

A PRACTITONER'S GUIDE TO

ALTERYX

BEGINNER
SERIES

VERSION 2019.2

A Practitioner's Guide to Alteryx

A Practitioner's Guide to Alteryx

Published by
USEReady

1407 Broadway, Suite 721, New York, NY 10018

www.useready.com

Copyright © 2020 by USEReady New York
Manufactured in the United States of America

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Sections 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher. Requests to the publisher for permission should be addressed to USEReady, 1407 Broadway, Suite 721, New York, NY 10018, or online at www.useready.com.

Limit of Liability / Disclaimer of Warranty: The publisher and the author make no representations or warranties with respect to the accuracy or completeness of the contents of this work and specifically disclaim all warranties, including, without limitation, warranties of fitness for a specific purpose. No warranty may be created or extended by sales or promotional materials. The advice and strategies provided herein might not be suitable for every situation. This work is sold with the understanding that the publisher is not engaged in rendering legal, accounting professional services. Neither the publisher nor the author is liable for damages arising as a result of information published in this book. The fact that an organization or a website is referred to in this book as a citation or potential source of further

information does not mean that the author or the publisher endorses the information the organization or website may provide or recommendations it may make. Further, readers should be aware that Internet sites listed in this work might have changed or disappeared between when this book was written and when it is read. The images provided in this book is for general informational purposes only.

Table of Contents

About USEReady.....	9
About Alteryx, Inc.....	10
Acknowledgments.....	11
Letter from the President.....	12
Letter from the CEO.....	15
Foreword	17
Preface.....	19
To the Reader	24
CHAPTER 1	25
An Introduction to Alteryx.....	25
1.1 <i>What Is the Alteryx Analytics Platform?.....</i>	26
1.2 <i>Alteryx Gallery.....</i>	26
1.3 <i>The Alteryx Interface.....</i>	27
1.4 <i>The Interface.....</i>	29
1.5 <i>File Menu</i>	30
1.6 <i>Edit Menu.....</i>	31
1.7 <i>View Menu.....</i>	32
1.8 <i>Options Menu</i>	32
1.9 <i>Help Menu</i>	33
1.10 <i>Samples.....</i>	34
1.11 <i>User Settings.....</i>	34
1.12 <i>Tool Palette.....</i>	35
1.13 <i>Overview</i>	37
1.14 <i>Results Window.....</i>	37
1.15 <i>Properties Window.....</i>	39
1.16 <i>Interface Designer</i>	45
1.17 <i>Canvas.....</i>	48
1.18 <i>Using Tools.....</i>	49
1.19 <i>Insert Tool Menu</i>	50

CHAPTER 2	51
The Games	51
2.1 <i>Tools and Concepts</i>	52
2.1.1 Input Data.....	52
2.1.2 Browse.....	52
2.1.3 Output Data.....	52
2.1.4 Comment.....	53
2.1.5 Filter.....	53
2.1.6 Formula.....	53
2.1.7 Running Total.....	54
2.1.8 Sample	54
2.1.9 Select.....	55
2.1.10 Sort.....	55
2.1.11 Summarize	55
2.1.12 Tool Container	56
2.1.13 Transpose.....	56
2.1.14 Union.....	56
2.1.15 Cross Tab	57
2.2 <i>Freestyle</i>	58
2.3 <i>Let's Tidy Things Up</i>	73
2.4 <i>Modern History</i>	82
2.5 <i>Brains vs. Brawn</i>	92
2.6 <i>How Are We Doing?</i>	109
CHAPTER 3	110
Unisex Baby Name.....	110
3.1 <i>Tools and Concepts</i>	111
3.1.1 Imputation.....	111
3.1.2 Multi-Field Formula	111
3.1.3 Multi-Row Formula	111
3.1.4 Text Input	112
3.1.5 Data Cleansing.....	112
3.2 <i>Gender Swapped</i>	113
3.3 <i>What about me?</i>	129

CHAPTER 4	135
The Direct Approach.....	135
4.1 <i>Tools and Concepts.....</i>	136
4.1.1 Append Fields.....	136
4.1.2 Auto Field.....	136
4.1.3 Date Time Now	136
4.1.4 Date Time.....	137
4.1.5 Directory.....	137
4.1.6 Visual Layout	137
4.2 <i>What's the Policy on That?.....</i>	138
4.3 <i>Where, Oh Where Have the Three Files Gone?.....</i>	150
CHAPTER 5	151
Cultural Musing.....	151
5.1 <i>Tools and Concepts.....</i>	152
5.1.1 Allocate Input.....	152
5.1.2 Find and Replace.....	152
5.1.3 Join Multiple	152
5.1.4 Text to Columns.....	153
5.1.5 Random % Sample.....	153
5.1.6 XML Parse.....	153
5.2 <i>Culturally Rich.....</i>	154
5.3 <i>Culturally Divided.....</i>	169
CHAPTER 6	170
Expensive Beauty Product	170
6.1 <i>Tools and Concepts.....</i>	171
6.1.1 Fuzzy Match.....	171
6.1.2 Record ID.....	171
6.1.3 RegEx.....	171
6.1.4 Unique.....	172
6.2 <i>Expensive Beauty Products.....</i>	173
6.3 <i>More Flags.....</i>	196
CHAPTER 7	197

Let's Do It (In-DB)	197
7.1 <i>Tools and Concepts</i>	199
7.1.1 Browse In-DB.....	199
7.1.2 Connect In-DB.....	199
7.1.3 Data Stream In	200
7.1.4 Data Stream Out	200
7.1.5 Filter In-DB	201
7.1.6 Formula In-DB	201
7.1.7 Join In-DB	201
7.1.8 Macro Input In-DB	202
7.1.9 Macro Output In-DB	202
7.1.10 Select In-DB	203
7.1.11 Summarize In-DB.....	203
7.1.12 Union In-DB	204
7.1.13 Write Data In-DB.....	204
7.1.14 Sample In-DB.....	204
7.1.15 Calgary Loader.....	205
7.1.16 Calgary Input.....	205
7.1.17 Calgary Join.....	205
7.1.18 Calgary Cross Count	206
7.1.19 Calgary Cross Count Append.....	206
7.2 <i>Superstore Database Report</i>	207
7.3 <i>Superstore Database Report with Macro</i>	214
CHAPTER 8	221
Alteryx for Good	221
8.1 <i>What Is Alteryx for Good?</i>	222
8.2 <i>Alteryx for Good in India</i>	222
8.3 <i>What Activities Does the AFG India Chapter Do?</i>	224
8.3.1 Report Card Solution	225
8.3.2 Certificates Solution.....	227
8.3.3 Donation Receipts Solution	229
8.3.4 ID Card Solution.....	230
8.3.5 Health Card Tracking Solution	231

8.3.6. Excel Training and Digitization	232
8.4 How Can You Join or Reach Alteryx for Good in India?.....	233
8.5 How Can You Join Alteryx for Good in Your Area?.....	234
CHAPTER 9	235
Self-Guided Solutions	235
9.1 The Games: How Are We Doing?	236
9.2 Unisex Baby Names: What's in a Name?.....	237
9.3 The Direct Approach: Where, Oh Where Have My Three Files Gone?	238
9.4 Cultural Musing: Culturally Divided	239
9.5 Expensive Beauty Products: More Flags.....	239
Appendices	242
Appendix A – File Types.....	243
Appendix B – Hot Keys	244
Appendix C – Downloads/Content.....	245
Appendix D – Field Types	246
Appendix E – Boolean Expressions.....	248
Appendix F – Data Components.....	249
Appendix G – RegEx Cheat Sheet	250
Appendix H – Directory Tool Data	251
Credited Original Data Sources	252

About USEReady

USEReady's mission is to help users succeed with data.

We achieve this mission with fanatical customer centricity, humility, and integrity. At USEReady, we love to solve customer problems, contribute to the community, and continue to improve. We aim to build a strong learning culture and have fun doing it together.

We have harnessed modern Business Intelligence solutions with cutting edge Artificial Intelligence (AI) platforms to unlock the power of data. Our result-oriented solutions ensure that our customers are successful in adopting self-service technologies.

We have partnered with the best in class BI, Data and Cloud products such as Tableau, Snowflake, Alteryx, Informatica, Exasol, AWS, and Azure. We have delivered proven success across Financial Services, Insurance, Retail, and Media verticals.

Check us out at www.useready.com or e-mail contact@useready.com.

About Alteryx, Inc.

As a leader in analytic process automation (APA), Alteryx unifies analytics, data science and business process automation in one, end-to-end platform to accelerate digital transformation.

Organizations of all sizes, all over the world, rely on the Alteryx Analytic Process Automation Platform to deliver high-impact business outcomes and the rapid upskilling of their modern workforce. For more information visit www.alteryx.com.

Alteryx is a registered trademark of Alteryx, Inc

Acknowledgments

When we started writing this edition of the book, we decided to use a team of authors instead of a single author. The team went through a very challenging process of reading, updating, and reviewing to get the book where it is.

I would like to express my gratitude to the team of authors and other colleagues who helped make this book a reality. Without them, the book truly would not be what it is today.

I would like to thank the following members of the team:

Uday Hegde: For being the driving force and a constant motivator.

Anand Vadul: For sharing the experiences and challenges faced in the last edition and being the technical guiding force.

Prashant Singh: For helping with content collation, editing, marketing, and an excellent cover design.

Alteryx for Good, Bangalore Team: For sharing their journey and their drive to serve charities using Alteryx.

- Deeksha Premchand Rao,
Technical Lead
USEReady

Letter from the President



USERReady is more than happy to have successfully published this latest edition of, A Practitioner’s Guide To Alteryx Beginner Series Version 2019.2 Over the years, Alteryx has become a leader when it comes to analytic process automation – successfully unifying multiple processes under one platform in a bid to help organizations transform at an accelerated pace. Today, we’re extremely happy to call them partners in our data journey.

This book has proven to be a huge success – seen by experts as a must-have guide for users setting out on using Alteryx, this book focuses on assisting users to be self-reliant and improve their skills, as they achieve a lot more from their Alteryx experience.

Multiple Alteryx practitioners at, USERReady have contributed towards piecing this book and making it what it is. The entire team consisting of Deeksha Premchand Rao, Sachet Kashyap, Abhilash Ramanathan, Harshita Paul, Priyanka Dobhal, and Meehir Mene who have strived to give us a book that is extremely refined and up to date with the most recent version of Alteryx, thank you. I would also like to thank Laxmi Shankar and Arun Prabhakara who was extremely influential in helping this project take shape and form - with a special mention to Subrat Das, Murtaza Farooqui, and Sharang Kulkarni.

At USERReady, we pay very close attention to the reaction and responses we receive from our readers. As per requests received

regarding USEReady's other guides published - we have further included additional hands-on exercises, upgraded product features, and new chapters for users learn from.

Feel free to drop by www.community.useready.com where you can explore extremely useful exercises that are beneficial to fellow Alteryx practitioners.

Apart from keeping our ears pressed to the ground when it comes to listening to our readers – we are also highly aware of the environment that we operate within and the ecosystem we are a part of. Right now, this means functioning in a world that is going through a pandemic.

Unemployment is on the rise, people who need to upgrade their skills will see an immense benefit when it comes to using this book. Today, we are also witnessing a steady increase in the number of people working from home - creating more free time that people can put to good use for developing and upgrading skillsets.

Learning and innovation must and will never stop, we must find new and better ways to move ahead as we understand how to pivot and readjust our thought process.

With that being said, we have decided to make this book completely free for everyone moving forward.

It took a lot of people and a collective effort in preparing this book - we are very grateful to have Alteryx part of the process and having put in the effort to help us review this book on priority and extremely quickly. They've helped push this book into the market for our readers, in a much shorter time than earlier anticipated.

I would like to personally thank, Alan Jacobson, Chief Data & Analytics Officer, Scott Jones - President & Chief Revenue Officer and Robbin Jang, Partner Manager, Alteryx without whose help this project would not have been possible.

Special mention towards Gretchen Roberts from Global Content Marketing and Tatiana Servin, Content Marketing at Alteryx for helping us tie up any loose ends and piece this book together, making it the finished and perfected guide that it is today. I would also like to thank Anabel Soto and Matt Wulkan from Alteryx Marketing for helping us through the initial stages of development to the final promotional process.

It's a massive collective effort, I'd like to personally thank everyone and anyone that I've failed to mention. Our long-standing relationship and ability to work together as one big data family has yet again produced a highly valuable and successful outcome.

Happy Reading Folks!

Lalit Bakshi
President,
USEReady

Letter from the CEO



It's my pleasure to introduce our readers to the latest and up to date publication on, Alteryx.

A public company, Alteryx has gained widespread adoption across industries and geographies - we are thrilled to have partnered with this successful product that's loved by all within the data community.

We are witnessing a steady increase in adoption through the orders that we receive for this book. This version of Alteryx incorporates all the feedback and suggestions received we have received from our readers. Learn through practicality - as suggested by our previous readers, this book further includes additional hands-on exercises based on real-world use cases that users can practice and learn from.

Certified Alteryx practitioners at, USEReady have contributed and authored this book. Their hard work and dedication over time, has resulted in producing this, and multiple other books that are loved by the data learning community all over.

Revised and perfected - this book consists of upgraded product features, exercises, and new chapters that are dedicated towards helping users understand Alteryx in a comprehensive manner. With content that is extremely refined, straightforward, and easy to understand - fellow practitioners will further find each and every exercise in this book to be extremely useful.

We are grateful to our readers of the previous editions - their constructive feedback has helped us improve and better the learnings published within this edition. We hope our efforts are well worth it and each one of you finds this book to be helpful.

Uday Hegde
Chief Executive Officer,
USEReady

Foreword



Welcome to an incredible journey into the world of analytics! While people around the world have been using Alteryx for over 20 years, we are excited each time a new user joins our data science and analytics community and the legion of digital transformation experts.

In this introductory book, USEReady quickly teaches you the core functionality of Alteryx including geospatial analysis and our rich data packages. It features real-world examples and solutions so relatable it's as if the very same assignment just came through your inbox.

Need your marketing program optimized by tomorrow morning? No problem for those that have been through this course. Need to re-engineer your financial consolidation processes with full automation on a timed schedule? You may be on vacation the next time it needs to run, but that's not a problem now because you've automated the entire process last night.

By the time you finish working through this manual, as someone who has never opened the Alteryx Designer before, you will be able to create powerful workflows with data, design engaging reports, develop applications, and write macros to solve any data need. USEReady has leveraged extensive experience that enables diverse users at large enterprises to make powerful deployments of analytic tools.

I invite you to supplement this guide with the resources provided by Alteryx in the Alteryx Community (community.alteryx.com). In the Alteryx Community, you will find a wide array of free resources that include the ability to get help on questions, weekly challenges to continue exercising your new skills, certification courses, podcasts, local user groups, and more.

This is an opportune time for people anywhere to make the best out of the unprecedented demand for data analytics with actionable information. Those who leverage these materials will be well on their way to accelerating their data know-how with Alteryx Designer.

We appreciate the partnership with companies like USEReady that know how to help analysts reduce the time to insight with Alteryx. We are thrilled to welcome you to our community and hope that we can help you accelerate your journey with this guide. If there is anything we can do to help, please reach out.

Alan Jacobson
Chief Data and Analytics Officer
Alteryx, Inc.

Preface

We are faced with options, questions, and choices every day. Our decisions, as we all know, are much easier to make when we're well informed. Let's say 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 \$5,000? What are we in the mood for?

This task that we each solve day in and day out depends on a considerable amount of information we know 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 – 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. Data will be our focus.

What Does Data Do?

Data does nothing. It simply exists. It's 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 the raw data and read every piece individually. We can use aggregation methods to create summary data so we can easily see high-level trends. We can visualize data. Since we often do not want to look directly at the original data source and reach each individual piece of data, we need to perform data preparation.

Introduction to Data Preparation

Data preparation is the process by which raw data is converted into a clean, usable source for later consumption. It is a process of standardizing the practice of data consumption, which helps create rich and insightful analysis.

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're going to avoid these technical terms and discuss the aspects of the processes as follows:

1. Data retrieval refers to the process of going to a data source, asking for data, and returning the desired data.
2. Data manipulation refers to anything we decide to do to the data between the time we retrieve it and the time we export it.
3. 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 straightforward. 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, filling the ice cube tray, putting the tray in the freezer, and letting the water have enough time to freeze so we have ice to consume. Going forward, we will use an analogy of a river to explain the entire process of data preparation and specifically data manipulation.

Data manipulation can come in many forms, which typically fall into three buckets: *combination*, *calculations*, and *transformation*.

Combination

One of the most common problems with data is that it comes from multiple sources. It is generally possible to perform an 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 unite these different water sources to form a river. Along the course of this river, we can perform calculations.

Calculations

If the data is to be used, it is generally advisable to have as much data pre-calculated as possible. One reason is because it allows an organization to create a standardized formula for everyone's

use. Another reason is because 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 existing resources in order to generate something new.

Because we're changing the flow of the water, we might be changing the landscape and 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 usable format.

Transformation

It is often the case that data is not in the format we need. We may have been given access to a database with 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 that has data extremely normalized, making it hard to use in a front-end system. Either way, we need to transform the structure of the data so it can be effectively consumed. Back to the river analogy, we can imagine this as the process of cutting a channel into the riverbed so the river is deep enough to move barges up and down. In doing so, we're fundamentally changing the structure of the river in order to make it more usable.

The Alteryx Analytics Platform was built with the above points in mind. It helps the user automate the manual data task using a repeatable analytic workflow approach, which is code-free and

helps the user focus on gaining insights from data and not spend time cleaning and preparing the data. It 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, it requires skilled individuals, and it is time consuming. Alteryx, Inc. has elegantly addressed both problems. It has designed an easy-to-use, drag-and-drop interface where all you need to do is give each tool the appropriate settings. This simple concept has allowed business users to develop workflows to meet their needs much faster than has ever been possible. It provides them with the tools to design or modify a fully or semi-automated data preparation process.

Historically, much of this task has been done with Excel spreadsheets. Companies have employed armies of analysts to help prepare and clean data, which, in turn, meant high cost of headcount and low employee morale, and that affected productivity. The introduction of Alteryx into such ecosystems has generally transformed the mindset of its workforce. Individuals now feel empowered to glean insights from the data at the speed of thought, and it has brought back the joy of learning and exploration to its user that has improved overall productivity and insights gained. Companies have seen tangible returns on investment in every scenario where the Alteryx Analytics Platform has been used.

To the Reader

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

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

The exercises will use data that can be downloaded from community.useready.com/publications by following the instructions on the website to unpack the file.

The diagrams in this book are not an accurate representation of the latest version of Alteryx. Additional data will also be needed to install the *U.S. 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 data sets in the *Statistics in Alteryx* section.

Let us know what you think by e-mailing us at publications@useready.com. We will try to incorporate reader requests going forward.

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

Best regards,
USEREADY

CHAPTER 1

An Introduction to Alteryx

1.1 What Is the Alteryx Analytics Platform?

The Alteryx Analytics Platform from Alteryx, Inc. consists of four products: Alteryx Designer, Alteryx Server, Alteryx Connect, and Alteryx Promote.

Alteryx Designer allows us to build a data stream (or module) and run it locally. In addition to a base package, Alteryx offers data sets and tool packages to expand the product to enhance computing capabilities. These packages permit additional features such as 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.

Alteryx Connect helps locate assets that can be used for data preparation. It provides a catalogue and metadata of all the assets, which are built collaboratively and in compliance with IT. *Alteryx Promote* helps deploy, manage, and monitor predictive and machine learning models in production. That is enabled by embedding Alteryx, R, and python models in production applications and referencing them through APIs.

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 this chapter, the reader should be able to understand all aspects of the Alteryx Designer Interface and how to work with the product.

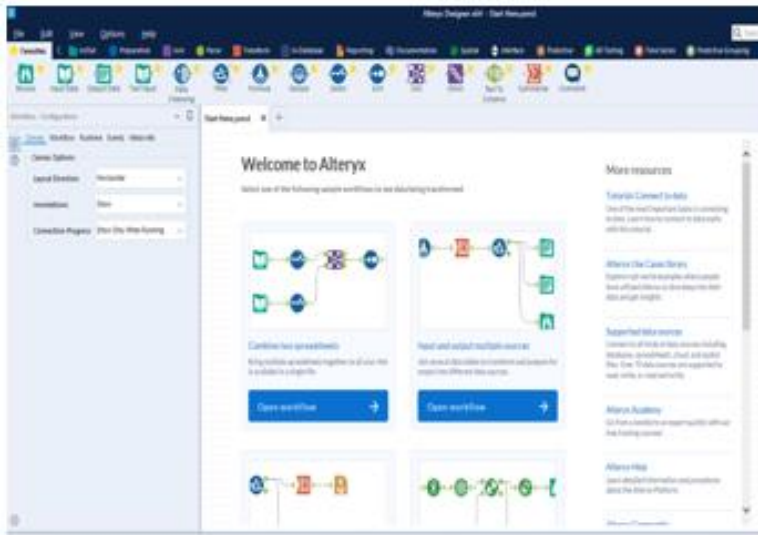


Figure-1-1-Alteryx Interface

Above is what we might see when we open Alteryx Designer. Alteryx Designer asks us what we want to do with the session. It is fairly clear that these options are to open up pre-built sample workflows to help the user understand how data is transformed in Alteryx, to open an existing module, or to open a new module.

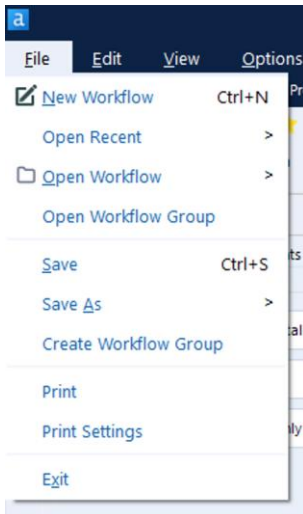


Figure-1-2-File Dropdown

The first thing we will do is create our module and save it. Open the file dropdown at the top left of the screen, and select the *New Workflow* option. Then click on the file dropdown again and click the *Save As* option.

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

- *.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 reuse of the data stream.

Please save this workflow as *The Basics.yxmd*.

You can also save a workflow to the Alteryx Gallery, which is hosted either on premise or on the cloud. While saving a workflow to the Alteryx Server Private Gallery, you can assign

the workflow to run on a specific worker if you have been given worker assignment permissions.

You can save multiple workflows as a workflow group, which can then be opened as one workflow group file (.yxwg).

You can save all open workflows at the same time by pressing Ctrl + Shift + S.

1.4 The Interface

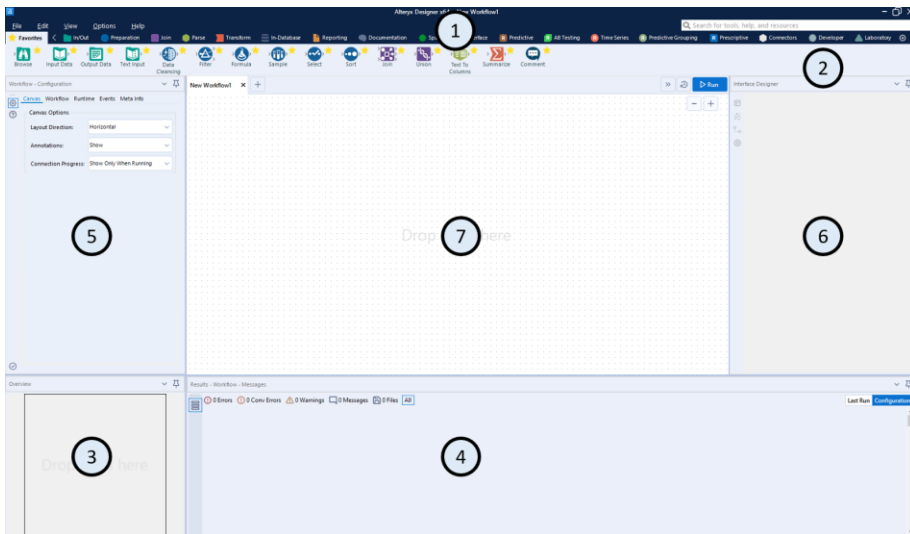


Figure-1-3-Alteryx Designer Interface

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 workflow

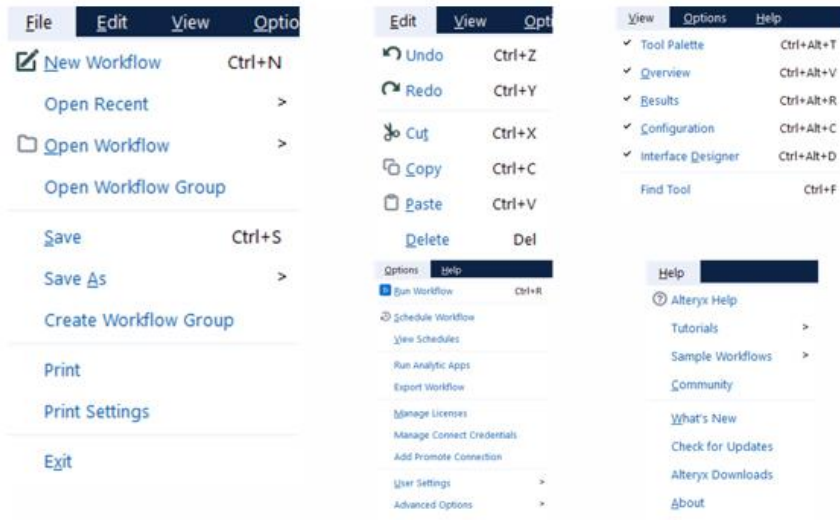


Figure-1-4-File/Edit/Options/Help Menu

1.5 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 there is a *Hot Key* that allows us to use this action without opening the dropdown. 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: By default, Alteryx saves your open workflows every 10 minutes, maintains the last three iterations of them, and keeps them for 30 days since the last save. To edit the Autosave settings, go to the 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 a copy.

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

1.6 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

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).

The close (X) button was removed from the Overview, Results, Configuration, and Interface Designer windows. You can still choose which windows to display using the View menu.

1.8 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, if we have Alteryx Server.

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

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

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

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

Manage Connect Credentials: Opens a window that allows us to sign in to our Alteryx Connect account using a URL for the instance we want to connect to.

Add Promote Connection: Opens a window that allows us to sign in to our Alteryx Promote account using a URL for the instance we want to connect to.

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 encrypted workflows.

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

1.9 Help Menu

Alteryx Help: Opens the Alteryx Help home page.

What's New: Opens the Alteryx Help release notes for the version of Alteryx 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. Fourteen new tutorials are added in the current version, available from the Help > Tutorials menu.

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 additional 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

Under the *Help Menu*, there is the option to open Tutorials. The sample workflows under the tutorials section is a good starting point to learn about the Alteryx tools 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.

1.11 User Settings

Edit User Settings: Opens a pop-up window so the user can edit the settings that affect 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.

Save Layout & Settings Now: Allows us to tell Alteryx to use the settings we currently have as the default behavior for the program.

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

1.12 Tool Palette



Figure-1-5-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 section below called *Canvas*, we will discuss the alternative method.

Below are the three main elements of the tool palette:

- *Search*: Located in the top-right corner of the tool palette, this element allows us to type something into the bar and see every tool associated with that word 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, or 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 is a dropdown when we type in the search box and shows all tools meeting either the criteria of the *Search* or *Tool Category* we have selected.

In the *Tool Categories*, there is one special category 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 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 it isn't in the *Favorites Category*.

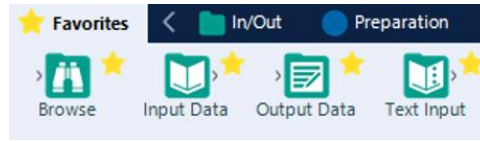


Figure-1-6-Favorites

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's 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>*. To unpin a category, right-click the category name and select *Unpin <Category Name>*.

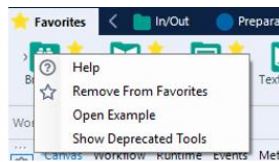


Figure-1-7-Tool Categories

We can see in Figure 1-7 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-1-8- Additional Tool Categories

All these categories may not be available to all users because Alteryx has many additional downloadable packages in addition to the basic package. One example is that if we hadn't chosen to download the Predictive package when we downloaded Alteryx Designer, we wouldn't see the *Predictive Category*. We will only

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

1.13 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.

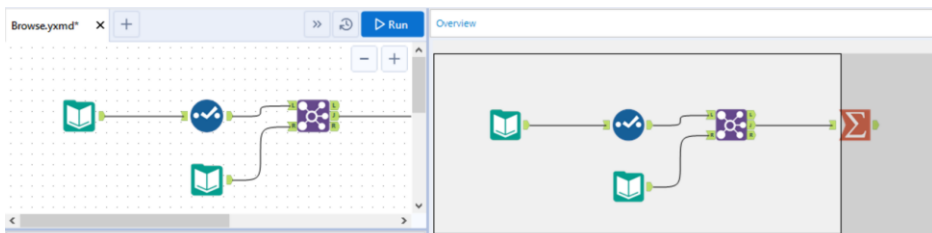


Figure-1-9-Overview Window

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

1.14 Results Window

Every time we run a workflow, we will get a series of returns that let us know things about the run. Below is one example of that.

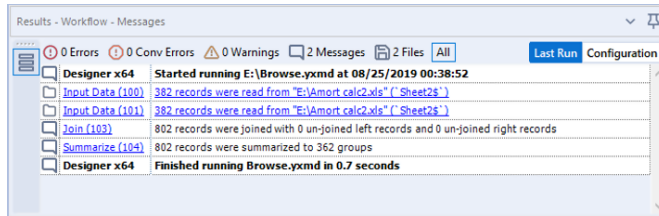


Figure-1-10-Results Window

In this run, we can see we had 0 Errors, 0 Conversion Errors, 0 Warnings, 2 Messages, and 2 Files. The counts here reflect the 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. That is because the first note is that the workflow started to run, and the last note is that the module has 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 stemming 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 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.

1.15 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 these symbols is in Appendix E.

Workflow Configuration

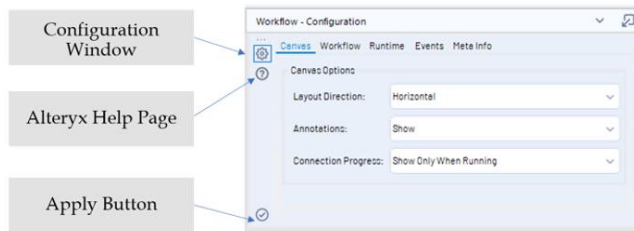


Figure-1-11-Workflow Configuration – Canvas Tab

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 we can modify the settings of the tool. As we discuss other tools, the importance here will become apparent.

A click on the symbol opens the Alteryx help page.

A symbol at the bottom-left corner is an apply button. We can click it to apply the changes we have made. We do not need to click the button; we can simply click anywhere 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 dropdown 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 dropdown 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 dropdown 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.

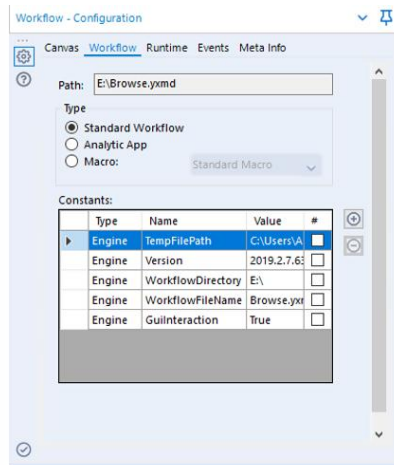


Figure-1-12-Workflow Configuration-Workflow Tab

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 we can use throughout our Alteryx Module.

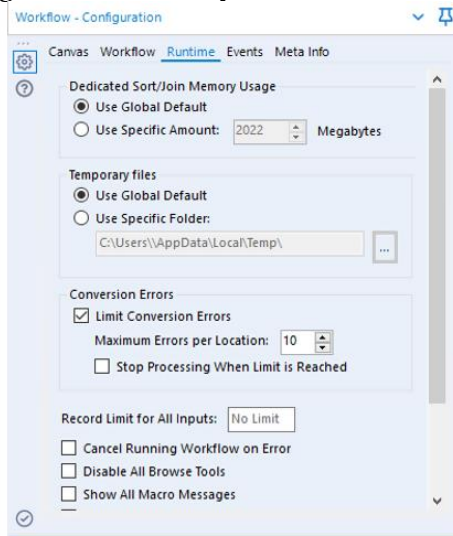


Figure-1-13-Workflow Configuration – Runtime Tab

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

- The *Dedicated Sort/Join Memory Usage* allows us to set the maximum amount of memory 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 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's better to use this than to delete all the tools because it will make modifying the workflow later easier.
- *Show All Macro Messages* allows us to see the messages we're getting in the macro tools so 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.

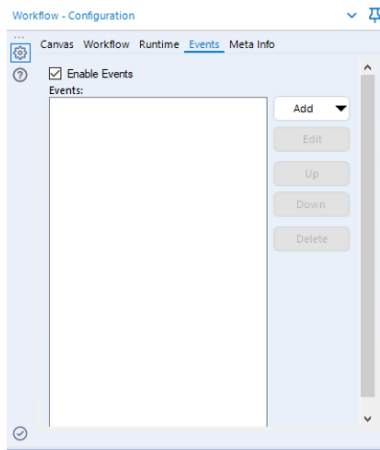


Figure-1-14-Workflow Configuration – Events Tab

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 e-mail when a condition is met. That 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. That is important when we are publishing a workflow to Alteryx Server or the public Alteryx Gallery.

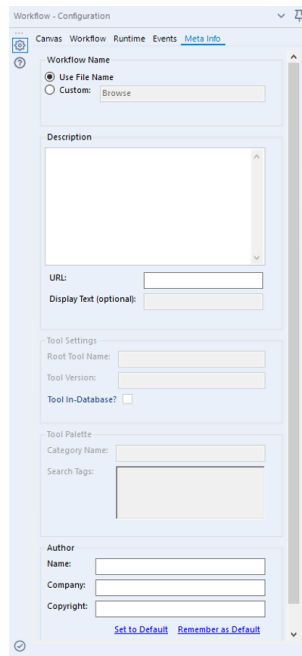


Figure-1-15-Workflow Configuration – Meta Info Tab

- *Workflow Name* allows us to customize what we want the workflow to be called (independent of the file name) so 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.

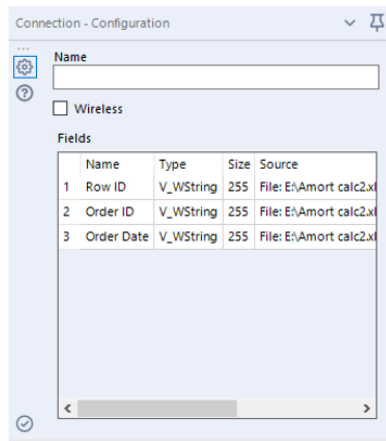


Figure-1-16-Connection Configuration

The *Connection Configuration window* has the following three components:

- *Name* allows us to rename our data connection. That 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.

1.16 Interface Designer

The Interface Designer is a window used in developing apps and macros. That is where we get to design the user interface the end user will see.

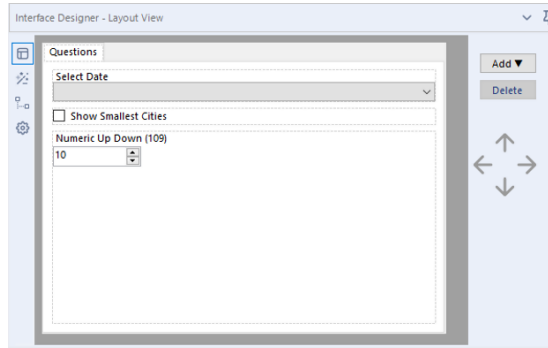


Figure-1-17-Interface Designer - Layout View

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

- *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 (independent of the Properties window itself).

In the *Layout View*, we design the interface and 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 by the end user. We can do this

by clicking the up and down arrows on the right 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. That 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.

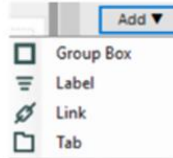


Figure-1-18-Interface Designer Dropdown

The next thing we can do is select the Add dropdown menu and see that we can add any of the following four things:

- *Group Box*: a text element we can put questions in
- *Label*: a text element 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. Following is an example of the properties window for the Label option described above.

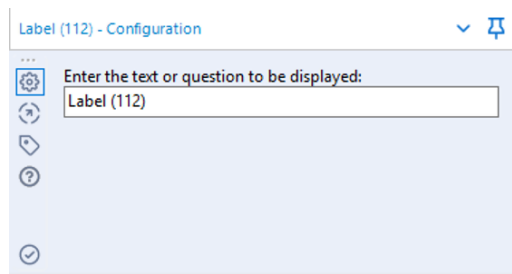


Figure-1-19-Label Configuration

All windows 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 new icons in the sidebar of the Properties window.



Figure-1-20-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.

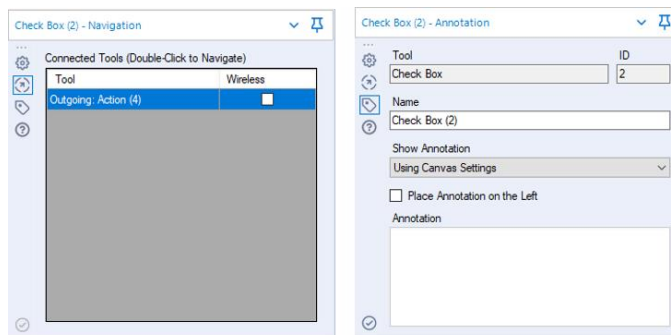


Figure-1-21-Check Box Navigation and Annotation

We can see that *Navigation* lists the tool this *Check Box* is connected to and tells us it's an outgoing connection. That means 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 the chapters called *Applications Wanted* and *Metamorphosis*.

The Annotation window tells us the tool is a Check Box with the ID number 2, which means it will try to execute this tool second if there are no 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).

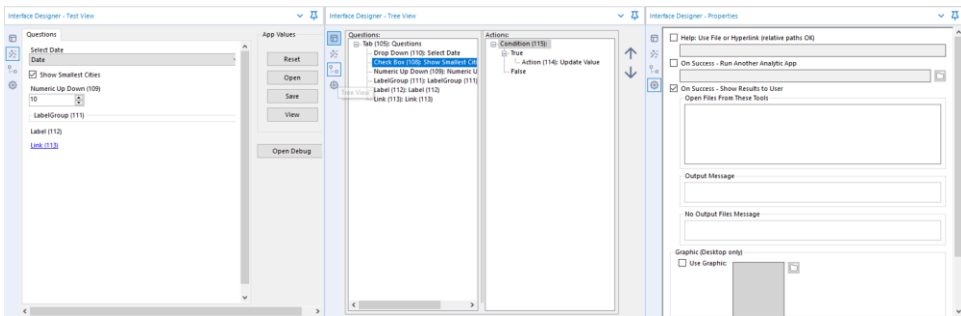


Figure-1-22-Test View / Tree View / Interface Designer Properties

Looking back at the Interface Designer, *Test View* allows us to answer the questions in the same way the end user would so we can test the app.

TreeView allows us to look at our user interface as a tree structure so it's easier to make sure our logic is properly grouped. *Interface Designer Properties* allows us to customize many aspects of our app or macro.

1.17 Canvas

The default canvas view of the Alteryx Designer has five elements: (1) tab name, (2) dropdown list of all the tabs, (3) button to allow the user to instantaneously add workflow to the

scheduler on the server, (4) button to run the workflow, and (5) white area that reads “Drop tools here.” The overview, output, interface designer, and properties windows allow us to move them around the screen by clicking and dragging their title bar. When we do this, we can snap them into the window by dropping them on one of the arrows you will see with drag-and-drop, or you can let it float in front of the canvas or other windows by letting go of it while not over an arrow.

1.18 Using Tools

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

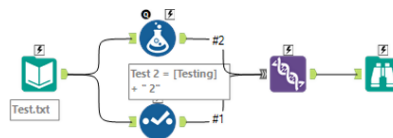


Figure-1-23-Data Stream

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.

We can see there is no parallel tool on the outgoing (right) side of the tools, despite the data stream having multiple outgoing

connections. That is because every tool allows us to branch the data stream off into many directions.

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

1.19 Insert Tool Menu

As an alternative to using the tool palette to insert new tools in the canvas, you can right-click on the canvas to add tools using the Insert Tool Menu. Simply navigate that menu to find the tool you're looking for.


CHAPTER 2

The Games

2.1 Tools and Concepts


2.1.1 Input Data

Concept: Input Data

 Figure 2-1-Input Data	Tool Palette: In/Out
	Imports data from different data sources. For more details, use the link below. bit.ly/31Lu6M3


2.1.2 Browse

Concept: Viewing Data

 Figure-2-2-Browse	Tool Palette: In/Out
	Gives a tabular view of the data in a data stream at the point it is connected. For more details, use the link below. bit.ly/2NfUxW4

2.1.3 Output Data

Concept: Outputting Data

 Figure 2-3-Output Data	Tool Palette: In/Out
	Writes the data stream out to a file or database. For more details, use the link below. bit.ly/31GD7WG


2.1.4 Comment

Concept: Adding Notes

 Figure-2-4-Comment	Tool Palette: Documentation
	<p>Gives us the ability to write notes on our workflows to add additional information on the data stream. For more details, use the link below. bit.ly/2TLIRvG</p>


2.1.5 Filter

Concept: Splitting Data

 Figure-2-5-Filter	Tool Palette: Preparation
	<p>Gives the ability to create a function that will split the data row by row into either the true or false outputs. For more details, use the link below. bit.ly/2KK64vk</p>


2.1.6 Formula

Concept: Creating Calculations

 Figure-2-6-Formula	Tool Palette: Preparation
	<p>Gives the ability to create a function that will update an existing column or will be written to a new column in our data. For more details, use the link below. bit.ly/30fFyPy</p>


2.1.7 Join

Concept: Combining Data

 <p>Figure-2-7-Join</p>	<p>Tool Palette: Join</p>
	<p>Combines two data streams by lining up records based on matching fields. For more details, use the link below. bit.ly/2TMLcGI</p>


2.1.7 Running Total

Concept: Cumulative sum

 <p>Figure-2-8-Running Total</p>	<p>Tool Palette: Transform</p>
	<p>Create a running sum for a numeric field in the incoming data stream. For more details, use the link below. bit.ly/2KGrS3</p>


2.1.8 Sample

Concept: Creating Data Subsets

 <p>Figure-2-9-Sample</p>	<p>Tool Palette: Preparation</p>
	<p>Allows us to create and work with a subset of data. For more details, use the link below. bit.ly/2lXoU81</p>


2.1.9 Select

Concept: Identifying Desired Results

 Figure-2-10-Select	Tool Palette: Preparation
	<p>Helps to include and exclude fields in the data stream and also modify the metadata associated with the data stream, including the order of fields. For more details, use the link below. bit.ly/2zaqF57</p>


2.1.10 Sort

Concept: Organizing Data

 Figure-2-11-Sort	Tool Palette: Preparation
	<p>Arranges the records of a table in alphabetical or numerical order of the specified fields. For more details, use the link below. bit.ly/2ZfKzdJ</p>


2.1.11 Summarize

Concept: Summarizing Data

 Figure-2-12-Summarize	Tool Palette: Transform
	<p>Helps to perform group operations on the data. For more details, use the link below. bit.ly/30itgpC</p>


2.1.12 Tool Container

Concept: Grouping Tools

 <p>Figure-2-13-Tool Container</p>	<p>Tool Palette: Documentation</p>
	<p>Groups tools together for clarity and allows the tools to be disabled when not required. For more details, use the link below. bit.ly/33IHJxl</p>


2.1.13 Transpose

Concept: Deformatizing Data

 <p>Figure-2-14-Transpose</p>	<p>Tool Palette: Transform</p>
	<p>Helps to deformatize the data. For more details, use the link below. bit.ly/2HgAnY9</p>

2.1.14 Union

Concept: Appending records

 <p>Figure-2-15-Union</p>	<p>Tool Palette: Join</p>
	<p>Appends records together one after another from multiple data sources. For more details, use the link below. bit.ly/2Z5UsLX</p>

2.1.15 Cross Tab

Concept: Normalizing Data Sets

 <p data-bbox="337 586 610 614">Figure-2-16-Cross Tab</p>	Tool Palette: Transform
	<p data-bbox="723 406 1096 539">Creates a normalized (more human-readable) data set by creating columns out of the rows of data.</p> <p data-bbox="723 548 1103 614">For more details, use the link below.</p> <p data-bbox="723 621 935 652">bit.ly/2zdUUbz</p>

2.2 Freestyle

To...	<input type="text"/>
Cc...	<input type="text"/>
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?

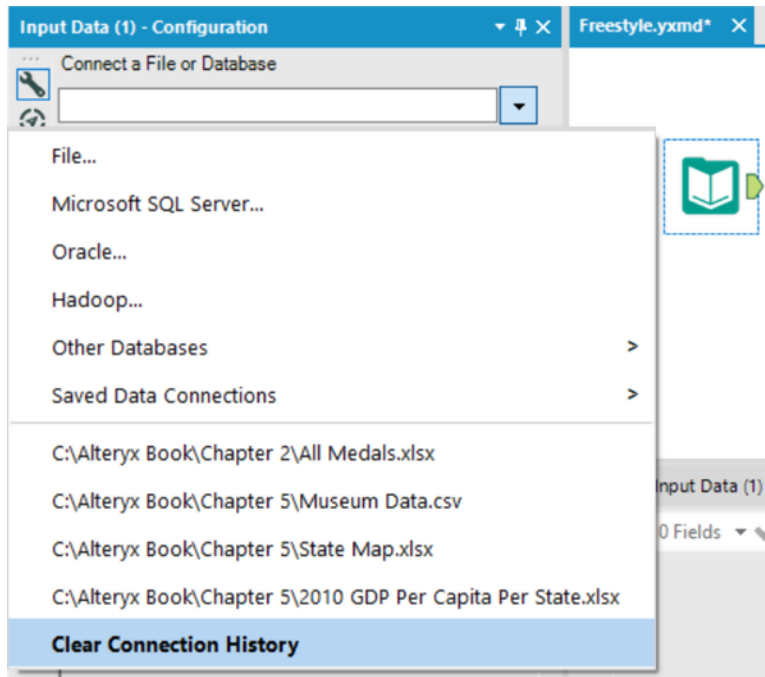
Assume each Gold is worth 3 points, Silver is worth 1.5 points and Bronze is worth 1 point.

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 we can connect to data.



**Figure-2-17-Freestyle Skiing
Input Data Configuration**

Now, navigate to where data files are unpackaged and connect to the file in *Chapter 2 – All Medals.xlsx*. For downloading the data associated with this book, please refer to the letter to the reader at the beginning of this book.

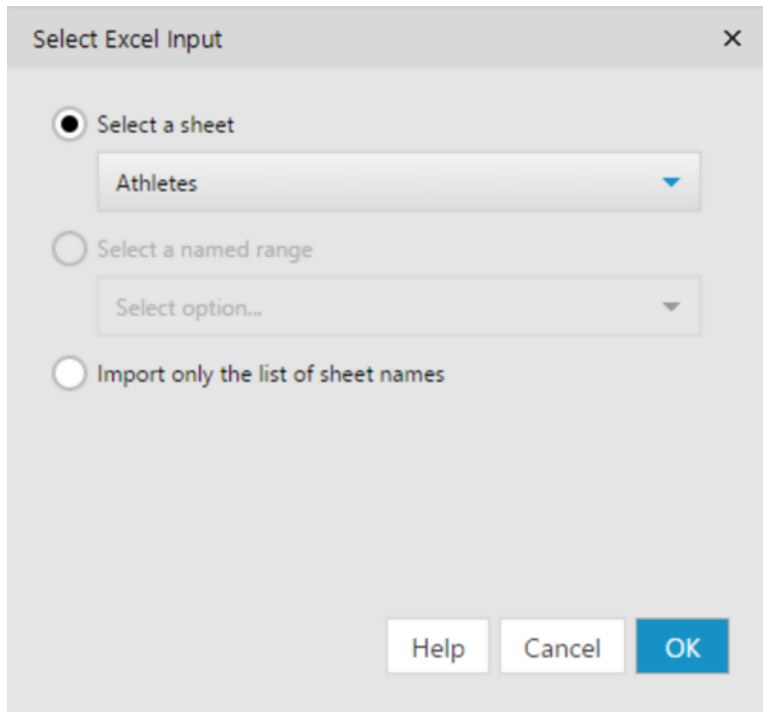
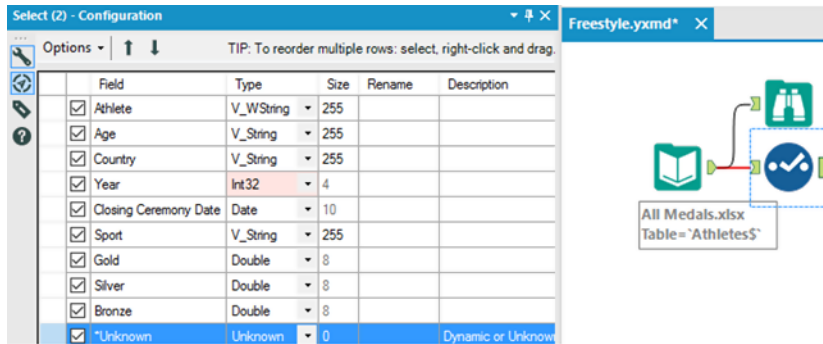


Figure-2-18-Freestyle Skiing - Data Input

When connected, we see this window pop up. Select Athletes and click OK to connect to the Athletes sheet in the *All Medals* Excel file, as 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 the data we're getting is correct.
- *Select* allows us to make sure the fields are in the right format from the beginning.



**Figure-2-19-Freestyle Skiing
Select Configuration**

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

Now that 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*, as 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 have not run it, we may not have enough information to filter it properly. In this case, we want to run the module so there's data in the *Browse* tool for us to work with.

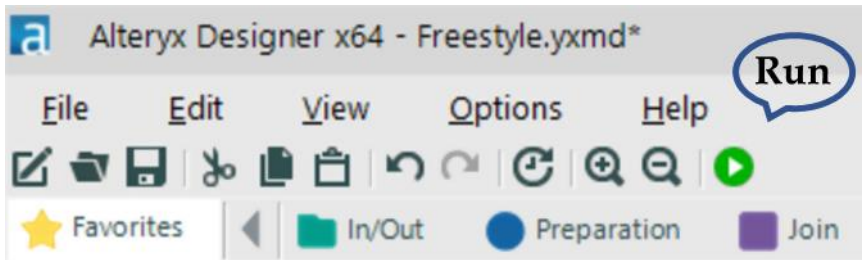


Figure-2-20 – Freestyle Skiing
Click Run to populate Browse

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 any errors.

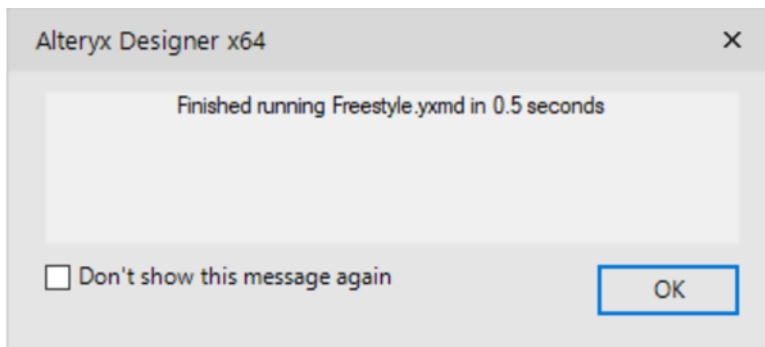


Figure-2-21-Freestyle Skiing – Message after Running Workflow

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

Now we can start thinking about the filter. We know we're 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*.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Gold	Silver	Bronze
5815	Irina Slutskaya	23	Russia	2002	2002-02-24	Figure Skating	0	1	0
5816	Aleksey Yagudin	21	Russia	2002	2002-02-24	Figure Skating	1	0	0
5817	Zhao Hongbo	28	China	2002	2002-02-24	Figure Skating	0	0	1
5818	Shannon Bahrke	29	United States	2010	2010-02-28	Freestyle Skiing	0	0	1
5819	Dale Begg-Smith	25	Australia	2010	2010-02-28	Freestyle Skiing	0	1	0
5820	Hedda Berntsen	33	Norway	2010	2010-02-28	Freestyle Skiing	0	1	0
5821	Alexandre Bilodeau	22	Canada	2010	2010-02-28	Freestyle Skiing	1	0	0
5822	Aleksey Grishin	30	Belarus	2010	2010-02-28	Freestyle Skiing	1	0	0
5823	Audun Grønvold	33	Norway	2010	2010-02-28	Freestyle Skiing	0	0	1

Figure-2-22-Freestyle Skiing – Browse

This is a crucial piece of information we didn't have before. Now that we know exactly what we need to look for in our data, 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 dropdown to *Sport*, and type *Freestyle Skiing* into the text box, like we see below.

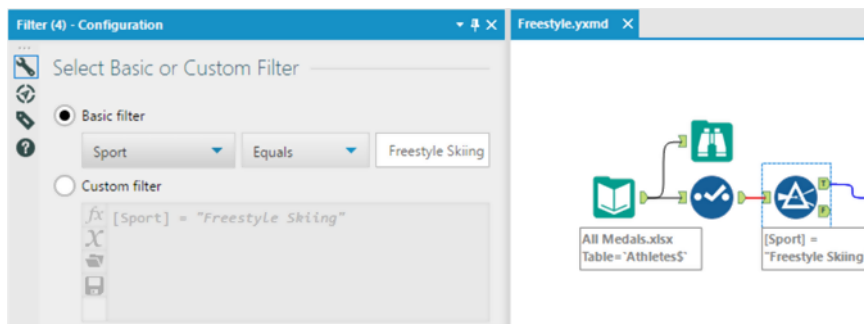


Figure-2-23-Freestyle Skiing – First Filter Configuration

Notice the Expression says `[Sport] = "Freestyle Skiing."` That is because field names are in square brackets, and string values are in quotes. What's happening here is that for each record, we test

to see if the value in Sport is 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 the records with years 2002 and 2006. If we look at the *Select* tool on the previous page, we will see that the *Year* field is a string. That is fine; we simply need to remember it while we're writing the filter formula. Drag a new *Filter* tool onto the canvas and make sure the first Filter's true (T) output is connected to the new Filter's input.

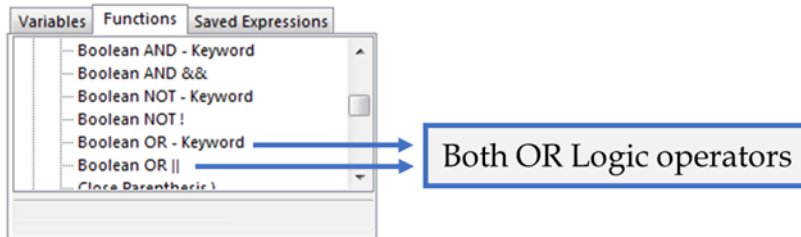


Figure-2-24-Freestyle Skiing
 Filter Configuration - Functions

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

One way we could do that 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 output from the two. The three Boolean operators we will be discussing are *AND*, *OR*, and *NOT*.

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

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

NOT: if a value is **true**, then **false**; if the value is **false**, then **true**. 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 who are 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 you might be used to.

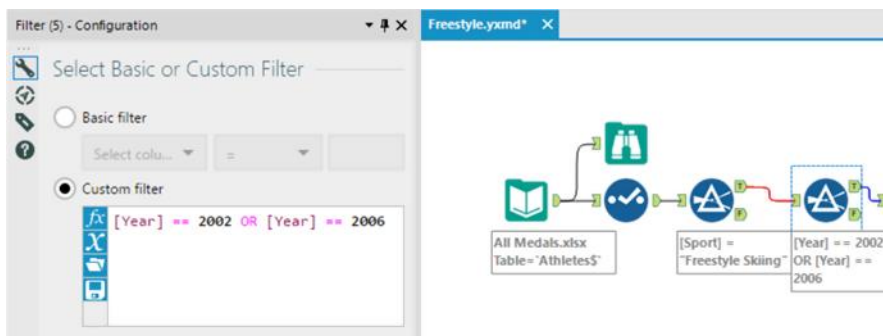


Figure-2-25-Freestyle Skiing Custom Filter Configuration

See Appendix F for examples of Boolean logic. Now that we know about logical operators, we can finally finish configuring the filter. We can use the formula `[Year] = "2002" OR [Year] = "2006"` in order to filter this data.

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

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

We have now removed the data we don't require in order to solve this question. But still, the problem remains that the data is too granular. We know who the athlete was and in which year that athlete won a medal(s). We should bring in a *Summarize* tool in order to aggregate 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.

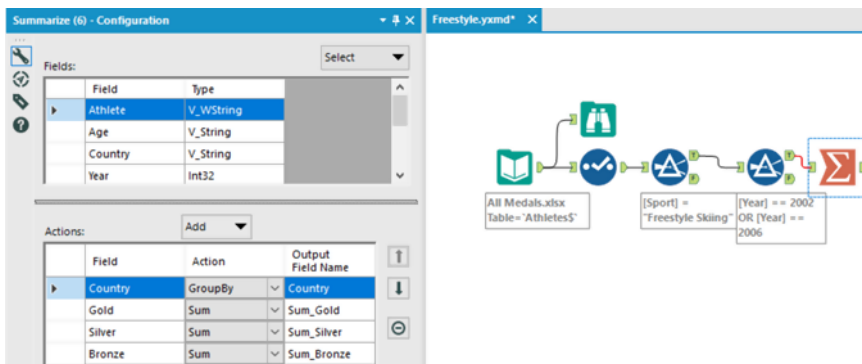
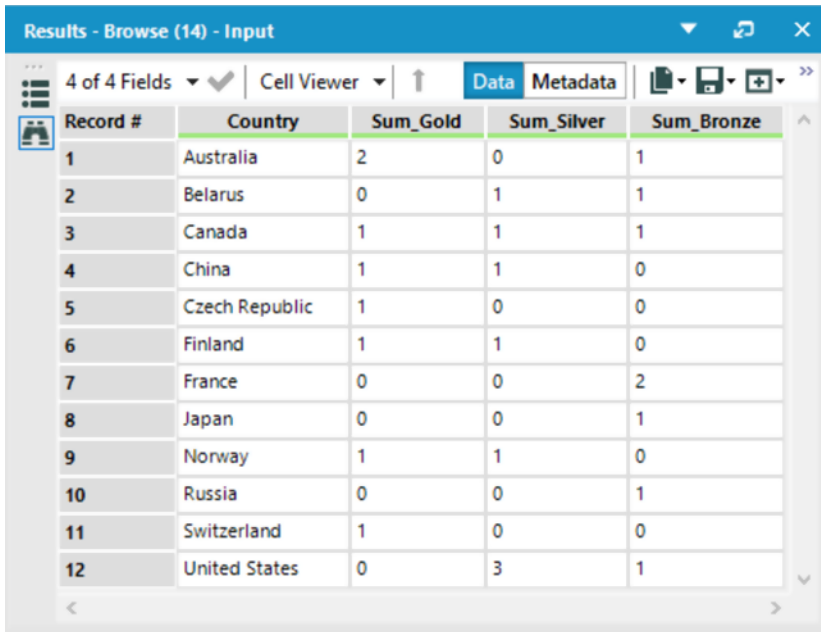


Figure-2-26-Freestyle Skiing Summarize Configuration

Best practice is to place a *Browse* tool after every tool that transforms the data into a significantly different format. Summarize is one of those tools.



Record #	Country	Sum_Gold	Sum_Silver	Sum_Bronze
1	Australia	2	0	1
2	Belarus	0	1	1
3	Canada	1	1	1
4	China	1	1	0
5	Czech Republic	1	0	0
6	Finland	1	1	0
7	France	0	0	2
8	Japan	0	0	1
9	Norway	1	1	0
10	Russia	0	0	1
11	Switzerland	1	0	0
12	United States	0	3	1

**Figure-2-27-Freestyle Skiing
Browse after Summarize**

We can see that there are four columns with the total counts of gold, silver, and bronze medals listed for each of the 12 countries that won freestyle skiing medals in 2002 and 2006. Notice that 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 performed the *best*. If we look back at the e-mail, 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 $[\text{Score}] = 3 * [\text{Sum_Gold}] + 1.5 * [\text{Sum_Silver}] + [\text{Sum_Bronze}]$.

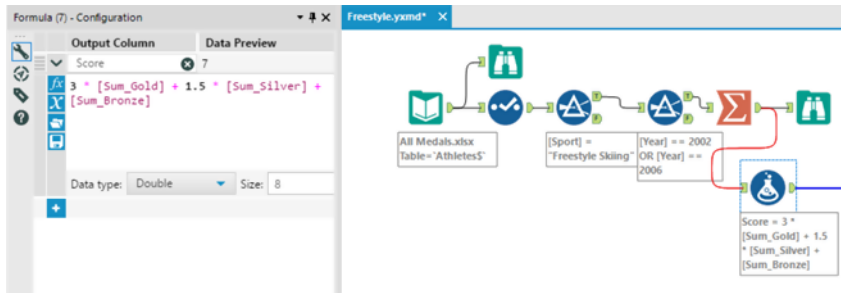
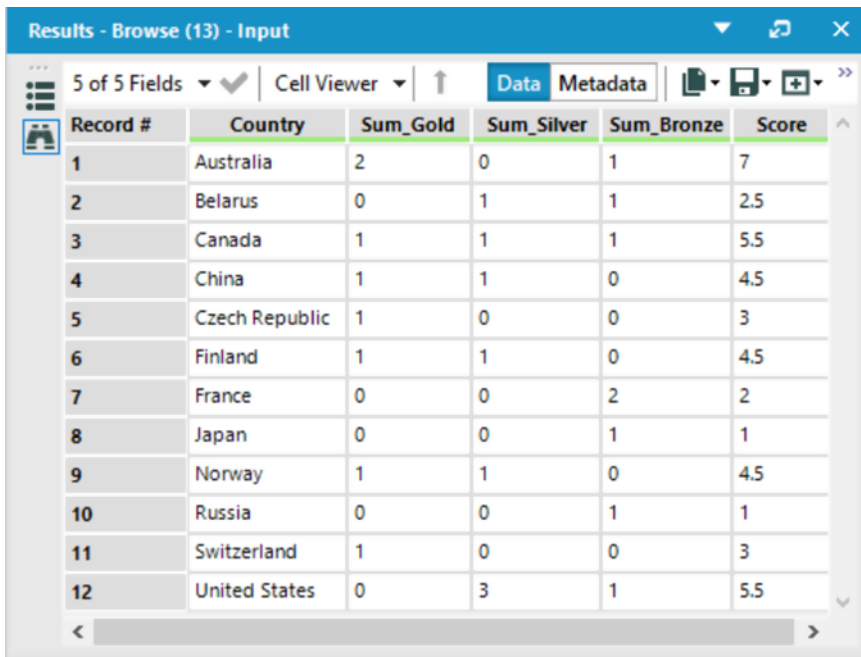


Figure-2-28-Freestyle Skiing - Formula Configuration

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



The screenshot shows the 'Results - Browse (13) - Input' window. It displays a table with 13 records. The columns are Record #, Country, Sum_Gold, Sum_Silver, Sum_Bronze, and Score. The data is as follows:

Record #	Country	Sum_Gold	Sum_Silver	Sum_Bronze	Score
1	Australia	2	0	1	7
2	Belarus	0	1	1	2.5
3	Canada	1	1	1	5.5
4	China	1	1	0	4.5
5	Czech Republic	1	0	0	3
6	Finland	1	1	0	4.5
7	France	0	0	2	2
8	Japan	0	0	1	1
9	Norway	1	1	0	4.5
10	Russia	0	0	1	1
11	Switzerland	1	0	0	3
12	United States	0	3	1	5.5

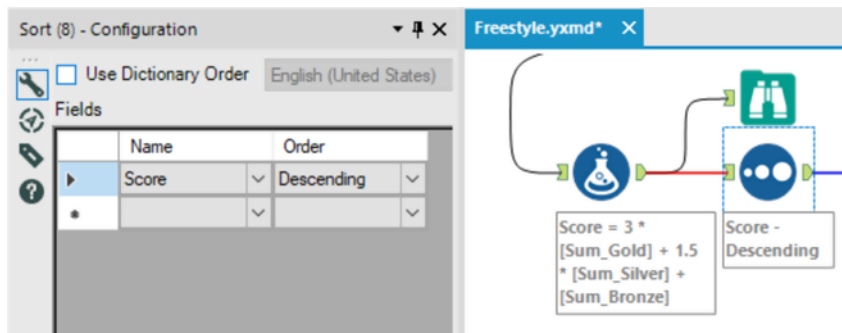
Figure-2-29-Freestyle Skiing
 Browse after Formula tool

We see there's a new field called Score that is an unordered data set 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 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.

As we have discussed, we will first reorder the data using a *Sort* tool. We will set up our data in a descending order based on Score, as shown below.



**Figure-2-30-Freestyle Skiing
Sort Configuration**

Next, we just need the first record, so we are going to use the *Sample* tool so we only get Top 1 Record as output from the Sort.

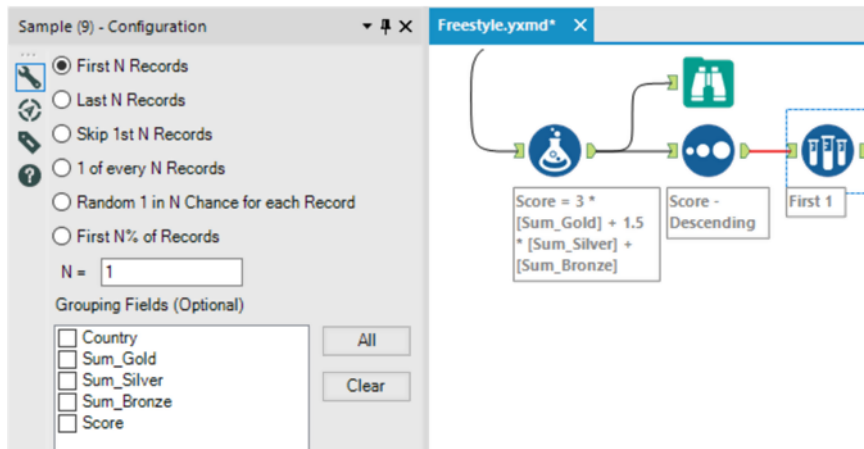


Figure-2-31-Freestyle Skiing
Sample Configuration

Since we will no longer need the rest of the fields, we can use a *Select* tool to remove everything that is not a country name.

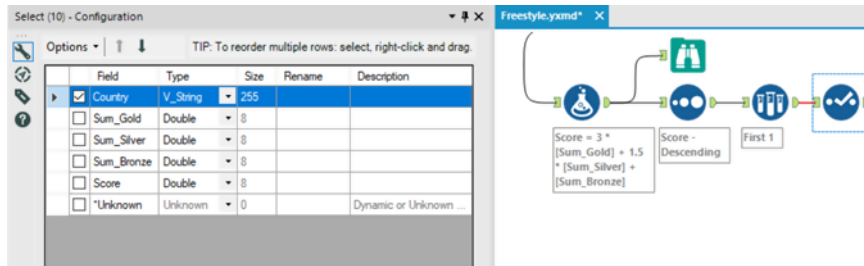
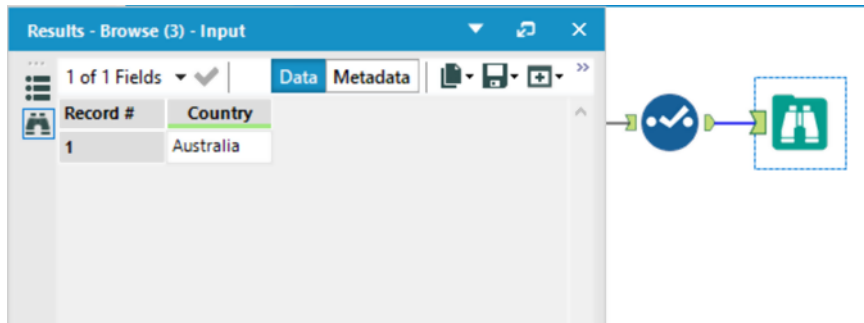


Figure-2-32-Freestyle Skiing
Select Configuration

Finally, we can put a *Browse* tool at the end and run the workflow to see the results.



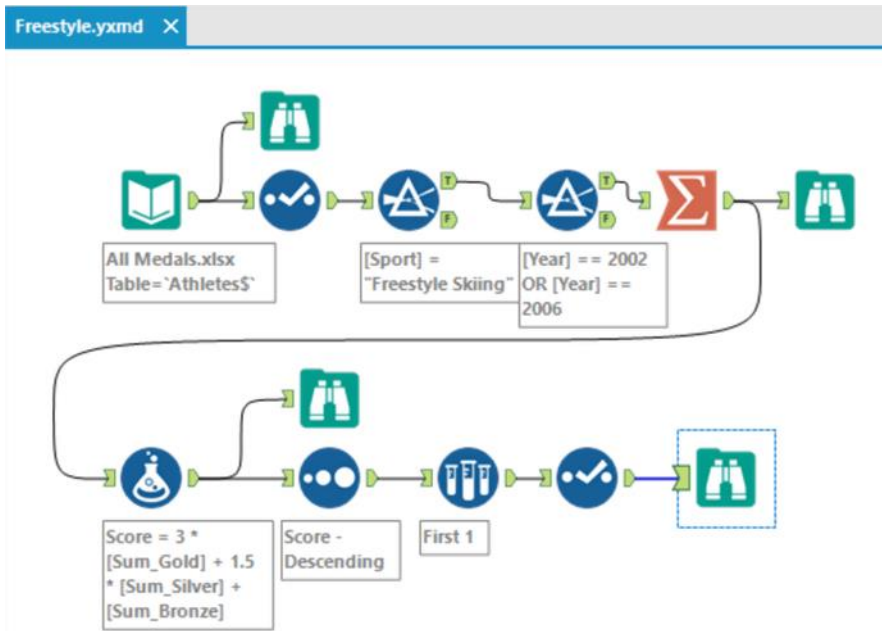
**Figure-2-33-Freestyle Skiing
Browse after Select**

We could have stopped when we first saw Australia had the highest score in the previous *Browse* tool. The reason we didn't is because when we are performing an analysis, we want our results to be perfectly repeatable. If we had interpreted the previous *Browse* tool incorrectly, there would have been no way of finding out why the error occurred.

That 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. Creating the workflow in this way affords us these two benefits:


- We would have a second verification that the answer was what we expected.
- There would be repeatability of the result so we could 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 when complete:



**Figure-2-34-Freestyle Skiing
Workflow after completion**

2.3 Let's Tidy Things Up

 Send	From ▾	
	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 data set.
2. Transpose the data set.
3. Make sure the fields are named correctly.
4. Export the data set.

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

1. Import the data.
2. Browse the data.
3. Make sure the data has the right data 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 data set.

Let's create a new workflow and save it as *Let's Tidy Things Up*. We need to import the same data 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 in *Chapter 2 - All Medals.xlsx*.

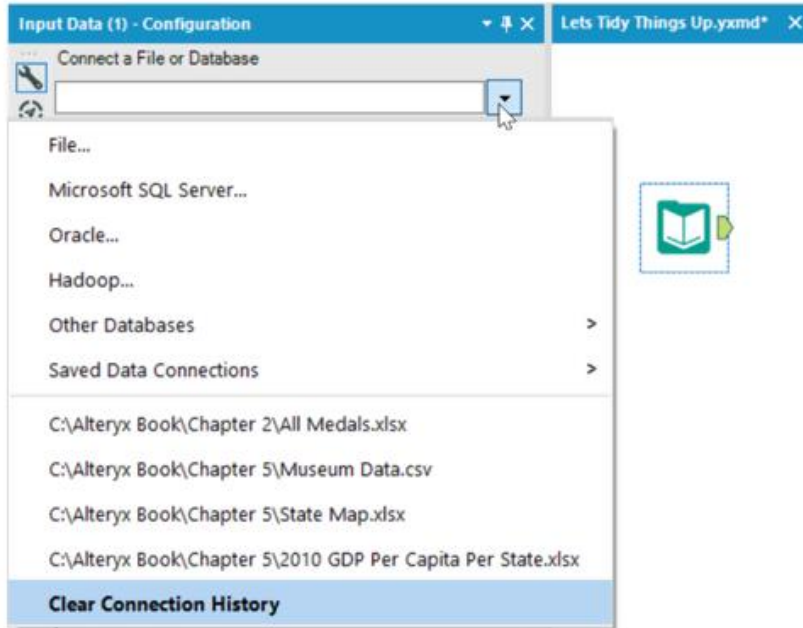


Figure-2-35 Let's Tidy Things Up – Input Data

Now we will put a *Browse* and select statement following the *Input* tool.

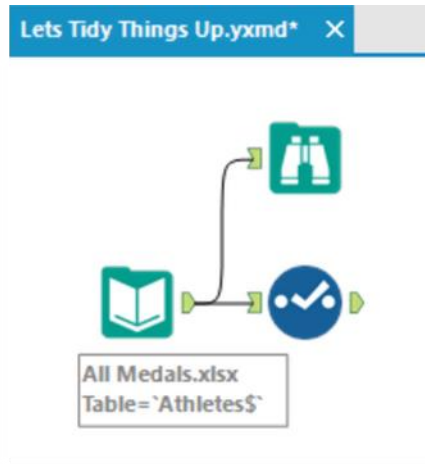


Figure-2-36-Let’s Tidy Things Up – Medals Data

All the fields have appropriate data types, so we can move directly to the transposition.

Let’s run the Module to see how the data is structured by clicking on the *Browse* tool.



Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Gold	Silver	Bronze
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	8	0	0
2	Michael Phelps	19	United States	2004	2004-08-29	Swimming	6	0	2
3	Michael Phelps	27	United States	2012	2012-08-12	Swimming	4	2	0
4	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	1	2	3
5	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	2	1	3
6	Alicia Coutts	24	Australia	2012	2012-08-12	Swimming	1	3	1
7	Missy Franklin	17	United States	2012	2012-08-12	Swimming	4	0	1
8	Ryan Lochte	27	United States	2012	2012-08-12	Swimming	2	2	1
9	Allison Schmitt	22	United States	2012	2012-08-12	Swimming	3	1	1

Figure-2-37-Let’s Tidy Things Up – Data in Browse Tool

The *Transpose* tool takes normalized data and denormalizes it. If we take the data stream coming out of *Select* and pass it into a *Transpose*, we can make the data tidier.

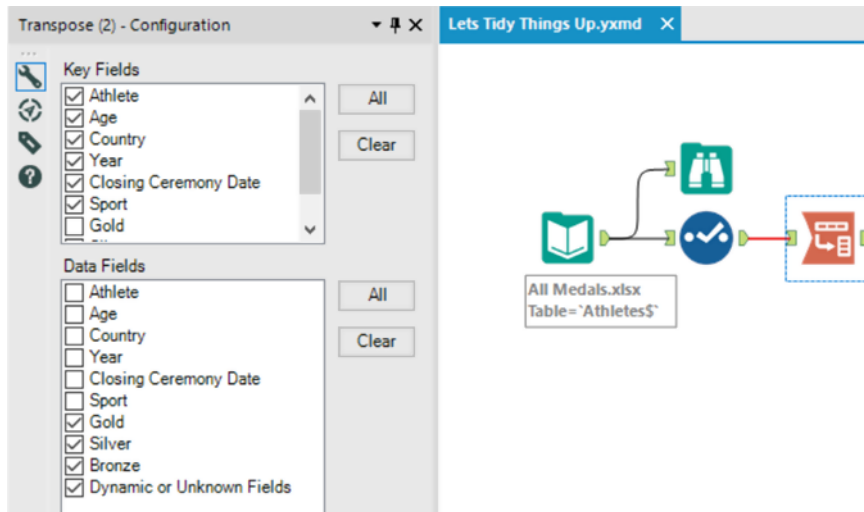
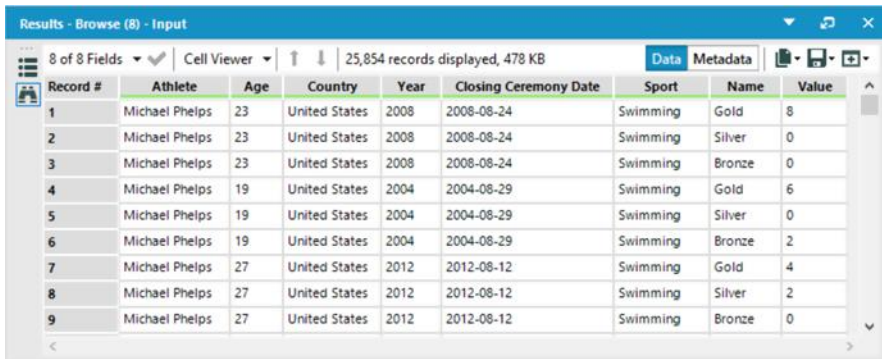


Figure-2-38-Let's Tidy Things Up – Transpose Configuration

We want to keep all the fields as they are, except for gold, silver, and bronze. So we select all but those three fields under Key Fields, and then we select gold, silver, and bronze under the Data Fields. If we want to drop a field entirely—say, Closing Ceremony Date—we can 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 those tools. Let's add a *Browse* to the end of the data stream and run it to see what we have.



The screenshot shows the Alteryx 'Results - Browse (8) - Input' window. The table displays 8 records for Michael Phelps, with columns for Record #, Athlete, Age, Country, Year, Closing Ceremony Date, Sport, Name, and Value. The data is as follows:

Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Name	Value
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Gold	8
2	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Silver	0
3	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Bronze	0
4	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Gold	6
5	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Silver	0
6	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Bronze	2
7	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Gold	4
8	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Silver	2
9	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Bronze	0

Figure-2-39-Let's Tidy Things Up – Data in Browse after Transpose

If we compare the top three records from the new *Browse* to the one that came out of the *Input* in Figure 2-37, we see we have two fields—*Name* and *Value*—and no longer have the fields *Gold*, *Silver*, and *Bronze*. We also notice from *Athlete* to *Sport* that all the fields are identical to the first three records in the original data set. That is because we replicated them for each column we created.

That is one of the reasons tidy data is not particularly human-readable but is highly computer-readable. Since all 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.

To clean this data entirely would mean we need to rename *Name* and *Value* to field names that will give better context to those fields. Add a *Select* statement and rename the *Name* and *Value* fields *Medal Type* and *Medal Count*, respectively.

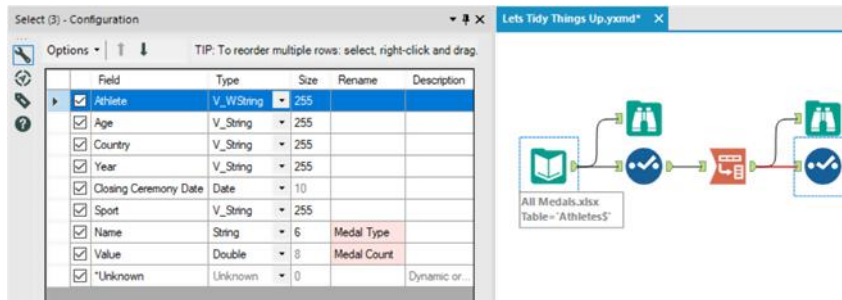


Figure-2-40-Let's Tidy Things Up – Select Configuration

By looking at the values in the last *Browse* tool we created, we know there are some records that show zero medals were won. We are going to filter out those data points by adding a *Filter* tool after Select.

Our goal is to filter out any records with 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 shown in the following figure.

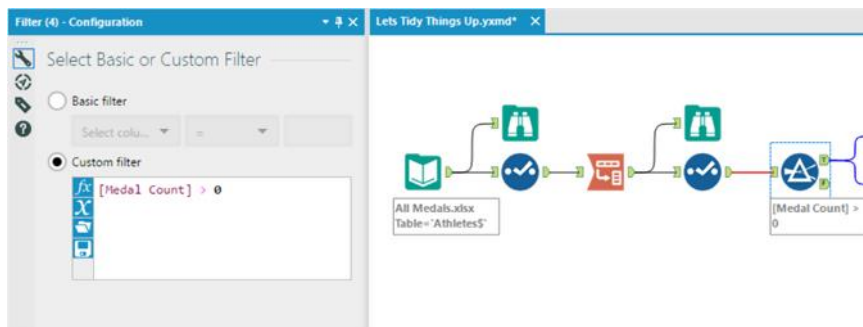


Figure-2-41-Let's Tidy Things Up – Filter Configuration

We can see that if we select Medal count, we have different options in the operator dropdown. That 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 it says **[Medal Count] > 0**. That is because we do not put numeric values in quotes. Alteryx recognized that when we selected a numeric field in the basic filter dropdown, the “0” we typed in meant the number 0 and not the string 0, so it adds 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 we don't need to open the file to make sure we have created it correctly.

We now add a *Browse* and an *Output Data* tool to the end of the data stream whose output is “True.” We are going to write the file to the same folder where we have saved the *Let's Tidy Things Up.yxmd*.

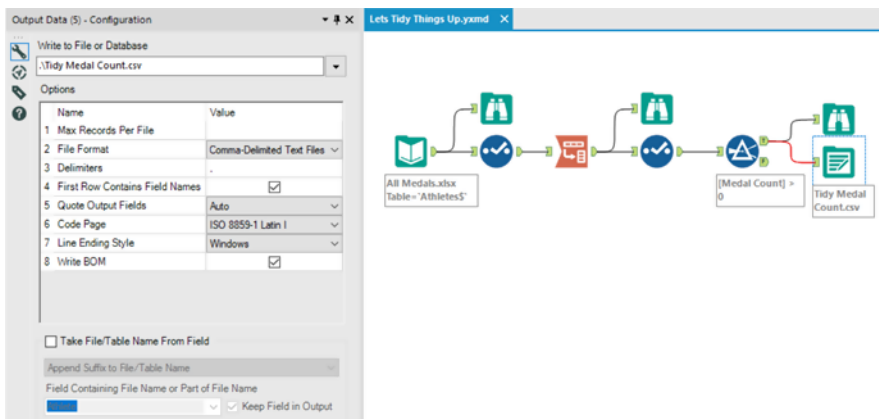


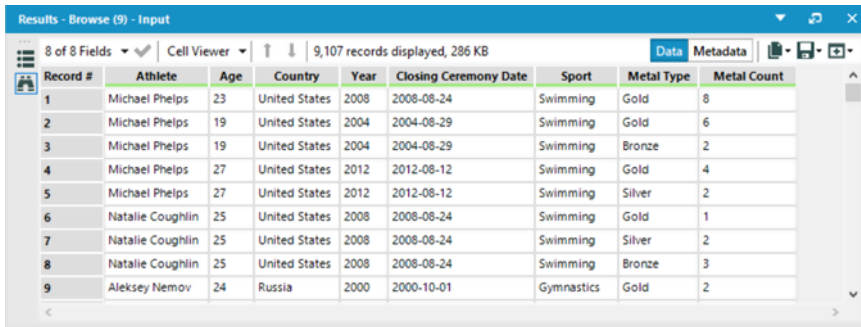
Figure-2-42-Let's Tidy Things Up – Output Data Configuration

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. That allows us to reference 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 the one where we have our 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 will see how the transformed data set looks. That ensures that the information written into the .csv was correct.



The screenshot shows the 'Results - Browse (9) - Input' window in Alteryx. It displays a table with 9 records and 8 fields. The fields are Record #, Athlete, Age, Country, Year, Closing Ceremony Date, Sport, Metal Type, and Metal Count. The data is as follows:

Record #	Athlete	Age	Country	Year	Closing Ceremony Date	Sport	Metal Type	Metal Count
1	Michael Phelps	23	United States	2008	2008-08-24	Swimming	Gold	8
2	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Gold	6
3	Michael Phelps	19	United States	2004	2004-08-29	Swimming	Bronze	2
4	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Gold	4
5	Michael Phelps	27	United States	2012	2012-08-12	Swimming	Silver	2
6	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Gold	1
7	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Silver	2
8	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Bronze	3
9	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	Gold	2

Figure-2-43-Let’s Tidy Things Up – Browse after Filter

Here is how the *Let’s Tidy Things Up* workflow looks upon completion.

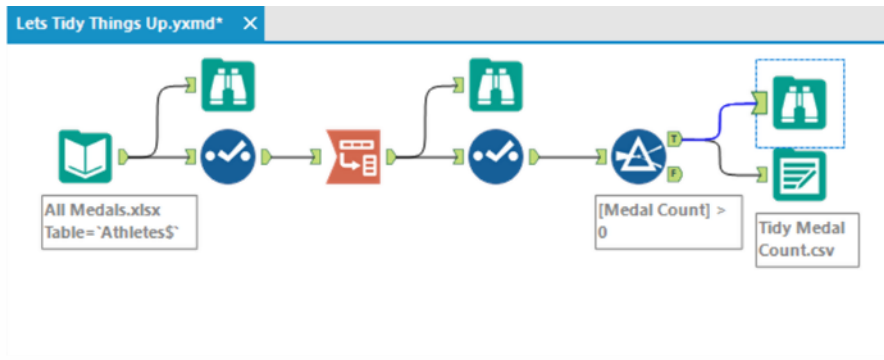
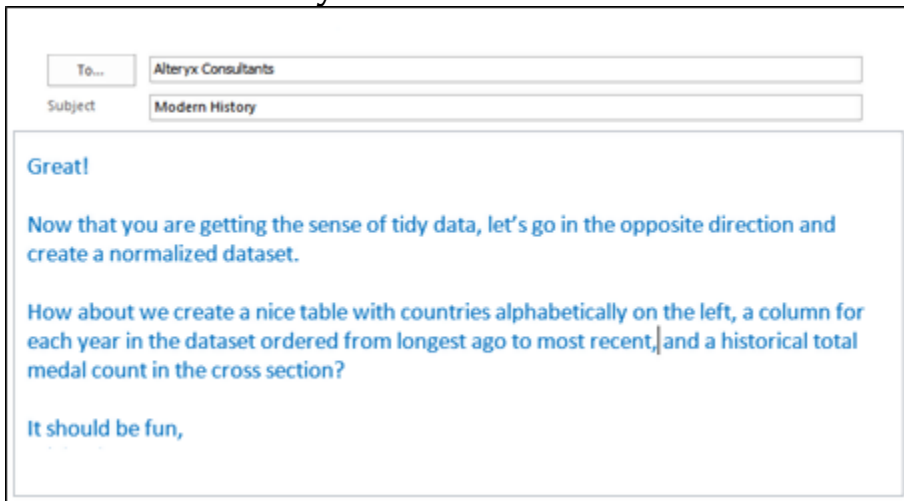


Figure-2-44-Let's Tidy Things Up – Final Workflow after completion

2.4 Modern History



Notice that there's considerably less context built into this e-mail. We often get very sparse information from people who assume we have the context. In this case, it was assumed we were talking about the medal data we've been working with during the training so far.

That is a much more complicated process than the last exercise. That's only because the data last time was set up very well for what we were doing, and this time it wasn't.

We are going to be connecting to the same data source we've 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 *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 the copied content. We see that the input has been copied over, and we do not need to recreate the connection. Connect to the file in *Chapter 2 - All Medals.xlsx*.

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.

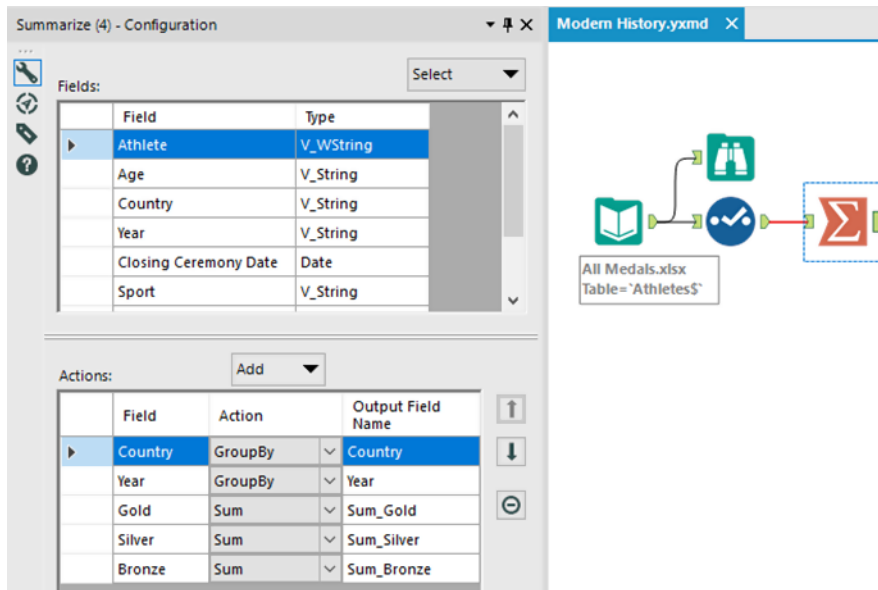


Figure-2-45-Modern History – Summarize Configuration

We know this data is too granular for our desired result, so we will summarize it. Based on the e-mail, we know the only information we will need in the end is the country, the year, and something to do with the medals. While using the Summary tool, we will 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.

We now add a formula that creates a *Total Medal* count by adding the gold, silver, and bronze fields for each record. (Remember, we used a Summarize tool, so we should have a *Browse* tool.)

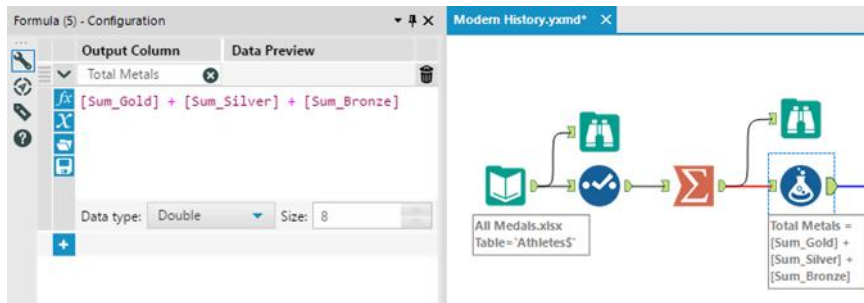


Figure-2-46-Modern History – Formula Configuration

We can now add a Select tool, allowing us to keep only the Country, Year, and Total Medals fields, which we will use to create the table.

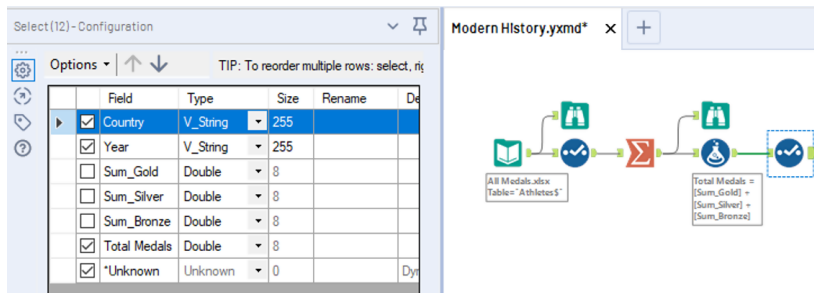


Figure 2-47- Modern History – Select Configuration

To get the historical medal count, we will 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 will sort the Country and Year in ascending order to help us in two ways. Initially, it will help create 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.

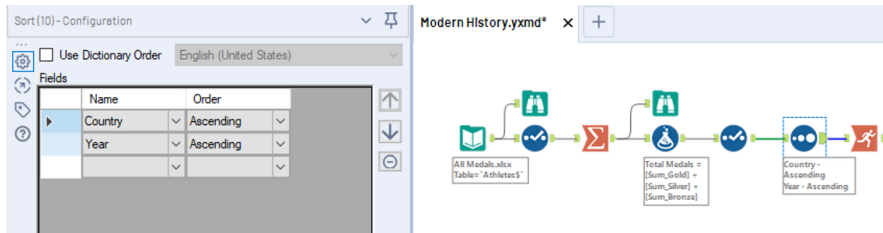


Figure 2-48- Modern History - Sort Configuration

Now that we have the data in a specific order, 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. That will 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.

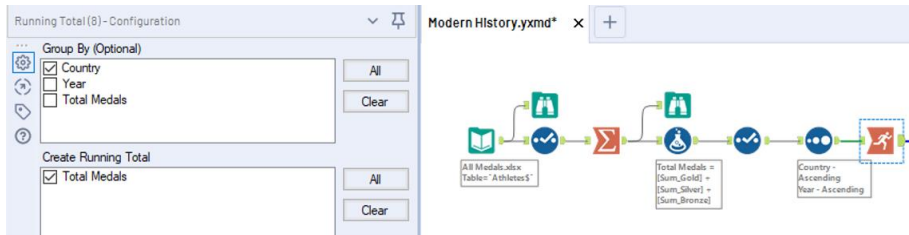


Figure 2-49- Modern History - Running Total Configuration

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

If we add a *Browse* tool and run the workflow, we can see 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 the country has won from one year to the next in a field called RunTot_Total Medals.

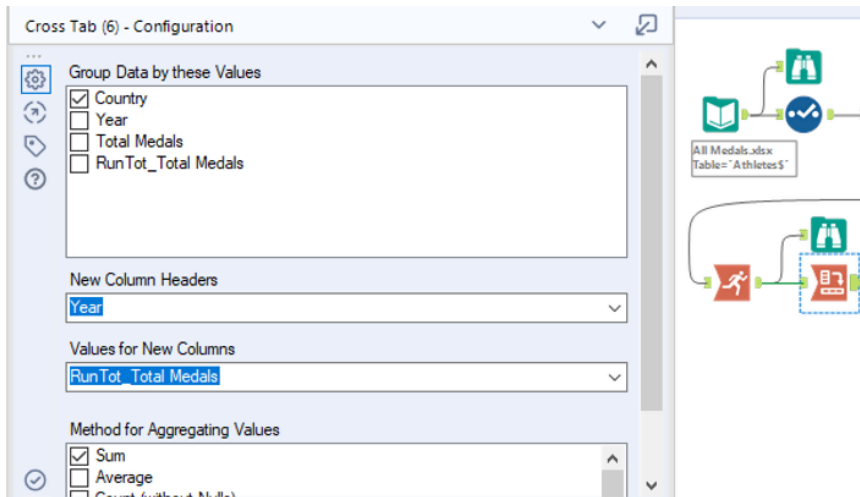
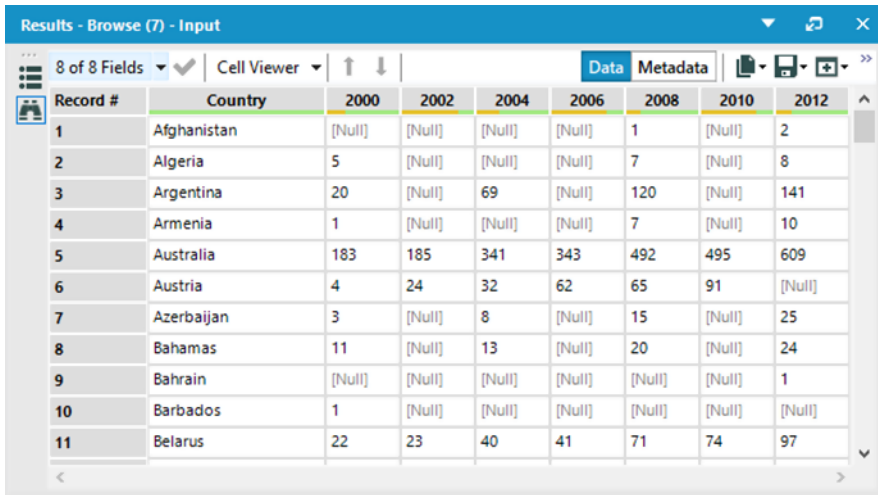


Figure 2-50- Modern History – Cross Tab

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 as in the above image, we will be close to our goal.

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



The screenshot shows the 'Results - Browse (7) - Input' window in Alteryx. The table displays medal counts for 11 countries across seven years (2000, 2002, 2004, 2006, 2008, 2010, 2012). Null values are present for several years, particularly for Afghanistan, Algeria, Armenia, Azerbaijan, Bahrain, Barbados, and Belarus.

Record #	Country	2000	2002	2004	2006	2008	2010	2012
1	Afghanistan	[Null]	[Null]	[Null]	[Null]	1	[Null]	2
2	Algeria	5	[Null]	[Null]	[Null]	7	[Null]	8
3	Argentina	20	[Null]	69	[Null]	120	[Null]	141
4	Armenia	1	[Null]	[Null]	[Null]	7	[Null]	10
5	Australia	183	185	341	343	492	495	609
6	Austria	4	24	32	62	65	91	[Null]
7	Azerbaijan	3	[Null]	8	[Null]	15	[Null]	25
8	Bahamas	11	[Null]	13	[Null]	20	[Null]	24
9	Bahrain	[Null]	[Null]	[Null]	[Null]	[Null]	[Null]	1
10	Barbados	1	[Null]	[Null]	[Null]	[Null]	[Null]	[Null]
11	Belarus	22	23	40	41	71	74	97

Figure-2-51-Modern History – Browse after Cross Tab

The results seen in the image are close but not exactly what we want. We get the correct running totals for the years when each country won medals; however, we get nulls for the years they did not.

What we need to do now is create a series of formulas to replace the nulls with zero or the previous value. Since we need to create formulas, we will have to use the formula tool. But this time, we have 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 value in the cell only if it is null. If the value in the column we're fixing is 2000, then it should be replaced with 0. If it is not 2000, then it should be replaced with whatever value is in the previously fixed column.

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

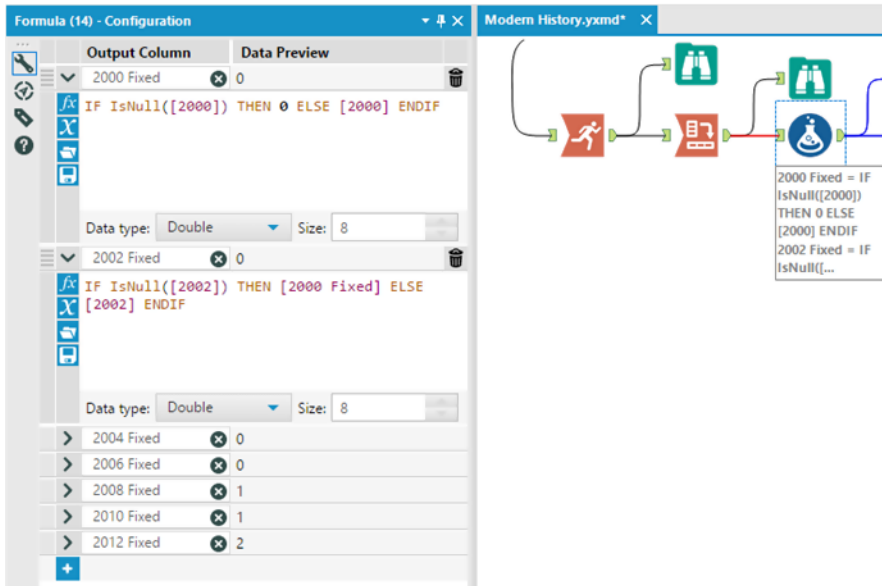
IF b1 THEN x ELSEIF b2 THEN y ELSE z ENDIF

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

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

IsNull(x)

Both of these syntaxes are under the Functions tab in the *Formula* tool if we need to reference them. The formulas we need are below:



The screenshot displays the Alteryx Formula tool configuration for a workflow named 'Modern History.yxmd'. The left pane shows the 'Output Column' list with the following formulas:

Output Column	Data Preview
2000 Fixed	0
2002 Fixed	0
2004 Fixed	0
2006 Fixed	0
2008 Fixed	1
2010 Fixed	1
2012 Fixed	2

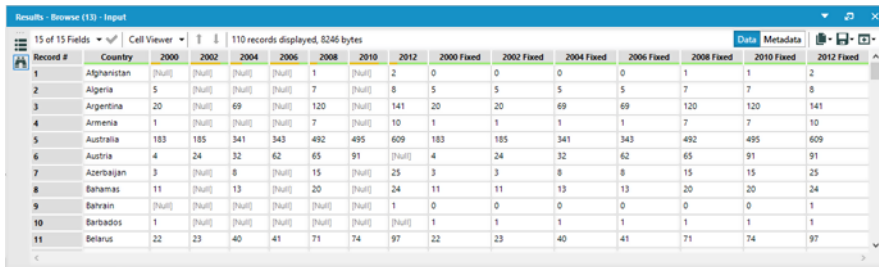
The right pane shows a workflow diagram with a Formula tool highlighted, displaying the formula for '2000 Fixed':

```
2000 Fixed = IF
IsNull([2000])
THEN 0 ELSE
[2000] ENDIF
```

Figure-2-52-Modern History – Formula Configuration

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

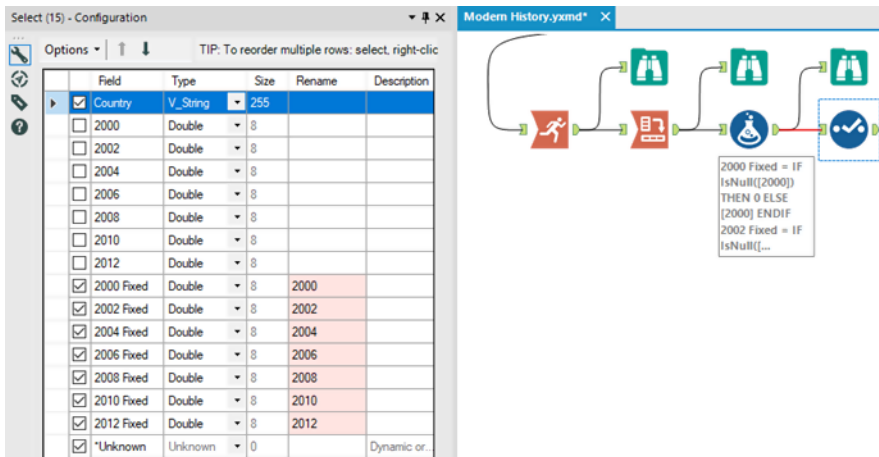
We can see we have two sets of fields: those with the original sparse data and those with the new dense data.



Record #	Country	2000	2002	2004	2006	2008	2010	2012	2000 Fixed	2002 Fixed	2004 Fixed	2006 Fixed	2008 Fixed	2010 Fixed	2012 Fixed
1	Alghanstan	[Null]	[Null]	[Null]	[Null]	1	[Null]	2	0	0	0	0	1	1	2
2	Algeria	5	[Null]	[Null]	[Null]	7	[Null]	8	5	5	5	5	7	7	8
3	Argentina	20	[Null]	69	[Null]	120	[Null]	141	20	20	69	69	120	120	141
4	Armenia	1	[Null]	[Null]	[Null]	7	[Null]	10	1	1	1	1	7	7	10
5	Australia	183	185	341	343	492	495	609	183	185	341	343	492	495	609
6	Austria	4	24	32	62	65	91	[Null]	4	24	32	62	65	91	91
7	Azerbaijan	3	[Null]	8	[Null]	15	[Null]	25	3	3	8	8	15	15	25
8	Bahamas	11	[Null]	13	[Null]	20	[Null]	24	11	13	13	20	20	24	24
9	Bahrain	[Null]	[Null]	[Null]	[Null]	[Null]	[Null]	1	0	0	0	0	0	0	1
10	Barbados	1	[Null]	[Null]	[Null]	[Null]	[Null]	[Null]	1	1	1	1	1	1	1
11	Belarus	22	23	40	41	71	74	97	22	23	40	41	71	74	97

Figure-2-53-Modern History – Browse after Formula

The next thing we need to do is remove and rename the columns we have, so add a *Select* tool to the end of the data stream.



Field	Type	Size	Rename	Description
<input checked="" type="checkbox"/>	Country	V_String	255	
<input type="checkbox"/>	2000	Double	8	
<input type="checkbox"/>	2002	Double	8	
<input type="checkbox"/>	2004	Double	8	
<input type="checkbox"/>	2006	Double	8	
<input type="checkbox"/>	2008	Double	8	
<input type="checkbox"/>	2010	Double	8	
<input type="checkbox"/>	2012	Double	8	
<input checked="" type="checkbox"/>	2000 Fixed	Double	8	2000
<input checked="" type="checkbox"/>	2002 Fixed	Double	8	2002
<input checked="" type="checkbox"/>	2004 Fixed	Double	8	2004
<input checked="" type="checkbox"/>	2006 Fixed	Double	8	2006
<input checked="" type="checkbox"/>	2008 Fixed	Double	8	2008
<input checked="" type="checkbox"/>	2010 Fixed	Double	8	2010
<input checked="" type="checkbox"/>	2012 Fixed	Double	8	2012
<input checked="" type="checkbox"/>	*Unknown	Unknown	0	Dynamic or...

```

2000 Fixed = IF
[IsNull]([2000])
THEN 0 ELSE
[2000] ENDIF
2002 Fixed = IF
[IsNull]([
    
```

Figure-2-54-Modern History – Select Configuration

The data preparation is now completed. We need to write the results 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*.

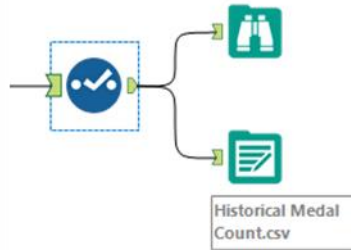


Figure-2-55-Modern History – Output

After doing these steps, the final workflow is as shown in the figure below:

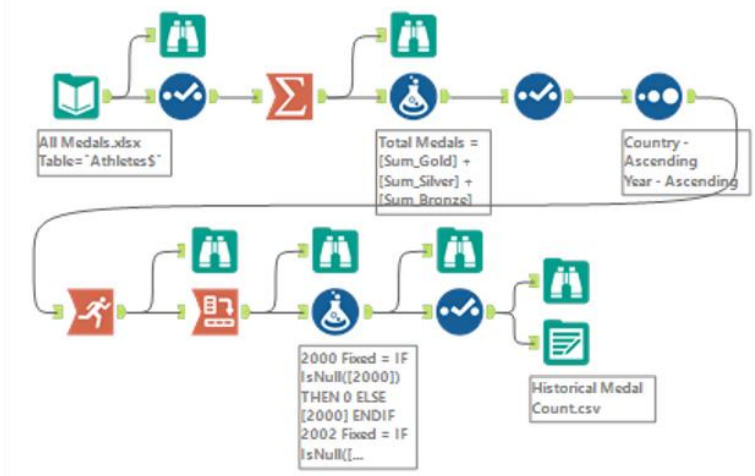
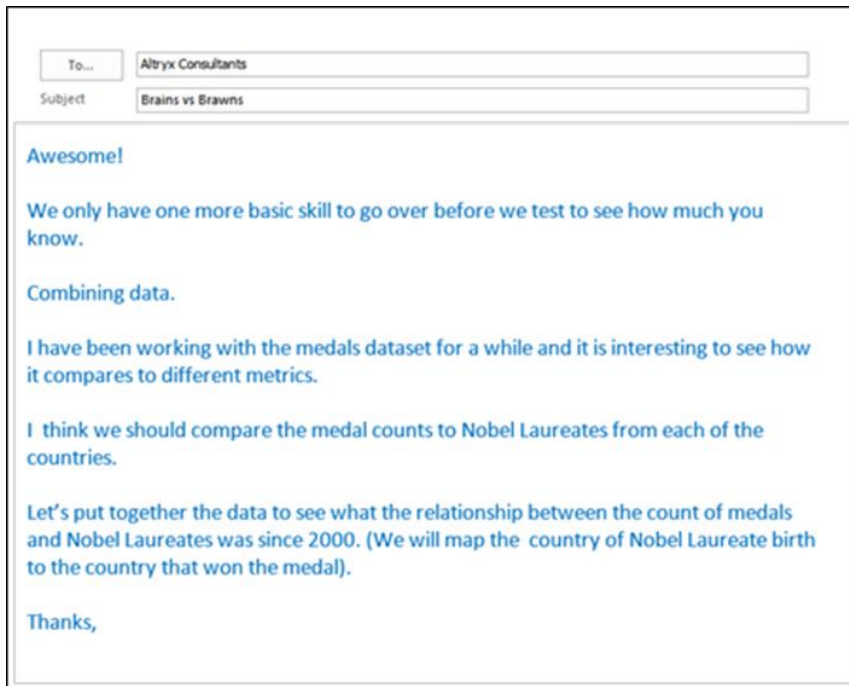


Figure 2-56- Modern History – Complete Workflow

2.5 Brains vs. Brawn



To... Alteryx 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 might have gone through different terrain and could have started from different sources. When they merge 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 preparing the medals data to be joined. We'll create a table with two columns called Country and Medal Count. Connect to the file *All Medals.xlsx* in *Chapter 2—All Medals.xlsx and Nobel Laureates.csv*.

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

1. Import data.
2. Transpose and rename the columns so the data is tidy.
3. Filter out the 0 medal records.
4. Summarize the data so 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 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's produced are shown in the following figures.

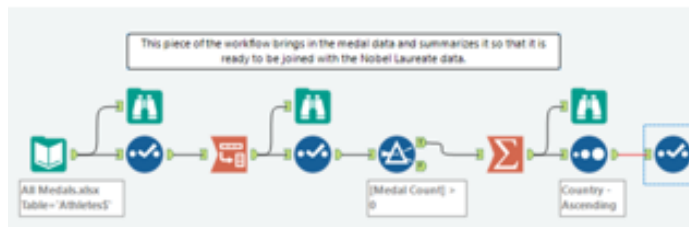


Figure-2-57-Brains vs. Brawn – Medal Count Preparation

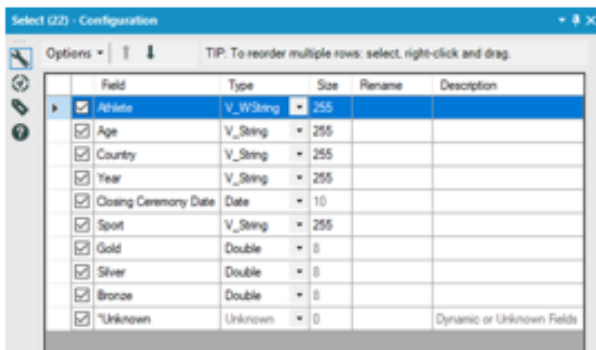


Figure-2-58-Brains vs. Brawn – Initial Steps - Select

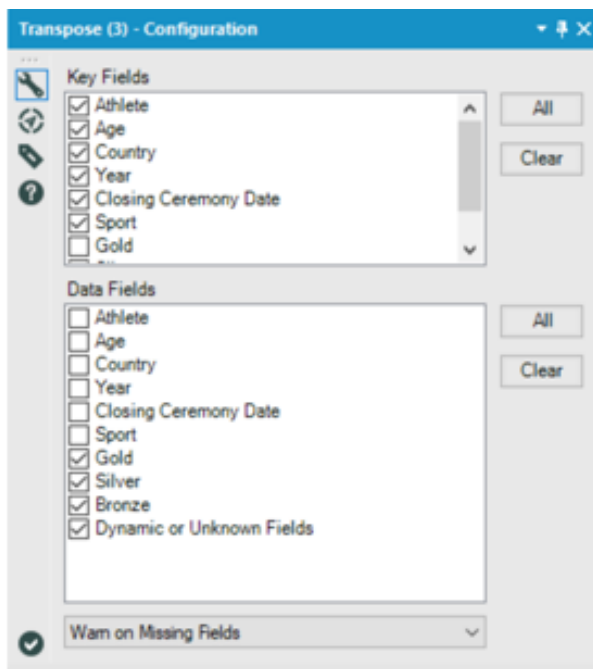


Figure-2-59-Brains vs. Brawn – Initial Steps - Transpose

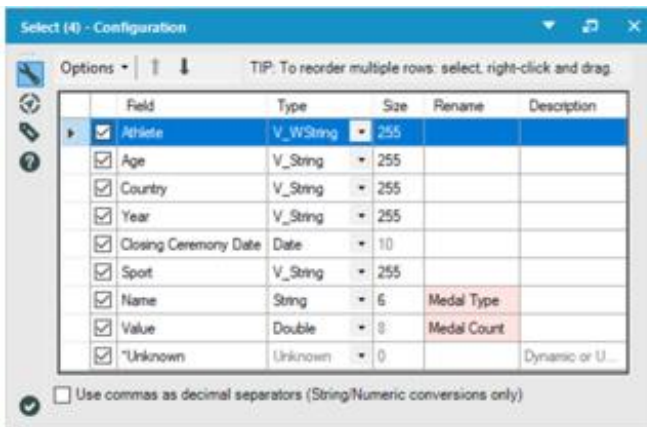


Figure-2-60-Brains vs. Brawn – Initial Steps – Select

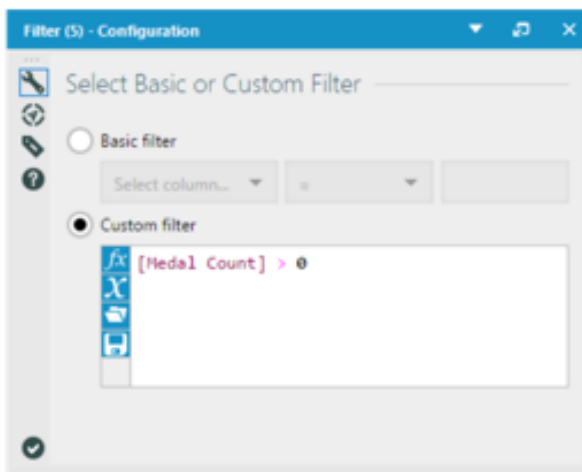


Figure-2-61-Brains vs. Brawn – Initial Steps – Filter

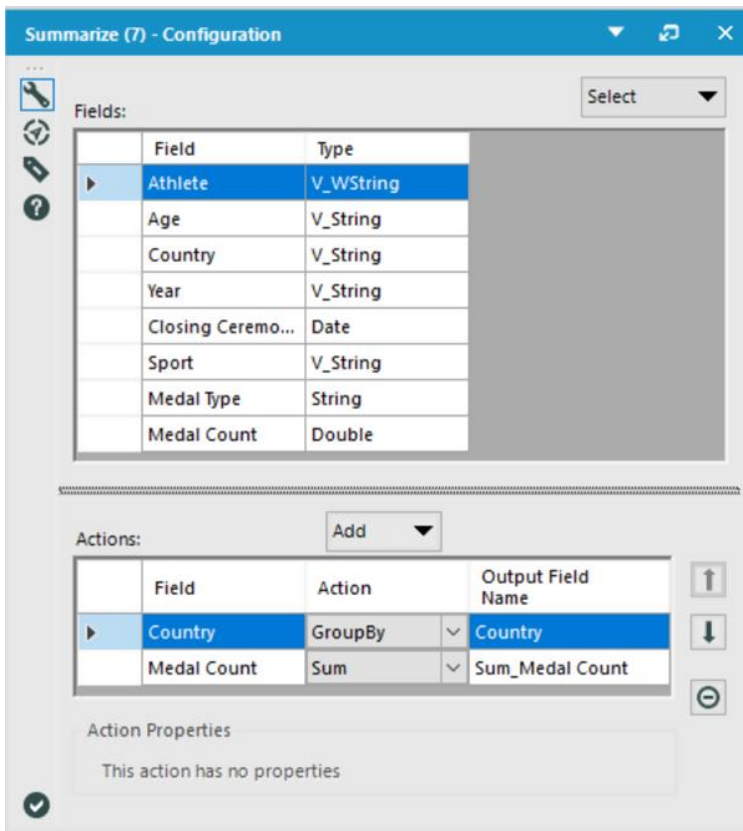


Figure-2-62-Summarize Configuration

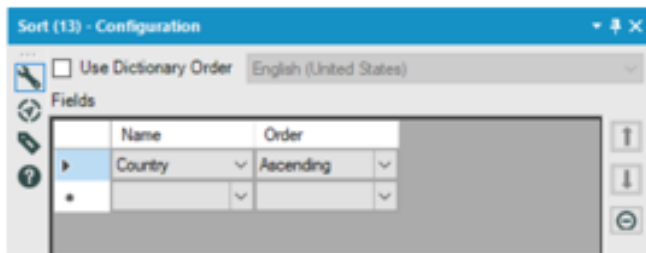
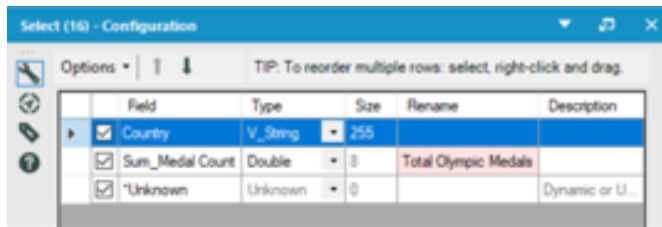


Figure-2-63-Brains vs. Brawn - Initial Steps - Sort



Field	Type	Size	Rename	Description
Country	V_String	255		
Sum_Medal Count	Double	0	Total Olympic Medals	
Unknown	Unknown	0		Dynamic or U...

Figure-2-64-Brains vs. Brawn - Initial Steps - Select

Now that we have the data in the above stream ready to be combined, we should prepare the other contributing data stream. Open the file called *Nobel Laureates.csv* in the folder Chapter 2 – The Games > Nobel Laureates. (Remember, we should always bring in a *Browse* and *Select* tool with an input.

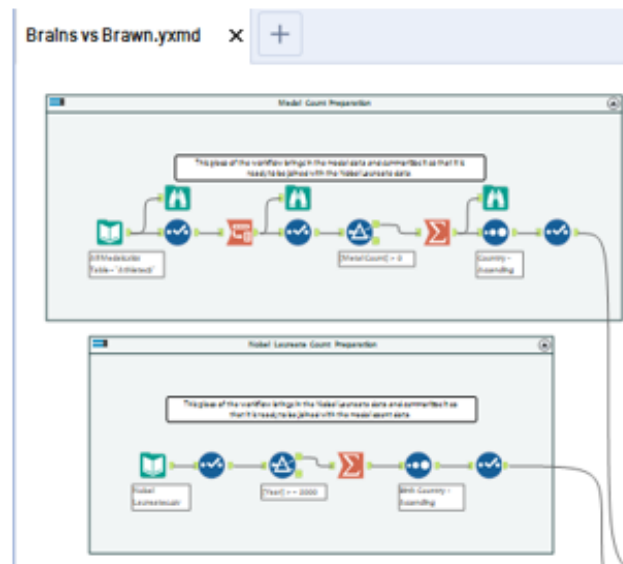
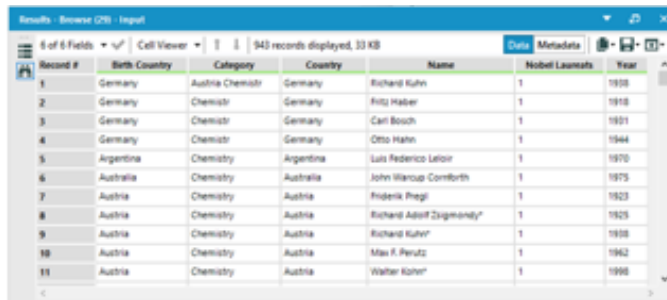


Figure-2-65-Brains vs. Brawn - Running unrelated analyses simultaneously

Notice that we now have two completely separate workflows. That is a useful feature because we can run unrelated analyses at

the same time, which aids in testing and conditional application development.

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

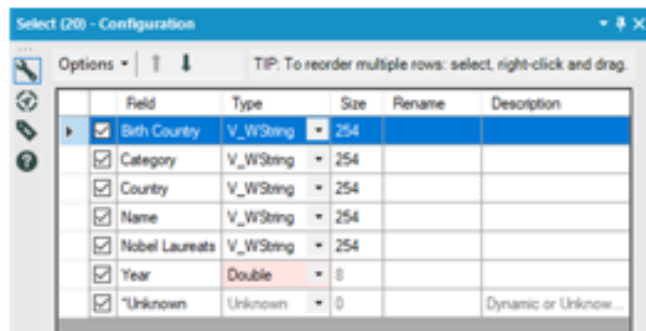


Results - Browse (20) - Input

6 of 6 Fields | Cell Viewer | 943 records displayed, 33 KB

Record #	Birth Country	Category	Country	Name	Nobel Laureate	Year
1	Germany	Austria Chemistry	Germany	Richard Kuhn	1	1938
2	Germany	Chemistr	Germany	Fritz Haber	1	1931
3	Germany	Chemistr	Germany	Carl Bosch	1	1931
4	Germany	Chemistr	Germany	Otto Hahn	1	1944
5	Argentina	Chemistry	Argentina	Luis Federico Leloir	1	1970
6	Australia	Chemistry	Australia	John Wilkinop Comforth	1	1975
7	Austria	Chemistry	Austria	Frideric Pregl	1	1937
8	Austria	Chemistry	Austria	Richard Adolf Zsigmondy	1	1925
9	Austria	Chemistry	Austria	Richard Kuhn	1	1938
10	Austria	Chemistry	Austria	Max F. Perutz	1	1962
11	Austria	Chemistry	Austria	Walter Koenig	1	1998

Figure-2-66-Brains vs. Brawn – Browse



Select (20) - Configuration

Options | TIP: To reorder multiple rows: select, right-click and drag.

Field	Type	Size	Rename	Description
<input checked="" type="checkbox"/> Birth Country	V_WString	254		
<input checked="" type="checkbox"/> Category	V_WString	254		
<input checked="" type="checkbox"/> Country	V_WString	254		
<input checked="" type="checkbox"/> Name	V_WString	254		
<input checked="" type="checkbox"/> Nobel Laureate	V_WString	254		
<input checked="" type="checkbox"/> Year	Double	8		
<input checked="" type="checkbox"/> *Unknown	Unknown	0		Dynamic or Unknow...

Figure-2-67-Brains vs. Brawn, Nobel Laureates – Select

As we can see, the data structure is similar to the medals file, and we can start our preparation for the join.

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

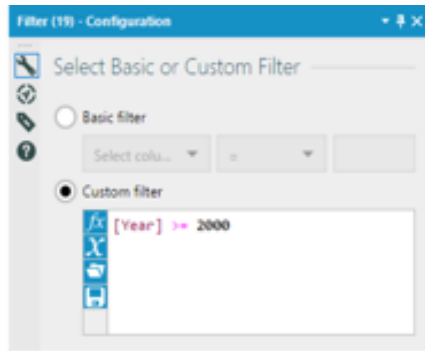


Figure-2-68-Brains vs. Brawn, Nobel Laureates – Filter

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

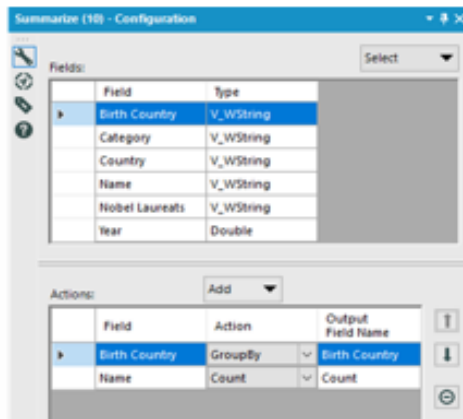


Figure-2-69-Brains vs. Brawn, Nobel Laureates –Summarize

Now we will sort the data in alphabetical order of the country of birth using the Sort tool.

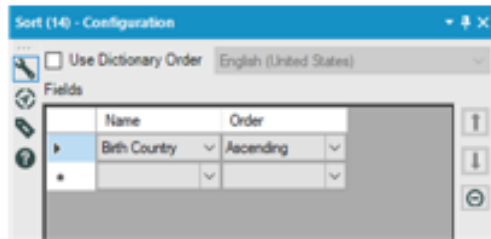


Figure-2-70-Brains vs. Brawn, Nobel Laureates – Sort

We will put a *Browse* tool after the *Sort* tool and see how the data looks.



Record #	Birth Country	Count
1	Australia	4
2	Austria	4
3	Bangladesh	2
4	Canada	2
5	China	2
6	Egypt	1
7	Finland	1
8	France	5
9	Germany	9
10	Ghana	1
11	Hong Kong	3
12	Hungary	3
13	India	2
14	Iran	3
15	Israel	4
16	Italy	4
17	Japan	10
18	Kenya	1
19	New Zealand	2

Figure-2-71-Brains vs. Brawn, Nobel Laureates – Browse after Sort

We see 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* using *Select* tool.

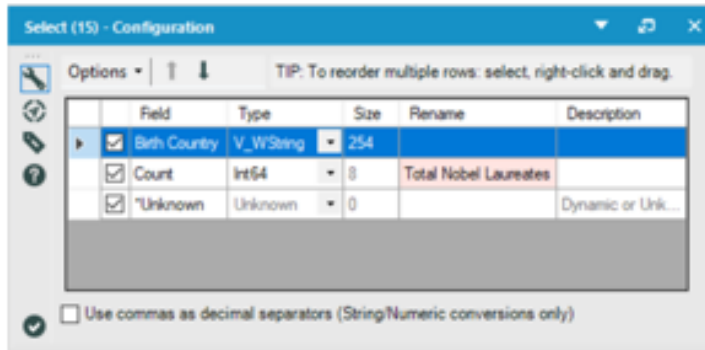


Figure-2-72-Brains vs. Brawn, Nobel Laureates – Rename Count using Select

We now have two data streams ready to be merged. We want to align the two data sets so matching countries from each of the data streams share the same record, which means we want to join the data. Since we do not want to lose any data points if we have countries in one data set but not the other, we will want to unite the three outputs from the join into a single data stream.

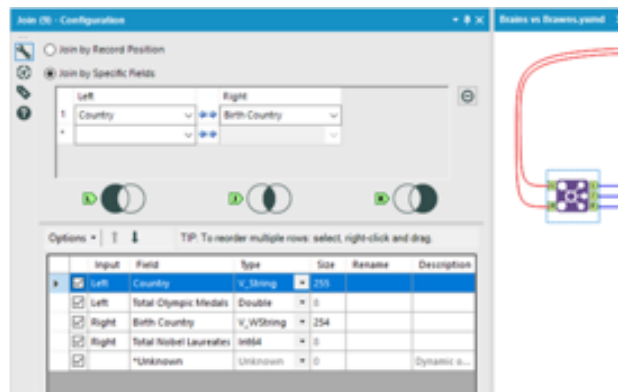


Figure-2-73-Brains vs. Brawn, Nobel Laureates – Join Configuration

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

In this instance, it's important we keep both of the 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*.

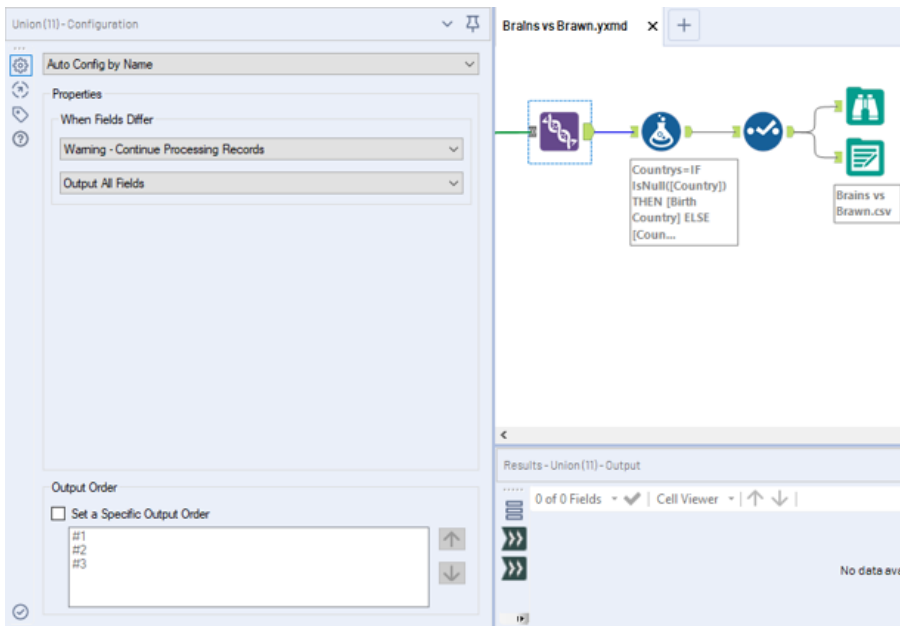
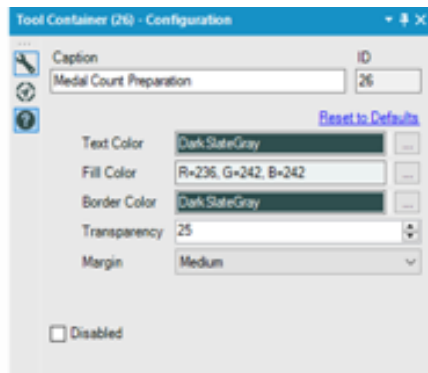


Figure-2-74-Brains vs. Brawn, Nobel Laurates – Union Configuration

Since we are doing a union of three output streams of a Join tool, we know we will have matching column names. That 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. That is to make sure the data looks the way we expect. Notice that we're doing that after the *Union* and not the *Join*. That's 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. Since this is a single step, we should check both the tools if any issue arises.

We are getting close to our goal; however, the data stream is also starting to become 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 following image.



**Figure-2-75-Brains vs. Brawn, Medal Count Preparation
– Tool Container Configuration**

We can now drag the appropriate tools into the tool containers so the data stream is easier to understand.

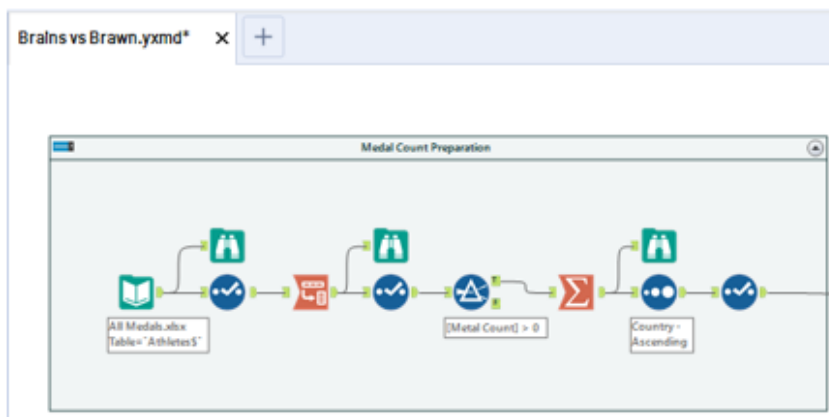


Figure-2-76-Brains vs. Brawn - Medal Count Preparation stream with Tool Container

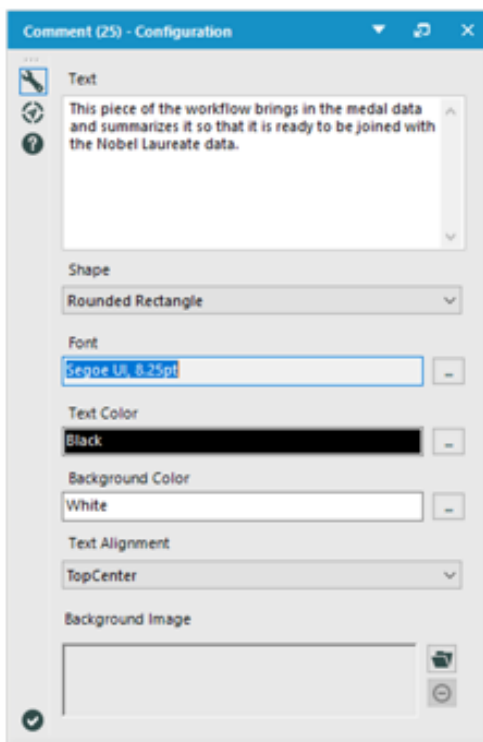


Figure-2-77-Brains vs. Brawn, Medal Count Preparation - Comment Configuration

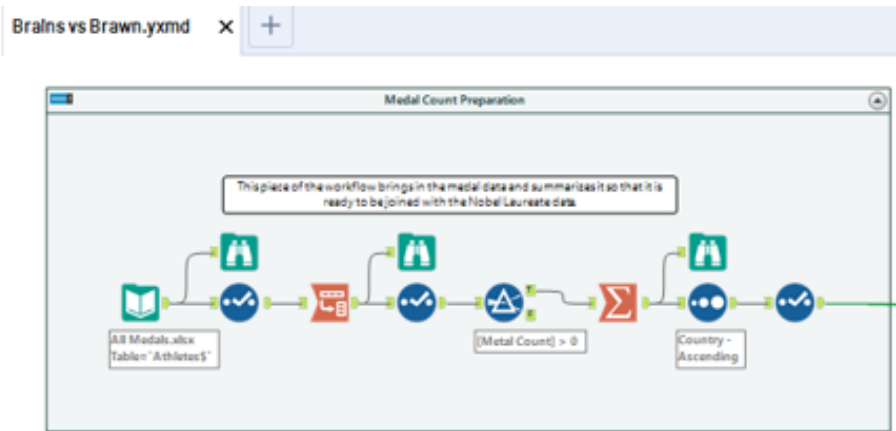


Figure-2-78-Brains vs. Brawn – Medal Count Preparation Comment

Looking at the data stream this way is helpful, but if we click on the arrows at the top right, we can condense what we are looking at.

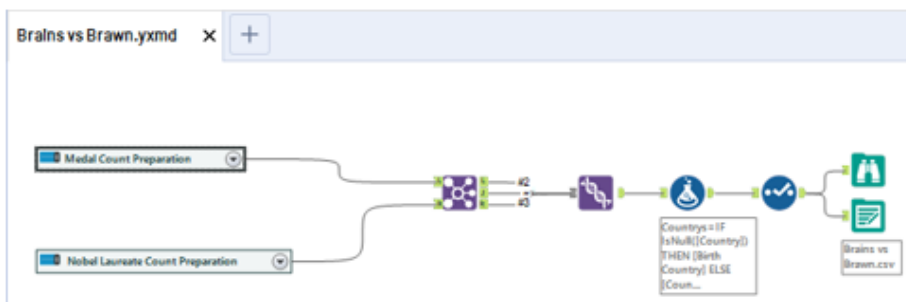
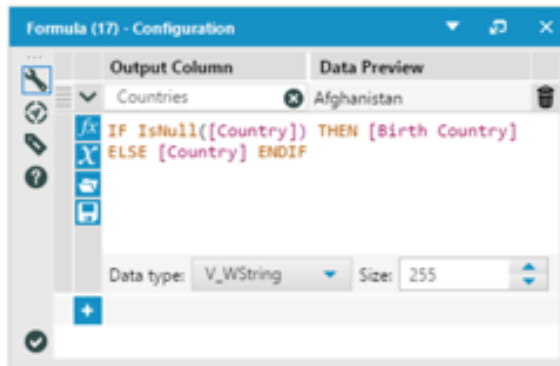


Figure-2-79-Brains vs. Brawn – Simplified Workflow

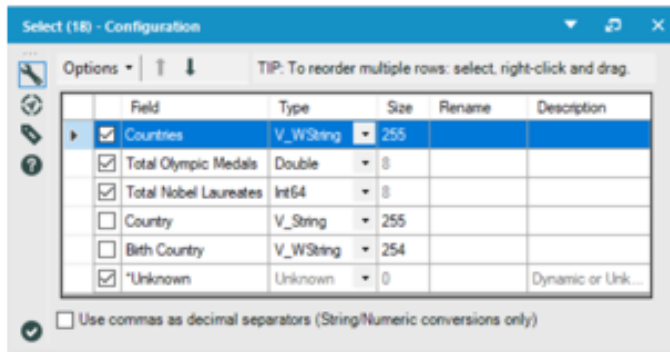
We now see the medal count preparation and the Nobel Laureate count preparation as two separate processes instead of a series of tools. Now that we have made the data stream easier to understand, we should finish building the workflow.

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 this formula: **IF IsNull([Country]) THEN [Birth Country] ELSE [Country] ENDIF**. That will take the Country value unless it is null, in which case it will take the Birth Country value.



**Figure-2-80-Brains vs. Brawn, Final Output
- Formula Configuration**

Now we only need to clean up the data and export it to a .csv file. Add a *Select* tool to move Countries to the top as dimensions are expected to the left, and uncheck Country and Birth Country. Then export the file to *Brains vs. Brawn.csv*.



**Figure-2-81-Brains vs. Brawn, Final Output
– Select Configuration**

The output of the entire workflow can be explored by adding a *Browse* tool after the *Select* tool. After running the workflow, the result will be as shown below:



Record #	Countries	Total Olympic Medals	Total Nobel Laureates
1	Alghanistan	2	[Null]
2	Algeria	8	[Null]
3	Argentina	141	[Null]
4	Armenia	10	[Null]
5	Australia	609	4
6	Austria	91	4
7	Azerbaijan	25	[Null]
8	Bahamas	24	[Null]
9	Bahrain	1	[Null]
10	Bangladesh	[Null]	2
11	Barbados	1	[Null]
12	Belarus	97	[Null]
13	Belgium	18	[Null]
14	Botswana	1	[Null]
15	Brazil	221	[Null]

**Figure-2-82-Brains vs. Brawn, Final Output –
Browse**

The final workflow is shown in the following figure:

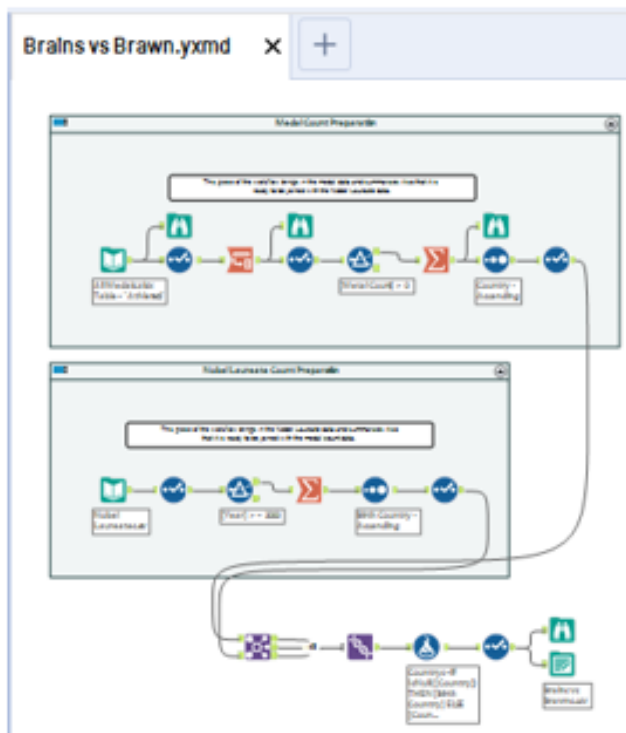



Figure-2-83-Brains vs. Brawn – Complete Workflow

2.6 How Are We Doing?

 Send	To...	Alteryx Consultants
	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

3.1 Tools and Concepts


3.1.1 Imputation

Concept: Handling Nulls

 <p>Figure-3-1-Imputation</p>	Tool Palette: Preparation
	Replaces a specific value in numeric fields. It is most commonly used to handle Null values in formulas. For more details, use the link below. bit.ly/2MFFTI6


3.1.2 Multi-Field Formula

Concept: Multiple Record Calculations

 <p>Figure-3-2-Multi-Field Formula</p>	Tool Palette: Preparation
	Provides the ability to create a formula that will be reused across multiple fields. For more details, use the link below. bit.ly/2LbmADy


3.1.3 Multi-Row Formula

Concept: Repetitive Calculations

 <p>Figure-3-3-Multi-Row Formula</p>	Tool Palette: Preparation
	Provides the ability to create a formula that will reference other records. For more details, use the link below. bit.ly/322VZ2j


3.1.4 Text Input

Concept: Ad Hoc Data


 <p>Figure-3-4-Text Input</p>	<p>Tool Palette: In/Out</p> <p>Allows the creation of data sets to be used by typing in the data. For more details, use the link below. bit.ly/2L9kd3X</p>
---	---

3.1.5 Data Cleansing

Concept: Problem Analysis

 <p>Figure-3-5-Data Cleansing</p>	<p>Tool Palette: Preparation</p> <p>Fixes common data quality issues using a variety of parameters. For more details, use the link below. bit.ly/2MF8Aou</p>
---	---

3.2 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's refer to the *Unisex Names.txt* file in the folder *Chapter 3 – Unisex Names.txt*. It is a tab-delimited file. We need to make sure the file pop-up window looks like the image represented here. Save the workflow as *Gender Swapped*.

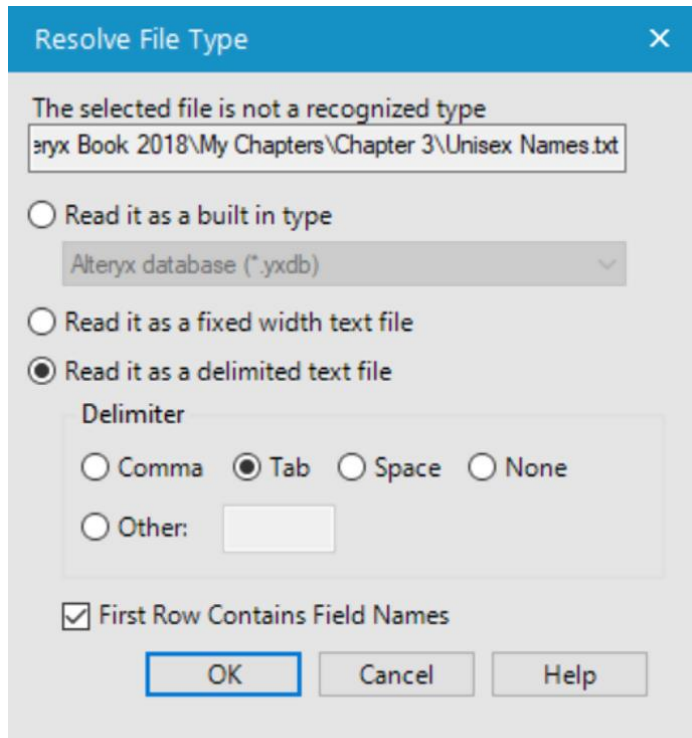


Figure-3-6 – Tab delineated pop-up window

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.

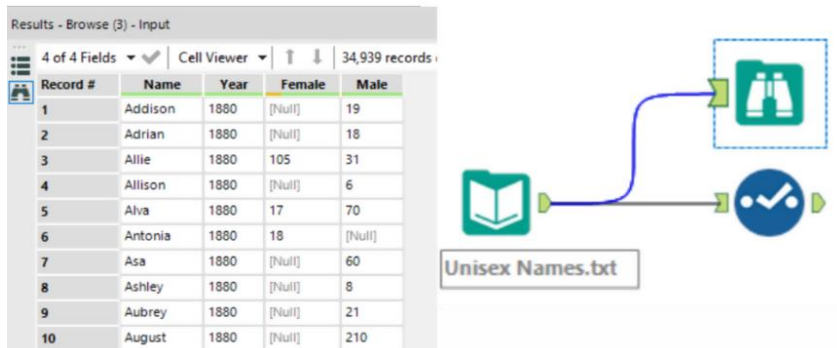


Figure-3-7-Gender Swapped Browse

If we look at the data, we will see that some numeric fields have *nulls* in them. Missing values are replaced with Null in Alteryx. 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 *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.

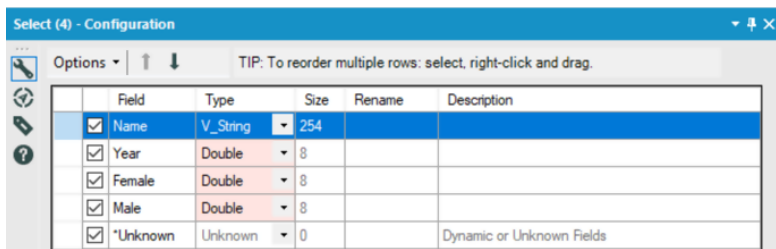


Figure-3-8 – Gender Swapped Select

If we add the *Imputation* tool with the settings as seen in the image, the tool will convert all the null fields in *Year*, *Female*, and *Male* fields to zeros.

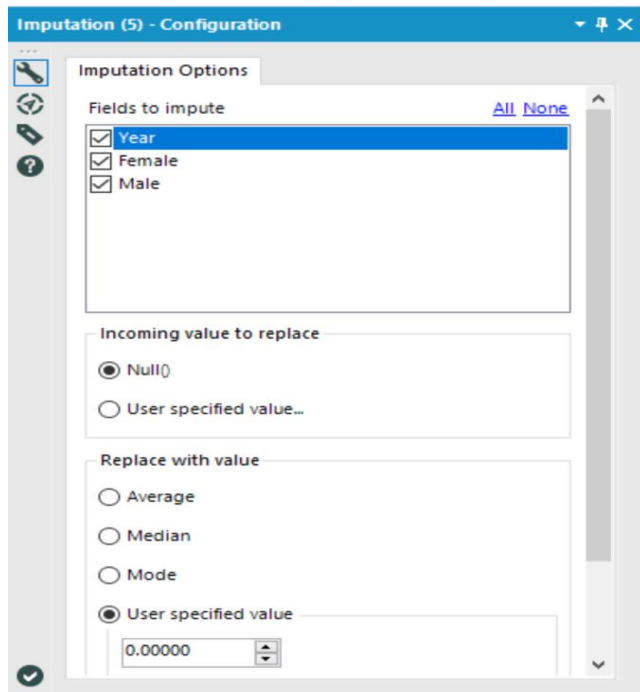


Figure-3-9-Gender Swapped Imputation Configuration

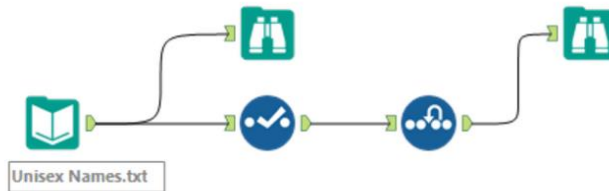
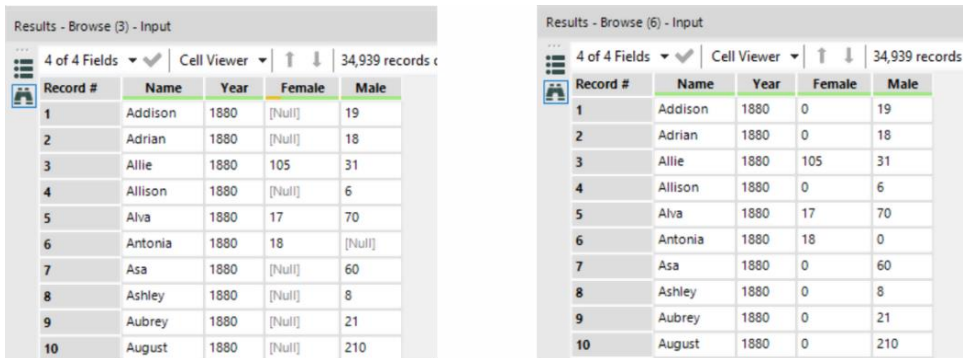


Figure-3-10-Gender Swapped data stream

Now let's see how the data has changed.



Record #	Name	Year	Female	Male
1	Addison	1880	[Null]	19
2	Adrian	1880	[Null]	18
3	Allie	1880	105	31
4	Allison	1880	[Null]	6
5	Alva	1880	17	70
6	Antonia	1880	18	[Null]
7	Asa	1880	[Null]	60
8	Ashley	1880	[Null]	8
9	Aubrey	1880	[Null]	21
10	August	1880	[Null]	210

Record #	Name	Year	Female	Male
1	Addison	1880	0	19
2	Adrian	1880	0	18
3	Allie	1880	105	31
4	Allison	1880	0	6
5	Alva	1880	17	70
6	Antonia	1880	18	0
7	Asa	1880	0	60
8	Ashley	1880	0	8
9	Aubrey	1880	0	21
10	August	1880	0	210

Figure-3-11- 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.



Figure-3-12 – Gender Swapped expression box

We have provided an alternative formula using `||` operator. We use `//` to comment lines in Alteryx.



Figure-3-13-Gender Swapped data stream

We have created a *flag* for the first and last 10 years for the data set. In order to do that, we will create a string field with *First 10 Years* or *Last 10 Years* in the field.

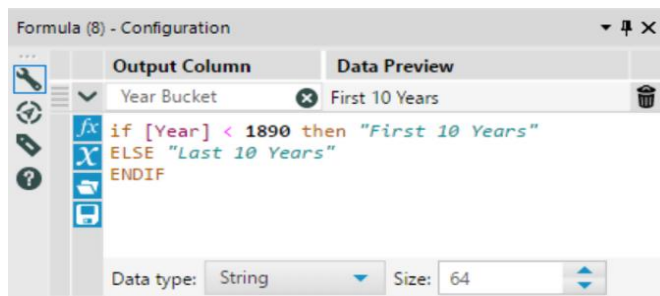


Figure-3-14-Gender Swapped Formula Configuration

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.

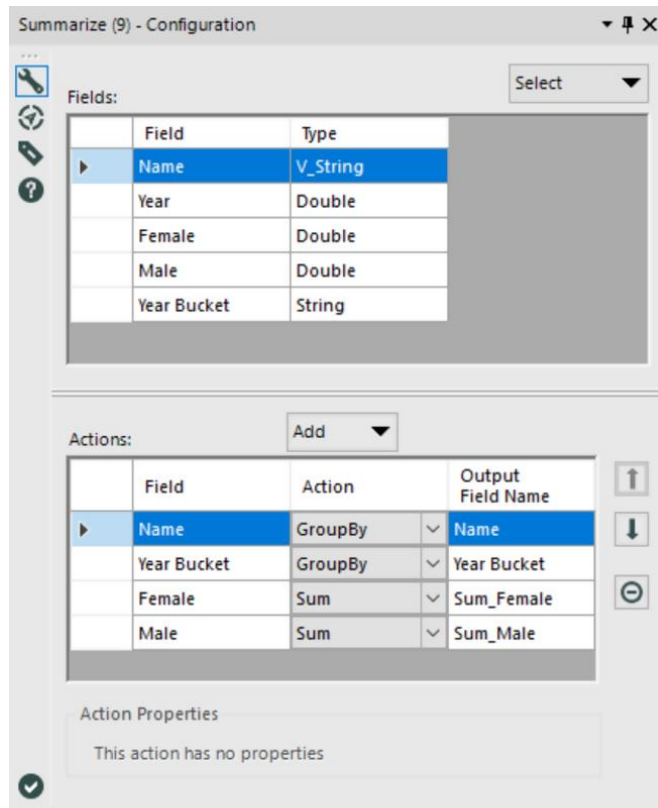


Figure-3-15- Gender Swapped Summarize Configuration

The data stream after *Summarize* looks like the image below:

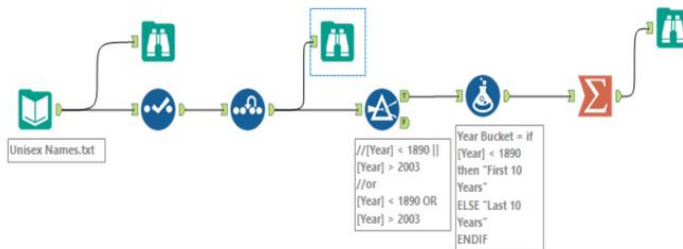
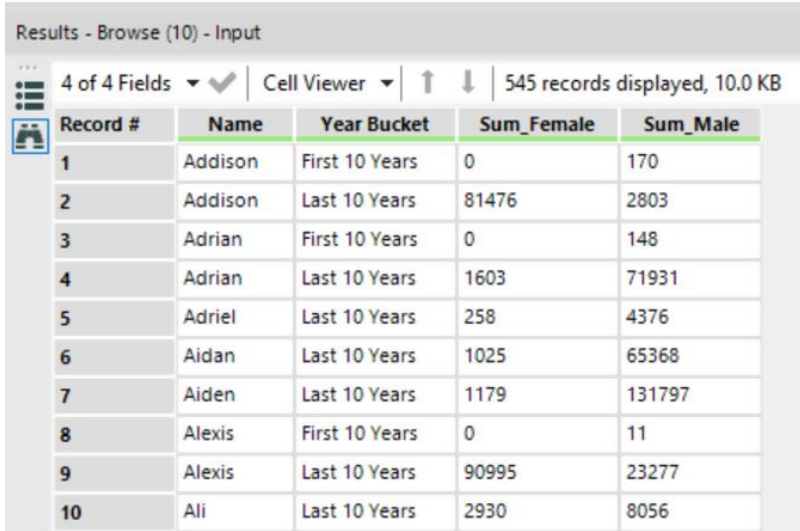


Figure-3-16- Gender Swapped data stream after summarize

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



Results - Browse (10) - Input

4 of 4 Fields | Cell Viewer | 545 records displayed, 10.0 KB

Record #	Name	Year Bucket	Sum_Female	Sum_Male
1	Addison	First 10 Years	0	170
2	Addison	Last 10 Years	81476	2803
3	Adrian	First 10 Years	0	148
4	Adrian	Last 10 Years	1603	71931
5	Adriel	Last 10 Years	258	4376
6	Aidan	Last 10 Years	1025	65368
7	Aiden	Last 10 Years	1179	131797
8	Alexis	First 10 Years	0	11
9	Alexis	Last 10 Years	90995	23277
10	Ali	Last 10 Years	2930	8056

Figure-3-17-Gender Swapped Browse Tool

From the *Browse* tool, we can see 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 that by using *Multiple Row Formula* to identify which records should be kept. Since we're using a Multi Row Formula, we need to make sure the data is in the right order. We will first use a *Sort* tool.

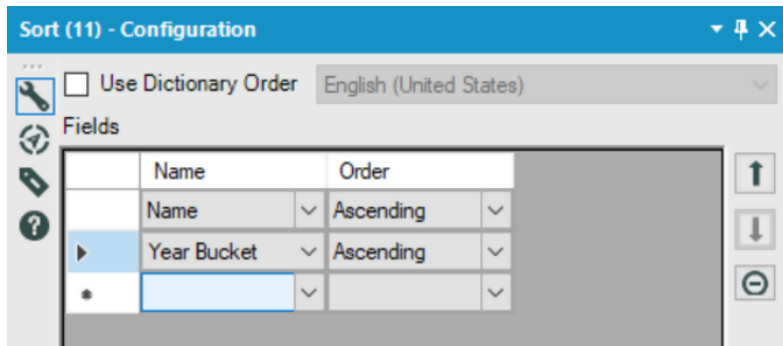


Figure-3-18-Gender Swapped Sort Configuration

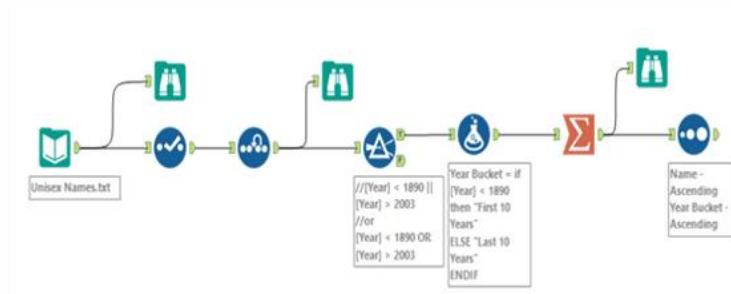


Figure-3-19-Gender Swapped data stream after adding sort tool

Now that we have made sure the data is sorted, we need to tackle a 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 doesn't 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's see how we can create that formula. At this point, much of the formula should be familiar.

We are looking at a conditional statement that tests two things, and if both are true, we write *Remove*. Otherwise, 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 that case, we are checking to see if the name of the current row (*[Name]*) equals the previous or following rows.

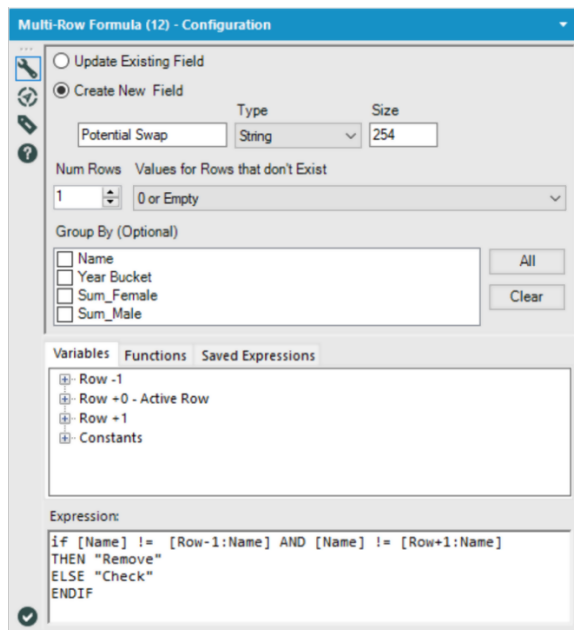


Figure-3-20-Gender Swapped Multi-Row Formula Configuration

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. That 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's 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 below:

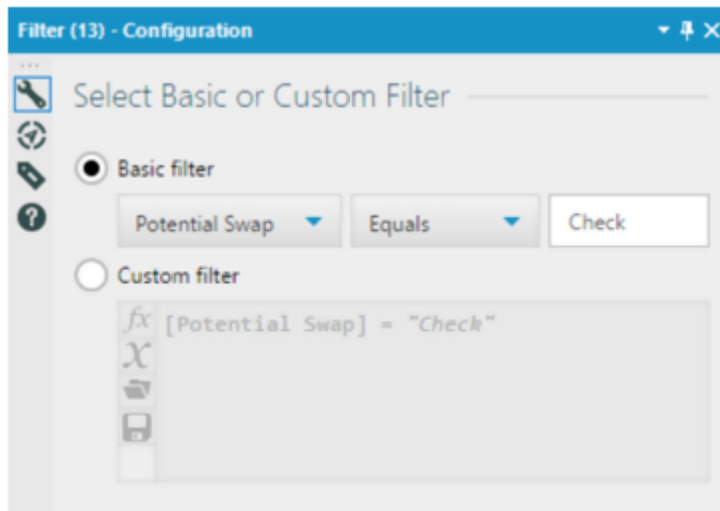


Figure-3-21-Gender Swapped Filter Configuration

The following table shows equations in the expression box.

```

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-3-22-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.

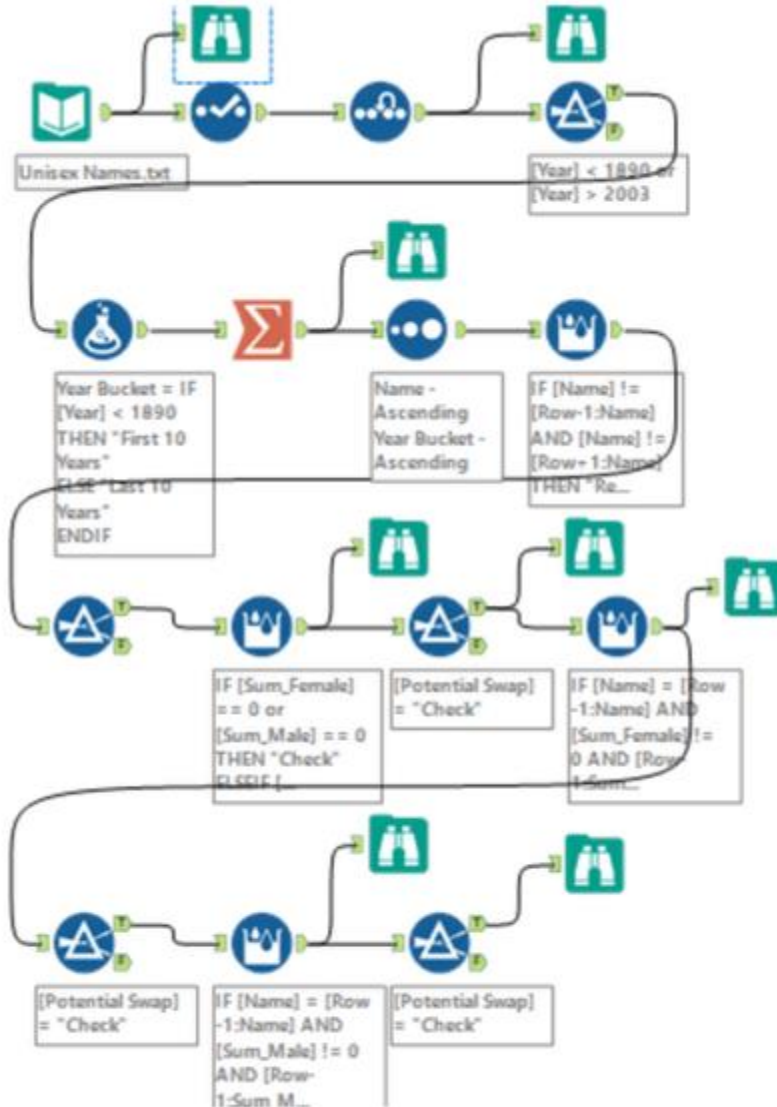


Figure-3-23-Gender Swapped Data Stream

Notice that a *Browse* tool has been added here to test for True or False conditions.



Results - Browse (15) - Input

5 of 5 Fields | Cell Viewer | 4 records displayed, 3015 bytes

Record #	Name	Year Bucket	Sum_Female	Sum_Male	Potential Swap
1	Beverly	First 10 Years	0	107	Check
2	Beverly	Last 10 Years	1472	0	Check
3	Hilary	First 10 Years	0	20	Check
4	Hilary	Last 10 Years	1284	0	Check

Figure-3-24-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 don't make a mistake. We can use *Summarize* by the *Name* field to get the results we're looking for.

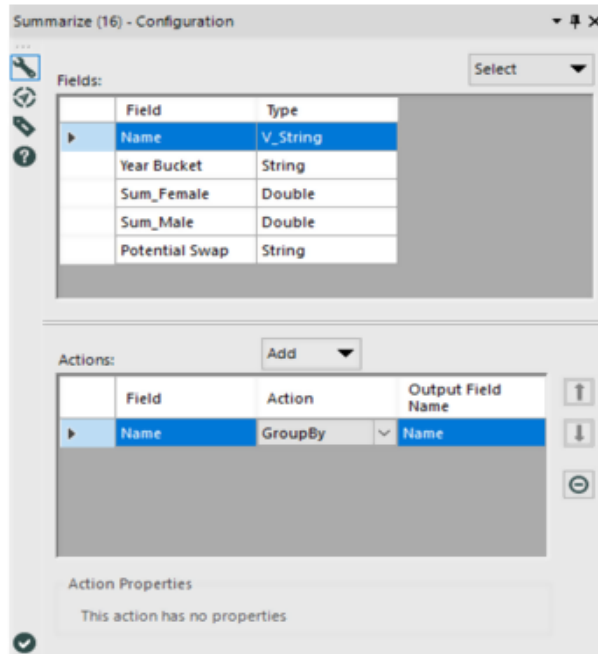


Figure-3-25-Gender Swapped Summarize Configuration

Running the *Browse* tool shows us the following:

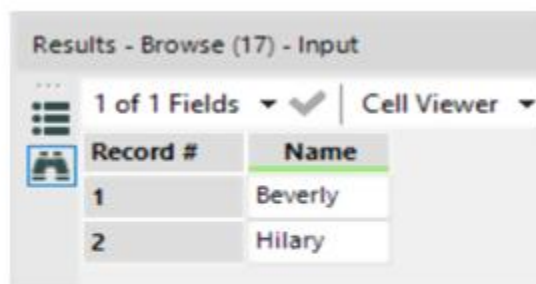


Figure-3-26-Gender Swapped Browse after Summarize

The *Gender Swapped* data stream should look like the image below when it is complete.

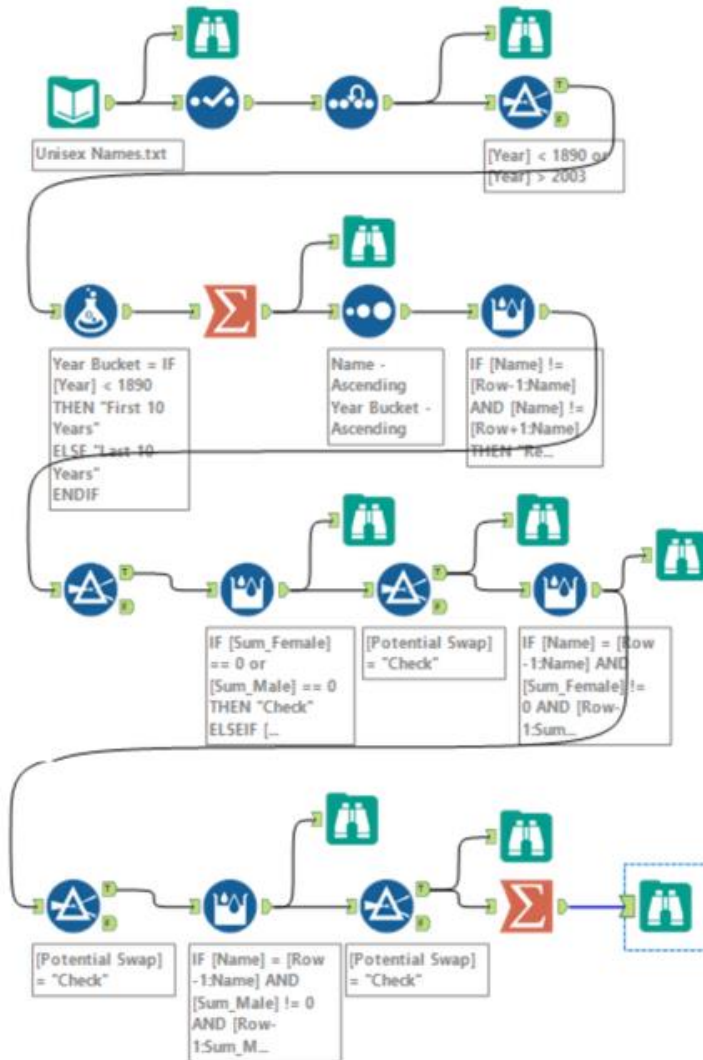


Figure-3-27-Gender Swapped Data Stream when complete

3.3 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 as follows:

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

We already know which issues we have in the data set, so we can copy the last data stream using the *Imputation* tool.

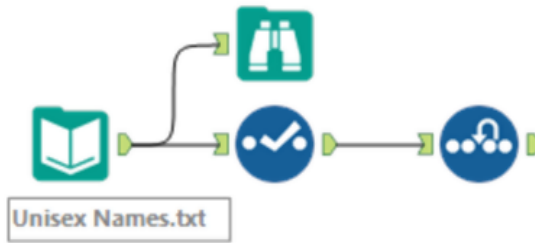


Figure-3-28-Reuse Gender Swapped for What about me

When *Text Input* is with a column called *Name*, we get the following:

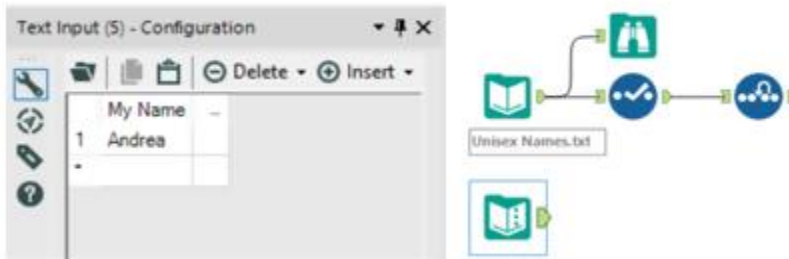


Figure-3-29-What about me Text Input

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

Unisex Names Select

Text Input Select

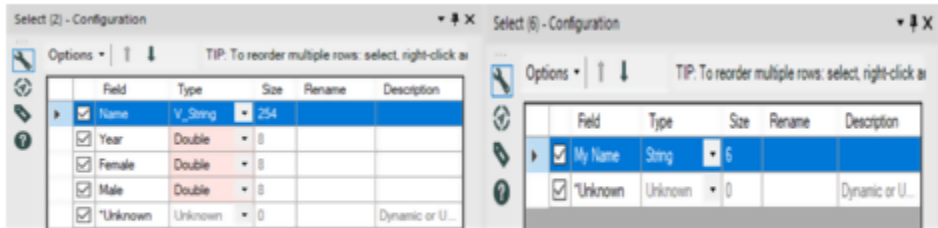


Figure-3-30-What about me Select Configuration

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 while 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*).

4 of 4 Fields ✓ | Cell Viewer | ↑ ↓

Record #	Name	Year	Female	Male
1	Andrea	1881	5	0
2	Andrea	1884	7	0
3	Andrea	1885	6	0
4	Andrea	1886	9	0
5	Andrea	1887	7	0
6	Andrea	1888	7	0
7	Andrea	1889	10	0
8	Andrea	1890	6	0
9	Andrea	1891	8	0
10	Andrea	1892	16	0

Figure-3-31-What about me Browse Configuration

We do not see that any records returned when we run using Browse tool. That is because the name entered in the *Text Input* does not match anything in the *Unisex Baby Names* data set. 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 percentage 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 that. 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 percentage of the total for Male and Female babies each year.

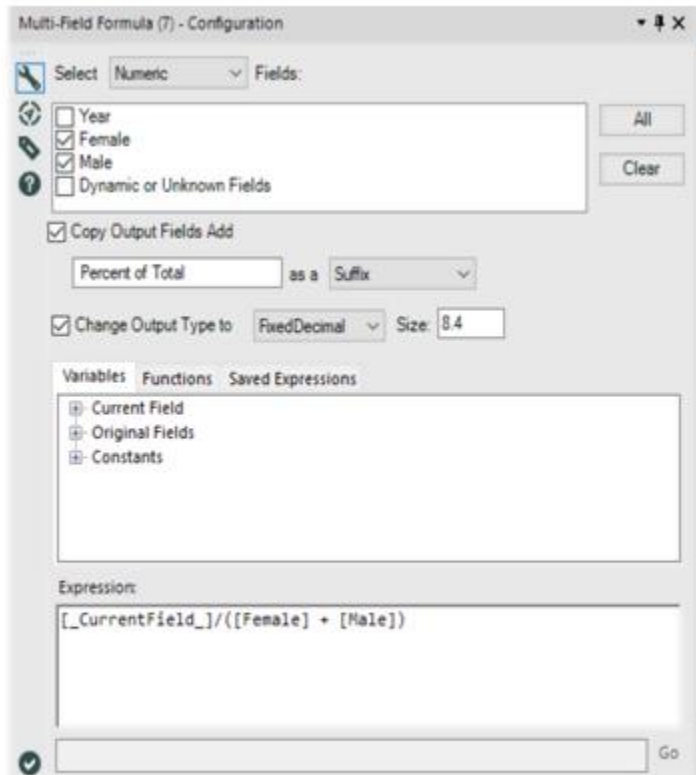


Figure-3-32-What about me Multi Field Formula Configuration

Results - Output Data (12) - Input

6 of 6 Fields | Cell Viewer | 131 records displayed

Record #	Name	Year	Female	Male	Female Percent of Total	Male Percent of Total
1	Andrea	1881	5	0	1.0000	0.0000
2	Andrea	1884	7	0	1.0000	0.0000
3	Andrea	1885	6	0	1.0000	0.0000
4	Andrea	1886	9	0	1.0000	0.0000
5	Andrea	1887	7	0	1.0000	0.0000
6	Andrea	1888	7	0	1.0000	0.0000
7	Andrea	1889	10	0	1.0000	0.0000
8	Andrea	1890	6	0	1.0000	0.0000

Figure-3-33-What about me-Output


CHAPTER 4

The Direct Approach

4.1 Tools and Concepts


4.1.1 Append Fields

Concept: Appending Data/Cartesian Product

 Figure-4-1-Append Fields	Tool Palette: Join
	<p>Adds each record from <i>S</i> (<i>source</i>) to the end of each record in <i>T</i> (<i>target</i>) providing a Cartesian product. For more details, use the link below. bit.ly/2TKjOZI</p>


4.1.2 Auto Field

Concept: Set Data Types

 Figure-4-2-Auto Field	Tool Palette: Preparation
	<p>Automatically sets the field to the smallest possible size and type to accommodate data in each string column. For more details, use the link below. bit.ly/2P6Fvoa</p>


4.1.3 Date Time Now

Concept: Current Date/Time

 Figure-4-3-Date Time Now	Tool Palette: In/Out
	<p>Gets the system time when the module starts executing. For more details, use the link below. bit.ly/2L89IOD</p>


4.1.4 Date Time

Concept: Working with Dates

 Figure-4-4-DateTime	Tool Palette: Parse
	Converts between String and Date format fields. For more details, use the link below. bit.ly/2KHEnTB


4.1.5 Directory

Concept: Working with Directories

 Figure-4-5-Directory	Tool Palette: In/Out
	Creates a data stream that has the contents of a Directory or Folder. For more details, use the link below. bit.ly/2HfWgqw

4.1.6 Visual Layout

Concept: Visualize Reports

 Figure-4-6 Visual Layout	Visual Layout Tool-Reporting
	Creates an output by bringing reporting tools together. For more details, use the link below. bit.ly/2KkqM4O

4.2 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 use 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 – What's the Policy on That?* Save the folder and workflow at the same location.

The plan is to do the following:

- Bring in a list of all of the *PDF* files in the directory.
- Get the current date.
- Tag the files to form three buckets:
 - a. Files created in last 30 days from today (current date).
 - b. Files created 30 to 90 days ago before today.
 - c. Files created prior to 90 days before today.

- Export the data to *New Policies.xlsx*.

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 the subfolders (subdirectories), we need to check the box at the bottom of the configuration window.



Figure-4-6-Directory Configuration

The *Directory Configuration* window has the following three components:

- *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 we've navigated to.

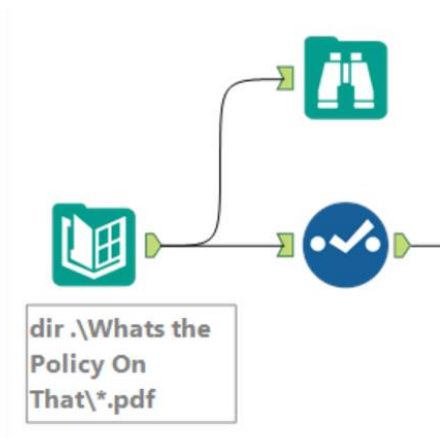


Figure-4-7-Policy Data Stream

As usual, we will add a *Browse* and *Select* tool following the data connection. Let's look at the *Select* tool to see what's in the data stream.

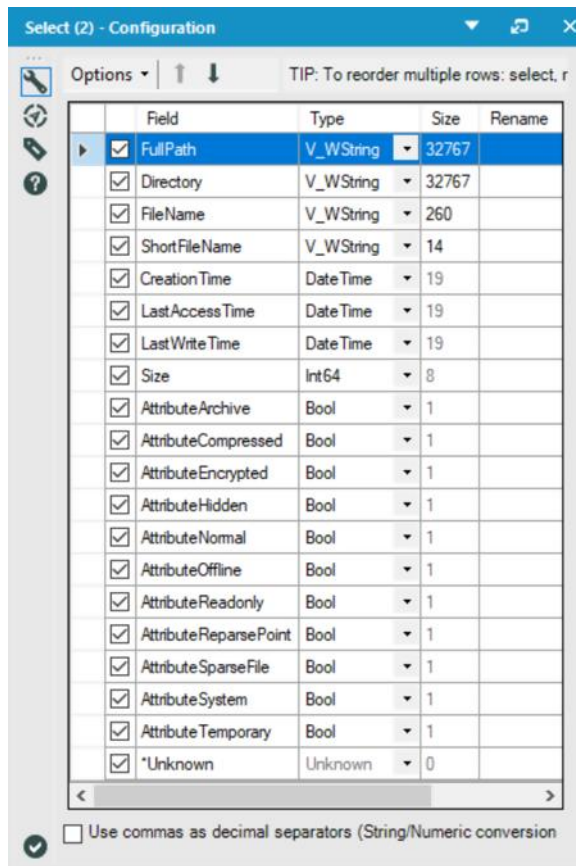


Figure-4-8-Select Configuration

We can see that 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*, *Directory*, *FileName*, and *CreationTime* fields. Now that we have the appropriate information from the list of files, the next thing we need to do is add the *current date* to the workflow. That is

where the *Date Time Now* tool comes in. Please note 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. The value is created as String data type. We need to convert to a Date data type using the *Date Time* tool.

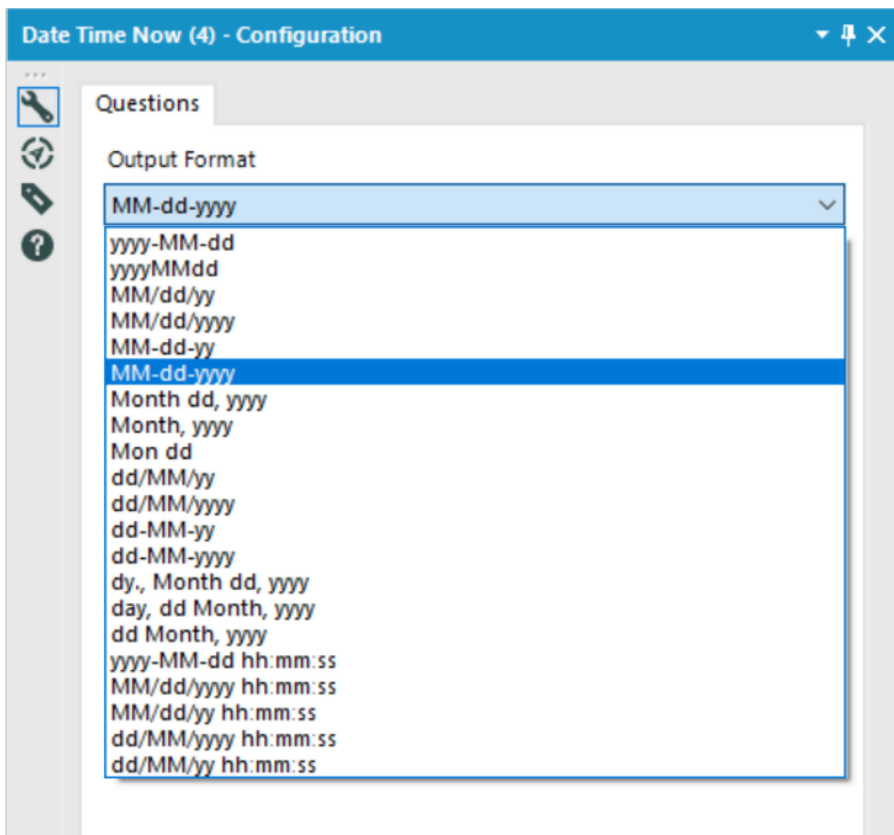


Figure-4-9-Date Time Now Configuration

The *Date Time Now Configuration* window allows us to select the format of the string required in the output.

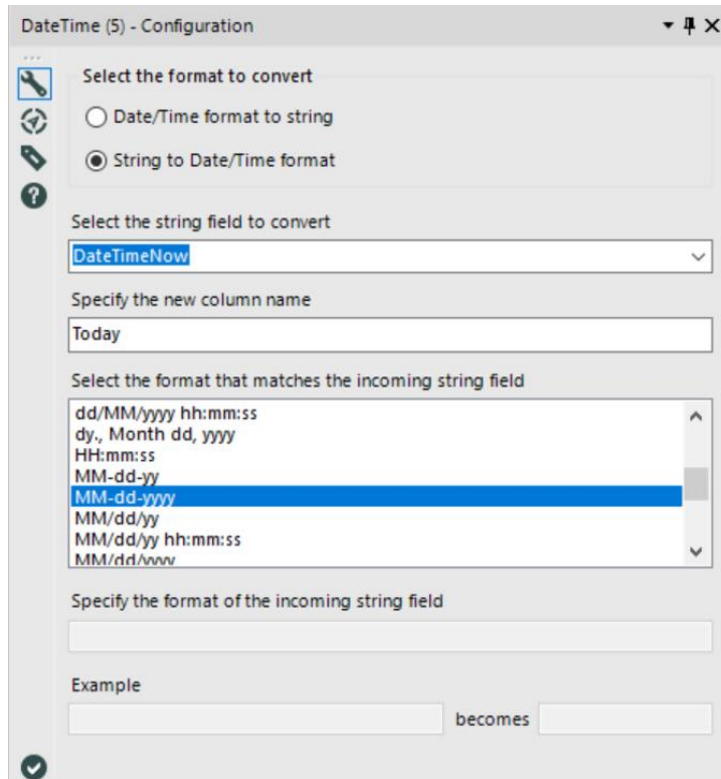


Figure-4-10-Date Time Configuration

The *Date Time Configuration* window has the following four components:

- *Conversion Mode* allows for conversion 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're creating.

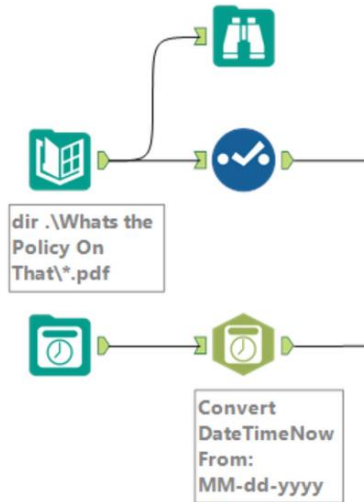


Figure-4-11-Policy Data Stream

Simultaneously, the *Auto Field* tool is being used to optimize the field lengths for fields that are String data type. We can also use the *Select* tool to manually change the data type, but *Auto Field* is used as a best practice here.

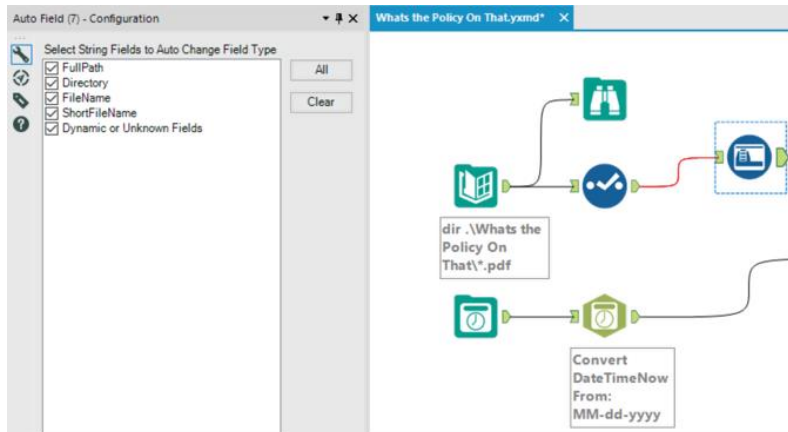


Figure-4-12-Auto Field Configuration

The *Auto Field configuration* window has only one setting. The *Select-String Fields to Auto Change Field Type* section allows us to select the columns for which optimization of field type and size is required.

Now that we have these two data streams ready to be combined, the question becomes this: 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 data set. A union will not work.

A *Join* would allow us to do that 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 will connect the list of *PDFs* to the *Target (T)* input and the date to the *Source (S)* input. When we test with a

Browse tool, the *Today* field would have been added as an extra column to every row, making it a Cartesian product.

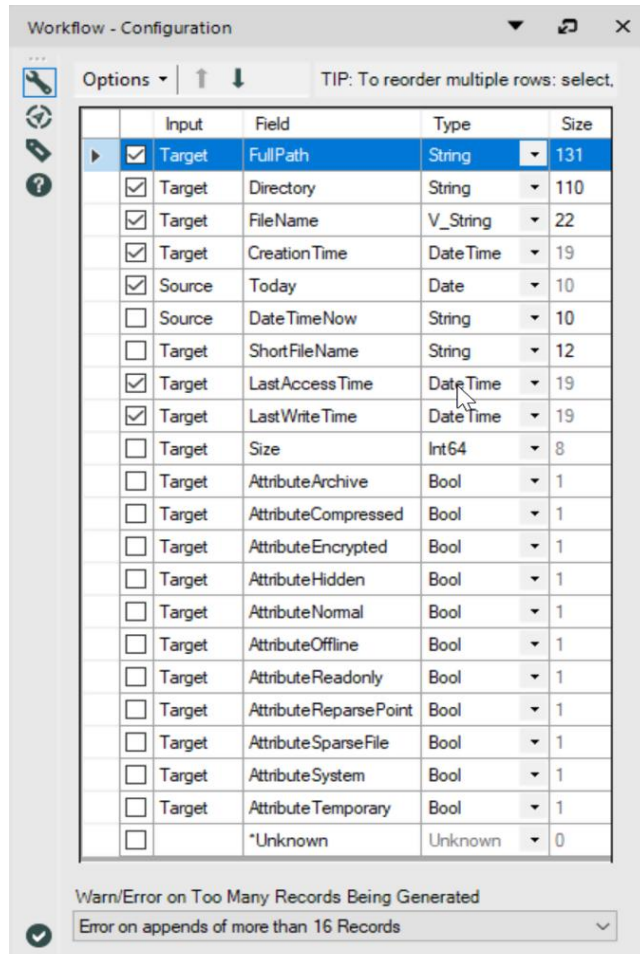


Figure-4-13-Append Field Configuration

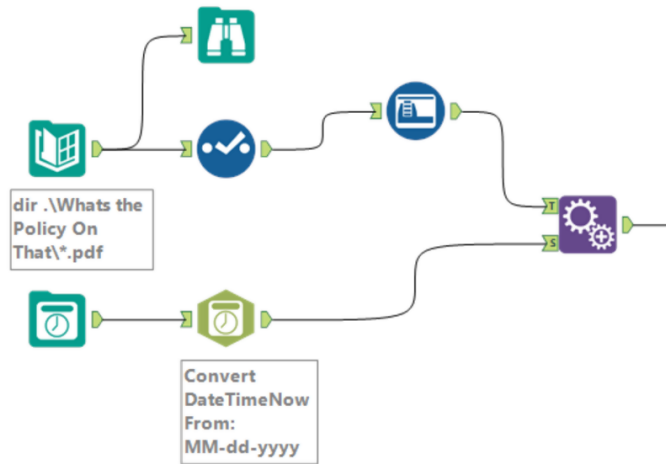


Figure-4-14-Policy Data stream

The *Append Fields Configuration* window is identical to the *Select Configuration* window, with these two exceptions:

- 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.

In order to create buckets, we need a *Formula* tool to create a new field called *Bucket*. That is done using the following formula:

```
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 *Bucket* will logically divide the data based on provided conditions.

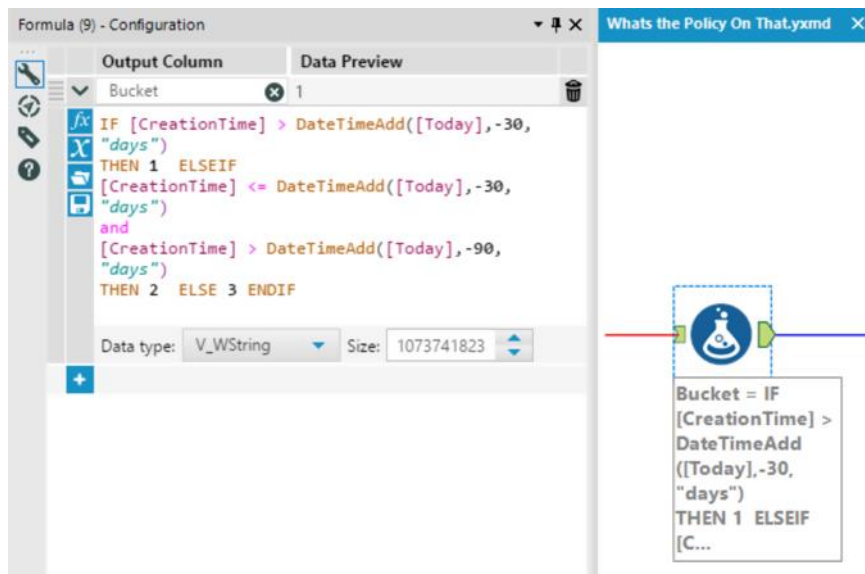


Figure-4-15-Formula Configuration

Then we use a *Filter* tool to restrict that data with *Bucket* value equal to 1.

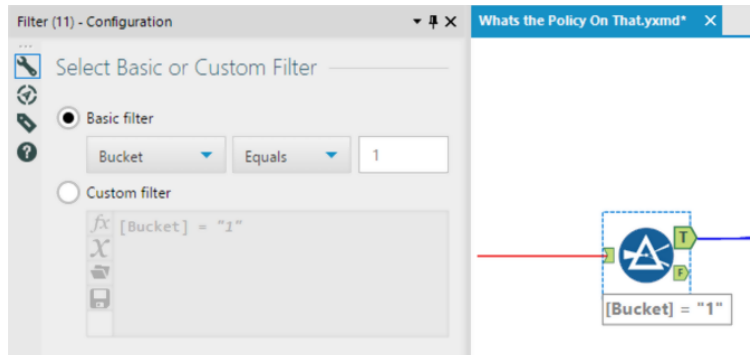


Figure-4-16-Filter Configuration

The last thing is to output the data to a file called *New Policies.xlsx* with a new sheet. The process is set up in such a way that every time the workflow is executed, it deletes the contents of the existing sheet and overwrites with new values for the current date (based on the system time).

Make sure the workflow is saved in the same location where the *What's the Policy on That* folder is saved.

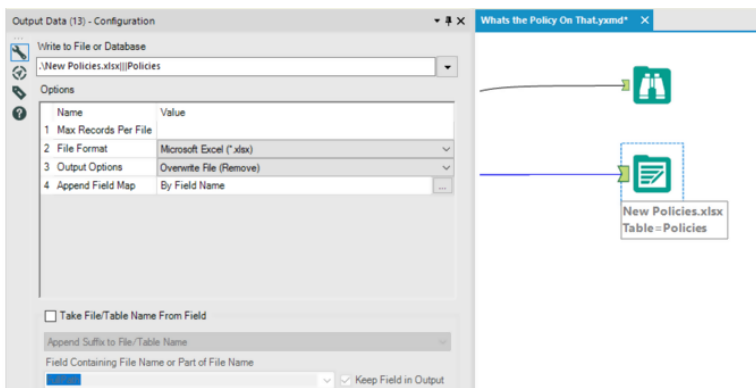


Figure-4-17-Output Data Configuration

The *What's the Policy on That?* data stream looks like the one below when it's complete.

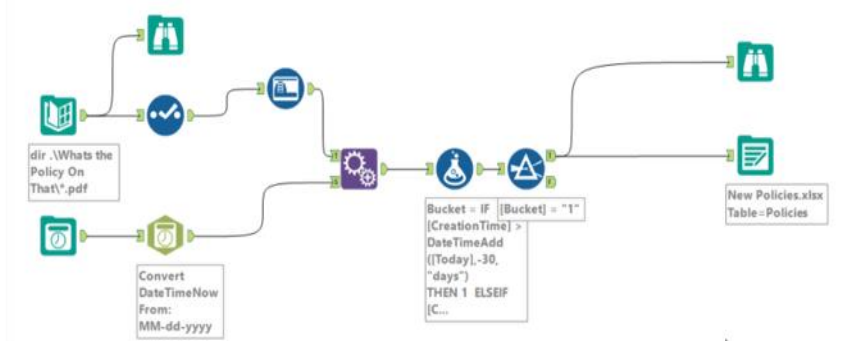


Figure-4-18-Policy Data Stream After Completion

4.3 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 Musing

5.1 Tools and Concepts


5.1.1 Allocate Input

Concept: Using Census Data

 <p>Figure-5-1-Allocate Input</p>	<p>Tool Palette: Demographic Analysis</p> <p>Gets input from demographic data using specific packages. For more details, use the link below. bit.ly/2zbWavC</p>
--	---


5.1.2 Find and Replace

Concept: Data Mapping

 <p>Figure-5-2-Find and Replace</p>	<p>Tool Palette: Join</p> <p>Replaces information in a data stream by entering the <i>F</i> (find) (target) input with information that matches in the <i>R</i> (replace) (source) input. For more details, use the link below. bit.ly/2KUxNbx</p>
--	--


5.1.3 Join Multiple

Concept: Complex Joins

 <p>Figure-5-3-Join Multiple</p>	<p>Tool Palette: Join</p> <p>Performs a full outer join or an inner join between data streams sharing a set of key fields. For more details, use the link below. bit.ly/2KOxxdO</p>
---	---


5.1.4 Text to Columns

Concept: Splitting field members at delimiters

 <p>Figure-5-4-Text To Columns</p>	<p>Tool Palette: Parse</p> <p>Breaks string fields into multiple string fields based on a delimiter. For more details, use the link below. bit.ly/2KTEhqV</p>
---	--


5.1.5 Random % Sample

Concept: Generate a random number or % of record

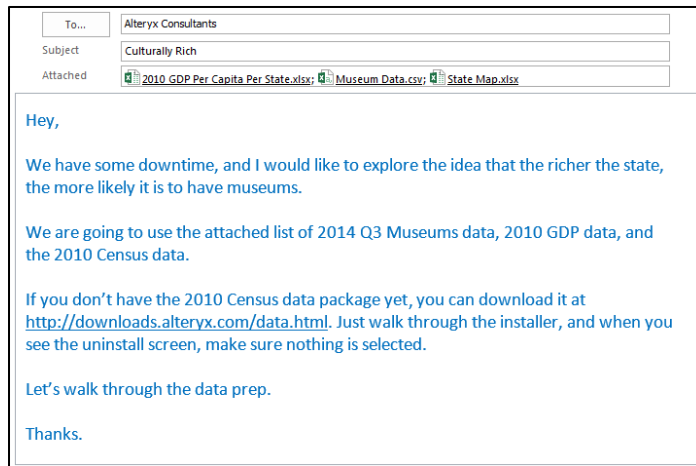
 <p>Figure-5-6-Random % Sample</p>	<p>Tool Palette: Preparation</p> <p>Returns an expected number of records resulting in a random sample of the incoming data stream. For more details, use the link below. bit.ly/2NfKIHL</p>
---	---

5.1.6 XML Parse

Concept: Read and Parse the XML snippet

 <p>Figure-5-6-XML Parse</p>	<p>Tool Palette: Parse</p> <p>Reads in XML snippets and parses them into individual fields. For more details, use the link below. bit.ly/2P2DaKU</p>
---	---

5.2 Culturally Rich



The files required for this chapter have been placed in clearly marked subfolders in *Chapter 5 – Culturally Rich*. We will start by bringing in each of the four data sources to see what we have. Let's open all three files and then connect to the *Census* data.

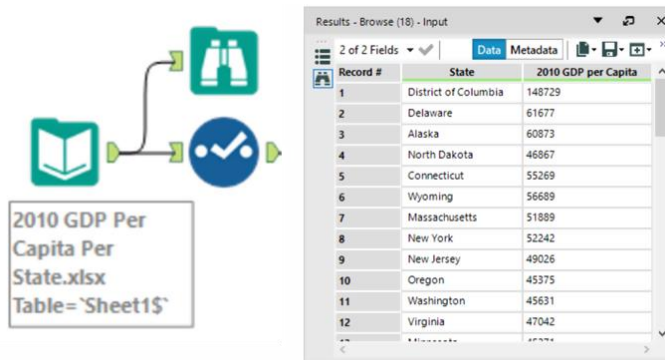


Figure-5-7 – Culturally Rich – GDP Data

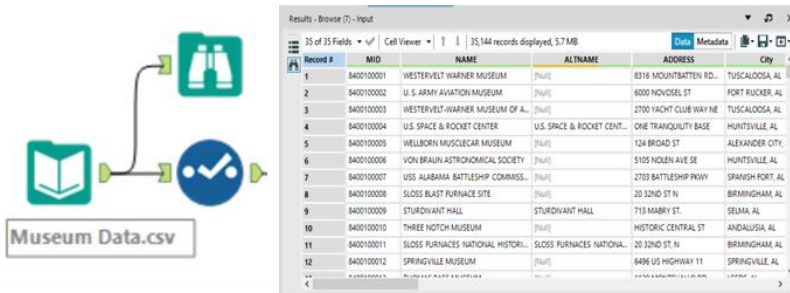


Figure-5-8 – Culturally Rich – Museum Data

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 in the city field.

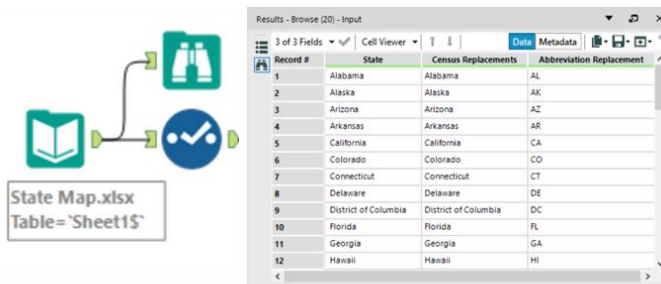


Figure-5-9 – Culturally Rich – State Map

However, the third file has three separate columns with either the complete state name or its abbreviation. That 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.

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's now bring in the *Census* data in order to plan what needs to be done with that data stream.

We will use an *Allocate Input* tool on the canvas and then choose the *U.S. Census 2010 – Most Recent Vintage* data set. Select the *States* option under *Pick Geography* and click the checkbox to select all states.

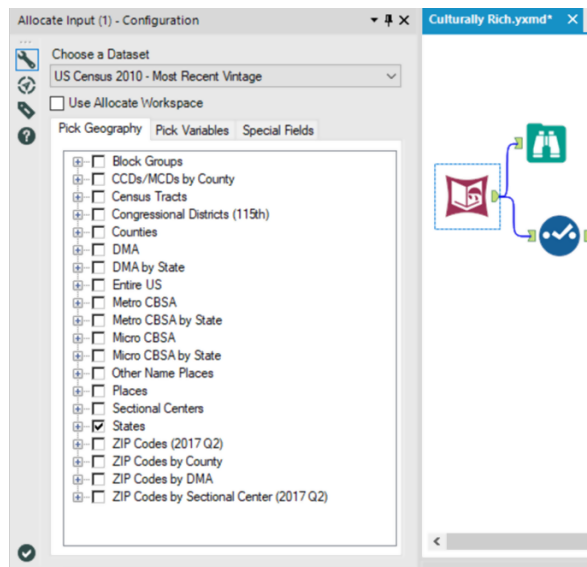


Figure-5-10 – Culturally Rich
 Allocate Input Configuration – Pick Geography

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.

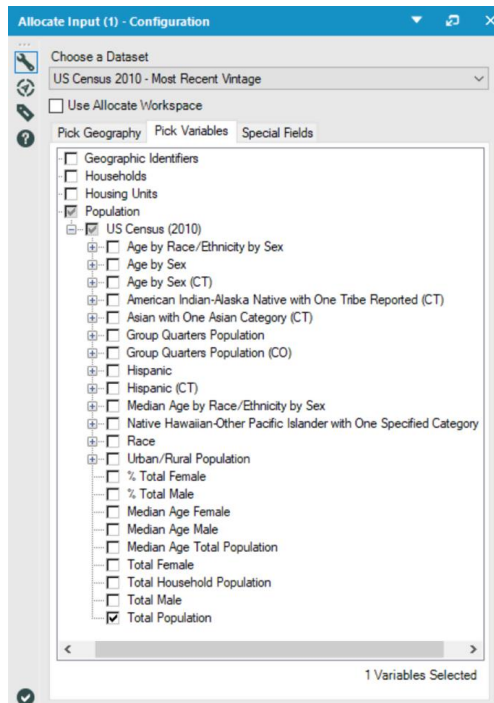


Figure-5-11 – Culturally Rich
 Allocate Input Configuration – Pick Variables

Under *Population* and *U.S. Census (2010)*, the last option is *Total Population*. That signifies the total number of people who have lived in each chosen state (Geography).

Since we need only that information, click on the *Special Fields* tab (as shown in the image below), and uncheck all options.

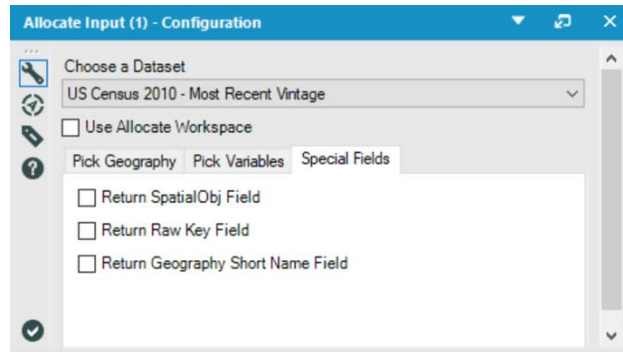
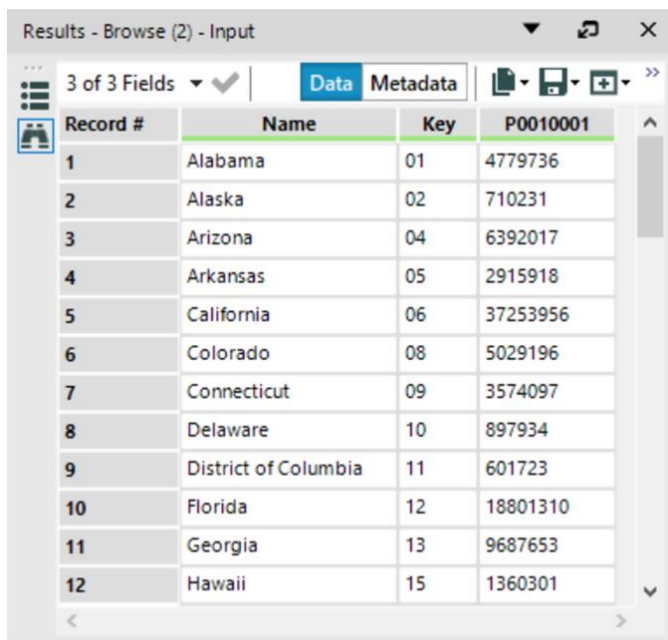


Figure-5-12 – Culturally Rich
 Allocate Input Configuration – Special Fields

Now that we have started with the final data stream, let's take a look at the data that comes out of the *Census* data connection.



Record #	Name	Key	P0010001
1	Alabama	01	4779736
2	Alaska	02	710231
3	Arizona	04	6392017
4	Arkansas	05	2915918
5	California	06	37253956
6	Colorado	08	5029196
7	Connecticut	09	3574097
8	Delaware	10	897934
9	District of Columbia	11	601723
10	Florida	12	18801310
11	Georgia	13	9687653
12	Hawaii	15	1360301

Figure-5-13 – 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. That is because the *Key* field uniquely identifies all geographies so 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*. That 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:

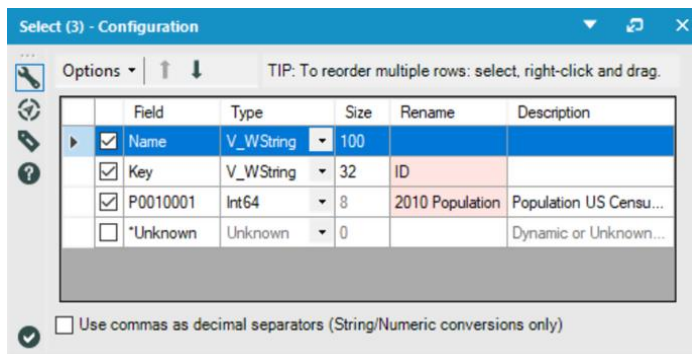


Figure-5-14 – Culturally Rich
 Select Configuration

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's start with mapping the *Census* data since it requires only one step before the join. In order to do that, we will use the *Find Replace* tool. Connect the *Census* data stream to the *Find (F)* input and the mapping data stream to the *Replace (R)* input, as shown in the following image:

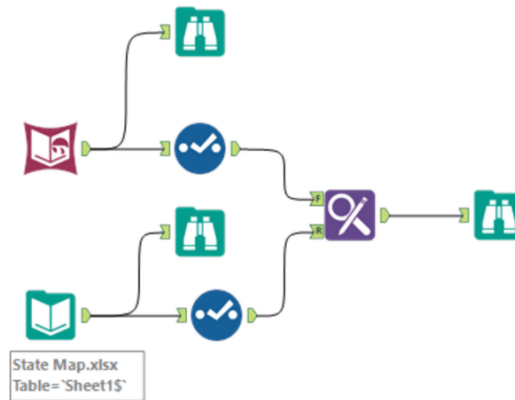


Figure-5-15 – Culturally Rich
 Connecting census data to Find and Status Map data to Replace

Now we'll take a look at the settings of the *Find Replace* tool.

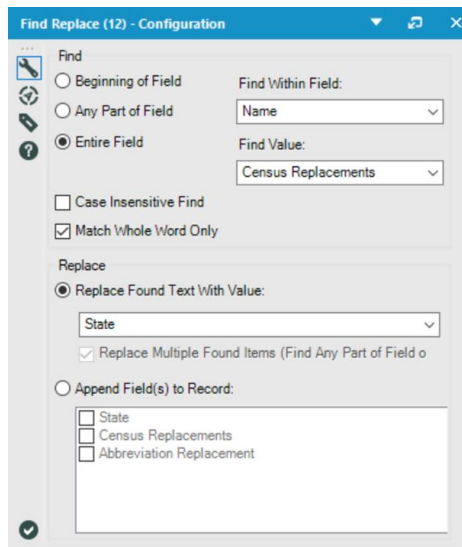


Figure-5-16 – Culturally Rich
 Census first Find Replace Configuration

Since the map has been structured in such a way that the field *Census Replacements* has all 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 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's get the *Museums* data ready.

We will first remove all the fields except *Name* and *City* in the *Select* tool following the *Input Data* tool. Then we will use the *Random % Sample* tool with the following settings to randomly retrieve 50% of the *Museums* data.

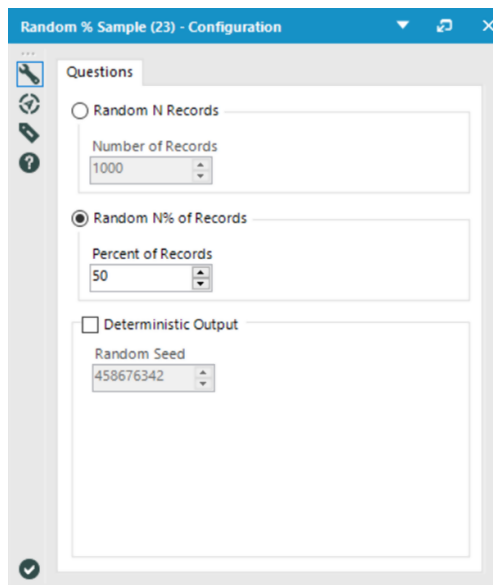


Figure-5-17 – Culturally Rich
Random % Sample Configuration

Next, we will split the data. To do that, use the *Text To Columns* tool and add it to the end of the *Museums* data stream with the following settings:

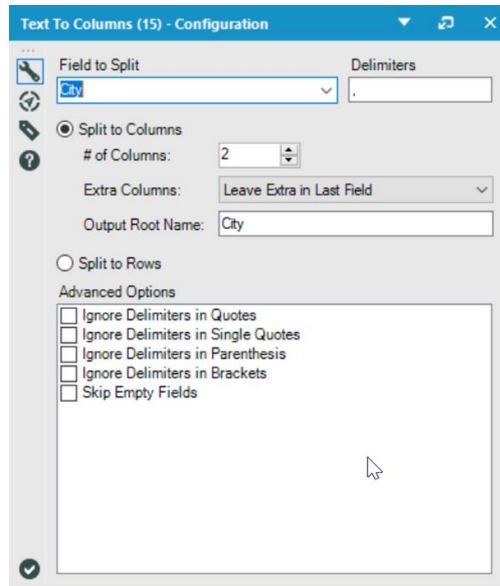
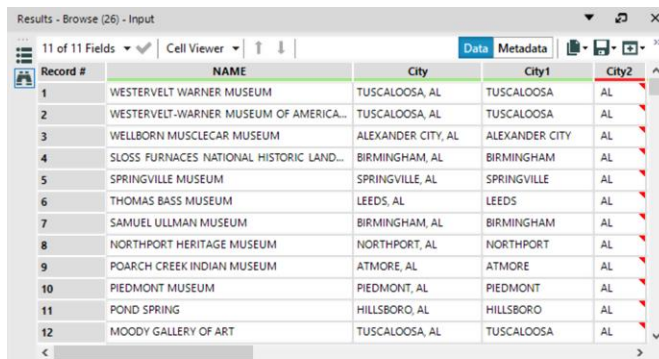


Figure-5-18 – Culturally Rich
 Text to Columns Configuration to split at a separator

Then, let's look at the updated data after it has been modified with the *Text to Columns* field.



Record #	NAME	City	City1	City2
1	WESTERVELT WARNER MUSEUM	TUSCALOOSA, AL	TUSCALOOSA	AL
2	WESTERVELT-WARNER MUSEUM OF AMERICA...	TUSCALOOSA, AL	TUSCALOOSA	AL
3	WELLBORN MUSCLECAR MUSEUM	ALEXANDER CITY, AL	ALEXANDER CITY	AL
4	SLOSS FURNACES NATIONAL HISTORIC LAND...	BIRMINGHAM, AL	BIRMINGHAM	AL
5	SPRINGVILLE MUSEUM	SPRINGVILLE, AL	SPRINGVILLE	AL
6	THOMAS BASS MUSEUM	LEEDS, AL	LEEDS	AL
7	SAMUEL ULLMAN MUSEUM	BIRMINGHAM, AL	BIRMINGHAM	AL
8	NORTHPORT HERITAGE MUSEUM	NORTHPORT, AL	NORTHPORT	AL
9	POARCH CREEK INDIAN MUSEUM	ATMORE, AL	ATMORE	AL
10	PIEDMONT MUSEUM	PIEDMONT, AL	PIEDMONT	AL
11	POND SPRING	HILLSBORO, AL	HILLSBORO	AL
12	MOODY GALLERY OF ART	TUSCALOOSA, AL	TUSCALOOSA	AL

Figure-5-19 – Culturally Rich
 Modified Museum Data

When we look at the *Browse* tool, we can see that *City2* has a red triangle at the top-right corner of the cell. It's 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 text box pop-up that reads “This cell has leading spaces.” That means the first character of the cell is a space (attributed to the split and a spacer after the “,” delimiter) and that we should address it. We will do this by using the *Formula* tool to create a new field, which we will call *State*.

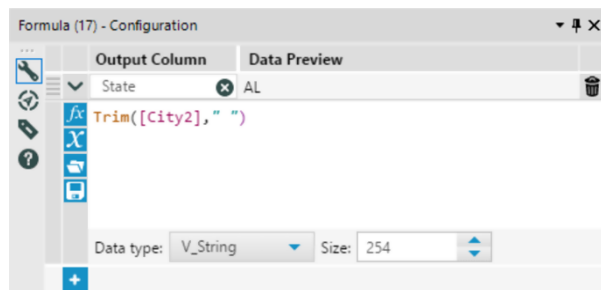


Figure-5-20- Culturally Rich
 Museum data using Formula

This time, we will be creating a string. If we look at the *String* functions below, 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 with trailing spaces that would cause issues with mapping.

Since we are removing spaces from the field, we could simply write the formula *Trim([City2])* 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([City2], " ")*.

Then, add a *Select* tool to rename *City1* to *City* and uncheck the *City2* field to remove it from the data stream.

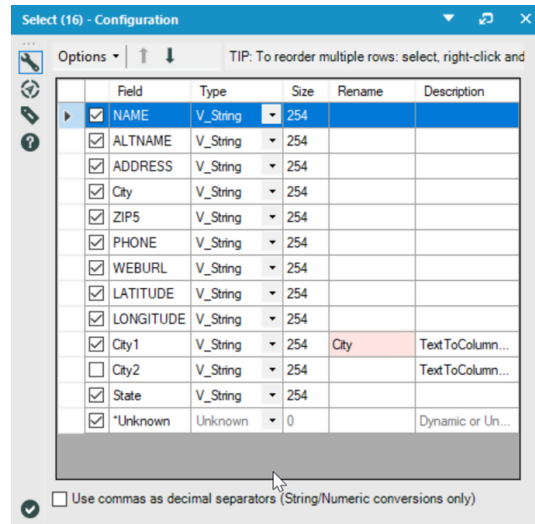


Figure-5-21- Culturally Rich
Museum Data Select Configuration after Formula Tool

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.

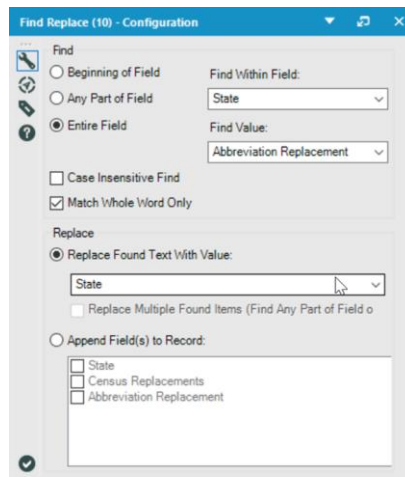


Figure-5-22- Culturally Rich
Second Find and Replace Configuration

Now that we have prepared all the contributing data streams, below is what the data stream should look like.

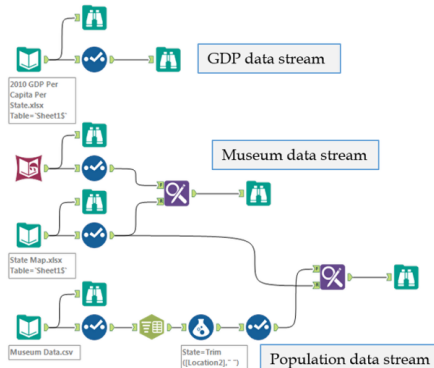


Figure 5-25 – Culturally Rich Consolidated Modified data streams

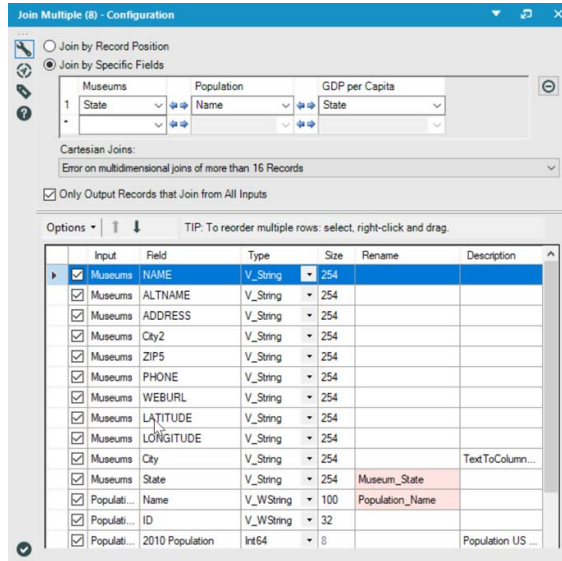
Notice that we've crossed the data streams. From a technical perspective, it does not matter if they are crossed. However, practically, the workflow becomes harder to read when the data streams are crossed frequently.

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 crossover 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* input 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 named *State*, the second occurrence is renamed with the data stream as an identifier: *Museum_State*.



**Figure-5-24- Culturally Rich
Second Join Multiple Configuration**

Below is what the data stream would look like.

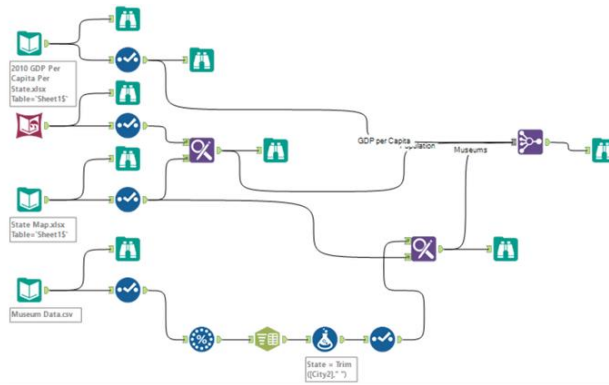


Figure-5-25 – Culturally Rich Unified Data Stream

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*) using a *Formula* tool.

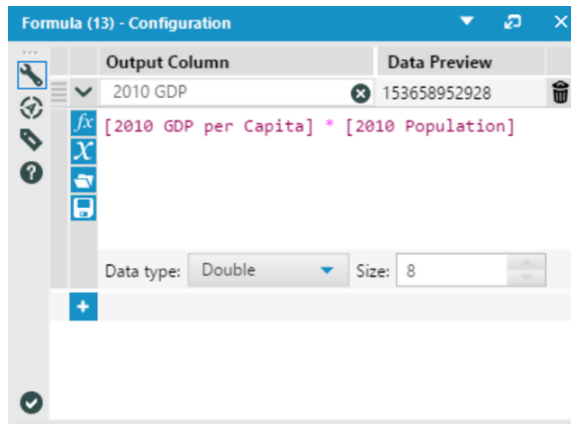


Figure-5-26 – Culturally Rich Total GDP Calculation

- Export the data to a file called *Culturally Rich* using the *Output* tool.

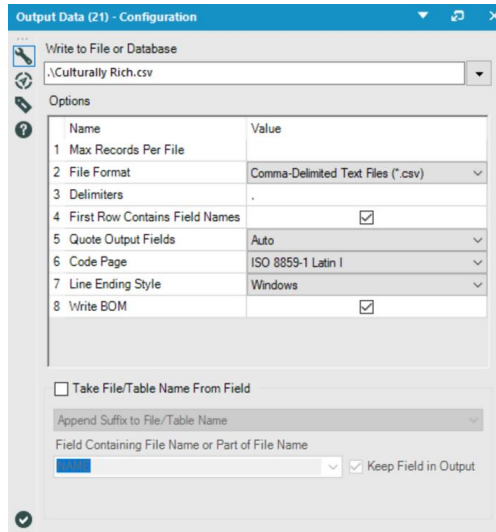


Figure-5-27 – Culturally Rich Output Data Configuration

The *Culturally Rich* data stream should look like the image below when it is complete.

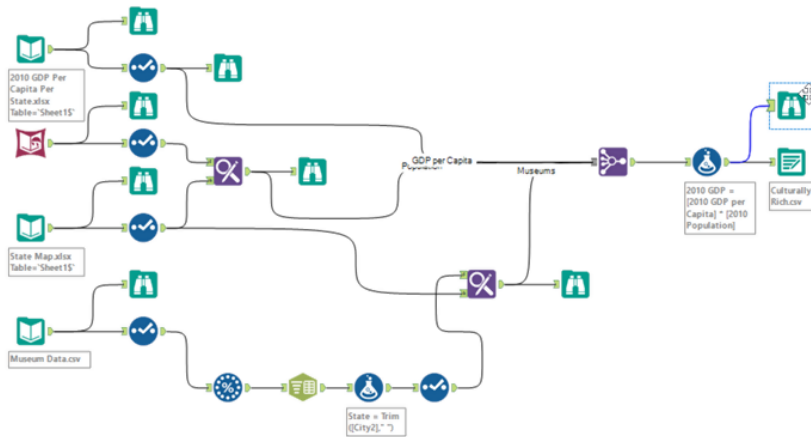


Figure-5-28 – Culturally Rich Complete Workflow

5.3 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

Expensive Beauty Product

6.1 Tools and Concepts


6.1.1 Fuzzy Match

Concept: Fuzzy Logic

 <p>Figure 6-1 - Fuzzy Match</p>	<p>Tool Palette: Join</p> <p>The <i>Fuzzy Match</i> tool allows us to apply logic to a data stream to match similar items. For more details, use the link below. bit.ly/2ZtGRsQ</p>
--	---


6.1.2 Record ID

Concept: Indexing

 <p>Figure 6-2 - Record ID</p>	<p>Tool Palette: Preparation</p> <p>The <i>Record ID</i> tool creates an index column in our data stream. For more details, use the link below. bit.ly/2KVPad3</p>
--	--

6.1.3 RegEx

Concept: Regular Expressions, Pre-Calculation

 <p>Figure 6-3 - RegEx</p>	<p>Tool Palette: Parse</p> <p>The <i>RegEx</i> tool allows us to run regular expressions on a field in our data stream. For more details, use the link below. bit.ly/2KWQhsR</p>
--	--

6.1.4 Unique

Concept: Removing Duplicates



Figure 6-4 - Unique

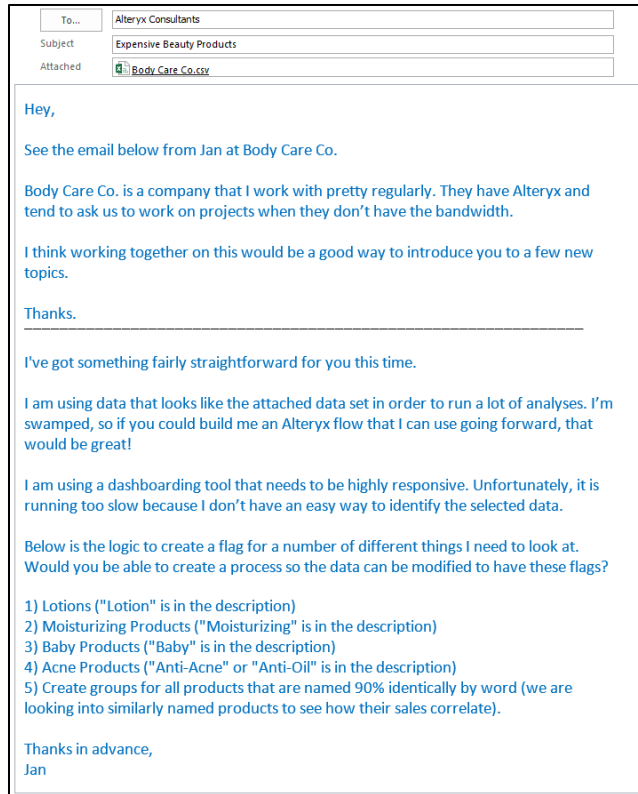
Tool Palette: Preparation

The *Unique* tool allows us to separate unique or duplicate records based on the field(s) we select.

For more details, use the link below.

bit.ly/33XrfkV

6.2 Expensive Beauty Products



The problem Jan describes 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. That means that as much as possible should be built into the data preparation process. In

Jan's case, they're doing some complex string calculations in order to appropriately filter the data.

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—a fundamental part of good data preparation—are performed.

Even when we are not preparing the data for analytics tools, it is important to pre-calculate fields the end user may need. That is because building and vetting a calculation once it's in a workflow means there will be a significant reduction in risk caused by human error downstream.

Let's 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 those analysts are each rewriting the same formula (*Profit/Sales*) every month, it means we need to make sure they don't accidentally type (*Sales/Profit*) in 240 different formulas per year. Instead, if we know they need a transaction-level profit ratio, we can create a standard formula in the data preparation process that writes this value into the source data they all use.

In addition to the pre-calculation, we can see from this e-mail that Jan is only interested in a module. That is the case because clients work with Alteryx and will do some modifications such as replacing the data connection and outputs. We will often come across things like that when supporting other Alteryx users because they either do not know how to or do not have time to build a portion of a module.

Considering the information in the e-mail, it seems like Jan was talking about flagging item names that fall into each of those buckets. We will use two techniques to achieve that. 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 the grouping for each of the tools.

Regular expressions are pattern-matching formulas 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's 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, that is limited to string parsing; however, more generally, it 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 that. The nuances of the differences are well documented, and it is suggested that the reader learn more about them if they plan to use Alteryx's *Fuzzy Match* tool.



Figure 6.5 – Expensive Beauty Products Body Care Company Data

If we connect to the file in *Chapter 6 – Body Care Co.csv*, we see that Alteryx is reading all the fields in this file as strings.

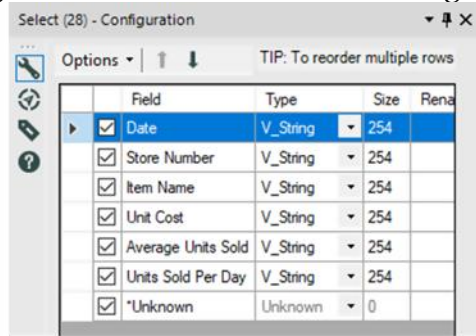


Figure 6.6 – Expensive Beauty Products Selection Configuration

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 the entire set of tools should be replaced when the connection is changed.

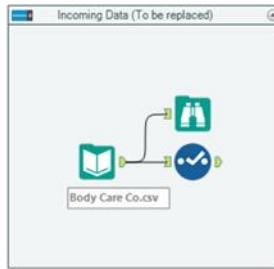


Figure 6.7 – Expensive Beauty Products Input Data – to be replaced in a container

Let's take a look at the contents of this file.



Record #	Date	Store Number	Item Name	Unit Cost	Average Units Sold	Units Sold Per Day
1	1/1/2014	1	Vitamin Restoring Anti-Aging Cream 1 Ounce	1.99	0	3
2	1/1/2014	1	Anti-Itch Cream 1 Ounce	1.99	0	3
3	1/1/2014	1	Anti-Itch Lotion 5 Ounce	1.99	1	2
4	1/1/2014	1	Natural Protection Sunscreen SPF 50 3 Ounce	10.99	0	3
5	1/1/2014	1	Baby Calming Bubble Bath 8 Ounce	3.99	0	3
6	1/1/2014	1	Baby Lotion 8 Ounce	2.99	0	2
7	1/1/2014	1	Baby Lotion 20 Ounce	5.99	0	1
8	1/1/2014	1	Baby Soothing Anti-Itch Cream 3 Ounce	3.99	0	3
9	1/1/2014	1	Baby Wash And Shampoo 18 Ounce	2.99	0	2
10	1/1/2014	1	Lavender Body Wash 14 Ounce	2.99	0	2
11	1/1/2014	1	Stress Relief Body Wash 10 Ounce	2.99	0	1
12	1/1/2014	1	Anti-Acne Medicated Body Cream 2 Ounce	11.99	0	2

Figure 6.8 – Expensive Beauty Products Input Data – Browse Configuration

Next, let's create flags using regular expressions. Let's create a new *Tool Container* called Regular Expressions with a *Regex* tool in it.

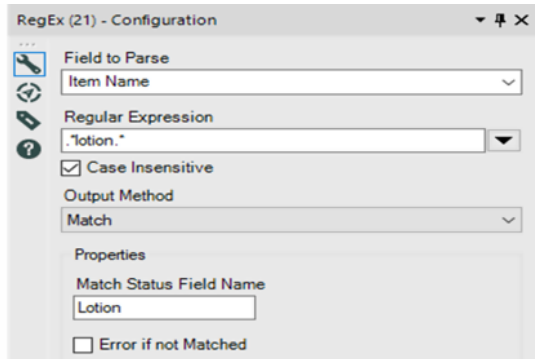


Figure 6.9 – Expensive Beauty Products Regex – “lotion” Flag Configuration

The *RegEx Configuration* window has the following five basic components:

- *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.

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.

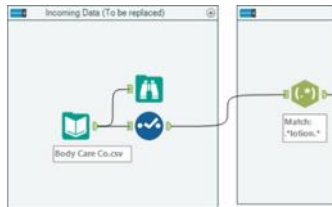


Figure 6.10 – Expensive Beauty Products Data Stream

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 returns true or false.

A familiarity with Body Care Co’s data lets us know their item names always have spaces between words. That 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.

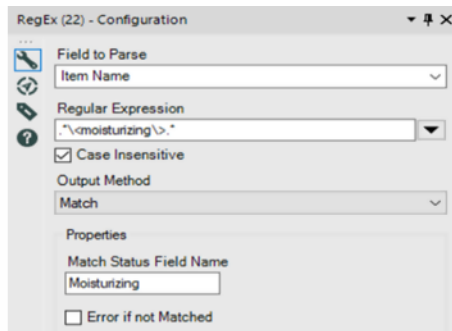


Figure 6.11 – Expensive Beauty Products RegEx – “moisturizing” Flag Configuration

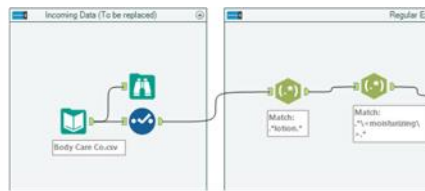


Figure 6.12 – Expensive Beauty Products Data Stream

Follow the same pattern to create a field called Baby Product.

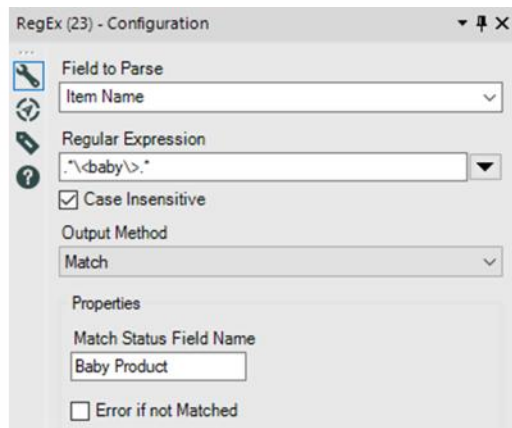


Figure 6.13 – Expensive Beauty Products Regex – “baby” Flag Configuration

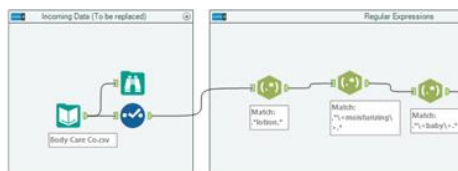


Figure 6.14 – Expensive Beauty Products Data Stream

We then make another for acne products.

Our last flag is more complicated because we have two different things that could define acne products.

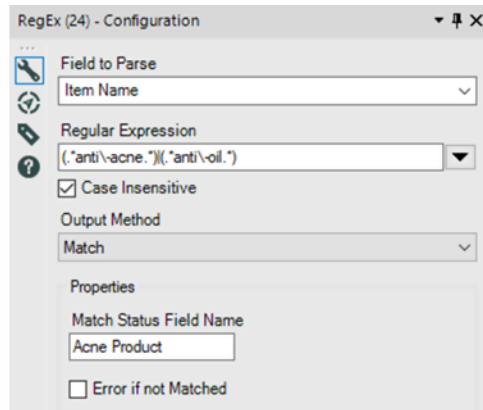


Figure 6.15 – Expensive Beauty Products RegEx – a Complex Flag - Configuration

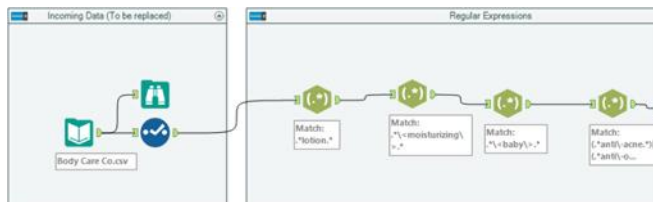


Figure 6.16 – Expensive Beauty Products Data Stream

We will combine what we know about Boolean statements and regular expressions to create this 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.”

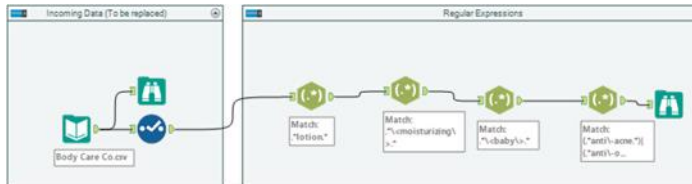
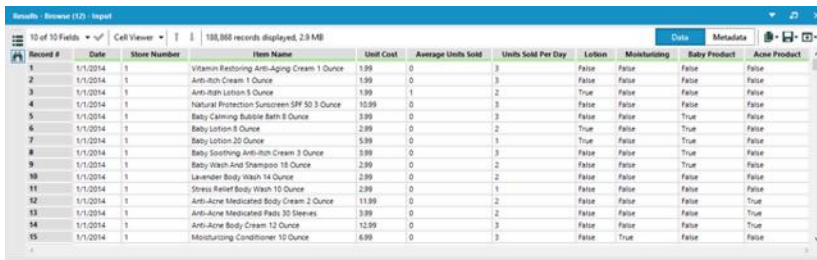


Figure 6.17 – Expensive Beauty Products Data Stream



Record #	Date	Store Number	Item Name	Unit Cost	Average Units Sold	Units Sold Per Day	Lotion	Moisturizing	Baby Product	Acne Product
1	1/1/2014	1	Vitamin Restoring Anti-Aging Cream 1 Ounce	1.99	0	3	False	False	False	False
2	1/1/2014	1	Anti-Itch Cream 1 Ounce	1.99	0	3	False	False	False	False
3	1/1/2014	1	Anti-Itch Lotion 5 Ounce	1.99	1	2	True	False	False	False
4	1/1/2014	1	Natural Protection Sunscreen SPF 30 3 Ounce	10.99	0	3	False	False	False	False
5	1/1/2014	1	Baby Calming Bubble Bath 8 Ounce	3.99	0	3	False	False	True	False
6	1/1/2014	1	Baby Lotion 8 Ounce	2.99	0	2	True	False	True	False
7	1/1/2014	1	Baby Lotion 20 Ounce	5.99	0	1	True	False	True	False
8	1/1/2014	1	Baby Soothing Anti-Itch Cream 3 Ounce	3.99	0	3	False	False	True	False
9	1/1/2014	1	Baby Wash And Shampoo 18 Ounce	2.99	0	2	False	False	True	False
10	1/1/2014	1	Lanolin Body Lotion 16 Ounce	2.99	0	2	False	False	False	False
11	1/1/2014	1	Stress Relief Body Wash 10 Ounce	2.99	0	1	False	False	False	False
12	1/1/2014	1	Anti-Acne Medicated Body Cream 2 Ounce	11.99	0	2	False	False	False	True
13	1/1/2014	1	Anti-Acne Medicated Pads 20 Sleeves	3.99	0	2	False	False	False	True
14	1/1/2014	1	Anti-Acne Body Cream 12 Ounce	12.99	0	3	False	False	False	True
15	1/1/2014	1	Moisturizing Conditioner 18 Ounce	6.99	0	3	False	True	False	False

Figure 6.18 – 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. We can now add a *Comment* tool that describes what’s happening in the *Tool Container* and moves on to the fuzzy logic.

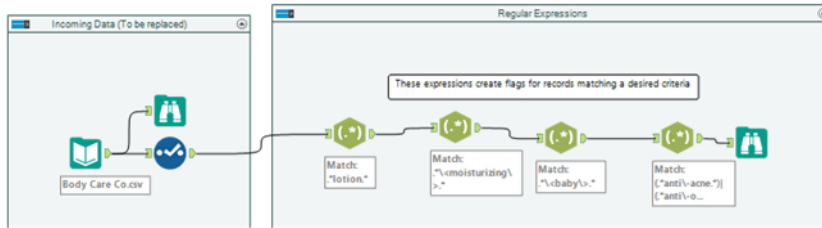


Figure 6.19 – Adding Comments to indicate contents in Tool Container and Updated Data Stream

We need to define the groups of products with similar names. The first thing we should do is create a unique list of product names. Let's introduce a *Summarize* tool within a container and connect it to the data input stream.

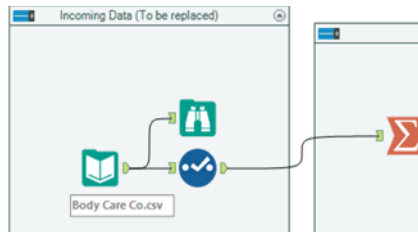


Figure 6.20 – Expensive Beauty Products second Data Stream for fuzzy match

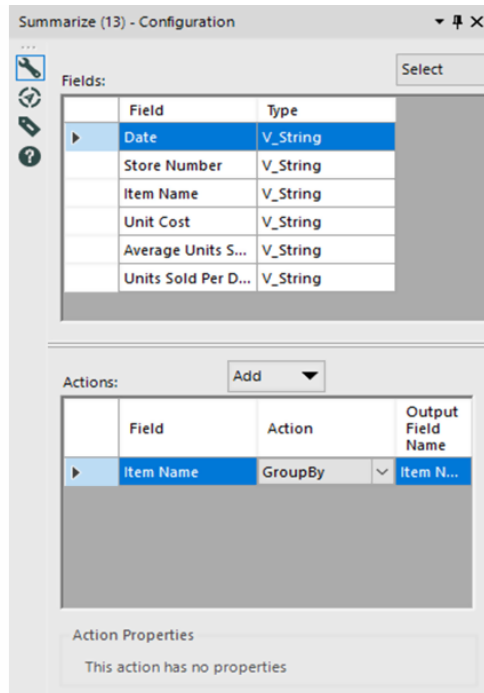


Figure 6.21 – Expensive Beauty Products Summarize tool configuration

Let's run the workflow and take a look at the unique list.

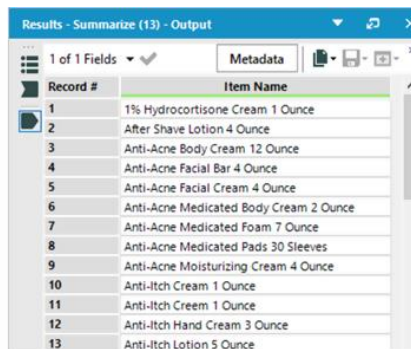


Figure 6.22 – Expensive Beauty Products Browse Configuration after Summarize tool

Now that we have a unique list of products, we can determine which of those products have similar names.



Figure 6.23 - Including Fuzzy Match Tool in Data Stream

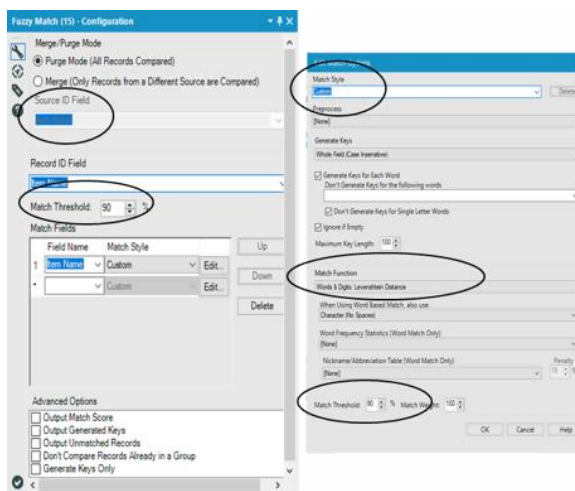


Figure 6.24 – Expensive Beauty Products Fuzzy Match Configuration

The *Fuzzy Match Configuration* window has the following five core components:

- *Merge/Purge Mode* allows us to decide if we need to compare all the records on the list or compare records with different sources. That means if we have data that's broken into logical buckets, we can compare across those buckets without comparing them.
- *Record ID Field* is the mapping field.

- *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.

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*. That algorithm is looking at the whole field by word (except for single-character word, up to 100 words per field) and keeping anything 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 that, we will isolate the unique records to be kept.

If we add a *Unique* tool after the *Fuzzy Match*, we can see that a new field has been created called *Item Name2*. The new field allows us to see which entries match.

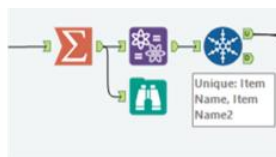


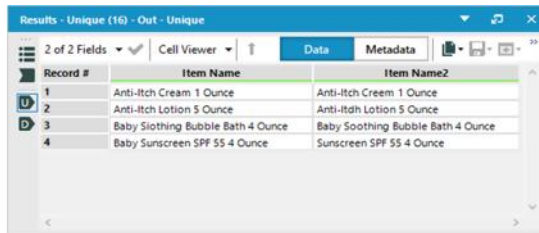
Figure 6.25 – Addition of Unique tool to fuzzy logic data stream



Figure 6.26 – Expensive Beauty Products Unique tool configuration

The *Unique Configuration* window is a list of each of the fields in the data set. We need to select all those we want to use in order to identify unique records.

A quick run using the *Browse* tool shows us the matched items.



Record #	Item Name	Item Name2
1	Anti-Itch Cream 1 Ounce	Anti-Itch Cream 1 Ounce
2	Anti-Itch Lotion 5 Ounce	Anti-Itch Lotion 5 Ounce
3	Baby Soothing Bubble Bath 4 Ounce	Baby Soothing Bubble Bath 4 Ounce
4	Baby Sunscreen SPF 55 4 Ounce	Sunscreen SPF 55 4 Ounce

Figure 6.27 – Expensive Beauty Products viewing matching 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.

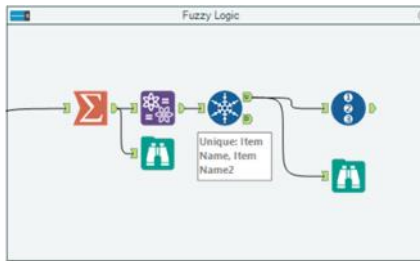


Figure 6.28 – Addition of Record ID tool to fuzzy logic data stream

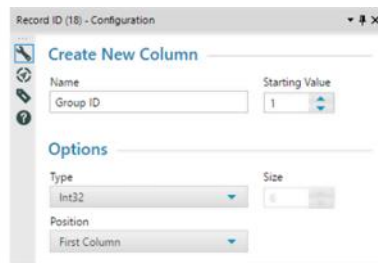


Figure 6.29 – Expensive Beauty Products Record ID Configuration

The *Record ID Configuration* window has the following four basic components:

- *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 data set.

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.

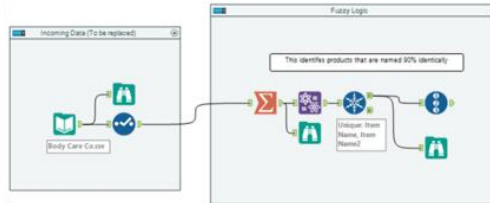


Figure 6.30 – Adding Comments to indicate contents in Tool Container and Updated Fuzzy Logic Stream

In order to combine the data streams, we need to join the data twice so each part of the group can be flagged with the *Group ID*.

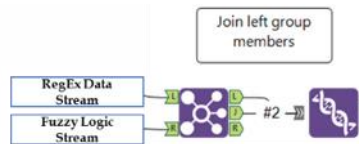


Figure 6.31 – Expensive Beauty Products First Join for Left Group Members

The *Join* tools settings are shown below.

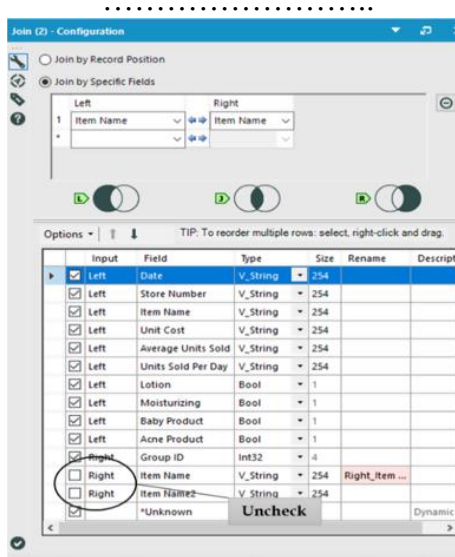


Figure 6.32 – Expensive Beauty Products Join Configuration for “Item Name”

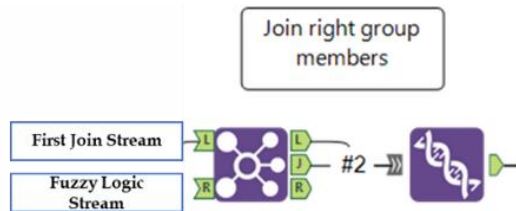


Figure 6.33 – Expensive Beauty Products Second Join for Right Group Members

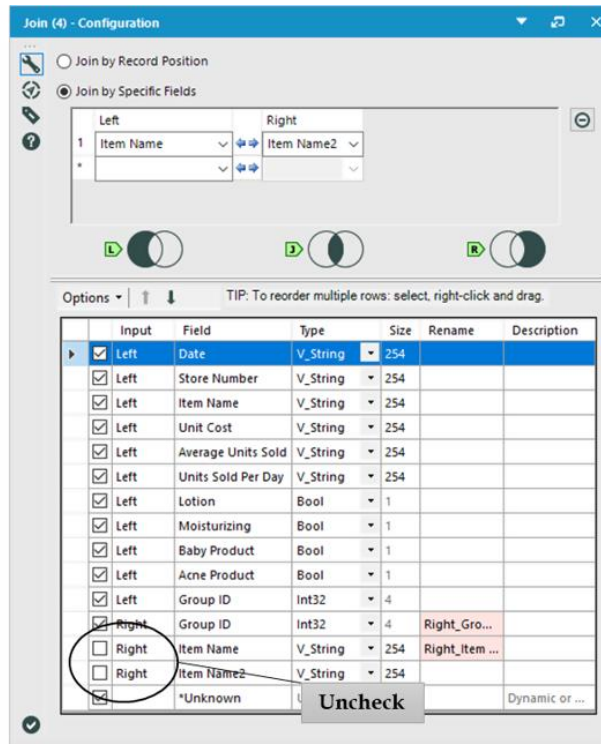
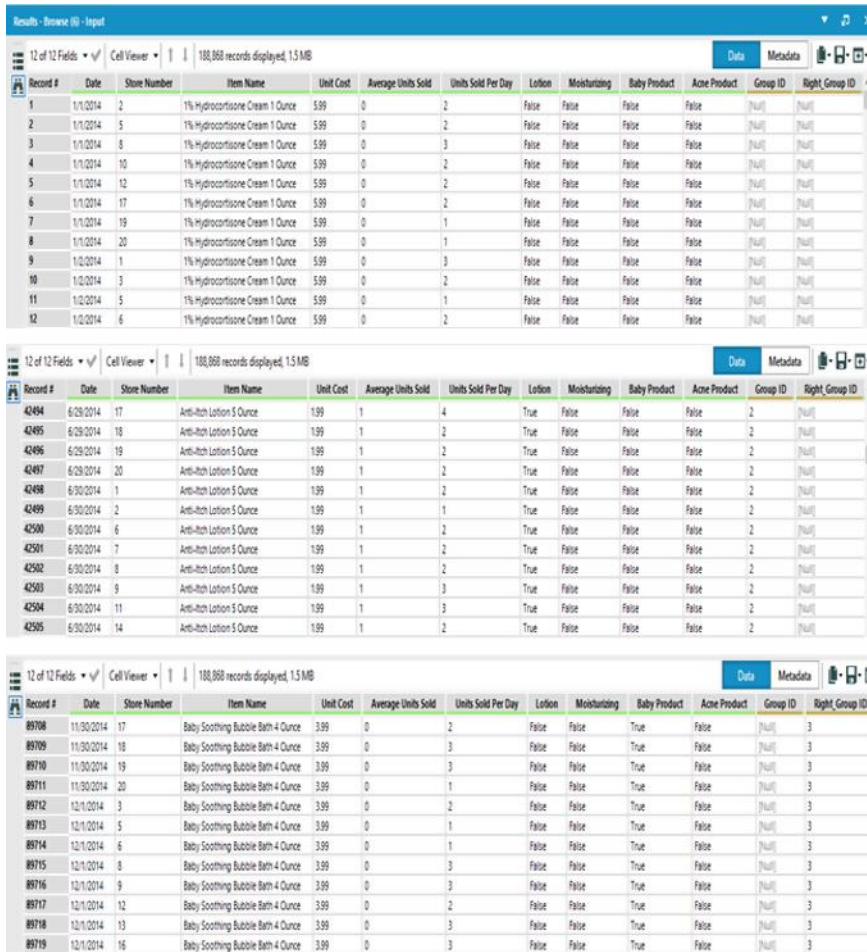


Figure 6.34 – Expensive Beauty Products Join Configuration for “Item Name 2”



The figure displays three screenshots of an Alteryx data stream, each showing a table of product sales data. The tables are filtered to show 12 of 12 fields and 108,868 records displayed (1.5 MB).

Table 1 (Top): Shows records 1 through 12. All items are '1% Hydrocortisone Cream 1 Ounce' with a unit cost of 5.99 and an average units sold of 0. The 'Units Sold Per Day' ranges from 1 to 3. The 'Lotion' and 'Moisturizing' columns are all 'False'. The 'Baby Product' and 'Acne Product' columns are all 'False'. The 'Group ID' and 'Right_Group ID' columns are both '(Null)'. The 'Date' column shows a range from 1/1/2014 to 1/2/2014.

Table 2 (Middle): Shows records 42494 through 42505. All items are 'Anti-Itch Lotion 5 Ounce' with a unit cost of 1.99 and an average units sold of 1. The 'Units Sold Per Day' ranges from 1 to 4. The 'Lotion' column is 'True' for all records, while 'Moisturizing' is 'False'. The 'Baby Product' and 'Acne Product' columns are all 'False'. The 'Group ID' and 'Right_Group ID' columns are both '2'. The 'Date' column shows a range from 6/29/2014 to 6/30/2014.

Table 3 (Bottom): Shows records 89708 through 89719. All items are 'Baby Soothing Bubble Bath 4 Ounce' with a unit cost of 3.99 and an average units sold of 0. The 'Units Sold Per Day' ranges from 1 to 3. The 'Lotion' and 'Moisturizing' columns are all 'False'. The 'Baby Product' column is 'True' for all records, while 'Acne Product' is 'False'. The 'Group ID' and 'Right_Group ID' columns are both '3'. The 'Date' column shows a range from 11/30/2014 to 12/1/2014.

Figure 6.35 – 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 data set to work from.

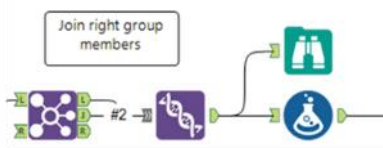


Figure 6.36 – Inclusion of Formula tool in data stream

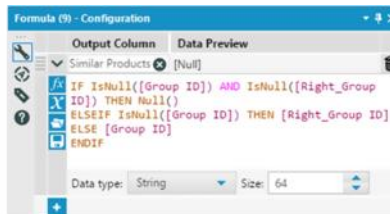


Figure 6.37 – Expensive Beauty Products Formula Tool Configuration

We can create a conditional formula (below) 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
```

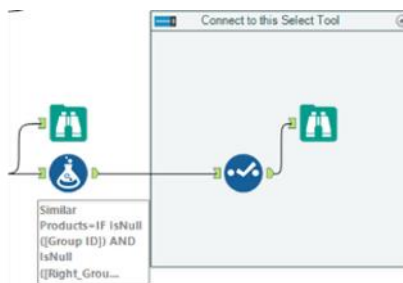


Figure 6.38 – Select tool included for final output

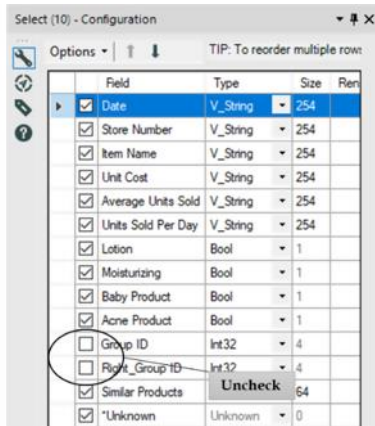


Figure 6.39 – Expensive Beauty Products output Select Configuration

In order to finish the 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.

Record #	Date	St...	Item Name	Unit...	A...	Un...	Lotion	Mo...	Baby...	Acn...	Similar Products
1	1/1/2014	2	1% Hydrocortisone Cream 1 Ounce	5.99	0	2	False	False	False	False	[Null]
2	1/1/2014	5	1% Hydrocortisone Cream 1 Ounce	5.99	0	2	False	False	False	False	[Null]
3	1/1/2014	8	1% Hydrocortisone Cream 1 Ounce	5.99	0	3	False	False	False	False	[Null]
4	1/1/2014	10	1% Hydrocortisone Cream 1 Ounce	5.99	0	2	False	False	False	False	[Null]
5	1/1/2014	12	1% Hydrocortisone Cream 1 Ounce	5.99	0	2	False	False	False	False	[Null]
6	1/1/2014	17	1% Hydrocortisone Cream 1 Ounce	5.99	0	2	False	False	False	False	[Null]

Figure 6.40 – Expensive Beauty Products Output

The *Expensive Beauty Products* data stream should look like the following image when it is complete:

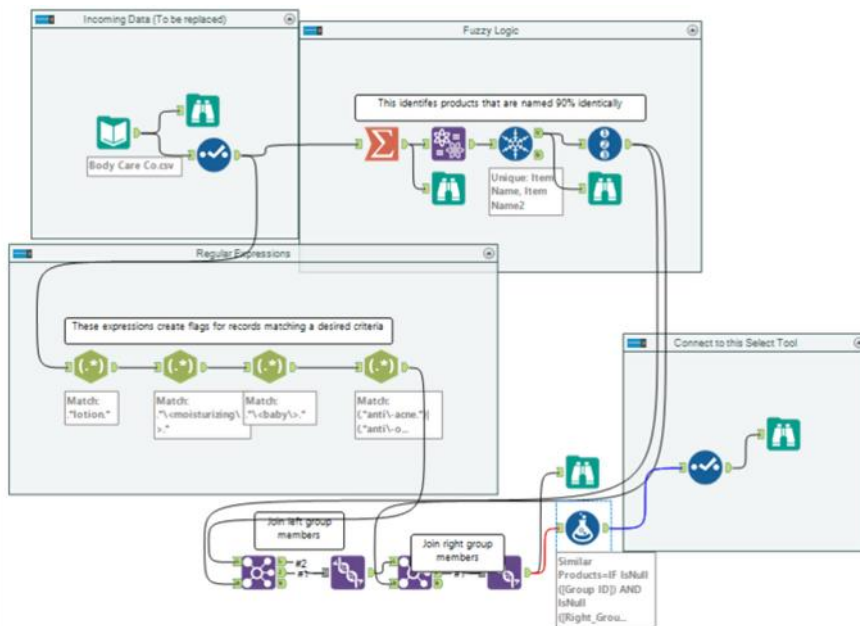


Figure 6.41 – Expensive Beauty Product Stream on Completion

6.3 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 7

Let's Do It (In-DB)

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.

7.1 Tools and Concepts


7.1.1 Browse In-DB

Concept: Viewing data present in the database

 <p>Figure 7-1 Browse-In-DB</p>	<p>Tool Palette: In-Database</p> <p>The <i>Browse In-DB</i> tool allows us to review the In-DB data stream at any point in the workflow. For more details, use the link below. bit.ly/2zsYgY4</p>
---	--


7.1.2 Connect In-DB

Concept: Connect to a database without bringing in the data (In-DB)

 <p>Figure 7-2 Connect In-DB</p>	<p>Tool Palette: In-Database</p> <p>The <i>Connect In-DB</i> tool allows us to connect to an In-DB data stream in the workflow. For more details, use the link below. bit.ly/2zof4Q3</p>
---	---


7.1.3 Data Stream In

Concept: Bring in external data into the In-DB data stream

 <p>Figure 7-3 Data Stream In</p>	Tool Palette: In-Database
	<p>The Data Stream In tool loads a standard input data into In-DB stream by creating a temporary database table to load the data.</p> <p>For more details, use the link below. bit.ly/342Ywvi</p>


7.1.4 Data Stream Out

Concept: Bring the data from In-DB stream to external workflow

 <p>Figure 7-4 Data Stream Out</p>	Tool Palette: In-Database
	<p>The Data Stream Out tool provides the ability to move data from In-DB Stream to the workflow.</p> <p>For more details, use the link below. bit.ly/2ksREFp</p>


7.1.5 Filter In-DB

Concept: Conditionally filter data present in the In-DB stream

 <p>Figure 7-5 Filter In-DB</p>	Tool Palette: In-Database
	<p>The Filter In-DB tool creates criteria and outputs only those records that match the criteria. For more details, use the link below. bit.ly/2HqDdd5</p>


7.1.6 Formula In-DB

Concept: Transformation data in the In-DB stream

 <p>Figure 7-6 Formula In-DB</p>	Tool Palette: In-Database
	<p>The Formula In-DB tool allows the In-DB fields to be created and updated using SQL expressions based on the native database language. For more details, use the link below. bit.ly/2ZidNta</p>


7.1.7 Join In-DB

Concept: Join multiple tables within the In-DB stream

 <p>Figure 7-7 Join In-DB</p>	Tool Palette: In-Database
	<p>The Join In-DB tool allows joining two database tables from In-DB stream. For more details, use the link below. bit.ly/2zqzd86</p>


7.1.8 Macro Input In-DB

Concept: Entry point for macro instructions for In-DB Stream

 <p data-bbox="306 531 508 583">Figure 7-8 Macro Input In-DB</p>	<p data-bbox="588 348 964 378">Tool Palette: In-Database</p> <p data-bbox="588 387 1116 560">The Macro Input In-DB tool is used to display In-DB input anchors on a macro tool for use in In-DB workflows. For more details, use the link below. bit.ly/345vM4L</p>
---	--


7.1.9 Macro Output In-DB

Concept: Exit point for macro instructions for In-DB Stream

 <p data-bbox="313 953 501 996">Figure 7-9 Macro Output In-DB</p>	<p data-bbox="588 748 964 777">Tool Palette: In-Database</p> <p data-bbox="588 795 1130 968">The Macro Output In-DB tool is used to display In-DB output anchors on a macro tool for use in In-DB workflows. For more details, use the link below. bit.ly/2Lfew4r</p>
--	--


7.1.10 Select In-DB

Concept: Select and rename fields present in the In-DB Stream

 <p>Figure 7-10 Select In-DB</p>	<p>Tool Palette: In-Database</p> <p>The Select In-DB tool allows manipulation of fields in an In-DB stream by selecting, deselecting, renaming, and reordering the fields. For more details, use the link below. bit.ly/324ujtO</p>
--	--


7.1.11 Summarize In-DB

Concept: Aggregate/Roll-up data present in the In-DB Stream

 <p>Figure 7-11 Summarize In-DB</p>	<p>Tool Palette: In-Database</p> <p>The Summarize In-DB tool summarizes the input data using different aggregate functions. For more details, use the link below. bit.ly/2PgwnGt</p>
--	---

7.1.12 Union In-DB

Concept: Vertically stack data sets present in the In-DB Stream with same table structure

 <p>Figure 7-12 Union In-DB</p>	<p>Tool Palette: In-Database</p> <p>The Union In-DB tool combines two or more In-DB streams of data with a similar structure.</p> <p>For more details, use the link below. bit.ly/2KZbL8x</p>
--	---


7.1.13 Write Data In-DB

Concept: Write to a database from the In-DB Stream

 <p>Figure 7-13 Write In-DB</p>	<p>Tool Palette: In-Database</p> <p>Creates a data stream that has the contents of a Directory or Folder.</p> <p>For more details, use the link below. bit.ly/2PgNL4W</p>
--	---


7.1.14 Sample In-DB

Concept: Provides selection of data subset

 <p>Figure 7-14 Sample In-DB</p>	<p>Tool Palette: In-Database</p> <p>The Sample In-DB tool allows sampling of the In-DB stream based on numbers or percentage of a total number of records.</p> <p>For more details, use the link below. bit.ly/2PglGd6</p>
---	--


7.1.15 Calgary Loader

Concept: Creating a Calgary database

 <p>Figure 7-15 Calgary Loader</p>	<p>Tool Palette: Calgary</p> <p>The Calgary Loader enables you to create a Calgary database (*.cydb) from any type of Input file. Each field contained in the Input file can be indexed to maximize the Calgary database performance.</p> <p>For more details, use the link below. bit.ly/30CSwXL</p>
---	---


7.1.16 Calgary Input

Concept: To query a Calgary database

 <p>Figure 7-16 Calgary Input</p>	<p>Tool Palette: Calgary</p> <p>The Calgary input tool is one way of accessing the raw records back from a Calgary database, generally by building a query on the records.</p> <p>For more details, use the link below. bit.ly/3479C2q</p>
--	--


7.1.17 Calgary Join

Concept: To join against a Calgary database

 <p>Figure 7-17 Calgary Join</p>	<p>Tool Palette: Calgary</p> <p>The Calgary join tool provides users with the ability to take an input file and perform joins against a Calgary database.</p> <p>For more details, use the link below. bit.ly/2KZW2pR</p>
---	---


7.1.18 Calgary Cross Count

Concept: To do a multidimensional crosstab on the fields in Calgary database


 <p>Figure 7-18 Calgary Cross Count</p>	<p>Tool Palette: Calgary</p> <p>The Calgary Cross Count tool enables users to aggregate data across multiple Calgary database fields to return a count per record group. For more details, use the link below. bit.ly/2zpSI0B</p>
--	--

7.1.19 Calgary Cross Count Append

Concept: To Append the counts to Calgary database

 <p>Figure 7-19 Calgary Cross Count Append</p>	<p>Tool Palette: Calgary</p> <p>The Calgary Cross Count Append tool provides users with the ability to take an input file and append counts to records that join to a Calgary database. For more details, use the link below. bit.ly/2Zzxoo0</p>
--	---

7.2 Superstore Database Report

Send	To...	Alteryx Consultants
	Cc...	
	Subject	Superstore Database Report
Attached	 Product Info.csv 160 KB	

Hey,

Superstore Inc. would like to generate data that 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).

The 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 in this case is SQL Server. 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.

It is important to understand while using In-DB tools that the data is not imported into the local environment. Rather, it continues to be processed using the DB's resources.

We are going to approach our solution by using the *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. The data set is available in *Chapter 11 – Product Info.csv*.

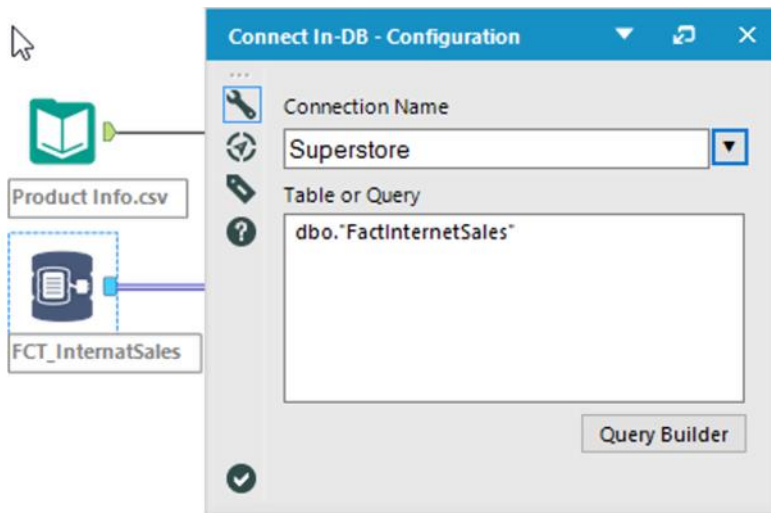


Figure 7-20 Superstore Data, Connect In-DB

We know from the e-mail that the data from the database needs to be considered for 2006 and on. So we need to make use of the *Filter In-DB* tool to restrict the records accordingly. Further, 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 set the *Creation Mode* in the *Data Stream In* tool to *Create Temporary Table*.

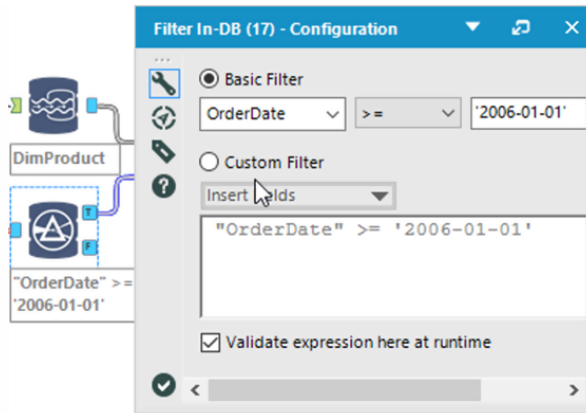


Figure 7-21 Superstore Data, Filter In-DB

Before proceeding, we need to understand how to bring *Product Name* into the data stream, considering this column isn't part of the Fact table. The *Product Info.csv* file has a *Product Name* column, which we require, and also the *ProductKey* column, which is also present in the database table. That field serves as a common column between the database table and the file. We will join the two data streams by using a *Join In-DB* tool and defining *ProductKey* as the join condition.

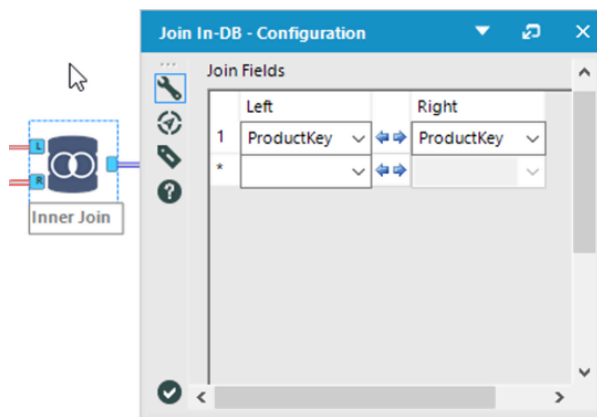


Figure 7-22 Superstore Data, Join In-DB

The output of the *Join In-DB* tool contains many fields, but moving forward, we require only a select few of these fields. To achieve that, we will use the *Select In-DB* tool to select only relevant columns such as *EnglishProductName* renamed *ProductName*, *OrderQuantity*, and *OrderDate*. Once selected, we will use the *Formula In-DB* tool to calculate the *year of OrderDate*. For the purpose of this calculation, we'll use `YEAR("OrderDate")` as the SQL expression in *Formula In-DB* tool. We'll rename the new field *OrderYear* and keep the *Type* as *Double*.

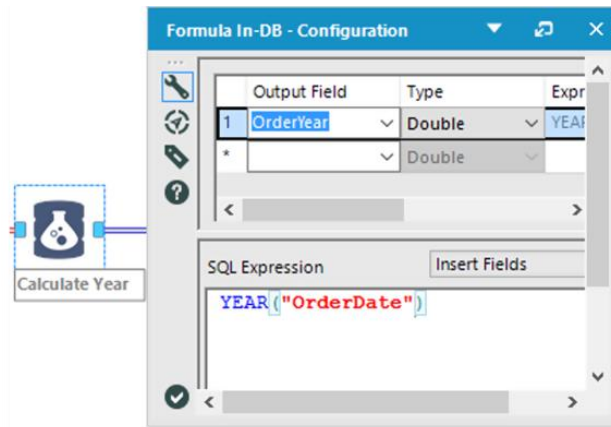


Figure 7-23 Superstore Data, Formula In-DB

After *OrderYear* has been calculated, we are now ready to aggregate *OrderQuantity* to obtain quantity values for each year. To achieve that, we will use the *Summarize In-DB* tool. Add the *GroupBy* action for *ProductName* and *OrderYear* while adding the *Sum* action from the *Numeric* section for the *OrderQuantity* field. For *Output Field Name*, keep the auto generated field names intact.

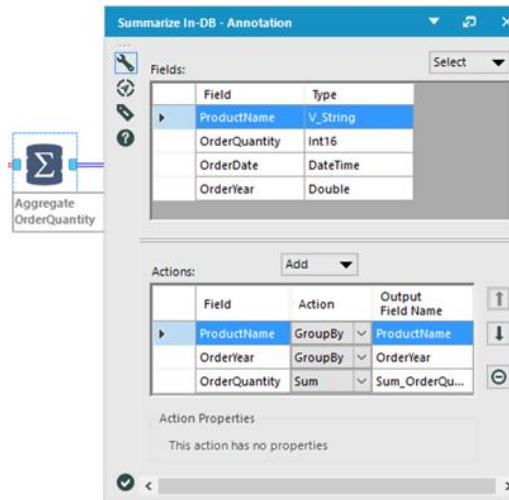
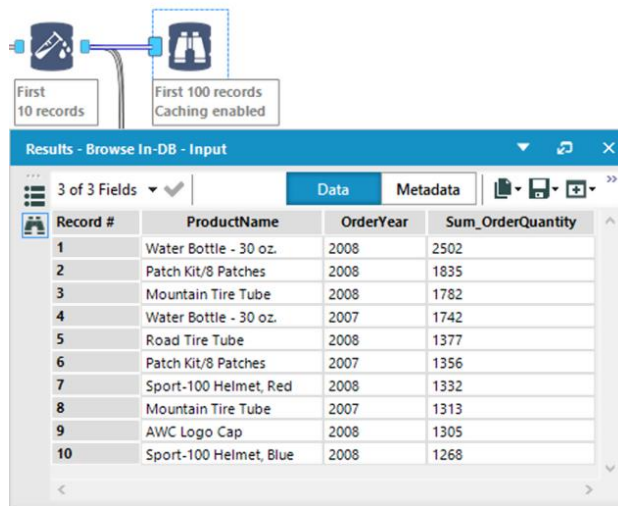


Figure 7-24 Superstore Data, Summarize In-DB

We now need to select only the records with Top 10 order quantities. In order to achieve that, we first need to sort the records in descending order of the sum of order quantity and then select the top 10 records from this list. Make use of the *Sample In-DB* tool for this purpose by specifying *Number of records to sample* as 10. Also, we need to select the *Sample records based on order* checkbox. Select *Sum_OrderQuantity* under the *Field* section and choose *Descending* as its order.

At this stage, the data should be ready according to the requirements. However, to confirm that, we have to add the *Browse In-DB* tool and then run the workflow to see the output in the *Results* window.



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 7-25 Superstore Data, Sample In-DB, Browse In-DB

Now that the data in the *Results* window appears according to the expected requirements, it is now ready for output. We need to output the results in two formats – first, a CSV file to be saved locally and as a database table. In order to do that, we will add *Data Stream Out* and the *Write Data In-DB* tool next to the *Sample In-DB* tool. While *Data Stream Out* does not need any configuration changes, for *Write Data In-DB*, select *Overwrite Table (Drop)* in *Creation Mode* and mention the table name as *ProductSummary*. Add the *Output Data* tool next to the *Data Stream Out* tool to write output data to a CSV file. Name the file *ProductSummary.csv*. The completed workflow would look something like the image below.

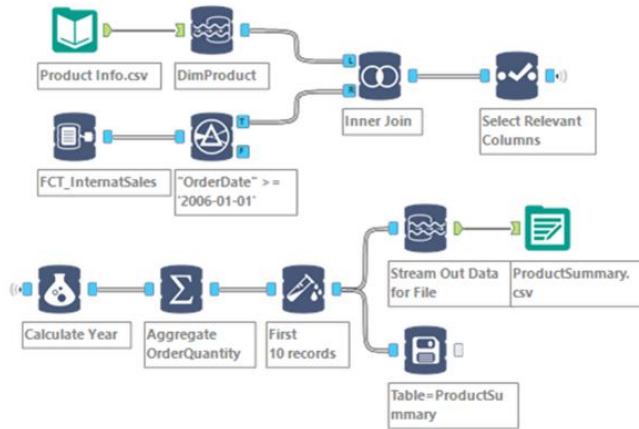


Figure 7-26 Superstore Data, Complete Workflow

ProductName	OrderYear	Sum_OrderQuantity
Water Bottle - 30 oz.	2008	2502
Patch Kit/8 Patches	2008	1835
Mountain Tire Tube	2008	1782
Water Bottle - 30 oz.	2007	1742
Road Tire Tube	2008	1377
Patch Kit/8 Patches	2007	1356
Sport-100 Helmet, Red	2008	1332
Mountain Tire Tube	2007	1313
AWC Logo Cap	2008	1305
Sport-100 Helmet, Blue	2008	1268

 Figure 7-27 Superstore Data,
 Report Output

7.3 Superstore Database Report with Macro

Send	To, ..	Alteryx Consultants
	CC, ..	
Subject		Re: Superstore Database Report

Hey,

Thank you for the earlier report, it has been of immense value.

As a follow up, Pete, our business analyst will be loading the Product Information CSV into our database. He will require help for the testing and would ideally want to see the following in a single table for a quick glance:

1. All Sales grouped by year
2. All Order Quantities grouped by year

As he'll be doing this over the weekend, I'd like you to build him an Alteryx Workflow that he can re-run whenever he modifies the data.

Please note, Pete isn't very comfortable with ETL workflows and seems to get overwhelmed by them. Could you find a method in which he'd only see a couple of icons at max with the same functionality?

The task at hand is very similar to the previous request. The difference is that we'll also be analyzing the sales amount by year. In addition, we'll have to condense the workflow into a simpler workflow for Pete.

To begin, let's consider how we'll go about condensing the workflow. Since we want to retain the power of the workflow and yet show only the minimum required workflow icons, we can consider using the concept of *Macros* – in this case, *Macro In-DB*. As a starting point, use the *Macro Input In-DB* tool and point it to one of the database tables (in our case, *FactInternetSales*). Since the macro will accept one input, we can set the *DimProduct* table (database table corresponding to Product Info.csv) as a source using the *Connect In-DB* tool.

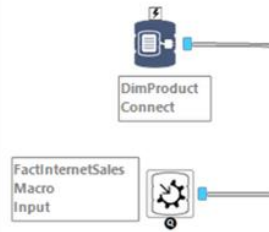


Figure 7-28 Superstore Database Report with Macro Data Stream

Connect the two data streams by joining them on *ProductKey* using the *Join In-DB* tool, similar to the previous task.



Figure 7-29 Join In DB

We now need to derive the year from *OrderDate* as well as fix the issue with *SalesAmount* not being a numeric value. For that, we use the *Formula In-DB* tool and the expression `YEAR("OrderDate")` to extract the *OrderYear*, and we use *SalesAmount* as is but change the *Type* to *Float* to derive the field *SalesAmountValue*.

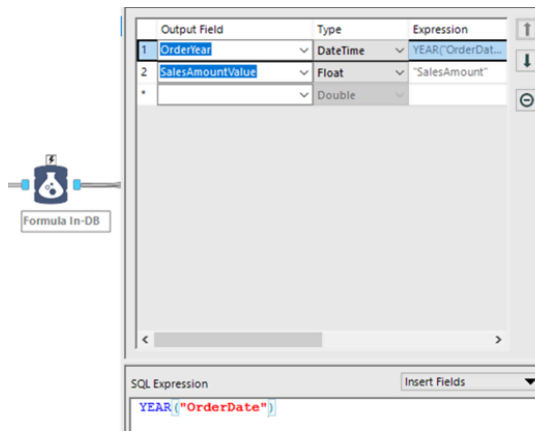


Figure 7-30 Formula In DB Configuration

According to the requirement, we'll have to aggregate the *Order Quantity* separately and the *SalesAmount* separately by year, later combining the data into one output. For now, let's branch the analysis into two streams: one for quantity and one for sales amount.

The two streams will be similar, both with select *EnglishProductName* and *OrderYear* using the *Select In-DB* tool. The only difference is that the quantity flow will also select the *OrderQuantity* field, and the Sales Amount flow will also select *SalesAmountValue*. We will be renaming these two numeric fields as *MetricValue* in the *Select In-DB* tool to enable us to combine the data in the future.

Now that we have chosen the select few fields we need, we will aggregate the *MetricValue* in their respective workflows. Similar to the previous task, add *EnglishProductName* and *OrderYear* to *GroupBy* Action, and add *MetricValue* to *Sum* Action under the *Numeric* option. That will roll up your *MetricValues* to total values per year per product.

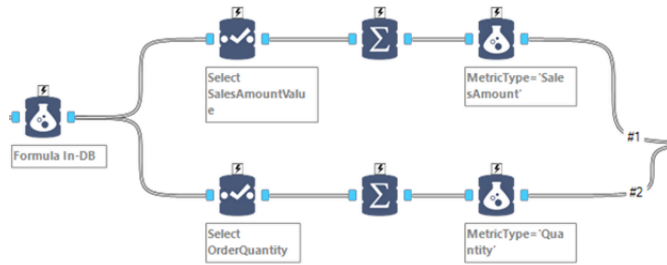


Figure 7-31 Superstore Database Report with Macro Data Stream

Fields:		Field	Type
▶	EnglishProduct...	V_WString	
	OrderYear	DateTime	
	MetricValue	Float	

Actions:			
Field	Action	Output Field Name	
▶	EnglishProduct...	GroupBy	EnglishProductName
	OrderYear	GroupBy	OrderYear
	MetricValue	Sum	Sum_MetricValue

Figure 7-32 Summarize In DB Configuration

Fields:	
Field	Type
EnglishProduct...	V_WString
MetricValue	Int16
OrderYear	DateTime

Actions:			
Field	Action	Output Field Name	
EnglishProduct...	GroupBy	EnglishProductName	
OrderYear	GroupBy	OrderYear	
MetricValue	Sum	Sum_MetricValue	

Figure 7-33 Summarize In DB Configuration

Before we go on to combine the two aggregated streams, we need to be able to identify what sort of metric exists at each row. To enable that, we will use a *Formula In-DB* tool to add a fixed string value to each of the streams. The field's name will be *MetricType*, fixed to the value *SalesAmount* for the *SalesAmountValue* stream and *Quantity* for the *OrderQuantity* stream.

Output Field	Type	Expression
1 MetricType	Double	Quantity
*	Double	

SQL Expression
'Quantity'

Figure 7-34 Formula In DB Configuration

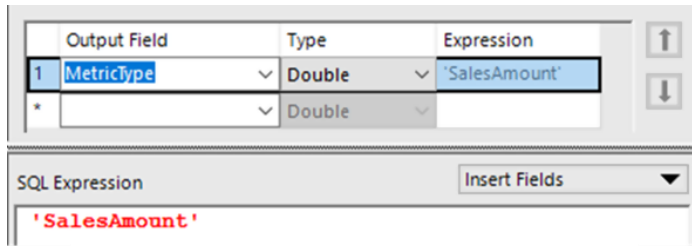


Figure 7-35 Formula In DB Configuration

You'll notice how the two streams now contain four columns with the same names: *EnglishProductName*, *OrderYear*, *MetricValue*, and *MetricType*. Having the very same columns lets us easily combine the two data sets using the *Union In-DB* tool. That tool reads columns from both of the data sets and stacks the data from similar column names on top of each other. While using the *Union In-DB* tool for this workflow, you can let the default tool configuration remain.

Finally, as we have our final output, we can exit the macro using the *Macro Output In-DB*. Proceed by saving the file with a *.yxmc* extension, which identifies it as a macro file.



Figure 7-36 Union In DB and Macro Output Data Stream

The final macro workflow should look like the workflow below (some connections have been made wireless for clarity).

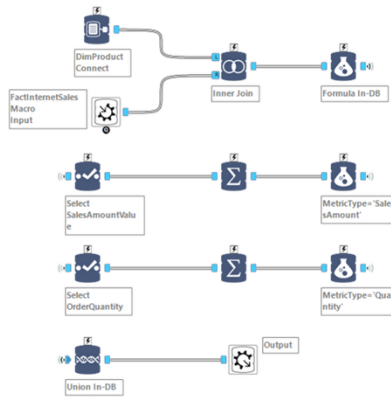


Figure 7-37 Superstore Database Report with Macro Data Stream when Complete

Start a new workflow file of the extension type `.yxmd`. That will be the main workflow Pete will interact with. Add the *Connect In-DB* tool to the workflow, and point it to *FactInternetSales*. That will mirror the fields that the *Macro Input In-DB* tool had been configured to read earlier.

Right-click anywhere on the workflow screen, and select *Insert>Macro*. You should be able to see the name of the macro you created here. Select the macro name, and it should add a nondescript circular icon to your workflow. That encapsulates the entire logic written earlier in the macro file. Finally, add a *Browse In-DB* to the workflow to enable Pete to see the results on refresh.

This simple workflow should look like the workflow below.



Figure 7-38 Superstore Database Report with Macro Data Stream with Macro UI when Complete

CHAPTER 8

Alteryx for Good

8.1 What Is Alteryx for Good?

Alteryx for Good (“AFG”) is an initiative run by Alteryx for Charities and Non-Government Organizations around the world. The program is a way for Alteryx to give back to the community at large and make lives of various individuals and organizations easier and more productive. The initiative is driven by a volunteer network of Alteryx experts who leverage their analytic expertise to non-profit organizations and educators to drive social change through the power of sharing.

The initiative is known around the world for benefitting organizations in multiple ways and generating greater profits and reduced costs. In the following sections, we will highlight some of the use cases that have been solved by the team in India using Alteryx in order to benefit the charities in Bangalore.

8.2 Alteryx for Good in India

The program was kick-started on March 14, 2017 in Bangalore, India. The initiative was accepted by the Alteryx User Group members, with two Alteryx ACEs who graced the occasion. One of them was Mark Frisch, aka Marquee Crew; Alteryx community’s celebrity and Sean Adams who is extremely passionate about all things Alteryx.

The event was a great success, and since that day, various charities have come on board in the program. We started off with Sneha Care Homes, an NGO that helps kids with HIV and AIDS lead a close-to-normal life. The event also kick-started a lot of activities at the charity that showcased the power of Alteryx and

the power of data digitization on their analytics journey. The event has been captured by the volunteers as an engaging post in the Alteryx community at community.alteryx.com/t5/Alter-Nation-Blog/Alteryx-For-Good-Launch-Event-in-India/bap/143952.



Figure 8-1 - AFG event at Sneha Care Homes

The next charity to be a part of the program was the Sai Krushna Charitable Trust. The trust runs a school that's completely free for 300 underprivileged students along with a free children's home that houses 10 boys who have single parents or are orphans.

Apart from education and housing, the trust also runs a free primary health care center in the same village (Hosadoddi on

Kanakapura Road in Bangalore). The Sai Krushna Charitable Trust is run by a group of highly accomplished and dedicated volunteers, including personalities such as Vinod Cartic, Dr. S. Rangarajan (former Director of ISRO), Mrs. Vijayalakshmy (passionate educationist for 30+ years), and many others. Vinod is an alumnus of IIT Madras Chemical Engineering (Gold Medalist) and an MBA from Sri Sathya Sai Institute of Higher Learning.



Figure 8-2 – AFG event at Sai Krushna Charitable Trust

8.3 What Activities Does the AFG India Chapter Do?

The AFG India chapter is involved in helping charities make their processes better. We have made multiple visits to the charities in

order to understand their mission, their challenges, and how we can contribute to them.

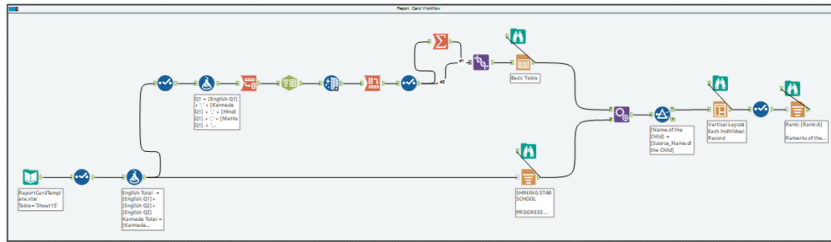
In the case of the Sai Krushna Charitable Trust and Sneha Care Homes, the AFG Team identified multiple areas where we can contribute. For the school, we have worked on report cards, student ID cards, and certificates—athletics, cultural, and academic for various age groups—for a variety of extracurricular activities at the school. For the hospital clinic, we have worked on creating an application that will help retrieve data for patients based on their previous history at the clinic.

There are other initiatives we have been part of in terms of volunteering, including painting the school, organizing the library, and mentoring the students and teachers on Alteryx and Excel.

We will now explain the use cases and the Alteryx solutions the AFG team has provided. We will also discuss the impact these Alteryx workflows have had on the charities and the future scope for any enhancements.

8.3.1. Report Card Solution

Prior to AFG, the report cards were generated manually by updating the data on multiple sheets of paper and then collated by manually calculating the total score for each student in all classes. That was a laborious task that took multiple days for the staff to accomplish and required them to work additional hours after school.



The workflow was designed by several volunteers after an hour of discussion with one of the senior teachers, the principal, and Vinod. The workflow was then developed in just two hours, and the output generated was a PDF report for each student. The report included a header with the school name, address, and logo. The student’s details, including name, parents’ names, class, and class teacher’s name, were also captured at the top of the report.

The bottom section of the report provided details of the grades in various subjects based on the class and the total grades, along with a percentage calculation for that period (monthly, semi-annually, annually). The report card is now ready to be printed on paper and shared with the student’s guardian.

8.3.1.1. Impact

The school has benefitted largely from time and effort saved for the teachers and administration. The teachers now have additional time to invest in teaching and bettering the students. These efforts have led to students winning top accolades at various district-level and national-level competitions. One such special accolade was winning first prize at ISRO’s national science day Essay Competition. That is just one of the things the students and teachers have done in their extra time. The report card has been in use over the past year, which has enabled

teachers to identify the needs of the students and hone their skills.

8.3.1.2. Enhancement Opportunities

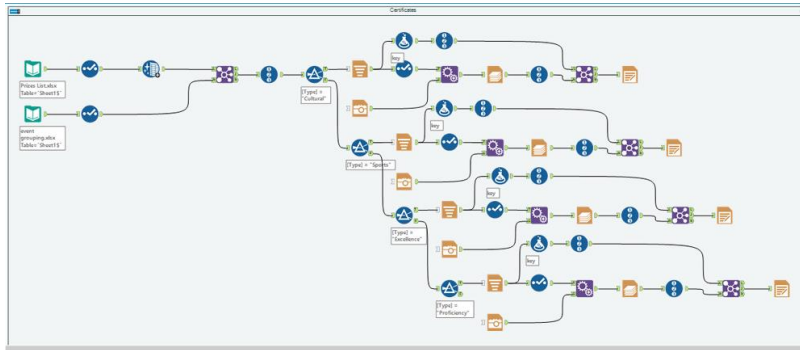
Currently, the school is moving to include a peer-to-peer learning methodology, which is currently being implemented under the guidance of Dr. S. Rangarajan and Mr. Ravikrishnan (former AVP at BGR Energy). The new report card shows the students' grades as well as the grades for the peer group that each student is part of. The students' performances will be monitored to help improve their academic skills.

Another aspect that is being worked on is the concept of sharing the report card with parents in a message or notification on their mobile phones. The approaches currently being researched are using a platform such as Twilio to send text messages to parents; however, most parents do not have smartphones on which they can access e-mails or web notifications.

8.3.2. Certificates Solution

The Sai Krushna Charitable Trust organizes competitions for the students throughout the year in multiple disciplines such as athletics; team games such as cricket, volleyball, and football; and cultural activities such as speech, acting, and debate. The students are then awarded certificates for their excellence in these activities. The certificate templates were mass-printed by providing a design, the students' names, the competition's name, and the positions (first, second, third, etc.), and then they were handwritten by the teachers, volunteers, and sometimes senior students of Classes 8 and 9. The most difficult part, which was shared with us as part of this exercise, is the principal, who has

to sign all of them manually. It took them 30–40 hours to complete the certificate printing and update the details.



With the Alteryx solution, all they have to do is maintain just one sheet with all the information, upload it into a designated folder, and execute the workflow. The certificates are now customized based on the event category (athletics, academics, individual sports, and team sports) and include a digital signature of the school’s principal.

8.3.2.1. Impact

The certificates are now prepared to be printed by updating the Excel spreadsheet and then distributing them to the students. Our certificates were used in the Annual Day celebrations in November 2018. The students liked the certificates for the beautiful images and nice font in which their names were printed. Some of the trust’s volunteers also pooled their money to buy a color printer so the complete process can now be completed in house. The best review of this task was provided by the school’s director (Mrs. Vijayalakshmy). She mentioned that this is something revolutionary that she has never seen before in her career.

8.3.2.2. Enhancement Opportunities

The certificates workflow mentioned above is also being upgraded to identify students who have been doing well in specific fields throughout the year. Students are being mentored for professional sports, and students are being identified for their talents in drama and writing and then being sent for interschool competitions. These efforts could open up a whole gamut of opportunities for these students.

8.3.3. Donation Receipts Solution

All charities depend on donations from interested people and organizations. One of the most repeated tasks for charities is providing receipts for donations, which are used for income tax or other purposes.

The AFG team has created a solution that takes from Excel the details of the donor, the amount donated, and the mode of transfer and embeds them in a standard form and then prints it as a PDF report. The workflow also has the capability to track the last row for which the output was created and work on the next row based on the date modified in the output file and also track the donation date.

8.3.3.1. Impact

The impact of the task at Sai Krushna Mandir was easily apparent since the charity used to get multiple donations every week, and all Vinod had to do was capture the details on an Excel sheet and print it in a batch or individually. This task used to take 10 minutes for each receipt, and now, receipts for the entire month can be done in less time than that.

8.3.3.2. Enhancement Opportunities

While the donation receipt solves the administrative overload, there is an opportunity to possibly use the financial data to track donation trends and send custom greeting cards that thank the donors for their contributions. In some cases, donors would like to hear how their donations have been used. That could easily be done by sending them the annual report for Sai Krushna Mandir every year, as well as sending them monthly newsletters that include information of the school's latest happenings. That would keep them aware of events and open up opportunities for future donations either from them or their families, relatives, or colleagues. The AFG team is also looking to integrate the workflow with bank reports.

8.3.4. ID Card Solution

The Sai Krushna Charitable Trust School is a private school affiliated with the government. The students and teachers previously did not have ID cards. Alteryx for Good volunteers noticed that when they interacted with the students a few times. They took on the challenge to create ID cards and do it in a way that could be easily sustained by the charity.

The solution required them to first take pictures of all the students and teachers, which was then mapped to each student and teacher using their identification numbers. Once this exercise was completed, the ID card template was formulated after collaboration with Vinod, Mr. Ravindranath (the school principal), and some of the teachers. The ID card now contains the name of the teacher or student, the parents' names, their class, their blood group, and an emergency contact number. The ID

card also contains the school's name, the school's logo, and the student's picture.

8.3.4.1. Impact

The ID card is now a thing of pride for each of these rural students, making them feel equal to the kids who go to international schools. Students wear their IDs with a lot of pride and also feel proud to be part of the school. The teachers also have pride in their school and feel more at par with teachers at other schools. This confidence and attitude translate to more effort by the teachers and the students.

8.3.4.2. Enhancement Opportunities

The ID card solution is comprehensive; however, there are some areas where improvements can be made. The ID cards can also be used with access cards, and data from that can then be used to track student attendance at school. The data about the number of registrations each year can be used to analyze the number of new students each year and look at student attrition and possible reasons for attrition.

All these solutions have been implemented at Sneha Care Homes as well by customizing it for their data.

8.3.5. Health Card Tracking Solution

In Sneha Care Homes, we have a unique scenario where the health and medical history of each student is kept in a notebook. Their medication dosages, the data on their nutrition, and the activities they need to be doing are also kept in that notebook. This notebook is important for regulatory purposes.

However, there was no easy way to notate the trends and improvements of the students. That became our problem statement, and we began solving it by initially digitizing the existing data and providing the school with a template they could fill in for future use and easier digitization. The next step was to create a workflow that captured all the inputs for a student and did the weekly and monthly comparisons presented in report format. This solution is still in the works and should be completed soon.

8.3.6. Excel Training and Digitization

In all the charities, one thing very evident is the lack of digitization and content readily available for analysis. That is important because a lot of their efforts go into administrative and academic exercises that require them to do lot of processes manually. The teachers and volunteers at the charities utilize Excel training so they can use it more effectively to digitize records and be able to do some manual tasks faster.

8.3.6.1. Impact

The Excel training has had a huge impact on the productivity of staff in both places. They are now able to think of solutions themselves for situations in which they can use digital records to better analyze and capture data for future reference. With the wide push for the Digital India campaign, most of the information today is being requested in soft copy format. That is helping staff become better at computers and able to do a lot of tasks faster.

One example that is noteworthy is the digitization of tests in the school run by the Sai Krishna Charitable Trust. Teachers now

have access to test banks for all subjects for grades three through seven. That makes their job of creating tests very easy since they can use the test bank at the click of a button.

8.3.6.2. Enhancement Opportunities

Excel training is an ongoing activity we are planning to do on almost every visit to the charity. On certain visits, we also trained some of the staff on Alteryx so they can access workflows and apps, run them with minimal input from us, and get their reports.

8.4 How Can You Join or Reach Alteryx for Good in India?

The Alteryx for Good India team is currently a subset of the Bangalore Alteryx User Group. The initiative is currently run by the User Group leads. The group can be reached at blralteryxusergroup@gmail.com for any queries or at [@BLRALteryxUserGroup](https://twitter.com/BLRALteryxUserGroup) on Twitter. The user group leads are Subrat Das and Yugandhar Muley.



Figure 8-5 – Bangalore Alteryx User Group Leads

Alteryx enthusiasts have made a wave across the AFG world by organizing various events at charities. They have won the

Alteryx Excellence Awards for their solutions and have been mentioned in Alteryx Inspire 2018.

8.5 How Can You Join Alteryx for Good in Your Area?

The Alteryx for Good initiative is now gaining momentum around the globe, a growing interest to know more about the use cases that Alteryx have solved is a testament to this fact. When it comes to solving problems within the non-profit sector, many corporates and individuals are more than willing to go that extra mile – working towards an ecosystem that gives more than it takes. The satisfaction of seeing a child smile is sometimes a greater reward than spending four hours binging on TV. The Alteryx for Good initiative is typically run by a subset of the regional Alteryx User Group.

CHAPTER 9

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 them on your own, which is the goal of these exercises. The screenshots of the solutions are provided so if you are stuck, you can refer to the logic of how the examples direct you to a solution.

9.1 The Games: How Are We Doing?

Since the complete workflow is large, it is broken down into two parts for better understanding.

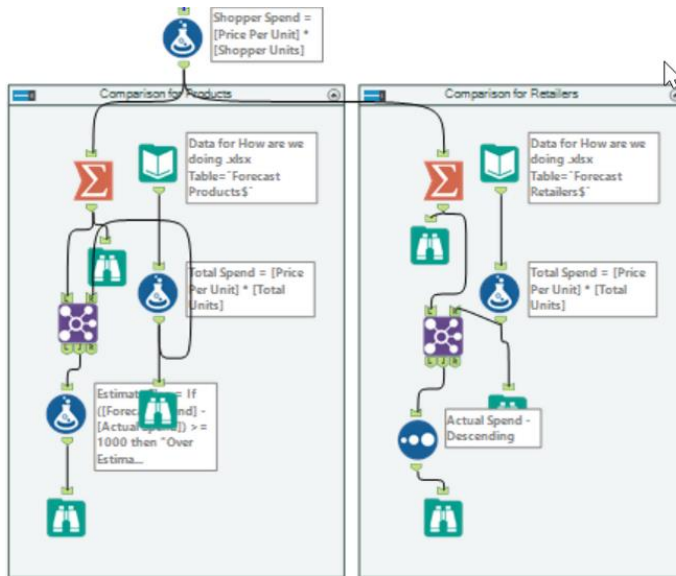


Figure 9-1 How Are We Doing? Part 1

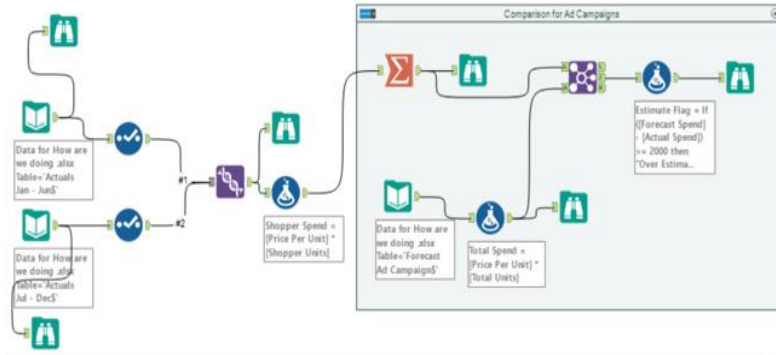


Figure 9-2 How Are We Doing? Part 2

9.2 Unisex Baby Names: What's in a Name?

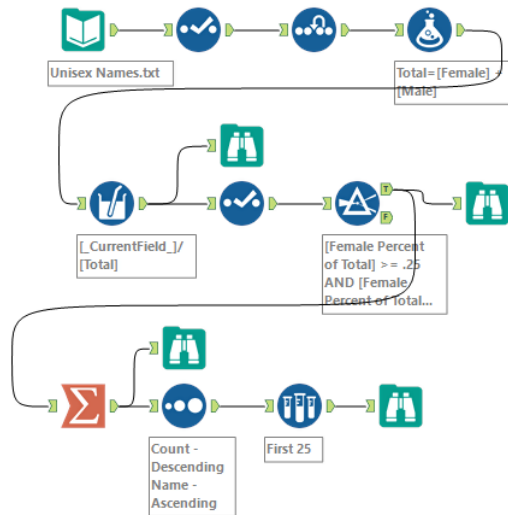


Figure 9-3 What's in a Name?

9.3 The Direct Approach: Where, Oh Where Have My Three Files Gone?

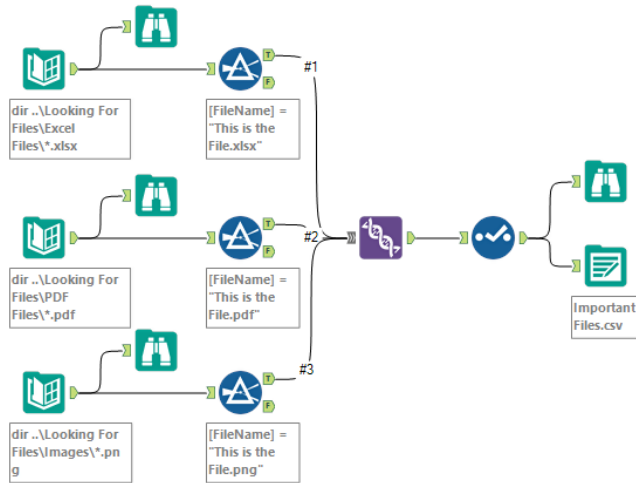


Figure 9-4 Where Have My Three Files Gone?

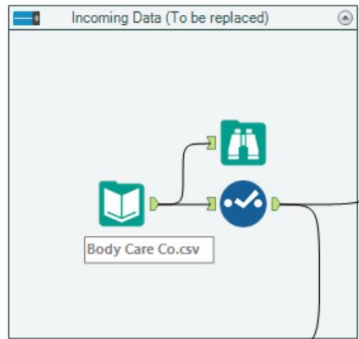


Figure 9-7a Incoming Data

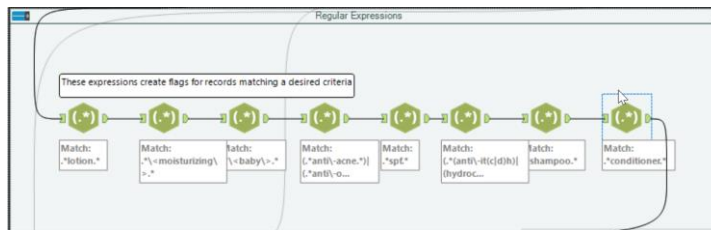


Figure 9-7b Regular Expressions

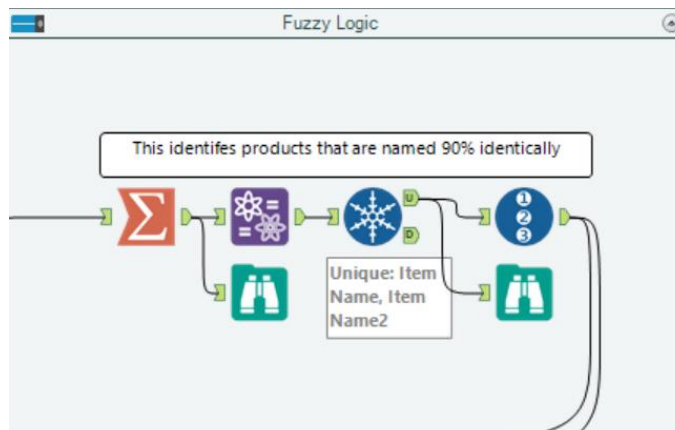


Figure 9-7c Fuzzy Logic

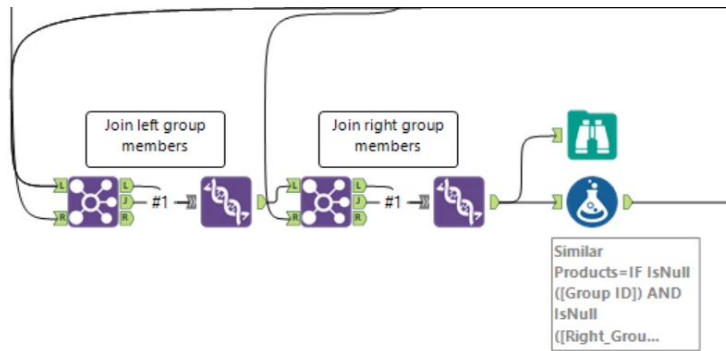
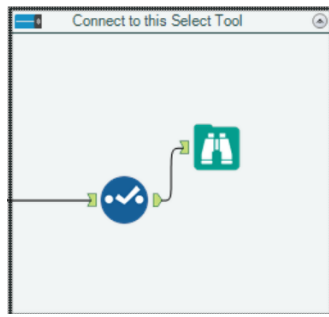


Figure 9-7d Join and Group



Appendices

Appendix A – File Types

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

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 <i>Browse</i> tool after all selected tools that a <i>Browse</i> 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

U.S. 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

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 8,192 characters
WString	String	A field with up to 8,192 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

Appendix D – Action Tool Sets

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 – Boolean Expressions

Appendix E – Boolean Expressions

Expression	Meaning
$A = B$	Expression A equals expression B
$A \neq B$	A does not equal B
$A > B$	A is greater than B
$A \geq B$	A is greater than or equal to B
$A < B$	A is less than B
$A \leq B$	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 F – Data Components

Appendix F – Data Components

yyyy-MM-dd	
yyyyMMdd	
MM/dd/yy	day - Spelled out weekday
MM/dd/yyyy	dd - 2 digit day of month
MM-dd-yy	dy - Abbreviated day of week
MM-dd-yyyy	hh - 2 digit hour
Month dd, yyyy	mm - 2 digit minute
Month, yyyy	MM - 2 digit month of year
Mon dd	Mon - Abbreviated month of year
dd/MM/yy	Month - Spelled out month
dd/MM/yyyy	ss - 2 digit second
dd-MM-yy	yy - 2 digit year
dd-MM-yyyy	yyyy - 4 digit year
dy., Month dd, yyyy	
day, dd Month, yyyy	
dd Month, yyyy	
yyyy-MM-dd hh:mm:ss	
MM/dd/yyyy hh:mm:ss	
MM/dd/yy hh:mm:ss	
dd/MM/yyyy hh:mm:ss	
dd/MM/yy hh:mm:ss	

Figure 12-12 – Date Components

Appendix G – RegEx Cheat Sheet

Appendix G – 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 H – Action Tool Sets

Appendix H – 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.
AttributeRespasePoint	This indicates that the file is associated with a repase 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.

Data Sources

Credited Original Data Sources

The Games

- Medals:
 - 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:
 - 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:
 - 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:
 - 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:
 - List of U.S. states by GDP per capita. (2015, March 28). In *Wikipedia, The Free Encyclopedia*. Retrieved 00:01, May 12, 2015
[fromhttp://en.wikipedia.org/w/index.php?title=List_of_U.S._states_by_GDP_per_capita&oldid=653869588](http://en.wikipedia.org/w/index.php?title=List_of_U.S._states_by_GDP_per_capita&oldid=653869588)
- Museum Data:
 - 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:
 - Created by author

Data Sources

Expensive Beauty Products

- Created by Author:

Let's Do It (In-DB)

Data was obtained from Adventure Works by using a sample from the below data set:

<https://msftdbprodsamples.codeplex.com/downloads/get/165405>

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/>.

About USEReady

USEReady's mission is to help users succeed with data.

We achieve this mission with fanatical customer centricity, humility, and integrity. At USEReady, we love to solve customer problems, contribute to community and continue to improve. We aim to build a strong learning culture and have fun doing it together.

We have harnessed modern Business Intelligence solutions with cutting edge Artificial Intelligence (AI) platforms to unlock the power of data. Our result-oriented solutions ensure that our customers are successful in adopting self-service technologies.

We have partnered with the best in class BI, Data and Cloud products such as Tableau, Snowflake, Alteryx, Informatica, Alation, Collibra, AWS and Microsoft. We have delivered proven success across Financial Services, Insurance, Retail and Media verticals.

We have been nominated and won several awards along this journey. Check us out at www.useready.com

Authored By

→ Deeksha Premchand Rao → Sachet Kashyap

→ Abhilash Ramanathan → Priyanka Dobhal

→ Meehir Mene → Harshita Paul

Thanking Subrat Das, Murtaza Farooqui and Sharang Kulkarni for all the support.