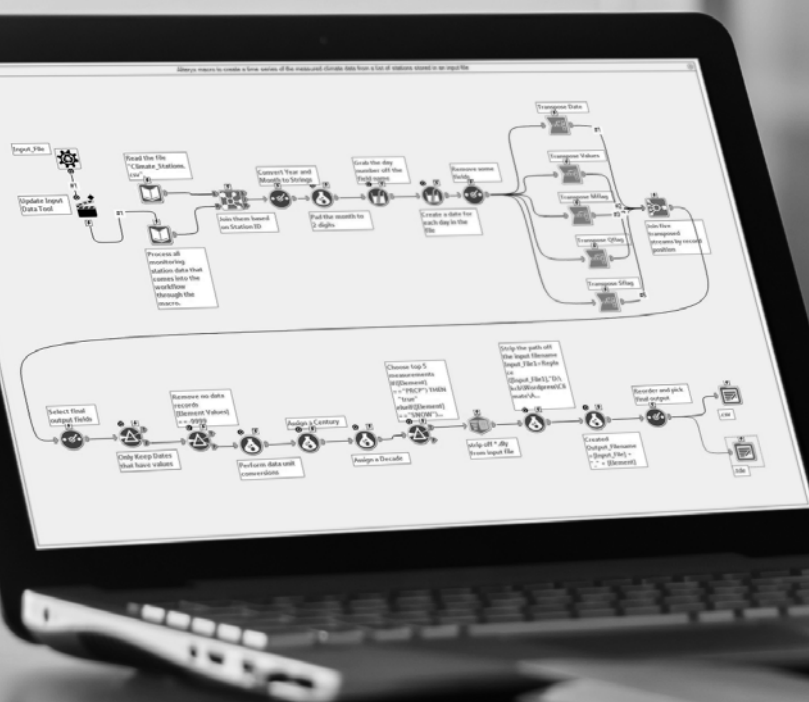# A PRACTITIONER'S GUIDE TO
ALTERYX

USEReady

Alteryx 11

A Practitioner's Guide to Alteryx®

A Practitioner's Guide to Alteryx®
Published by
USEReady
805 Third Avenue, 11th floor, New York, NY 10022
Manufactured in the United States of America

About USEReady

USEReady's mission is to help organizations succeed by fast tracking their business performance. We relentlessly strive towards value-driven innovation and digital transformation using advanced business intelligence, data management, and infrastructure management services. Our global delivery orientation is enriched with winning strategies, rich industrial and technology expertise and customer fanaticism. Our proven track record of success in versatile domains such as capital markets, insurance, healthcare, pharma, retail, media etc. is a testimony to our performance.

Visit www.useready.com or e-mail info@useready.com

About Alteryx, Inc.

Alteryx, Inc. is the leader in data blending and advanced analytics software. Alteryx Analytics provides analysts with an intuitive workflow for data blending and advanced analytics that leads to deeper insights in hours, not weeks, which is typical of traditional approaches. Analysts love the Alteryx Analytics Platform because they can deliver deeper insights by seamlessly blending internal, third party and cloud data; and then analyze it using spatial and predictive drag-and-drop tools. This is all done in a single workflow, with no programming required. More than 1,000 customers and thousands of data analysts worldwide rely on Alteryx daily.

Visit www.alteryx.com or call 1-888-836-4274.

Alteryx is a registered trademark of Alteryx, Inc.

Acknowledgements

Letter from the CEO

As we are publishing this third edition, Alteryx is a public company. Alteryx has gained a wide spread adoption across industries and geographies. We are glad that we partnered with this successful product early on. We see an increased adoption through the orders that we receive for this book. This edition incorporates many feedback and suggestions received from our readers.

As suggested by many of you, we have incorporated additional content by adding five new chapters and several hands-on exercises. Many newly certified Alteryx practitioners at USEReady have contributed to this book. Their hard work and dedication has resulted in this book that is now reaching 1000 pages in content. A total of 22 practitioners have revised this book with upgraded product features, exercises and new chapters.

The team has gone great heights to ensure the content is refined to the most recent version of Alteryx and exercises are useful to a fellow practitioner.

We are grateful to our readers of the previous editions and their constructive feedback has helped us improve this edition. We hope our efforts are well worth it and you are going to find this book useful.

Uday Hegde
Chief Executive Officer
USEReady

Foreword

As Vice President of product management at Alteryx, Inc. working closely with our product team, our customers and our partners, a training manual from USEReady is testament to the growing demand for easy-to-use data blending and advanced analytics solutions. USEReady knows the analytics industry and has created this book as both an independent guide and as a classroom aid to help its customers and others not only quickly learn Alteryx products, but more importantly grow in their ability to help gather deeper insights from their data.

The book has created a great launching point for beginners who want a manual, in addition to the real-world use cases, so you can easily learn how best to use Alteryx Designer. By the time, you finish working through this manual, someone who has never opened the Alteryx Designer before, will be able to create workflows, design reports, develop applications, and write macros to solve any of their data needs.

These are exciting times as the Alteryx community continues to grow globally, and we continue to witness an unprecedented demand for data analytics with actionable information. The Alteryx Designer and materials like this manual help the community of self-service data analysts make the most of their data.

We appreciate the partnership with companies like USEReady that know how to help these self-service analysts and reduce the time to insight with Alteryx.

Laura Sellers
Vice President, Product Management of Alteryx

Preface

Every day we are faced with options, questions, and choices. These decisions, as we all know, are much easier to make when we are well informed. Let's say that we want to eat. We literally have an entire world of possibilities, given the proper resources. However, practically, there are real limitations. Are we at home without transportation? Are we backpacking in the mountains? Are we in the middle of Times Square? Do we have food restrictions for health reasons? Do we have $5 or $5000? What are we in the mood for?

This task that we all solve day in and day out depends on a considerable amount of information that we know about our world, and often take for granted. This information is all based on data about our world.

What Is Data?

Data is stored information. It comes in various forms ranging from the number and types of items on our desk to the total mass of the universe to the contents of this book to the information in digital files and systems, which will be our focus.

What Does Data Do?

Data does nothing. It simply exists. It is what we do with data that is important. When we look at data we interpret it to create meaningful information, which gives us the ability to make better-informed decisions.

How Do We Consume Data?

Data can be consumed in many forms. We can look at all of the raw data and read every piece individually. We can use aggregation methods to create summary data so that we can

easily see high-level trends. We can visualize the data because a picture truly is worth a thousand words. Since we often do not want to look directly at the original data source and read each individual piece of data, we need to perform data preparation.

What Is Data Preparation?

Data Preparation is the process by which raw data is converted into a clean, usable source for later consumption.

The three core components of data preparation are data retrieval, data manipulation, and data export. In more traditional analytic terms, data preparation refers to the extract-transform-load process referred to as ETL. However, in order to ease communication, we are going to avoid these technical terms and discuss the aspects of the processes as follows:

Data retrieval refers to the process of going to a data source, asking for data, and returning the desired data.

Data manipulation refers to anything we decide to do to the data between the time we retrieve it and the time we export it.

Data export refers to what we do with data after we have extracted and manipulated it even if we haven't finished transforming it.

What is Data Manipulation?

Data retrieval and Data export are fairly straightforward; respectively, they can be likened to drawing water from a well and putting an ice cube in someone's drink. However, data manipulation is that tricky process of running the water through the pipes into our house, then filling the ice cube tray, then putting the tray in the freezer, and letting the water have enough time to freeze so that we have ice to consume. Going forward, we

will be using an allegory to a river to explain the entire process of the data preparation and specifically the data manipulation portion.

Data manipulation can come in many forms which typically fall into three buckets as follows:

Combination

One of the most common problems with data is that it comes from multiple sources. It is generally possible to perform the analysis separately, or through a significant amount of manual effort, but these methods often leave something to be desired or are too slow for effective use. In order to solve this problem, we will be designing data streams that come together.

If we think about data streams as actual rivers, original data locations can be thought of as glacial streams, smaller rivers, or lakes. Bringing data together is like the tributaries that bring these different water sources together to form a river. Along the course of this river, way we can perform calculations.

Calculation

If the data is to be used, it is generally advisable to have as much data pre-calculated as possible. One reason for this is that it allows an organization to create a standardized formula for everyone's use. Another is that when we can run calculations before data is provided to a front-end user or system, the consumer will experience a much faster process.

If the data is to be used in a report, then the calculations are often fundamental aspects of that report.

Returning to the river analogy, we can think of calculations as hydroelectric dams along a river, we are using the resources that already exist in order to generate something new.

We may change the landscape because we are changing the flow of water, and we are also slowing down the river (introducing calculations will slow down the data preparation process).

We also have the ability to transform the data stream into a more useable format.

Transformation

It is often the case that data is not in the format that we need. We may have been given access to a database that has data stored in a very machine readable format, and we need to pivot the table to make it human readable, or we may have been working with an Excel file which has data extremely normalized that makes it hard to use in a front-end system. Either way, we need to transform the structure of the data so that it can be effectively consumed. In thinking about the river, we can imagine this as the process of cutting a channel into the riverbed so that the river is deep enough to move barges up and down. In doing this, we are fundamentally changing the structure of the river in order to make it more useable.

To the reader

In the following chapters, we are going to cover many topics, but the format of the chapters will all be the same.

You will assume the role of a new consultant at a company that works with Alteryx. We introduce a business scenario, discuss the tools that we will use to solve the problem(s), walk through the initial problem(s), and then provide you a self-guided exercise. We conclude this book with a capstone assignment in *NYC.*

The exercises will use data that can be download from http://www.useready.com/resources/a-practitioners-guide-to-alteryx by following the instructions on the website to unpackage the file.

Additional data will also be needed to install the *US 2010 Census SF1* and *USGS North America Map* packages from http://downloads.alteryx.com/data.html which we will start using in *Cultural Musings.* We will also be using the Solocast Datasets in the *Statistics in Alteryx* section.

Let us know what you think by emailing us at AlteryxBook@USEReady.com and we will try to incorporate reader requests going forward.

If interested in Alteryx training sessions or Alteryx consulting, visit http://www.useready.com/.

Best of Luck,

USEReady

Table of Contents

CHAPTER 1
An Introduction to Alteryx

1.1   What Is Alteryx Analytics Platform?

The Alteryx Analytics Platform from Alteryx Inc. consists of two products: Alteryx Designer and Alteryx Server.

Alteryx Designer allows us to build a data stream (or module) and run it locally. In addition to a base package, Alteryx offers datasets and tool packages to expand the product to enhance computing capabilities. These packages permit additional features like drive-time analysis when TomTom data is available.

Alteryx Server has two main functions. The first allows users who have Alteryx Designer to publish applications for consumption. The second allows scheduled workflows to run without external intervention.

The *Alteryx Analytics Platform* has solved two major issues affecting the data preparation process: time to run the process and high specialization of labor. Typically, the preparation process is highly time-intensive because it is largely manual. While automation solves the time required to run, it still requires personnel with skills in computer programming. This means we need additional resources allocated to every project, eventually raising the project costs and often preventing projects from being taken on. Alteryx, Inc. has elegantly addressed both of these problems. It has designed an easy-to-use drag-and-drop interface where all we need to do is give each tool the appropriate settings. This simple concept has allowed business users to develop workflows that meet their exact needs much faster than has ever been possible because it provides business people with the tools to design or modify a fully or semi-automated data preparation.

1.2   Alteryx Gallery

The *Alteryx Gallery* is a public version of *Alteryx Server* where the community can upload interesting data streams, share ideas, and get inspired. The *Alteryx Gallery* can be accessed at gallery.alteryx.com.

1.3   The Alteryx Interface

By the end of the chapter, the reader should be able to understand all aspects of the Alteryx Designer Interface and how to work with the product.
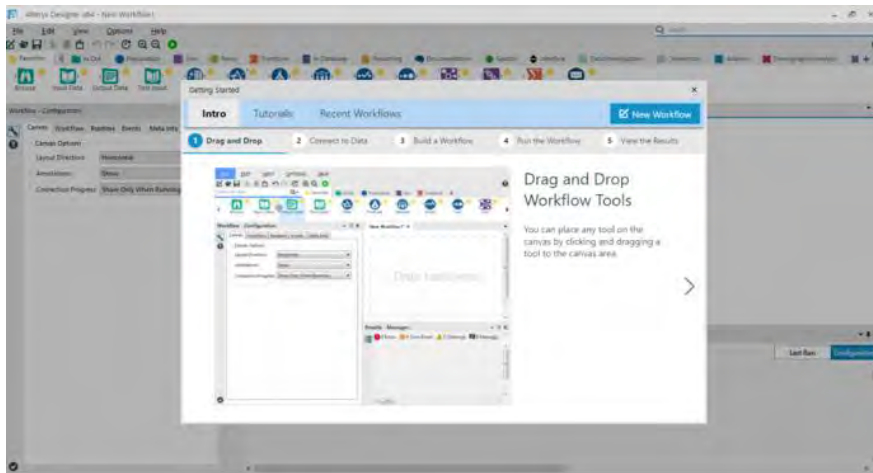
Figure 1-1 - Alteryx Intro
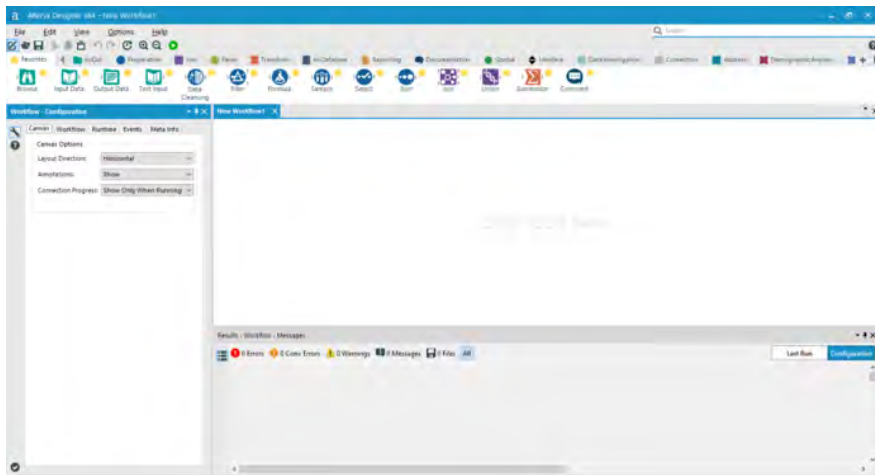
Figure 1-2 - Workflow Configuration

Above are the two things we may see when we open Alteryx Designer. The first image is the default for when the program is first opened. Alteryx Designer asks us what we want to do with the session. These options, fairly clearly, are to open up the basic tutorials, open an existing module, or open a new module. The second image is what comes up when we have previously selected the option *Don't show this again and always open a blank canvas*. When using this option, Alteryx Designer always opens a new module. (For those who see the first image, please select "New Workflow.")
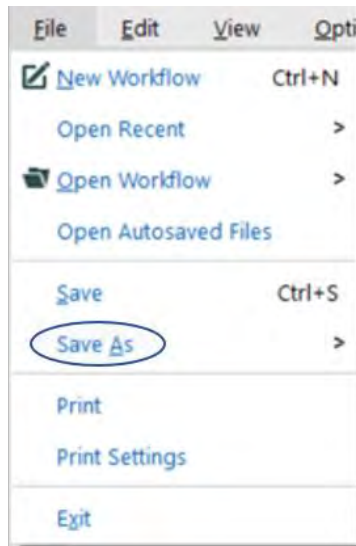
Figure 1-3 - File Dropdown

The first thing we will do is save our module. Open the file drop-down at the top left of the screen, and select the *Save As* option as indicated above.
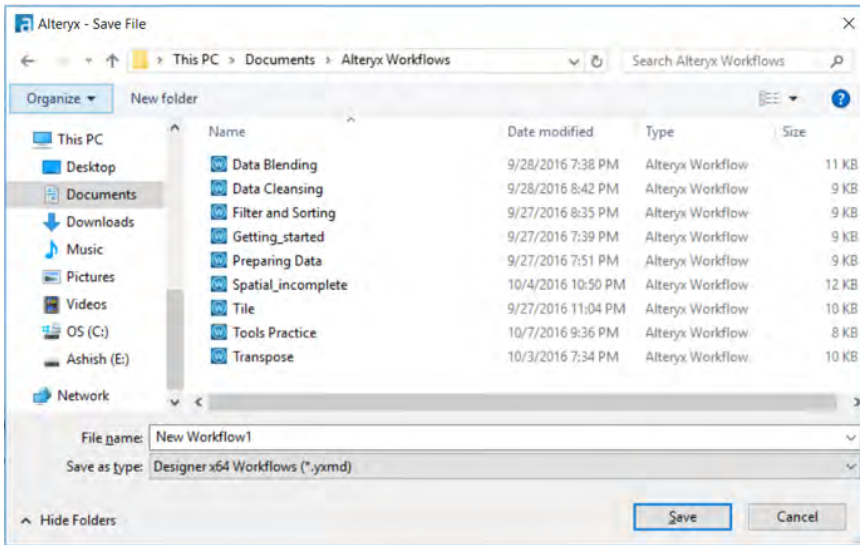
Figure 1-4 - Save As

We can save our data stream as one of three file types. At a high level, we have the ability to save our data streams as:

- .yxmd

  Files saved in this format are data streams in a standard *Alteryx Workflow.*

- .yxwz

  Files saved in this format are data streams that have been designed as *Analytic Applications*, which means they have user inputs and can be run on *Alteryx Server*.

- .yxmc

  Files saved in this format are data streams that have been designed as *Macros* that allow the re-use of the data stream.

Please save this workflow as *The Basics.yxmd*

1.4   The Interface
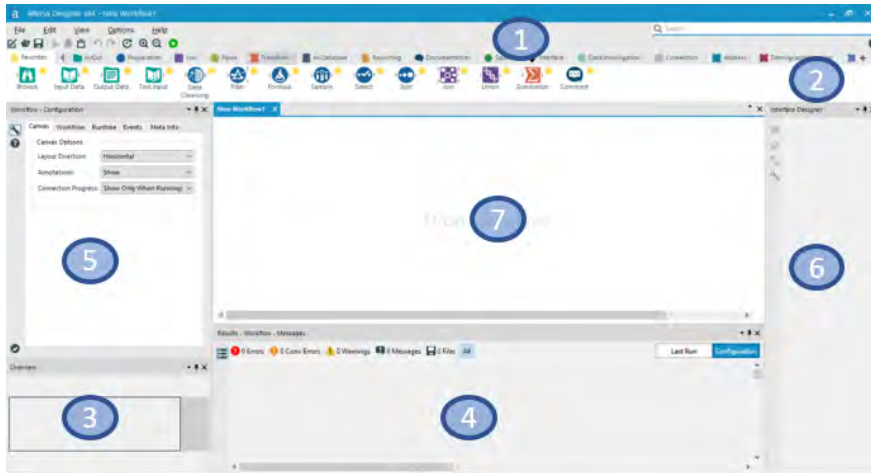
Figure 1-5 – Alteryx Designer Interface

This is one potential view of the *Alteryx Designer Interface* if every *View* option is on. The numbers represent each of the seven primary aspects of the interface. One through six can be disabled and will be discussed under the corresponding sections in *View*. The seventh is the canvas, where we build workflows.

1.5   File Menu
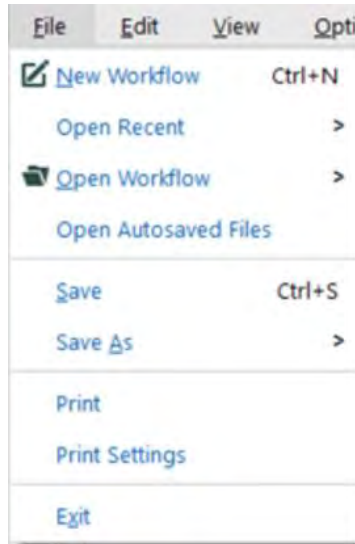
Figure 1-6 – File Menu

The *File Menu* functions in the same way as most programs, with a few specialized options:

New Workflow: Opens a blank canvas to work in. (*Ctrl+N* means that there is a *Hot Key* that allows us to use this action without opening the drop-down. Pressing the *Ctrl* and *N* buttons at the same time does the same thing as selecting *New Workflow*.) (Hot Key: Ctrl+N)

Open Recent: Opens a list of the most recently opened workflows.

Open Workflow: Opens a file browser that allows us to open an existing *Workflow*, *Analytic App*, or *Module* in a tab of the canvas. (Hot Key: Ctrl+O)

Open Autosaved Files: Unbeknown to many, by default Alteryx saves your open workflows every 10 minutes, keeps the last three iterations of them, and keeps them around for 30 days since the last save. To edit the Autosave settings, go to 'Advanced' tab under Options > User Settings > Edit User Settings.

Save <File Name>: Saves the workflow that is in the active canvas. (We will see the name of the current workflow instead of *<File Name>)* (Hot Key: Ctrl+S)

Print: Prints a copy of the workflow.

Print Setup: Opens a window that allows us to configure how the workflow will print if we choose to print out a copy.

Exit: Closes Alteryx Designer entirely. This differs from *Close* because *Close* will close the active canvas but *Exit* will close every open data stream.

1.6   Edit Menu

Figure 1-7 – Edit Menu

Undo: Allows us to undo the steps just performed, in reverse order. (Hot Key: Ctrl+Z)

Redo: Allows us to restore a step from what was undone, as long as we have not changed anything else. (Hot Key: Ctrl+Y)
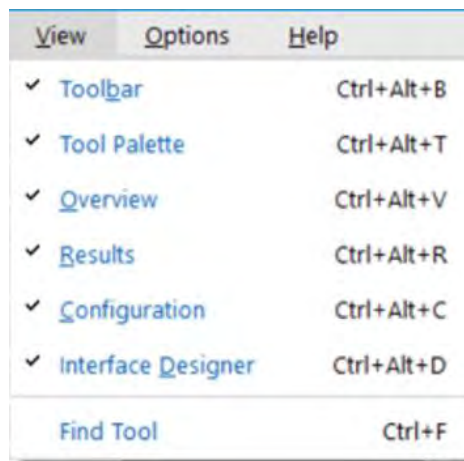
Cut: Allows us to remove the selected items on the canvas while keeping a copy on the clipboard. (Hot Key: Ctrl+X)

Copy: Allows us to copy the selected items on the canvas to the clipboard. (Hot Key: Ctrl+C)

Paste: Allows us to take the most recent thing copied or cut into the clipboard and put it on the canvas. (Hot Key: Ctrl+V)

Delete: Removes the selected items on the canvas. (Hot Key: Del or Delete)

1.7   View Menu

Figure 1-8 – View Menu

Toolbar: Turns the toolbar on and off. When it is checked, the buttons on the toolbar are visible (1 in Figure 1-5). (Hot Key: Ctrl+Alt+B)

Tool Palette: Turns the tool palette on and off. When checked, we will see the two rows of drag-and-drop tools (2 in Figure 1-5). (Hot Key: Ctrl+Alt+T)

Overview: Turns the overview window on and off. When checked, we can see a high-level overview of our canvas (3 in Figure 1-5). (Hot Key: Ctrl+Alt+V)

Results: Displays the Results window docked in its last visible position. (Hot Key: Ctrl+Alt+R)

Configuration: Displays the Configuration window docked in its last visible position. (Hot Key: Ctrl+Alt+C)

Interface Designer: Turns the interface designer window on and off. When it is checked, we will see a window that allows us to modify the user interface (6 in Figure 1-5). (Hot Key: Ctrl+Alt+D)

Find Tool: Opens a pop-up window that allows us to quickly find particular tools on the canvas. (Hot Key: Ctrl+F)

1.8   Options Menu
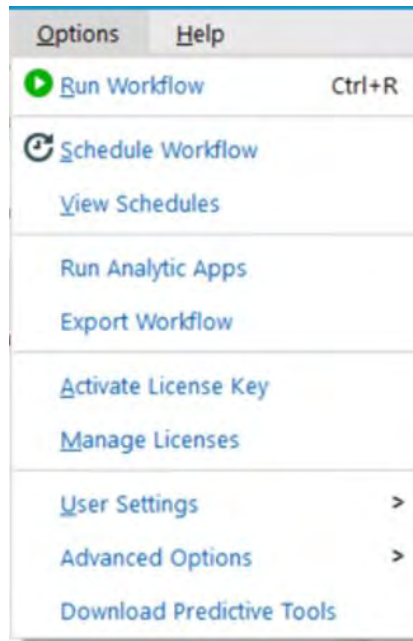
Figure 1-9 – Options Menu

Run Workflow: Runs the workflow. Ctrl+R will also run the workflow. The icon changes to Cancel Workflow while the workflow is running.

Schedule Workflow: Allows us to set up a schedule to run our workflow as long as we have Alteryx Server.

View Schedules: Allows us to look at all of the scheduled jobs that we have access to on Alteryx Server.

Run Analytic Apps: Allows us to run Alteryx Analytical Apps that we have created.

Export Workflow: Allows us to export the workflow along with the packages that have been created.

Activate License Key: Opens a window that asks for an activation key so we can use Alteryx Desktop.

Manage Licenses: Opens a window that allows us to look at historical license keys and see what we currently have available.

User Settings: Opens a window that allows us to customize the way the canvas looks and the way some of the default settings behave.

Advanced Options: Allows us to manage aliases, workflow dependencies, and encrypt workflows.

Download Predictive Tools: Allows us to download tools for the purpose of predictive analytics from the Alteryx repository.

1.9   Help Menu
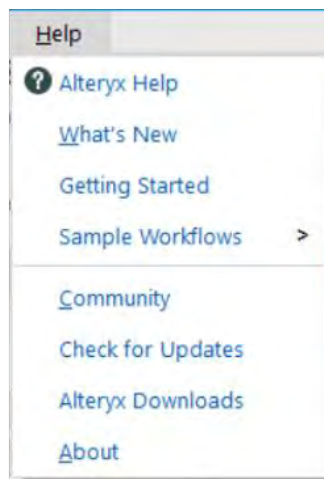
Figure 1-10 – Help Menu

Alteryx Help: Opens the Alteryx Help home page.

What's New: Opens the Alteryx Help release notes for the version of Alteryx that we are currently using.

Getting Started: Opens the Alteryx Getting Started page that is loaded in the beginning.

Sample Workflows: Allows us to access the Sample Workflows that are shipped with Alteryx.

Community: Opens the Alteryx Community web page.

Check for Updates: Checks our current version of Alteryx Designer and verifies if it is the most recent version.

Alteryx Downloads: Opens the Alteryx Download page where we can find additionally downloadable content. One example is the U.S. Census data.

About: Opens a window that tells us version, publisher, and serial number information about Alteryx Designer.

1.10   Samples
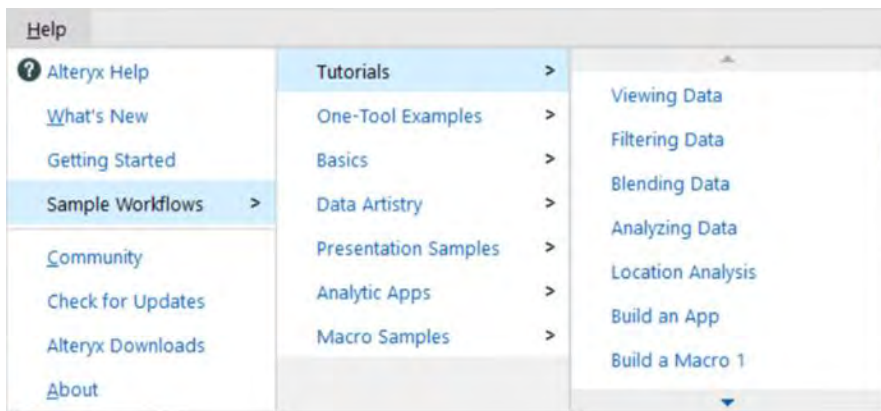
Figure 1-11 – Samples

Under the *Help Menu,* there is the option to open Sample Workflows. Depending on what we have downloaded from Alteryx, we will see different samples.

These samples are good starting points to get used to Alteryx and to learn how to use tools that we do not know.

Walking through the samples is beyond the scope of this book, as all the samples are well documented and show how to build the workflow step by step. However, below is a picture of what the *Viewing Data* sample looks like before you begin to work with it.
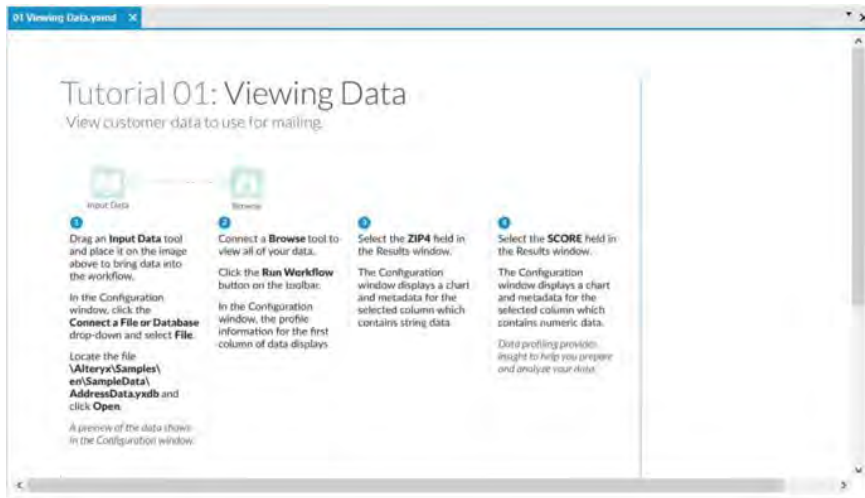
Figure 1-12 – Sample Tutorial

1.11   User Settings

Under the *Options menu,* there is an option called *User Settings*. This sub-menu has four options.
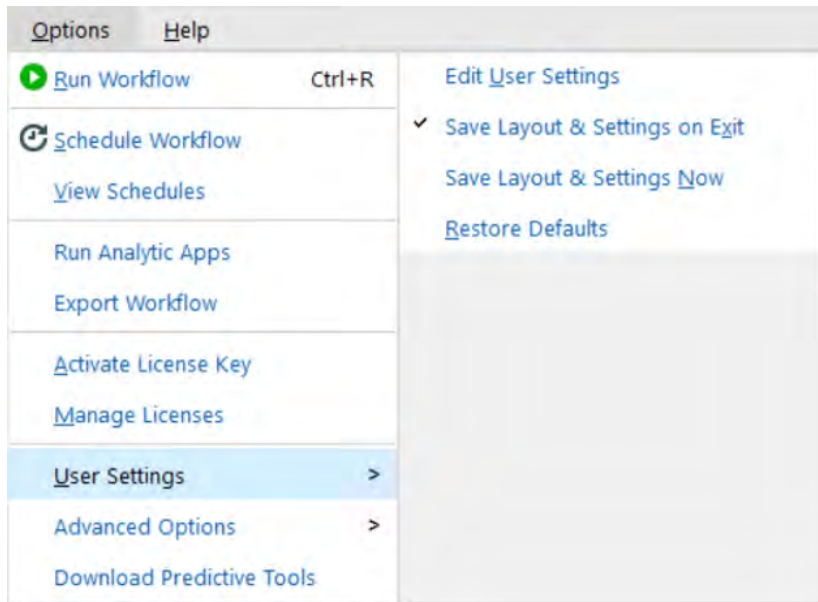
Figure 1-13 – User Settings

Edit User Settings: Opens a pop-up window for the user to edit the settings affecting the way Alteryx Designer looks and behaves.

Save Layout & Settings on Exit: Allows Alteryx to copy the settings we had when Alteryx was last closed as the default when we next open the program. Unchecking the option can turn off this functionality.

Save Layout & Settings Now: Allows us to tell Alteryx to use the settings we currently have as the default behavior for the program. This is most effectively used when we are not using *Save Layout & Settings on Exit*.

Restore Defaults: Allows us to revert Alteryx Designer to the default settings.

1.12   Toolbar

Figure 1-14 – Toolbar

The *Toolbar* has buttons we can use to do the following tasks. It is the same as if we were to navigate to the parallel options in the menus described above. The buttons simply make it easier to perform the most common actions.

Allows us to create a new workflow. (Same as File > New Workflow) (Hot Key: Ctrl+N)

Allows us to open a workflow. (Same as File > Open Workflow) (Hot Key: Ctrl+O)

Allows us to save our current workflow. (Same as File > Save <File Name>) (Hot Key: Ctrl+S)

Allows us to cut our current selection. It is grayed out and inaccessible when we have nothing selected and dark when we do. (Same as Edit > Cut) (Hot Key: Ctrl+X)

Allows us to copy our current selection. It is grayed out and inaccessible when we have nothing selected and dark when we do. (Same as Edit > Copy) (Hot Key: Ctrl+C)

Allows us to paste what we have on the clipboard. If we do not have something on the clipboard that can be pasted into Alteryx, we will get a pop-up that says *Nothing to paste.* (Same as Edit > Paste) (Hot Key: Ctrl+V)

Allows us to undo what we just did. It is grayed out when we have nothing that we can undo. (Same as Edit > Undo) (Hot Key: Ctrl+Z)

Allows us to redo what we just undid. It is grayed out when we have nothing that we can redo. (Same as Edit > Redo) (Hot Key: Ctrl+Y)

Allows us to schedule the workflow on Alteryx Server. (Same as Tools > Schedule Workflow)

Allows us to zoom in on our workflow. (Same as View > Zoom In)

Allows us to zoom out on our workflow. (Same as View > Zoom Out)

Allows us to run the workflow. (Hot Key: Ctrl+R)

Allows us to cancel the workflows while running. (Hot Key: Ctrl+R)

1.13   Tool Palette

Figure 1-15 – Tool Palette

The tool palette is one of the most important aspects of the Alteryx Designer interface. It is the easiest way to bring the tools onto our canvas. In the later section below called *Canvas,* we will discuss the alternative method.

The three main elements of the tool palette are:

- *Search*: Located in the top-right corner of the tool palette, this element allows us to type something into the bar, and

every tool associated with that word is displayed in the Tool Selection.

- *Add/Remove Tools* ⊞ : Located to the far right, below *Search* and next to the *Tool Categories*, this element allows us to access/add/remove all the tools grouped into categories that Alteryx has defined. We can add to this list by developing macros, which we will discuss toward the end of the book.
- *Tool Selection*: This comes as a drop-down when we type in the search box and shows all tools meeting either the criteria of the *Search* or *Tool Category* that we have selected.

There is one special category in the *Tool Categories* called *Favorites*. It is special because it does not have tools of its own. Instead, we can add and remove tools from this list by clicking the star at the top-right corner of each tool. When it is yellow, it means that it will show up in the *Favorites Category*. When we cannot see the star unless we point our mouse at the tool, and it shows up gray, that means that it is not in the *Favorites Category*.
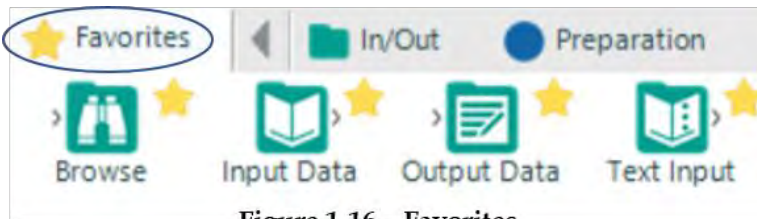
Figure 1-16 – Favorites

The arrows on either side of the *Tool Categories* offer the ability to scroll across to see each of the categories. We see that, by default, *Favorites* doesn't follow this behavior. That is because it is pinned outside the scroll arrows. We can do this with any of the categories by right-clicking on them and selecting *Pin <Category Name>*. See the example below.

Right click on In/Out:
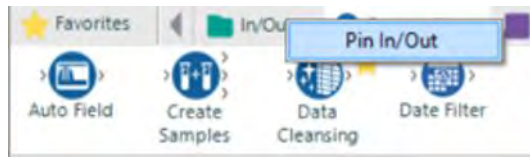
Figure 1-17 – Pin In/Out

Select Pin In/Out:
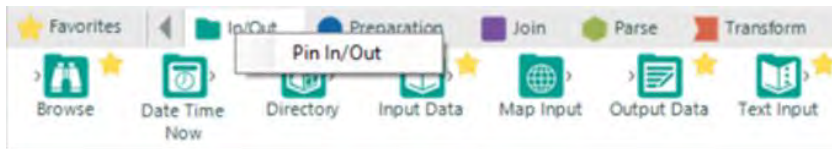
Figure 1-18 – Selecting Pin In/Out

In/Out is pinned:
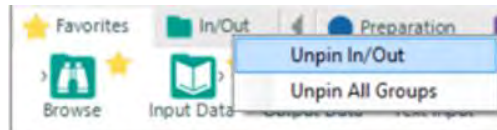
Figure 1-19 – Unpin In/Out

Select Unpin In/Out:
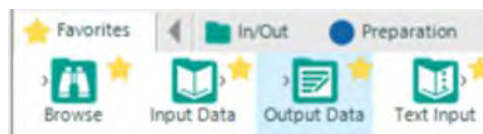
Figure 1-20 – Tool Categories

We can see that certain icons show up on the *Tool Category* names. That is because each of the categories has a *Default Tool*. If we were to click on the *Tool Category* and drag it to the canvas, the tool in the icon is what we bring onto the canvas.
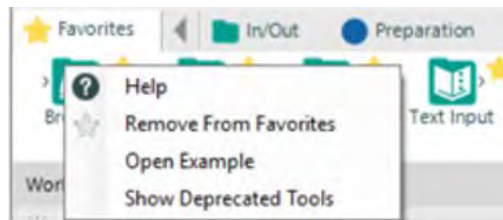
Figure 1-21 – Tool Categories

We can see in Figure 1-21 an option called *Show Deprecated Tools*. Selecting this option shows tools that belong to outdated versions of Alteryx. If we need a former functionality, we can access it here.

Figure 1-22 – Additional Tool Categories

All of these categories may not be available to all users because Alteryx has many additional packages that we can download in addition to the basic package. One example is if we had not chosen to download the Predictive package when we downloaded Alteryx Designer, we would not see the *Predictive Category.*

We will only cover the basic package in this book, plus some downloaded content.

1.14   Overview

Below is a picture of the canvas and overview window with a simple data stream that is too wide to see what is happening in the view of the canvas. The right side of the image is the *Overview* window, and it shows the view of the canvas in relation to the entire workflow.
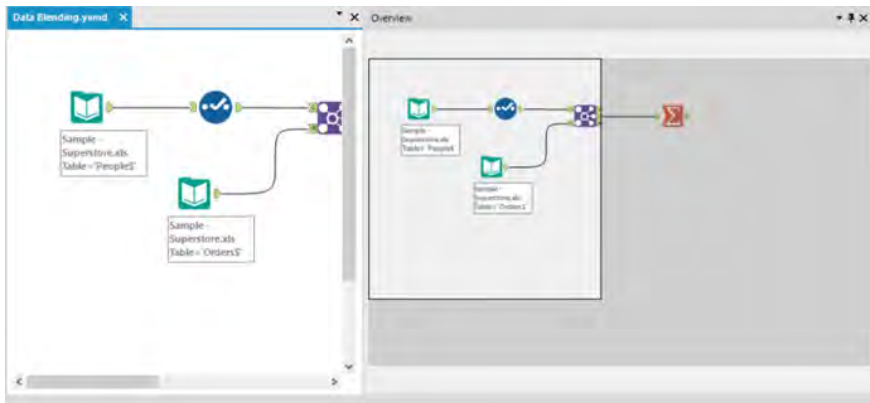
Figure 1-23 – Overview Window

In addition to showing where we are, the overview window makes it easy to find the part of our program that we are interested in because we can move the view of the canvas by clicking and dragging the white box in the overview window.

1.15   Results Window

Every time we run a workflow, we will get a series of returns that lets us know things about the run. One example of this is below.
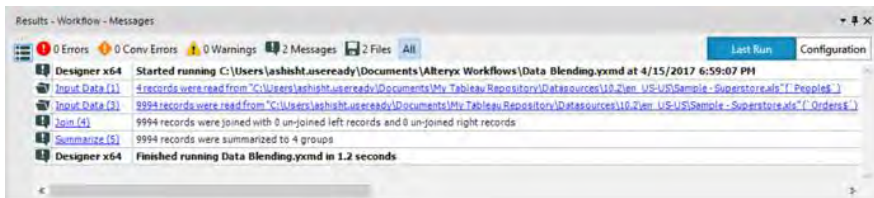
Figure 1-24 – Results Window

In this run, we can see that we had 0 Errors, 0 Conversion Errors, 0 Warnings, 2 Messages, and 2 Files. The counts here reflect the eight notes we see about the run in the text portion of the Output window. We have six notes instead of the four we would expect from the counts of the messages. This is because the first note is that the workflow started to run, and the last note

is that the module had finished the run. Even if we have no issues or messages during the run, we will still have these two notes because they are always written. The last message changes depending on whether or not we have any errors during our execution. The output window is important in diagnosing issues with the data stream.

- *Errors*: These are critical failures that will terminate the data stream they are in.
- *Conversion (Conv) Errors*: These are potentially problematic issues that stem from a data mismatch when converted from one format to another.
- *Warnings*: These are things we should be aware of but could be expected, so the results are printed in the Output for us to know about.
- *Messages*: These are text in the program that are designed to print when certain stages of the data stream have been finished or certain conditions were met.
- *Files*: These are links that will allow us to open any file created during the running of the workflow

1.16   Properties Window

The properties window is where we will spend the vast majority of our time. Every tool has a different properties window, and we will discuss each of them as we discuss the individual tools. Here, we will discuss the properties window of the workflow and connections.

There is a pane on the left side of every properties window that has a universal set of symbols. We will discuss each of these as they first appear in this book. A list of all of these symbols is in Appendix E.

Workflow configuration
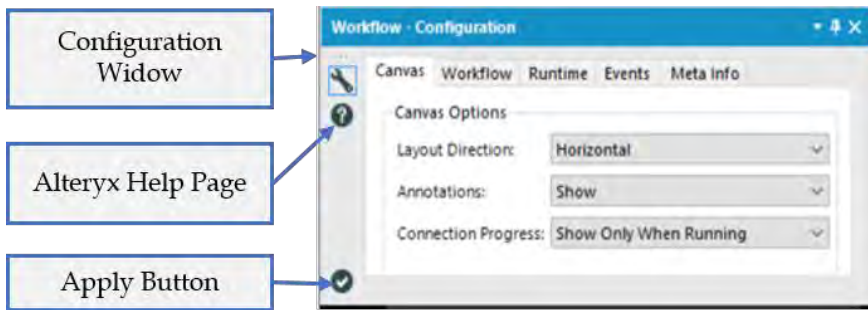
Figure 1-25 – Workflow Configuration – Canvas Tab

Notice the button that looks like a wrench at the top-left of each of these properties windows. Selecting the wrench shows the *configuration window* so that we can modify the settings of the tool. As we discuss other tools, the importance here will become apparent.

A click on ❓ symbol opens the Workflow Alteryx Help page.

A ✅ symbol at the bottom-left corner is an apply button. We can click this to apply the changes we have made. We do not need to click this button; we can just click somewhere on the canvas.

The first tab of the *Workflow configuration* window is Canvas. We can change three things in this window.

- *Layout Direction* is a drop-down list that allows us to rotate our data stream from a horizontal layout, which is the default, to a vertical layout, which is the layout for older versions of Alteryx. We must set either *Horizontal* or *Vertical* and cannot use a combination of the two.
- *Annotations* is a drop-down list that allows us to change what the text boxes under Tools say. When this option is set to *Hide,* the text does not show. When this option is set to *Show*, the default text is shown for each tool. When this

option is set to *Show w/ Tool Names,* the annotations tell us what the tool names are as well as the default text.

- *Connection Progress* is a drop-down list that allows us to change when we see the progress labels on each of the tools. By default, this is set to *Show Only When Running,* but we can also set it to *Hide* and *Show,* which will never or always show the last run's progress respectively.
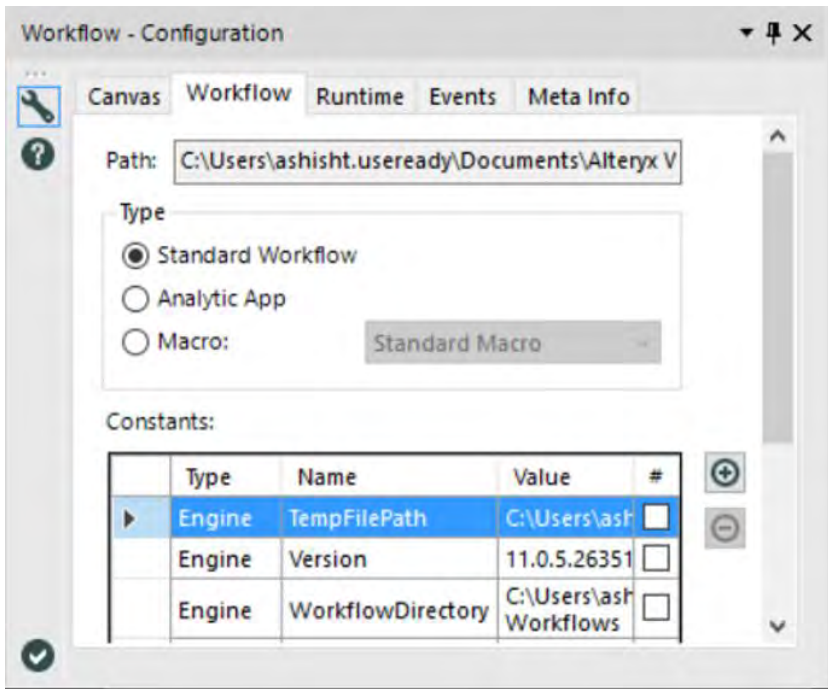
Figure 1-26 – Workflow Configuration – Workflow Tab

The next tab is *Workflow.*

- The *Path* (file location) shows up in this box when the file has been saved.
- The *Type* allows us to choose which format the file should be. This is also where we select the type of *Macro* we want

to create. (This will be discussed in *The Mermaid Coffee Company* chapter.)

- The *Constants* section allows us to define variables that we can use throughout our Alteryx Module.
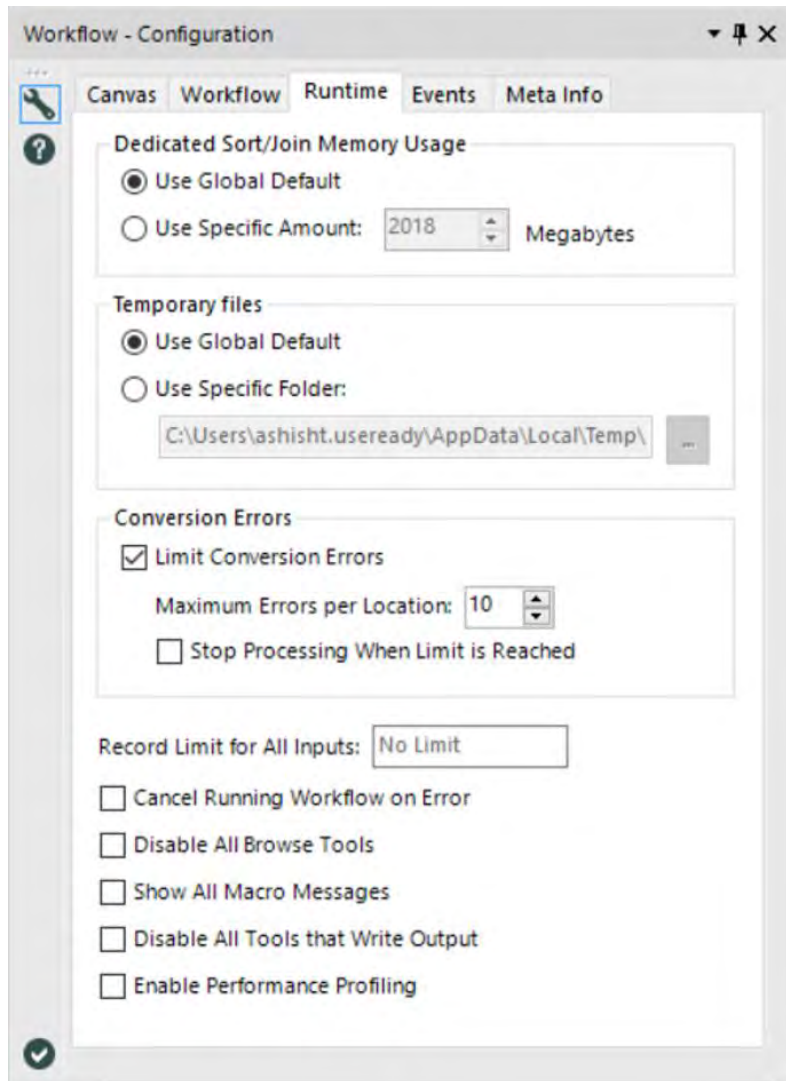
Figure 1-27 – Workflow Configuration – Runtime Tab

The *Runtime* tab allows us to configure settings that impact the way that Alteryx runs.

- The *Dedicated Sort/Join Memory Usage* allows us to set the maximum amount of memory that Alteryx is allowed to use for the *Sort* and *Join* functions, which can be highly memory-intensive.
- The *Temporary Files* setting allows us to define the folder that our temporary files are written to.
- The *Conversion Errors* setting allows us to change the way the workflow behaves if it experiences conversion errors, including limiting them from showing up for each tool and stopping the workflow if they occur.
- The *Record for All Inputs* setting allows us to set a maximum number of records read from any of our input files.
- The *Cancel Running Workflow on Error* is useful if we are trying to diagnose a specific issue in a workflow.
- The *Disable All Browse Tools* should be set after we finish testing and the process is in production. We want to do this because browse tools are slow to create and unnecessary unless testing. It is better to use this than to delete all of the tools because it will make modifying the workflow easier later.
- *Show All Macro Messages* allows us to see the messages we are getting in the macro tools so that when we are running a workflow, we can see everything that may be an issue.
- *Disable All Tools that Write Output* is useful for testing because we can turn off only the outputs so we are not rewriting the data every time we run the workflow.
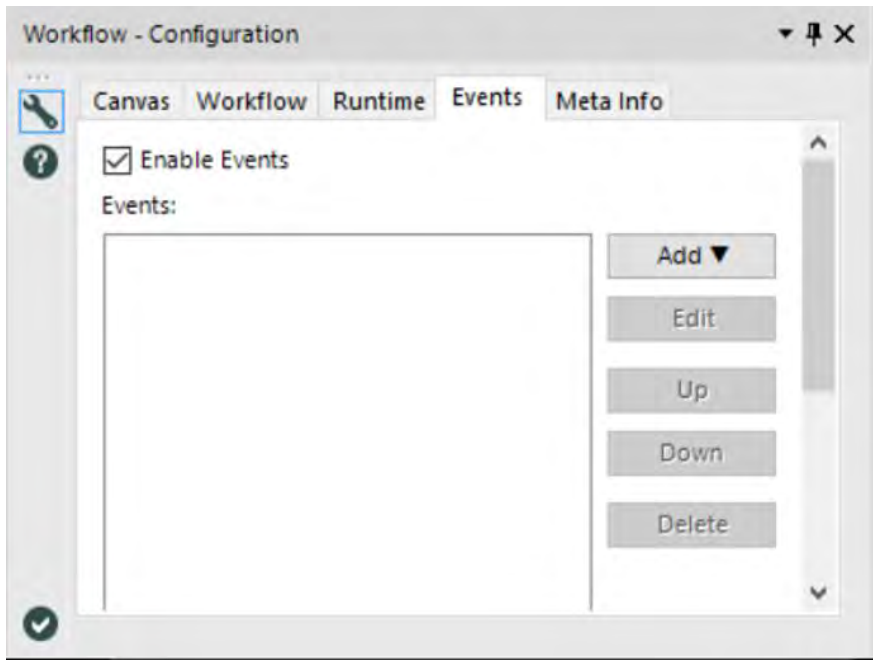
Figure 1-28 – Workflow Configuration – Events Tab

The *Events* tab allows us to define events that will allow secondary actions to take place.

When we add an event, we can choose to define a command line code or send an email when a condition is met. This is useful for security or maintenance because we can set up our workflows to let us know anytime the workflow ran or anytime it ran with issues.

The *Meta info* tab allows us to define meta information about our workflow. This is important when we are publishing a workflow to Alteryx Server or the public Alteryx Gallery.
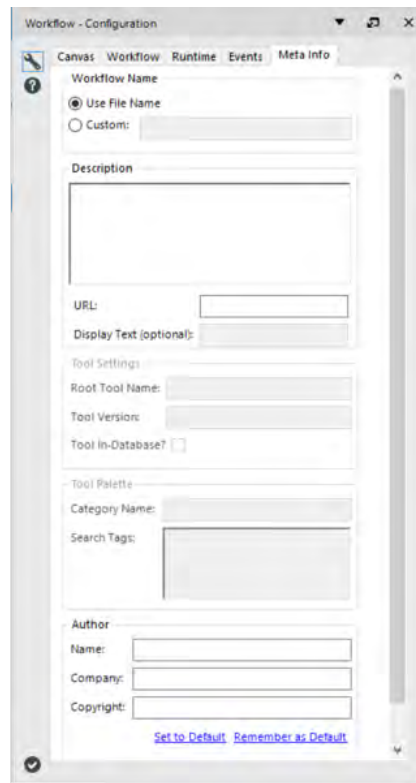
Figure 1-29 – Workflow Configuration – Meta Info Tab

- *Workflow Name* allows us to customize what we want the workflow to be called (independent of the file name) so that versioned workflows can have the same name, as far as the end user is concerned.
- The *Description* is a place for us to describe what the workflow does.
- The *URL and Display Test* fields allow us to identify where the module is published.
- The *Author* section allows us to publish information about us and our company along with the workbook.

Connection configuration

When using the default settings, connections are the curved black lines that connect one tool to the next.
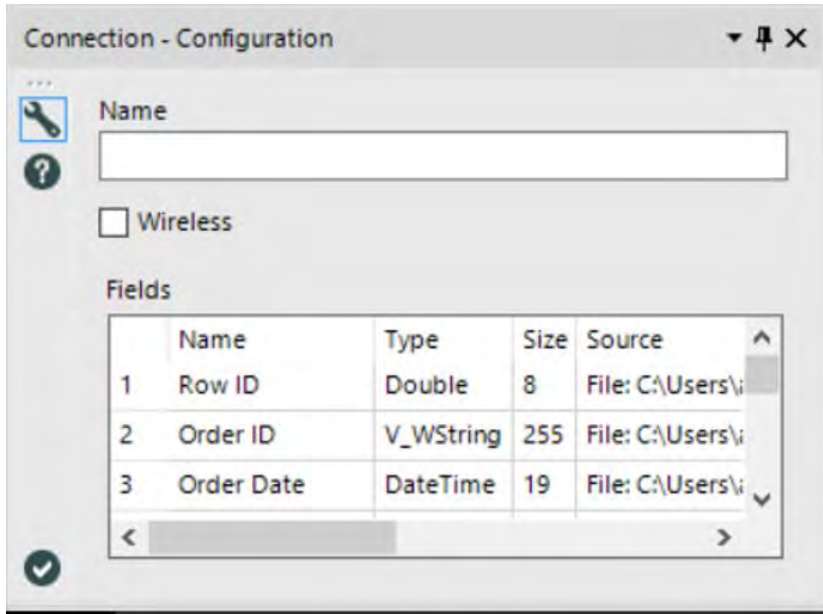
Figure 1-30 – Connection Configuration

The *Connection configuration window* has three components.

- *The name* allows us to rename our data connection. This is most useful when we have multiple data connections going into the same tool connector.
- The *Wireless* checkbox allows us to make the connection wireless (invisible) unless one of the tools it connects is selected.
- The *Fields* box allows us to see some of the metadata about each field in the data stream in this connection.

1.17   Interface Designer

The Interface Designer is a window used in developing apps and macros. This is where we get to design the user interface the end user will see.
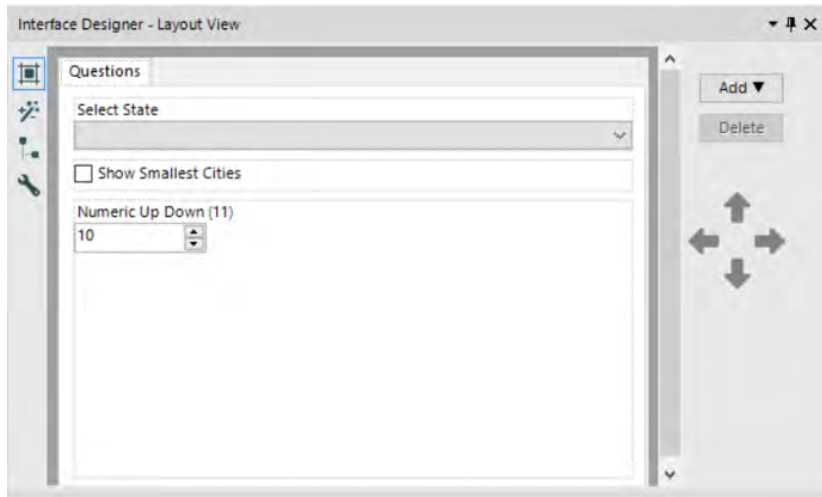
Figure 1-31 – Interface Designer – Layout View

The default tab view of the Interface Designer window is of the Layout View, which allows us to design the user interface in a graphical format.

Figure 1-32 – Interface Designer Settings

The Interface Designer, like the Properties window, has a sidebar that tells you what you are looking at. From top to bottom, the icons are:

- *Layout View*, where we design the interface graphically.
- *Test View*, where we see what the end user would see.
- *Tree View*, where we look at the structure of our questions in a tree structure
- *Properties*, where we set properties associated with the interface (this is independent of the Properties window itself).
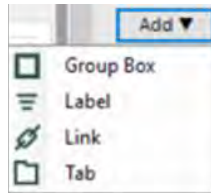
Figure 1-33 – Interface Designer Dropdown

In the *Layout View,* we design the interface and thus have a significant amount of flexibility in what we can do.

The first and most intuitive thing to do is rearrange questions in the order they need to be asked from the end user. We can do this by clicking the up and down arrows on the right-hand side. We may notice that while we are moving questions, they sometimes move within a box that frames the question we were trying to pass. This is because we can make one question determine if the other is also asked. By leaving a question inside another, we make that question dependent.

The next thing we can do is select the Add drop-down menu and see that we can add any of four things. Those things are:

- *Group Box*: a text element that we can put questions in.

- *Label*: a text element that we cannot put questions in.
- *Link*: a hyperlink in the interface.
- *Tab*: a tab in the interface.

When we are working with the *Interface Designer*, we will be opening the Properties windows associated with whichever tool we have selected at the time. The following are the properties windows of the four objects described above.
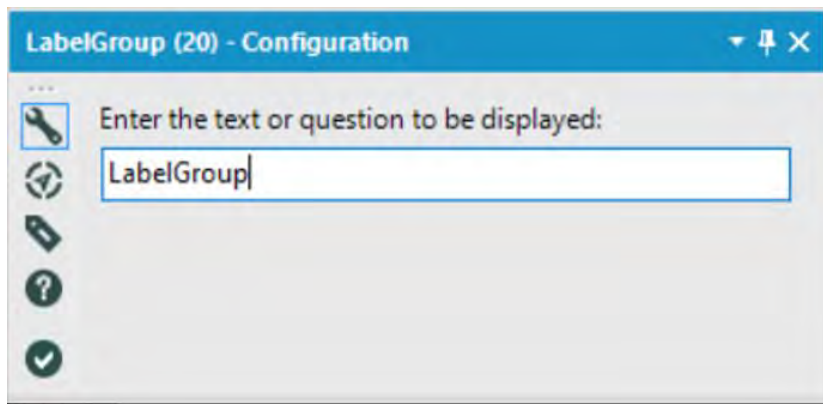
Figure 1-34 – Label Group Configuration
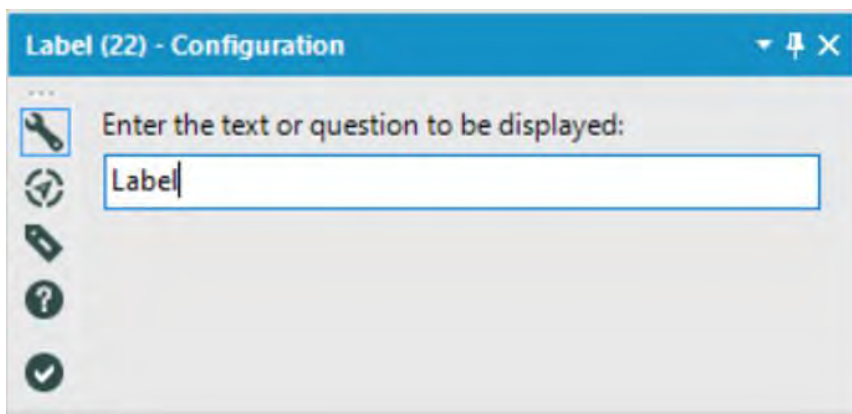
Figure 1-35 – Label Configuration
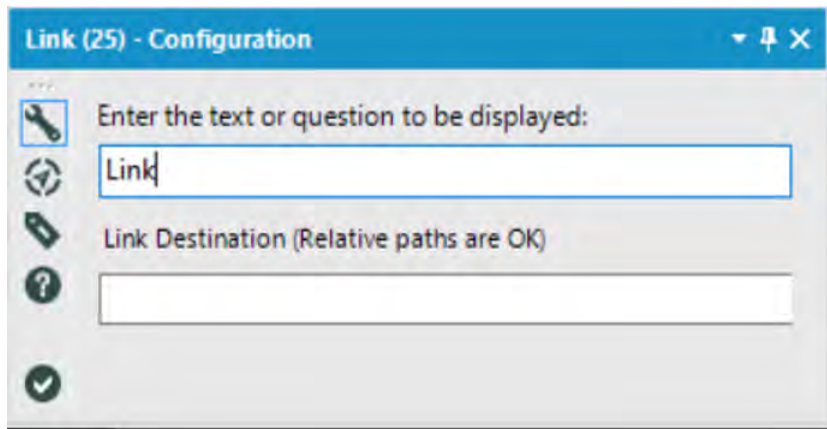
Figure 1-36 – Link Configuration
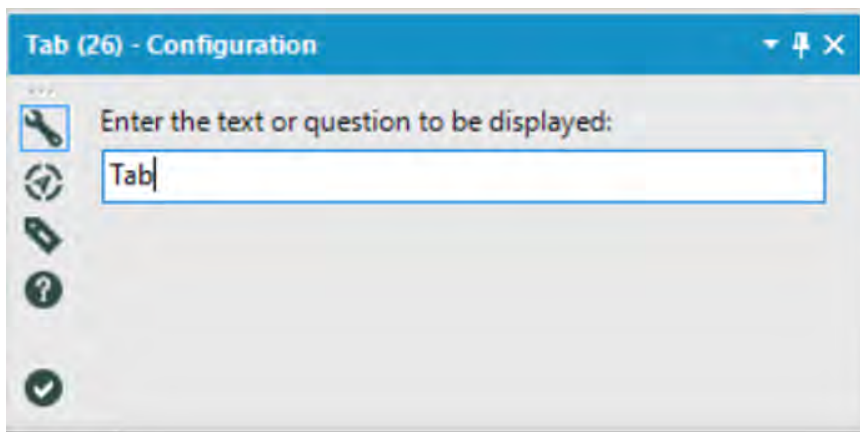
Figure 1-37 – Tab Configuration

As we can see, the four windows are similar. All four ask what we want the element in the interface to be called, and *Link* also asks us what link address we want to add to the interface. In addition to these questions, we see for the first time that we have three new icons in the sidebar of the Properties window.

Figure 1-38 – Sidebar Icons

- *Navigation*: Allows us to move directly to the incoming or outgoing tools.
- *Annotation*: Allows us to change how this tool is named and how its annotation behaves.

Navigation and *annotation* are part of every tool properties window, and they always behave the same.

Below are examples of a *Check Box* Navigation and Annotation window when connected to a tool.
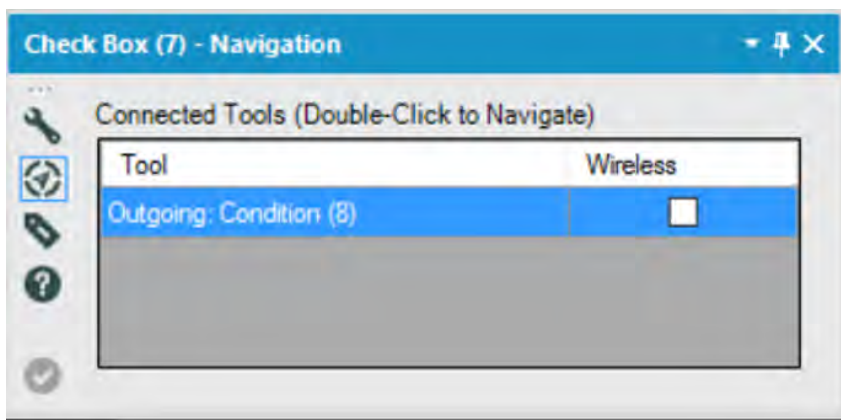
Figure 1-39 – Checkbox Navigation

We can see that *Navigation* lists the tool that this *Check Box* is connected to, Action (20), and tells us that it is an outgoing connection. That means that the Action is downstream of the Check Box. If the Wireless box were checked, the wire would be invisible. We will discuss both tools, as well as wireless connections, in chapters *Applications Wanted* and *Meta-morphosis*.
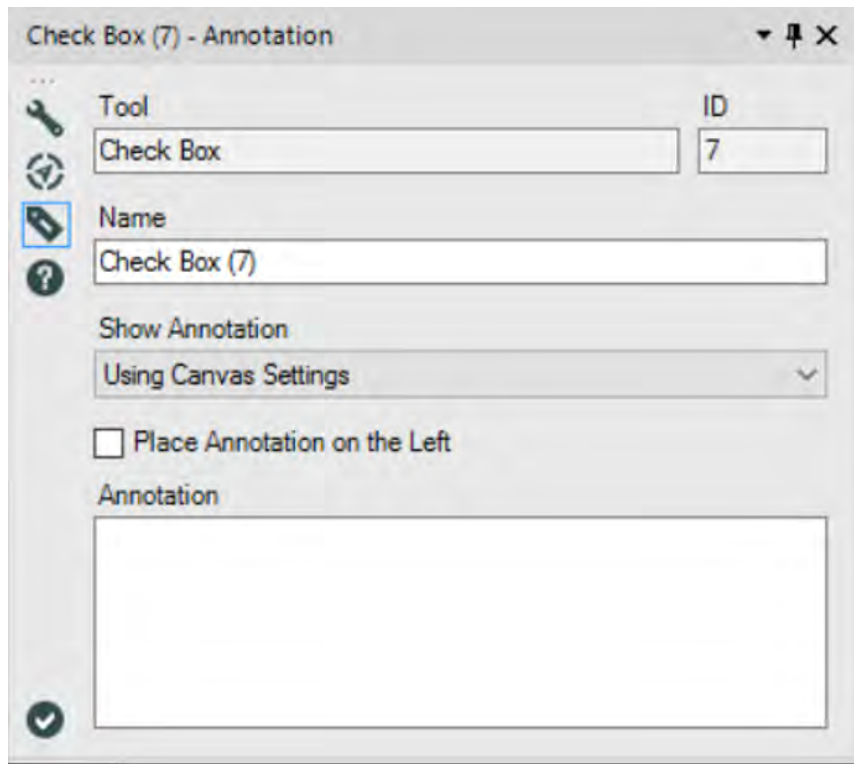
Figure 1-40 – Checkbox Annotation

The Annotation window tells us that the tool is a Check Box and that it has the ID number 7 (which means it will try executing this tool third if there are not outside influences). We then can change the name of the tool. We can change the annotation setting specific to this tool as well as provide a custom annotation (this will show up under the tool in the data stream).
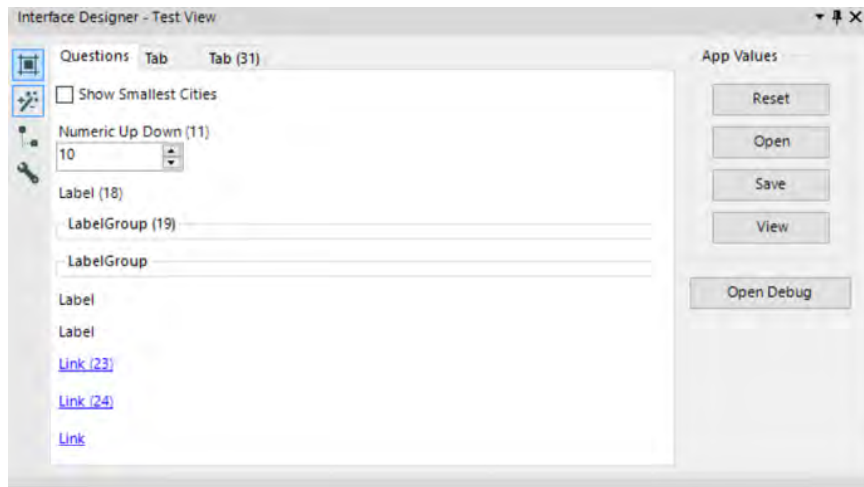
Figure 1-41 – Test View

Looking back at the Interface Designer, *Test View* allows us to answer the questions in the same way that the end user would so we can test the app.
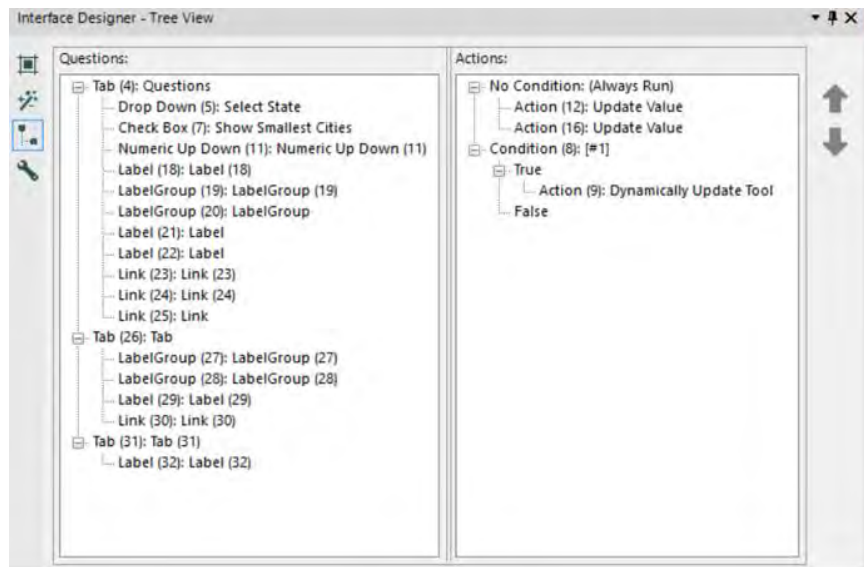
Figure 1-42 – Tree View

Tree-View allows us to look at our user interface as a tree structure so it's easier to make sure our logic is properly grouped.
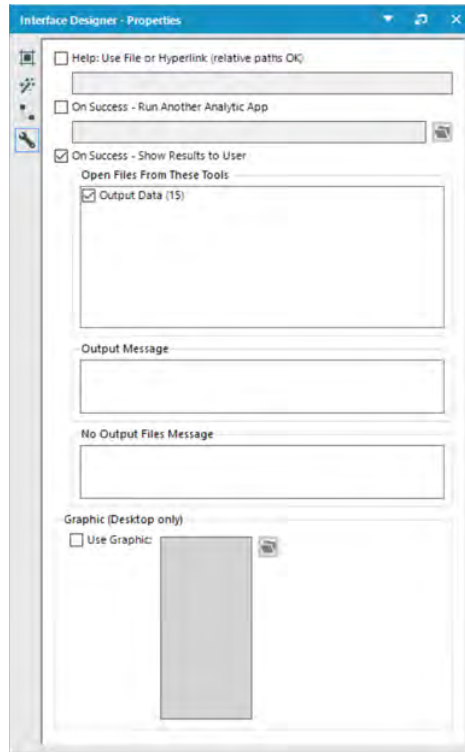
Figure 1-43 – Interface Designer Properties

Interface Designer Properties allows us to customize many aspects of our app or macro.

1.18 Canvas

This is what the default canvas looks like before we have brought any tools onto it.
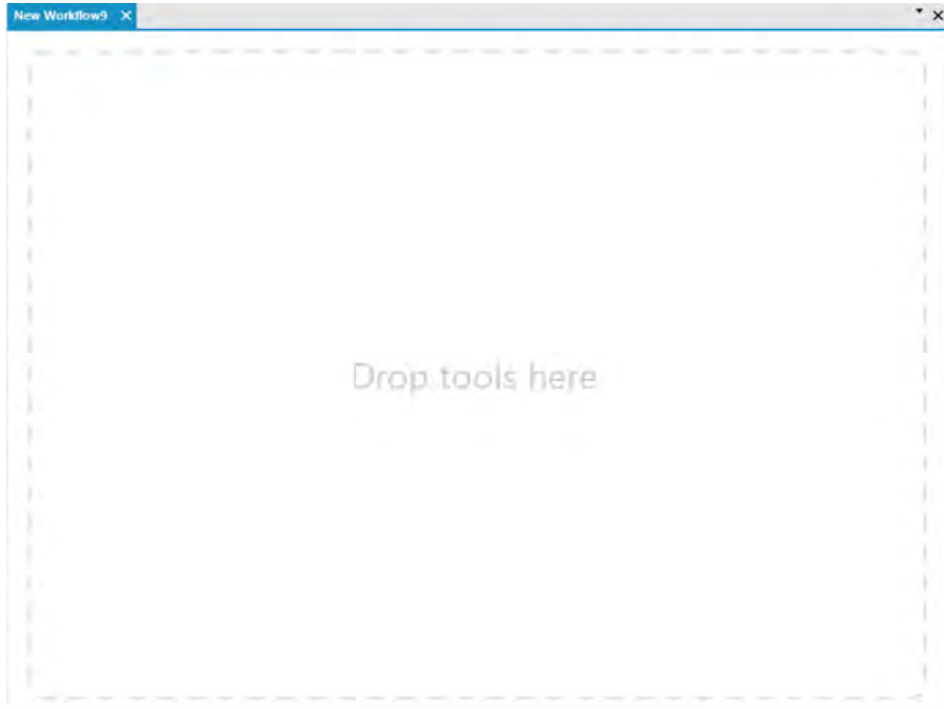
Figure 1-44 – Default Canvas

We can see three elements: the tab name, a drop-down list of all the tabs, and a white area that reads "Drop tools here."

Let's see what happens if we follow the following steps.

1. Open three new windows.
2. Click on the third tab.
3. Click on Window > New Horizontal Tab Group.
4. Click and drag the fourth tab (now third on the top) from the top group to the bottom group.
5. Click on Window > New Vertical Tab Group.

We should see that our canvas has split into three sections and looks like the image below.
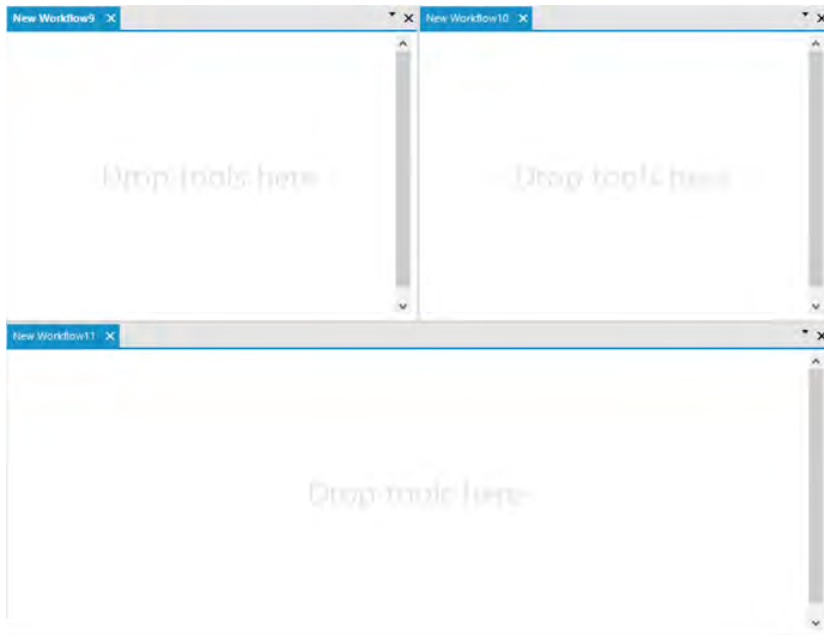
Figure 1-45 – Canvas with Elements

The ability to display workflows next to one another makes it much easier to test parts of a program. We can copy and paste portions between canvases to test, compare potential changes, or just work with multiple workflows at the same time.

Moving the locations of window elements

The overview, output, interface designer, and properties windows allow us to move them around the screen by clicking and dragging the rough part of the title bar (▨▨▨ or ▨▨▨). When we do this, we can snap them into the window by dropping them on one of the arrows or let it float in front of the canvas or other windows by letting go of it while not over an arrow.
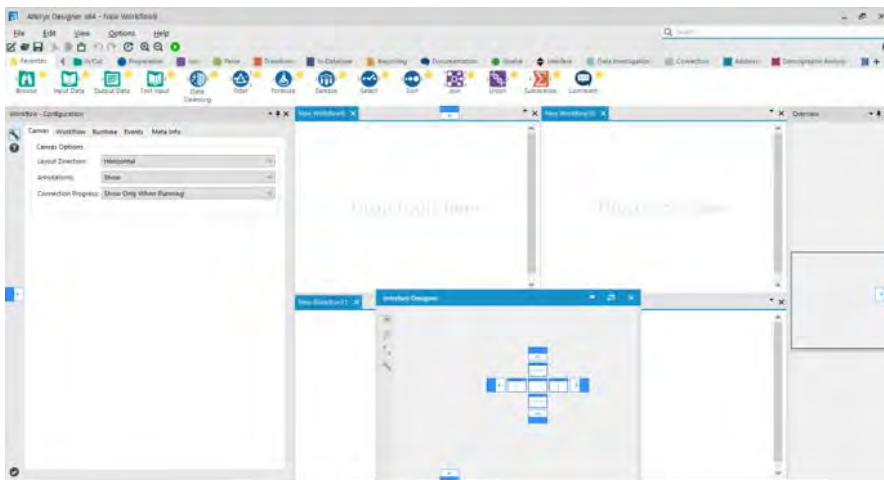
Figure 1-46 – Movement of Canvas Elements

1.19   Using Tools

Data streams start with some sort of data source, and in most cases, this will be the input tool. From there, we may do a series of operations before bringing the data back together to look at.

We will cover each of these methods in detail in the next chapter, but for now, we will look at the structure of a data stream.

This Alteryx Workflow takes input data from a file called Test.txt, splits the data stream (doubling the data), adds a new

field to one side, and brings the two streams together to be viewed.

Notice that there are two types of arrows on the incoming (left) side of the tools. A green arrow indicates that a single input can be connected there, while a gray double arrow indicates that multiple connections can be made. This is important because it allows us to know how to connect items to each of these tools.

We can see that there is no parallel tool on the outgoing (right) side of the tools, despite the data stream having multiple outgoing connections. This is because every tool allows us to branch the data stream off into many directions. We also see the lightning bolt and question anchors, which we will discuss when we first use them in *Applications Wanted*.

There are two ways to bring tools onto the canvas: We can both drag the tool from the Tool Palette or right-click on the canvas and navigate the menu on the next page.

1.20   Insert Tool Menu
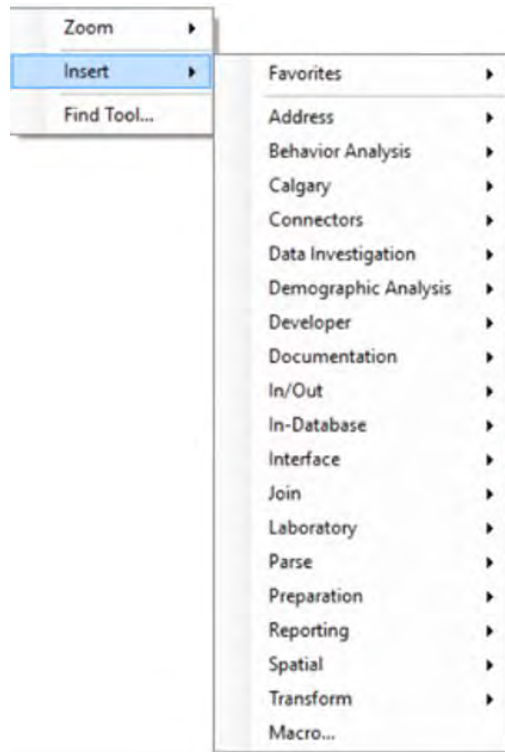
Figure 1-47 – Insert Tool Menu

As an alternative to using the tool palette, you can right-click on the canvas to add tools using the Insert Tool Menu. Simply navigate this menu to find the tool you are looking for.

CHAPTER 2
The Games

We all love playing make believe, don't we? As we discussed in the last chapter, assume that you, dear reader, are the new consultant at a company that works with the Alteryx Analytical Platform.

This book is your mentor and here is our first problem to solve!

To...	
Cc...	
Subject	Welcome - Let's Get Started

Hey,

Welcome Aboard!

We try to get all of our new hires a basic understanding of Alteryx as quickly as possible.

We will center the basic training around the most important sporting event in the world, which should need no introduction.

I will be asking you a few questions and walking you through examples until I feel like you are ready to handle it yourself.

The first question we are going to explore: Which country has produced the best Freestyle Skiing results overall in the 2002 and 2006 Winter Games?

Assume each Gold is worth 3 points, Silver is worth 1.5 points and Bronze is worth 1 point.

Something important to recognize is that I am asking you for the answer to a very specific question. Once you have some of the basics down, we will talk about making a generalized tool for you or your end user to ask related questions. For now, just understand that when you are asked about a specific answer, they are going to only want the result.

I'll show you how this works.

Thanks,

2.1   Tools & Concepts

In this chapter, we will cover the improved features for Alteryx designer 11.x and the various tools and concepts as mentioned below:

Tools	Concepts
Browse	Importing Data
Comment	Viewing Data
Filter	Outputting Data
Formula	Identifying Desired Results
Input Data	Answers to Questions
Join	Tidy Datasets
Output Data	Normalized Datasets
Running Total	Creating Calculations
Sample	Combining Data
Select	Creating Data Subsets
Sort	Summarizing Data
Summarize	Organizing Data
Tool Container	Organizing Workflows
Transpose	Documenting Your Work
Union	
Cross Tab	
Arrange	
Weighted Average	
Count Records	

2.2   Improved Features

Several updates were implemented since Alteryx 10.0 was released. Most of the tool icons got a new look, and some tools got new features. New features for each tool will be covered in their respective sections. Here, we will cover the new UI and Alteryx Designer's updated features.

- Reading and writing Excel files (.xlsx) is done using a new Alteryx driver for Microsoft Excel.
- The Predictive Tools Installer has been separated from the Alteryx Designer install to allow for future updates to the Predictive Tools and packages without the need for an update to the Designer.
- After running a workflow, clicking on the Input or Output tools will automatically populate results (up to 1MB).
- Tool palette is customizable to show only those categories and tools you want to use.
- The File menu is reorganized to be simpler and more intuitive to use.
- The enhanced integration between the Designer and Gallery makes it easier to collaborate on workflows hosted in a Gallery. From within the Designer, one can add and maintain connections to a Gallery, open Gallery workflows and edit them, and then save a new version of the workflow back to the Gallery. Previous versions of the workflow are maintained and can be retrieved via a version history window in both the Gallery and the Designer. Additionally, any workflow version can be made the "published" version that users will see by default.
- In-Database support is added for these Data Platforms: Amazon Redshift, Impala, Teradata, and Spark.

- Changes to the Connect In-DB tool include: new streamlined UI, new option to allow password decryption, and a new option for creating file-based In-Database connections in order to simplify obtaining connection information from IT Admin/DBA for server users.

- Changes to the Data Stream In tool include: new option to allow password decryption, users can write out to a permanent table when streaming data from an external data source.

- New Macro Input In-DB and Macro Output In-DB tools have been added for In-Database processing so users can build macros with In-Database.

- New Dynamic Input In-DB and Dynamic Output In-DB tools have been added for In-Database processing so users can retrieve the underlying SQL query and other metadata info being sent to the database or to the In-Database tools in a workflow.

- These connectors are now supported: Amazon Redshift bulk load (write), Netsuite (read/write), Qlik (read), PostgreSQL 9.4 (read/write), SAP Hana (read/write), Spark SQL (read).

- Users can now read and write to JSON from the Input and Output Data tools.

- Users can now browse to their Hadoop distributed file system and read/write to HDFS via the new connections in the Input Data tool. (HDFS tools released in 9.5 have been deprecated).

- Salesforce and Marketo connector tools built on the REST API provide enhanced functionality. The existing SOAP-based connectors are now deprecated.

- MongoDB Input and Output tools now support version 3.0. Mongo input will also read from replica set members including primary and secondary.
- Users now have the ability to retrieve a list of sheet names from an Excel (.xlsx) file and read the list via the Alteryx XLSX driver.
- Apache Avro support is no longer listed as being in beta. Support for SQLite has now been upgraded to 3.8.9.
- The Download tool now supports multithreading to increase the speed at which data is retrieved. The maximum number of connections that can be used in the same tool is 32.
- Authenticated proxy can now be enabled from User Settings. The Download Tool, Amazon Redshift Bulk loader, Amazon S3 Upload, and Amazon S3 Download tools will function through a proxy server.
- Alias Repository has been renamed to Alias Manager and supports both standard as well as In-Database connections.
- A new option that provides information about the performance of tools in a workflow is available. Select the "Enable Performance Profiling" checkbox on the Runtime tab of the workflow, and run the workflow to view the percentage of time spent processing each tool in the Results window. This option should only be used when debugging a specific workflow, as it may decrease the performance of the workflow slightly. Sample workflows that demonstrate different configuration options for one tool at a time are one example. Twenty new examples have been added, which can be accessed via Help > Sample Workflows > One Tool Examples. In/Out: Directory; Map

Input Preparation: Auto Field; Generate Rows; Imputation; RecordID; Tile; Unique Join: Append Fields; Find/Replace; Fuzzy Match; Make Group Parse: Date/Time; RegEx; Text To Columns; XML Parse Transform: Count Records; Cross Tab; Running Total; Transpose.

- The Block Until Done tool now has numbered outputs that output "in order."
- The Map tool now has a zoom/bounds option that makes the map zoom/pan to the reference file.
- In the Alteryx Designer version 11.x, the Configure Workflow and Results windows are split into two distinct views.
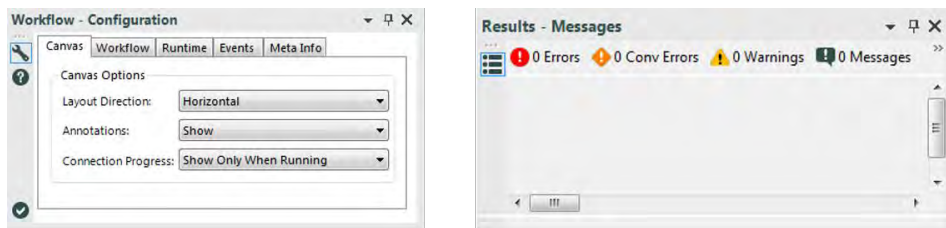
Figure 2-1 – Workflow and Results windows

- Configuration and Results windows can be dragged and displaced to different possible positions.
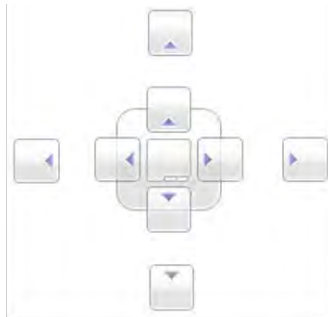
Figure 2-2 – Possible positions

- Every tool in Alteryx now has a Preview feature to show
  a snapshot of the data it contains.
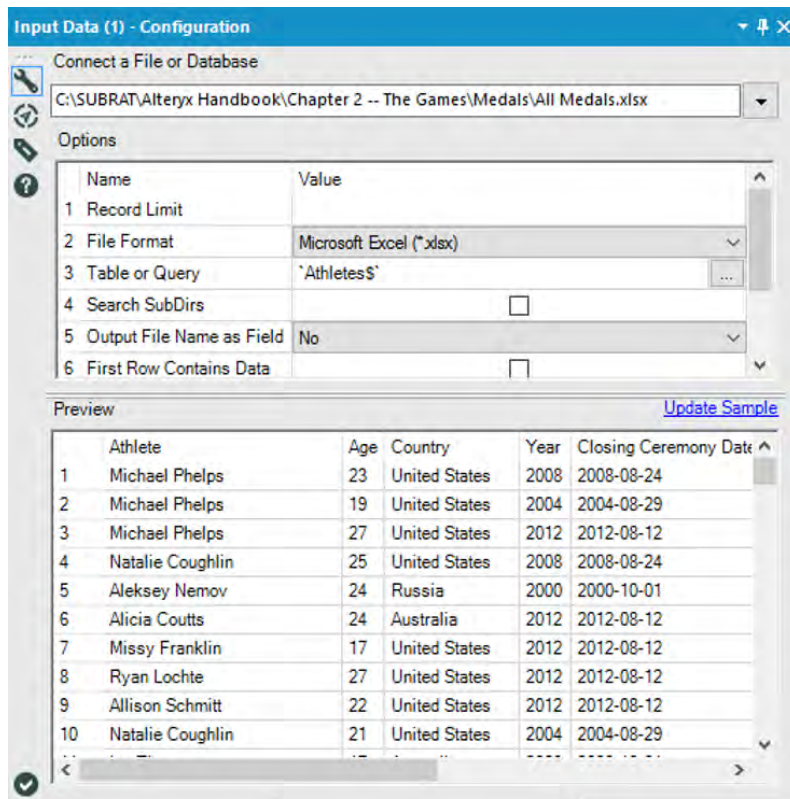
Figure 2-3 –Preview Data

2.3   Browse

	The *Browse* tool gives us a tabular view of the data in a data stream at the point it is connected.		
Figure 2-4 - Browse	**Group**	**Input**	**Output**
	In/Out	Data stream	None

Note: It is very important for the development of workflows, applications, and macros but should be disabled when development is completed to improve speed.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Results Window:

The *Browse Results* Window allows us to view the data that was in the data stream during the last run.

The message option shows the number of records processed, processing time, and any errors during processing.

Figure 2-5 – Browse, Results

The two icons on the right side (shown above) are *Copy* and *Save* functions. They allow us to directly copy the data out of the browse tool or save it to a file.
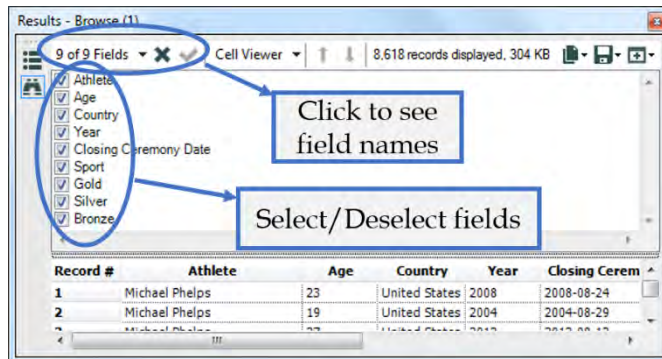
Figure 2-6 – Browse, Field Names

Clicking on the down arrow shown above gets us a list of all field names so we can select only the relevant fields to be displayed. The checkboxes ✓ in this list allow us to select or deselect every field.

The text shown after clicking on *Cell Viewer* depends on what is selected in the records.
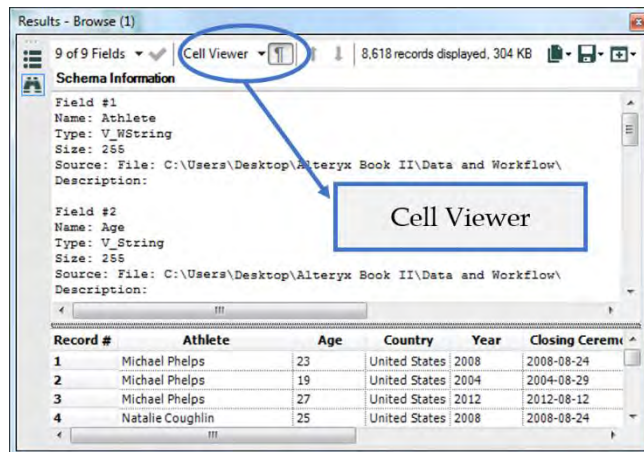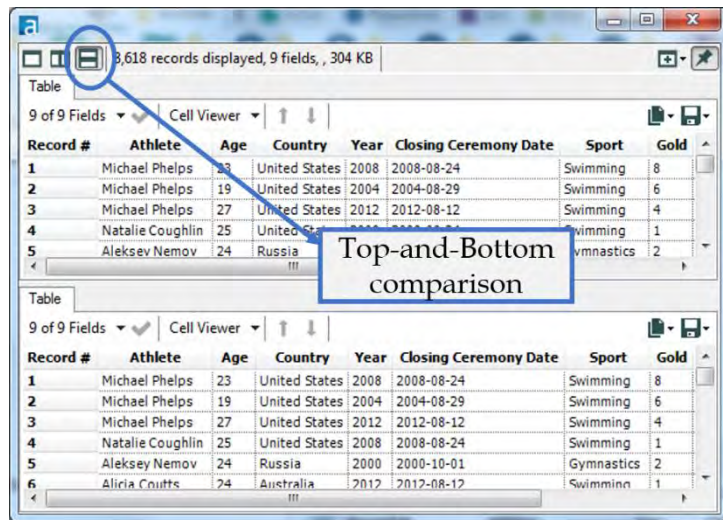
Figure 2-7 – Browse, Cell Viewer

Clicking *Record* # selects everything, allowing us to see the information about each column. If a single column is selected, we see metadata about that column. If a row is selected, we see how much data the record has in it. If a cell is selected, we can see the contents of that cell formatted with all line breaks.

Figure 2-8 – Browse, Side by Side

We have the ability to compare the dataset either Side-by-Side (vertical compare) or Top-and-Bottom (horizontal compare) by clicking on the new window button located at the top-right of the results window.

Figure 2-9 – Browse, Top and Bottom

We can also take a look at the metadata of the input data stream.

Figure 2-10 –Metadata Preview

We can know the data types of all the columns in the data.In this case, most of them are V_WString and the Medal Counts(Gold, Silver and Bronze) are in Double.
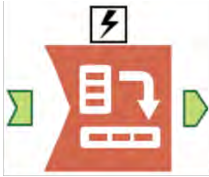
2.4   CrossTab

	The *Crosstab* tool creates a normalized (more human-readable) dataset by creating columns out of the rows of data.		
Figure 2-11 - Crosstab	**Group**	**Input**	**Output**
	Transform	Any data stream	Data stream wider than input

Note: The *Crosstab* tool will convert all spaces and special characters to underscores in the column headers.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The Cross-Tab Properties window has four components, as shown in the following figure.
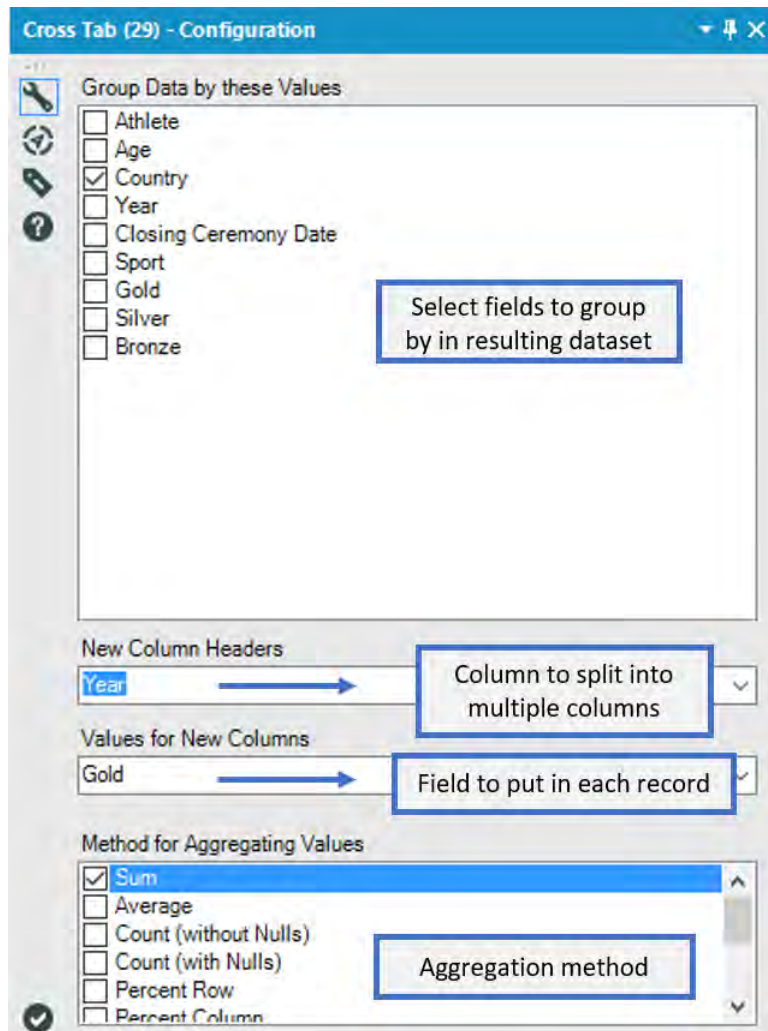
Figure 2-12 – Crosstab,configuration

We notice the gray boxes that have arrows at the top-left and bottom right corners. These show us what the input and output fields are. These boxes are standard for tools that have input and output.

- *Grouping Fields* allows us to select which fields we want to group by in the resulting dataset. If nothing is selected, we will only get a single record as output.
- *Header Field* is the field we are splitting into multiple columns.
- *Data Field* is the field we want to put in each record for the columns created by the Header Field.
- *Methodologies* (Numeric Data Field) allows us to select the type of aggregation method used if we have multiple data entries that fit into the same cell of the resulting dataset.

2.5 Comment

	The *Comment* tool gives us the ability to write notes on our workflows to add additional information on the data stream.		
	Group	**Input**	**Output**
Figure 2-13 - Comment	Documentation	None	None
Note: Comment is an annotative tool to help give meaning to developers using this workflow.			

Properties Window:

The *Comment* Properties window allows us to customize the comment field that appears on the canvas.
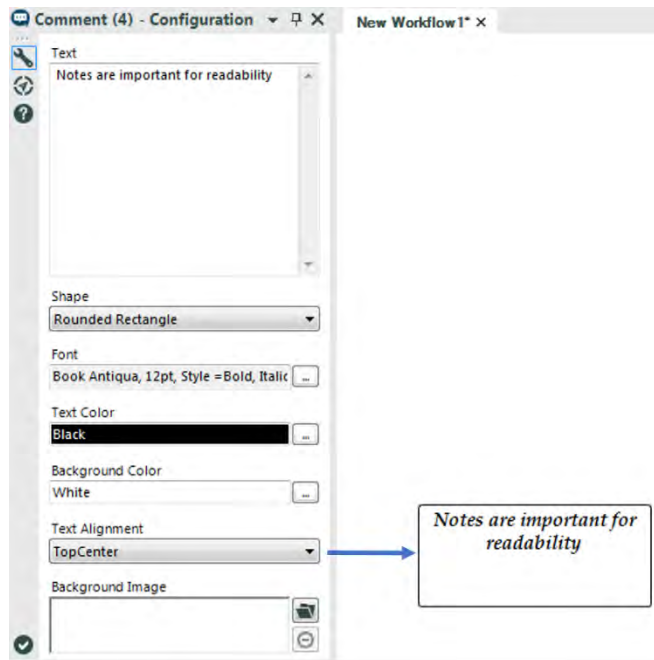
Figure 2-14 – Comment, configuration

The configuration settings in the figure above create the comment to the right of the window. We can edit text, shape, font and color of the comment background, adjust the alignment of the text, or select an image to write over.

Using the settings in the *Comment* tool allows us to create easily recognizable, distinct comments throughout our data stream.
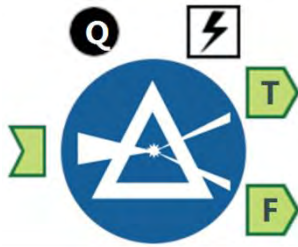
2.6   Filter

	The *Filter* tool gives us the ability to create a function that will split the data row by row into either the true or false outputs.
Figure 2-15 - Filter	
	Group	**Input**	**Output**
	Preparation	Any data stream	T & F section below

Note: The formula must evaluate to a True or False Boolean value (null evaluates to False unless the formula is looking for nulls). The formulas we create here can be arbitrarily complex, and thus can significantly slow our data stream. "//"is the comment character.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output T: This is the set of records where the formula evaluated to true.

Output F: This is the set of records where the formula evaluated to false.

A basic filter was added to the Filter tool. One can use this basic filter to quickly construct a simple query on a single field in the incoming data stream.

Properties Window:

We see that there are two different ways to create a filter: *Basic Filter* and *Custom Filter*. The regions for each are as shown below.
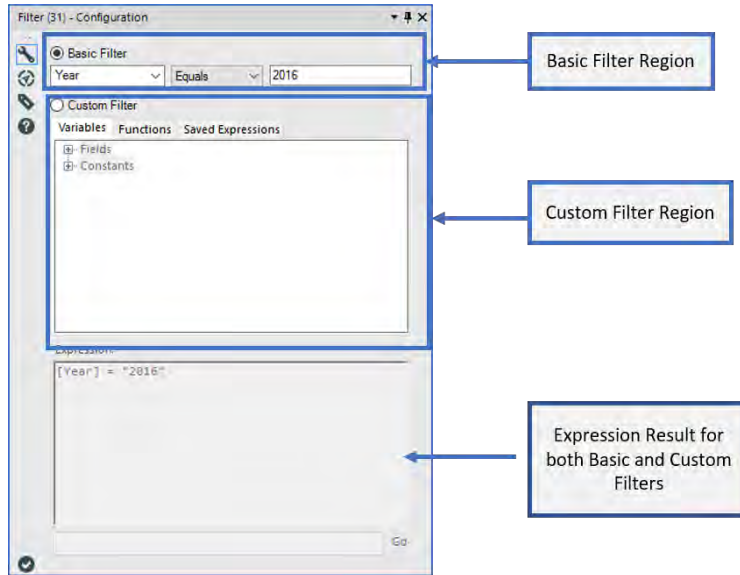
Figure 2-16 – Filter,basic filter

- *Basic Filter*: Allows us to pick a field and operator, and type in the value that the field should be compared to.

The options for the *Basic Filter* change depending on the field type. The *Basic Filter* option allows us to easily create simple filters as well as start building formulas before we know the syntax.

As we enter values into the *Basic Filter* options, it populates the Expression box at the bottom of the window. This helps us learn the associated syntax.

- *Custom Filter:* allows us to click on variables, function, and saved expressions to populate the expressions window or type the formula directly. A sample custom filter is shown below.
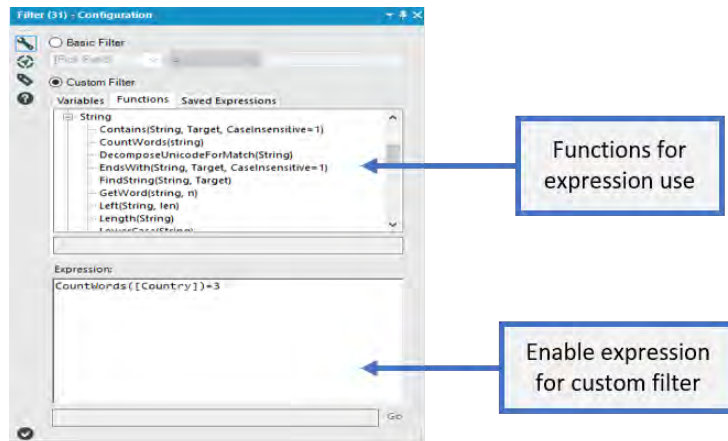
Figure 2-17 – Filter, custom filter

We can also use saved expressions using formulas in the *Formula* tool and throughout the exercises in this book.
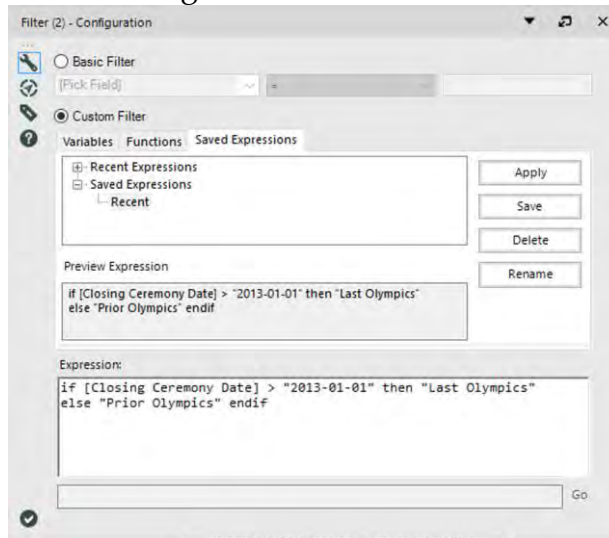
Figure 2-18 –Filter, Saved Expressions

2.7   Formula
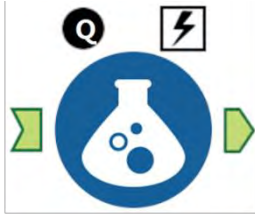
Figure 2-19 - Formula	The *Formula* tool gives us the ability to create a function that will be written to a new column in our data.		
	Group	**Input**	**Output**
	Preparation	Any data stream	Augmented original data stream

Note: The formulas created here can be arbitrarily complex, and thus can significantly slow the data stream. Ensure the created output field has a file type that's compatible with the result being created. We can use the formulas created higher in the list in calculations lower in that list. "//" is the comment character.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with one additional field for each formula we create.

A new function is added to the Formula tool: Starts With, Ends With, and Contains.

Properties Window:

The formula window looks similar to the filter window; in fact, the expression building section here is identical to the *Custom Filter*.

The top of the *Formula Properties* window has an option to define multiple calculations, change the order the fields are created, and define the major metadata for the field that we are creating.
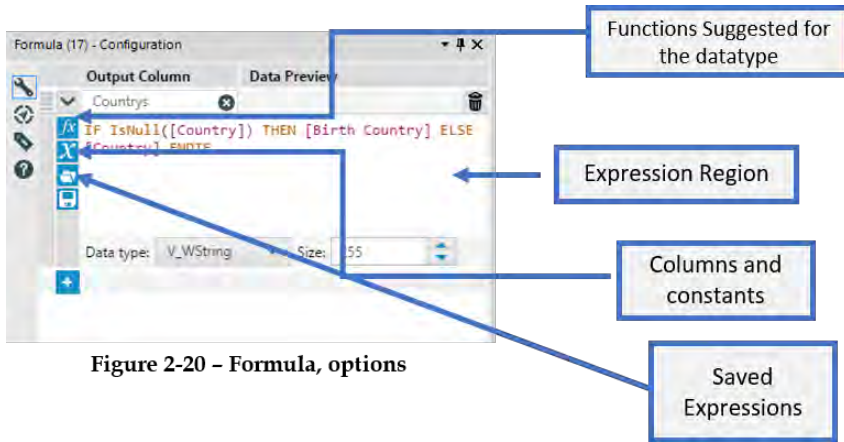
Figure 2-20 – Formula, options

The arrows on the side move the created formula up and down the list while the circle with the line in it removes the highlighted formula.

Each of these output fields will have an associated formula and will add a column to our data stream.

Under the *Variables Tab*, we can see *Fields* and *Constants*. The incoming data determines what list of Fields, and the *Environmental Variables* determine what the Constants are. (*Environmental Variables* can be defined in the workflow properties window; see Properties Window in Chapter 1 for more information.)

Under the *Functions tab*, we see a tree structure. This allows us to look for the functions needed by double-clicking on and moving them into the expressions window to work with.

The *Saved Expressions Tab* allows us to access recent and saved expressions, as well as save our current expression for later use.

2.8   Input Data

	The *Input Data* tool gives us the ability to import data from specific databases.		
Figure 2-21 – Input Data	**Group**	**Input**	**Output**
	In/Out	None	Data stream in initial data format

Note: This is the most common start to a data stream. We can use full or relative file paths to files as well as database connections.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

A new option is available for relational database connections. When enabled, the data is stored in a. yxdb file on disk so that data sources aren't hit repeatedly during workflow development.

Properties Window:

The *Input Data Property* window has three main components. The first is the field that shows our data connection. When connected to a data source, we can see the address of the file or database that we are connected to.
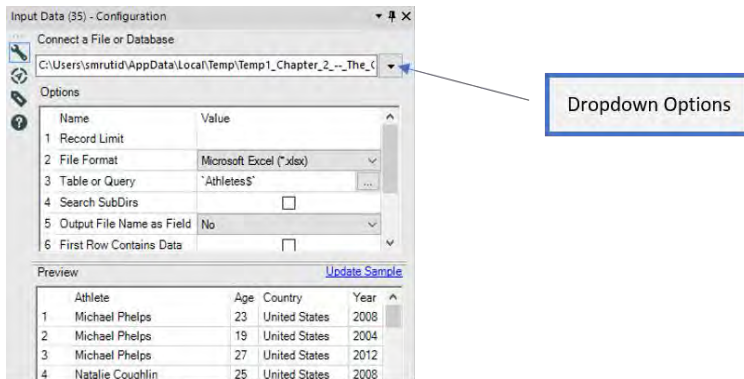
Figure 2-22 – Input Data, options

When we click on the drop-down arrow on the right-hand side of the field, we see this menu and can use this option to connect to data sources that are accessible. If present, a list of recent connections is shown, allowing us to save connections. This would appear under the Alias Link shown above.

The second main component of the window is the Options section where we can change the setting associated with the data connection to modify exactly what we are connecting to. This will allow us to modify many of the options that define the connection.

The third component is the Preview, which gives us a view of the first 100 records to help ensure that we are connected to the correct data source.

Table or Query: This option needs to be called out separately because it allows us to open a new connection window by clicking on the "…" button. That menu looks similar to this, depending on what we are connecting to.

The image below describes each of the tabs available when clicking on the "…" button.
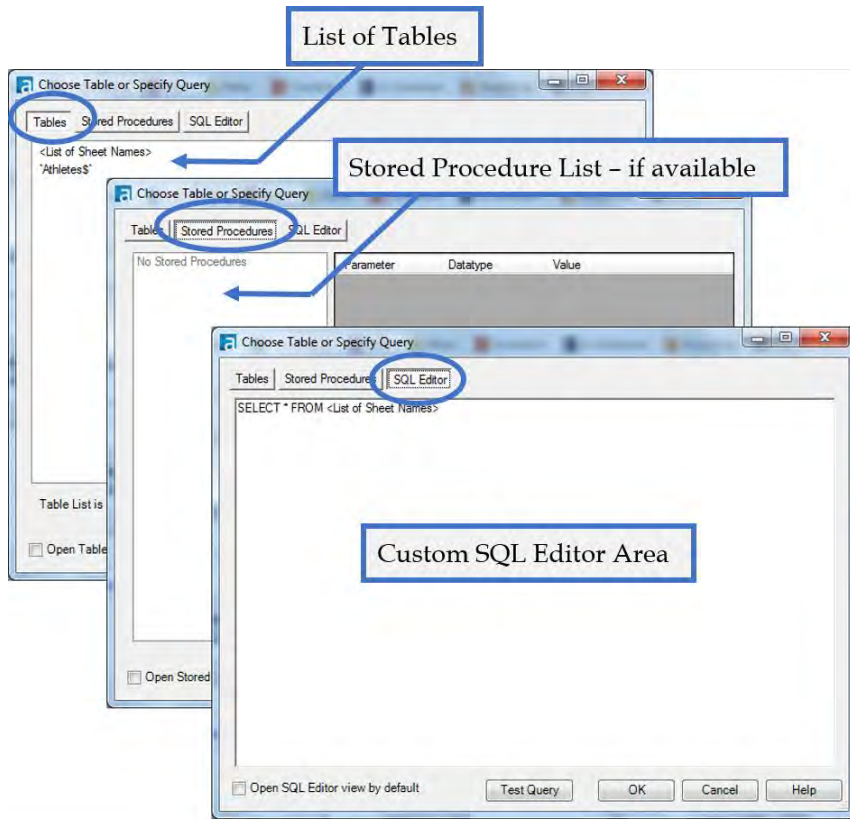
Figure 2-23 – Input Data, table and query options
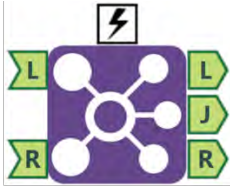
2.9   Join

	The *Join* tool gives us the ability to combine two data streams by lining up records based on matching fields.
Figure 2-24 - Join	**Group** / **Input** / **Output**
	Join — See Input Left and Input Right — See Output Left, Output Join, and Output Right

Note: Join does not work like a join in SQL; the tool creates three groups in order to perform a left, right, or full outer join We need a *Union* tool after the join to identify which outputs should be brought together. (See example "Brains vs. Brawns.") If we have multiple fields that match each other, records are replicated from the original data stream.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input Left: A data stream with at least one common field to Input R (the fields do not need to share a name).

Input Right: A data stream with at least one common field to Input L (the fields do not need to share a name).

Output Left: Data stream containing records that did not match anything from the right input.

Output Join: Data stream containing records that match both left and right inputs. Records may be replicated as a result of this operation.

> *Output Right:* Data stream containing records that did not match anything from the left input.

Properties Window:

The *Join Properties* window has two major components.

The top asks how we want to join the two data sets. We can join by position or by specific fields. Most of the time, we would be joining on specific fields because it allows greater control over the join.
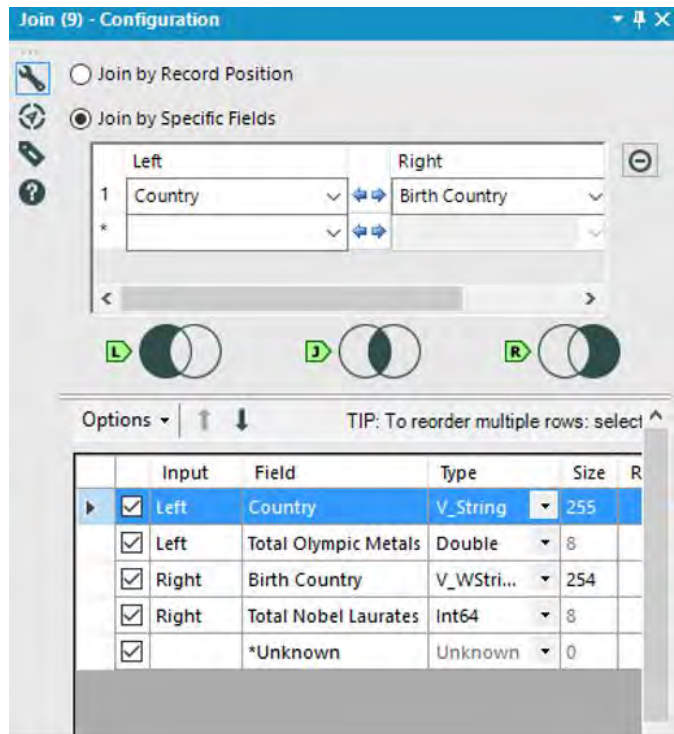
Figure 2-25 - Join Configuration

The bottom allows us to define which fields will be in the output as well as define the metadata for those fields (more on this when we talk about the *Select* tool).

We can see that there are three images separating the top and bottom of the window. These Venn diagrams show us what will be in each of the three outputs. More succinctly, we can consider the image below. The two inputs are the pink and blue circles, and the three outputs are the pink, purple, and blue shaded regions.
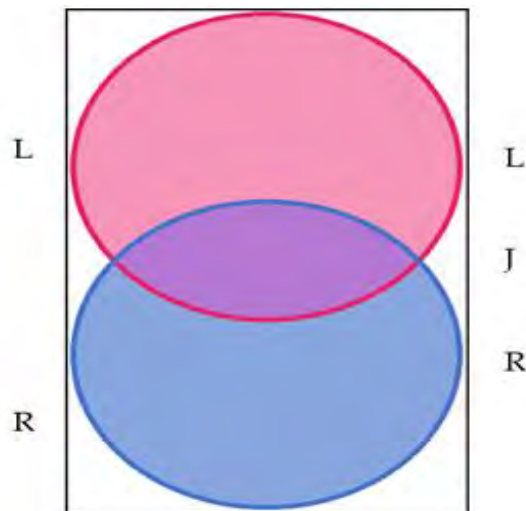
Figure 2-26 – Join Properties Venn Representation

2.10 Output Data

	The *Output Data* tool allows us to write the data stream out to a file or database		
Figure 2-27– Output Data	**Group**	**Input**	**Output**
	In/Out	Any data stream	File Or Database

Note: The output window has the ability to write to files or to databases using SQL.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The top half of the *Output Data Property* window is very similar to the Input Data window. Both allow us to navigate to a file or database and set options related to the dataset.
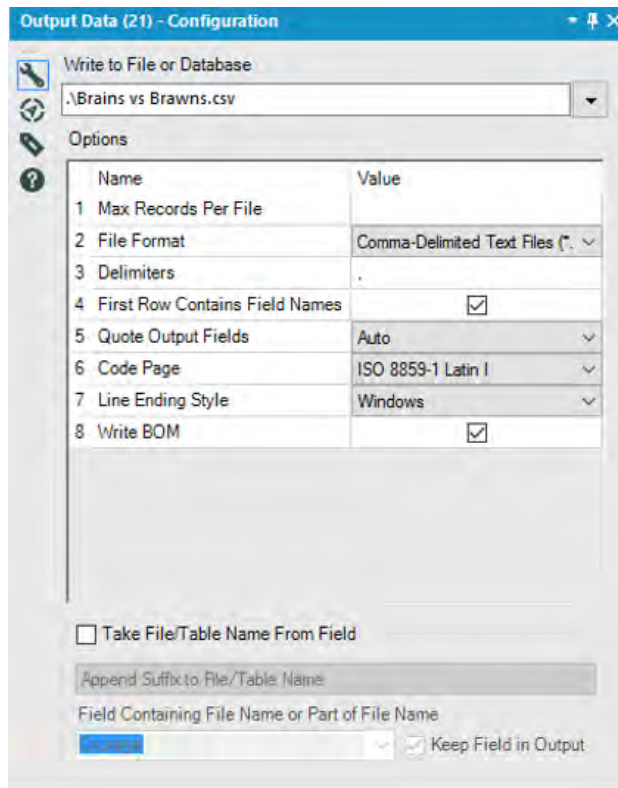
Figure 2-28 – Output Data Properties

The difference is that at the bottom, there are some special options that allow us to modify the way the metadata is written based on the incoming data stream.

We can see here that there is a file format called an "Alteryx database (*. yxdb)." Alteryx allows us to store data in files specifically designed to work well with Alteryx. These file types are native to Input and Output Data tools, so they do not need to perform any conversion to connect to these files. We can also use this file type to store both spatial and non-spatial data together.

2.11 Running Total
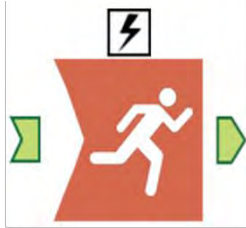
	The *Running Total* tool allows us to create a running sum for a numeric field in the incoming data stream
Figure 2-29 – Running Total	**Group** / **Input** / **Output** below

	Group	**Input**	**Output**
	Transform	See details below	See details below

Note: Running Total produces the running sum of the data from the top of the column down, so it is important to make sure the data is properly ordered (See the Sort Tool).

Input: Any data stream with at least one numeric field.

Output: The original data stream with additional columns called *RunTot_<Original Field Name>* for each of the selected "Create Running Total" fields.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Running Total Properties* window has two components.
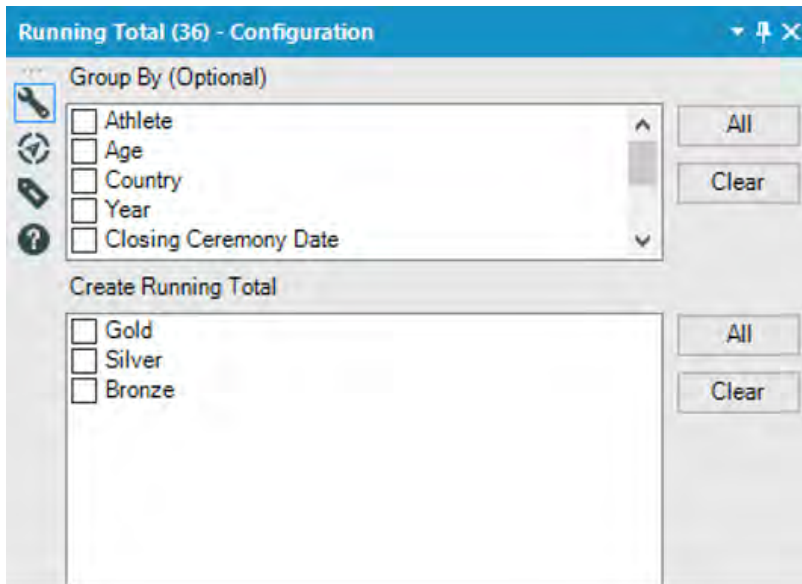
Figure 2-30 – Running Total Properties

The first is *Group By*, which allows us to define a field or set of fields that the running sum will be unique to the set of elements in our group by fields.

The second is the selection of which fields we want to create a running total on.

2.12   Sample

	The *Sample* tool allows us to work with a subset of data		
	Group	**Input**	**Output^**

|

Figure 2-31 - Sample	Preparation	Any data stream	See details below

Note: This is useful for limiting the amount of data we've run through our data stream when we are testing, creating different samples of our dataset, and skipping header or footer information that may exist in our data.

Output: The original data stream with potentially modified metadata and truncated fields.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Sample Properties* window has three different settings.
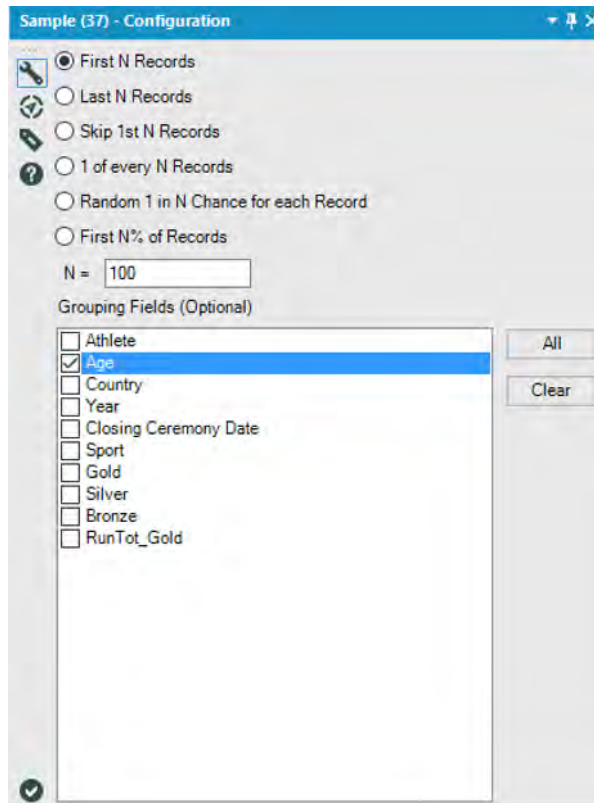
Figure 2-32 – Sample Properties

The first of these settings is the type of sampling that is needed.

First N Records: the first N records in our data stream.

Last N Records: the last N records in our data stream.

Skip 1st N Records: all but the first N records in our data stream.

1 of every N Records: create groups of records based on the order, and take one of each of those records.

Random 1 in N Chance for each Record: Every record has a 1/N chance of being kept.

First N% of Records: the first N percent of Records in our data stream.

The second setting is 'N' to be used in the sampling.

The third is the ability to select the fields we want to group the sampling by. In the scenario pictured above, this filtering will keep the first 100 records for each date in the data.

2.13   Select

Figure 2-33 - Select

The *Select* tool allows us to modify metadata associated with the data stream, including the order of columns		
Group	**Input**	**Output**
Preparation	Any data stream	See details below

Note: Select is used after every data connection and periodically throughout the data stream to ensure everything in the data stream is in the right format, named appropriately, and necessary. Use this tool to drop fields that are no longer needed to save space.

Output: The original data stream with potentially fewer fields with modified metadata.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Select Properties* window has multiple important tasks associated with the maintenance of data and metadata. The window shown below shows us a list of every field name coming in so we can reset the information if needed.
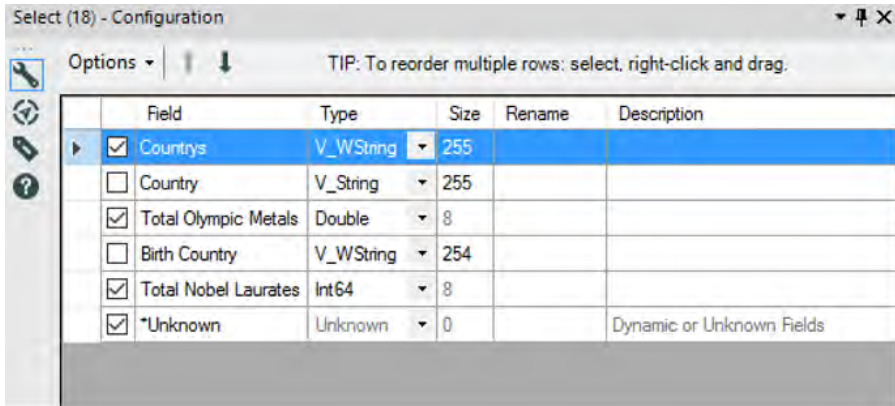
Figure 2-34 – Select Properties

In this case, the *Date* field has been renamed to *Sales Date*, and the *Sales* field was converted to a float type and given a description saying that it is the *Total Cash Sales*. The red cells above indicate that the metadata in the cell has been modified.

There are two other things we can do with the fields without going into the options menu: The first is checking and unchecking the boxes at the left of each row in order to drop that field from the data stream. The second is reordering the fields so that the columns are in a different order downstream. This can be accomplished by selecting a field and clicking the up and down arrows or by right-clicking in the space to the left of the checkmark and dragging the field up or down in the list.

In addition to the two fields that are part of the incoming data stream, there is a special field called *Unknown* that acts as a placeholder for all new fields that come into this tool.

Select Options Menu:

The options menu allows us to systematically modify the fields.
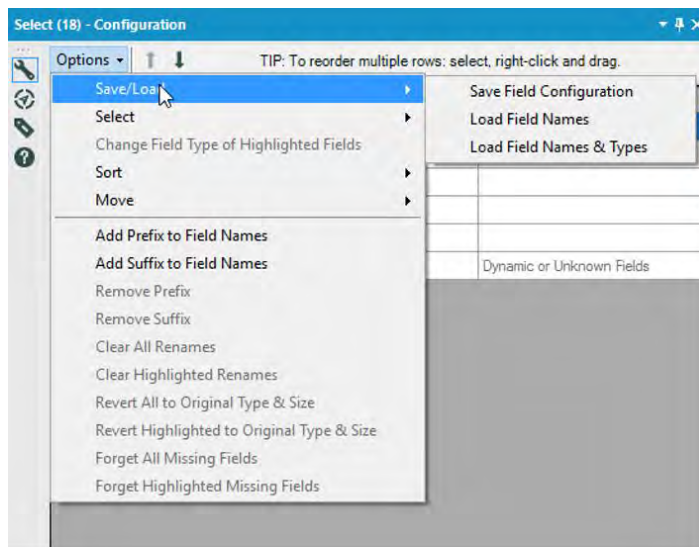
Figure 2-35 – Select Options1

Save/Load allows us to create or load a Field Type File (. yxft). Field Type Files are a metadata file that can be used to appropriately define or redefine the columns in our data stream.

Save Field Configuration creates a new yxft file.

Load Field Names imports field names from an existing yxft file.

Load Field Names & Types imports field names and the type of field that should be allocated.

Other options are as shown in the figure below.
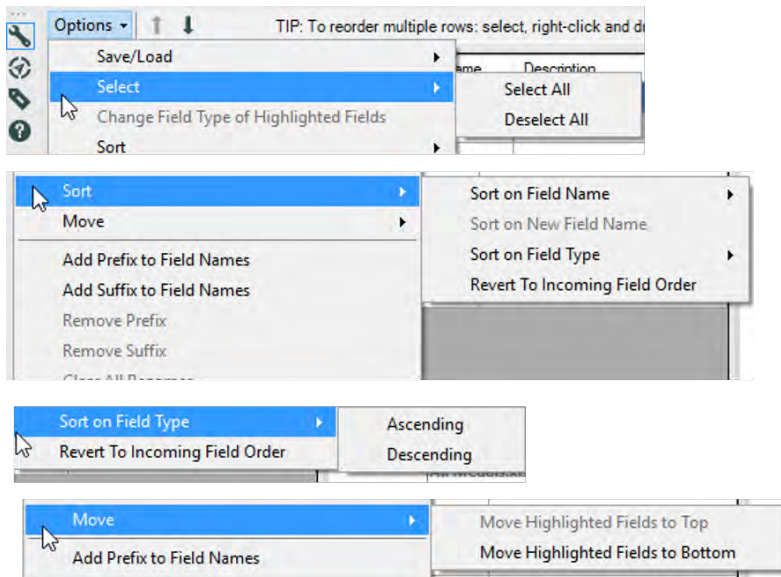
2-36 – Select Options 2

Select allows us to select or deselect all fields.

Sort has four primary methods of ordering fields.

Sort on Original Field Name will alphabetically sort our fields in either ascending or descending order based on the field names that came in from the data stream.

Sort on New Field Name will alphabetically sort our fields in either ascending or descending order based on the field names that leave the tool.

Sort on Field Type will group all fields that have the same data type together.

Revert To Incoming Field Order will clear the ordering of fields.

If we have selected fields, Move allows us to group them all at the top of the data field list.

Add Prefix to Field Names will allow us to add a prefix to all fields or all selected fields.

Add Suffix to Field Names will allow us to add a suffix to all fields or all selected fields.

Remove Prefix will allow us to remove a common prefix between selected fields.

Remove Suffix will allow us to remove a common suffix between selected fields.

Clear All Renames will remove all renaming that has been defined for this select.

Clear Highlighted Renames will remove all renaming in the highlighted (selected) fields.

Revert All To Original Type & Size will remove all changes to the field types or allocated data sizes.
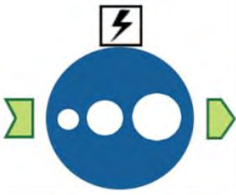
Revert Highlighted To Original Type & Size will remove all changes to the field types or allocated data sizes for highlighted fields.

Forget all Missing Fields will remove this tool's metadata about fields that are no longer coming into the tool from the data stream.

Forget Highlighted Missing Fields will remove this tool's metadata about fields that are no longer coming into the tool from the data stream that is highlighted.

Type: Type is an important thing to know because each one has a different attribute and means differently in other tools. Types are described in Appendix D.

2.14   Sort

|
Figure 2-37 - Sort	The *Sort* tool allows us to reorder records.		
	Group	**Input**	**Output**
	Preparation	Any data stream	See below

Note: Sorting is most important when we are working with calculations that consider multiple rows or ordering data for normalized consumption.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with records sorted in a different order.

Properties Window:

The *Sort Properties Configuration* window allows us to select one or more fields by name, and Ascending or Descending for each to determine an order to our records. We can change the order of these sorts by moving them up and down the list.
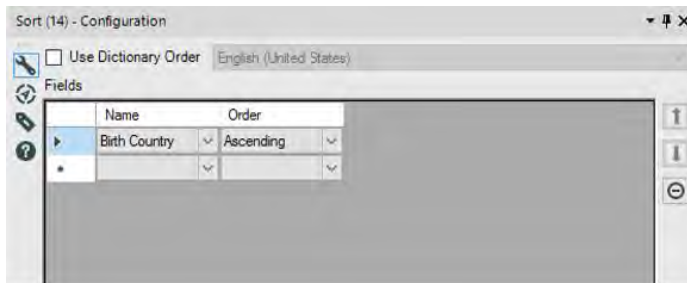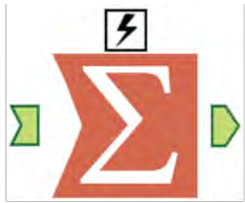
Figure 2-38 – Sort Configuration

By checking the *Use Dictionary Order* option, we can select the dictionary order that should be applied to sort the data when appropriate.

2.15   Summarize

	The *Summarize* tool allows us to summarize data in our data stream.		
Figure 2-39 - Summarize	**Group**	**Input**	**Output**
	Transform	See below	See below

Note: When summarizing data, it may be necessary to reexamine the underlying calculations because aggregating those calculations may not make sense. Summarizing a single field using Group By is a good way to get a unique list of the data. Running complex analysis like geocoding is often more efficient to do on a summarized list and then join back onto the full dataset.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with too granular a level of detail.

Output: A summarized data stream with a less granular level of detail.

Properties Window:

The *Summarize Configuration* window has two basic components.
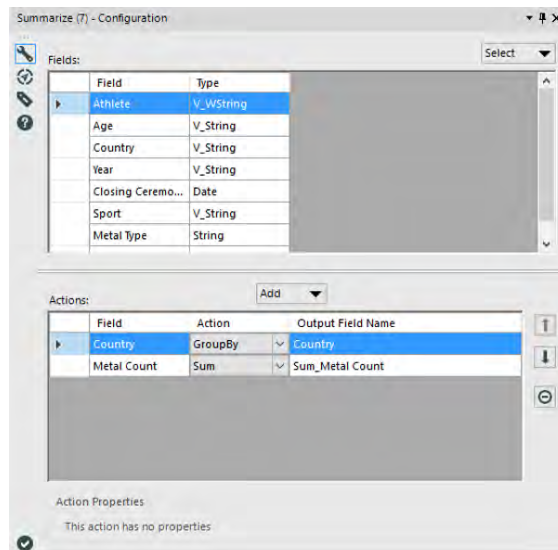
Figure 2-40 – Summarize Configuration

The *Fields* list shows each of the incoming fields, and the *Actions* list shows the fields created in this tool.

The select button at the top-right allows us to select tools in a systematic way so that if we want to take the sum of all of our Numeric fields, we can select them all and add them in one step.

When we have something in the fields list selected, we can click on the *Add* button, which shows the drop-down menu shown above. It lists every operation that can be used to aggregate the data using *Summarize*.

We will not be going through this menu in detail, as it is a list of aggregation methods; however, *Group By* needs to be given special attention.

When we use *Group By* on fields in the summary, we will end up with one line item for each combination of field elements

that we grouped by. If this is unclear, it should make more sense as we go through exercises.

2.16   Tool Container

Figure 2-41 – Tool Container	The *Tool Container* tool allows us to group tools together for clarity and allows the tools to be disabled when unnecessary.		
	Group	**Input**	**Output**
	Documentation	None	None
Note: Click and drag tools onto the box to put them into the tool container. Tool containers make it much easier to navigate our data stream because they allow us to consider a series of tools as a single unit. If we click on the arrow at the top-right corner, it will collapse the box without disabling it. An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Tool Container Configuration* window allows us to customize the text and formatting of the container as well as disable the tools inside it.
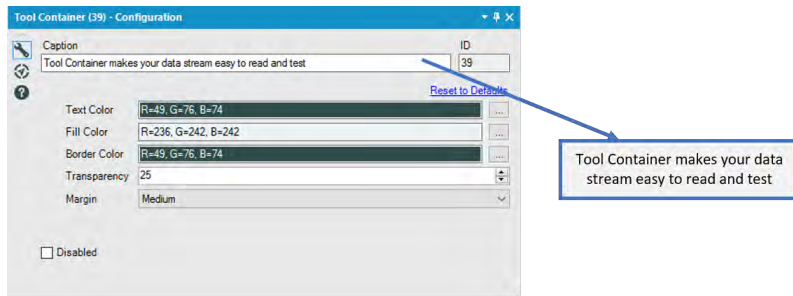
Figure 2-42 – Tool Container Configuration

The *Disabled* option allows us to turn off sections of our data stream. This is typically used in testing data and application building.
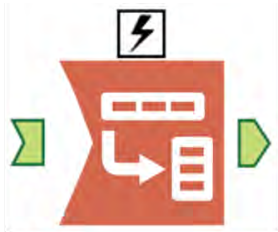
2.17   Transpose

	The *Transpose* tool allows us to de-normalize data		
  Figure 2-43– Transpose	**Group**	**Input**	**Output**
	Transform	See below	See below

Note: This tool converts all spaces and special characters in the titles into underscores in each record.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with multiple fields that need to be combined into rows.

Output: A taller data stream because records have been duplicated in order to consolidate the columns. The columns consolidate into two columns; *Name*, which is a column with

each of the former column names in it, and *Value*, which is a column with each of the data values.

Properties Window:

The *Transpose Configuration* window has three elements, as shown in the figure below.
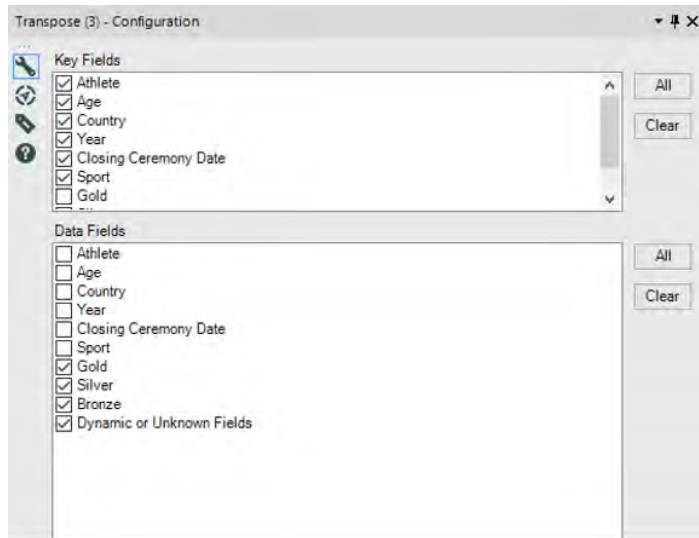
Figure 2-44 – Transpose Configuration

Key Fields allows us to select the fields that will be maintained after the transposition.

Data Fields allows us to select the fields that will be combined following the transposition.

The drop-down at the bottom allows us to change the message behavior if there are missing fields in the incoming data stream.
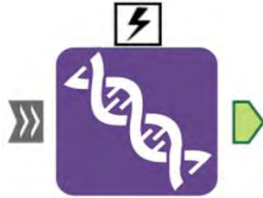
2.18   Union

	The *Union* tool allows us to append records together one after another from multiple data sources.		
	Group	**Input**	**Output**
	Join	See below	See below

Figure 2-45 - Union

Note: The order that we connect data stream to this tool's input will determine the default order that they are combined. In this tool, naming the incoming connections is often helpful.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Multiple data streams that should be combined by adding the records from one set to the end of the others.

Output: A data stream that has the records from multiple data streams.

Properties Window:

The *Union Configuration* window has three core elements.

The first drop-down allows us to change the method that is used to align the columns from the different inputs.
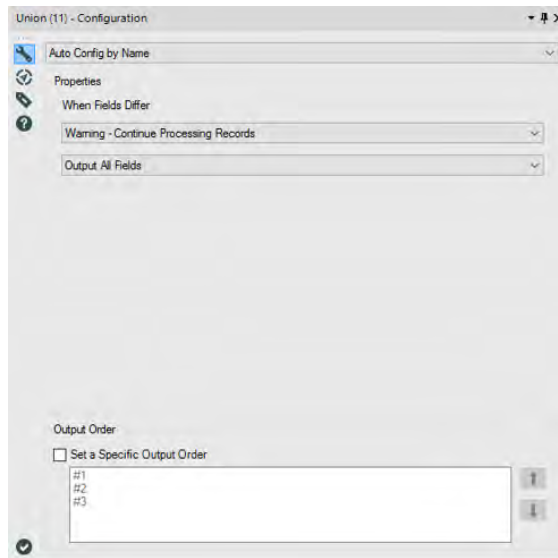
Figure 2-46 – Union Configuration by Name

Auto Config by Name makes the union align the fields that have the same name. This is best used if we know that our data will always be named the same way.
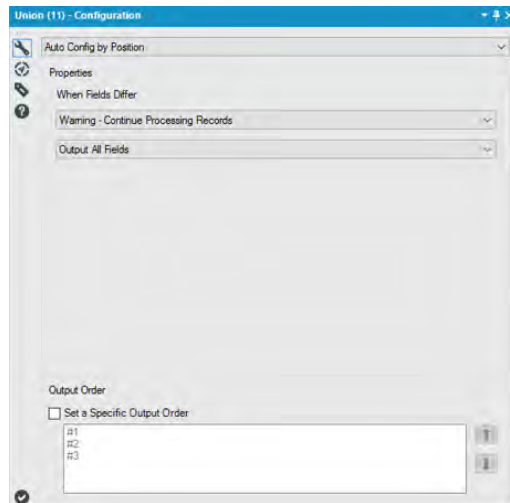
Figure 2-47 – Union Configuration by Position

Auto Config by Position makes the union align the fields by the column number. This is best used if we know our data will always be in the same order but may have different (or no) field names.

In both of these options, we see the same Properties section asking what should happen when the fields differ. The first drop-down allows us to change the behavior between an error, a warning, and nothing. The second drop-down allows us to decide if we want all fields or only the fields matched from all outputs to be in the output.
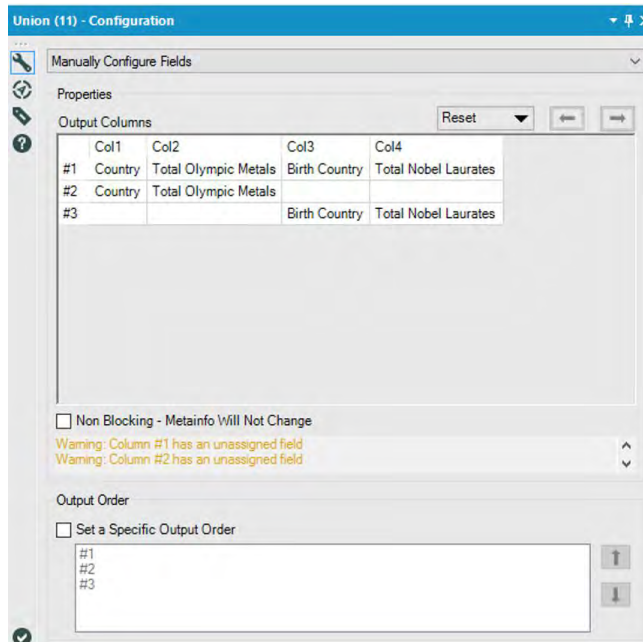
Figure 2-48 – Union Configuration Manual Configuration

Manually Configure Fields is the third option in the first drop-down. It allows us to select exactly which fields to be brought together by manipulating the Output Columns portion of the configuration window. This is best used if we know that our fields may be named differently or may be in different order.

The final component is the *Output Order*. It allows us to set the order of the records by choosing the order that the data streams are combined.

2.19   Arrange

	The *Arrange* tool allows you to manually transpose and rearrange your data fields for presentation purposes.		
Figure 2-49 – Arrange	**Group**	**Input**	**Output**
	Transform	See below	See below

Note: Data is transformed so that each record is turned into multiple records and columns can be created by using field description data or manually created. It can be used for staging data for reports.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with multiple fields that need to be combined into rows.

Output: A data stream that has the records from multiple data streams.

Properties Window:

The *Arrange Configuration* window has following elements, as shown in the figure below.
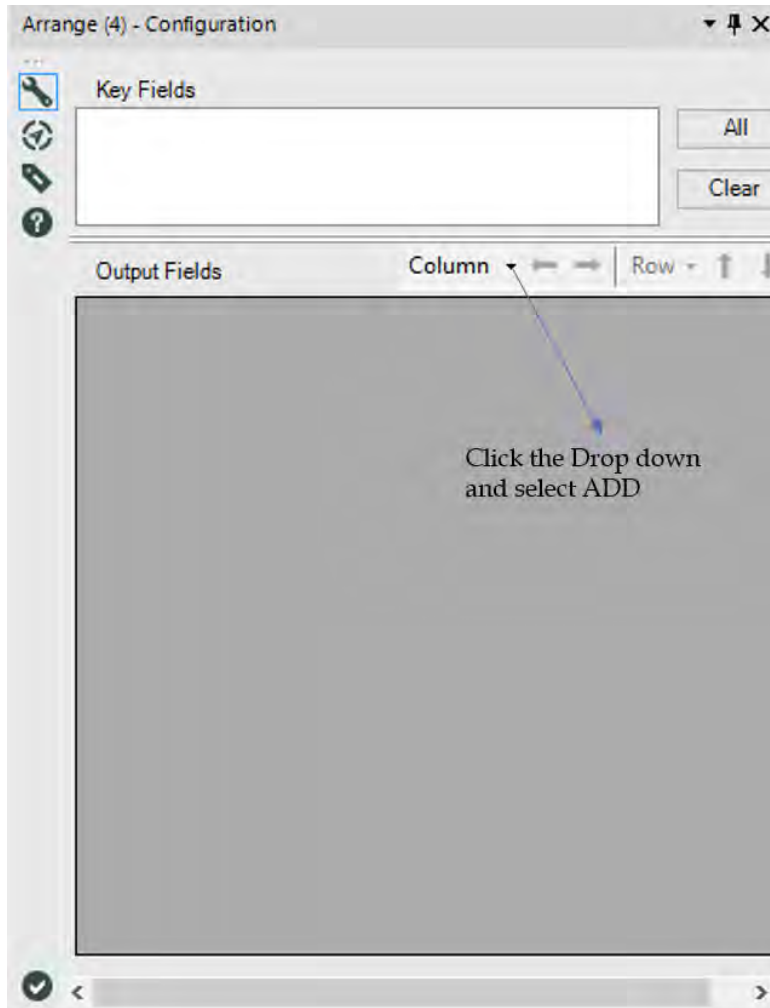
Figure 2-50 – Arrange Configuration

Key Fields allows us to select the fields that will be maintained after the transposition.

Construct the *output fields* from the Column drop-down, click Add. The Add Column dialog opens.

Figure 2-51 – Arrange Configuration: Add Column

Column Header: Type the name of the new column of data.

Fill in Description Column: Adds a new column to your description value of the selected fields. Specify None and no description Column will be added.

Description Mode: How the description field is populated. Choices include:

- Take from Descriptions: The description for the field is used from its meta info.
- Take from Field Names: The field name is used.

- Leave Empty: The description is left empty.

Description Header: Type the desired column header name for the new Description field.

Fields: Select the fields that will make up the data values for the new column of data. Use the All and Clear buttons to select All fields or Clear your selection.

Additional Column Menu items include:

- Delete: deletes the entire column of the highlighted record.
- Rename Column Header: prompts the user with a rename box for the column of the highlighted record.
- Replace Highlighted Description: prompts the user with a search and replace box and replaces the text as specified for the highlighted records.
- Replace Entire Column Description: prompts the user with a search and replace box and replaces the text as specified for the entire description column.
- Left/Right arrows: re-orders the selected columns left or right.

Row Menu Items Include:

- Add: Adds a blank row. The user can type the description manually and use the drop down to select a field.
- Delete: Deletes a selected row.
- Up/Down arrows: re-orders the selected rows up and down.

2.20   Count Records

Figure 2-52-Count Records	The Count Records tool returns a count of the number of records passing through the tool.		
	Group	**Input**	**Output**
	Transform	See below	See below

Note: The Count Records tool returns a count of the number of records passing through the tool.

This tool will come in handy when the user wishes to report on the resulting record count of a process. It will even return a count of zero which is a scenario that a Summarize tool will not handle.

Input: Any data stream with multiple fields that need to be counted for further data processing.

Output: Count of a set of data.

Configuration Properties

This tool does not need to be configured.

Simply connect the stream you want a count on into the Count Records Macro and connect its output to the desired tool.

2.21   Weighted Average Tool

	The Weighted Average tool calculates the weighted average of an incoming data field.		
	Group	**Input**	**Output**
Figure 2-53 –Weighted Average	Transform	See below	See below

Note: A weighted average is similar to a common average, but instead of all records contributing equally to the average, the idea of weight means some records contribute more than others.

Input: Any data stream where we need to calculate the weighted average of a set of values where some records are configured to contribute more than others.

Output: A data stream that has weighted average based on the input contributions.

Properties Window:

The *Weighted Average Configuration* window has following elements, as shown in the figure below.

Figure 2-54 Weighted Average Configuration

Value Field allows us to select the numeric field that contains the data to return the weighted average on.

Weight Field allows us to select the numeric field that contains the weight to calculate the weighted average of.

Note: When all records have the same weight the returned value is a common average.

Output Field Name asks us to Type in the desired outputted field name for the calculated Weighted Average.

Grouping Fields is an optional setting. The user can elect to specify a grouping field and the weighted average will be calculated for each group of values in the data stream.

2.22   Freestyle

To...	
Cc...	
Subject	Welcome - Let's Get Started

Hey,

Welcome Aboard!

We try to get all of our new hires a basic understanding of Alteryx as quickly as possible.

We will center the basic training around the most important sporting event in the world, which should need no introduction.

I will be asking you a few questions and walking you through examples until I feel like you are ready to handle it yourself.

The first question we are going to explore: Which country has produced the best Freestyle Skiing results overall in the 2002 and 2006 Winter Games?

Assume each Gold is worth 3 points, Silver is worth 1.5 points and Bronze is worth 1 point.

Something important to recognize is that I am asking you for the answer to a very specific question. Once you have some of the basics down, we will talk about making a generalized tool for you or your end user to ask related questions. For now, just understand that when you are asked about a specific answer, they are going to only want the result.

I'll show you how this works.

Thanks,

Let's start building a workflow that will answer our question. We are going to start with a blank canvas and save it as Freestyle Skiing. Next, bring an Input Data tool so that we can connect to data.

Figure 2-49 – Freestyle Skiing, Data Input

Now navigate to where data files are unpackaged, and connect to the file *All Medals.xlsx* in *Chapter 2 -- The Games> Medals*. For downloading the data associated with this book, please refer to the letter to the reader on this chapter's first page.

When connected, we see this window pop up. Click on *Athletes$* then OK to connect to the Athletes sheet in the All Medals Excel file. This is shown in the figure above.

Best practices are to put a select and a browse after every input.

- Browse helps us check the data at the time of import. This ensures that the data we are getting is correct.

- Select allows us to make sure that the fields are in the right format from the beginning.
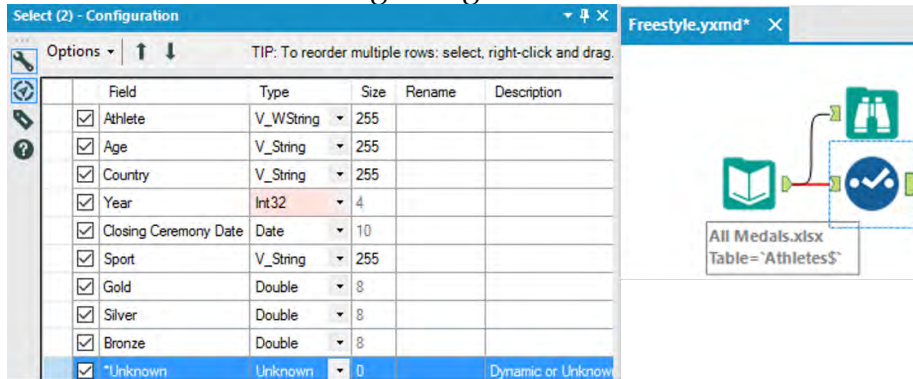
Figure 2-50 – Freestyle Skiing, Select Configuration

If we click on Select, we should see that our fields are of different types than the above image. Change them to match what is shown.

Now that we have the data and the fields are the right type, the first thing we should do is filter the data. We always want to *limit the data as soon as possible*, since this will speed up our data stream and prevent memory errors by limiting the information.

Best practice is to remove data as soon as it is no longer needed.

It makes sense that the first step in filtering would be to bring in the Filter tool; however, if we are not familiar with the data set and we have not run it, we may not have enough information to filter properly. In this case, we want to run the module so that there's data in the Browse tool for us to work with.
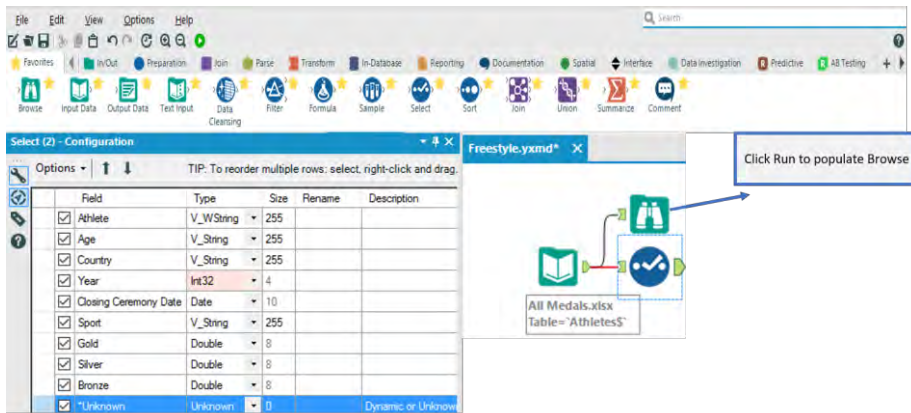
Figure 2-51 – Freestyle Skiing, Populate Browse

We can see that when the module finishes running, we get a pop-up window that lets us know how long it took to run and if there were errors.
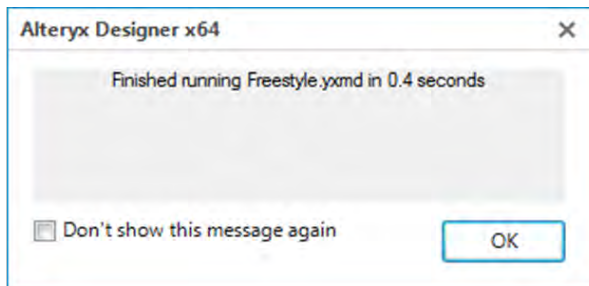
Figure 2-52 – Freestyle Skiing, message after Running

Feel free to click on the *Don't show this message again* check box before closing if the pop-up window is distracting.

Now, we can start thinking about the filter. We know that we are only interested in freestyle skiing results for the 2002 and 2006 games. So the first thing we are going to filter is the sport of "freestyle skiing." If we look at the browse tool, we first see freestyle skiing at row 5818, identified by the string *Freestyle Skiing*.

Figure 2-53 – Freestyle Skiing, Browse configurations

This is the crucial piece of information we didn't have before. Now that we know exactly what we need to look for in our data so we can create the filter.

Drag a Filter tool after the Select, and make sure there is a connection between the Select output and the Filter input. This time, we will use the Basic Filter builder. Set the field drop-down to *Sport*, and type *Freestyle Skiing* into the text box like we see below.
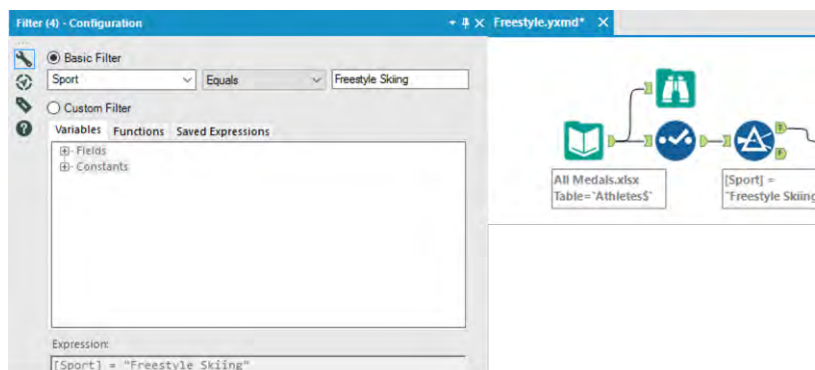
Figure 2-54 – Freestyle Skiing, Browse Configuration

Notice that the Expression says **[Sport] = "Freestyle Skiing."** This is because field names are in brackets, and string values are in quotes. What is happening here is that for each record, we test to see if the value in Sport in exactly *Freestyle Skiing.* If it is, then True; if it isn't, then False.

The next thing we want to do is create a filter to keep only 2002 and 2006. If we look at the Select tool on the previous page, we will see that the *Year* field is a string. This is fine; we need to remember that when we are writing the filter formula. Drag a new Filter tool onto the canvas, and make sure that the first Filter's true (*T*) output is connected to the new Filter's input.
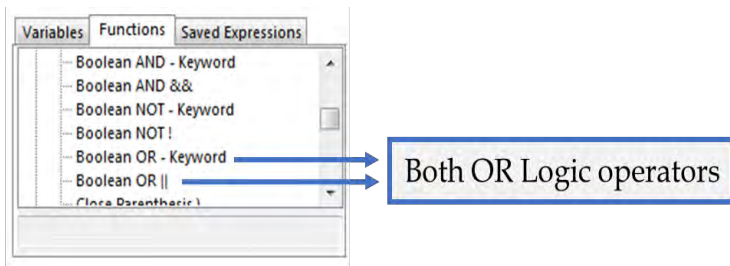
Figure 2-55 – Freestyle Skiing, Filter configurations - Functions

This time, we are going to create the filter logic on our own using the Custom Filter option. We know from the previous filter that **[<Field Name>] = "<Value>"** is the syntax for filtering a string field, so creating the first half of the filter is not hard – it is **[Year] = "2002"** – but what we need to do now is make sure 2006 is also kept.

One way we could do this is by using logical operators. *Logical operators* are terms that allow us to combine two or more Boolean (true or false) values to create a single Boolean from the two. The three Boolean operators that we will be discussing are *and*, *or*, and *not*.

AND: if **both** the value to the **left and** the value on the **right are true, then true**.

OR: if **either** the value to the **left or** the values to the **right or both** are **true, then true**.

NOT: **if** a value is **true** then **false**; **if** the value is **false,** then **true**.

      We can see in the *Functions Tab* that we have the option to use the *Boolean OR - Keyword* or the *Boolean OR ||*. There is no computational difference between using the keyword or the double vertical bar symbol. Both options are available for our convenience. For those not used to programming, the keyword *OR* is much easier to remember and use, but for those who program, double vertical bars (||) is a common standard they may be used to.
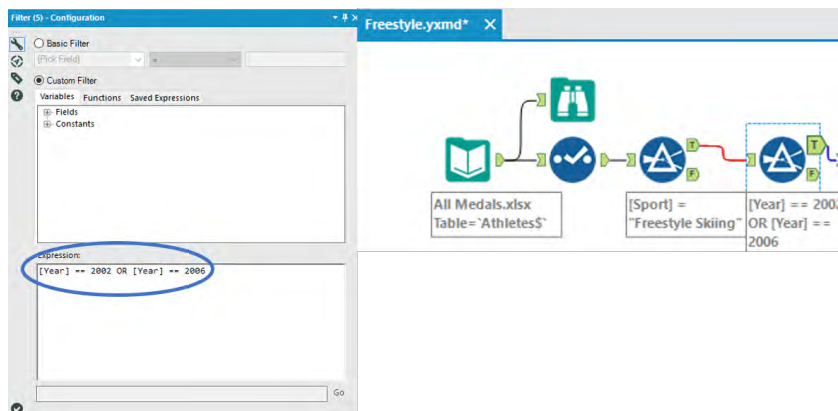
Figure 2-56 – Freestyle Skiing, Custom Filter

      See Appendix F for examples of Boolean logic.

      Now that we know about logical operators, we can finally finish putting the filter together. We can use the formula **[Year] = "2002" OR [Year] = "2006"** in order to filter out this data.

Note: We could have combined both of these filters together by using the following: **[Sport] = "Freestyle Skiing" AND ([Year] = "2002" OR [Year] = "2006")**

The parentheses allow us to change the order of operations so that this formula reads "Freestyle Skiing in the years 2002 or 2006" instead of "Freestyle Skiing in 2002 or anything in 2006."

We now have removed all of the information we don't need in order to answer the question. But we have the problem that the data is too granular. We know who the athlete was and in which year they won their medal(s). We should bring in a Summarize Tool in order to bring the data up to the country level. If we add Country using Group By, and Gold, Silver, and Bronze using Sum, we will get a list of countries and their total medal count for Freestyle skiing for 2002 and 2006. Place a Browse Tool, and run the module to see what we have so far.
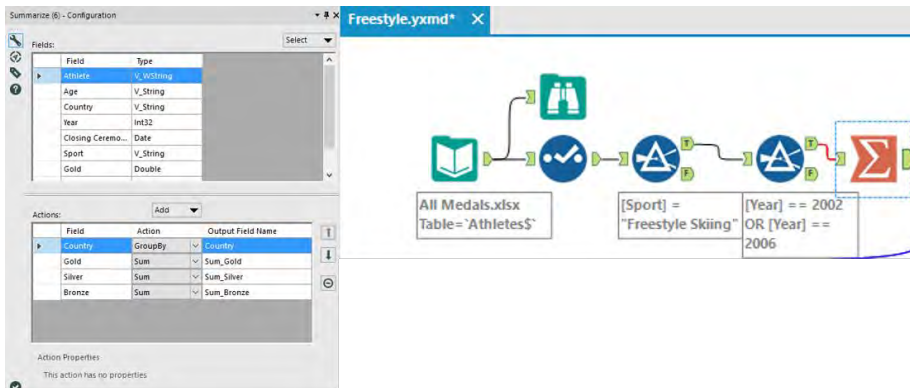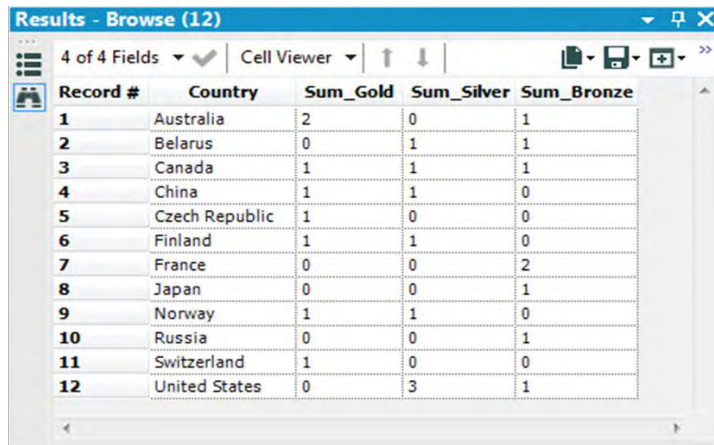
Figure 2-57 – Freestyle Skiing, Summary Configuration

Best practice is to place a browse tool after every tool that transforms data into a significantly different shape. Summarize is one of the tools.

Figure 2-58 – Freestyle Skiing, Browse configuration after summarize

We can see that we have four columns with the total counts of gold, silver, and bronze medals listed for each of the 12 countries that won freestyle skiing medals during 2002 and 2006.

Notice the fields are titled *Sum_* followed by the original field name. Alteryx is making sure we know the method used to summarize the data.

The next thing we need to do is determine which country was the *best*. If we look back at the email, we can see that *best* is defined as a function of the medals won; 3 points for gold, 1.5 points for silver and 1 point for bronze.

Bring a Formula Tool onto the canvas following the Summarize tool, and we are going to create a calculation called Score that has the Type Double, with the formula [Score] = 3*[Sum_Gold] + 1.5*[Sum-Silver] + [Sum_Bronze]
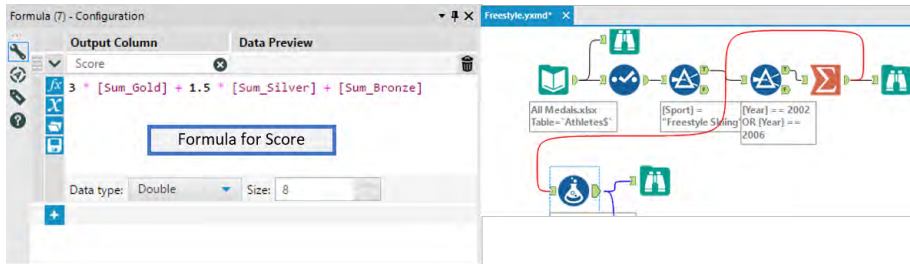
Figure 2-59 – Freestyle Skiing, Formula Configuration

We can now add another Browse after the Formula Tool to see what the data looks like.

Figure 2-60 – Freestyle Skiing, Browse configuration after formula

We see there is a new field called Score that is an unordered data set and with multiple unnecessary values. We can also see that Australia has the highest score and therefore is the answer to the original question. But for good practice, we are going to continue to build this workflow so that no interpretation is needed.

This process is going to take four steps:

1. Reorder the data based on the score field.
2. Select only the top-scoring country.
3. Remove all data other than the name of the best country.
4. Browse that data.

Like we discussed, the Sort Tool is how we reorder the data and will be our first step. We will set up our data in a descending order based on Score like we see here.
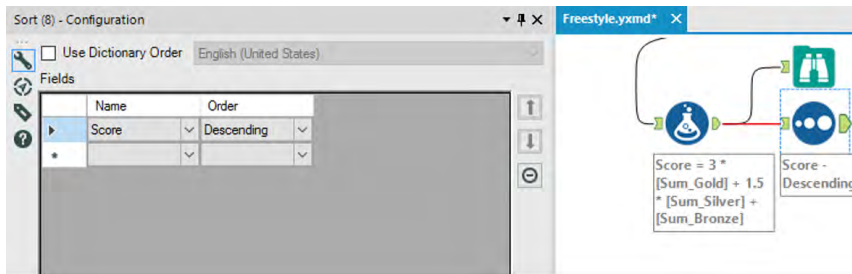
Figure 2-61 – Freestyle Skiing, Sort Configuration

Next, we need only the first record, so we are going to use the Sample Tool to keep only the Top 1 Record coming out of the Sort.
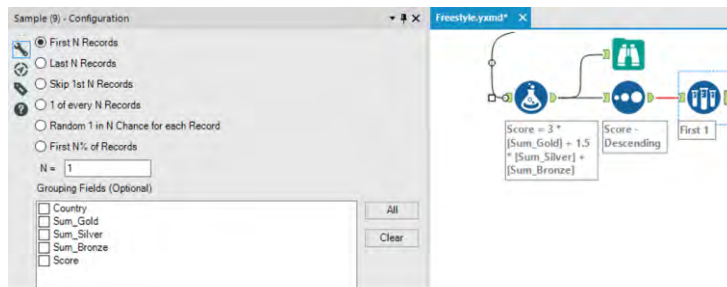
Figure 2-62 – Freestyle Skiing, Sample Configuration

We know we have fields we no longer need, so we can use a Select Tool to eliminate everything that is not the country name.
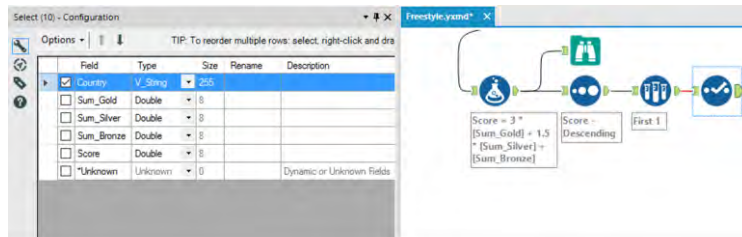
Figure 2-63 – Freestyle Skiing, Select Configuration

Finally, we can put a browse tool at the end and run the workflow to see the results.
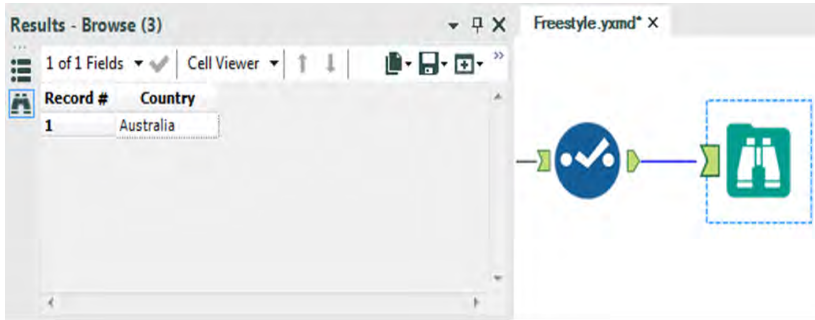
Figure 2-64 – Freestyle Skiing, Browse after Select

We could have stopped when we first saw Australia had the highest score in the previous browse tool. The reason we did not, is that when we are performing an analysis, we want our results to be perfectly repeatable. If we had interpreted the previous browse tool incorrectly then there would be no way of finding out why an error occurred. This is a problem because it makes the individual analyst entirely responsible for the answer, and anyone who checked the results could easily find the correct answer where we mistakenly picked the wrong one. Finishing the workflow in this way affords us two benefits:

- We would have a second verification that the answer was what we expected.

- Repeatability of the result so we can point to a single issue in the data preparation process that needs fixing instead of not being able to fix it at all.

Here is how the workflow would look like when complete:
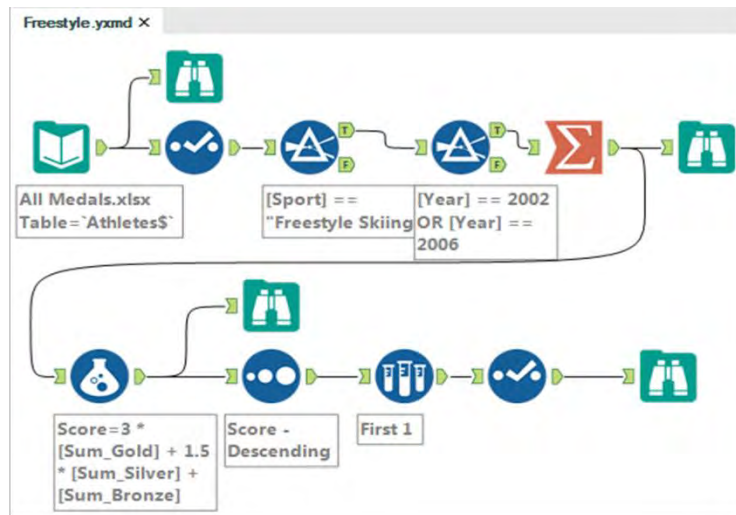
Figure 2-65 – Freestyle Skiing, Data stream after completion

2.23 Let's Tidy Things Up

	From ▾	
⊟ Send	To...	Alteryx Consultants
	Cc...	
	Subject	Let's Tidy Things Up

Hey,

That was great!

So the next thing we are going to cover is a question that requires you to produce a dataset instead of just an answer.

The goal of most data manipulation is to get the data in a more useable format. Typically, there are two formats that are most appropriate. Which you create is going to depend on what you are trying to do. The first and more relatable is to have a wide *normalized* data structure, which you can think of as "human friendly". These datasets tend to have multiple columns that have the same metrics in them but are replicated because you have a variable you want to compare across. The other has two common names that we will be using interchangeably; "tidy" is one and "denormalized" is the other. These data sets are categorized by having a single field for each variable and are often very "tall" (long). You can think of it as "computer friendly".

Let's consider the data source on the historical medal counts that we were just working on. In this case, we have a mostly denormalized dataset. Let's take that last step in creating a truly tidy dataset by converting the three columns gold, silver, and bronze into "Medal Type" and "Medal Count" so that downstream systems can process the data better.

Thanks,

This process must include at least four steps:

    1) Import the dataset.
    2) Transpose the dataset.
    3) Make sure the fields are named correctly.
    4) Export the dataset.

However, we are going to make the data cleaner and employ best practices. So our process is:

    1) Import the data.
    2) Browse the data.
    3) Make sure the data has the right type.
    4) Transpose the data.
    5) Browse the restructured data.
    6) Make sure the fields are named correctly.
    7) Remove records that say there were no medals won.
    8) Browse the data that will be exported.
    9) Export the dataset.

Let's create a new workflow and save it as *Let's Tidy Things Up*.

We need to import the same data that we used in the last example. Bring an input tool onto the canvas, navigate to where we saved this book's data, and connect to the file *All Medals.xlsx* in *Chapter 2 -- The Games> Medals*.
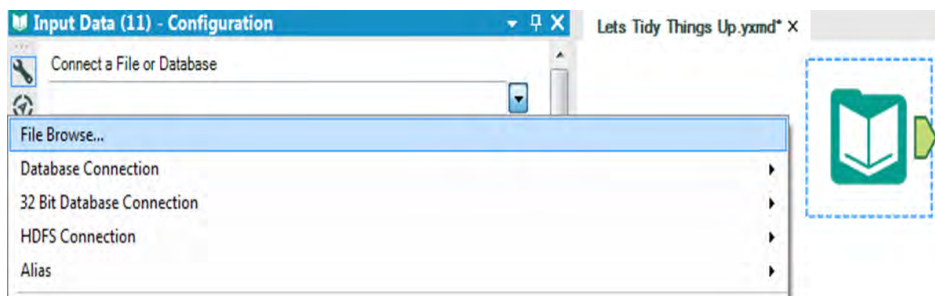
Figure 2-66 – Let's Tidy Things Up, Data Input

Now we will put a Browse and select statement following the Input Tool.
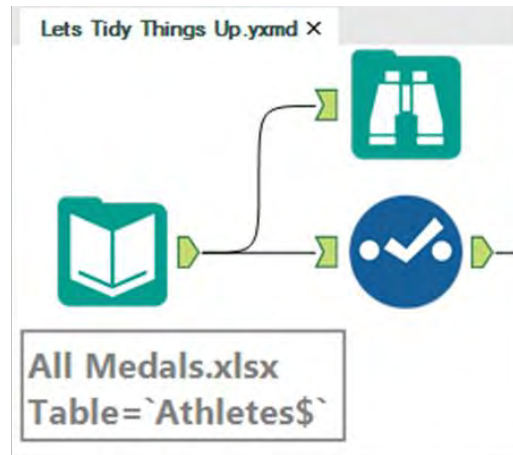
Figure 2-67 – Medals data

All of the fields are inappropriate types of what we are trying to do, so we can move directly to the transposition.

Let's run the Module to see how the data is structured.

Figure 2-68 – Let's Tidy Things Up, Browse configuration

The Transpose tool takes normalized data and de-normalizes it. If we take the data stream coming out of *Select* and pass it into a Transpose, we can make the data tidier.
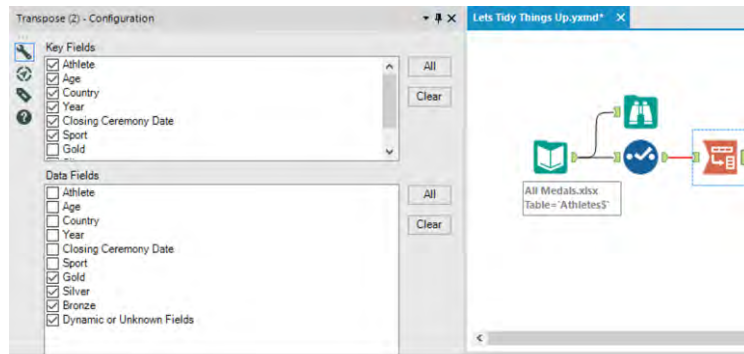
Figure 2-69 – Let's Tidy Things Up, Transpose Configuration

We want to keep all of the fields as they are except for gold, silver, and bronze. So we select all but those three fields under Key Fields and we select gold, silver, and bronze under the Data Fields. If we had wanted to drop a field entirely – say, Closing Ceremony Date – we could leave it unchecked in both lists.

Best practice is to always include a Browse after a tool that modifies the structure of a data stream. Transpose is one of these tools. Let's add a Browse to the end of the data stream and run it to see what we have.

Figure 2-70 – Let's Tidy Things Up, Browse configuration

If we compare the top three records from the new Browse to the one that came out of the Input in figure 2-66, we see that we have two fields called *Name* and *Value* and no longer have the fields *Gold, Silver,* and *Bronze*. We also notice from *Athlete* to *Sport*, all fields are identical to the first three records in the original dataset. This is because we replicated them for each column we created.

This is one of the reasons that tidy data is not particularly human readable but is highly computer readable. Since all of the information is displayed in each record and there is only a single column to work on, interactive front-end software can work very fast with the data.
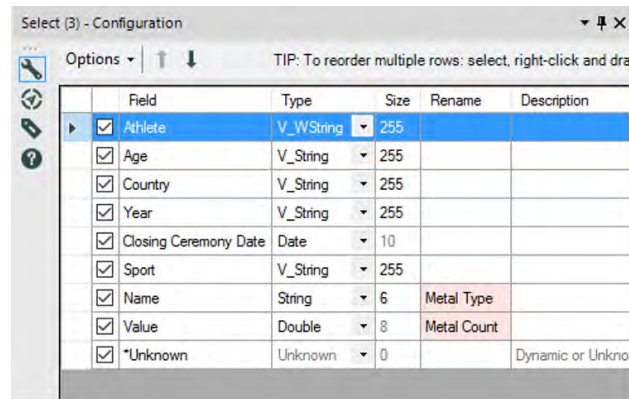
Figure 2-71 – Let's Tidy Things Up, Select Configuration

Making this data truly tidy would mean we need to rename *Name* and *Value* to names that give better context to the field. Add a Select statement, and rename the Name and Value fields *Medal Type* and *Medal Count*, respectively.

We know we have rows that say zero medals were won by looking at the values in the last Browse tool we created. We are going to filter those data points out by adding a Filter tool after the Select.

Our goal is to filter out any records that have zero medals. We are filtering on a numeric field for the first time, which means we should use the Basic Filter to learn about the syntax. The configuration is as shown in the following figure.
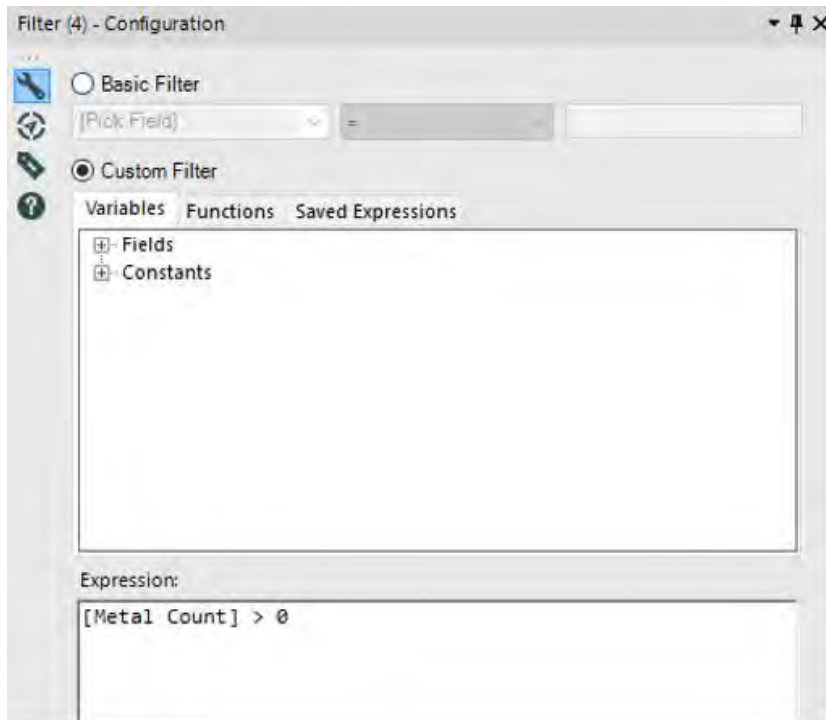
Figure 2-72 – Let's Tidy Things Up, Filter Configuration

We can see how if we select Medal count, we have different options in the operator drop-down. This is because numeric fields allow different comparison methods than string fields.

We want to select greater than – ">" – and type "0" in the text box. When we look at the Expression below, we see that it says **[Medal Count] > 0**. This is because we do not put numeric values in quotes. Alteryx recognized that when we selected a numeric field in the basic filter drop-down, the "0" we typed in meant the number 0 and not the string 0, so it put the numeric value into the formula.

The last step involves two tools: the *Browse* tool and the *Output Data* tool.

Best practice dictates that we put a Browse before every data output so that we do not need to open the file to make sure we created it correctly.

We now add a Browse tool to the end of the data stream and also add an Output Data tool. We are going to write the file to the same folder we have saved the *Let's Tidy Things Up.yxmd*.
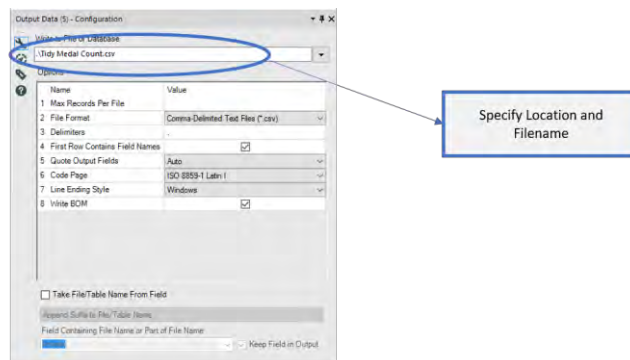
Figure 2-73 – Let's Tidy Things Up, Output Data Configuration

To do this, we are going to type *.\Tidy Medal Data.csv* in the text box labeled *Write to File or Database*.

We just used a relative file path. Which allows us to reference files in relation to where we currently are. Some basics of relative paths are ".\", which means the current folder. "..\" means the parent folder (the folder that our current folder is in). ".\Folder Name\" will move our file into a folder below where we have the workflow.

We do not necessarily need to use relative paths, but if we are sharing Alteryx files, it is very beneficial to do so. We can use absolute paths (full file locations) by pasting them into this box or navigating through them in the File Browse option.

If we run the module, we can see what the transformed dataset looks like. This ensures that the information written into the .csv was correct.

Figure 2-74 – Let's Tidy Things Up, Final Browse configuration

Here is how the *Let's Tidy Things Up* data stream looks on completion.

Figure 2-75 – Let's Tidy Things Up, Data stream after completion

2.24   Modern History

| To... | Alteryx Consultants |
| Subject | Modern History |

Great!

Now that you are getting the sense of tidy data, let's go in the opposite direction and create a normalized dataset.

How about we create a nice table with countries alphabetically on the left, a column for each year in the dataset ordered from longest ago to most recent, and a historical total medal count in the cross section?

It should be fun,
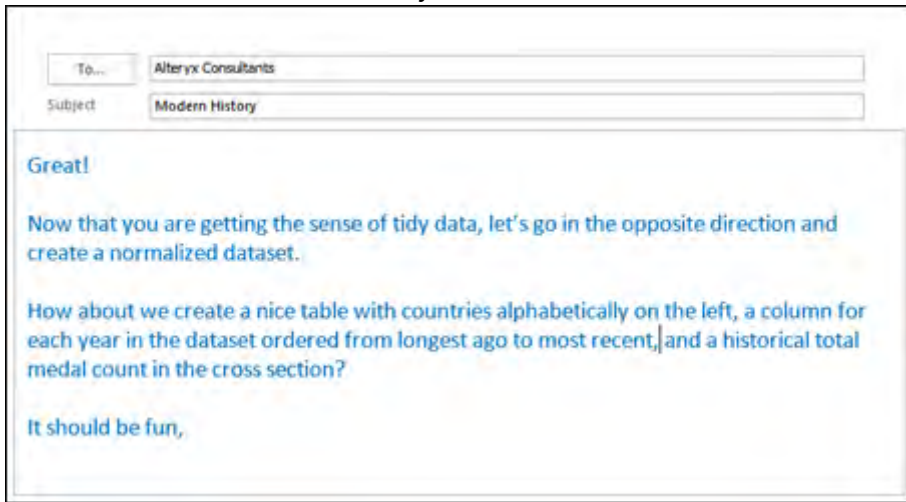
Notice that there is considerably less context built into this email. We often get very sparse information from people, and they will assume we know the context. In this case, it was assumed we were talking about the medal data that we have been working with during the training so far.

This is a much more complicated process than the last exercise, but that is only because the data was set up very well for what we were doing last time, and it isn't here.

We are going to be connecting to the same data source that we have been using, but we are going to use a shortcut in the connection process. Open a new workflow and save it as *Modern History.yxmd*, but make sure that *Let's Tidy Things Up.yxmd* is still open.

Click on the data input in *Let's Tidy Things Up* and copy it. Move over to the Modern History canvas, and paste what we have copied. We see that the input has been copied over and we do not need to recreate the connection.
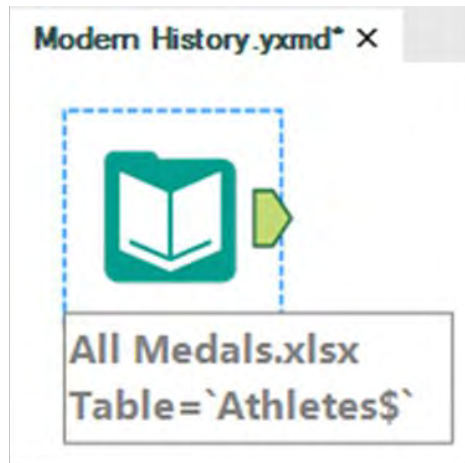
Figure 2-76 Modern history, Input Data

Best practice will once again bring in Browse and Select tools. But since we know from past experience what the data looks like and how it is read in, we will move directly to the next step.
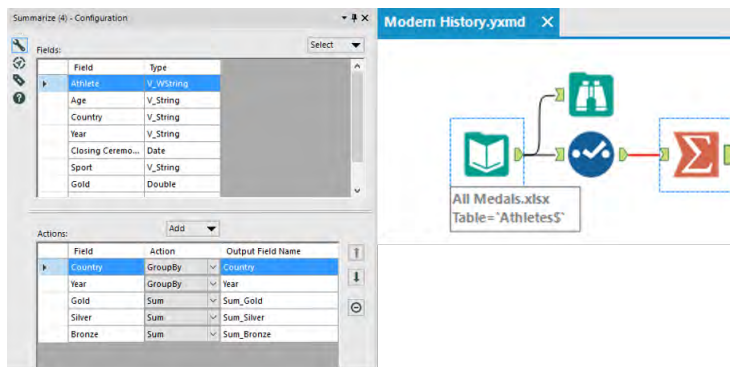
Figure 2-77 – Modern History, Summarize Configuration

We know that this data is too granular for our desired result. So we can summarize it. Based on the email, we know the only information we will need in the end is the country, year, and something to do with the medals.

So when using the Summary tool, we can group by the country and year fields and take the sum of each of the medal counts to take our first step down this path.
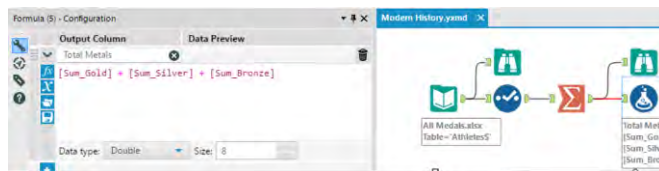
Figure 2-78 – Modern History, Filter Configuration

We now add a formula that creates a *Total Medal* count by adding the gold, silver and bronze fields for each record. (Remember that we used a Summarize tool so we should have a Browse tool.)
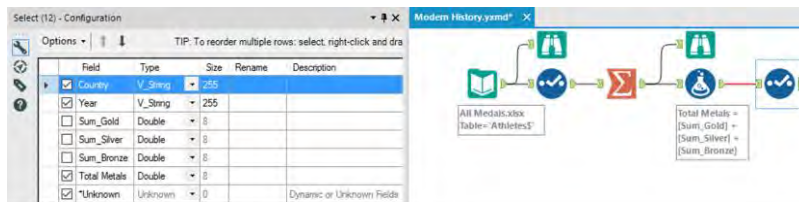
Figure 2-79 – Modern History, Select Configuration

We can now add a select statement that will allow us to keep only the Country, Year, and Total Medals fields, which we will use to create the table.

We know we need a historical medal count, which means we are going to need to take the running total along with the country and year. But because Running Total is a tool where order matters, we need to sort the data.

We can sort the Country and Year in ascending order to help us in two places: Initially, this will help because we are creating the order for the Running Total, but it will also help us

with the order of records and columns when we normalize the data set.

Figure 2-80 – Modern History, Sort Configuration

Now that we have ordered the data, we can create the Running Total for each country across years. To do this, we *Group By* Country and *Create Running Total* on Total Medals. What this will do is create the running sum of Total Medals down the data set (as time increases) and have that count restart every time a new country shows up.

Figure 2-81 – Modern History, Running Total Configuration

Let's take a look at what we have created so we can get a better sense of what the process so far has done.

**Figure 2-82 – Modern History Browse
Configuration after Total**

If we add a Browse tool and run the workflow, we can see that we have an alphabetical list of countries with a record for every year they won a medal. We can also see the year is increasing as we move down the list within a country. We then see the Total Medal count for that year and the running total of medals that the country has won going from one year to the next in a field called RunTot_Total Medals
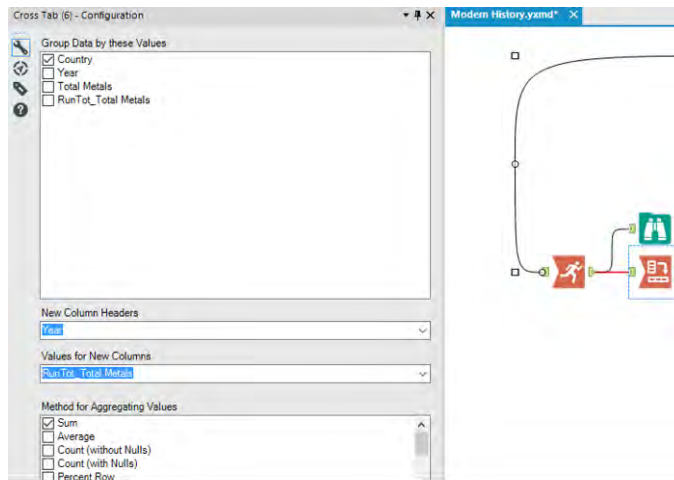
Figure 2-82 – Modern History, Cross Tab Configuration

The next step in this process is to convert the data into a Cross Tab. If we add the Cross Tab tool to the end of the data stream and apply the settings in the above image, we will be close to our goal.

Let's add a browse tool and see what we have so far.

Record #	Country	2000	2002	2004	2006
1	Afghanistan	[Null]	[Null]	[Null]	[Null]
2	Algeria	5	[Null]	[Null]	[Null]
3	Argentina	20	[Null]	69	[Null]
4	Armenia	1	[Null]	[Null]	[Null]
5	Australia	183	185	341	343
6	Austria	4	24	32	62
7	Azerbaijan	3	[Null]	8	[Null]
8	Bahamas	11	[Null]	13	[Null]
9	Bahrain	[Null]	[Null]	[Null]	[Null]
10	Barbados	1	[Null]	[Null]	[Null]

Figure 2-84 – Modern History Browse Configuration
Cross Tab

The result in the image is close but not exactly what we wanted. We get the correct running totals in the years that each country won medals, however, we get nulls in the years that they did not.

What we need to do now is create a series of formulas that replace the nulls with zero or the previous value as appropriate. Since we need to create formulas. we are going to need the formula tool; but this time, we are going to need to create seven similar calculations because we need to replace the values in seven different fields.

Let's think through these formulas. We want to change the cell only if it is null. If the column we are fixing is 2000, then it should be replaced with 0, and if the column is not 2000, it should be replaced with whatever is in the previously fixed column.

For those familiar with conditional statements, the syntax for an if-then statement is:

IF b1 THEN x ELSEIF b2 THEN y ELSE z ENDIF

For those unfamiliar with conditional statements, the concept is: Given a true or false (Boolean) expression, the calculation should do one of two things. The logic is if something is true, then do that; else, if the previous is false and something else is true, do the second option; else, do the default.

The other thing we need to know in writing these formulas is the test to see if something is null. The function used is:

IsNull(x)

Both of these syntaxes are under the functions tab in the Formula tool if we need to reference them.

The formulae that we need are:
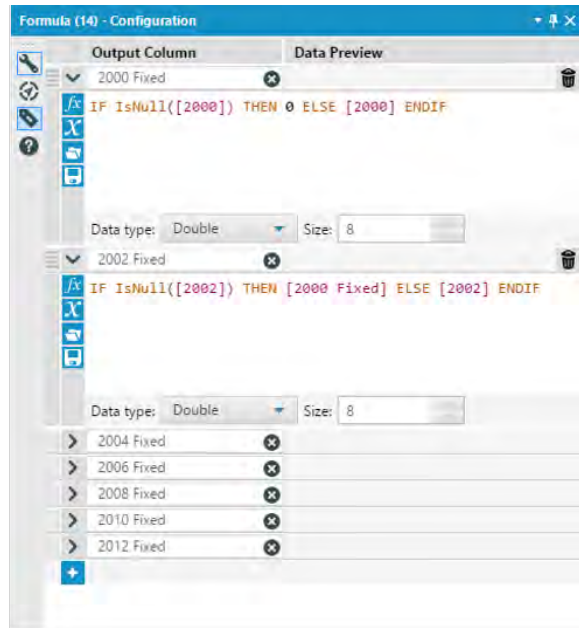
Figure 2-85 – Modern History, Formula Configuration

Add a Formula tool to the end of the data stream, and add the seven formulas we see here with corresponding field names. We can also add a browse tool after that to see what we have created.

We can see that we have two sets of fields: those with the original sparse data, and those with the new dense data.

Figure 2-86 – Modern History Formula List

The next thing that we need to do is remove and rename the columns that we have, so add a Select tool to the end of the data stream.
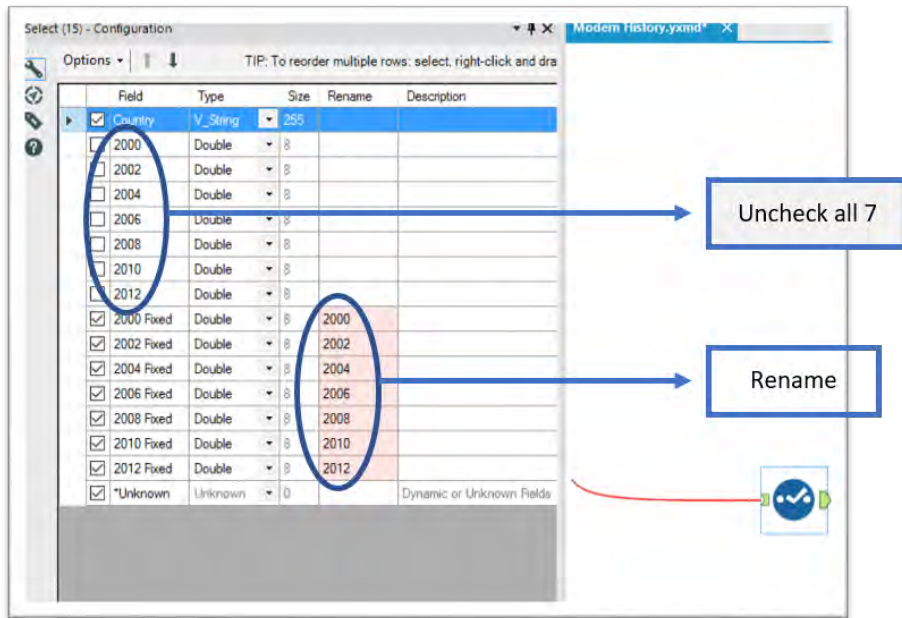
Figure 2-87 – Modern History, Select Configuration After Formula

Now the data has finished being prepped. We need to write it out, which we know because we were asked for a data set and not a specific answer. We should add a Browse tool and an Output Data tool to end the data flow. Save the output as Historical Medal Count.csv.
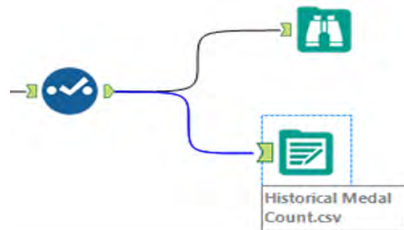
Figure 2-88 – Modern History Output

After doing these steps, the final workflow is as shown in the below figure.
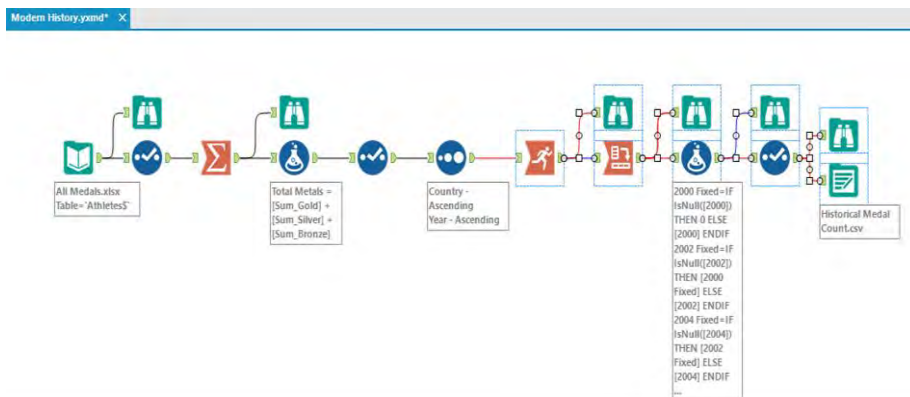
Figure 2-89 – Modern History, Data Stream when Complete

2.25   Brains vs Brawns

To...   Altryx Consultants

Subject   Brains vs Brawns

Awesome!

We only have one more basic skill to go over before we test to see how much you know.

Combining data.

I have been working with the medals dataset for a while and it is interesting to see how it compares to different metrics.

I think we should compare the medal counts to Nobel Laureates from each of the countries.

Let's put together the data to see what the relationship between the count of medals and Nobel Laureates was since 2000. (We will map the country of Nobel Laureate birth to the country that won the medal).

Thanks,

Since we are combining data, let's revisit the analogy presented in the preface. When we look at a river, we see there are tributaries all along its length. Each of these tributaries may have gone through different terrain and could have started as very different sources. When they come together, they add whatever they carried along with them into the river they form.

To relate it to the task at hand, tributaries are branches of our data stream that come together, and when they come together, we have a richer data stream because we have the information that comes from everything contributing to it.

We are going to start by prepping the medals data and preparing them to be joined. We'll create a table with two columns called Country and Medal Count.

In order to do this, we are going to follow the following steps:

1. Import data
2. Transpose and rename the columns so that the data is tidy.
3. Filter out the 0 medal records
4. Summarize the data so that we only have one record per country and the total medal count.
5. Rename the medal count column *Total Medal Count*.

Since we have covered the tools and the concepts used in this exercise in previous exercises, overall flow should look familiar. Please rebuild the following workflow with the following configurations.

The properties windows for each of these tools as well as the data stream that is produced are shown in the following figures.
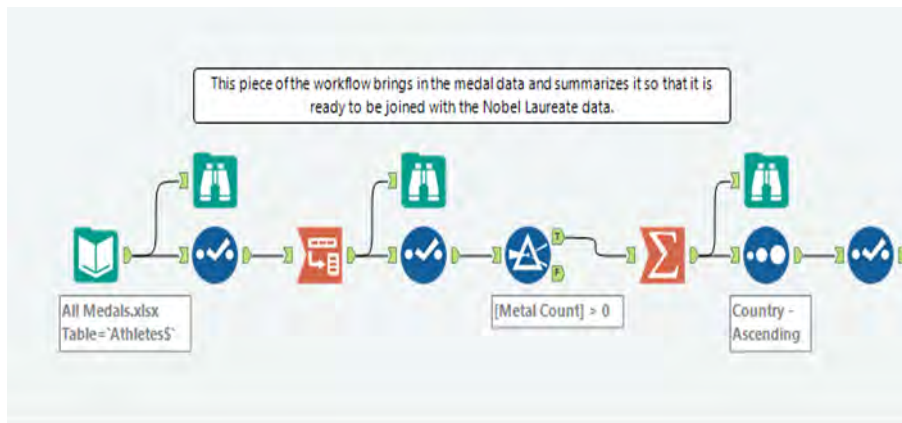
Figure 2-90 – Brains vs Brawns data stream

Figure 2-91 – Brains vs Brawns, Initial Steps - Select
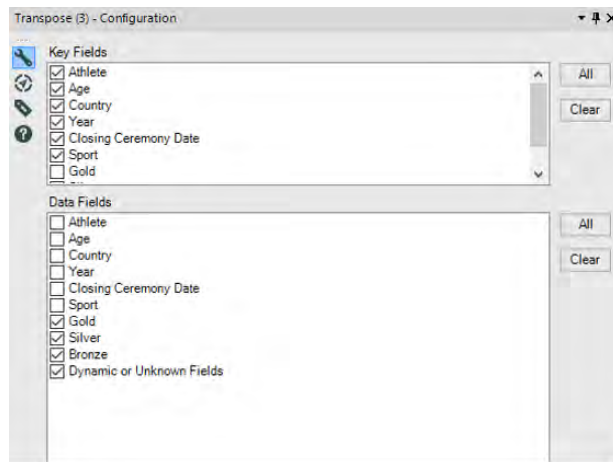
Figure 2-92 – Brains vs Brawns, Initial Steps - Transpose
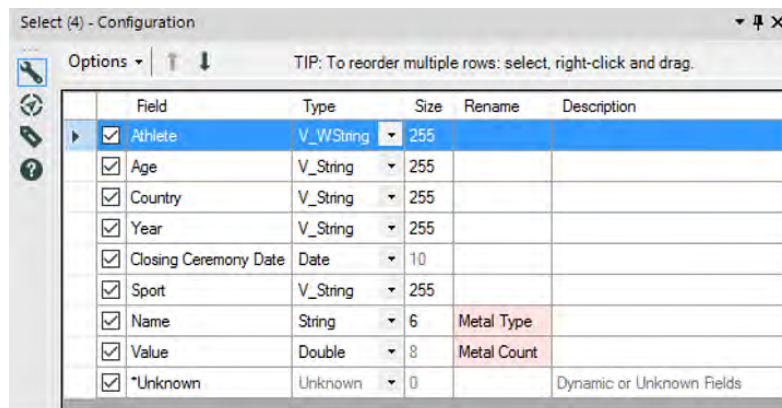
Figure 2-93 – Brains vs Brawns, Initial Steps - Select
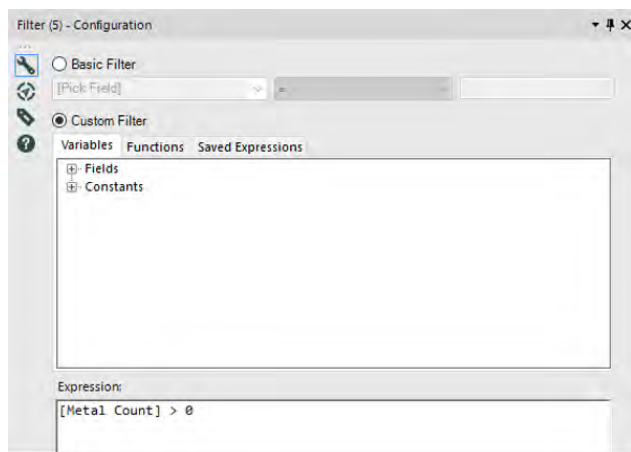
Figure 2-94 – Brains vs Brawns, Initial Steps - Filter
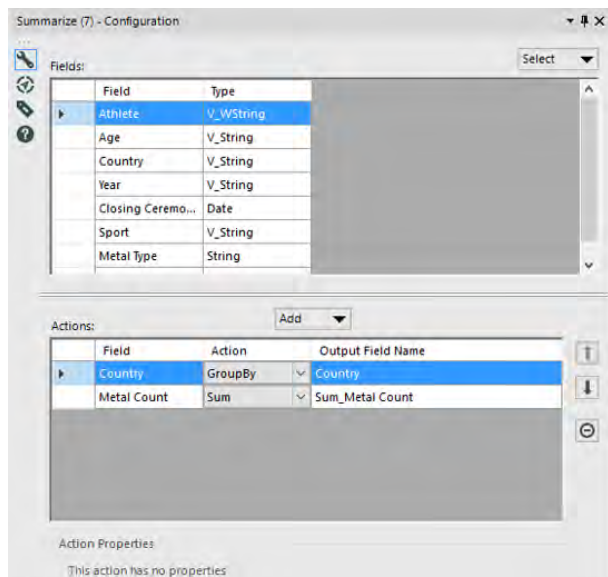
Figure 2-95 – Brains vs Brawns, Initial Steps - Summarize
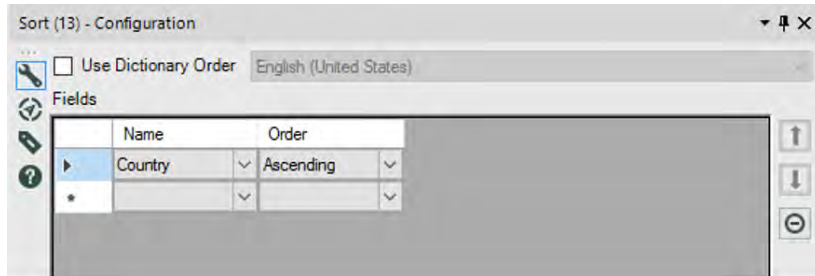
Figure 2-96 – Brains vs Brawns, Initial Steps - Sort
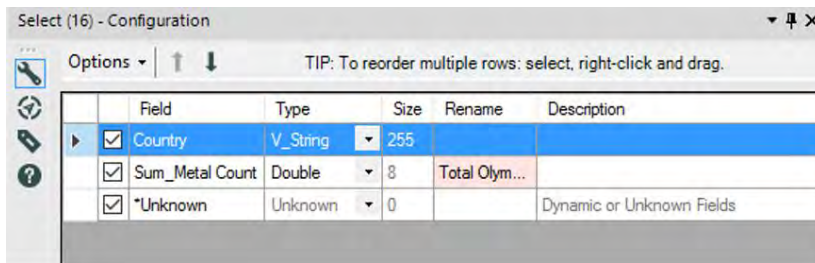
Figure 2-97 – Brains vs Brawns, Initial Steps - Select

Now that we have the data in the above stream prepared to be combined, we should prepare the other contributing data stream.

Let us open the file called *Nobel Laureates.csv* in the folder Chapter 2 -- The Games > Nobel Laureates. (Remember that we should always bring in a Browse and Select Tool with an input.)
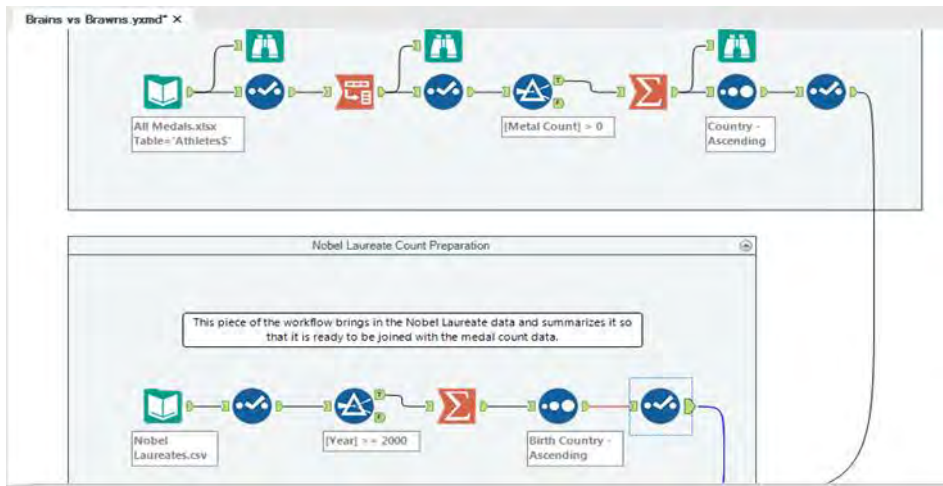
Figure 2-98 – Running unrelated analysis simultaneously

Notice that we now have two completely separate workflows. This ability is often a useful feature because we can run unrelated analyses at the same time, which aids in testing and in conditional application development.

If we run the workflow, we can look at the structure of the Nobel Laureates dataset. Here, we want to make sure that the field we plan on joining (Birth Country) is the same type as County in the medal data stream.

Figure 2-99 – Nobel Laureates – Browse Configuration

Figure 2-100 – Brains vs Brawns, Nobel Laureates- Select

Now that we know what the data structure is and that it parallels the medals file, we can start our preparation for the join.
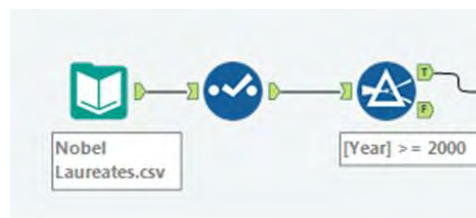
Figure 2-101 – Nobel Laureates –
Preparing to join Medals

We know we want to limit this data to years starting in 2000. One way we can do this is to convert Year to a Double Type and set up a filter to be **[Year] >= 2000**.

Since we only need to know what the total number of Nobel Laureates there were for each country of birth, we can summarize the data.

Figure 2-102 – Nobel Laureates – Browse after Summarize
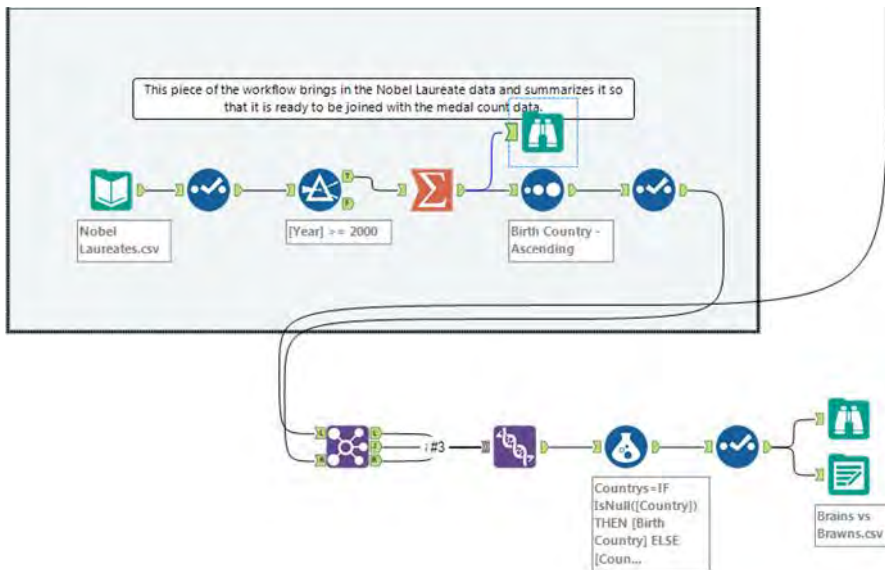
Figure 2-103 – Nobel Laureates – vis a vis All Medals

We see that we have a list of countries and a count of the number of Nobel Laureates. However, it is unclear what the number is because the field is called *Count*. We should rename it *Total Nobel Laureates*.
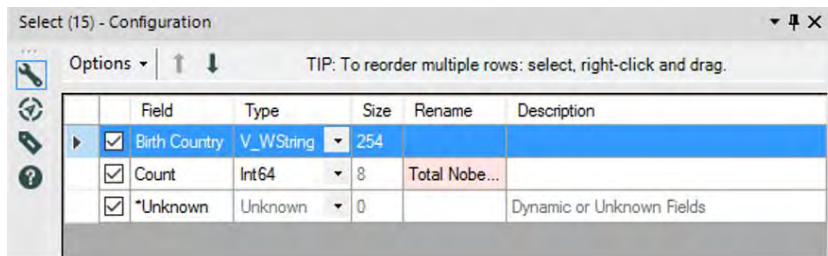
Figure 2-104 – Brains vs Brawns, Nobel Laureates- Select Configuration

We now have two data streams ready to be merged. We want to align the two datasets so that matching countries from each of the data streams share the same record, which means we want to join the data. Because we don't want to lose any data

points if we have countries in one dataset but not the other, we will want to unite the three outputs from the join into a single data stream.
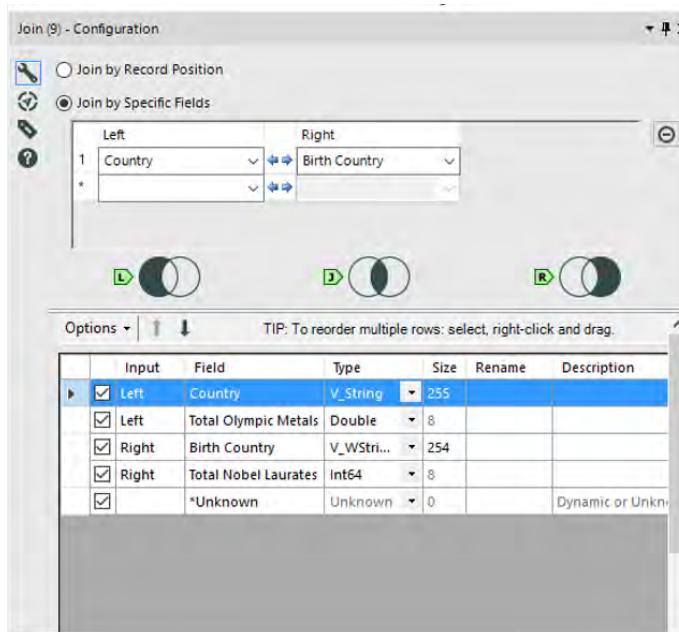
Figure 2-105 – Brains vs Brawns, Nobel Laureates- Join Configuration

We want to join on Country field from the Left (Input L) with Birth Country field from the Right (Input R).

It is important in this instance that we keep both joining fields because we intend to combine all three outputs in the next step. However, if this was not our intention, we could have removed the joining field from one of the two inputs.

Best practice is to give useful names to every connection that enters a multiple connection anchor.

Thus, we can see in the following image that we have relabeled the connections from *#1*, *#2,* and *#3* to *Left*, *Join,* and *Right*.
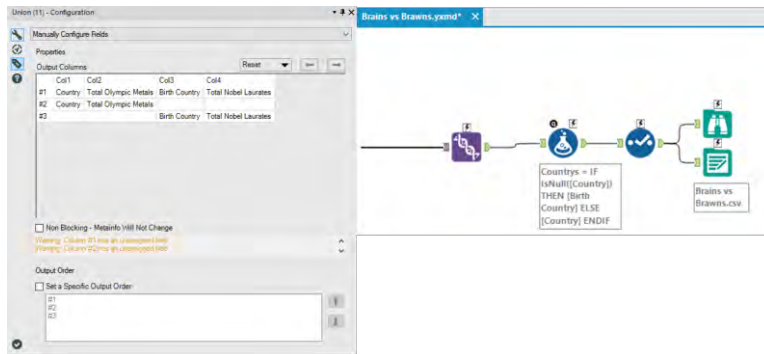
Figure 2-106 – Brains vs Brawns, Nobel Laureates- Union Configuration

Since we are doing a union of three output streams of a Join tool, we know that we will have matching column names. This allows us to use the *Auto Config by Name* setting for the Union tool and leave the rest of the defaults.

We need to add a browse tool again as we have just altered the structure of the data. This is to make sure the data looks the way we expect. Notice that we are doing this after the Union and not the Join. That is because when we are combining the three output streams of a Join tool using a Union, we are performing a single logical step called an outer join. Because this is a single step, we know that we should check both tools if an issue arises.

We are getting close to our goal; however; the data stream is also starting to get complex. So we should take a minute to annotate what we have so it will be easier to follow later. We are going to add *Tool Containers* and *Comments* to the two contributing data streams so we can easily identify different parts of this data stream. We can create the comments and containers like we see in the next image.

This piece of the workflow brings in the Nobel Laureate data and summarizes it so that it is ready to be joined with the medal count data.
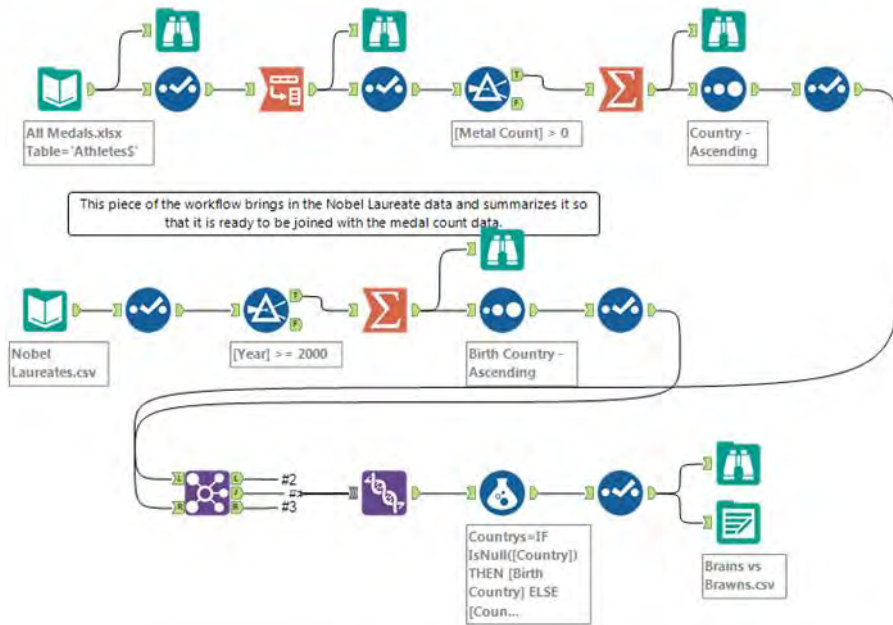
Figure 2-107 – Nobel Laureates – Comments and Containers

We can now drag the appropriate tools into the tool containers so the data stream is easier to understand.
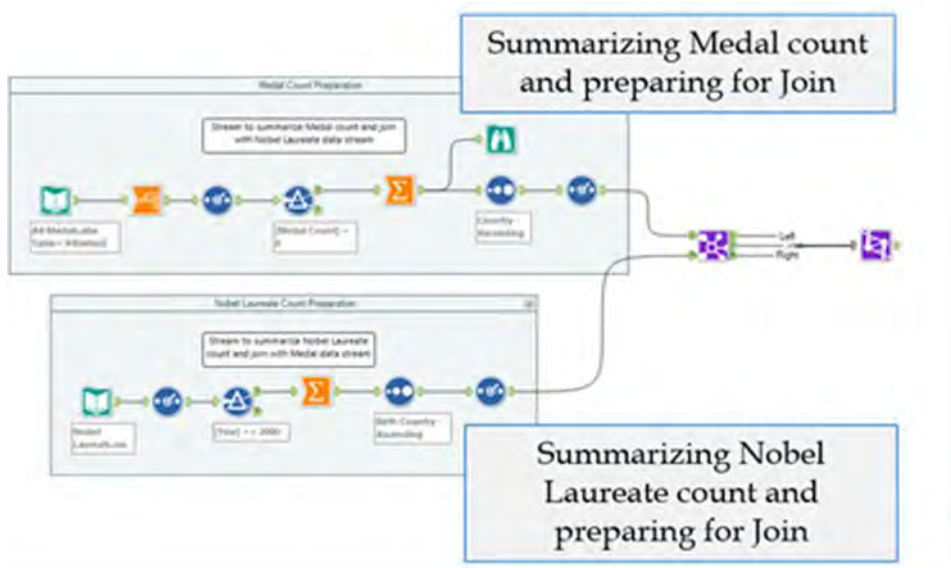
Figure 2-108 – Nobel Laureates –Output Stream with Comments

Looking at the data stream this way is helpful, but if we click on the arrows at the top-right corner, we can condense what we are looking at.
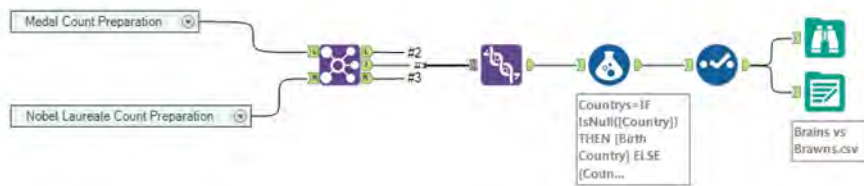
Figure 2-109 – Nobel Laureates –Simplified Output Stream

Now we can easily see the medal count preparation and the Nobel Laureate count preparation as two single processes instead of a series of tools. Now that we have made the data stream easier to understand, we should finish building the workflow.

Figure 2-110 – Nobel Laureates –Browse Configuration

We can observe from the Browse that the country names matched the names in both the Country and Birth Country fields. Let's create a conditional formula, **Countries** with the formula: **IF IsNull ([Country]) THEN [Birth Country] ELSE [Country] ENDIF**. This will take the Country value unless it is null and the Birth Country if it is.

Now we only need to clean up the data and export it to a .csv file. Add a Select tool with the following configuration, and export the file to *Brains vs Brawns.csv*.

Figure 2-111 – Brains vs Brawns, Nobel Laureates- Select Configuration

The final workflow is as shown in the following figure.

Figure 2-112 – Nobel Laureates –Complete Workflow

2.26   How are we doing?

	To...	Alteryx Consultants
Send	Cc...	
	Subject	How are we doing?

Hey,

I just got a call from the Ad Agencies marketing team. They have got some shopper marketing data over a period of time and the forecast for the same.

They want us to help them with the following:
   a.   Get all the actuals into one dataset
   b.   Find the totals, Spend values for Products, Retailers and Ad Campaigns

And further,
   a.   Find the spend for Products from forecast. Compare the values from the respective Part A.
        If the difference in spend is 1000 then provide flags to highlight the forecast as over or under estimate
   b.   Find the spend for Retailers from forecast. Compare the values from the respective Part A, Arrange values from High to Low
   c.   Find the spend for Ad Campaign from forecast. Compare the values from the respective Part A.
        If the difference in spend is 2000 then provide flags to highlight them as over/under estimate.

I am into meetings for the rest of the afternoon, I cannot show them the turn around, but, I am sure you know enough to get these in place.

If we can get this back to them in next couple of hours, I think it would guarantee that they go with us.

Thanks

CHAPTER 3
Unisex Baby Name

From ▾	
To...	Alteryx Consultants
Cc...	
Subject	Gender Swapped
Attached	📄 Unisex Names.txt

Hey,

A major newspaper is considering writing an article on the change in assignment of names to different sexes, and they would like our help finding out if there is anything definitive that they can say.

They provided the attached file that has unisex baby names from 1880 to 2013 and the count of children given each name by sex.

This is a pretty interesting dataset, so I am going to work on this with you.

The first things they want to know are: Are there any names that between 1880 and 1889 were completely assigned to one sex and then completely assigned to another between 2004 and 2013? If so, which were they?

Thanks,

3.1 Tools & Concepts

Tools	Concepts
Imputation	Handling Nulls
Multi-Field Formula	Multiple Record Calculations
Multi-Row Formula	Repetitive Calculations
Text Input	Ad Hoc Data
Data Cleansing	Problem Analysis
Generate Rows	
Multi-Field Binning	
Select Records	

3.2   Imputation

Figure 3-1 – Imputation	The *Imputation* tool replaces a specific value in numeric fields. It is most commonly used to handle Null values in formulas.		
	Group	**Input**	**Output**
	Preparation	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with a numeric field.

Output: A data stream with at least one imputed field.

The tool will only modify an individual value and cannot be used to impute a range of values, such as all negatives.

Properties Window:

The *Imputation Configuration* window has five components.
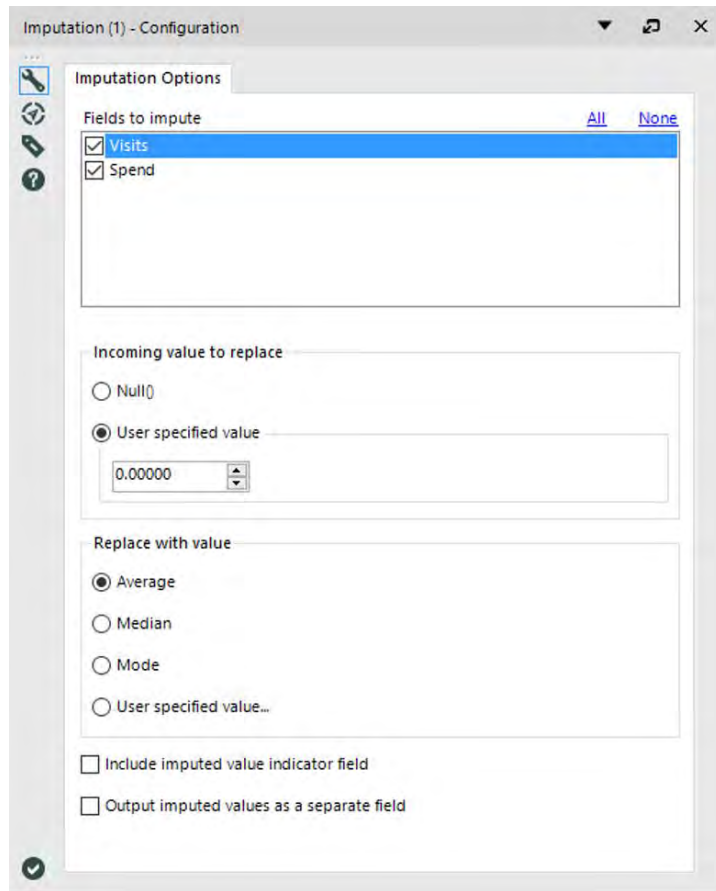
Figure 3-2 – Imputation Configuration

- *Fields to Impute* is a list of numerical fields from the incoming data stream that one can choose to impute.
- *Incoming value to replace* allows us to choose to replace nulls or a specific numeric value that exists in the incoming dataset.

- *Replace with value* allows us to replace the value selected in *Incoming value to replace* with either the average (mean), median (middle), mode (most common), or a custom value that we select.
- *Include imputed value indicator field* creates a field for each of the *Fields to impute* with ones and zeros, where 1 indicates that the data was imputed and 0 that it was not.
- *Output imputed values as a separate field* creates a new field for each of the Fields to impute instead of replacing the values in the original field.

3.3   Multi-Field Formula

	The *Multi-Field Formula* tool provides the ability to create a formula that will be reused across multiple fields.		
	Group	**Input**	**Output**
Figure 3-3 – Multi-field formula	Preparation	See below	See below

The formulas created here are arbitrarily complex, and thus can significantly slow down the data stream. Care must be taken to make sure the output field created has a field type compatible with the result created. // is the comment character.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with at least two fields of the same type.

Output: The original data stream with fields modified or original data stream with a new field for every selected field from the input.

Application questions can be connected to the *Black Question Anchor* for "Use Answers" in this tool.

Properties Window:

The *Multi-Field Formula Configuration* window has five main components.
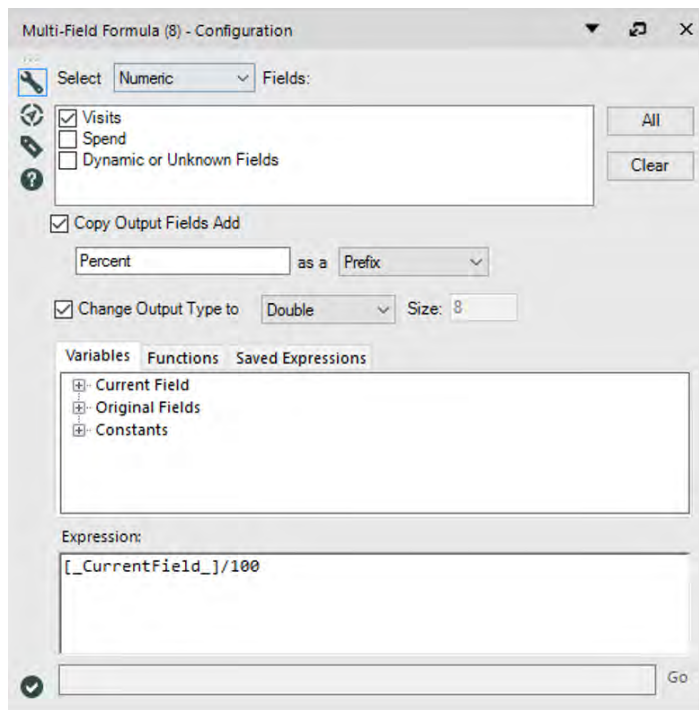
Figure 3-4 – Multi-Field Formula Configuration

- *Select … Fields*: populates the list with the fields that match what we have selected in the drop-down. The calculation is performed on each of the fields selected from the generated list.
- *Copy Output and Add … as a …*: when selected, creates a new field as the output instead of replacing the ones that we are performing the calculation on. *Decimal*, in the

example, will be appended to the field name as a Suffix or Prefix.

- *Change Output Type to … Size …*: when selected, we can change the metadata of the output that we create.
- The last two sections are similar to the formula window that we are familiar with, except that since we are working across multiple fields, we have the ability to use the *_CurrentField_* in our calculations.

3.4 Multi-Row Formula

Figure 3-5 – Multi-row formula	The *Multi-Row Formula* tool provides the ability to create a formula that will reference other records.		
	Group	**Input**	**Output**
	Preparation	Any data stream	See below

The values are updated by moving down the records in order to modify the current record based on the result of the formula in the previous record. The formulas we create here can be arbitrarily complex, and thus can significantly slow down the data stream. Care has to be taken to ensure the output field created has a file type that is compatible with the result being created. The *//* is the comment character.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the top *Black Question Anchor* for "Use those Answers" in this tool.

> *Output*: The original data stream with a field modified or the original data stream with a new field.

Properties Window:

The *Multi-Row Formula Configuration* window has six core components.
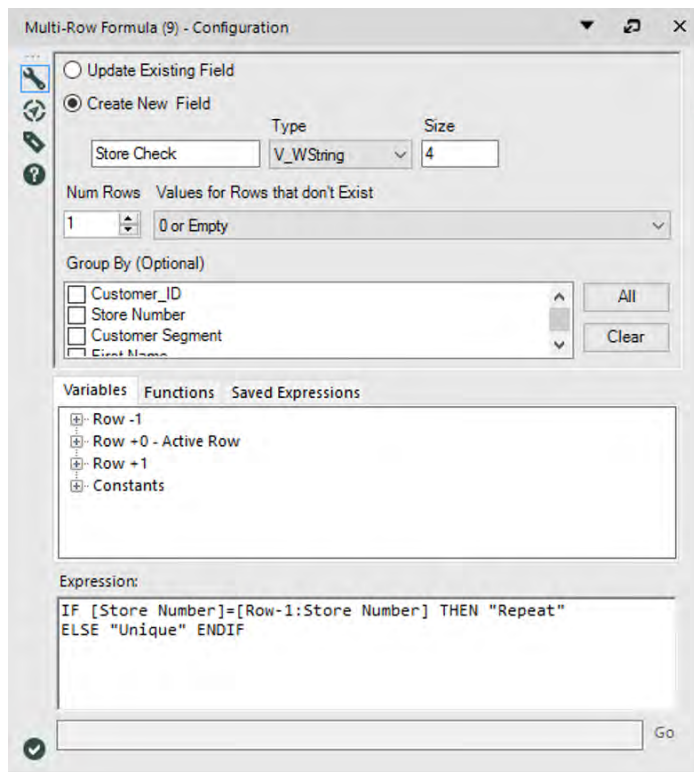
Figure 3-6 – Multi Row Formula Configuration

- *Update Existing Field* allows us to replace the entries in the field selected in the list below.
- *Create New Field* is the other option for the output, which allows us to create a new field instead of updating the

current one. With this selection, we need to name the field and set the metadata of the field.

- *Num Rows* changes the number of rows before and after the current row we have access to in the formula.
- *Values for Rows that Don't Exist* allows us to define the behavior for records that do not actually exist. (In this case, for the first record, there is no *[Row-1: …]*. So the field is replaced by a 0 since the field is numeric.)
- *Group By* allows us to reset the calculation based on changes in values in the fields we select.
- The bottom two sections are similar to the formula boxes we are used to seeing except that we have the added ability to refer to field values in different rows.

3.5   Text Input

Figure 3-7 – Text input	The *Text Input* tool allows the creation of datasets to be used by typing in the data.		
	Group	**Input**	**Output**
	In/Out	None	Data stream

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Text Input Configuration* window allows us to create a data source that is internal to the module. The configuration has five options.
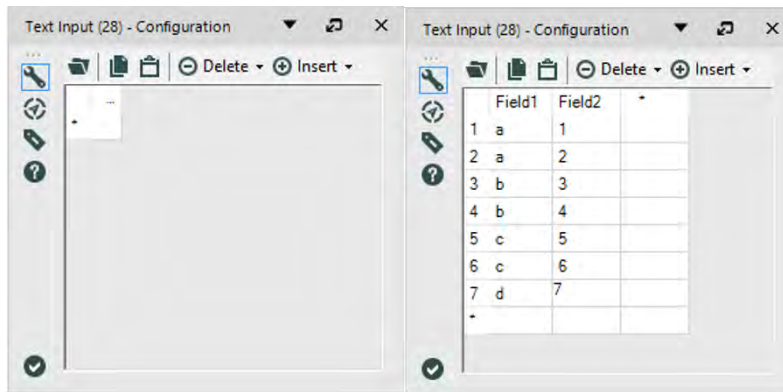
Figure 3-8 – Text Input Configuration

- *Import* allows us to import a dataset.
- *Copy* allows us to copy the cells we highlighted in the table.
- *Paste* allows us to paste values into the table.
- *Delete* allows us to delete everything, or the row or column we have selected.
- *Insert* allows us to create a row or column in the data set.

An example of what this would look like is provided in the previous image.

3.6 Gender Swapped

From ▾	
To...	Alteryx Consultants
Cc...	
Subject	Gender Swapped
Attached	Unisex Names.txt

Hey,

A major newspaper is considering writing an article on the change in assignment of names to different sexes, and they would like our help finding out if there is anything definitive that they can say.

They provided the attached file that has unisex baby names from 1880 to 2013 and the count of children given each name by sex.

This is a pretty interesting dataset, so I am going to work on this with you.

The first things they want to know are: Are there any names that between 1880 and 1889 were completely assigned to one sex and then completely assigned to another between 2004 and 2013? If so, which were they?

Thanks,

Let us refer to the *Unisex Names.txt* file in the folder *Chapter 3 – Unisex Names.* It is a tab-delineated file. We need to make sure the file pop-up window looks like the image represented here. Let us save the workflow as *Gender Swapped*.
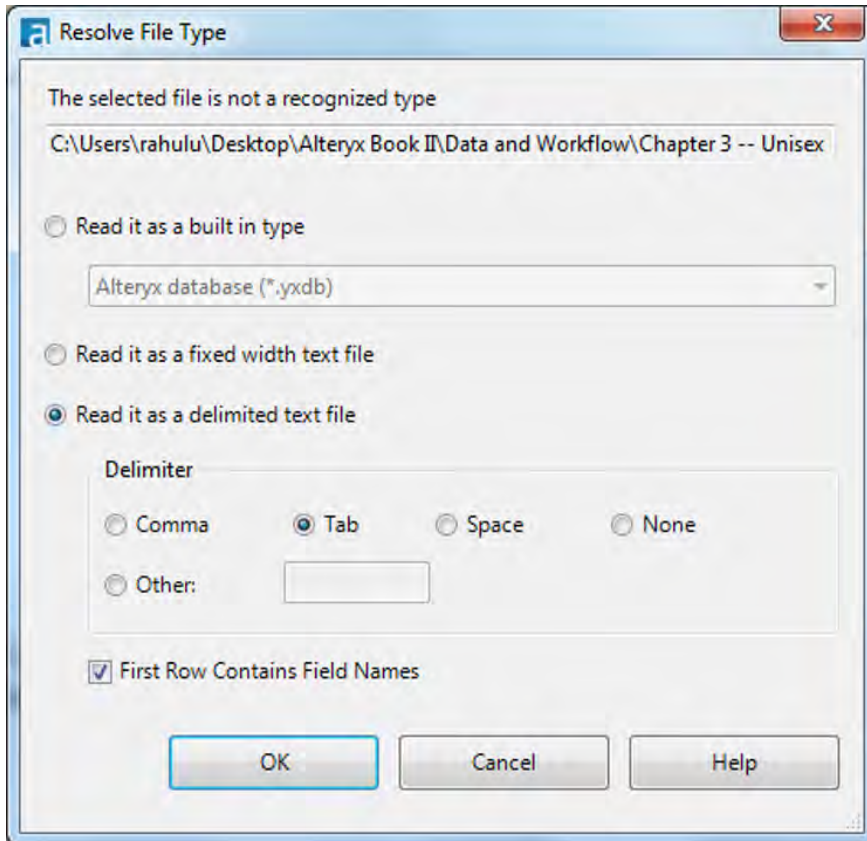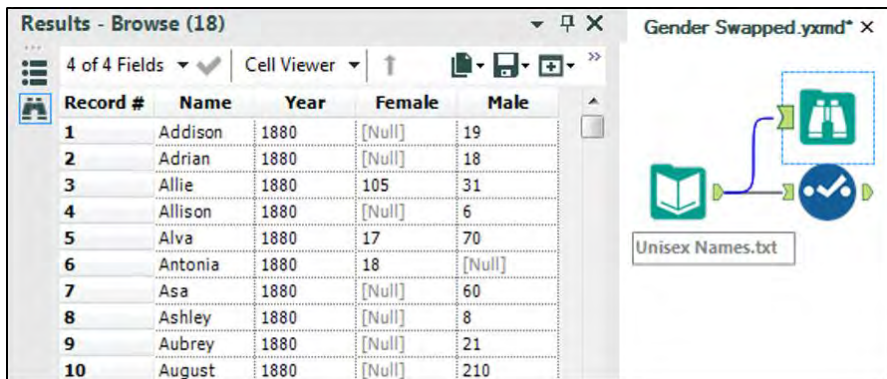
Figure 3-9 – Tab delineated pop-up window

Add a *Browse* tool and a *Select* tool to the input. After running the workflow, we should be able to see the records in the file.

Figure 3-10 – Gender Swapped Browse

If we take a look at the data, we will see that some numeric fields have *nulls* in them. Whenever values are missing, Alteryx fills those cells with a "[Null]" placeholder. We could leave these values as nulls; however, it is easier to work with numeric fields when they do not have nulls in them. Instead, what we will do is impute the nulls to 0s.

Unfortunately, Alteryx did not recognize these fields as numeric, so we need to make modifications in the *Select* tool so that *Female* and *Male* are converted into *Double*.

Since we know we need to work with two 10-year periods, we should also convert *Year* to *Double* because it will make our formulas simpler and faster when we filter and flag the data.
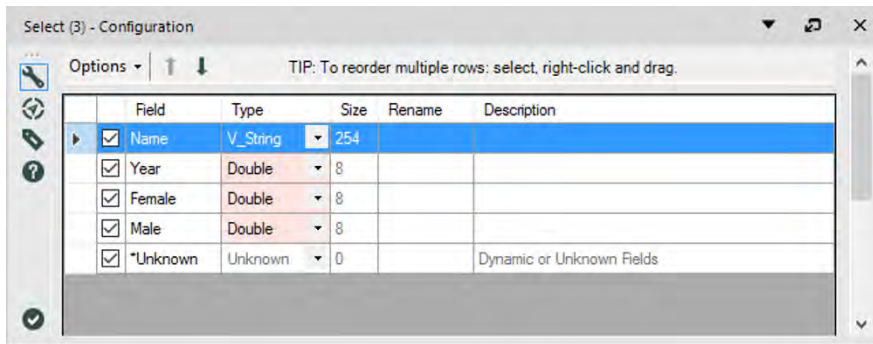
Figure 3-11 – Gender Swapped Select

If we add the *Imputation* tool with the settings as seen in the image, the tool will convert all of the null fields in *Female* and *Male* fields to zeros.
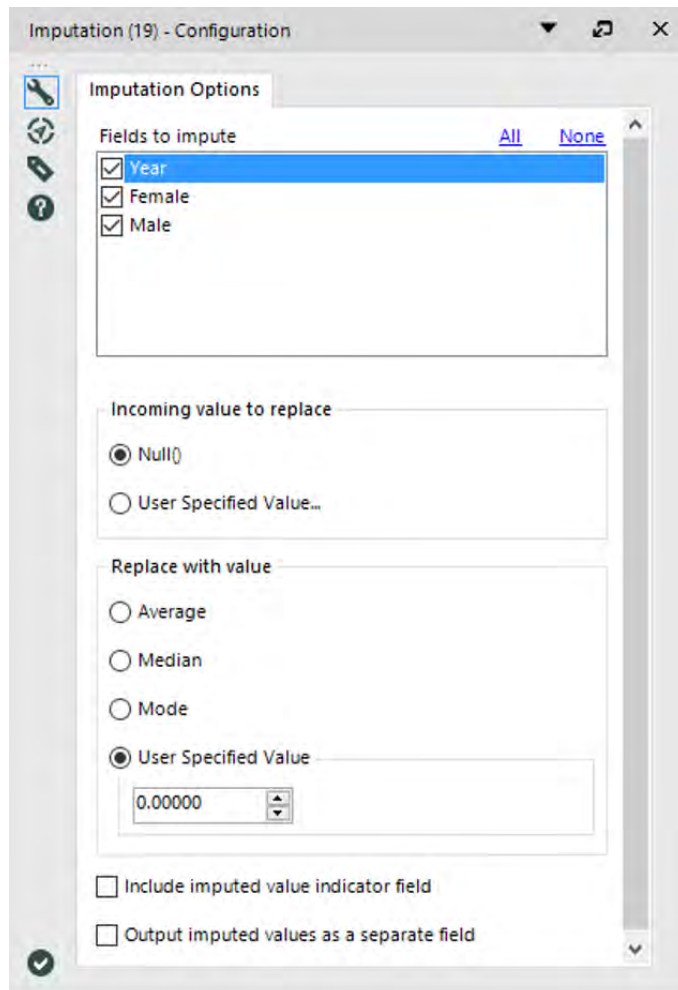
Figure 3-12 – Gender Swapped Imputation Configuration
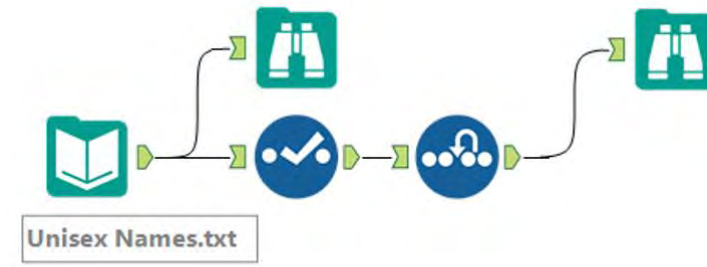
Figure 3-13 – Gender Swapped data stream

Now let's see how the data has changed.

Figure 3-14 – Gender Swapped Browse - Before and After

We can create a filter that limits the data set to *Years before 1890 or years after 2003* to isolate the two date ranges 1880 to 1889 and 2004 to 2013. We will add a filter to the end of the data stream. Use the expression box below to verify the Boolean formula.

Expression:

```
// [Year] < 1890 || [Year] > 2003
// or
[Year] < 1890 OR [Year] > 2003
```

Figure 3-15 – Gender Swapped expression box

Notice that in this expression, we have two lines starting with //. These two forward slashes mean that the line is a comment. It is a way to tell Alteryx to ignore that specific line from the calculation. We used it to show the two intended methods of filtering this data, but it can be used to maintain old versions of the formulas or leave descriptions of the formula for future benefit.

Unisex Names.txt

```
// [Year] < 1890 ||
[Year] > 2003
// or
[Year] < 1890 or
[Yea...
```
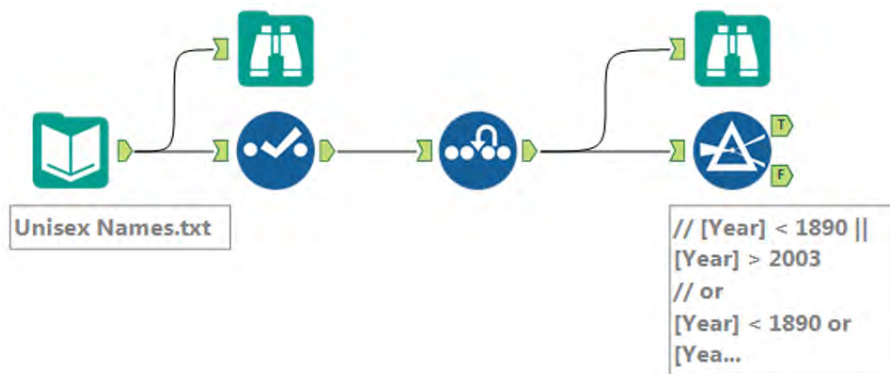
Figure 3-16 – Gender Swapped Data stream

The next thing we can do is create a *flag* for the first and last 10 years for the dataset. We want to do this to create a comparison between the two-time frames. In order to do this, we will create a string field that will have *First 10 Years* or *Last 10 Years* in the field.
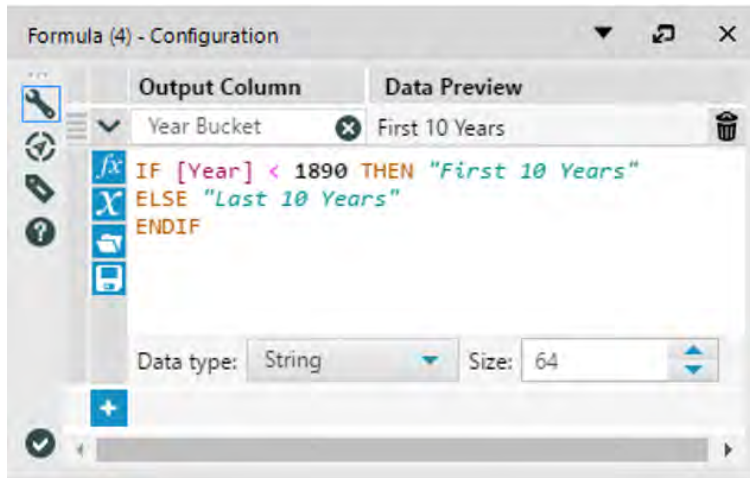
Figure 3-17 – Gender Swapped Formula Configuration

We need to change the field type to string with a conditional formula to test if the record is in the first 10 years. Since the data is already filtered, we do not need to test the other values.

Now that we identified the data with the *Year Bucket* field, we can summarize the data to find out the total number of babies given each name during the 10 years we want to analyze.
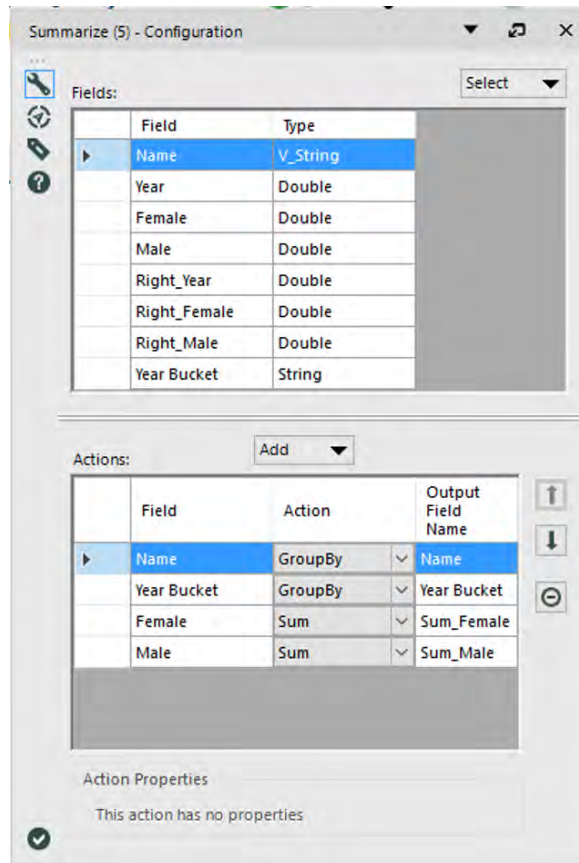
Figure 3-18 – Gender Swapped Summarize Configuration

The data stream after *Summarize* looks as below:
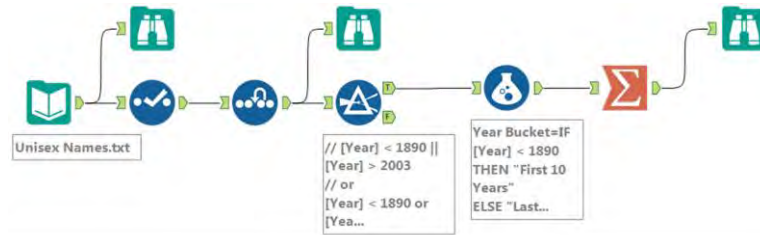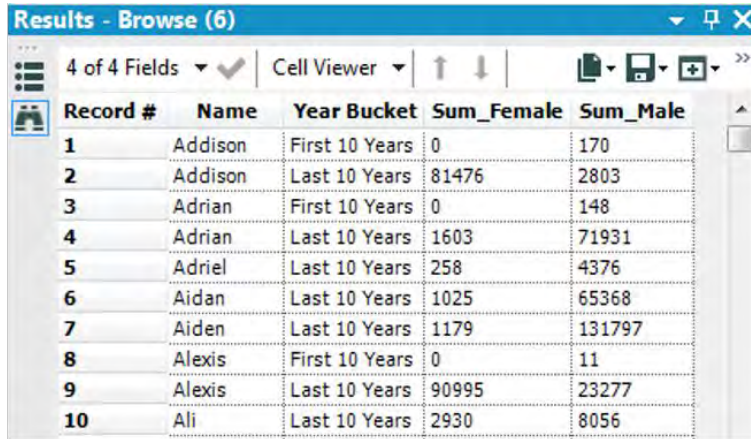
Figure 3-19 – Gender Swapped Data stream after summarize

We will next run the workflow and look at the *Browse* tool to see the data we have.

Figure 3-20 – Gender Swapped Browse configuration

From the *Browse* tool, we can see that we have the data structured in a way we need. However, since we were asked to find the baby names that were previously only assigned to one sex and now are only assigned to the other, we need a way to check each of the names to see if they meet these criteria.

We will do this by using *Multiple Row Formula* to identify which records should be kept. Since we are using a Multi Row Formula, we need to make sure the data is in the right order. So, we will first use a *Sort* tool.
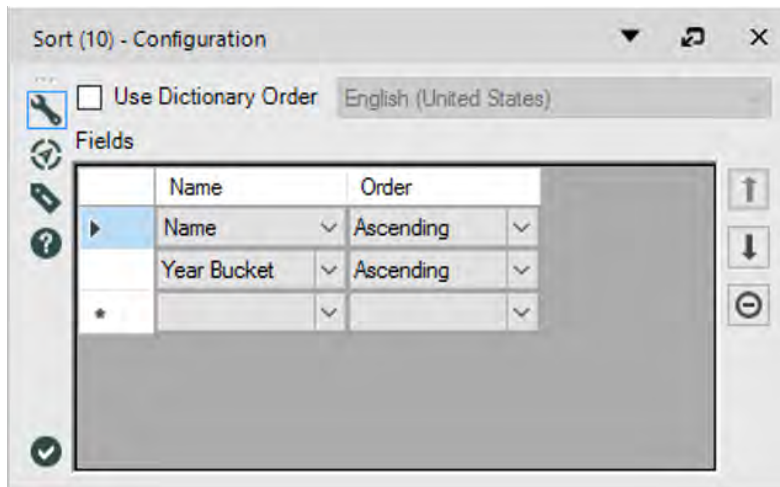
Figure 3-21 – Gender Swapped Sort Configuration
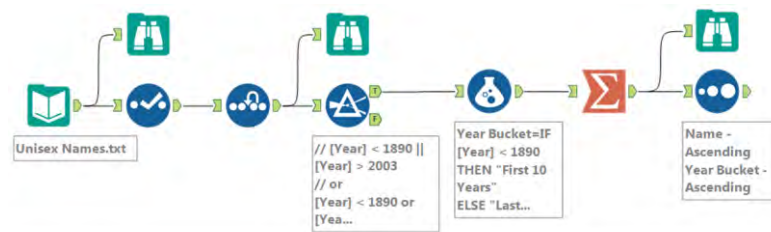
Figure 3-22 – Gender Swapped data stream after Sort

Now that we have made sure the data is sorted, we need to tackle the complex question. In order to do so, we will break it down into smaller, simpler ones:

- Which names occur in both the first and last 10-year lists?
- Which names have at least one count of zero?
- Which names have a female count alternate between something and zero?
- Which names have a male count alternate between something and zero?

By asking these four relatively simple questions, and filtering out the data that does not meet the criteria, we can answer the complex question we have been asked. We will use the same field as we move forward to filter the data, which we will call *Potential Swap*. We will create it in the first *Multi-Row Formula* tool and update it in all subsequent ones. Let us see how we can create this formula.

At this point, much of this formula should be familiar. We are looking at a conditional statement that tests two things, and if both are true, we write *Remove*; else, we write *Check*. What is different about this is that we have special operators in the field names *Row-1:* and *Row+1:*. These allow us to look at the row (record) above and below the current one so we can use the value there. In this case, we are checking to see if the name of the current row (*[Name]*) equals the previous or following rows.

Figure 3-23 – Gender Swapped Multi-Row formula Configuration

Those familiar with Boolean logic may have understood what the formula here actually does. The formula checks to see if the *[Name]* does not match both the previous and following rows. This is a logically equivalent statement called the *contrapositive*, and we are bringing it up here to demonstrate two things: First, *!=* is the *logical operator* that means *does not equal*. The second is to show if we are having trouble with a logical statement. There are multiple ways one can approach it.

Between each *Multi-Row Formula*, we can add a *Filter* to improve speed by removing records we know are not needed for analysis. Let us add a series of alternating *Filters* and *Multi-Row Formulas* to answer the remaining three questions.

All *Filter* configuration windows for the next three steps should look like the image represented here.
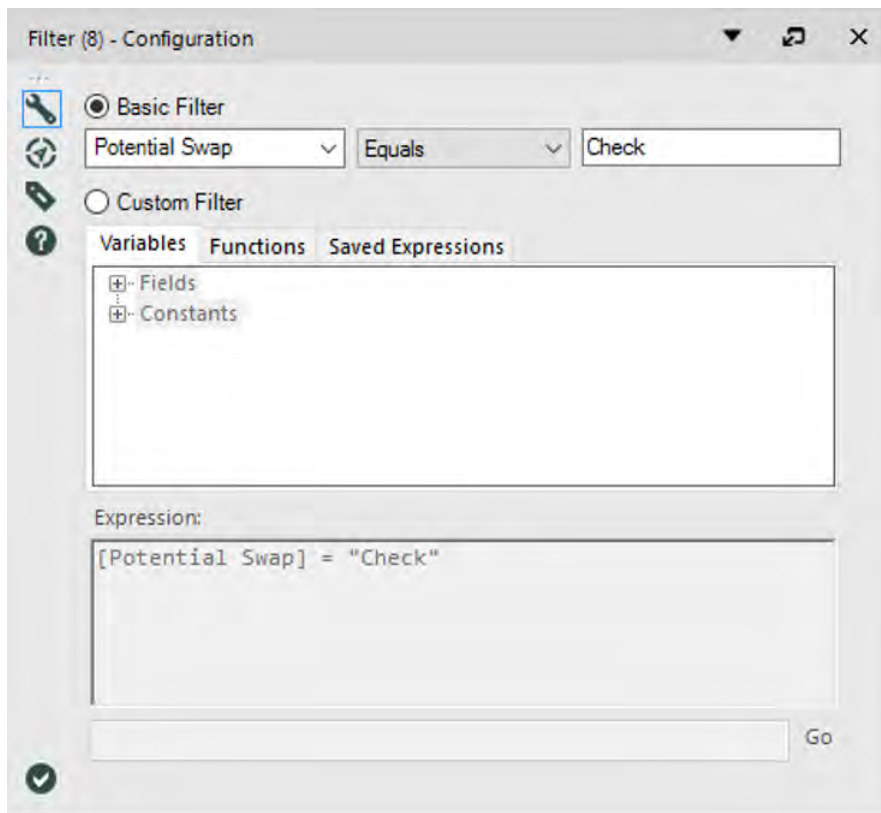
Figure 3-24 – Gender Swapped Filter Configuration

The remaining *Multi-Row Formula* properties windows should look like the provided image, with the following table of equations in the expression box.
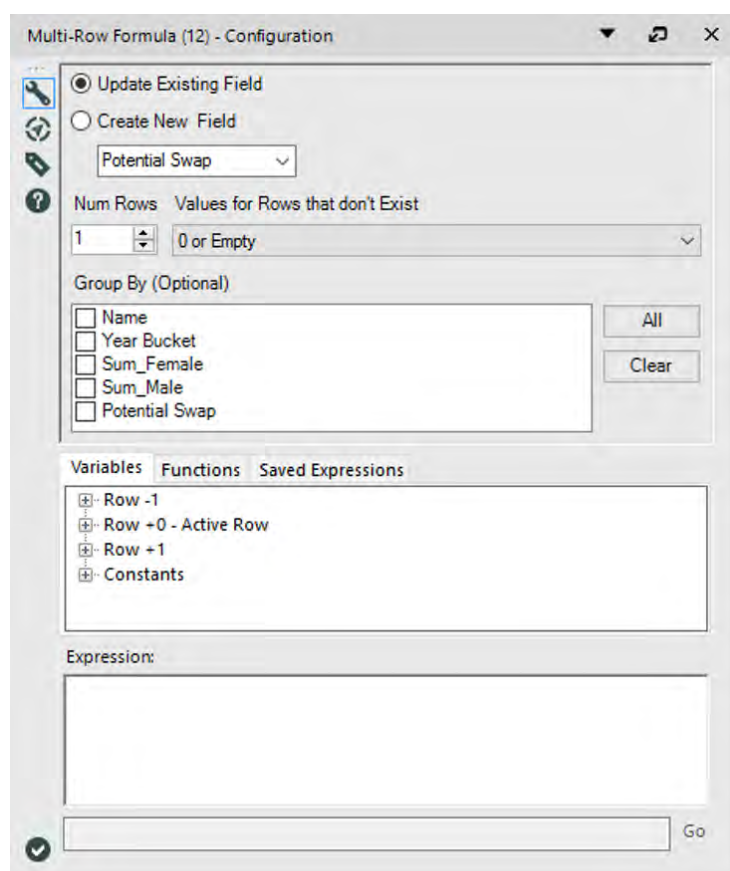
Figure 3-25 – Gender Swapped remaining Multi –row Formula

1   IF [Sum_Female] == 0 or [Sum_Male] == 0

         THEN "Check"

         ELSEIF [Name] == [Row-1:Name] AND ([Row-1:Sum_Female] ==
         0 or [Row-1:Sum_Male] == 0) THEN "Check"

         ELSEIF [Name] == [Row+1:Name] AND ([Row+1:Sum_Female] ==
         0 or [Row+1:Sum_Male] == 0) THEN "Check"

         ELSE "Remove" ENDIF

2   IF [Name] = [Row-1:Name] AND [Sum_Female] != 0 AND [Row-
         1:Sum_Female] != 0

         THEN "Remove"

         ELSEIF [Name] = [Row+1:Name] AND [Sum_Female] != 0 AND
         [Row+1:Sum_Female] != 0

         THEN "Remove"

         ELSE "Check" ENDIF

3   IF [Name] = [Row-1:Name] AND [Sum_Male] != 0 AND [Row-
         1:Sum_Male] != 0

         THEN "Remove"

         ELSEIF [Name] = [Row+1:Name] AND [Sum_Male] != 0 AND
         [Row+1:Sum_Male] != 0

         THEN "Remove"

         ELSEIF [Name] = [Row+1:Name] AND [Sum_Male] !=0 AND
         [Row+1:Sum_Male] !=0

         THEN "Remove"

         ELSE "Check" ENDIF

Figure 3-26 - Gender Swapped Formula

At this point, the data stream is quite long, so we have moved the tools below one another to make it easier to see.
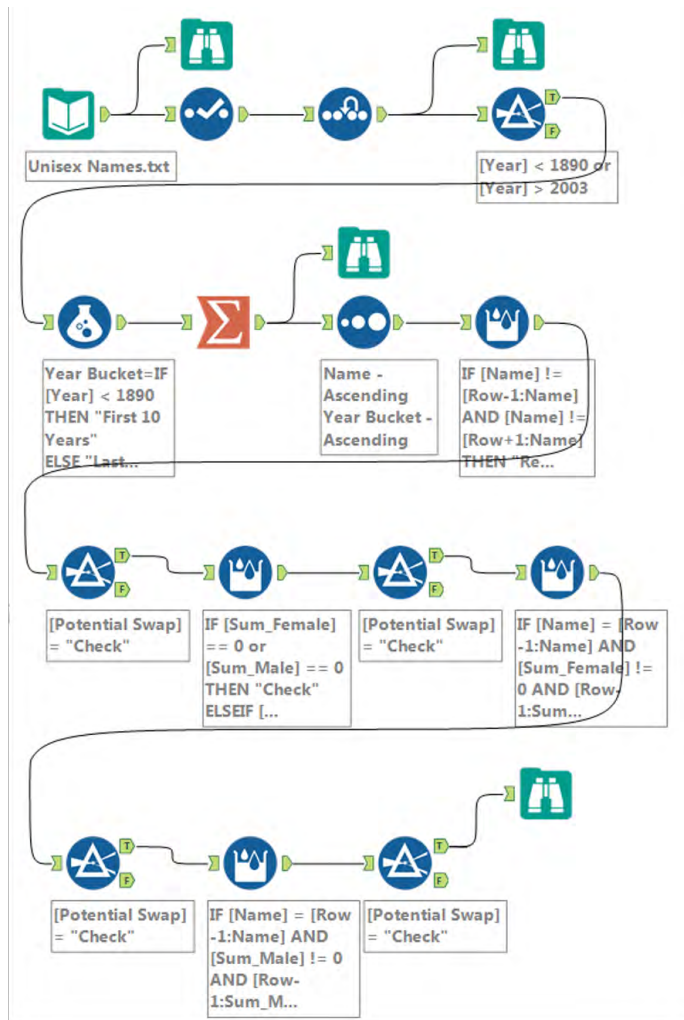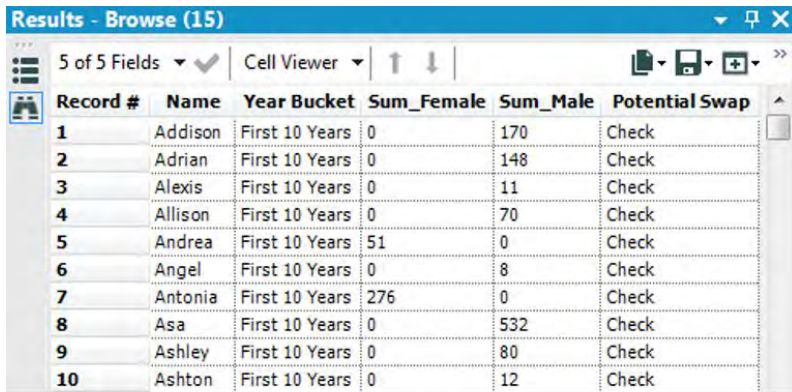
Figure 3-27 – Gender Swapped data stream

We will notice that a *Browse* tool has been added here. Typically, if the formulas are being tested, then a *Browse* tool should be used on both the *True* and *False* outputs of every *Filter* tool. However, since these formulas were already tested, we are just going to take a look at the end result.

Figure 3-28 – Gender Swapped Browse configuration

We can now see that Beverly and Hilary are the only names that meet our criteria.

However, just like in the *Freestyle Skiing* example, we should make Alteryx give us exactly the results we will give to the newspaper so we do not make a mistake. This time, all we need to do is *Summarize* by the *Name* field to have the results that we are looking for.
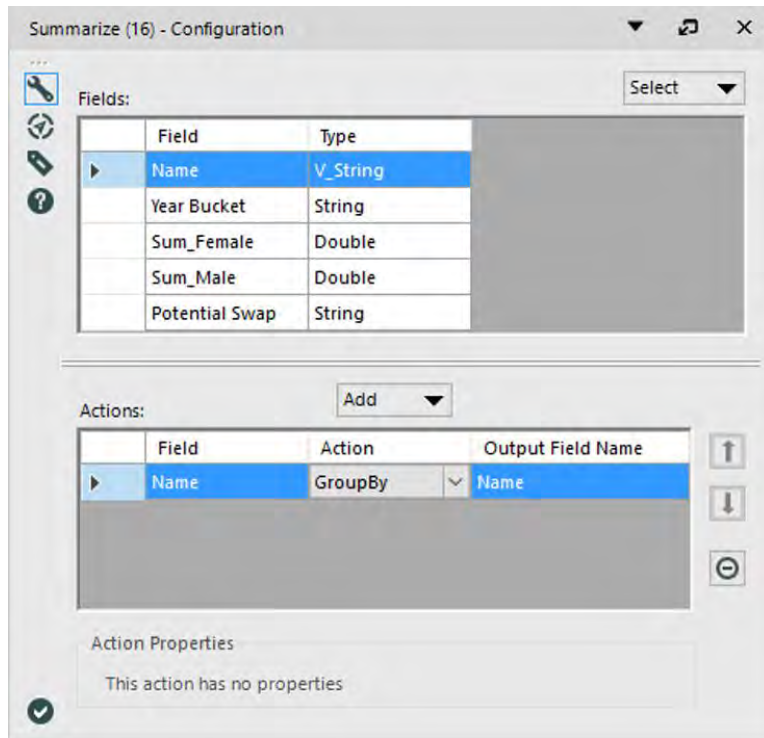
Figure 3-29 – Gender Swapped Summarize configuration

Running the *Browse* tool shows us the following:

Figure 3-30 – Gender Swapped Browse after summarize

The *Gender Swapped* data stream should look like the image on the next page when it is complete.
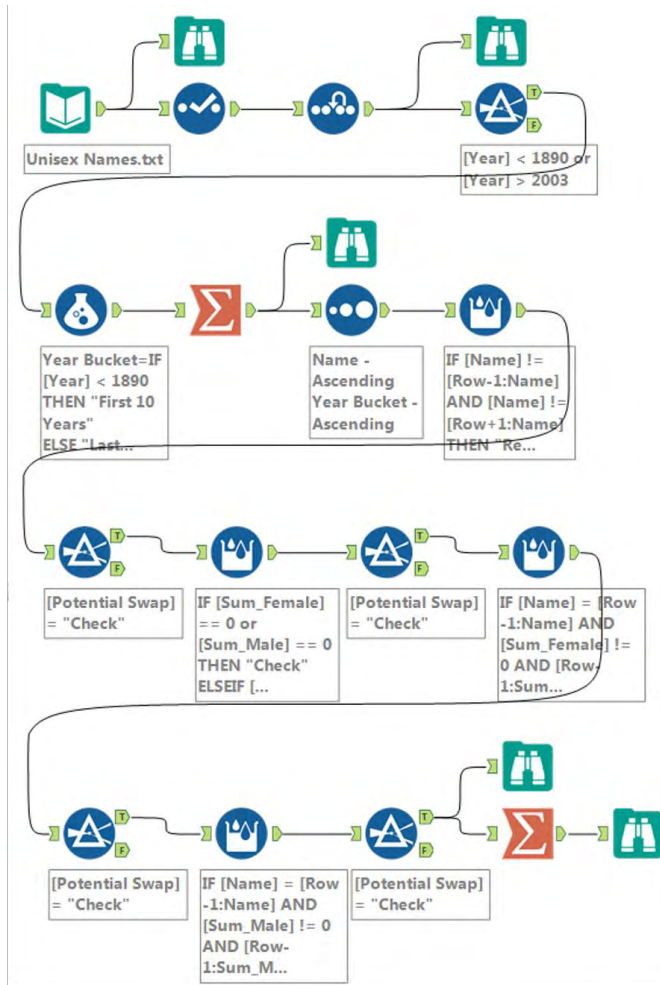
Figure 3-31 – Gender Swapped data stream when complete

3.7   Data Cleansing

	The *Data Cleansing* tool fixes common data quality issues using a variety of parameters.		
	Group	**Input**	**Output**
Figure 3-32 – Data Cleansing	Preparation	Any Data Stream	Data stream

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

The tool has the capabilities to remove nulls, eliminate extra white space, clear numbers from a string entry from a data stream.

Properties Window:

The Data Cleansing Configuration Window has four components:
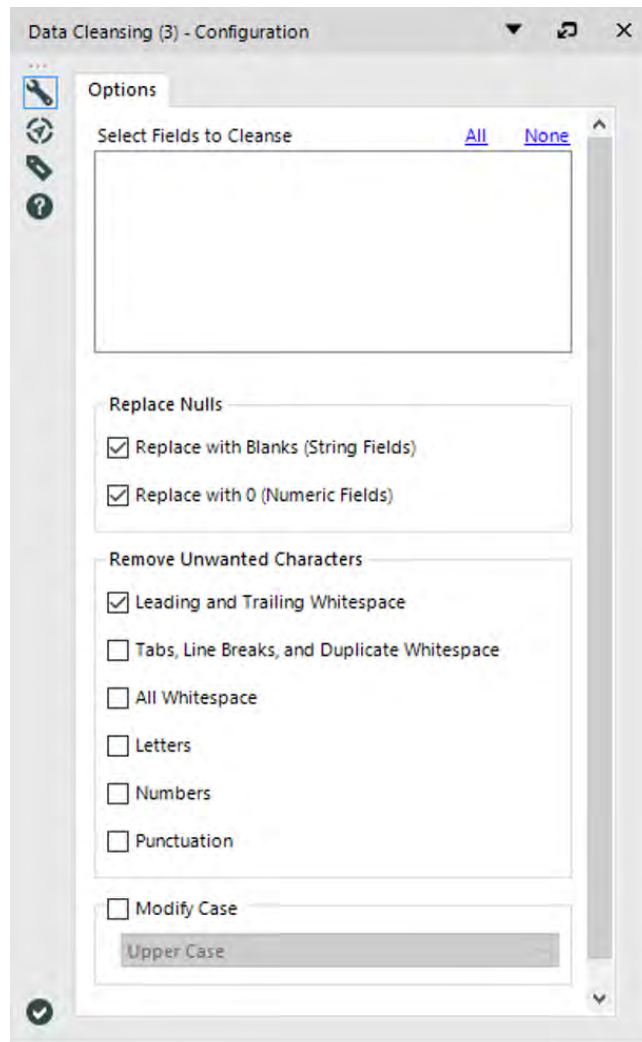
Figure 3-33 – Data Cleansing Configuration

- *Select the fields to cleanse* using the configuration options below. All options, except for Replace Nulls with 0, apply to string data types and to specify different options for different fields, use multiple Data Cleansing tools in your workflow.

- *Select an option to replace* nulls with blanks (for string data types) or replace nulls with 0 (for numeric data types). Both options are selected by default.
- *Select an option to remove* letters, numbers, punctuation, or multiple forms of whitespace. *Leading and Trailing Whitespace* is selected by default.
- *Select an option* to change the capitalization of string data types. *Upper Case* capitalizes all letters in a string. *Lower Case* converts all letters in a string to lowercase. *Title Case* capitalizes the first letter of all words in a string.

3.8   Generate Row

Figure 3-34 –Generate Rows

	The *Generate Rows* tool creates new rows of data at the record level. Use this tool to create a sequence of numbers, transactions, or dates		
	Group	**Input**	**Output**
	Preparation	See Below	Data stream

The Generate Rows tool follows a process to generate rows of data. That process consists of an initial expression (applied to record 1), then a loop expression is applied (such as an increment) that will build subsequent rows, based on a condition (true or false) that will ultimately build rows until the condition is false, when it will terminate the loop.

An Action tool can connect to the Lightning Bolt Anchor to modify how this tool works in apps and macros.

An input connection into this tool is optional, as indicated by the gray input connection arrow.

Properties Window:

The Generate Rows Configuration Window has five components:
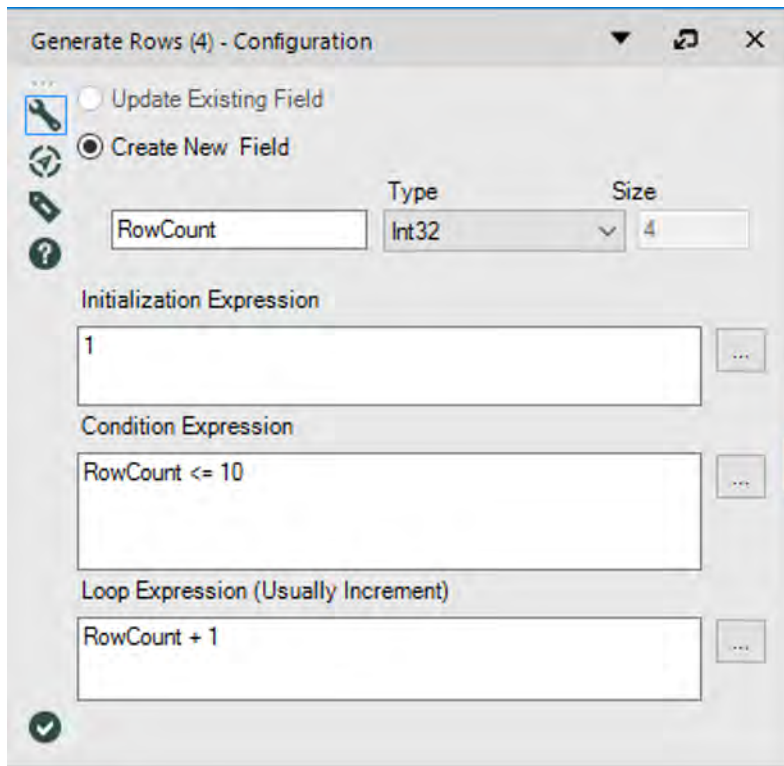
Figure 3-35 – Generate Rows Configuration

- *Create New Field* will be configured as an input. If this method is chosen, specify the new Field Name and appropriate Type and Size.
- *Update Existing Fields*: Assesses the rows coming in and adds new records accordingly. One example would be if your input contains a start and an end value and you would like to generate a row for each value in between.

- Specify the Initialization Expression to start the creation of rows. You can place a value here, create an expression, or click [⚏] to open the Expression Editor.
- *Specify the Condition Expression*, where the condition is either true or false. If the condition is true, additional rows will be generated according to the Loop Expression (step 4), until the condition is false, at which time the generation of rows will terminate. Click [⚏] to open the Expression Editor.
- *Specify the Loop Expression*, usually, an increment of some sort, that will generate subsequent rows until the condition (step 3) is false. Click [⚏] to open the Expression Editor.

3.9   Multi-Field Binning

Figure 3-36 – Multi-Field Binning

The *Multi-Field Binning* is a macro that replicates some of the functionality of the Title tool but has the added feature of being able to tile or bin on multiple fields at the same time.

Group	Input	Output
Preparation	Any Data Stream	See Below

This tool was built primarily for the predictive toolset and as such will only accept numeric fields for binning.

The fields are sorted ascending by default, and the user has the option to tile based on equal records or equal intervals.

An Action tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output includes all fields that went into the tool and a new field for each field selected for binning named FieldName_Tile_Num

Properties Window:

       The Multi-Field Binning Configuration Window has three components:
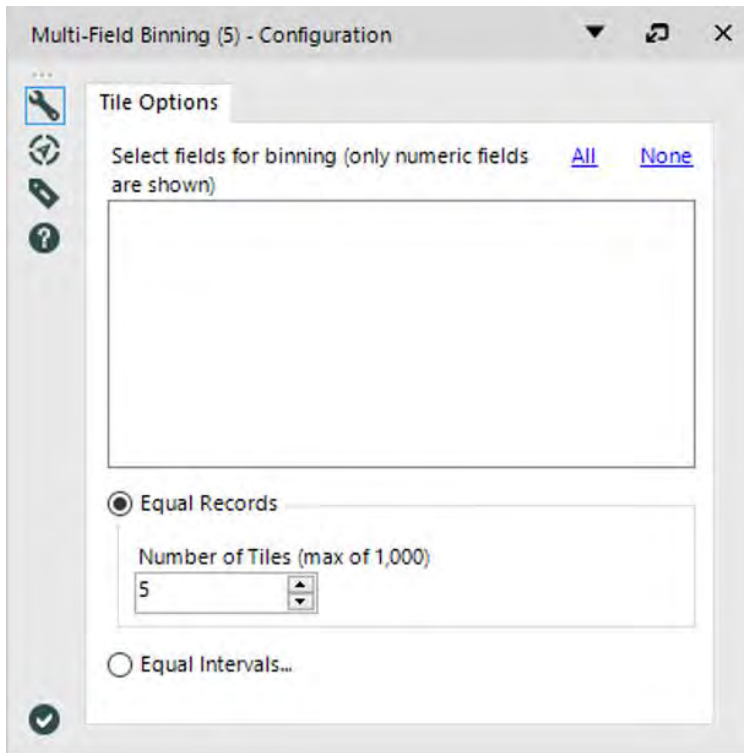
Figure 3-37 – Multi-Field Binning Configuration

- *Select the fields for binning (only numeric fields are shown)* is used to select the numeric fields for binning.
- *Tile Method* chooses the tile methods for binning. Choices are
  1) *Equal Records*: Input records are divided into the specified number of tiles so that each tile is assigned the same amount of records. This is based solely on the record's position in the input file.
  2) *Equal Intervals*: The minimum and maximum values of the tile field are determined. The range is split into equal sized sub-ranges and records are assigned to tiles based on these ranges.
- *The numeric up/down selector* can be used to specify how many bins to create.

3.10   Select Records

Figure 3-38 – Select Records	The *Select Records* tool returns records and ranges of records that are specified, including discontinuous ranges of records.		
	Group	**Input**	**Output**
	Preparation	Any Data Stream	Data stream
An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The Select Records Configuration Window is shown below:
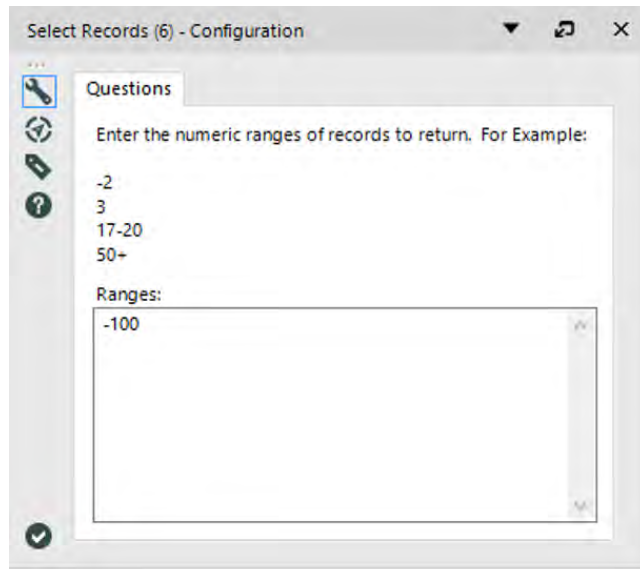
Figure 3-39 – Select Records Configuration

- Enter the records or range of records to return.
- A single digit like "3" returns only row 3.
- A minus sign in front of a number like "-2" returns everything from row 1 through row 2.
- A range of numbers like "17-20" returns rows 17, 18, 19, and 20.
- A plus sign after a number like "50+" returns everything from row 50 through the last row.
- Any combination of these can be specified at the same time.
- **Example:**
  -2
  3
  17-20
  50+
  The following records are returned: 1, 2, 3,17, 18, 19, 20, 50, 51, 52, ... to last row

3.11   What about Me?

From ▾	
To...	Alteryx Consultants
Cc...	
Subject	FW: What About Me?

That is pretty interesting.

Now that we have taken a look at which names were completely reassigned, it would be interesting to find out if your name has a history of being unisex.

A quick look will show you that Michael doesn't even appear in the data set, so I'm going to use the name Andrea, but feel free to use your name.

Let's see what the yearly percent breakdown for male and female children was for the name you choose.

Thanks,

We will add a few things to this list to adhere to best practices, but the necessary steps we need are:

3. Import the data.
4. Clean up the fields.
5. Input a name.
6. Limit the records by that name.
7. Create calculated fields that show us the percent breakdown.
8. Export the data

We already know what issues we have in the dataset, so we can copy the last data stream until the *Imputation* tool.
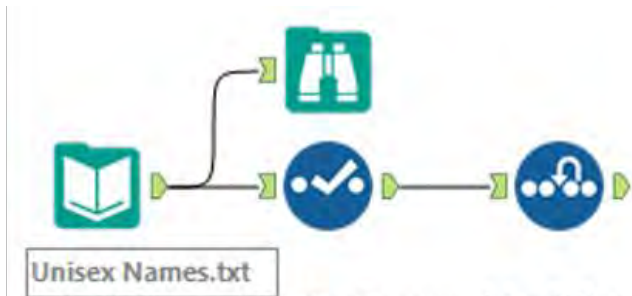
Figure 3-40- Reuse Gender Swapped for What about me

Now that we have this, the next step is to add the name to the data stream. If we add a *Text Input* with a column called *Name*, and name as the only record, below is what we will get.
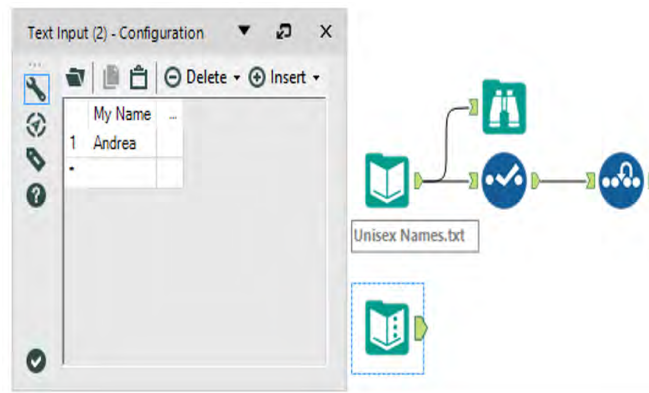
Figure 3-41- What about me Text Input

Since the *Text Input* will show us all of the data, we do not need to add a *Browse*. But it is a good practice to add the *Select* so that we can ensure that the data types from both data streams match.

Unisex Names Select   **Text Input Select**
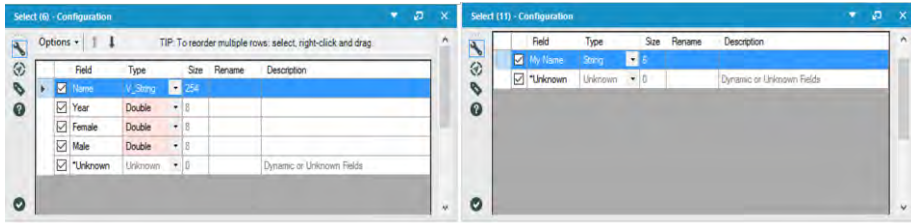
Figure 3-42 – What about me Select Configuration

Though we have different types in our data field, we do not need to convert them because they are both strings. If they were not, it would cause an issue in joining the data. Due to the way we have approached this problem, we can use the *Join* tool to filter the data to the appropriate records. So we will take only the joined section and not keep the name field coming out of the right (*Text Input*).

Figure 3-44 – What about me Browse Configuration

Running the *Browse* tool for the *J* output will show us the following:
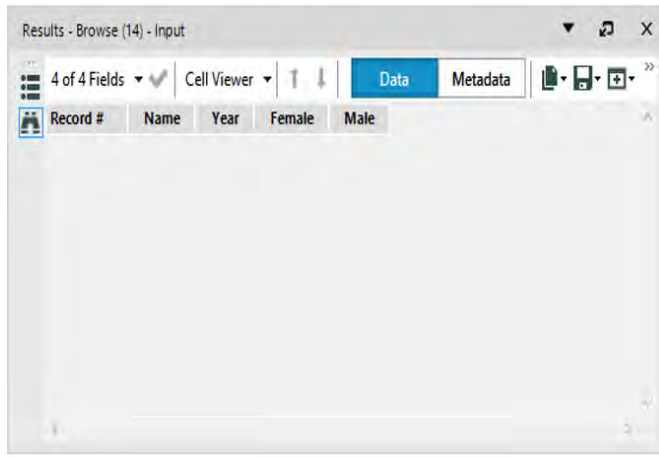
Figure 3-45 – What about me Browse Configuration

If we see the image here, we see that no records have been returned. It is because the name entered in the *Text Input* does not match anything in the *Unisex Baby Names* dataset. Change the *Text Input* name to *Andrea* to follow along more easily.

At this point, we have covered steps 1 through 4. The next step is for us to create the percent breakdown for both male and female babies in each year. Since we intend to perform the same calculation on two different fields, we can use the *Multi-Field Formula* tool to accomplish this.

If we add the *Multi-Field Formula* tool after the *J* output from the *Join* with the following configuration, we will be able to create the percent of the total for Male and Female babies each year.
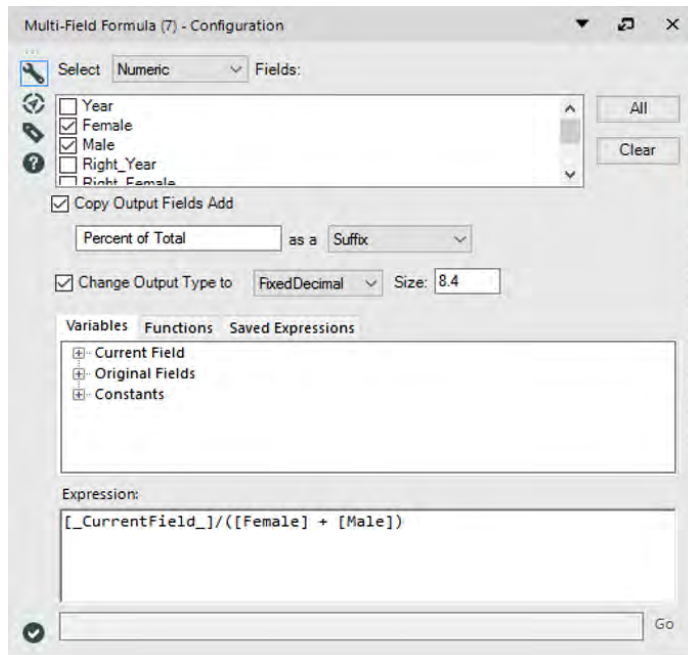
Figure 3-46 – What about me Multi-Field formula Configuration

Now, all that's left is to write the file out to *What About Me.csv*. The *What About Me* data stream should look like the below image when it is complete.
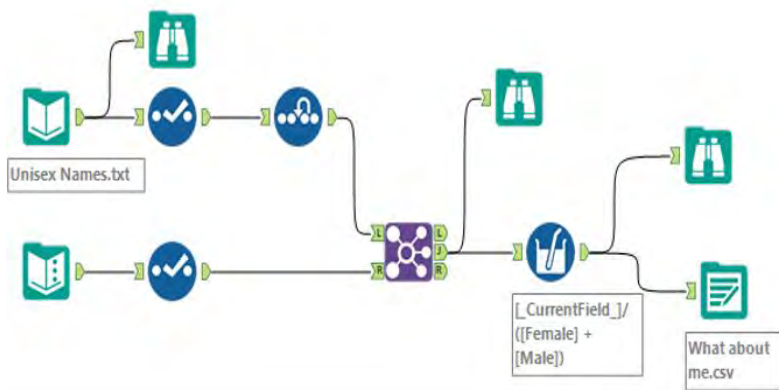
Figure 3-47 – What about me data stream when complete

3.12   What's In A Name?

From ▾	
To...	Alteryx Consultants
Cc...	
Subject	FW: What's In A Name

Hey,

So I heard back from the newspaper, and they decided to run the article. But they are curious about something that they think will be an interesting spin.

See the email below.

I don't have time to help you out this time, but I'm interested in the result. Let me know when you've got it.

Thanks,

From: Erin
To: Michael Davis
Subject: Unisex Baby Names

Hello,

What you spun up was good, but I think that we are going in a different direction, as long as the numbers work out.

For some context:

When someone tells us their name, we make assumptions about them, one of the first being their sex.

I want to know the 25 names most likely to be heard and have an incorrect assumption made about them concerning sex, based on this data.

When I say "most likely" let's call it the most amount of years when names had at least a 25% chance of referring to males and females. (If there are ties, they should be alphabetical. You can remove "Unknown" and "Baby" from the list of names).

Deadline's approaching,
Erin

CHAPTER 4
The Direct Approach

To...	Alteryx Consultants
Subject	What's The Policy On That?

Hey,

As I'm sure you're aware, we are working for a small company.

Frank has asked me to build a process that helps us stay on top of our new policies. I will be using a software program that, given a link, can automatically open the file or web address that it points to. But first, I need to prep the data.

I'm going to have you sit with me on this so you can see some more of Alteryx's functionalities.

They only want policies that were published in the last 30 days to show up.

Thanks.

4.1   Tools & Concepts

Tools	**Concepts**
Append Fields	Appending Data
Date Time Now	Cartesian Product
DateTime	Working With Dates
Directory	Working With Directories

4.2   Append Fields

Figure 4-1 – Append fields	The *Append* tool adds each record from *S (source)* to the end of each record in *T (target)*.		
	Group	**Input**	**Output**
	Join	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input S: Any data stream.

Input T: Any data stream.

Output: A data stream of records that has the Cartesian product from inputs *T* and *S*, with the records of *S* appended to the end of *T* records.

This tool is very useful for appending a single data value onto a full dataset, such as appending the mean of a field to every row for calculation purposes. Since we are taking the Cartesian product of two data streams, we run the risk of creating a very large data stream, which could cause memory errors.

<u>Properties Window:</u>

The *Append Fields Configuration* window is identical to the *Select Configuration* window, with two exceptions.
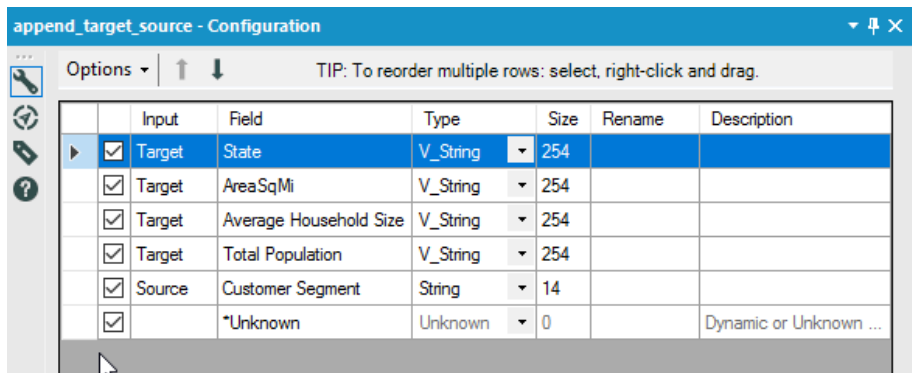
Figure 4-2 Append Field Tool Configuration

- There is an additional element in the metadata section called *Input*. It identifies if the data is coming from the *Target (T)* or *Source (S)* inputs for the tool.
- There is a *Warn/Error on Too Many Records Being Generated*, which allows us to decide if and how we should be alerted to a high rate of replication of the *Target* field.

4.3   Date Time Now

  Figure 4-3 – Date time now	The *Date Time Now* tool gets the system time when the module starts executing.		
	Group	**Input**	**Output**
	In/Out	None	String

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A string representing the date or the date and time, depending on the settings.

The tool is useful if we want to limit our data stream based on the current time, or if we want to publish the time of a report or dataset when created.

Properties Window:

The *Date Time Now Configuration* window allows us to select the format of the string required in the output.
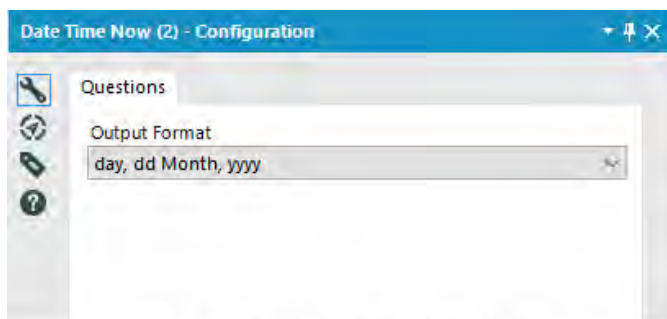
Figure 4-4 – Date time now configuration

The following image represents the format of the strings available. See *Appendix G* for a list of character descriptions.

Figure 4-5 – Date time now formats
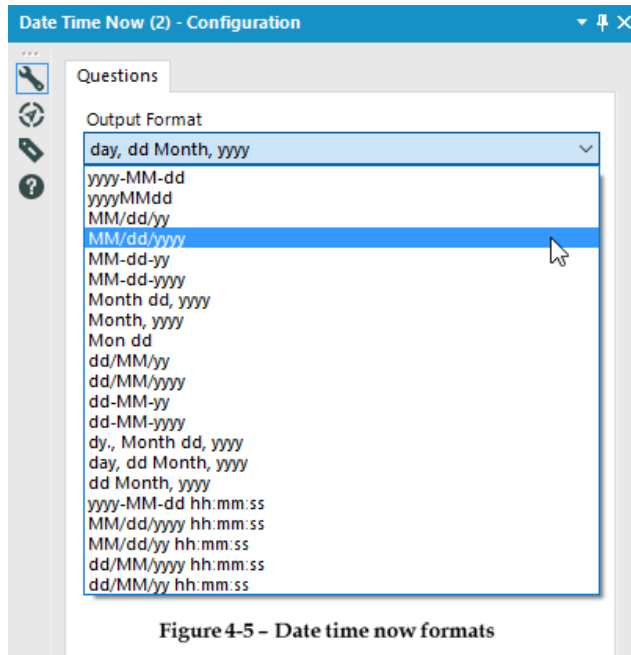
4.4   DateTime

	The *DateTime* tool converts between String and Date format fields		
Figure 4.6 – Date time	**Group**	**Input**	**Output**
	Parse	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with a date field in a *string*, *date*, or *date/time* field.

Output: The original data stream with a new field. The new field can be a *string* representing the date or the date/time depending on a *date* or *date/time* field that was represented as a string field.

The tool can only convert the formats in the menu. We need to write a *Formula* that does string conversions in addition to using this tool if we need a different type of date format.

<u>Properties Window</u>:

The *DateTime Configuration* window has four components.
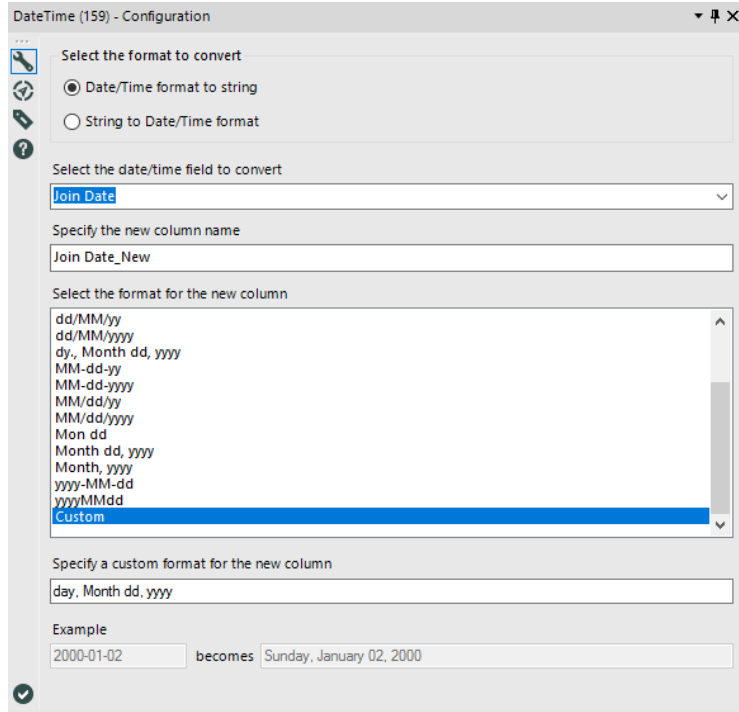
Figure 4-7 DateTime Tool Configuration

- *Conversion Mode* allows for converting from or to a string field.
- *Formatted Input String Field (Input Date/Time Field to be formatted)* is the field we want to convert.
- *Format of Input String (Desired Format of Output String)* is the format the input string is in.
- *Output Date/Time Field (Output Formatted String Field)* lets us name the field we are creating.

4.5   Directory

	The Directory tool creates a data stream that has the contents of a directory.		
Figure 4-8 - Directory	**Group**	**Input**	**Output**
	In/Out	None	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream containing metadata about all of the files in a given directory.

We can use special characters to look for a file meeting a pattern. For example, *.* will return all files, and *a *.csv* will return all *.csv* files starting with the letter *a*. These file-matching commands are different from the regular expression syntax, which we will discuss in *Expensive Beauty Products*.

<u>Properties Window</u>:

The *Directory Configuration* window has three components.
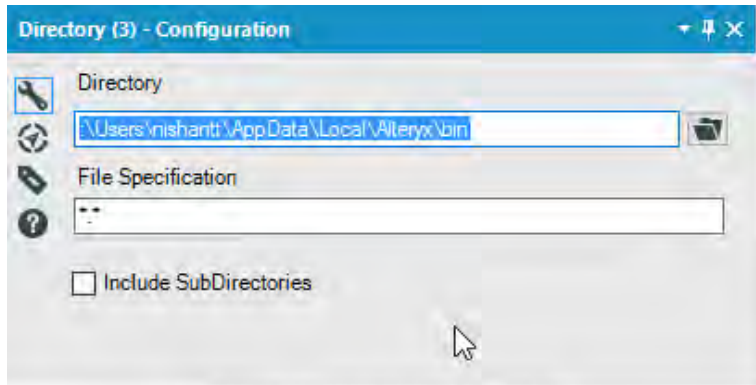
Figure 4-9 – Directory configuration

- *Directory* lets us navigate to a folder.
- *File Specification* allows us to type the generic format for a file name to be found in the *Directory*, where * denotes zero or more characters or spaces, while *?* denotes exactly one character or space.
- *Include SubDirectories,* when checked, includes each of the subfolders when it's looking for files. Otherwise, it will only look in the directory that we have navigated to.

4.6   Auto Field

	The Auto Field automatically sets the field for each string to the smallest possible size and type to accommodate data in each column.		
Figure 4-10 - Auto Field Tool	**Group**	**Input**	**Output**
	Preparation	See below	See below

Input: A data stream with String data in any column/columns.

Output: A data stream with optimized metadata of the column containing string data (*Field Type and Size is set to a smallest possible value which can hold the data*).

The workflow must be run at this point for the configured fields to be properly loaded and recognized downstream. Connect the Auto Field tool to a Browse tool and run the workflow before configuring additional downstream tools.

<u>Properties Window</u>:

The *Auto Field* configuration has only one configuration setting.
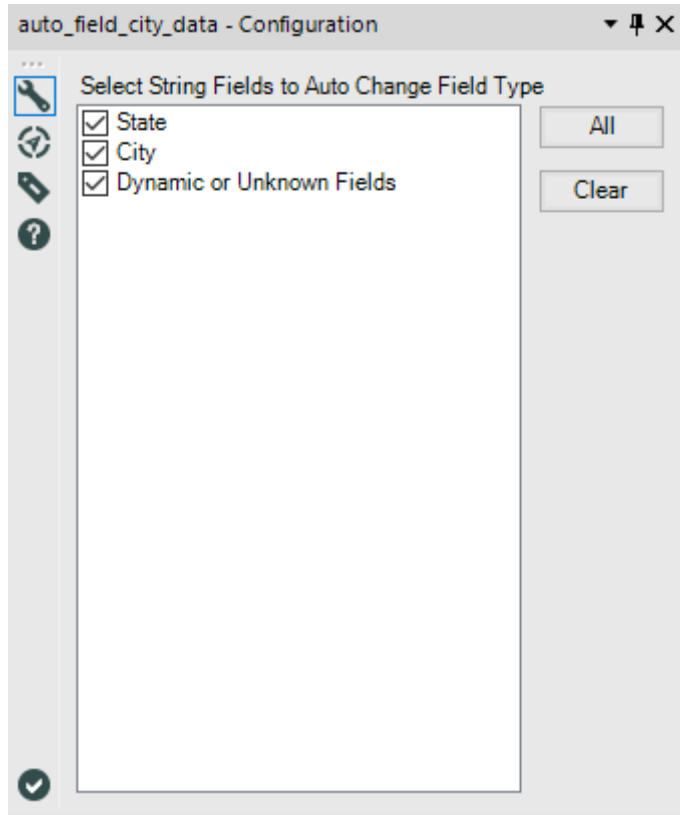
Figure 4-11 - Selection of string fields configuration

- *Select-String Fields to Auto Change Field Type* section allows you to select the columns for which optimization of field type and size is required.

4.7   Oversample Field

	The Oversample Field creates a new dataset that would have a defined

Figure 4-12 - OverSample Tool	ratio of positive and negative response samples (records) based on the selected response criteria.		
	Group	**Input**	**Output**
	Preparation	See below	See below

Input: A data stream, specifically one used for creating binary classification model. For example, Yes/No decision prediction model.

Output: A data stream with defined number of positive and negative response records.

Properties Window:

The *Oversample Field* configuration has three components.
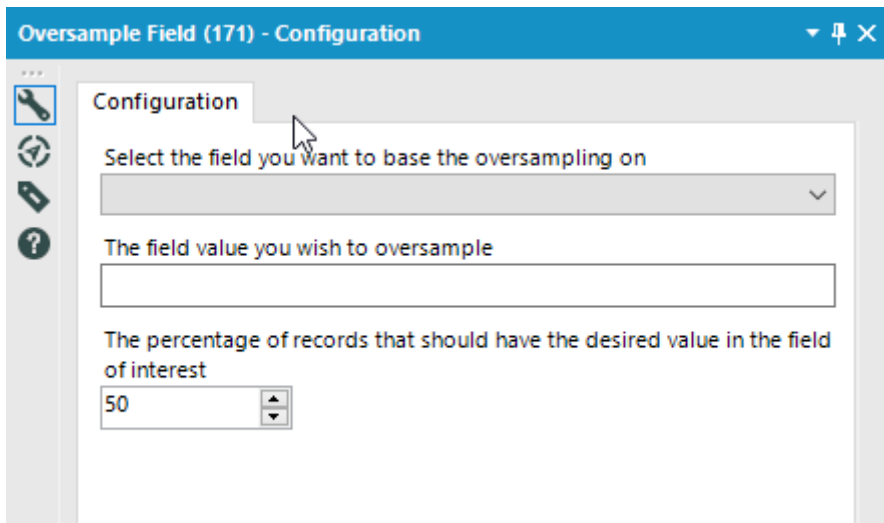
Figure 4-13 - Oversample Field Tool Configuration

- *Select the field you want to base the oversampling on* allows us the select the field containing which needs to be considered for oversampling
- *The field value you wish to oversample* allows us to select the values in the field that needs to be oversampled, typically it will contain binary classification supporting values i.e. "Yes" and "No"
- *The percentage of records that should have the desired values in the field of interest* allows us to select the percentage of binary classification samples you need in the output stream

4.8   Tile Tool

	The Tile Tool helps to logically tile (divide & group) the data stream. This is done by adding two more fields to input stream Tile Number and Tile sequence number.
Figure 4-14 - Tile Tool	

	Group	Input	Output
	Preparation	See below	See below

Input: A data stream, which specifically needs to be logically divided/grouped.

Output: A data stream with two extra fields added; Tile number assigned Tile of the record and Tile sequence number assigned records position within the Tile.

Properties Window:

The *Tile Tool* configuration has 5 methods to perform tiling.
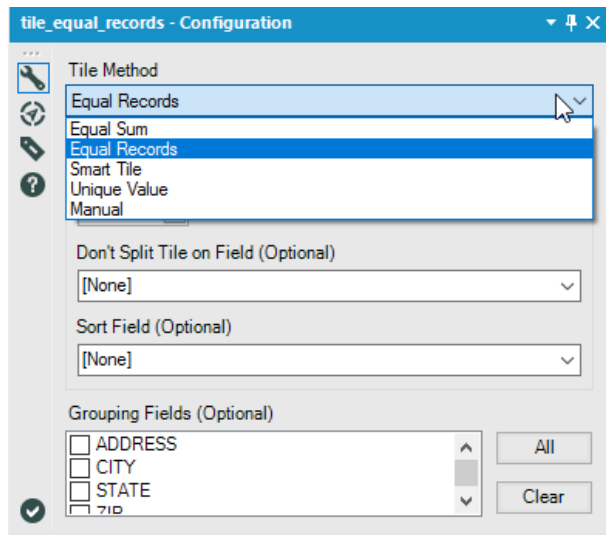
Figure 4-15 - Time Tool Configuration

- Tile Method *Equal Records* assigns Tile to records making sure that an equal number of records exist in each tile.
  - o *Select Number of Tiles* allows us to specify the number of tiles to assign to input data stream
  - o *Don't Split Tile on Field (Optional)* allows us to select the field which should not be considered for Tile process i.e. Tile would not split across selected field.
  - o *Sort Field (Optional)* allows us to select the field which needs to be sorted prior to split and tiling process.
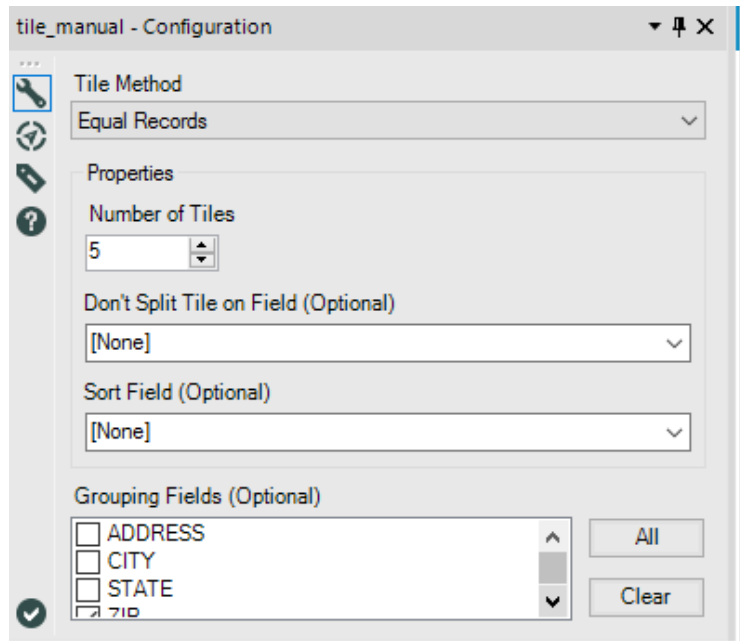
Figure 4-16 - Tile Method *Equal Records* Configuration

- Tile Method *Equal Sum* splits the data in such a way that the sum of the selected *Sum Field* for each tile is equal.
    - o *Number of Tiles* allows us to specify the number of tiles to assign to input data stream
    - o *Sort Field (Optional)* allows us to select if we would like to pre-sort data based on the specified field. If nothing is specified the tool takes data and tiles based in the incoming sort order.
    - o *Sum Field* allows us to select the numeric field for which sum of the field needs to be equally distributed across the tile.

Figure 4-17 - Tile Method *Equal Sum* configuration

- Tile Method *Smart Tile* creates tiles based on the standard deviation of the value in specified fields. The tiles assigned indicate whether the record value falls within the average, above average or below average etc.
  - o *Tile Field (Must be Numeric)* allows us to select the field which would be considered for standard deviation.
  - o Set of Radio Buttons are provided to specify how the additional output (Tile information) fields would be generated
    - ▪ *Do not output Name Field* can be selected if not additional output field is required except the Tile number and Tile sequence number

- *Output Name Field* can be selected to create an additional output field with descriptors like *Average, Above Average, Below Average, High, Extremely High etc.*
- *Output verbose Name Field* can be selected to generate additional descriptor field which would specify the data range in parenthesis along with the descriptor generated by *Output Name Field* option. For example, *High (12750 to 15500)*
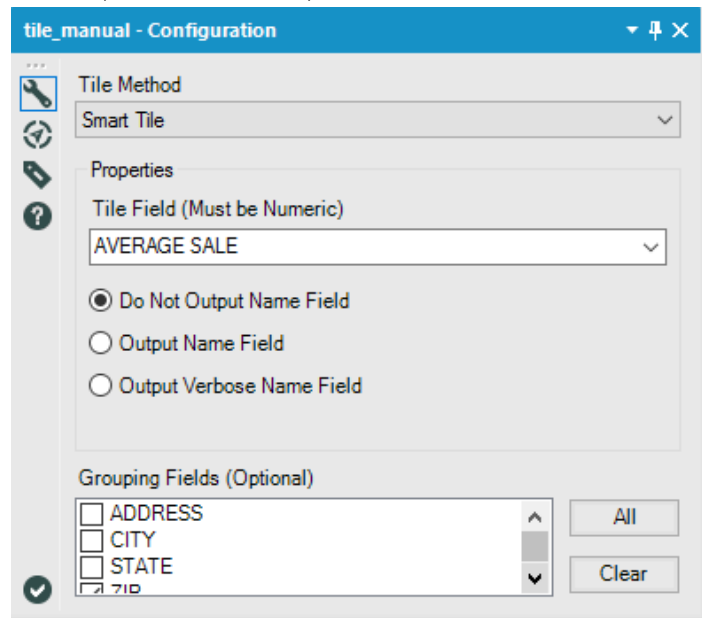
Figure 4-18 · Tile Method *Smart Tile* configuration

- Tile Method *Unique Value* allows us to create a separate tile for each unique value in the selected field.
  - *Check the box of field or fields to Tile by* allows us to select the field /fields which would be tile criteria

o   If the *Leave Unsorted* check box is left unchecked, the table will be automatically sorted by the field or fields you specified for the tile
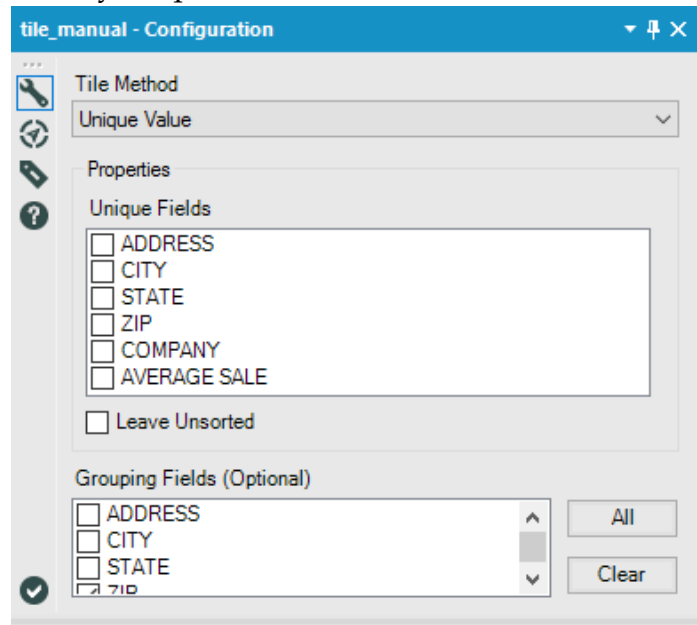
Figure 4-19   Tile Method *Smart Tile* **configuration**

- Tile Method *Manual* allows us to specify the cutoff range for tiles by allowing to specify each range
    o   *Tile Field (Must be Numeric)* allows us to select the field to be tiled
    o   *Enter one of more Tile cutoffs* allows us to specify the cutoffs as be our requirement of Tile
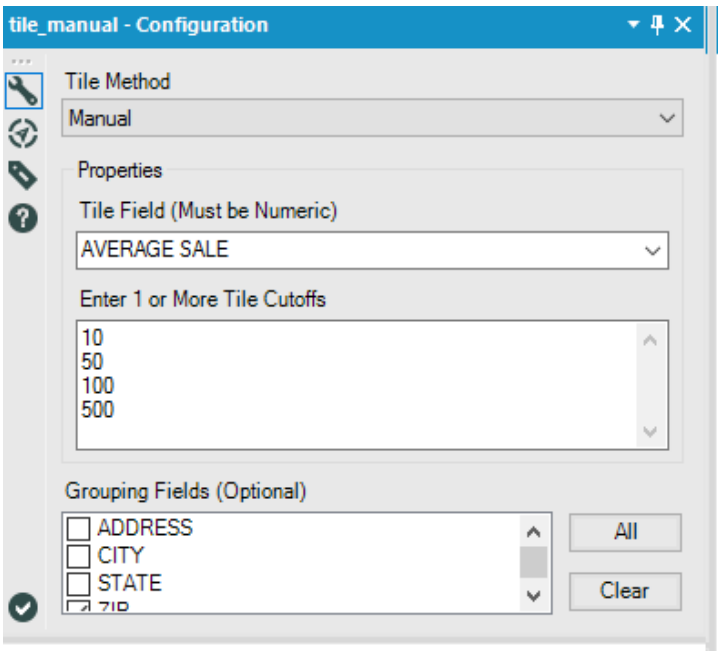
Figure 4-20 - Tile Method *Manual* configuration

The Tile Tool allows us to select the grouping field, which is optional. By selecting group a series of tiles will be generated for every record within the specified group.

4.9   Explorer Box

	The *Explorer Box Tool* displays the specified URL/location path. It is similar to windows browser/Explorer.		
Figure 4-21 - Explorer Box	**Group**	**Input**	**Output**
	Documentation	NA	NA

Properties Window:

The *Explorer Box* configuration have two components.
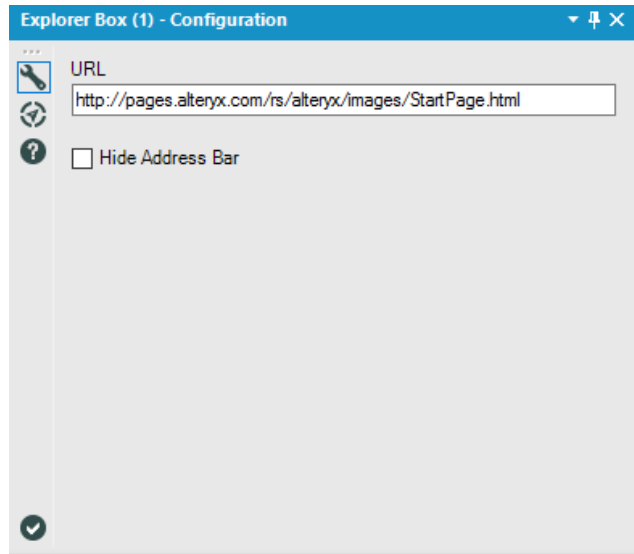
Figure 4-22 - *Explorer Box* Configuration

- *URL* allows you to specify the webpage or folder/file location which you would like to display in the tool area
- *Hide Address Bar* check box allow to hide/show the address bar of the webpage

4.10   Make Columns

Figure 4-23 - Make Columns Tool

The Make Columns tool takes rows of data and arranges them by wrapping records into multiple columns. The user can specify how many columns to create and whether they want records to layout horizontally or vertically.

	Group	Input	Output
	Laboratory	See Below	See below

Input: A data stream rows which need to be converted into a small set of columns and arranged vertically or horizontally.

Output: An output stream of data

This is a Laboratory tool and is not for production use. It may have documented known issues, may not be feature complete, and is subject to change.

Properties Window:

The *Make Columns* configuration has below three selection;
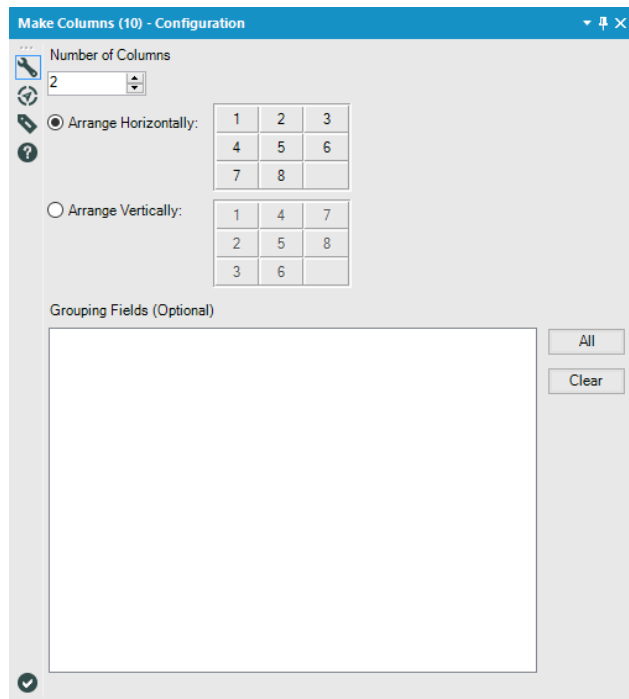
Figure 4-24 - Make Columns Tool Configuration

- *Number of Columns* to specify the number of columns required in the output stream.
- *Arrange Horizontally / Arrange vertically* to specify whether the incoming stream has to be horizontally processed or vertically
- *Grouping Fields (Optional)* to specify the sorting and the group criteria to arrange the incoming stream into columns.

4.11   What's The Policy On That?

| To... | Alteryx Consultants |
| Subject | What's The Policy On That? |

Hey,

As I'm sure you're aware, we are working for a small company.

Frank has asked me to build a process that helps us stay on top of our new policies. I will be using a software program that, given a link, can automatically open the file or web address that it points to. But first, I need to prep the data.

I'm going to have you sit with me on this so you can see some more of Alteryx's functionalities.

They only want policies that were published in the last 30 days to show up.

Thanks.

Since we need to search for files, we are going to be using the directory tool. Our finalized policies are published as *pdf* files, so we can use that to limit our search. We have three divisions responsible for publishing policies right now, and they all

publish them to subfolders in *Chapter 4 – The Direct Approach >*
What's The Policy On That.

The plan is to:

- Bring in a list of all of the *PDF* files in the directory.
- Get the current date.
- Tag the files to form 3 buckets (Extended Functionality to explain How Tile Tool can be used to create buckets)
  a. Files created in last 30 days from today
  b. Files created in between 30 to 90 days ago from today
  c. Files created prior to 90 days from today
- Export the data to *What's The Policy On That.csv*.

When we bring the *Directory* tool onto the canvas and navigate to the *Policy* folder in the *Directory* file browse, we can use the expression **.pdf* to find all *PDF* files in the directory. But since we need to search all of the subfolders (subdirectories), we need to check the box at the bottom of the configuration window.
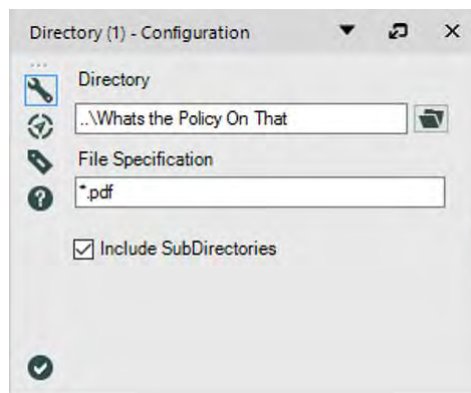
Figure 4-25 – Policy Directory Configuration

Figure 4-26 - Policy Data Stream

As usual, we will add a *Browse* and S*elect* tool following the data connection. Let's look at the *Select* tool to see what's in the data stream.
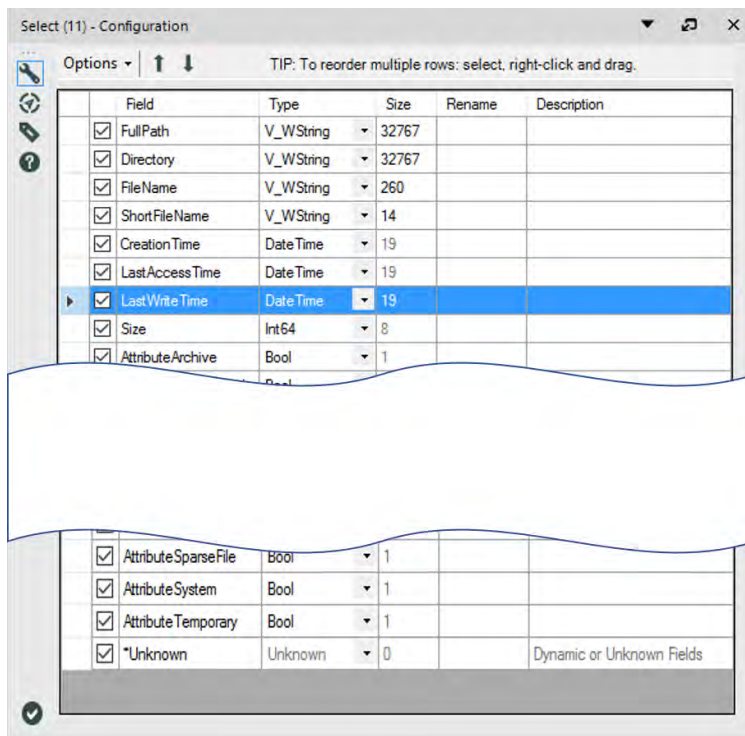
Figure 4-27 – Policy Select Configuration

We can see the data stream has 19 fields. Whenever we use the *Directory* tool, it will return these 19 metadata fields about

each of the files that were found. See *Appendix K* for details on each.

In our case here, we will only be working with the *FullPath*, *FileName*, and *CreationTime* fields. Now that we have the appropriate information from the list of files, the next thing we need is to add the *current date* to the workflow. This is where the *Date Time Now* tool comes in. Please note here that the annotations are hidden to save space.

We will set the output format of the *Date Time* input to *MM-dd-YYYY*. It's important to note that the *Date Time* tool creates a single field with a single record, which is in the format as specified in the configuration.

Figure 4-28 - DateTime Tool Configuration to generate DateTimeNow

Now that we have these two data streams ready to be combined, the question becomes: What method do we use?

A *Union* would allow us to combine the data. Unfortunately, since we will be performing a calculation to see if

the dates were in the last 30 days, we need the *Today* in every record not added to the end of the dataset. A union will not work.

A *Join* would allow us to do this if we had a field we could match, which means we could use a function on both data streams that just writes 1 to the field, and then join on that. But that's difficult to explain and maintain.

Fortunately, Alteryx has a tool called *Append Fields* that does exactly what we want to do with the *Formulas* and a *Join*, in an optimized way. We shall connect the list of *PDFs* to the *Target (T)* input and the date to the *Source (S)* input. Since we are changing the structure of the data by combining two data streams, we should put a *Browse* tool following the *Append Fields* tool to follow the output at this stage.

Auto Field Tool is being used to optimize the field lengths for fields which string data type.

Figure 4-29 - Policy Append Configuration
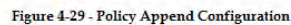
In order to create buckets, we need a Formula Tool to set a new parameter *Bucket.* This is done using the formula given below;

IF [CreationTime] > DateTimeAdd([Today],-30, "days")

THEN 1 ELSEIF

[CreationTime]   =   DateTimeAdd([Today],-30, "days") and [CreationTime] > DateTimeAdd([Today],-90, "days")

THEN 2 ELSE 3 ENDIF

The buckets will logically divide the data based on decided conditions.
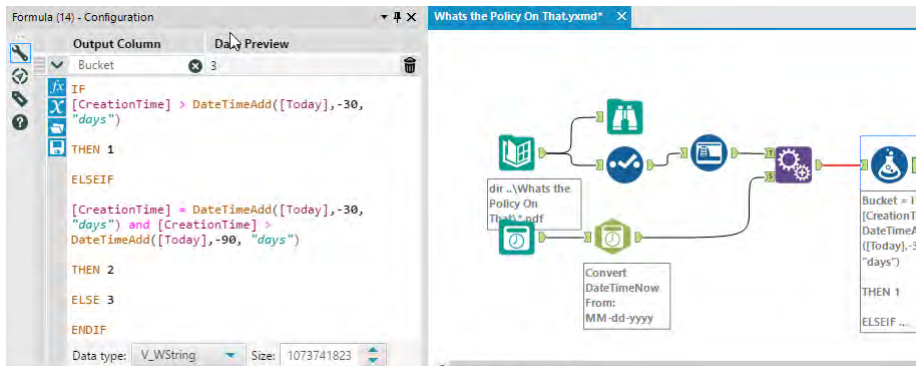
Figure 4-30 - Formula Configuration to create buckets

Next Tile Tool is used to logically group the data. Followed by a Filter Tool to filter data with Bucket Value as 1.
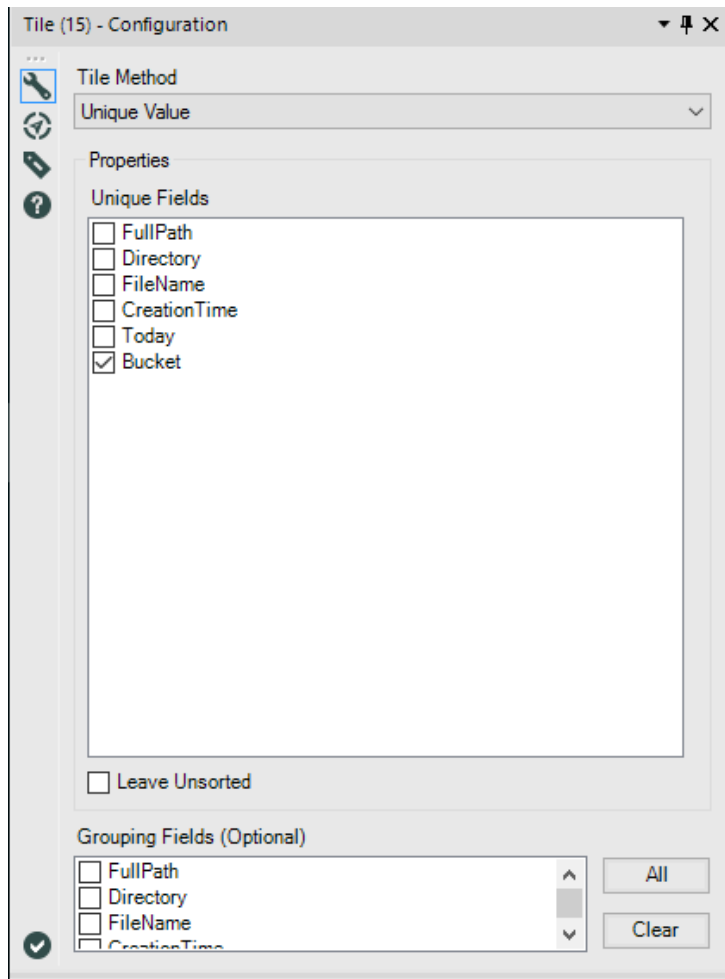
Figure 4-31 - Tile Tool Configuration to assign Tile Numbers

Now, let's output the data to a file called *What's The Policy On That.csv*.

The *What's The Policy On That* data stream should look like the below when it's complete.
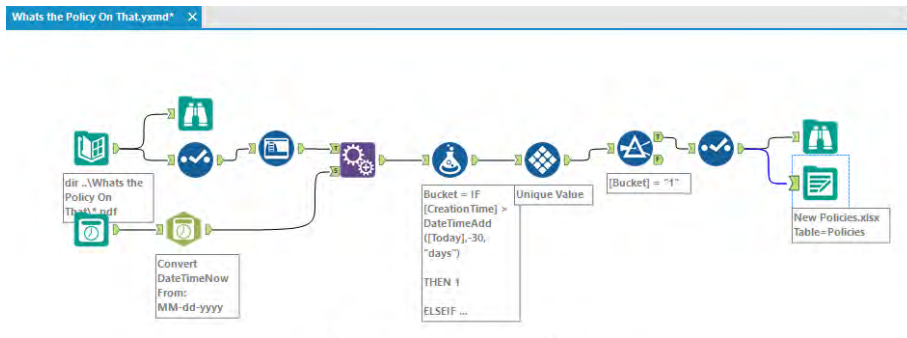
Figure 4-32 - Policy Data Stream when Complete

4.12   Where, Oh Where Have the Three Files Gone?

To...	Alteryx Consultants
Subject	Where, Oh Where Have The Three Files Gone?

Hey,

I know this is a little unorthodox, but it will be a good way to score points with me and my boss.

Usually, I would take care of things like this, but Frank needs me in a meeting.

Frank has lost some important files, and they don't remember what the files are called.

Frank knows they saved the files somewhere on their network drive under a particular folder each, and that you will recognize the names of the files as soon as you find them, but searching the network drive is something Frank doesn't have time to do.

I need you to build a directory and sub directory search for an Excel file (.xlsx) in the "Excel Files" folder, a PDF file (.pdf) in the "PDF files" folder, and an image file (.png) in the "Images" folder in "Chapter 4 – The Direct Approach" > "Looking for Files." Look at the results of each of the three independent file lists, and identify the ones we need.

After that, bring the data together into an output file with the two columns "File Name" and "Full Path", so that by the time the meeting is over, we can direct Frank right to the files.

I really appreciate this.

CHAPTER 5
Cultural Musings

To...	Alteryx Consultants
Subject	Culturally Rich
Attached	2010 GDP Per Capita Per State.xlsx; Museum Data.csv; State Map.xlsx

Hey,

We have some downtime, and I would like to explore the idea that the richer the state, the more likely it is to have museums.

We are going to use the attached list of 2014 Q3 Museums data, 2010 GDP data, and the 2010 Census data.

If you don't have the 2010 Census data package yet, you can download it at http://downloads.alteryx.com/data.html. Just walk through the installer, and when you see the uninstall screen, make sure nothing is selected.

Let's walk through the data prep.

Thanks.

5.1   Tools & Concepts

Tools	Concepts
Allocate Input	Using Census Data
Find Replace	Data Mapping
Join Multiple	Complex Joins
Text to Columns	Splitting field members at delimiters
Create Samples	Splitting the Data stream
Random % Sample	Generate a random number or % of record
XML Parse	Read and Parse the XML snippet

5.2   Allocate Input

Figure 5-1 – Allocate Input

The *Allocate Input* tool allows us to get input from demographic data using specific packages.

Group	Input	Output
Demographic Analysis	None	See below

Note: Specific packages have to be downloaded to use this tool. The *Census Data* is a free package to use here.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream that contains the selected fields from an Allocate Dataset.

Properties Window:

The *Allocate Input Configuration* window has three core components and two sub-components for selection and customization.
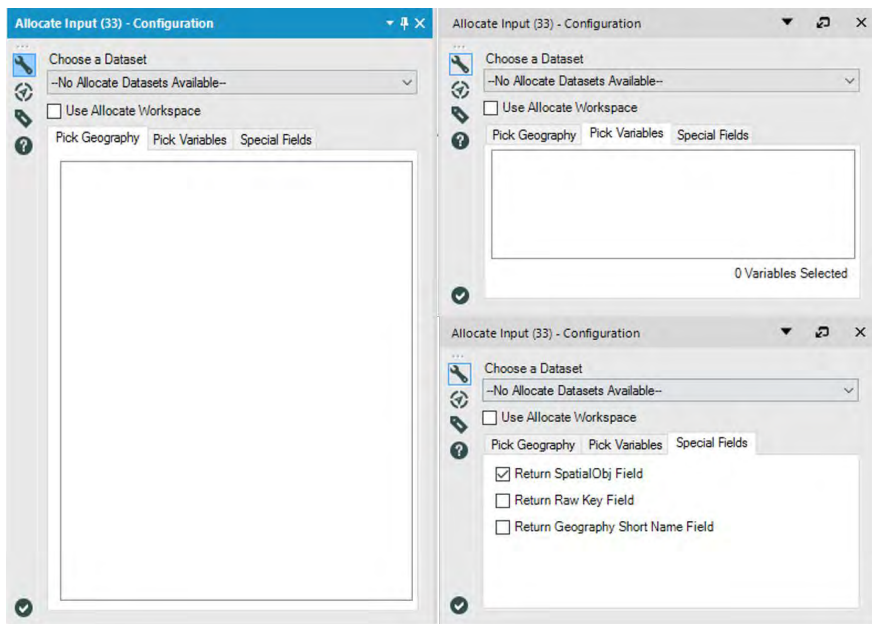
Figure 5-2 – Allocate Input Configuration

Choose a Dataset drop-down allows us to select the dataset associated with one of the packages we have downloaded.

Use Allocate Workspace check box helps us define the fields and geographies we want to use.

The selection window is broken down into tabs. We see different tabs depending on whether or not the *Use Allocate*

Workspace is checked. When unchecked, we see the *Pick Geography*, *Pick Variables* and *Special Fields* tabs.

- *Pick Geography* tab allows us to select the geographic level of the data required.
- *Pick Variables* tab allows us to select the demographic variables needed in the output.
- *Special Fields* tab allows us to select the required special fields relevant to our selections in the output.

When *Use Allocate Workspace* is checked, we see the *Workspace* and *Special Fields* tabs.
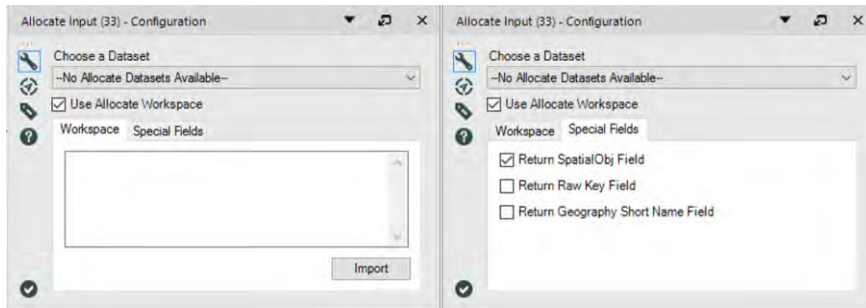
Figure 5-3 – Allocate Input Configuration(continued..)

- *Workspace* can import and modify the *Allocate Workspace* file (.aws), which allows us to save tree-based selections.
- *Special Fields* allows us to select the special fields needed in the output.
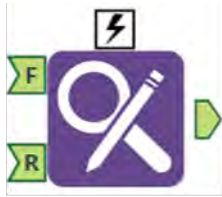
5.3   Find Replace

	The *Find Replace* tool allows us to replace information in a data stream by entering the *F* (find) (target) input with information that matches in the *R* (replace) (source) input.

Figure 5-4 – Find Replace

Group	Input	Output
Join	See below	See below

Note: The *Find Replace* tool is useful in renaming field entries or for replacing specific information in a data stream. In simple words, it works like a lookup function.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input F: A data stream with at least one string field

Input R: A data stream with at least two fields: one mapping to the string field being replaced in the *F* input, and the other field that's required in the output by replacing the original.

Output: The *F* input data stream with an updated field or the *F* input data stream with additional fields from the *R* input.

<u>Properties Window</u>:

The *Find Replace Configuration* window has two main components: *Find* and *Replace*.
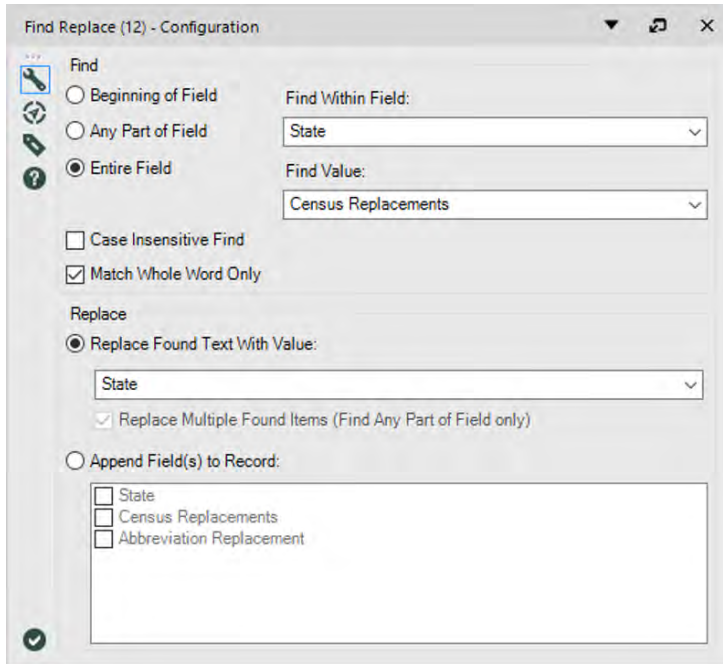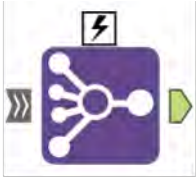
Figure 5-5 – Find Replace Configuration

- *Find* allows us to define the matching criteria.
  - ○ *Find Within Field (Target)* is the field we want to match in the data stream connected to the *F* input.
  - ○ *Search Term Field (Source)* is the field that we want to match in the data stream connected to the *R* input.
    We can select one of the three ways to match the two fields:

- *Beginning of Field* will match only if the *Search Term Field (Source)* matches the beginning of the *Find Within Field (Target)*.
- *Any Part of Field* will match if the Search Term Field record is found anywhere in the *Find Within Field* record.
- *Entire Field* will match if the entire *Search Term Field* record matches with the *Find Within Field* record.

- o *Case Insensitive Find* allows us to match a field irrespective of uppercase or lowercase in both the sources.
- o *Match Whole Word Only* allows us to match only if the *Search Term Field* matches to a word or set of words in the *Find Within Field*. For example, if this option were checked, *apple* would not match *pineapple*, but if it were unchecked, it would match.

- *Replace* allows us to define what the replacement field is
  - o *Replace Found Text With Field* allows us to select a field from the R input with which to replace the values.
  - o *Replace Multiple Found Items (Find Any Part of Field only)* allows us to choose if we want multiple occurrences of a string to be replaced in a single record.
  - o *Append Field to Record* allows us to add specific fields to the data instead of just the replaced value.

5.4   Join Multiple

	The *Join Multiple* tool will perform a full outer join or an inner join between data streams sharing a set of key fields.

Figure 5-6 – Join Multiple

Group	Input	Output
Join	See below	See below

Note: If there are multiple records that match each other (based on the joining field(s)), records from the original data streams be replicated.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

The *Input* must have at least two data streams with the same mapping fields.

Output will be a data stream with all records joined by the key fields. It can either include all records from each data stream or only those that match across all data sources.

Properties Window:

The *Join Multiple Configuration* window looks and functions similarly to the *Join Properties* window with only the following three differences.
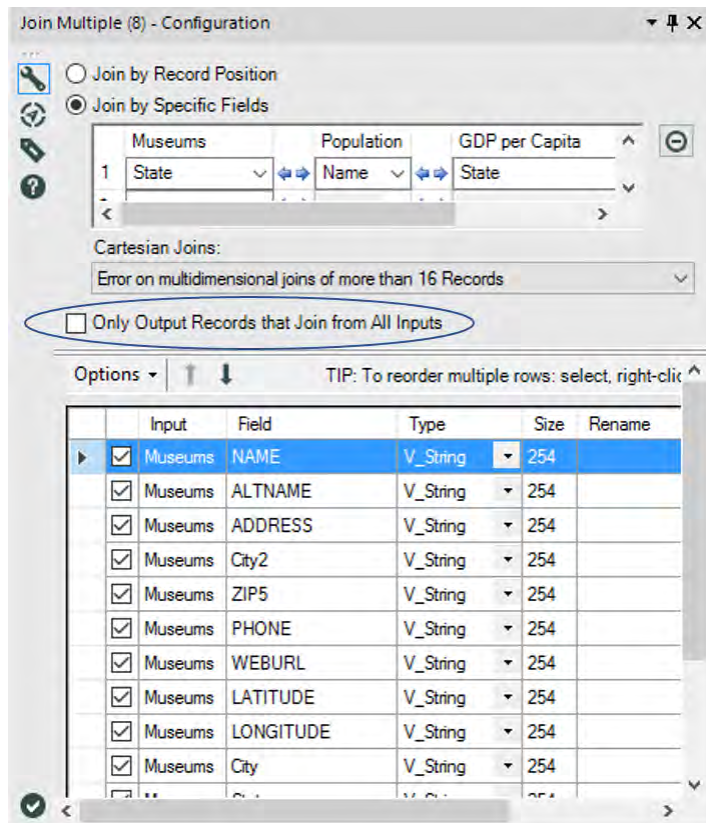
Figure 5-7 – Join Multiple Configuration

- It will have as many columns to join by, as there are input data streams.
- It has the ability to alert in case of too many replications.
- Instead of displaying Venn diagrams for the potential outputs, there is a checkbox (*Only Output Records that Join*

from All Inputs) for deciding if we want the intersection of all of the datasets (the *J* output in the *Join* tool – Inner Join) or the full outer join (when unchecked).

5.5   Text To Columns

 Figure 5-8 – Text to columns	The *Text To Columns* tool will break string fields into multiple string fields based on a delimiter.

Group	Input	Output
Parse	See below	See below

Note: This tool is useful in breaking down strings like email addresses or transaction elements so as to be able to look at the component pieces. It uses a single character to split the fields.

For example, if we type @| into the *Delimiters* text box, *Text To Columns* will split fields at @ and again at |, and not when it encounters @| together.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

The *Input* data stream should have at least one string field.

The *Output* will have the original data with either additional columns or rows generated by splitting the original field.

<u>Properties Window</u>:

The *Text To Columns Configuration* window has the following five core components.
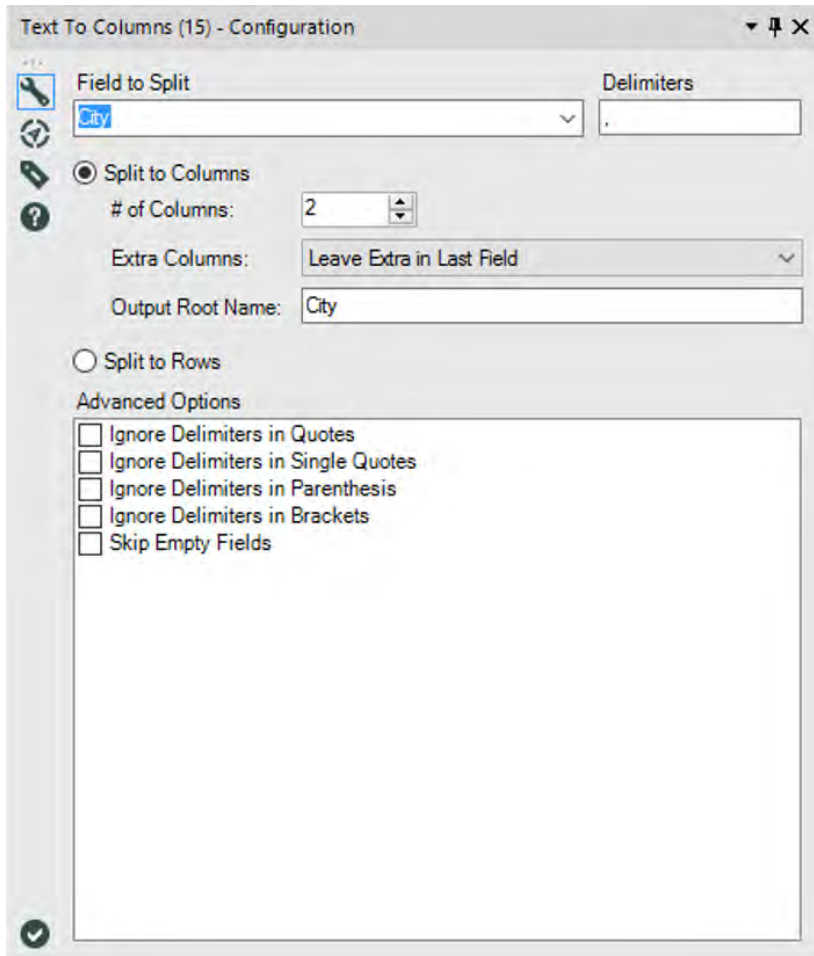
Figure 5-9 – Text To Columns Configuration

- *Field to Split* allows us to select the field to be split from the incoming data stream.

- *Delimiters* are the list of special characters or separators used to split the field by.
- *Split to Columns* allows the creation of new columns for each split and has the following three secondary questions:
    - o *# of Columns* allows us to select the number of new columns to be created.
    - o *Extra Columns* is a drop-down menu that allows us to decide what happens to that data that would be split into more than the number of columns selected.
    - o *Output Root Name* is the beginning of the name of the columns created by the split.
- *Split to Rows* changes the behavior such that new rows are created instead of additional columns for every split that is found.
- *Advanced Options* allow us to decide if there are more granular ways to control the splits.

5.6   Create Sample

	The *Create Sample* tool splits the input records into two or three random samples.		
Fig 5.6.1- Create Sample	**Group**	**Input**	**Output**
	Preparation	See below	See below

Note: In this tool you specify the percentage of records that are in the estimation and validation samples. If the total is less than 100%, the remaining records fall in the holdout sample.

The *Input* data stream will be used with the Predictive Analytics tools.

Output E: This is an Estimation output stream to sample input records randomly. The count of records in this stream will be equal to the percent of total records specified in the Estimation property.

Output V: This is a Validation output stream which contain random sample of input records. The count of records in this stream will be equal to the percent of total records specified in the Validation sample property.

Output H: This is a Holdout output stream which includes any leftover records that were not placed in either the Estimation or Validation samples.

Properties Window:

The *Create Sample Configuration* window has the following five core components.

- *Estimation Sample Percent* allows us to select the percentage of the data which is to be placed in the estimation sample (between 1% and 99%).
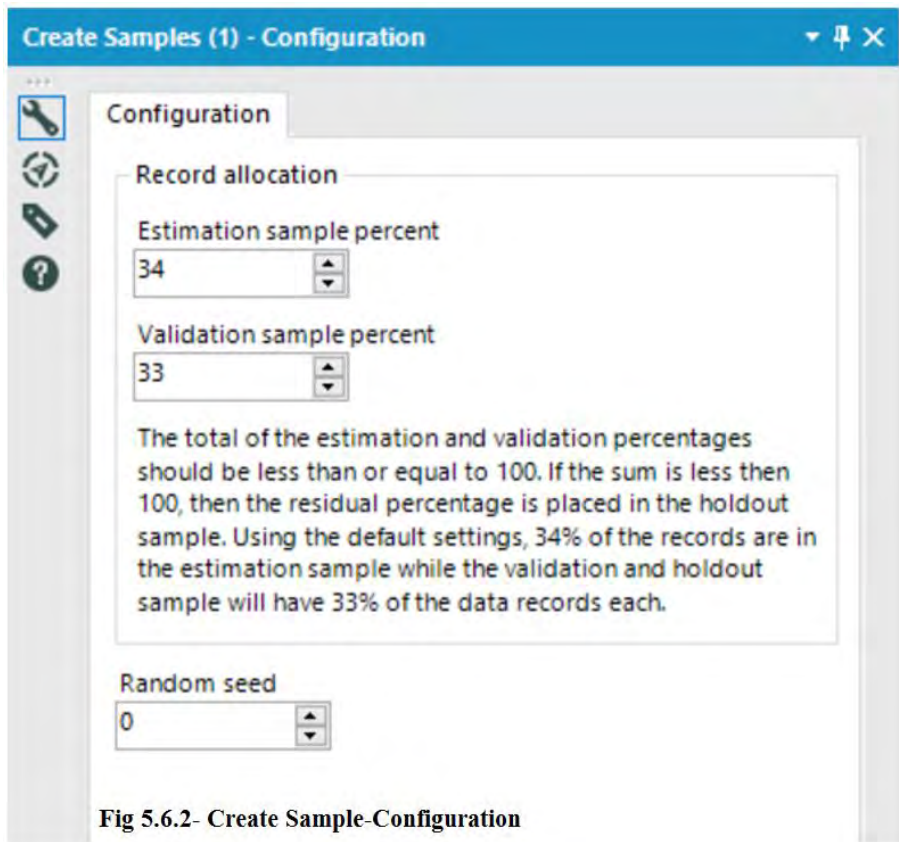
Fig 5.6.2- Create Sample-Configuration

- *Validation Sample Percent* allows us to select the percentage of the data which is to be placed in the validation sample (between 1% and 99%).

5.7   Random % Sample

	The *Random % Sample* Macro tool returns an expected number of records resulting in a random sample of the incoming data stream.
  Fig 5.7.1- Random % Sample	**Group** | **Input** | **Output**
	Preparation | See below | See below

The *Input* data stream should have at least one incoming connection.

The Output will have random percentage of incoming data stream.

Properties

The *Random % Sample Configuration* window has the following three main components.

- *Random N records* allows us to specify N, this N is nothing but the number of records that will be outputted from the Macro.
- *Random N% of Records* allows us to specify the percentage of incoming records to be outputted from the Macro.
- *Deterministic Output* when checked will ensure the same records come out of this macro each time, as long as the random seed is the same at runtime. The random seed

Fig 5.7.2- **Random % Sample- Configuration**

Value can be either typed or selected by using up/down arrows.

5.8   XML Parse

	The XML parsing tool reads in XML snippets and parses them into individual fields.
Fig 5.8.1- XML Parse	

Group	Input	Output
Parse	See below	See below

Note: In versions prior to 9.0, the use of namespaces was "undefined." Namespaces are no longer supported in Alteryx 9.0. If the XML contains namespaces (e.g. "w:document") the namespace will be stripped off in the output (e.g. "document" is returned).

The *Input* data stream should have at least one incoming connection.

The Output parses out information from structured data into separate fields.

Properties

The *XML Parse Configuration* window has the following main components.

- *Field with XML Data* allows us to specify fields that we would like to parse. The dropdown list will display the string fields included in the input.

Fig 5.8.2- XML Parse- Configuration
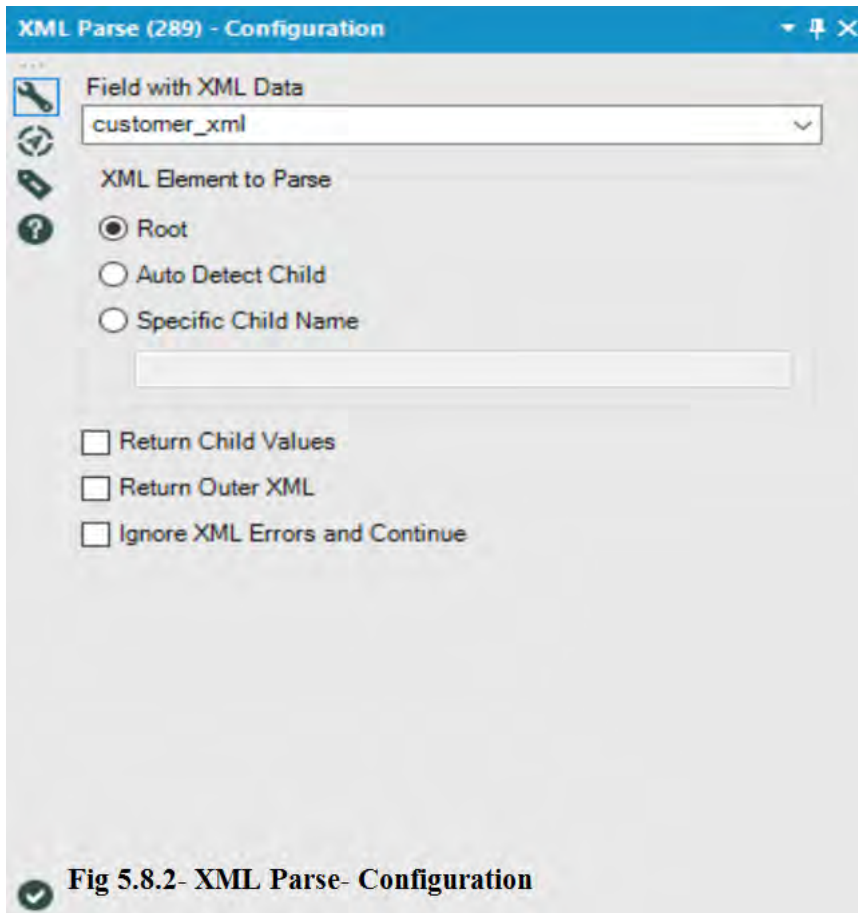
- *XML Element to Parse* allows us to select the particular element between start tag to end tag. This is an optional setting. The XML Element is case sensitive. Here, Root specifies the Parent tag from the XML file. When "Auto Detect Child" is selected, the XML Parse tool will do its best to find the most relevant child data to parse out. Also, we can specify specific child values to parse out only specific tag from the XML file.

- When *Return Child Values* check box is checked, the first set of child values are returned. When unchecked, child values are left out. The child values refer to the elements that are one level deep below the parsed element.
- When *Return Outer XML* is checked, the outer XML tags are returned. One additional field for each parsed field will be created with the suffix "_OuterXML" and the tags are included around the values in the data table.
- We can also select *Ignore XML Errors and Continue* if required.

5.9 Culturally Rich
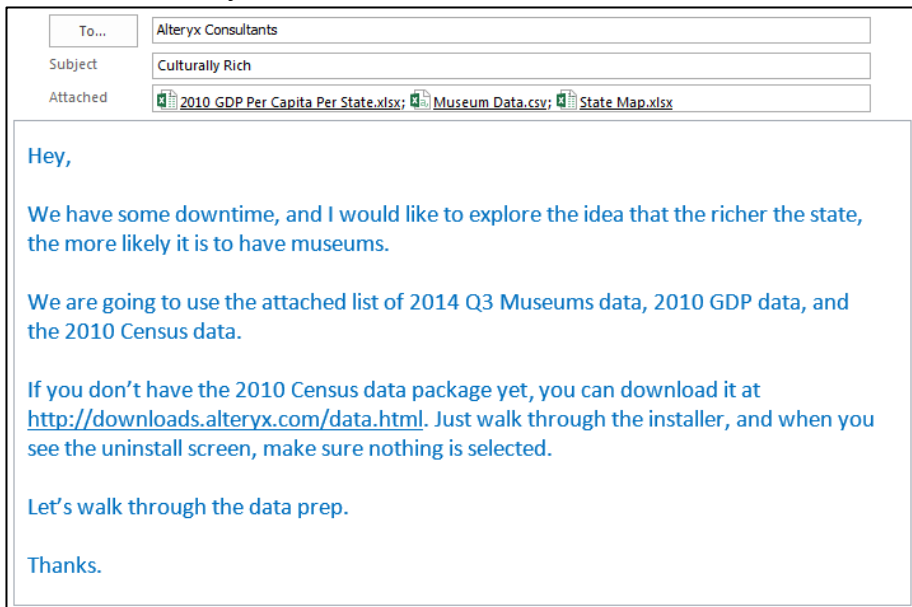
The files required for this chapter have been placed in clearly marked sub-folders in *Chapter 5 – Cultural Musings > Culturally Rich*. We will start by bringing in each of the four data sources to

see what we have. Let us open all three files and then connect to the *Census* data.
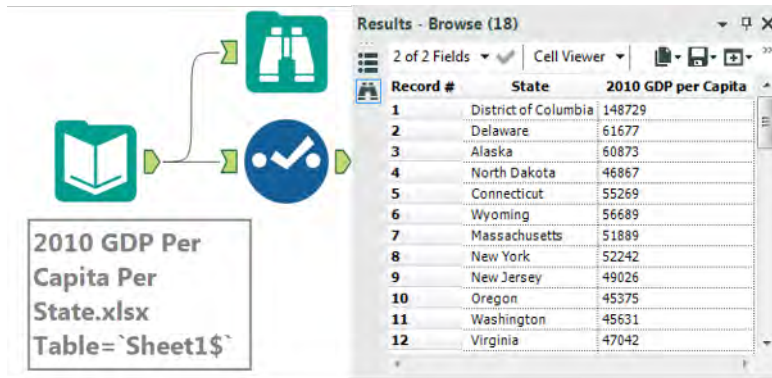
Figure 5-10 – Culturally Rich – GDP data

Figure 5-11 – Culturally Rich – Museum Data

As can be seen in these two files, the names of the states appear in different formats, with the *GDP per Capita* file having the entire state name spelled out, and the *Museum* data having only the two-letter state abbreviation within the city field.
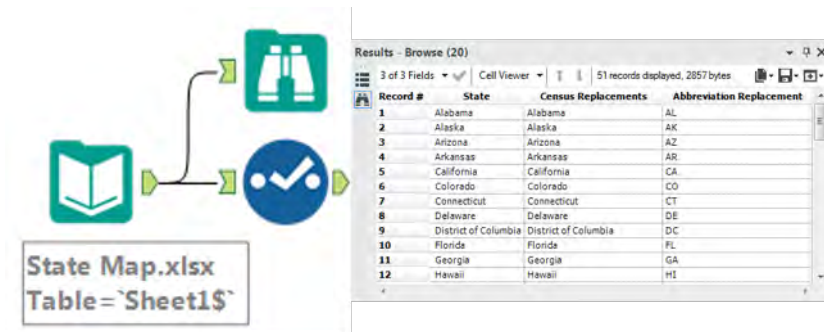
Figure 5-12 – Culturally Rich – State Map

However, the third file has three separate columns with either the complete state name or its abbreviation. This is because *State Map.xlsx* is a file created especially for the purpose of field mapping by linking the data sources from the *GDP*, *Museum* and *Census* data sets, which all have state identifiers in different formats.

The following are the observations from the data in the three files we connected to.

- The *GDP per Capita* data does not need any preparation before the join.
- The *State* associated with each museum needs to be parsed out of *City* and then mapped to the *GDP per Capita* name.
- The *State Map.xlsx* file can be used for mapping all data sources together.

Let us now bring in the *Census* data in order to plan what needs to be done with that data stream.

We shall use an *Allocate Input* tool on the canvas, then choose the *US Census 2010 – Most Recent Vintage* dataset. Let us

select the *States* option under *Pick Geography*, and click the
checkbox to select all states.
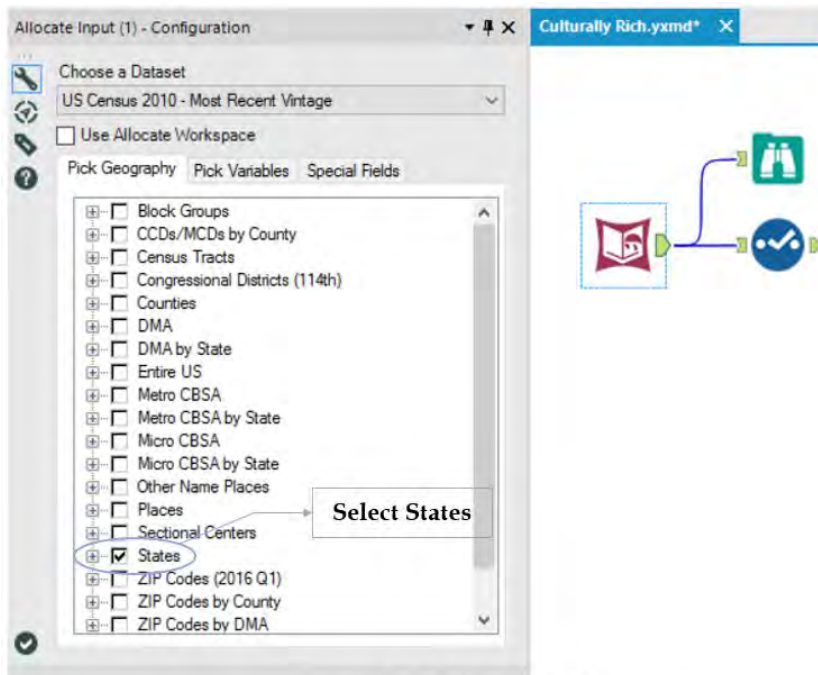
Figure 5-13 Culturally Rich
Allocate Input Configuration – Pick Geography

The purpose of bringing in the *Census* data is to compare
the *GDP* to the number of museums in each state instead of the
GDP per Capita. We can use the two data sources to generate the
GDP value.

Now that the goal is identified, figuring out the variable
needed becomes easy.
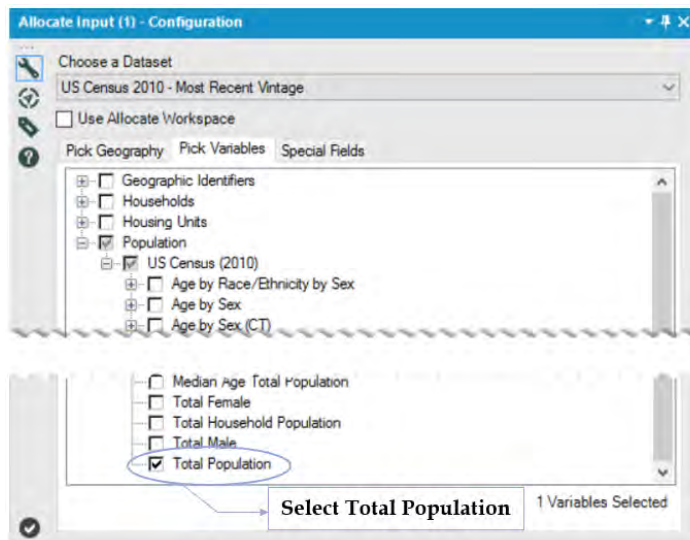
Figure 5-14 Culturally Rich
Allocate Input Configuration – Pick Variables

Under *Population* and *US Census (2010)*, the last option is *Total Population*. This signifies the total number of people that have lived in each chosen State (Geography).

Since we need only this information, click on the *Special Fields* (as shown in the image below) and uncheck all options.
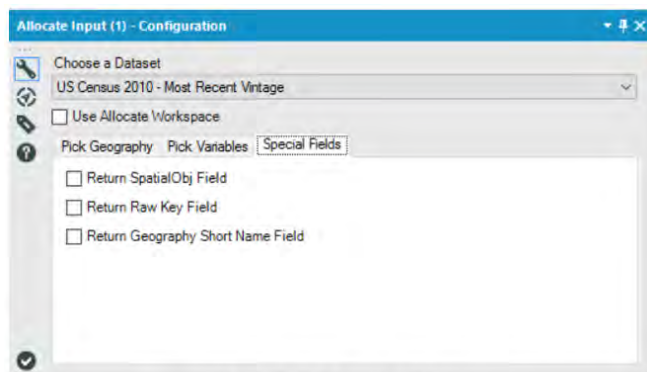
Figure 5-15 Culturally Rich
Allocate Input Configuration – Special Fields

Now that we have started with the final data stream, let us take a look at the data that comes out of the *Census* data connection.

Figure 5-16 – Culturally Rich Allocate Input Browse configuration

The following two points about this data stand out:

- Even though we have only selected one geography and one variable, we still have three fields. This is because the *Key* field uniquely identifies all geographies so that even if we have regions with the same name, they have unique identifiers. In this case, because they are States, we do not need the *Key*.

- There is a field called *P0010001*. This is because the data is stored with keyed column headers. We will simply rename the field to *2010 Population* in the *Select* tool, as shown in the following image.
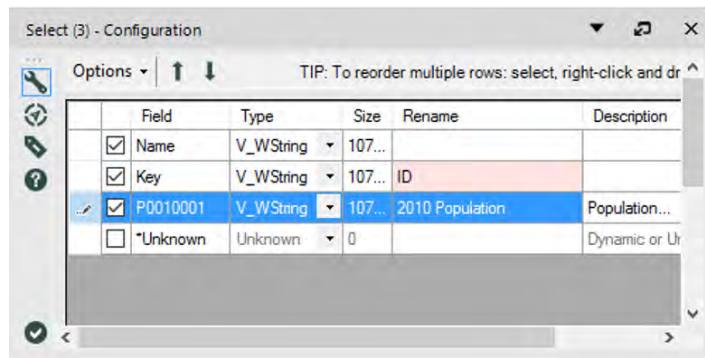
Figure 5-17 – Culturally Rich Census Population data
Select configuration

Now, the remaining preparation is to map the state names from the data sources back to the *GDP per Capita State* names before joining these data streams.

Let us start with mapping the *Census* data since it requires only one step before the join. In order to do this, we will use the *Find Replace* tool. Connect *Census* data stream to the *Find (F)* input and the mapping data stream to the *Replace (R)* input as shown in the following image.
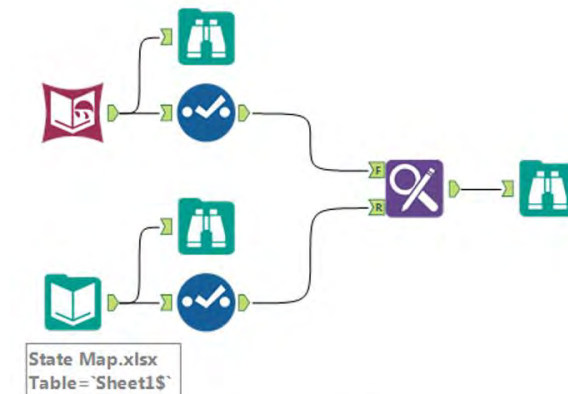
Figure 5-18 – Culturally Rich
Connecting census data to Find and State Map data to Replace

Now, let us take a look at the settings of *Find Replace* tool.
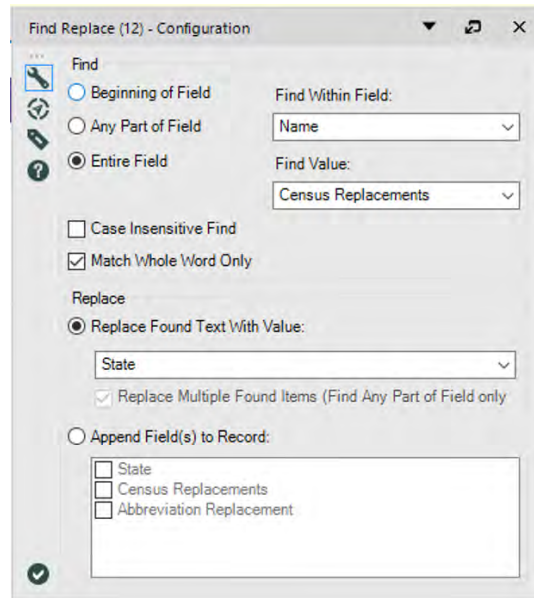
Figure 5-19 – Culturally Rich Census first Find Replace configuration

Since the map has been structured in such a way that the field *Census Replacements* has all of the *Census State* names in it, we will be looking for the *Entire Field* matches of the *Census Replacements* in the *Census Name* field. We also want to make sure that we replace the fields found in the *State* field (which has the *State* names from the *GDP* data). Since the *Census* data stream and the *GDP* data stream share a mapping field, let us get the *Museums* data ready.

We will first remove all of the fields except *Name* and *City* in the *Select* tool following the *Input Data* tool. Then, we will use Random % Sample tool with following settings to randomly retrieve 50% *Museums* data.
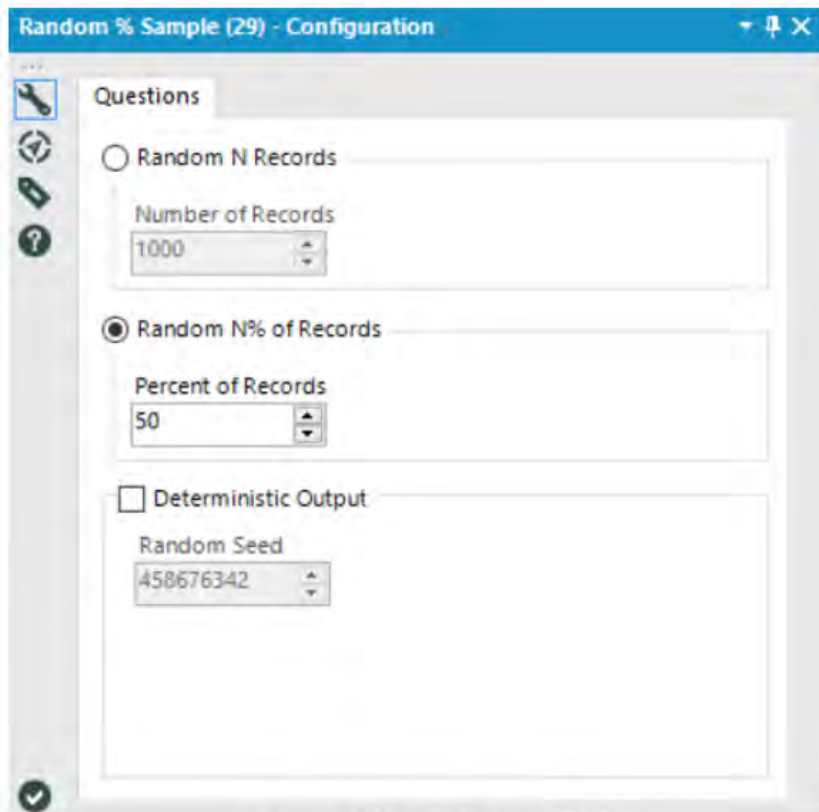
**Figure 5-20 – Culturally Rich
Random % Sample configuration**

Next, we will split the data apart. To do this, use the *Text to Columns* tool and add it to the end of the *Museums* data stream with the following settings.
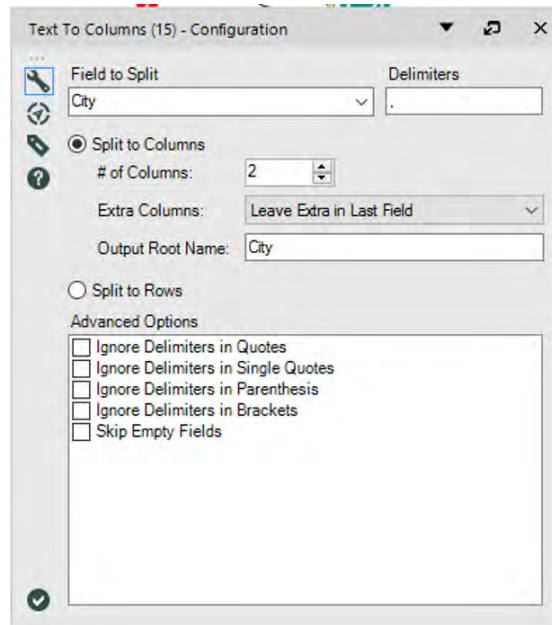
Figure 5-21 – Culturally Rich Text to Columns
configuration to split at a separator

Then, let us look at the updated data after it has been modified with the *Text to Columns* field.

Figure 5-22 – Culturally Rich
Modified Museum data

When we look at the *Browse* tool, we can see that *Location2* has a *red triangle* at the top-right corner of the cell. It is an alert indicating that Alteryx recognizes a potential issue with the data. If we hover the mouse over the cell, we will see a little textbox pop-up that reads "This cell has leading spaces". This means the first character of the cell is a space (attributed to the split and a spacer after "," delimiter) and that we should address it. We will do this by using the *Formula* tool to create a new field, we will call *State*.
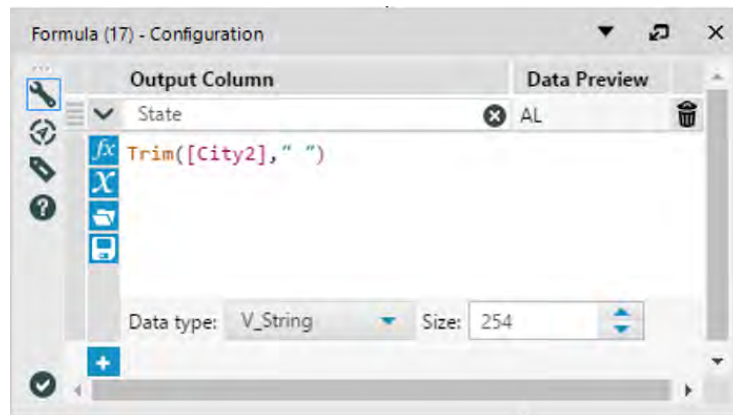
Figure 5-23 – Culturally Rich Trimming Museum data using Formula

This time, we will be creating a string. If we look below the *String* functions, we can find three functions designed to remove one string from another. We will use the *Trim* function because we do not know if there are cells having trailing spaces which would cause issues with mapping.

Since we are removing spaces from the field, we could simply write the formula *Trim([Location2])* because the function removes white spaces by default. However, for the purpose of

clarity and readability, the best practice is to explicitly define all the variables. In this case, it would mean writing the formula as *Trim([Location2]," ")*.

Now that we are left with clean two-character *State* abbreviations, we can map them to the *GDP State* names. Once again, we will bring a *Find Replace* tool onto the canvas and connect the mapping data to the *Replace* input.
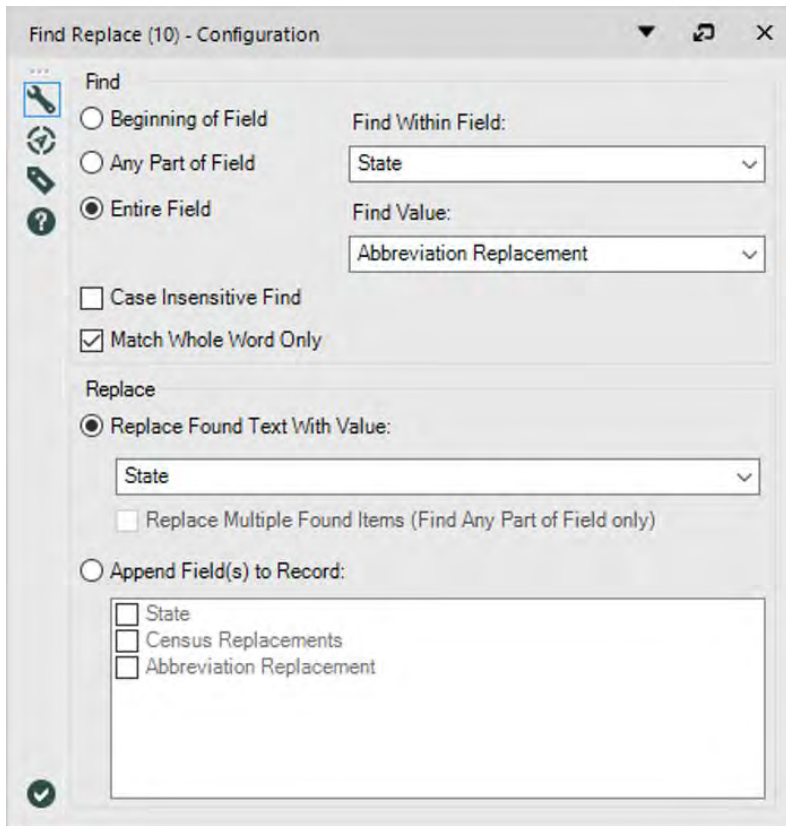
**Figure 5-24 – Culturally Rich Second
Find Replace configuration**

Now that we have prepared all the contributing data streams, this is what the data stream should look like.
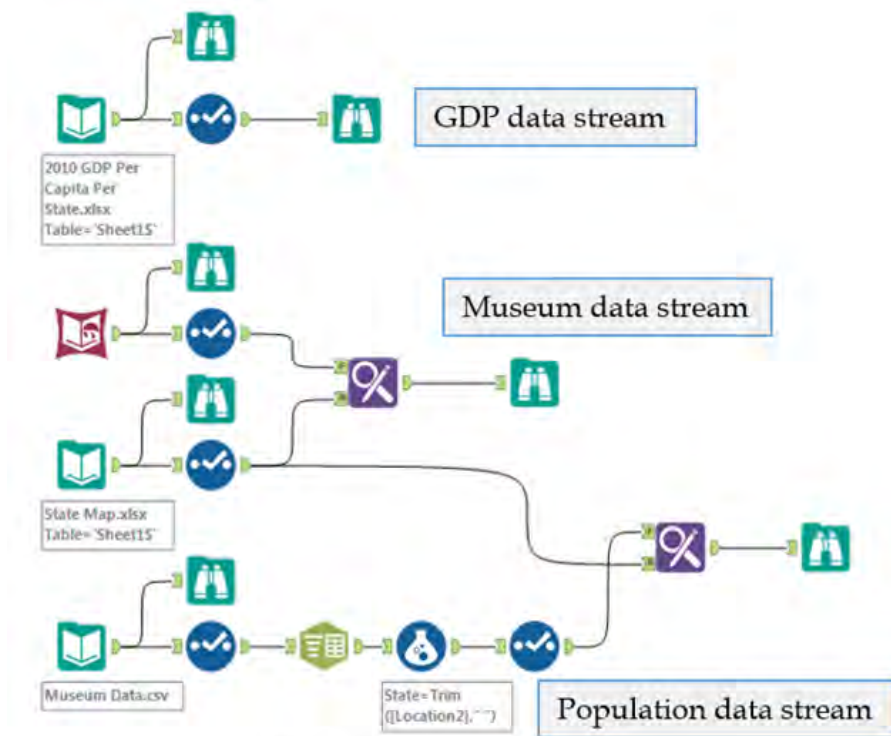
Figure 5-25 – Culturally Rich
Consolidated Modified data streams

Notice that we have crossed the data streams. From a technical perspective, it does not matter if they are crossed. However, practically, the workflow becomes harder to read when the data streams are crossed frequently.

Best practice is to create workflows with as few crossovers as possible. In this case, we leave it as-is because there will invariably be at least one cross over in this module.

We could use a couple of *Join* tools now, but since all data streams share the same key (*State* name), we can use a *Join Multiple* tool to join all three data streams at once. We will connect the *GDP Per Capita*, *Museum,* and *Population* data streams to a *Join Multiple* inputs in the same order. We can then rename the connectors similarly and configure it as shown in the image.

One thing to note is that Alteryx renames conflicting fields with the data stream name and an "_". For example, since there are two fields that are named *State*, the second occurrence is renamed with the data stream as an identifier: *Museum_State*.
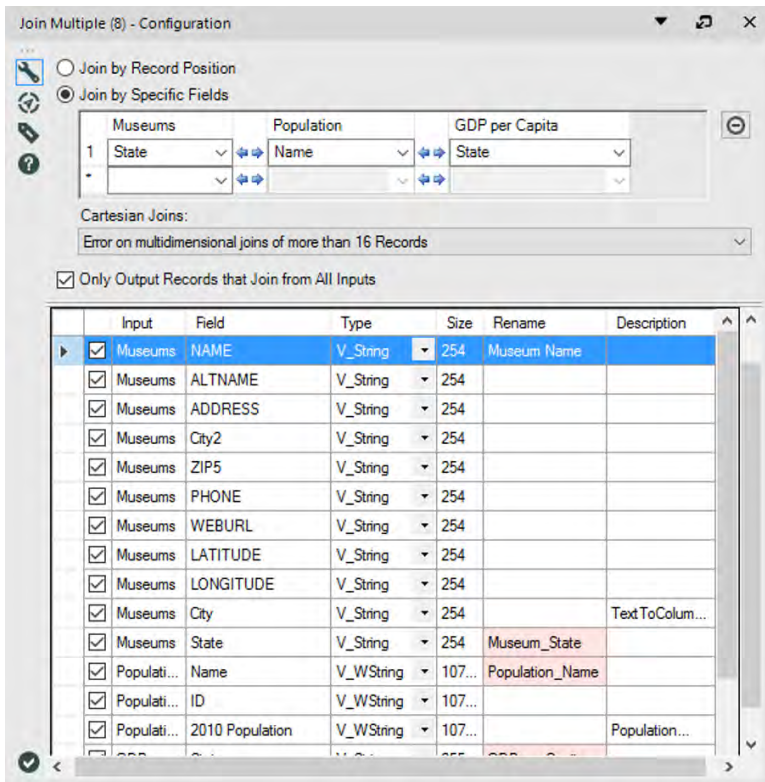
Figure 5-26 – Culturally Rich Second
Join Multiple configuration

This is what the data stream would look like.
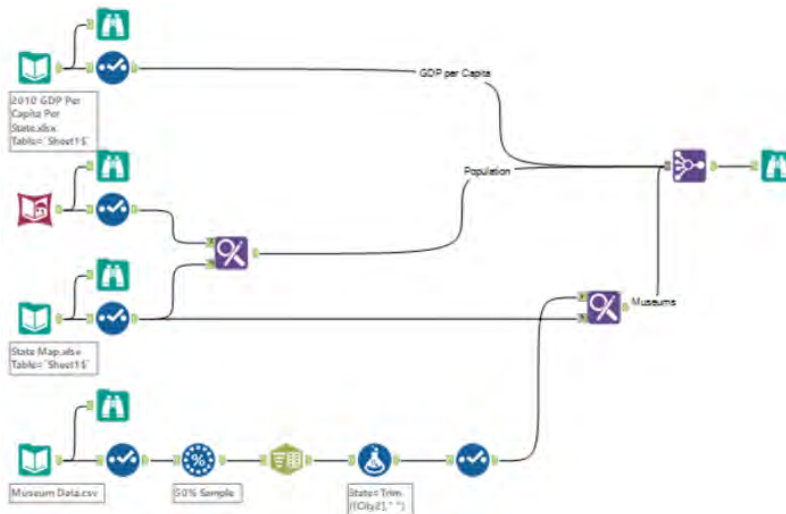
Figure 5-27 – Culturally Rich
Unified data stream

Once we perform the following steps, our data preparation will be complete.

- Add a field called *Total GDP* (product of GDP Per Capita and Population).
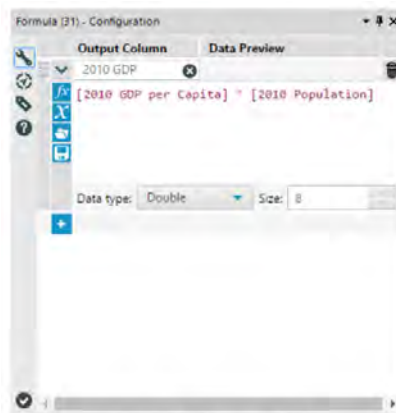
Figure 5-28 Culturally Rich
Total GDP calculation

- • Export the data to a file called *Culturally Rich*.

The *Culturally Rich* data stream should look like this when it is complete.
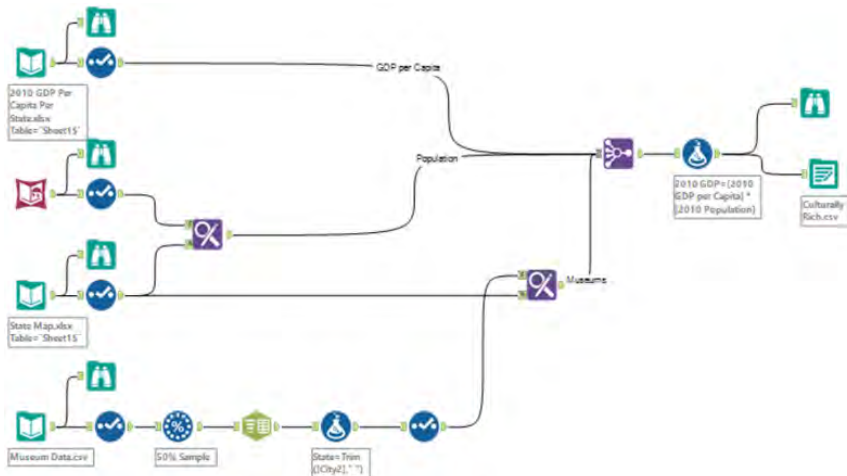
Figure 5-29 – Culturally Rich Complete data stream

5.10 Culturally Divided

To...	Alteryx Consultants
Subject	Culturally Divided

Hey,

I want to create a map in another tool that will approximate the cultural appreciation of each state by identifying what the museum's per capita rates are. This may not be an appropriate measure, but I'm curious. Note that this time, we will need to take a count of the museums per state, not just apply the state data to each museum.

It would also be nice if I had a flag for the comparison to the average museum's per capita. (Above average, average, and below average).

I need to run to a meeting; do you think you can get this done by the time I get back?

Thanks.

CHAPTER 6

The Sport Report

To...	Alteryx Consultants
Subject	The Sport Report
Attached	Major Sport Teams.xlsx

Hey,

I got a call from a sports bar chain. They want to see a rough report that could show them exactly where the major professional baseball, basketball, football, and soccer teams in the United States and Canada are so they can target new locations.

I want you to help me build a report with the following properties.

- Every sport should start on a new page.
- There should be a generic company header and footer (we can use the default for now).
- It should say the name of the sport.
- There should be a map of all of the stadium locations for that sport.
- There should be a bar graph that has the total count of teams by the governing association.
- There should be a table that has the name of the association, the league, or conference, the division, the team, the city, the home park, the street address, and the zip code, for each team. (The table should be allowed to take in as many pages as necessary.)

Thanks.

6.1 Tools & Concepts

In this chapter, we will be covering the following Tools and Concepts:

Tools	Concepts
Charting	Report Building
Create Points	Spatial Objects
Layout	Maps
Render	
Report Footer	
Report Map	
Report Header	
Table	
Email	
Image	
Map Legend Builder	
Map Legend Splitter	
Overlay	
Report Text	

To the reader,

We will be covering the basics of report building in Alteryx. However, since building reports in Alteryx is an advanced technique, we will not be exploring the Report tools as completely as the tools in other sections.

USEReady

6.2 Charting

	The *Charting* tool creates a chart object for reporting.		
Figure 6-1 - Charting	**Group**	**Input**	**Output**
	Reporting	See below	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Input: A data stream with at least one numeric field.

Output: A data stream with a record for every element in fields grouped by, a column for each grouping field, and a field called *Chart* with a *Chart object* in each record.

This tool creates a reporting object.

<u>Properties Window:</u>

The *Charting Configuration* window has four components: *Chart Type, Select Field(s) To Chart, Preview,* and *Setup Chart Appearance.*
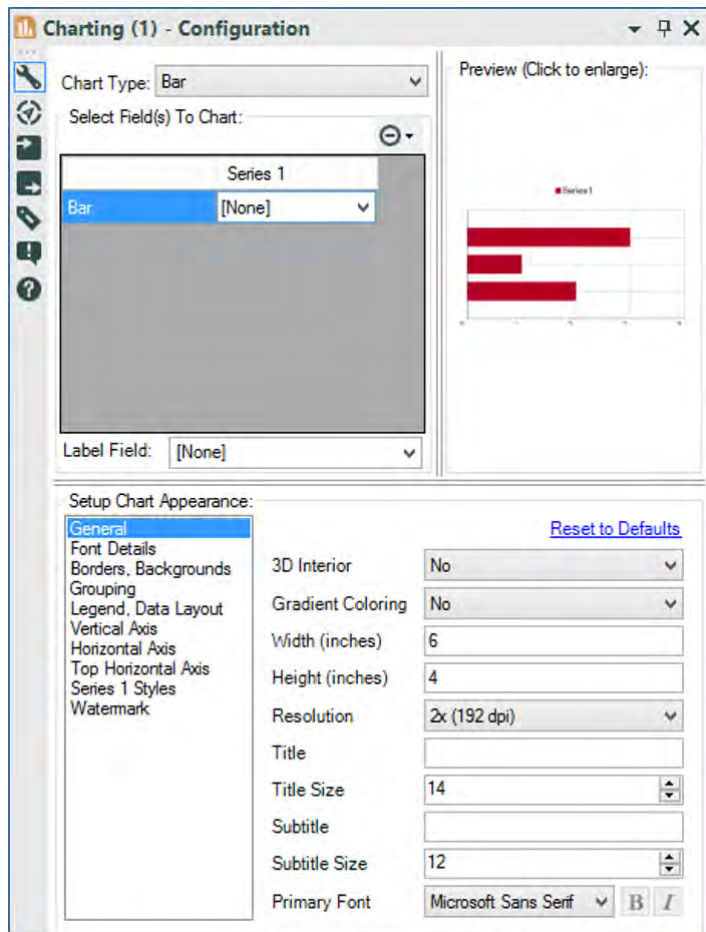
Figure 6-2 – Charting Configuration

1
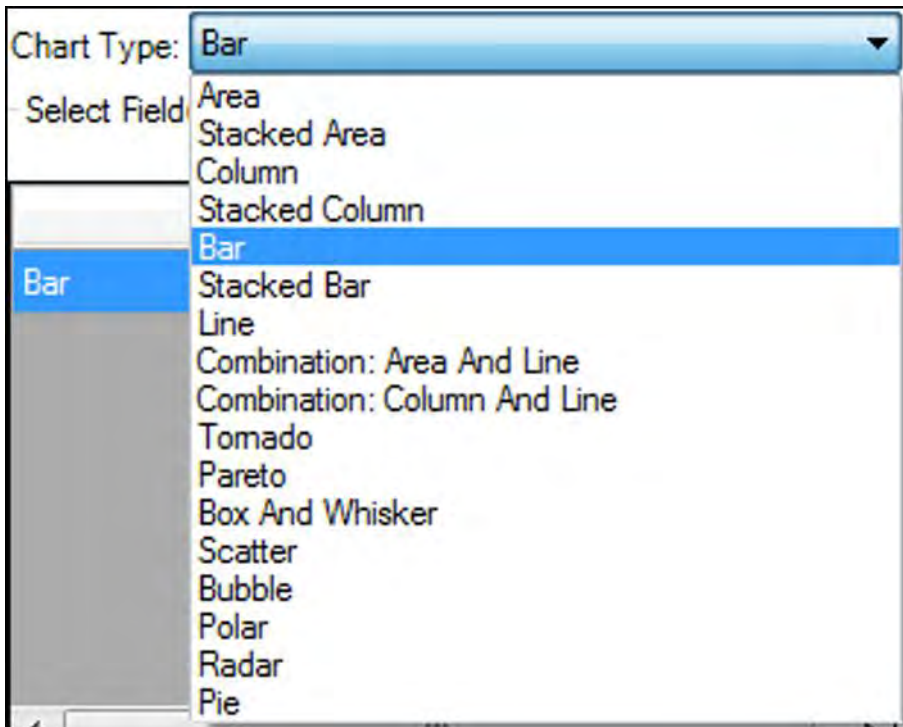
Figure 6-3 - Chart Types

Chart Type allows us to select from a list of charts.

For each of these charts, the *Setup Chart Appearance* will reflect the appropriate settings. We will be walking through the basic settings when we use a bar chart.

Select Fields to Chart is where we can select each of the individual items that represent a series in the data. When working with a *Bar* chart, this means we can take additional columns to create additional bars for each field. At the bottom of this window, there is an ability to select the *Label Field*, which allows us to change the label on the bars.

Preview shows us a sampled version of what our resulting chart will look like. It only shows the formatting and does not refer to the real data stream. If we click on the image, it opens in a pop-up window so we can get a clearer look at what we are designing.

The *Setup Chart Appearance* has two main components: the list on the left side, which allows us to select which aspect of the chart we want to modify, and the settings on the right side that we can modify. Most of the settings we will see in the following options are fairly self-explanatory, and some may have secondary questions that appear when we select them. We will be covering all of these options at a conceptual level.
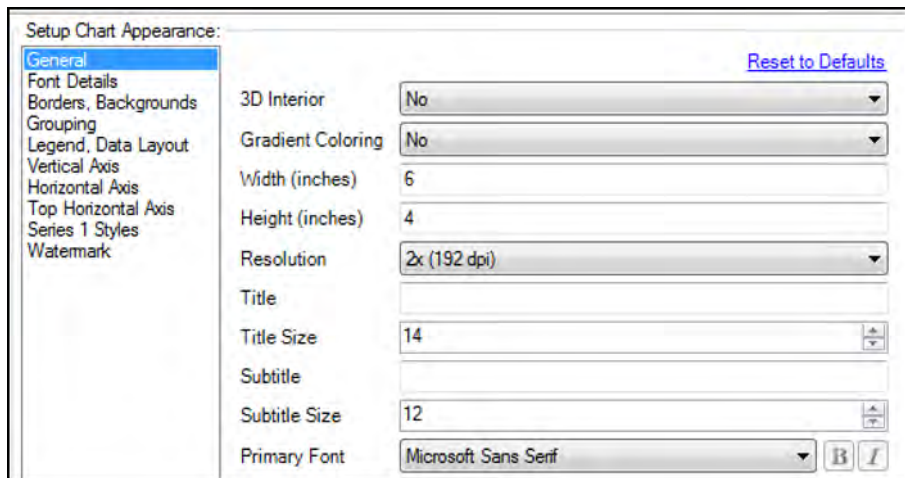
Figure 6-4 - Chart Appearance - General

Under *General* setting, the first two settings change the formatting of the bars. *3D interior* makes the bar appear 3D while *Gradient Coloring* makes the bars look rounded. With *Width* and *Height*, we can set the dimensions of the graph image. *Resolution* allows us to set the quality of the image. *Title* and *Subtitle* allow

us to type a title onto the graph, while *Title Size* sets the size of the font. *Primary Font* allows us to set the default font used for the graph.
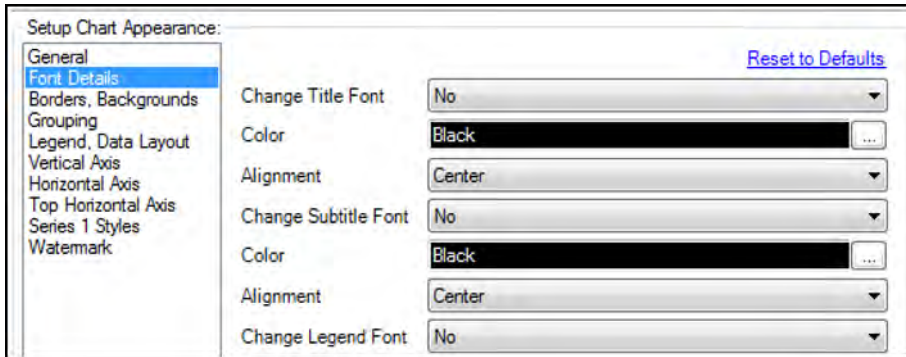
Figure 6-5 - Chart Appearance - Font Details

Font Details allow us to change the font settings for the Title, Subtitle, and Legend.
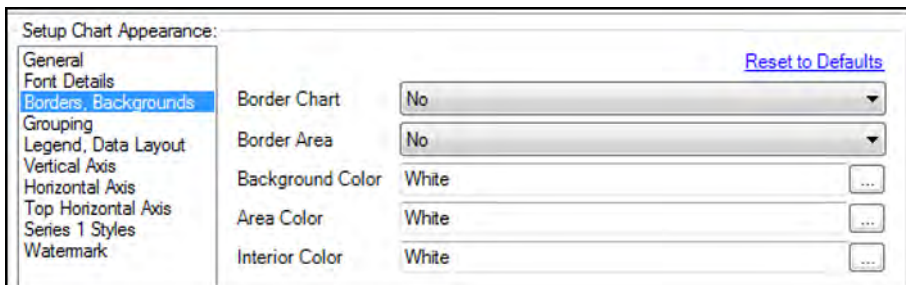
Figure 6-6 - Chart Appearance - Borders, Backgrounds

Borders, Backgrounds allow us to change the color and size of borders throughout the graph (excluding the bars).

Figure 6-7 - Chart Appearance - Grouping

With *Grouping*, for each unique element in the field(s) selected, a different chart will be created (and a different record will be created in the output).

Figure 6-8 - Chart Appearance - Legend, Data Layout

In *Legend, Data Layout*, the first five tools allow us to customize the legend. *Plot Data By* changes the way the data is colored within the graph. The default is to color by series; h. However, if we change this setting from *Record* to *Field*, we will be able to change the color of the data by a specific field.
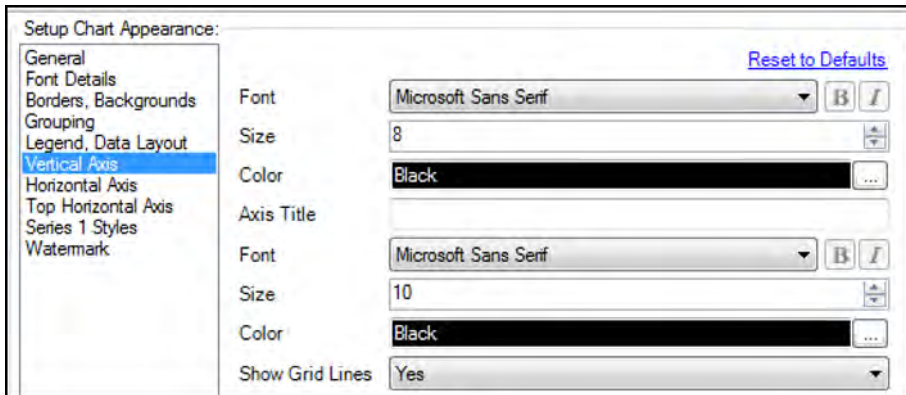
Figure 6-9 - Chart Appearance - Vertical Axis

Vertical Axis allows us to customize the font of the *Axis Title* and labels, as well as turn the gridlines on or off.
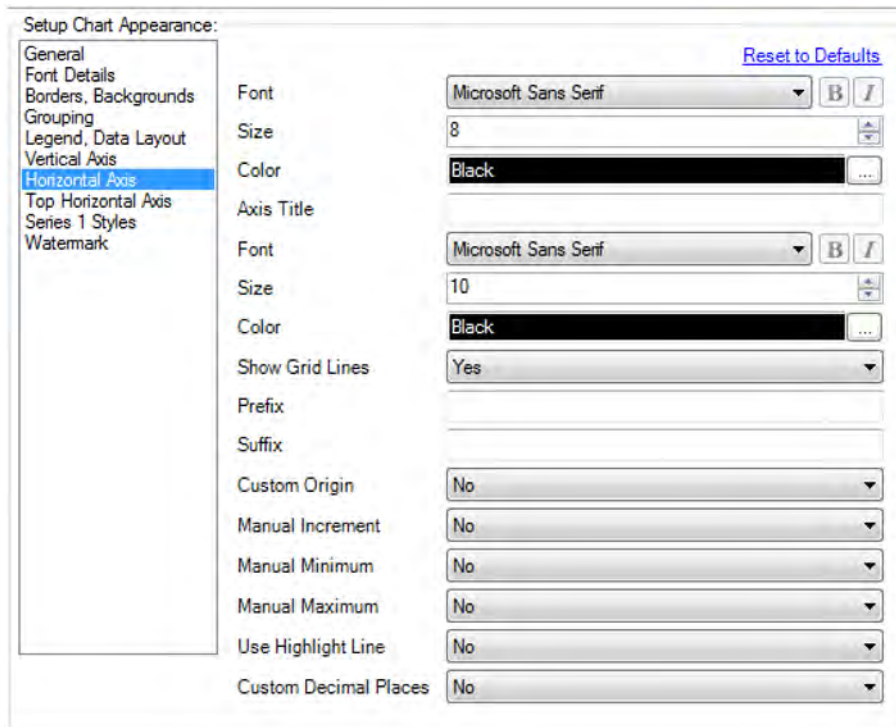
Figure 6-10 - Chart Appearance - Horizontal Axis

Horizontal Axis allows us to customize the horizontal axis. The top half is the same as the *Vertical Axis* settings. However, this being a numeric axis, it gives us the ability to customize the range, set a reference (highlight) line, and change the number format.

Figure 6-11 - Chart Appearance - Top Horizontal Axis

The *Top Horizontal Axis* allows us to set, define, and customize a secondary axis for the chart.
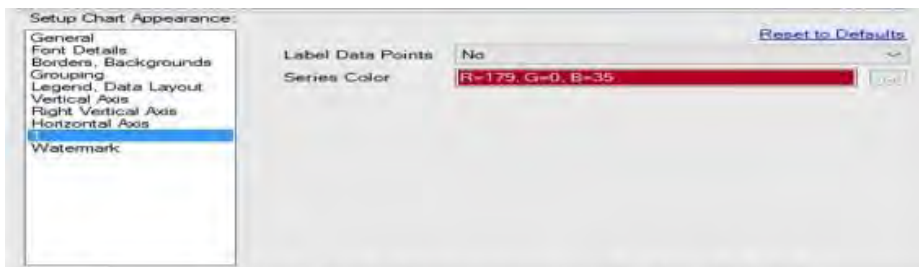
Figure 6-12 – Chart Appearance – Series X Style

The *Series X Styles* allows us to set the style for the series X so we can differentiate them.

Figure 6-13 - Chart Appearance - Watermark

Watermark allows us to set a watermark on the graph.

6.3   Create Points

Figure 6-14 – Create Points	The *Create Points* tool converts decimal latitudes and longitude columns into a new field called *Centroid* with a point object for each record.		
	Group	**Input**	**Output**
	Spatial	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with latitude and longitude columns.

Output: The incoming data stream with an additional field containing point objects.

This tool creates a spatial object field.

<u>Properties Window:</u>

The *Create Points Configuration* window has two core components.
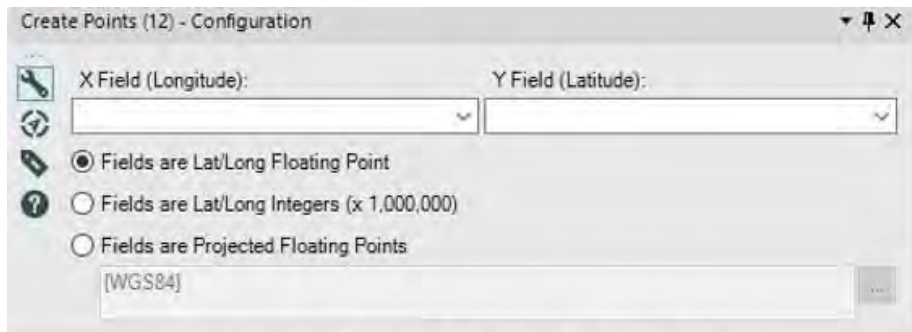
Figure 6.15 – Create Points Configuration

The *X Field (Longitude)* and *Y Field (Latitude)* are drop-downs that allow us to select the longitude and latitude fields from the input data stream.

The single value list following the field selection allows us to define how the latitude and longitude are defined.

- *Fields are Lat/Long Floating Point* means the fields are in a standard decimal format.
- *Fields are Lat/Long Integers (x 1,000,000)* means the fields have been standardized to an integer format.
- *Fields are Projected Floating Points* allows us to identify a specific formatting method to pinpoint spatial fields.

6.4   Layout

Figure 6-16 - Layout	The *Layout* tool creates a data stream with reporting objects and allows us to structure their layout.		
	Group	**Input**	**Output**
	Reporting	See below	See below

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with at least one report object field.

Output: The original data stream with a single *Layout* object that combines other report objects. Depending on the settings, the original report object fields may be removed.

This tool creates a reporting object.

Properties Window:

The *Layout Configuration* window has the following five core components.
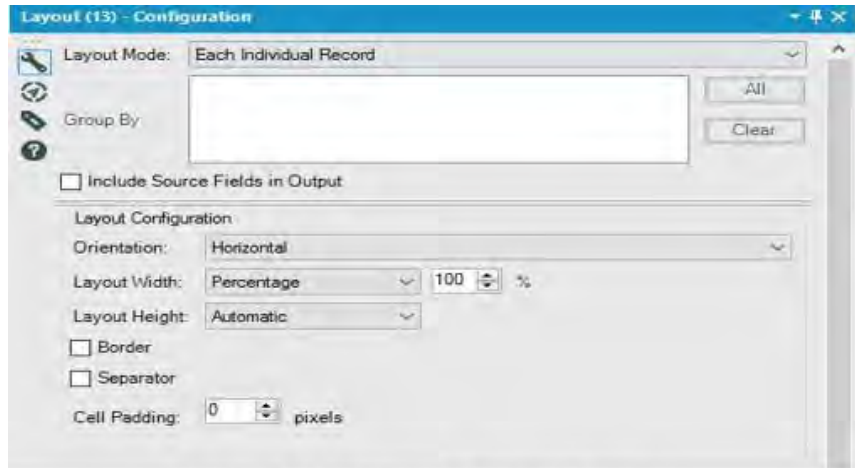
Figure 6.17 – Layouts Configuration

- *Layout Mode* allows us to define the method used to combine the different report objects.
- *Group By* (when using the *Each Group of Records* option from the *Layout Mode* drop-down) allows us to define what is to be combined with the group of records that share a field(s).
- *Include Source Fields in Output* allows us to decide whether or not the original report fields that have been combined should be in the outgoing data stream (assuming we have chosen the *Each Individual Record* from the *Layout Mode* drop-down).
- *Layout Configuration* allows us to set some parameters for the look and feel of the resulting *Layout* object(s).
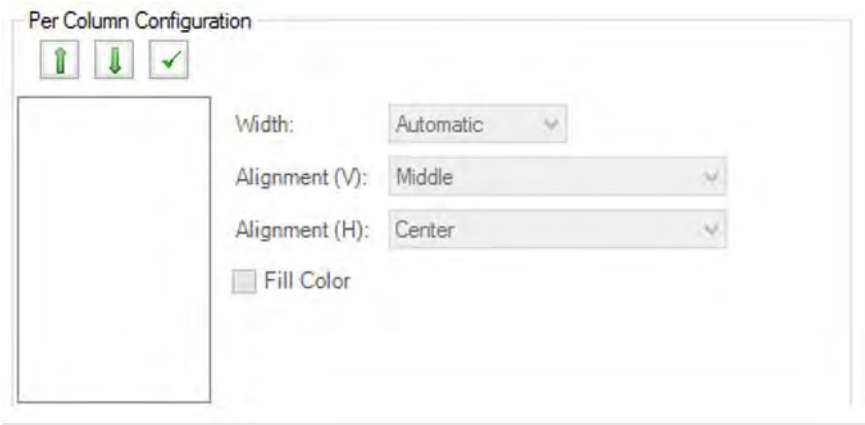
Figure 6-18 – Per Column Configuration

Per Column Configuration has two setups. *Each Individual Record* mode gives us the ability to select the reporting objects that we want to combine across a record, while the other two modes allow us to set the field(s) that we want to combine when creating a single record per group.

6.5   Render

The *Render* tool creates a file that allows us to see the structured reporting that we have designed.		
Group	**Input**	**Output**
Reporting	See below	File

Figure 6-19 - Render

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

> Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.
>
> *Input:* Any data stream with at least one report object.
>
> *Output* is a file that has the structured report we've designed.
>
> If we require more than one report object, we need to combine them with the *Layout* tool and use the *Layout* object.

Properties Window:

The *Render Configuration* window has three core components.
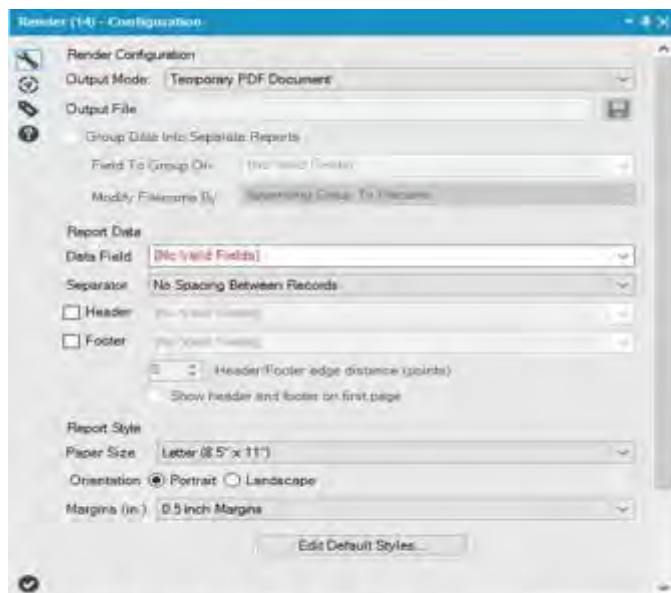
Figure 6.20 – Render Configuration

Render Configuration allows us to set the type and location of the report output.

Output Mode allows us to define a temporary file in one of a few different formats, as well as define a permanent file to write to.

Output file lets us define a path and filename for the file.

Group Data into Separate Report, when selected, allows us to create multiple output reports based on a grouping field selected in *Field to Group On*.

Modify Filename By allows us to define how each of the files should be named in the output.

Under *Report Data*, *Data Field* allows us to select the field in the body of the report.

Separator allows us to set what is between each record.

Header and *Footer* allow us to define the header and footer fields respectively.

Header/Footer edge distance (points) is the number of points everything is from the edge of the paper.

Show header and footer on the first page allows us to turn those aspects on or off for the first (often title) page of the report.

Report Style allows us to set the paper and layout properties of the final report.
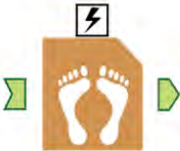
Paper Size allows us to define page dimensions.

Orientation defines the way the report is laid out.

Margins (in.) define the margin width in inches.

Edit Default Styles allows us to change the default fonts and coloring.

6.6   Report Footer

Figure 6-21 – Report Footer	The *Report Footer* tool allows us to create a footer to be used in reports.		
	Group	**Input**	**Output**
	Reporting	Data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream, with the additional field *Footer*, which has a *Layout* object in each record.

This tool creates a reporting object and should be used after report elements have been joined together.

<u>Properties Window:</u>

The *Report Footer Configuration* window allows us to define the footer.
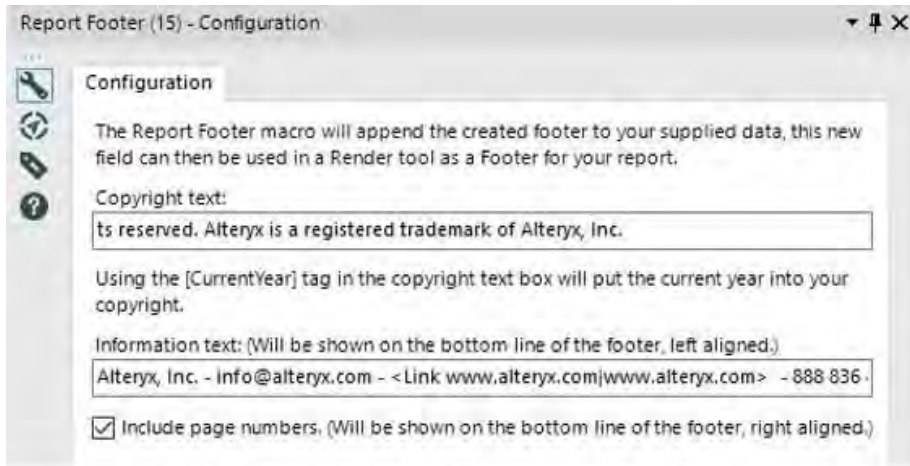
Figure 6.22 – Report Footer Configuration

Copyright text is intended for a copyright line on each page.

Information Text is a left-aligned text field that can include hyperlinks in the format <Link text to display URL>.

Include page numbers allows to set the visibility of the page number at the bottom-right corner of the report.

6.7   Report Map

Figure 6-23 – Report Map

	The *Report Map* tool creates a data stream with one or more maps.	
Group	**Input**	**Output**
Reporting	See below	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Input: Any data stream with at least one spatial object field.

Output: A data stream with a record for every element in fields grouped by, a column for each grouping field, and a field called *Map* with a *Map object* in each record.

This tool creates a report object.

Properties Window:

The *Report Map Configuration* window has two main components. The *Preview* window gives us a look at the map to be generated.
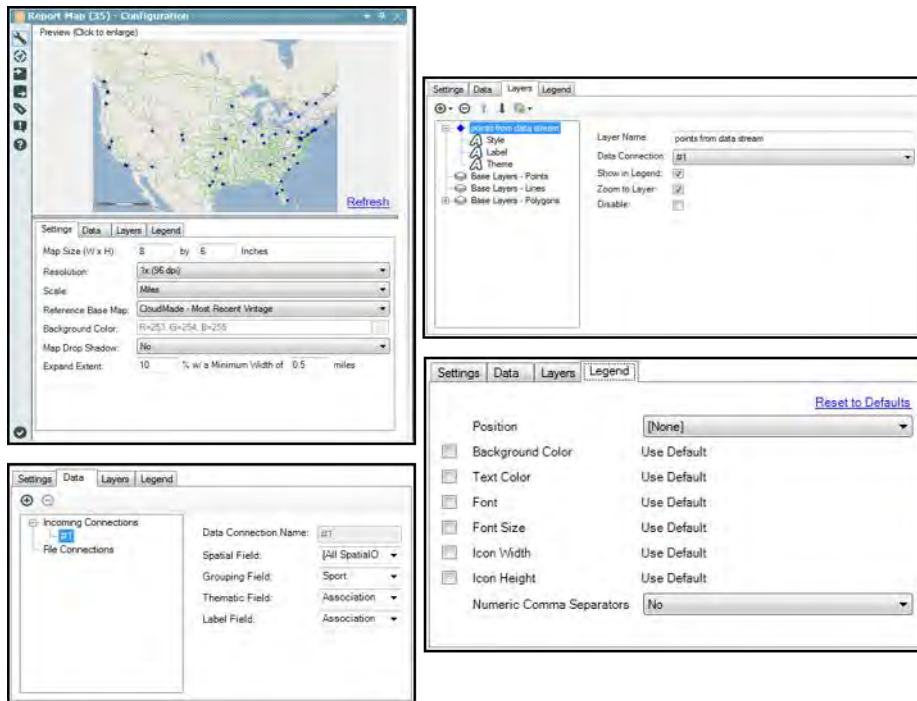
Figure 6-24 - Report Map Configuration
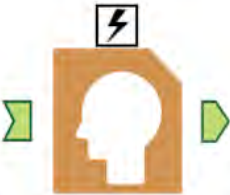
The bottom of the window is broken into four tabs:

- The *Settings* tab allows us to set high-level options including dimensions, resolution, scale units, the background map to be used, background color (if not using a map), whether or not the objects should have a drop shadow, and some zoom settings.

- The *Data* tab allows us to select data streams and add them as data layers. When an incoming connection or file connection is selected, we can define how the layers that use that connection will behave. *Data connection name* is how it will be identified. *Spatial Field* allows us to select one or all of the spatial object fields in the dataset to define.

Grouping Field will create a new map for each unique element in the selected field. *Thematic Field* allows us to color the elements based on a particular field. If a string field is to be used, we need to set the theme to *unique* and type in the values. *Label Field* allows us to select the field we will display the text from on the map.

- The *Layers* tab allows us to modify specific aspects of the map. We can add and remove layers to the map using the plus and minus symbols, as well as change the floating order of the objects. By selecting different objects in the tree, we can modify various settings. Some of these settings will be covered in the following exercise.
- The *Legend* tab allows us to set a number of options related to including a legend with the map. By default, the legend is off and accepts the default values.

6.8   Report Header

Figure 6-25 - Header	The *Report Header* tool allows us to create a header to be used in reports.		
	Group	**Input**	**Output**
	Reporting	Any data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with the additional field *Header*, which has a *Layout* object in each record.

> This tool creates a reporting object and should be used after report elements have been joined together.

Properties Window:

The *Report Header Configuration* window allows us to design a header for the report.
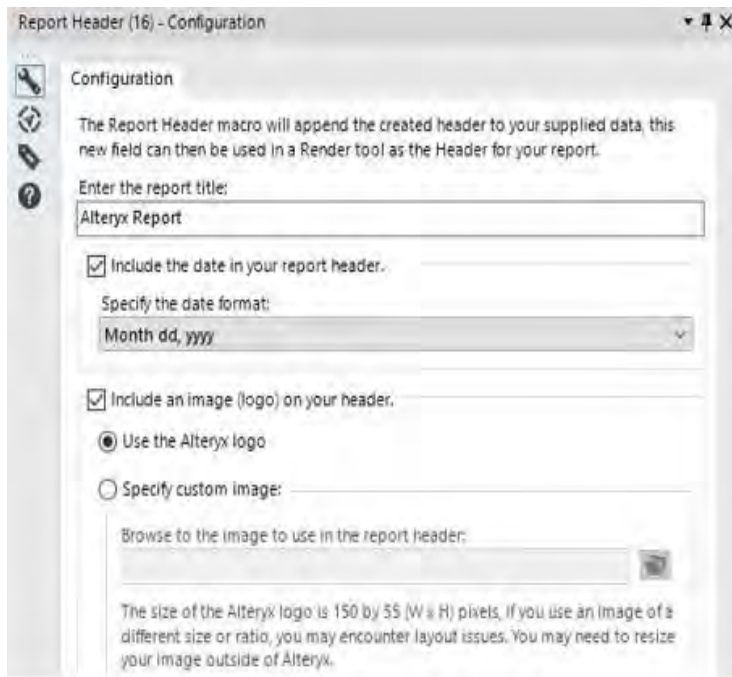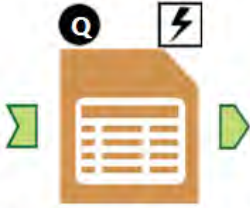
Figure 6.26 – Report header Configuration

We can type in a title, decide if we want a date and set its format, and also decide if we want an image logo in the header.

6.9   Table

\n\n**Figure 6-27 - Table**	The *Table* tool creates a data stream with one or more tables.

Group	Input	Output
Reporting	Any data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: A data stream with a record for every element in the fields grouped by, a column for each grouping field, and a field called *Table* with a *Table object* in each record.

This tool creates a report object, which can be used with other report objects in the table tool.

Properties Window:

The *Table Configuration* window allows us to create a report table.

Figure 6.28 – Table Configuration

Table Mode has the following two options:

- *Basic* will create a table from the rows and columns we have.
- *Pivot (CrossTab)* allows us to use fields we have created using a *CrossTab* tool in order to create a series of more flexible tables.

Group By will create a different table for each of the unique elements in the grouping fields.

Table Configuration allows us to modify the appearance of the table. *Table Width* allows us to set a standard width or let it vary with the available space. *Pivot Style* allows us to display elements of the pivoted data from a *Crosstab* tool in different

ways. *Bar Graph* allows us to create a spark bar for each element in the table. *Show Column Headings* adds a label to the top of each of the columns.

Per Column Configuration allows us to select columns to be displayed on the table and to set the format for each column. *Column Rules* allows us to create a conditional formatting for the column.

Create Row Rule allows us to create a record level formula.

Default Table Settings allows us to change the default formatting of the table.

6.10 Email

	The *Email* tool allows you to use a data stream to send emails for each record.		
	Group	**Input**	**Output**
Figure 6.29 - Email	Reporting	Any data stream	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

This tool creates a report object, which can be used with other report objects in the table tool.

<u>Properties Window:</u>

The *Email Configuration* window allows us to send mail.
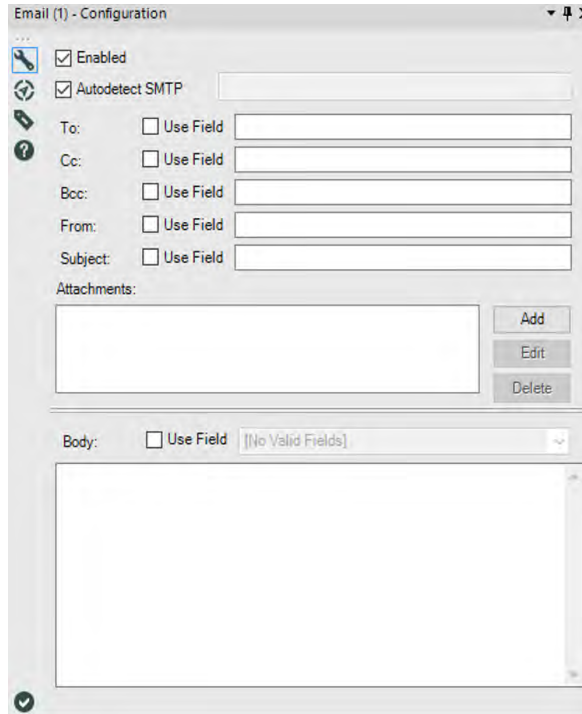
<div align="center">Figure 6.30 – Email Configuration</div>

Email Configuration has the following properties:

Enabled is selected by default. If not selected, the tool will be disabled and no emails will be sent, but the tool retains all other configuration properties.

Autodetect SMTP is used as a control for detecting Simple Mail Transfer Protocol. The default is to Autodetect, however, if the organization receiving the email has restrictive IT infrastructure and firewalls, the email will likely be blocked. In

this case, the SMTP server must be entered in manually. The SMTP setting uses port 25 by default.

Example: mail.alteryx.com

If the SMTP server uses a port other than the default 25 then a colon and the port number should be appended to the server name.

Example: mail.alteryx.com:587

It is recommended to specify SMTP whenever possible for better overall reliability. If using a specific SMTP server that is different from the recipient domain, errors will come back in an email to the sender from the System Administrator outlining the problem.

For the following settings, either type the relevant information in the field or select the Use Field option to select a field from your data stream that contains valid data.

To is the recipient of the email. Must be a valid email address.

Cc is the carbon copy recipients of the email. Must be a valid email address.

Bcc is the blind carbon copy recipients of the email. Must be a valid email address.

From is the sender of the email. Must be a valid email address.

Subject is the subject line of the email.

Attachments contain any additional files to be attached to the email. Email attachments have a 10 MB limit.

- *Add* is clicked to specify each attachment to send with the email. You can either browse to the file location or use a field from your input stream that contains the file location of the attachment.
- *Edit* is clicked to change a previously specified attachment. The same options for Add are available on Edit.
- *Delete* is clicked to remove the attachment. If there are multiple attachments specified, highlight to select the attachment for deletion and hit Delete.

Body is the content of the email.

6.11 Image

	The *Image Tool* allows for a static image to be stored in the workflow or for an image to be retrieved from disk at run time.		
	Group	**Input**	**Output**
Figure 6.31 - Image	Reporting	Image	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: An image is stored that can be retrieved from disk at run time.

This tool creates a report object, which can be used with other report objects in the table tool.

Properties Window:

The *Image Configuration* window allows us to specify image file name to be retrieved.
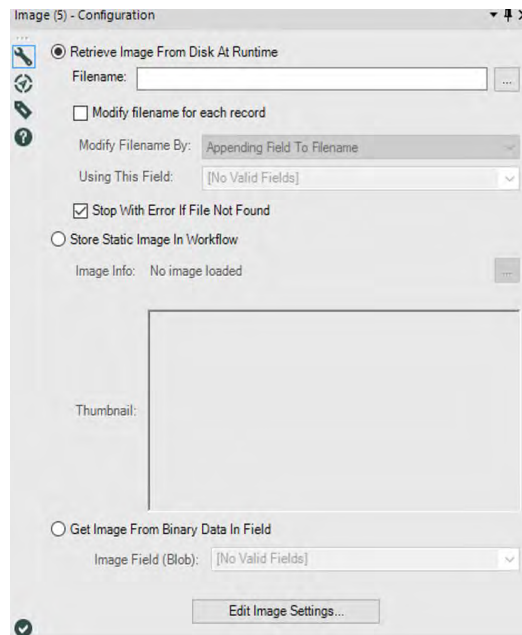
<div align="center">Figure 6.32 – Image Configuration</div>

Image File Editor allows adjustment to the following Image Style properties:

- *Use Fixed Width*: Specify if a fixed width should be used. If choosing to use a fixed width the image is restrained to the width chosen in the menu below.

- *Image Width*: Specify the image width in pixels. Aspect ratio will be preserved.

- *Show Border*: Specify if a border is desired around the image.

- *Border Width*: Specify the border width in pixels.

- *Border Color*: Specify the border color.

6.12 Map Legend Splitter

	The *Map Legend Splitter Tool* takes a legend from the Report Map tool and splits it into its component parts.		
Figure 6.33 – Map Legend Splitter	**Group**	**Input**	**Output**
	Reporting	Report snippet with a map legend	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: A data stream with *Text, Image, Theme Fields* and *Style Fields*. Once split, the legend can be customized by the use of other tools.

Properties Window:

The *Map Legend Splitter Configuration* window allows us to set the *Legend Field*.
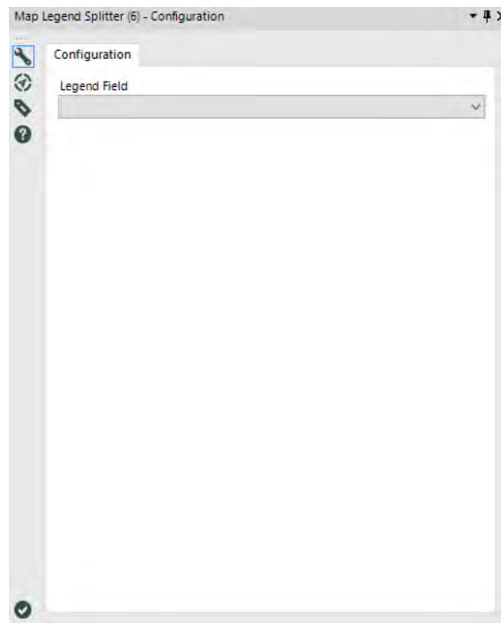
Figure 6.34 – Map Legend Splitter Configuration

This tool is expecting a report snippet that consists of a map legend. This information is contained in the "*Legend*" field. Prior to using this tool, ensure a *Report Map* tool is upstream. On your *Report Map* tool, set your *Legend Position* to "*Separate Field*".

- On the *Map Legend Splitter* tool ensure the *Legend Field* is set to "*Legend*".

6.13 Map Legend Builder

	The *Map Legend Builder Tool* takes the components output from the *Map Legend Splitter* macro and builds them back into a legend table.		
Figure 6.35 – Map Legend Builder	**Group**	**Input**	**Output**
	Reporting	Report snippet with a map legend	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: A data stream with *Text Field, Image Field, Text Style Field* and *Image Style Field*.

Properties Window:

The *Map Legend Builder Configuration* window allows us to specify the output of the field from the *Map Legend Splitter* tool:
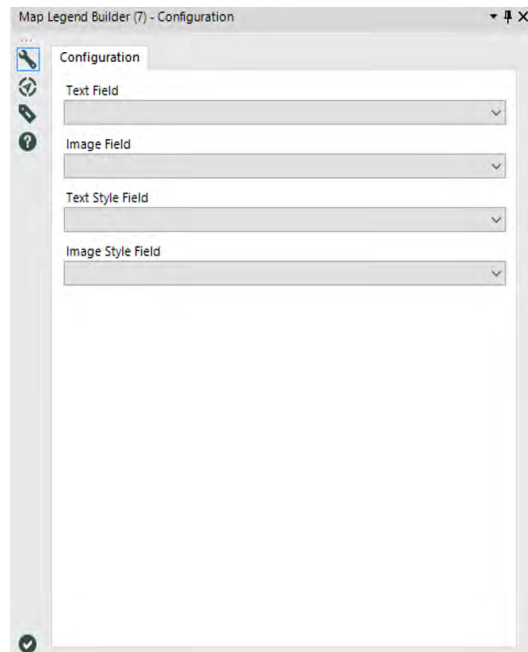
Figure 6.36 – Map Legend Builder Configuration

- *Text Field*: the text you can see in the legend.

- *Image Field*: contains the report snippet with the legend image.

- *Text Style Field*: the table styling elements for the PCXML table, such as alignment of text.

- *Image Style Field*: the table styling elements for the PCXML table, such as alignment of the image.

6.14 Overlay

	The *Overlay Tool* allows reporting snippets (maps, text, images, etc.) to be placed on top of one another.		
 Figure 6.37 - Overlay	**Group**	**Input**	**Output**
	Reporting	Report snippets	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: A data stream with reporting snippets placed on top of one another.

Properties Window:

The *Overlay Configuration* window allows us to specify the fields mentioned below:

Figure 6.38 – Overlay Configuration

Select the *Base Field*. The base field is the main reporting snippet. Overlay fields will be placed on top of the Base field.

Specify the *Overlay Fields*. Use the drop-down lists in the desired positions where you would like to overlay other snippets.

Specify the *Padding* in which the overlay objects will place relative to the Base field. Default is 0.5 inches from the top, bottom, left and right.

Include Source Fields in Output when checked, will output all snippet fields. This box is unchecked, by default, outputting only the resulting Overlay snippet.

6.15 Report Text

	The *Report Text Tool* adds text to reports and documents.		
Figure 6.39 - Report Text	**Group**	**Input**	**Output**
	Reporting	Reports and documents	See below

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the *Top Black Question Anchor* to use those answers in this tool.

Output: Reports and documents with text.

Properties Window:

The *Report Text Configuration* window shows us 2 modes:
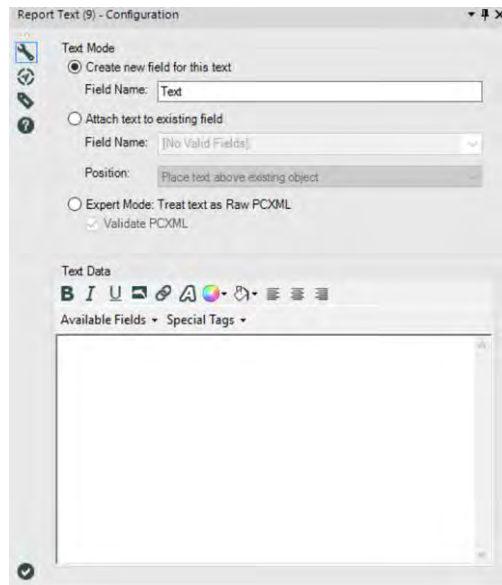
Figure 6.40 – Report Text Configuration

Text Mode:

Create New Field for this Text: Specify a field name. Specifying this field name will help keep track of what is contained in this tool when using the layout tool downstream.

Attach Text to Existing Field: When adding a text tool downstream from another reporting tool, the user is able to attach the text to the Tool so that a layout tool does not need to be used to combine the two *Reporting Snippets*.

Field Name: Select the *Field Name* of the *Snippet* to attach the text to. All available *Snippets* will be available in this pull-down list.

Position: Where to place the text, above or below the Snippet specified in *Field Name*.

Expert Mode: For users who are experts in writing PCXML code. When checked, all text styling is disabled.

Validate PCXML will indicate any syntax errors in the custom PCXML code entered.

Text Data:

Use the box to type the desired text. Additionally, you can include field data from your input by selecting Available Fields. You can also add Special Tags, and Bold, Italicize, and/or Underline text.

WYSIWYG Editor: Use the text editor buttons to style the text as you would like. You can either type your text and then apply styles by highlighting different elements and then selecting one of the buttons described below, or select the button and then type in the selected style.

Text Buttons: Use the buttons to stylize text.

B: Makes text Bold.

I : Makes text *Italicized.*

U: Makes text <u>Underlined.</u>

: Inserts an image. Supports PNG, JPEG, and GIF.

: Inserts a hyperlink. Clicking the Hyperlink button launches the Add Hyperlink dialog box.

Text To Display: This is the text that will appear in the report snippet stylized to reflect it is a hyperlink. You can type the text

in this box or choose from fields in your data stream from the Available Fields dropdown.

Address: This is the location of the content you would like to link to. You can type the address into this box or choose from fields in your data stream from the Available Fields dropdown. Web addresses must include the Hypertext Transfer Protocol (HTTP) such as *http://www.alteryx.com/* NOT www.alteryx.com

: Exposes a font picker that controls font styles.

: Exposes a color picker for the text font.

: Exposes a color picker for the background color of the text.

: Left justified text.

: Centers text.

: Right-justified text.

: Exposes additional options described below:

Available Fields: Data fields contained in the input stream are available on this drop-down menu and they can be added to your text. Data fields are displayed in the drop down with their field type. Some field types have additional display options (such as # of decimal places to display) and they will be indicated with an additional arrow.

Special Tags: Drop down the list of options that will populate the text box to return different objects. Choices include:

Horizontal Bar: A Horizontal bar will appear after the entered text

Current Page: Use this tag when creating a Header or Footer to display the current page number of the report.

Page Break: Use this tag to indicate a new page for subsequent reporting elements. The page break is IGNORED in the following situations:

- When the Text tool feeds into a *Layout* tool with and orientation other than Vertical with Section Breaks. If reporting snippets are being arranged vertically, consider using an *Arrange* tool instead of a *Layout* tool.
- When one Text tool feeds into another Text tool and Attach text to the existing field is checked with a left or a right position selected.
- When the Text tool feeds into a *Table* tool.

Subscript Selection: makes selected text Superscript[Superscript]

Superscript Selection: makes selected text Subscript[Subscript]

Total Pages: Use this tag when creating a Header or Footer to display the total number of pages of the report.

6.16 Major Sports Teams

To...	Alteryx Consultants
Subject	The Sport Report
Attached	📊 Major Sport Teams.xlsx

Hey,

I got a call from a sports bar chain. They want to see a rough report that could show them exactly where the major professional baseball, basketball, football, and soccer teams in the United States and Canada are so they can target new locations.

I want you to help me build a report with the following properties.

- Every sport should start on a new page.
- There should be a generic company header and footer (we can use the default for now).
- It should say the name of the sport.
- There should be a map of all of the stadium locations for that sport.
- There should be a bar graph that has the total count of teams by the governing association.
- There should be a table that has the name of the association, the league, or conference, the division, the team, the city, the home park, the street address, and the zip code, for each team. (The table should be allowed to take in as many pages as necessary.)

Thanks.

When building reports in Alteryx, we need to remember the following key points:

- Most report objects should be created in individual data streams and then brought together. The exceptions are *Layout*, *Header*, and *Footer* objects.
- Report objects should be in the same record for each grouping.

Let us start by connecting to the data (*Chapter 6 – Sport Report*) > M*ajor Sports Teams.xlsx*.
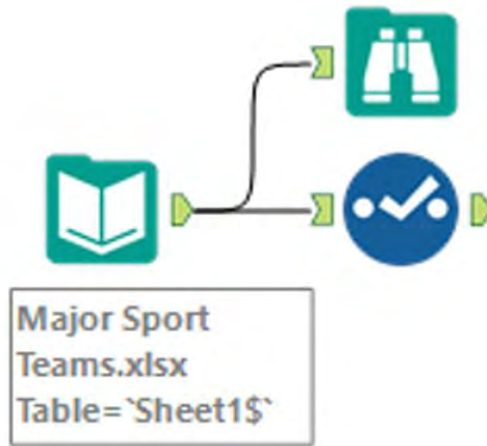
Figure 6-29 – Opening Sport Report

As in any other workflow, we will bring in the data and ensure that it is in the correct format.
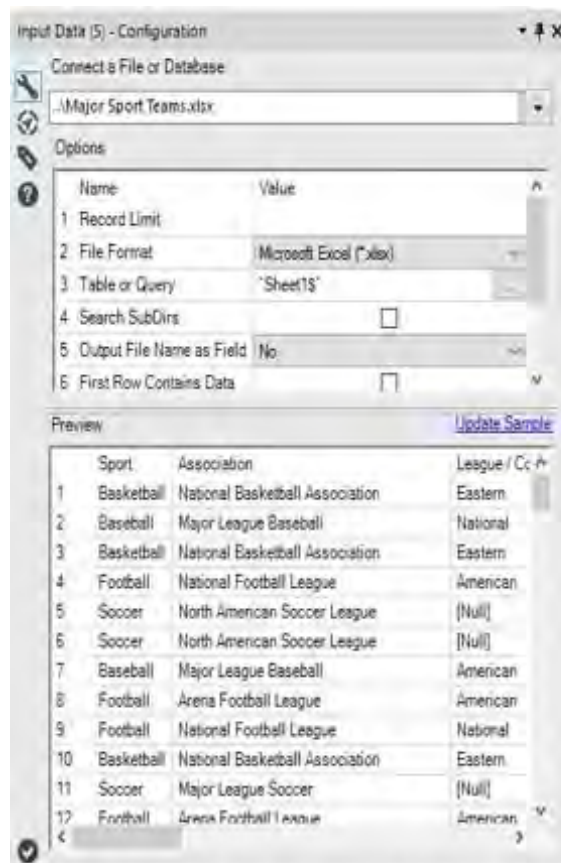
Figure 6.30 – Sports Report – Data formatting

Since we have *Latitude* and *Longitude* fields, we can easily start tackling the mapping aspect of our report. We start by adding a *Create Points* tool to the end of the workflow and using the accompanying configuration.
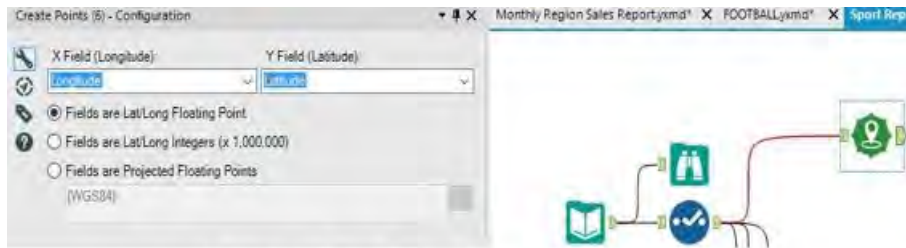
Figure 6.30 – Sports Report – Mapping and creating points

Then, when we add a *Browse* tool and run the workflow, we will see that we have a field called *Centroid* in the last column with a green point object in each record. This is a special spatial field, and we will use this field in order to plot the locations of each team's home field.
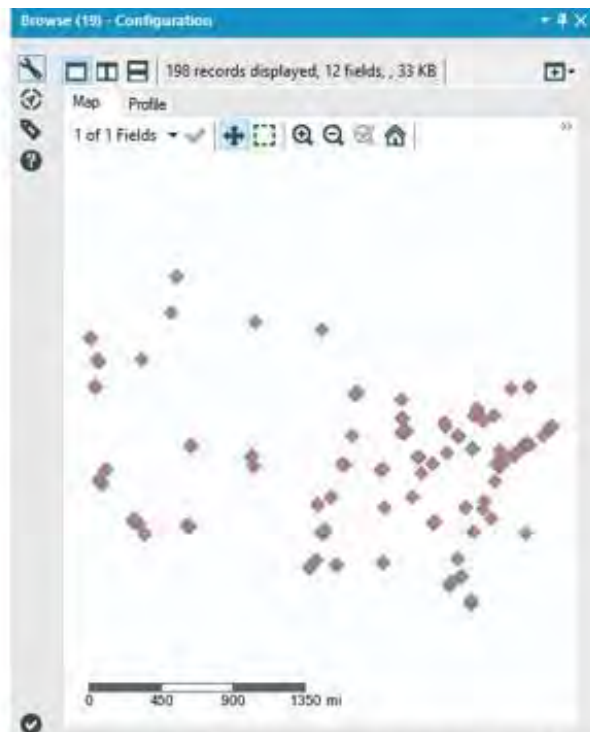
Figure 6.30 – Sports Report – Run Module Using Browse

The resulting data stream will look like the below.

Major Sport
Teams.xlsx
Table=`Sheet1$`
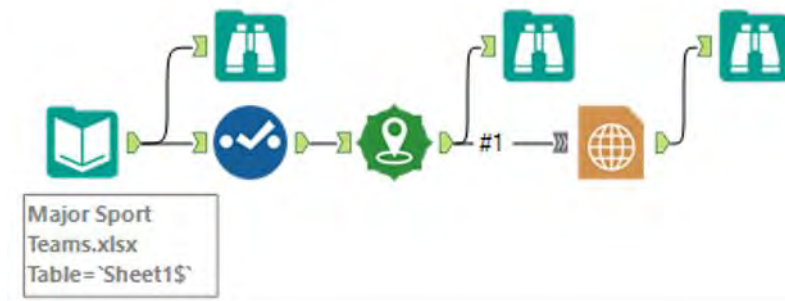
Figure 6-33 – Sports Report Data Stream

Since we are only going to use the one input for this *Report Map*, we can leave the connection labeled *#1*. We need to think about the way the report is going to look, so we should consider the overall layout. It will likely flow better if the map and chart are placed next to each other. This way, we know that image will not take more than half of the page.

In the following pages, the map configuration will allow us to create the desired maps.

Figure 6.30 – Report Map Setting Configuration

The following are the Report Map Data and Layers settings.

Note that we want to group by sport to make one map for each.
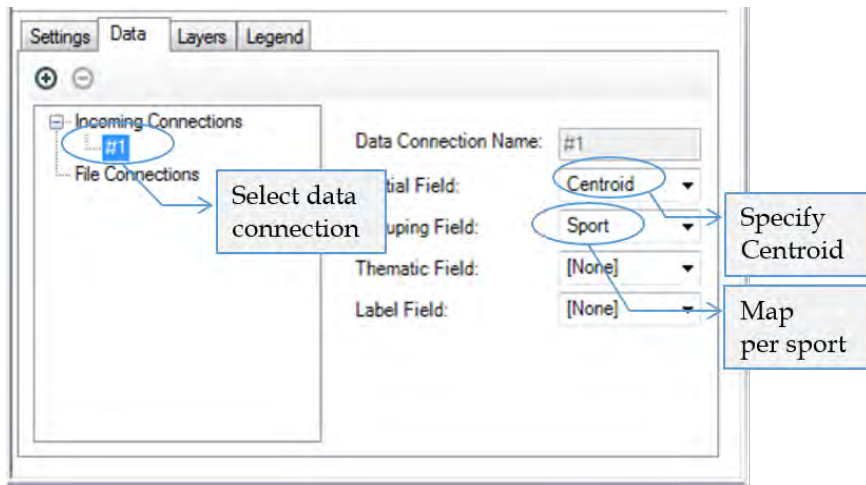
Figure 6-35 - Sport Report - Report Map Data configuration
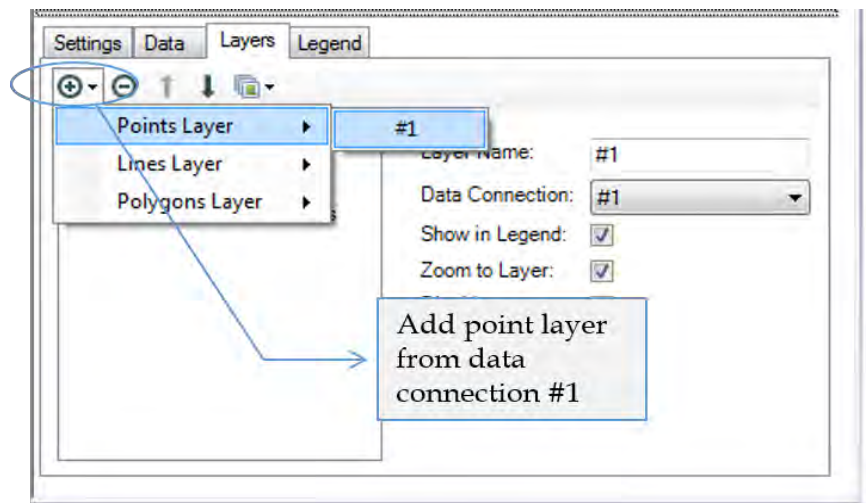
Figure 6-36 - Sport Report - Report Map Layers configuration

Normally, we would not put a *Browse* tool after *Report* tools, but it will help us understand the output of the *Report* tools.

Figure 6.38 – Sports Report – Browse

If we run the module, we should see the output in the *Browse* tool at the end of the data stream.

We see that we have four records, one for each of the four sports we grouped by, identified by the *Group* column at the right. A *Map* field contains the *Report* object map for each sport. The *BoundingRect* spatial field can be used to create new maps that are zoomed to the same area as these. Since we will not be creating any more maps, we can ignore this field.

Also, we should note that we can preview the map and report it if we want by selecting the tabs along the top.

Figure 6.38 – Sports Report – Chart Creation using Map

Now that we have our maps, let us create our charts. We will be making a bar graph for each sport, showing the number of teams in each association. This means that the first thing we should do is summarize the data by sport and association, then take the count of the team names.

Figure 6.39 – Sports Report – Chart Creation using select

Since we want to have clean field names while reporting, we will change the name of the *Count* column to *Number of Teams*.

Figure 6-40 – Sport Report - Adding Charting Tool to end of data stream

Now that we have the data prepared we can add a *Charting* tool to the end of this data stream and apply the settings as shown in the image.

Figure 6.41 – Sports Report – Charting tool

The images below show the grouping, legend, and data layout settings made to the *Charting* tool.
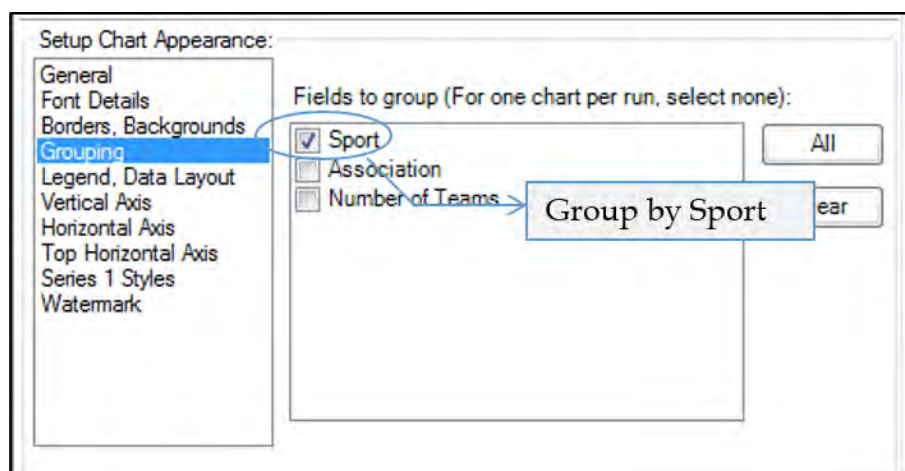
Figure 6-42 - Sport Report – Setup Chart Appearance, Grouping set to Sport
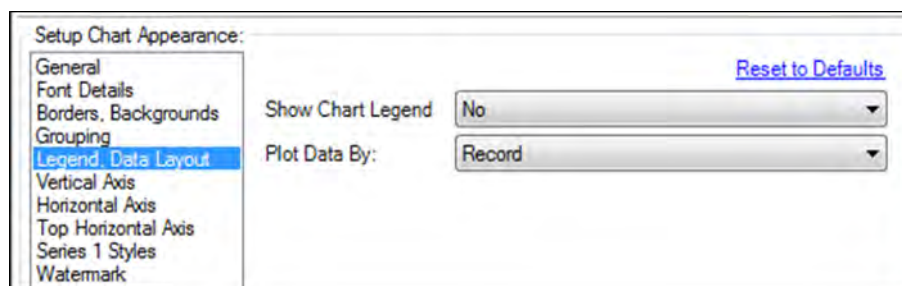
Figure 6-43 - Sport Report –
Setup Chart Appearance – Setting Legend, Data Layout
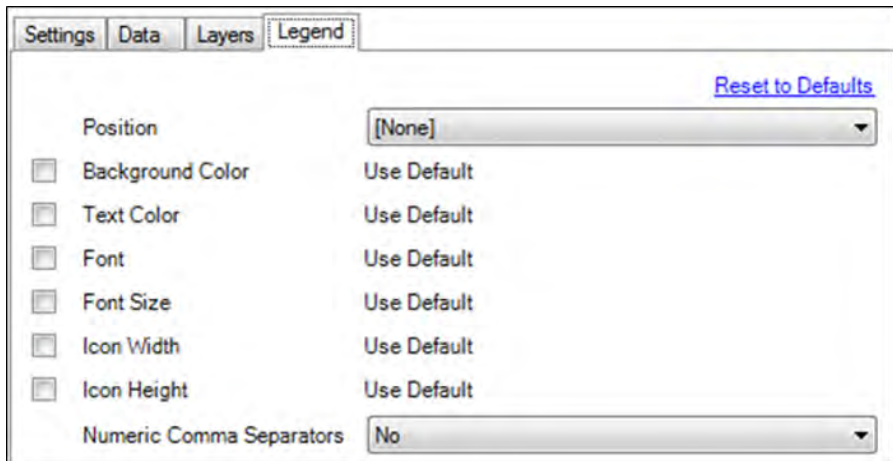
**Figure 6-44 – Sport Report – Setting Legend settings
(refer Report Map tool)**

Let us understand the settings we have changed. The high-level settings are easy to understand. Since we wanted to create a bar chart, we selected *Bar*. Since we wanted to compare the number of teams, we selected that field for our series, and as we wanted to know the governing body related to each bar, we chose *Association*.

Under the *General* settings, we set the width and height to the same dimensions as the maps because we want them to fit side by side. We increased the resolution to have a cleaner image. We provided a descriptive title for the chart with a reduced text size so it fits on one line and we have more room for the chart. We grouped by *Sport* to make one chart for each. We removed the chart legend because we do not need it in this case.

Now that we have covered our maps and charts, we should set a title for each section. We have made a map and chart for each of the sports, but we haven't mentioned anywhere what the sport actually is. We need an object that we can put at the top of the page, with the sport in it.
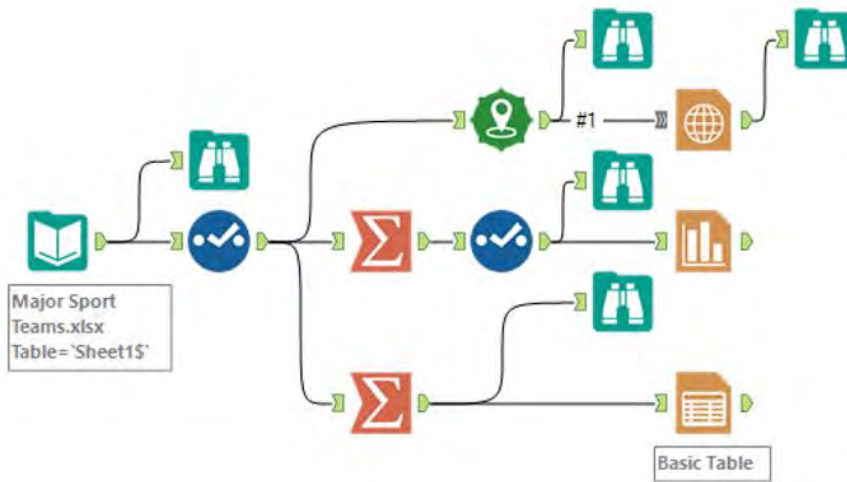
Figure 6-45 – Sports Report – Data Stream Branching

In order to create this object, we will branch off into another data stream, as shown in the previous image. We will use the *Summarize* tool to group by *Sport* so that we have a single record for each. We are then going to add a *Table* tool to create our *Report* objects.
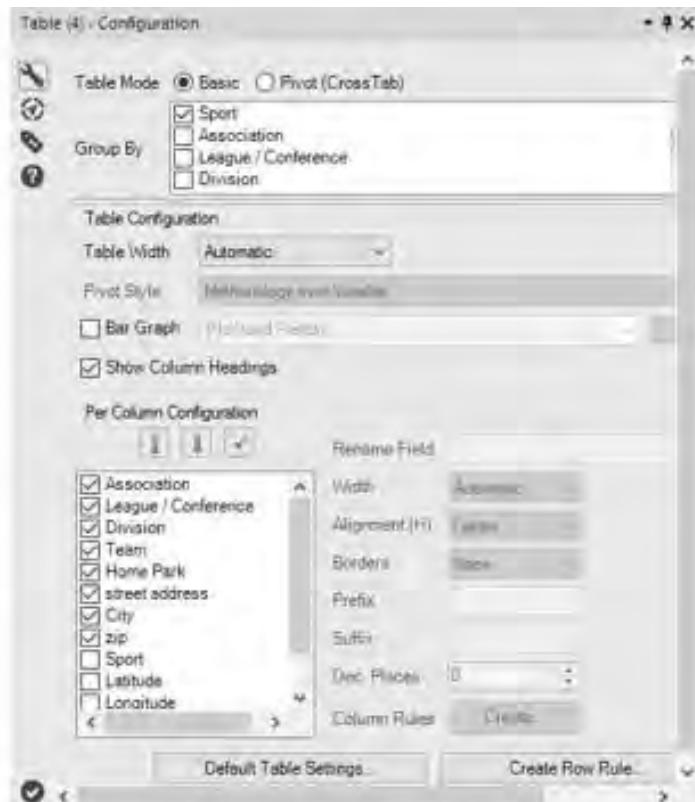
Figure 6.46 – Sports Report – Grouping by sports using Table

Since we want a *Table* object for each sport, we will group by Sport. Since we are using *Sport* as a section title, we will be able to identify the sections we are interested in. Additionally, since we want to ensure that only *Sport* makes it into this table, we can deselect the *Dynamic or Unknown Fields* option.
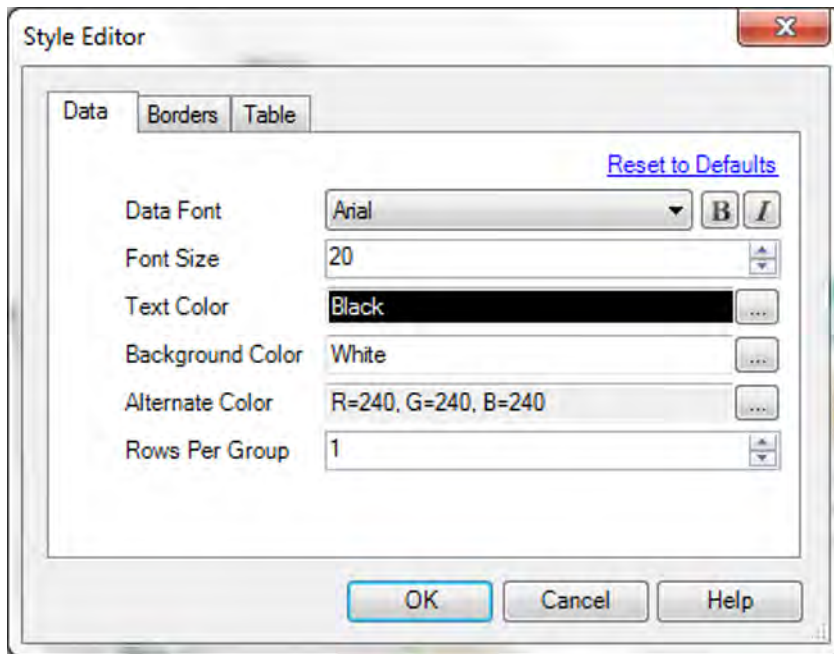
Figure 6-47 - Sport Report – setting title style using Style Editor

Since this will be the title in each section, we should increase its font size.

At this point, the following image is what our data stream will look like.
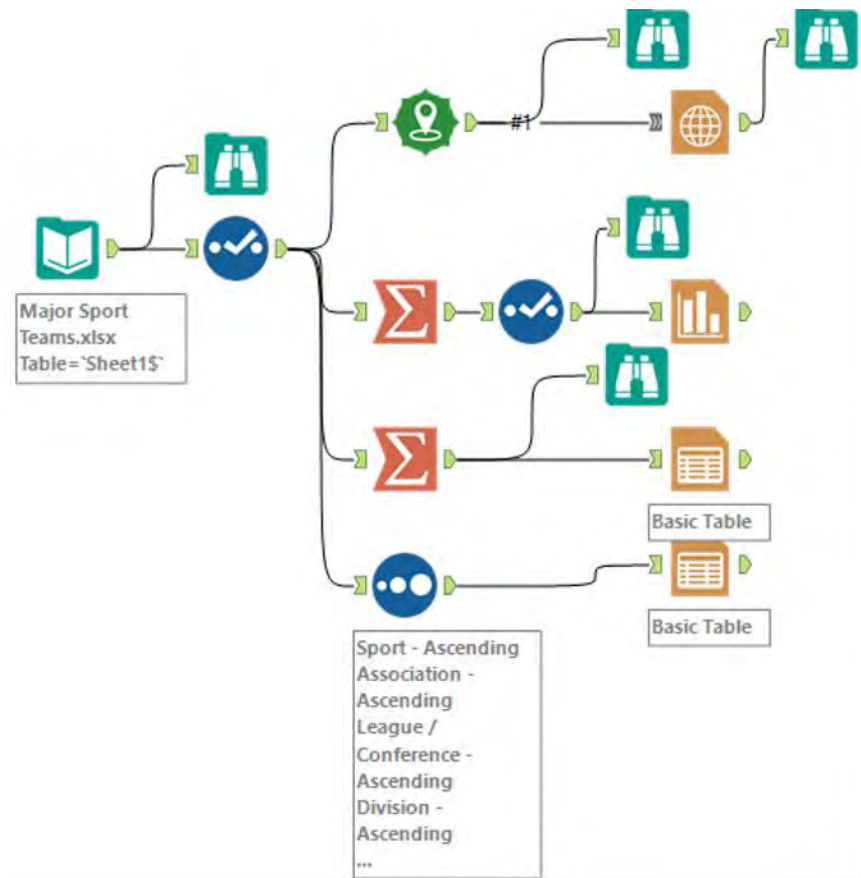
Figure 6-48 – Sport Report - data stream group by each sport

We now need to create a table for the demographic information about the teams. Since this table will be in the report, it should be sorted so the consumer can easily see what's going on.

Figure 6.49 – Sports Report – Applying Sort to Data

Let us sort the data like we see in the *Sort Configuration* window. This will allow the report consumer to easily scan the list for the team they are interested in.

Figure 6.50 – Sports Report – Team Information using Tables

Using these settings, we can create a table with the desired useful information about each team.
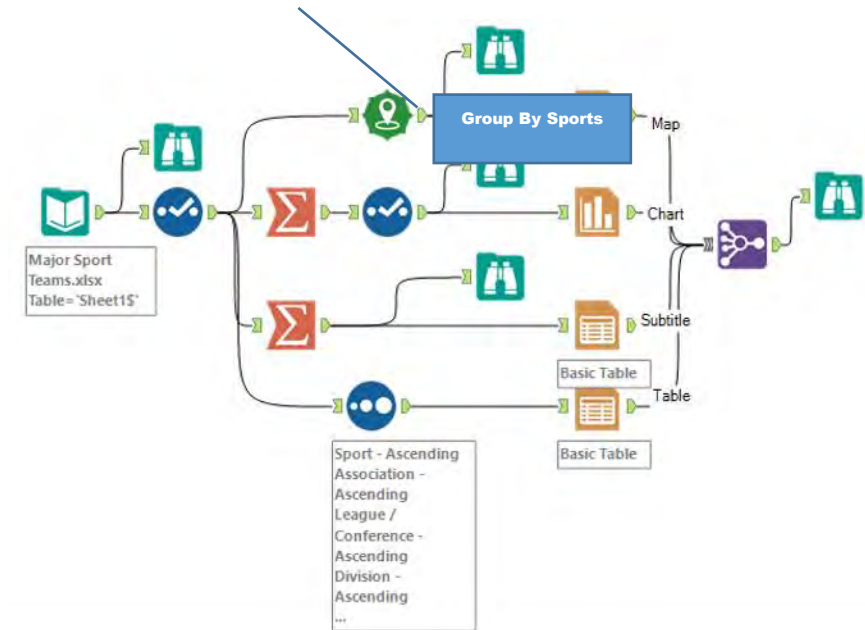
Figure 6-51 – Sport Report – with separate data streams

Now that we have created each of the objects for the body of our report, we need to bring them into a single data stream so we can format the layout, header, and footer.

In the *Browse* tool following the map we created, we saw a single record for each sport because that's the field we grouped by. Adding a *Browse* tool after each of the report objects would also result in the same, except that instead of the field being called *Group*, it would retain the title *Sport*. Since we know that the sport name makes a unique key in all of our data streams, we can use the *Multiple Join* tool to bring them together.

Figure 6.52 – Sports Report – Multiple Joins using Sports name

If we had named the incoming connections the way we see them in the previous image, we would see something similar to the above *Join Multiple Properties* window. Let us use the same setup as above to keep the information that will help us downstream.

Record #	Group	Chart	Table	Subtitle_Table
1	Baseball	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab
2	Basketball	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab
3	Football	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab
4	Soccer	Chart - View Browse Tool Report Tab	Table - View Browse Tool Report Tab	Table - View Browse Tool Report Tab

Figure 6-53 – Sport Report – using Browse to get data after Join

At this point, we should have five fields: The *Sport* field for our understanding and the four report object fields use in the workflow.

Now that we have the objects for the body of our report, we need to incorporate the following into the layout:

- The sport should be at the top of each section.
- We designed the map and chart object to fit side by side, but there is no rule for where they should go.
- We have a table that can vary in length because it has one row for each team. In such a situation, the table is typically put at the bottom of the report body.

- Based on what we see here, we need two *Layout* tools. The first is a horizontal *layout* tool that will align the map and chart objects. The second *Layout* tool is to align the section title, map-chart layout, and team information table.
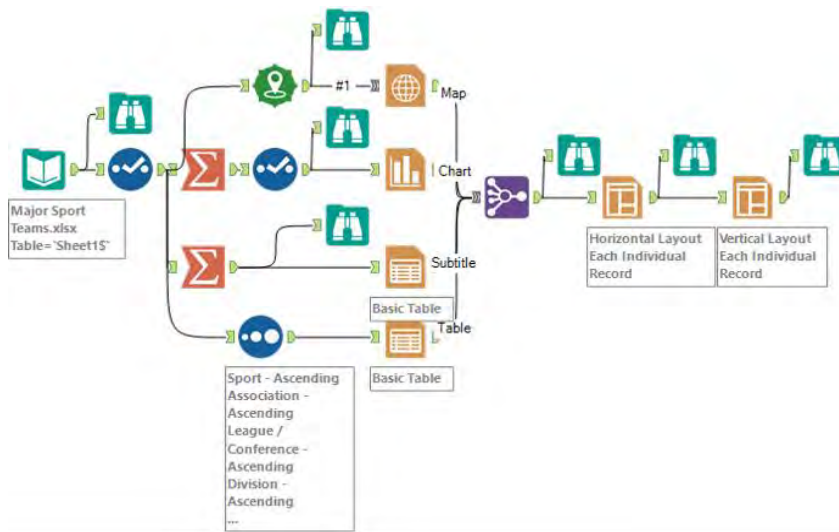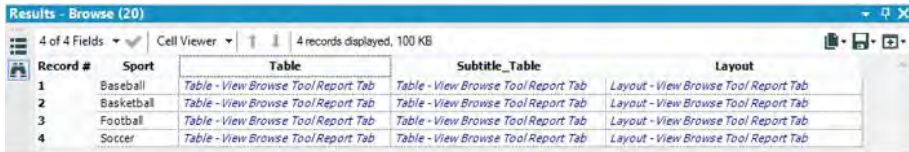
Figure 6-54 – Sport Report – Updated Data Stream

Here are the settings for the first *Layout* tool.

Figure 6.55 – Sports Report – Using Layouts to align maps, team info table etc.

A *Browse* will show us the four columns for the first *Layout* tool.

Figure 6-56 – Sport Report – Data after applying Layouts

The image below shows the settings for the second *Layout* tool.

Figure 6-57 – Sports Report – Applying Layout Configuration

Rerunning the flow now would show the two columns in the second *Layout* tool.

Figure 6-58 – Sport Report – selected data after applying Layout

The data stream after having the Layouts in place would look like the below.
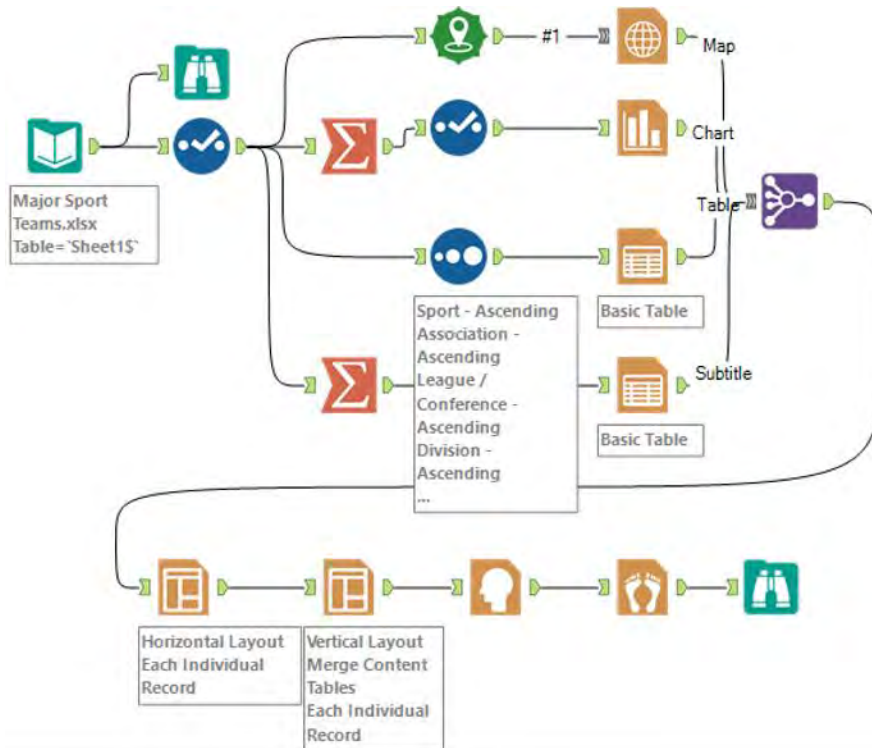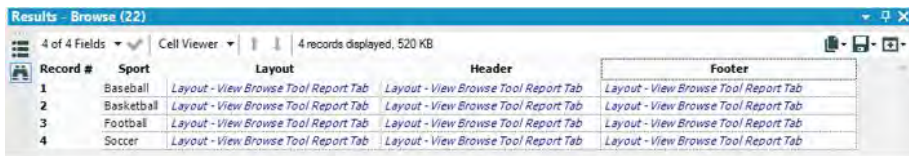
Figure 6-59 – Sport Report – data stream with Layouts in place

Now that we have combined all of the elements of the body of the report into a single report object, we can add a *Report Header* and *Report Footer* to the report. We will use the default settings for both because the client wants to see what would typically be there. We are ready to render the report, but before we do that, let us see what the data we are feeding into the *Render* tool looks like.

Figure 6-60 – Sport Report – Data selected after adding
Report Header and Report Footer

As we can see, there is no special format for the *Header* and *Footer* fields. This means we could put any object into the *Header* and *Footer* options when we render the report. However, the *Report Header* and *Report Footer* tools create appropriately sized objects that are designed for standard report information.
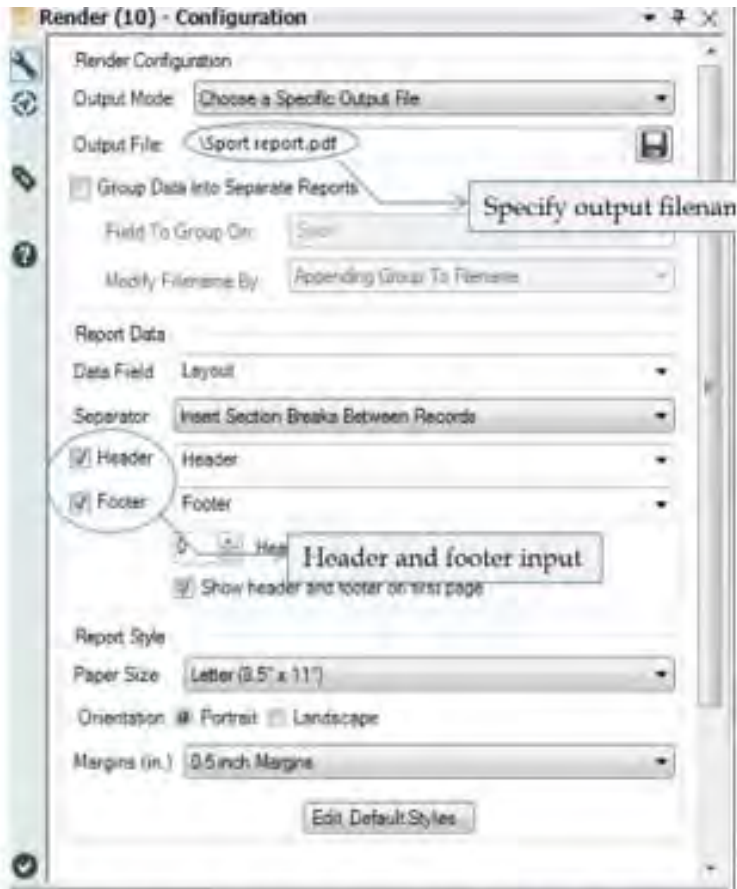
Figure 6-61 – Sports Report – Applying Render to specify path

In the *Render Properties* window, we will specify that we are creating an output file, and specify a path for it. Since we want to start each sport on a new page, we will use the section break option. Finally, we will turn the Header and Footer fields on and define them. The Major Sports Teams data stream should look like the following when it's complete.
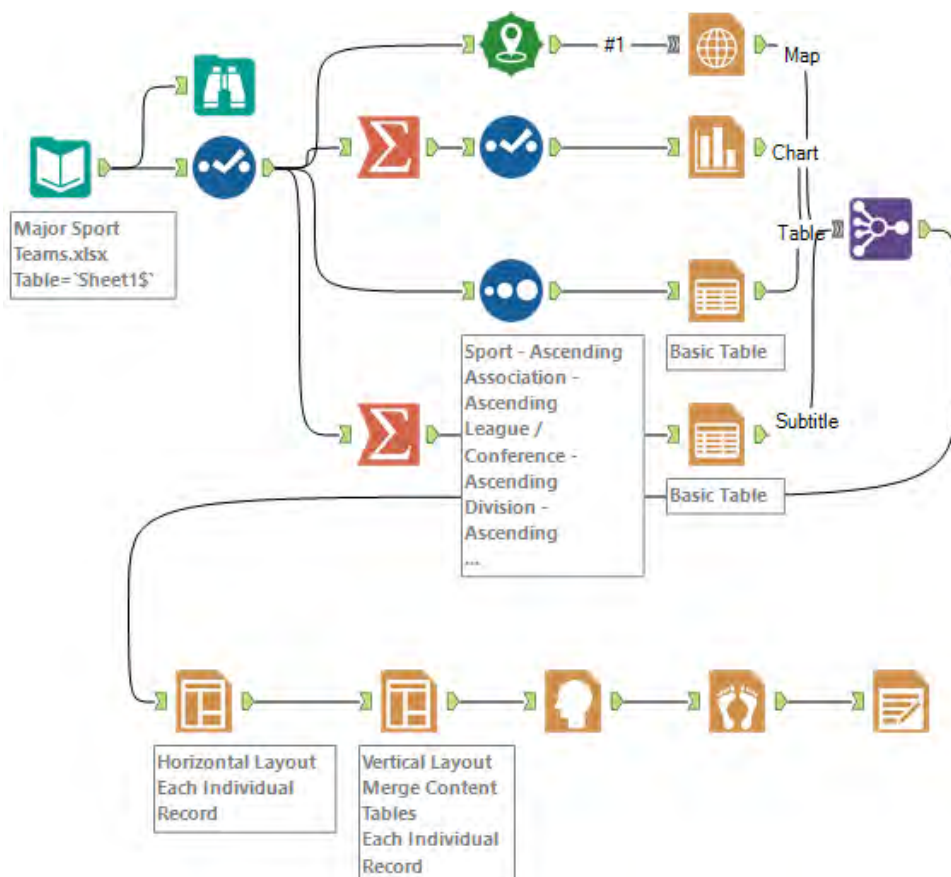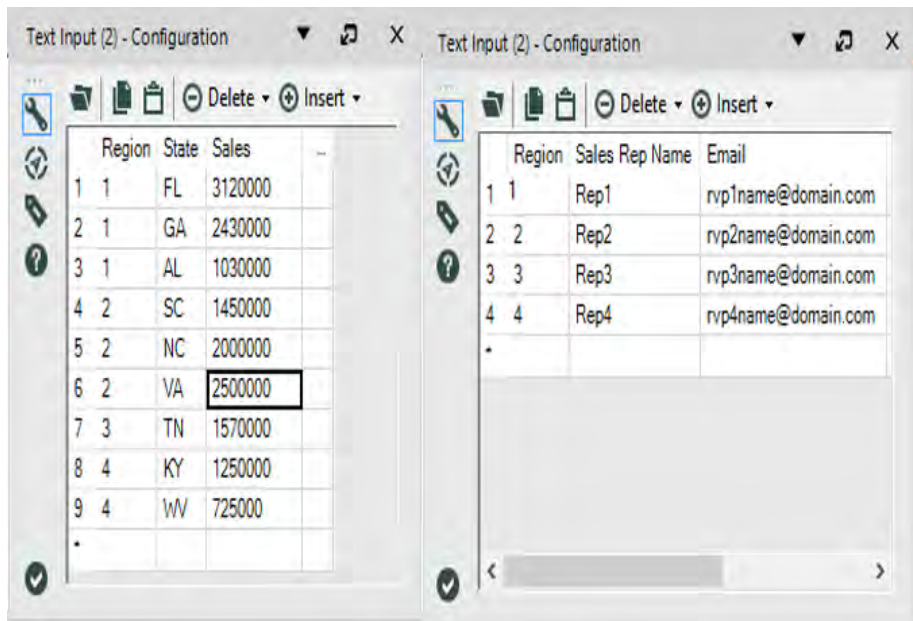
Figure 6-62 – Sport Report data stream on completion

6.17 Regional Sales Monthly Update
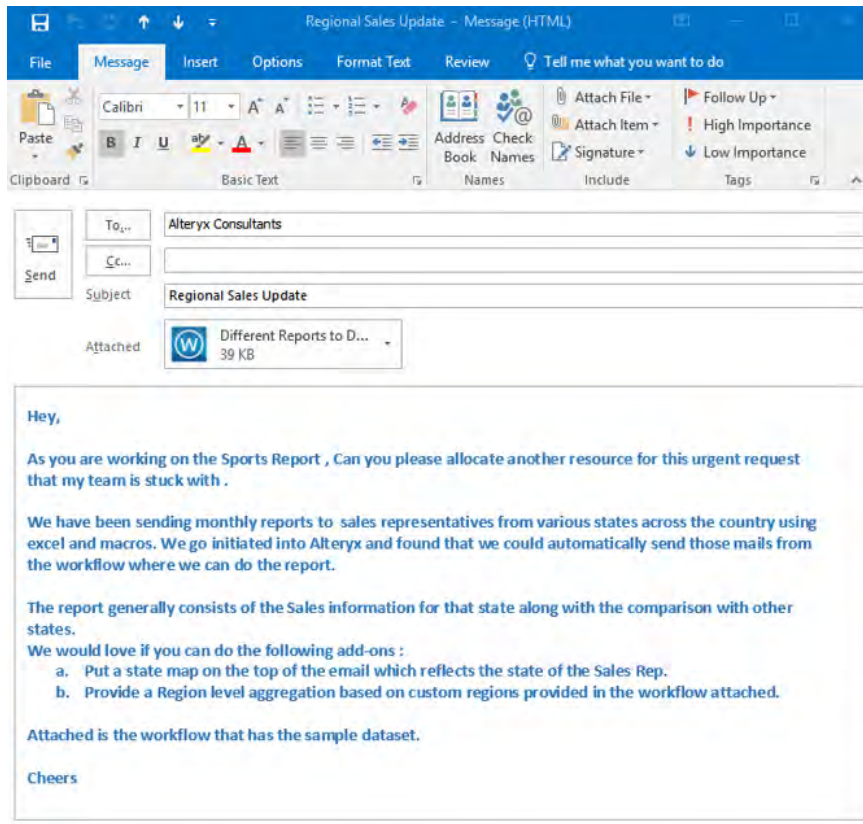
       This sounds like an interesting assignment. Let us start by checking the dataset.

Figure 6-63 – Region Sales Data

The next step that we have is to create a *Table* that formulates the sales numbers and join the email information of the different representatives.
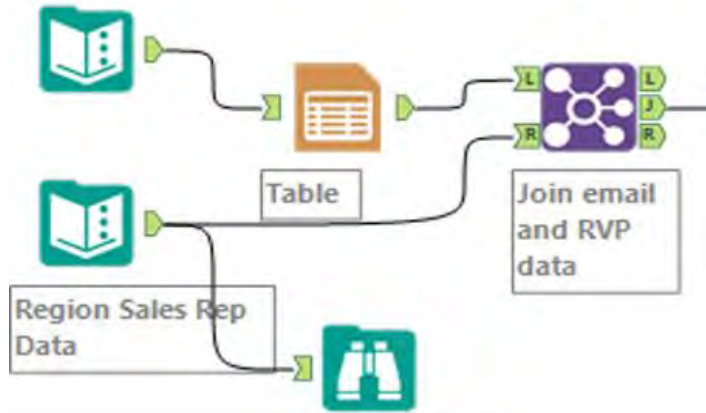
Figure 6-64 – Sales Data joined with Email

The next step is to create the *Title* using *Report Text* of the report, add the *Image* of the states and create the *Report Text* and *Layout* of the E-mail that we need to send out.

Figure 6-65 – Components of the Email

In this step, we added an additional component where we added a salutation based on the time of the day that the *Email* is being sent to add to the aesthetics and added a title that would change as per Region.
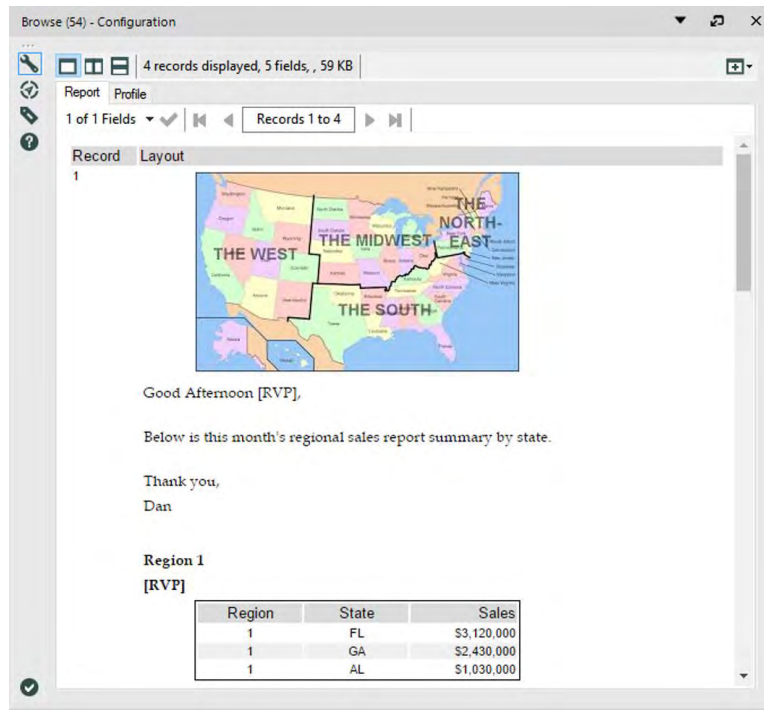
Figure 6-66 – Email Preview

The preview of the email shows us the complete map of the United States and the state wise split of the region's sales numbers for the current month.

Now all that is required to provide the necessary email id information and configure the email tool and the output would be sent to multiple Sales representatives across the country.

Figure 6-67 – Regional Sales Monthly Report data stream on completion

6.18 Football!

To...	Alteryx Consultants
Subject	FW: FOOTBALL!

Hey,

The sports bar just called me back. They said that the first draft of the report shows promise, and are likely to use us, as long as we can quickly modify the report to show the following : Limit to a report about Football. Instead of breaking it down by sport, break it down by the association. Additionally, instead of summarizing the chart by association, summarize it by division.

I am in meetings for the rest of the afternoon and I cannot show them the turn around, but, I think you know enough to make these changes.

If we can get it back to them in the next 20 minutes, I think it would guarantee that they go with us.

Thanks.

CHAPTER 7

Expensive Beauty Products

To...	Alteryx Consultants
Subject	Expensive Beauty Products
Attached	Body Care Co.csv

Hey,

See the email below from Jan at Body Care Co.

Body Care Co. is a company that I work with pretty regularly. They have Alteryx and tend to ask us to work on projects when they don't have the bandwidth.

I think working together on this would be a good way to introduce you to a few new topics.

Thanks.
_____

I've got something fairly straightforward for you this time.

I am using data that looks like the attached data set in order to run a lot of analyses. I'm swamped, so if you could build me an Alteryx flow that I can use going forward, that would be great!

I am using a dashboarding tool that needs to be highly responsive. Unfortunately, it is running too slow because I don't have an easy way to identify the selected data.

Below is the logic to create a flag for a number of different things I need to look at. Would you be able to create a process so the data can be modified to have these flags?

1) Lotions ("Lotion" is in the description)
2) Moisturizing Products ("Moisturizing" is in the description)
3) Baby Products ("Baby" is in the description)
4) Acne Products ("Anti-Acne" or "Anti-Oil" is in the description)
5) Create groups for all products that are named 90% identically by word (we are looking into similarly named products to see how their sales correlate).

Thanks in advance,
Jan

7.1   Tools & Concepts

Tools	Concepts
Fuzzy Match	Fuzzy Logic
Record ID	Regular Expressions
RegEx	Pre-Calculation
Unique	Data Flags
Make Group	Removing Duplicates
	Indexing
	Group Tools
	Organize Module
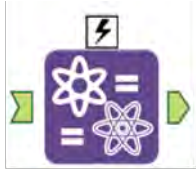
7.2   Fuzzy Match

	The *Fuzzy Match* tool allows us to apply fuzzy logic to a data stream to match similar items.
Figure 7-1 – Fuzzy Match	**Group** **Input** **Output** Join See below See below

Note: The *Fuzzy Match* tool has multiple different methods of determining what matches. If the reader is not familiar with fuzzy logic, then it is suggested to research these techniques in order to understand how each works.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: Any data stream with a text field

Output: A data stream has the key field and selected *Fuzzy Match* information fields.

Properties Window:

The *Fuzzy Match Configuration* window has the following five core components.
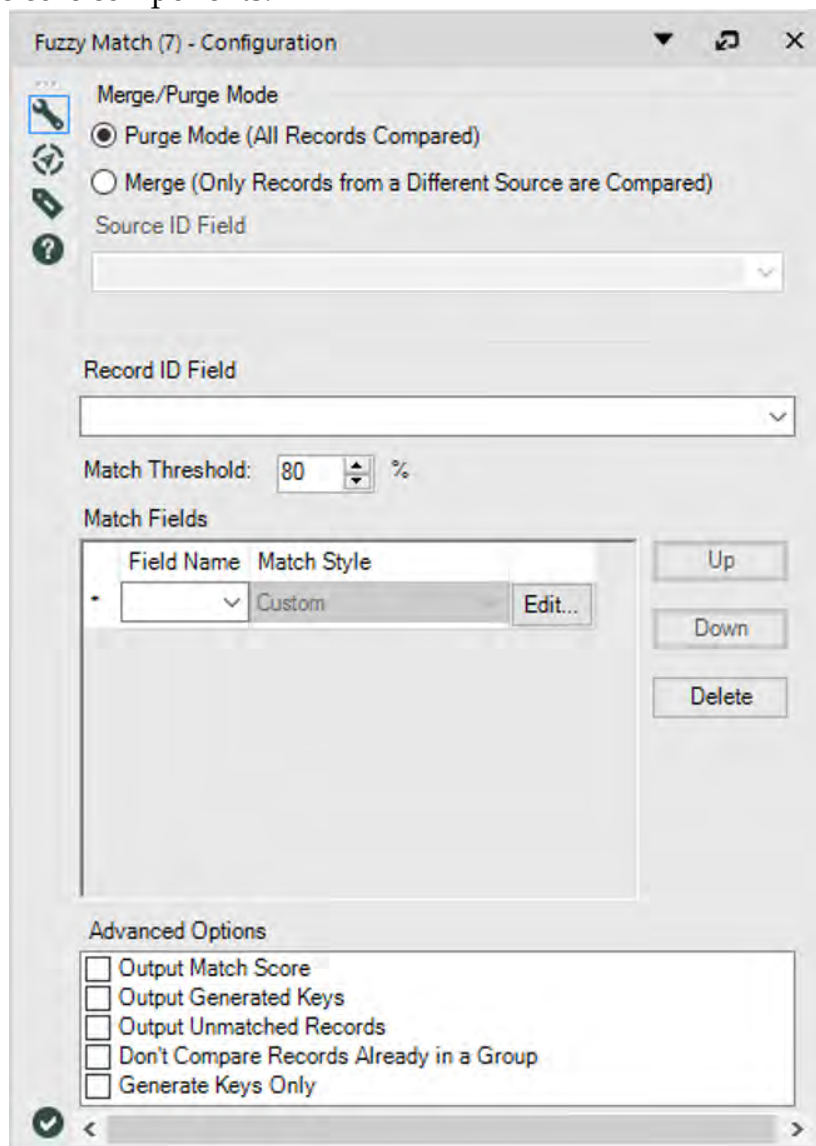
Figure 7-2 – Fuzzy Match Configuration

- *Merge/Purge Mode* allows us to decide if we need to compare all of the records in the list, or to compare records that have different sources. This means if we have data that is broken into logical buckets, we can compare across those buckets without comparing them.
- *Record ID Field* is the field we will match on.
- *Match Threshold* is the minimum match percentage to have a positive result.
- *Match Fields* allows us to select fields and methodologies to run the analysis for matches.
- *Advanced Options* allows us to modify the output of the *Fuzzy Match*.
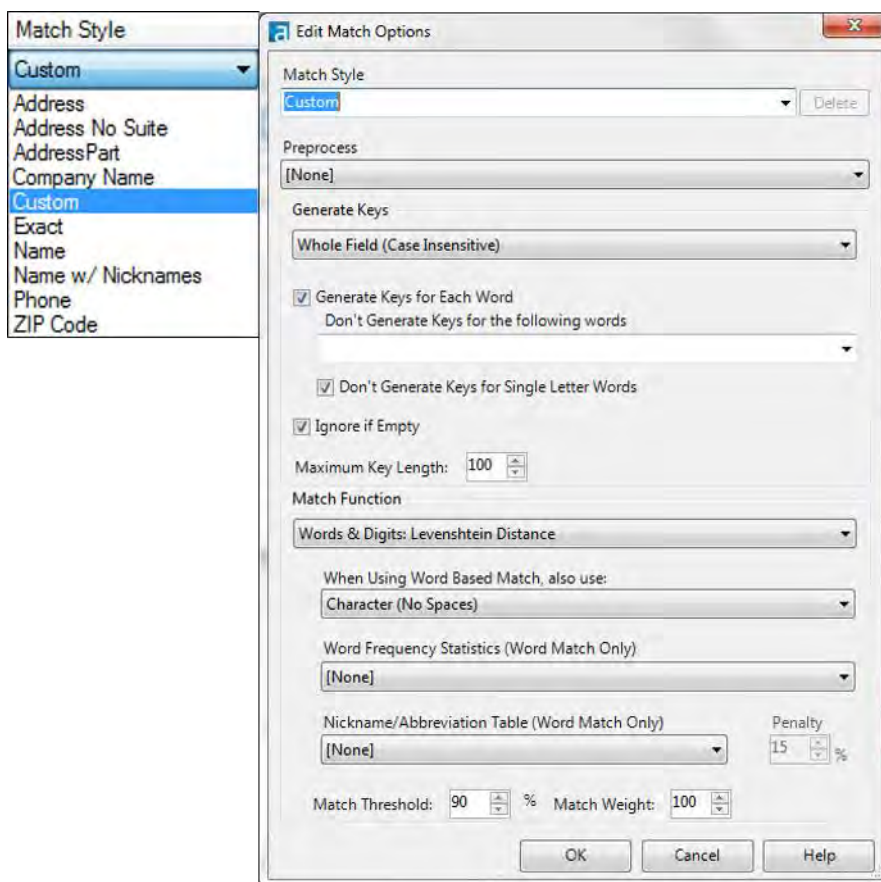
Figure 7-3 – Fuzzy Match Configuration
Edit Match options

Under *Match style,* there are some predefined matching methods. These match styles are Alteryx's suggested fuzzy logic methodologies for the type of things they name.

Custom allows us to define a methodology, which can be saved as a permanent setting if we are an administrator or super user.

The *Edit* option under the *Match Fields* section opens the *Match Options* window and offers more customization. There are many options here that allow us to define a matching algorithm.

Many of the options are self-explanatory. However, if we are not familiar with fuzzy logic, we should research Levenshtein Distance and Jaro Distance before designing our own match methodology.

7.3   Record ID
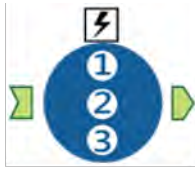
Figure 7-4 – Record ID

	The *Record ID* tool creates an index column in our data stream.	
Group	**Input**	**Output**
Preparation	Any data stream	See below

Note: This is useful for creating an index

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: The original data stream with a new field, which numerically identifies each record.

Properties Window:

The *Record ID Configuration* window has four basic components:
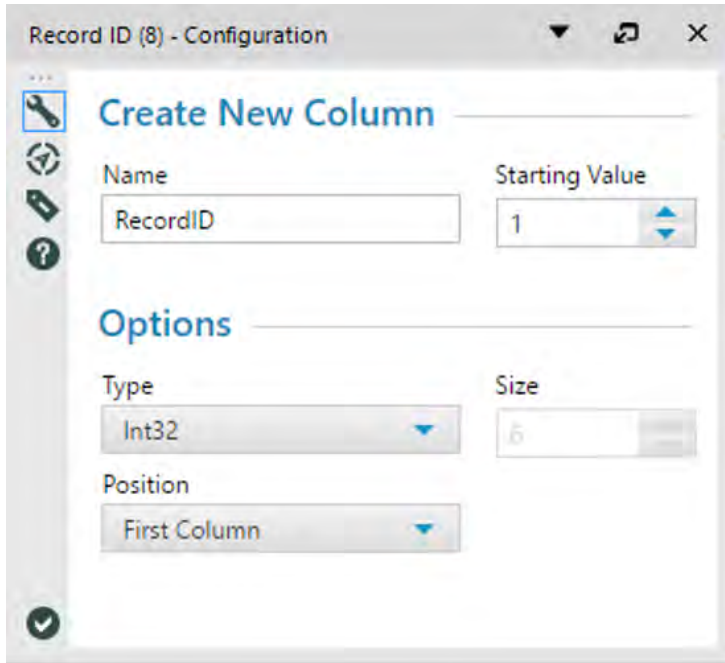
Figure 7-5 – Record ID Configuration

- *Starting Value* is the number we want to assign to the first record (which will be incremented following that record).
- *Field Name* is the resulting field name for our index.
- *Field Type* allows us to select the type and size of the resulting field.
- *Field Position* allows us to set the new field to the beginning or end of the dataset.

7.4   RegEx

Figure 7-6 – RegEx	The *RegEx* tool allows us to run regular expressions on a field in our data stream.

	Group	Input	Output
	Parse	See below	See below

Note: Alteryx uses the Perl regular expression syntax. See Appendix I.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Input: A data stream with a string field.

Output: The original data stream and additional columns or rows depending on the selections made in the tool.

<u>Properties Window</u>:

The *RegEx Configuration* window has five basic components:
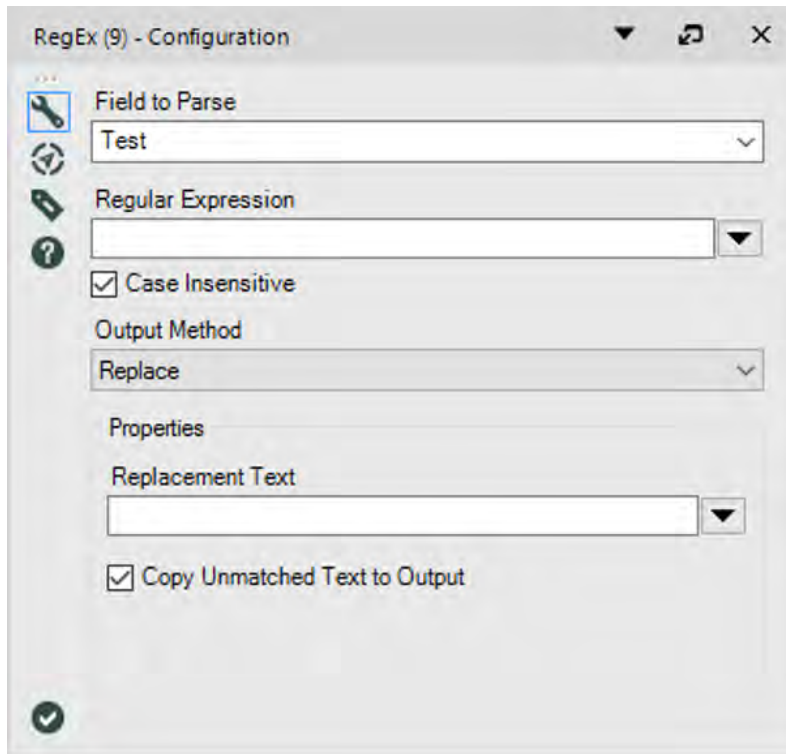
Figure 7-7 – RegEx Configuration Replace Method

- *Field to Parse* is the text field we want to manipulate.
- *Regular Expression* is the expression we use in order to identify sections of the string.
- *Case Insensitive* allows us to specify if the case is important to our search.
- *Output Method* allows us to select what the regular expression will result in.
- *Properties* define specific options depending on which *Output Method* is selected.

Replace allows us to replace the text every time the *Regular Expression* is found in the *Replacement Text* string.

Copy Unmatched Text to Output will maintain any text before, between, or after the occurrences of the *Regular Expression*. If this option is not checked, the output will only have the *Replacement Text* as many times as the mentioned *Regular Expression* is found. In the above case, every time a space (\s) is in the field, it is replaced by the string characters "- -" (two hyphens).
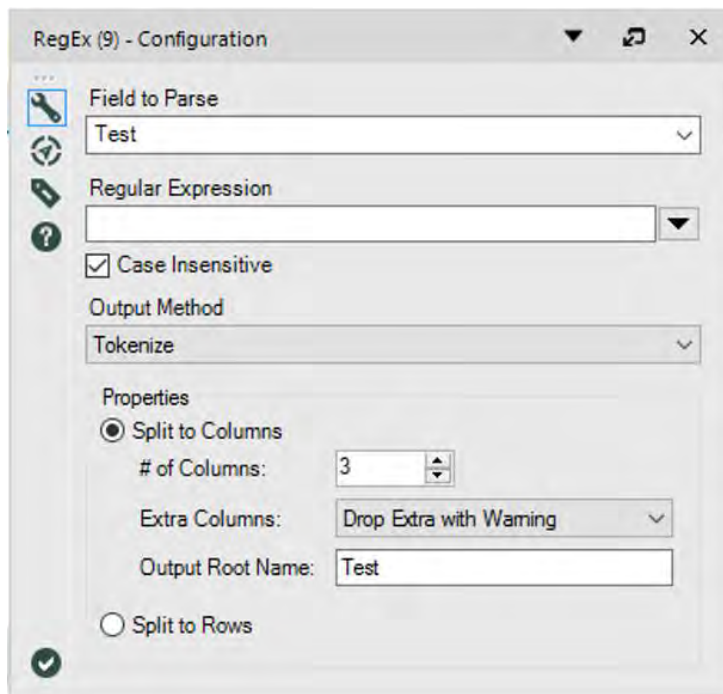
Figure 7-8 – RegEx Configuration Tokenize Method

Tokenize allows us to split the field we have, into an additional column or row every time we find the *Regular Expression* in the field.

In this case, we are creating a new record for every word in a field. Since the regular expression will look for the longest occurrence of the format we need to specify that we do not want

any spaces in the unknown characters between the beginning and ending of a word.

We can see that \<[^\s]*\> looks for *words* and creates a new record with all other data replicated for each of the resulting words. This is because:

\< identifies the beginning of a word
[^\s] means any character that is not a space
* means repeat the previous thing 0 or more times
\> identifies the end of a word
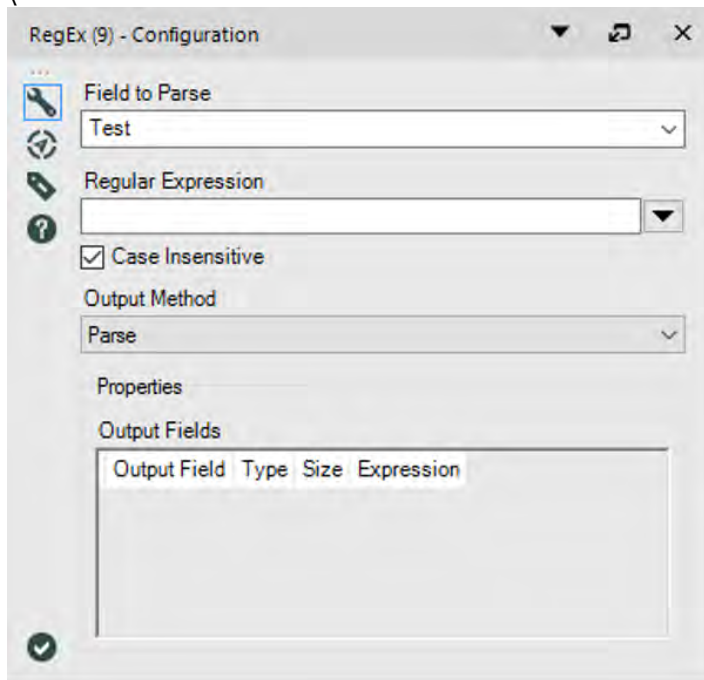
Figure 7-9 – RegEx Configuration Parse Method

Parse allows us to define *Marked Groups* in order to break down fields with specific syntax into new columns, which can be named in the *Output* fields section.

In this example image, the tool isolates the numbers from a list of phone numbers formatted as *(XXX) XXX – XXXX.*

Marked groups are identified by parentheses. The marked group (X) would select X and create a column.

The expression is made of six marked groups:

(\() Since "\(" means the "(" character, the first group is an opening parenthesis.

(…) Since "." means any single character the second group is the three characters following the previous group (which would be the first three digits of the phone number).

(\)\s) Since "\)" means the ")" character, the third group is the closing parenthesis and space before the second set of numbers.

(…) As seen earlier, it represents the next three characters. So we have the second three digits of the phone number in the fourth group.

(\s-\s) Since "-" is literally the hyphen character, the fifth group is the "space hyphen space" between the second and third set of digits.

(….) gets us the last four digits of the phone number.
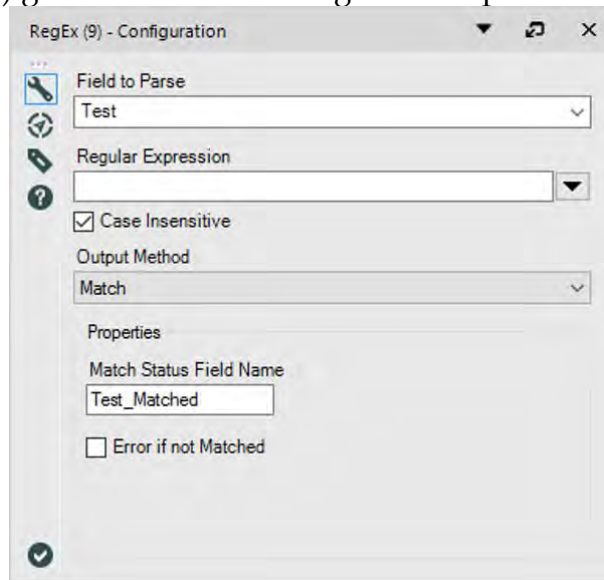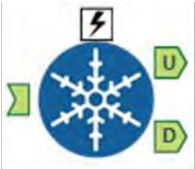
Figure 7-10 – RegEx Configuration Match Method

Match creates a Boolean field such that anytime the *Regular Expression* is in the field, the result is true else it is false.

In this case, we have the regular expression, .*world.*. This means that the word *world* with any characters before or after it.

The new field created is called *Test_Matched*, which can be used downstream.

7.5   Unique

	The *Unique* tool allows us to separate unique or duplicate records based on the field(s) we select.		
Figure 7-11 – Unique	**Group**	**Input**	**Output**
	Preparation	Any data stream	See below

Note: Unique records are kept if the record is the first to pass through the data stream. Unique differs from grouping by the same fields in a *Summarize* tool. It keeps all of the data associated with the record instead of just the used fields.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output U: a data stream where each record is unique based on the fields that were selected.

Output D: a data stream where each record is a duplicate of the one in the U (unique) output based on the fields that were selected.

Properties Window:

The *Unique Configuration* window is a list of each of the fields in the dataset.
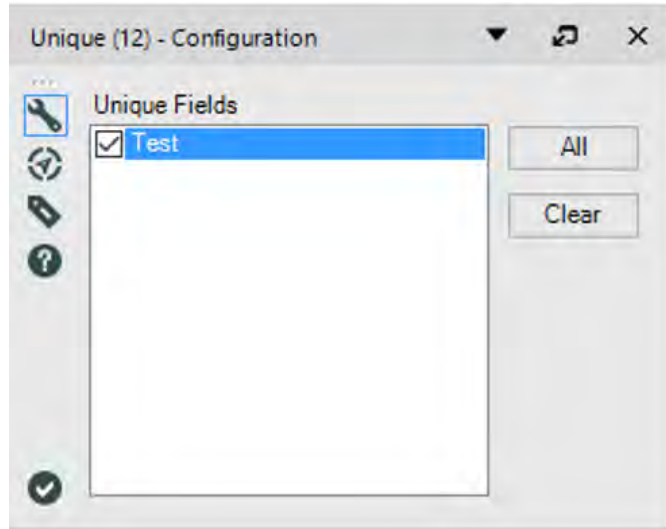
Figure 7-12 – Unique Configuration

We need to select all those we want to use to identify unique records.

7.6   Make Group

	The *Make Group* tool will Assemble pairs of matches into groups based on their relationships. Generally used with Fuzzy Match tool.		
Figure 7-13 - Make Group	**Group**	**Input**	**Output**
	Join	See below	See below

Note: The Make Group tool is designed to group values in two columns together (if A~B and B~C then A, B, and C are in the same group).

This makes it well suited for use in conjunction with the Fuzzy Match tool, which outputs two columns of associated values.

The *Output* will have Group column based on input provided.

Properties Window:

The *Make Group Configuration* window has the following two core components
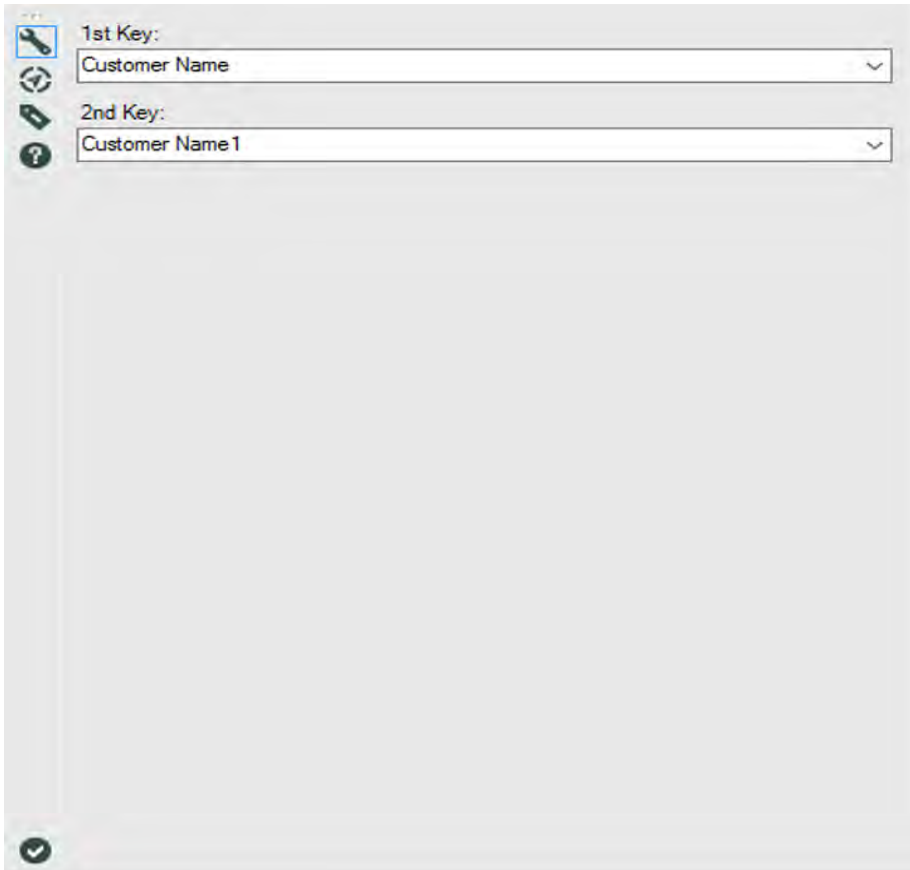
Figure 7-14 – Make Group Configuration

1st Key allows as to select column 1 AND *2nd Key* allows to select Column 2

The two keys are interchangeable, and the order is irrelevant -- the important thing is the relationships that connect two things together. Using those relationships, this tool is able to put everything together and tell you who is in the same group! (The group will be named after one of the members of the group almost arbitrarily, taking the first value from a sorted list of all the members in the group.)

7.7 Expensive Beauty Products

To...	Alteryx Consultants
Subject	Expensive Beauty Products
Attached	Body Care Co.csv

Hey,

See the email below from Jan at Body Care Co.

Body Care Co. is a company that I work with pretty regularly. They have Alteryx and tend to ask us to work on projects when they don't have the bandwidth.

I think working together on this would be a good way to introduce you to a few new topics.

Thanks.
_____

I've got something fairly straightforward for you this time.

I am using data that looks like the attached data set in order to run a lot of analyses. I'm swamped, so if you could build me an Alteryx flow that I can use going forward, that would be great!

I am using a dashboarding tool that needs to be highly responsive. Unfortunately, it is running too slow because I don't have an easy way to identify the selected data.

Below is the logic to create a flag for a number of different things I need to look at. Would you be able to create a process so the data can be modified to have these flags?

1) Lotions ("Lotion" is in the description)
2) Moisturizing Products ("Moisturizing" is in the description)
3) Baby Products ("Baby" is in the description)
4) Acne Products ("Anti-Acne" or "Anti-Oil" is in the description)
5) Create groups for all products that are named 90% identically by word (we are looking into similarly named products to see how their sales correlate).

Thanks in advance,
Jan

The problem that Jan is describing is important to be aware of when we create data for dashboard and reporting tools. These tools are expected to create visuals in real time based on dynamic user inputs. Ideally, the performance is fast enough that the dashboard consumers do not observe a lag in the data load.

In order to achieve this goal, dashboard creators need well-designed and usually tidy data. This means that as much should be built into the data preparation process as possible. In Jan's case, they are doing some complex string calculations in order to filter the data appropriately.

Best practice is that any calculation the dashboard creator needs to run at a record level should be done in the data preparation stage.

To improve performance, pre-calculation steps are performed and it is a fundamental part of good data preparation.

Even when we are not preparing the data for analytics tools, it is important to pre-calculate fields the end user may need. This is because building and vetting a calculation once in a workflow means there will be a significant reduction in risk caused by human error downstream.

Let us imagine that we work for a large retailer that wants to know the profit ratios for each transaction. We may have 20 different analysts working on this project and reproducing the results every month. If these analysts are each rewriting the same formula *(Profit/Sales)* every month, it means we need to make sure they do not accidently type *(Sales/Profit)* in 240 different formulas per year. Instead, if we know they need transaction level profit ratio, we can create a standard formula in the data

preparation process that writes this value into the source data that they all use.

Besides the pre-calculation, we can see from this email that Jan is only interested in a module. This is the case because they work with Alteryx and will do some modifications, like replacing the data connection and outputs. We will often come across things like this when we are supporting other Alteryx users because they either do not know how or do not have time to build a portion of a module.

Considering the information in the email, it seems Jan was talking about flagging item names that fall into each of those buckets. We will use two techniques to achieve this. For requirements one through four, we will write regular expressions that create flags for matches and mismatches. And for the fifth, we will use fuzzy logic to define grouping for each of the tools.

Regular expressions are pattern-matching formulae that allow us to define a pattern to look for in a string. Alteryx uses the Perl programming language syntax for its regular expressions, so any documentation related to Perl regular expressions will help us get up to speed. A cheat sheet is provided at the end. (See Appendix I.)

Fuzzy logic is essentially a logical process that allows computers to say, "Eh, close enough!" As far as Alteryx is concerned, this is limited to string parsing; however, more generally this extends to programming a definition of *tall*, *heavy*, and *blue*. There are various methodologies of matching strings. Alteryx uses the *Levenshtein Distance* and *Jaro Distance* to achieve this. The nuances of the differences are well documented and it

is suggested that the reader learns more about them if they plan to use Alteryx's *Fuzzy Match* tool.

Figure 7-15 – Expensive Beauty Products
Body Care Company data

If we connect to the file in *Chapter 7 -- Expensive Beauty Products > Body Care Co.csv*, we see that Alteryx is reading all of the fields in this file as strings.

Figure 7-16 – Expensive Beauty Products
Select Configuration

Ideally, we should change the last three fields to numeric. But since we are only going to use *Item Name* and we don't know

what the rest of Jan's process looks like, we should leave the fields the way they are.

Since we need to make sure Jan knows what each piece of this module is doing, we have to make sure that it is well documented. The first thing we will do is put our initial connection into a tool container that indicates that the entire set of tools should be replaced when the connection is changed.
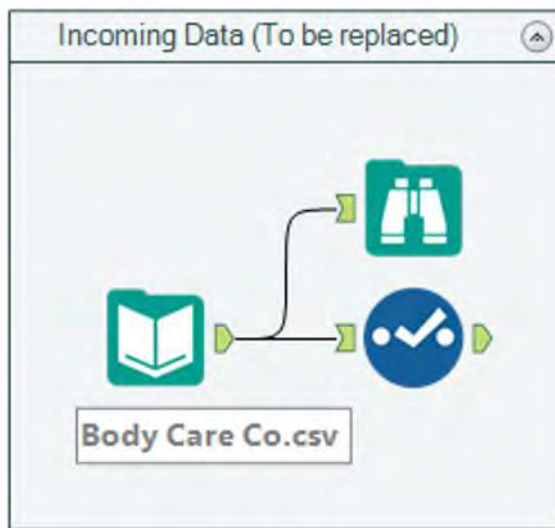
Figure 7-17 – Expensive Beauty Products
Input data – to be replaced in a container

Let us take a look at the contents of this file.

**Figure 7-18 – Expensive Beauty Products
Input data – Browse configuration**

Next, let us create flags using regular expressions. We will create a new *Tool Container* called Regular Expressions with a *RegEx* tool in it.
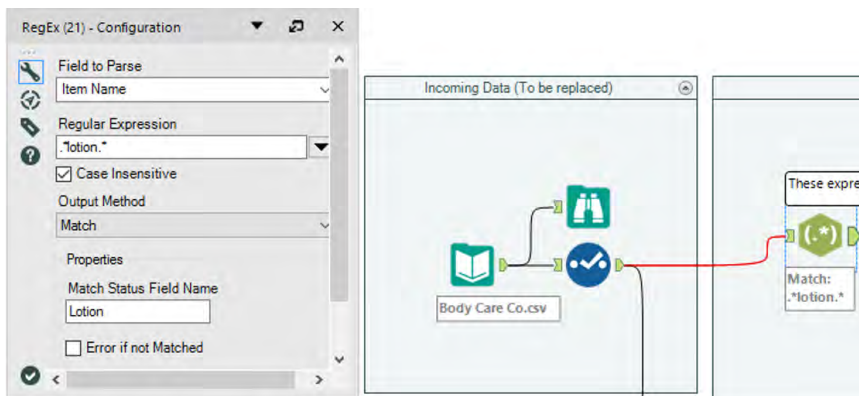
Figure 7-19 – Expensive Beauty Products Regex- "lotion"
flag –Match method

The first requirement we want is a flag to identify if *lotion* is in the *Item Name* field. We can use the expression. *lotion. * to identify any field that has the string "lotion" in it.

This regular expression checks to see if the literal pattern "0 or more characters followed by 'lotion' followed by 0 or more characters" is in each record and returning true or false.
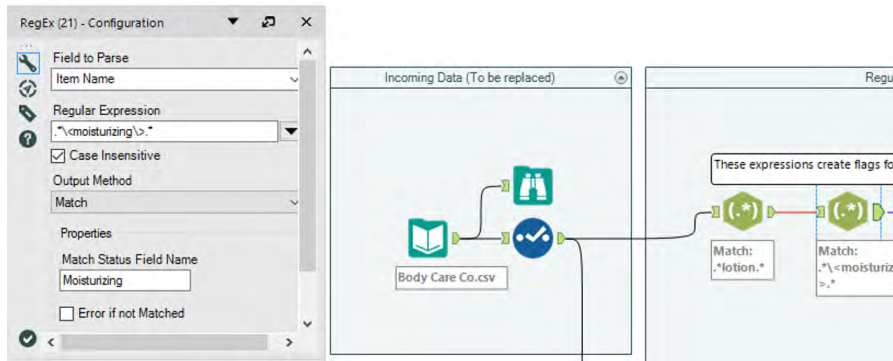
Figure 7-20 – Expensive Beauty Products Regex-
"moisturizing" flag –Match method

A familiarity with Body Care Co's data lets us know their item names always have spaces between words. This means we can actually make our expression more specific as shown in the image, with *.*\\<moisturizing\\>.**

This regular expression reads "0 or more characters followed by the word 'moisturizing' followed by 0 or more characters."

The special characters \\< and \\> in regular expressions take into account specific characters preceding or ending a word to identify if the entire word is matched.

Following the same pattern, we can create a field called *Baby Product*.

Figure 7-21 – Expensive Beauty Products Regex- "baby" flag –Match method

We then make another for acne products.
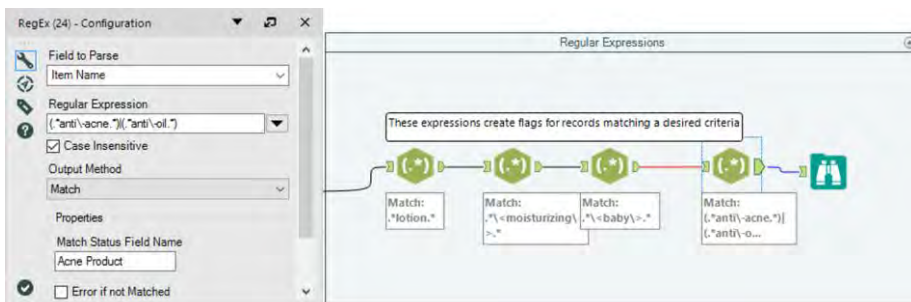
Figure 7-22 – Expensive Beauty Products Regex- a complex flag –Match method

Our last flag is more complicated because we have two different things that could define acne products.

We will combine what we know about Boolean statements and regular expressions to create the expression

(.*anti\-acne.*) | (.*anti\-oil.*)

This regular expression is checking to see if the pattern "0 or more characters followed by 'anti-acne' followed by 0 or more

characters, or 0 or more characters followed by 'anti-oil' followed by 0 or more characters' exists.

As we can see, regular expressions can get complex, but there are always multiple ways to create them. In this case, we could have also used *.*anti\-(acne|oil).**, where it would have looked for the pattern "0 or more characters followed by 'anti-' followed by 'acne' or 'oil' followed by 0 or more characters".

Figure 7-23 – Expensive Beauty Products
Boolean filed (flags) for RegEx Match

We can see that we have the four Boolean fields to identify each of the four sets of products we have defined.
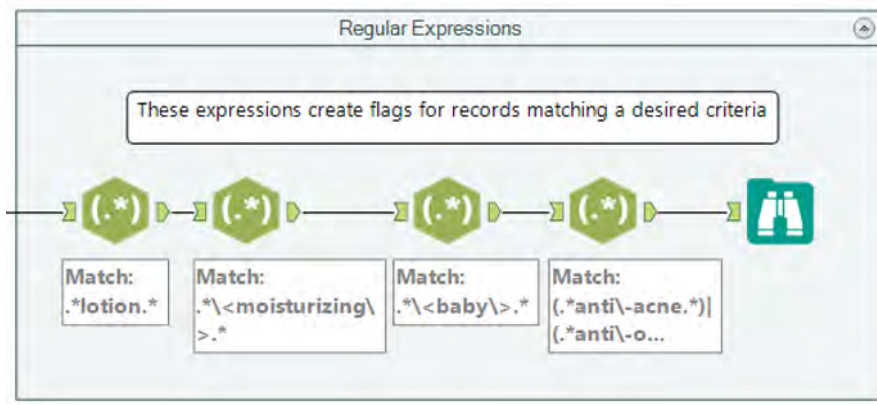
Figure 7-24 – Expensive Beauty Products
Adding comments to indicate contents in Tool Container

We can now add a *Comment* tool that describes what is happening in the *Tool Container* and moves onto the fuzzy logic.
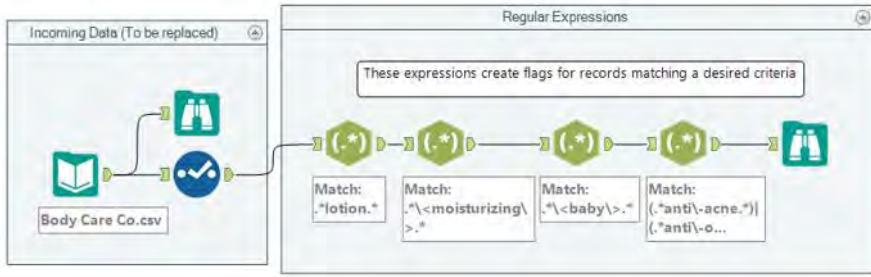
Figure 7-25 – Expensive Beauty Products
Updated data stream

We need to define the groups of products with similar names. The first thing we should do is to create a unique list of product names. Let us introduce a *Summarize* tool within a container and connect it to data input stream.
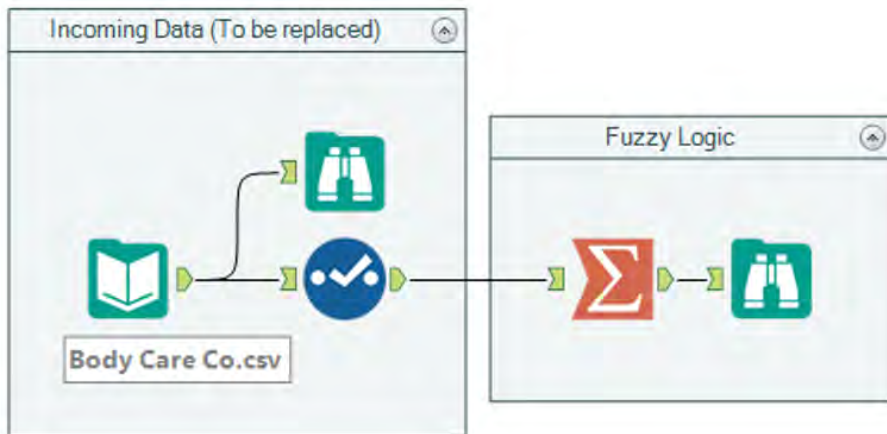
Figure 7-26 – Expensive Beauty Products
second data stream for fuzzy match
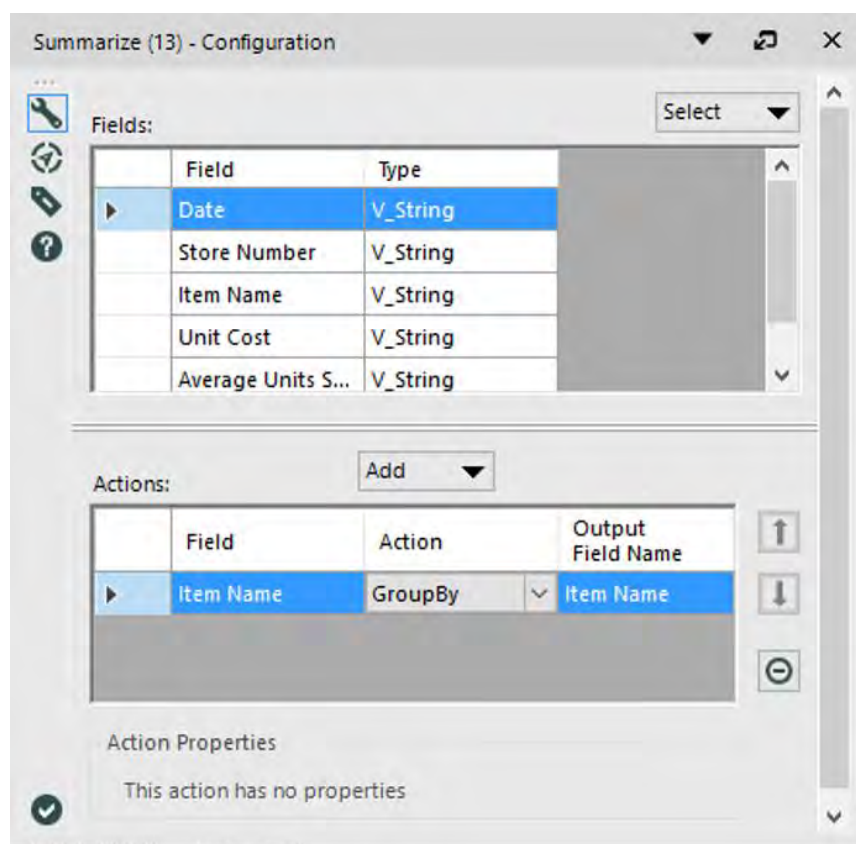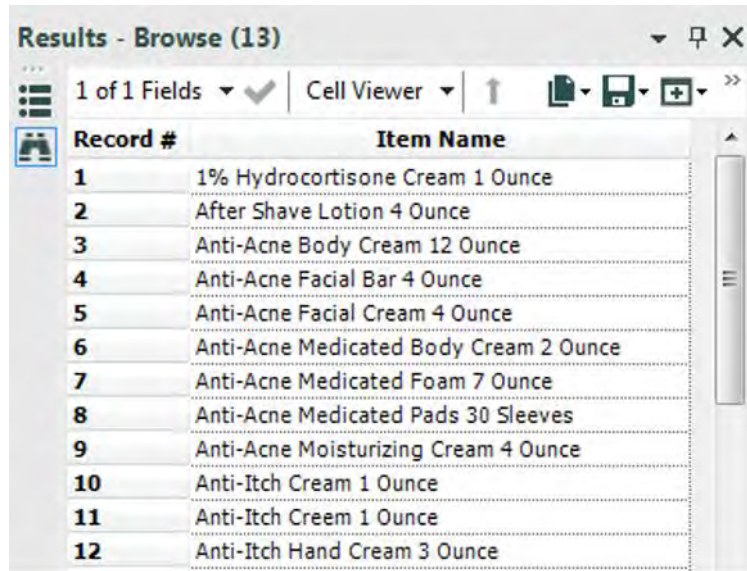
**Figure 7-27 – Expensive Beauty Products
Summarize Configuration**

Let us run the workflow and take a look at the unique list.

Figure 7-28 – Expensive Beauty Products
Browse configuration after Summarize

Now that we have a unique list of products, we can determine which of those products have similar names.
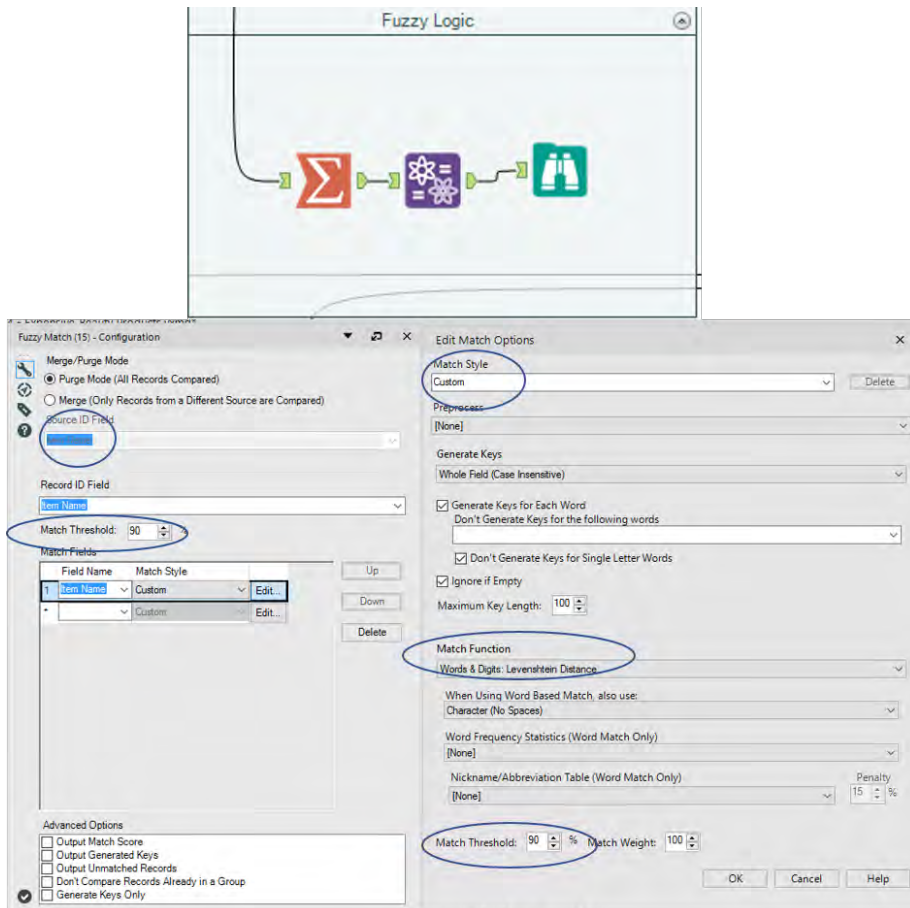
Figure 7-29 – Expensive Beauty Products
Fuzzy Logic Configuration

If we add a Fuzzy Match tool with the above settings, we know we are matching *Item Name* at a minimum 90% threshold. The settings here define a matching algorithm using a word and digit-based *Levenshtein distance method*. This algorithm is looking at the whole field by word (except for single-character word; up to 100 words per field) and keeping anything that it finds with at least a 90% match.

Because of the nature of this methodology, we will introduce an issue in the output data stream. That will result in duplicate records for the fields that match based on multiple keys. In order to correct this, we will isolate the unique records to be kept.

If we add a *Unique* tool after the *Fuzzy Match,* we can see a new field has been created called *Item Name2.* The new field allows us to see which entries match.
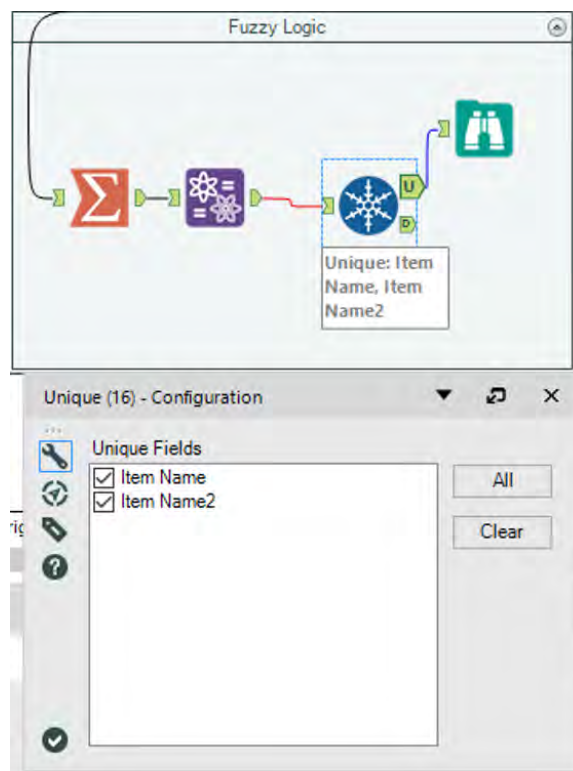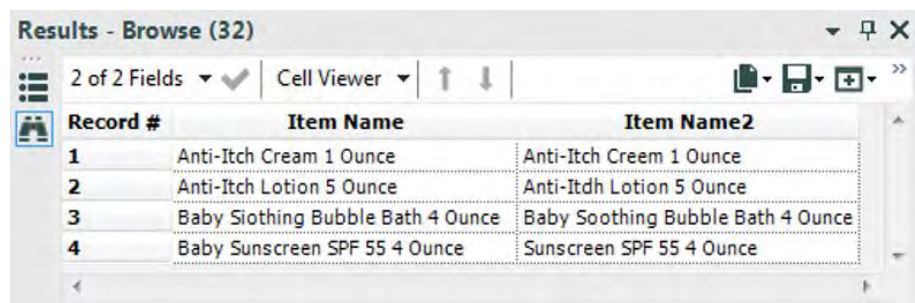
Figure 7-30 – Expensive Beauty Products Unique Configuration

A quick run using the *Browse* tool shows us the matched items.

Figure 7-31 – Expensive Beauty Products
Viewing matching items
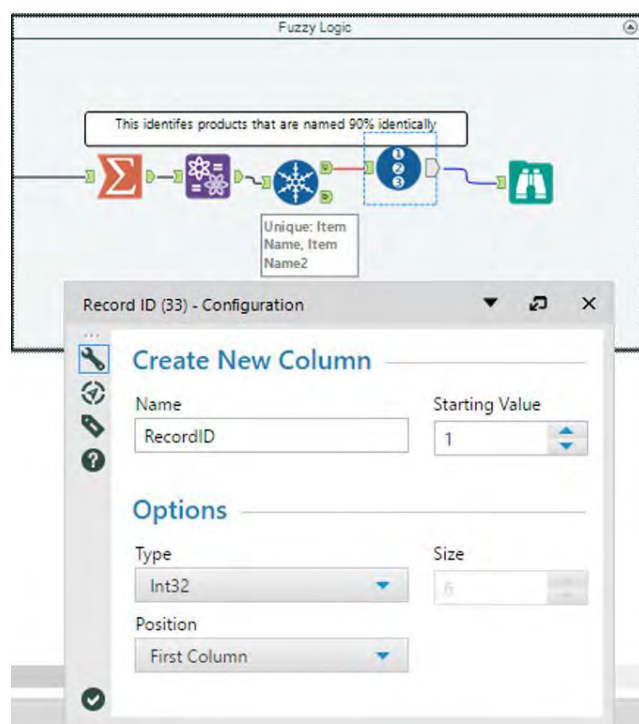
Figure 7-32 – Expensive Beauty Products
Record ID for matched items

We can see by looking at the resulting data stream that we have four groups of items, and three of those are a result of misspellings in the data.

What we can do is provide a numeric grouping for each of these matches, which will solve Jan's problem of data comparison. Since each of the groups is unique, we can use the *Record ID* field to identify them.

Now we can add a comment that will allow Jan to understand what we did, and then we will be ready to combine the data streams.
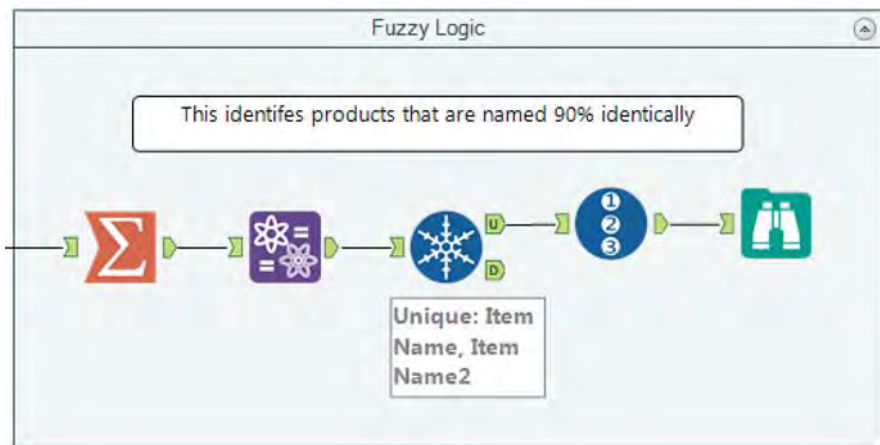
Figure 7-33 – Expensive Beauty Products
Adding comment to indicate the process

In order to combine the data streams, we need to join the data twice so that each part of the group can be flagged with the *Group ID*.
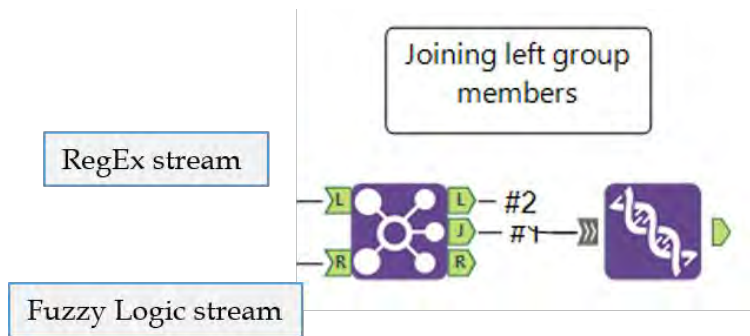
Figure 7-34 – Expensive Beauty Products
First Join for left group members

The *Join* tools settings are shown below.
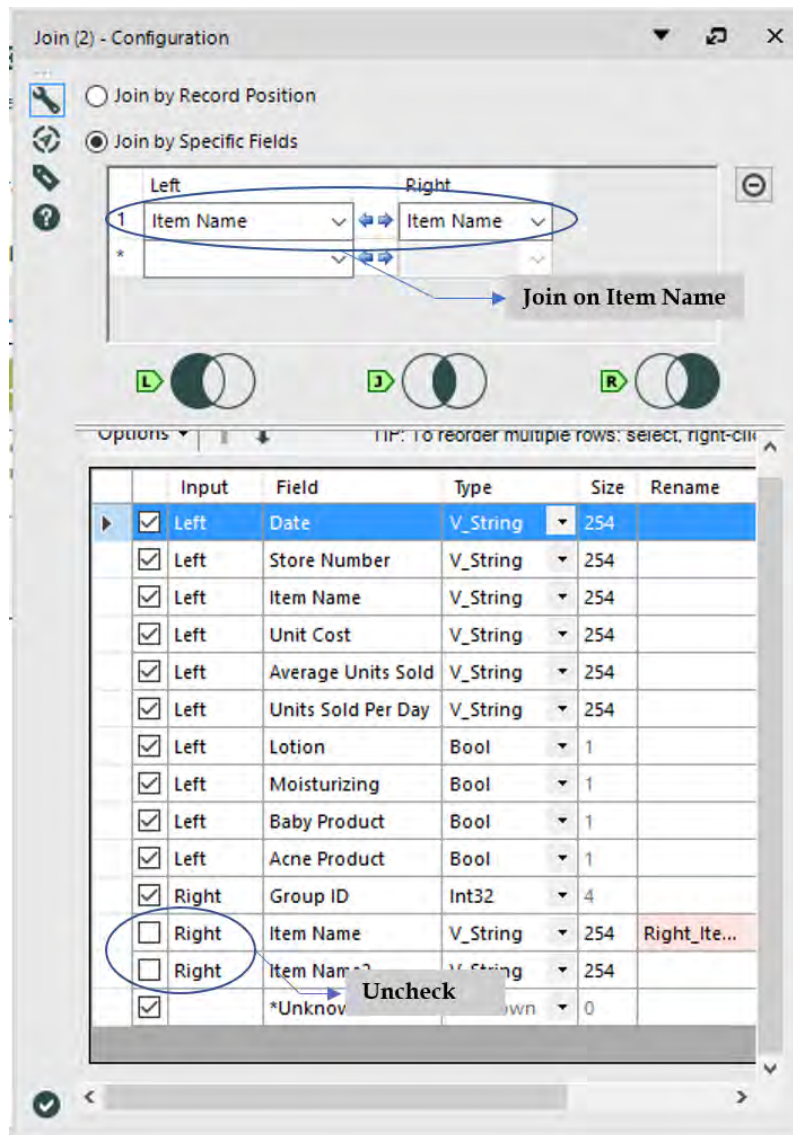
Figure 7-35 – Expensive Beauty Products
Join Configuration for "Item Name"
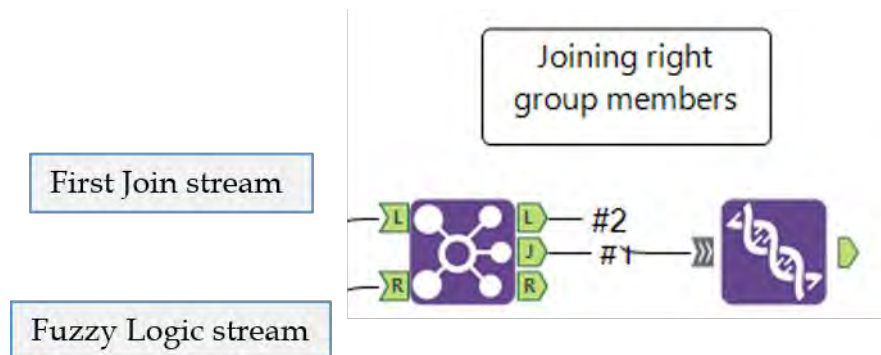
Figure 7-36 – Expensive Beauty Products
Second Join for right group members
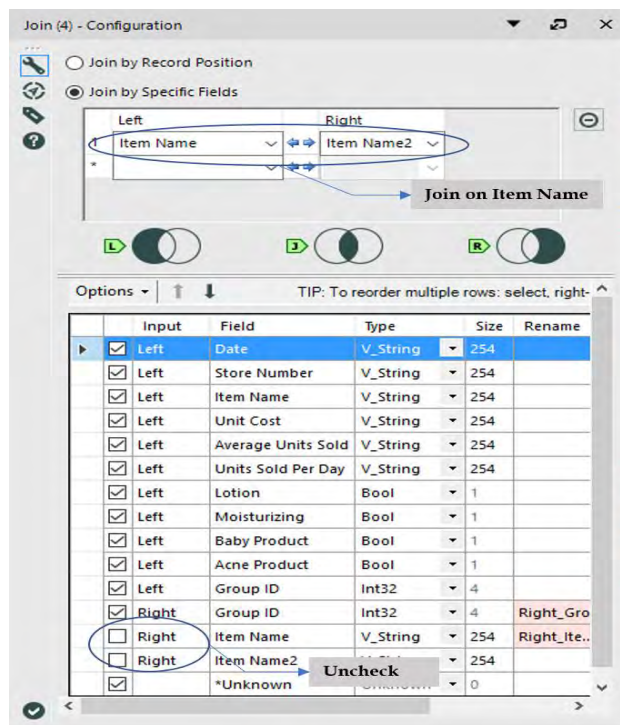
Figure 7-37 – Expensive Beauty Products
Join Configuration for "Item Name2"

Figure 7-38 – Expensive Beauty Products
Browse configuration after Joins

If we applied the previous settings along with the default *Union* settings, we should see that we have two new fields at the end of the data stream. We need to combine these fields in order to give Jan a simple dataset to work from.
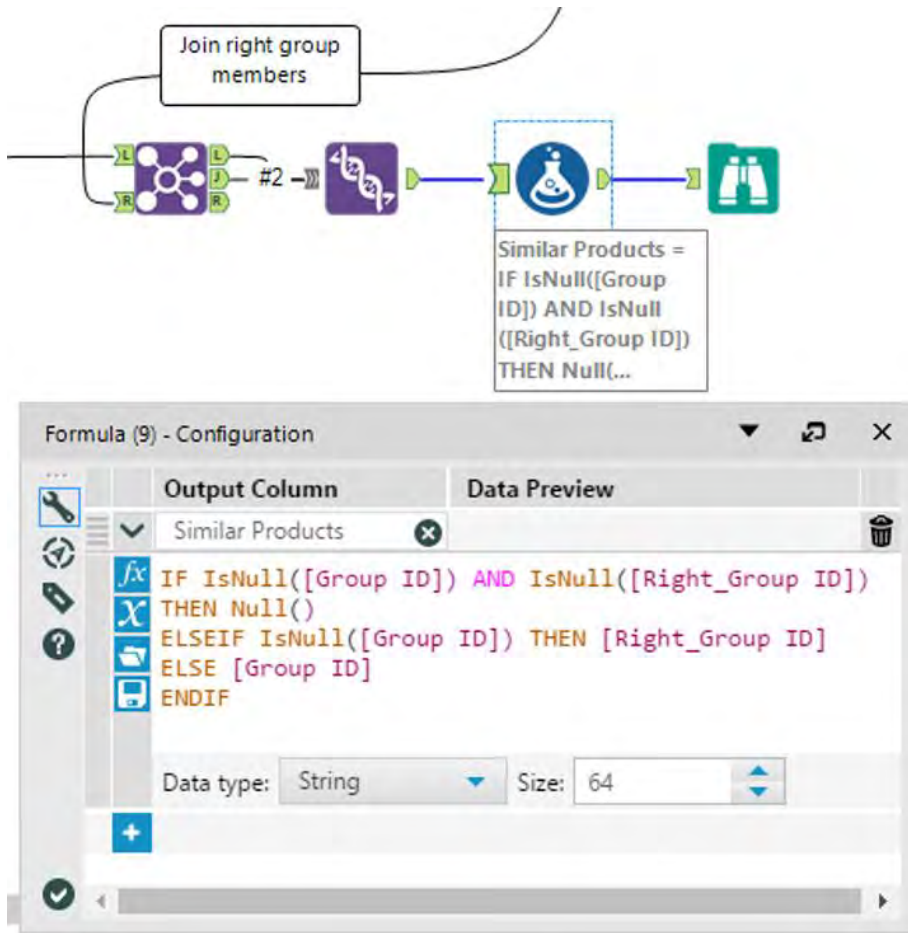
**Figure 7-39 – Expensive Beauty Products
Formula to collate fields**

We can create a conditional formula called *Similar Products* that will bring the fields together:

IF IsNull([Group ID]) AND IsNull([Right_Group ID]) THEN Null()

ELSEIF IsNull([Group ID]) THEN [Right_Group ID]

ELSE [Group ID]

ENDIF
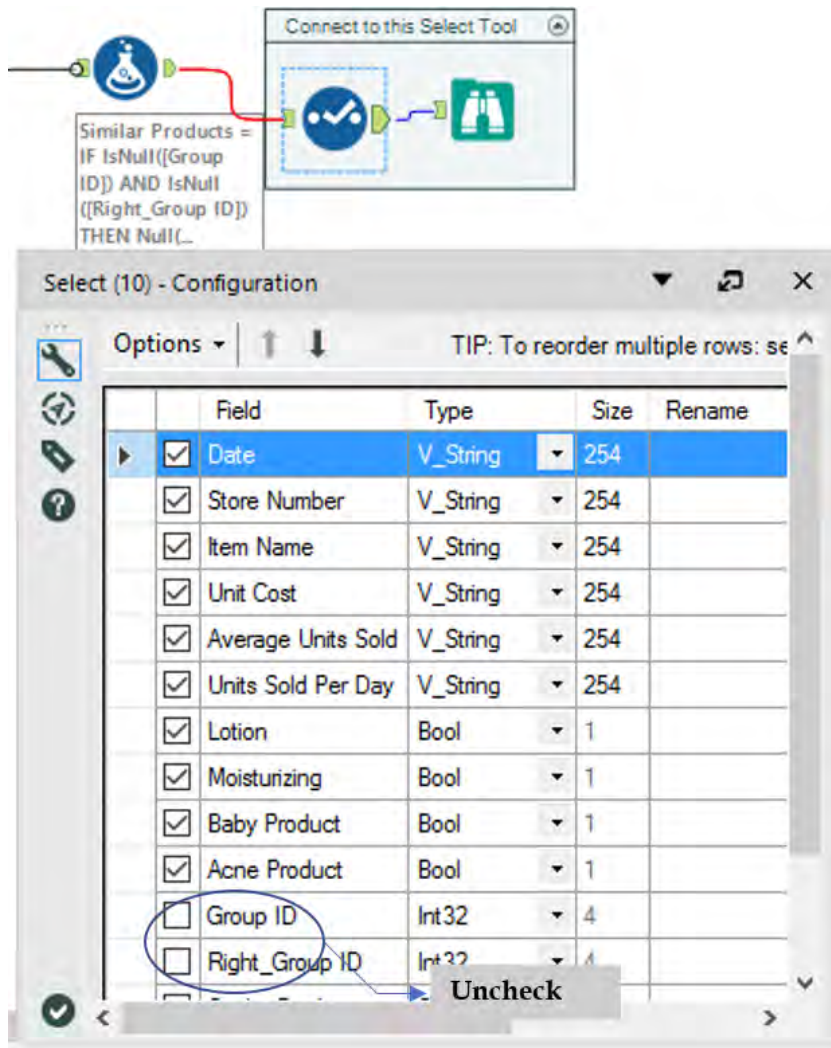
Figure 7-40 – Expensive Beauty Products
Select fields for final output

In order to finish this data stream, we can add a select statement that has the two *Group ID* fields removed. If we then use a *Tool* container, we can very clearly show where any downstream tools should be connected.

The *Expensive Beauty Products* data stream should look like the following image when it is complete.
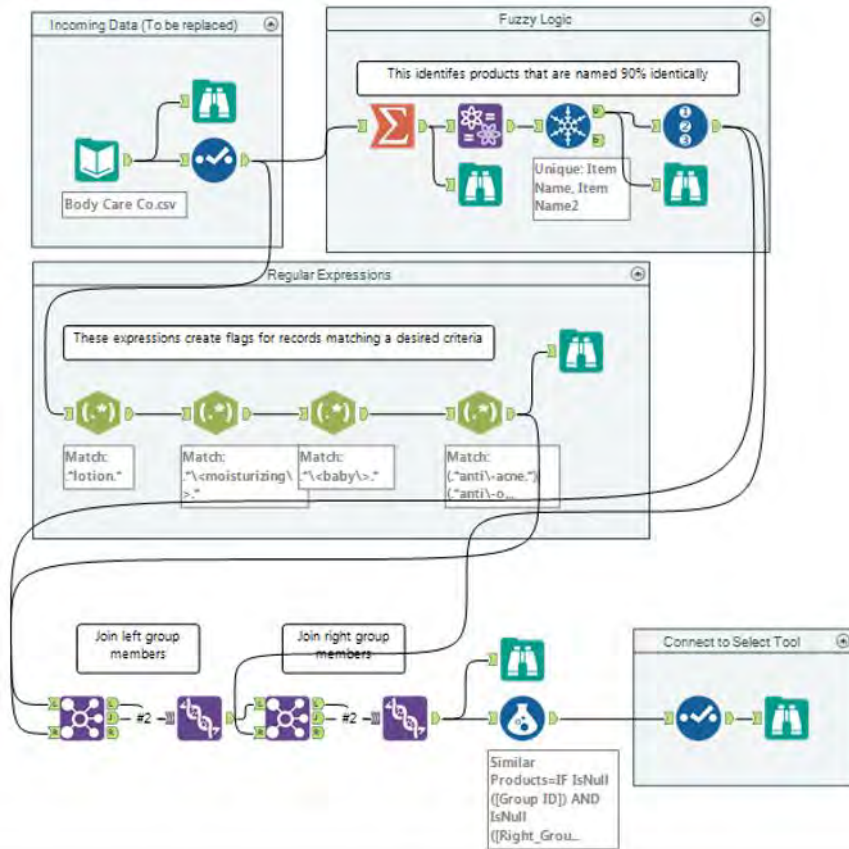
Figure 7-41 – Expensive Beauty Products
Complete data stream

7.8 More Flags

To...	Alteryx Consultants
Subject	More Flags

Hey,

Check out the response we got!

Think you can handle it?
_____

Hey,

That last dataset you created was great. I have a few more items that I need to add to the analysis.

Can you add a flag for the following?

1) All sun protective products (includes "SPF" in the description)
2) All itch relief (includes "anti-itch", "anti-itdh" (I noticed an issue with the data), or "hydrocortisone" in the description)
3) All shampoos (includes "shampoo" in the description)
4) All conditioners (includes "conditioner" in the description)

Thanks again,
Jan

CHAPTER 8

Applications Wanted

To...	Alteryx Consultants
Subject	Let's Build An App For That!
Attached	📄 All Recorded Traffic Tickets.csv

Hey,

I need to pull you in on a project with the Baltimore Ticket Team.

They have asked us if we can build an application that will allow them to do the following:
1. select a file with violations
2. select a date range to limit the data
3. have it create output file(s) of their choice
   a. in-state plates
   b. out-of-state plates
   c. in-state and out-of-state plates (as a single or multiple files).

I think we will hear back from them soon if we build it to these exact standards. We should design this analytic application such that the default is to select Maryland, yet it also gives the flexibility to select any state. Also include a text box so they can limit to a specific location of interest.

Should be fun.

8.1 Tools & Concepts

In this chapter, we will be covering the following Tools and Concepts:

Tools	Concepts
Action	Building Applications
Condition	
Control Parameter	
Date	
Date Filter	
Drop Down	
Error Message	

File Browse

Folder Browse

List Box

Radio Button

Text Box

Tree

8.2 Action

	The *Action* tool modifies the values in other tools.		
	Group	**Input**	**Output**
Figure 8-1 – Action	Interface	None	None

Application questions can be connected to the top *Black Question Anchor*.

A condition tool can be connected to the *Diamond Anchor*.

The *Lightning Bolt Anchor* can be connected to a tool to update that tool based on the *Action* tool's settings.

Action tools are special tools that update settings for the connected tools in the workflow.

Properties Window:

The *Action Configuration* window has only one component.
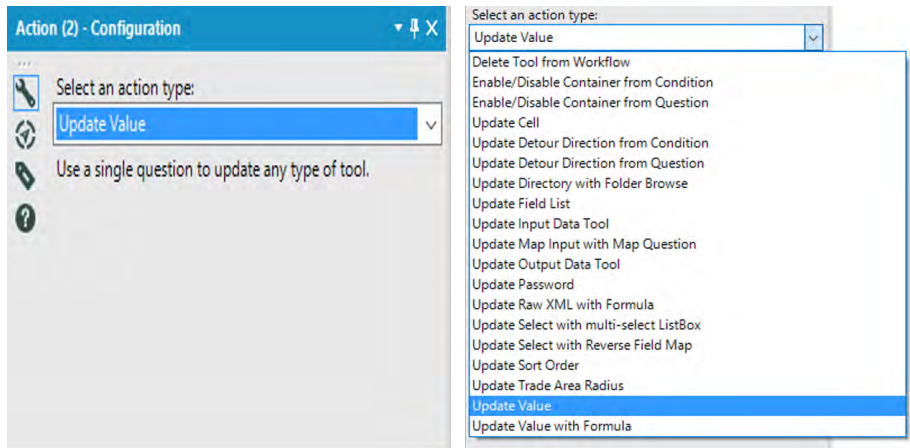
Figure 8-2 – Action configuration with list of Action types

Selecting an action type allows us to configure the changes in the tool connected to the action.

The list of options appears depending on the tool that is connected, and we can see a list of tools that behave in the same way as seen in Appendix J.

The image is a comprehensive list of action types. The methods are self-explanatory, and we will almost always use the default action associated with each tool.

8.3 Condition

	The *Condition* tool allows us to trigger different results depending on the condition in the tool.		
Figure 8-3 – Condition	**Group**	**Input**	**Output**
	Interface	None	None

At least one *Action* tool should always follow this tool.

Multiple Application questions can be connected to the top *Black Question Anchor*.

Diamond Anchor T connects to an Action tool and provides True result if the condition is true.

Diamond Anchor F connects to an Action tool and provides False result if the condition is false.

Properties Window:

The *Condition Configuration* window allows us to create a Boolean expression using the connected questions. The configuration is similar to a Filter tool. An Expression is written and if values going in meet the criteria, they come out the True output. If they do not meet the criteria, they come out of the False output.

Use the tabs to build an Expression:

- Variables: There are 2 types of variables in the Condition tool:

- o Variables refer to incoming connections and will be the naming convention: #*Connection Number: (Interface Tool Name) Description*. You can reference multiple connections in a formula as long as they are all connected to the Condition tool.
- o Constants refer to global variables in a workflow. Engine, Question and User constants are available to build the expression.
- Functions: The entire functions library is available to build the expression.
- Saved_Expressions: Allow you to save and load custom expressions that can be used over and over without having to formulate the expression from scratch.
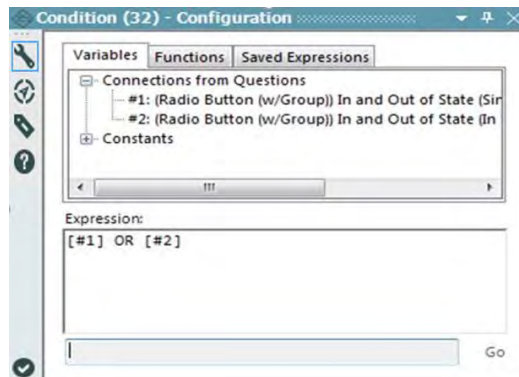
Figure 8-4 - Condition Configuration

In this case, since connections 1 and 2 are both *Radio Button* tools, their values are either true or false.

The expression is true if either of these options was selected.

- If the expression is true, then the true (T) path has a true value passed down it, and the false (F) path has a false value passed down it.
- If the Expression is false, then the true (T) path has a false value passed down it, and the false (F) path has a true value passed down it.

8.4 Control Parameter

Figure 8-5 – Control Parameter	A *Control Parameter* is the input for each iteration of a Batch Macro		
	Group	**Input**	**Output**
	Interface	None	None
The *Black Question Anchor* connects the results to other tools, which have incoming anchors.			

The Control Parameter input will appear as a ¿ below its input arrow on the macro tool icon. For each record coming into the Control Parameter input, the entire macro will be re-configured and run beginning to end. The Control Parameter must have an Action associated with it, such as Update Value, Update Input, and Update Output Actions.

Properties Window:

The *Control Configuration* window has only one component.
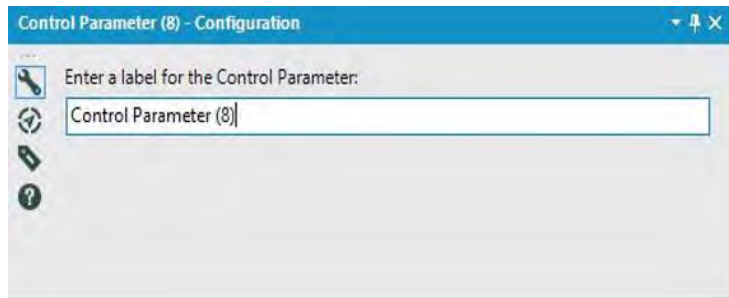
Figure 8-6 - Control Configuration

Enter the label to be displayed allows us to write the prompt that the end user sees.

8.5 Date

Figure 8-7 –Date	The *Date* tool allows for date selection from a calendar menu.		
	Group	**Input**	**Output**
	Interface	None	None
The *Black Question Anchor* connects the results to other tools, which have incoming anchors. The *Date* tool creates a string of the selected date in the format YYYY-MM-DD.			

Properties Window:

The *Date Configuration* window has only one component.
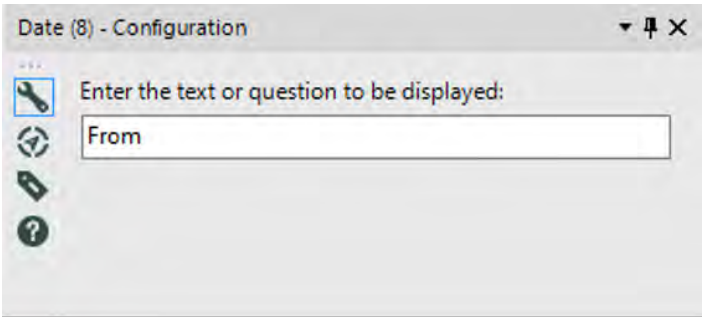
Figure 8-8 - Date Configuration

Enter the text or question to be displayed allows us to write the prompt that the end user sees.

8.6 Date Filter

Figure 8-9 –Date Filter	The *Date Filter* tool provides the ability to filter based on dates.		
	Group	**Input**	**Output**
	Preparation	See below	See below
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros. *Input:* A data stream with at least one date field. *Output:* The input data stream limited to an appropriate date range.			

Properties Window:

The *Date Filter Configuration* window has two components.
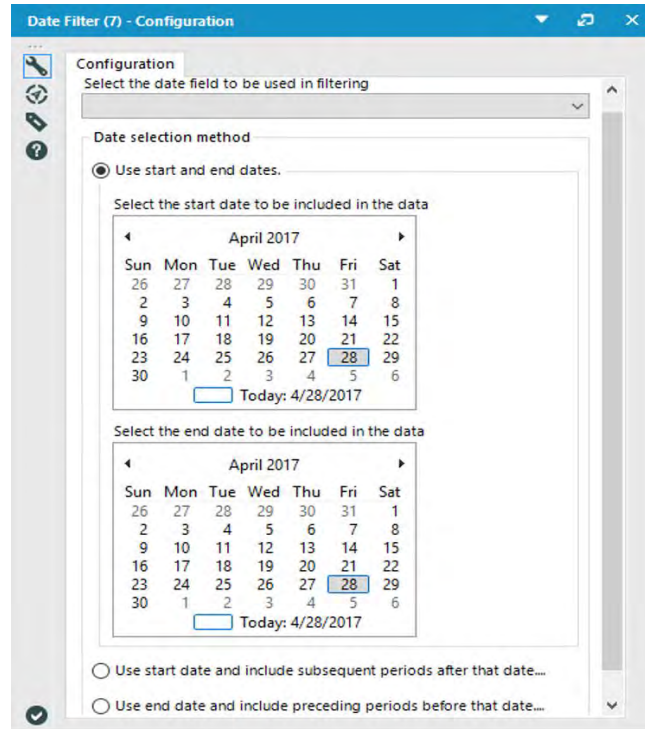
Figure 8-10 – Date Filter Configuration

- The *Select the date field to be used in filtering* allows in selecting the date field in the incoming data source.
- *Date selection method* allows us to define which method the date filter will use to limit the data.
- *Select the start and end dates* allows us to select two dates on a calendar and only keep the dates restricted between them.

Figure 8-11 – Date Filter Configuration

- *Use start date and include subsequent periods after that date* allows us to select a date to start from and then define how long after that date we want to include.
- *Use end date and include preceding periods before that date* allows us to select a date to end the data and then define how long before that date we want to include.

8.7 Drop Down
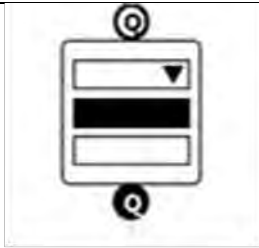
  Figure 8-12 – Drop Down	The *Drop-Down* tool allows for the selection of a single value from a provided list.

Group	Input	Output
Interface	None	Data stream

The *White Question Anchor* receives information from other tools to populate the list.

The *Black Question Anchor* connects the results to other tools that have incoming anchors.

The result of this question is the string of the selected option.

Properties Window:

The *Drop-Down Configuration* window has three basic components.

- *Enter the text or question to be displayed* allows us to write the prompt that the end user sees.
- *List Values* allows us to select a method of populating a list and any related properties.
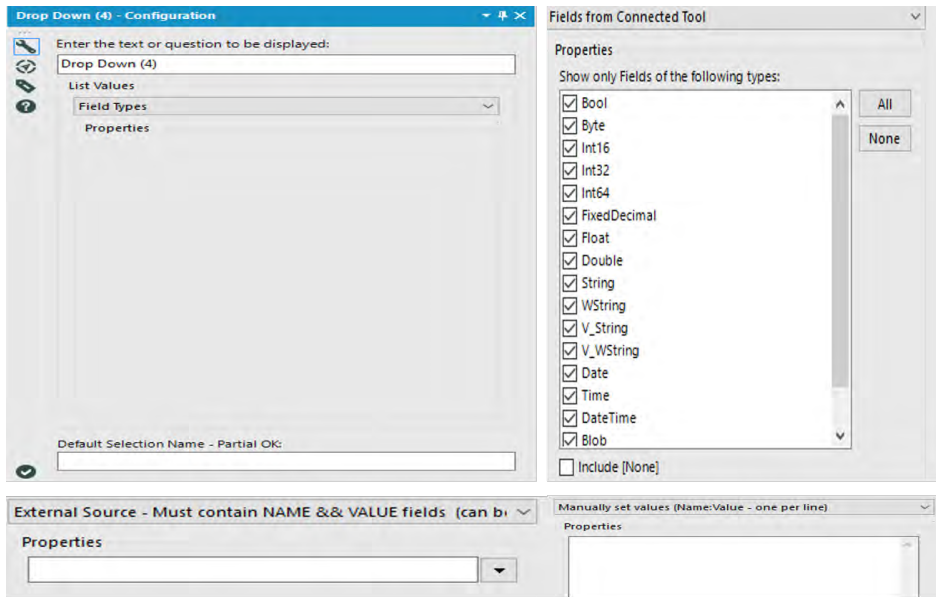
Figure 8-13 – Dropdown Configuration

- *Default Selection Name – Partial OK* allows us to set a default value by typing in a unique part of the option.
- *Field Types* are the list of field types.
  - *External Source – Must contain NAME & VALUE fields (can be a relative path)* uses a reference file that has both a Name and a Value field.
  - *Field from Connected Tool* allows us to connect to a tool and select one of the fields from that data connection, as long as they match the file type in the multi-select list.
  - *Manually set Values (Name: Value – one per line)* allows us to create a list of options in the properties box. The format for this is: *Text to display: Resulting value in data.*
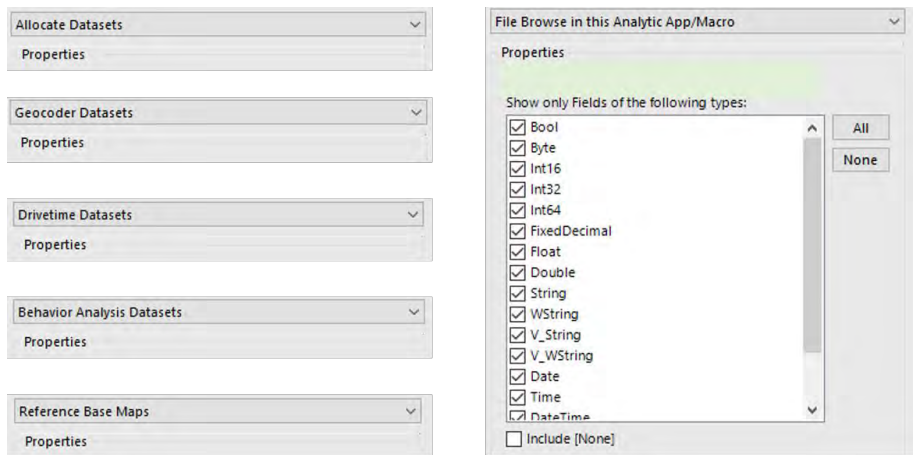
Figure 8-14 – Dataset Types

- *Allocate Datasets* allows us to select from the Allocate dataset list we have available.
- *Geocoder Datasets* allows us to select from the Geocoder dataset list we have available.
- *Drivetime Datasets* allows us to select from the Drivetime dataset list we have available.
- *Behavior Analysis Datasets* allows us to select from the behavior dataset list we have available.
- *Reference Base Maps* allows us to select from the base maps we have available.
- *File Browse in this Analytic App/Macro* connects to a file browse tool and allows for selection of one of the fields from that data connection, as long as they match the file type in the multi-select list. *Include [None]* means we are providing an option that will result in no selection.

8.8 Error Message

Figure 8-15 –Error Message	The *Error Message* tool will throw an Error message to the end user of an app or macro. Once the error message is thrown, all downstream processing stops.		
	Group	**Input**	**Output**
	Interface	None	None
The *Black Question Anchor* at the top connects the results to other tools, which have incoming anchors.			

Properties Window:

The *Error Configuration* window has only one component.
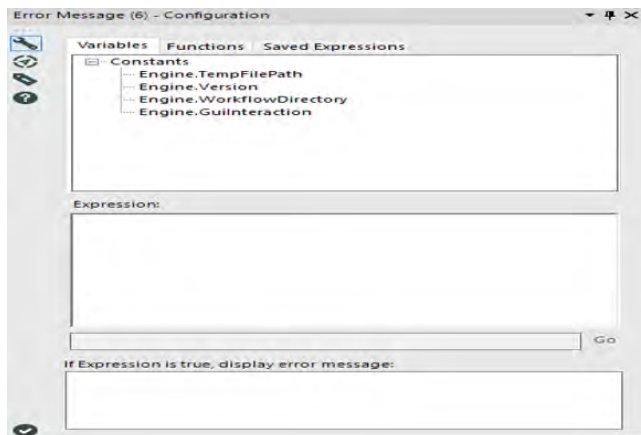
Figure 8.16 - Error Configuration

Using the Formula editor, construct an expression where if the criteria is met, an error will be thrown. Use the tabs to build an Expression:

- Variables: There are 2 types of variables in the Condition tool:
    1. Variables refer to incoming connections and will be the naming convention: #*Connection Number: (Interface Tool Name) Description*
    2. Constants refer to global variables in a workflow. Engine, Question and User constants are available to build the expression.
- Functions: The entire functions library is available to build the expression.
- Saved_Expressions: Allow you to save and load custom expressions that can be used over and over without having to formulate the expression from scratch.

8.9 File Browse

The *File Browse* tool allows connecting to a file of choice instead of a predesignated file.		
Group	**Input**	**Output**
Interface	None	None

Figure 8-17 – File Browse

> The *Black Question Anchor* connects the results of questions to anchors on other tools.
>
> The tool can be used to update any tool that used a file connection. When using this tool, the end user should know what information should be included in the data so the module runs correctly.

Properties Window:

The *File Browse Configuration* window has three components.
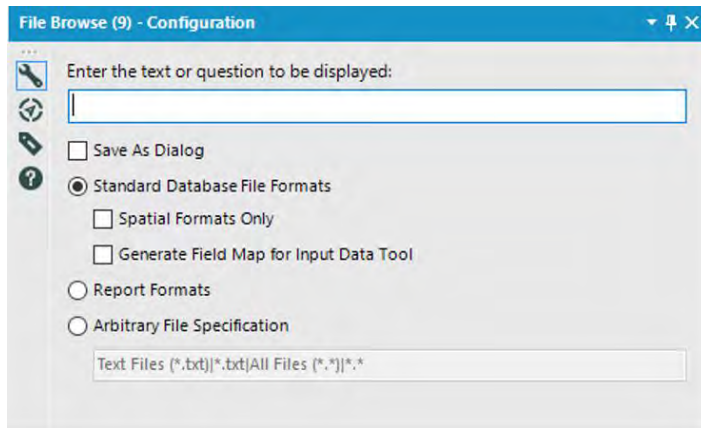
Figure 8-18 – File Browse configuration

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.
- *Save As Dialog* allows us to output the data as a dialog box instead of a file.
- *Standard Database File Formats* are a typical file browse.

o *Spatial Formats Only* will allow only file types that can have spatial objects.

o *Generate Field Map for Input Data* forces the end user to align the fields in the selected data source with the fields in the development data source.

● *Report Formats* option allows us to select only report formats that are compatible with the *Render* tool.
● *Arbitrary File Specification* allows us to define specific file formats that we want to be accessed or preferred.

8.10 Folder Browser
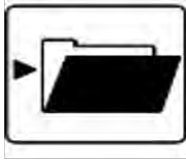
	The *Folder Browse* tool displays a folder browse control in an app or macro. The directory path specified by the user is passed to downstream tools.		
Figure 8-19 – Folder Browse	**Group**	**Input**	**Output**
	Interface	None	None
The *Black Question Anchor* connects the results of questions to anchors on other tools.			

Properties Window:

The *Folder Browser Configuration* window has only one component, the text here will be presented to the user instructing them how to use the Folder Browse control.
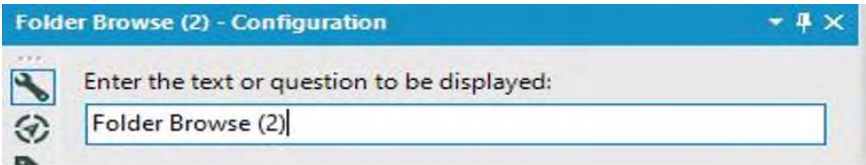
Figure 8-20 - Folder Browser

8.11 List Box

Figure 8-21 – List Box

The *List Box tool* will display a multi-selection check box list in an app or macro. The selections specified by the user are passed as values to downstream tools.

Group	Input	Output
Interface	None	None

The *Black Question Anchor* connects the results of questions to anchors on other tools.

Properties Window:

The text here will be presented to the user instructing them how to use the List Box control. When checked, all items in the list box will be selected (checked) for the user.
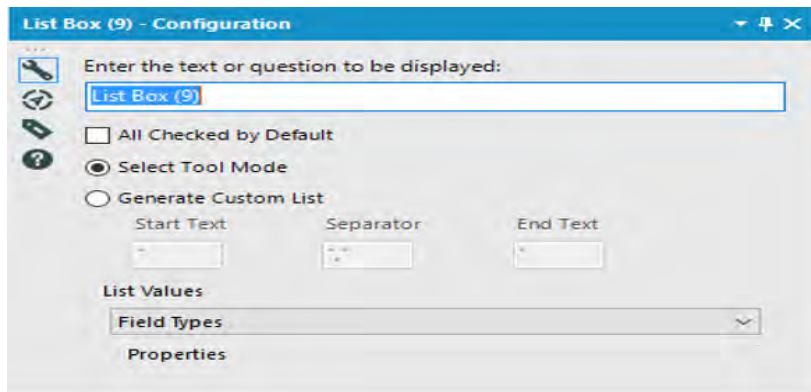
Figure 8-21 – List Box

There are two modes in which this control should behave. Choices are:

- Select tool Mode: Use this option when updating a Select tool.
- Generate custom list: Not visible to the App Consumer, this option returns a list of the selected values with the specified Start, Separator, and End characters. This option is usually used in tandem with the List Values: "External Source" or "Manually set values" described below.

The List Values drop down selection determining how the list is populated. Options are explained in detail below:

Field Types: The App Author specifies what field types to present to the App Consumer. The App Consumer will only see fields that match those field types.

External Source: A file contains the list of values to present to the App Consumer. The user will see the names in the file.

Fields from Connected Tool: The App Consumer will see a populated list box of fields present in the tool connected to this tool.

Fields from Macro Input: The Macro Consumer will see a populated list box of fields present in the Macro Input tool connected to this tool.

Manually set values: The App Author manually enters the list values to display to the end user.

File Browse in this App/Macro: The App Consumer will see available fields from their specified input that match selected field types.

8.12 Radio Button
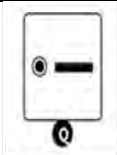
<table>
<tr>
<td rowspan="3">Figure 8-22 – Radio Button</td>
<td colspan="3">The Radio Button tool creates a single select option for use in the Interface designer.</td>
</tr>
<tr>
<td>Group</td>
<td>Input</td>
<td>Output</td>
</tr>
<tr>
<td>Interface</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td colspan="4">The Black Question Anchor connects the results of questions to anchors on other tools.

The question creates a Boolean. It will be a single select option along with all other single select options in the group (all, if no groups are defined).</td>
</tr>
</table>

Properties Window:

The *Radio Button Configuration* window has three components.
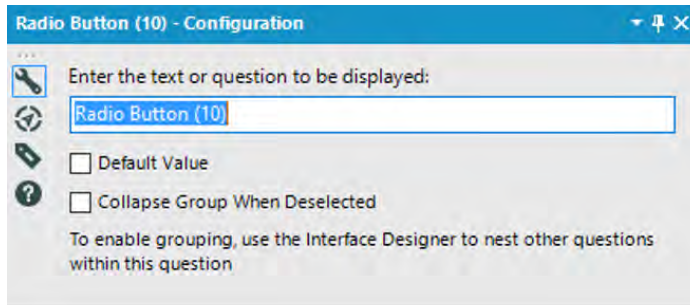
Figure 8-23 - Radio Button

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.
- *Default Value*, when selected, will use this question as the default selected.
  Best practice is to make sure this question is first in the list of values.
- *Collapse Group When Deselected* allows us to create a group of questions that only display when the question is selected.

8.13 Text Box

Figure 8- 24 – Text Box	The *Text Box* tool allows the user to enter a string of choice.		
	Group	**Input**	**Output**
	Interface	None	None
The *Black Question Anchor* connects the results of questions to anchors on other tools. The questions produce a string output.			

Properties Window:

The *Text Box Configuration* window has five components.
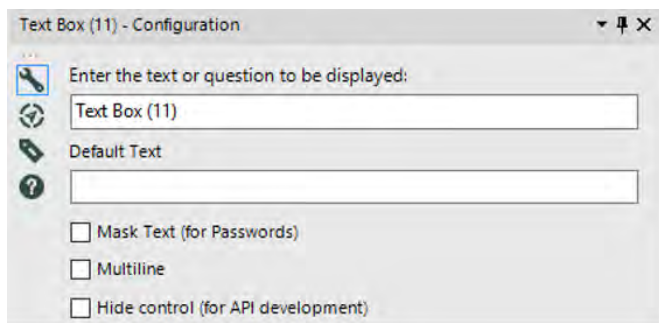
Figure 8- 25 Text Box

- *Enter the text or question to be displayed* allows us to write the prompt the end user sees.

- *Default Text* is a string that will populate the text box.
- *Mark Text (for Passwords)*, when checked, replaces all visible text characters in the response with "*".
- *Multiline*, when checked, allows the end user to create a multiple line input and makes the text entry larger.
- *Hide control (for API development)* gives the ability to hide the entry field from the interface designer but allows the API to continue to edit the entry.

8.14 Tree

	The *Tree tool* will display an organized, hierarchical data structure in an app or macro. The selections made by the user are passed as values to downstream tools. The values returned from trees are separated by a new line character (\n).

Figure 8- 26 – Tree

Group	Input	Output
Interface	See Below	None

The *Black Question Anchor* that can accept a list of Allocate Datasets.

<u>Properties Window:</u>

The *Tree Configuration* window has the below components.
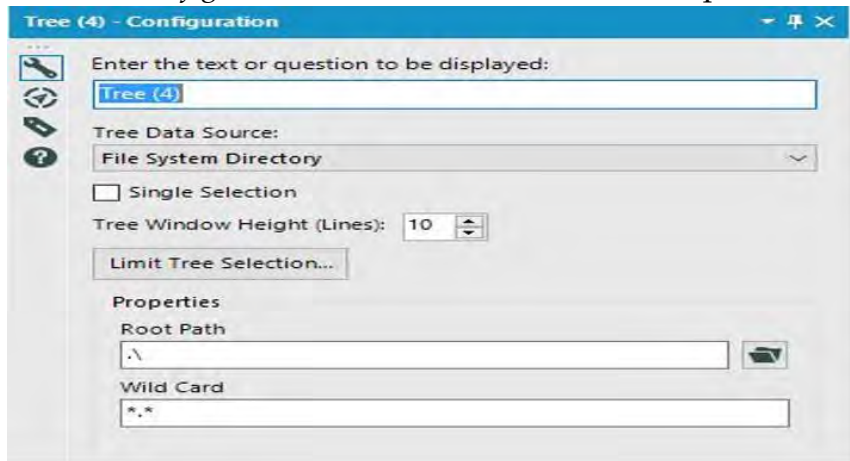
Figure 8-27 - Tree

Enter the text or question to be displayed allows us to write the prompt the end user sees.

There are 5 types of trees, each with different additional Configuration Properties. Click on the tree data source below to reveal configuration instructions each.

- Allocate Geography: Use this option to display a list of Allocate Geographies to the App Consumer.
- Allocate Variables: Use this option to display a list of Allocate Variables to the App Consumer.
- File System Directory: Use this option to display a file system directory to the App Consumer.
- Custom XML: Use this option to display your own list to the App Consumer. They will see the values, not the XML.

- Custom File/Database: Use this option to display a list to the App Consumer that comes from a custom file or database.

Single Selection: When checked, the App Consumer will only be able to select one value from the tree.

Tree Window Height: specify how many lines tall the Tree control should display to the user.

Limit Tree Selection: Exposes the Filter Tree Options dialog box where the App Author can pare down what levels of the tree get displayed to the App Consumer.

8.15 Let's Build an App For That!

To...	Alteryx Consultants
Subject	Let's Build An App For That!
Attached	All Recorded Traffic Tickets.csv

Hey,

I need to pull you in on a project with the Baltimore Ticket Team.

They have asked us if we can build an application that will allow them to do the following:
1. select a file with violations
2. select a date range to limit the data
3. have it create output file(s) of their choice
    a. in-state plates
    b. out-of-state plates
    c. in-state and out-of-state plates (as a single or multiple files).

I think we will hear back from them soon if we build it to these exact standards. We should design this analytic application such that the default is to select Maryland, yet it also gives the flexibility to select any state. Also include a text box so they can limit to a specific location of interest.

Should be fun.

Alteryx allows us to develop Analytic Applications that make it so the workflow can be modified by user inputs. Apps have a couple of special properties that allow us to:

- Use an interface to run the workflow instead of opening the file in Alteryx Designer.
- Publish the workflow to an Alteryx Gallery, giving users without Alteryx Designer the ability to run workflows.
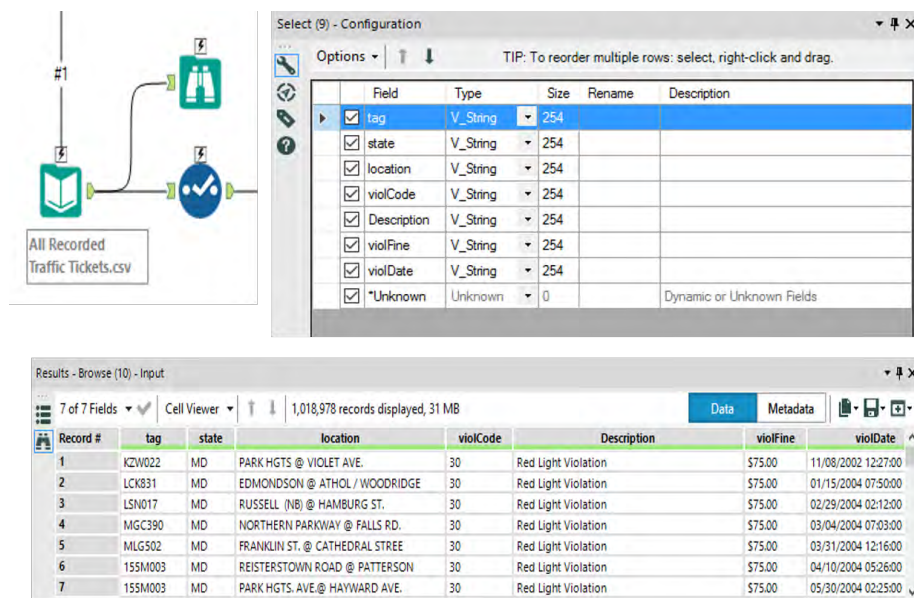
Figure 8-28 - Traffic Tickets- Opening , Selecting and Viewing data

We are going to approach this by laying out the workflow and then adding the interface tools that allow for the modification. So, let us get started by connecting to the file *All Recorded Traffic Tickets.csv.* Since they haven't asked us to do anything with the Fine amount field, we can leave it in a string format.

However, since we know we need to limit the data by dates, we are going to want the *violDate* in a date format. If we take a look at the format in *violDate*, we see it is a string formatted date with an *AM/PM* flag. If we look at the *DateTime*, tool we can see that we do not have a matching date format, so we need to use a formula tool to manipulate the string first.

Figure 8-29 - Traffic Tickets-Conversion from string to date

Select the format that matches the incoming string field

day, dd Month, yyyy
dd-MM-yy
dd-MM-yyyy
dd-Mon.-yy
dd Month, yyyy
dd/MM/yy
dd/MM/yy hh:mm:ss
dd/MM/yyyy
dd/MM/yyyy hh:mm:ss
dy., Month dd, yyyy
HH:mm:ss
MM-dd-yy
MM-dd-yyyy
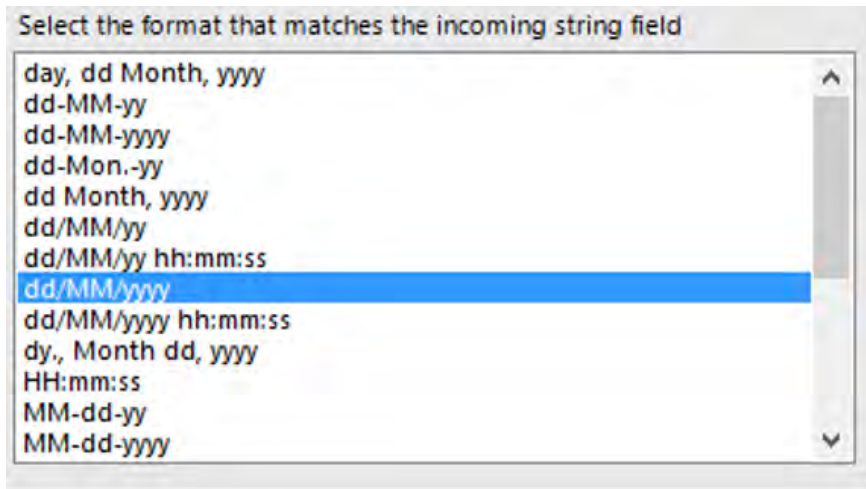
Figure 8-30 - Traffic Tickets- Input String Format

Since we are only going to allow the end user to select dates, we can drop the time portion of the field. This means that we only need the first 10 characters. The formula is: *LEFT([violDate],10)*
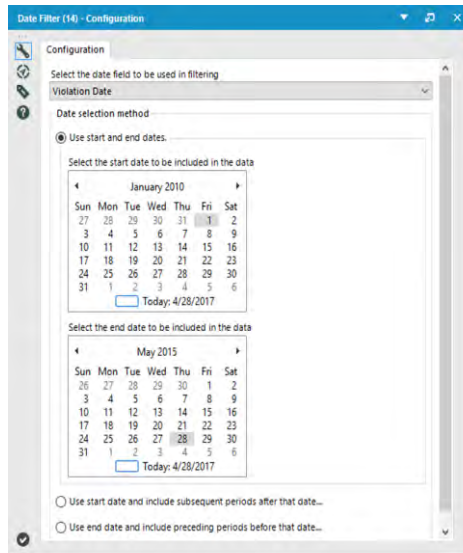
Figure 8-31 - Traffic Tickets- Using Date Filter tool
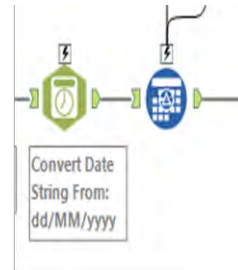
Now that we have a field in date format, we can create the *Date Filter* tool. Because we only have one date type field, *Filter Date* populates in the field selection. We are going to allow the end user the ability to select the beginning and end date, so we can select that option and leave all other settings alone. It does not matter that both dates are referring to today because we are going to replace these values.

Note: While we are testing, it may make it hard to see what is going on if we do not select a date range that is reflected in the data.
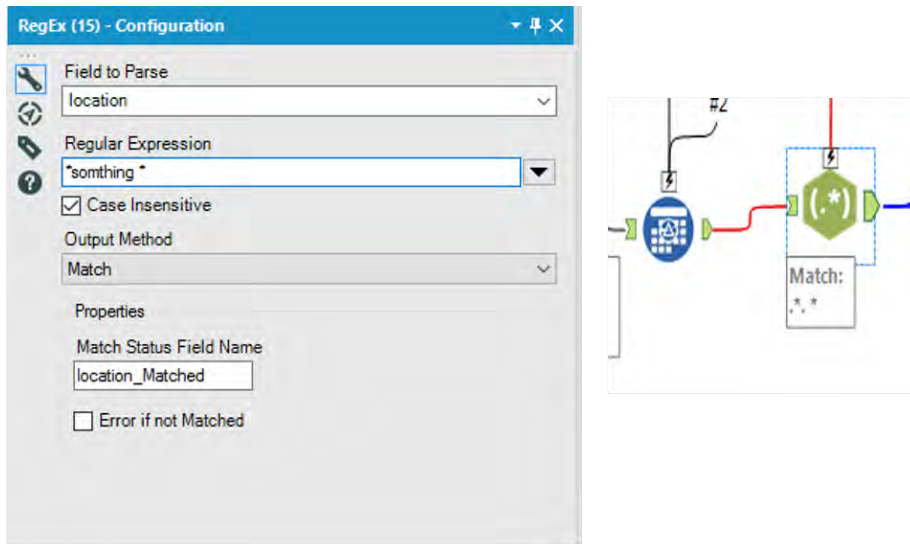
Figure 8-32 - Traffic Tickets- Using RegEx to flag string

Now we can add the feature they didn't ask for. We can write a regular expression that allows us to capture partial or full location names to limit the data. We want to look at the location field and then create a regular expression that will tell us if the field matches the user input. We can use the following expression along with a *text box interface* tool to do this.

.*something.*

Using the above expression, we will update *something* with the user-specified value so we can flag the appropriate records. (If the user wants everything, they can leave the field blank resulting in the *expression .*.*, which will mark all fields as true.)
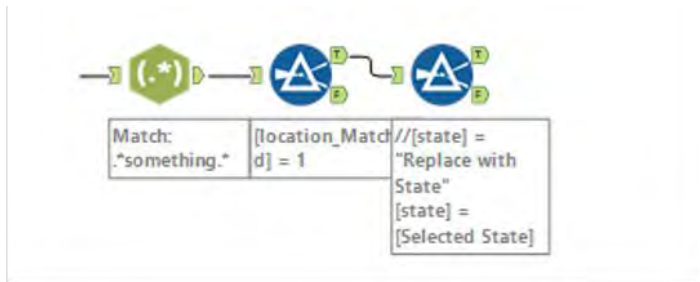
Figure 8-33- Traffic Tickets- Match with flagged string

Now that we have flagged which records match the desired location, we can use the *location_Matched* field that was generated in the *RegEx* tool to filter the data. If we use the basic filter for the *location_Matched,* we can see the field is true and the function becomes the expression *location_Matched* = 1.

Now that we have limited the data to the appropriate locations, we can flag which state they may want to analyze in detail. For now, we can use a custom filter expression in order to finish building the core data stream.

[state] = "Replace with State"
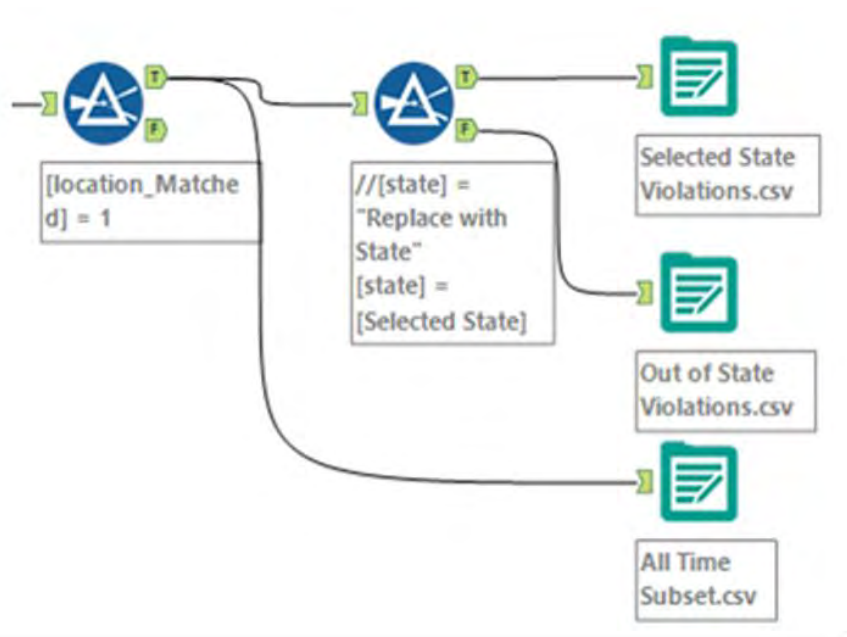
Figure 8-34- Traffic Tickets- Getting three possible outputs

The last step in the workflow is to create the three possible outputs. We can add them to the end of the data stream where appropriate.

- Connect the *Selected State Violations.csv* to the True output on the state *filter*.
- Connect the *Non-Selected State Violations.csv* to the False output on the state *filter*.
- Connect the *All Matching Violations.csv* to the True output on the Location Match *filter*.

We can put the three outputs in *Tool Containers*. This is not for organizational purposes like we saw earlier. When we are building applications, we can enable and disable *Tool Containers* based on the user selection. In this case, we are going to disable

all three tool containers so the only time the output is created is when we have selected that option.
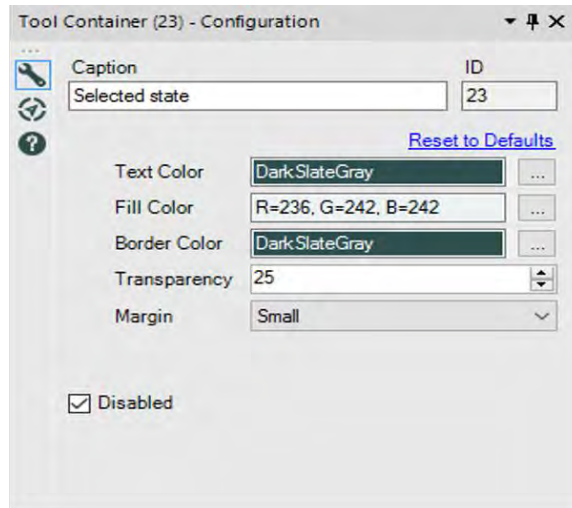
Figure 8-35 - Traffic Tickets- Using Tool Container to disable it except when appropriate option is selected

We can disable the containers using the *Disabled* option in the bottom-left corner of the properties windows. This option only shows the container and not the contents and will turn off all the three outputs.
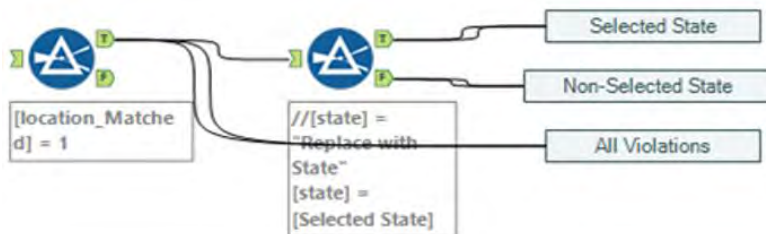
Figure 8-36- Traffic Tickets- Selected Options
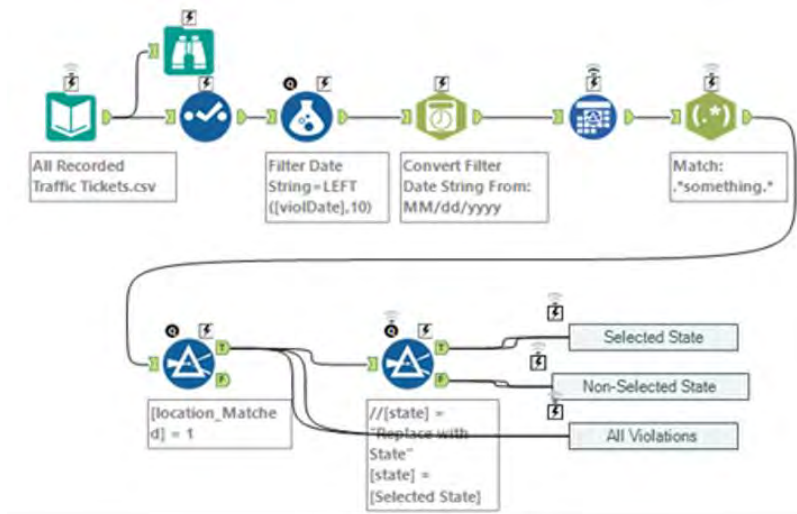
Figure 8-37- Traffic Tickets- Basic data stream

We now have the basic data stream designed for the analytic app. We will have to modify some of the tools, but almost everything is ready for the transition. Let us look at the list of interactions we are providing the Baltimore Ticket Team.

1. Select files.
2. Choose the date range.
3. Name a location (optional).
4. Select a State (Maryland, by default).
5. Select which combination of the three output files is to be returned.
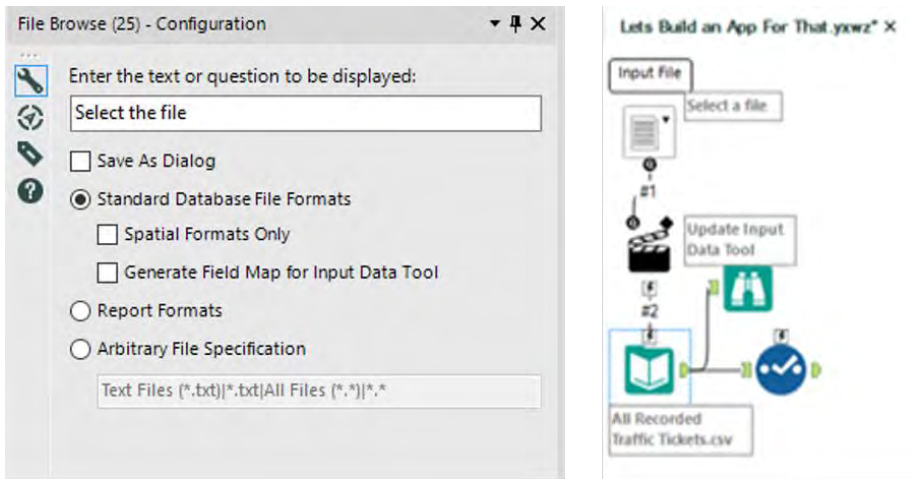
Figure 8-38 - Traffic Tickets- File Browse

The first question we can address is which file should be selected. If we drop a *File Browse* tool onto the canvas, we can see that the tools show their *lightning bolt* and *question mark* anchors indicating we are now working with an *Analytic Application or Macro*. We can then change the text so it reads *Select a file* and use the remaining default settings.

We know we want to connect the file selection question to the *Input Data* tool; however, the *lightning bolt anchor* only connects to *Action* tools. We could bring in the tool, but Alteryx has a clever way of handling it: When we connect a question directly to a *lightning bolt anchor,* it brings an *Action* tool onto the canvas with all of the default settings in place. In this case, it will do exactly what we want, and we can move onto the next question. (Click the Action tool On and Off to confirm the settings.)
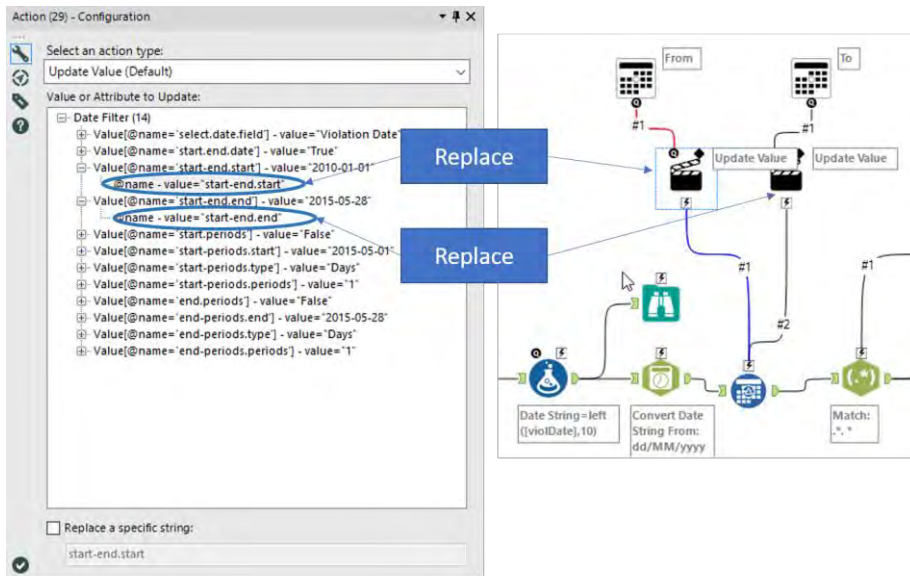
Figure 8-39 - Traffic Tickets- Using Action Tool for Date

The next question is slightly more complicated. We need to set the beginning and end dates for the selected timeframe. We can drag in two *Date* tools and label one *From* and the other *To* so we know which dates refer to the beginning and end of the range. This time, when we connect the questions to the *Date Filter* tool, there is no indication what value should be updated. What we want to do is update two values in the *Date Filter*. However, since there are multiple inputs, we need to select which value each action should update.

- From's *Action* tool should update the variable s*tart-end.start*
- To's *Action* tool should update the variable *start-end.end*

Now that we have this filter getting its values updated, we can move onto the next question.
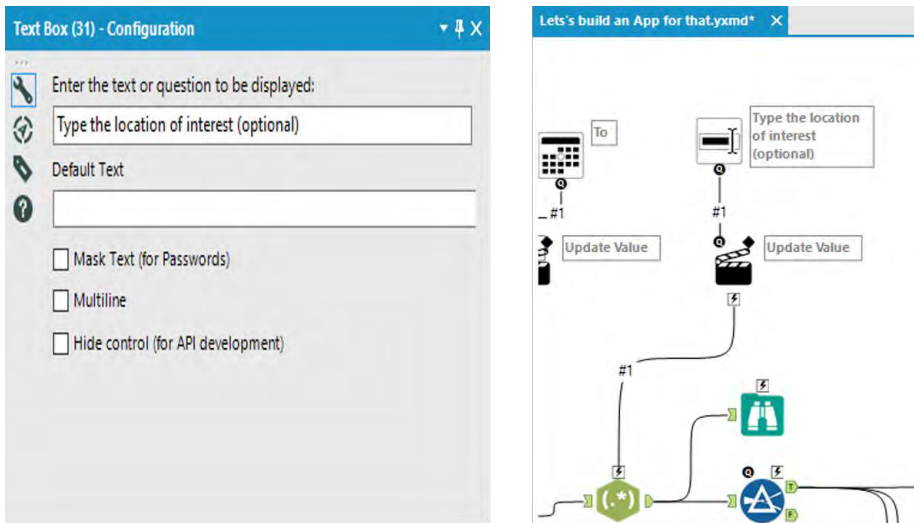
Figure 8-40 - Traffic Tickets- Using Text Box to specify location

This time, we are going to add a *Text Box* tool with the prompt *Type a location here (optional)*. This will allow the end user to type anything they want to limit the locations required. We want this tool to modify the regular expression in the *RegEx* tool.
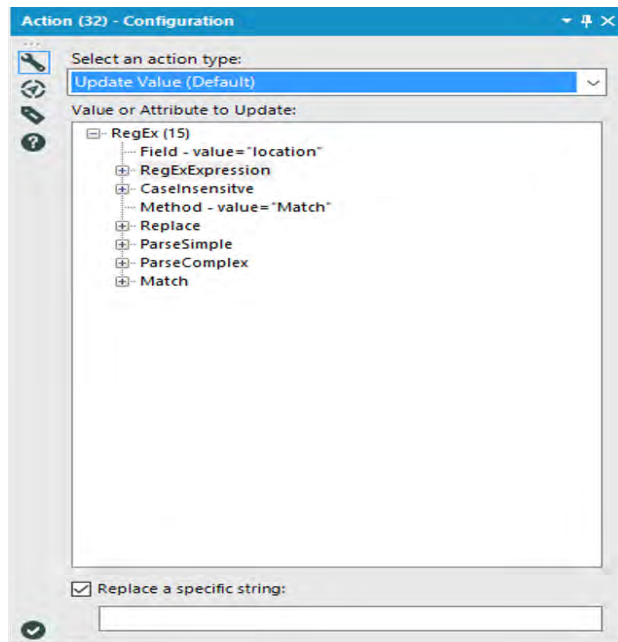
Figure 8-41 - Traffic Tickets- Using Action with RegEx to specify string

We can see the *Action* tool is giving us an error, and if we take a look, we need to specify what should be updated. In this case, we know that in the *RegEx* tool, there is the expression string *.*something.** We want to replace *something* with the value the end user types in. Thus, we can use the *Replace a specific string* option to replace the string *something* within the regular expression. Now let's see how we can modify the state filter.
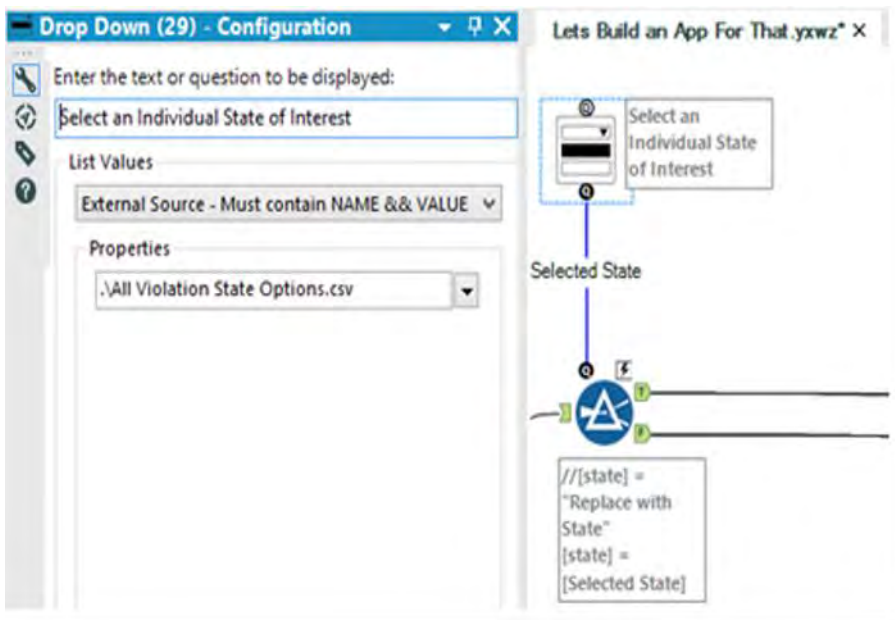
Figure 8-42- Traffic Tickets- Using Drop Down for selecting an individual State of Interest

This time, we are going to feed the answer of a question directly into a tool for use in the expression. We are going to create a *Drop Down* question with the prompt *Select an Individual State of Interest*. We are going to use the second file we received from the Baltimore Ticket Team to import the list of possible state codes from a file called *All Violation State Options.csv*. Finally, we are going to set the default value to *MD* so the team can quickly run the in-state analysis.
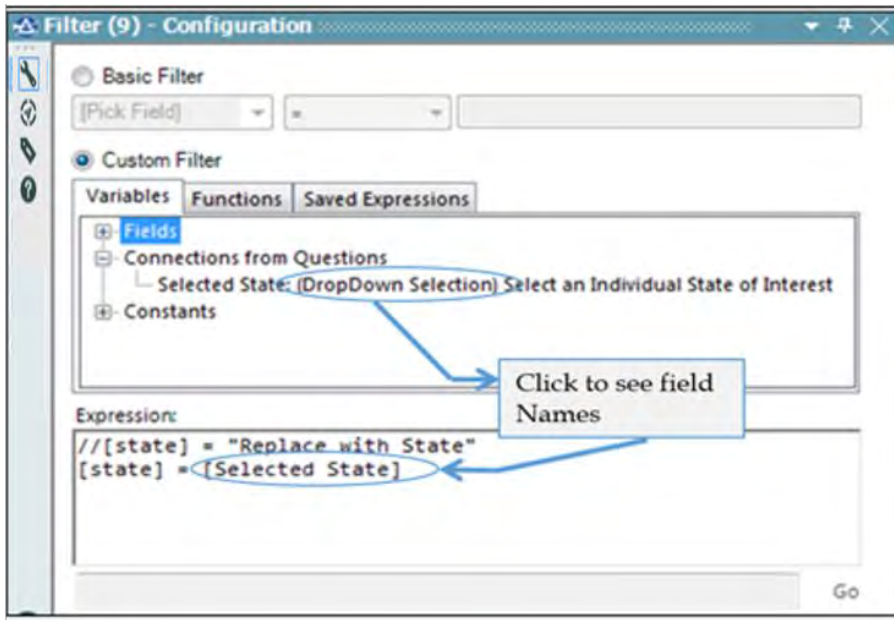
Figure 8-43- Traffic Tickets- Using Formula to filter the data

We are almost done with this question, except that we need to update the formula used to filter the data. We can replace the original formula with the one in the expression mentioned. (If we do not rename the connection, it should say [#1] instead of [Selected State].)

There is only one more question, but this one is the most complicated. This last question involves selecting which output(s) get created. Since the end user may want to output any combination of three different files (we will force them to select at least one), there are five different options we need to provide.
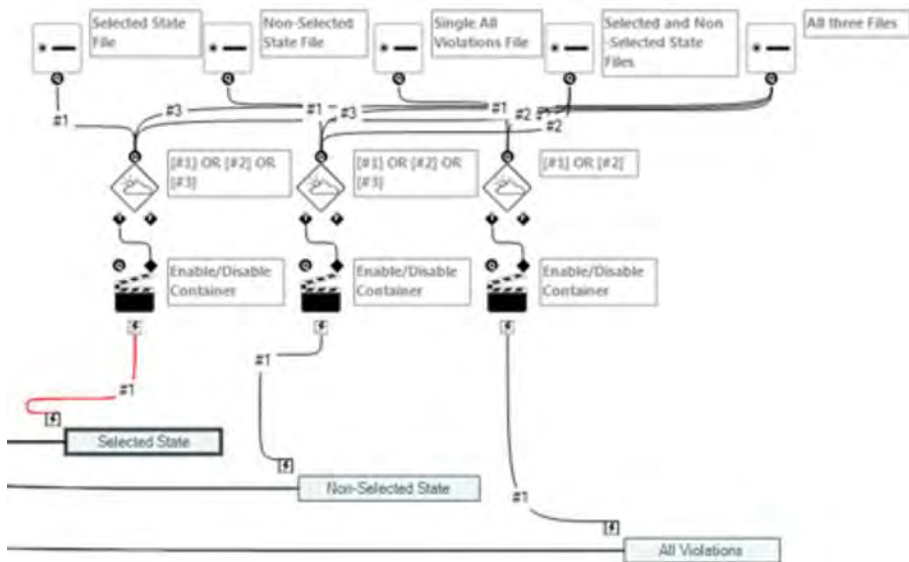
Figure 8-44- Traffic Tickets- Choosing from 5 different option

Create five Radio Button tools with the default settings other than the labels:

1. Selected State File
2. Non-Selected State File
3. All Violations File
4. Selected State and Non-Selected Files
5. All three Files

The goal here is to develop a structure where anytime we want a particular file created, we enable the *Tool Container(s)*. In order to activate the tool containers, we need *Action* tools connected to them that enable the container if the incoming connection is true.

Since multiple values can result in each file getting produced, we have complex conditions that need to be met. In

order to evaluate these conditions, we need to add a *Condition* tool before each *Action* tool.

Each of these Condition tools should be connected in the following way:

1. Condition connected to the Selected State Output
   - Selected State File (connection #1)
   - Selected and Non-Selected State Files (connection #2)
   - All three files (connection #3)
2. Condition connected to the Non-Selected State Output
   - Selected State File (connection #1)
   - Selected and Non-Selected State Files (connection #2)
   - All three files (connection #3)
3. Condition connected to the All Violations Output
   - All Violations File (connection #1)
   - All three Files (connection #2)

The expressions in each of the three conditions should be:

1. [#1] OR [#2] OR [#3]
2. [#1] OR [#2] OR [#3]
3. [#1] OR [#2]

Figure 8-45- Traffic Tickets- Interface designer

Now that we have the workflow developed, we can see the *Interface Design*. The interface is how the end users are going to interact with the workflow. As we can see from the previous page, the *Interface Designer* shows us the list of questions we created in the workflow. We could leave the questions like this, yet it provides a nicer user experience if we organize the tools.
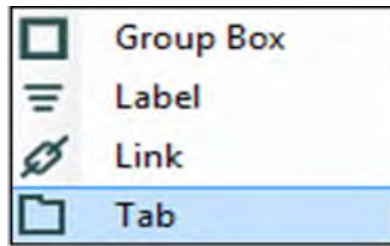
Figure 8-46- Traffic Tickets- Choosing Tab from the drop down

We can create a new tab by clicking *add drop down* and selecting *Tab*. We can then click on questions we want to move in order to have related questions together.
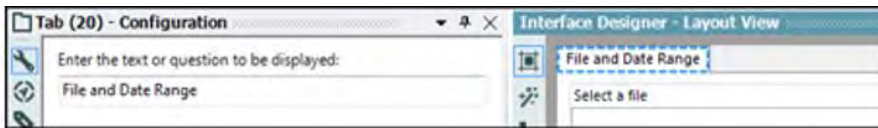
Figure 8-47- Traffic Tickets- Entering Text to be displayed

We can rename this tab by clicking on the tab at the top of the list and changing the text in the *Tab configuration window*.
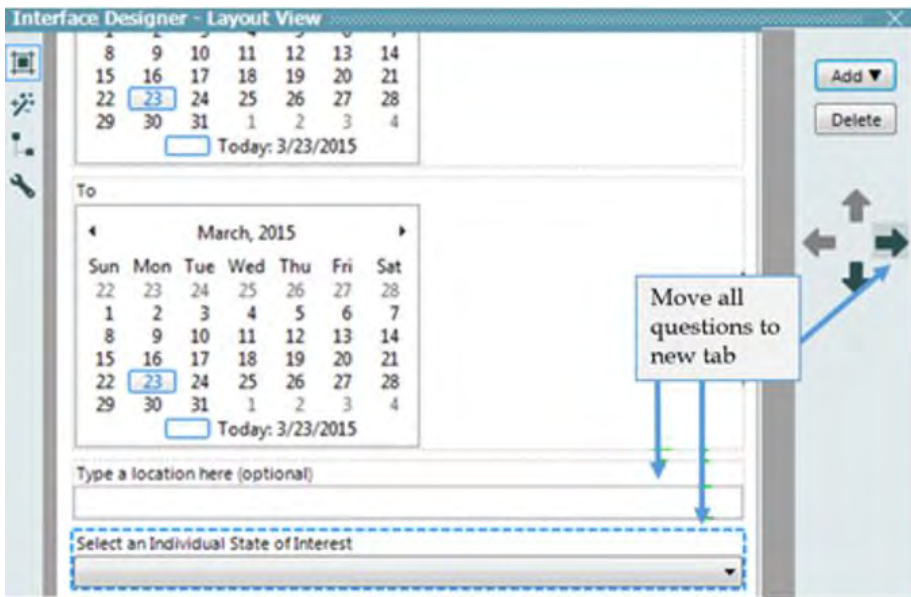
Figure 8-48- Traffic Tickets- Location information

We can then add a new tab and rename it *Location Information*. Once it is created, we can move the questions in the previous image to that tab that is about Location (of the driver's home state or violation).
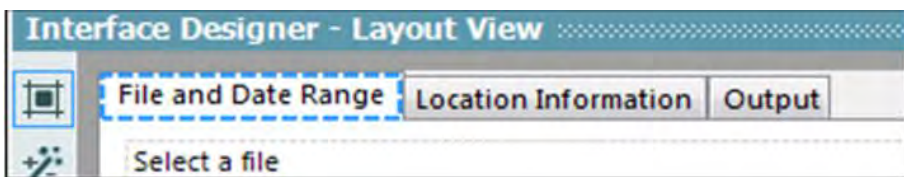
Figure 8-49- Traffic Tickets- Adding a tab called Output

Now we can create one more tab called *Output*. But before we move questions, we need to add a *Group Box* from the *Add* menu.
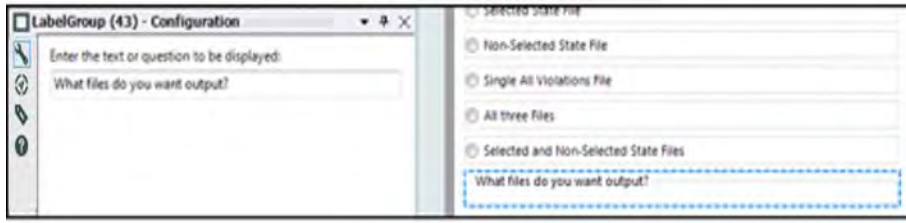
Figure 8-50- Traffic Tickets- Moving all questions into a group using LabelGroup
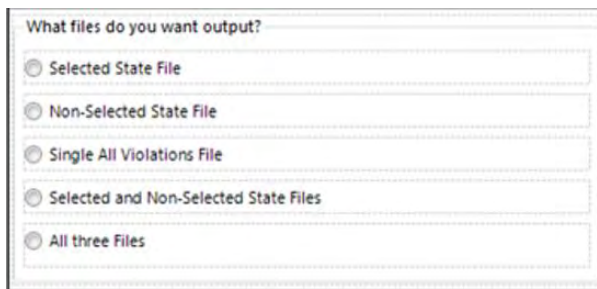
Figure 8-51- Traffic Tickets-Grouped questions moved to Output tab

Now we will move all the required questions down into the *Group Box* by selecting the question and clicking down until our list looks like the image.

Once we have all of our output questions grouped together, it is easy to move them across into the Output tab. Then we can click on the *Group Box* title and move that.

After completion, we should have the three tabs for our questions.

Figure 8-52- Traffic Tickets-All three tabs

For better representation purposes, some of the connections are made wireless. The final workflow after adding the Interface tools looks as shown in the image.
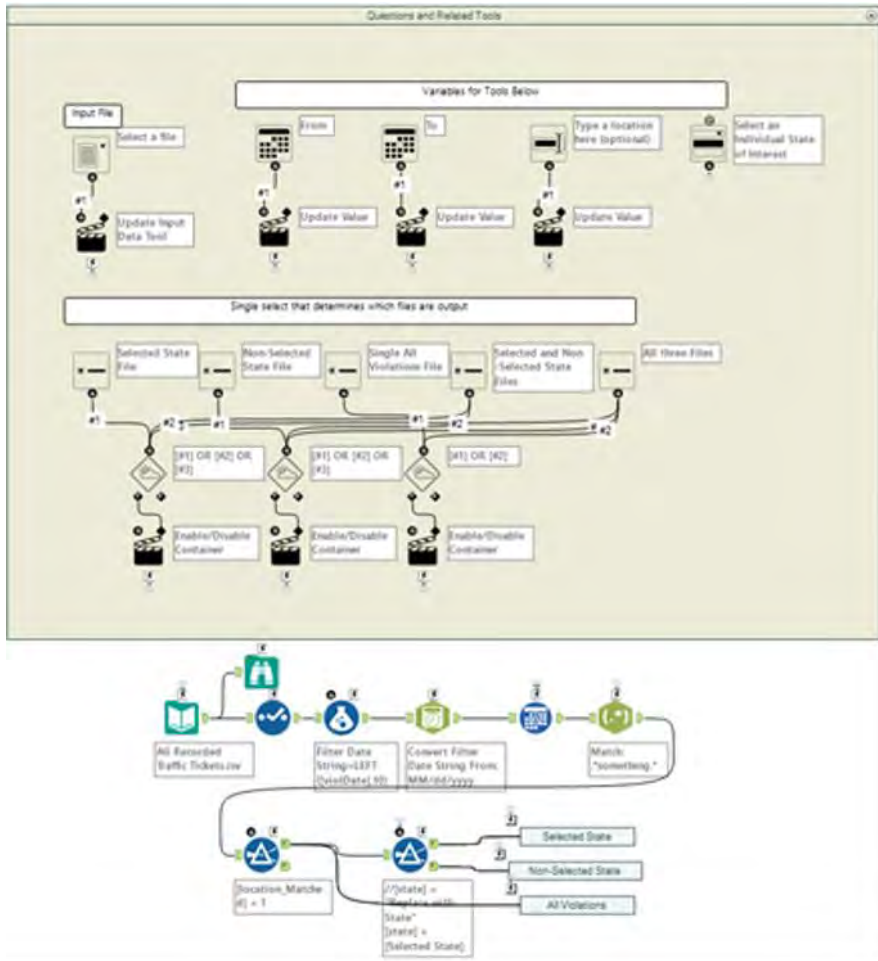
Figure 8-53- Traffic Tickets- Data stream when complete

8.16 To Summarize or Not to Summarize: That is the Question

To...	Alteryx Consultants
Subject	To Summarize, or Not to Summarize, That is the Question

Hey,

As it turns out, the Baltimore Ticket Team really likes what we put together but would like to add one more feature.

They want to keep everything that is currently there, plus add an option to summarize the output files by state to see the total count of violations.

Let me know when you've come up with a solution.

Thanks.

CHAPTER 9

Where's the Joe?

To... Alteryx Consultants

Subject Where's the Joe?

Attached Mermaid Coffee Company Locations.csv

Hey,

We were asked by The Mermaid Coffee company to create a map of all of their locations in the United States for an advertising campaign.

They have provided us a map of all of their global locations to work with.

They want to see the location as well as a set radius around all of their stores, which should be configurable from 1 mile to 10 miles, depending on the user's preferences.

They want a report to be created with a map of all U.S. locations drawn within a selected area, as well as the individual maps for each state for stores in that selected area.

This is a good time to introduce you to a concept that will save you a lot of time down the road. Many requests have similar core elements, and it saves a lot of time when you can reuse them. Thus, Alteryx allows you to make macros.

We should develop a macro that will allow you to pass map inputs as well as a numeric field for the radius before we tackle this project.

Thanks.

9.1   Tools & Concepts

In this chapter, we will be covering the following Tools and Concepts:

Tools	**Concepts**
Macro Input	Building Macros
Macro Output	Using Macros
Map	Mapping
Map Input	
Numeric Up Down	
Spatial Match	
Trade Area	
Buffer	
Distance	
Find Nearest	
Generalize	
Heat Map	
Make Grid	
Poly-Build	
Poly-Split	
Smooth	
Spatial Info	
Spatial Process	

9.2 Macro Input

	The *Macro Input* tool allows for the creation of an incoming data connection for a macro.		
Figure 9-1 Macro Input	**Group**	**Input**	**Output**
	Interface	See below	Any data stream

Note: The *Macro Input* tool receives data from the calling workflow.

The bottom *Black Question Anchor* can be connected to a question anchor on top of a tool to pass the result to the subsequent tool.

Input: The sample data connection or the calling module's data stream.

Properties Window:

The *Macro Input Configuration* window has four basic components.
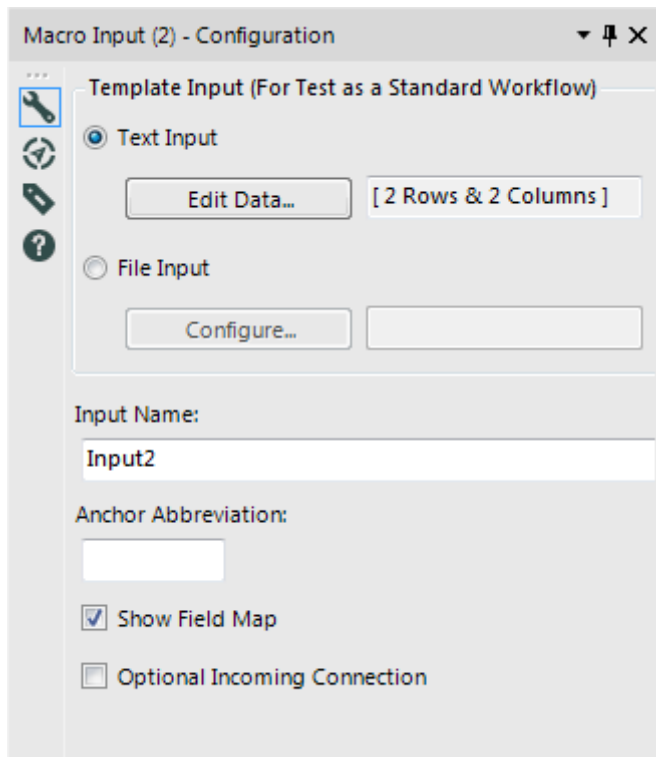
Figure 9-2 Macro Input Configuration

- *Template Input (For Test as a Standard Workflow)* is the data we will use to build the macro's data stream. The fields here need to match the incoming connections. We can use either a text input or a file input for the source.
- *Input name* is the name of the data connection.
- *Anchor Abbreviation* allows us to type a single character to be displayed on the macro's input connection.

Typically, this is used to identify which data stream should be connected to which connector if we have multiple inputs.

- *Show Field Map* allows us to manually match the fields to be used in the data stream if the names do not match the original data source.

9.3 Macro Output

	The *Macro Output* allows us to create an outgoing data connector for a macro.		
	Group	**Input**	**Output**
Figure 9-3 Macro Output	Interface	Any data stream	See below

Note: The *Macro Output* tool returns data back to the calling workflow.

Output: The incoming data stream into the workflow that called this macro.

Properties Window:

The *Macro Output Configuration* window has two components as shown in the following figure.
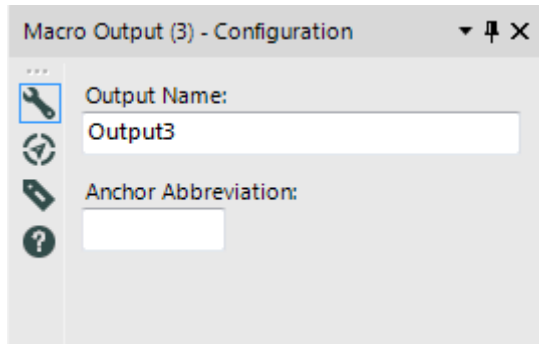
Figure 9-4 Macro Output Configuration

- *Output Name* allows us to rename the output data connector.
- *Anchor Abbreviation* allows us to type a single character to be displayed on the macro's output connection. Typically, this is used to identify which data will be coming out of the connector if we have multiple outputs.

9.4 Map

|

Figure 9-5 Map	The *Map* tool allows us to create a spatial object(s) on a map for use in the data stream.		
	Group	**Input**	**Output**
	Interface	None	None

Note: The *Map* tool creates spatial objects to be used with a Map Input tool.

Application questions can be connected to the top *White Question Anchor* to update other questions.

The bottom *Black Question Anchor* can be connected to a question anchor on top of a tool to pass the result to the subsequent tool.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to change the spatial objects created.

Properties Window:

The *Map Configuration* window has six basic components.

Figure 9-6 Map Configuration

Enter the text or question to be displayed allows us to write the prompt the end user sees.

- *Base Map* allows us to select the map the end user will be able to use in order to understand where they are clicking.
- *Select/Draw* allows us to decide if the end user is allowed to create single or multiple spatial objects.
- *Zoom To* allows us to decide what area of the map should be shown by default.
- *Reference Layer* allows us to use a spatial file in order to feed spatial information into the map. We would want to do this so end users can select created objects or use those objects for reference points when they create their own. If we choose to add a reference layer, we can feed it in from a *File Browse* question or set a specific file to always use. When we are using either of these modes a Label Field must be selected (if we use the *File Browse* option setting the label field is done in the interface).
- *Mode* allows one of two input methods. *Draw* allows the end user to create objects on their own, while *Select* allows them to choose objects in the reference layer. When we are in *draw* mode, we have the ability to allow the end user to create points, lines or polygons and decide if the user can create custom labels for the objects.

9.5   Map Input

	The *Map Input* allows us to use the map as a source to create spatial objects.		
Figure 9-7 Map Input	**Group**	**Input**	**Output**
	In/Out	None	See below

Note: This tool creates spatial objects.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream with spatial objects and a label field.

<u>Properties Window:</u>

The *Map Input Configuration* window has four basic components.
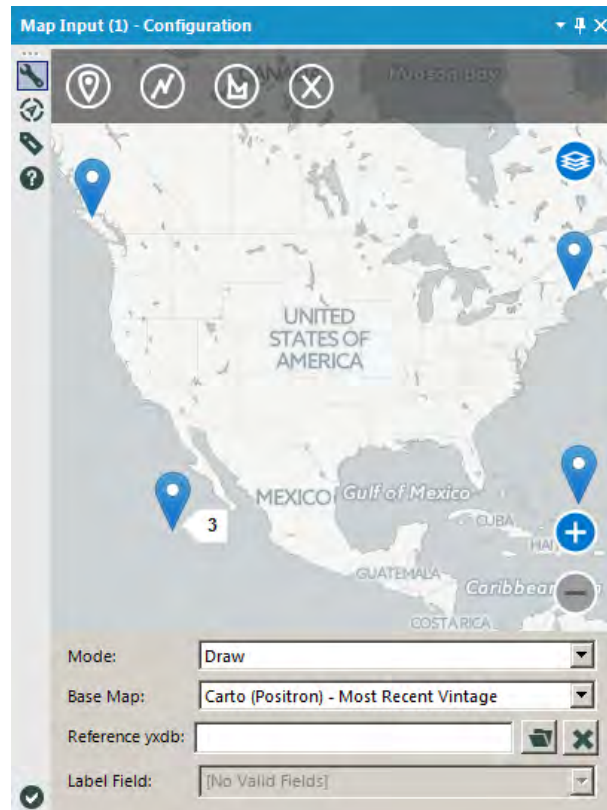
Figure 9-8 Map Input Configuration

- *Mode* allows us to select between *Draw*, where we create the objects, and *Select*, where we can use objects from a file.
- *Base Map* is the background map we are using to give a reference.
- *Reference yxdb* allows us to select an Alteryx database file with spatial objects defined so we can reuse previously created map objects.

- *Label Field* allows us to define which field in the Alteryx database file has the spatial object label names.

The map allows us to create spatial objects. The icons labeled 1 – 6 are:

1. create point objects

2. create line objects

3. create polygon objects

4. clear all created objects

5. show the base map or objects

6. zoom controls

9.6   Numeric Up Down

	The *Numeric Up Down* question tool allows us to input a numeric value to be used in the data stream.

Figure 9-9 Numeric Up Down

Group	Input	Output
Interface	None	None

The bottom *Black Question Anchor* can be connected to a question anchor on top of another tool to pass the result to the subsequent tool.

Properties Window:

The *Numeric Up Down Configuration* window has six
components.

Figure 9-10 Numeric Up Down Configuration

- *Enter the text or question to be displayed* allows us to write
  the prompt the end user sees.
- *Minimum* allows us to set a lower threshold for the input.
- *Maximum* allows us to set an upper threshold for the input.
- *Increment* allows us to define the most granular level of
  change.
- *Default* is the value that will auto-populate.
- *Decimal Places* is the number of decimal places that are
  displayed.

9.7   Spatial Match

	The *Spatial Match* allows us to use one spatial field to filter the data from another.		
Figure 9-11 Spatial Match	**Group**	**Input**	**Output**
	Spatial	Input T & Input F below	Output M & Output U below

Note: One of the two inputs need to use a polygon. When we have multiple target spatial objects that match multiple universe spatial objects, the *Spatial Match* tool will replicate data in the output.

Input T: Any data stream with a spatial object field.

Input U: Any data stream with a spatial object field (optional depending on settings).

Output M: The target (T) input data stream limited by the records that matched in universe (U) input data stream.

Output U: The target (T) input data stream limited by the records that did not match (unmatched records) in universe (U) input data stream.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Spatial Match Configuration* window has five components.
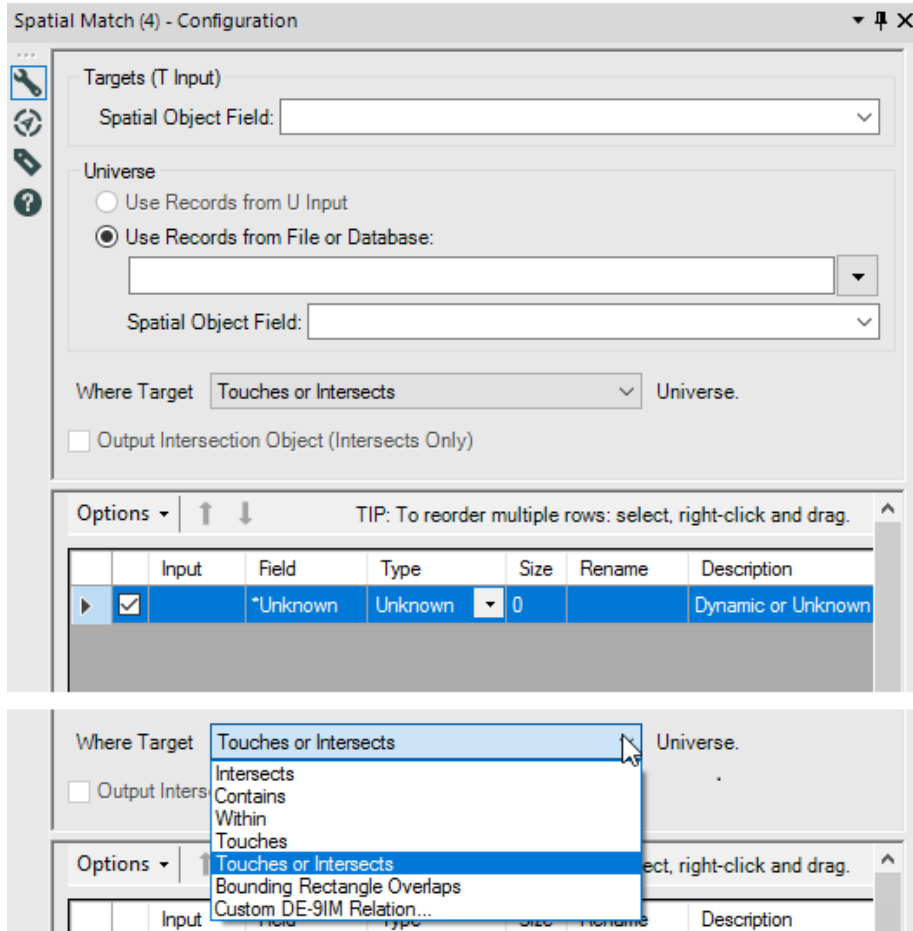
Figure 9-12 - Spatial Match Tool Configuration

- *Targets (T Input)* allows us to select a spatial object field from the target input.
- *Universe* allows us to define if we are going to use a field in the universe input or connect to a spatial data file and then select the field.

- *Where Target* _____ *Universe* allows us to define the relationship that defines a match between the target and universe spatial fields, while the check box will include the intersection as an additional spatial object field.

The types of interaction are:

- *Intersects* means they share the same space in at least a partial manner.
- *Contains* means the entire universe spatial object is in the target spatial object.
- *Within* means the entire target spatial object is in the universe spatial object.
- *Touches* means the two spatial objects share a border but do not intersect.
- *Touches or Intersects* means the two spatial objects share a border or intersect.
- *Bounding Rectangle Overlaps* means the fields are matched if the smallest rectangle to contain the entire target spatial object intersects the smallest rectangle to contain the entire universe spatial object.
- *Custom DE-91M Relation* allows us to define customized relationships between the objects.
- The bottom of the tool has an embedded *Select* tool.

9.8   Trade Area

Figure 9-13 Trade Area	The *Trade Area* tool draws a polygon centered on a point object that is provided.		
	Group	**Input**	**Output**
	Spatial	Any data stream with a point spatial field	None

Note: In order to use the drive radius data functionality, we need to have the *TomTom* data package.

Lightning Bolt Anchor: An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros

Properties Window:

The *Trade Area Configuration* window has four basic components.
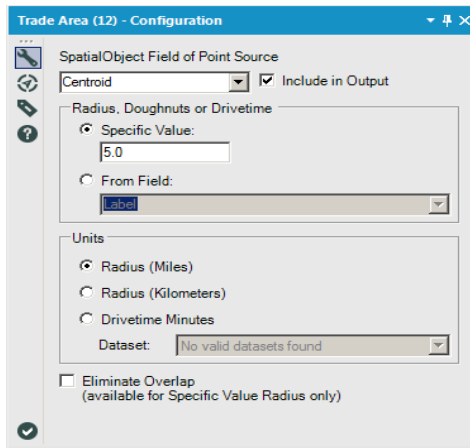
Figure 9-14 Trade Area Configuration

- *Spatial Object Field of Point Source* shows us the list of spatial object fields in the incoming data and allows us to select the one we are interested in. Checking the *Include in Output* box maintains the point field in the output.
- *Radius, Doughnuts or Drivetime* allows us to set a numeric value for the distance between the point and the edge of the polygon.
- *Units* allow us to define what the unit of measurement Specific Value represents and if it is Drivetime Minutes, which dataset to use to create the distance.
- *Eliminate Overlap* (available for Specific Value Radius only) allows us to create non-overlapping polygons.

9.9   Where's the Joe?

To...	Alteryx Consultants
Subject	Where's the Joe?
Attached	Mermaid Coffee Company Locations.csv

Hey,

We were asked by The Mermaid Coffee company to create a map of all of their locations in the United States for an advertising campaign.

They have provided us a map of all of their global locations to work with.

They want to see the location as well as a set radius around all of their stores, which should be configurable from 1 mile to 10 miles, depending on the user's preferences.

They want a report to be created with a map of all U.S. locations drawn within a selected area, as well as the individual maps for each state for stores in that selected area.

This is a good time to introduce you to a concept that will save you a lot of time down the road. Many requests have similar core elements, and it saves a lot of time when you can reuse them. Thus, Alteryx allows you to make macros.

We should develop a macro that will allow you to pass map inputs as well as a numeric field for the radius before we tackle this project.

Thanks.

Macros are tools we develop so processes we need to reuse in the same workflow or others don't need to be replicated. This is useful for multiple reasons: The first is it will save us time in creating workflows, and the second is it makes maintenance much easier by allowing us to make a change to a single macro instead of each instance in all files that use the macro.

Let's start by building a macro that is given a data stream with latitude and longitude field. *Macros* are a special type of Analytic Application, so we will be taking the same approach in designing it as we do with apps.
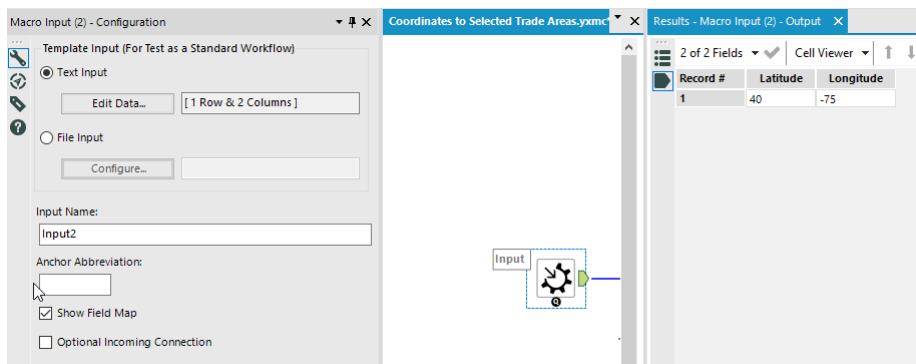
Figure 9-15 - Mermaid Coffee Macro Input for Location

We are going to build this macro by starting with a *Macro Input* tool. We are going to create a sample data using the *Text Input* option and make sure that *Show Field Map* is checked. We are setting up the data this way because we need to make sure we have latitude and longitude fields in any data that uses the tool. In the event, the incoming data doesn't have fields named *Latitude* and *Longitude* we are giving the person using this macro the ability to map the fields accordingly.
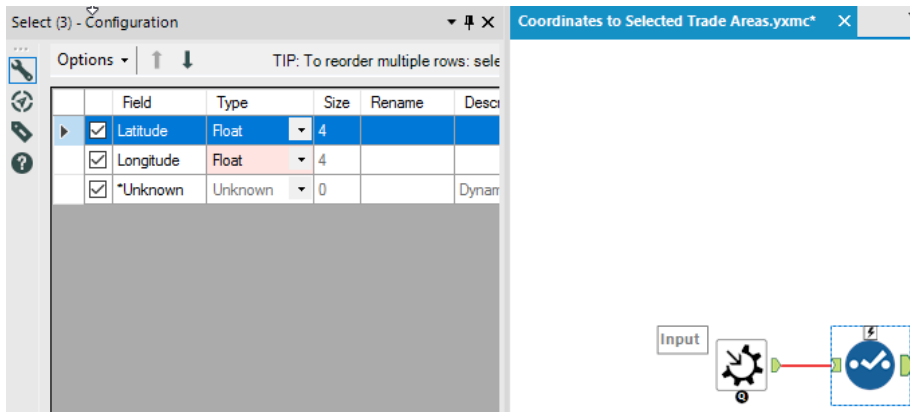
Figure 9-16 - Mermaid Coffee using Select for getting the data

Now for the first time, we pay attention to the *Unknown* option in the select tool. Because we are building a macro, we don't know what data will be coming into this tool; this option allows us to pass through any unknown fields in the data stream without modifying them.
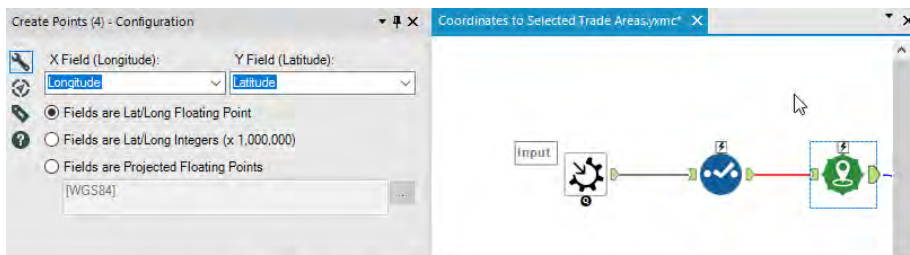
Figure 9-17 - Mermaid Coffee Creating Points using default settings

Since we have named the latitude and longitude fields in a way that the *Create Points* tool recognizes, it will auto populate with those fields in the appropriate places in the settings. We are going to use the default settings to create our points.
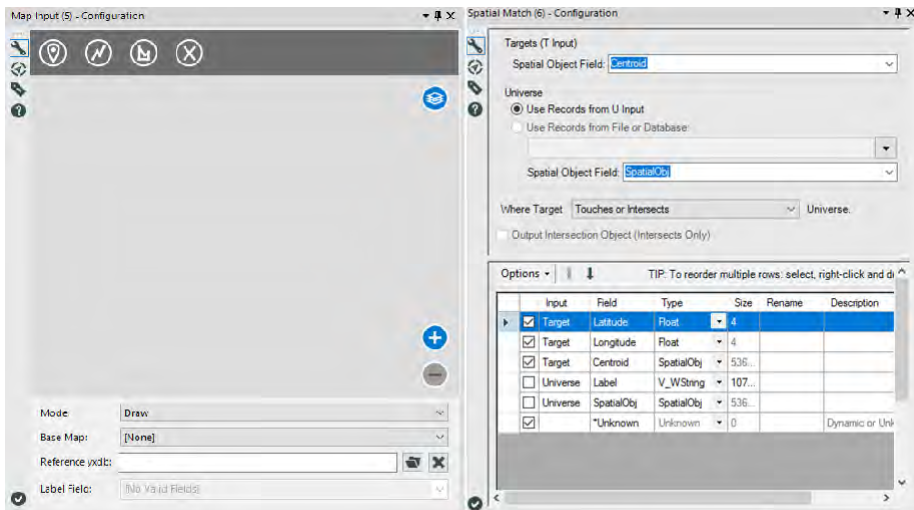
Figure 9-18 - Mermaid Coffee using Map Input and Spatial Match to filter incoming data
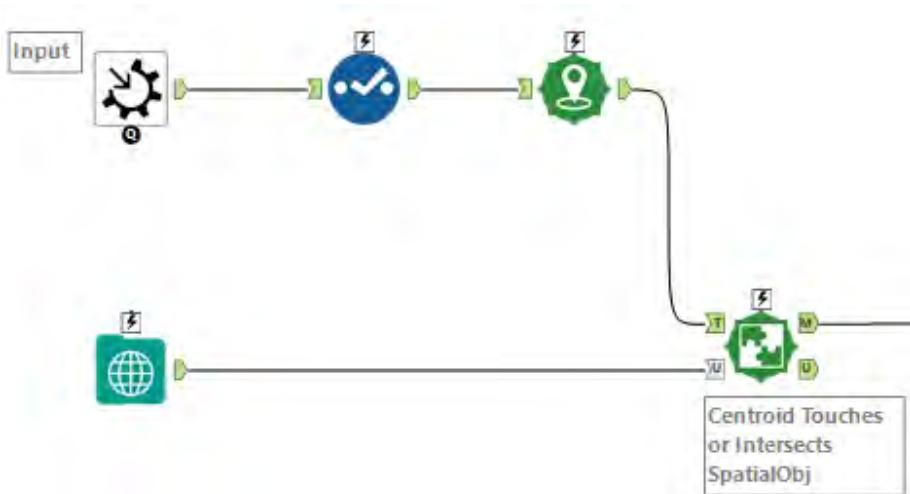
Figure 4-19 - Mermaid Coffee Incoming Data Stream

In order to limit the incoming data, we are going to use a *Map Input* tool to create a spatial object that will be passed to the *Spatial Match* tool. The *Spatial Match* tool has the ability to compare different spatial objects to determine if they share space.

We are not going to configure the *Map Input* tool because we are going to be giving it spatial objects from an interface tool called *Map* once we have finished the basic workflow.

The *Spatial Match* tool will use the configuration shown in the image to include only locations that are in/or touch the user-defined Map Input region.
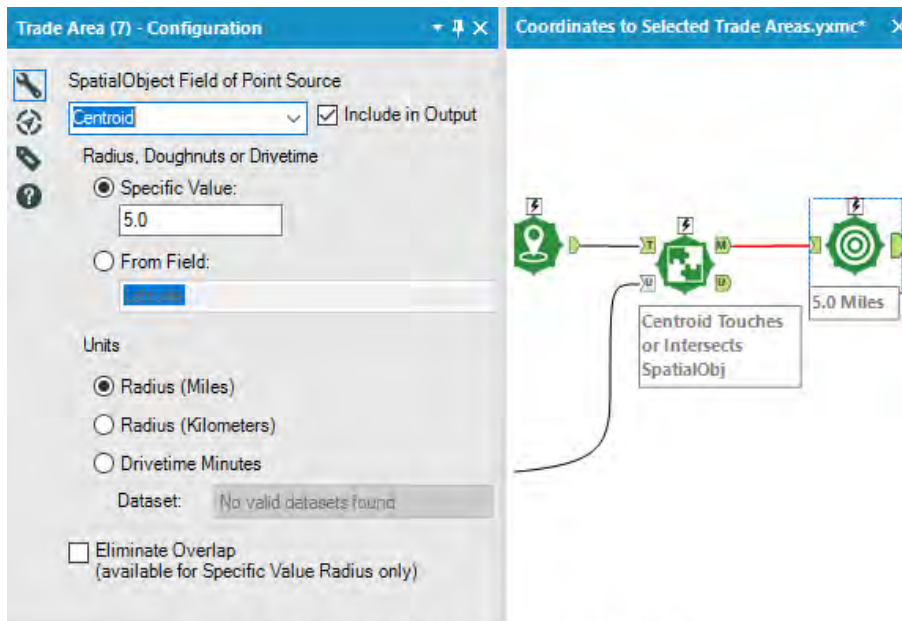
Figure 4-20 - Mermaid Coffee with Points and Limits

Now that we have limited the points to only those in the area of interest, we can create the *Trade Area* polygons for those points. We are going to make sure that the units are miles, but it doesn't matter that the *Specific Value* is 5 because we are going to use a *Numeric Up Down* to set the distance.
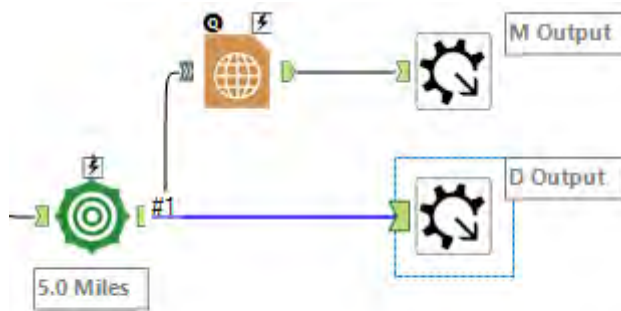
Figure 9-21 - Mermaid Coffee Trade Area Polygons

Now that we have created our trade areas, we need to create outputs.
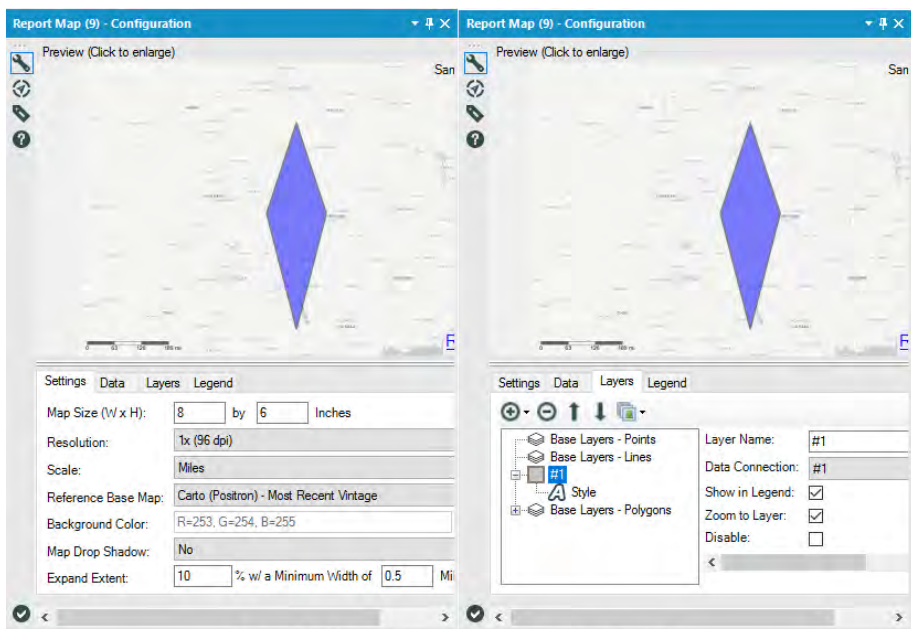
Figure 9-22 - Mermaid Coffee - Using Report Map to create output

For the map output, we need to create a map report object. Use any setting of choice; just make sure to include a base layer and the polygon layer from the data stream.
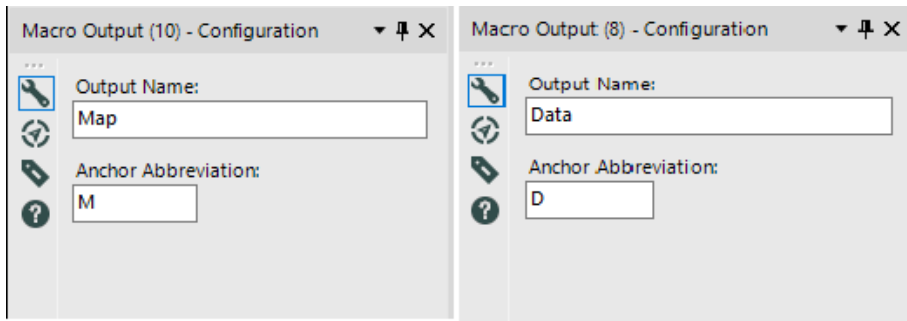
Figure 4-23 - Mermaid Coffee Macro Output

Now all that is left for the outputs is to name them and provide them with an identifying anchor character.
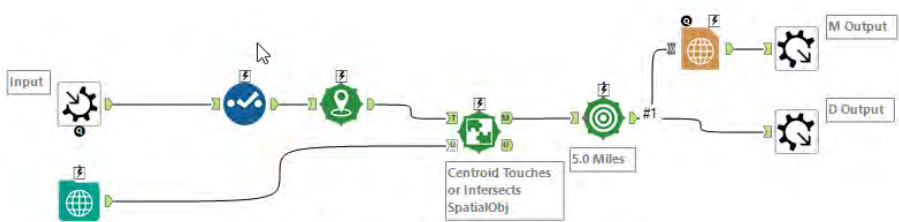
Figure 9-24 - Mermaid Coffee data stream after Macro Output

This macro now only needs the user settings so that the controls can be defined for the two tools we want to be able to change.
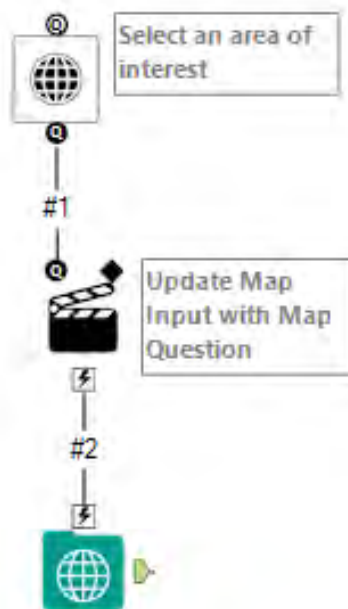
Figure 9-25 - Mermaid Coffee using Map Tool
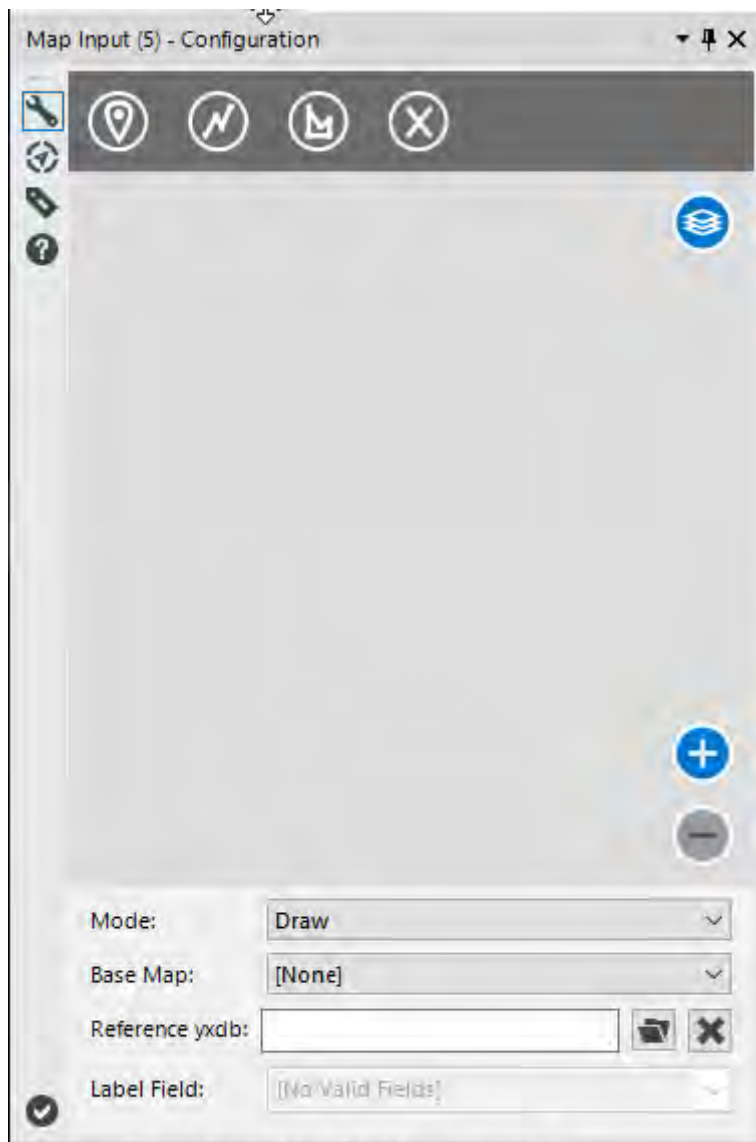
Figure 9-26 - Map Tool Configuration

We can use the *Map* tool to create a map in the interface, which allows the end user to select a specific region.

If we connect the *Map* tool directly to the *Map Input* tool, we see the *Action* tool will auto-populate with the appropriate settings.
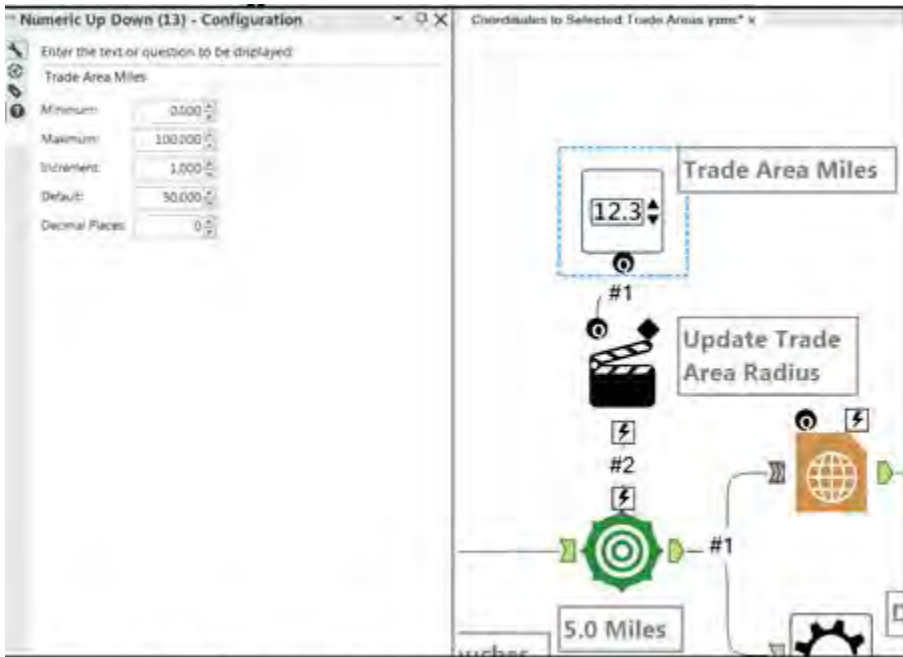
Figure 9-27 - Mermaid Coffee using Numeric Up Down to set
Number of miles a trade should cover

If we add a *Numeric Up Down* tool, we will have a numeric input so the user can set the number of miles a *Trade Area* should cover. If we connect this directly to the *Trade Area* tool, we can see the *Action* tool uses the special *Trade Area Radius* setting (so we don't need to configure the *Action* tool). We are giving a much wider set of selected radii here because we don't know how we may want this tool to be used in the future. Since we will be embedding this in another tool we can further limit the options for our current project.
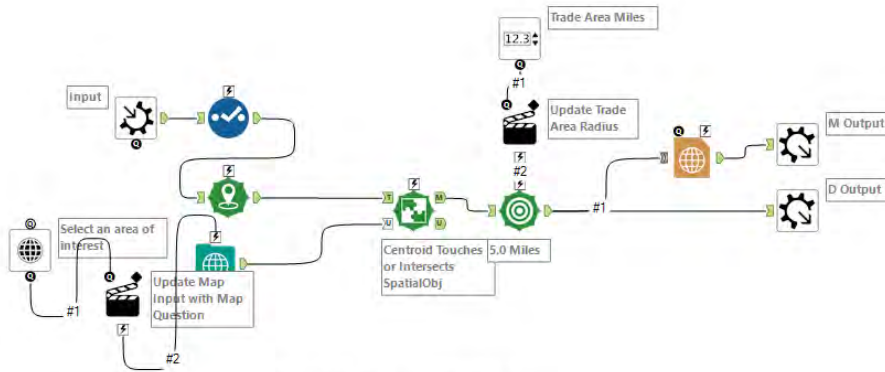
Figure 9-28 - Mermaid Coffee Data Stream

Now that we have this macro, we can save it to use in other data streams. We are going to create a macro folder in *My Documents* called *Alteryx Macros* and create a sub-folder called *Training Macros*. Let us save this macro there, and we should modify the user settings so our macro is available in the *Tool Palette*.

Navigate to the Edit User Setting window in Tools > User Settings > Edit User Settings.
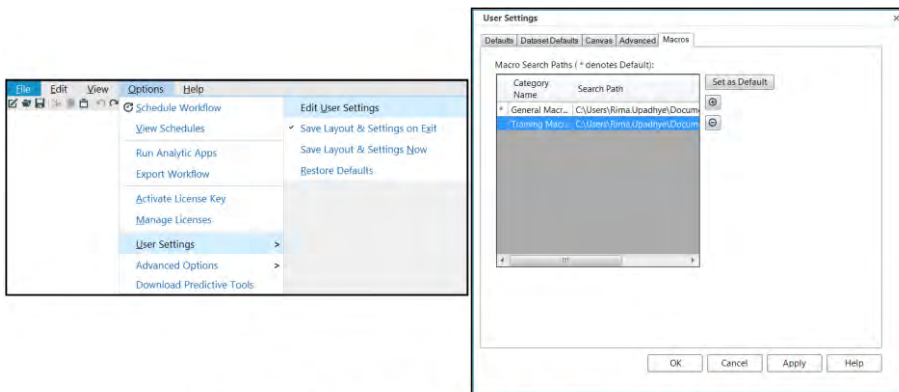
Figure 9-29 - Editing User Settings of Macro

If we go to the Macros Tab, we see the categories we defined and the folder location related to those categories. We have a *General Macros* category and a *Training Macros* category. If we click on the add symbol to the right of the window, it navigates to the *Training Macros*. Let's add it to this list with the title *Training Macros*.
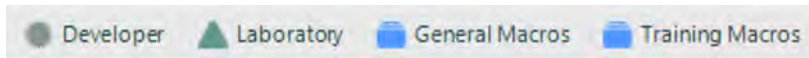
Figure 9-30 – Tool Palette

If we take a look at the *Tool Palette,* we see at the far-right side a Category called *Training Macros.* The image shown here as well as description information can be edited if we modify the properties in the workflow *Configuration window* and *Interface designer.*

Figure 9-31 – Newly saved as Macro
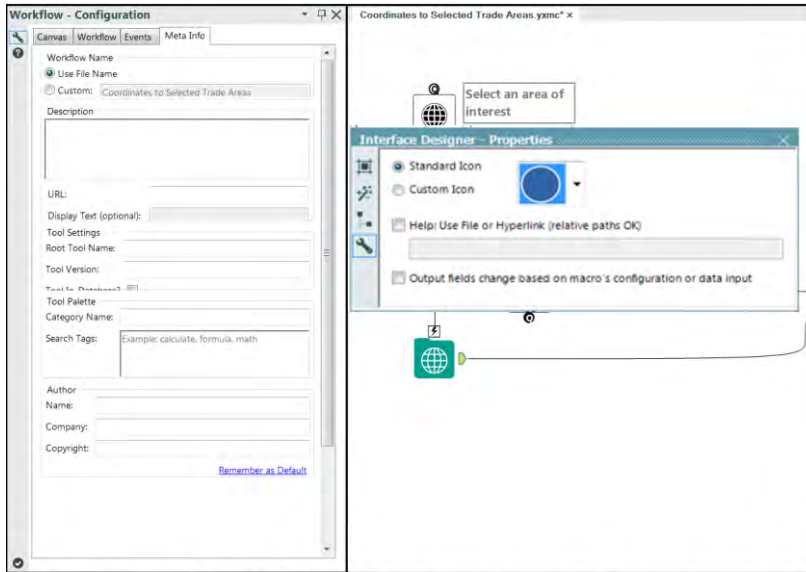
Figure 9-32 - Training Macro Workflow Configuration

Now that we have created the macro, we can build the app specific to the Mermaid Coffee Company. Open a new workflow and connect to the Mermaid Coffee Company Locations file. We can see that all fields are V_String. But we need Latitude and Longitude in a decimal format, which means we need to convert those fields to double so we will have no issue using our macro.
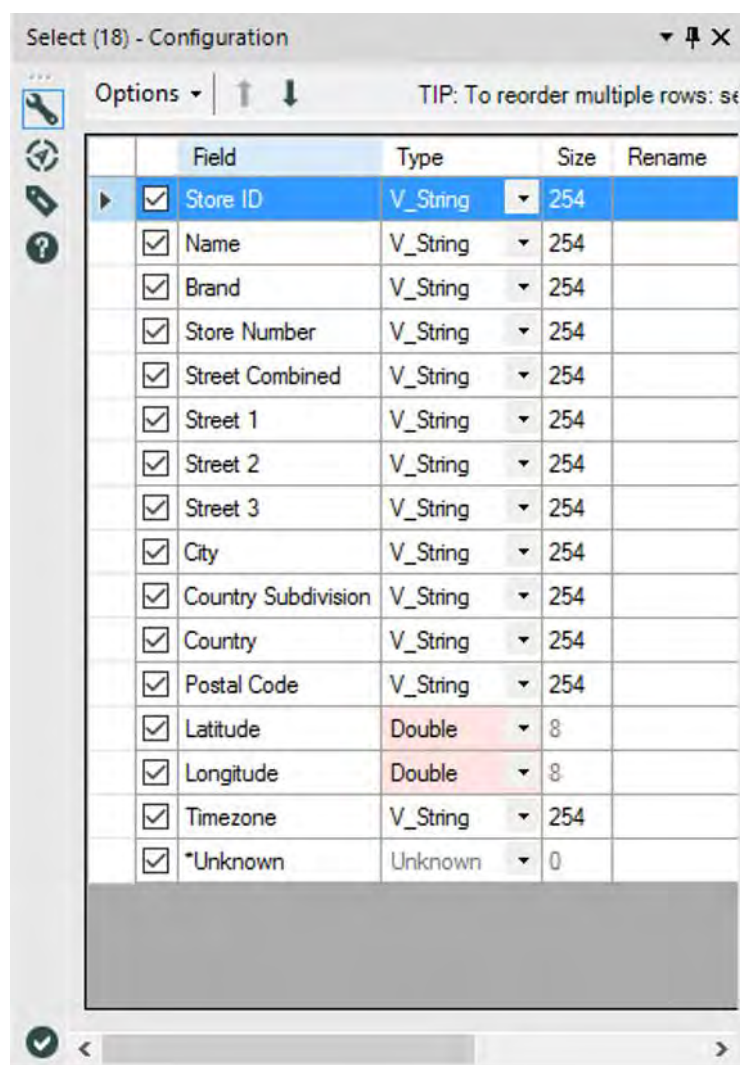
Figure 9-33 – Mermaid Coffee Select

Figure 9-34 – Mermaid Coffee Browse
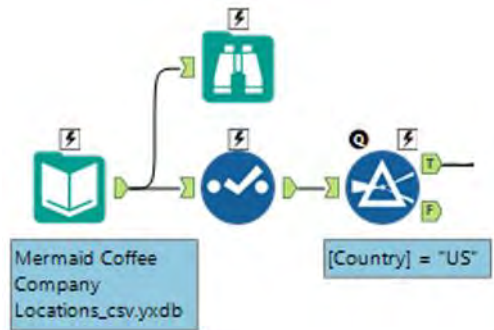
Figure 9-35 – Mermaid Coffee US Location

We can see from the email that they are only interested in US locations, so we can remove all other countries. This means the data is ready for use in our macro.
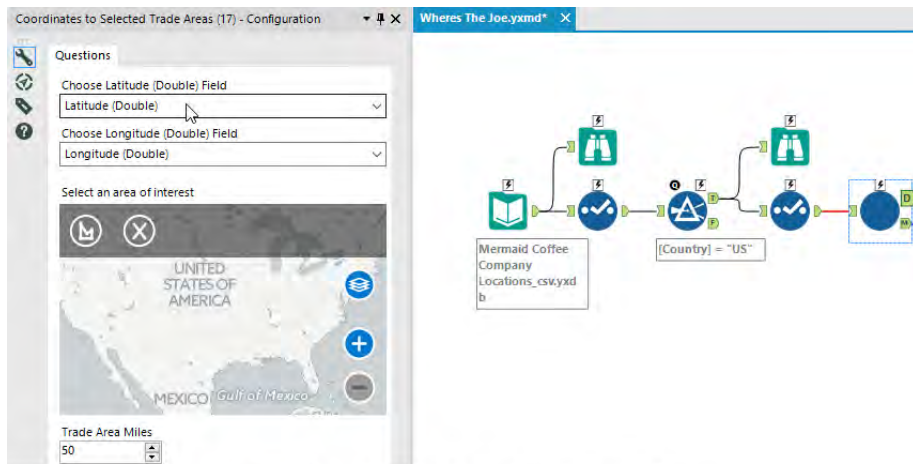
Figure 9-36 - Mermaid Coffee Trade Area configuration to restrict to US

We must make sure the Latitude and Longitude fields match correctly but leave the other *Map* and *Trade Area Miles* as defaults for now. We will connect a *Map* tool and *Numeric Up Down* here to allow the end user to update these values.
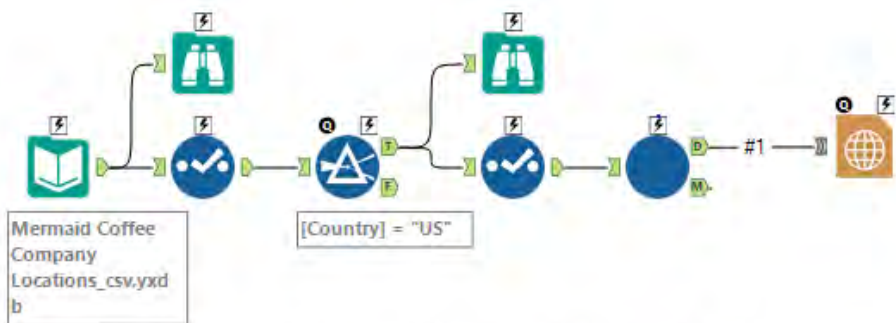
Figure 9-37 - Mermaid Coffee US Locations after setting Trade Area

Figure 9-38 - Mermaid Coffee Report Map Settings
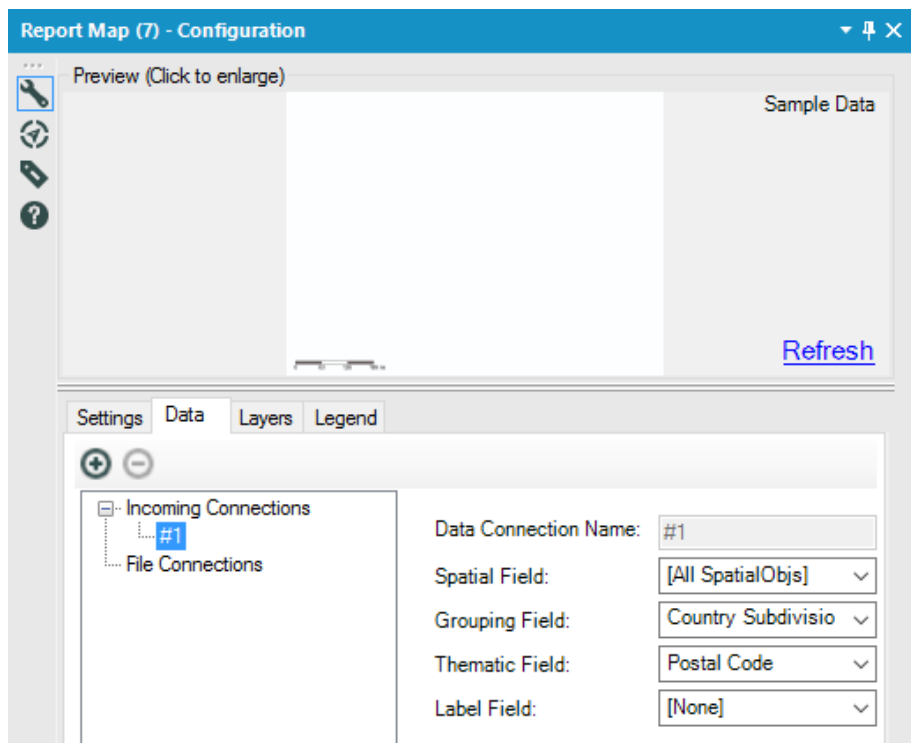
Figure 9-39 - Mermaid Coffee Report Map Data
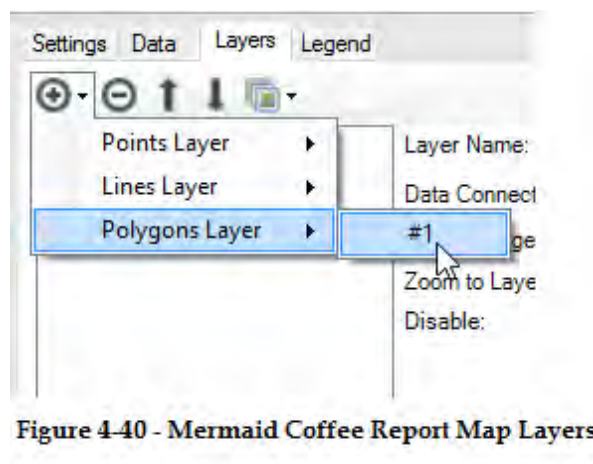
Figure 4-40 - Mermaid Coffee Report Map Layers

We know our macro created a universal map, so we can use that directly. However, we also need individual maps for each state for which we use another *Report Map* tool with the previously detailed settings to create each of the individual state maps.
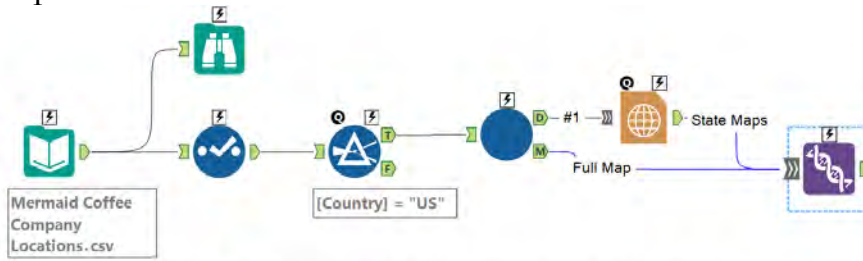
Figure 9-41 - Mermaid Coffee data stream with Maps

Now that we have both sets of maps, we can combine them. If we want the overall map to precede the others, we need to adjust the order in the *Union* tool.

We have all of our maps in a single field. We can use the *Render* tool to create our report.
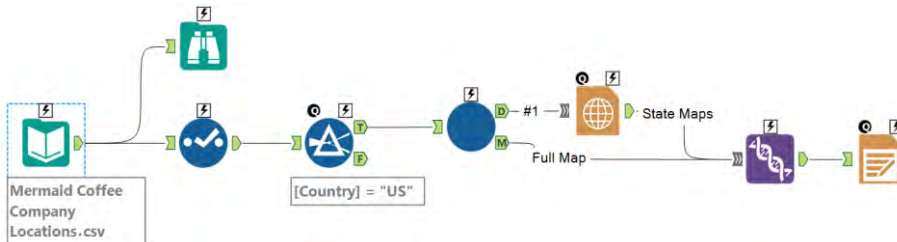
Figure 9-42 - Mermaid Coffee using Union Tool to get overall map
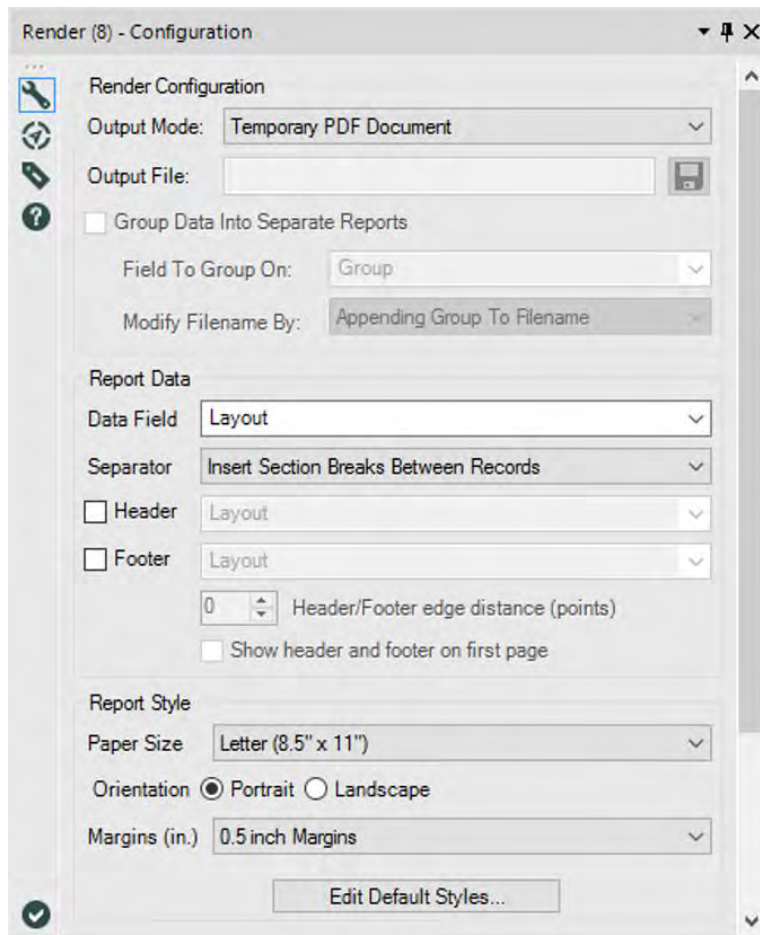
Figure 9-43 – Mermaid Coffee creating Report from Maps

Note that we are inserting section breaks between each record to make sure each map is on a page by itself.
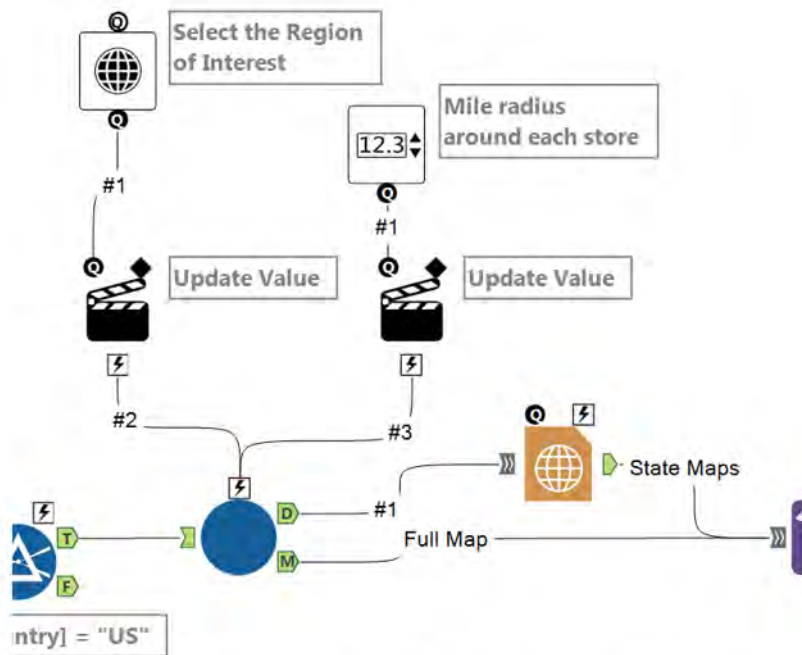
Figure 9-44 – Mermaid Coffee selection Region of interest
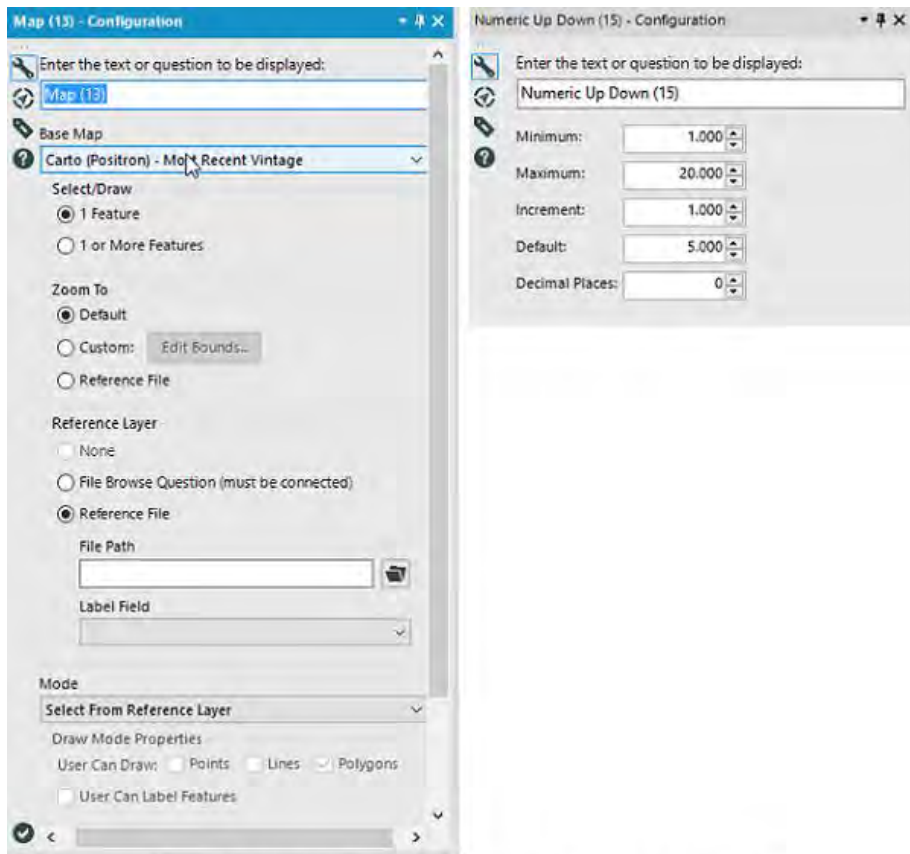
Figure 9-45 - Mermaid Coffee Map and Numeric Up Down to get desired View

Now that we have the workflow developed, we can finish off this application by adding the two desired user inputs. We will define the *Map* input the same way we did in the macro. However, the *Numeric Up Down* will have a different range of possible values and default value so it reflects what the Mermaid Coffee Company would like to see.

The Where's The Joe? data stream should look like this when it's complete.
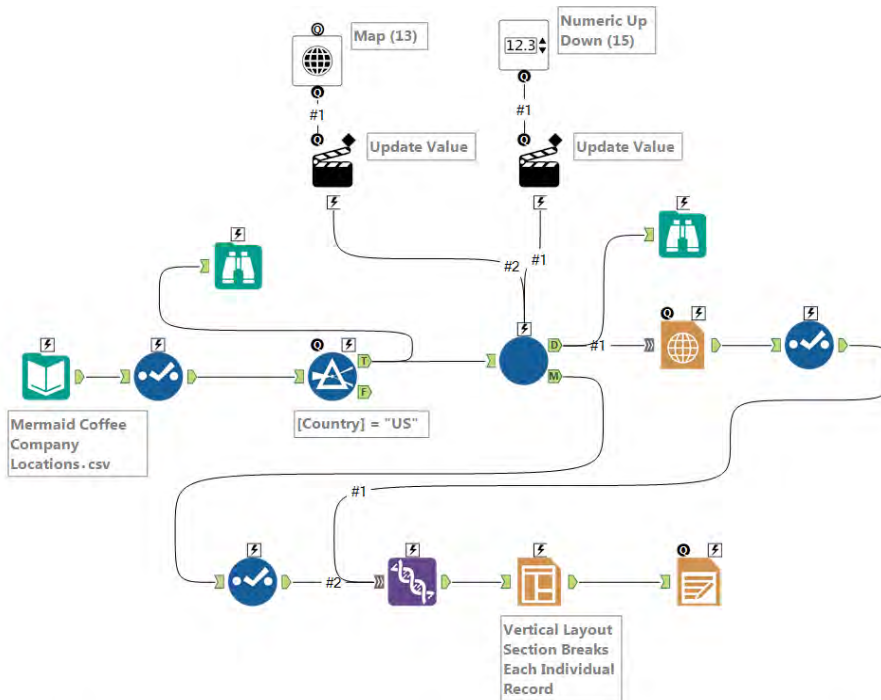
Figure 9-46 - Mermaid Coffee Final data stream when complete

9.10   Buffer Tool

	The Buffer tool will take any polygon or polyline spatial object and expand or contract its extents by the user specified value.		
Figure 9-47 – Buffer	**Group**	**Input**	**Output**
	Spatial	Any polygon or polyline spatial object	None

Special Note: The Buffer tool will take any polygon or polyline spatial object and expand or contract its extents by the user specified value.

Properties Window:

Figure 9-48 – Buffer Configuration

Spatial field: User has to specify the spatial field that contains spatial objects to be buffered.

Include in output: User can choose whether or not to include the original spatial object in the output stream. The default is unchecked so the object is not included.

Generalize to 1% of Buffer size: This is a speed optimization as it will cut down on the number of nodes in the buffered object. By default, this option will be selected and the user can check and uncheck as per requirement.

Buffer Size parameters
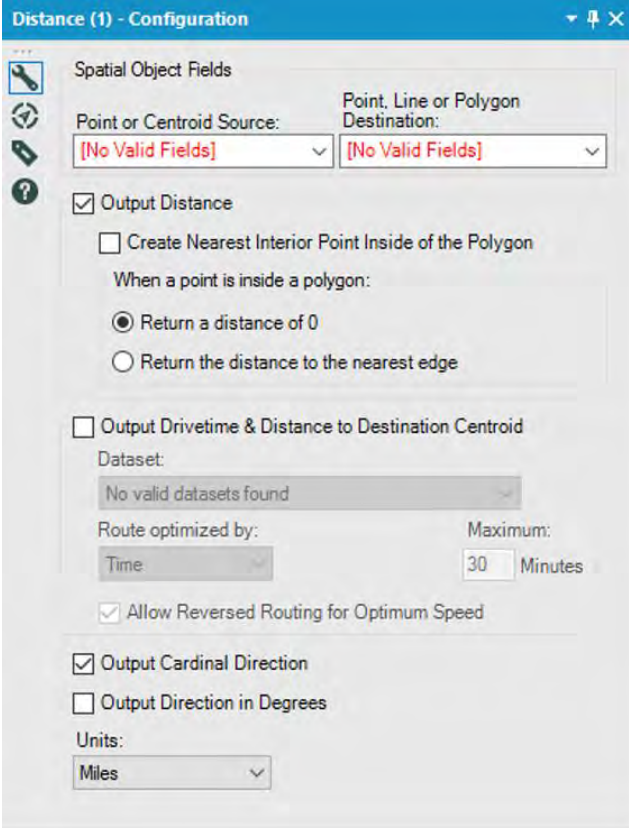
- *Specific Value:* User can input a numerical value and all records will be buffered to this size

- *From Field:* User can select a numerical field in the connection stream and each record will be buffered by the value of that field.

- *Units:* User can choose to buffer the object in units of Miles or Kilometers

9.11   Distance Tool

	The Distance tool calculates the ellipsoidal direct point-to-point, point-to-edge, or the drive distance between two sets of spatial objects.		
	Group	**Input**	**Output**

Figure 9-49 – Distance	Spatial	Spatial object	Any type of spatial object
Special Note: The Source MUST be a point or the centroid of a polygon.			

Properties Window:

Figure 9-50 – Distance Configuration

Spatial Object fields:

Point or centroid source: User has to specify the field contains points or centroids of the polygon.

Destination: In the destination, field user can specify any type of spatial object: point, line, polyline or polygon

Output Distance Checkbox: When checked, the calculated distance will be outputted in an appended data field.

Create the Nearest Interior Point Inside of the Polygon: Checking this box will add a spatial point object to the data stream. It is the distance between the specified Source and this newly created Point that is returned.

Select the appropriate radio button to specify the preferred behavior of what to do if a point is within a polygon. Choices include: Return a value of 0 or Return the distance of the nearest edge.

Output Drivetime & Distance to Destination Centroid checkbox: When checked, the drive time in minutes from source to destination will be calculated.

Choose the dataset to use in calculating the drive time and drive distance.

Select the method for your route to be optimized. Choices include:

Time: This is the default route. This method finds the route based on the least amount of time between points.

When this method is selected, both drive time and drive distance are returned.

Distance: This method ignores road speeds to find the most direct route between points. When this method is selected, drive distance is returned. Drivetime is not returned because road speeds were all set to be the same to calculate the most direct route.

Allow Reversed Routing for Optimum Speed: When checked, the engine will try to determine if you have significantly more "Source" points than "Destination" points and if so, it will reverse the two. This will result in somewhat different drivetimes (because of one-way streets) but can yield enormous performance improvements.

Output Cardinal Direction checkbox: When checked, the direction (N, NE, SW, etc.) will be outputted in an appended data field.

Output Direction in Degrees checkbox: When checked, the distance direction in degrees will be outputted in an appended data field.

Units: Click either Miles, Kilometers, Meters, or Feet to specify the unit of measure.

9.12   Find Nearest

	The Find Nearest Identifies the shortest distance between spatial objects in one file and the objects in a second file.		
Figure 9-51 – Find Nearest	**Group**	**Input**	**Output**
	Spatial	1.*Target Input* (Any spatial object field) 2.*Universe Input* (From input/Data source/spatial object)	Matched and unmatched target inputs

Special Note: Any object type can be chosen for the Target, but if it is a not a point type object, the centroid is used for analysis.

Properties Window:

Figure 9-52 – Find Nearest Configuration

The Find Nearest Point tool accepts two spatial inputs.

Target input: Specify Spatial object field.

Universe object: There are two choices:

Use Records from U Input: Select the Spatial Object field from the data going into the tool.

Use Records from Data Source: When reading in spatial objects from a data source, make sure the data source that is being brought in has already been sorted on the spatial object. Ensure no connection is going into the U input.

Spatial Object field: Select the field from the input data source to calculate the nearest distance to.

How many nearest points to find: Specify how many nearest Universe objects to find for each Target.

Maximum distance: Specify the maximum distance and units of measure these objects can be from the Target.

Ignore 0 distance matches: Choose whether or not to Ignore 0 Distance Matches. When this box is checked, a point will never match itself.

9.13   Generalize Tool

	The Generalize Tool Decreases the number of nodes that make up a polygon or polyline, making a simpler rendition of the original spatial object.
Figure 9-53 – Generalize	
	Group	Input	Output

	Spatial	Any spatial object	Simplified Spatial Object
Special Note: Polygons are generalized using the Ramer-Douglas-Peucker algorithm.			

Properties Window:
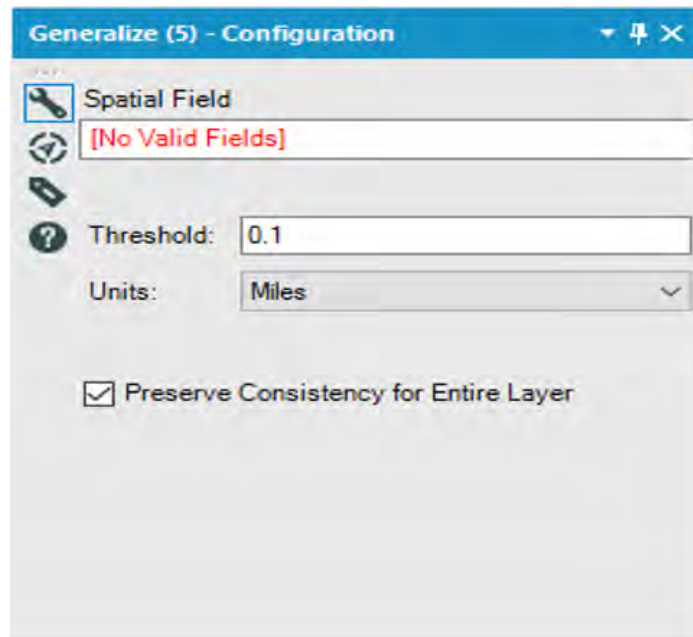
Figure 9-54 – Generalize Configuration

Spatial Field: Specify the Spatial Field that contains the spatial object to be generalized.

Threshold: Specify the Threshold, default is 0.1.

Units: Specify the Units of the threshold, default is miles.

Preserve Consistency for Entire Layer: when checked (default), line segments that are shared by multiple objects are generalized at once, ensuring no gaps or overlaps occur between those objects.

9.14 Heat Map

	The Heat Map tool divides the world into grid cells and calculates a heat level for each cell by giving the cell full credit for heat sources within it, and some smaller amount of heat from nearby cells.	
Figure 9-55 – Heat Map	**Group**	**Input**	**Output**
	Spatial	Any spatial object that contains points that represent heat source	Polygons representing different levels of heat
Special Note: The Heat Map tool divides the world into grid cells and calculates a heat level.		

Properties Window:
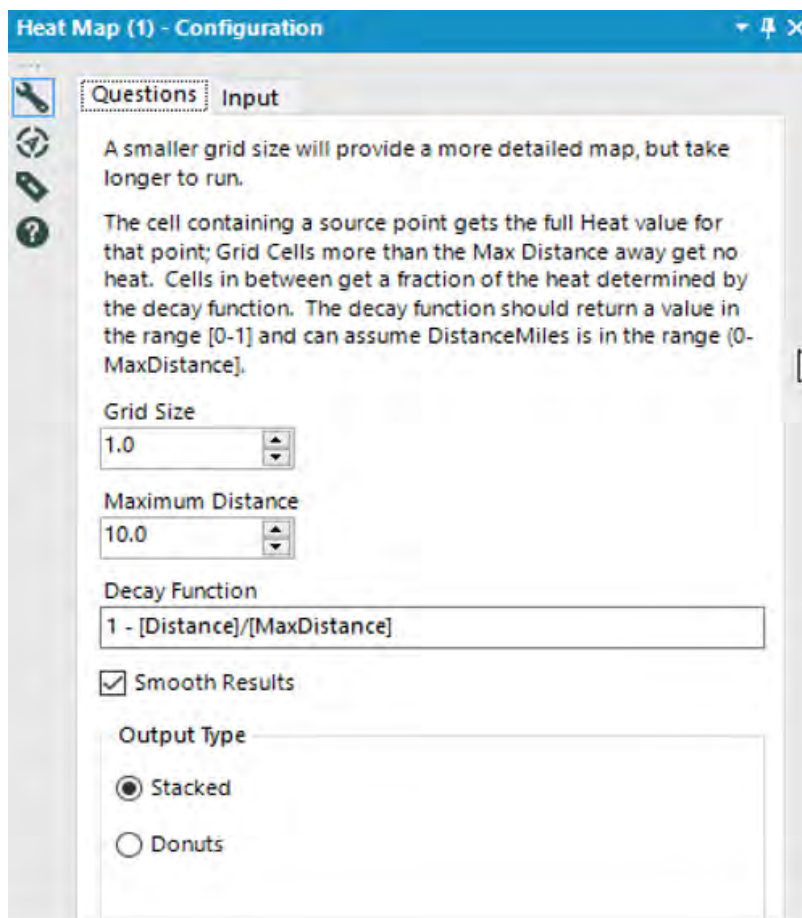
Figure 9-56 – Heat Map Configuration

There are two tabs that require user input to successfully configure the tool:

Questions - The tool is configured with 5 parameters:

Grid Size: The tool's process is based on grid cells. A smaller grid size will produce more grid cells, giving

a more detailed map as a result, but will take longer to process. Since the processing time is related to the grid size exponentially, even a small increase in size can impact the runtime considerably.

Max Distance: This is the maximum distance at which heat from one cell can contribute to the final heat level of another cell. While heat propagation generally decreases with distance, it is helpful to entirely cut it off at some point to avoid excessive processing time calculating insignificant contributions from far away cells.

Decay Function: Cells get full heat from points inside them, and no heat from points beyond the max distance, but in between, they must get some fraction based on the distance to the heat source. This is controlled by the decay function. The default (1 - [Distance]/[MaxDistance]), gives a simple linear slope.

Smooth Results: The results of the tool consist of aggregated grid cells, which can look quite jagged. Selecting this will cause a series of smooth/generalize operations designed to straighten edges and round corners.

*Output Type***:** The tool produces polygons representing different levels of heat. It can be helpful to have these as either non-overlapping "Donuts" each containing a single heat level or as "Stacked" polygons each containing their own heat level, and all higher levels.
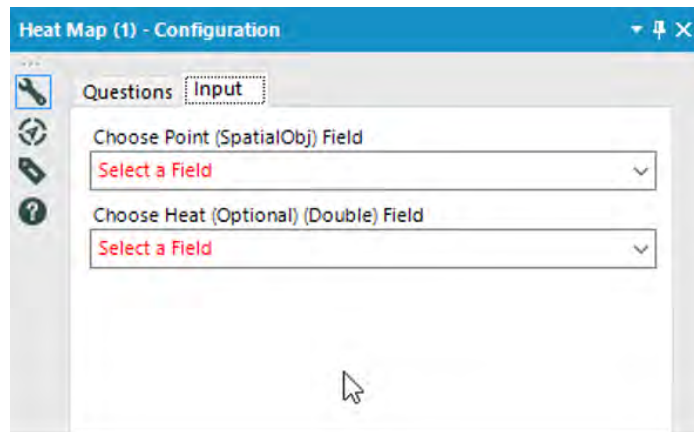
Figure 9-57 – Heat Map Configuration(Inputs)

Input - Two input fields are considered:

> *Choose Point Field:* A spatial object field containing points the represent the heat sources.
>
> *Choose Point Field:* An optional numeric field specifying how "hot" this particular point is. If the points represent customers, this might be a revenue figure, for example. If this is omitted, all points will be considered to generate equal heat.

9.15 Make Grid

 Figure 9-58 – Make Grid	The Make Grid tool takes a specified spatial object (point or polygon) and creates a grid based on the spatial object. The resulting grid is either a single grid, bound to the extent of the input spatial objects, or individual grids that dissect each input polygon.

Group	Input	Output
Spatial	Spatial object (point or polygon)	Grid based on the spatial object

Special Note: The Make Grid tool takes a specified spatial object (point or polygon) and creates a grid based on the spatial object.

Properties Window:
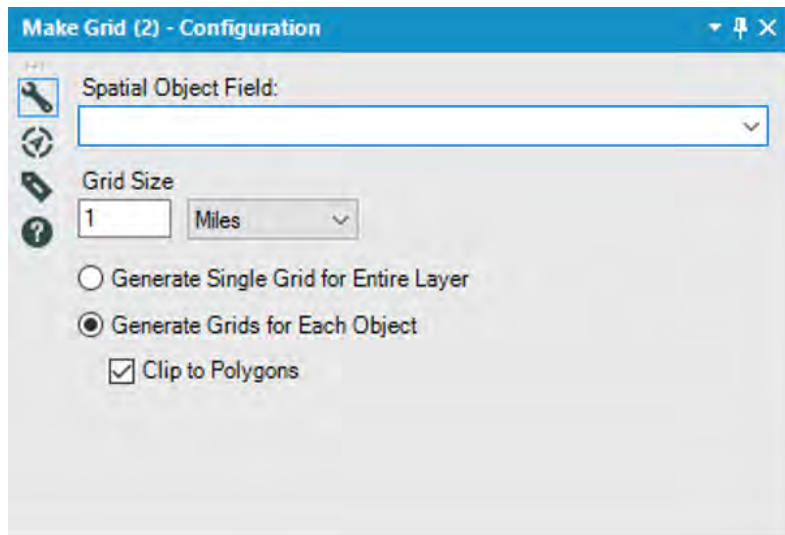
Figure 9-59 – Make Grid Configuration

Spatial Object Field: Choose the spatial object field from the input table to build a grid from. This field can contain either point type or polygon objects.

Grid Size: Specify the size and the units of the grid

Select the type of grid to make:

Generate Single Grid for Entire Layer: A single grid is created to the extent of the spatial objects (green polygons) in one layer. You could also use a point object for this type of grid.

Generate Grids for Each Object: A grid is created for each spatial object (blue polygons) in the layer and covers the extent of that object. You can only create this type of grid using polygon spatial objects

Clip to Polygons: By checking the "Clip to Polygons" box, the grid is cut to the boundary of the polygon used to create the grid.

9.16 Poly Build Tool

	The Poly-Build tool takes a group of spatial point objects and draws a polygon or polyline in a specific sort order to represent that group of points.		
  Figure 9-60 – Poly-Build	**Group**	**Input**	**Output**
	Spatial	Spatial object	Polygon or Polyline

Special Note: This tool can also be used for spatial layer development. A typical example where you would use this tool is if you had a collection of GPS data that you needed to translate into polygon or polyline objects. If you were mapping out a river, road, pipeline, reservoir or property boundary, for instance, this tool would be quite useful.

Properties Window:
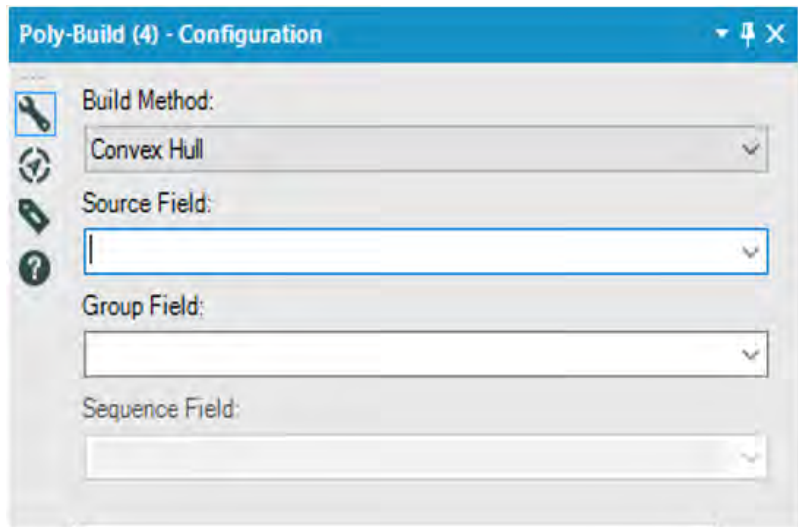
Figure 9-61 – Poly-Build Configuration

Build Method: Choices Include

> **Convex Hull:** The Convex Hull polygon is the smallest convex polygon that can be drawn to include a set of points. The polygon will not contain any concave angles, so it will never turn in on itself.

> **Sequence Polygon:** The polygon will be drawn based on all the points within their sequential order.

> **Sequence Polyline:** The polyline will be drawn based on all the points within their sequential order.

Source Field: Specify the field contains the points that tool will use to create the polygon object.

Group Field: The group field should be the field that represents the poly object, so all the records that fall into the group will make up one poly object.

Sequence Field: The sequence field is what the tool uses to draw the poly object. It sorts the records based on the sequence field and uses that order to draw the poly object.

9.17 Poly Split Tool

	The Poly-Split tool splits polygon or polyline objects into their component point, line, or region objects.		
 Figure 9-62 – Poly-Split	**Group**	**Input**	**Output**
	Spatial	Spatial object	Split Spatial object
Special Note: This is a very specialized tool used for spatial layer development. A typical use of this tool would be to disaggregate complex regions that may contain more than one polygon or to separate a polyline into its individual nodes			

Properties Window:
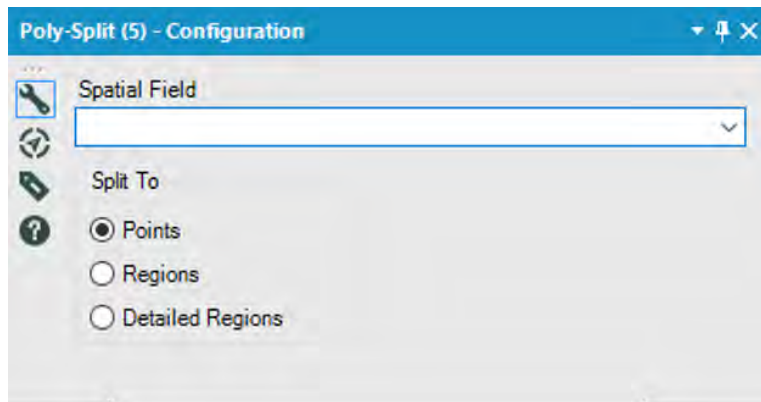
Figure 9-63 – Poly-Split Configuration

Spatial field: Select object to split from the drop-down list.

Split to: Choose from the radio buttons which type of object the split will result in. Choices are:

> **Points**: Splits regions into the points or nodes that make up that region. Each point becomes a record in the output.
>
> **Regions**: Splits regions into its individual component polygons. Each individual polygon becomes a record in the output.
>
> **Detailed Regions:** Splits regions into its individual component polygons and any holes in a polygon are also output as polygons. An additional field, split_IsHole will also be output. See below.

9.18 Smooth Tool

Figure 9-64 – Smooth	The Smooth tool is used to round off the sharp angles of polygon or polyline objects by adding nodes along the lines that make up the object.		
	Group	**Input**	**Output**
	Spatial	Spatial object	Smoothed Spatial object
Special Note: The Smooth tool uses a "B-spline" smoothing algorithm to round off the sharp angles of polygon or polyline objects.			

Properties Window:
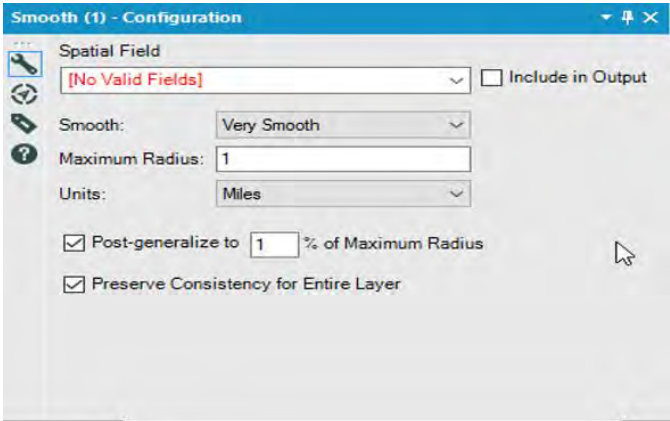
Figure 9-65 – Smooth Configuration

Spatial Field: Specify the field that contains the spatial object to be smoothed.

Include in output: User can choose whether or not to include the original spatial object in the output stream. The default is unchecked so the object is not included.

Smooth: Specify the amount of Smoothing desired. This setting controls the number of nodes that will be added to each curve. Choices include:

> Smooth: The number of nodes of the original object contained will be multiplied by 2.

> **Very Smooth:** The number of nodes of the original object contained will be multiplied by 4.

> **Super Smooth:** The number of nodes of the original object contained will be multiplied by 8.

Maximum Radius: Allow you to specify the maximum radius for smooth. The default is 1.

Units: Specify the units of the maximum radius. Choices are miles or kilometers. The default is Miles.

Post Generalize to N % of Maximum Radius: when checked (default), the smoothed object will be generalized to eliminate redundant nodes and cut down on file size.

Preserve Consistency for Entire Layer: when checked (default), line segments that are shared by multiple objects are smoothed at once, ensuring no gaps or overlaps occur between those objects.

9.19 Spatial Info Tool

 Figure 9-66 – Spatial Info	The Spatial Info tool extracts tabular information about the spatial object. Attributes such as area, spatial object, number of parts, number of points, and centroid Latitude/Longitude coordinates can be revealed.
	Group / **Input** / **Output** below

Group	Input	Output
Spatial	Spatial object	Spatial information for given input Spatial object

Special Note: Columns will be added to the input dataset which contains the information required for the spatial object.

Properties Window:

Figure 9-67 – Spatial Info Configuration

Spatial object: Provides option to select a spatial object from drop down list that to append spatial information to.

Items To Output: The information to append to the table by checking the appropriate box. Choices include:

Area (Square Kilometers) appends the calculated area in square kilometers of the spatial objects in the table.

Area (Square Miles) appends the calculated area in square miles of the spatial objects in the table.

Bounding Rectangle as Points appends the spatial objects (2 points) of the record to its bounding rectangle, based on the geographic extent of the incoming spatial object.

Bounding Rectangle as Polygon appends a spatial object (1 polygon) of the record to its bounding rectangle, based on the geographic extent of the incoming spatial object.

Bounding Rectangle as X and Y Fields appends four fields (BR_Left, BR_Top, BR_Right, BR_Bottom), directly related to the bounding rectangle extents' geographic coordinates.

Centroid as Spatial Object appends the spatial object of the record to the centroid.

Centroid as X and Y Fields appends two fields (X, Y) depicting the geographic coordinates of the center of its object.

End Points as Spatial Object appends the end point (extent or where the object sequence ends) of the records' spatial object.

End Points as X and Y Fields appends two fields (X, Y) depicting the geographic coordinates of the end point of a record's object.

Length (Kilometers) appends the line length, in kilometers, of the record's object. If the object is a line, this field will reflect the total line length. If the object is a polygon, this field will reflect the perimeter of the polygon object.

Length (Miles) appends the line length, in miles, of the record's object. If the object is a line, this field will reflect the total line length. If the object is a polygon, this field will reflect the perimeter of the polygon object.

Number of Parts appends the number of parts that make up the object (polylines and polygons only).

Number of Points appends the number of points that make up the object (polylines and polygons only).

Peano Key is used to merge the x-coordinate and y-coordinate of a point so the value can be stored in a one-dimensional array. To produce a Peano key the binary bits of the latitude and longitude value for each point are merged by alternating the bits to produce a new binary number.

Spatial Object Type appends the type of spatial object is associated with the file (Point, Line, Polyline, Polygon).

Projection for XY Output(Optional): By default, Alteryx reports spatial information in WGS 84 projection. The user can select a different known projection and return

spatial information in that projection. Click [ ] to display the Edit Projection box.

9.20 Spatial Process Tool

	The Spatial Process performs high-level spatial object editing from a simple, single tool. You can combine multiple objects or cut the spatial objects of the input table.

Figure 9-68 – Spatial Process | |

	Group	Input	Output
	Spatial	Spatial object	Spatial information for given input Spatial object

Special Note: Columns will be added to the input dataset which contains the information required for the spatial object.

Properties Window:

Figure 9-69 – Spatial Process Configuration

1st Spatial Field: First spatial field selection from input data.

2nd Spatial Field: Second spatial field selection from input data.

Action: User has the option to select the actions from below choices to perform on selected input fields. Choices include:

Combine Objects: Creates a third object by combining both spatial field objects into one object.

Cut 1st from 2nd: Creates a third object by removing the first object where the second object overlaps the first.

Cut 2nd from First: Creates a third object by removing the second object where the first object overlaps the second.

Create Intersection Object: Creates a third object made up of the area where the first and second objects occupy the same area.

Create Inverse Intersection Object: Creates a third object made up of the entire area that the first and second objects occupy, dropping the area where they intersect.

Don't Output Empty Objects: Check this box so that empty objects are not included in the output.

9.21 CASS Tool

<table>
<tr>
<td rowspan="2">

Figure 9-70 – CASS

</td>
<td colspan="3">The CASS Tool takes the input address file and checks it against the USPS Coding Accuracy Support System. The tool cleans up the data by correcting address information to the US Postal Service Standard. It also appends valuable address-specific information to the data.</td>
</tr>
<tr>
<td>**Group**</td>
<td>**Input**</td>
<td>**Output**</td>
</tr>
<tr>
<td>Address</td>
<td>Address</td>
<td>CASS Address</td>
</tr>
<tr>
<td colspan="4">*Special Note:* Records will be shuffled in the output dataset.</td>
</tr>
</table>

Properties Window:

Figure 9-71 – CASS Configuration

- The first two fields, *Company* and *Urbanization* are optional.
- Choose the *Address* fields from the next two boxes. These are mandatory fields. The address field will contain a house number and the Street. Only one address field is required, but the file may have a second address field containing a building, apartment or suite number.
- The last Line is a single field checkbox. Check this box if the *City*, *State,* and *Zip Code* information are contained

within a single field. Then choose the Last Line Field from the drop-down list. If they are in multi-fields, go to Step 4.

- If the *City State* and *Zip Code* information are in separate fields, choose them individually from the drop-down lists.
- The next section has 4 tabs:

> **Output Fields:** Select the relevant boxes next to the fields to include in the analysis. If you want to choose Defaults for U.S. or Canadian addresses, select All fields, or Clear all fields by clicking the appropriate button on the right.

> **U.S. CASS Report:** The user can choose to generate a CASS report of the analysis. Just specify the file location path to save the CASS Report to. The CASS form 3553 is automatically generated when the workflow runs. This report is required to submit to the USPS for bulk mail discounts, as it proves the data has been CASS-Certified.

> **Canada SOA Report:** If you want to generate a Canadian SOA (Statement of Accuracy) report of the analysis. Just specify the file location path to save the SOA Report to. The report is automatically generated when the workflow runs.

> **Options:** Force results to CAPS, when checked will return all field values in all CAPS. This is a requirement for writing the CASS Report.

9.22 Parse Address

|

Figure 9-72 – Parse Address	The Parse Address tool breaks down a street address into its component parts, such as street number, directional (S, NW, and so on), street name, and suffix (ST, RD, BLVD).
	Group	**Input**	**Output**
	Address	Address	CASS Address

Special Note: The CASS Address Parse Tool uses the CASS library; a separate data install that is only available with the Alteryx Spatial offering.

Properties Window:

The tool can parse Last Line data (city, state, and ZIP Code) into the separate parts. This is handy if your address data comes to you as lines of plain text instead of discrete fields.

Figure 9-73 – Parse Address Configuration

- Specify the Address field to parse using the dropdown list.
- Specify the Last Line field to parse using the dropdown list. This is an Optional field.
- Choose to output a single parsed candidate or to output all parsed candidates.

- o **Output Single Parse Candidate:** Each record gets parsed and a single record is returned. This option typically returns the best parse candidate, but may not when there are multiple candidates.

- o **Output All Parse Candidates:** Each record gets parsed and every possible parse candidate gets returned, meaning a record may have multiple ways of being parsed and each method becomes its own record on output.

- Select the Output Fields to return using the checkbox to the left of the field name. The All, US Default, CA Default, and Clear Button should help with making multiple selections. Output fields are defined below:

  - o **CASSParse_City:** The city name is returned only if the "Last Line" field is specified in step 2.

  - o **CASSParse_Garbage:** Any garbage characters left over after the address has been parsed. The returned value is a maximum 50-character string. Common items found in this field are community center names, names of individuals, additional delivery instructions and so on.

  - o **CASSParse_Plus4:** The plus 4 extension of the ZIP Code is returned if the "Last Line" field is specified in step 2.

  - o **CASSParse_PostDirection:** The geographic directional that follows the street name. The value is returned as a maximum 2-character string.

o **CASSParse_PreDirection:** The geographic directional that precedes the street name is returned. The value is returned as a maximum 2-character string.

o **CASSParse_PrivateMailboxName:** The private mailbox number portion is returned if the input address contains the "PMB" abbreviation.

o **CASSParse_PrivateMailboxNumber:** The private mailbox name portion is returned if the input address contains the "PMB" abbreviation.

o **CASSParse_Range:** The delivery number of the parsed address is returned. The value is returned as a maximum 10-character string.

o **CASSParse_State:** The state name abbreviation is returned if the "Last Line" field is specified in step 2.

o **CASSParse_StreetName:** The name of the street is returned. The value is returned as a maximum 28-character string.

o **CASSParse_Suffix:** The suffix portion of an address is returned. The value is returned as a maximum 4-character string.

o **CASSParse_SuiteName:** The name of the secondary unit is returned. The value is returned as a maximum 4-character string containing the suite name portion of the full input address.

- o **CASSParse_SuiteNumber:** The range of any secondary unit is returned. The value is returned as a maximum 6-character string containing the suite number portion of the full input address.

- o **CASSParse_Zip:** The ZIP Code is returned if the "Last Line" field is specified in step 2.

- o **CASSParse_CA_DeliveryInstallation:** Canada Addresses only - The Delivery Installation information is returned. The delivery installation is the post office facility responsible for delivering to the current address. It is often used for rural addresses or when multiple post offices service the same municipality.

- o **CASSParse_CA_LockBox:** Canada Addresses only - The lock box information is returned. A lock box is the Canadian equivalent of a Post Office Box in the US. The two terms are used interchangeably in Canada.

- o **CASSParse_CA_RouteService:** Canada Addresses only - The route service information is returned. The route service is a term used to refer to what the USPS would call a Rural Route address.

9.23 Street Geocoder

Figure 9-74 – Street Geocoder	The Street Geocoder tool places a point object based on address interpolation by converting a multi-line address into a normalized form, with latitude and longitude, spatial object, and additional fields specific to the coding process.		
	Group	**Input**	**Output**
	Address	Address and Geocoder Dataset	Match Status, Score, Spatial Object
Special Note: The Street Geocoder tool requires a separate installation located on the Core Data Bundle Installation Disk.			

Properties Window:

The *Street Geocoder Configuration* window has four components.

- Select A Geocoder: From the dropdown menu, select a geocoder. All Alteryx Geocoders or Alteryx-supported geocoders will be selectable from this list.

Figure 9-75 – Street Geocoder Configuration

- *Input Field Names:* From the dropdown menu, select the appropriate field from your file for the following inputs:
  - *Address:* First line address typically contains the number and street.
  - *City or Last Line:* Last Line is used if you have a field that combines City, State, and ZIP Code fields all in one field

- o *State:* State or Province of the address.
- o *Postcode:* the ZIP Code or Postal Code field.
- *When there is a multiple match:* Specify how the geocoder should handle multiple matches. This setting only applies to geocoders that support multiple matches.
  - o *Prompt to select:* Each time a Multiple Match is found, a dialog box will pop up asking how it should handle the multiple match.
  - o *Return None:* No match will be returned. Reference the Match Status field which will indicate whether a failed record had a multiple match.
  - o *Return 1st candidate:* The first match (with the highest Score) will automatically be returned.
  - o *Return all candidates:* All matches will be returned. In the case of multiple matches, a record for each match is returned. Therefore, the number of records inputted into the Geocoder may be a different number than the number of records coming out of the Geocoder. Reference the Match Status field which will indicate whether a matched record had a multiple match.

9.24 US ZIP9 Coder

Figure 9-76 – US ZIP9 Coder	The US ZIP9 Coder tool associates geographic coordinates with input ZIP9 (also known as ZIP+4) codes in an address file, enabling the user to carry out geography-based analyses.

The US ZIP9 Coder appends latitude and longitude, county and state FIPS code, and census tract and block number.		
Group	**Input**	**Output**
Address	ZIP Code and Geocoder Dataset	Match Status, Score, Spatial Object
Special Note: The US ZIP9 Coder tool requires a separate installation located on the Core Data Bundle Installation Disk.		

Properties Window:

The *US ZIP9 Coder Configuration* window has the below options.
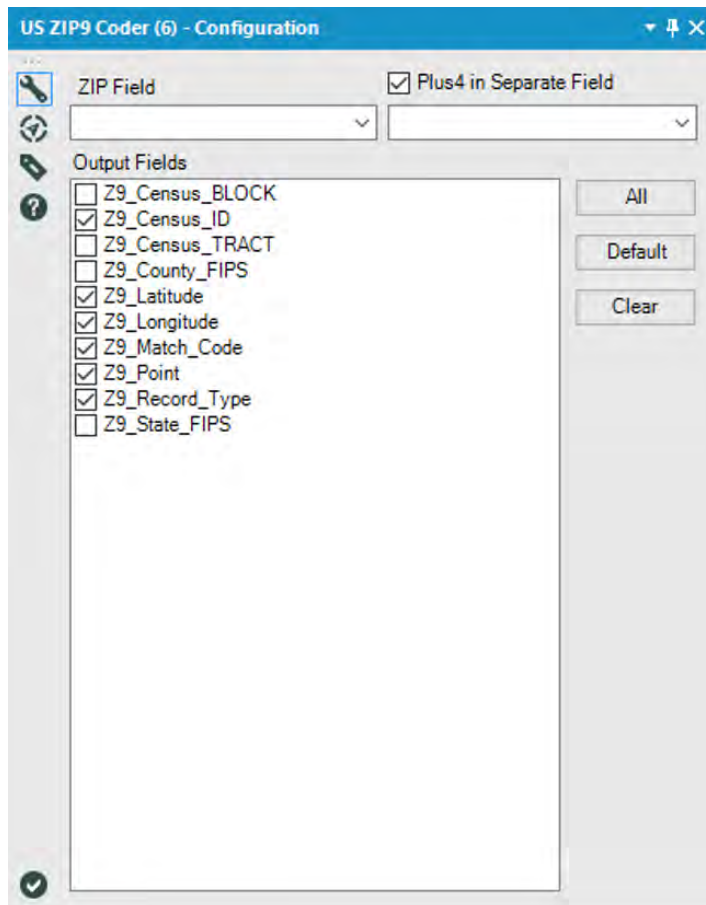
Figure 9-77 – US ZIP9 Coder Configuration

- Discern whether the input table has the 9-digit ZIP Code in a single field, or if it is parsed into 2 fields, a 5-digit ZIP Code and then the 4-digit appendage.
  - o Single Field: The ZIP9 code is contained in one data field.
  - o Two Fields: The 5-digit ZIP Code is in one data field and the ZIP+4 Code is in a different data field.
- Choose the fields to add to the output.

The ZIP9 Coder tool can produce any or all of the following outputs (defaults in green):

- o *Z9_Census_BLOCK*: Appends the Census Block ID that the ZIP9 falls in. (The Census block group code is the first digit of the Census block code)
- o *Z9_Census_ID*: Depending on the Match Code the record was coded to, the CensusID will go to that level of granularity. Census Block Code being the most granular level.
- o *Z9_Census_TRACT*: Appends the Census Tract ID that the ZIP9 falls in.
- o *Z9_County_FIPS*: Appends the County FIPS ID that the ZIP9 falls in.
- o *Z9_Latitude*: Assigns the Latitude (Y) coordinate to the ZIP9
- o *Z9_Longitude*: Assigns the Longitude (X) coordinate to the ZIP9
- o *Z9_Match_Code*: The ZIP level the record was coded to.
- o *Z9_Point*: The spatial object of the ZIP9
- o *Z9_Record_Type*: describes the type of facility the ZIP9 code represents.
- o *Z9_State_FIPS*: Appends the State FIPS ID that the ZIP9 falls in.

9.24 Who wants Coffee?

To...	Alteryx Consultants
Subject	Who Wants Coffee?

Hey,

The Mermaid Coffee company got back to me and really like what we gave them.

They want to add another output to the app. In addition to the report, it should also return the data with total population joined onto each zip code. They said if the zip code is invalid, just return the record without the population.

Since adding the total population is something that I know you will have to do regularly, you might as well make a macro that joins total census data by zip code onto the data stream that you give it.

The data coming into the macro needs a string field called "Zip" with Zip 5s in it. If the zip doesn't exist, it should return the record without a population.

I'll check on you in a little bit.

CHAPTER 10

Meta-Morphosis

To...	Alteryx Consultants
Subject	All The Tools Along the way
Attached	JSON.txt

Hey,

I have been asked to make sure you know how to use everything in a list of tools from the higher-ups.

There are a few that you haven't gotten to know yet, so I have designed something for us to go over and then for you to work through.

Attached, you will find a file called JSON.txt. We will be using it to create an application such that when you check a field, the JSON data will be converted into a relational structure. If you don't check the field, it will output a file identical to the original JSON file.

Thanks

10.1 Tools & Concepts

Tools	**Concepts**
Check Box	Alternative Data Prep
Detour	JSON Files
Detour End	Documentation
Dynamic Rename	Data Cleaning
JSON Parse	JSON Files
Message	Information Display

10.2 Check Box

Figure 10-1 – Check Box	The *Check Box* interface tool allows us to create a check box question.		
	Group	**Input**	**Output**
	Interface	None	None
Special Note: This *Check Box* returns a Boolean value. The bottom *Black Question Anchor* connects to question anchors on top of other tools to pass the results to the subsequent tool.			

<u>Properties Window:</u>

The *Check Box Configuration* window has three components.
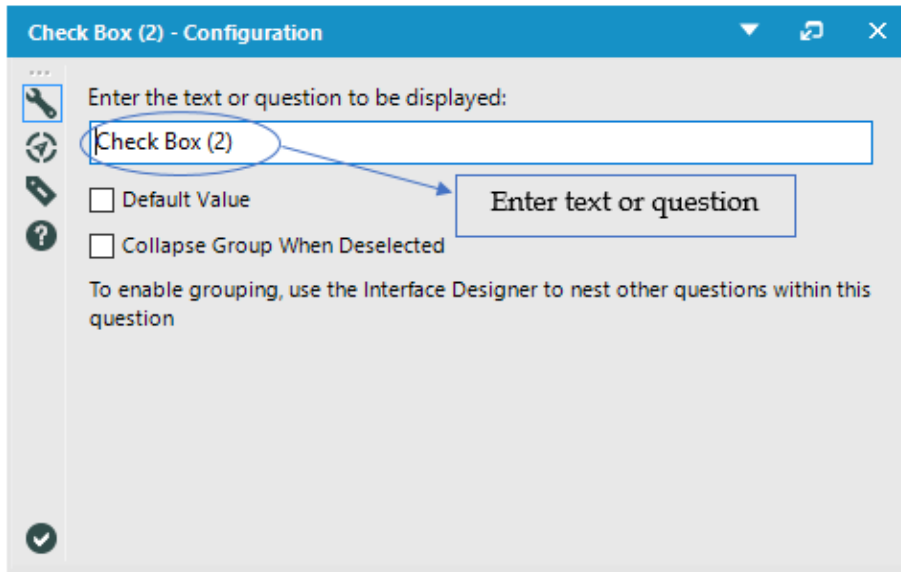
Figure 10.2 – Check Box Configuration

- *Enter the text or question to be displayed* is a text box that will be the question prompt.
- *Default Value* can either be checked or unchecked to determine what the default result will be.
- *Collapse Group When Deselected* allows us to hide the relevant group in the interface designer.

10.3 Detour

	The *Detour* tool allows us to create alternative paths dependent on the user input.		
	Group	**Input**	**Output**
	Developer	Any data stream	Output L & R sections below

10.3 Detour

Note: Detour is useful if we are developing flexible tools that allow choices in data inputs or outputs. This tool should almost always be used with an action tool. Only the Left or Right stream will be active.

An Action tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output L: Nothing or the input data stream.

Output R: Nothing or the input data stream.

Properties Window:

The *Detour Configuration* window has only one component.
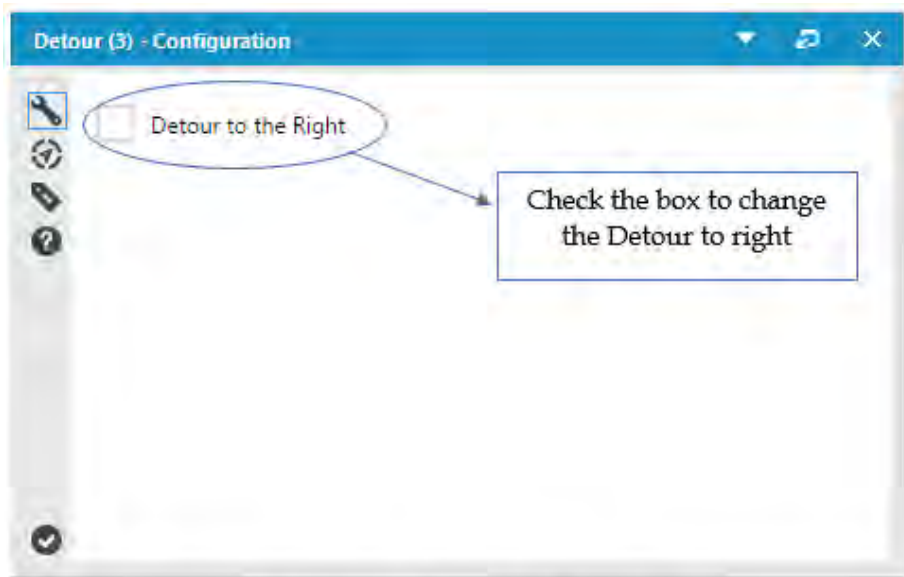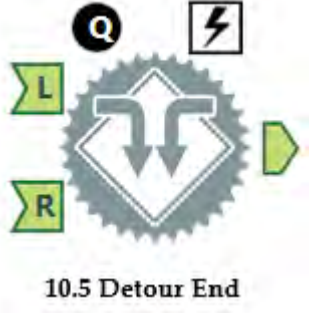
Figure 10.4 – Detour Configuration

- The *default* behavior for the *detour* tool is to always divert to the left (L) data stream.
- *Detour to the Right* allows us to reverse this default behavior.

10.4 Detour End

	The *Detour End* tool rejoins two optional paths.		
	Group	**Input**	**Output**
10.5 Detour End	Developer	Input L & Input R below	Any data Stream

Note: The *Detour End* tool should be used after we have used a Detour tool in order to bring the data streams back together. Only the Left or Right stream can be active.

Input L: Nothing or any data stream.

Input R: Nothing or any data stream.

An Action tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Detour End Configuration* window has no configuration.

Figure 10.6 – Detour End configuration

10.5 Dynamic Rename

	The *Dynamic Rename* tool allows us to systematically rename fields.		
Figure 10.7 – Dynamic Rename	**Group**	**Input**	**Output**
	Developer	Input L & Input R below	Any data stream

Note: The *Dynamic Rename* tool is useful in properly identifying column headers in data that either have header rows before the column names or no column names.

Input L: Any data stream.

Input R: This input is optional, and its structure depends on the Rename Mode.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Application questions can be connected to the top *Black Question Anchor* to use answers in the tool.

<u>Properties Window:</u>

The *Dynamic Rename Configuration* window has three core components and eight different modes.
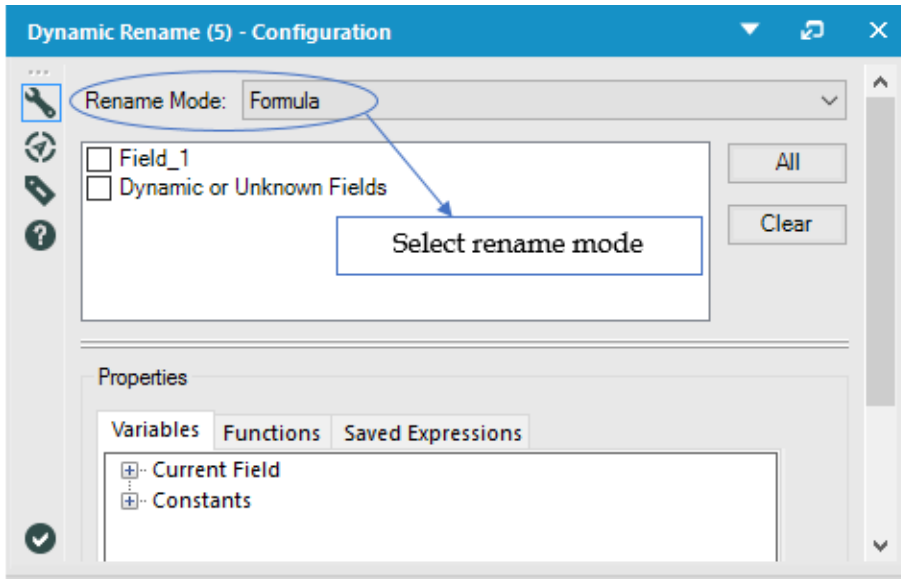
Figure 10.8 – Dynamic Rename configuration

- *Rename mode* allows us to select the method used to rename the fields in the left (L) input.

If we are using one of the single input rename modes, we will see a field selection window.

- The *Properties* section changes, depending on the Rename Mode.
- *Formula Rename Mode* allows us to define an expression that will name our columns.
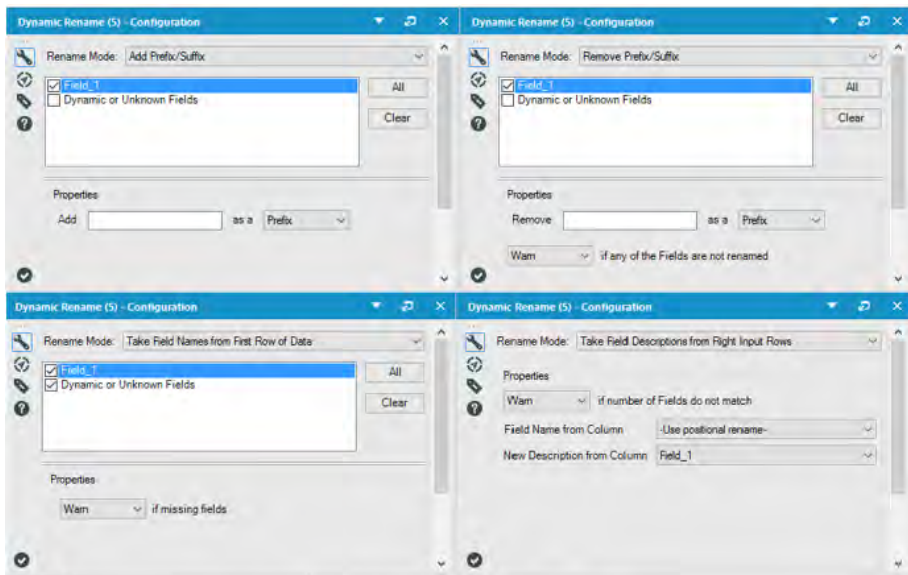
Figure 10-9a – Dynamic Rename (contd.)

- *Add Prefix/Suffix Rename Mode* allows us to define a string to add to the beginning or end of specific fields.
- *Remove Prefix/Suffix Mode* allows us to remove a common string from the beginning or end of specific fields.
- *Take Field Names from First Row of Data* allows us to replace the column headers with the first record of our data.
- *Take Field Descriptions from Right Input Rows* allows us to use a definition file to redefine the fields.
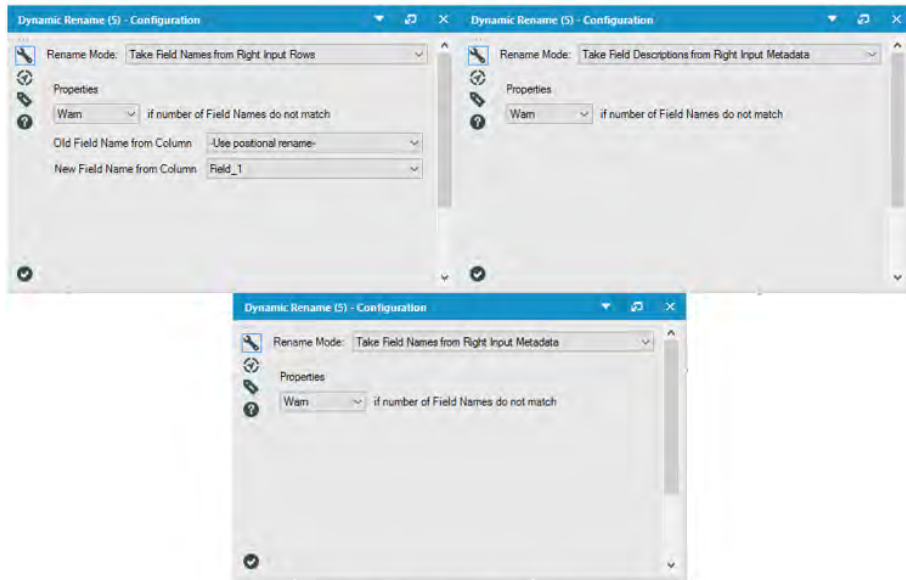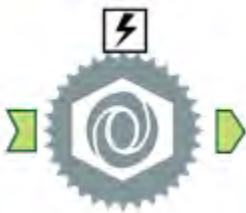
Figure 10.9b – Dynamic Rename (contd.)

Take Field Names from Right Input Rows allows us to use a definition file to rename the fields.

Take Field Names from Right Input Metadata allows us to use a data stream that we know matches to rename the fields.

Take Field Descriptions from Right Input Metadata allows us to use a data stream that we know matches to redefine the fields.

10.6   JSON Parse

Figure 10-10 – JSON Parse	The *JSON Parse* tool allows us to read JSON formatted files into relational tables.		
	Group	**Input**	**Output**
	Developer	JSON formatted data stream	See below

Note: For those unfamiliar with JSON: JSON stands for JavaScript Object Notation, which is a structured data format typically used in data transfer.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Output: A data stream in relational data format.

Properties Window:

The *JSON Parse Configuration* window has three components.
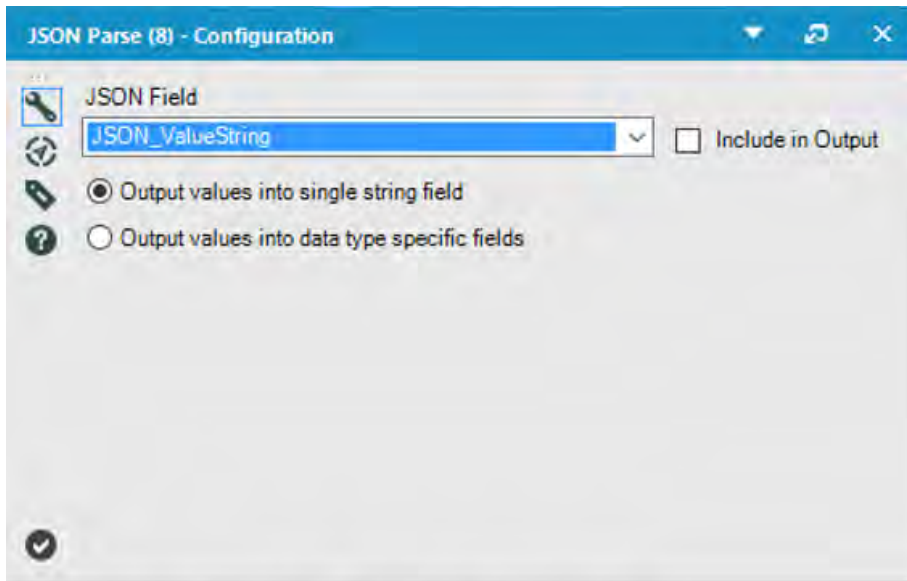
Figure 10.11 – JSON Parse Configuration

- *JSON Field* is the field in the incoming data stream that has JSON data written with one complete record in the cell.
- *Include in Output*, when checked, will maintain the JSON field when it is output from this tool.
- *The Output Values* options allow us to decide if each of the data values should be included in a single string field or in a field defined by the JSON data type.

10.7   Message

|

Figure 10-12 - Message	The *Message* tool allows us to create a message in the output window.
	Group	**Input**	**Output**
	Developer	Any data stream	Any data stream

Note: Message helps find out where something went wrong and log the settings during a run. It also provides updates about parts of our workflow that have completed.

Application questions can be connected to the top *Black Question Anchor* to use those answers in this tool.

An Action tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Message Configuration* window has five basic components.
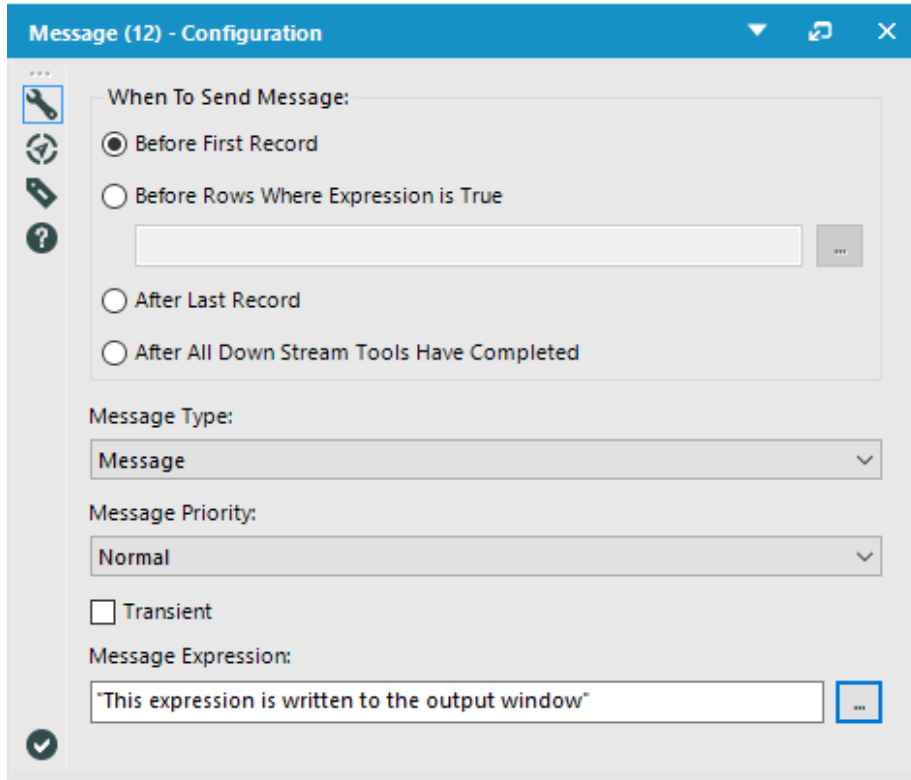
Figure 10.13 – Message Configuration

- *When To Send Message* defines what the trigger for writing out the message is. It can be just before the first record, before every record where a Boolean expression is true after the last record has passed through the tool, or when the data stream has terminated.
- *Message Type* defines what type of output should be written out and if it should stop the data flow (see Output Window in The Interface for more details).

- *Message Priority* determines what the message would do by default if it were in a macro. Normal will show the message if the tool is in the workflow. Medium will show the message if the tool is directly in a macro used in the workflow. High will show the message no matter how many macros deep the message occurred.
- *Transient*, when checked, means that if the Message tool writes more than one message, it will replace the single message instead of creating a new one every time.
- *Message Expression* is an expression field that will output the result to the output window.

10.8   All the Tools Along the Way

To...	Alteryx Consultants
Subject	All The Tools Along the way
Attached	JSON.txt

Hey,

I have been asked to make sure you know how to use everything in a list of tools from the higher-ups.

There are a few that you haven't gotten to know yet, so I have designed something for us to go over and then for you to work through.

Attached, you will find a file called JSON.txt. We will be using it to create an application such that when you check a field, the JSON data will be converted into a relational structure. If you don't check the field, it will output a file identical to the original JSON file.

Thanks,

The option for a tool to essentially just copy the file seems a little strange, but it will make a more sense when we start working on the follow-up assignment. We can write the file to a .csv instead of a .txt file at the end of the workflow. Let us start by connecting to the data.
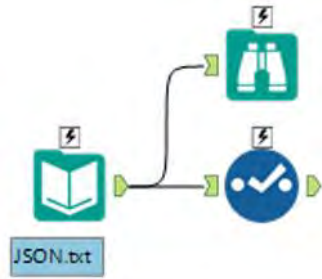
Figure 10-14 – Open JSON File
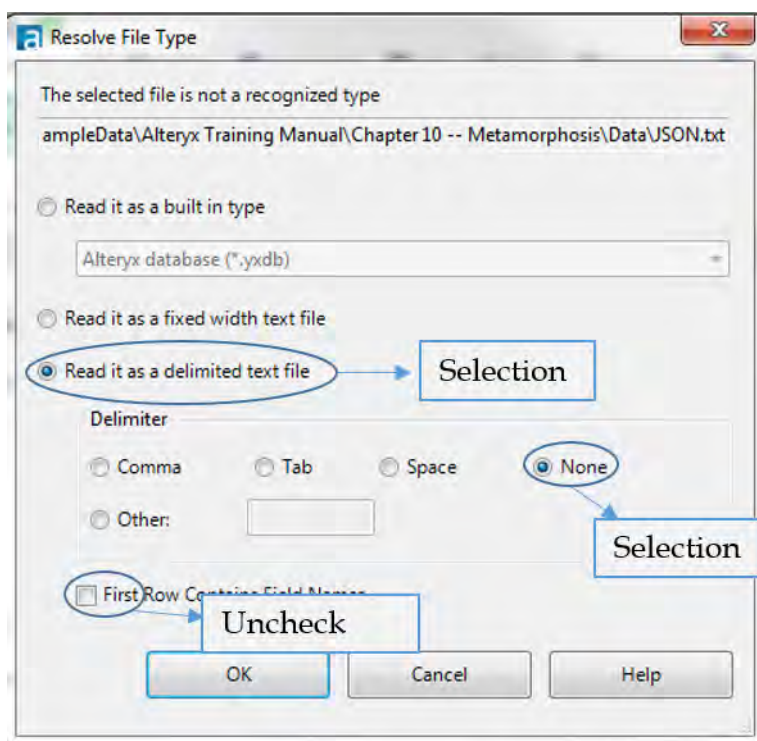
Figure 10-15 – Read JSON File
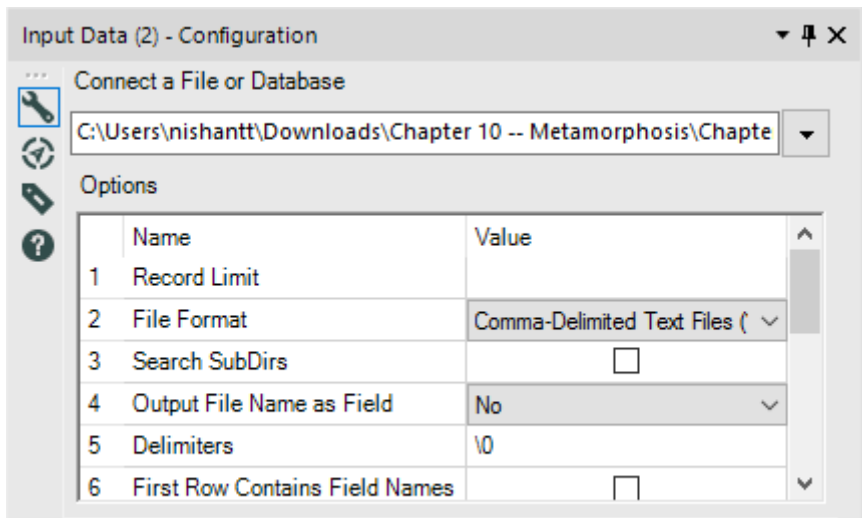
Figure 10-16 - JSON Configuration
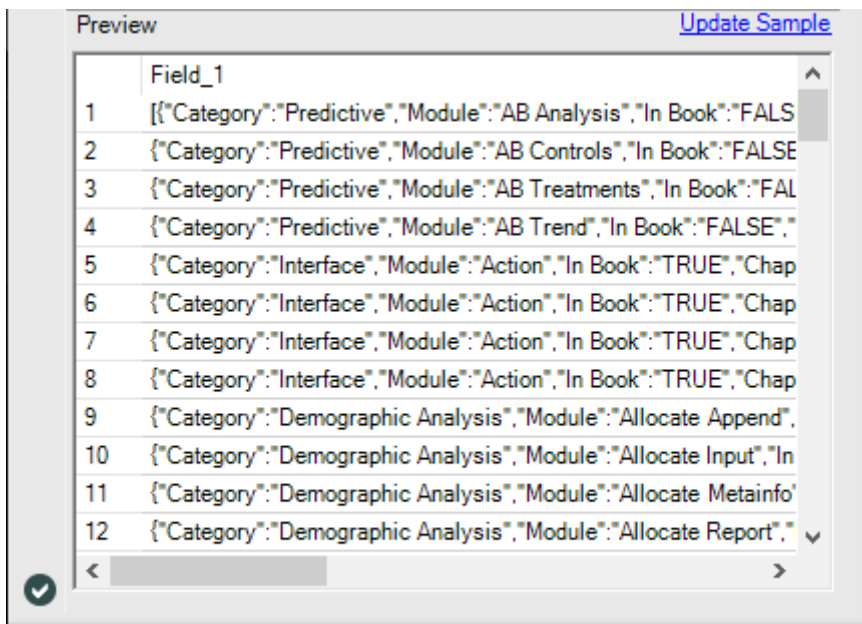
Figure 10-17 - JSON Preview

When we connect to this file, because it is *.txt*, Alteryx needs to know how it should read it. This is by design because

file formats like JSON need to be read so that all data is in a single field and ideally one record per row. Because we want these settings, we can choose to have the file delimited with no delimiter so that nothing will break the field. If we take a look at the file, we can see there is no header row. This is because JSON carries the header names in every record and thus does not need a devoted row.
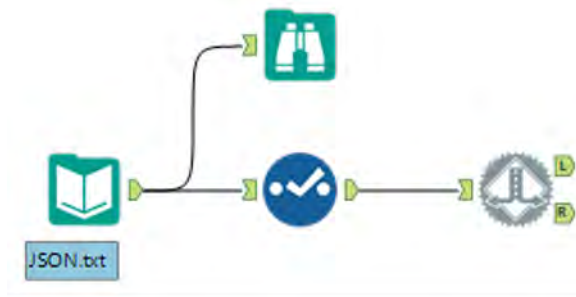
Figure 10-18 – JSON Detour

Our next step will be to add a *Detour* tool. This tool allows us to use a question in order to decide which data stream to follow. The *Detour to the Right* option is so that while we are testing our workflow, we can properly check both sides. By design, the Detour tool detours to the left unless updated.
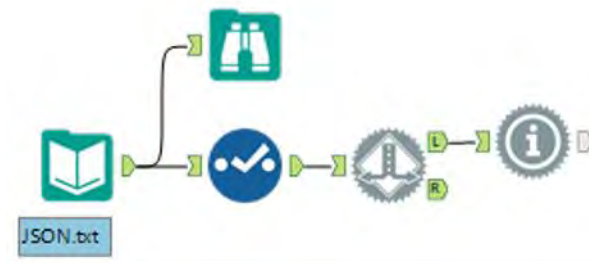
Figure 10-19 – JSON Detour to the Left

Let's start by designing the left (L) data stream. We are going to make this the side where we do not convert the data to a relational structure.

We are going to use *Message* tools to record what will happen to the data. The Message tool allows us to write something to the output window. In this case, we are going to write "*User Decided not to reformat the JSON data file*" so that it will be in the output log.
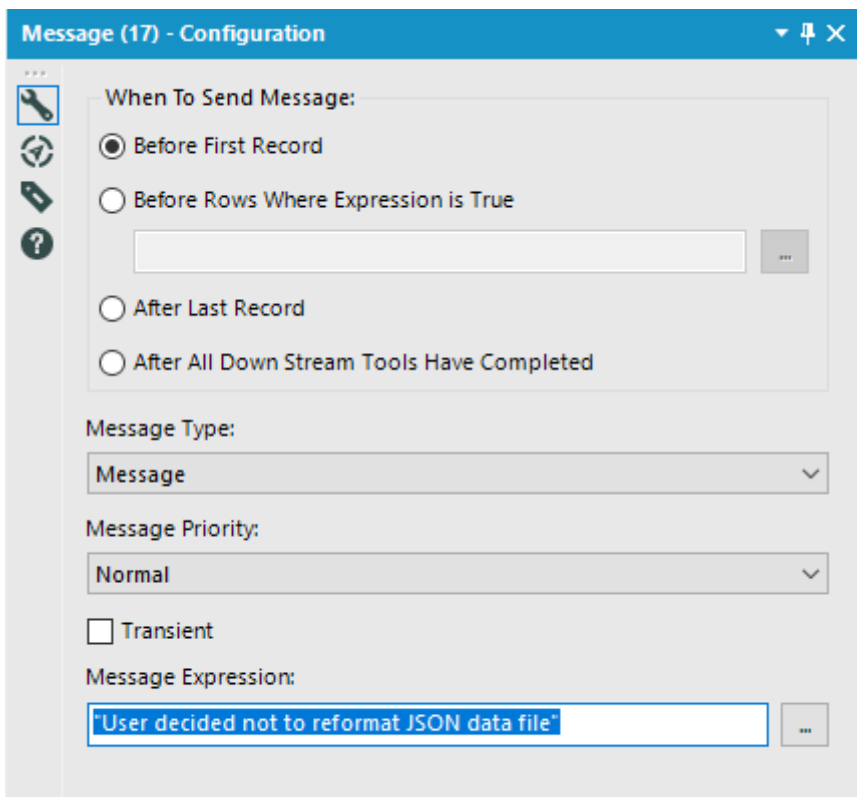
Figure 10-20 - JSON Message Configuration
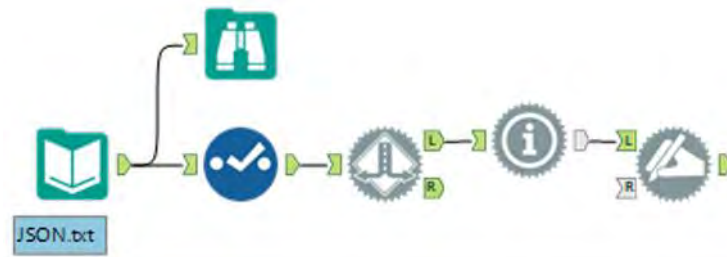
Figure 10-21 – JSON Left data stream
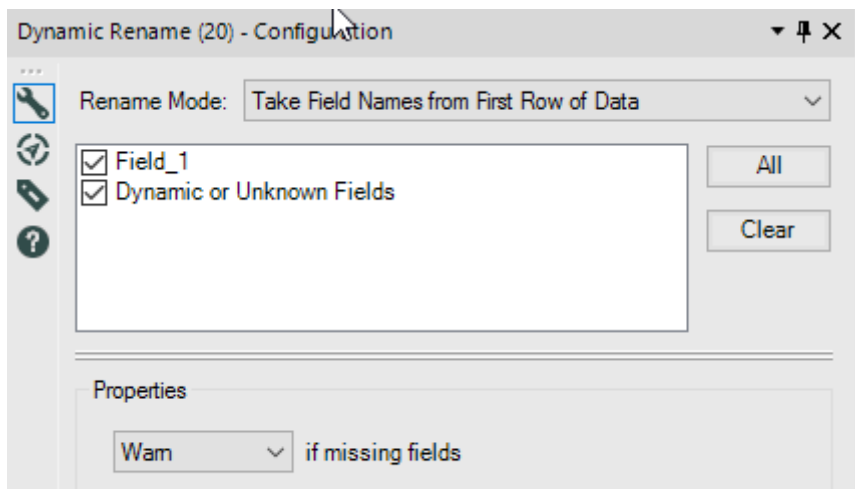
Figure 10-22 - JSON Dynamic Rename Configuration

Now let us add a *Dynamic Rename* tool. This is because once we bring the data back together to write, we will need to write the header. Since we don't want a header for this option on the data stream, we will modify it such that the header is actually the first record of data. This will complete all of the data prep we need before rejoining the data streams to be written out.
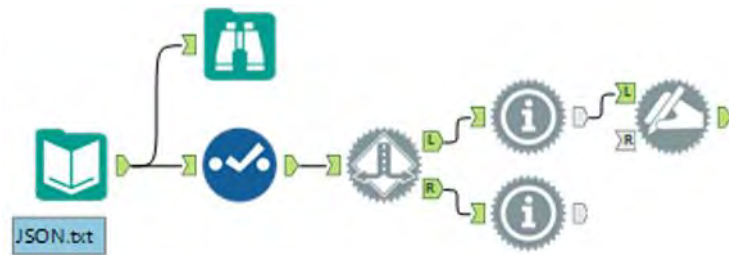
Figure 10-23 – JSON Right data stream
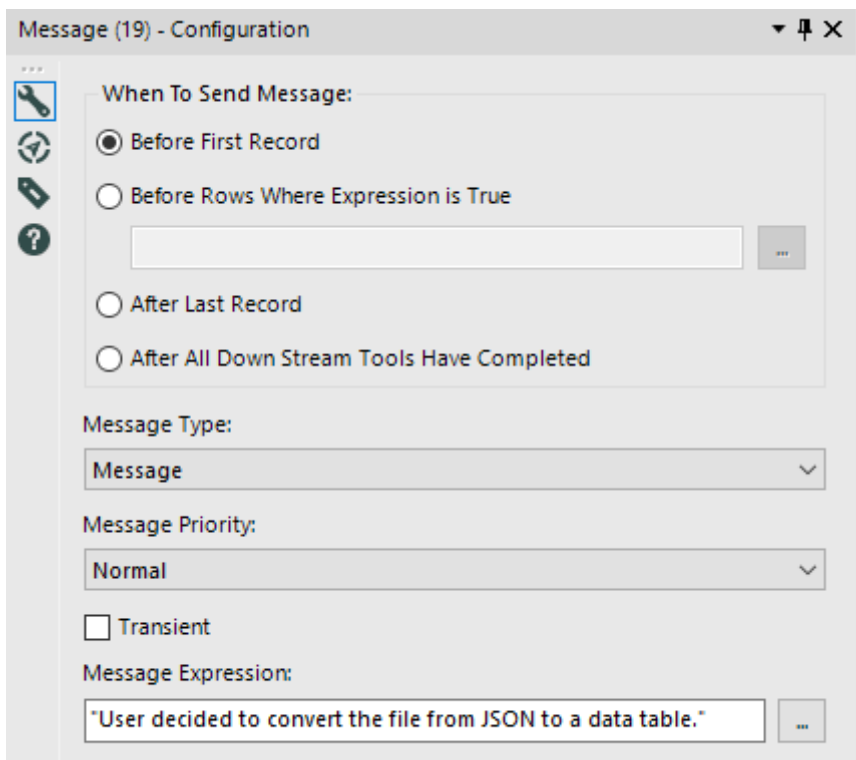
Figure 10-24 - JSON Message configuration for Right Side data stream

We are going to start the right (R) side of the data stream by creating a parallel message to the left that reads *"User decided to convert the file from JSON to a data table"*.
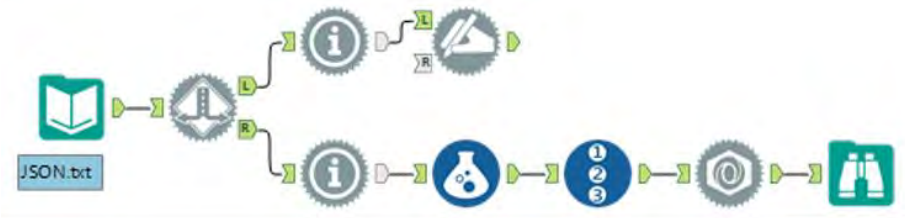
Figure 10-25 - JSON Right side data stream completed

We will introduce a *Formula* tool now and convert each row of the incoming data into an array by introducing '[' and ']' symbols at the start and end of each row.
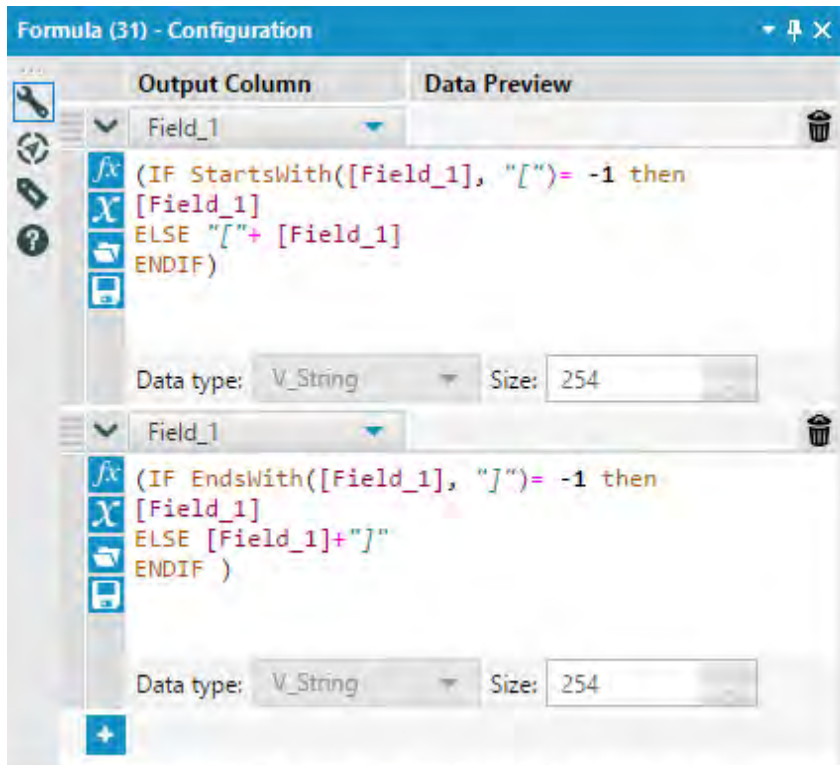
Figure 10-25(Contd) - Formula Field Calculation

We then introduce Record ID tool to number each row which will eventually help us in grouping the data as we shall see. Now, we will bring the *JSON Parse* tool onto the canvas. This tool is designed to look at JSON data and convert it into relationally structured data. Unfortunately, if we just put the tool at the end of the data stream, it does not recognize any fields. This is because the *Detour* is diverting all information to the left, including the metadata. In order to configure the tools on this side of the *Detour*, we need to divert the tool to the right.
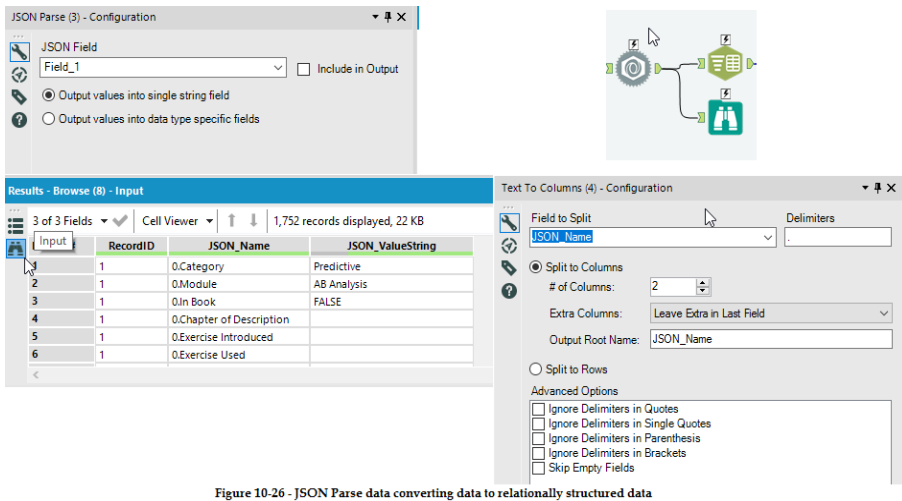
Figure 10-26 - JSON Parse data converting data to relationally structured data

Now that we have restructured the data, we should take a look at the result so we know what our next steps should be. We can see the JSON Parse tool has created two fields: *JSON_Name*, which has the record number and field name separated by a period, and *JSON_ValueString*, which has the values associated with each record and variable.

Note: If we wanted to convert data into JSON format using the *JSON Build* tool, this is the format it should be in.

Since we know we want a relational dataset that is easy to work with, we should make it tidy by creating a single column for each variable.

In order to do this, we first need to identify the unique column names by splitting them off of the record number. We can use the *Text To Columns* tool in order to create a separate column for both the record number and the column name so we can use the two of them to restructure the data.
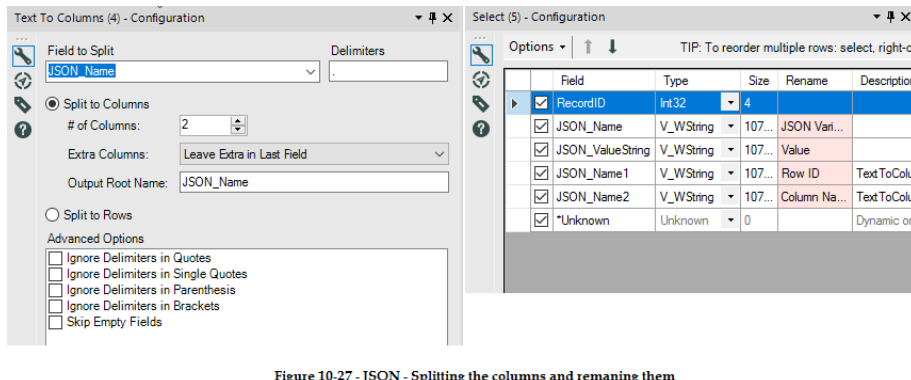
Figure 10-27 - JSON - Splitting the columns and remaning them

In order to make the data easier to understand we should rename our fields.

Figure 10-28 – JSON – Relationally structured data

Now that we have our fields well named, we can pivot the data into a more usable format. We are going to group the data by the *Record ID*, use the *Field Names* as our columns, and use the

value field as the data in the intersection. We need to set a methodology. However, since we are using a unique combination of header and grouping fields, we do not need to worry about clashing data.
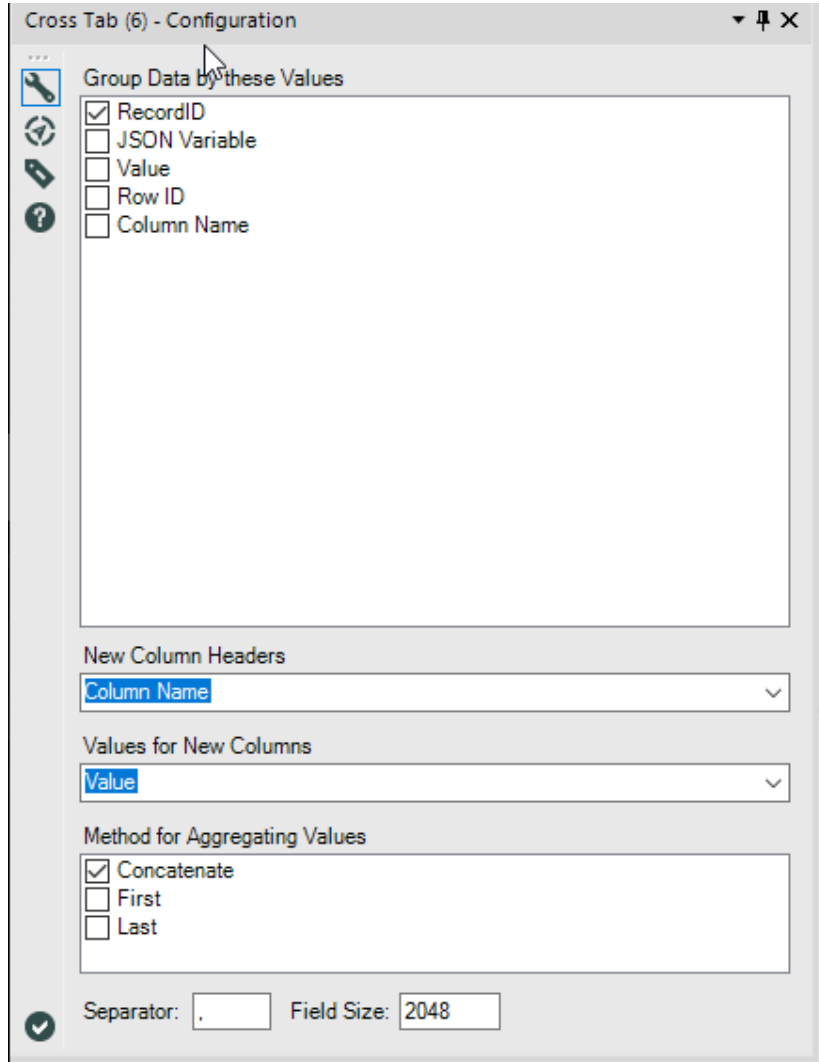
Figure 10-29 - JSON Data Grouping by Record ID

Figure 10-30 - JSON Groupped Data Set

If we take a look at the data, we can see we have a tidy dataset, which means we can end this specialized portion of the workflow and bring the two data streams back together.
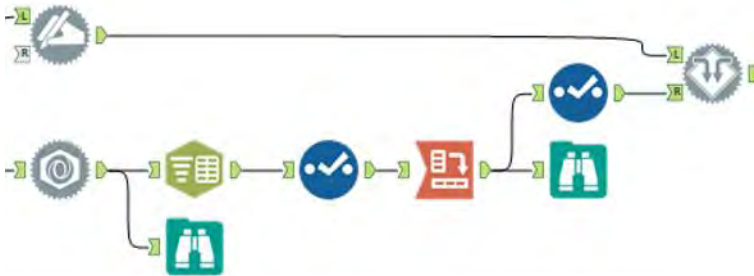
Figure 10-31 – JSON – Bringing both data streams together

Now we can connect both sides of the data stream to a *Detour End* tool to bring them back together so we can use a single output.

Figure 10-32 – JSON – Using Detour End Tool to bring together the data streams

Once we add the Output Data tool, we are almost done. As we can see, we have a fully operational data stream. The only problem we have is that we have no way to control if the file will be written as JSON or a relational table.

We are going to add a *Check Box* tool to the workflow so we can give the option of selecting between converting the data or not. If we connect the question directly to the *Detour* tool, the action will be created exactly how we want it to work, except that we should take a look at the behavior of this action because it doesn't just change the direction, like implied by the action type.
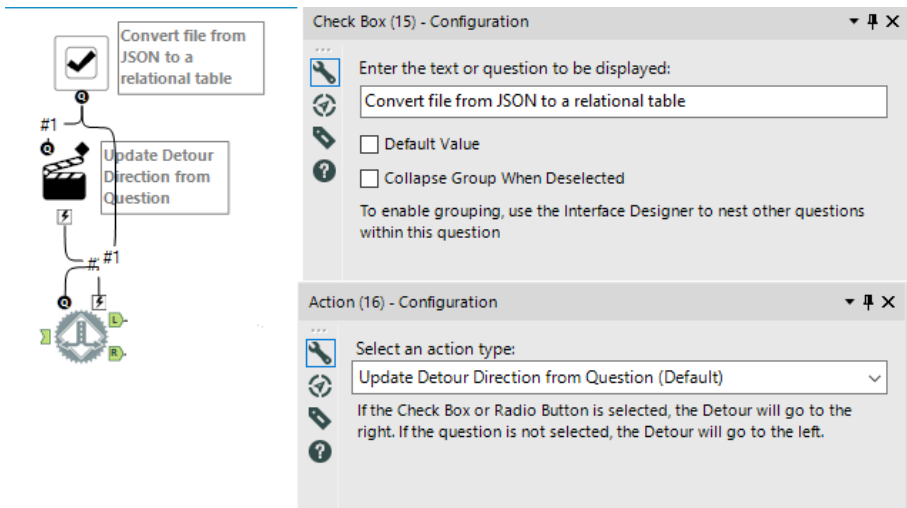
Figure 10-33 - JSON Adding Checkbox to provide an option for conversion

Anytime we are running the Detour tool as an application, the default path is to go to the left. When we have an action to update this behavior, it will change the result to divert to the right. We should name the question to reflect this behavior.

Figure 10-34 – JSON data stream on completion

10.9   Narrowing Down the Search

To...	Alteryx Consultants
Subject	Narrowing Down The Search

Hey,

This has come along nicely, but here's where it gets complicated; I need you to add a question that will filter the data streams to the specific tool you type into it.

Thanks

CHAPTER 11
Let's Do It (In-DB)

	From ▼	Google Apps
Send	To...	Alteryx Consultants
	Cc...	
	Subject	Superstore Database Report
	Attached	Product Info.csv 160 KB

Hey,

Superstore Inc. would like to generate data which shows Top 10 products by order quantity for Internet Sales since 2006. The data should also include the year in which this product achieved its highest sales.

However, there's a catch. While their Fact table of Internet Sales is in a SQL Server database, the Product information is available as a file (attached).

Customer would like to have the output in two different ways.
1. As a CSV file for use by other systems.
2. As a table on the SQL Server database where the fact table resides.

Please implement the logic using Alteryx and let us know when this is ready.

11.1 Tools & Concepts

In this chapter, we will cover the features for in-database processing in Alteryx designer 11.0 and the various tools and concepts as mentioned below:

Tools	Concepts
Browse In-DB	Reviewing In-DB Stream
Connect In-DB	Connecting to In-DB Stream
Data Stream In	Bring Data into In-DB Stream
Data Stream Out	Export Data from In-DB Stream
Dynamic Input In-DB	Query Data into In-DB Stream
Dynamic Output In-DB	Output Data for Predictive In-DB
Filter In-DB	Filtering In-DB using Queries
Formula In-DB	Create/Update In-DB Fields
Join In-DB	Combining Two In-DB Stream
Macro Input In-DB	In-DB Stream Input using Macro
Macro Output In-DB	In-DB Stream Output using Macro
Sample In-DB	Limiting In-DB Stream
Select In-DB	Organizing In-DB Stream
Summarize In-DB	Aggregating In-DB Stream
Union In-DB	Combine multiple In-DB Streams
Write Data In-DB	Updating DB using In-DB Stream

11.2 Browse In-DB

|
Figure 11-1 –
Browse In-DB	The *Browse In-DB* tool allows us to review the In-DB data stream at any point in the workflow.		
	Group	**Input**	**Output**
	In-Database	In-DB Stream	None

Note: This tool plays a very important role development of a workflow as you can view the data at every stage. However, In-DB browse triggers a database query every time it is used and hence it degrades performance. It is important to disable this tool when development is complete to improve performance.

An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Browse In-DB* tool provides us with the following set of options.

Figure 11-2 – Browse In-DB, Configuration

- *Browse first*: This option allows us to set the number to records to display in the *Results* window when the workflow is run. The default value for this configuration in 100 whereas specifying a "0" means allowing maximum numbers of records to be displayed. The maximum number of records is 2 billion.

- *Enable Caching*: Enabling caching option allows the Browse In-DB tool not to re-run the query if there are no changes to the database connection or the query. In such a case, the results to be displayed are fetched from cache. This option helps improve the performance of the workflow.

- *Clear Cache*: This option forces cache to be cleared irrespective of any changes to database connection or

query. The cache is rebuilt the next time workflow is run if the *Enable caching* option is checked.

11.3 Connect In-DB

Figure 11-3 - Connect In-DB	The *Connect In-DB* tool allows us to connect to an In-DB data stream in the workflow.		
	Group	**Input**	**Output**
	In-Database	Connection Name	In-DB Stream
An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for Connect In-DB tool provides us with the following set of options.
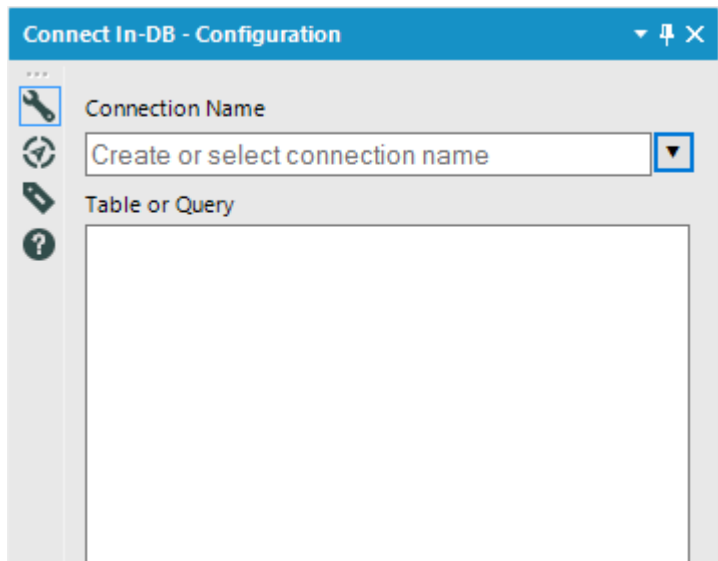
Figure 11-4 – Connect In-DB, Configuration

- *Connection Name*: This dropdown option provides us with the option to *Manage Connections* or *Open File Connection*.

  Manage Connections: Selecting this option to open the window for us to create/modify an In-DB connection. This window has the following options to offer

Figure 11-5 – Connect In-DB, Manage In-DB Connections

- *Data Source*: Allows you to select the type of database you would like to connect. The options available are

  Amazon Redshift, Databricks, Hive, IBM Netezza, Microsoft Analytics Platform System, Microsoft Azure SQL Database, Microsoft Azure SQL Data Warehouse, Cloudera Impala, Microsoft SQL Server, Oracle, SAP, Hana, Spark, Teradata

- *Connection Type*: Allows you to specify the type of connection as *User* or *File*.

- *Connections*: Here you can either select an existing connection from the dropdown list or you can create a new one using the *new* button. If you select a connection from dropdown list, the delete option is enabled to let you delete the connection or you can continue to edit the connection.

- *Connection Name*: Lets you specify a name for the connection being created.

- *Password Encryption*: This provides you with options to specify the level of security if the database connection contains the password.

- *Driver*: This option on the *Read* tab helps you select the driver for the Data Source that you had opted for. You can leave the driver to the default value if unsure about which driver to choose.

- *Connection String*: This dropdown list provides you with the option to either select a previously saved connection string or to create a new one. *New database connection* option in the list allows you to create a new connection string by providing the connection details specific to the type of driver chosen by you in the previous steps.

- *Options*: This section has different values for Name and Value columns based on the type of data source chosen by you. If unsure, leave the default values in place.

  Similar to *Read* tab, *Write* tab also needs the set of details for *Driver*, *Connection String,* and *Options*.

  If you are working with a database connection which contains several tables, the *Choose Table or Specify Query* window opens after clicking OK on the *Manage In-DB Connections* window. This window allows you can select a particular table, stored procedure or write a query you want to execute to fetch data.

  Open-File Connection: Allows you to open a .indbc file which was created earlier using *Manage Connections* option in the dropdown list.

- *Table or Query:* This field in the configuration window is auto-populated with information of the table or query specified in the *Choose Table or Specify Query* window.

- *Query Builder*: Clicking on this button opens the *Choose Table or Specify Query* window to allow making any changes to the database object or query selected while setting up the connection.

11.4 Data Stream In

Figure 11-6 - Data Stream In	The *Data Stream In* tool loads a standard input data into In-DB stream by creating a temporary database table to load the data		
	Group	**Input**	**Output**
	In-Database	Input Data	In-DB Stream
Note: If using Oracle database for creation of the temporary table, you must have permission to write to temp space assigned to GLOBAL TEMPORARY. An *Action* tool can connect to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for *Data Stream In* tool provides us with the following set of options.
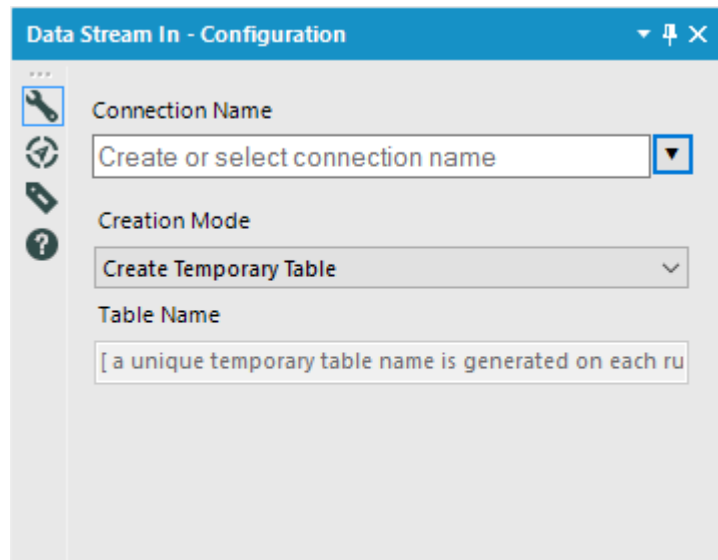
Figure 11-7 – Data Stream In, Configuration

- *Connection Name*: The dropdown list provides options to *Manage Connections* or *Open File Connection*. Using these options, you can use any existing database connection saved in Alteryx Designed or open the *.indbc* file for setting up the database connection.

- *Creation Mode*: This configuration provides option to

  Create Temporary Table for creating a new temporary table with a unique name every time the workflow is run.

  Create New Table for creating a new database table to hold the input data which is not dropped when the workflow execution completes.

Overwrite Table (Drop) to drop any existing table and then recreate the table with the same name. After recreation, the table is populated with the input data.

- *Table Name*: This option is enabled if you have selected *Create New Table* or *Overwrite Table (Drop)* option in Creation Mode configuration. This field is used to mention the name of the table as required to be created in the database.

11.5 Data Stream Out

| Figure 11-8 - Data Stream Out	The *Data Stream Out* tool provides the ability to move data from In-DB Stream to the workflow.		
	Group	**Input**	**Output**
	In-Database	In-DB Stream	Output Data
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for *Data Stream Out* tool allows you to *sort* the output records based on the order defined in the *fields* section.
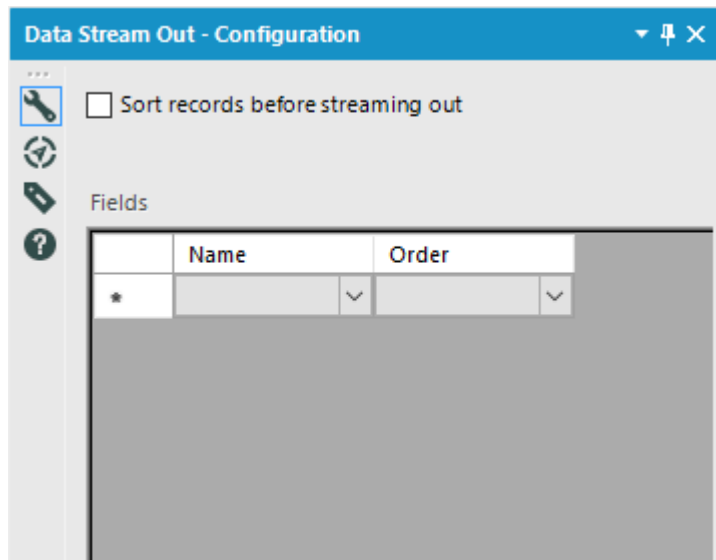
Figure 11-9 – Data Stream Out, Configuration

Sorting of records before streaming out is completely optional and may be skipped if not required.

11.6 Dynamic Input In-DB

Figure 11-10 - Dynamic Input In-DB

The *Dynamic Input In-DB* tool accepts the connection name and query as a field from input data and then converts it into an In-DB stream of data.

Group	Input	Output
In-Database	Input Data	In-DB Stream

> *Dynamic Input In-DB* tool is used alongside *Dynamic Output In-DB* tool in a macro meant for predictive analysis.
>
> An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Dynamic Input In-DB* tool provides us with the following set of options.
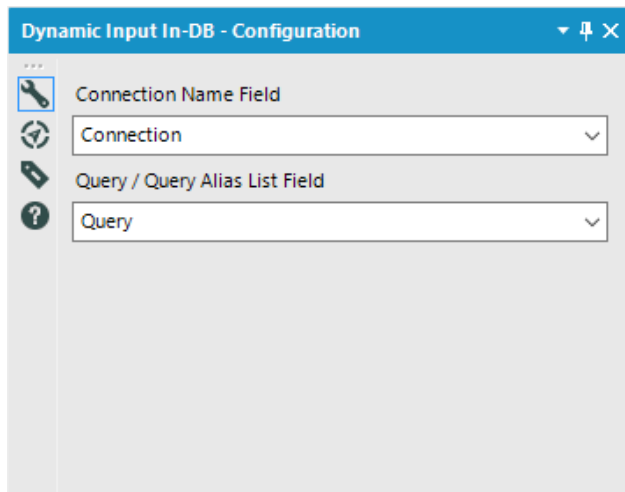
Figure 11-11 – Dynamic Input In-DB, Configuration

- *Connection Name Field* dropdown list is populated using the fields from Input data. The list allows selecting the field which contains the connection name to be used.

- *Query / Query Alias List Field* dropdown list is also populated using the fields from Input data. This list allows us to the field which contains the query information to be run against the connection name selected.

11.7 Dynamic Output In-DB

 Figure 11-12 - **Dynamic Output In-DB**	The *Dynamic Output In-DB* tool outputs information about In-DB stream to the workflow for Predictive In-DB		
	Group	**Input**	**Output**
	In-Database	In-DB Stream	Data Output
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for *Dynamic Output In-DB* tool allows us to select the type of data required as part of the output.
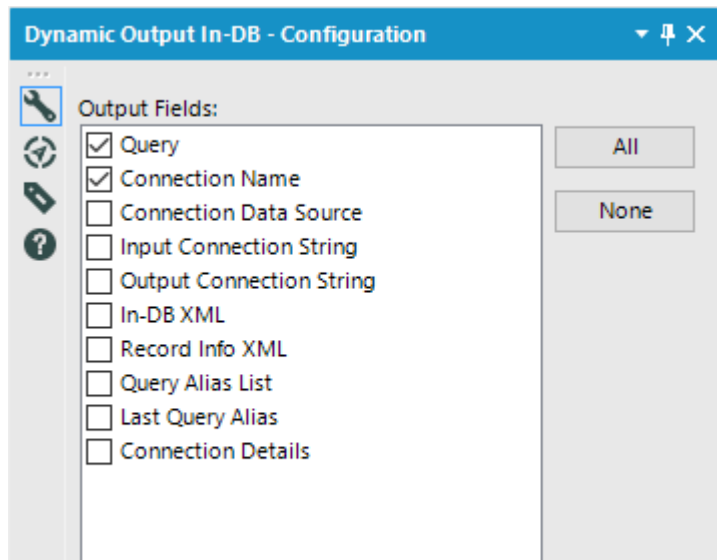
Figure 11-13 – Dynamic Output In-DB, Configuration

- *Query* outputs the In-DB query in use right before *Dynamic Output In-DB tool.*

- *Connection Name* outputs the database connection name alias for In-DB connection.

- *Connection Data Source* outputs the type of database.

- *Input Connection String* outputs the database information required to establish connection

- *Output Connection String* outputs the database information required to establish a connection.

- *In-DB XML* outputs the In-DB query in XML format

- *Record Info XML* outputs the database fields in XML format

- *Query Alias List* outputs a common table expression which shows each segment of the query

- *Last Query Alias* outputs the last alias from *Query Alias List*

11.8 Filter In-DB

Figure 11-14 - Filter In-DB	The *Filter In-DB* tool creates a criteria and outputs only those records which match the criteria		
	Group	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream

Records which matches the criteria are sent to "T" (True) output of the tool whereas those which do not meet the criteria are sent to "F" (False) output of the tool.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Filter In-DB* tool allows us to select the type of filter to specify the criteria for output records.
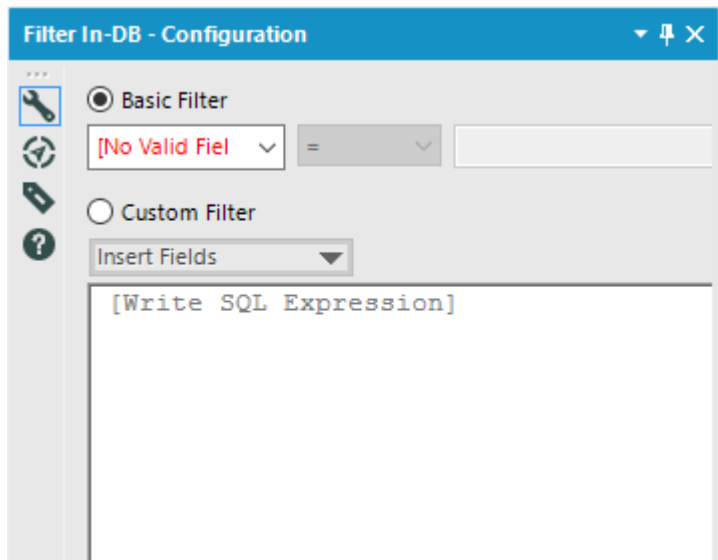
Figure 11-15 – Filter In-DB, Configuration

- *Basic Filter* allows us to write basic criteria for filtering the input records.

- *Custom Filter* allows writing custom or complex criteria in the form of a query which is not offered by the dropdown list in *Basic Filter*.

11.9 Formula In-DB

	The *Formula In-DB* tool allows the In-DB fields to be created/updated using SQL expressions based on the native database language
Figure 11-16 - Formula In-DB	**Group**	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for *Formula In-DB* tool provides us with the following set of options.
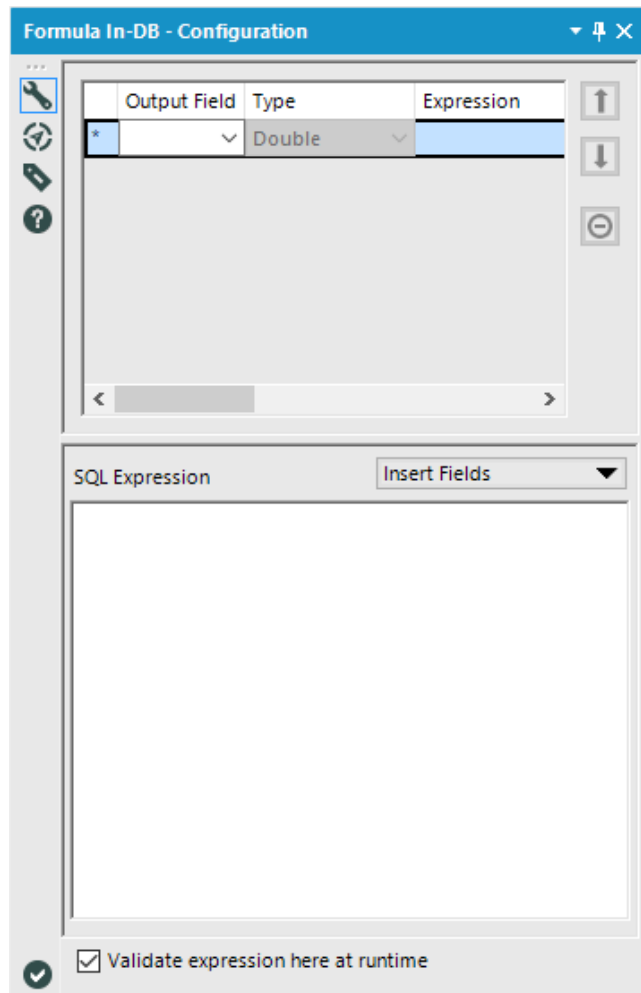
Figure 11-17 – Formula In-DB, Configuration

- *Output Field* allows to choose from a dropdown list of available fields in the In-DB stream or you can name a new field by typing into it. The field chosen here will be the one which will have the expression applied to it.
- *Type* allows selection of output data type of the field. If chosen from existing fields, this information is auto-populated and cannot be changed.

- *Size* allows specifying the width of output data. If an existing field was chosen, this property cannot be changed.

- *Expression* This box is populated by the SQL expression written in the box below. This property is for reference only.

- *Navigation Buttons* on the right side of the configurations window allows changing the order in which the expressions will be applied or to delete any existing expression from the list.

- *SQL Expression* window allows writing SQL code in the native language of the database connection. *Insert Fields* dropdown list can be used to insert fields into the SQL expression from the In-DB stream.

- *Validate expression here at runtime* option when checked, allows the query to be sent to the database and any error gets reported immediately. If unchecked, the query is not sent to the database until the workflow is executed.

11.10 Join In-DB

	The *Join In-DB* tool allows joining two database tables from In-DB stream
Figure 11-18 - Join In-DB	

Group	Input	Output
In-Database	In-DB Stream	In-DB Stream

Not all type of fields can be joined to other types of fields. In few cases, incompatible join types may result in an error/warning message.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Join In-DB* tool provides us with the following set of options.
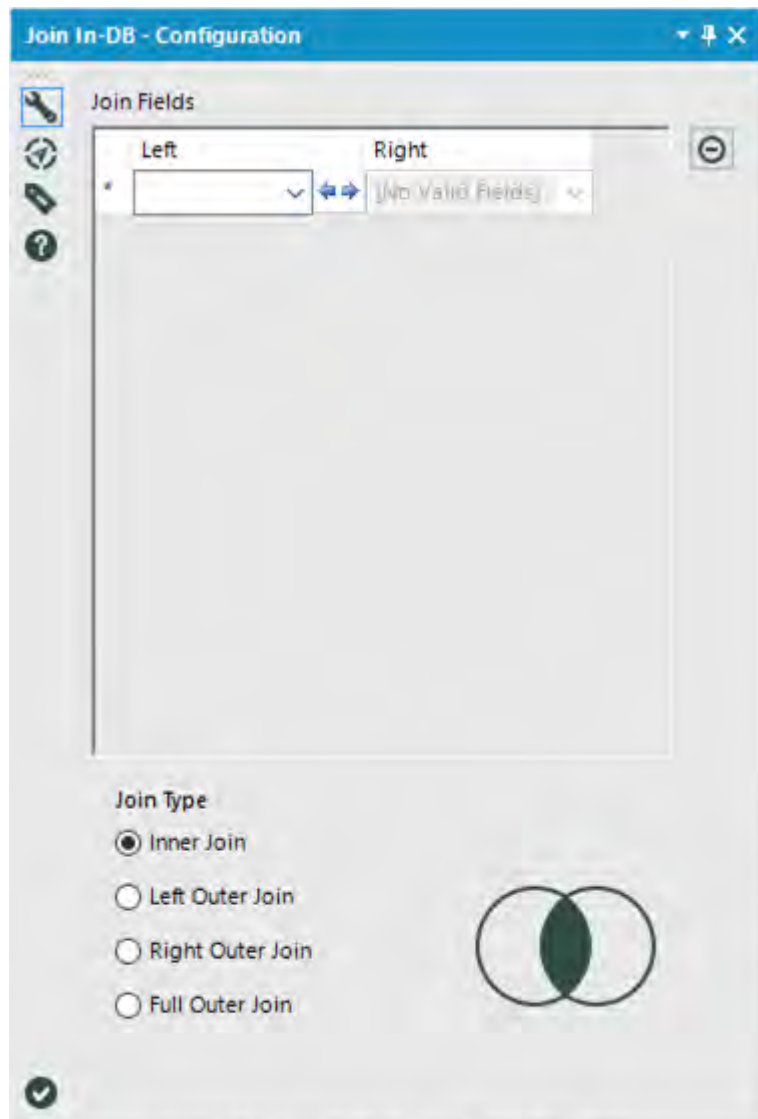
Figure 11-19 – Join In-DB, Configuration

- *Join Fields* allows choosing the field from Left and Right table for the join condition. There can be multiple pairs of fields to join which can be specified by adding more rows in the *Join Fields* property

- *Delete Button* on the right side of the *configuration* window allows removal of any join condition which is no longer required.

- *Join Type* lets us specify the type of join. There are *four* type of joins

  Inner Join returns only the records from left table which matches the records from a right table based on the join condition.

  Left Outer Join returns all records from a left table including those which matches the records from a right table based on the join condition.

  Right Outer Join returns all records from a right table including those which matches the records from a left table based on the join condition.

  Full Outer Join returns all records from both left and a right table including those which matches based on the join condition.

11.11 Macro Input In-DB
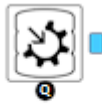
	The *Macro Input In-DB* tool is used to display In-DB input anchors on a macro tool for use in In-DB workflows.
Figure 11-20 - Macro Input In-DB	**Group**	**Input**	**Output**
	In-Database	See below	In-DB Stream

Note: An input anchor appears for each *Macro Input In-DB* tool used in the macro workflow in the same order as they were brought into the macro workflow. These input anchors can be re-ordered using the *Tree* view of *Interface Designer*.

The bottom *Black Question Anchor* can be connected to a question anchor on top of a tool to pass the result to the subsequent tool.

Input: The In-DB Connection Name and Query from the calling module's data stream.

Properties Window:

The *Configurations* window for *Macro Input In-DB* tool provides us with the following set of options.

Figure 11-21 – Macro Input In-DB, Configuration

- *Template Input* allows to use of temporary details to make sure that the resulting macro is functional and also determines the field requirement for user of the macro.

Connection Name specifies the database to connect to. The dropdown list provides the options to *Manage Connections* or *Open File Connection.*

Table or Query shows the name of the table or the query selected after setting up connection name. *Query Builder* button can be used to make any changes to this information.

- *Input Name* specifies the name which will be visible to the user of the macro.

- *Anchor Abbreviation* allows short name to be displayed on the input anchor of the macro tool

- *Show Field Map* option when checked asks the user to map the input fields with those specified in the *Template Input*.

11.12 Macro Output In-DB

Figure 11-22 - Macro Output In-DB	The *Macro Output In-DB* tool is used to display In-DB output anchors on a macro tool for use in In-DB workflows.		
	Group	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream
Note: An output anchor appears for each *Macro Output In-DB* tool used in the macro workflow in the same order as they were			

brought into the macro workflow. These input anchors can be re-ordered using the *Tree* view of *Interface Designer*.

Properties Window:

The *Configurations* window for *Macro Output In-DB* tool provides us with the following set of options.
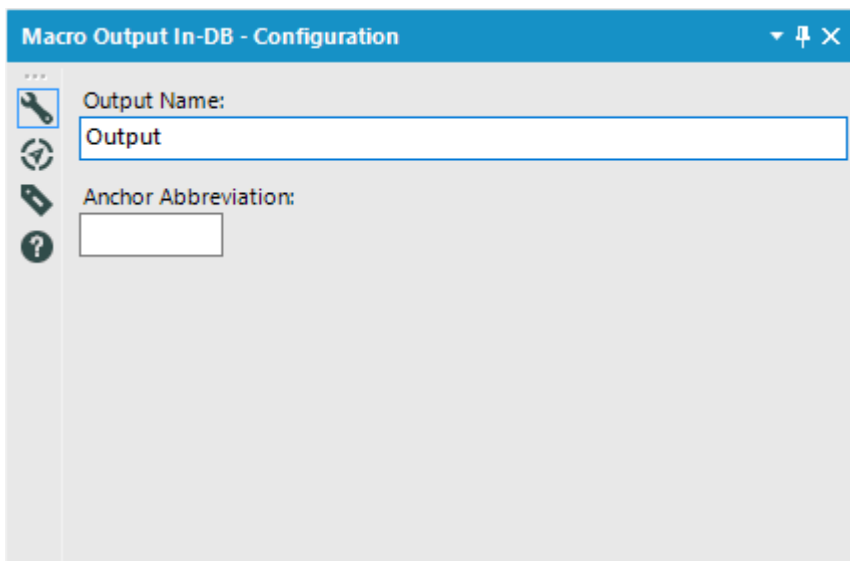
Figure 11-23 – Macro Output In-DB, Configuration

- *Output Name* specifies the output name which will be visible to the user of the macro.

- *Anchor Abbreviation* allows short name to be displayed on the output anchor of the macro tool

11.13 Sample In-DB

	The *Sample In-DB* tool allows sampling of the In-DB stream based on numbers or percentage of a total number of records.		
Figure 11-24 - Sample In-DB	**Group**	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for *Sample In-DB* tool provides us with the following set of options.
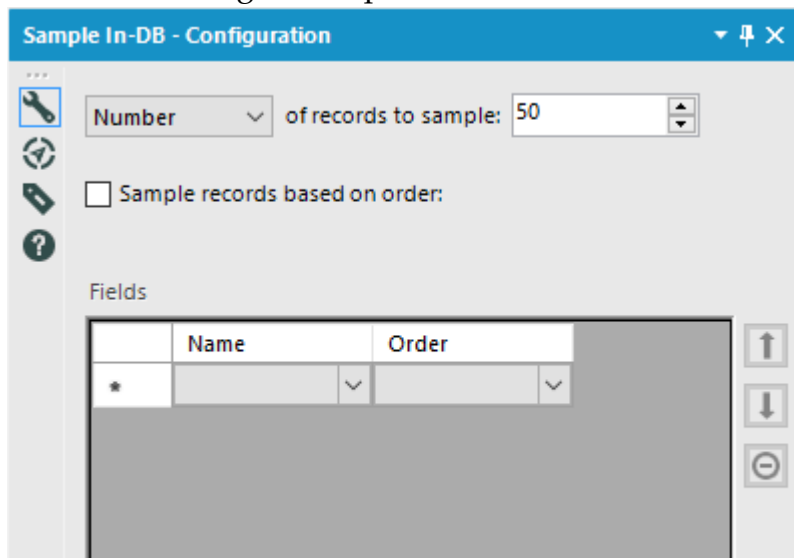
Figure 11-25 – Sample In-DB, Configuration

- *Records to sample* option allows to choose if the sampling needs to be done as a fixed number or percentage of a total number of records. The text box allows us to specify the number or percent figure for a sampling of records.

- *Sample records based on order* when checked allows records to first sorted before the sampling is done.

- *Fields* option is enabled when *Sample records based on order* option is checked. Once enabled, this section is used to specify the column names and the order of sorting to be applied to records before getting sampled.

- *Navigation Buttons* on the right side of *configuration* windows allows changing the order of fields for sorting. The *delete* button allows removal of any field from sorting that is no longer required.

11.14 Select In-DB

	The *Select In-DB* tool allows manipulation of fields in an In-DB stream by selecting, deselecting, renaming and reordering the fields.		
Figure 11-26 - Select In-DB	**Group**	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

<u>Properties Window:</u>

The *Configurations* window for *Select In-DB* tool provides us with the following set of options.
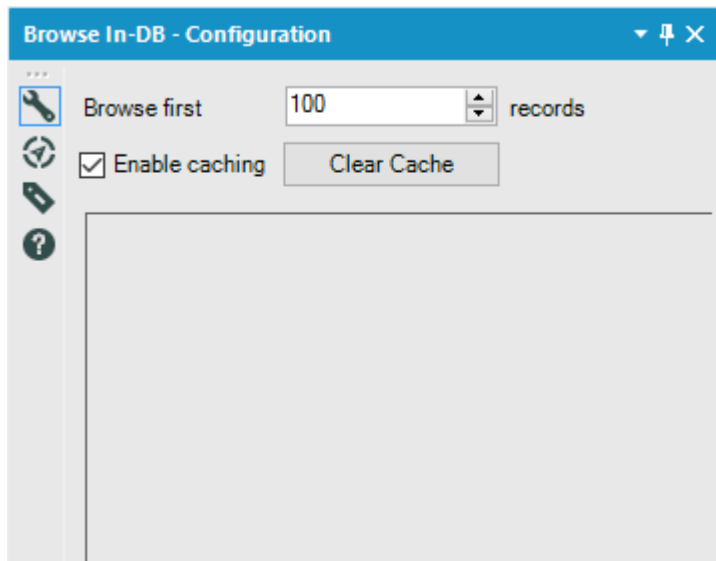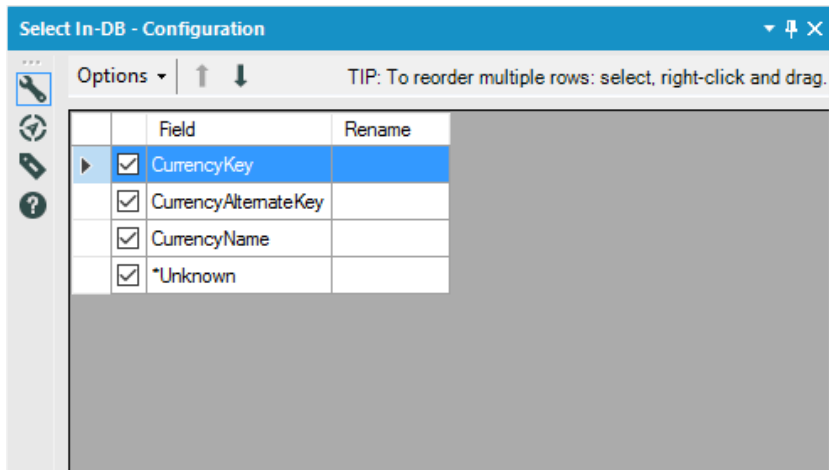

Figure 11-27 – Select In-DB, Configuration

- *Options* button on the top provides option to

  Select, Sort, Move, Add Prefix to Field Names, Add Suffix to Field Names, Remove Prefix, Remove Suffix, Clear All Renames, Clear Highlighted Renames, Forget All Missing Fields, Forget Highlighted Missing Fields, Deselect Duplicate Fields.

- *Arrow Keys* to change the order of the fields.

- *Rename* box allows to directly rename the field.

11.15 Summarize In-DB

	The *Summarize In-DB* tool summarize the input data using different aggregate functions.		
Figure 11-28 - Summarize In-DB	**Group**	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream

The output contains only the fields which are part of the calculation in the tool and not the fields which were part of the input.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Summarize In-DB* tool provides us with the following set of options.
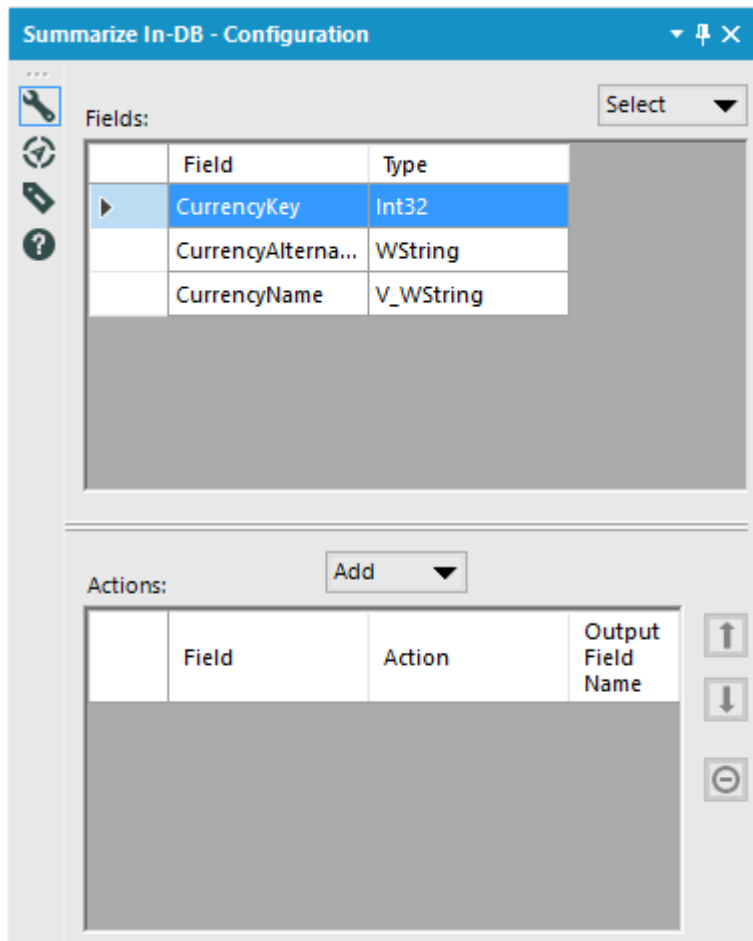
Figure 11-29 – Summarize In-DB, Configuration

- *Fields* section lists all the fields that are part of the input In-DB stream along with their data types.

- *Select* button on top right corner of *configurations* window allows selecting fields for aggregation operation
  All, None, Numeric, String, Spatial

- *Add* button allows adding aggregate operation on the selected fields. The list of functions includes

  Group By, Count, Count Distinct, Count Non-Null, First, Last, Min, Max

  Numeric is a group of functions which can be applied to fields with numeric or related data types only. These are
  Sum, Average, Standard Deviation, Variance

- *Actions* window allows us to select a field which was added using the *add* button earlier. Once selected we can use *Up* or *Down* arrow buttons to change the order of output or use *Delete* button to remove the field from output. We can also change the action selected earlier or rename the output field.

- *Action Properties* section is available for *Concatenate String* or *Finance* actions. It specified additional properties for such actions.

11.16 Union In-DB

	The *Union In-DB* tool combines two or more In-DB streams of data which have a similar structure.		
Figure 11-30 - Union In-DB	**Group**	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream

The structure of the In-DB streams for *Union In-DB* tool need to be similar in structure.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Union In-DB* tool provides us with the following set of options.
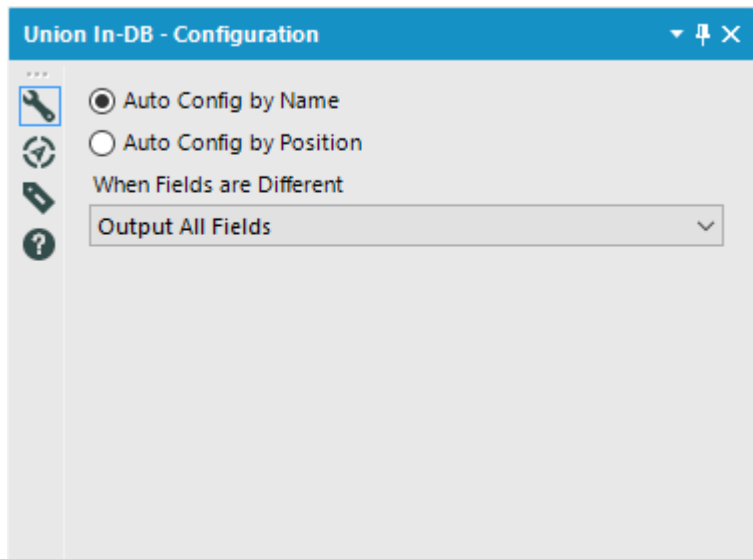
Figure 11-31 – Union In-DB, Configuration

- *Auto Config by Name* option combines fields having the same name into a single field

- *Auto Config by Position* option combines fields by their order in the input streams

- *When Fields are different* option allows us to specify the action required to be taken if a field does not matches by Name/Position
Error – Stop Processing, Output All Fields, Output Common Subset of Fields

11.17 Write Data In-DB

Figure 11-32 - Write Date In-DB	The *Write Data In-DB* tool is used create or update a database table directly.		
	Group	**Input**	**Output**
	In-Database	In-DB Stream	In-DB Stream

The *Write Data In-DB* tool outputs the same In-DB stream as the input In-DB stream.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Write Data In-DB* tool provides us with the following set of options.
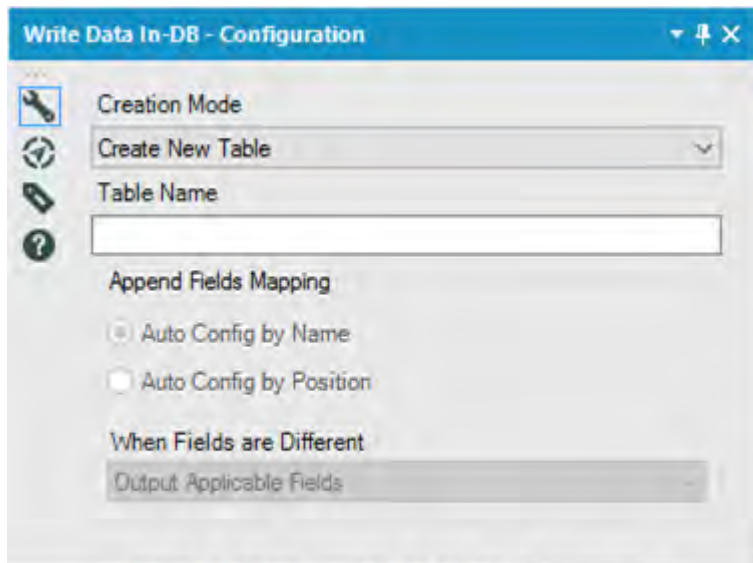
Figure 11-33 – Sample In-DB, Configuration

- *Creation Mode* specifies the option for writing data into the database.

  Append Existing, Delete Data & Append, Overwrite Table (Drop), Create New Table, Create Temporary Table

- *Table Name* option allows specifying the table name to be created or for data to be appended to. If *Create Temporary Table* option is selected in *Creation Mode,* then this value is defaulted to *[a unique temporary table name is generated on each run].*

- *Append Field Mapping* option is enabled if *Append Existing* or *Delete Date & Append* options are selected in *Creation Mode*. This option specifies how the data should be appended to existing data

Auto Config by Name, Auto Config by Position

- *When Fields are Different* option allows us to specify the action required to be taken if an append fields do not match by Name/Position
  Error – Stop Processing, Output Applicable Fields

11.18 Calgary

Calgary is a list count data retrieval engine. It is primarily used for analysis for huge datasets with millions of records.

Calgary makes use of database indexes to enable faster retrieval of data. Such indexes can be created on more or more columns from the dataset.

The number of records which *Calgary* database can handle is 2^{31} (approx. 2 billion). Considering this figure, the Calgary database works best with a dataset of 100-300 million records. In the case of more than 300 million records, the 64-bit version of Alteryx engine works better.

11.19 Calgary Input

	The *Calgary Input* tool is used to read an existing Calgary Data File and output the data as per the specified query.		
Figure 11-34 - Calgary Input	**Group**	**Input**	**Output**
	Calgary	Data File	Data Stream

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Calgary Input* tool provides us with the following set of options.
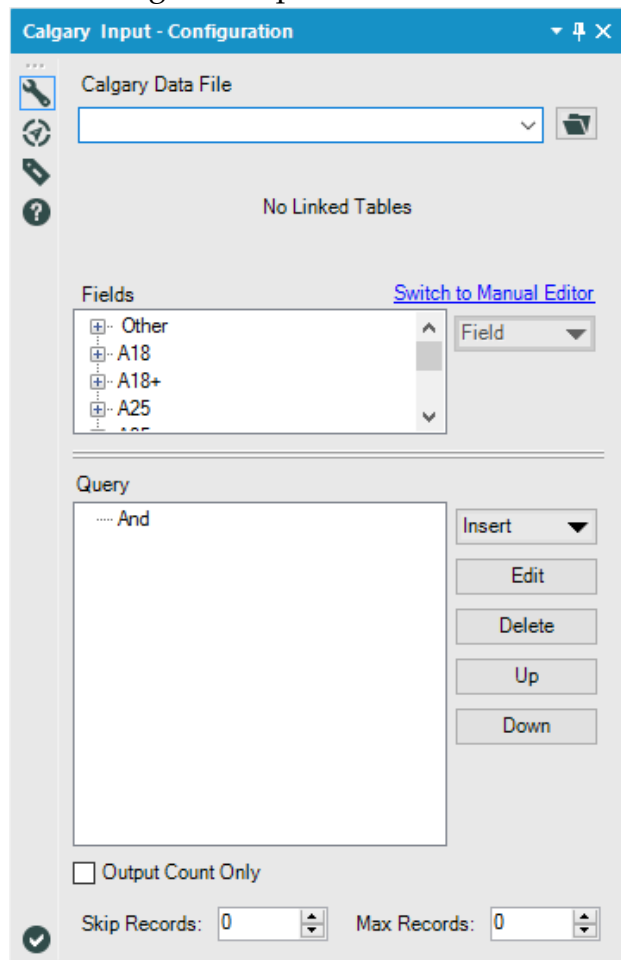
Figure 11-35 – Calgary Input, Configuration

- *Calgary Data File* specifies the Calgary database file (*.cydb* file) to read the data from. One can type into the textbox or use the browse button on the right to locate the file.
- *Linked Data File* section shows if the selected Calgary database file (primary) is linked to another Calgary database file (secondary). In such a case, there are different *Linked Options* for returning records for *Query* criteria.
- *Fields* section shows a list of all the fields which are indexed and can be used in the query. The fields are arranged in a tree structure and grouped together based on their name. For example, *Customer_First_Name*, *Customer_Middle_Name* and *Customer_Last_Name* will be grouped under *Customer* node.
- *Edit Query Item* interface can be accessed by either double-clicking on a field or by using the *Field* button on the right after clicking on the field.
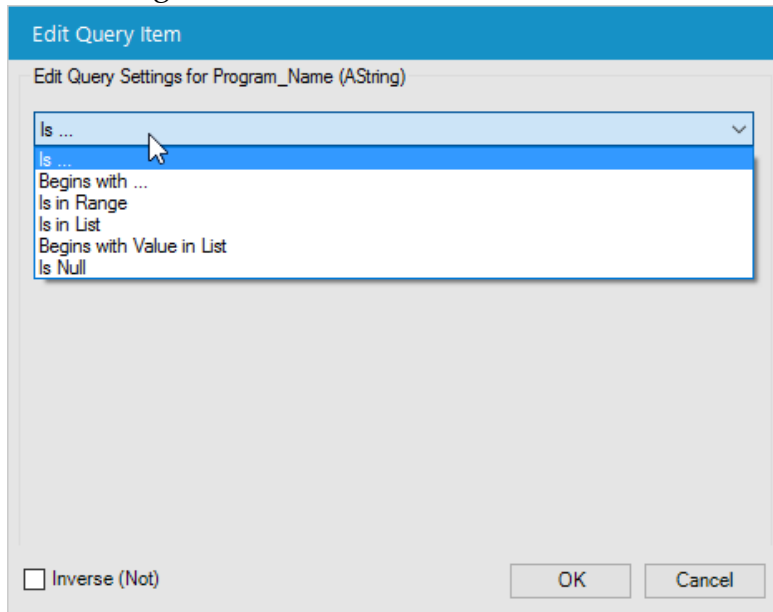
Figure 11-36 – Calgary Input, Edit Query Item

Field button can have a downward arrow if there is a predefined query specified using *Meta Info Editor* interface. If not using a predefined query, one can select the *Custom* option in the *Field* button to access *Edit Query Item* interface.

To access *Meta Info Editor* interface, please select and right-click any field in the *Fields* section and then select *Edit Index MetaInfo*.

- *Query* area displays the query created using *Edit Query Item* interface. This section allows reorganizing the different filters using the button on the right i.e. *Edit*, *Delete*, *Up* and *Down.*

- *Output Count Only* checkbox when checked will return the count of all records matching the Calgary query criteria.

- *Skip Records* option allows the specified number of rows to be removed from the output of the Calgary query.

- *Max Records* option limits the number of records returned by the Calgary query as the number specified.

- *Switch to Manual Editor* option allows constructing the Calgary database query using XML editor.

11.20 Calgary Join

	The *Calgary Join* tool is used to return records after joining data from an Input file with an existing Calgary database		
Figure 11-37 - Calgary Join	**Group**	**Input**	**Output**
	Calgary	Data File	Data Stream

Calgary Join tool outputs all the fields from the Input file and the Calgary database file. In order to select specific fields, *Select* tool must be used.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Calgary Join* tool provides us with the following set of options.
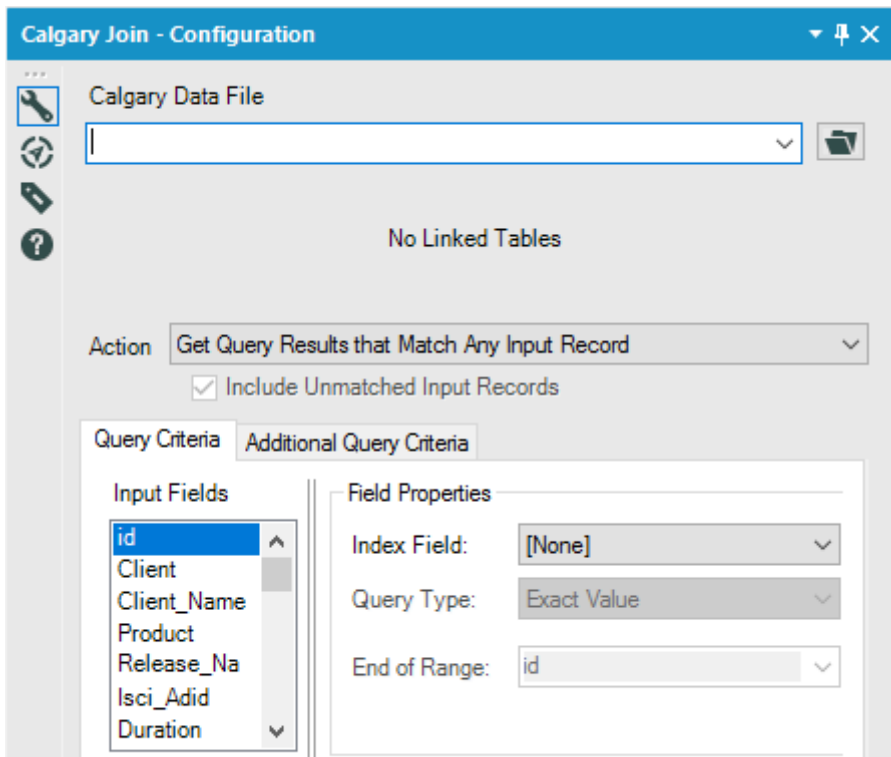
Figure 11-38 – Calgary Join, Configuration

- *Calgary Data File* specifies the Calgary database file (*.cydb* file) to join the data from. One can type into the textbox or use the browse button on the right to locate the file.

- *Linked Data File* section shows if the selected Calgary database file (primary) is linked to another Calgary database file (secondary). In such a case, there are different *Linked Options* for returning records.

- *Action* list specifies 6 different type of outputs and we need to select the one required by the *Calgary Join* tool. The list includes

Count Query Results that Match ANY Input Record
Get Query Results that Match ANY Input Record
Count Query Results that Match ALL Input Records
Get Query Results that Match ALL Input Records
Append Query Count to Each Input Record
Join Query Results to Each Input Record

- *Query Criteria* tab includes the option to select

  Input Fields to select a field from Input file for the join condition.

  Field Properties to specify the field properties from Calgary database file for the join condition.

  Index Fields is meant for selecting the field from Calgary database for the join condition.

  Query Type specifies the type of join to perform. End of Range specifies the end of range value for Range type

- *Additional Query Criteria* tab is used the filter the Calgary database file similar to how this is done in *Calgary Input tool*.

11.21 Calgary Loader

Figure 11-39 - Calgary Loader	The *Calgary Loader* tool is used to create a new Calgary database file.		
	Group	**Input**	**Output**
	Calgary	Data Stream	.cydb File

Each field included in the Calgary database file can be indexed for performance.

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.

Properties Window:

The *Configurations* window for *Calgary Loader* tool provides us with the following set of options.
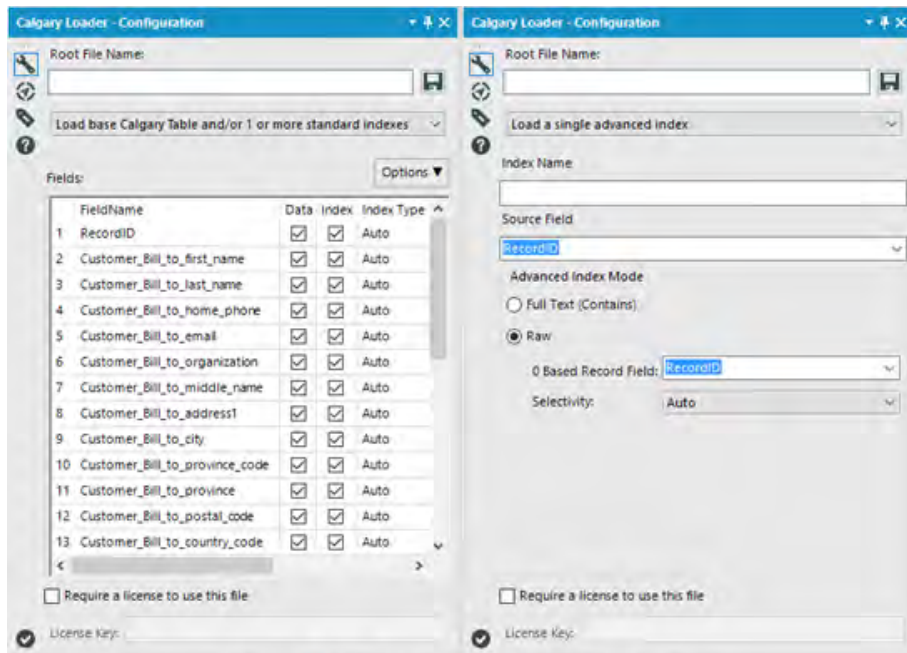
Figure 11-40 – Calgary Loader, Configuration

- *Root File Name* specifies the name of the Calgary database file (.cydb file) to be created. One can type into the textbox or use the save button to browse a folder location for saving the file. There is an option to either choose to *Load base Calgary Table and/or 1 or more standard indexes* or *Load a single advanced index*.

- *Load base Calgary Table and/or 1 or more standard indexes* enables *Fields* section which contains a list of all the fields from the input data. The options button on the right enables *Select All/Deselect All* fields from the list for the purpose of *Data* or *Index*.

- *Index Type* column in the field section allows to specify the type of index to be created on the field. The options include *Auto*, *Low Selectivity*, *High Selectivity*.

- *Load a single advanced index* enables the options to specify the *Index Name*, *Source Field,* and *Advanced Index Mode*.

- *Advanced Index Mode* section allows choosing the type of index to be created among the options of *Full Text (Contains)* or *Raw* index. For *Raw* Indexes, the *0 Based Record Field* specifies the *Record ID* and the *Selectivity* (High, Low or Auto).

11.22 Cross Count

	The *Cross-Count* tool is used to aggregate Calgary data by grouping on a set of fields to return the count.		
Figure 11-41 - Cross Count	**Group**	**Input**	**Output**
	Calgary	Calgary DB	Data stream
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

Properties Window:

The *Configurations* window for *Cross Count* tool provides us with the following set of options.
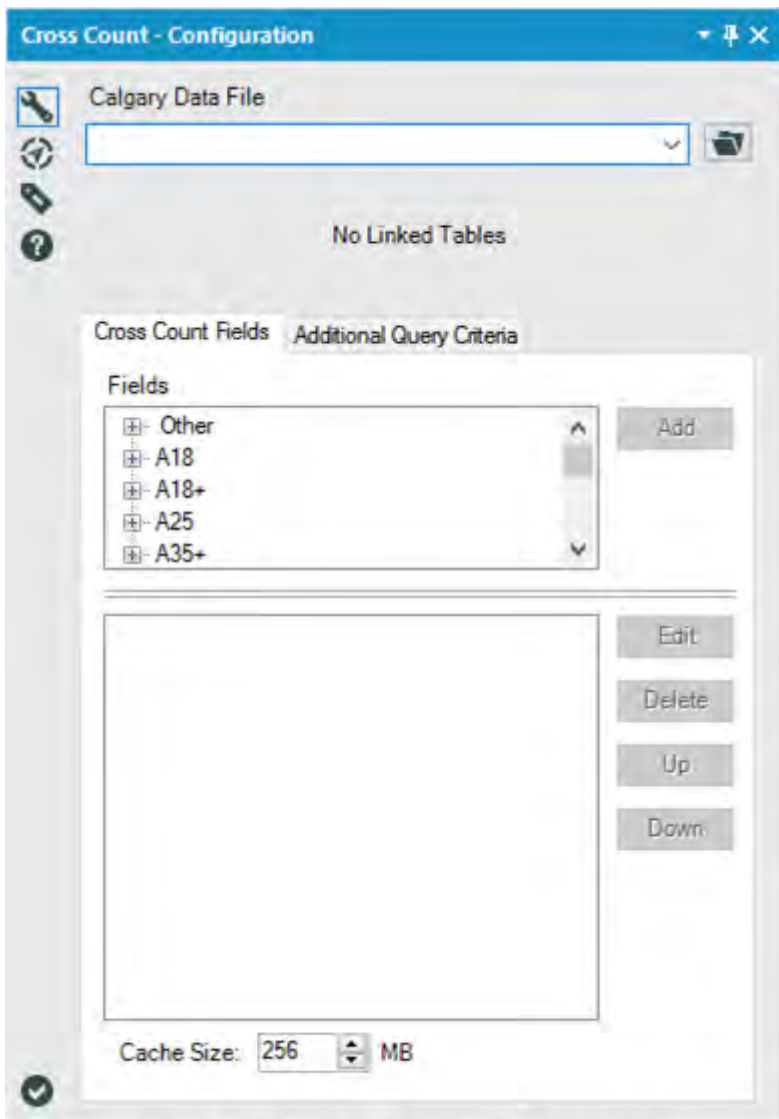
Figure 11-42 – Cross Count, Configuration

- *Calgary Data File* specifies the Calgary database file (*.cydb* file) to join the data from. One can type into the textbox or use the browse button on the right to locate the file.

- *Linked Data File* section shows if the selected Calgary database file (primary) is linked to another Calgary database file (secondary). In such a case, there are different *Linked Options* for returning records.

- *Cross Count Fields* tab includes options to select the fields from *Fields* section and then add then using the *Add* button to aggregate data on that field. The *Edit* button allows specifying *Custom CrossCount Field* to aggregate the data. The *Delete*, *Up* and *Down* buttons are used to remove or reorganize the order of the aggregation of data.

- *Cache Size* specifies the size of Cache for after retrieval of *Cross Count* data. This value is 256 MB by default.

- *Additional Query Criteria* tab is used the filter the Calgary database file similar to how this is done in *Calgary Input tool*.

11.23 Cross Count Append

	The *Cross Count Append* tool is used to append count to the input data which joins with Calgary database.		
Figure 11-41 - Cross Count	**Group**	**Input**	**Output**
	Calgary	Data Stream	Data Stream
An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.			

<u>Properties Window:</u>

The *Configurations* window for *Cross Count Append* tool provides us with the following set of options.
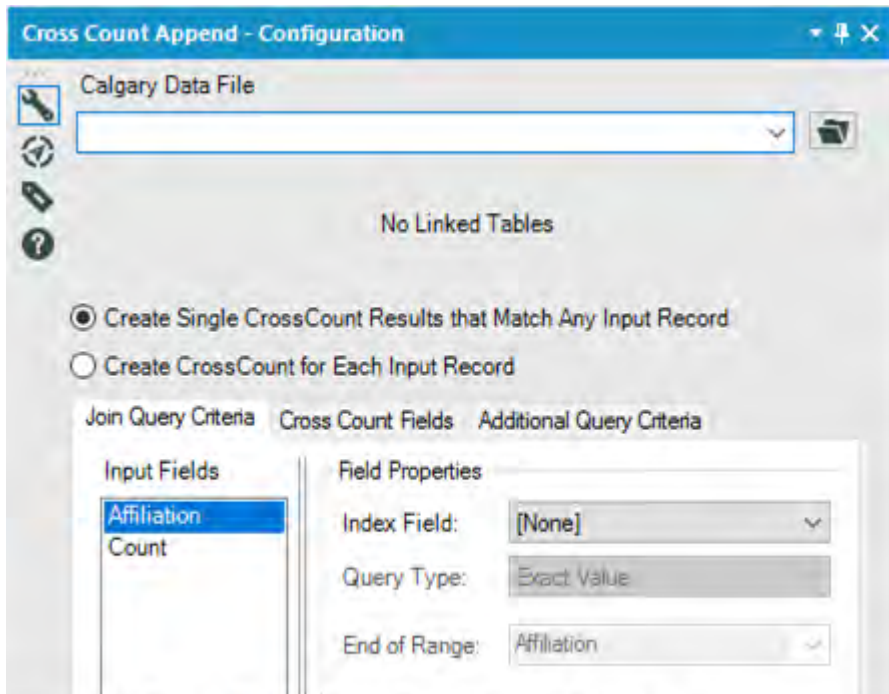
Figure 11-44 – Cross Count Append, Configuration

- *Calgary Data File* specifies the Calgary database file (*.cydb* file) to join the data from. One can type into the textbox or use the browse button on the right to locate the file.

- *Linked Data File* section shows if the selected Calgary database file (primary) is linked to another Calgary database file (secondary). In such a case, there are different *Linked Options* for returning records.

- *Create Single CrossCount Results that Match Any Input Record* and *Create CrossCount for each Input Record* are two type of outputs which can be generated by the tool. Only one of these options can be selected.

- *Join Query Data* tab provides options to select

   Input Fields to select a field from the input file to join with the Calgary database.

   Query Type specifies the type of join for joining Input file and Calgary database.

   End of Range defines the end of range value which is applicable to Range query types.

- *Cross Count Fields* tab includes options to select the fields from *Fields* section and then add them using the *Add* button to aggregate data on that field for getting the count of records per group. The *Edit* button allows specifying *Custom CrossCount Field* to aggregate the data. The *Delete*, *Up* and *Down* buttons are used to remove or reorganize the order of the aggregation of data.

- *Cache Size* specifies the size of Cache for after retrieval of *Cross Counted* data. This value is 256 MB by default.

- *Additional Query Criteria* tab is used the filter the Calgary database file similar to how this is done in *Calgary Input tool*.

11.24 Superstore Database Report

	From ▼	Google Apps
	To...	Alteryx Consultants
Send	Cc...	
	Subject	Superstore Database Report
	Attached	Product Info.csv 160 KB

Hey,

Superstore Inc. would like to generate data which shows Top 10 products by order quantity for Internet Sales since 2006. The data should also include the year in which this product achieved its highest sales.

However, there's a catch. While their Fact table of Internet Sales is in a SQL Server database, the Product information is available as a file (attached).

Customer would like to have the output in two different ways.
1. As a CSV file for use by other systems.
2. As a table on the SQL Server database where the fact table resides.

Please implement the logic using Alteryx and let us know when this is ready.

Alteryx allows us to use data from different databases using In-database tools the database being SQL Server in this case. Further, the functionality can be extended by importing data from a file input and using it alongside data from the database to produce desired results.

We are going to approach our solution by using *Connect In-DB* tool to connect to the Fact table from SQL Server and *Input Data* tool to connect to the flat file which contains Product information.
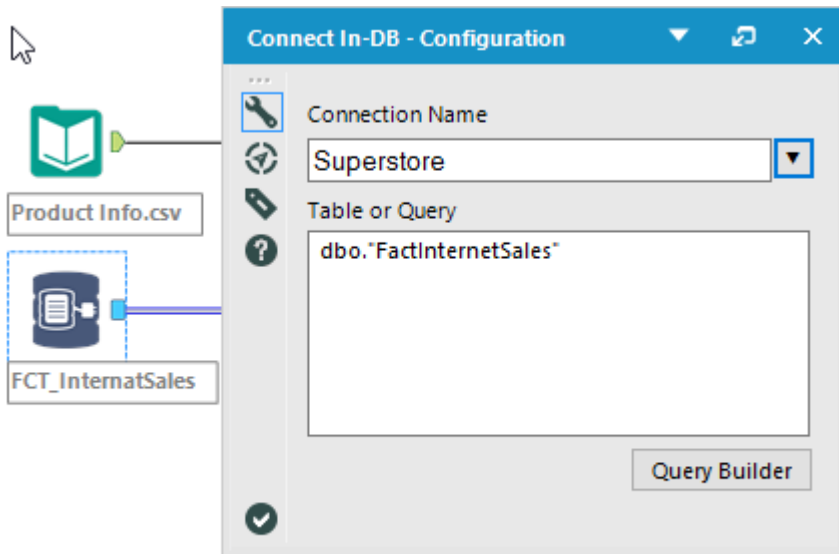
Figure 11-45 – Superstore Data, Connect In-DB

We now already know from the email that the data from the database needs to be considered for 2006 onwards hence we need to make use of *Filter In-DB* tool to restrict the records. Also, we need to use a *Data Stream In* tool to convert the file input stream to an In-DB stream. For the purpose of this exercise, we are going to use the Creation Mode as Create Temporary Table.
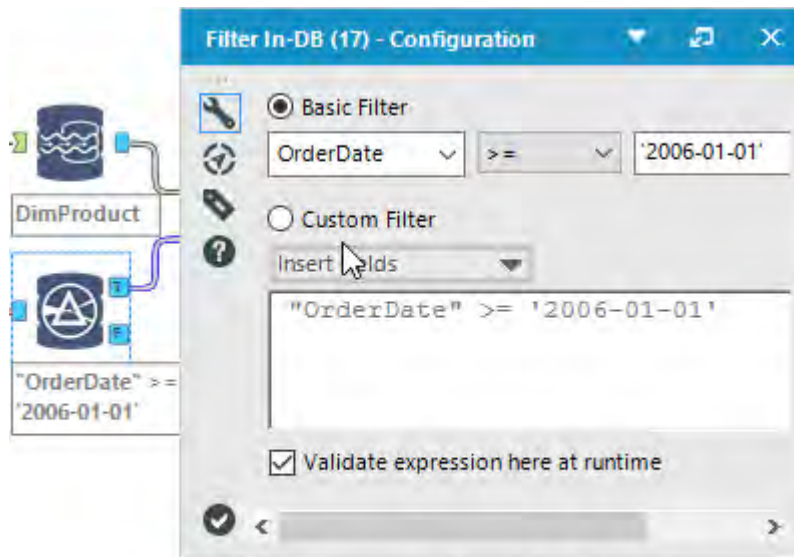
Figure 11-46 – Superstore Data, Filter In-DB

Before we can go any further, we need to understand how are we going to get the Product Name into the data stream since this column is not part of the Fact table. However, Product Info.csv file is having a Product Name column. The common column between the database table and the file is ProductKey. Hence, we need to join the two data streams by using a *Join In-DB tool* and defining ProductKey for the join condition.
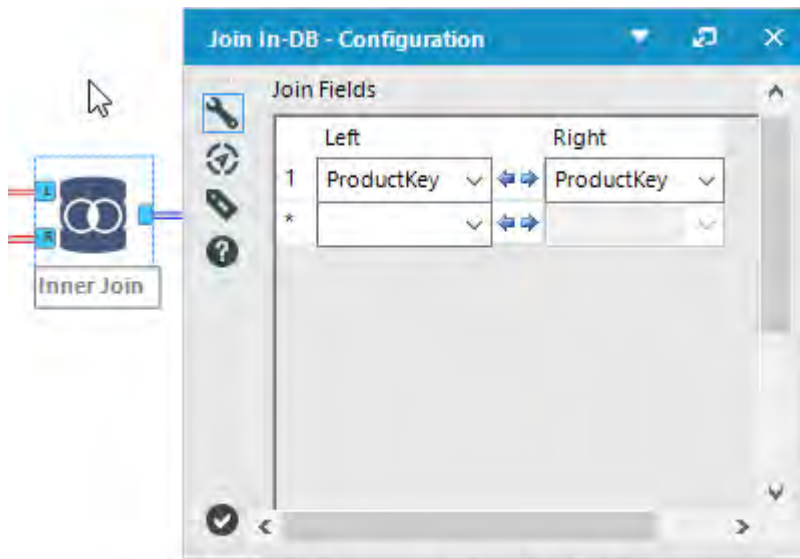
Figure 11-47 – Superstore Data, Join In-DB

From here on, we are going to use only a few columns compared to the output of *Join In-DB* tool. We will use *Select In-DB* tool to select only the relevant columns i.e. *EnglishProductName* renamed as *ProductName*, *OrderQuantity*, and *OrderDate*. Once selected, we will use the *Formula In-DB* tool to calculate the year of *OrderDate*. For the purpose of calculation, we'll use *YEAR("OrderDate")* as the SQL expression in Formula In-DB tool. We'll rename the new field as *OrderYear* and keep the *Type* as *Double*.
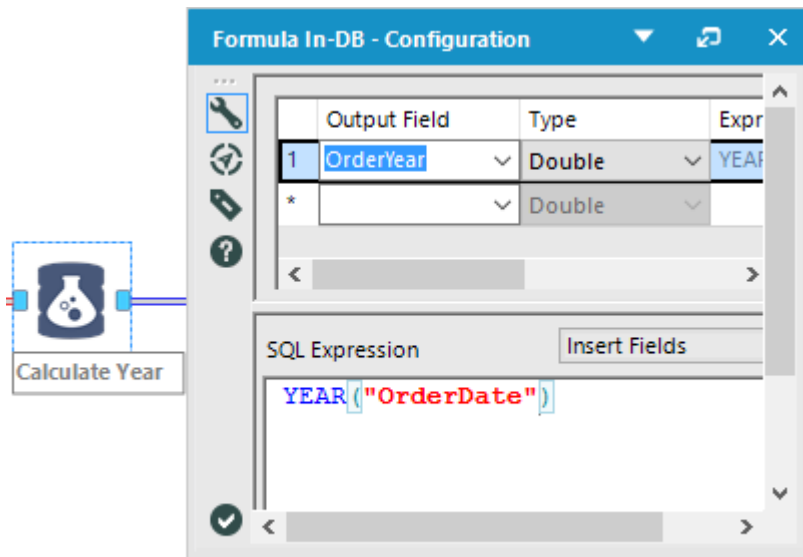
Figure 11-49 – Superstore Data, Formula In-DB

With the Calculate field of *OrderYear* in place, we are now ready to aggregate *OrderQuantity*. For this purpose, we have to make use of *Summarize In-DB* tool. Add *GroupBy* action for *ProductName* and *OrderYear* while we'll be adding *Sum* action from the *Numeric* section for *OrderQuantity* field. For *Output Field Name*, keep the auto generated field names intact.
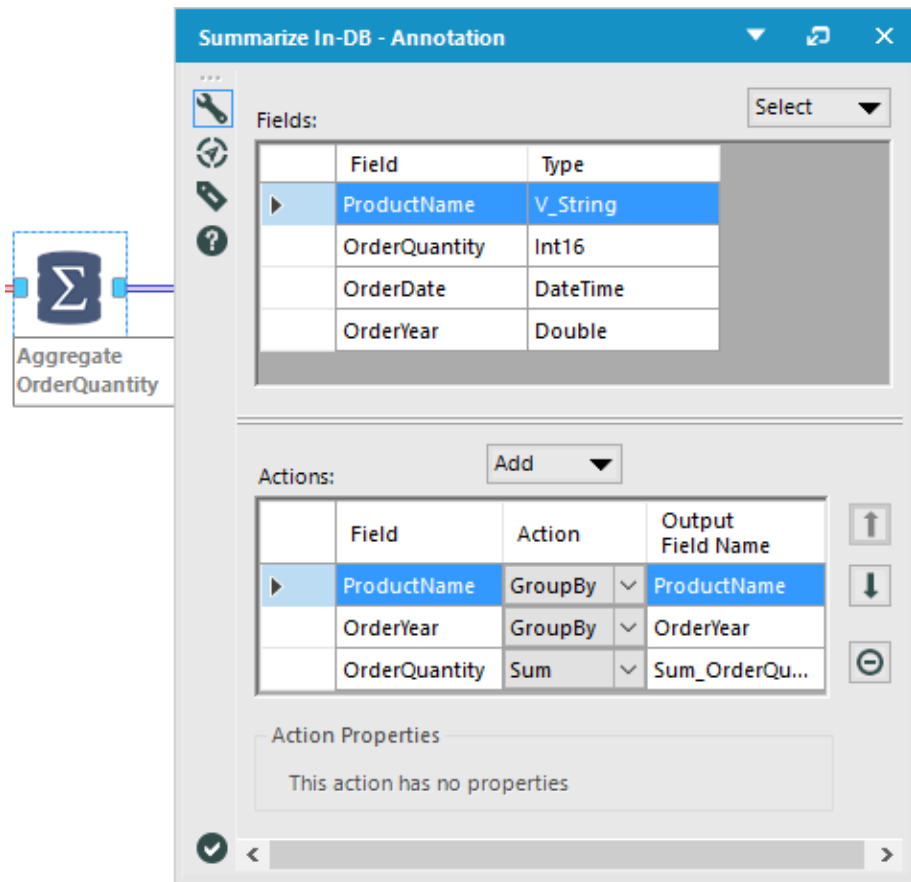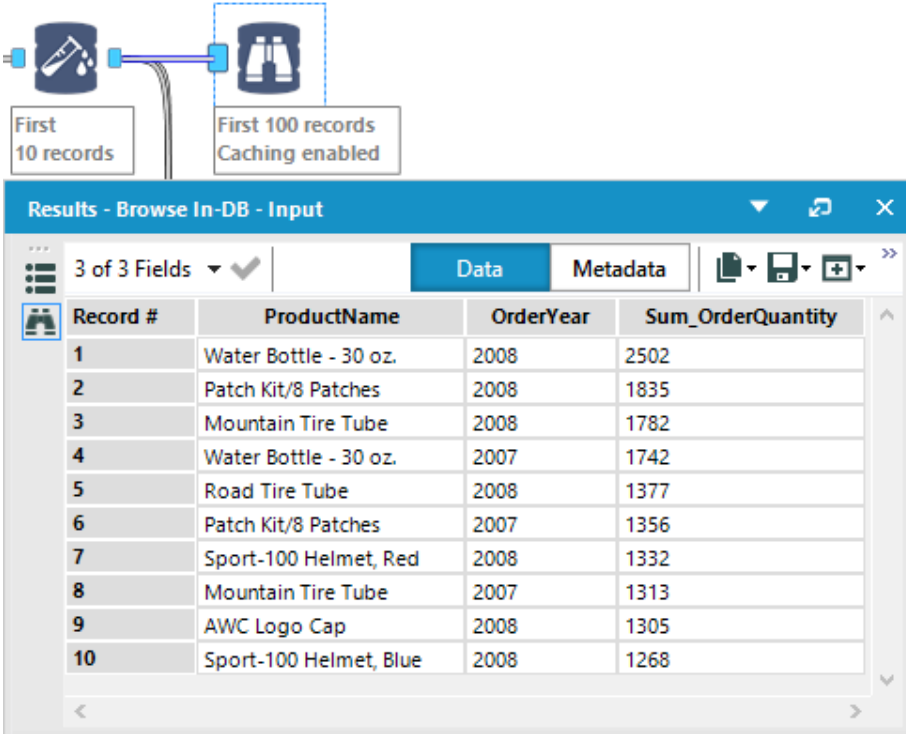
Figure 11-50 – Superstore Data, Summarize In-DB

Now we need to select only the records with Top 10 order quantities. In order to achieve this, we first need to sort the records in descending order of the sum of order quantity and then select top 10 records from it. Make use of *Sample In-DB* tool for this purpose by specifying *Number* of records to sample as *10*. Also, we need to check the *Sample records based on order* checkbox. Afterward, select *Sum_OrderQuantity* under *fields* section and choose *Descending* as its order.

At this stage, the data should be ready and as per the requirements. However, to confirm this, we have to add *Browse In-DB* tool and then run the workflow to see the output in the *Results* window.

First
10 records

First 100 records
Caching enabled

Record #	ProductName	OrderYear	Sum_OrderQuantity
1	Water Bottle - 30 oz.	2008	2502
2	Patch Kit/8 Patches	2008	1835
3	Mountain Tire Tube	2008	1782
4	Water Bottle - 30 oz.	2007	1742
5	Road Tire Tube	2008	1377
6	Patch Kit/8 Patches	2007	1356
7	Sport-100 Helmet, Red	2008	1332
8	Mountain Tire Tube	2007	1313
9	AWC Logo Cap	2008	1305
10	Sport-100 Helmet, Blue	2008	1268

Figure 11-51 – Superstore Data, Sample In-DB, Browse In-DB

Now that the data in the Results window appears as per the expected requirements, it is now ready for the output. We need to output the results in two formats. First, a CSV file to be saved locally and secondly, as a database table. In order to do this, we are going to add *Data Stream Out* and *Write Data In-DB* tool next to *Sample In-DB tool*. While *Data Stream Out* does not need any configuration changes, for *Write Data In-DB*, select

Overwrite Table (Drop) as Creation Mode and mention the table name as *ProductSummary*. Add *Output Data* tool next to *Data Stream Out* tool to write output data to a CSV file. Name the file as *ProductSummary.csv*

The completed workflow would look something like the below image.
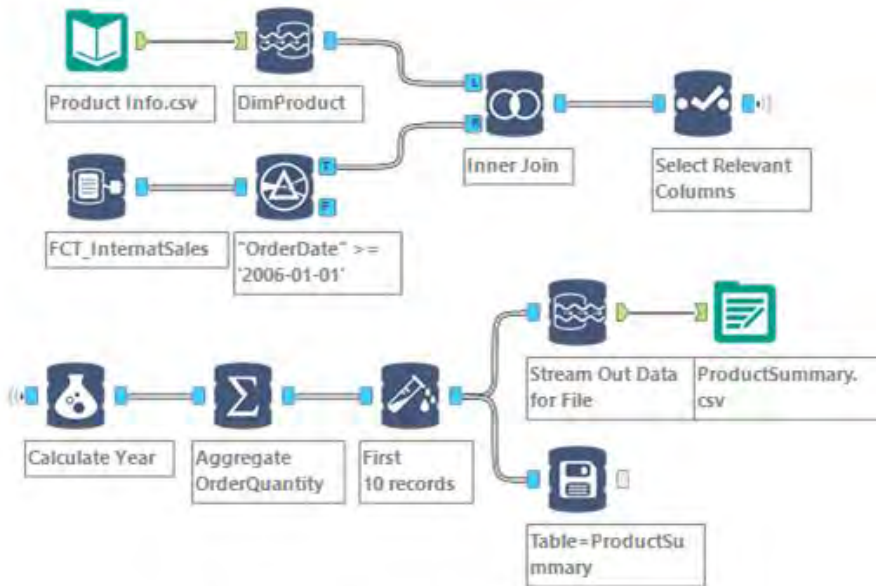
Figure 11-53 – Superstore Data, Complete Workflow

CHAPTER 12
Green On The Go

	To...	Alteryx Consultants
Send	Cc...	
	Subject	Tesla

Hey,

A courier delivery services client is considering going green. So to reduce their carbon footprint, they are planning to introduce Tesla cars for their delivery services.

Before cars are introduced, they want to provide the drivers' information about all the Tesla Supercharger station. So that the drivers are well aware of nearest station, options to charge based on their delivery route etc on their company mobile app.

First they would want Tesla Supercharger station information to be updated periodically from the Tesla website to their database and Also they want data from social media like Twitter to see what is trending about the Tesla. Here is the link to get the Supercharger details:
https://www.tesla.com/findus/list/superchargers/United+States

I have heard that Alteryx is good in doing spatial and geo analysis, to get started could you please help me out in getting this data for the analysis.

Thanks

12.1 Tools & Concepts

Tools	Concepts
Tools	**Concepts**
Download	Retrieve data from cloud
Amazon-S3 Download	Retrieve data from internet/intranet environment
Amazon-S3 Upload	
Google Analytics	Push data to the cloud
Foursquare Search	Push data to the internet /intranet environment
Marketo Append	
Marketo Input	
Marketo Output	
MongoDB Input	
MongoDB Output	
Salesforce Input	
Salesforce Output	
SharePoint List Input	
SharePoint List Output	
Twitter Search	

12.2 Download

Figure 12.1– Download	The *Download* tool will retrieve data from a specified URL to be used in downstream processing or to be saved to a file.		
	Group	**Input**	**Output**
	Connectors	URL	Data field or output to a file

The *Download* tool can also download or upload data via FTP and SFTP.

Note: To avoid connection delays when using the Download tool, ensure that the "Automatically detect settings" option is deselected within your account's internet properties (Control Panel > Internet Options > Connections > LAN Settings).

An *Action* tool can be connected to the *Lightning Bolt Anchor* to modify how this tool works in apps and macros.*:*

Properties Window:

There are four tabs with configurable options on the *Download* tool: *Basic*, *Headers*, *Payload* and *Connection*. Only the *Basic* tab is required for proper configuration.
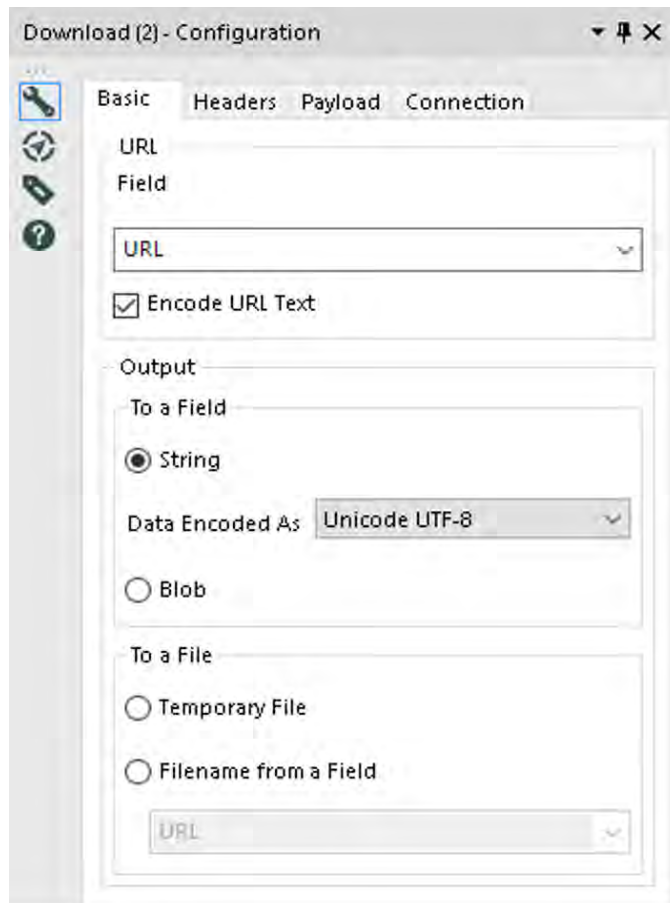
Figure 12.2 – Download Configuration

- *URL Field*: Specify the field from the incoming data stream that contains the URL to pull data from.

- *Encode URL Text*: When checked, the specified URL will be encoded as needed where unsafe ASCII characters are converted into a format that can be transmitted over the internet. An example of this would be the substitution of %20 for a space.

Output: Specify how the returned data should be formatted. The data can be returned in a data field or output to a file.

- *To a Field*: Downloaded content is returned in the data stream as a data field. The downloaded contents will be in a single field called "DownloadData". You will likely have to parse this data using downstream tools such as the Text to Columns, RegEx, or Formula tools.

  o String: Data is returned as a new wide string type field. A wide string supports Unicode characters.
  o Blob: Data is returned as a new blob type field. Blob is also known as Binary large objects. Image files are usually stored in this format. To use the image, configure a Report Image tool downstream and specify the Image or Blob field.

- *To a File:*

  o *To a Temporary File*: Data is output to a temporary file and will be located in the user's temporary directory. For more on temp file handling in Alteryx, review the Temp Files page.
  o *Filename from a Field*: Data is output to a specific file where the file specification is in an incoming field. Use the dropdown to select the field that contains the file name to output to.

The *headers tab* allows you to modify the HTTP headers sent with the web request.
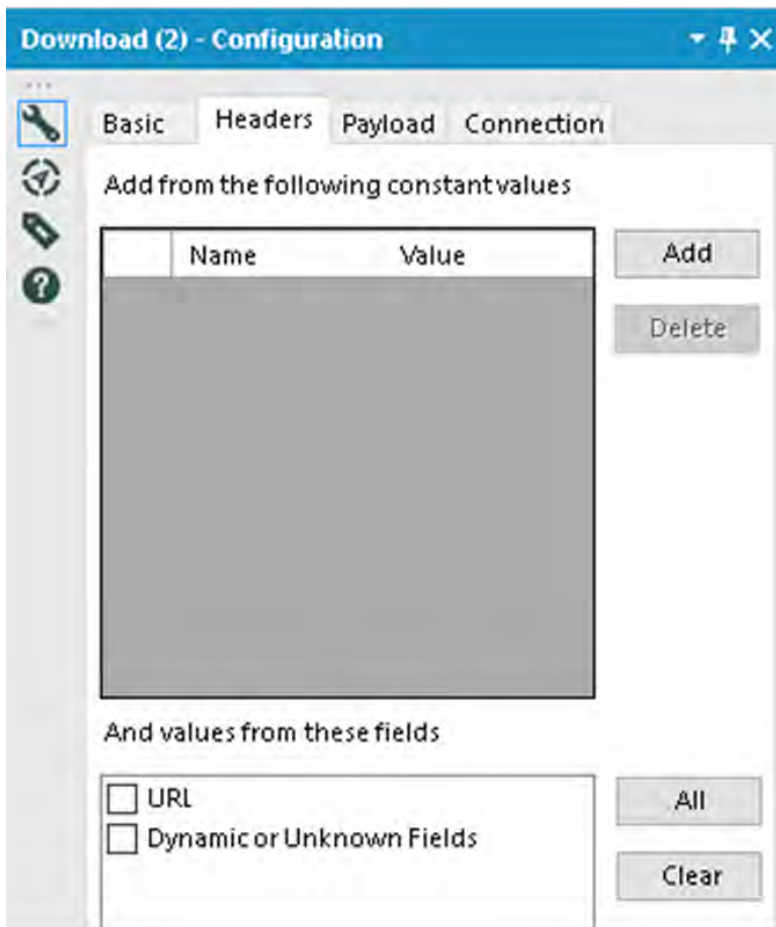
Figure 12.3–Download Configuration

- *Add from the following constant values*: Allows adding fixed header values.
- *And values from these fields*: Takes values from the record data and creates header values.

The *payload tab* allows you to set the HTTP Action you would like to perform and optionally build the Query String or Body for the web request.
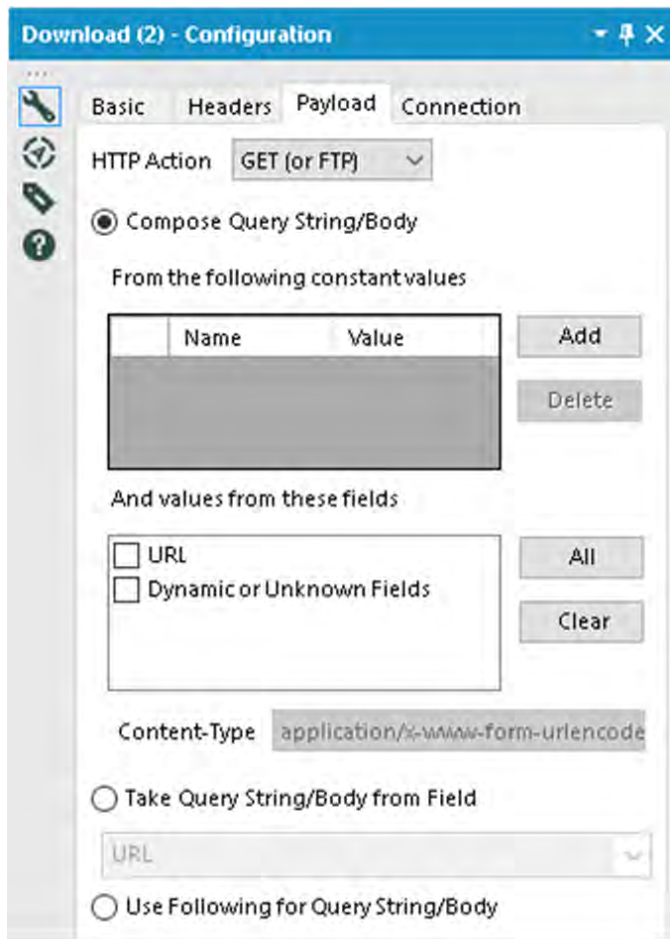
Figure 12.4–Download Configuration

- HTTP Action: Select the HTTP Action for the web request. Choices include:
  - ○ GET (or FTP): Perform a GET request or download a file from an FTP or SFTP site. This option can be used without entering any other options on this tab and will download the selected URL.

- o POST: Performs a POST request to the selected URL. Typically when using this option you would specify a POST body using the below options.
- o PUT: Performs a PUT request. This option only allows you to take the query body from a blob field via the Blob Input tool. Typically used to upload a file to the remote server.
- o DELETE: Performs a DELETE request. Typically used to ask the remote server to delete a specified resource.
- o HEAD: Performs a HEAD request. This asks the server to return the header data, but not the body data.
- o Custom: Allows you to enter a custom verb in a text box. Everything else about this request behaves the same as a POST. The POST verb is replaced by the custom verb just before the request is made.

These options will only work if the remote server (that you are sending the requests to) supports a particular verb. Check the API documentation of the URL you are using to see what requests are supported.

Choose from the following Query String / Body Options:

Compose Query String/Body:

From the Following Constant Values: Adds constant name value pairs to the query string/body

And values from these fields: Takes name value pairs from the incoming data record. The field name is the name used in the Query String/Body

Content-Type: Controls how the name value pairs are encoded.

Application/x-www-form-urlencoded: e.g.
Name1=Value1&Name2=Value2&Name3=Value3. Unsafe
ASCII characters are automatically encoded, so make sure your
data is not already encoded.

Multipart/form-data: Only available when using HTTP Actions
POST and Custom.

Take Query String/Body from Field: Select the field in the input
data which contains the Query String/Body data. Blob fields
can be selected when using HTTP Actions POST, PUT and
Custom.

Use Following for Query String/Body: Text box allowing you to
manually type in the Query String or Body contents.

All text data is UTF-8 encoded before being sent to the remote
web server.

Connection tab:
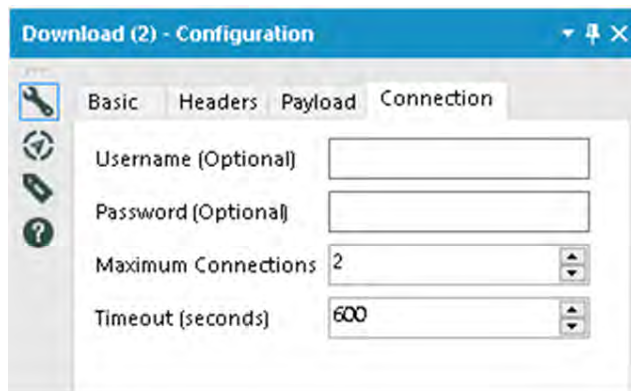
Figure 12.5–Download Configuration

- *Username*: Enter the user name if required by the URL specification above.
- *Password*: Enter the password if required by the URL specification above.
- *Maximum Connections*: Specify the maximum number of simultaneous transfers for the Download tool to perform. Transfers are only done in parallel when there are multiple input records sent to the Download tool. Multiple Download tools operate independently but do not typically function at the same time. For new Download tools added to a workflow, the default number of connections is 2 and the maximum number of connections is 32.

  Increasing the number of connections may reduce the total time taken to complete all transfers, but please use caution not to set the number too high as it could overload the server being used. It is possible the server could quit responding, report errors, or even refuse connections if it believes you are misusing it. This is particularly important when accessing a public site that is not under your control. Most web browsers will do as many as 6 simultaneous transfers, but these would typically be relatively small transfers as a part of a web page. For a server inside your own business where you have more control over how it is configured, using a higher number of connections may be okay.

  Additionally, because the Download tool will send records downstream as transfers complete, it will likely result in a change in the order of records as they pass

through the tool. If the order matters to your workflow, make sure to sort the results or limit the number of connections to 1. Finally, please note that empty URLs will be processed ahead of those that require an actual transfer.

- Timeout (seconds): Specify the number of seconds to wait before reporting a timeout due to an unresponsive connection. Select a number from 0 (never timeout) to 10,000.

12.3 Twitter Search
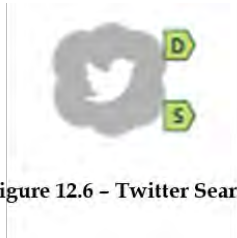
Figure 12.6 – Twitter Search

The *Twitter Search* tool allows you to search tweets by given search terms, with the location as an optional property. The search will only retrieve tweets from the previous seven days.

Group	Input	Output
Connectors	None	See Output data and Output Summary

Note: Before you can you use this tool, you must register an application with Twitter. Log in to your Twitter account at https://apps.twitter.com, click "Create a new application", and complete the form (a placeholder website may be used and there is no need for a Callback URL). Once you have submitted the application form, you will be provided with a Consumer Key and Consumer Secret that you can use to configure the tool.

> The search will only retrieve tweets from the previous seven days.
>
> *Output Data:* The selected data from your query.
>
> *Output Summary:* Summary information from your query.

Properties Window:
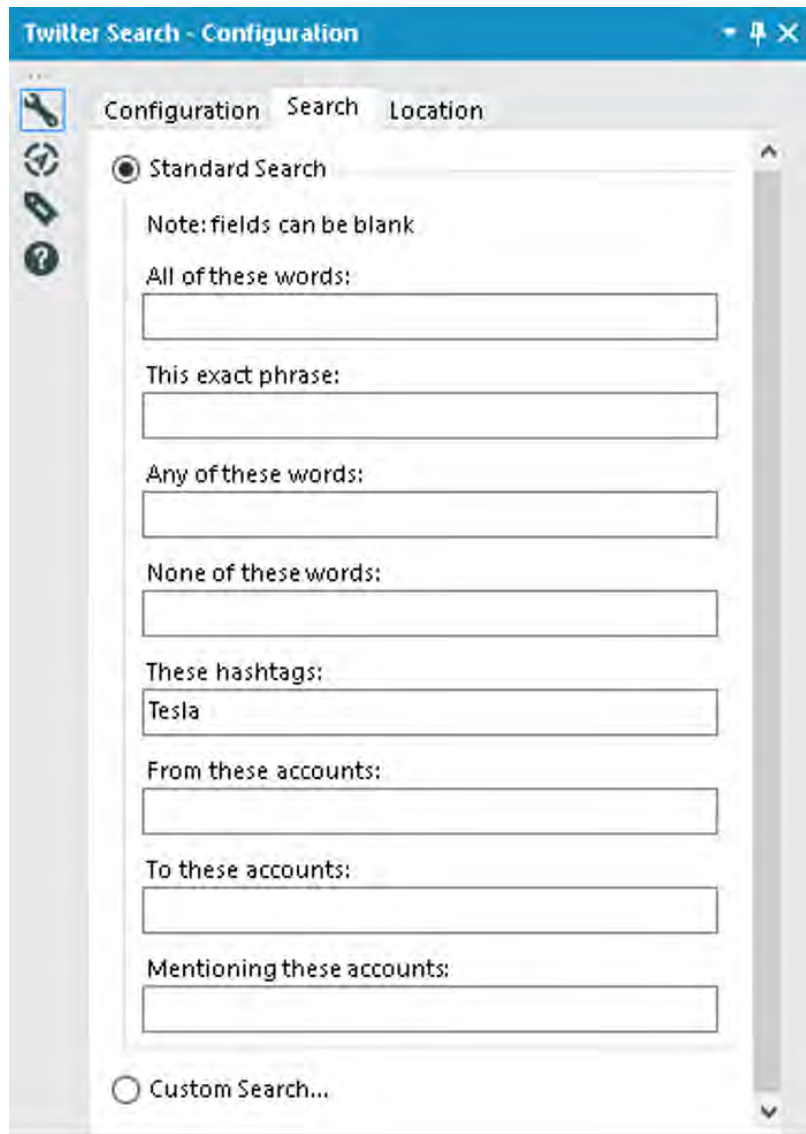
Configuration tab

Figure-12.7 – Twitter Search Configuration

- *Authentication:* Enter your credentials.

- *Record Limit:* Specify the number of records to request, up to a maximum of 100,000 records.

Search tab with two options – Standard search and Custom Search

Figure 12.8–Twitter Search Configuration

If you choose *Standard Search*, you'll see eight text boxes in which to enter search criteria. You can enter criteria into one or all text boxes. The tool takes all the criteria you enter and configures it into one combined query.

- *All of these words:* Return Tweets containing all of the words (in any order) in the body of the Tweet.
- *This exact phrase:* Return Tweets containing this exact phrase in this exact order. This is one of the only text boxes in the Standard Search that allows punctuation (e.g., if the exact phrase you are searching for contains a comma or quote, do include the comma or quote in the search).
- *Any of these words:* Return Tweets containing at least one of these words.
- *None of these words:* Exclude Tweets containing any of these words.
- *These hashtags:* Return Tweets containing at least one of these hashtags.
- *From these accounts:* Return Tweets from any of these accounts.
- *To these accounts:* Return Tweets set to any of these accounts.
- *Mentioning these accounts:* Return Tweets that mention any of these accounts.

The *Custom Search* option allows users to do a more advanced search than the Standard Search by utilizing the proper Twitter API syntax. You can learn about proper Twitter API syntax via these links: https://dev.twitter.com/docs/using-search.

Location tab
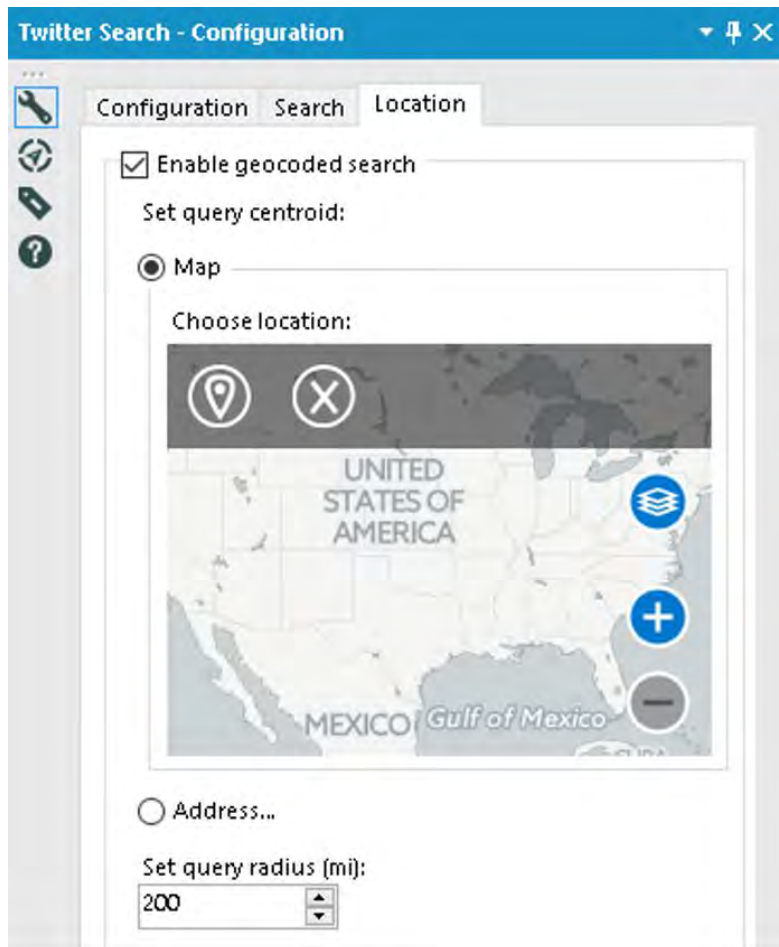
Figure 12.9–Twitter Search Configuration

- *Enable geocoded search:* Returns tweets by users located within a radius of a given latitude/longitude.

- *Set query centroid:* Specifies the latitude and longitude coordinates for the center of the search radius.

- *Map:* The geo-location of the point placed by the user on the map will be used to specify the latitude/longitude coordinates.
- *Address:* The geo-location of the address submitted by the user will be used to specify the latitude/longitude coordinates.
- *Set query radius (mi):* Sets the size of the search radius for the geo search. Defaults to five, with a max size of 1000.

12.4 Amazon S3 Download

Figure 12.10 – Amazon S3 Download

The *Amazon S3 Download* tool will retrieve data stored in the cloud where it is hosted by Amazon Simple Storage Service (Amazon S3).

Group	Input	Output
Connectors	None	Any data Stream

Output Data: Data stream containing records from cloud where it is hosted by Amazon Simple Storage Service.

Read CSV, BDF and YXDB files from Amazon S3.

Properties Window:

Figure 12.11 – Amazon S3 Download Configuration

- **AWS Access Key**: Specify the Amazon Web Services Access Key to use to download data.
- **AWS Secret Key**: Specify the Amazon Web Services Secret Key to use to access the data for download.
- In the drop-down, select an encryption option for the *AWS Secret Key*:

- o **Hide (Default)**: Hide the password using minimal encryption.

- o **Encrypt for Machine**: Any user on the computer will be able to fully use the connection.

- o **Encrypt for User**: The logged in user can use the connection on any computer.

Save Current AWS Credentials As Default: Saves the AWS credentials to the machine's registry.

Delete Saved Default AWS Credentials: Deletes any previously saved AWS credentials from the machine's registry.

- Endpoint: Select Default to allow Amazon to determine the endpoint automatically based on the bucket you select. To specify an endpoint for private S3 deployments, or if you know a specific bucket region, you can alternately select an endpoint (S3 region), enter a custom endpoint, or select from one of ten previously-entered custom endpoints.

Use Signature V4 for Authentication: Select this option to use Signature Version 4 instead of the default Signature Version 2. This will increase security, but connection speeds may be slower. This option is automatically enabled for regions requiring Signature Version 4.

- **Bucket Name**: AWS stores data objects in Buckets. Type a Bucket name or select one from the list of available Buckets.
- **Object Name**: Specify the Object name (data file) to be stored in the previously specified Bucket. Type an Object name or select one from the list of available Objects.

12.5 Amazon S3 Upload

	The *Amazon S3 Upload* tool will transfer data from Alteryx to the cloud where it is hosted by Amazon Simple Storage Service.

Figure 12.12 – Amazon S3 Upload

Group	Input	Output
Connectors	Any data stream	None

Write CSV, BDF and YXDB files from Amazon S3.

Properties Window:

Figure 12.13– Amazon S3 Upload Configuration

- *AWS Access Key:* Specify the Amazon Web Services Access Key to use to upload data.
- *AWS Secret Key:* Specify the Amazon Web Services Secret Key to use to access the data for upload.

- In the drop-down, select an encryption option for the *AWS Secret Key*:

  - *Hide (Default)***:** Hide the password using minimal encryption.

  - *Encrypt for Machine***:** Any user on the computer will be able to fully use the connection.

  - *Encrypt for User***:** The logged in user can use the connection on any computer.

*Save Current AWS Credentials as Default***:** Saves the AWS credentials to the machine's registry.

*Delete Saved Default AWS Credentials***:** Deletes any previously *saved AWS credentials from the machine's registry.*

- *Endpoint:* Select Default to allow Amazon to determine the endpoint automatically based on the bucket you select. To specify an endpoint for private S3 deployments, or if you know a specific bucket region, you can alternately select an endpoint (S3 region), enter a custom endpoint, or select from one of ten previously-entered custom endpoints.

*Use Signature V4 for Authentication***:** Select this option to use Signature Version 4 instead of the default Signature Version2. This will increase security, but connection speeds may be slower. This option is automatically enabled for regions requiring Signature Version 4.

- *Bucket Name:* AWS stores data objects in Buckets. Type a Bucket name or select one from the list of available Buckets.

- *Object Name:* Specify the Object name (data file) to be stored in the previously specified Bucket. Type an Object name or select one from the list of available Objects.
- *Enable Server-Side Encryption:* Select this option to allow files to be uploaded to an encrypted Amazon S3 bucket. The only method supported at this time is SSE-S3.

12.6 Foursquare Search

Figure 12.14 –
Foursquare Search

	Search Foursquare Venues by location with an option to filter by a search term.		
	Group	**Input**	**Output**
	Connectors	None	See Primary Output and Secondary Output

Note: Before you can use this macro, you must register an application with Foursquare. This can be set up by visiting https://foursquare.com/login and selecting 'Create a New App'. The only required fields are the 'Download/welcome page URL' and 'Redirect URL(s)', but http://www.foursquare.com is accepted as a placeholder in those fields https://foursquare.com/login. Once you have submitted your application form you will be provided a 'Client ID' and 'Client Secret' token, which are needed to configure the macro.
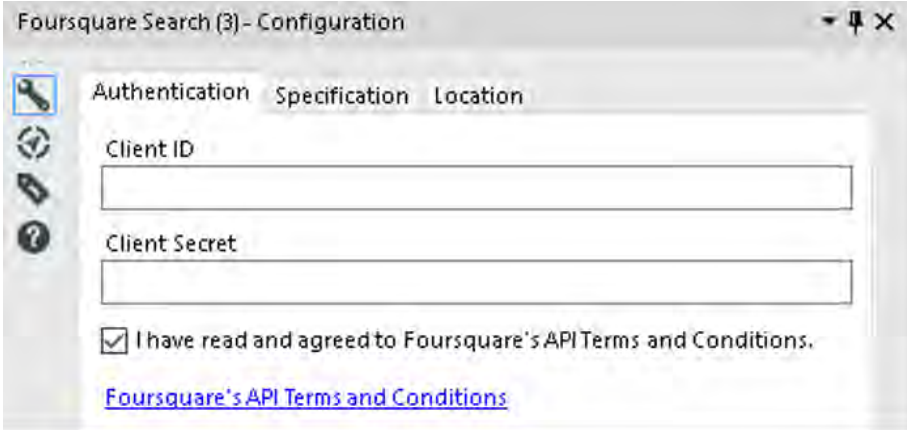
Primary output. This is the list of venues that fit the criteria selected in the interface, returned via the "P" macro output.

Secondary output. This optional output returns via the "S" macro output and contains either the NextVenues, Tips, or Photos if one of those options is selected in the interface; it will be empty if no secondary output is selected.

Properties Window:

There are 3 tabs: Authentication, Specification, Location

Authentication Tab

Figure 12.15– Foursquare Search Configuration

- *Client ID:* Enter your Foursquare Application Client ID.
- *Client Secret:* Enter your Foursquare Application Client Secret.
- *Foursquare API Terms and Conditions*: In order to run this macro, you must read and agree to the Foursquare API Terms and Conditions.

Specification Tab
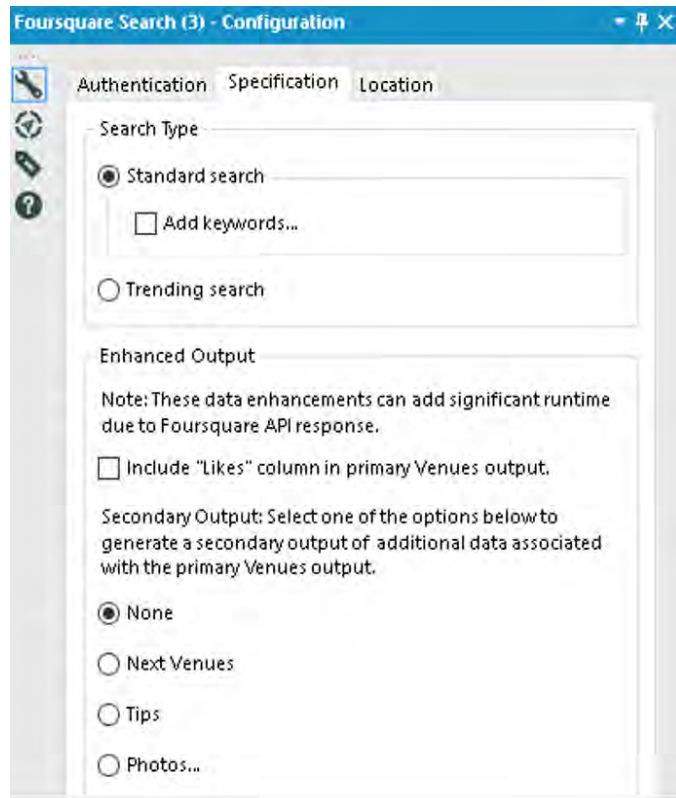
Figure 12.16– Foursquare Search Configuration

- Specify the *Search type*:

  o *Standard Search*: Returns a list of venues near the specified location.
  o *Add keywords*: When checked, the venues will be filtered to return only those relevant to the search terms entered.
  o *Keywords*: Specify keyword search terms, separated by commas. A single 'search term' can be 1 or more words, and a maximum of 10 search terms may be specified at a time. (e.g. burgers,ice cream,steak) Each comma-separated search term will have a maximum of
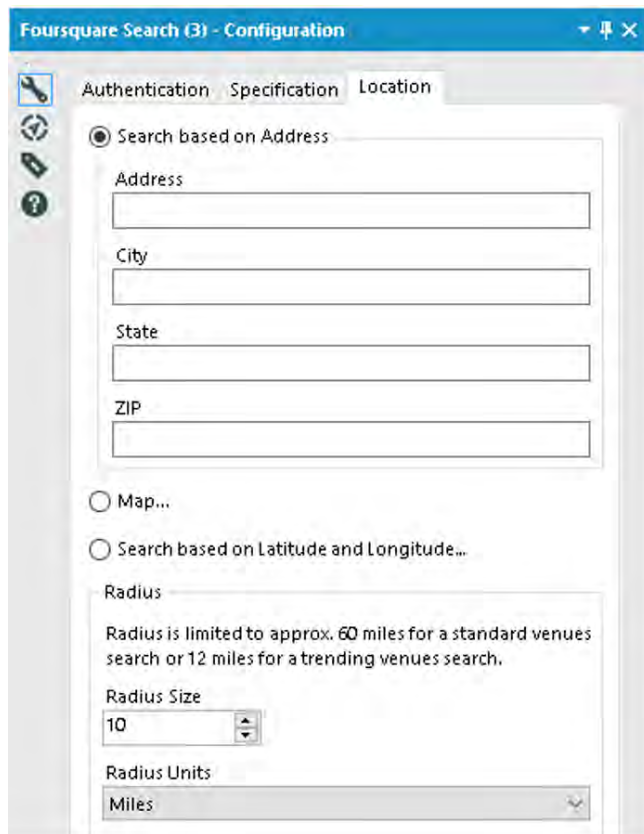
50 venues returned. (So in the previous example, you can get up to 150 records in total—50 for each search term.)

o *Trending Search*: Returns a list of up to 50 venues near the specified location with the most people currently checked in. (Keywords are not available in Trending searches.)

- Enhanced *Output***:**

  o The Foursquare macro allows you to enhance the primary venue-related data returned, by enhancing the primary output and/or producing an additional secondary output. Be aware that selecting these options may—to varying degrees—add significant runtime to your process due to awaiting responses from the Foursquare API.

  o Include "Likes": Check this box to populate the Foursquare "Likes" column in the primary Venues output.

- *Secondary Output***:** Select one of these options to generate a 'Secondary' output stream of data associated with the primary venues output. Join the secondary data output to the primary venues output by the FoursquareVenueID column that's found in both the primary and secondary output data streams. Choices include:

  o *None*: No secondary output is generated.

  o *Next Venues***:** Returns venues that people often check into after the current (i.e. 'primary') venue. Up to 5 'next' venues are returned for each 'primary' venue.

  o *Tips***:** Tips that members have posted to Foursquare.

    o   ***Photos*:** Photos related to the venue. Up to 200 photo records are returned for each venue.

Include actual image data: By default, the Photos output contains URL information and other metadata (username, photo width/height, etc) related to the photo, but not the actual image. Check this box to include the image. Including the actual image greatly increases both the time required to run the macro and the volume of data returned.

Location Tab

Figure 12.17– Foursquare Search Configuration

- **Enter** *the search location* **in 1 of 2 ways***:*

  o *Address*: Specify a Street Address, City, State and ZIP Code. To search based on an address, you must have licensed CASS and Geocoder.
  o *Latitude/Longitude*: Specify a latitude and longitude point.
  o *Radius Size*: Specify a radius for the venue search. Note: The maximum radius search supported by Foursquare is 100km (~60 miles) for a standard venues search or 20km (~12 miles) for a trending search. If the radius entered is larger, it will be automatically shortened to the appropriate limit.

- **Radius** *Units***:** Specify Miles, Kilometers, Meters or Feet as the unit of measure of the radius.

12.7 Google Analytics

 Figure 12.18 – Google Analytics	The *Google Analytics* tool downloads data from Google Analytics directly into your Alteryx workflow, allowing non-technical business users to utilize the Google Analytics API.

Group	Input	Output
Connectors	None	Any data stream

Output Data: The selected data from your query.

Output Summary: Summary information from your query.

Properties Window:

- **In** the *Configuration* **window**, select a sign-in method.
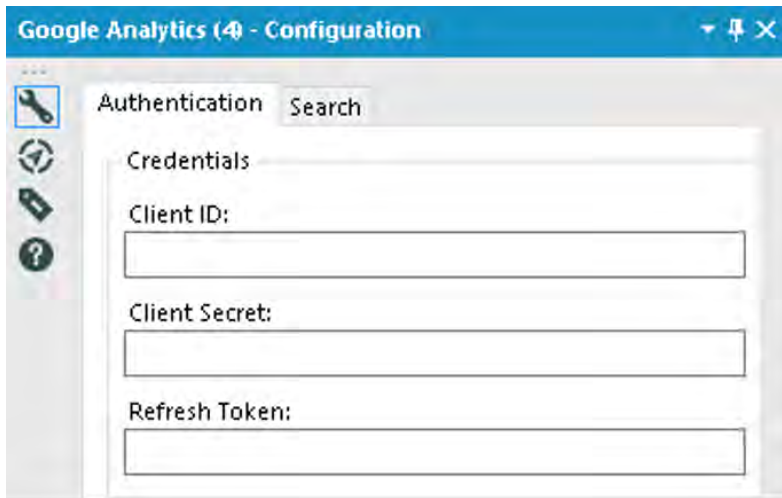
Figure 12.19– Google Analytics Configuration

o *Online*: Use this method for ad-hoc workflows. Sign in using Google account credentials. This option requires re-entering credentials every 60 minutes or any time a new workflow is opened. This method will not work for scheduled workflows.

o *Offline*: This method is required for scheduled workflows. Sign in using Google API credentials. This option requires a Client ID, Client Secret, and Refresh Token and does not require re-entering credentials to run a workflow.
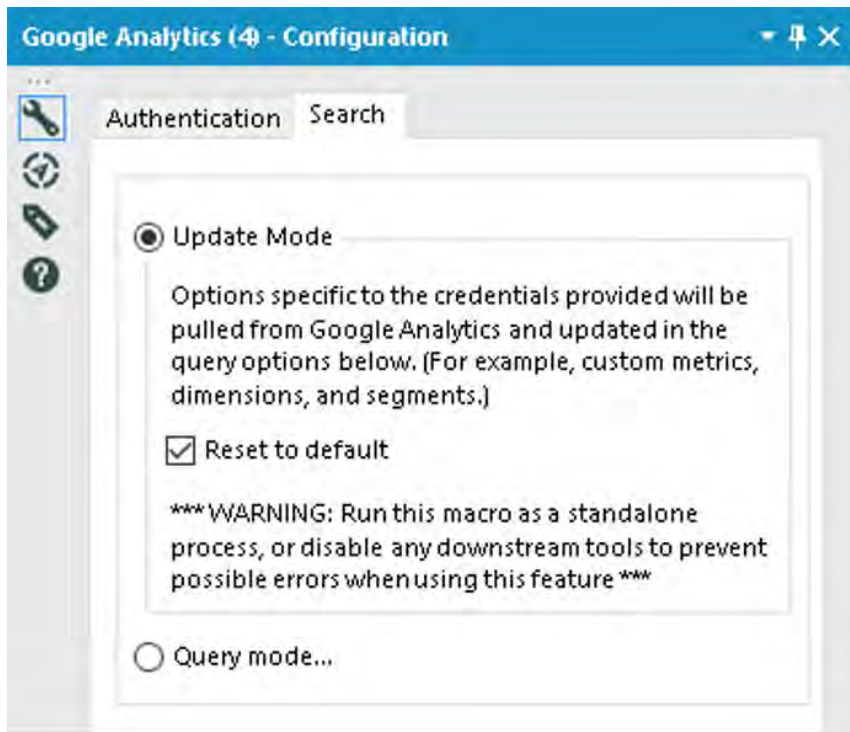
Figure 12.20– Google Analytics Configuration

- **Select** *an account*, *web property, and profile.*

  o *Available Accounts* is determined by your login account.
  o *Available WebProperties* is determined by your account selection.
  o *Available Profiles* is determined by your WebProperties selection. This is a unique table ID associated with the data query.

- **Select a** *date range.* Select either a preset range (for example, Today, Last Month, Year to Date) or Custom to specify a start and end date.

- **Select at least one *metric* and a maximum of ten metrics and goals. combinations** (**Optional**) **Select a maximum of seven dimensions**.

Dimensions break down metrics by common criteria. Only valid combinations of metrics and dimensions can be used.

- **(Optional) Select a maximum of four segments.** Selecting multiple segments limits the results to data included in all selected segments.

12.8 Marketo Append
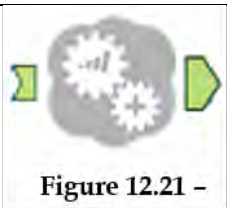
Figure 12.21 – Marketo Append

The *Marketo Append* tool retrieves Marketo records and appends them to the records of an incoming data stream.

Group	Input	Output
Connectors	Any data stream	Any data stream

Using the Append tool would be most helpful in situations where you have a list of email addresses or other data that you are already working with, and want to join that data with additional information from Marketo, or see if they already exist in your Marketo instance

Note: All dates in Marketo are stored with a UTC offset. This format is comprised of the local time with an appended offset that can be positive or negative.
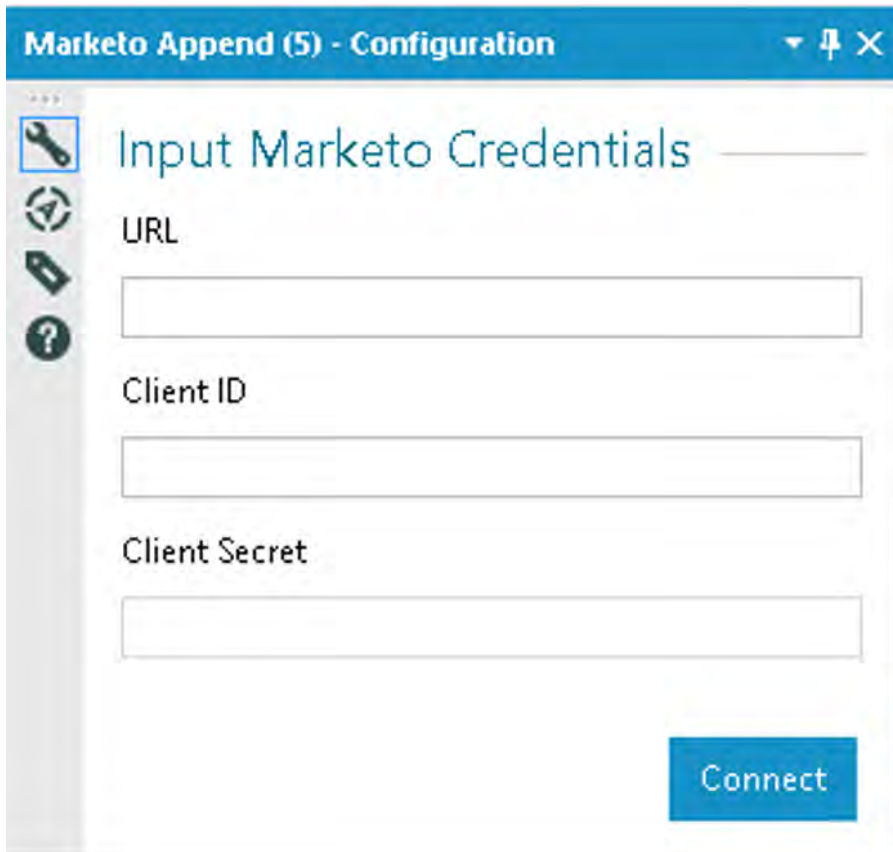
Properties Window:

Figure 12.22– Marketo Append Configuration

- *URL:* The Marketo REST Instance to retrieve records from. This information is found in Marketo (Admin > Integration > Web Services > REST API).
- *Client ID:* Client ID for the API Role user. This information can be found in Marketo (Admin > Integration > LaunchPoint).

- *Client Secret:* Client Secret for the API Role user. This information can be found in Marketo (Admin > Integration > LaunchPoint).
- *Incoming Field:* This drop-down will show the list of fields that you have used as input for the tool. Marketo will use this field to find lead records with this field value in the Marketo instance.
- *Incoming Field Type*: What type of records the incoming field is (email, SFDC id, etc.).
- *Output Fields*: The fields you want to retrieve from the Marketo database. These fields, as well as the fields that were passed into the tool, will be returned in the output data stream. If the output fields that you select are also contained in the incoming data stream, the output field coming from Marketo will be renamed with a '2' at the end.

12.9 Marketo Input

Figure 12.23 – Marketo Input

	The *Marketo Input* tool reads Marketo records for a specified date range.		
	Group	**Input**	**Output**
	Connectors	None	Any data stream
Use the Marketo Input tool if you do not have any data to start with, or to sample data from your Marketo instance. If you have data that you want to join to additional data from Marketo, the Marketo Append tool is a better option			

Two types of Marketo records can be retrieved:

LeadRecords: These are lead records and there will be one record for each lead.

Lead Activity Records: These records track the activities for each lead. There are potentially many Lead Activity Records for each Lead Record.

The Properties Window is the same as for Marketo Append: refer to [Section 12.8](#)

Based on the selections of the type of record, different options are available and based on that selection there will be parameters available to configure, some are optional and some are required. The table below details which parameters are required based on your chosen configuration.

Choose parameters

- *Lists*: All static lists for the Marketo instance specified in the configuration are listed here (smart lists are not currently available via the API)-only 1 list selection is allowed per request.
- *Activity Types*: All activity types available in Marketo are listed here, there is a maximum of 10 activity type selections per request.
- *Output Fields*: All available fields available in the Marketo instance will be listed. If no selection is made the default list of fields will be returned: Email, Lead ID, First Name, Last Name, Updated At, Created At.
- *Start Date/Time*: Specifies the starting point in the database to begin retrieving records.

Note: Error messaging such as authentication failures, etc. will be displayed in the configuration window. Error messaging such as missing configuration selections will be shown over the tool in the workflow.

The tool sends records in batches of 300 records or less, each batch counts as a separate API call towards the daily limit set by Marketo.

The Marketo REST API limits accounts to 10,000 requests per day, requests to the REST API do not count towards the SOAP API daily limitations.

You want to make sure to maintain Marketo field structures in Alteryx, before writing back to Marketo to avoid errors. The Marketo connector will convert the data to the proper datatypes before outputting to Marketo as long as the field names are valid. If the field names are not found in Marketo, the output will fail.

12.10 Marketo Output

Figure 12.24 – Marketo Output

The *Marketo Output* tool helps to write back data to Marketo using an 'Upset' operation.		
Group	**Input**	**Output**
Connectors	Any data stream	None

The Marketo Output tool makes a call to the Marketo REST API endpoint: Create/Update Leads. Data is output to Marketo based on the action you select in the configuration.

The Properties Window is the same as for Marketo Append: refer to Section 12.8

- *Key Field*: This drop-down will show the list of fields that you have used as input for the tool. Marketo will use this field to find any duplicate records. If no field is specified, then the email will be used by default. If the field 'id' is included in your records, 'id' must be selected as the key field.
- *Output Action*
  - ○ Create or Update (default): Based on the key field, create a lead if the lead does not exist, update the lead if it does exist.
  - ○ *Create Duplicate*: Create another lead even if it already exists.
  - ○ Update Only: Update the lead if it already exists, do nothing if it does not.
  - ○ *Create Only*: Create a new lead if it does not exist, if it does exist, skip it.
- *Partition Name*: If the Marketo Instance being accessed has partitions set up, they will be listed in the drop-down. This field selection is required if the instance being written out to has lead partitions set up. If there are no partitions available, 'Default' will be used.

12.11 MongoDB Input

  Figure 12.25 – MongoDB Input	The *MongoDB Input* tool reads data stored in MongoDB databases.		
	Group	**Input**	**Output**
	Connectors	None	Any data stream

The MongoDB Input tool is used for reading data stored in MongoDB databases. MongoDB is a scalable, high-performance, open source, NoSQL database. You can learn more about MongoDB here: http://www.mongodb.org/.

MongoDB databases store data in a BSON format which is binary JSON. You can learn more about BSON here: http://bsonspec.org/.

<u>Properties Window:</u>

Figure 12.26– MongoDB Input Configuration

- *Server*: The name of the MongoDB server you wish to connect to. Enter localhost to connect to a MongoDB instance you have on the machine running Alteryx.
- *User Name/Password* (optional): If your MongoDB instance is running with the –auth option this is where you enter the username/password you wish to connect to.

- Database: The name of the MongoDB database you wish to connect to.
- *Collection*: The name of the MongoDB collection you wish to read data from.
- *Mode*: There are 2 modes for reading the MongoDB. Depending on the mode chosen, additional configuration is necessary. Choices include:
  - *Automatic*: Alteryx will read the data in 2 passes: the first pass will scan through all documents to determine the table schema (columns, data type, and size); the second pass will return the data.
  - *Manual*: In Manual mode, you specify the schema or you can choose to scan a specified amount of documents to determine the schema.

12.12 MongoDB Output

	The *MongoDB Output* tool writes data to MongoDB databases.		
Figure 12.27 – MongoDB Output	**Group**	**Input**	**Output**
	Connectors	Any data stream	None
MongoDB is a scalable, high-performance, open source NoSQL database. You can learn more about MongoDB here: http://www.mongodb.org/.			
MongoDB databases store data in a BSON format which is binary JSON. You can learn more about BSON here: http://bsonspec.org/.			

Properties Window:

Figure 12.28– MongoDB Output Configuration

- *Server:* The name of the MongoDB server you wish to connect to. Enter localhost to connect to a MongoDB instance you have on the machine running Alteryx.
- *Database:* The name of the MongoDB database you wish to connect to.
- *Collection***:** The name of the MongoDB collection you wish to push data to.
- *User Name/Password (optional)***:** If your MongoDB instance is running with the –auth option this is where you enter the username/password you wish to connect to.
- *Output Options*:
  o *Append Existing*: Append the new records onto the end of your collection
  o *Delete Data & Append***:** Deletes the collection and then adds the new records
  o *Update Using _id*: Will attempt to match an existing record using the MongoDB _id element to the field specified in "_id field". If a match is found then the existing record will be updated with the new one. If no match is found then the new record will be appended onto the collection.
- *Mongo_id Field*: Used to select the Alteryx field which contains the _id value when using "Update Using _id" mode. Note if you wish to update based on the MongoDB generated ID then this should be a JSON object e.g. { "_id" : { "$oid" "4fad55603346998a9f7d6841" }}
- *BSON view*: The tree view at the bottom of the tool configuration provides a visual representation of what the BSON object will look like which is going to be inserted into your collection.
  The default options use the dot notation to build the structure of the BSON object so, for example, passing fields Member ID; Name. First; Name. Last would by default give you a JSON object which looked like this {

Member ID: 123, Name: { First: "Alistair", Last : "Terry" } }. However, you can restructure the BSON object to look however you would like by using the up/down/add/remove buttons.

- *Name :* Allows you to rename any given BSON element (Note: BSON names cannot begin with $ nor contain '.'s)
- *Folders:*
  - *As Name/Value*: Will create child nodes as name value elements e.g. Name : { First : "Alistair" , Last: "Terry" }
  - *As Array*: Will add child nodes as member of an array e.g. Name: ["Alistair" , "Terry"]
- *Mode :(*for the incoming Alteryx fields)
  - *Data is Value*: The data in the Alteryx field is plain data to add to the data element of the BSON pair.
  - *Data is JSON*: The data in an Alteryx string field is already a JSON and should be appended as such.
  - *Data is BSON*: The data in an Alteryx blob field is already a BSON and should be appended as such.
- *Unknown Fields:*
  - *Position Using Field Name*: The unknown fields will be positioned using the dot notation of their names. Starting with the location of the Dynamic or Unknown Fields node as their root position.
  - *Position as Left*: The unknown fields will be added wherever the Dynamic or Unknown Fields node is positioned.

12.13 Salesforce Input

	The *Salesforce Input* tool allows you to read and query tables from Salesforce.com into Alteryx.

Figure 12.29 – Salesforce Input

Group	Input	Output
Connectors	None	Any data stream

This tool requires that your Salesforce account is "API Enabled". Contact your Salesforce administrator for assistance with granting your account API user permissions.

Use Query Builder to browse for and select a Salesforce table, output fields, and other parameters. Use Custom Query to specify a table and parameters as a Salesforce Object Query Language (SOQL) query.

Properties Window:
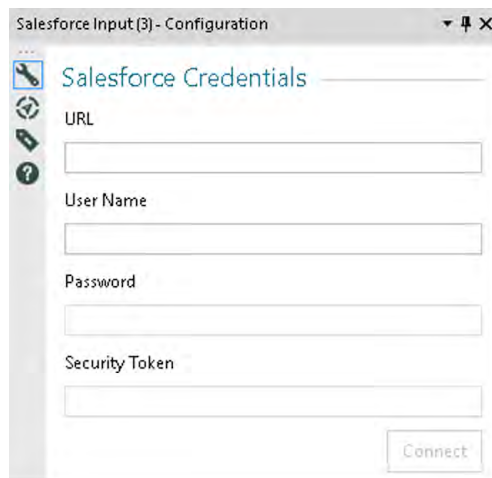
Figure 12.30– Salesforce Input Configuration

- *URL*: Enter your Salesforce URL in the following format: https://[instance].salesforce.com where [instance] refers to the specific server that pertains to your Salesforce environment. (For example https://na9.salesforce.com). The easiest way to determine your instance is to log into Salesforce in a browser at https://login.salesforce.com. Once you are logged in you will be redirected to a URL that contains your instance.
- *User Name*: Enter your Salesforce username associated with the Salesforce URL specified above. This is often an email address.
- *Password*: Enter your password for the Salesforce username specified above. This information will be encrypted.
- *Security Token*: You may need to enter your Salesforce Security Token. This information will be encrypted. You can find instructions on how to get your security token from Salesforce Help.
- *Connect*: Click this button to establish a connection. If the credentials are accurate, the Configuration window will display.
- *Query Builder*
  o *Table*: Click a table to read into Alteryx. Type in the search box to filter for a table or browse for a table by using the arrows – but you must click on a table name to register your selection. This list will only include queryable tables. Queryable here is a reference to a flag returned from the API. If this is set to false, then those tables will not be displayed in the list.
  o *Output Fields*: Select the fields you wish to retrieve from the above table. When no fields are selected, all output fields will be returned. This field is optional.

- o *Record Limit*: Enter a number of rows to return. If left blank, all rows will be returned. This field is optional.
- o *WHERE Clause (SOQL):* Specify a Query on the table specified above. Query language must be SOQL (Salesforce Object Query Language). This field is optional.
- SOQL Query: Specify a custom SOQL statement in the text box. If you previously selected a table using the Query Builder option, you will be prompted to load that query into Query Builder as a starting point.
    - o Attempt to Parse JSON Response: When selected, Alteryx will attempt to parse the query response and will display the parsed output in the Results window. If not selected, the response is output as a single field (called 'JSON'), which you can then parse with the JSON Parse tool.
    - o Validate: Click this button to submit the query to the Salesforce API to determine if the query is valid. Alteryx will also determine if the query results can be parsed. If Alteryx cannot parse the results, the response will be output as a single field (called 'JSON'), which you can then parse with the JSON Parse tool.

12.14 SalesForce Output

	The *Salesforce Output* tool allows you to write to Salesforce.com tables from Alteryx.		
	Group	Input	Output

Figure 12.31 – Salesforce Output	Connectors	Any data stream	None
This tool requires that your Salesforce account is "API Enabled". Contact your Salesforce administrator for assistance with granting your account API user permissions Note: Do not use Blob and SpatialObj field types in your workflow, as they cannot be output to Salesforce.			

Properties Window:

The SalesForce Output has the same configuration window as SalesForce Input, where you need to fill Salesforce credentials.

- *Table*: Click a table to write to from Alteryx. Type in the search box to filter for a table or browse for a table by using the arrows – but you must click on a table name to register your selection.
- *Output Operation*: When writing data to a Salesforce.com table, the field names must be the same as the field names in the table, including capitalization, but the type and size of the data do not matter. Choose from the following output options:
- *Update*: Replaces existing records in the specified table with the contents of the input stream. The ID field is required. All other fields must be within the target table.

- *Insert*: Adds to existing records in the specified table with the contents of the input stream. The ID field cannot be included. All other fields must be within the target table.
- *Delete*: Removes the records in the input stream from the specified table. The ID field is required and should be the only field provided.
- *Change Credentials*: Click the link at the bottom to change connection details, if necessary.

12.15 SharePoint List Input

	The *SharePoint Input* tool reads lists from SharePoint to be used as a data input in a workflow.		
Figure 12.32 – SharePoint List Input	**Group**	**Input**	**Output**
	Connectors	None	Any data stream
SharePoint automatically stores date/time data in UTC format. Date/time data is automatically converted to your local timezone when Alteryx reads a SharePoint list.			

Properties Window:
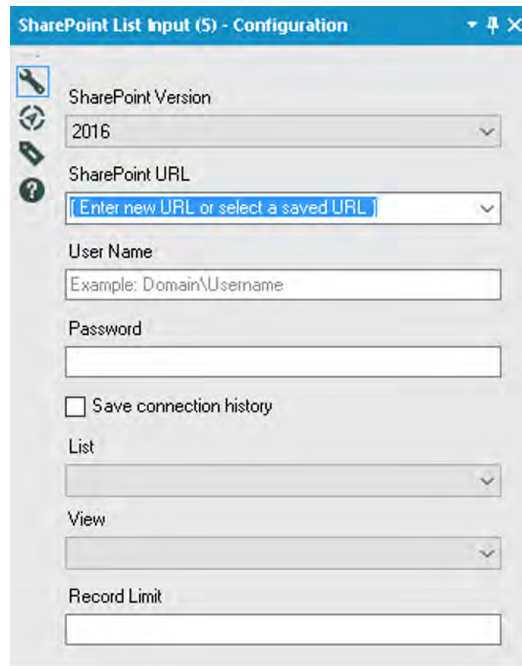
Figure 12.33– SharePoint List Input Configuration

- Select a *SharePoint Version:* 2007, 2010, 2013, 2016, or Online.
- Type the full *SharePoint URL* or click the drop-down to select a saved URL.
- Type your *User Name and Password.* The user name must include the domain name.
- (Optional) Select *Save connection history* to save the connection URL and credentials when you run the workflow. You can save up to ten connections.
- Select a *Lis*t from the drop-down. The List drop-down contents are specific to the URL you specify. If your URL contains a directory, only lists for that directory are shown.
- Select a *View* from the drop-down. The available views are determined by the list you select.

- (Optional) Specify a *Record Limit*. To read in all records, leave this field blank.

12.16 SharePoint List Output

Figure 12.34 – SharePoint List Output	The *SharePoint output* tool writes the content of a data stream to a Sharepoint list.		
	Group	**Input**	**Output**
	Connectors	Any data stream	None

Note: SharePoint automatically stores date/time data in UTC format. Date/time data is automatically converted to UTC when Alteryx writes a SharePoint list

Properties Window:

Figure 12.35– SharePoint List Output Configuration

- *Select a SharePoint Version*: 2007, 2010, 2013, 2016, or Online.
- Type the full *SharePoint URL* or click the drop-down to select a saved URL.
- Type your *User Name and Password*. The user name must include the domain name.
- (Optional) Select *Save connection history* to save the connection URL and credentials when you run the workflow. You can save up to ten connections.

- Select *a List* from the drop-down. The List drop-down contents are specific to the URL you specify. If your URL contains a directory, only lists for that directory are shown.
- In *Output Options*, select an option for writing the list:
  - *Append Existing*: Appends all the data to an existing table.
  - *Delete List and Append*: Deletes all the original records from the table and then appends the data into the existing table.
  - *Update; Warn on Update Failure*: Updates only existing records with the IDs you specify. If a record cannot be updated, a warning is reported.
  - *Update; Error on Update Failure*: Updates only existing records with the IDs you specify. If a record cannot be updated, an error is reported and processing will stop.

12.17 Green on the go

To,.. Alteryx Consultants

Cc...

Subject   Tesla

Hey,

A courier delivery services client is considering going green. So to reduce their carbon footprint, they are planning to introduce Tesla cars for their delivery services.

Before cars are introduced, they want to provide the drivers' information about all the Tesla Supercharger station. So that the drivers are well aware of nearest station, options to charge based on their delivery route etc on their company mobile app.

First they would want Tesla Supercharger station information to be updated periodically from the Tesla website to their database and Also they want data from social media like Twitter to see what is trending about the Tesla. Here is the link to get the Supercharger details:
https://www.tesla.com/findus/list/superchargers/United+States

I have heard that Alteryx is good in doing spatial and geo analysis, to get started could you please help me out in getting this data for the analysis.

Thanks

Based on the request to create the supercharger station data for the spatial and geo analysis, we will start looking at how to pull the information which is available on the Tesla website.

First, let us list down what we should be doing to extract the supercharger station data for the website in a specific format.

1. Get the URL of Tesla supercharger station information web page.
2. Look at the webpage, see what data would you extract for analysis. Like,

   Supercharger Station Name, Street Address, State, Zip and Roadside assistance

3. Download the page on Alteryx.
4. Parse the data from the above data points
5. Prepare and filter the data

   We already know the URL where we can find the Tesla supercharger station information

URL:https://www.tesla.com/findus/list/superchargers/United+States

Let us use the URL in Text Input and pass the URL to the Download tool.
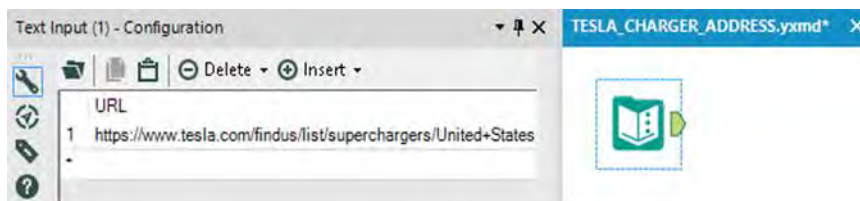
Figure 12.36– Green on the go – Text Input

Download tool takes the URL as input and retrieves data from a specified URL. We will configure the download tool to output data as a string. This option returns the data as a new wide string type field. A wide string supports Unicode characters.
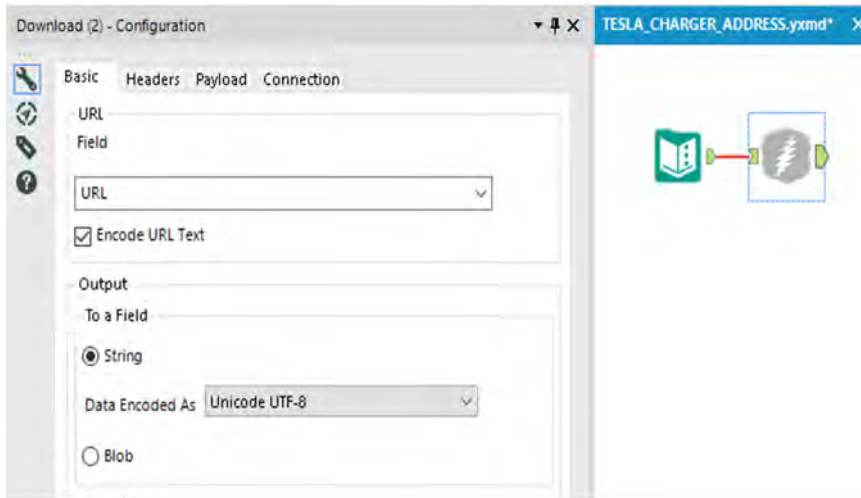
Figure 12.37– Green on the go – Download configuration
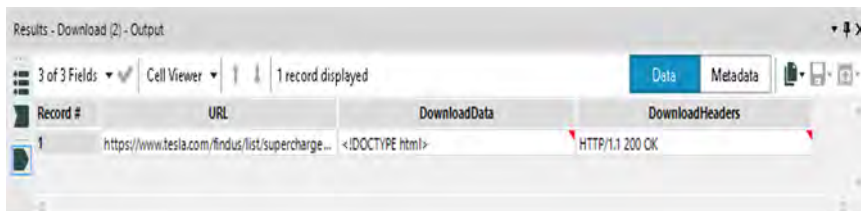
Figure 12.38– Green on the go – Download output

So now we have the result in a string or the text. Next step is to convert the string or the text to rows. We use the Text to column or row tool for this task.

At this stage running process shows us the following,
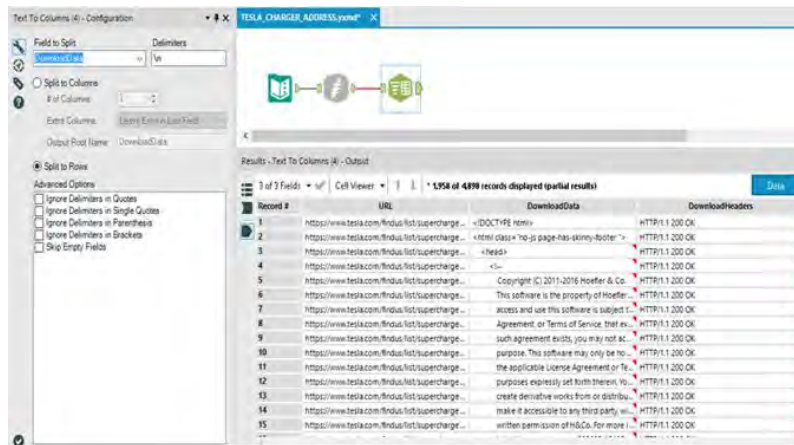
Figure 12.39– Green on the go – Text to column configuration and output

After converting the data to rows, the challenge is to get rid of the unwanted rows which were part of the string/text converted from the URL.
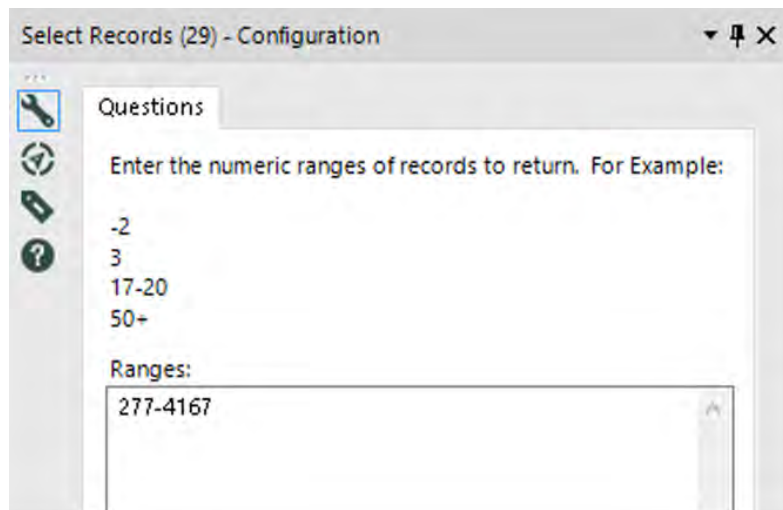
Figure 12.40– Green on the go – Select Records configuration

Using the select record, we will select the records from the row number 277-4167. Looking at the previous output we found the

first 276 rows does not contain the information that we want. So, we are filtering them out for the data stream.

Now is time to clean the data further. Using the data cleansing tool, let us do few cleansing operation.

Like,
- o Replacing NULL with Blank for the string data
- o Replacing NULL with 0 for the numeric data
- o Removing unwanted characters like leading and trailing whitespaces, Tabs, Line Breaks etc.
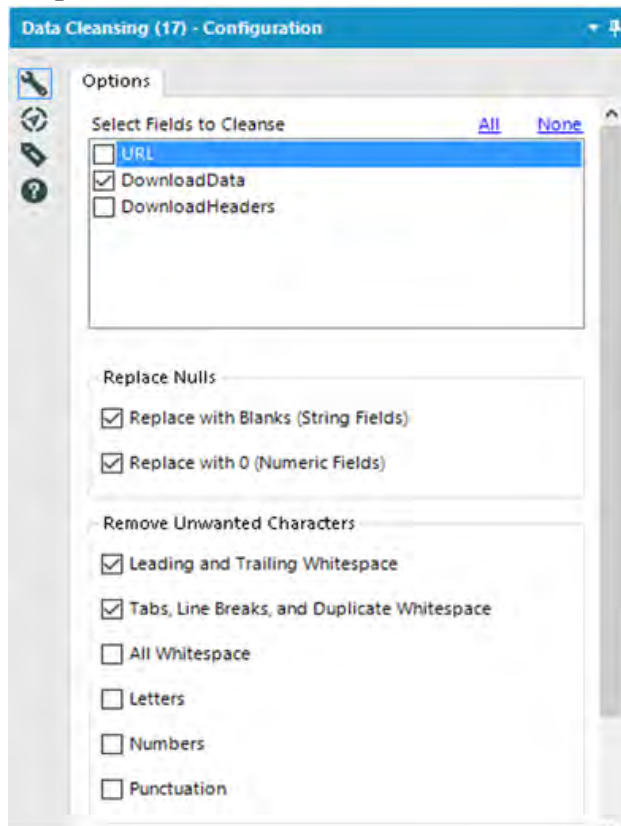
Figure 12.41– Green on the go – Data Cleansing configuration

Now we got a lot cleaner data.

Next step would be to create a mapping Create a mapping table using text input. Mapping table will have the data nodes that we would like to extract.

Roadside Assistance: <span class="value">

The above data nodes are mapped to the data points,
    Supercharger Station Name
    Street Address
    State & Zip
    Roadside assistance

Figure 12.42– Green on the go - Text input
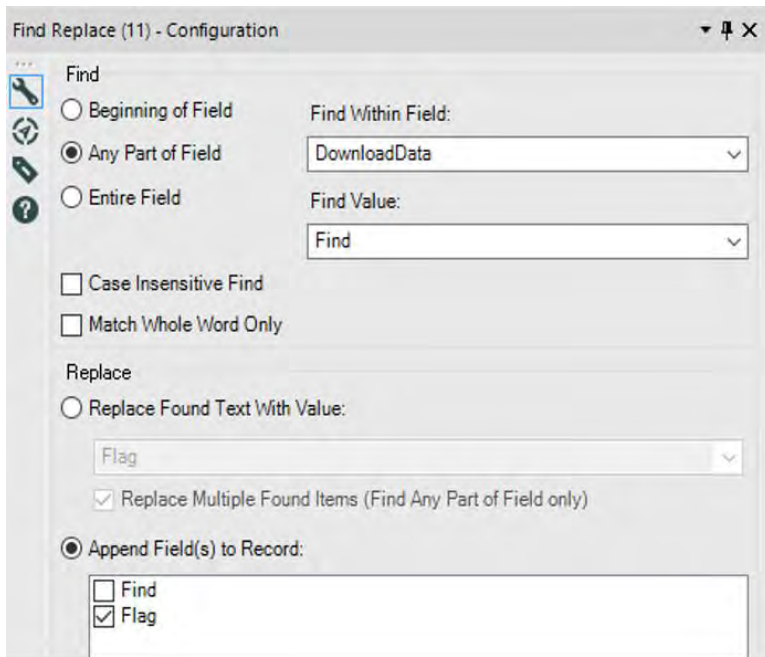
Figure 12.43– Green on the go –Find Replace Configuration

Filter the rows containing the data for the data point mentioned above.
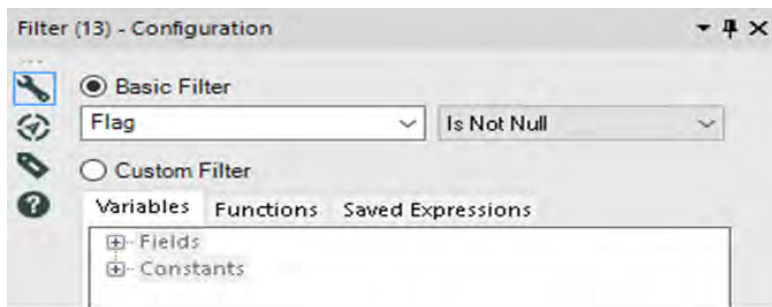
Figure 12.44– Green on the go –Filter Configuration

We have the data for Supercharger Station Name, Street Address, State & Zip and Roadside assistance

Since we have all the information which was requested, now time to clean up the unwanted HTML tags in the row and assign the unique ID to each Supercharger Station records.

Next 2 tools Multi-Row Formula and Formula tools will help us in assigning a unique number with a new column ID and removing unwanted HTML tags.
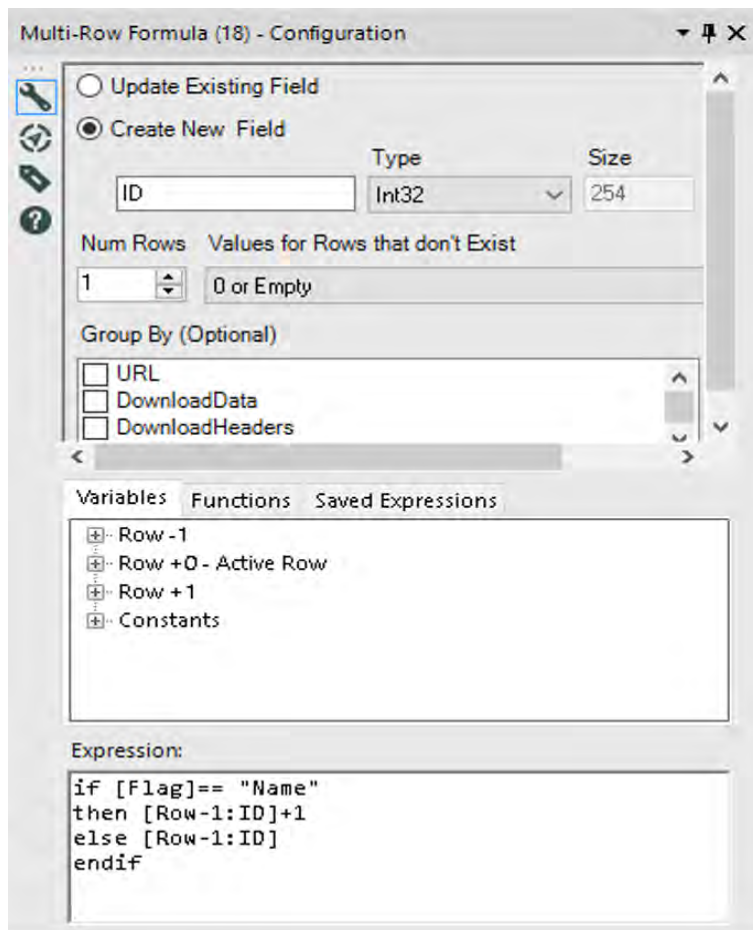
Figure 12.45– Green on the go –Multi-Row Configuration
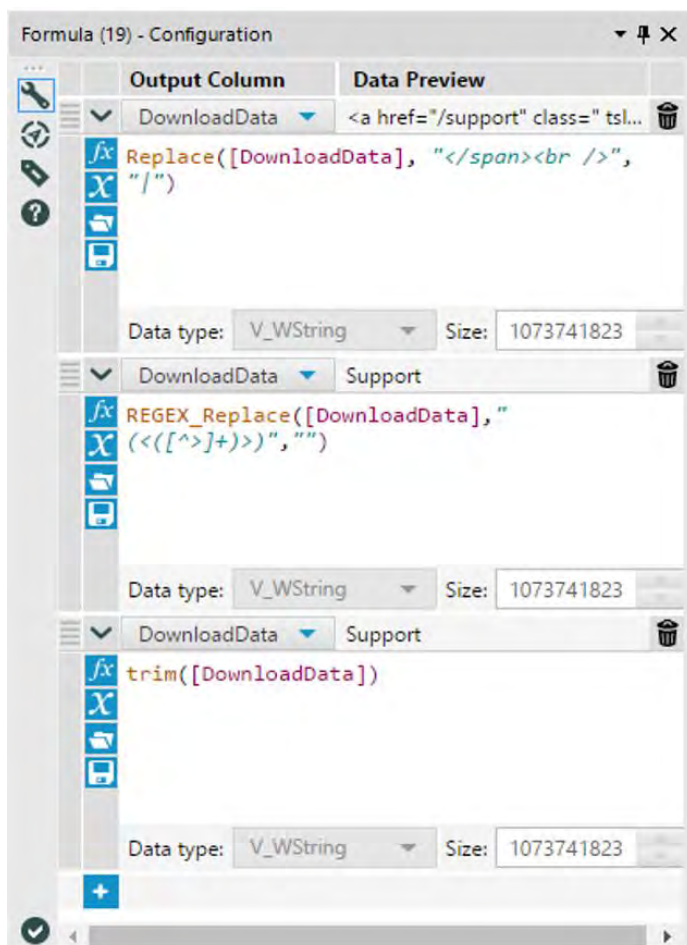
Figure 12.46– Green on the go –Formula Configuration

Till now we were massaging Supercharger Station data which were in rows. So now we would like to convert the relevant information as a column.
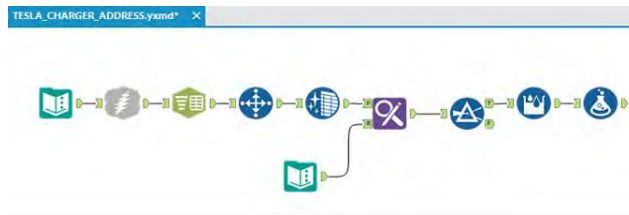
Figure 12.47– Green on the go –Data Stream

ID column which we just generated will help us to do a group for Supercharger Station and convert other information like the address, state, and zip into respective columns. We will use Cross Tab tool to reshape the data stream.
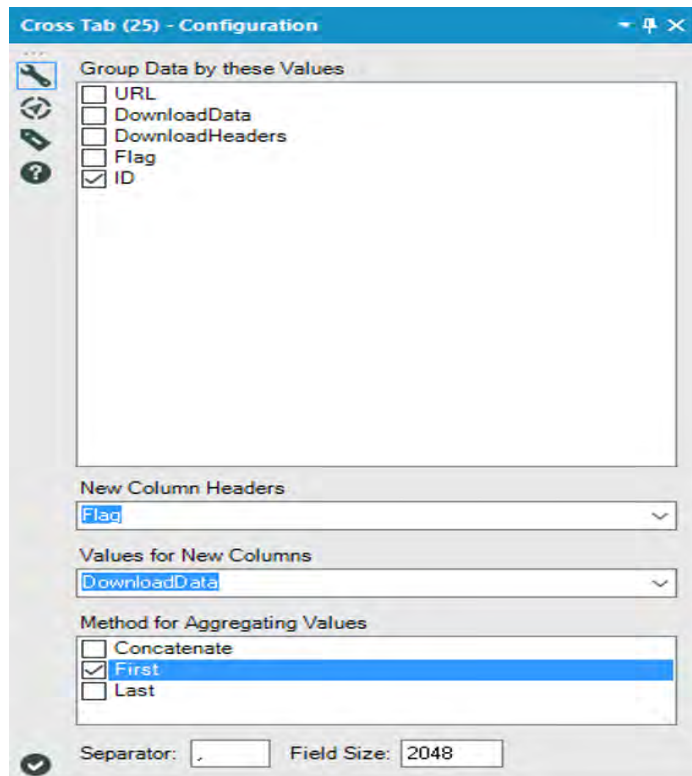
Figure 12.48– Green on the go –Cross Tab Configuration

The last step is to get rid of the unnecessary column and renaming the output column as required.
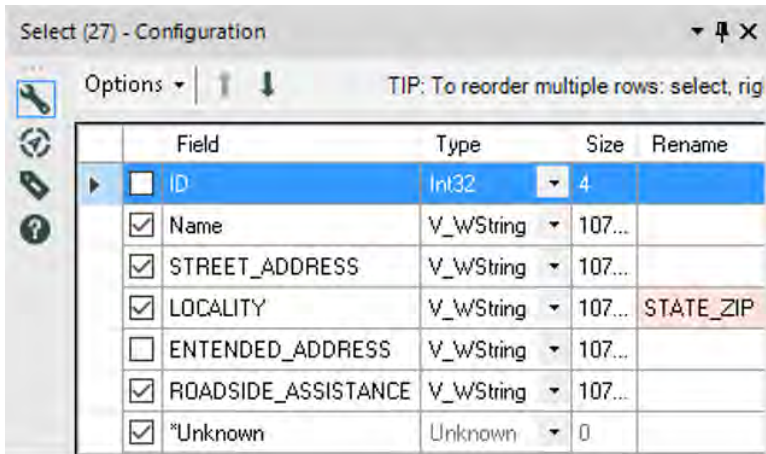
Figure 12.49– Green on the go –Select Configuration

One last time, make sure that there are no NULL rows in the data stream. We find first 7 rows are showing NULL values so we would like to filter out them for the data stream.

Figure 12.50– Green on the go –Cross Tab Output
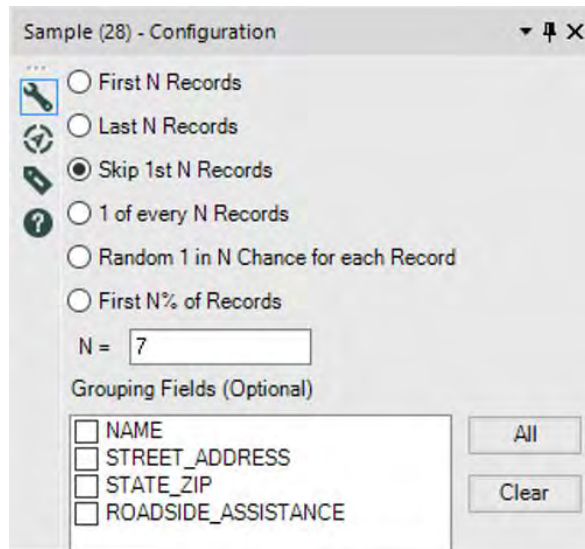
Figure 12.51– Green on the go –Sample
Configuration

That's it. We have the data in a required format for the spatial and geo analysis.

Below is the full view of the workflow that we just created.

Figure 12.52– Green on the go –data stream on
completion

12.18 What is trending for Tesla?

	To...	Alteryx Consultants
Send	Cc...	
	Subject	Tesla

Hey,

Tesla Supercharger station data has come along nicely. Now please help me out to see what is trending for Tesla on twitter.

Thanks

We would like to search for the hashtag Tesla in Twitter to see what is trending for Tesla. It could be about their car models, new release, supercharger stations, car recalls etc.

Steps below will help us accomplish the goals above,

The Twitter Search tool has to be downloaded from the Alteryx gallery.

To configure Twitter Search, first, we need a Twitter account and have to generate token.

Log in to your Twitter account at https://apps.twitter.com, click "Create a new application", and complete the form (a placeholder website may be used and there is no need for a Callback URL). Once you have submitted the application form, you will be provided with a Consumer Key and Consumer Secret that you can use to configure the tool.
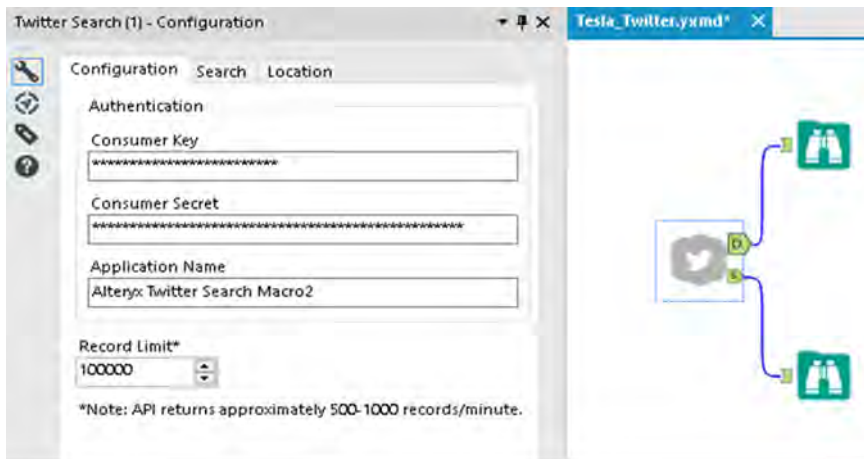
Figure 12.53– What is trending for Tesla –Twitter Search Configuration

Enter the hashtag to look for on Twitter. For us, it is "Tesla"

Figure 12.54– What is trending for Tesla –Twitter Search Configuration

We get a lot of information from the twitter. To reduce the size of data, we will select only specific fields that we need for analysis.
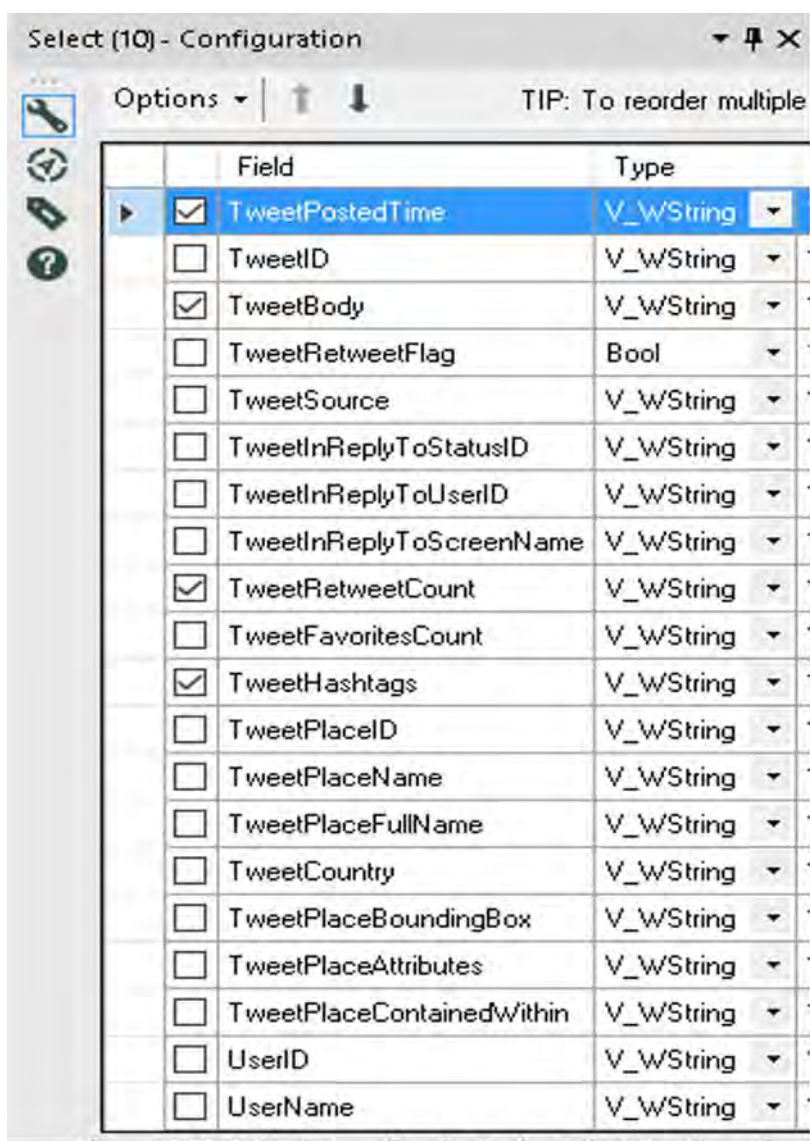
Figure 12.55– What is trending for Tesla –Select Configuration

Let us select below data point from the Twitter search output,
    TweetPostedTime
    TweetBody
    TweetRetweetCount

TweetHashtags
Dynamic or Unknown Fields

Figure 12.56– What is trending for Tesla –Auto field Configuration

All tweet fields are a string. we use Auto field tool which reads through all the records of an input and sets the field type to the smallest possible size relative to the data contained within the column.

Now we have to find the patterns for these tweets. You can use multiple tools to find the patterns. Like formula toll and Regex tool. In our approach, we will use the Formula tool.
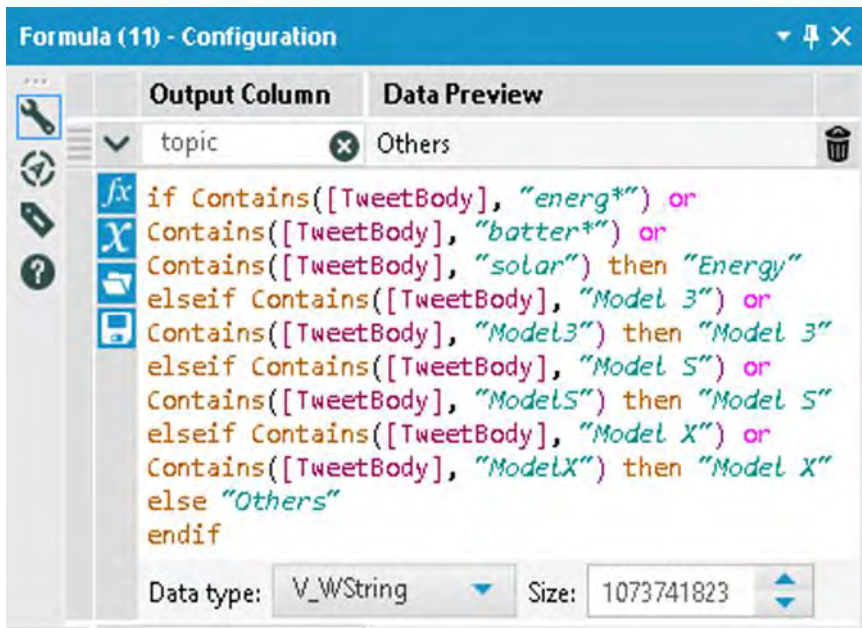
Figure 12.57– What is trending for Tesla –Auto field Configuration

Now we have a lot of rows with the same patterns and we need to aggregate them using the Summarize tool.

We use the Summarize tool twice.

First, to group by topic and get the number of tweets and retweets.

Second, to calculate a total number of tweets. This number or measure will help us in calculating % of a particular topic in all tweets.
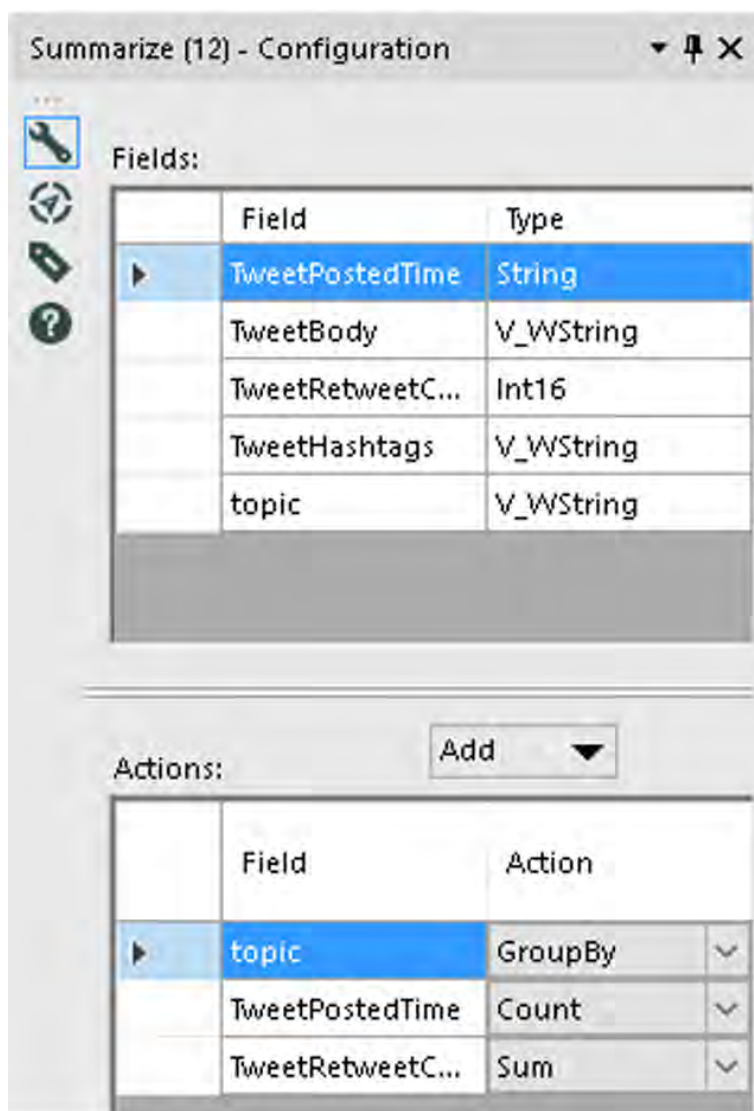
Figure 12.58– What is trending for Tesla –Summarize
Configuration
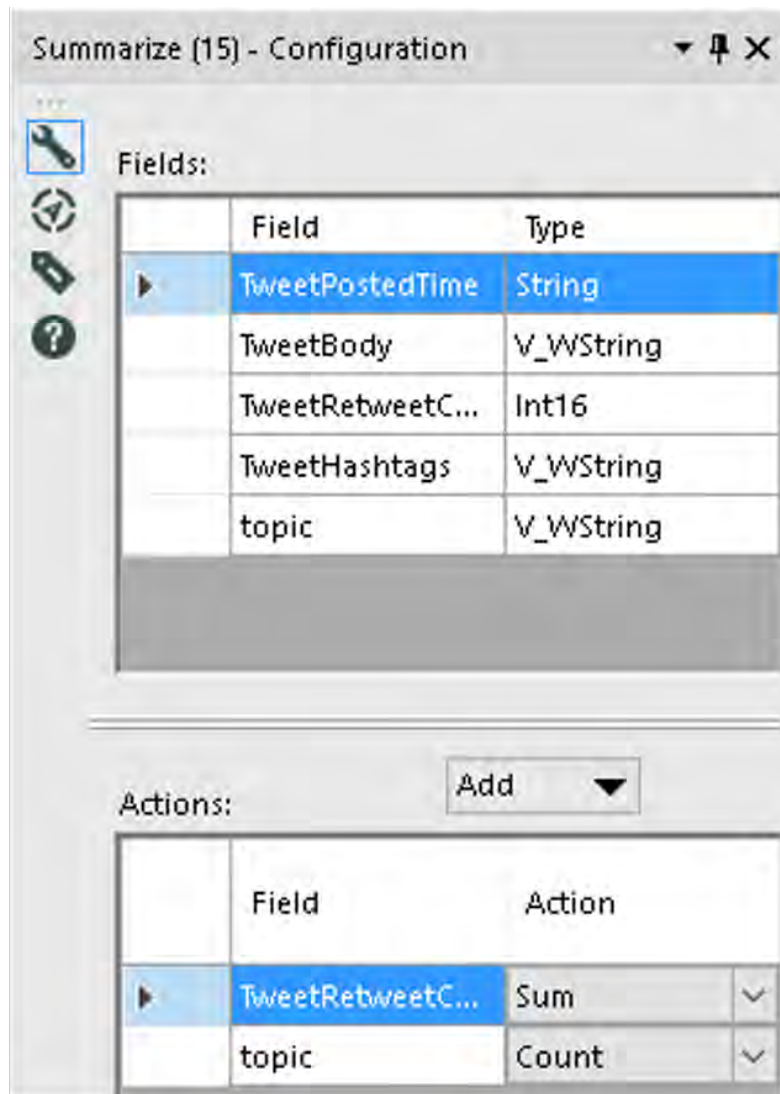
Figure 12.59– What is trending for Tesla –Summarize
Configuration

We will use the append fields tools to append a total number of tweets to each topic group.
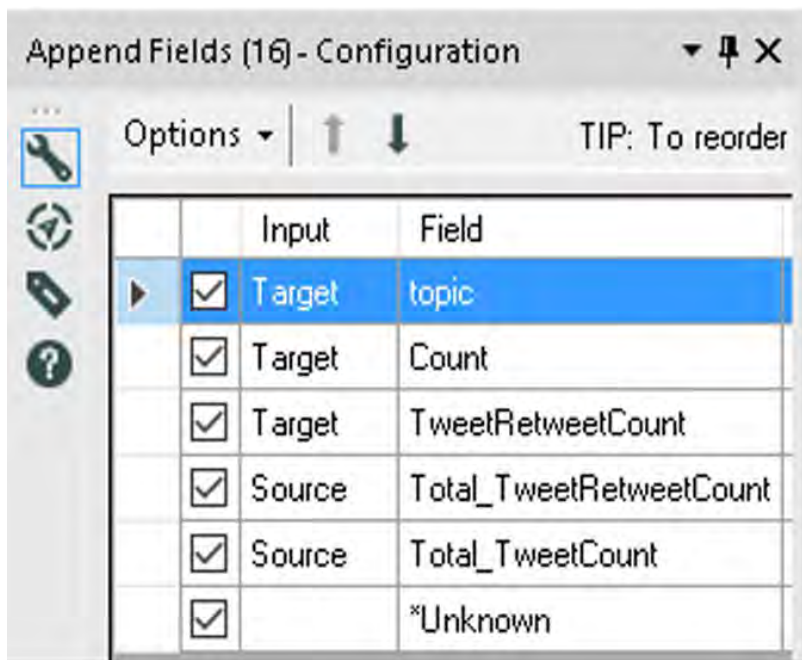
Figure 12.60– What is trending for Tesla –Summarize Configuration

Now Let's check the result. See if we have all the numbers and fields that we need for analysis.

Record #	topic	Count	TweetRetweetCount	Total_TweetRetweetCount	Total_TweetCount
1	Energy	927	7258	893081	17300
2	Model 3	1185	6712	893081	17300
3	Model S	782	109254	893081	17300
4	Model X	229	568	893081	17300
5	Others	14177	769289	893081	17300

Figure 12.61– What is trending for Tesla –Browse tool

Let's calculate % of tweets and retweets for each group.
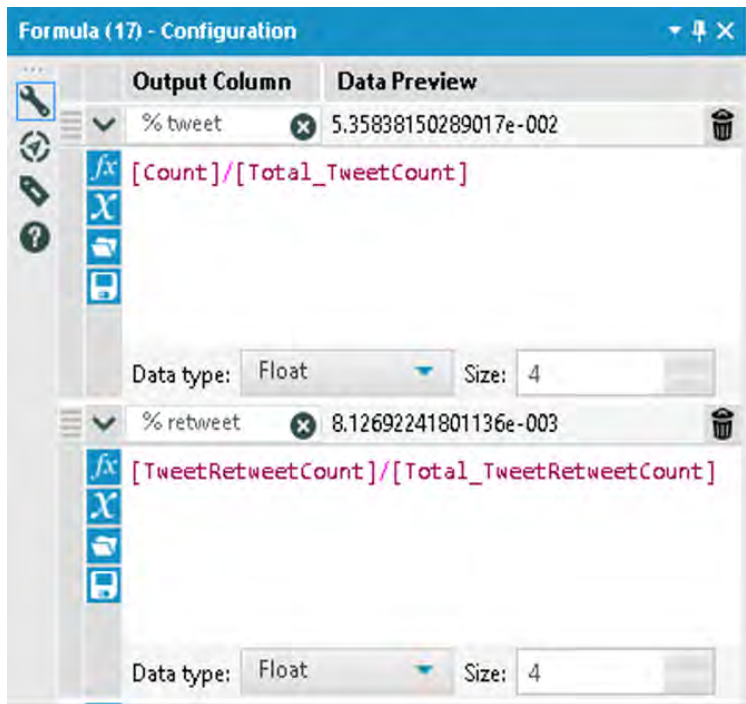
Figure 12.62– What is trending for Tesla –Formula tool

We are almost there. Now let's select all necessary fields and give them appropriate names.
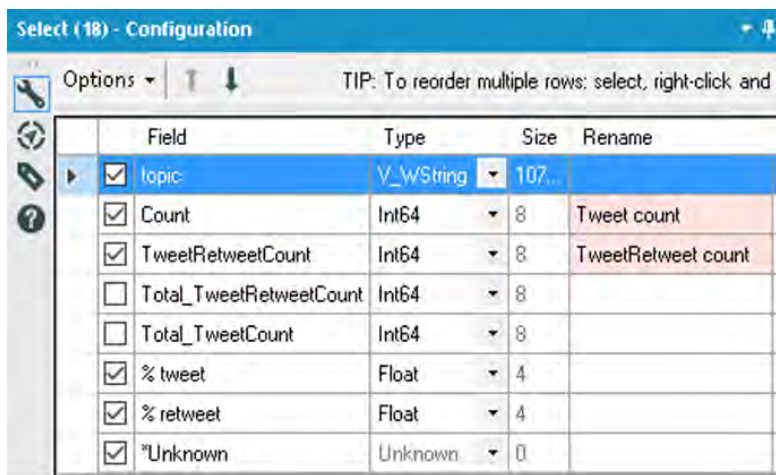
Figure 12.63– What is trending for Tesla –Select tool

Here is our final output. We have sorted the output based on the % of tweets.

Figure 12.64– What is trending for Tesla –Browse the result

The whole workflow is here.
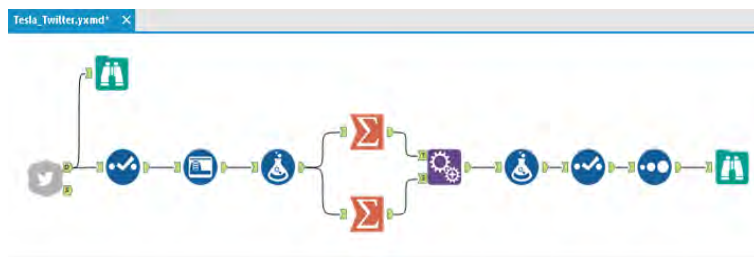
Figure 12.65– What is trending for Tesla –data stream on completion

CHAPTER 13

Statistics in Alteryx

To...	Alteryx Consultants
Subject	All The Tools Along the way
Attached	JSON.txt

Hey,

I have been asked to make sure you know how to use everything in a list of tools from the higher-ups.

There are a few that you haven't gotten to know yet, so I have designed something for us to go over and then for you to work through.

Attached, you will find a file called JSON.txt. We will be using it to create an application such that when you check a field, the JSON data will be converted into a relational structure. If you don't check the field, it will output a file identical to the original JSON file.

Thanks

13.1 Tools & Concepts

Tools	Concepts
Basic Data Profile	
Field Summary	
Frequency Table	
Pearson Correlation	
Spearman Correlation	
Allocate Append	
Allocate Metainfo	
Allocate Report	
Behavior Metainfo	
Cluster Code	
Compare Behavior	
Create Profile	
Detail Fields	
Profile Input	
Profile Output	
Output Tableau Workbook	
Polygon For Tableau	

Publish to Tableau
Server

Optimization

Simulation Sampling

Simulation Scoring

Simulation Summary

13.2 Basic Data Profile

Figure 13-1 – Basic Data Profile	The *Basic Data Profile* tool provides metadata for each of the columns in the data stream. It analyzes data and outputs the information.		
	Group	**Input**	**Output**
	Data Investigation	Any data stream	See Below

Special Note: Use this along with *Browse* tool to view the visual output.

Output: A table which provides the details of each column present in the input.

It is similar to Field Summary Tool as it provides information like the length, data type, source etc. The Basic Data Profile tool is more equipped to provide information related to missing data. It gives you a count of null and blank values. It also shows the number of records which have leading or trailing blank white spaces which tell you the number of records that need data cleaning.

Properties Window:

The *Basic Data Profile Configuration* window has three components.
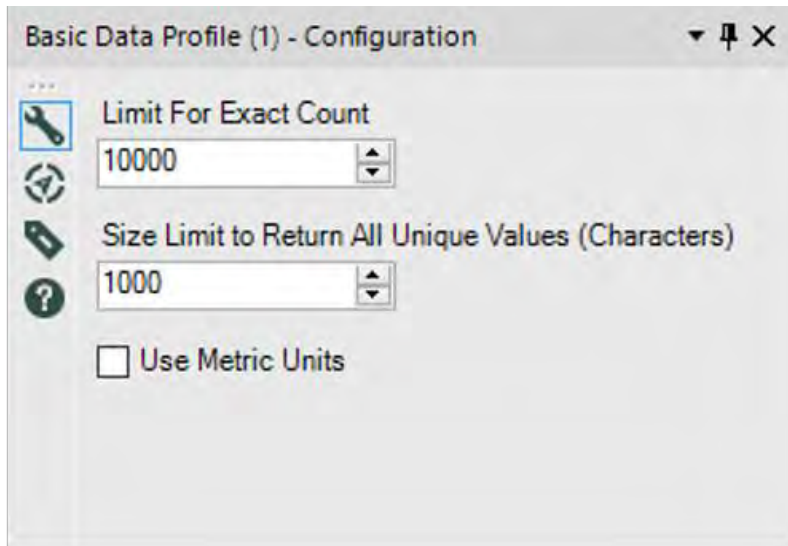
Figure 13-2 – Basic Data Profile Configuration

- *Limit For Exact Count:* The default limit is advised for a good performance. But if you need more information or number of rows, the limit can be increased. This value is the number of unique values that you want Alteryx to identify.
- *Size Limit to Return All Unique Values (Characters):* The default limit is advised for a good performance. The value can be increased to get more information. This value is the number of characters that you want Alteryx to check to determine whether the value is unique.
- *Use Metric Units:* This field is checked to use the metric units of measure. This is only applicable to spatial data.

13.3 Field Summary

Figure 13-3 – Field Summary	The *Field Summary* tool provides a summary report with descriptive statistics for the selected columns.		
	Group	**Input**	**Output**
	Data Investigation	Any data stream	Output O, R & I sections below
Note: Use this along with *Browse* tool to view the visual output.			

The Field Summary Tool is used to gain more information about data and suggestions for managing data. It provides information like the length, data type, shortest value, longest value, minimum value count and max value count. The data types which are supported in this tool are – Numeric, string, spatial and data/time. Also, a set of descriptive statistics is calculated and shown as output.

Properties Window:

The *Field Summary Configuration* window has two components.

Figure 13-4 – Field Summary Configuration

Input

- *Select the fields to product summary info*: This component is used to select the fields which are to be analyzed and included in the output data. Descriptive statistics will be provided for the selected columns only.
  All: This option lets you select all the columns to be included in the output.
  None: This option is used to deselect all the previously selected columns
- *Sample Input Data*: This checkbox is selected if you want to sample your input data. After the checkbox is selected, the user is asked to select the sampling method for the input data.
  Random N records: The user can specify the value of N and accordingly the number of records will be used to calculate the summary and descriptive statistics of the data. The user can type or use up and down arrow keys to enter the value for N.
  Random N% of records: The user can specify the percentage of records and accordingly the number of records will be used to calculate the summary and descriptive statistics of the data. The user can type or use up and down arrow keys to enter the value for a percentage.

Output

The Field Summary tool gives three outputs:

- *O (Output)*: It is an Alteryx data stream with descriptive statistics for the selected fields along with suggestions named as "Remarks" to manage the data. You will need to connect a Browse tool to the O output to view the visual report.

- *R (Report)*: It is a static report along with a scatter plot and descriptive statistics for the selected fields along with the suggestions named as "Remarks" to manage the data. You will need to connect a Browse tool to the R output to view the report.
- *I (Interactive)*: It is an interactive dashboard with statistics for each column. You will need to connect a Browse tool to the R output to view the report. Each column has a histogram, column chart and summary statistics report.

13.4 Frequency Table

 Figure 13-5 – Frequency Table	The *Frequency Table* tool produces a frequency table for the fields and gives a summary in terms of frequency counts and percentages.

Group	Input	Output
Data Investigation	Any Data Stream	Output D, R & I sections below

Note: Use this along with *Browse* tool to view the visual output.

Properties Window:

The *Frequency Table configuration* window has only one component.
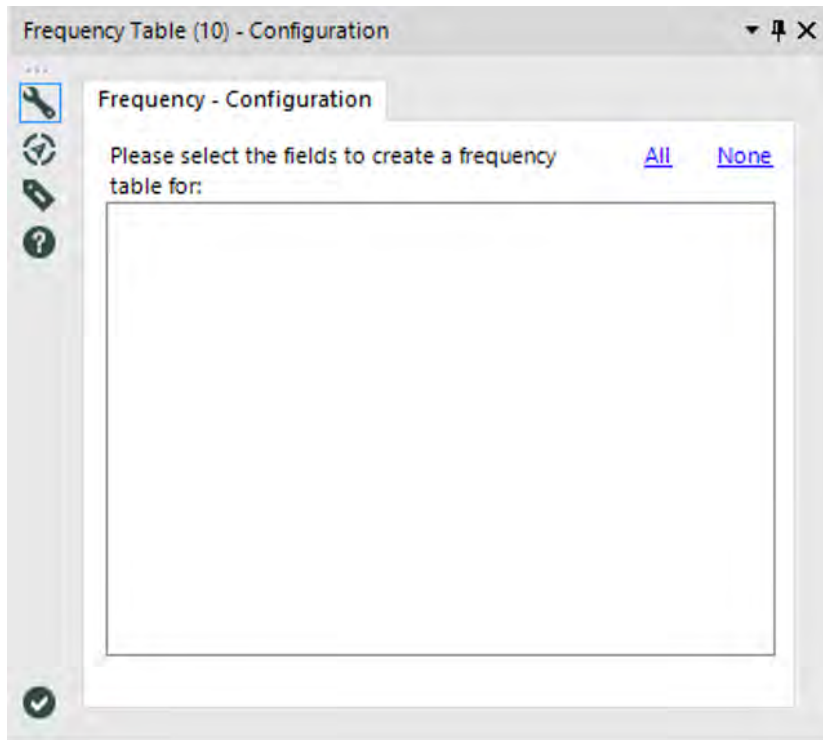
Figure 13-6 – Frequency Table Configuration

- *Please select the fields to create a frequency table for*: Select the columns on which you want to perform frequency analysis. Some data types cannot be selected for calculating frequency like Date, Time, DateTime, Blob, SpatialObj, and FixedDecimal.

Output

- *D (Data)*: This output gives us the statistics about the data of the chosen field. It has fields – Field_Name, Field_Value, Frequency, Percent, Cumulative_Frequency

and Cumulative Percent. You will need to connect a Browse tool to the D output to view the report.

- *R (Report)*: This is a report output with a frequency table for each selected column. It is shown on the left side pane. The header is the field name. The rest of the values for Field_Value, Frequency, Percent, Cumulative_Frequency and Cumulative Percent are shown in the report. You will need to connect a Browse tool to the R output to view the report.

- *I (Interactive)*: This is the same report as R output but with an interactive dashboard. It is shown on the left side pane. The header is the field name. There is a drop-down menu with a list of the selected columns for which frequency tables were calculated. Select one column to view the statistics for that column. The dashboard has values for Field Value, Frequency, Percent, Cumulative.Frequency and Cumulative.Percent are shown in the report. You will need to connect a Browse tool to the R output to view the report.

13.5 Pearson Correlation

 Figure 13-7 – Pearson Correlation	The *Pearson Correlation* tool gives the correlation of two variables. It determines the strength of their linear relationship.		
	Group	**Input**	**Output**
	Data Investigation	Any Data Stream	See Below
Note: Use this along with *Browse* tool to view the visual output.			

In statistics, the Pearson Correlation is also known as product-moment correlation. It is obtained by dividing the covariance of the two variables by the product of their standard deviations.

Properties Window:

The *Pearson Correlation configuration* window has two components.
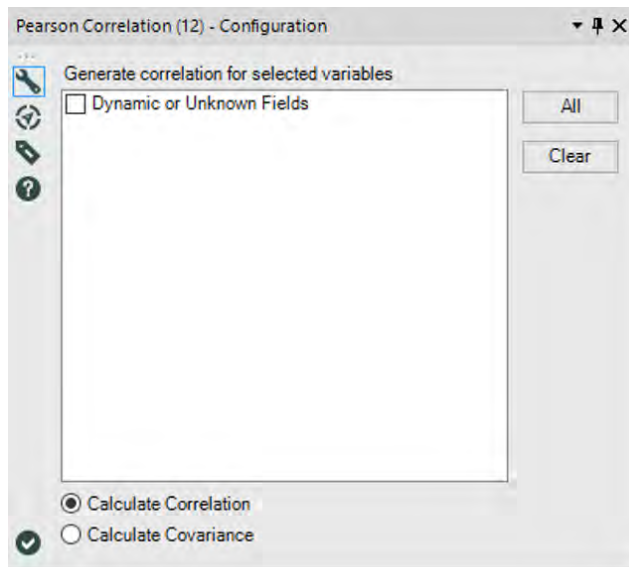
Figure 13-8 – Pearson Correlation Configuration

- *Generate correlation for selected variables*: Select the columns on which you want to perform the Pearson correlation. The fields must be numeric.
  Note: The fields should not be any kind of identifiers like primary or foreign keys as they will not be useful in finding a predictive value and can cause runtime exceptions.
- *Calculate Correlation/Covariance*: Select one of the two options – Correlation and Covariance. Correlation

calculates the Pearson Correlation and Covariance calculates the statistical formula Covariance among the fields.

13.6 Spearman Correlation

Figure 13-9 – Spearman Correlation	The *Spearman Correlation* tool measures the strength and direction of the association between two variables.		
	Group	**Input**	**Output**
	Data Investigation	Any Data Stream	See Below
Note: Use this along with *Browse* tool to view the visual output.			

The *Spearman Correlation* tool assesses the monotonic relationship between two variables without forming any other assumptions about the nature of the relationship between two variables.

Properties Window:

The *Spearman Correlation configuration* window has two components.
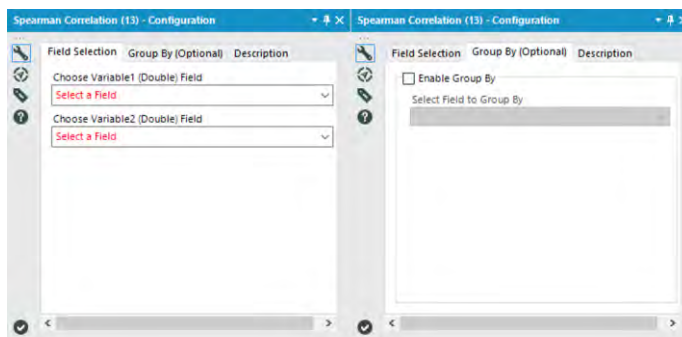
Figure 13-10a – Spearman Correlation Configuration

- *Field Selection*

  The variables are the two fields whose relationship we are going to assess using Spearman Correlation. Select the fields from the drop-down menu given for each field in the Field Selection pane.

  Note: The fields should not be any kind of identifiers like primary or foreign keys as they will not be useful in finding a predictive value and can cause runtime exceptions.

  - *Choose Variable1*:
  - *Choose Variable2*:
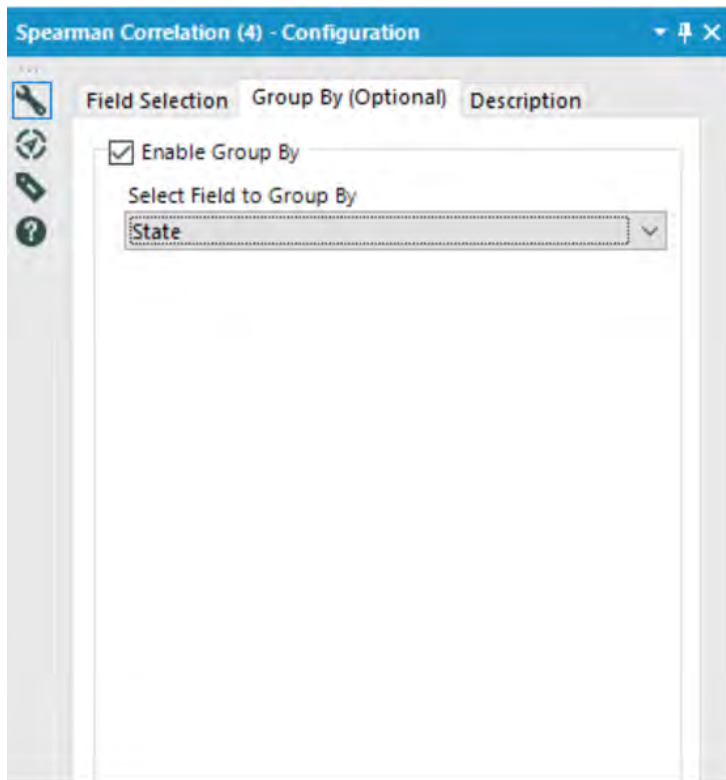
Figure 13-10b – Spearman Correlation Configuration

- *Enable Group By*: This is a checkbox on another pane "Group By (Optional)". As the header suggests, this field is optional. If you want to group the data using any field, check the checkbox and then select the field using which you want to group the data.

13.7 Allocate Append

	The *Allocate Append* tool is used to append demographic fields from an existing Allocate installation.

Figure 13-11 – Allocate Append

Group	Input	Output
Demographic Analysis	Allocate Dataset	Allocate Dataset with appended fields

Note: Use this along with *Browse* tool to view the visual output.

This tool can be used only if you have licensed installation of Allocate. The datasets or variables licensed on Allocate can be used directly on Alteryx through the Allocate Data Append tool.

<u>Properties Window:</u>

The *Allocate Append Configuration* window has six components.
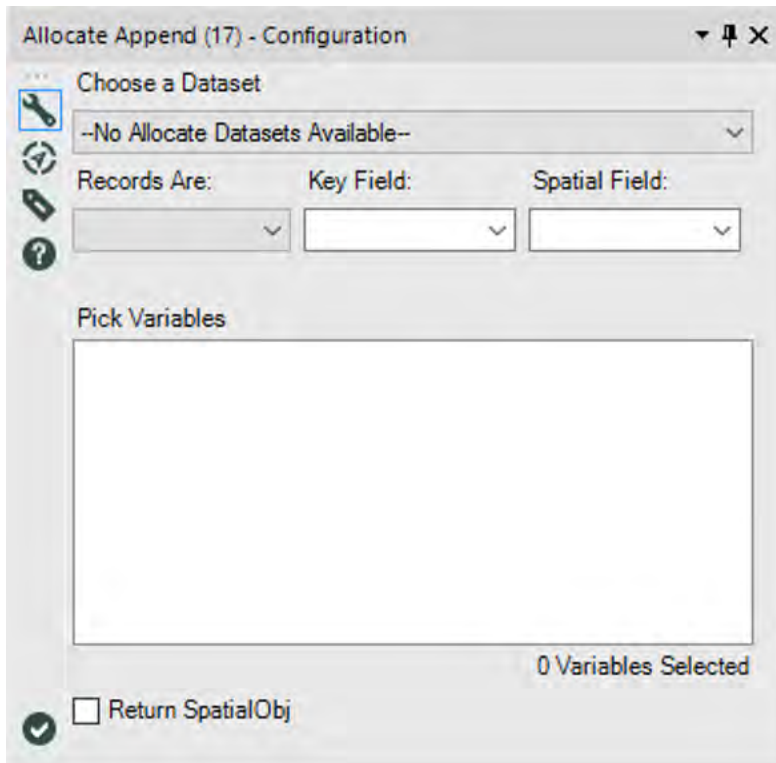
Figure 13-12 – Allocate Append Configuration

- *Choose a Dataset*: This component helps you to select the Allocate dataset that you want to work on.
- *Records Are*:
- *Key Field*: Choose the field that has key values.
- *Spatial Field*: Choose the field which has the spatial value.
- *Pick Variables*: This component helps you to pick up the variables that you need.

- *Return SpatialObj*: If this checkbox is checked, the spatial object will be returned for the particular Geography Allocate.

13.8 Allocate Metainfo

Figure 13-13 – Allocate Metainfo	The *Allocate Metainfo* tool gives the information about the installed Allocate datasets.		
	Group	**Input**	**Output**
	Demographic Analysis	None	MetaInfo Information
Note: Use this along with *Browse* tool to view the visual output.			

This tool can be used only if you have licensed installation of Allocate.

Properties Window:

The *Allocate Metainfo Configuration* window has two components.
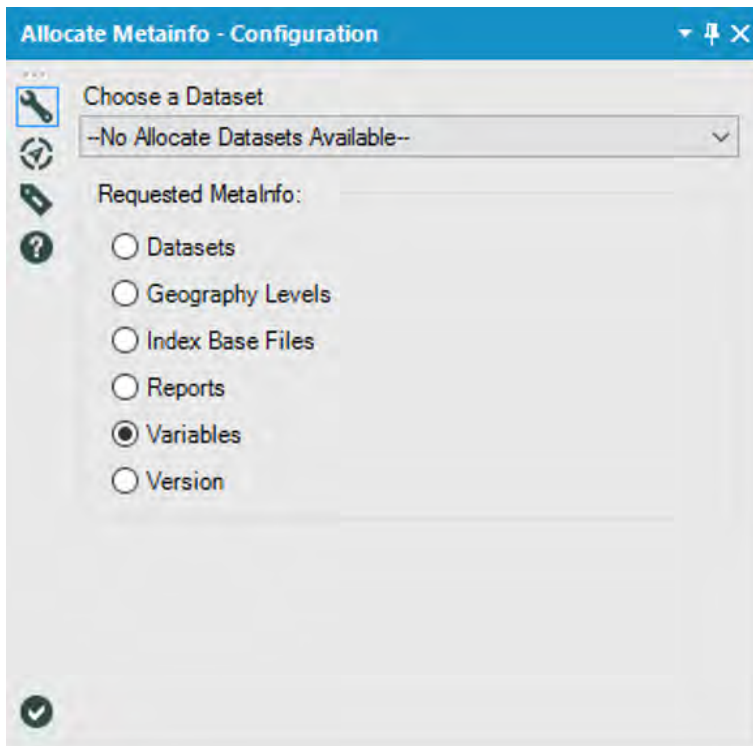
Figure 13-14 – Allocate Metainfo Configuration

- *Choose a Dataset*: This component helps you to select the Allocate dataset for which you want information like metadata.
- *Requested Metainfo*: This is a radio button list. It has list of options of the metainfo that you can retrieve with the help of this tool.

The options are:

- Datasets: This option returns all the Allocate datasets and its information – short name (internal name) and long name (Alteryx GUI name).

- Geography Levels: This option returns all the geography levels like short name, long name (descriptive name), alias short name and isIntermediate is a boolean variable which indicates if the field is a geography level or not.
- Index base files: This option returns all the index base files the user has created and saved in the dataset.
- Reports: This option returns installed *.rptx files along with the report names and its type.
- Variables: This option returns the information related to all the variables from the dataset. Information like name, name_long, description, category, SubCategory, Units, Vintage, Year, Action, Weight, Is Licensed and Data Table.
- Version: This returns the version number of the dataset.

13.9 Allocate Report

	The *Allocate Report* tool helps the user to retrieve custom reports		
	Group	**Input**	**Output**
Figure 13-15 – Allocate Report	Demographic Analysis	Allocate Dataset	Custom Reports
Note: Use this along with *Browse* tool to view the visual output.			

This tool can be used only if you have licensed installation of Allocate. The user can get the reports in any desired format and

can be exported to HTML, PDF, XLS, RTF and Alteryx Composer's PCXML format.

Properties Window:

The *Allocate Report Configuration* window has nine components.
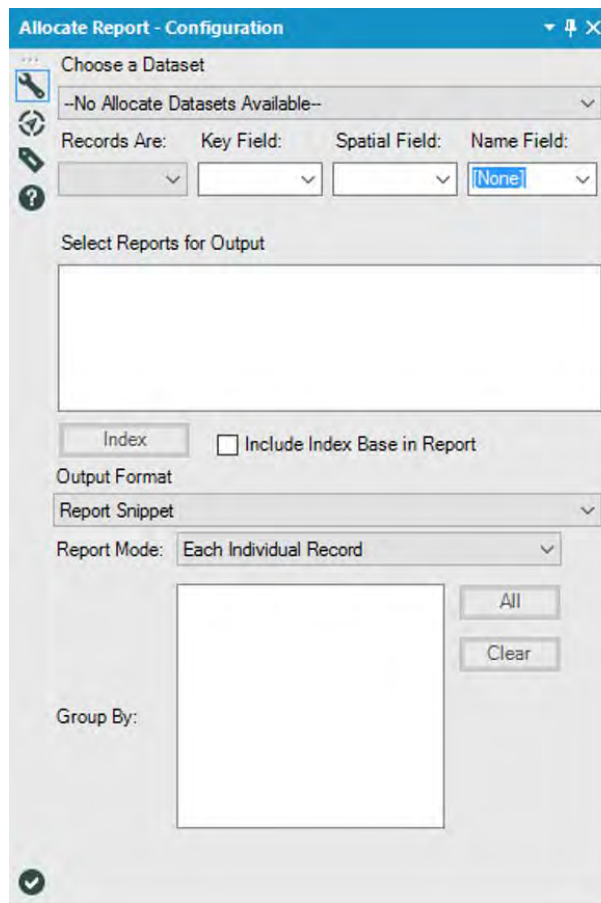
Figure 13-16 – Allocate Report Configuration

- *Choose a Dataset*: This component helps you to select the Allocate dataset that you want to work on.

- *Records Are*:
- *Key Field*: Choose the field that has key values.
- *Spatial Field*: Choose the field which has the spatial value.
- *Name field*: It acts as a row header.
- *Select Reports for Output*: The options for reports with pre-formatting will be available here to choose from. They are saved in Allocate Reports Directory.
- *Index*: This is optional. When this button is clicked, a geography tree is shown and you have to expand the tree levels so that they can be used as an index.
- *Include Index Base in Report*: This checkbox includes the index base file format in the report.
- *Output Format*: This is a dropdown menu with options to choose the output format for the report like PDF, Report Snippet, Alteryx Composer (PCXML), Microsoft Word etc.
- *Report Mode*: This specifies the mode of the report.
- *Group By*: Also, group by clause can be used to summarize the data.

13.10 Behavior Metainfo

Figure 13-17 – Behavior Metainfo

	The *Behavior Metainfo* tool gives the information about the installed Behavior Analysis datasets.		
	Group	**Input**	**Output**
	Behavior Analysis	None	Details of Behavior Analysis Datasets
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

The *Behavior Metainfo Configuration* window has two components.
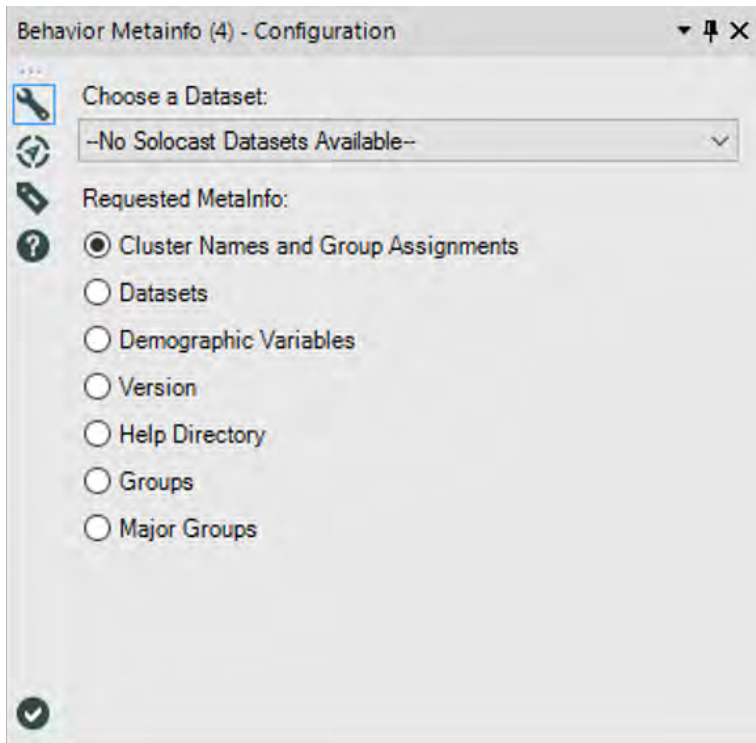
Figure 13-18 – Behavior Metainfo Configuration

- *Choose a Dataset*: This component helps you to select the Behavior Analysis dataset for which you want information like metadata. These datasets can be obtained if you have that particular license and subscription.
- *Requested Metainfo*: This is a radio button list. It has list of options of the metainfo that you can retrieve with the help of this tool.

- o *Cluster Names and Assignments:* The details for segmentation system classification will be returned using this option:
  - AGS datasets will return the following fields
    - **Cluster Number**: Cluster Identifier
    - **Cluster Name**: Short name of the cluster
    - **Cluster Link**: This is a link to report snippet which is linked to the online documentation with a description of a cluster. This report can be viewed using a Browse tool.
    - **Mosaic Group Number**
    - **Global Mosaic Groups Number**
  - STI Popstats Dataset will return the following fields:
    - **Cluster Number**: Cluster Identifier
    - **Cluster Name**: Short name of the cluster
    - **Cluster Link**: This is a link to report snippet which is linked to the online documentation with a description of a cluster. This report can be viewed using a Browse tool.
- o *Datasets:* It has information like the short name (an internal name which references the registry key of the dataset) and long name (Alteryx GUI name).
- o *Demographic Variables:* This option returns information about variables like household, population, person, adult etc.

o *Version:* This option returns the Solocast engine version number of the dataset.

o *Help Directory:* This option returns a web address to a directory which gives the description of clusters for the selected dataset.

o *Groups:* This option is related to the AGS datasets only. It returns fields like:

- **Mosaic Group Number***:* Mosaic Group Identifier
- **Mosaic Group Name***:* Short description of the group
- **Mosaic Groups Link***:* This is a link to report snippet which is linked to the online documentation with a description of the Mosaic Group. This report can be viewed using a Browse tool.

o *Major Groups:* This option is also related to the AGS datasets only. It returns fields like:

- **Global Mosaic Group Number***:* Global Mosaic Group Identifier
- **Mosaic Group Name***:* Short description of the Global Mosaic Group.
- **Mosaic Groups Link***:* This is a link to report snippet which is linked to the online documentation with a description of the Global Mosaic Group. This report can be viewed using a Browse tool.

13.11 Cluster Code

	The *Cluster Code* tool will append a Cluster Code field to the records using a Cluster ID
Figure 13-19 – Cluster Code	

Group	Input	Output
Behavior Analysis	Dataset	Dataset with a cluster code

Note: Use this along with *Browse* tool to view the visual output.

Properties Window:

The *Cluster Code Configuration* window has two components.

Figure 13-20 – Cluster Code Configuration

- *Choose a Dataset*: This component helps you to select the Behavior Analysis dataset with clustering system for which you want to find the clusters. These datasets can be obtained if you have that particular license and subscription.
- *Cluster Level ID*: This is a dropdown list from which you have to select the field from the dataset which is the Cluster Level ID (also called Block Group Key). This field should be a complete key. Partial IDs will not be allowed.

13.12 Compare Behavior

	The *Compare Behavior* tool compares two datasets		
	Group	**Input**	**Output**
Figure 13-21 – Compare Behavior	Behavior Analysis	One single data stream with two profile sets	Gives comparing analysis
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

The *Compare Behavior Configuration* window has seven components.
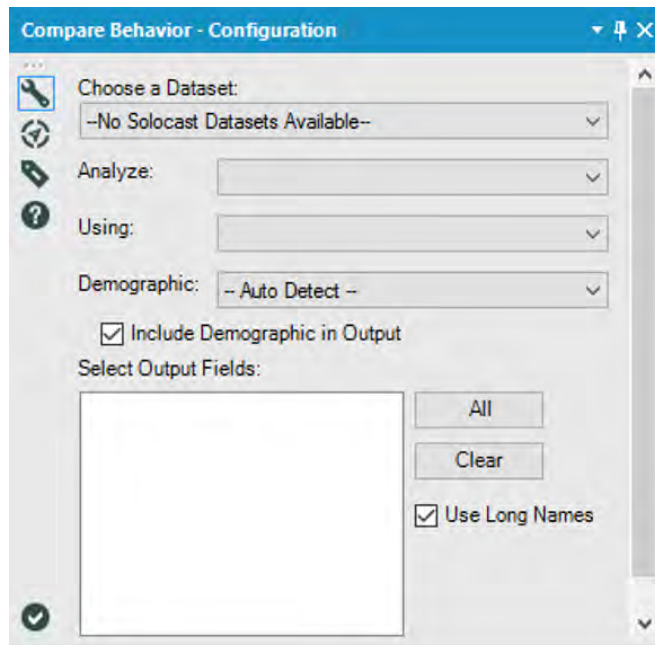
Figure 13-22 – Compare Behavior Configuration

- *Choose a Dataset*: This component helps you to select the single data stream. This data stream is a combined set of two profile sets with clustering system whose behavior you want to compare. These datasets can be obtained if you have that particular license and subscription.
- *Analyze*: This component is a dropdown list from where you need to choose a field which represents the profile to use for analysis.
- *Using*: This component is a dropdown list from where you need to choose a field which represents the profile to use to compare the Analyze Profile.
- *Demographic*: This component represents the classification level of the Profile. The options are Auto-Detect, Household, Person or Adult.

- *Include Demographic in Output*: This option is a checkbox to include the demographic information in the output.
- *Select Output Fields*: This component helps to decide which fields should be included in the output. The buttons All and Clear help to select the fields.
- *Use Long Names*: With this checkbox, the long descriptive names of the fields will be displayed instead of the shorter names.

13.13 Create Profile

Figure 13-23 – Create Profile	The *Create Profile* tool takes the data stream and creates a Behavior Profile		
	Group	**Input**	**Output**
	Behavior Analysis	Data Stream	Profile is created
Note: Use this along with *Browse* tool to view the visual output.			

This profile can be created using a database, Polygon Files, Rings or Standard Geographies. We can use a Write Behavior Profile Set Tool to build the Profiles.

Properties Window:

The *Create Profile Configuration* window has three components.

Figure 13-24 – Create Profile Configuration

- *Choose a Dataset*: This component helps you to select the single data stream with a set of profiles with clustering system. These datasets can be obtained if you have that particular license and subscription.
- *Mode*: This component lets you select the mode from the dropdown menu. The options are Spatial Object, Known Geography Key, Combine Profiles, Cluster Code and Cluster Level ID.
- *Summarize*: This component decides the method to summarize the data. The options are as follows:
  - *All Records*: This option combines all records into one single record.

   o *Group By*: This option summarizes the common values into groups.

   o *None*: This option means that each record is different.

- *Profile Name:* This is the field where you can name the Profile.

- *Spatial Object:* This is the dropdown menu from which you need to select the Spatial Object from the data stream.

- *Base Profile:* This is the dropdown menu from which you need to select the Base Profile field from the data stream.

13.14 Detail Fields

Figure 13-25 – Detail Fields	The *Detail Fields* tool returns detailed information at the Cluster or Group specific to that Profile.		
	Group	**Input**	**Output**
	Behavior Analysis	Data Stream	Detailed Information
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

   The *Detail Fields Configuration* window has eight components.
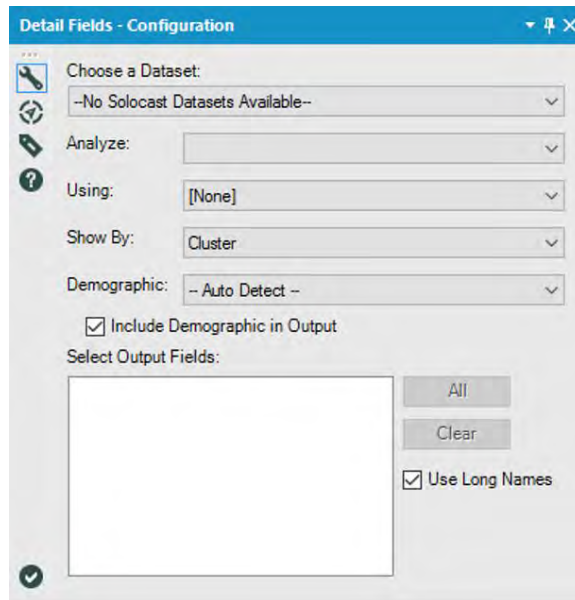
Figure 13-26 – Detail Fields Configuration

- *Choose a Dataset*: This component helps you to select the single data stream with a set of profiles with clustering system. These datasets can be obtained if you have that particular license and subscription.
- *Analyze*: This component is a dropdown list from where you need to choose a field which represents the profile to use for analysis.
- *Using*: This component is a dropdown list from where you need to choose a field which represents the profile to use to compare the Analyze Profile.
- *Show By*: This component gives the option to show the data at a record level. The options are Cluster or Group.
- *Demographic*: This component represents the classification level of the Profile. The options are Auto-Detect, Household, Person or Adult.

- *Include Demographic in Output*: This option is a checkbox to include the demographic information in the output.
- *Select Output Fields*: This component helps to decide which fields should be included in the output. The buttons All and Clear help to select the fields.
- *Use Long Names*: With this checkbox, the long descriptive names of the fields will be displayed instead of the shorter names.

13.15 Profile Input

  Figure 13-27 – Profile Input	The *Profile Input* tool helps you to select a Profile Set to use as an input to the workflow	
	Group	**Input**	**Output**
	Behavior Analysis	None	Name, Profile, and SpatialObj
Note: Use this along with *Browse* tool to view the visual output.		

Properties Window:

The *Profile Input Configuration* window has three components.
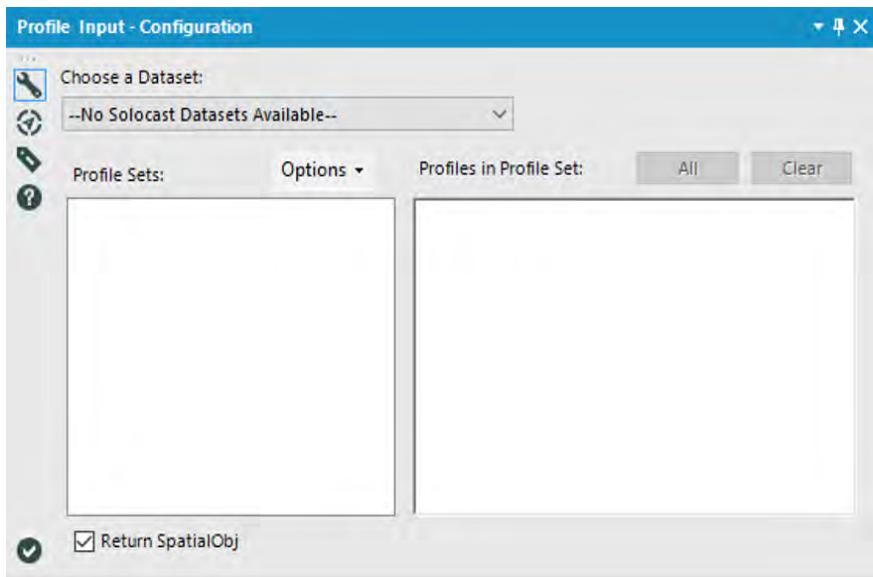
Figure 13-28 – Profile Input Configuration

- *Choose a Dataset*: This component helps you to select the single data stream with a set of profiles with clustering system. These datasets can be obtained if you have that particular license and subscription.
- *Profile Sets*: This pane has the list of Profile Sets.
- *Profiles in Profile Set*: When you select, the Profile Set in the left pane and the information about that set will appear on the right pane.
- *Options*: This is the dropdown menu which has a list of options available while selecting Profile Sets. The options are Open Profile Set, Clear All, Expand All and Collapse All.
- *Return Spatial Obj*: This is a checkbox which if checked returns the Spatial Object field with the geography level highlighted. If it is unchecked, it returns null.

13.16 Profile Output

\n\nFigure 13-29 – Profile Output	The *Profile Output* tool takes an input stream and writes it into Profile Set output *.scd file.		
	Group	**Input**	**Output**
	Behavior Analysis	Data Stream	None
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

The *Profile Output Configuration* window has seven components.

Figure 13-30 – Profile Output Configuration

- *Choose a Dataset*: This component helps you to select the single data stream with a set of profiles with clustering system. These datasets can be obtained if you have that particular license and subscription.
- *Output File*: In this textbox, you need to specify the name and location of the output file.
- *Output Options*: This is a dropdown menu of types of output methods. The options are:

- o *Create New Profile*: This option lets you create a new Profile Set.
- o *Append to Existing Profile:* This option lets you add profiles to an existing Profile Set.
- o *Overwrite Profile:* This option removes the existing Profile Set and writes a new on with the same name.
- o *Update Profile:* The new values are updated in the existing Profile Set.

- *Profile Name*: This is the field where you can name the Profile.
- *Profile*: This is the dropdown menu from which you can select the profile field from the data stream.
- *Spatial Object*: This is the dropdown menu from which you need to select the Spatial Object from the data stream.
- *License Required*: This component is a checkbox. If it is checked, you need a license to use the Profile Set.

13.17 Optimization

	The *Optimization* tool helps to find the values of the decision variables that minimize or maximize the objective.		
	Group	**Input**	**Output**
Figure 13-31 – Optimization	Prescriptive	O: To input names of decision variables A: Matrix	I: Interactive Dashboard D: Summary, Variables and

		Input Mode	Constraints output
		B:	S: Single Table with Optimal Value
		Q: Using this option, the quadratic value u a wtdu room.	
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

     The *Optimization Configuration* window has six components.
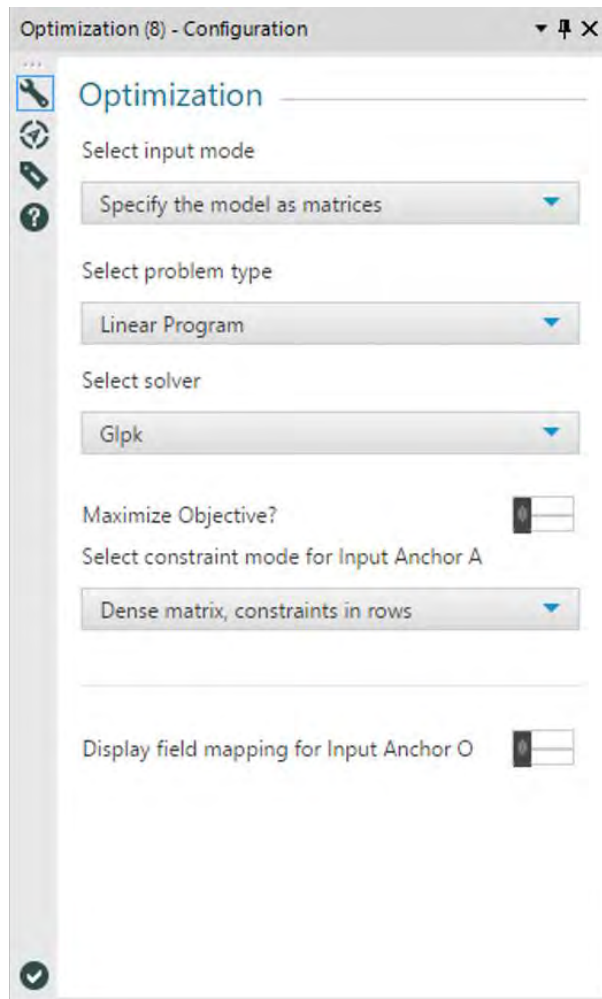
Figure 13-32 – Optimization Configuration

- *Select input mode*: This option lets you select the type of input.
- *Select problem type*: There are different types of problems types available like Linear Program, Mixed Integer Program, and Quadratic Program.
- *Select solver*: You select the feasible option.
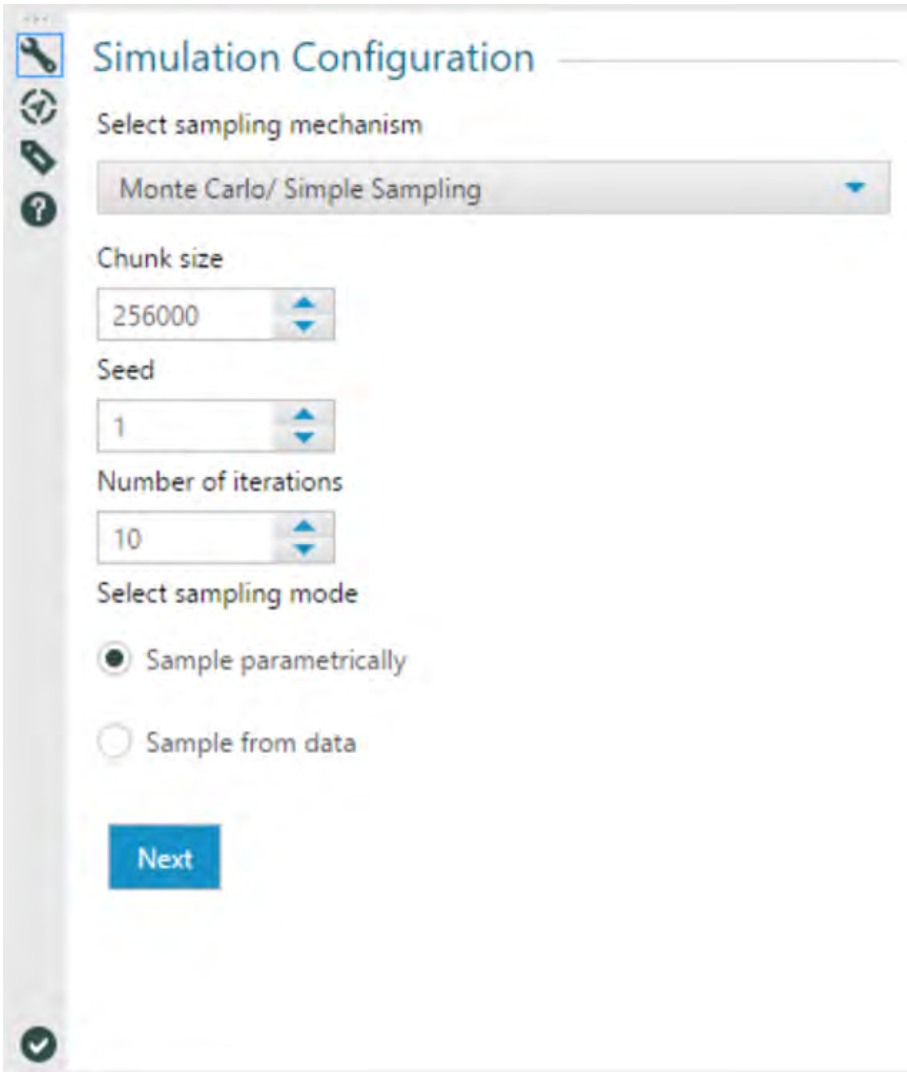
- *Maximize Objective*: This option is activated if there is a maximization problem.
- *Select constraint mode for Input Anchor A*: This is a dropdown menu to select the constraint mode for input A.
- *Display field mapping for Input Anchor O*: This option will help you to display field mappings for input A.

13.18 Simulation Sampling

	The *Simulation Sampling* tool performs sampling and gives you sampled data.		
	Group	**Input**	**Output**
Figure 13-33 – Simulation Sampling	Prescriptive	D: Sample Data S: Simulation Data	D: Data Output. Result of Simulation
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

The *Simulation Sampling Configuration* window has seven components.

Figure 13-34 – Simulation Sampling Configuration

- *Select Sampling Mechanism*: Here we can choose the sampling type – Monte Carlo/Simple Sampling or Latin HyperCube / Stratified Sampling.
- *Chunk Size*: This is the maximum size of data which can be evaluated at a time instant. It is also the maximum size of strata in stratified sampling.
- *Seed*: This random seed is used for sampling.
- *Number of iterations*: This field is the number of samples to select. If the dataset is connected to S input, then that option is not available.
- *Select Sampling Mode*: It allows you to select a type of sampling mode.
  - *Sample parametrically*: For this type of sampling, we do not need to connect to the D input.
  - *Sample from data*: Connect to the D input or enter the data manually.

13.19 Simulation Scoring

	The *Simulation Scoring* tool gives a range of possible values to predict the mean predicted value while considering the error distribution.		
Figure 13-35 – Simulation Scoring	**Group**	**Input**	**Output**
	Prescriptive	M: Model Object V: Validation Dataset	D: Data to be scored along with the simulated score.

Figure 13-35 – Simulation Scoring

		S: Simulatio n data	
Note: Use this along with *Browse* tool to view the visual output.			

Properties Window:

The *Simulation Scoring Configuration* window has four components.
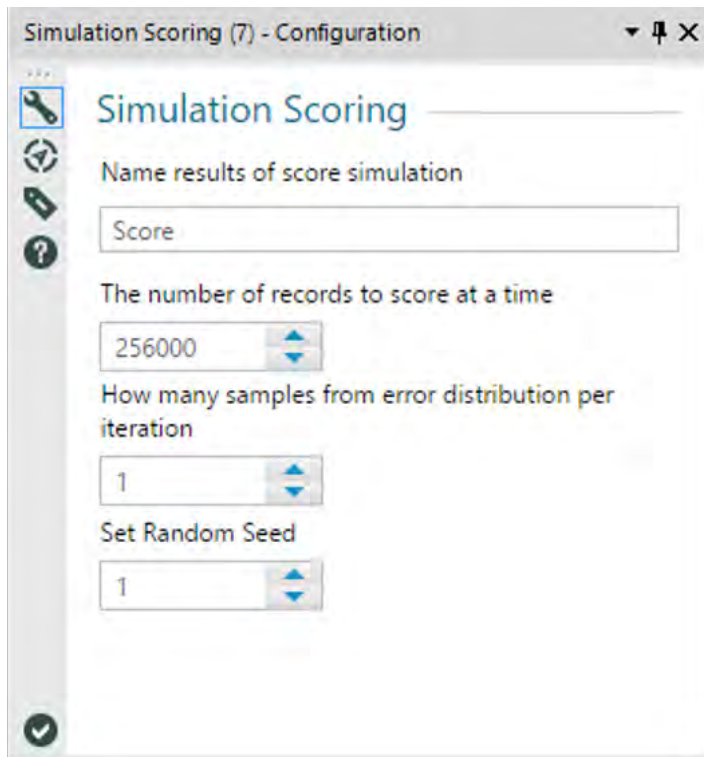
Figure 13-36– Simulation Scoring Configuration

- *Name results of score simulation*: This is the name of the field for results.
- *The number of records to score at a time*: This is the number of records allowed for each chunk of data in the input.
- *How many samples from error distribution per iteration*: This is the number of draws from the error distribution of the model for every record.
- *Set Random Seed*: This is a Random Seed.

13.20 Simulation Summary

<table>
<tr>
<td rowspan="3">Figure 13-37 – Simulation Summary</td>
<td colspan="3">The *Simulation Summary* tool gives visualizations of various operations on distributions.</td>
</tr>
<tr>
<td>Group</td>
<td>Input</td>
<td>Output</td>
</tr>
<tr>
<td>Prescriptive</td>
<td>Alteryx Data Stream with two fields</td>
<td>D: Data Output

R: Output displays the Sensitivity Analysis Report.

I: Output shows an interactive dashboard.</td>
</tr>
<tr>
<td colspan="4">*Note:* Use this along with *Browse* tool to view the visual output.</td>
</tr>
</table>

Properties Window:

The *Simulation Summary Configuration* window has two components.
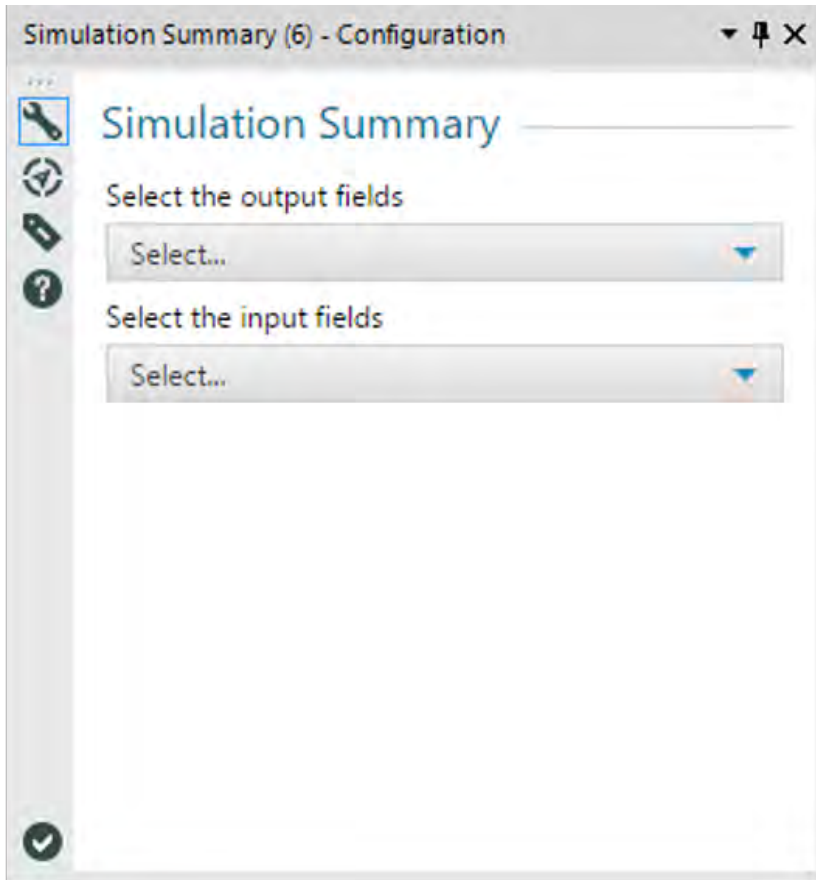
Figure 13-38 – Simulation Summary Configuration

- *Select the output fields*: Using this dropdown menu you can select the fields to analyze the effects of independent variables.
- *Select the input fields*: Using this dropdown menu you can select the fields to analyze the effects of dependent variables.

13.21 Which car has the best performance?

Let us start with having the Auto dataset as the data source for the Alteryx workflow. The file Auto.csv is located on the desktop. Drag the *Input Data* tool from the *In/Out* group and connect to the file.
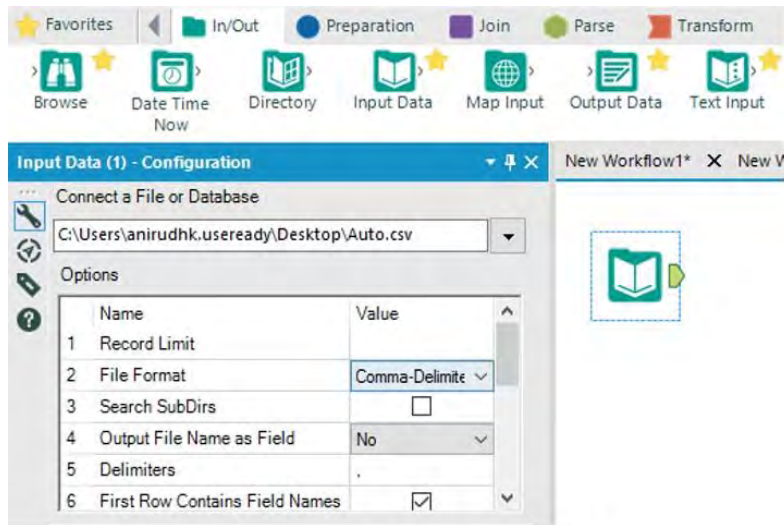
Figure 13-39 – First let's get the data into Alteryx

Basic Data Profile tool from the *Data Investigation* group will us understand the metadata for each of the columns present in the dataset. Add *Browse* to get a detailed report. Run (Ctrl+R) the workflow and select the *Browse* tool to view the detailed report.
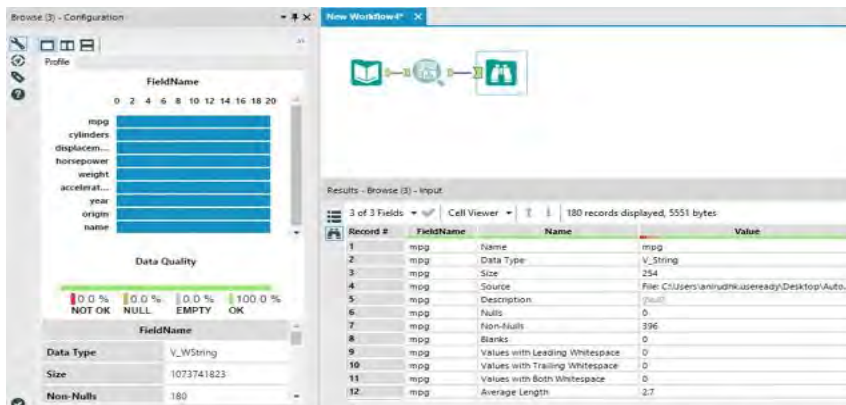
Figure 13-40 Running the Data through the Basic Data Profile

From the output, we understand that the Data Type is V_String for all the columns.

Add the *Field Summary* tool followed by *Browse* to *Output O* to view the statistical description for each of the columns in the data set. Select 'All' in configuration window while configuring the *Field Summary* tool.
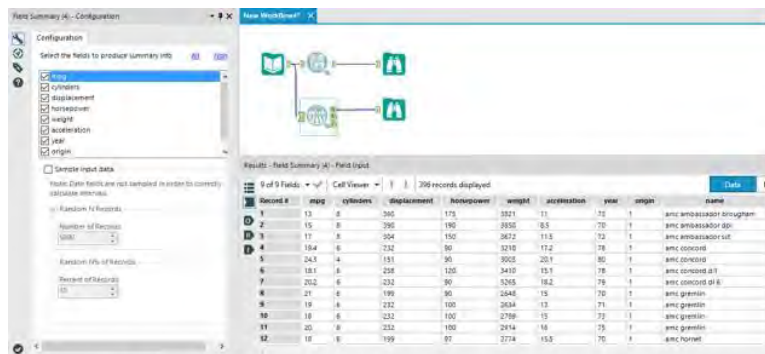
Figure 13-41 Let's check Field Summary to know the data types

Run (Ctrl+R) the workflow and click on *Browse* to view the output.
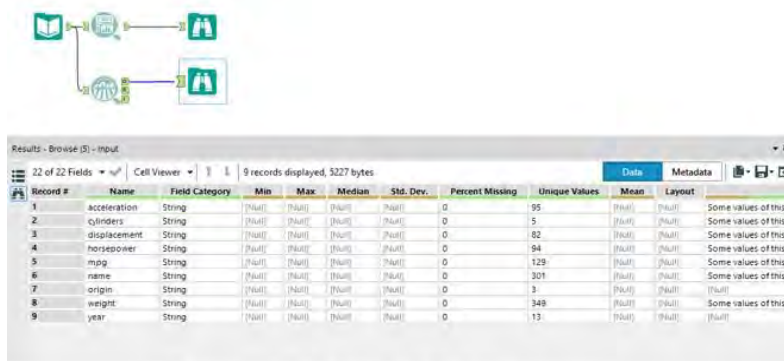
Figure 13-42 Output of Field Summary

A summary of each of the columns can be studied from the output. For example, we can observe that the percent missing is 0 for all the columns and understand that none of the columns have missing values.

Except for Name, Origin and Year, rest are supposed to be numeric. So, we need to change the data type for each of those to fields to a numeric data type like *Double.* Use the *Select* tool from *Preparation* group to change the data types. Click on the dropdown under *Type* to make the appropriate selection.
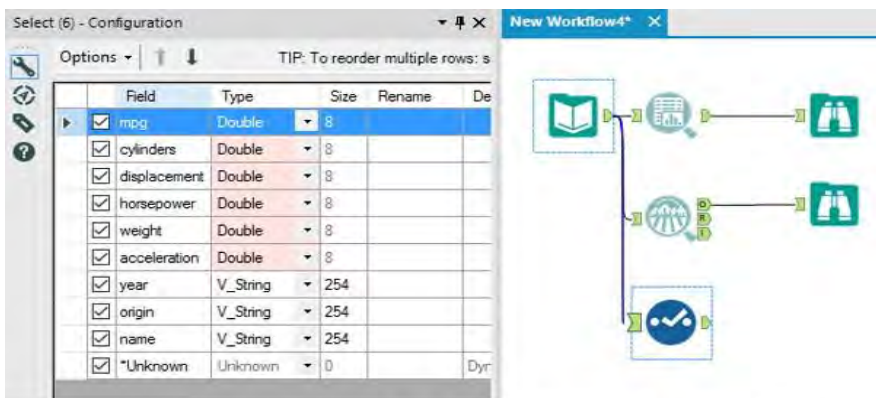
Figure 13-43 Selection of Columns for Further Analysis

Name column has the names of all the cars. We can use the *Formula* tool from the *Preparation* group to get the Brand of each car.

The first word in the *Name* can be considered as the Brand. The formula below can be used to get the Brand of the car.

left([name],FindString([name], " "))
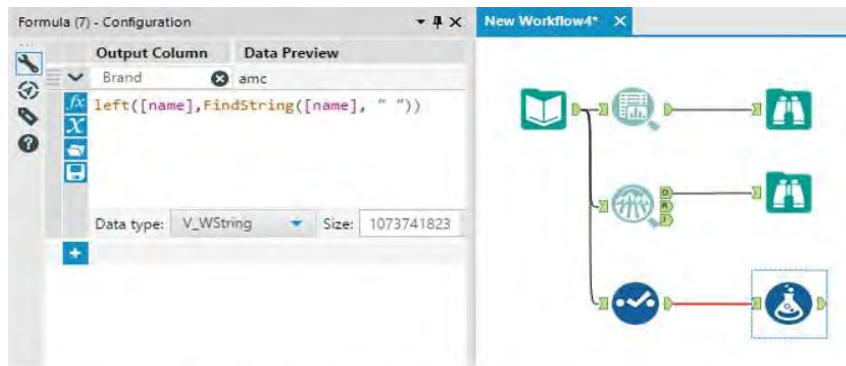
Figure 13-44 Retrieving the Brand name from the Column

We would like to understand the correlation between cylinders and acceleration and also between mpg and displacement to conclude on the performance of each car.

We can connect the current output to two *Spearman Correlation* tools present in the Data Investigation group and join these with the output using the *Join Multiple* tool present in the *Join* group.
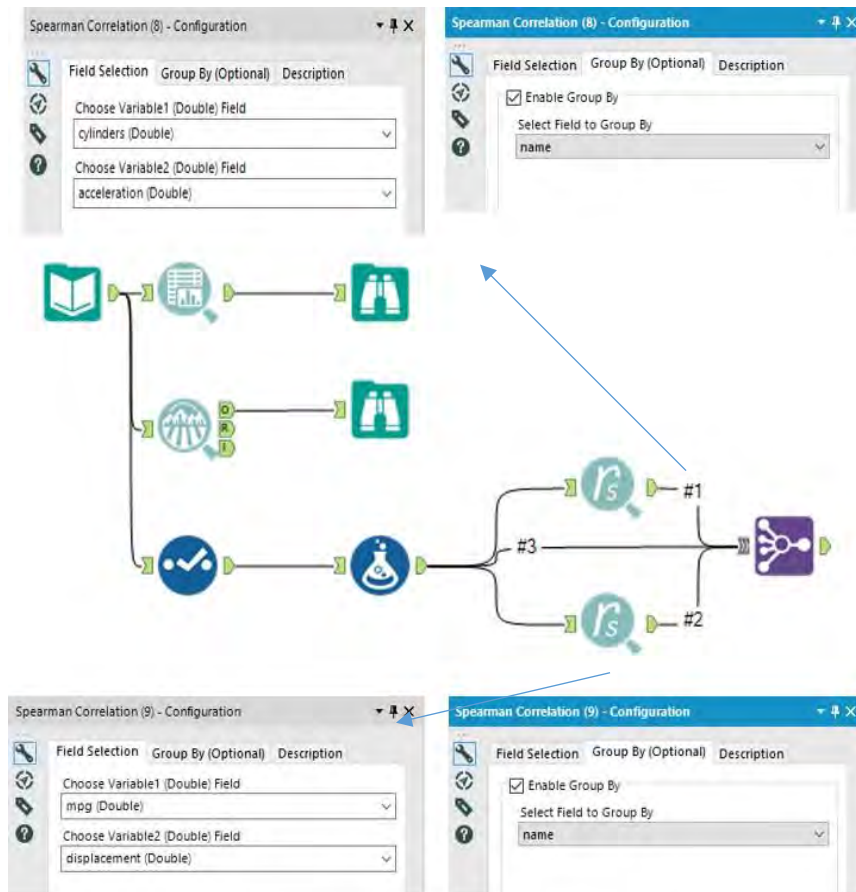
Figure 13-45 Introducing Spearman Correlation based on Grouping

The *Spearman Correlation* tools are configured as shown in the figure above. They are grouped by *Name* so that we get the values for each car. The outputs of this tool are similar to what is shown below.

Record #	Group	Result
1	amc ambassador brougham	[Null]
2	amc ambassador dpl	[Null]
3	amc ambassador sst	[Null]
4	amc concord	-1

Figure 13-46 Sample Output of Spearman Correlation

These outputs are joined (using *Multiple Join*) with the previous output (output at *Formula*) using the following condition.
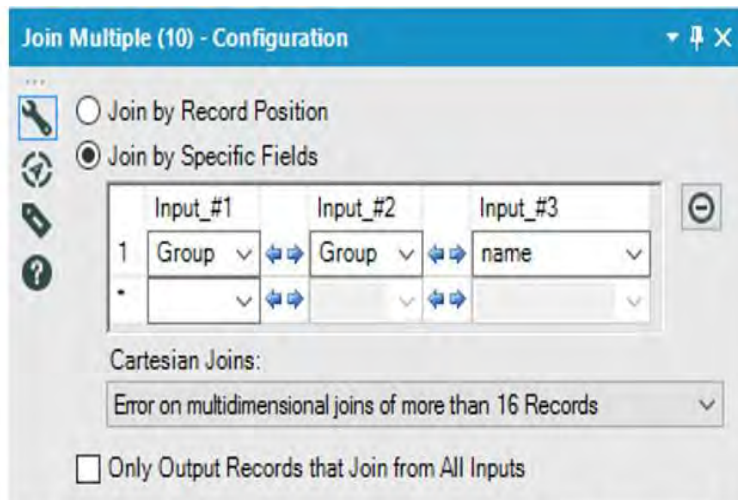
Figure 13-47 Joining the Spearman Correlation Outputs based on Group

We can use a *Select* tool after the *Multiple Join* to carry forward only the required fields and rename the ones required.
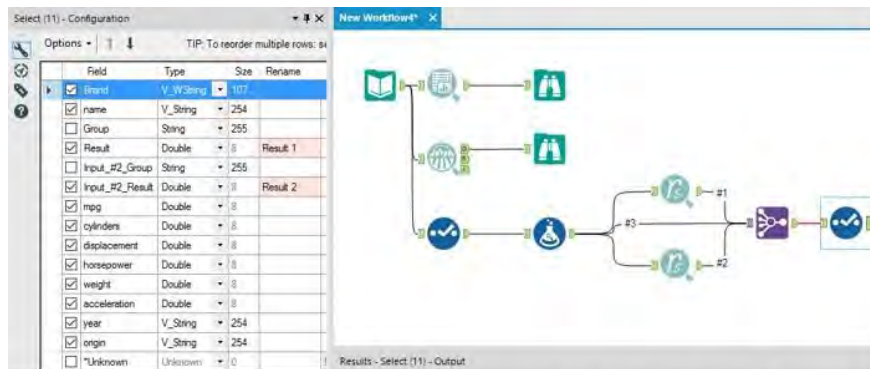
Figure 13-48 Selecting Relevant Columns for further Analysis

Using the *Formula* tool, we can create a column to get the priority. This can be followed by a *Sort* tool from *Preparation* group sort the data according to Priority. Add *Browse* to view the data.
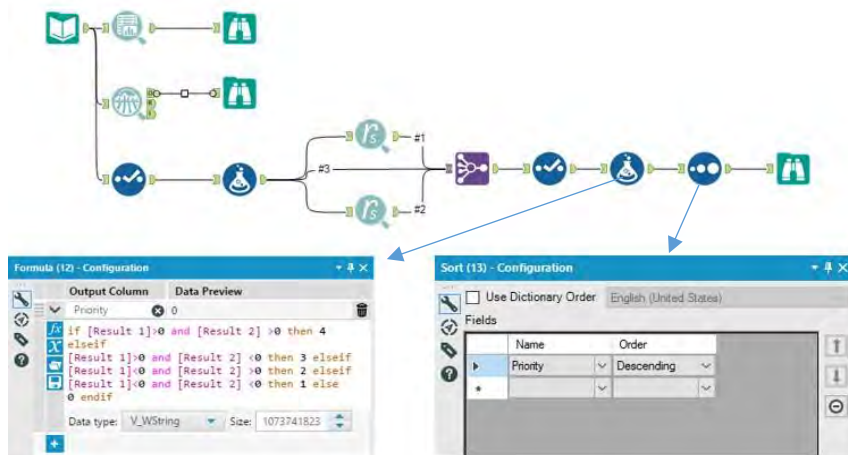
Figure 13-49 Data Transformation to calculate Priority

We get a result similar to the one displayed below.

Figure 13-50 Data Output Post Priority calculation

Higher the *Priority* value, better is the performance.

To understand the correlation between each of the fields (only numeric), we connect the data to *Pearson Correlation* tool under *Data Investigation* group. Select all the numeric fields in the configuration window and connect it to *Browse* to view the output.

Figure 13-51 Pearson Correlation Variables Selection

The figure below shows the output.

Record #	FieldName	mpg	cylinders	displacement	horsepower	weight	acceleration
1	mpg	1	-0.776606	-0.804501	[Null]	-0.831784	0.422462
2	cylinders	-0.776606	1	0.951008	[Null]	0.897083	-0.505084
3	displacement	-0.804501	0.951008	1	[Null]	0.933066	-0.544692
4	horsepower	[Null]	[Null]	[Null]	1	[Null]	[Null]
5	weight	-0.831784	0.897083	0.933066	[Null]	1	-0.419933
6	acceleration	0.422462	-0.505084	-0.544692	[Null]	-0.419933	1

Figure 13-52 Pearson Correlation Output to show how Variables are related to each other

CHAPTER 14

Alteryx for the Developer

To...   Alteryx Consultants

Cc...

Send

Subject   Energy Savings Implementation

Hey,

We are a firm operating in the Energy Management domain . We are planning to implement a Proof of Concept for a customer in Chicago around the New Year season.

We have to take into count various external factors when planning for these activities. We will be using data for Weather, Crude Oil, Utilities and need your assistance in putting it in a reasonable format for us to be able to take a clear call about the activity.

Here is a link to get the data for any day from a source we trust-
"http://www.wunderground.com/history/airport/KMDW/2017/01/01/CustomHistory.htm

Can you please help us with this weather information first and we can then look at the others?

Regards

14.1 Tools & Concepts

Tools	Concepts
API Output	
Base64 Encoder	
Blob Convert	
Blob Input	
Blob Output	
Block Until Done	
Dynamic Input	
Dynamic Replace	
Dynamic Select	
Field Info	
Run Command	
Test	
Throttle	
JSON Build	
Transpose In-DB	

14.2   API Output Tool

Figure 14-1-API Output Tool	Return the results of a data stream directly to an API callback function. For use with custom application development.		
	Group	**Input**	**Output**
	Developer	None	See Below

The API Output tool allows users to display the contents of a data stream in the Results window as a comma-delimited, quote-qualified string. No configuration is needed for the tool.

The first record that displays in the Results window contains the field names, and each additional record contains a row of data from the incoming data stream.

The API Output tool outputs a maximum of 255 characters per row, and the data from each field is contained within double quotes. If a string value contains double quotes, each quote is replaced by two consecutive quotes. Blob fields display as the size of the blob (e.g., "14 Bytes").

Figure 14-2-API Output Configuration

14.3   Base64 Encoder

	The Base 64 Encoder macro issues a base 64 encode a string for a specified string field.		
	Group	**Input**	**Output**
Figure 14-3-Base64 Encoder	Developer	All file Types	See Below

Output: Includes all incoming data fields plus a field named Base64_Encoded that includes the base 64 encode string for the specified string field.

Example: This Macro is used in our Social Media tools

String Field	Base64_Encoded
Writing Macros is fun!	V3JpdGluZyBNYWNyb3MgaXMgZnVuIQ==

Useful where there is a need to encode binary data that needs to be stored and transferred over media that is designed to deal with textual data. This is to ensure that the data remains intact without modification during transport.

Configuration Properties

Use the drop down to pick the string field from the incoming data stream to encode in base 64.
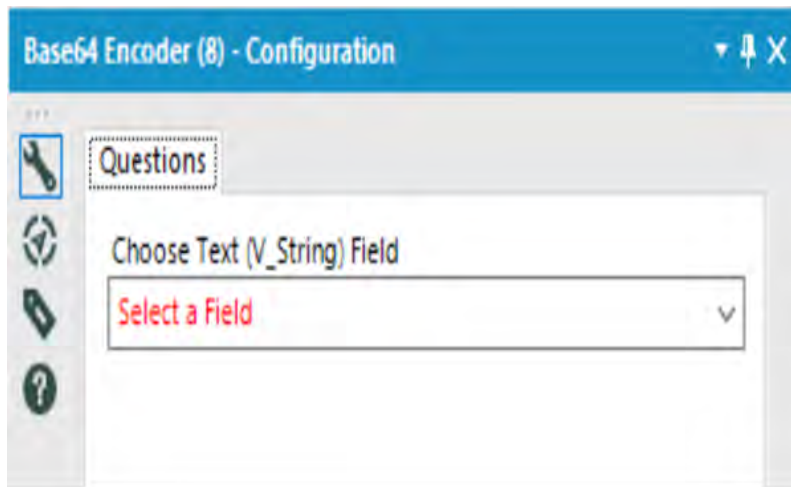
Figure 14-4-Base64 Encoder Configuration

14.4   Blob Convert

	The Blob Convert tool will take different data types and either convert them to a Binary Large Object (Blob) or take a Blob and convert it to a different data type.		
 Figure 14-5-Blob Convert	**Group**	**Input**	**Output**
	Developer	Any Data type	None
Example: Convert a PNG, GIF or JPG Blob to Report Snippet. Allows the Reporting Tools to recognize the incoming images as report snippets for building reports.			

Configuration Properties

Conversion mode: Choose the type of conversion to perform. Choices are Convert from a blob field or Convert to a blob field. Based on your selection, other options are available and listed out below.

Convert from a Blob field:
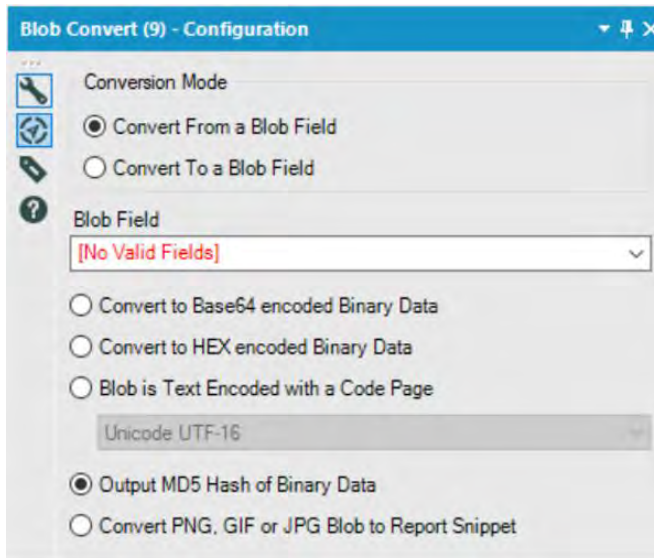
Figure 14-6-Convert from a Blob Filed

Takes an incoming Blob and converts it to another data type.

Blob Field: Use the drop-down list to select the data field that contains the Blob data to be converted.

Specify a data type to convert the Blob to.

- Choices are:
- Convert to Base64 encoded Binary Data
- Convert to HEX encoded Binary Data
- Blob is Text Encoded with a Code Page: It has a drop-down list to choose

- Output MD5 Hash of Binary Data
- Convert PNG, GIF or JPG Blob to Report Snippet: choosing this option will allow the Reporting Tools to recognize the incoming images as report snippets for building reports.
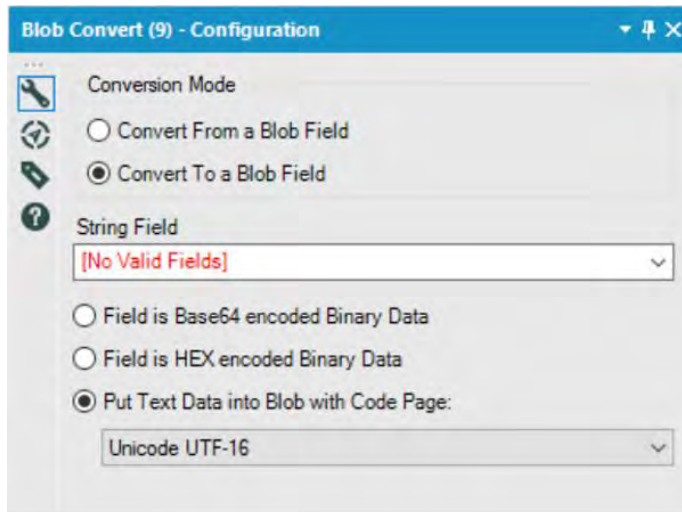
Figure 14-7-Convert to a Blob Field

Convert to a Blob Field:

Takes incoming data and converts it to a Blob.

String Field: Use the drop-down list to select the data field that will be converted to a Blob.

Specify the type of data contained within the specified field.

Choices are:

- Field is Base64 encoded Binary Data
- Field is HEX encoded Binary Data

- Put Text Data into a Blob with Code Page: It has a drop-down list to choose from.

14.5   Blob Input

	The Blob Input tool will read a Binary Large Object such as an image or media file, by browsing directly to a file or passing a list of files to read.		
Figure 14-8-Blob Input	**Group**	**Input**	**Output**
	Developer	Any File Type	None
Example: Read a list of documents from disk either as blobs or encoded strings.			

Configuration Properties
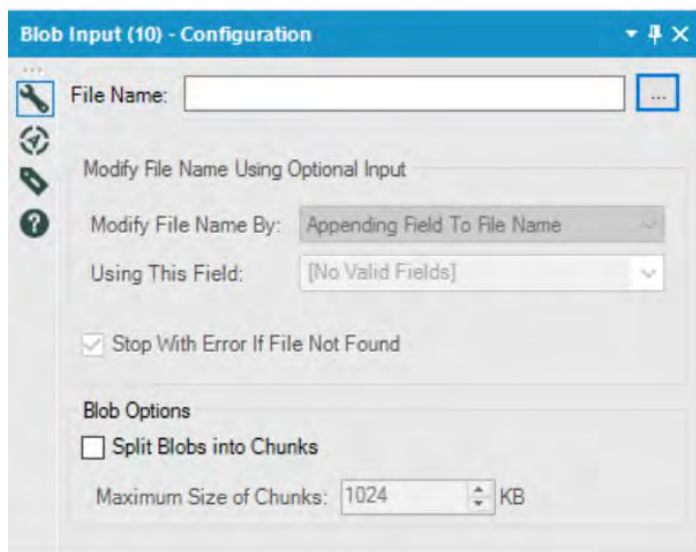
Figure 14-9-Blob Input Configuration

The Blob Input tool can either be an input tool or a pass-through tool where a list of files is passed to the tool and the tool reads them.

Input tool Configuration

Specify the file name of the Blob to be read. All file types are supported.

Pass through tool Configuration

1. The incoming data stream should contain a list of files (Blobs) to be read.

2. Specify the file name of the Blob to be read. All file types are supported. Alteryx will be replacing the string with the contents of the incoming stream.

3. *Modify File Name Using Optional Input*: Select the appropriate action to replace the string specified in the file name above.

- *Modify File Name By:* Choose the action to replace what was specified in the file name above. For example, the file name is set to: .\data\images\image.png
- *Appending Field to File Name*: The contents of the specified field are added after the file name and before the extension.
  Using the example above, say the value of the specified field is "apple" the resulting file name will be: .\data\images\imageapple.png
- *Prepending Field To File Name*: The contents of the specified field are added before the file name.
  If the value of the specified field is "banana" the resulting file name will be: .\data\images\bananaimage.png

- *Replacing File Name with Field*: The contents of the specified field will be used as the file name before the extension.
  If the value of the specified field is "grape" the resulting file name will be: .\data\images\grape.png
- *Replacing Entire Path With field*: The contents of the specified field will be used as the location directory, file name, and extension.
  If the value of the specified field is "C:\Temp\rainbows_and_unicorns.jpg" the resulting file name will be: C:\Temp\rainbows_and_unicorns.jpg
- *Using This Field*: Use the drop-down list to select the incoming field that contains the data replacement for the file name.

4. *Stop with Error If File Not Found*: When checked, Alteryx will throw an error if the file is not present. The error will say Cannot access file: (path).

Input or Pass-through Tool Configuration

1. Specify the file name of the Blob to be read. All file types are supported.

2. *Blob Options*: Provides options for handling large files.

- Split Blobs into Chunks: When checked, blobs will be broken up into smaller blobs, or "chunks" and sent downstream as individual records. Each record will contain this blob and an additional field called "BlobNumber" which is the sequence number of each chunk. The first is 1, the second is 2, etc.
- Maximum Size of Chunks: Allows the size (KB) of each chunk that will be created to be specified. If the original blob that is read is not an exact multiple of the chunk size,

the final chunk will be smaller than this size. If the original blob is smaller than the chunk size, it will be output as a single blob and will still have the "BlobNumber" field added to the record.

14.6   Blob Output

	The Blob Output tool writes out each record into its own file.		
	Group	**Input**	**Output**
Figure 14-10-Blob Output	Developer	All File Type	None
Example: Write a list of documents to disk either as blobs or encoded strings.			

1. Specify the file name of the Blob to write.

Blob Output (17) - Configuration

File Name:

Modify File Name By: Appending Field To File Name

Using This Field: [No Valid Fields]

Blob Field: [No Valid Fields]
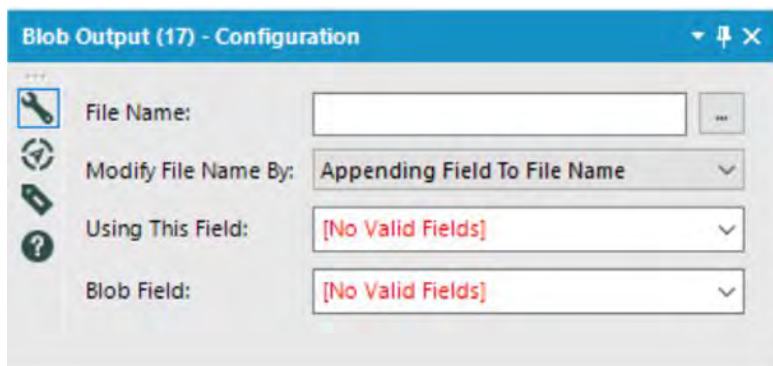
Figure 14-11-Blob Output Configuration

2. Modify File Name By: Choose the action to replace what was specified in file name above. For example, the file name is set to: .\data\images\image.png
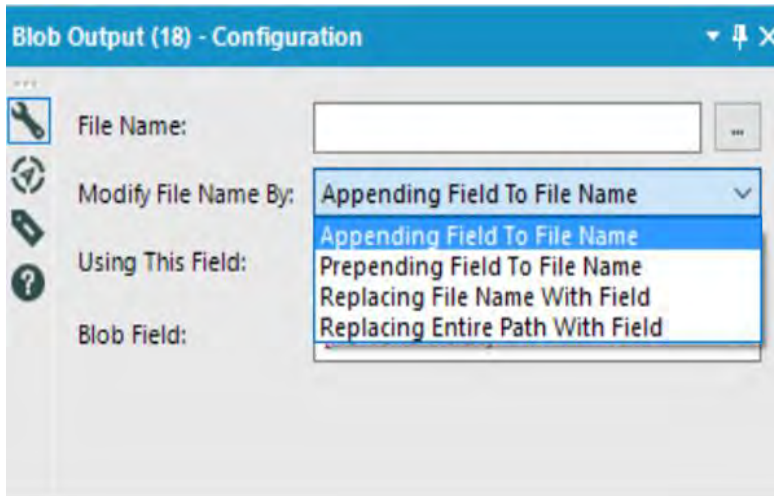
Figure 14-12-Blob Output Configuration

Appending Field to File Name: the contents of the specified field are added after the file name and before the extension.

Using the example above, say the value of the specified field is "apple" the resulting file name will be: .\data\images\imageapple.png

Prepending Field to File Name: the contents of the specified field are added before the file name.

If the value of the specified field is "banana" the resulting filename will be: .\data\images\bananaimage.png

Replacing File Name with Field: the contents of the specified field will be used as the file name before the extension.
If the value of the specified field is "grape" the resulting file name will be: .\data\images\grape.png

Replacing Entire Path with field: the contents of the specified field will be used as the location directory, file name, and extension. The value of the specified field is "C:\Temp\rainbows_and_unicorns.jpg" the resulting file name will be: C:\Temp\rainbows_and_unicorns.jpg

1. Using this field: Use the drop-down list to select the incoming field that contains the data replacement for the file name.

2. Blob Field: Use the drop-down list to select the incoming field that contains the Binary Large Object data.

14.7   Block Until Done

	The Block Until Done tool stops downstream processing until the very last record has arrived. On output, it also ensures that only a single output stream gets records at a time. Subsequent streams will be blocked until all the records are pushed through the first.

Figure 14-13-Block Until Done

Group	Input	Output
Developer	None	See Below

Note: It will slow down execution because it requires a large amount of memory. For larger streams, it may require a temp file.

Output: The output streams will write in the sequential order of the three output anchors.

> *Example*: Used when you want to make sure part of the process is finished before the next part of the process begins

- *The Block Until Done* tool has no configuration.

Figure 14-14-Block Until Done Configuration

- A common scenario is that we are reading and then writing to the same file. Block Until Done ensures that the reading of the file will be closed before Alteryx tries to write.
- We might also use it for writing multiple tables to an Excel file. The Excel file will only let one output tool access a single file at a time.

14.8 Dynamic Input

Figure 14-15 – Dynamic Input	The *Dynamic Input* tool allows us to read from input files or databases at runtime and dynamically choose what records to read in.		
	Group	**Input**	**Output**

	Developer	Any data stream	Any data stream
Note: The *Dynamic Input* tool allows for dynamically generated queries. Alteryx filters the data and only returns the user specified criteria and joins it to the data coming into the tool.			

Properties Window:

The *Dynamic Input Configuration* window has three components.
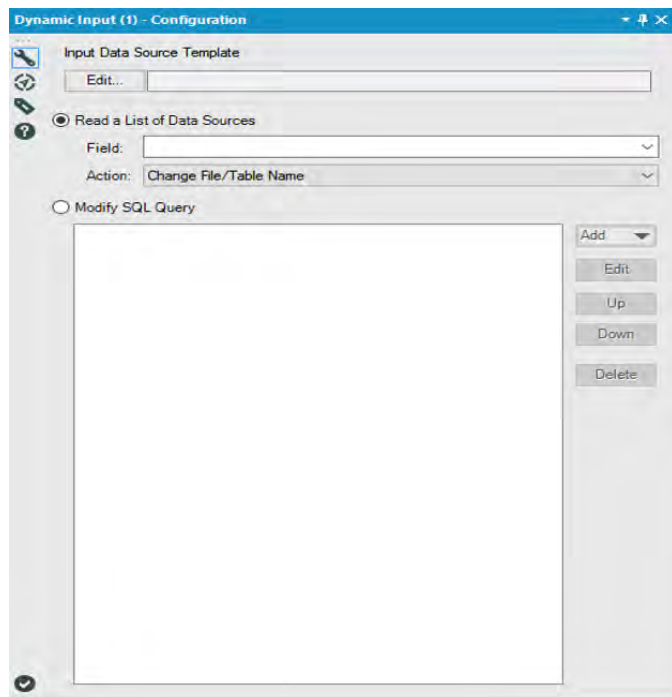
Figure 14-16 – Dynamic Input Configuration

Input Data Source specifies the input path to the database table for reading.

There are two choices on how to read and the action to execute with this input.

Read a List of files reads a list of files contained within the connection stream and performs the specified action, updating the input path with incoming field data.

Modify SQL Where Clause allows the user to specify which records and values get returned from the database.

14.9   Dynamic Replace

<table>
<tr>
<td rowspan="3">

Figure 14-17 – Dynamic Replace</td>
<td colspan="3">The *Dynamic Replace* tool allows us to replace data values based on a condition.</td>
</tr>
<tr>
<td>**Group**</td>
<td>**Input**</td>
<td>**Output**</td>
</tr>
<tr>
<td>Developer</td>
<td>See below</td>
<td>See below</td>
</tr>
</table>

Note: The *Dynamic Replace* tool stops at the first match. Test for both sides of the range is not required.

Input D: This is the original data needing replacement and requires no configuration

Input R: This is the replacement data. The data stream must contain *Field Name*, *Output Value,* and an *Expression* field.

The tool matches the Field Name of *Input R* to the field names in *Input D*, and replaces the value of the D connection with the *Output Value* of the R connection, using the *Expression* field.

Output O: This is the original data structure with the replacement data.

Output S: This is a summary of the replacements. It provides the count of each *Field Name* and *Output Value* combinations that were replaced.

Properties Window:

The *Dynamic Replace Configuration* window has five components.
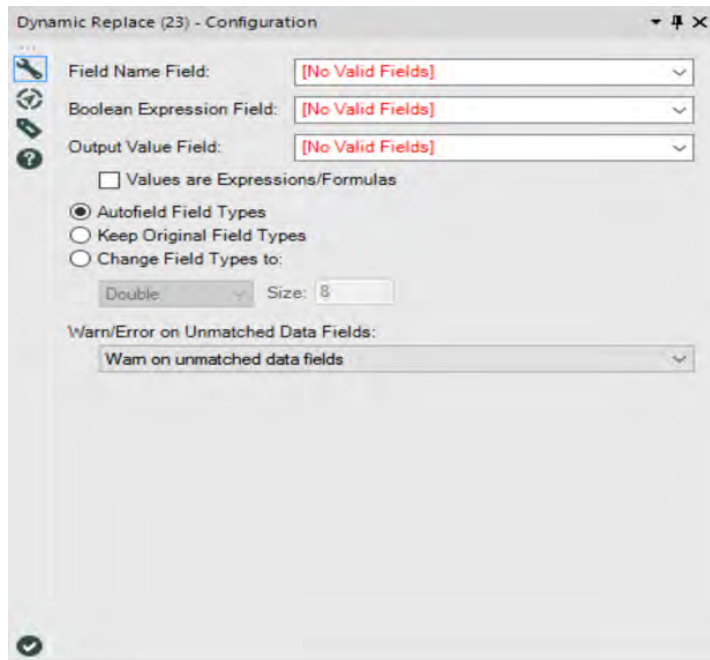
Figure 14-18 – Dynamic Replace Configuration

Field Name Field is the field where data will be replaced.

Boolean Expression Field is a condition where any value other than zero gets through the tool. The field contains the expression to determine the replacement.

Output Value Field is the data replacement. It can contain static values or a formula.

- *Values are Expressions/Formulas:* Check this box if the values in the field are formulas.

Field Type Handling: There are three options for changing the field type.

- *Autofield Field Types* sets the field type to accommodate the data contained within the column.
- *Keep Original Field Types* maintains the original field type.
- *Change Field Types to* allows the user to specify the field type and size.

Warn/Error on Unmatched Data Fields: The drop-down at the bottom allows us to change the message behavior if there are unmatched data fields in the incoming data stream.

14.10   Dynamic Select

Figure 14-19 – Dynamic Select

	The *Dynamic Select* tool allows us to select or de-select fields by a field type or an expression.	
Group	**Input**	**Output**
Developer	Any data stream	Any data stream

Note: The *Dynamic Select* tool allows fields to be selected (or de-selected) either by field type or via a formula. At runtime, dynamic or unknown fields will also be selected by field type or via formula.

Most tools cannot handle a stream with no fields and will cause an error.

Properties Window:

The *Dynamic Select Configuration* window has two modes of field selection.
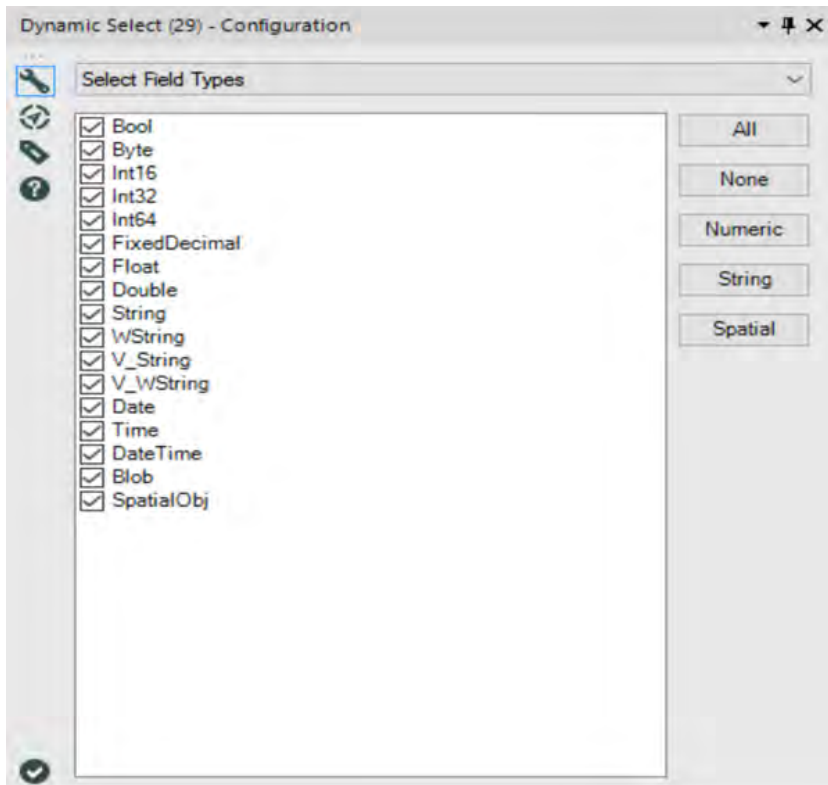
Figure 14-20 – Dynamic Select Configuration Field Types

Select Field Types selects fields based on their field type. All field types are listed. Check the box to the left of the field type to add to the selection.
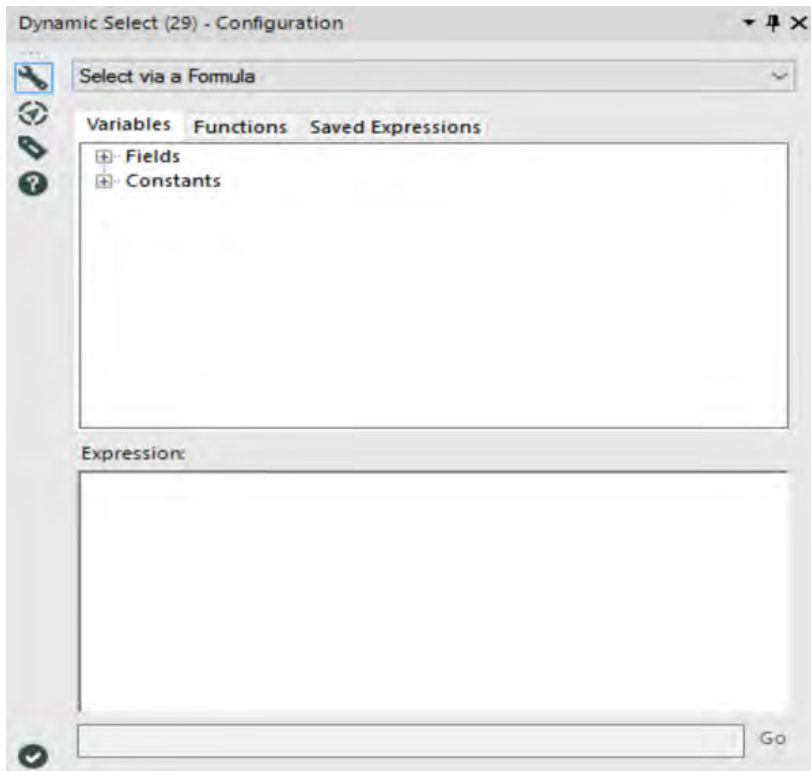
Figure 14-21 – Dynamic Select Configuration Formula

Select via a Formula selects fields based on criteria specified via the formula processor. The field information data fields available for building the expression are listed in Fields.

14.11   Field Info

Figure 14-22 – Field Info

The *Field Info* tool outputs the description of the data.

Group	Input	Output
Developer	Any data stream	Any data stream

Note: The *Field Info* tool outputs records even when there are no records coming into it.

The *Field Info* tool displays field names, field order, field type and field size in a tabular format. The output is similar to Metadata.

This tool can be used for validation of schema before scheduling processing. A typical scenario is to combine with the *Test* tool and at times *Block Until Done* tool.

Properties Window:

The *Field Info Configuration* window has no configuration. A *Browse* tool or *Output Data* tool needs to be linked in order to utilize the functionality of the tool.
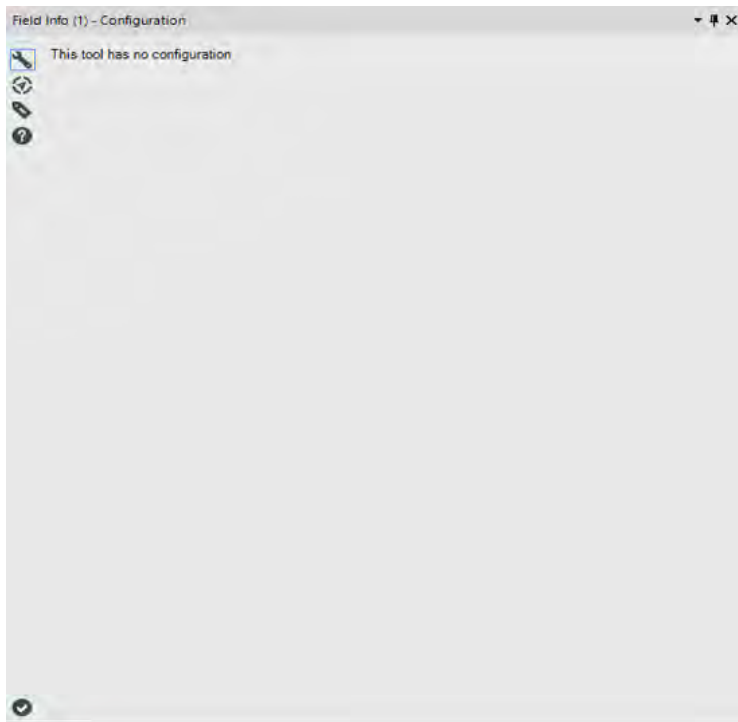
Figure 14-23 – Field Info Configuration

14.12 Run Command

	Run command tool is used to run external programs as part of an Alteryx process.		
	Group	**Input**	**Output**
Figure 14-24-Run Command	Developer	None	None

Example: Call other programs from inside Alteryx- example, curl.exe was widely used to scape data, before Alteryx had the download tool.

Note: The Run Command tool cannot be used in apps saved to the Alteryx Analytics Gallery.

If the program you are using doesn't set error flags, you might want to wrap it in a batch file to make sure the output file is deleted before it is run.

If you are using the Run Command tool as an input, the Write Source should be blank. If you are using the Run Command tool as an output, the Read Results should be blank.

The Run Command tool allows the user to run external command programs within Alteryx. This tool can be used as an Input, Output or as a pass through, intermediary tool.

The concept of the Run Command tool is that it is a process identical to running applications directly from the Windows Command line with the convenience of remaining within the Alteryx GUI.

This tool is useful when functionality is needed that Alteryx doesn't have. It can be used in conjunction with curl.exe to download data from web or FTP sites. It can be used to call to 3rd party applications or for wrapping custom functionality written in other programming languages.

Configuration Properties
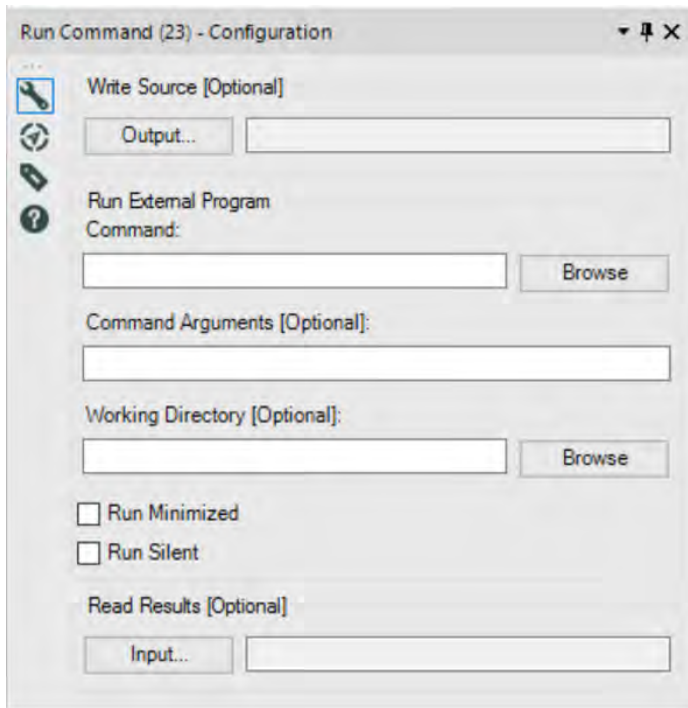
Figure 14-25-Run Command Configuration

The settings specified in the Configuration Properties happen sequentially when the workflow is run: Write Source (Alteryx), Run Program (Command line > External Program), Read Results (Alteryx)

Specify the Write Source (optional): The Write Source is the data that comes into the Run Command tool as a file, SQL server any supported output type. The write source will only be specified if you are using the Run Command tool as an output or intermediary tool.

Specify the external command program to run. A typical example is where the command program will act on write source and generate read results. See Using the Run Command Tool below.

If the external program is a UNIX style command line app that takes input from stdin and writes to stdout, you can use the << and >> operators to pipe the information to and from files.

Type the Command Arguments, if any. Most commands are used together with one or more arguments. For some commands these arguments are optional; other commands require you to use them with an argument. See Using the Run Command Tool below.

Specify the Working Directory. If this is left blank, the default will be the same directory where the workflow is saved.

Run Minimized and Run Silent checkboxes: refers to the Run Command Window that pops up when an external command is executed.

Run Minimized: when Alteryx executes the command, the Run Command window will not pop up. it will be minimized on the task bar.
Run Silent: when Alteryx executes the command, the Run Command window will not display to the user.

Specify the Read Results. This specified file will be carried through the rest of the workflow. The Read Results only needs to be

specified if you are using the Run Command tool as an input or intermediary tool.

Note: Click Apply to have the configurations accepted.

See the Example below:

- The Run Command tool in this example is being used as an Input file in the Alteryx Workflow. Therefore, the first section, Write Source, is left blank
- Run External Program:
  The command is curl.exe this is a free and open source program that allows users to download files from a web page. Documentation for cURL can be downloaded from http://curl.haxx.se/docs/
- Command Arguments when utilizing cURL should include:
  The location of the file to be downloaded from its URL.

In this case, it is:

http://www.wunderground.com/weatherstation/WXDailyHistory.asp?ID=KCOBOULD16&month=6&day=19&year=2005&format=1

After the complete path, the command argument should include an output command and the file you would like to output. In this case: -o temp.CSV What this part means: output (-o) the content from the web page as a file called temp.csv

- The working directory defaults to the directory where the workflow is saved - left blank in this case.
- Read Results refers to the temp.CSV file we are returning from the downloaded content, with its relative path (%temp% references the user's temporary directory). The cURL program has already downloaded the content and is handling the data as this CSV file. Alteryx is reading the CSV file as if it came from an Input Data Tool. The file can now be processed downstream.

14.13 Test

Figure 14-26 – Test	The *Test* tool checks conditions to verify data.		
	Group	**Input**	**Output**
	Developer	Any data stream	No output connection

Note: The *Test* tool checks condition in modules and throws an error when the user-specified condition is not met.

The *Test* tool is useful for testing assumptions about the data. It is a good practice to verify data before an update is committed to the database.

This tool is extensively used for regression testing.

Input: Can accept multiple inputs.

Output: No output connection. The test will pass if the condition is met or result in a descriptive error message in case of failure.

Properties Window:

The *Test Configuration* window has three actions: Add, Edit and Delete.
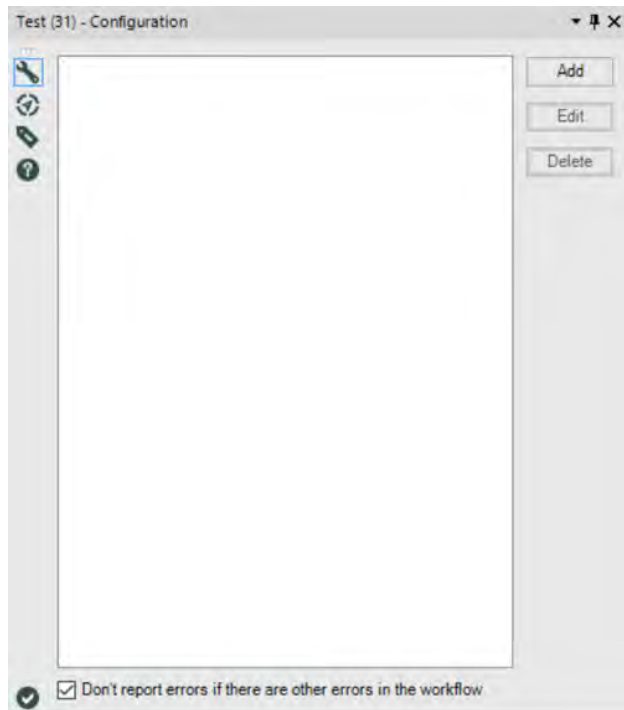
Figure 14-27 – Test Configuration

Each test requires four user specifications:

Test Name is the name of the test to conduct. This name appears in the properties window.
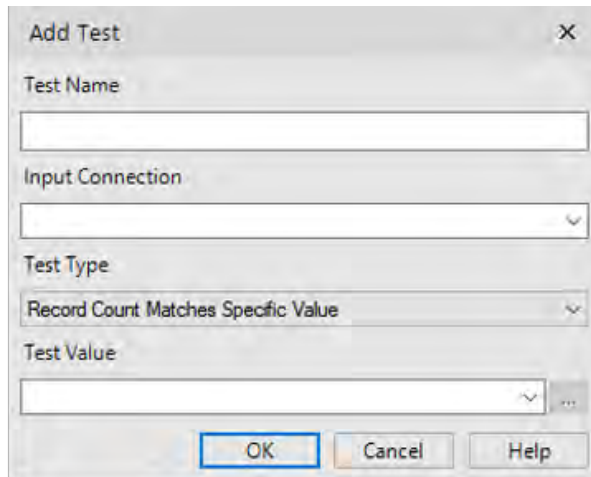
Figure 14-28 – Test Configuration Test Name

Input Connection specifies the input connection to test.

Test Type specifies the type of test to be conducted. There are five choices available:
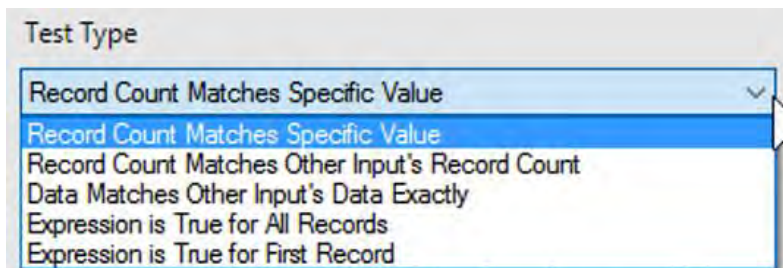
Figure 14-29 – Test Configuration Test Type

1. *Record Count Matches Specific Value* verifies that a number of records coming into the tool match with the value specified.
2. *Record Count Matches Other Input's Record Count* compares the record count of one specified connection to another connection.
3. Data Matches Other Input's Data Exactly compares inputs to verify that the data, data type, and data order in each input are identical.
4. *Expression is True for All Records* checks that all records through the specified connection must meet the specified criteria.
5. *Expression is True for First Record* checks that the first record coming through the specified connection meets the specified criteria.

Test Value is the assumption we are testing for.

14.14   Throttle

Figure 14-30 – Throttle	The *Throttle* tool slows down the runtime of the workflow.		
	Group	**Input**	**Output**
	Developer	Any data stream	Any data stream

Note: The *Throttle* tool slows down the speed of downstream processing by limiting the number of records that are passed through it.

The *Throttle* tool is useful when there are limits to how many records can be processed or when sending or retrieving data from the web where requests have a rate limit.

This is useful for slowing down requests sent per minute when there are limits to how many records can be sent in.

<u>Properties Window:</u>

The *Throttle Configuration* window has one property.
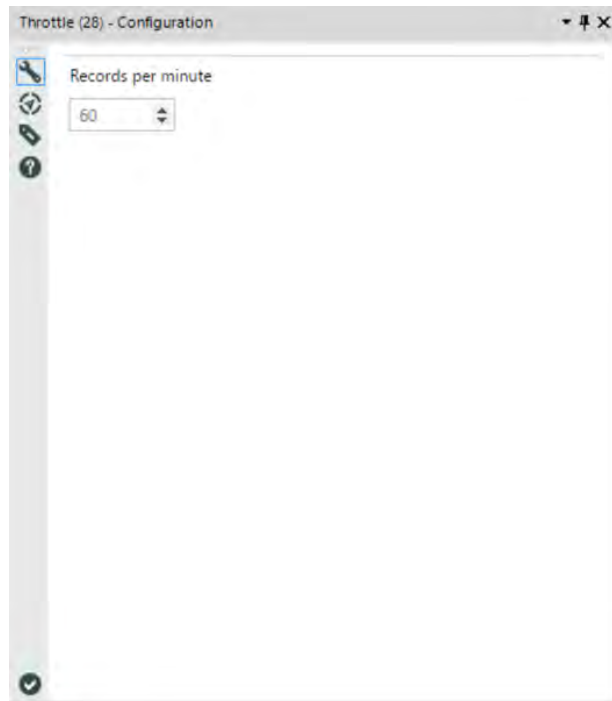
Figure 14-31 – Throttle Configuration

Records per minute specifies the number of records to send through each minute.

With the default of 60 records per minute, one record is sent through each second. The tools sleeps at 1ms intervals, so it cannot process more than 1000 records per second or 60,000 records per minute.

14.15   JSON Build

	The *JSON Build* tool allows us to read table schema and build it into Java Script Object Notation.
Figure 14-32 – JSON Build	**Group** **Input** **Output**

	Group	**Input**	**Output**
Figure 14-32 – JSON Build	Laboratory	Table schema of *JSON Parse*	JSON formatted data stream

Note: For those unfamiliar with JSON: JSON stands for JavaScript Object Notation, which is a structured data format typically used in data transfer.

The *JSON Build* tool takes the table schema of the *JSON Parse* tool and builds it back into properly formatted JSON.

Properties Window:

The *JSON Build Configuration* window has six components.
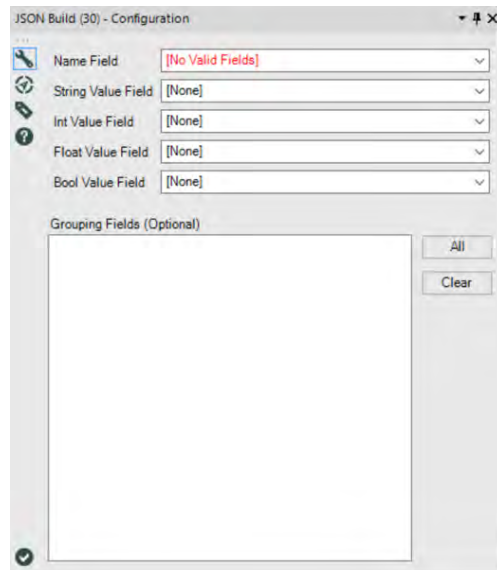
Figure 14-33 – JSON Build Configuration

- *Name Field* is the key of a JSON object. This is a required field.
- *String Value Field* is the corresponding string value of the JSON Object.
- *Int Value Field* is the corresponding integer value of the JSON Object.
- *Float Value Field* is the corresponding float value of the JSON Object.
- *Bool Value Field* is the corresponding Boolean value of the JSON Object.
- *Grouping Fields* allows us to build a JSON object for each group or groups specified. Specifying a grouping field is optional.

14.16   Transpose In-DB

	The *Transpose In-DB* tool allows us to de-normalize data in a database workflow.		

Figure 14-34 – Transpose In-DB | **Group** | **Input** | **Output** |
| | Laboratory | See below | See below |

Note: This tool converts all spaces and special characters in the titles into underscores in each record.

Input: Any data stream in a database with multiple fields that need to be combined into rows.

Output: A taller data stream because records have been duplicated in order to consolidate the columns. The columns consolidate into two columns; *Name*, which is a column with each of the former column names in it, and *Value*, which is a column with each of the data values.

Properties Window:

The *Transpose In-DB Configuration* window has three elements, as shown in the figure below.
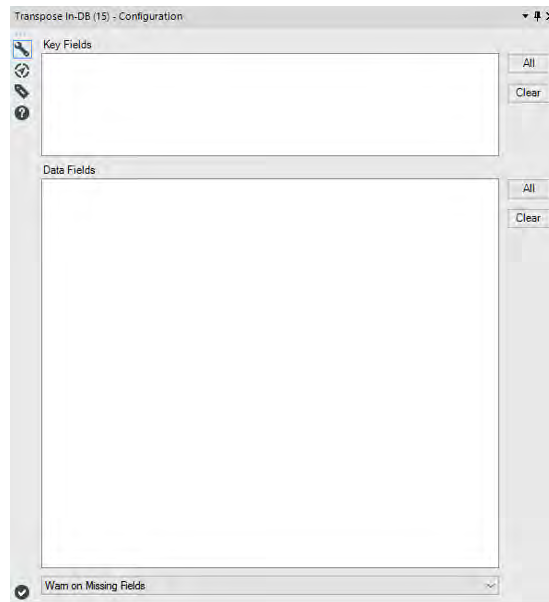
Figure 14-35 – Transpose in-DB Configuration

Key Fields allows us to select the fields that will be maintained after the transposition.

Data Fields allows us to select the fields that will be combined following the transposition.

The drop-down at the bottom allows us to change the message behavior if there are missing fields in the incoming data stream.

14.17   Energy Management Company

Let's start building a workflow (*What's the Temperature Like*) that will answer our question. We are going to start with a blank canvas and save it as Weather Info. Next, bring the Text Input tool; type in the URL to pull data.

Figure 14-36 – Energy Firm Input Section

Use the formula tool, we will now add this text input to a new column with correct field type. Best practice is to add Browse tool after each input. The Browse helps us check data and ensure that the data we are getting is correct.

Figure 14-37– Energy Firm Browse Tool for Input

Now that we have the URL in the correct format, the next step is to import data from the URL. To achieve this, we should use "download" tool.

URL =
"http://www.wun
derground.com/hi
story/airport/KM
DW/2017/01/01/C
ustomHistory..
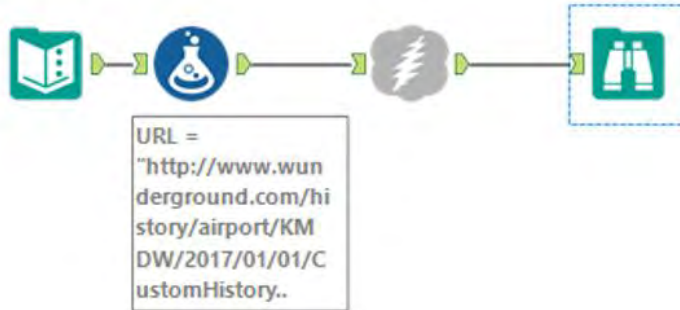
Figure 14-38– Energy Firm Download Tool

Best practice is to remove data as soon as it is no longer needed. It makes sense that the first step in filtering would be to bring in the Filter tool; however, if we are not familiar with the data set and we have not run it, we may not have enough information to filter properly. In this case, we want to run the module so that there's data in the Browse tool for us to work with.

We can see that when the module finishes running, we get a pop-up window that lets us know how long it took to run and if there were errors.
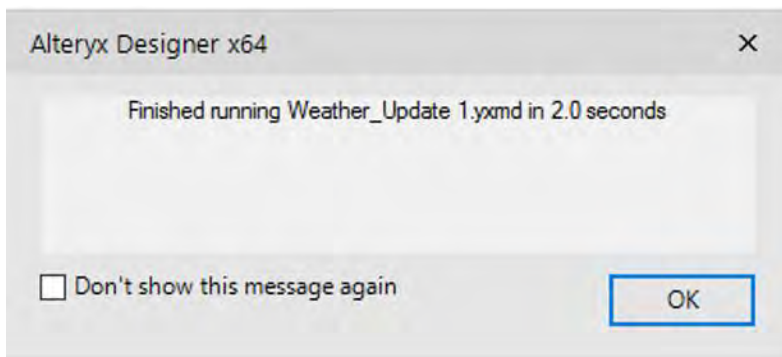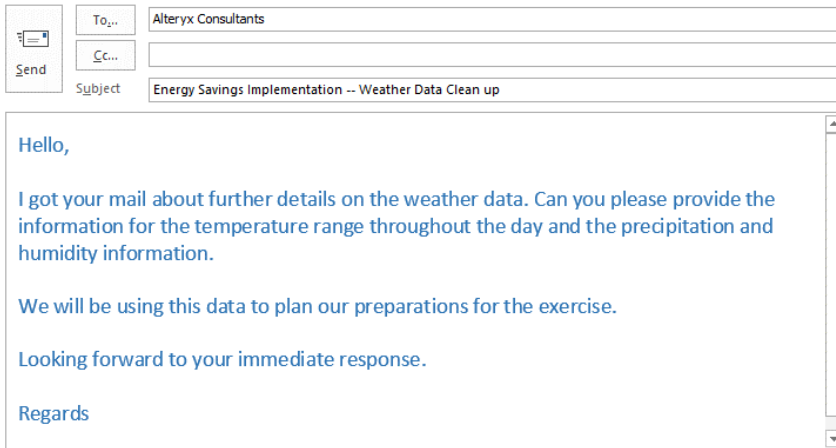
Figure 14-39– Energy Firm Execution Time

Feel free to click on the 'Don't show this message again' check box before closing if the pop-up window is distracting.

What's the Weather Like?

We have weather info data, but unfortunately, it's not in usable form. We are going to make the data cleaner and employ best practices. So, the process that we are going to follow is:

1. Text input URL.
2. Input URL with right field type via formula tool.
3. Browse the data.
4. Make sure the data has the right type.
5. Import data from URL via Download.
6. Browse the data.
7. Make sure data is imported successfully.
8. Remove redundant columns via the select tool.
9. Browse the data.
10. Export the dataset.

We have already completed few of the steps listed above. We would be continuing from step 6. The next step is to make sure that the data is imported successfully.

First, let's run the Module to see how the data is structured.
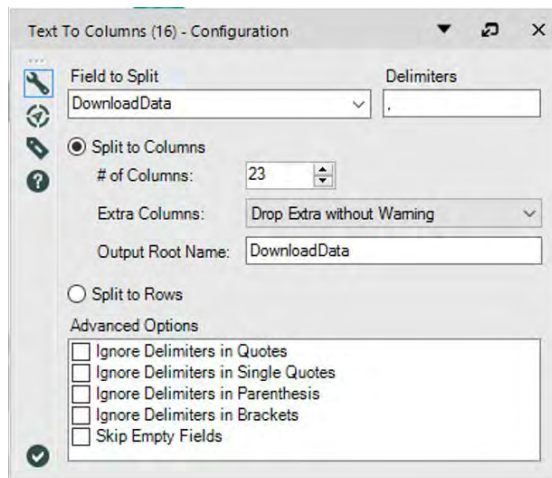
Figure 14-40- What's the Temperature Like Download Browse Output

We observe that the fields are in a single cell and need to be moved to a text to columns tool.

The Text to columns tool takes text data and splits it into columns. If we take the data stream coming out of *text to columns* and pass it into Data Cleanse tool, we can make the data tidier.
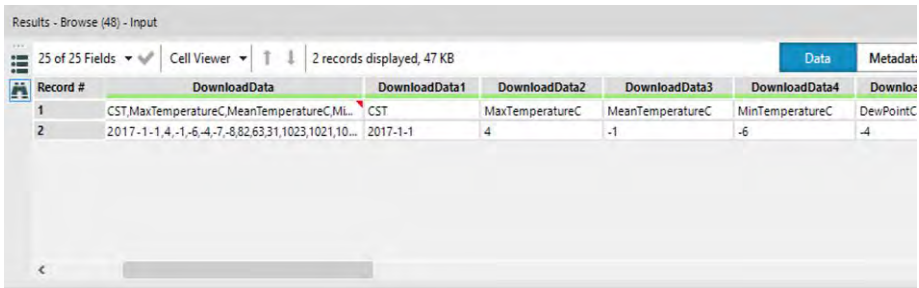
Figure 14-41– What's the Temperature Like Text To Column Configuration

We want to split the downloaded data into as many fields as required. So, we split the downloaded data column into child columns.

Best practice is to always include a Browse after a tool that modifies the structure of a data stream. We will now include a *Data Cleansing* Tool after the Text to Columns to cleanse the data generated from the Text to Columns output. Let's add a Browse to the end of the data stream and run it to see what we have.

Figure 14-42– What's the Temperature Like Browse Output after Data Cleansing

If we compare the record from the new Browse to the one that came out of the download tool in figure 2-66, we see that we have split the data into multiple child columns. We also notice 2 columns URL and Download Headers in the original dataset are missing. This is because we unselected them in the Select tool.

The last step involves two tools: The *Browse* tool and the *Output Data* tool. Best practice dictates that we put a Browse before every data output so that we do not need to open the file to make sure we created it correctly.

We now add a Browse tool to the end of the data stream and also add an Output Data tool. We are going to write the file to the same folder we have saved the *What's the Weather Like yxmd*.
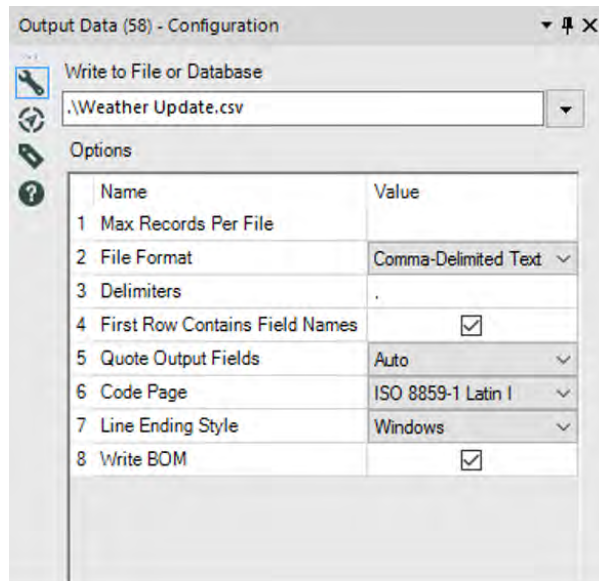
Figure 14-43– What's the Temperature Like Output Configuration

To do this, we are going to type *.\Weather Update.csv* in the text box labeled *Write to File or Database*.

We just used a relative file path. Which allows us to reference files in relation to where we currently are. Some basics of relative paths are ".\", which means the current folder. "..\" means the parent folder (the folder that our current folder is in). ".\Folder Name\" will move our file into a folder below where we have the workflow.

We do not necessarily need to use relative paths, but if we are sharing Alteryx files, it is very beneficial to do so. We can use absolute paths (full file locations) by pasting them into this box or navigating through them in the File Browse option.

If we run the module, we see what the final dataset looks like. This ensures the information written into the .csv was correct.

Here is how the *What's the Weather Like* data stream looks on completion.

URL =
"http://www.wun
derground.com/hi
story/airport/KM
DW/2017/01/01/C
ustomHistory...

Weather
Update.csv

Figure 14-44– What's the Temperature Like Data Stream
when complete

How do I read this?

To..,   Alteryx Consultants

Cc...

Send   Subject   Energy Savings Implementation -- Weather Data Final Output

Hello,

Thanks for the csv file with the relevant data.

We would like you to include the following points into the data output and in the workflow logic:

1) Column Names should be relevant names
2) Can you make the workflow to be able to check the data , if it is exporting the data for only that date and provide a message that says Success at the end of the test.

Thanks for your efforts, we plan to use your workflow for future exercises, so if you can also include some documentation that would be great.

Regards

When we were working on the weather data notice that the column names are out of context.

This is a much more complicated process than the last exercise, but that is only because the output was a simple .csv file the last time, and it isn't here.

We are going to be connecting to the same data source that we have been using, but we are going to use a shortcut in this process. Open a new workflow and save it as *Success Message workflow. yxmd*, but make sure that *What's the Weather Like yxmd* is still open.

Drag/Click on the workflow in *What's the Weather Like. yxmd* and copy it. Move over to the Success History Workflow canvas, and paste what we have copied. We see that the workflow has been copied over and we do not need to recreate it.
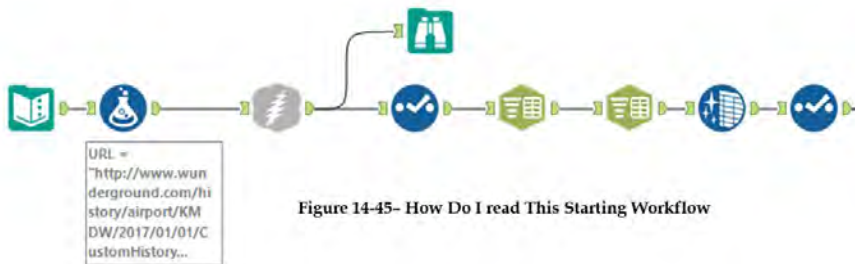
URL =
"http://www.wun
derground.com/hi
story/airport/KM
DW/2017/01/01/C
ustomHistory...

Figure 14-45– How Do I read This Starting Workflow

Best practice will once again bring in Browse and Select tools. But since we know from past experience what the data looks like and how it is read in, we will move directly to the next step.

We know that this data doesn't have the correct column headers. So, we have to add it. Based on the email, we know the only information we will need is already present in the downloaded data as 1st row.
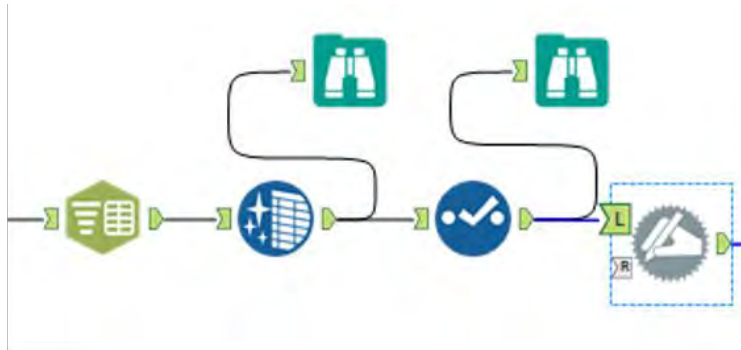
Figure 14-46– How Do I read This Dynamic Rename Input

We now add a Dynamic Rename tool and select "Take field names from the first row of data" option.

Let's take a look at what we have created so we can get a better sense of what the process so far has done.

Figure 14-47- How Do I read This Dynamic Rename Output

If we add a Browse tool and run the workflow, we can see that we have the entire weather info along with relevant column headers.

The next part is where we have to implement a success criteria check on the dataset as requested by the client. To do that we will use the functionality of some Developer Tools like *Block Until Done, Test* and *Run*.
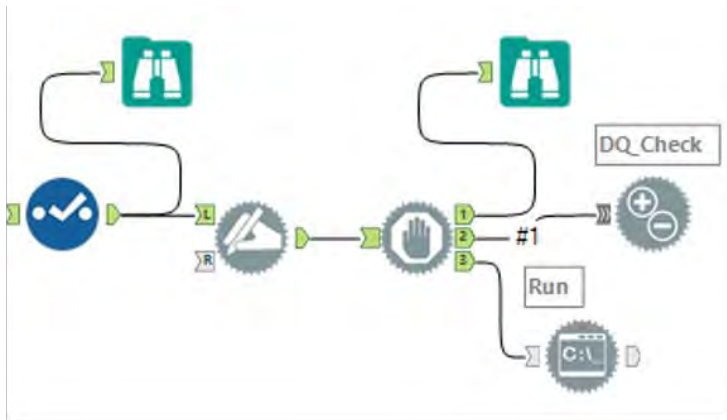
Figure 14-48– How Do I read This Success Check

What we have done here is created three outputs for the Block Until Done, so that the first output is a regular .csv output and then we do a Test. The final output is used for the Success Message.
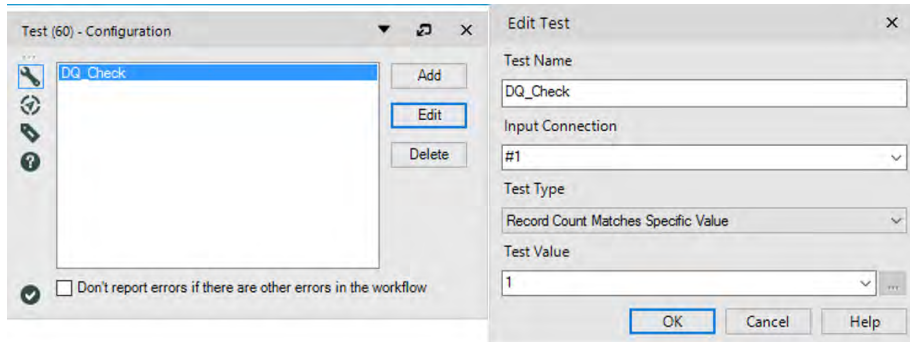
Figure 14-49– How Do I read This Test Configuration

The test configuration checks for the number of rows in the data output. Please note the various options that we get as part of the Test. Here we have made use of the type where we check if the Record Count matches a specific value, in this case, 1.

There may or may not be any historic weather info for a particular weather station on a specific day. In this case, how does the team know the workflow is executed successfully and info available for use? It was assumed we would always have would have the correct URL and historic data in the training so far. There's a way out to determine if workflow was executed successfully outside of Alteryx by using Run tool.

Figure 14-50– How Do I read This Data Run Configuration

Figure 14-51– How Do I read This Data Stream on Completion

14.18   How's the oil market doing?

	To...	Alteryx Consultants
Send	Cc...	
	Subject	How's the Oil Market Doing

Hey,

Thanks for your analysis on the Weather Data .

Here is some data from our sources which we would use for Oil prices information . However we are not able to understand anything as this is JSON format.

Can you please put it in a reasonable format for us to use it for our analysis ?

Thanks

CHAPTER 15

Predictive Analytics with Alteryx

	From ▼	Google Apps
Send	To...	Alteryx Consultants
	Cc...	
	Subject	Forecasting Business Value

Hey,

We are ABC LLC (a **boutique investment bank**) and have stock information for a merger deal and want to do some predictive analysis on the dataset. We think the best way is to forecast (within a small error margin) the price of the stocks , based on its previous values. The dataset for the purpose of the exercise has been shared.

Based on the domain understanding , we think that we could use the data for time series modelling to do some forecasting .

Can one of you help us to look at time series forecasting tools in Alteryx and then compare the output of the different models with the performance metrics and the best one can be finally used to forecast for the future?

Let's get the ball rolling on the Forecasting bit and we can then do the model comparison post that.

15.1   Tools & Concepts

Tools	Groups
Find Nearest Neighbours	Predictive Grouping
K-Centroids Cluster Analysis	Predictive Grouping
K-Centroids Cluster Diagnostics	Predictive Grouping
MB Affinity	Predictive Grouping
MB Inspect	Predictive Grouping
MB Rules	Predictive Grouping
Principal Components	Predictive Grouping
Boosted Model	Predictive
Count Regression	Predictive
Decision Tree	Predictive
Forest Model	Predictive
Gamma Regression	Predictive
Lift Chart	Predictive
Linear Regression	Predictive
Logistic Regression	Predictive
Naïve Bayes Classifier	Predictive
Nested Test	Predictive
Network Analysis	Predictive
Neural Network	Predictive
Score	Predictive
Spline Model	Predictive
Stepwise	Predictive
Support Vector Machine	Predictive
Test of Means	Predictive
AB Analysis	AB Testing

AB Controls	AB Testing
AB Treatments	AB Testing
AB Trend	AB Testing
ARIMA	Time Series
ETS	Time Series
TS Compare	Time Series
TS Covariate Forecast	Time Series
TS Filler	Time Series
TS Forecast	Time Series
TS Plot	Time Series

15.2   Append Cluster

Figure 15.1 Append Cluster	The Append Cluster tool appends the cluster assignments from a K-Centroids Cluster Analysis tool to a knowledge stream. The info stream doesn't need to be constant one that the cluster answer relies on.

Group	Input	Output
Predictive Grouping	See below	See below

Note: The data stream does not have to the same one the cluster output is based. However, it must contain the fields that were used to create the original cluster.

Input: Data stream with the appropriate fields.

Output: A data stream with cluster assignment variable affixed.

The field name for the cluster assignments: The clusters names are consecutive integers beginning with one. this feature determines the name of the cluster assignment field, with the default field name being "Cluster." The cluster assignment field name should begin with a letter and solely embrace letters, numbers, and "_" or "." characters.

15.3   Find Nearest Neighbors

	The find Nearest Neighbors tool finds the chosen range of nearest neighbors within the "data" stream that corresponds to every record within the "query" stream supported their Euclidean distance.		
Figure 15.2 Find Nearest Neighbors	**Group**	**Input**	**Output**
	Predictive Grouping	See below	See below

Note: The tool provides the user an alternative of algorithms for locating the closest neighbors that take issue with their speed and potential accuracy. The default is to try to the search supported the KD-Tree algorithmic program that contains a typically smart combination of speed and accuracy.

Input: Two Alteryx data streams. The right stream is the "query" stream, the rows for which the selected number of nearest neighbors in the left stream (the "data" stream)

Output: Comprises of a table that gives the remarkable key value and separation to the desired number of close neighbors to each point in the query stream.

15.4   K-Centroids Cluster Analysis

Figure 15.3 K-Centroids Cluster Analysis	K-Centroids speak to a class of calculations for doing what is known as partitioning cluster examination. These techniques work by taking the records in a database and isolating (dividing) them into the "best" K bunches in view of a few criteria.		
	Group	**Input**	**Output**
	Predictive Grouping	See below	See below

Note: Three specific types of K-Centroids cluster analysis can be carried out with this tool: K-Means, K-Medians, and Neural Gas clustering. K-Means uses the mean value of the fields for the points in a cluster to define a centroid, and Euclidean distances are used to measure a point's proximity to a centroid.

Input: Data stream with the appropriate fields.

Output: There are 2 output streams:

O Output: consists of a table of the serialized model with model name and the size of the object.

R Output: consists of the report snippets generated by the K-Centroids Cluster Analysis Tool: a statistical summary and cluster solution plots.

15.5   K-Centroids Cluster Diagnostics

 Figure 15.4 K-Centroids Cluster Diagnostics	The K-Centroids Diagnostic tool is intended to permit the client to make an appraisal of the fitting number of clusters to determine given the information and the chose grouping calculation (K-Means, K-Medians, or Neural Gas).

Group	Input	Output
Predictive Grouping	See below	See below

The tool is graphical and depends on figuring two unique measurements over bootstrap that recreates tests of the original information for a scope of grouping arrangement that contrast in the quantity of clusters indicated. The inspiration driving this approach is that if the records in a database genuinely fall into an arrangement of stable groups, then the reality of the situation should prove that an arrangement of various irregular specimens of those records ought to bring about roughly the arrangement of bunches over the bootstrap repeats, aside from little contrasts that are because of both arbitrary example changeability and to the haphazardness prompted by the strategy used to create the beginning arrangement of centroids, through choosing K focuses indiscriminately, in the general K-Centroids calculation.

Input: Data stream with the appropriate fields.

Output: The yield of the tool is data about the dispersion of the two measurements for varying quantities of groups over the bootstrap repeats. The data is passed on by means of two box and bristle plots (one each for the balanced Rand list and the Calinski-Harabasz list) and outline insights for the two measures.

15.6   MB Affinity

	The MB Affinity tool takes "transaction" information and builds a framework where each line is a transaction and the segments are the arrangement of "items" that could show up in the transaction.		
Figure 15.5 MB Affinity	**Group**	**Input**	**Output**
	Predictive Grouping	See below	See below

On the off chance that an item was available in a transaction, it has an estimation of 1 in the lattice, and 0 in the event that it didn't. The lattice is then used to develop a network of proclivity measures between various items regarding their probability of being in a similar transaction. Subsequently, if there are 500 items, the subsequent framework will have 500 lines and sections. The accessible affinity measures are cosine similarity, cosine distance, and co-occurrence (e.g., the quantity of times two items showed up in a similar transaction).

This tool scales with the quantity of transaction records, yet is constrained in the quantity of remarkable items it can deal with. Under regular tablet and desktop setups, the cutoff is probably going to be in the vicinity of 3000 and 5000 unique items.

Input: Data stream with the appropriate fields.

Output: A data matrix based on the input fields.

15.7   MB Inspect

Figure 15.7 MB Inspect

The MB Inspect tool takes as info a standards or item sets yield of the MB Rules tool and gives a posting and investigation of those principles that can be sifted on a few criteria so as to lessen the number or returned tenets or item sets to a sensible number.

Group	Input	Output
Predictive Grouping	See below	See below

Input: MB rules output 'O' data.

Output: A data stream with analysis of the input data.

The Output ("O") connection returns a data stream that contains the association rules or item sets.

The Report ("R") connection returns a set of report snippets, that include a summary of the filters rules to develop the inspection report, the rules or item sets generated (the same

information as returned in via the Output connection), and one (for item sets) or two (for association rules) diagnostic plots to aid the user in understanding the relationships between different items and rules.

15.8   MB Rules

	The MB Rules tool takes transaction data and, after transforming the data, creates either a set of association rules or frequent item sets.

Figure 15.7 MB Rules

Group	Input	Output
Predictive Grouping	See below	See below

Input: Data stream with the appropriate configuration.

Output: A summary report of both the transaction data and the rules/item sets is produced, along with a model object that can be further investigated in a downstream process.

15.9   Principal Components

Figure 15.8 Principal Components

The Primary Parts tool can lessen the dimensions (the number of numeric fields) in a database. It does this by changing the first arrangement of fields into a littler set that records for the greater part of the variance (i.e., information) in the data. The new fields are called factors, or principal components.

Group	Input	Output
Predictive Grouping	See below	See below

The main parts are extricated successively, with the primary vital segment representing the most difference in the information. Naturally, the main essential part is a vector that focuses toward the path in which the information is most "spread out." The second vital segment is set up comparably, however with the extra imperative that it must be uncorrelated with the first. Each ensuing vital part catches an inexorably bring down the rate of variety in the information, and is uncorrelated with the already removed chief segments. There can be the same number of primary segments as there are numeric fields in the information. Be that as it may, it is normally conceivable to catch the change in the information utilizing the initial few chief parts rather than the full arrangement of unique numeric fields. An essential part is comprised of a weighted straight blend of the first numerical fields. Together they can be utilized to frame another arrange

framework, where each measurement is uncorrelated to the others.

Input: Data stream with the appropriate fields.

Output: A data stream with reduced dimensions.

O: Consists of the input data stream with the Principal Components appended.

R: Consists of the report snippets generated by the Principal Component tool: a statistical summary, basic plots, and biplots

15.10   Boosted Model

Figure 15.9 Boosted Model

The Boosted Model tool gives summed up supported relapse models in light of the gradient boosting techniques for Friedman.

Group	Input	Output
Predictive	See below	See below

Note: This technique is a present day factual learning model that self-figures out which subset of fields best anticipate an objective field of premium, can catch profoundly nonlinear connections and collaborations amongst fields, and can naturally address a wide scope of relapse and arrangement issues in a way that can be straightforward to the client.

The tool is relevant to an extensive variety of issues, including characterization, tally information, and persistent target relapse issues and works by serially adding straightforward choice tree models to a model gathering to limit a proper misfortune work.

Input: Data stream with the appropriate fields. Configuration can vary based on the input type, database or Alteryx data stream.

Output: A data stream with, O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

15.11   Count Regression

Figure 15.10 Count Regression

The Count Regression tool makes a regression model that relates a non-negative whole number value (0, 1, 2, 3, and so forth.) field of interest (an objective variable) to at least one fields that are relied upon to have an impact on the objective variable, and are regularly called predictor variables.

Group	Input	Output
Predictive	See below	See below

Input: Data stream with the appropriate fields or an XDF metadata file, that includes a target field of interest along with one or more possible predictor fields.

Output: Output: Consists of a table of the serialized model with its model name.

R Output: Consists of the report snippets generated by the Count Regression tool: a statistical summary, a Type II Analysis of Deviance (ANOD), and Basic Diagnostic Plots. The Type II Analysis of Deviance table and the Basic Diagnostic Plots are not produced when the model input comes from an XDF Output or XDF Input tool.

15.12   Decision Tree

	The Decision Tree tool develops an arrangement of if-then split decides that enhance a criterion to make a model that predicts an objective variable utilizing at least one indicator factors.		
Figure 15.11 Decision Tree	**Group**	**Input**	**Output**
	Predictive	See below	See below

Note: The criteria used to shape these tenets relies on upon the way of the objective variable. On the off chance that the objective variable recognizes enrollment in one of an arrangement of classifications, then a grouping tree is developed. On the off chance that the objective variable is a ceaseless variable, then a relapse tree is built.

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields or an in-database information stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: Connect a Browse tool to each output anchor to view results.

O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

I (Interactive): Displays an interactive dashboard of supporting visuals that allow you to zoom, hover, and click.

15.13   Forest Model

	The Forest Model tool makes a model that builds an arrangement of choice tree models to foresee an objective variable in view of at least one indicator factors.		
	Group	**Input**	**Output**
Figure 15.12 Forest Model	Predictive	See below	See below

Note: The distinctive models are built utilizing irregular specimens of the first information, a methodology known as bootstrapping. Moreover, just a predetermined number of factors is considered at each tree split, with the number decided set either naturally by R or set by the client.

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields or an in-database information stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: Connect a Browse tool to each output anchor to view results.

O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

15.14 Gamma Regression

Figure 15.13 Gamma Regression

The Gamma Regression tool relates a gamma conveyed, the entirely positive variable of intrigue (target variable) to at least one factors (indicator factors) that are relied upon to have an impact on the objective variable.

Group	Input	Output
Predictive	See below	See below

Note: In various applications, the estimations of the objective variable are dependably entirely positive (i.e., are never zero or negative), yet tend to group toward the lower scope of the watched values, yet in a little minority of cases go up against expansive qualities. Target factors of this nature speak to an information era handle that is not steady with the Normality suspicions basic the customary straight relapse demonstrate.

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields or an in-database information stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: O Output: Consists of a table of the serialized model with its model name.

R Output: Consists of the report snippets generated by the Gamma Regression tool: a statistical summary, a Type II Analysis of Deviance (ANOD), and Basic Diagnostic Plots. The Type II Analysis of Deviance table and the Basic Diagnostic Plots are not produced when the model input comes from an XDF Output or XDF Input tool.

15.15   Lift Chart

Figure 15.14 Lift Chart

The Lift Chart tool delivers an aggregate captured reaction graph (also called a gains chart) and an incremental reaction rate diagram that are utilized to outwardly evaluate the relative precision of various parallel (yes/no) grouping models to foresee new information and to make an appraisal of the normal financial ramifications of the utilization of a prescient model in a business procedure.

Group	Input	Output
Predictive	See below	See below

Input: The tool just works with paired characterization models from the Neural Network, Logistic Regression, Decision Tree, Forest Model, Boosted Model, Spline Model, or Stepwise tools that have been joined together into a single Alteryx information stream, and either an Alteryx information stream or XDF metadata stream that is predictable with the model object (as far as field names and the field sorts), and all have the same binary target variable.

Output: A data stream with cluster assignment variable affixed.

15.16   Linear Regression

	The Linear Regression tool develops a linear function to make a model that predicts an objective variable in light of at least one indicator factors.		
Figure 15.15 Linear Regression	**Group**	**Input**	**Output**
	Predictive	See below	See below

Note: There are two principle sorts of linear regression: non-regularized and regularized. Non-regularized direct linear regression linear models that limit the aggregate of squared error between the real and anticipated estimations of the preparation information target variable. Regularized linear regression adjusts a similar minimization of the whole of squared errors with a penalty term on the measure of the coefficients and tends to create less complex models less inclined to overfitting.

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields or an in-database information stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: Standard processing: Connect a Browse tool to each output anchor to view results.

O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

I (Interactive): Displays an interactive dashboard of supporting visuals that allow you to zoom, hover, and click.

In-Database processing: Connect a Browse tool to each output anchor to view results.

O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

15.17   Logistic Regression

Figure 15.16 Logistic Regression

The Logistic Regression tool makes a model that relates an objective twofold factor, (for example, yes/no, pass/fail) to at least one indicator factors to get the assessed likelihood for each of two conceivable responses for the target variable.

Group	Input	Output
Predictive Grouping	See below	See below

Note: Common logistic regression models incorporate logit, probit, and complementary log-log.

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields or an in-database information stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: Standard processing: Connect a Browse tool to each output anchor to view results.

O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

I (Interactive): Displays an interactive dashboard of supporting visuals that allow you to zoom, hover, and click.

In-Database processing: Connect a Browse tool to each output anchor to view results.

O (Output): Displays the model name and size of the object in the Results window.

R (Report): Displays a summary report of the model that includes a summary and plots.

15.18 Naïve Bayes Classifier

Figure 15.17 Naïve Bayes Classifier	The Naive Bayes Classifier tool makes a binomial or multinomial probabilistic arrangement model of the connection between an arrangement of indicator factors and a categorical target variable.		
	Group	**Input**	**Output**
	Predictive	See below	See below

Note: The Naive Bayes classifier assumes that all indicator factors are free of each other and predicts, based on sample output, a probability distribution over an arrangement of classes, along these lines computing the likelihood of belonging to each class of the target variable. One of the main advantages of the Naive Bayes Classifier is that it performs well even with a small training set.

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: Object (O) output: Consists of a table of the serialized model with its model name.

Report (R) output: Consists of the report snippets generated by the Naive Bayes Classifier tool: a basic model summary, as well as main effect plots for each class of the target variable.

15.19   Nested Test

	The Nested Test tool inspects whether two models, one of which contains a subset of the factors contained in the other, are measurably proportional as far as their predictive capability.
Figure 15.18 Nested Test	

Group	Input	Output
Predictive	See below	See below

 The bigger of the two models is referred to as the "full" model, while the model that contains a subset of the factors is referred to as the "reduced" model. In the statistics writing, the "reduced" model is regularly called the "null" model, while the "full" model is known as the "alternative" model.

Input: The output stream from either a Linear Regression Tool or a GLM family tool (Count Regression Tool, Gamma Regression Tool, or Logistic Regression Tool) used to create the "full" model, the one with a greater number of variables. The stream can be entered as any of the three inputs.

The output stream from either a Linear Regression Tool or a GLM family tool (Count Regression Tool, Gamma Regression Tool, or Logistic Regression Tool) used to create the "reduced" model, the one with a subset of the variables included in the "full" model. The stream can be entered as any of the three inputs.

The Alteryx data stream that was used to create both the "full" and "reduced" models. The stream can be entered into either side of the tool.

Output: A data stream which shows whether the models are measurably proportional as far as their predictive capability.

15.20   Network Analysis

	The Network Analysis tool creates an intuitive dashboard of a system, to investigate connections between the different hubs. The tool gives a visual portrayal of the system alongside key outline insights that describe the system.

Figure 15.19 Network Analysis

Group	Input	Output
Predictive	See below	See below

Input: N Input: A Nodes data stream that contains a field named _name_ that uniquely identifies each node in the network.

E Input: An Edges data stream that contains fields named "from" and "to" identifying nodes that are connected by an edge. Note that the "from" and "to" fields must use the same unique node identifier.

Output: D Output: An Alteryx data stream with network centrality measures for each node.

I Output: An interactive dashboard of the network that consists of:
An interactive force diagram to visualize the network. Aggregate statistical measures that characterize the entire network. A histogram of node centrality measures.

15.21   Neural Network

Figure 15.20 Neural Network

The Neural Network tool makes a feedforward perceptron neural system model with a single hidden layer.

Group	Input	Output
Predictive	See below	See below

Note: While more present day factual learning strategies, (for example, models delivered by the Boosted, Forest, and Spline Model tools) ordinarily give greater predictive accuracy in respect to neural system models, in some particular applications (which can't be resolved before the fact), neural system models outflank different techniques for both arrangement and relapse models. Besides, in a few ranges, for example, in financial appraisal, neural system models are viewed as a "standard" strategy that is broadly acknowledged.

Input: Data stream with the appropriate fields.

Output: Object (O) output: Consists of a table of the serialized model with its model name.

Report (R) output: Consists of the report snippets generated by the Neural Network tool: a basic model summary, as well as main effect plots for each class of the target variable.

15.22 Score

Figure 15.21 Score
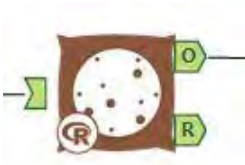
The Score tool takes the predicted values (scores) from an R-based predictive model, and either an Alteryx information stream or XDF metadata stream that contains all the predictor fields used to make the model, and yields the information stream with a one (for a model with a persistent target) or at least two (for a model with a downright target) "Score" (fitted value) field(s) added to the information stream.

Group	Input	Output
Predictive	See below	See below

Input: Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside all conceivable indicator fields or an in-database information stream that incorporates an objective field of interest alongside all conceivable indicator fields.

Output: The output incorporates the original information streams with the predicted estimations of the model given the information. On account of a model that uses an all-out focus on, a predicted likelihood for each level of the target variable is given, with the field name introduced by the user provided prefix and the addition of the field relating to the conceivable level of the target variable.

15.23   Spline Model

Figure 15.22 Spline Model

The Spline Model tool gives the multivariate adaptive regression splines (or MARS) calculation of Friedman, this technique is a cutting edge factual learning model that: (1) Finds out which subset of fields best predict an objective field of interest; (2) can catch highly nonlinear connections and interactions amongst fields and naturally address a wide scope of regression and grouping issues in a way that can be straightforward to the user.

Group	Input	Output

	Predictive	See below	See below
Note: The user can do as little as determining an objective field and an arrangement of predictor fields, yet the tool can be broadly tweaked by advanced clients.			
Input: The Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.			
Output: O Output: Consists of a table of the serialized model with its model name.			
R Output: Consists of the report snippets generated by the Spline Model tool: a basic model summary, a Variable Importance Plot (which indicates the relative importance of the different predictor fields), a Basic Model Diagnostics Plot, and (optionally) the Effects Plots.			

15.24   Stepwise

Figure 15.23 Stepwise

The Stepwise tool determines the best predictor variables to include in a model out of a larger set of potential predictor variables for linear, logistic, and other traditional regression models.		
Group	**Input**	**Output**
Predictive	See below	See below

Input: The output stream from a Count Regression, Gamma Regression, Linear Regression or Logistic Regression tool used to create the "maximal" model. The stream can be entered into either side of the tool.

The same Alteryx data stream or XDF metadata stream that was used to create the "maximal" model. The stream can be entered into either side of the tool.

Output: O Output: Consists of a table of the serialized model with model name and the size of the object.

R Output: Consists of the report snippets generated by the Stepwise tool: a statistical summary, Type II Analysis of Deviance or ANOVA Tests, and basic diagnostic plots. The Type II Analysis of Deviance or ANOVA table and the Basic Diagnostic Plots are not produced when the data input comes from an XDF Output or XDF Input tool.

15.25　Support Vector Machine

Figure 15.24 Support Vector Machine

Support Vector Machines (SVM), or Support Vector Networks (SVN), are a prevalent arrangement of directed learning calculations initially created for grouping (all out target) issues, and late reached out to regression (numerical target) issues.

Group	Input	Output
Predictive	See below	See below

Input: The Alteryx information stream or XDF metadata stream that incorporates an objective field of interest alongside at least one conceivable indicator fields.

Output: O Output: The "O" output consists of a table of the serialized model with its model name. A Score tool and test data set can be used after obtaining the output from the SVM tool.

R Output: The "R" output consists of the report snippets generated by the Support Vector Machine tool. The report is different for classification and regression since they have different performance evaluation methods.

Classification

A basic model summary: The function call in R, target, predictors and related parameters.

Model performance

A Confusion Matrix

The SVM classification plots

Regression

A basic model summary: The function call in R, target, predictors and related parameters.

Model performance

Root mean squared error

R-squared

Mean absolute error

Median absolute error

Residual plot

Residual distribution

15.26   Test of Means

Figure 15.25 Test of Means

The Test of Means tool plays out a Welch's two specimen t-test of the distinction in mean qualities for a numeric response field between a control gathering and at least one treatment groups where group membership is shown by a gathering field that can either be a character or number field that demonstrates to which assemble each record has a place.

Group	Input	Output
Predictive	See below	See below

Note: The tests are completed between the assigned control (examination) group and alternate group demonstrated in the group field.

Input: Data streams with no less than one numeric and no less than one character or number field.

Output: D Output: The D output consists of a table that gives the two group labels for each test along with the test statistic,

degrees of freedom, and p-value of the tests which can potentially be used in downstream processing.

R Output: The R output provides a set of report snippets of the test results.

15.27   AB Analysis

Figure 15.26 AB Analysis

Determine which group is the best fit for AB testing.

Group	Input	Output
AB Testing	See below	See below

Note: This tool thinks about the rate change in a performance measure to a similar measure either over a similar era one year prior, controlling for conceivable occasional impacts) or a client determined a day and age, for two unique groups: a treatment group and control group.

Input: Data stream with the appropriate fields.

Output: A data stream with cluster assignment variable affixed.

15.28   AB Controls

 Figure 15.27 AB Controls	The tool compares the rate change in an execution measure to a similar measure either over a similar day and age one year prior (controlling for conceivable occasional impacts), or a client indicated day and age, for two unique groups (a treatment group and control group).

Group	Input	Output
AB Testing	See below	See below

Input: C Input: A Controls information stream that contains a field with a novel identifier or each control unit that permits the control units to be recognized in the execution information and a field that gives the identifier of the treatment unit to which each control unit is relegated.

T Input: A Treatments information stream that contains a field with a novel identifier for every treatment unit that permits the treatment units to be recognized in the execution information.

P Input: A Performance information stream that contains a field with an execution measure important to be thought about between the treatment and control groups, a date field recognizing the period each estimation of the execution measure to which the esteem relates, and a unit identifier field that permits the treatment and control units to be appropriately distinguished.

Output: O Output: Includes a printed outline of the test comes about, essential rundown insights of the test comes about, various perceptions of the test comes about, and a table of the aftereffects of a Welch's two samples (the treatment and control tests) t-test of the distinction in the rate change for the execution measure of intrigue.

E Output: The Extended yield is an information table comprising of the qualities used to populate the Dot Plot found in the O and I fields.

In a Dot Plot, every segment of dabs gives the rate change in Traffic from an indistinguishable period from the test time frame, yet one year prior, and the test time frame for a treatment unit and the control units allocated to that treatment unit.

G Output: The Grouped Data yield is an information table comprising of the qualities used to populate the Time Comparison Plot found in the O and I fields.

The Time Comparison Plot gives a representation of the normal impact of the test treatment over the Treatment units for the execution measure of intrigue.

I Output: The Interactive Dashboard is HTML that presentations in your program. You can communicate with the representations by tapping on the diverse realistic components to uncover more data, qualities, measurements and investigation.

15.29   AB Treatments

	The *Append cluster* tool allows us to append assignments from a K-Centroid Cluster analysis tool to a data stream.		
Figure 15.28 AB Treatments	**Group**	**Input**	**Output**
	AB Testing	See below	See below

Note: The data stream does not have to the same one the cluster output is based. However, it must contain the fields that were used to create the original cluster.

Input: Data stream with the appropriate fields.

Output: A data stream with cluster assignment variable affixed.

15.30   AB Trend

	The AB Trend tool makes measures of pattern and regular examples that can be utilized as a part of coordinating treatment to control units for A/B testing.		
Figure 15.29 AB Trend	**Group**	**Input**	**Output**
	AB Testing	See below	See below

Note: The data stream does not have to the same one the cluster output is based. However, it must contain the fields that were used to create the original cluster.

Input: Data stream with the appropriate fields:

(1) a unit identifier (e.g., a customer or store identification code); (2) a Date or Date Time field with date information for the reporting periods; and (3) a performance measure (such as sales or visitor traffic) that will be the basis of the trend and seasonality measures.

Output: A data stream with Identifier (the unit identifier code).

Trend: the calculated trend value.

Seasonality the calculated relative seasonal pattern value.

15.31   ARIMA

	The ARIMA tool assesses a time series model, either as a univariate model or one with covariates (predictors), utilizing an autoregressive integrated moving average (or ARIMA) technique.		
	Group	**Input**	**Output**
Figure 15.30 ARIMA	Time Series	See below	See below

Note: ARIMA is the most commonly used forecasting approach, and is considered to be the most general class of models for forecasting a time series field.

Input: An Alteryx information stream containing recorded information on the time arrangement to be forecast and optionally an arrangement of covariates. Fields that won't be utilized as a part of model creation can likewise be available in the information stream.

Output: O Output: Consists of an output stream containing the ARIMA model object that can be used for both point forecasts and a user-specified percentile confidence interval surrounding those forecasts.

R Output: Consists of the report snippets generated by the ARIMA tool: a statistical summary, autocorrelation diagnostic plots and forecast plots.

I Output: An interactive HTML dashboard consisting of plots and metrics. You can interact with the visualizations by

clicking on the different graphical elements to reveal more information, values, metrics and analytics.

15.32   ETS

Figure 15.31 ETS

The ETS tool estimates a univariate time series forecasting model using an exponential smoothing method.

Group	Input	Output
Time Series	See below	See below

Input: Data stream with the appropriate fields.

Output: O Output: Consists of an output stream containing the ARIMA model object that can be used for both point to point forecasts and a user-specified percentile confidence interval surrounding those forecasts.

R Output: Consists of the report snippets generated by the ARIMA tool: a statistical summary, autocorrelation diagnostic plots and forecast plots.

I Output: An interactive HTML dashboard consisting of plots and metrics. You can interact with the visualizations by clicking on the different graphical elements to reveal more information, values, metrics and analytics.

15.33   TS Compare

|

Figure 15.32 TS Compare	The TS Compare tool analyzes one or more models made with either the ETS or ARIMA macros, including ARIMA models that use covariates.

Group	Input	Output
Time Series	See below	See below

Input: A set of time series models that predict the same field and the Alteryx data set with same fields but for a time period following the one used in the models.

Output: O Output: Contains a data stream of the names of each model examined and its accuracy statistics.

R Output: Consists of the report snippets of a table with the actual and forecast values, a table of the accuracy statistics for each model, and a plot that shows all the values of the time series and forecast values for all the models being compared.

I Output: An interactive HTML dashboard consisting of plots and metrics. You can interact with the visualizations by clicking on the different graphical elements to reveal more information, values, metrics and analytics.

15.34   TS Covariate Forecast
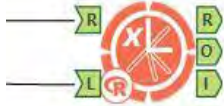
	The TS Covariate Forecast tool provides forecasts from an ARIMA model estimated using covariates for a user-specified number of future periods.		
Figure 15.33 TS Covariate Forecast	**Group**	**Input**	**Output**
	Time Series	See below	See below

Input: The output data stream from either a TS ARIMA or TS ETS model tool.

A data stream that contains the relevant (future) covariate information, can be forecasted values as well.

Output: O Output: Contains a data stream of the forecast values and confidence bounds.

R Output: Consists of the report snippets of a table with the forecast values and confidence bounds and a forecast plot that shows both the historical values of the time series and forecast values with error bounds.

I Output: An interactive HTML dashboard consisting of plots and metrics.

15.35   TS Filler

Figure 15.34 TS Filler	The Time Series Filler tool fills any gaps in a time series data stream.		
	Group	**Input**	**Output**
	Time series	See below	See below

Input: Data stream with the appropriate fields with date or date time column.

Output: Every row entering the macro will appear on the output. In addition, the macro will evaluate if any periods in the time series are missing, and if so, will generate a row to 'fill' the gap.

15.36   TS Forecast

Figure 15.35 TS Forecast	The TS Forecast tool provides forecasts from either an ARIMA or ETS model, for the number of periods the user has specified.		
	Group	**Input**	**Output**
	Time series	See below	See below

Input: Output from ARIMA or ETS model tool.

Output: O Output: Contains a data stream of the forecast values and confidence bounds.

R Output: Consists of the report snippets of a table with the forecast values and confidence bounds and a forecast plot that shows both the historical values of the time series and forecast values with error bounds.

I Output: An interactive HTML dashboard consisting of plots and metrics.

15.37   TS Plot

  Figure 15.36 TS Plot	The TS Plot tool gives various distinctive univariate time arrangement plots that are valuable in both better understanding the time information and deciding how to build up a determining model.		
	Group	**Input**	**Output**
	Time Series	See below	See below

Note: The *Accessible Plot* helps survey whether the first run through arrangement should be changed and checks for exceptions; the *Occasional Plot* considers an evaluation of the presence and nature of any regularity; The *Occasional Deviation Plot* takes into account an appraisal of whether the time arrangement fluctuates crosswise over seasons; *Auto Regression Capacity* and *Halfway Auto Regression Work Plots* help decide the way of any basic autocorrelation and in evaluating the information differencing for the formation of an ARIMA display; and *Period Arrangement Disintegration Plot* takes into consideration a visual examination of the first information, pattern, regularity in the information, and data about the residuals once regularity and pattern have been considered.

Input: Data stream with the appropriate fields.

Output: An Alteryx R graph

I Output: A HTML dashboard with plots and metrics

15.38   Boutique Investment Bank

	From ▼	Google Apps
Send	To...	Alteryx Consultants
	Cc...	
	Subject	Forecasting Business Value

Hey,

We are ABC LLC (a **boutique investment bank**) and have stock information for a merger deal and want to do some predictive analysis on the dataset. We think the best way is to forecast (within a small error margin) the price of the stocks , based on its previous values. The dataset for the purpose of the exercise has been shared.

Based on the domain understanding , we think that we could use the data for time series modelling to do some forecasting .

Can one of you help us to look at time series forecasting tools in Alteryx and then compare the output of the different models with the performance metrics and the best one can be finally used to forecast for the future?

Let's get the ball rolling on the Forecasting bit and we can then do the model comparison post that.

To start working on this deliverable, we will need to keep Predictive analytics sample workflow 15 as a reference and use any stock data from the public forum, here we have used a dataset from google.
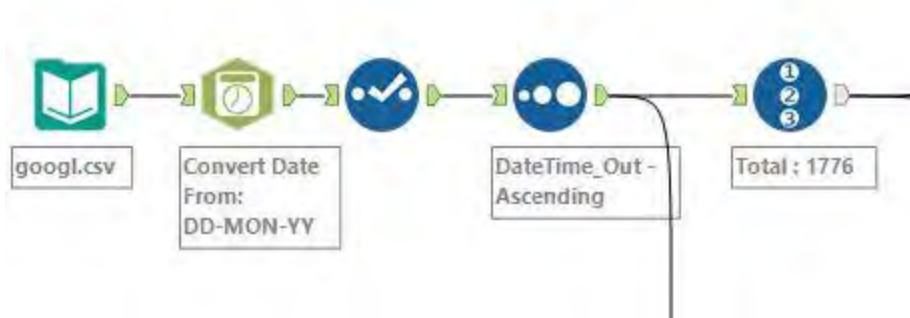
Figure 15-37 – Boutique Investment Firm Input Section

Once we input the data, we would need to convert the date string to date in the required format since that is the most important metric for the analysis. We then select the appropriate columns which are Date Time_Out and Close and sort it in ascending order of the Dates. Post that we add a column for record count using *Record ID* tool. This is relevant when have to split the input into 90% and 10% for training and verification purposes to make it ready for the Predictive Analysis.

Using the 90% of the data train the time series models you want to test.
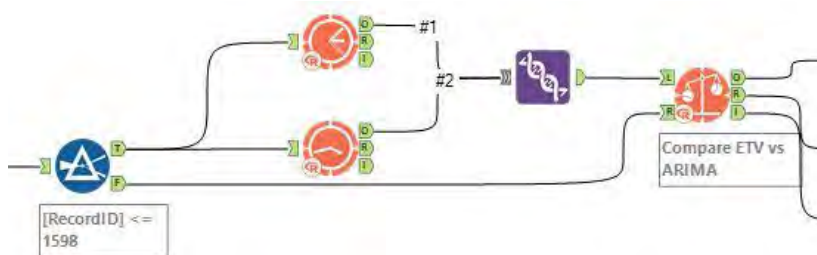
Figure 15-38 – Boutique Investment Firm -ETS Model and ARIMA

The Model1 is an *ETS model* and Model 2 is an *ARIMA Model*. For the *ETS Model*, we need to make sure we select the appropriate target field and target field frequency in the tool configuration. A union of these two is fed to a TS compare tool that returns various parameters than can be used to determine a suitable model for the given data.

Suppose we use the *ARIMA Model* with the given data set, the output from this tool is fed to a TS forecast tool with the number of periods into the future the forecast is necessary(configuration). At the end of the workflow, we would have tested out the time series models and then used the best one for forecasting.
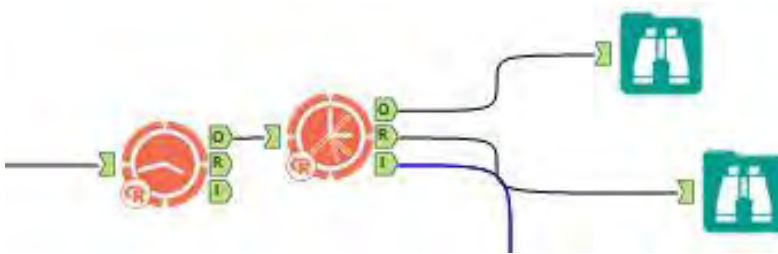
Figure 15-39 – Boutique Investment Firm -Comparison of ETS Model and ARIMA using TS Compare

When we compare the model outputs using the TS compare we get three outputs – *Output*, *Report*, and *Interactive.* The output is of highest importance since it provides us the complete statistics of the models' performance using the training set. The two main components to understand here are the mean absolute percentage forecast error (MAPE), and the mean absolute scale error (MASE). The MASE is a better metric to compare the numbers and the lower the value, the better is the fit. In our case, we can see that the MASE for ARIMA is lower than ETS.

Figure 15-40 – Boutique Investment Firm -Statistics of ETS Model and ARIMA using TS Compare

The ARIMA model hence is a better fit to the current stock data. The report generated from the tool is very important to note the exact values about the Model.

Figure 15-41 – Boutique Investment Firm -ARIMA Model Statistics

This report has all the details about the model and also contains forecasted values for a given date range. The final workflow would be in the below format
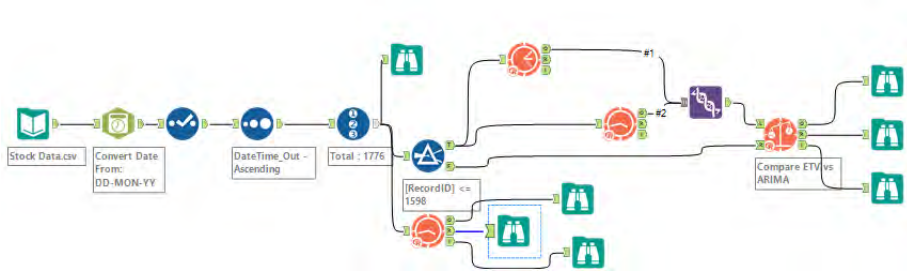
Figure 15-42 – Boutique Investment Firm stream on completion

15.39   Blood Donation

The data to be used here for the analysis is from New donor score sample example in predictive analytics since they

wanted us to build it on generic data which they would then use with their dataset. The app is an extension of the existing New donor score sample example - sample workflow. The first step to solving this is to load the data in Alteryx and take a look at it using the browse tool.

Figure 15-43 – Blood Donation -Input Data

From the Browse tool, we understand that the data is about all the details of various students and their activities in the college. The next step is to create sample data from this dataset to be used for the prediction, so we use a *Create Samples* tool to do that.
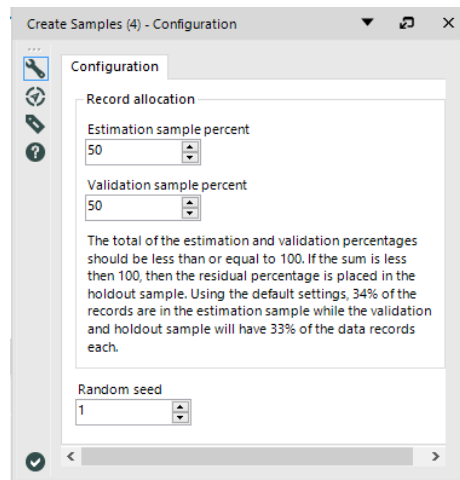
Figure 15-44 – Blood Donation –Samples Configuration

We use a 50-50 split to split it right in the center and have a reasonably same data.
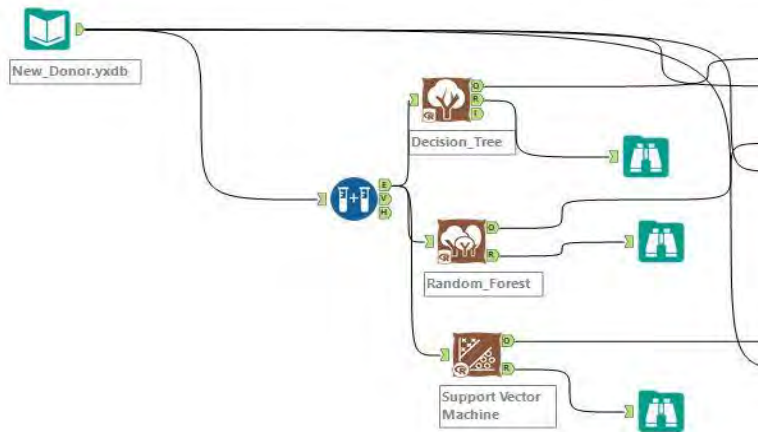
Figure 15-45 – Blood Donation – Predictive Analytics Tools

Let us start with the idea that, we have to classify donor and say whether the donate field is 'YES' or 'NO'. For the purpose of this case the tools used are *Decision Tree, Random Forest,* and *Support Vector Machine* tool. *Note*: The use of Random forest will help us negate the overfitting that can happen with a decision tree. The sampling of data will split the data into estimation and validation sets.
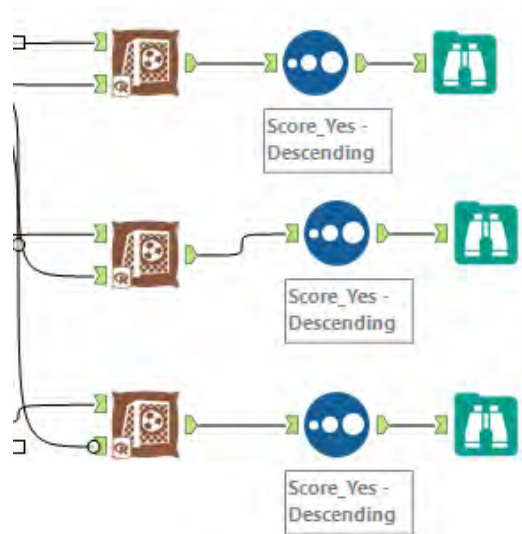
Figure 15-46 – Blood Donation – Predictive Analytics Tools Scoring

The output from the models is scored against the input data and then fed into a sorting tool. We can compare the outputs from the sorted data to see how the models perform.
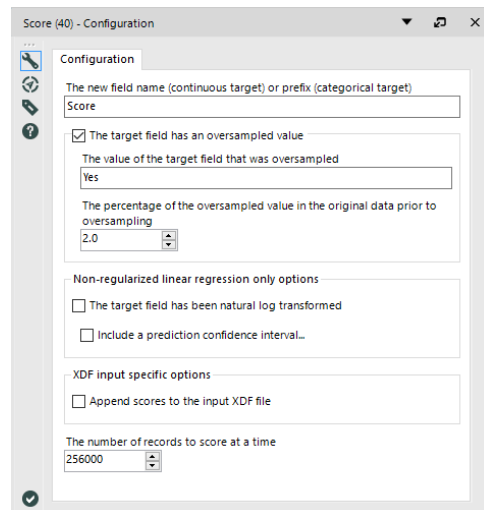
Figure 15-47 – Blood Donation –Score Tool Configuration

We have set up the configuration for the scoring model as shown above to get the appropriate solution. We can use this technique to determine the best prediction model.

Here is a sample of the output generated from one model's score.

Figure 15-48 – Blood Donation –Score Tool Sample Output

The final output shows us that the higher the *Score_Yes* value is, higher is the chance of the person donating blood.

The *Blood Donation* data stream should look like this when it's complete.
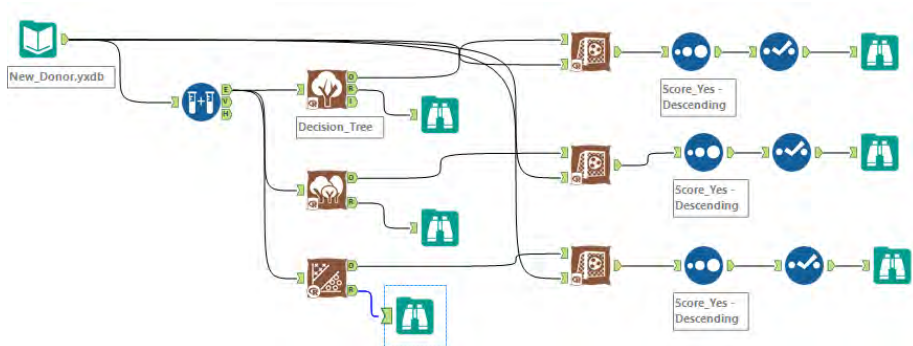
Figure 15-49 – Blood Donation final data stream when complete

CHAPTER 16

Self-Guided Solutions

Readers are strongly encouraged not to use these solutions when solving the self-guided examples. There are many approaches to answering these problems, and figuring out how to do it on your own is the goal of these exercises. These screenshots of solutions are provided so that if you're stuck, you can refer to the logic of how the examples direct you to a solution.

16.1 The Games: How are we doing?

Since the complete workflow is large, it is broken down into 2 parts for better understanding.
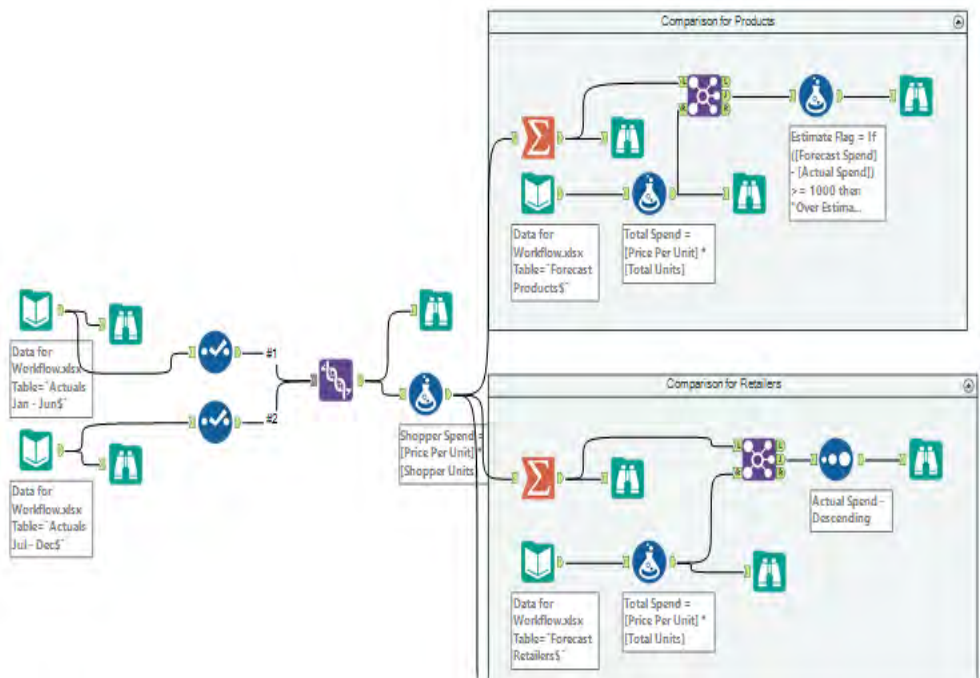
Figure 16-1 How are we doing - part1
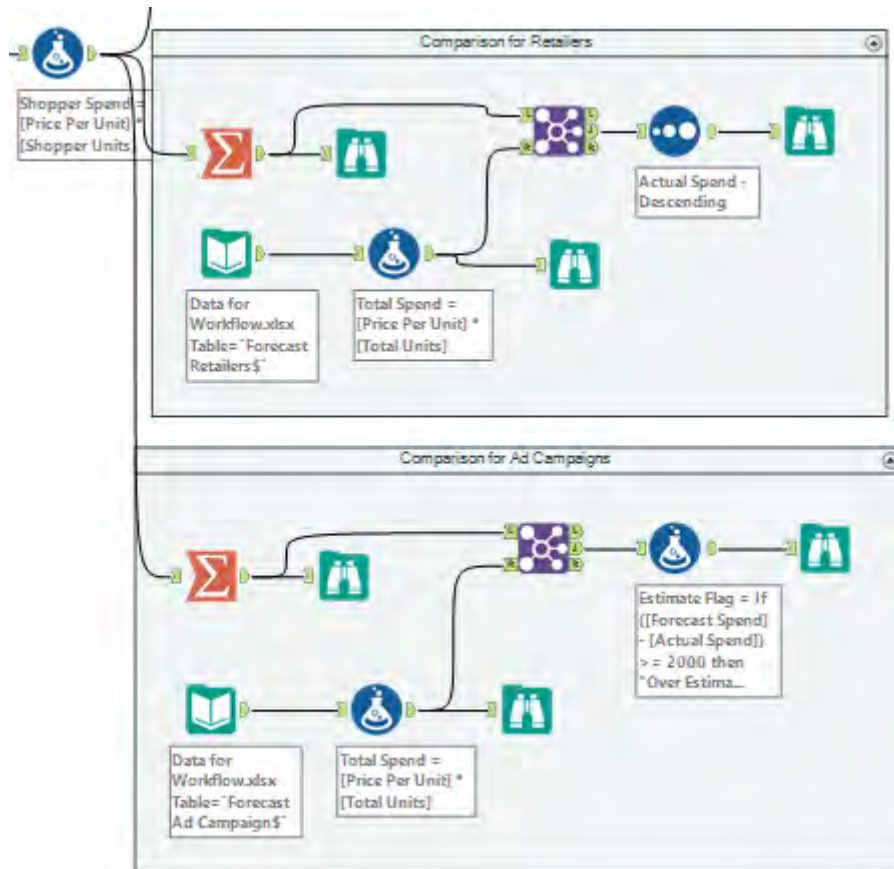
Figure 16-2 How are we doing - part2

16.2   Unisex Baby Names: What's In a Name?
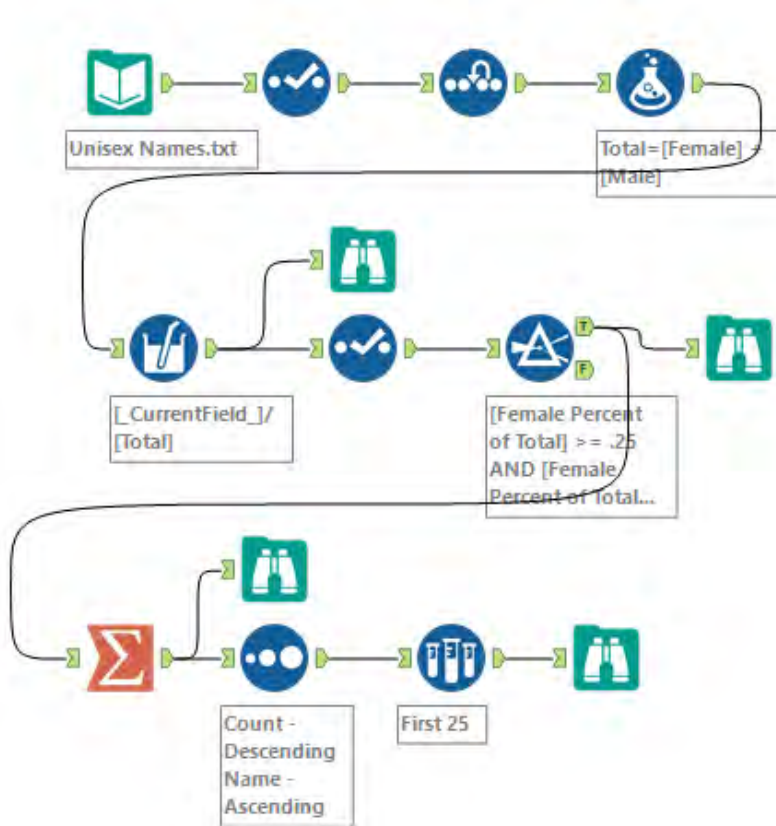
Figure 16-3 What's in a Name

16.3 The Direct Approach: Where, Oh Where Have My Three Files Gone?
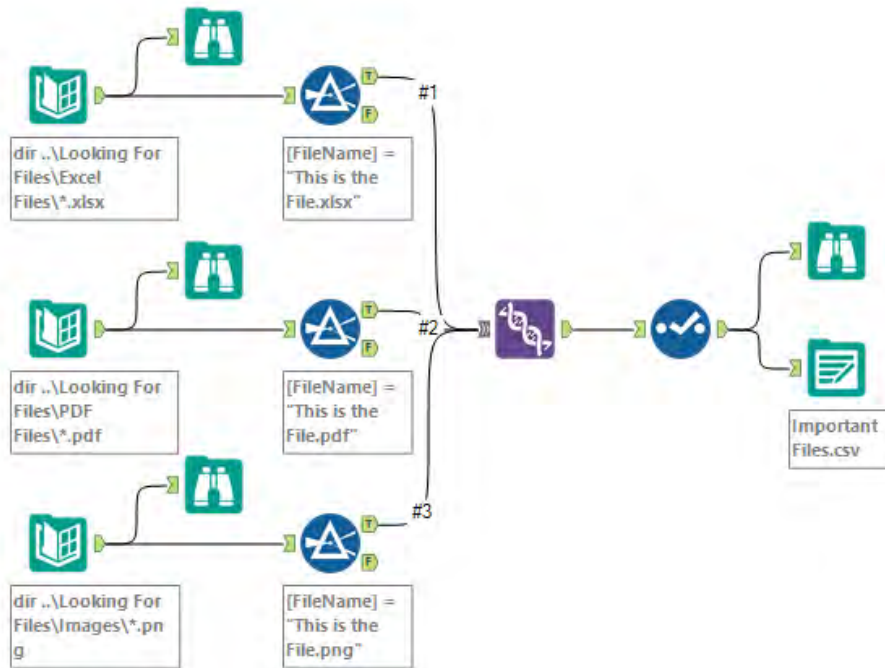
Figure 16-4 Where Have my Three Files Gone

16.4 Cultural Musing: Culturally Divided
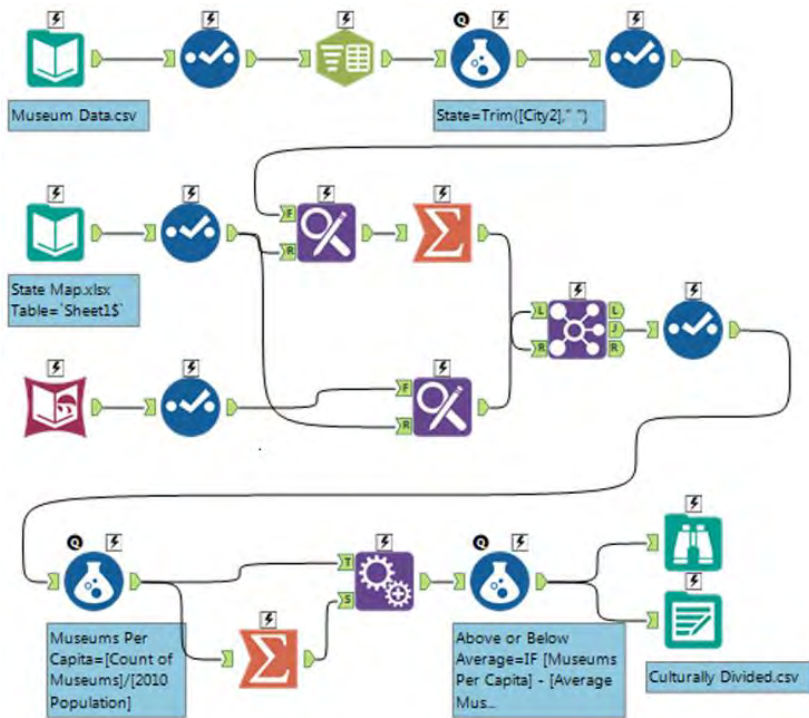
Figure 16-5 – Culturally Divided

16.5 The Sport Report: FOOTBALL!
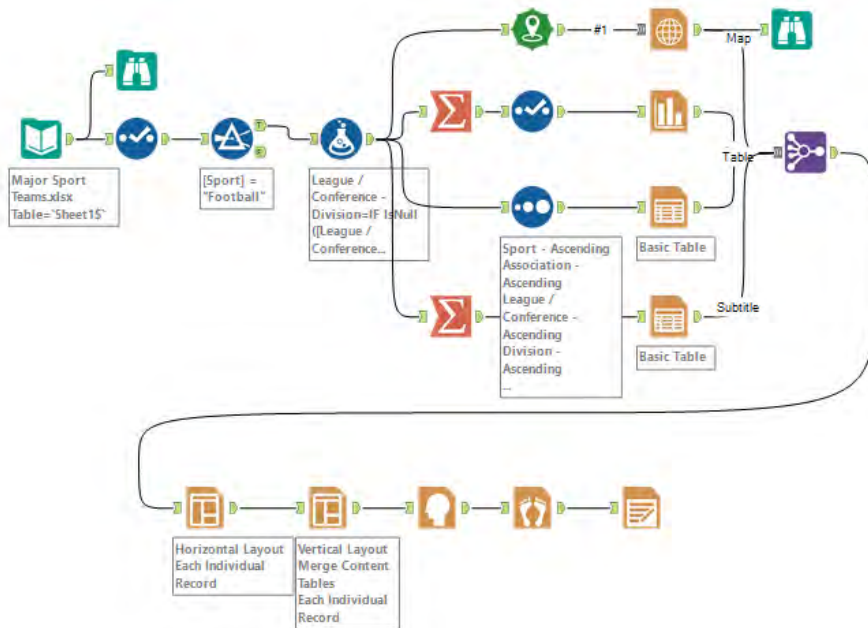
Figure 16-6 Football!

16.6 Expensive Beauty Products: More Flags

      Since the complete workflow is large, it is broken down into segments for better understanding.

Figure 16-7 More Flags
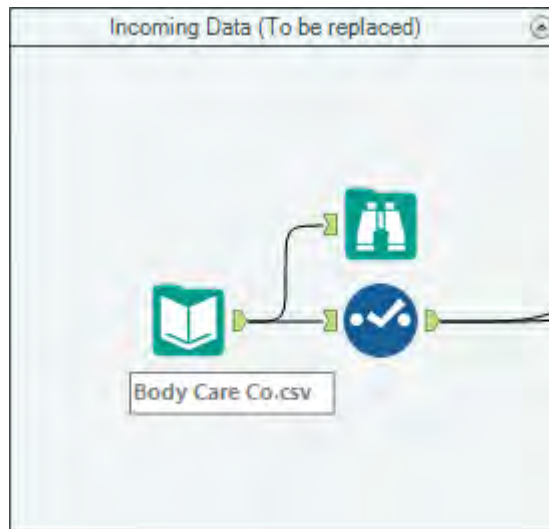
Figure 16-8a – Incoming Data
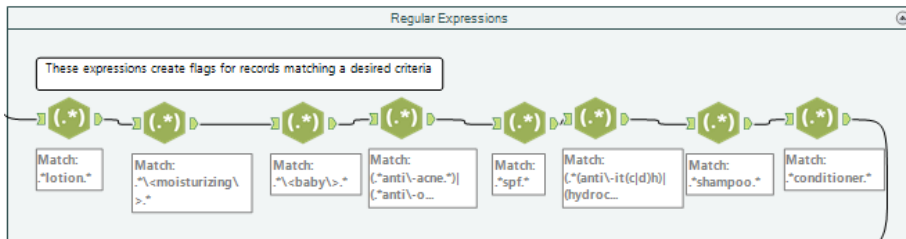
Figure 16-8b -Regular Expressions
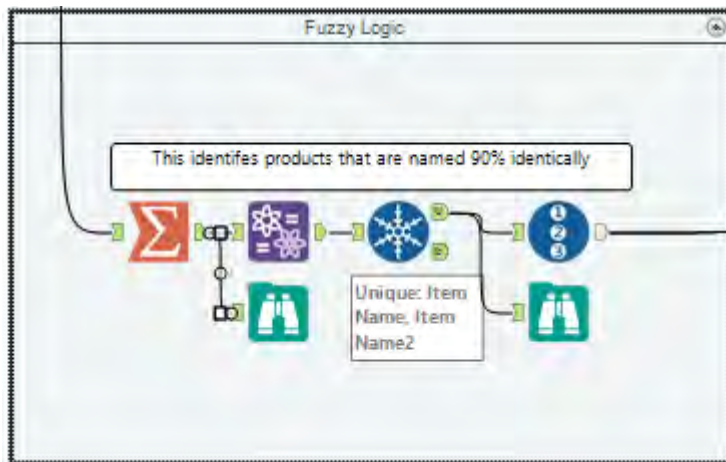
Figure 16-8c – Fuzzy Logic
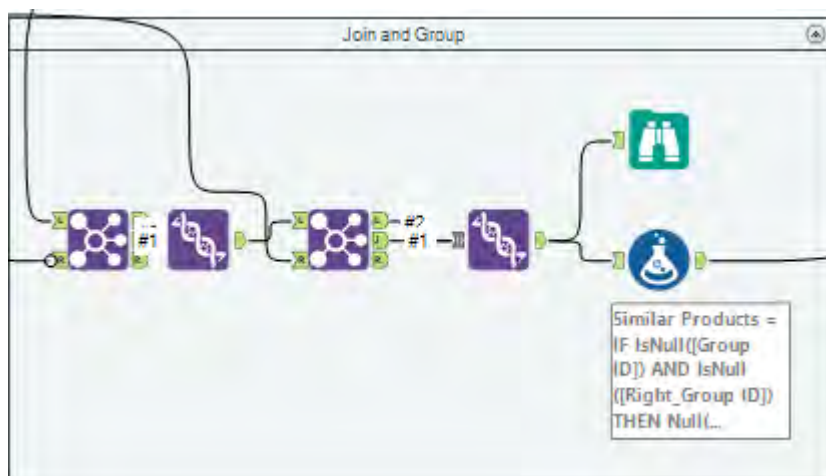
Figure 16-8d – Join & Group
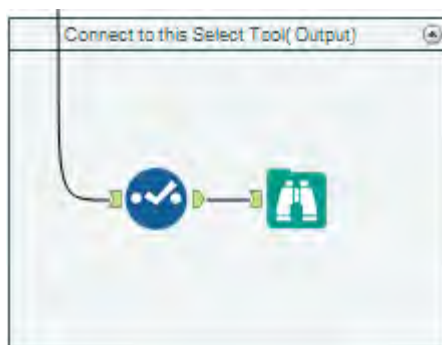
Figure 16-8e - Output

16.7 Applications Wanted: To Summarize or Not to Summarize

Since the complete workflow is large, it is broken down into segments for better understanding.
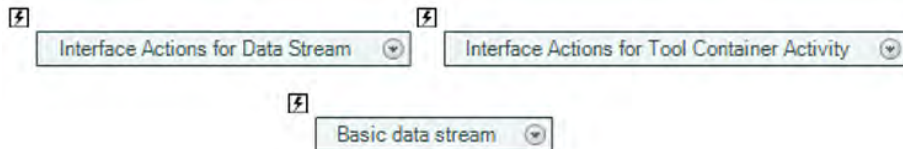
Figure 16-9 – To Summarize or Not to Summarize
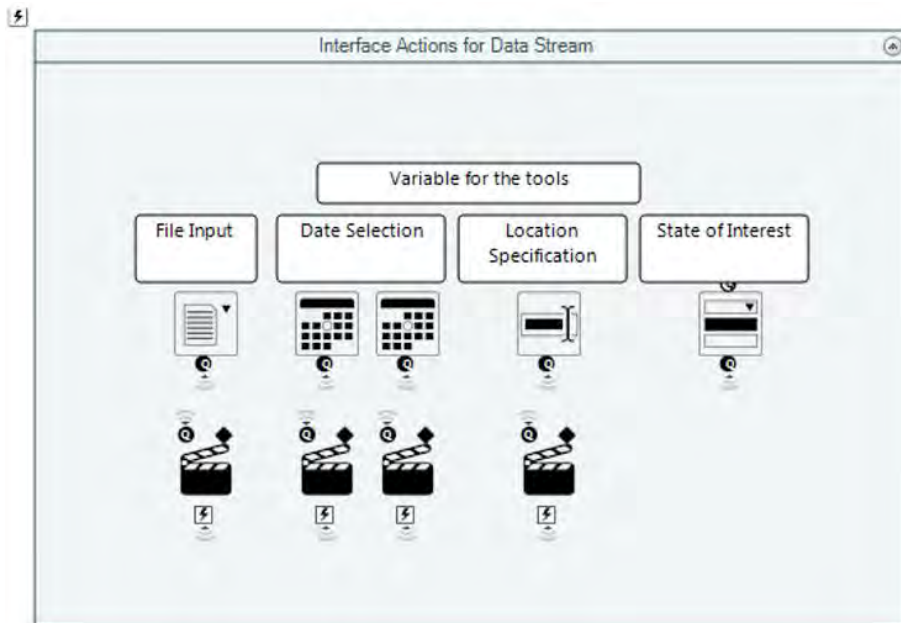
Figure 16-9a -Interface Actions for Data Stream
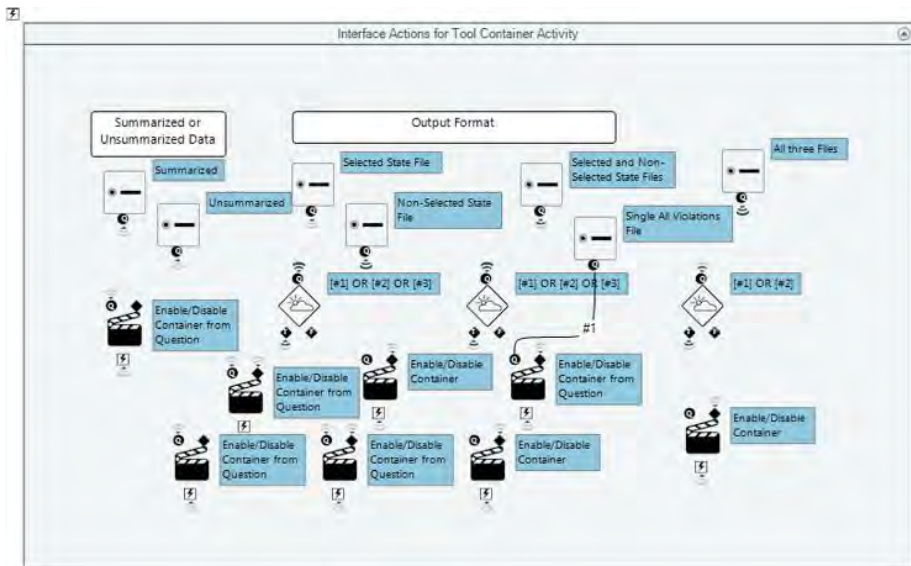
Figure 16-9b – Interface Actions for Tool Container Activity
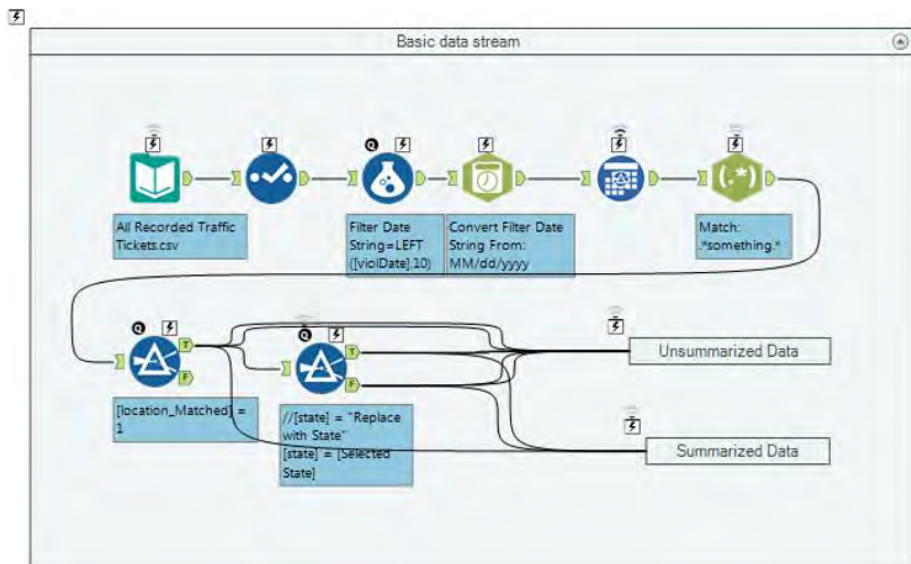
Figure 16-9c – Basic Data Stream

16.8 Where's The Joe? Who wants Coffee
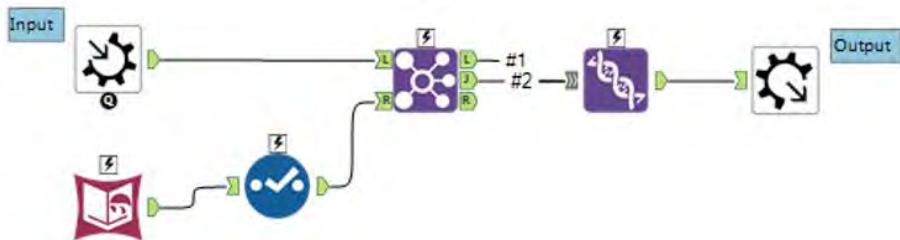
Figure 16-0-10a – Who Wants Coffee? – Macro
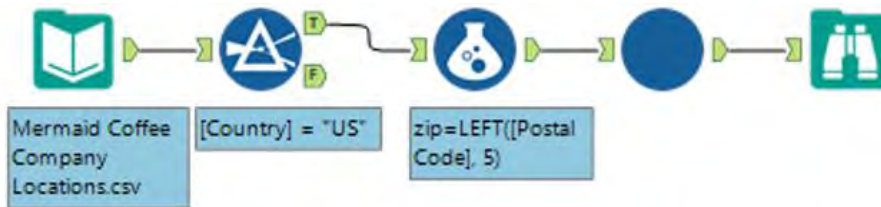
Figure 16-10b – Who wants Coffee? - Workflow

16.9 Metamorphosis: Narrowing Down the Search
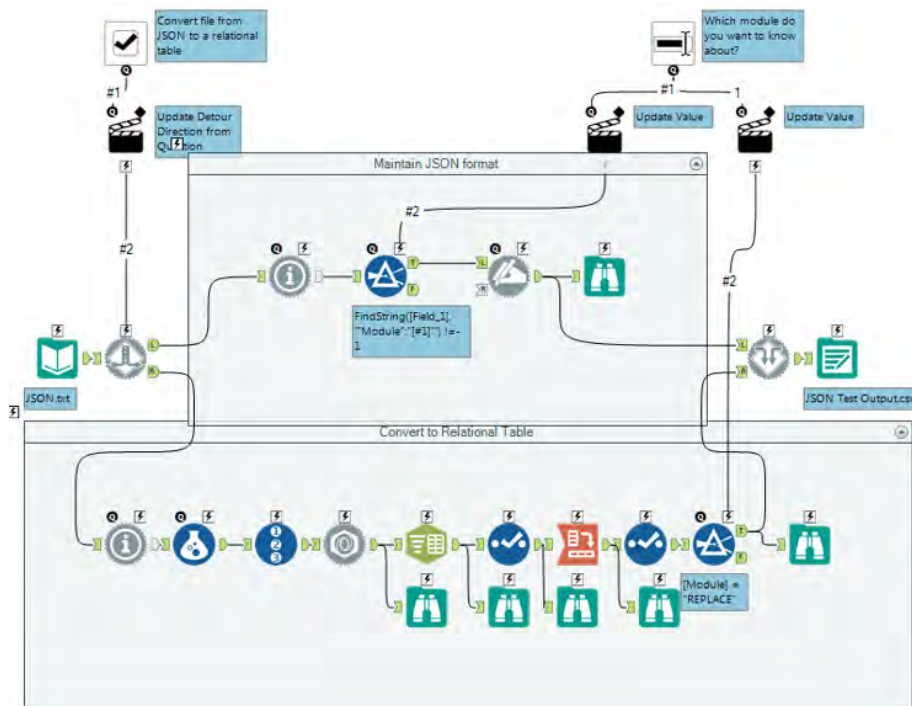
Figure 16-11 -Metamorphosis

16.10 How's the oil market doing?
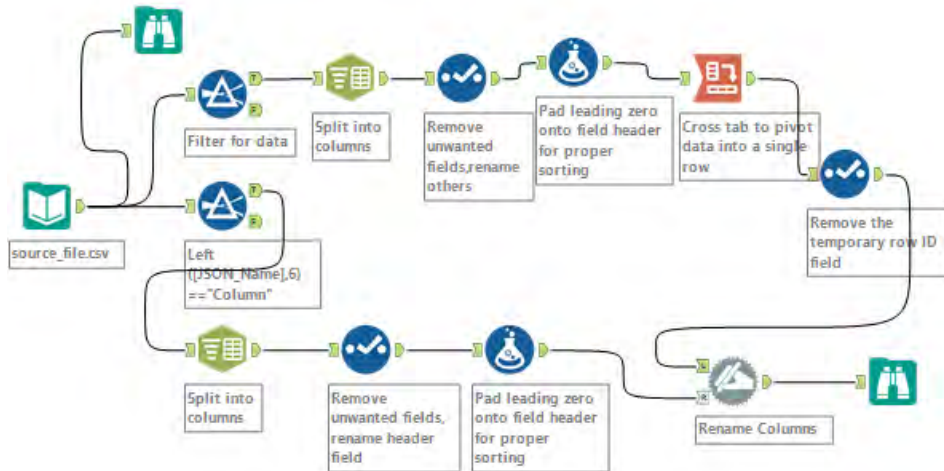
Appendices

Appendix A – File Types

.aws	Allocate Workspace
.cydb	Calgary database
.pcxml	XML data output type that allows the end user to select from multiple file types for a final output when using Alteryx Gallery
.yxdb	Alteryx database
.yxft	Alteryx Header File
.yxlc	Alteryx License File
.yxmc	Alteryx Macro File
.yxmd	Alteryx Workflow File
.yxwv	Alteryx Analytic App Values File which allows predefined selections to be made.
.yxwz	Alteryx Analytic App File
.yxzp	Packaged Alteryx File

Appendix B – Hot Keys

Ctrl+Alt+B	Show/Hide Toolbar
Ctrl+Alt+D	Show/Hide Interface Designer
Ctrl+Alt+O	Show/Hide Output
Ctrl+Alt+P	Show/Hide Properties
Ctrl+Alt+T	Show/Hide Tool Palette
Ctrl+Alt+V	Show/Hide Overview
Ctrl+C	Copy
Ctrl+F	Fine
Ctrl+F4	Close workflow
Ctrl+N	New workflow
Ctrl+O	Open workflow
Ctrl+R	Run/Cancel Workflow
Ctrl+S	Save workflow
Ctrl+Shift+B	Add a Browse tool after all selected tools that a Browse tool can be connected to
Ctrl+V	Paste
Ctrl+X	Cut
Ctrl+Y	Redo
Ctrl+Z	Undo
Ctrl++	Vertically align selected tools
Ctrl+-	Horizontally align selected tools
Del (Delete)	Deletes selected part of workflow
F1	Open Help menu
F5	Refresh Configuration

Appendix C – Downloads/Content

US 2010 Census SF1
http://downloads.alteryx.com/data.html

USGS North America Map
http://downloads.alteryx.com/data.html

Visual Analytics Kit (Tableau)
http://alteryx.com/kit

Visual Analytics Kit (Qlik)
http://pages.alteryx.com/VisualAnalyticsKitforQlik_Reg-LP.html

Alteryx Gallery
https://gallery.alteryx.com

Appendix D – Field Types

Type	Class	Description
Bool	Boolean	A field with two values, True and False
Byte	Numeric	A field with integers between 0 and 255
Int16	Numeric	A field with integers between -32,768 and 32,767
Int32	Numeric	A field with integers between -2,147,483,648 and 2,147,483,647
Int64	Numeric	A field with integers between –9,223,372,036,854,775,808 and 9,223,372,036,854,775,807
Fixed Decimal	Numeric	A field with a specific width including the number of decimal places
Float	Numeric	A field with up to 7 digits of accuracy between -3.4^38 and 3.4^38
Double	Numeric	A field with up to 15 digits of accuracy between -1.7^308 and 1.7^308
String	String	A field with up to 8192 characters
WString	String	A field with up to 8192 characters that will accept Unicode characters
V_String	String	A field that will adjust the size depending on the length of the strings in it

V_WString	String	A field that will adjust the size depending on the length of the strings in it and will accept Unicode characters
Date	String	A field in the format "yyyy-mm-dd"
Time	String	A field in the format "hh:mm:ss"
DateTime	String	A field in the format "yyyy-mm-dd hh:mm:ss"
Blob	Blob	A field with image or sound files
SpatialObj	Blob	A field with points, lines, polylines, or polygons.

Appendix E- Properties Window
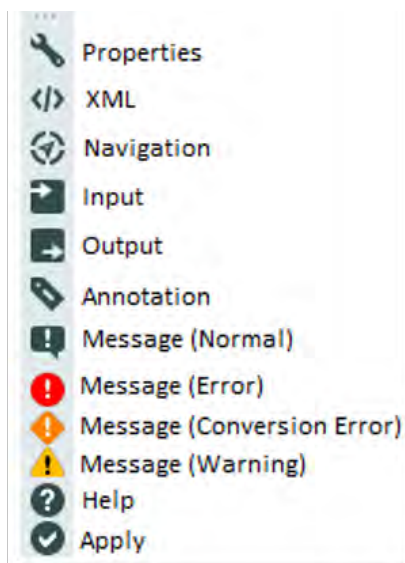
Figure 12-11 – Properties Window

Appendix F – Boolean Expressions

Expression	Meaning
A = B	Expression A equals expression B
A != B	A does not equal B
A > B	A is greater than B
A >= B	A is greater than or equal to B
A < B	A is less than b
A <= B	A is less than or equal to B
A	A is True
NOT A	A is False
A IN B	A is in the set of B things
A NOT IN B	A is not in the set of B things
A OR B	Expression A is True or expression B is True or both expressions are True
A AND B	Expression A is True and expression B is True
A OR (B AND C)	Expression A is True or expression B and expression C are both True or expression A and expression B and C are True
A AND (B OR C)	Expression A is True and expression B, expression C, or expressions B and C are True

Basic Boolean expressions can be combined to create more complex logic by replacing A, B or C with any basic Boolean Expression.

Appendix G – Data Components

yyyy-MM-dd	
yyyyMMdd	
MM/dd/yy	
MM/dd/yyyy	day - Spelled out weekday
MM-dd-yy	dd – 2 digit day of month
MM-dd-yyyy	dy – Abbreviated day of week
Month dd, yyyy	
Month, yyyy	hh – 2 digit hour
Mon dd	
dd/MM/yy	mm – 2 digit minute
dd/MM/yyyy	MM – 2 digit month of year
dd-MM-yy	
dd-MM-yyyy	Mon – Abbreviated month of year
dy., Month dd, yyyy	Month – Spelled out month
day, dd Month, yyyy	
dd Month, yyyy	ss – 2 digit second
yyyy-MM-dd hh:mm:ss	yy – 2 digit year
MM/dd/yyyy hh:mm:ss	
MM/dd/yy hh:mm:ss	yyyy – 4 digit year
dd/MM/yyyy hh:mm:ss	
dd/MM/yy hh:mm:ss	

Figure 12-12 – Date Components

Appendix H – Date/Time Units

Keywords
Years
Months
Days
Hours
Minutes
Seconds

Appendix I – RegEx Cheat Sheet

.	Any Character
$	End of a line
()	Marked Group
(?:)	Unmarked Group
*	Repeat the previous thing 0 or more times
+	Repeat the previous thing 1 or more times
[ ]	A set to be used for optional lists of single characters
[^]	A set to be used for optional lists of single characters to be excluded
[[:alpha:]]	Any letter
\	Escape the following symbol
\<	Beginning of a word
\>	End of a word
\d	Digit
\l	Lowercase letter
\n	New line character
\s	Space
\t	Tab character
\u	Uppercase letter
\w	Word characters
\W	Non-word characters
^	Beginning of a new line or "not" depending on context
|	Or

Appendix J – Action Tool Sets

These tools do not receive connections from an Action tool.

Action	Date	Folder Browse	Numeric Up Down
Checkmark	Drop Down	List Box	Radio Button
Comment	Error Message	Macro Input	Text Box
Condition	Explorer Box	Macro Output	Tree
Control Parameter	File Browse	Map	

All other tools have at least these basic methods

Delete Tool from Workflow
Update Field List
Update Raw XML with Formula
Update Select with Reverse Field Map
Update Value
Update Value with Formula

These tools only have the basic methods.

AB Analysis	Data Stream In	Join In-DB	Report: Comparison
AB Controls	Data Stream Out	Join Multiple	Report: Detail
AB Treatments	DataSift Connector	JSON Build	Report: Rank

AB Trend	Date Filter	JSON Parse	Run Command
Allocate Append	Date Time Now	K-Centroids Cluster Analysis	Running Total
Allocate Input	DateTime	K-Centroids Diagnostics	Sample
Allocate Metainfo	Decision Tree	Layout	Sample In-DB
Allocate Report	Detail Fields	Lift Chart	Scatterplot
Amazon S3 Download	Detour End	Linear Regression	Score
Amazon S3 Upload	Distance	Logistic Regression	Select In-DB
API Output	Distribution Analysis	Make Columns	Select Records
Append Cluster	Dynamic Input	Make Grid	Smooth
Append Fields	Dynamic Rename	Make Group	Spatial Info
ARIMA	Dynamic Replace	Map Legend Builder	Spatial Match
Arrange	Dynamic Select	Map Legend Splitter	Spatial Process
Association Analysis	Email	Marketo Append	Spearman Correlation
Auto Field	ETS	Marketo Input	Spline Model
Base 64 Encoder	Field Info	Marketo Output	Stepwise
Behavior Metainfo	Field Summary	MB Inspect	Street Geocoder

Blob Convert	Filter	MB Rules	Summarize
Blob Output	Filter In-DB	Message	Summarize In-DB
Block Until Done	Find Nearest	Multi-Field Binning	Support Vector Machine
Boosted Model	Find Nearest Neighbors	Multi-Field Formula	Table
Browse	Find Replace	Multi-Row Formula	Tableau Workbook Macro
Browse Data In-DB	Footer	Naïve Bayes Classifier	Test
Buffer	Forest Model	Nested Test	Test of Means
Business Match (US)	Formula	Neural Network	Text To Columns
Calgary Input	Formula In-DB	Non Overlap Drivetime	Throttle
Calgary Join	Foursquare Search	Overlay	Tile
Calgary Loader	Frequency Table	Oversample Field	Transpose
Canada Geocoder	Fuzzy Match	Parse Address	TS Compare
CASS	Gamma Regression	Pearson Correlation	TS Covariate Forecast
Charting	Generalize	Plot of Means	TS Filler
Cluster Code	Generate Rows	Poly-Build	TS Forecast
Compare Behavior	Gnip Search	Poly-Split	TS Plot

Connect In-DB	Google Analytics	Principal Components	Twitter Search
ConsumerView Matching	HDFS Input	Profile Input	Union
Contingency Table	HDFS Output	Profile Output	Union In-DB
Count Records	Header	R	Unique
Count Regression	Heat Map	Random % Sample	US Geocoder
Create Points	Heat Plot	Record ID	US Zip 9 Coder
Create Profile	Histogram	RegEx	Violin Plot
Create Samples	Household File Matching	Render	Weighted Average
Cross Count	Image	Report Map	Write In-DB
Cross Count Append	Imputation	Report Text	XML Parse
Cross Tab	Join		

The following tools have the basic tools plus

Update Password

Download	MongoDB Input	Salesforce Input	SharePoint List Input
	MongoDB Output	Salesforce Output	SharePoint List Output

The following tools have the basic tools plus

Update Input Data Tool

Blob Input	Input Data

The following tools have the basic tools plus unique methods

Detour	Update Detour Direction from Condition
	Update Detour Direction from Question
Directory	Update Directory with Folder Browse
Map Input	Update Map Input with Map Question
Output Data	Update Output Data Tool
Select	Update Select with multi-select ListBox
Sort	Update Sort Order
Text Input	Update Cell
Tool	Enable/Disable Container From Condition
Container	Enable/Disable Container From Question
Trade Area	Update Trade Area Radius

Appendix K – Directory Tool Data

FullPath	The direct path to open the file.
Directory	The direct path to the folder the file is in.
FileName	The actual name and extension of the file.
ShortFileName	A short file name provided for compatibility with legacy software.
CreationTime	This indicates the first time the file was created.
LastAccessTime	This indicates the last time the file was opened.
LastWriteTime	This indicates the last time the file was saved.
Size	This is the total size of the file in bytes.
AttributeArchive	This indicates if the file is archived.
AttributeCompressed	This indicates if the file is compressed.
AttributeEncrypted	This indicates if the file is encrypted.
AttributeHidden	This indicates if the file is hidden.
AttributeNormal	This indicates if the file has normal attributes.
AttributeOffline	This indicates if the data is unavailable.
AttributeReadonly	This indicates if the file is read-only.
AttributeResparsePoint	This indicates that the file is associated with a reparse point.
AttributeSparseFile	This indicates that the file is sparse.
AttributeSystem	This indicates that the file is a system file.
AttributeTemporary	This indicates that the file is a temporary file.

Credited Original Data Sources

The Games

- Medals:
- o Access through Tableau Public and assumed to be in the public domain
  https://public.tableau.com/profile/a.m.5517#!/vizhome/OlympicMedals2000-2012_0/Dashboard2 published by A.M.
- Nobel Laureates:
- o Access through Tableau Public and assumed to be in the public domain:
  http://tableausoftware.co.nz/public/gallery/nobelprizes published by Ross Perez
- GDP per Capita:
- o Accessed through GitHub
  https://github.com/cschin/ipython_d3_mashup/blob/master/ipython_13_vis_example/gm_data/GDPpercapitaconstant2000US.csv published by **Jason Chin**

Unisex Baby Names

- Unisex Names:
- o Access through Tableau Public and assumed to be in the public domain
  http://tableausoftware.co.nz/public/gallery/unisex-baby-names published by Steve Ruble

The Direct Approach

- Created by Author
- Special Thanks to http://watchout4snakes.com/ for use of their random word generator in creating the names of the folders in the exercise Where, Oh Where Have The Three Files Gone?

Cultural Musings

- GDP per Capita:
- o List of U.S. states by GDP per capita. (2015, March 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 00:01, May 12, 2015,
  from http://en.wikipedia.org/w/index.php?title=List_of_U.S._states_by_GDP_per_capita&oldid=653869588
- Museum Data:
- o Data retrieved from Data.Gov (http://data.gov) Direct Link:
  http://catalog.data.gov/dataset/museums-universe-data-file-mudf-fy-2014-3rd-quarter/resource/d4b36e0f-1d9d-4c2a-8882-138c15763717
- State Map:
- o Created by Author

Sport Report

- Major US Sport Teams:
o The author created this data using Wikipedia pages and tools
▪ Primary Wikipedia Article
• List of professional sports teams in the United States and Canada. (2015, March 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 14:34, January 7, 2015, from
http://en.wikipedia.org/w/index.php?title=List_of_professional_sports_teams_in_the_Unite
d_States_and_Canada&oldid=650502141
▪ The Geocoded locations of each of the sites were accessed by using the coordinates link
embedded in their web pages with utilizes a Wikimedia Tool called GeoHack which can be
accessed directly at http://tools.wmflabs.org/geohack/
• Turner Field. (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:02, April 7,
2015, from http://en.wikipedia.org/w/index.php?title=Turner_Field&oldid=653079052
• Marlins Park. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Marlins_Park&oldid=649550406
• Citi Field. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Citi_Field&oldid=653482419
• Citizens Bank Park. (2015, March 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7,
2015, from http://en.wikipedia.org/w/index.php?title=Citizens_Bank_Park&oldid=649435153
• Nationals Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:03, April 7,
2015, from http://en.wikipedia.org/w/index.php?title=Nationals_Park&oldid=655186324
• Wrigley Field. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:04, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Wrigley_Field&oldid=655275853
• Great American Ball Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:04,
April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Great_American_Ball_Park&oldid=6552856
31
• Miller Park (Milwaukee). (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:04,
April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Miller_Park_(Milwaukee)&oldid=655121188
• PNC Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:05, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=PNC_Park&oldid=654838712
• Busch Stadium. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:05, April 7,
2015, from http://en.wikipedia.org/w/index.php?title=Busch_Stadium&oldid=649550165
• Chase Field. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:05, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Chase_Field&oldid=655057408
• Coors Field. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:09, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Coors_Field&oldid=654538824
• Dodger Stadium. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:09, April 7,
2015, from http://en.wikipedia.org/w/index.php?title=Dodger_Stadium&oldid=654869845
• Petco Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:10, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=Petco_Park&oldid=654538750
• AT&T Park. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:10, April 7, 2015,
from http://en.wikipedia.org/w/index.php?title=AT%26T_Park&oldid=654538692

- Oriole Park at Camden Yards. (2015, March 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Oriole_Park_at_Camden_Yards&oldid=651898983
- Fenway Park. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Fenway_Park&oldid=653504103
- Yankee Stadium. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Yankee_Stadium&oldid=654863343
- Tropicana Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:11, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Tropicana_Field&oldid=655160166
- Rogers Centre. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:12, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Rogers_Centre&oldid=654913445
- U.S. Cellular Field. (2015, March 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:12, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=U.S._Cellular_Field&oldid=649856024
- Progressive Field. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:12, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Progressive_Field&oldid=653499953
- Comerica Park. (2015, March 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Comerica_Park&oldid=651832995
- Kauffman Stadium. (2015, March 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Kauffman_Stadium&oldid=650377832
- Target Field. (2015, February 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Target_Field&oldid=647933510
- Minute Maid Park. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Minute_Maid_Park&oldid=655303381
- Angel Stadium of Anaheim. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Angel_Stadium_of_Anaheim&oldid=654460841
- O.co Coliseum. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:13, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=O.co_Coliseum&oldid=653612043
- Safeco Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Safeco_Field&oldid=655148228
- Globe Life Park in Arlington. (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Globe_Life_Park_in_Arlington&oldid=652959003
- TD Garden. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=TD_Garden&oldid=653403579
- Barclays Center. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Barclays_Center&oldid=654631943
- Madison Square Garden. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:14, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Madison_Square_Garden&oldid=654828742
- Wells Fargo Center (Philadelphia). (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Wells_Fargo_Center_(Philadelphia)&oldid=649738895

- Air Canada Centre. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Air_Canada_Centre&oldid=655130412
- United Center. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=United_Center&oldid=655268263
- Quicken Loans Arena. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:15, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Quicken_Loans_Arena&oldid=655272955
- The Palace of Auburn Hills. (2015, February 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=The_Palace_of_Auburn_Hills&oldid=648272487
- Bankers Life Fieldhouse. (2014, December 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Bankers_Life_Fieldhouse&oldid=640160898
- BMO Harris Bradley Center. (2015, February 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=BMO_Harris_Bradley_Center&oldid=645646883
- Philips Arena. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Philips_Arena&oldid=654538478
- Time Warner Cable Arena. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Time_Warner_Cable_Arena&oldid=653373673
- American Airlines Arena. (2015, February 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:16, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=American_Airlines_Arena&oldid=646426983
- Amway Center. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Amway_Center&oldid=655268871
- Verizon Center. (2015, March 14). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Verizon_Center&oldid=651299892
- Pepsi Center. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Pepsi_Center&oldid=653560944
- Target Center. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:17, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Target_Center&oldid=655045980
- Chesapeake Energy Arena. (2015, March 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Chesapeake_Energy_Arena&oldid=650053351
- Moda Center. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Moda_Center&oldid=652902770

- EnergySolutions Arena. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=EnergySolutions_Arena&oldid=654209905
- Oracle Arena. (2015, February 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Oracle_Arena&oldid=648884284
- Staples Center. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:18, April 7, 2015, from http://en.wikipedia.org/w/index.php?title=Staples_Center&oldid=654943633
- Talking Stick Resort Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:19, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Talking_Stick_Resort_Arena&oldid=654788366
- Sleep Train Arena. (2015, February 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:19, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Sleep_Train_Arena&oldid=648040550
- AT&T Center. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:40, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Center&oldid=654184289
- Smoothie King Center. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:40, April 8, 2015,
  from http://en.wikipedia.org/w/index.php?title=Smoothie_King_Center&oldid=653411690
- FedExForum. (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:40, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=FedExForum&oldid=649655797
- Toyota Center. (2015, February 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:41, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Center&oldid=649249468
- American Airlines Center. (2015, March 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:41, April 8, 2015,
  from http://en.wikipedia.org/w/index.php?title=American_Airlines_Center&oldid=654000621
- Sleep Train Arena. (2015, February 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 15:19, April 7, 2015,
  from http://en.wikipedia.org/w/index.php?title=Sleep_Train_Arena&oldid=648040550
- Philips Arena. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Philips_Arena&oldid=654538478
- Allstate Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Allstate_Arena&oldid=654835689
- Mohegan Sun Arena. (2015, March 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015,
  from http://en.wikipedia.org/w/index.php?title=Mohegan_Sun_Arena&oldid=650179511
- Bankers Life Fieldhouse. (2014, December 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015,
  from http://en.wikipedia.org/w/index.php?title=Bankers_Life_Fieldhouse&oldid=640160898
- Madison Square Garden. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015,
  from http://en.wikipedia.org/w/index.php?title=Madison_Square_Garden&oldid=654828742
- Verizon Center. (2015, March 14). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Verizon_Center&oldid=651299892

- Staples Center. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:43, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Staples_Center&oldid=654943633
- Target Center. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Target_Center&oldid=655045980
- Talking Stick Resort Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Talking_Stick_Resort_Arena&oldid=654788 366
- AT&T Center. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Center&oldid=654184289
- KeyArena. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=KeyArena&oldid=653237504
- BOK Center. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=BOK_Center&oldid=654259841
- Santander Arena. (2015, February 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Santander_Arena&oldid=648485678
- Odeum Expo Center. (2015, February 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Odeum_Expo_Center&oldid=648219741
- The Gardens Ice House. (2014, December 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=The_Gardens_Ice_House&oldid=636303525
- Tiger Arena. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Tiger_Arena&oldid=653415784
- Dow Event Center. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Dow_Event_Center&oldid=654204684
- York City Ice Arena. (2014, October 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=York_City_Ice_Arena&oldid=629018943
- Talking Stick Resort Arena. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Talking_Stick_Resort_Arena&oldid=654788 366
- Honda Center. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Honda_Center&oldid=654822150
- Alamodome. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Alamodome&oldid=654207806
- Moda Center. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Moda_Center&oldid=652902770
- SAP Center at San Jose. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=SAP_Center_at_San_Jose&oldid=654649938
- Spokane Veterans Memorial Arena. (2015, March 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Spokane_Veterans_Memorial_Arena&oldid=654081587

- Jacksonville Veterans Memorial Arena. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Jacksonville_Veterans_Memorial_Arena&oldid=652941155

- CFE Arena. (2015, February 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=CFE_Arena&oldid=646752542

- Amalie Arena. (2015, February 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Amalie_Arena&oldid=646850271

- Quicken Loans Arena. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:44, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Quicken_Loans_Arena&oldid=655272955

- Smoothie King Center. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Smoothie_King_Center&oldid=653411690

- Wells Fargo Center (Philadelphia). (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Wells_Fargo_Center_(Philadelphia)&oldid=649738895

- Tim Hortons Field. (2015, March 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Tim_Hortons_Field&oldid=651817295

- Percival Molson Memorial Stadium. (2015, January 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Percival_Molson_Memorial_Stadium&oldid=642808624

- TD Place Stadium. (2015, February 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=TD_Place_Stadium&oldid=648816970

- Rogers Centre. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Rogers_Centre&oldid=654913445

- BC Place. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=BC_Place&oldid=653675532

- McMahon Stadium. (2015, February 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=McMahon_Stadium&ol Gillette Stadium. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Gillette_Stadium&oldid=654585813did=649207077

- Commonwealth Stadium (Edmonton). (2015, March 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Commonwealth_Stadium_(Edmonton)&oldid=649791012

- Mosaic Stadium at Taylor Field. (2015, February 15). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015,

fromhttp://en.wikipedia.org/w/index.php?title=Mosaic_Stadium_at_Taylor_Field&oldid=647219959

- Investors Group Field. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=Investors_Group_Field&oldid=655386188
- Ralph Wilson Stadium. (2015, April 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=Ralph_Wilson_Stadium&oldid=655091171
- Sun Life Stadium. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Sun_Life_Stadium&oldid=652918431
- MetLife Stadium. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=MetLife_Stadium&oldid=655501155
- M&T Bank Stadium. (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:45, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=M%26T_Bank_Stadium&oldid=638418974
- Paul Brown Stadium. (2014, December 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:46, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=Paul_Brown_Stadium&oldid=637052857
- FirstEnergy Stadium (Cleveland). (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:46, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=FirstEnergy_Stadium_(Cleveland)&oldid=654531854
- Heinz Field. (2015, March 25). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Heinz_Field&oldid=653482405
- NRG Stadium. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=NRG_Stadium&oldid=655407096
- Lucas Oil Stadium. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Lucas_Oil_Stadium&oldid=655376758
- EverBank Field. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=EverBank_Field&oldid=653161581
- LP Field. (2014, November 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:47, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=LP_Field&oldid=634894166
- Sports Authority Field at Mile High. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015,

  fromhttp://en.wikipedia.org/w/index.php?title=Sports_Authority_Field_at_Mile_High&oldid=653126660
- Arrowhead Stadium. (2015, March 29). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=Arrowhead_Stadium&oldid=654006833
- O.co Coliseum. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=O.co_Coliseum&oldid=653612043
- Qualcomm Stadium. (2015, March 27). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015,

  from http://en.wikipedia.org/w/index.php?title=Qualcomm_Stadium&oldid=653741585
- AT&T Stadium. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=AT%26T_Stadium&oldid=654731734

- MetLife Stadium. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=MetLife_Stadium&oldid=655501155
- Lincoln Financial Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Lincoln_Financial_Field&oldid=655256338
- FedExField. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=FedExField&oldid=654429456
- Soldier Field. (2015, March 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Soldier_Field&oldid=651010515
- Ford Field. (2015, March 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Ford_Field&oldid=653890809
- Lambeau Field. (2015, March 10). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Lambeau_Field&oldid=650772069
- TCF Bank Stadium. (2015, March 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=TCF_Bank_Stadium&oldid=649686714
- Georgia Dome. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Georgia_Dome&oldid=653308324
- Bank of America Stadium. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Bank_of_America_Stadium&oldid=653374205
- Mercedes-Benz Superdome. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:48, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Mercedes-Benz_Superdome&oldid=654152346
- Raymond James Stadium. (2015, February 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Raymond_James_Stadium&oldid=649202435
- University of Phoenix Stadium. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, fromhttp://en.wikipedia.org/w/index.php?title=University_of_Phoenix_Stadium&oldid=653559579
- Edward Jones Dome. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Edward_Jones_Dome&oldid=654167126
- Levi's Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=Levi%27s_Stadium&oldid=655210318
- CenturyLink Field. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 13:49, April 8, 2015, from http://en.wikipedia.org/w/index.php?title=CenturyLink_Field&oldid=652787982
- Toyota Park. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:30, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Park&oldid=654161546
- Mapfre Stadium. (2015, March 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Mapfre_Stadium&oldid=652025679
- Robert F. Kennedy Memorial Stadium. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015,

from http://en.wikipedia.org/w/index.php?title=Robert_F._Kennedy_Memorial_Stadium&oldid=652754664

- Saputo Stadium. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Saputo_Stadium&oldid=653113798
- Gillette Stadium. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Gillette_Stadium&oldid=654585813
- Yankee Stadium. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Yankee_Stadium&oldid=655513009
- Red Bull Arena (New Jersey). (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Red_Bull_Arena_(New_Jersey)&oldid=653149464
- Orlando Citrus Bowl Stadium. (2015, March 21). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Orlando_Citrus_Bowl_Stadium&oldid=652829636
- PPL Park. (2015, March 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=PPL_Park&oldid=651972956
- BMO Field. (2015, March 24). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BMO_Field&oldid=653369346
- Dick's Sporting Goods Park. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Dick%27s_Sporting_Goods_Park&oldid=655566313
- Toyota Stadium (Texas). (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Toyota_Stadium_(Texas)&oldid=638418651
- BBVA Compass Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BBVA_Compass_Stadium&oldid=655235636
- StubHub Center. (2015, March 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:31, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=StubHub_Center&oldid=651093792
- Providence Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Providence_Park&oldid=654711739
- Rio Tinto Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Rio_Tinto_Stadium&oldid=654304261
- Avaya Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Avaya_Stadium&oldid=655230355
- CenturyLink Field. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=CenturyLink_Field&oldid=652787982
- Sporting Park. (2015, March 13). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sporting_Park&oldid=651202621

- BC Place. (2015, March 26). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BC_Place&oldid=653675532
- Harvard Stadium. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Harvard_Stadium&oldid=654508027
- Village of Lisle-Benedictine University Sports Complex. (2014, September 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:32, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Village_of_Lisle-Benedictine_University_Sports_Complex&oldid=624371085
- BBVA Compass Stadium. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=BBVA_Compass_Stadium&oldid=655235636
- Providence Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Providence_Park&oldid=654711739
- Memorial Stadium (Seattle). (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Memorial_Stadium_(Seattle)&oldid=652979450
- Yurcak Field. (2014, October 11). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Yurcak_Field&oldid=629218625
- Maryland SoccerPlex. (2014, October 17). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Maryland_SoccerPlex&oldid=629989133
- Sahlen's Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sahlen%27s_Stadium&oldid=654403301
- Atlanta Silverbacks Park. (2014, November 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Atlanta_Silverbacks_Park&oldid=632159843
- WakeMed Soccer Park. (2015, February 18). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=WakeMed_Soccer_Park&oldid=647633122
- Clarke Stadium. (2015, April 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Clarke_Stadium&oldid=654678732
- Lockhart Stadium. (2014, November 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Lockhart_Stadium&oldid=632454890
- Baseball Grounds of Jacksonville. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Baseball_Grounds_of_Jacksonville&oldid=649550076
- IU Michael A. Carroll Track & Soccer Stadium. (2014, December 5). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:33, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=IU_Michael_A._Carroll_Track_%26_Soccer_Stadium&oldid=636779965

- National Sports Center. (2014, December 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=National_Sports_Center&oldid=636289847
- James M. Shuart Stadium. (2015, January 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=James_M._Shuart_Stadium&oldid=6406885
  11
- Toyota Field. (2015, April 1). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Toyota_Field&oldid=654448149
- Al Lang Stadium. (2015, March 20). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Al_Lang_Stadium&oldid=652776072
- Blackbaud Stadium. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Blackbaud_Stadium&oldid=653227162
- American Legion Memorial Stadium. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  fromhttp://en.wikipedia.org/w/index.php?title=American_Legion_Memorial_Stadium&oldid=654180309
- Skyline Sports Complex. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Skyline_Sports_Complex&oldid=654929230
- Louisville Slugger Field. (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Louisville_Slugger_Field&oldid=655428425
- Saputo Stadium. (2015, March 23). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Saputo_Stadium&oldid=653113798
- Highmark Stadium. (2015, April 4). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Highmark_Stadium&oldid=654950422
- City Stadium (Richmond). (2015, February 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=City_Stadium_(Richmond)&oldid=647801967
- Sahlen's Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Sahlen%27s_Stadium&oldid=654403301
- St. Louis Soccer Park. (2015, April 3). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:34, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=St._Louis_Soccer_Park&oldid=654795723
- Ontario Soccer Centre. (2015, March 2). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:35, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Ontario_Soccer_Centre&oldid=649486610
- Legion Stadium. (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:35, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Legion_Stadium&oldid=638361469
- Scottsdale Stadium. (2015, March 19). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015,
  from http://en.wikipedia.org/w/index.php?title=Scottsdale_Stadium&oldid=652123780

- House Park. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=House_Park&oldid=654152211
- Sand Creek Stadium. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Sand_Creek_Stadium&oldid=654238924
- StubHub Center. (2015, March 12). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=StubHub_Center&oldid=651093792
- Bishop McGuinness Catholic High School (Oklahoma). (2015, April 8). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, fromhttp://en.wikipedia.org/w/index.php?title=Bishop_McGuinness_Catholic_High_School_(Oklahoma)&oldid=655587468
- Anteater Stadium. (2014, December 16). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Anteater_Stadium&oldid=638362970
- Merlo Field. (2015, March 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Merlo_Field&oldid=654210299
- Rio Tinto Stadium. (2015, March 31). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Rio_Tinto_Stadium&oldid=654304261
- Bonney Field. (2015, April 6). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Bonney_Field&oldid=655241105
- Starfire Sports. (2015, March 22). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Starfire_Sports&oldid=653021031
- ONEOK Field. (2015, January 30). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:38, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=ONEOK_Field&oldid=644849140
- Thunderbird Stadium. (2015, April 7). In *Wikipedia, The Free Encyclopedia*. Retrieved 16:39, April 9, 2015, from http://en.wikipedia.org/w/index.php?title=Thunderbird_Stadium&oldid=655408047
- https://community.alteryx.com/t5/Alteryx-Knowledge-Base/Emailing-Different-Reports-to-Different-Recipients/ta-p/38517

Expensive Beauty Products

- Created by Author

Applications Wanted

- All Recorded Traffic Tickets:
o Data provided by the Baltimore Department of Finance on Open Baltimore (https://data.baltimorecity.gov/) Direct Link: https://data.baltimorecity.gov/Transportation/All-Time/ks7u-tsjz (accessed on

Where's the Joe?

- Mermaid Coffee Company: Data modified to exclude "Starbucks" from everywhere in the data.
o Corporation

Meta-Morphosis

- Created by Author

Let's Do It (In-DB)

- Data was obtained from Adventure Works by using a sample from the below dataset: https://msftdbprodsamples.codeplex.com/downloads/get/165405

Green on the Go

- Data for Supercharger Stations was downloaded from the below link using Alteryx tools: https://www.tesla.com/findus/list/superchargers/United+States
- https://gallery.alteryx.com/#!app/Twitter-Search/574db168a248970bdc36d527

Statistics in Alteryx

- https://rpubs.com/jbowmer/autoassessment1
- https://rstudio-pubs-static.s3.amazonaws.com/20858_82b28110c8ec44129848a691c4f02246.html

Alteryx for the Developer

- http://www.wunderground.com/history/airport/KMDW/2017/01/01/CustomHistory.html ?

Predictive Analytics with Alteryx

- Workflows were used from the below description, section 11 and 15 from the Alteryx help options in the tool. https://help.alteryx.com/9.5/SamplesByDescription.htm

Content:

The Alteryx Help documentation was used extensively in the research for this manual. Alteryx Help can be accessed through the Help Menu, Properties Window, or with the Hot Key F1 in Alteryx or can be accessed directly at http://help.alteryx.com/.

Authored and Published by

 USEReady

USEReady's mission is to help organizations succeed by fast tracking their business performance. We relentlessly strive towards value-driven innovation and digital transformation using advanced business intelligence, data management, and infrastructure management services. Our global delivery orientation is enriched with winning strategies, rich industrial and technology expertise and customer fanaticism. Our proven track record of success in versatile domains such as capital markets, insurance, healthcare, pharma, retail, media etc. is a testimony to our performance.

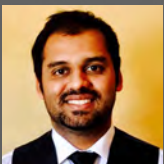|					
Abhilasha Shinde	Anirudh Kalva	Ashish Tergaonkar	Honey Shahi	Liyun Zhou	Lokesh Bansal
Lokesh Bhat	Madhusudhan G	Nishant Thakar	Olga Chirkova	Prateek Kale	Priya Raghuveer
Raghavendra Pai K	Rahul Shetty	Rahul Upadhye	Sachet Kashyap	Shamalee Thakur	Smruti Dash
Subrat Das	Suman Joshi	Tanvi Shah	Tharun Kumar	Vijay Gowtham	Vishnudas P N

www.useready.com
contact@useready.com