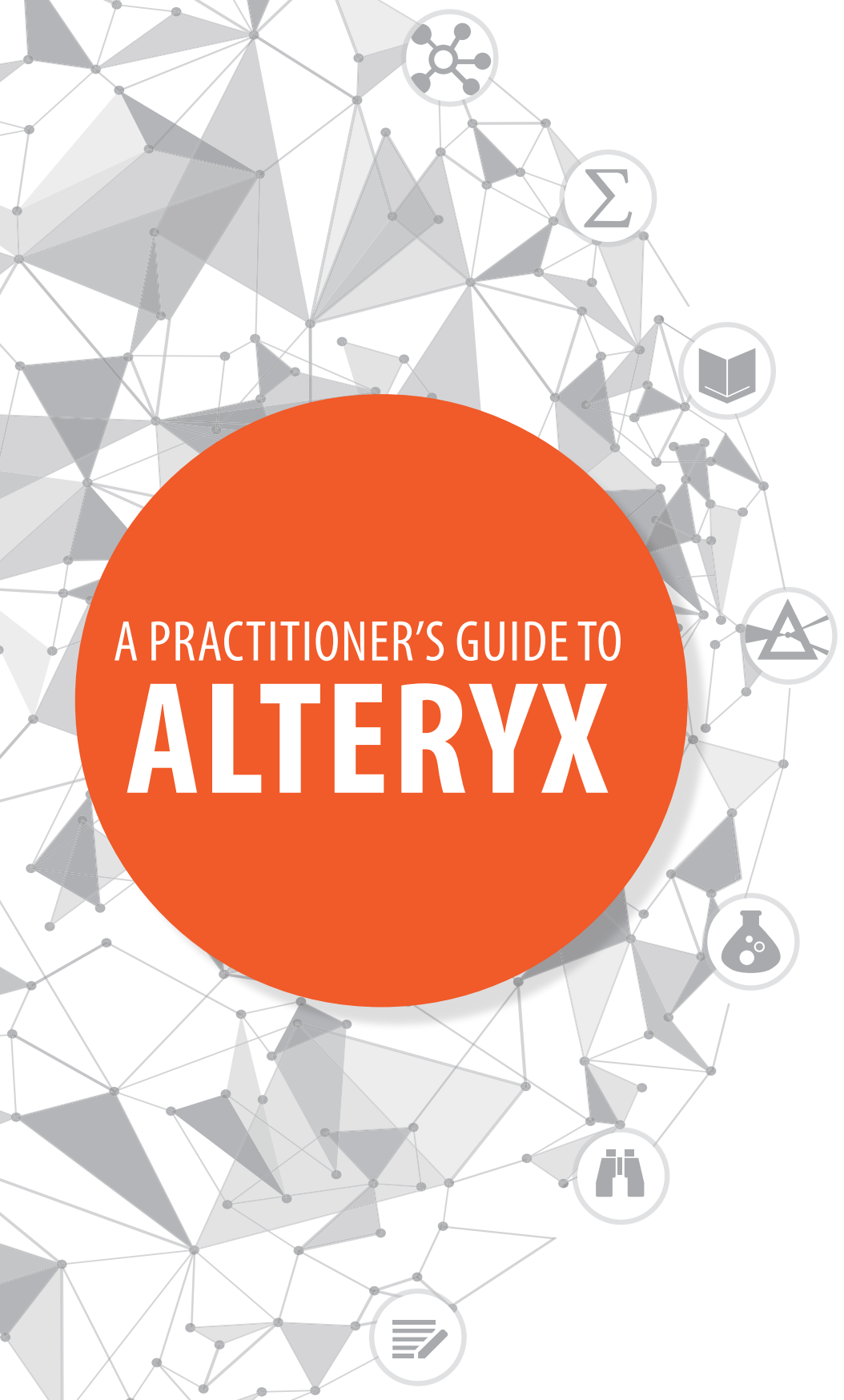


USEREADY

A PRACTITIONER'S GUIDE TO ALTERYX

Alteryx 10

A Practitioner's Guide to Alteryx®

Published by
USEReady
469 7th Ave Suite #745, New York, NY - 10018
www.useready.com

ISBN: 978-0-692-44795-6
Manufactured in the United States of America

About USEReady

USEReady is a leading provider of advanced business intelligence, data management, and network security solutions. Our mission is to help businesses succeed by fast-tracking their business performance - through our digital world ready solutions, industry and technological expertise, agile global delivery practices, and our customer fanaticism.

Alteryx Solutions

Being their long-term partner, we understand Alteryx like no one else. Staying ahead of the rapidly evolving data-scape, USEReady has developed a specific, unified and democratized approach to data management to ensure quality, accelerated decision-making and collaboration. From profiling to optimization, we leverage Alteryx's powerful capabilities to address the biggest challenge faced by analysts today – how to access the right data quickly.

Our Alteryx expertise includes:

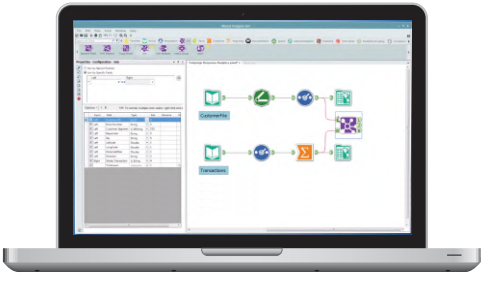
- Domain specific data preparation
- API-driven Extract-Transform-Load
- Reusable data preparation applications
- Analytics-oriented prepared answer sets
- Cloud and on-premise integration
- BI-ready modern data warehousing

Alteryx Training

We aim to give you the best chance of understanding the capabilities of Alteryx and make analytics interactive and explorable. You can choose from: virtual class-room or instructor-led, on-site format; as well as from Beginners, Intermediate or Advanced Training in Alteryx. We also offer customized Alteryx courses.

Our training will enable you to:

- Access, clean and blend data from multiple sources
- Join data sources of different types
- Create workflows, macros, and applications
- Prepare and deliver data sets for visualization
- Perform advanced predictive and spatial analytics



USEREADY

For more information, visit us at: **www.useready.com**, or write to us at: **contact@useready.com**

tableau SERVICES PARTNER OF THE YEAR 2015 USEREADY

RED HERRING N. AMERICA 100 WINNER

AMERICA'S FASTEST-GROWING PRIVATE COMPANIES #113 Inc. 5000 2015

About Alteryx, Inc.

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

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

Acknowledgements

When we started writing this edition of the book, we decided to go with a team of authors instead of a single author. The team went through a rigorous iterative process of reading, updating and reviews to get the book where it is today.

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

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.

Vijai Narasimha: For being one of main drivers of the project and putting in those long hours updating and reviewing the content.

Abhilash Ramanathan: For being a leader and helping the team with a very fine attention to detail.

Vijay Gowtham: For putting up with meetings at odd hours and being that much needed funny bone during our meetings.

Rahul Upadhye: For putting in endless hours of work in editing the book and always being committed.

Ashish Tergaonkar: For his enthusiasm in editing the book and his constructive suggestions during the meetings.

Rima Upadhye: For being involved despite a very busy schedule.

Prashant Singh: For creating marketing, a web presents, and an excellent cover design.

- Jagdeesh Narayanan

Letter from the CEO

First of all, we would like to thank our readers for making the first edition a grand success that we are super encouraged to attempt this edition. This edition is also a testament to growing popularity of Alteryx and appetite for self reliant solutions in the industry today.

We took a little bit of gamble with this edition. We invited a team of Alteryx practitioners within USEReady to contribute while enhancing the content with lot of experience and gotchas we learnt from the first edition. Jagdeesh Narayanan, Abhilash Ramanathan, Vijai Narasimha, Rahul Upadhye, Rima Upadhye, Anand Vadul, Vijay Gowtham, Ashish Tergaonkar and Prashant Singh have worked countless hours in shaping this book. The result has been phenomenal as you glean from the chapters.

Team has gone great heights to ensure the content is refined to the most recent version of Alteryx and exercises are useful to a fellow practitioner. With your valuable feedback we plan to augment the content to facilitate class-room like trainings.

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

Uday Hegde
Chief Executive Officer
USEReady

Foreword

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

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

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

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

Laura Sellers
Vice President, Product Management of Alteryx

Preface

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

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

What Is Data?

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

What Does Data Do?

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

How Do We Consume Data?

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

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

What Is Data Preparation?

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

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

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

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

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

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

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

Combination

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

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

Calculation

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

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

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

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

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

Transformation

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

To the reader

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

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

The exercises will use data that can be download from https://resources.useready.com/publications/a-practitioners-guide-to-alteryx-alteryx-version-10/ by following the instructions on the website to unpackage the file.

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

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

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

Best of Luck,

USEReady

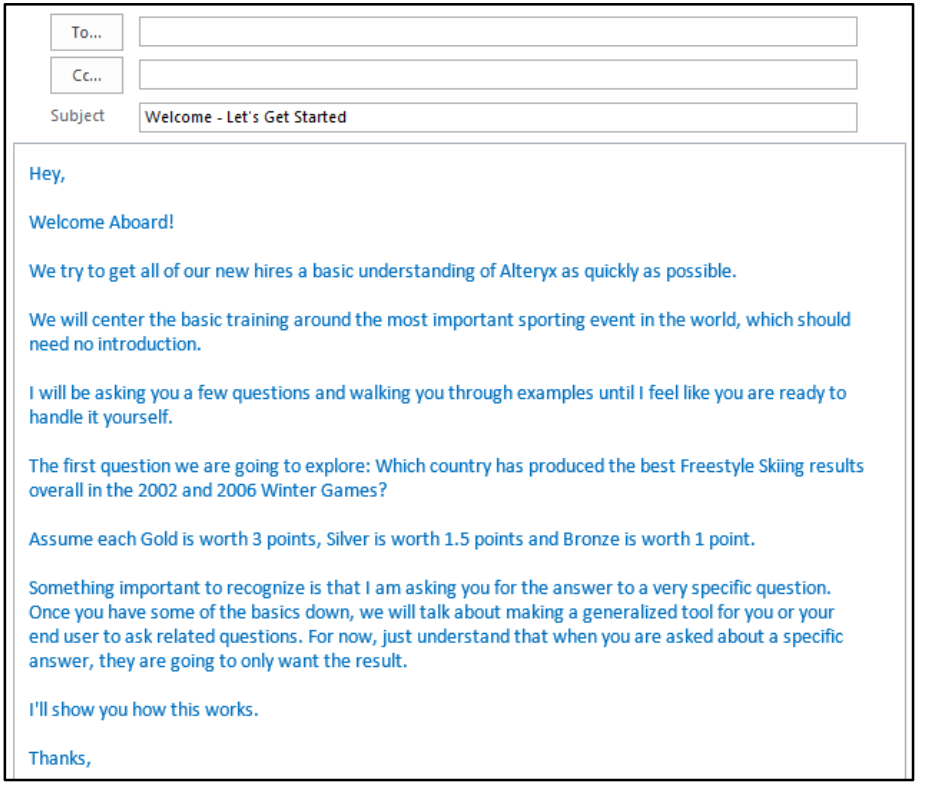
Table of Contents

CHAPTER 2

The Games

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

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



2.1 Tools & Concepts

In this chapter, we will cover the improved features for Alteryx designer 10.x and the various tools and concepts as mentioned below:

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

2.2 Improved Features

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

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

- Changes to the Connect In-DB tool include: new streamlined UI, new option to allow password decryption, and new option for creating file-based In-Database connections in order to simplify obtaining connection information from IT Admin/DBA for server users.
- Changes to the Data Stream In tool include: new option to allow password decryption, users can write out to a permanent table when streaming data from an external data source.
- New Macro Input In-DB and Macro Output In-DB tools have been added for In-Database processing so users can build macros with In-Database.
- New Dynamic Input In-DB and Dynamic Output In-DB tools have been added for In-Database processing so users can retrieve the underlying SQL query and other metadata info being sent to the database or to the In-Database tools in a workflow.
- These connectors are now supported: Amazon Redshift bulk load (write), Netsuite (read/write), Qlik (read), PostgreSQL 9.4 (read/write), SAP Hana (read/write), Spark SQL (read).
- Users can now read and write to JSON from the Input and Output Data tools.
- Users can now browse to their Hadoop distributed file system and read/write to HDFS via the new connections in the Input Data tool. (HDFS tools released in 9.5 have been deprecated).
- Salesforce and Marketo connector tools built on the REST API provide enhanced functionality. The existing SOAP-based connectors are now deprecated.

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

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

- The Block Until Done tool now has numbered outputs that output "in order."
- The Map tool now has a zoom/bounds option that makes the map zoom/pan to the reference file.
- In the Alteryx Designer version 10.x, the Configure Workflow and Results windows are split into two distinct views.

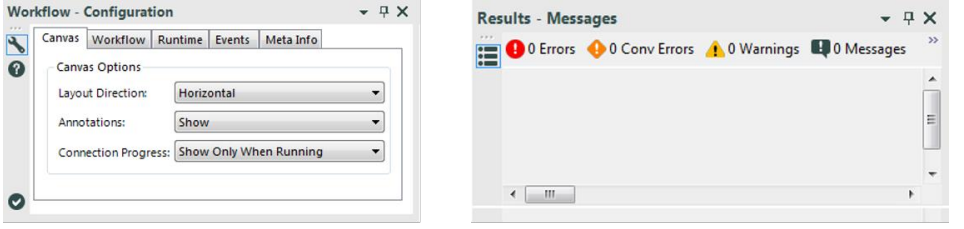


Figure 2-1 – Workflow and Results windows

- Configuration and Results windows can be dragged and displaced to different possible positions.

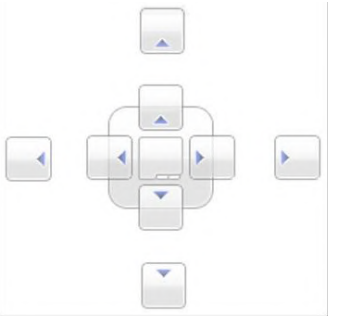


Figure 2-2 – Possible positions

- Every tool in Alteryx now has a Preview feature to show a snapshot of the data it contains.

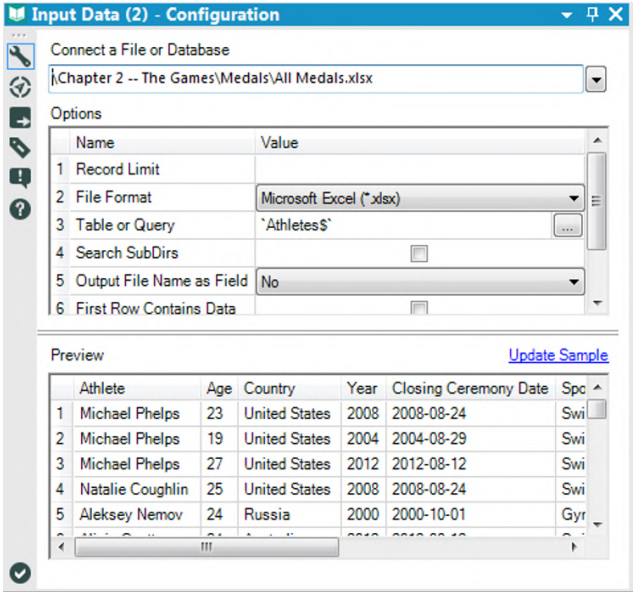


Figure 2-3 – Preview Data

2.3 Browse

Figure 2-4 - Browse

The *Browse* tool gives us a tabular view of the data in a data stream at the point it is connected.

Group	Input	Output
In/Out	Data stream	None

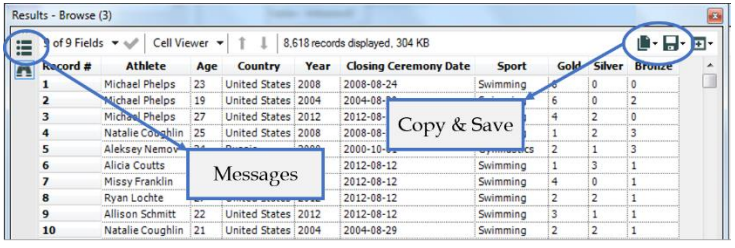
Note: It is very important for the development of workflows, applications, and macros but should be disabled when development is completed to improve speed.

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

Results Window:

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

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



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-		6	0	2
3	Michael Phelps	27	United States	2012	2012-08-		4	2	0
4	Natalie Coughlin	25	United States	2008	2008-08-		1	2	3
5	Aleksey Nemov	24	Russia	2000	2000-10-	Gymnastics	2	1	3
6	Alicia Coutts				2012-08-12	Swimming	1	3	1
7	Missy Franklin				2012-08-12	Swimming	4	0	1
8	Ryan Lochte				2012-08-12	Swimming	2	2	1
9	Allison Schmitt	22	United States	2012	2012-08-12	Swimming	3	1	1
10	Natalie Coughlin	21	United States	2004	2004-08-29	Swimming	2	2	1

Figure 2-5 – Browse, Results

The two icons on the right side (shown above) are *Copy* and *Save* functions. They allow us to directly copy the data out of the browse tool or save it to a file.

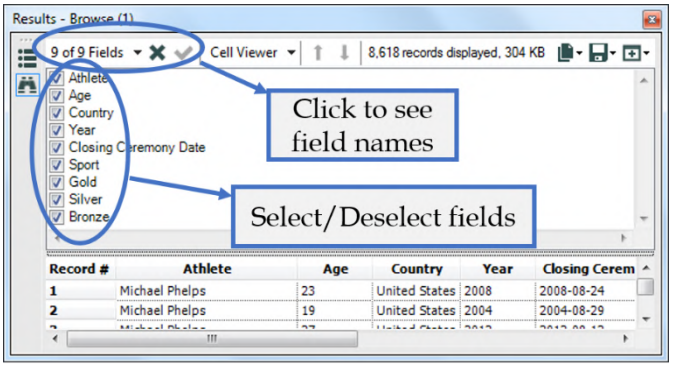


Figure 2-6 – Browse, Field Names

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

The text shown after clicking on *Cell Viewer* depends on what is selected in the records.

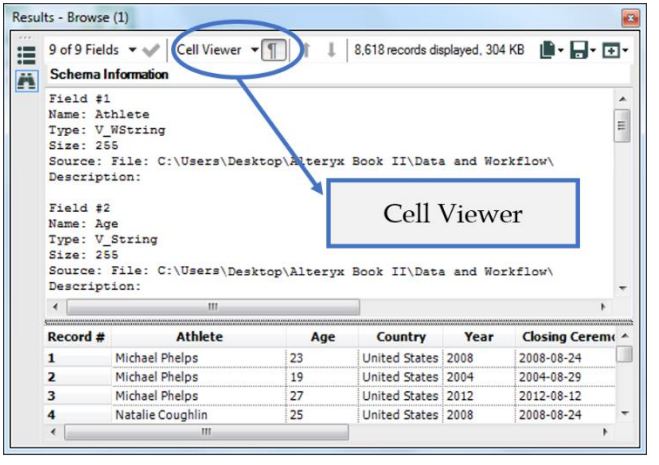


Figure 2-7 – Browse, Cell Viewer

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

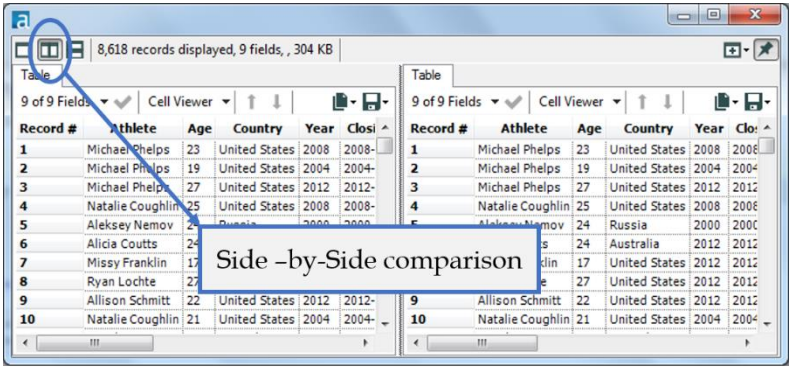
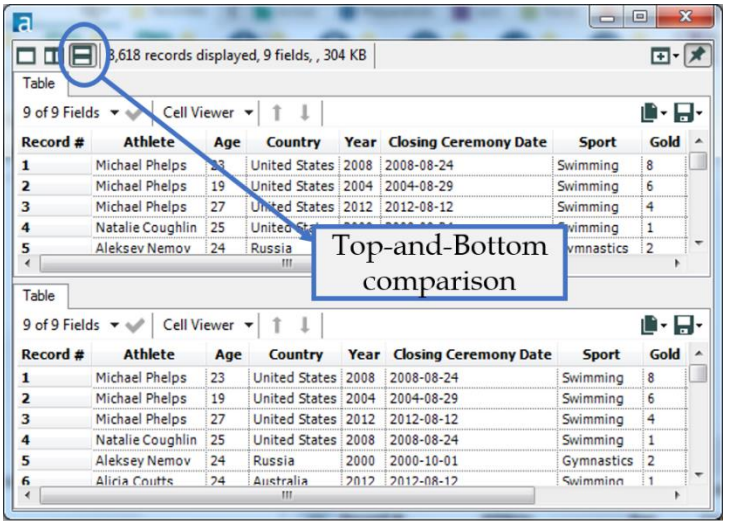


Figure 2-8 – Browse, Side by Side

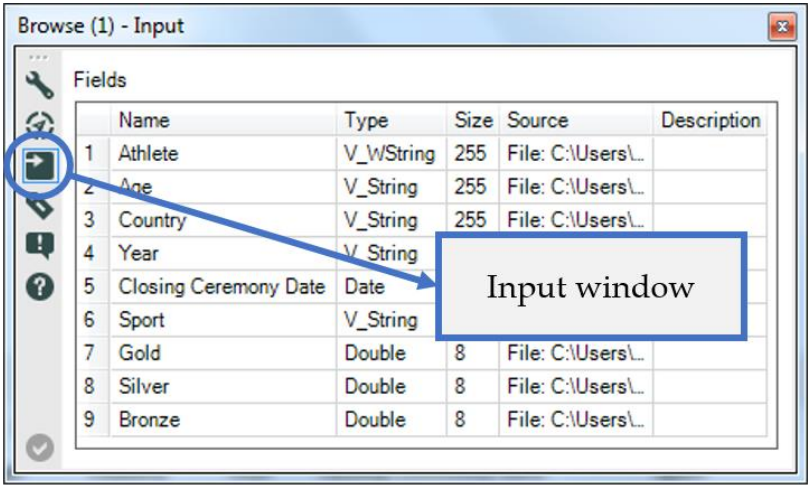
We have the ability to compare the dataset either Side-by-Side (vertical compare) or Top-and-Bottom (horizontal compare) by clicking on the new window button located at the top-right of the results window.



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

Figure 2-9 – Browse, Top and Bottom

You will also notice that we have an Input button on the Properties window side pane.



	Name	Type	Size	Source	Description
1	Athlete	V_WString	255	File: C:\Users\...	
2	Age	V_String	255	File: C:\Users\...	
3	Country	V_String	255	File: C:\Users\...	
4	Year	V_String			
5	Closing Ceremony Date	Date			
6	Sport	V_String			
7	Gold	Double	8	File: C:\Users\...	
8	Silver	Double	8	File: C:\Users\...	
9	Bronze	Double	8	File: C:\Users\...	

Figure 2-10 – Browse, Input

This is the *Input* window. It tells us information about each field that comes into the tool: the field name, field type, total size of each cell, the original source of the data, and a description of the field.

2.4 CrossTab

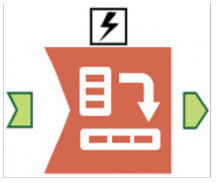


Figure 2-11 - Crosstab

The *Crosstab* tool creates a normalized (more human-readable) dataset by creating columns out of the rows of data.

Group	Input	Output
Transform	Any data stream	Data stream wider than input

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

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

Properties Window:

The Cross Tab Properties window has four components, as shown in the following figure.

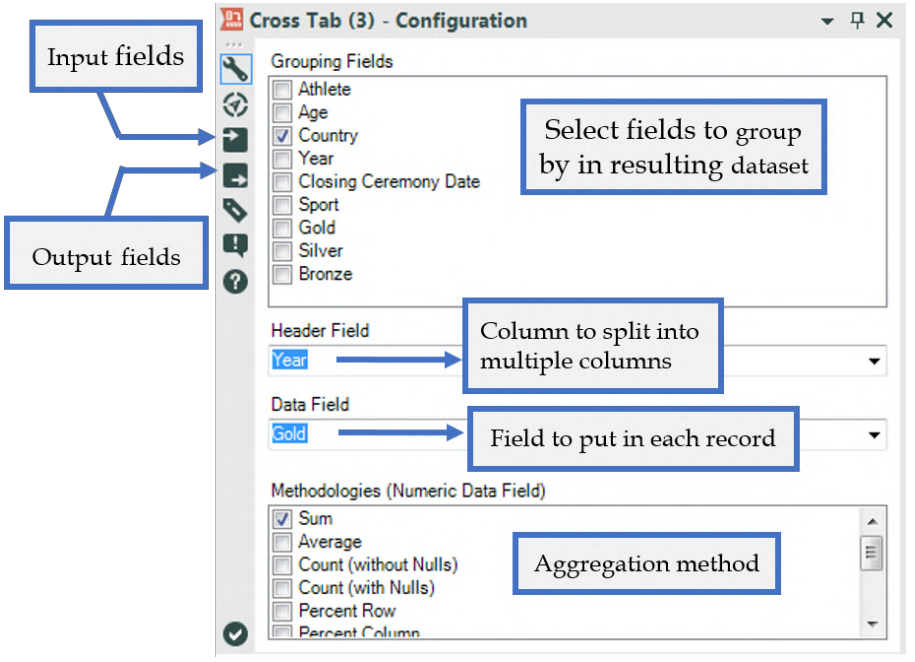


Figure 2-12 – Crosstab, configuration

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

- *Grouping Fields* allows us to select which fields we want to group by in the resulting dataset. If nothing is selected, we will only get a single record as output.
- *Header Field* is the field we are splitting into multiple columns.
- *Data Field* is the field we want to put in each record for the columns created by the Header Field.

- *Methodologies* (Numeric Data Field) allows us to select the type of aggregation method used if we have multiple data entries that fit into the same cell of the resulting dataset.

2.5 Comment

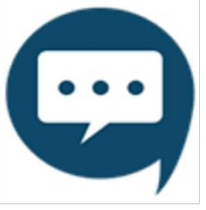


Figure 2-13 - Comment

The *Comment* tool gives us the ability to write notes on our workflows to add additional information on the data stream.

Group	Input	Output
Documentation	None	None

Note: Comment is an annotative tool to help give meaning to developers using this workflow.

Properties Window:

The *Comment* Properties window allows us to customize the comment field that appears on the canvas.

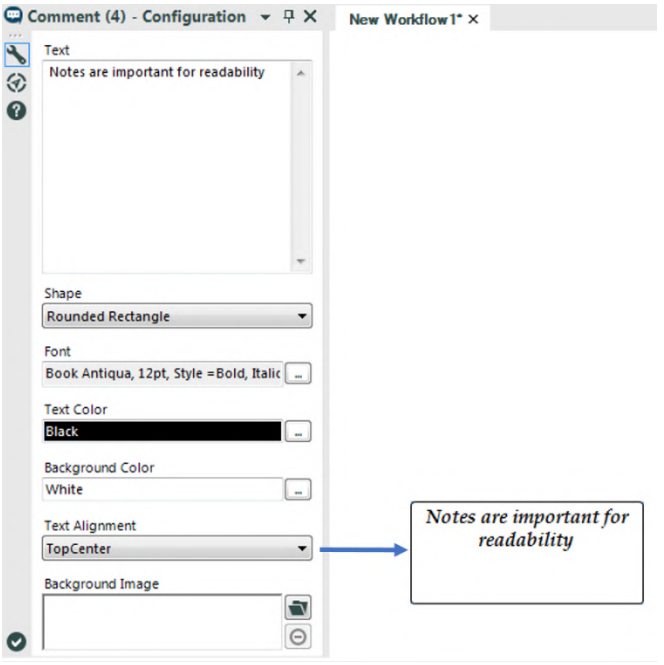


Figure 2-14 – Comment, configuration

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

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

2.6 Filter

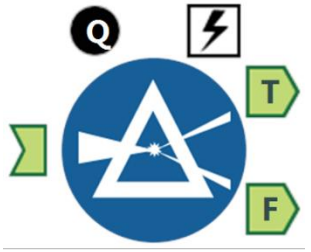


Figure 2-15 - Filter

The *Filter* tool gives us the ability to create a function that will split the data row by row into either the true or false outputs.

Group	Input	Output
Preparation	Any data stream	T & F section below

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

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

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

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

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

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

Properties Window:

We see that there are two different ways to create a filter: *Basic Filter* and *Custom Filter*. The regions for each are as shown below.

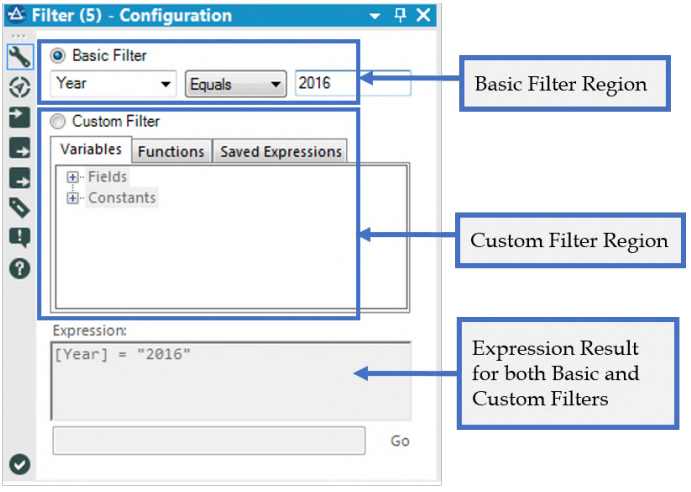


Figure 2-16 – Filter, basic filter

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

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

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

- *Custom Filter:* allows us to click on variables, function, and saved expressions to populate the expressions window or

type the formula directly. A sample custom filter is shown below.

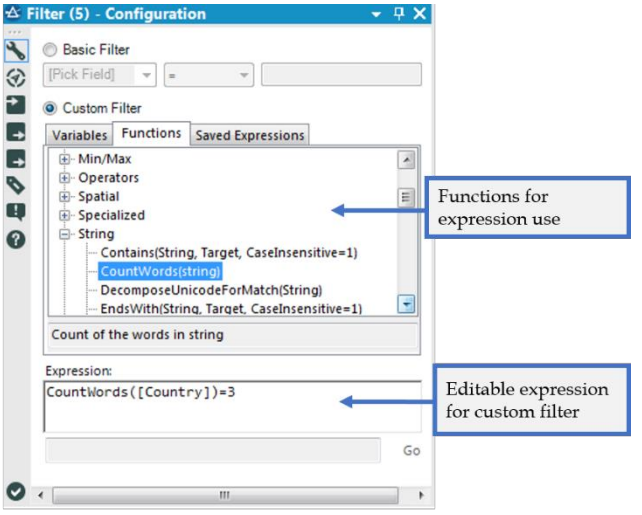


Figure 2-17 – Filter, custom filter

We will discuss creating formulas in the *Formula* tool and throughout the exercises in this book. On an error, the red Error symbol replaces the message symbol, as shown in the figure below.

Figure 2-18 – Filter, error view

2.7 Formula

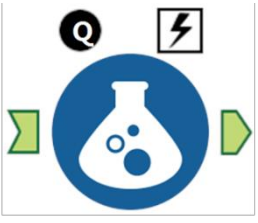


Figure 2-19 - Formula

The *Formula* tool gives us the ability to create a function that will be written to a new column in our data.

Group	Input	Output
Preparation	Any data stream	Augmented original data stream

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

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

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

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

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

Properties Window:

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

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

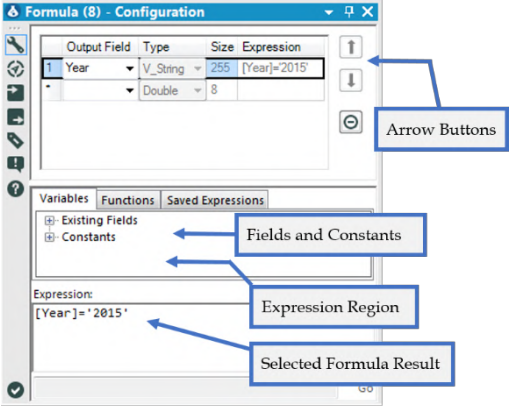


Figure 2-20 – Formula, options

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

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

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

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

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

2.8 Input Data

Figure 2-21 – Input Data

The *Input Data* tool gives us the ability to import data from specific databases.

Group	Input	Output
In/Out	None	Data stream in initial data format

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

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

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

Properties Window:

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

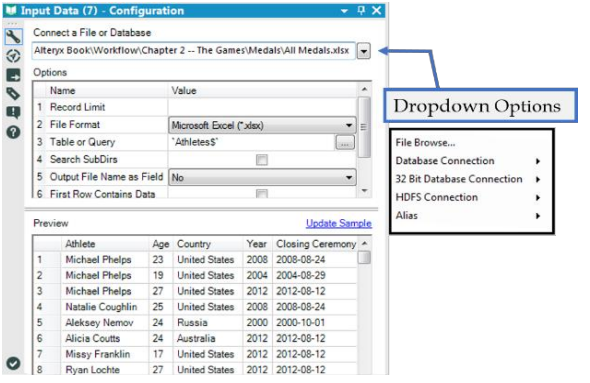


Figure 2-22 – Input Data, options

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

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

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

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

The image below describes each of the tabs available when clicking on the "…" button.

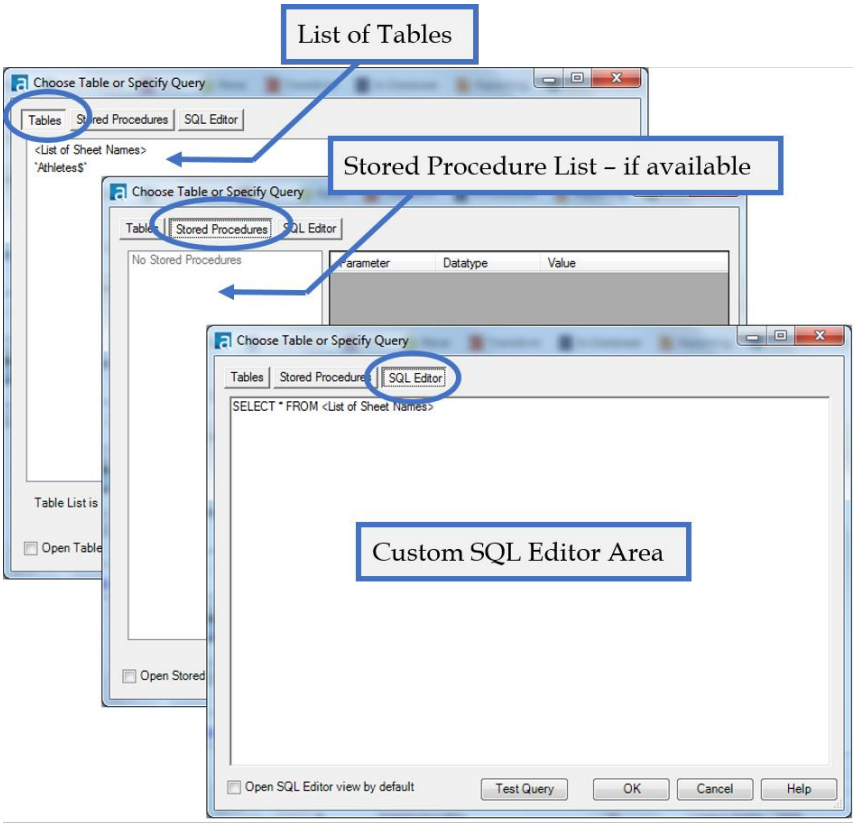


Figure 2-23 – Input Data, table and query options

2.9 Join

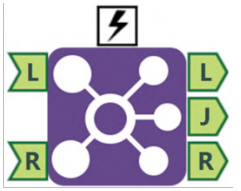


Figure 2-24 - Join

The *Join* tool gives us the ability to combine two data streams by lining up records based on matching fields.

Group	Input	Output
Join	See Input Left and Input Right	See Output Left, Output Join, and Output Right

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

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

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

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

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

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

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

Properties Window:

The *Join Properties* window has two major components.

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

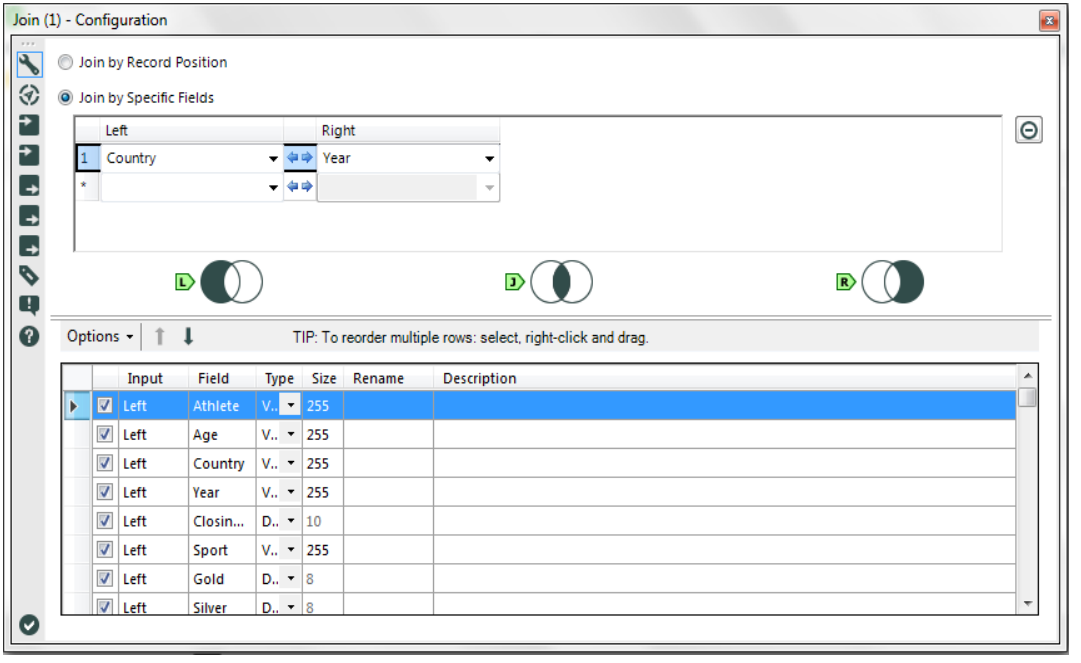


Figure 2-25 – Join Properties

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

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

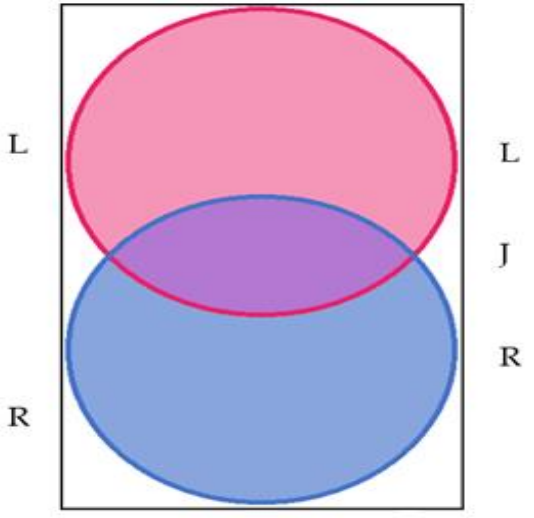


Figure 2-26 – Join Properties Venn Representation

2.10 Output Data

Figure 2-27– Output Data

The *Output Data* tool allows us to write the data stream out to a file or database

Group	Input	Output
In/Out	Any data stream	File Or Database

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

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

Properties Window:

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

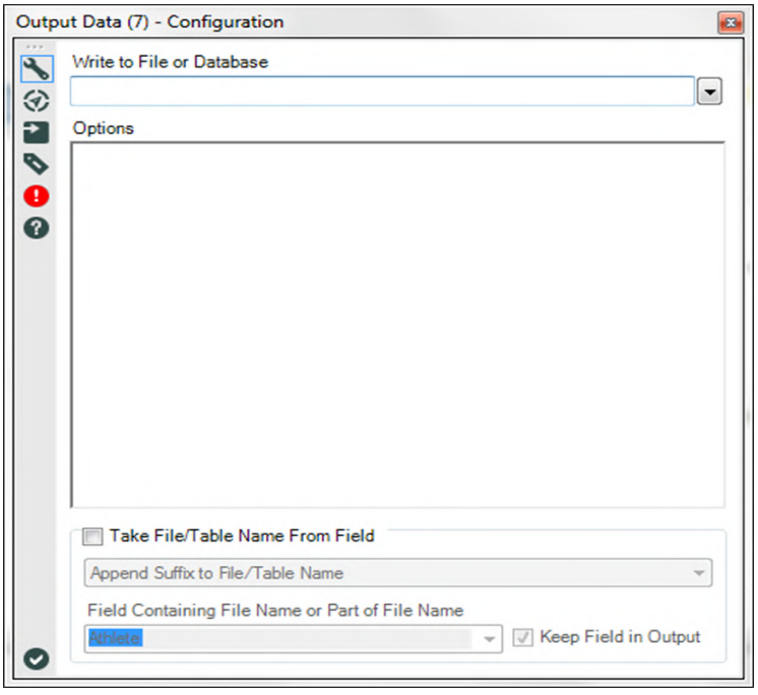


Figure 2-28 – Output Data Properties

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

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

2.11 Running Total

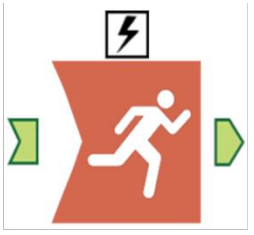


Figure 2-29 – Running Total

The *Running Total* tool allows us to create a running sum for a numeric field in the incoming data stream

Group	Input	Output
Transform	See details below	See details below

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

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

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

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

Properties Window:

The *Running Total Properties* window has two components.

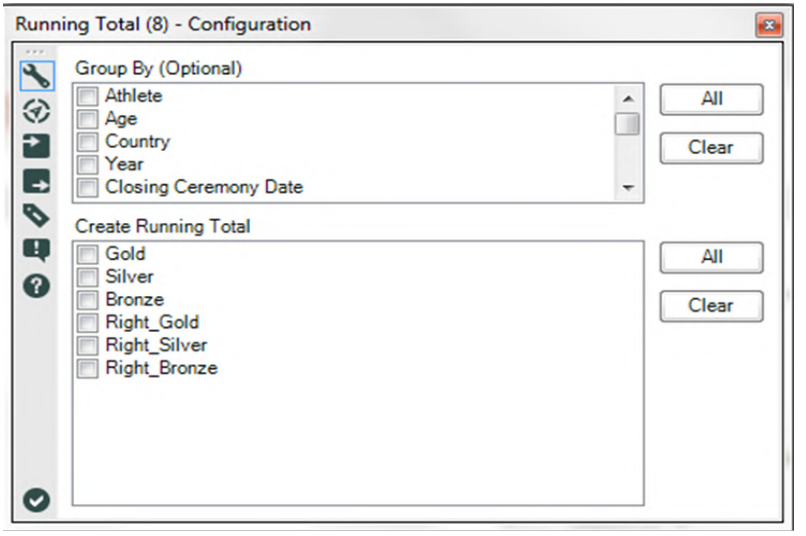


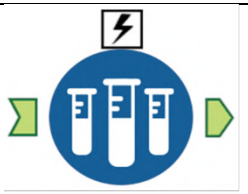
Figure 2-30 – Running Total Properties

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

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

2.12 Sample

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

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

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

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

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

Properties Window:

The *Sample Properties* window has three different settings.

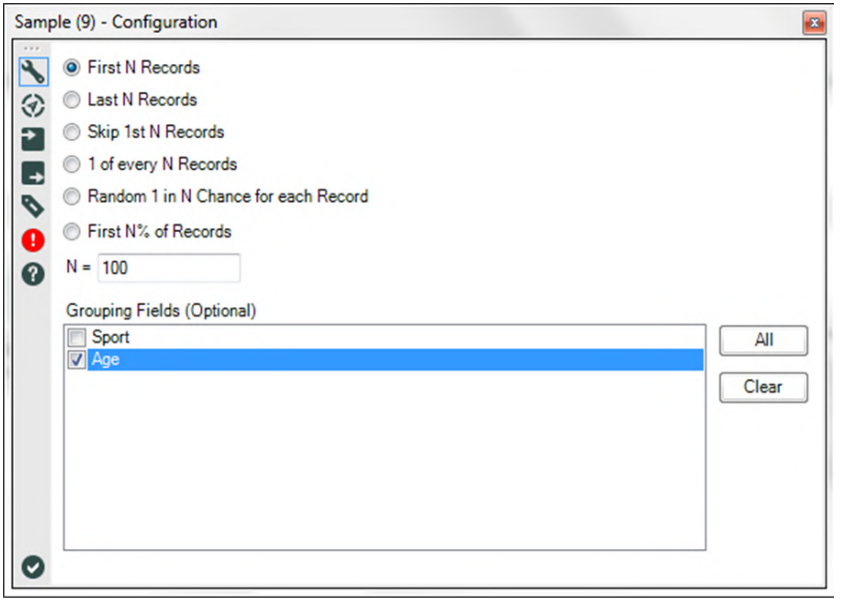


Figure 2-32 – Running Total Properties

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

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

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

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

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

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

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

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

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

2.13 Select

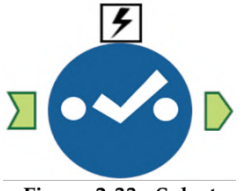


Figure 2-33 - Select

The *Select* tool allows us to modify metadata associated with the data stream, including the order of columns

Group	Input	Output
Preparation	Any data stream	See details below

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

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

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

Properties Window:

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

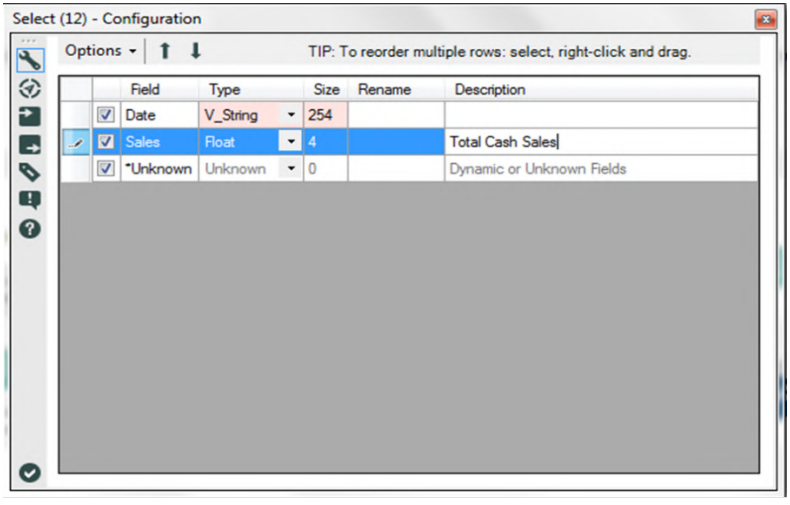


Figure 2-34 – Select Configuration

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

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

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

Select Options Menu:

The options menu allows us to systematically modify the fields.

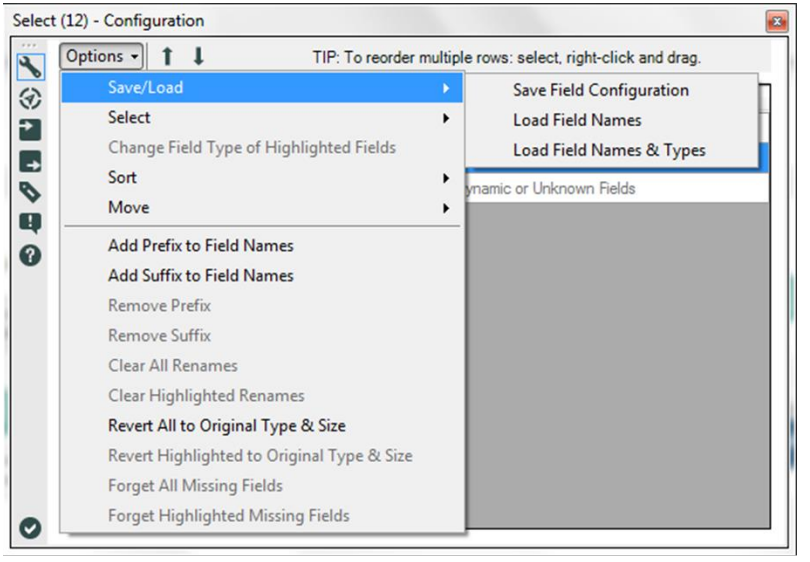


Figure 2-35 – Select Options 1

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

Save Field Configuration creates a new yxft file.

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

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

Other options are as shown in the figure below.

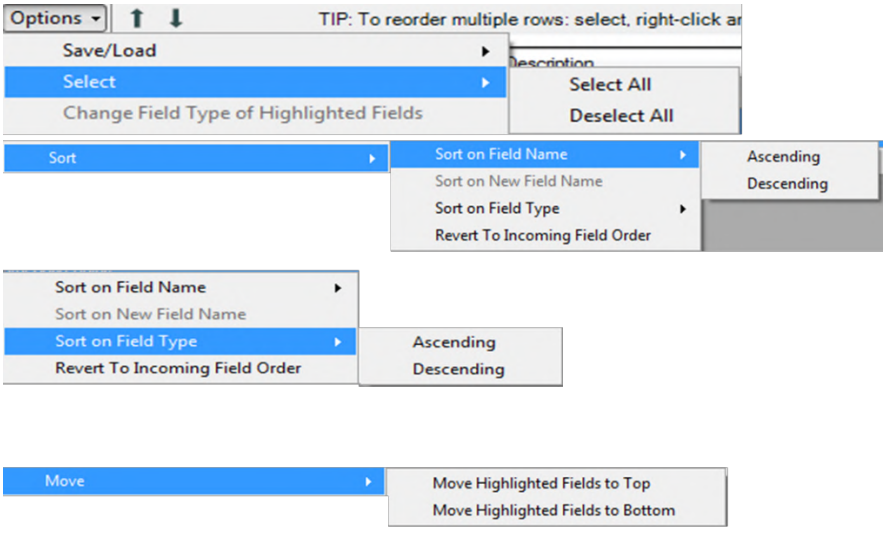


Figure 2-36 – Select Options 2

Select allows us to select or deselect all fields.

Sort has four primary methods of ordering fields.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Type: Type is an important thing to know because each have different attributes and mean different things to other tools. Each of the types are described in Appendix D.

2.14 Sort

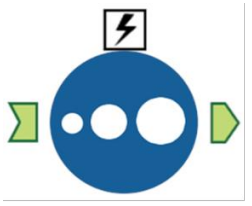


Figure 2-37 - Sort

The *Sort* tool allows us to reorder records.

Group	Input	Output
Preparation	Any data stream	See below

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

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

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

Properties Window:

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

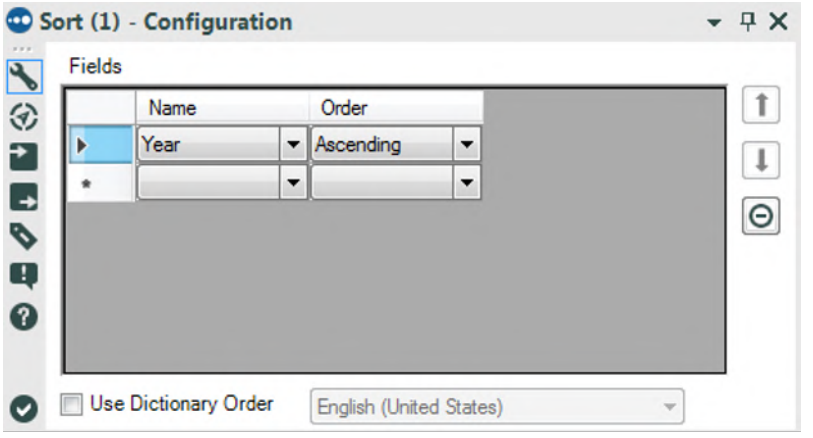


Figure 2-38 – Sort Configuration

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

2.15 Summarize

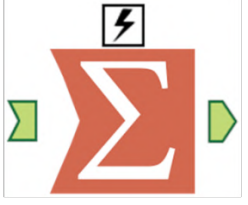


Figure 2-39 - Summarize

The *Summarize* tool allows us to summarize data in our data stream.

Group	Input	Output
Transform	See below	See below

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

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

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

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

Properties Window:

The *Summarize Configuration* window has two basic components.

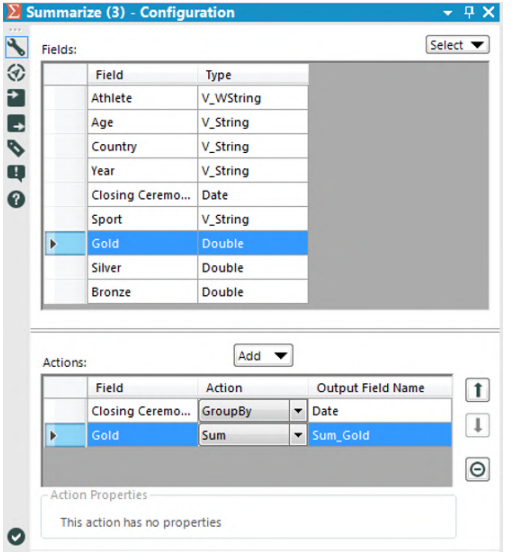


Figure 2-40 – Summarize Configuration

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

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

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

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

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

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

2.16 Tool Container

Figure 2-41 – Tool Container

The *Tool Container* tool allows us to group tools together for clarity and allows the tools to be disabled when unnecessary.

Group	Input	Output
Documentation	None	None

Note: Click and drag tools onto the box to put them into the tool container. Tool containers make it much easier to navigate our data stream because they allow us to consider a series of tools as a single unit. If we click on the arrow at the top-right corner, it will collapse the box without disabling it.

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

Properties Window:

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

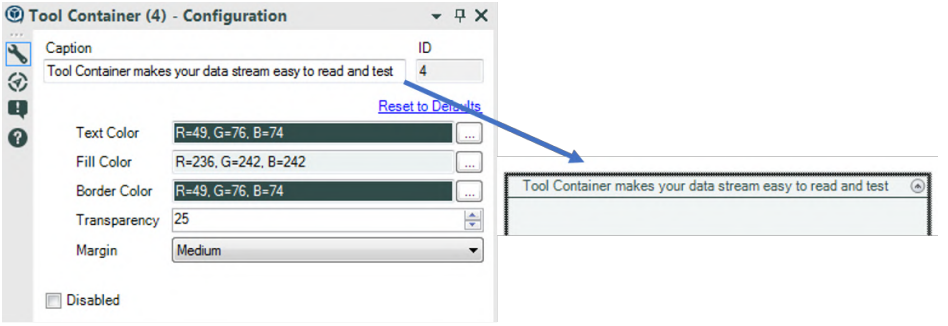


Figure 2-42 – Tool Container Configuration

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

2.17 Transpose

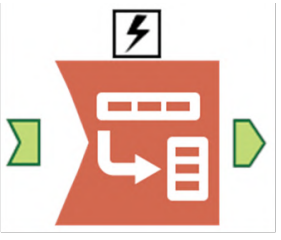


Figure 2-43– Transpose

The *Transpose* tool allows us to de-normalize data

Group	Input	Output
Transform	See below	See below

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

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

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

Output: A taller data stream because records have been duplicated in order to consolidate the columns. The columns consolidate into two columns; *Name*, which is a column with each of the former column names in it, and *Value*, which is a column with each of the data values.

Properties Window:

The *Transpose Configuration* window has three elements, as shown in the figure below.

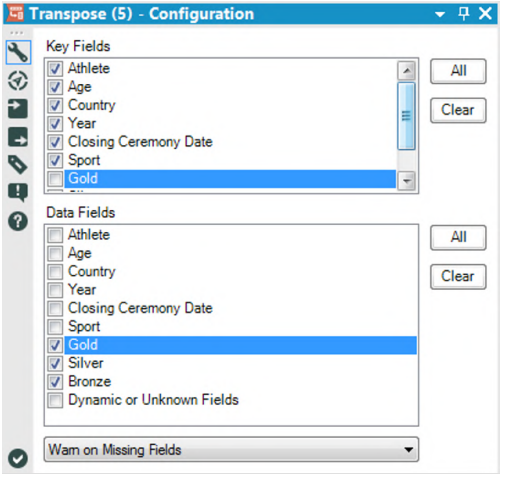


Figure 2-44 – Transpose Configuration

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

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

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

2.18 Union

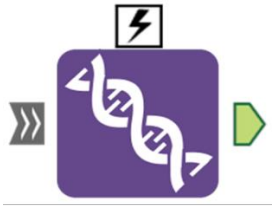


Figure 2-45 - Union

The *Union* tool allows us to append records together one after another from multiple data sources.

Group	Input	Output
Join	See below	See below

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

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

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

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

Properties Window:

The *Union Configuration* window has three core elements.

The first drop-down allows us to change the method that is used to align the columns from the different inputs.

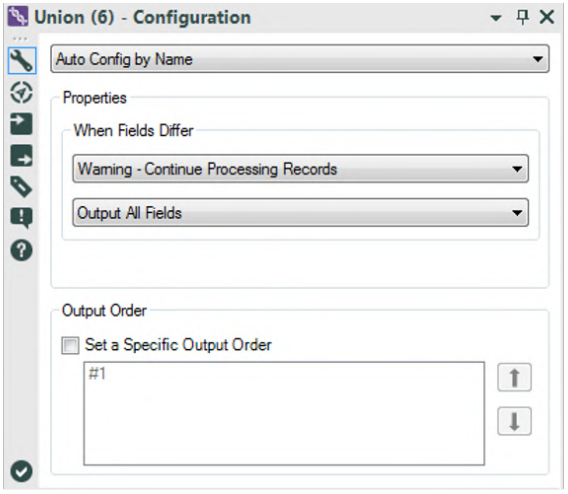


Figure 2-46 – Union Configuration by Name

Auto Config by Name makes the union align the fields that have the same name. This is best used if we know that our data will always be named the same way.

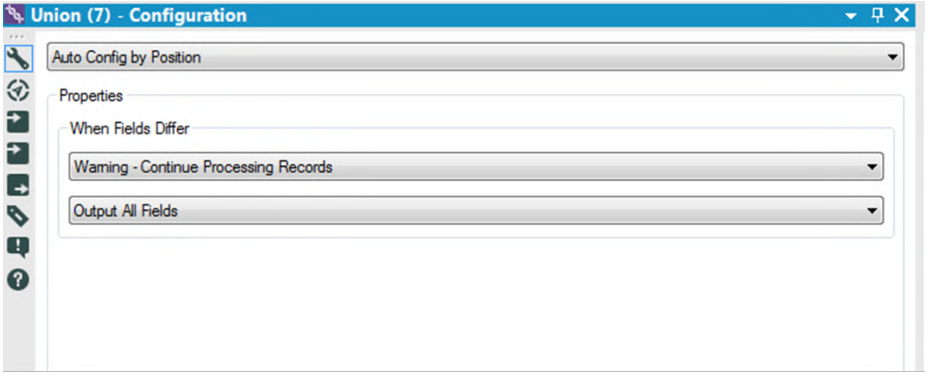


Figure 2-47 –Union Configuration by Position

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

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

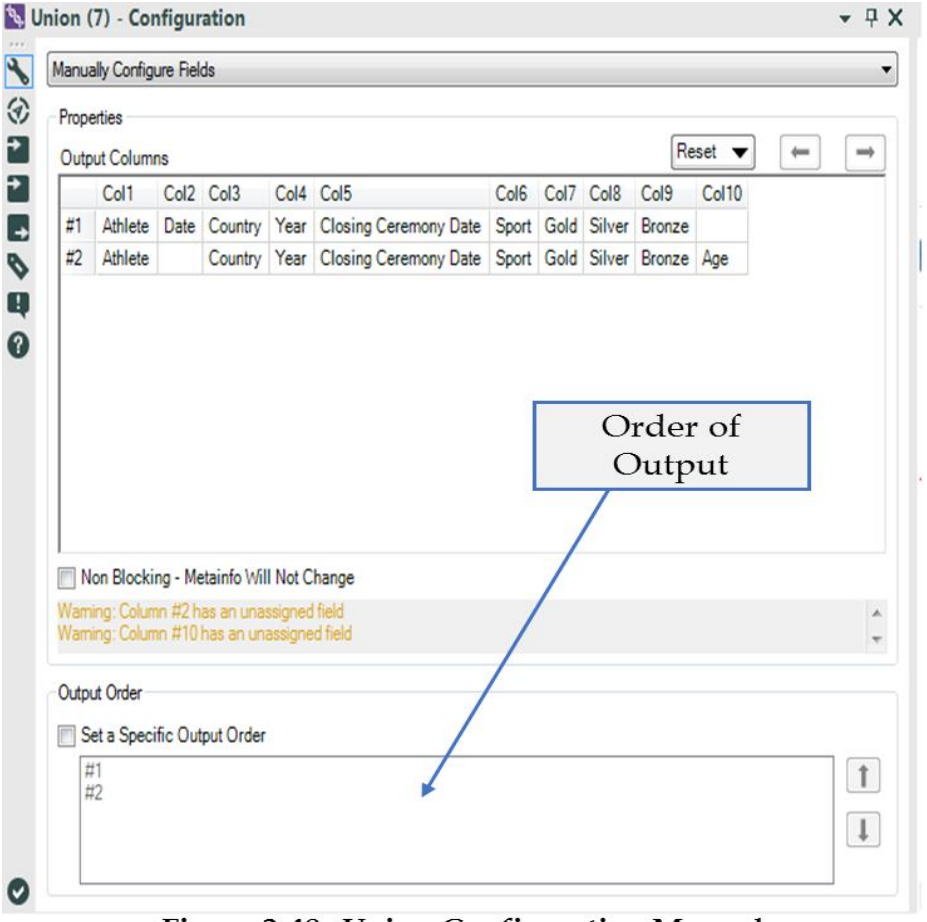


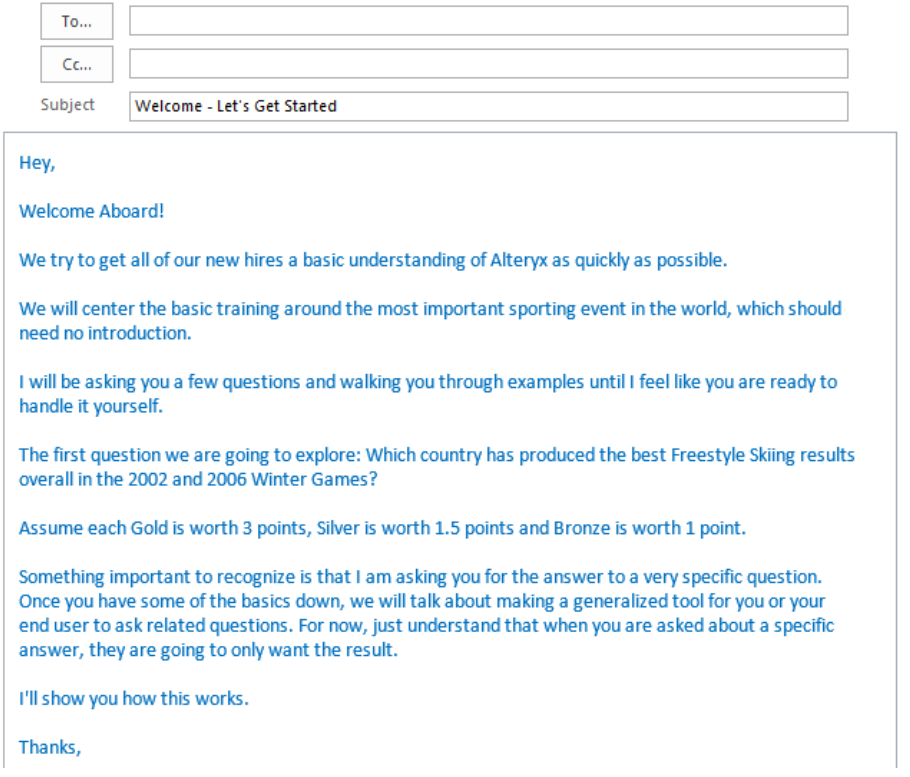
Figure 2-48 –Union Configuration Manual Configuration

Manually Configure Fields is the third option in the first drop-down. It allows us to select exactly which fields to be

brought together by manipulating the Output Columns portion of the configuration window. This is best used if we know that our fields may be named differently or may be in different order.

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

2.19 Freestyle



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

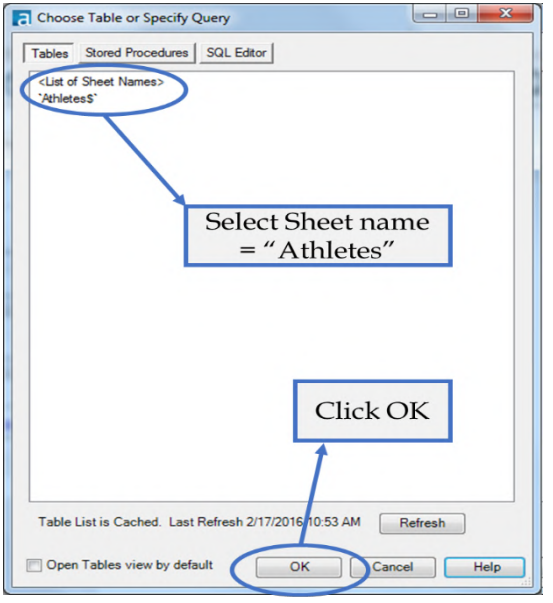


Figure 2-49 – Freestyle Skiing, Data Input

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

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

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

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

- Select allows us to make sure that the fields are in the right format from the beginning.

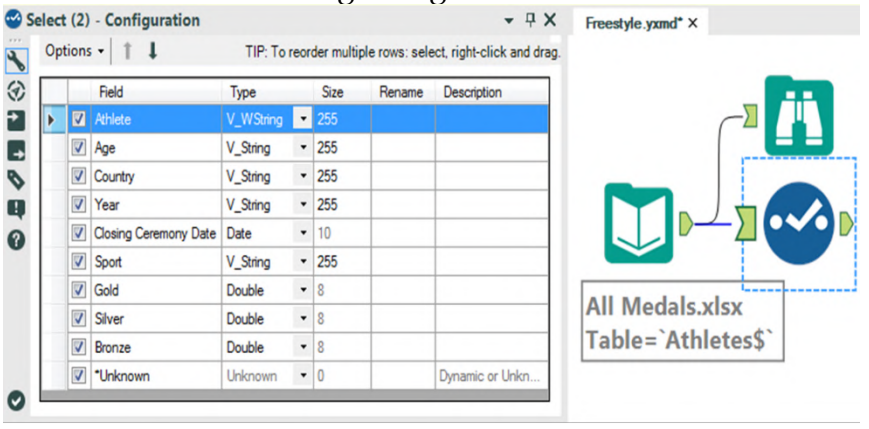


Figure 2-50 – Freestyle Skiing, Select configuration

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

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

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

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

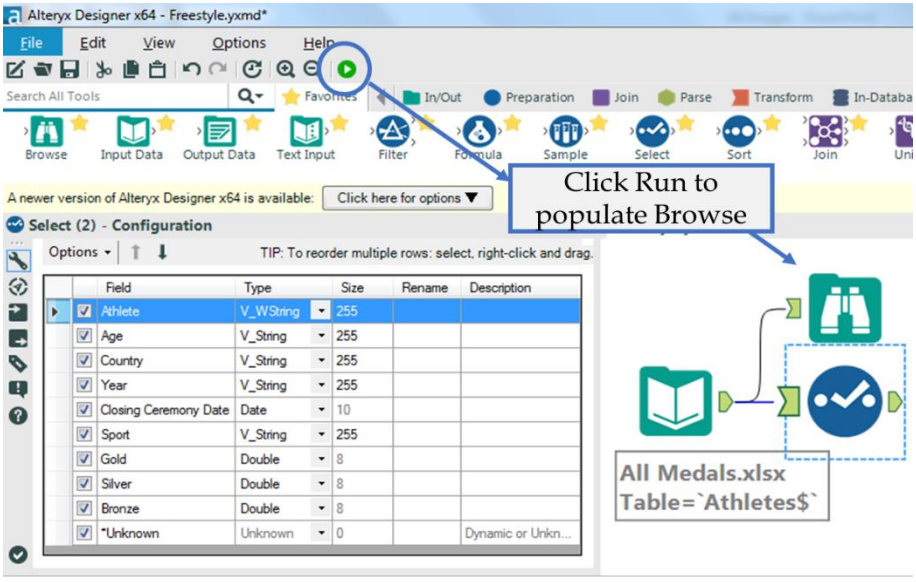


Figure 2-51 – Freestyle Skiing, populate Browse

We can see that when the module finishes running, we get a pop-up window that lets us know how long it took to run and if there were errors.

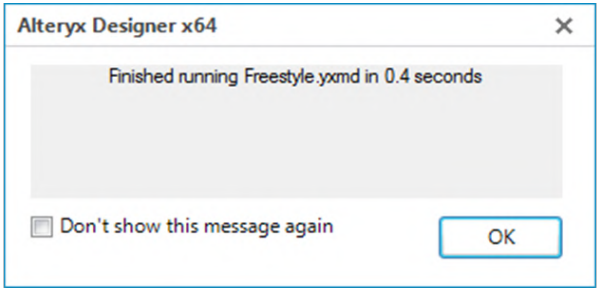
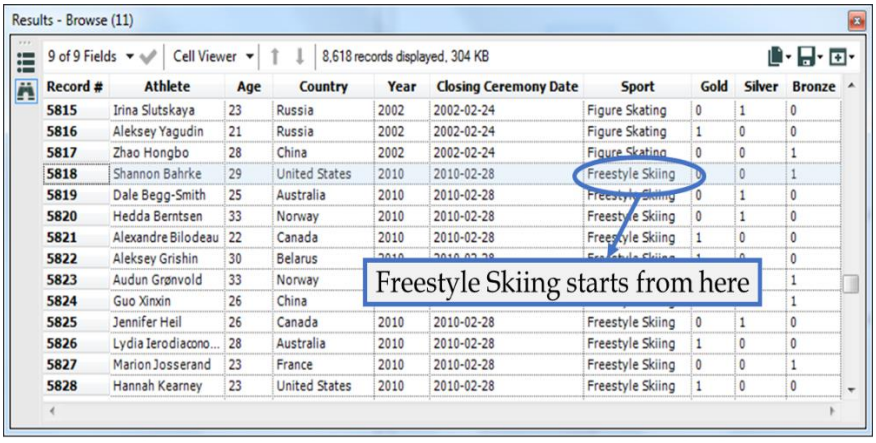


Figure 2-52 – Freestyle Skiing, message after Running

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

Now, we can start thinking about the filter. We know that we are only interested in freestyle skiing results for the 2002 and

2006 games. So the first thing we are going to filter is the sport to "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						0
5823	Audun Grønvold	33	Norway						1
5824	Guo Xinxin	26	China						1
5825	Jennifer Heil	26	Canada	2010	2010-02-28	Freestyle Skiing	0	1	0
5826	Lydia Ierodiaconou...	28	Australia	2010	2010-02-28	Freestyle Skiing	1	0	0
5827	Marion Josserand	23	France	2010	2010-02-28	Freestyle Skiing	0	0	1
5828	Hannah Kearney	23	United States	2010	2010-02-28	Freestyle Skiing	1	0	0

Figure 2-53 – Freestyle Skiing, Browse configurations

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

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

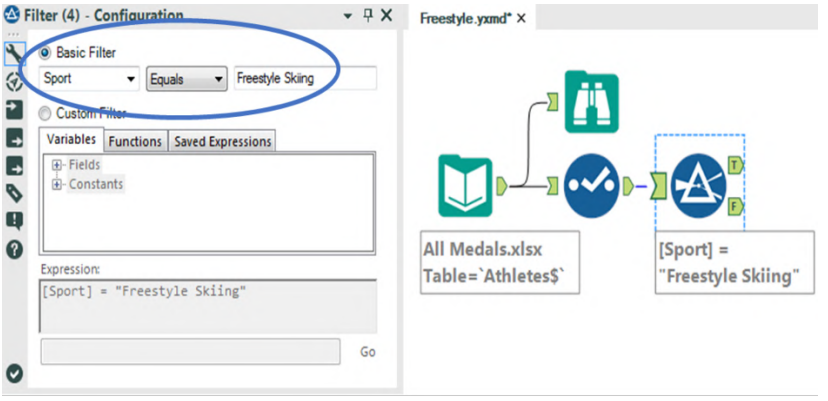


Figure 54 – Freestyle Skiing, Browse configurations

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

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

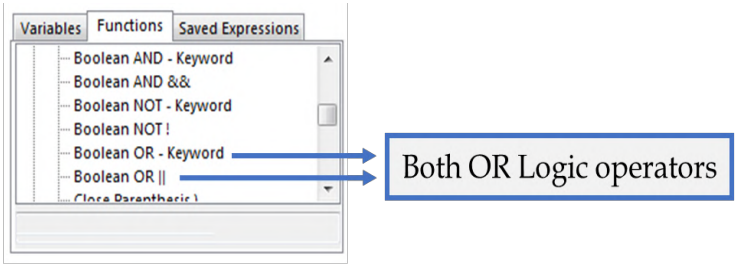


Figure 2-55 – Freestyle Skiing, Filter configurations - Functions

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

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

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

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

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

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

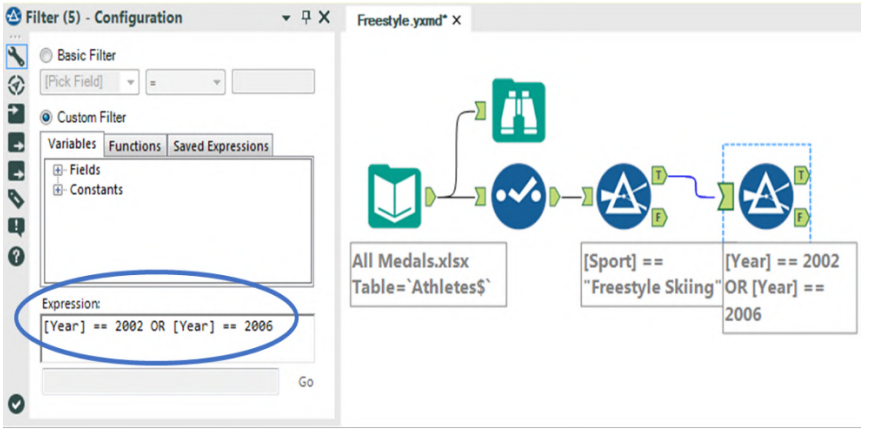


Figure 2-56 – Freestyle Skiing, Custom filter

See Appendix F for examples of Boolean logic.

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

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

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

We now have removed all of the information we don't need in order to answer the question. But we have the problem that the data is too granular. We know who the athlete was and in which year they won their medal(s). We should bring in a Summarize Tool in order to bring the data up to the country level. If we add Country using Group By, and Gold, Silver, and Bronze using Sum, we will get a list of countries and their total medal

count for Freestyle skiing for 2002 and 2006. Place a Browse Tool, and run the module to see what we have so far.

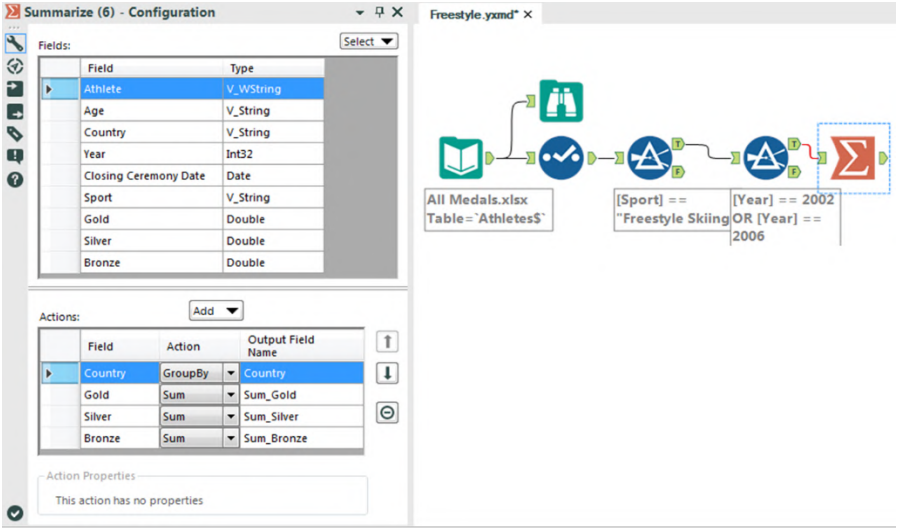


Figure 2-57 – Freestyle Skiing, Summary configuration

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

Results - Browse (12)

4 of 4 Fields | Cell Viewer

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 2-58 – Freestyle Skiing, Browse configuration after summarize

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

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

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

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

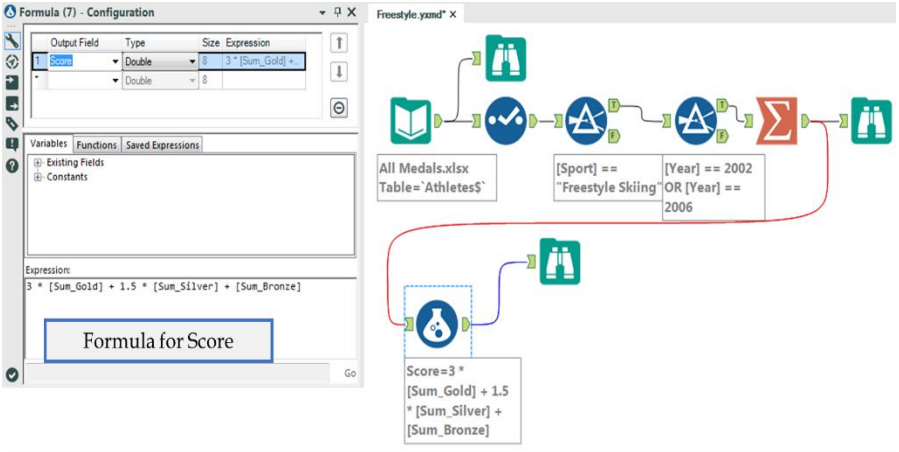


Figure 2-59 – Freestyle Skiing, Formula configuration

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

Results - Browse (13)

5 of 5 Fields | Cell Viewer

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 2-60 – Freestyle Skiing, Browse configuration after formula

We see there is a new field called Score that is an unordered dataset and with multiple unnecessary values. We can also see that Australia has the highest score and therefore is the answer to the original question. But for good practice, we are

going to continue to build this workflow so that no interpretation is needed.

This process is going to take four steps:

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

Like we discussed, the Sort Tool is how we reorder the data and will be our first step. We will set up our data in a descending order based on Score, like we see here.

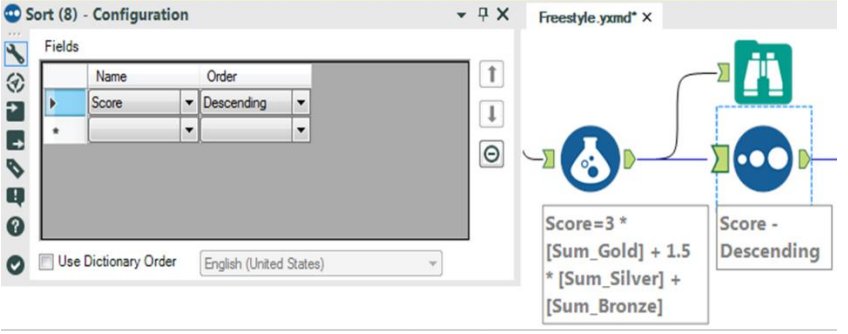


Figure 2-61 – Freestyle Skiing, Sort configuration

Next, we need only the first record, so we are going to use the Sample Tool to keep only the Top 1 Record coming out of the Sort.

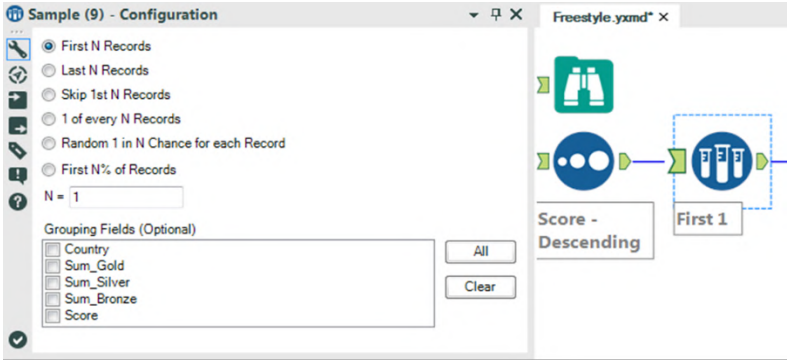


Figure 2-62 – Freestyle Skiing, Sample configuration

We know we have fields we no longer need, so we can use a Select Tool to eliminate everything that is not the country name.

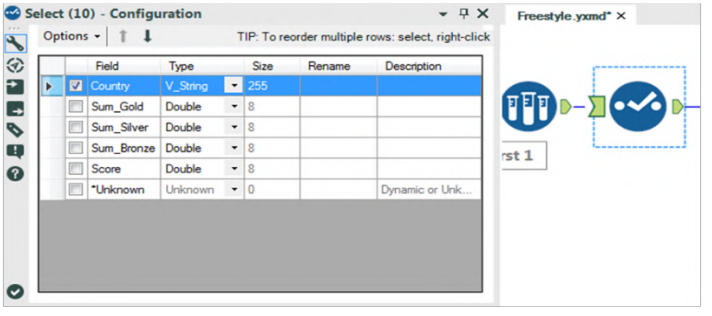


Figure 2-63 – Freestyle Skiing, Selection after Sample

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

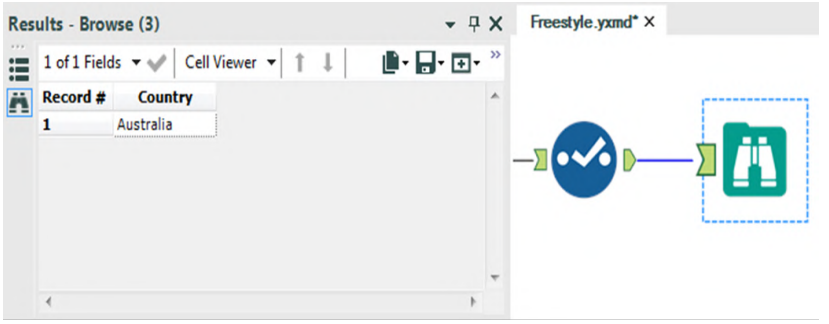


Figure 2-64 – Freestyle Skiing, Browse after Select

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

- We would have a second verification that the answer was what we expected.
- Repeatability of the result so we can point to a single issue in the data preparation process that needs fixing instead of not being able to fix it at all.

Here is how the workflow would look like when complete:

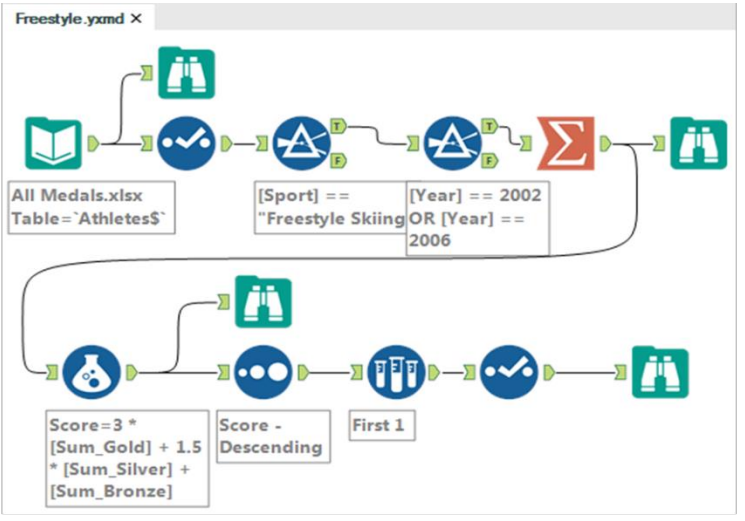
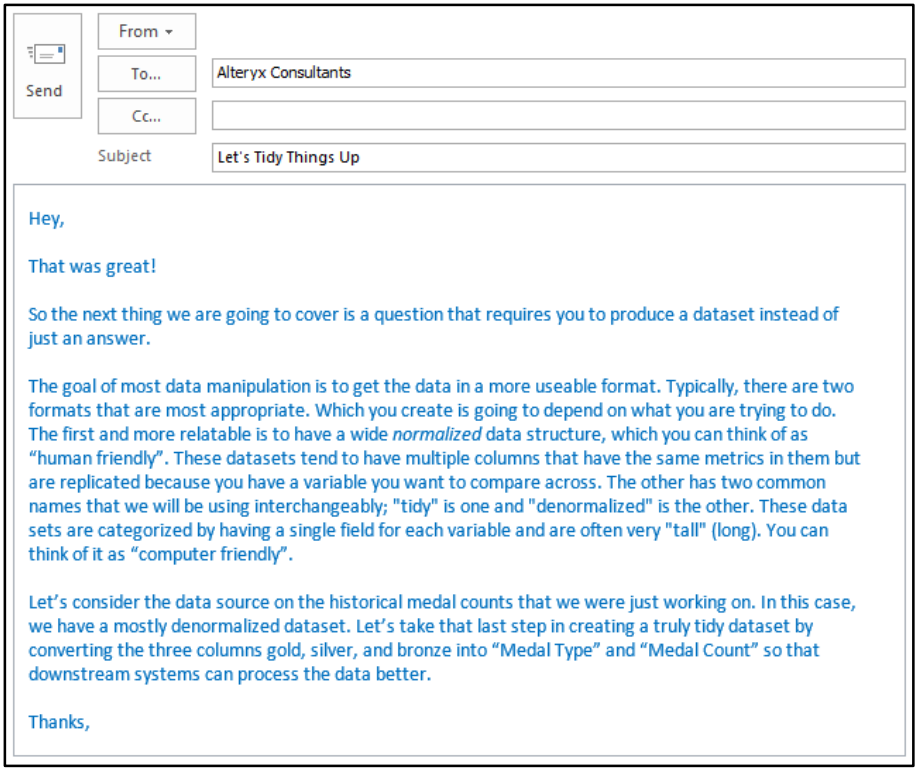


Figure 2-65 – Freestyle Skiing, Data stream after completion

2.20 Let's Tidy Things Up



This process must include at least four steps:

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

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

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

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

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

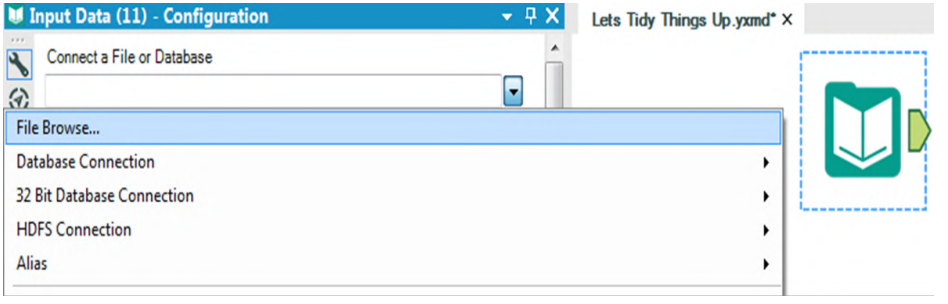


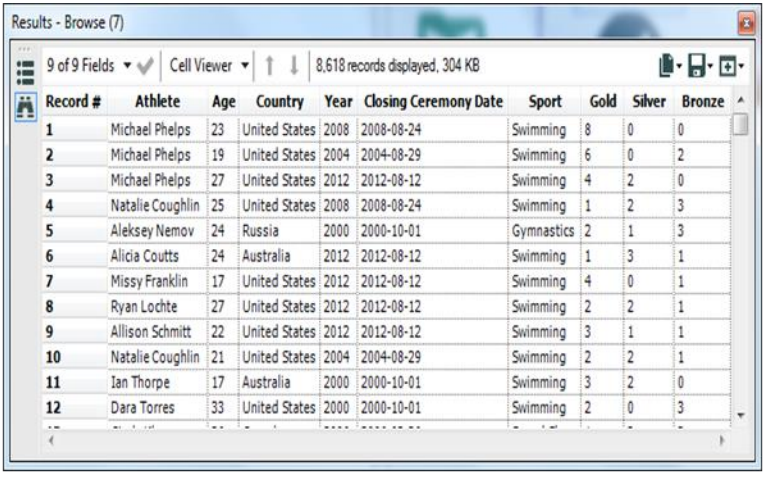
Figure 2-66 – Let's Tidy Things Up, Data Input

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

Figure 2-67 – Medals data

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

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



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
10	Natalie Coughlin	21	United States	2004	2004-08-29	Swimming	2	2	1
11	Ian Thorpe	17	Australia	2000	2000-10-01	Swimming	3	2	0
12	Dara Torres	33	United States	2000	2000-10-01	Swimming	2	0	3

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

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

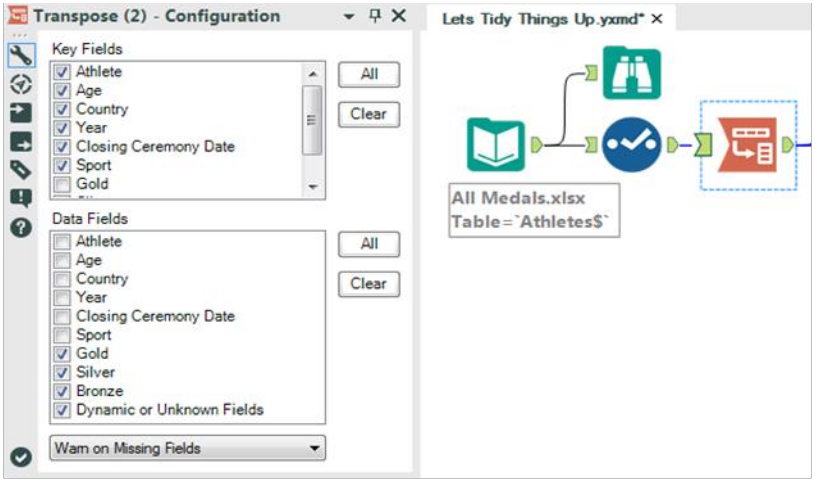
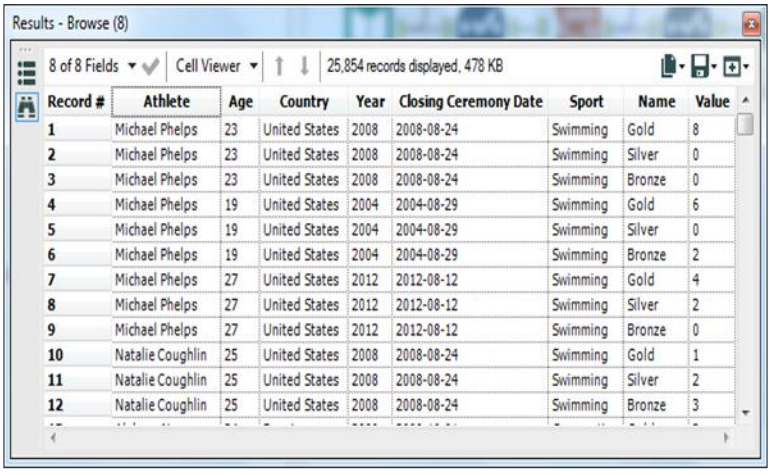


Figure 2-69 – Let's Tidy Things Up, Transpose configuration

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

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



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
10	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Gold	1
11	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Silver	2
12	Natalie Coughlin	25	United States	2008	2008-08-24	Swimming	Bronze	3

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

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

This is one of the reasons that tidy data is not particularly human readable but is highly computer readable. Since all of the information is displayed in each record and there is only a single column to work on, interactive front-end software can work very fast with the data.

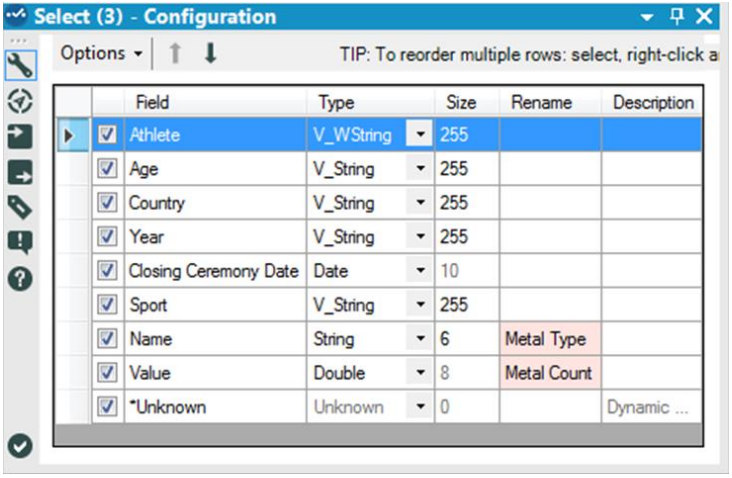


Figure 2-71 – Let's Tidy Things Up, Select configuration

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

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

Our goal is to filter out any records that have zero medals. We are filtering on a numeric field for the first time, which means we should use the Basic Filter to learn about the syntax. The configuration is as shown in the following figure.

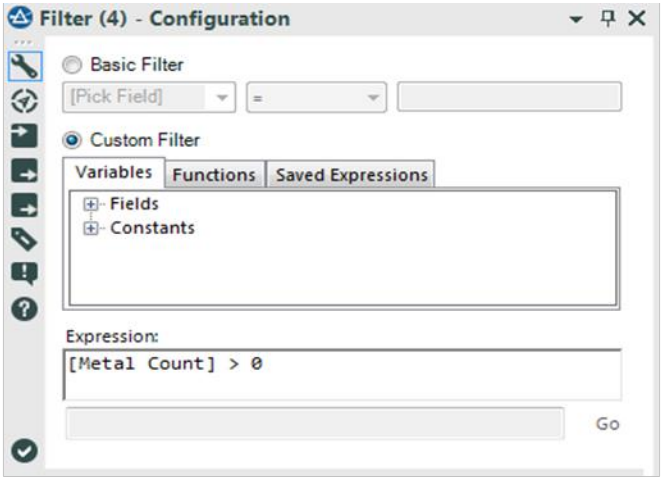


Figure 2-72 – Let's Tidy Things Up, Filter configuration

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

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

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

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

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

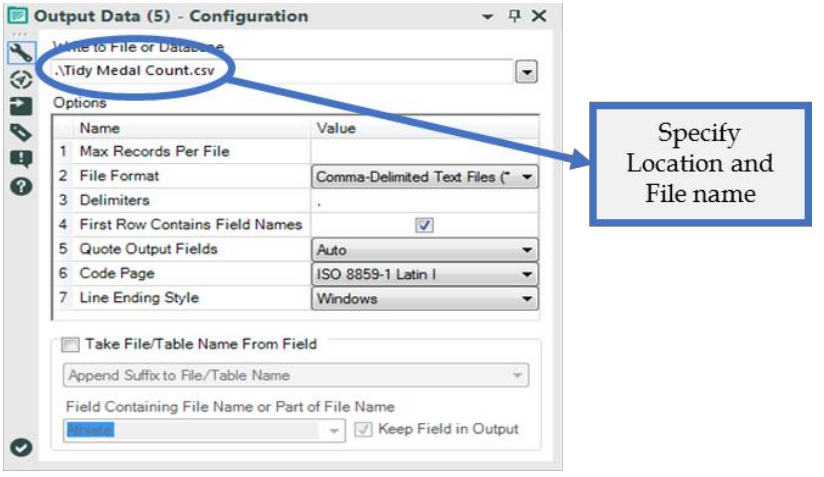


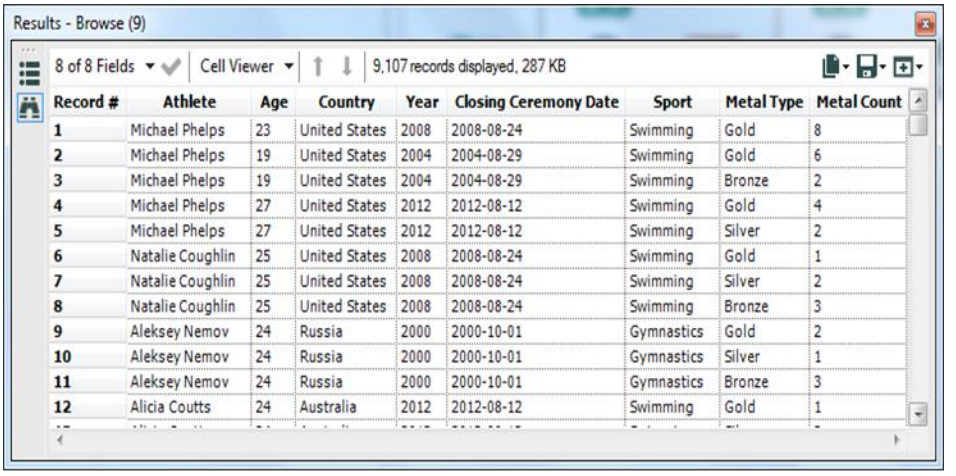
Figure 2-73 – Let's Tidy Things Up, Output Data configuration

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

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

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

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



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
10	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	Silver	1
11	Aleksey Nemov	24	Russia	2000	2000-10-01	Gymnastics	Bronze	3
12	Alicia Coutts	24	Australia	2012	2012-08-12	Swimming	Gold	1

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

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

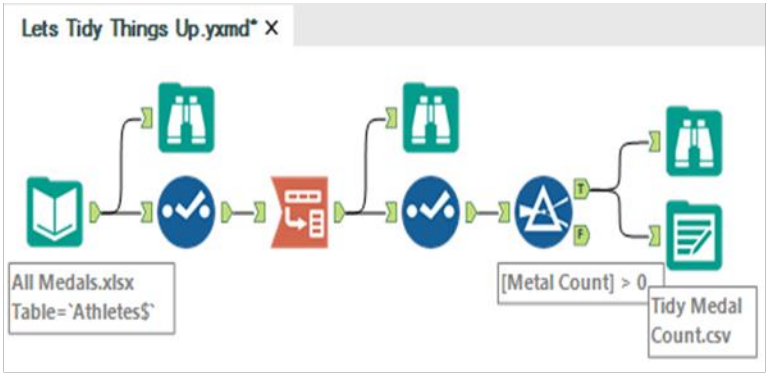
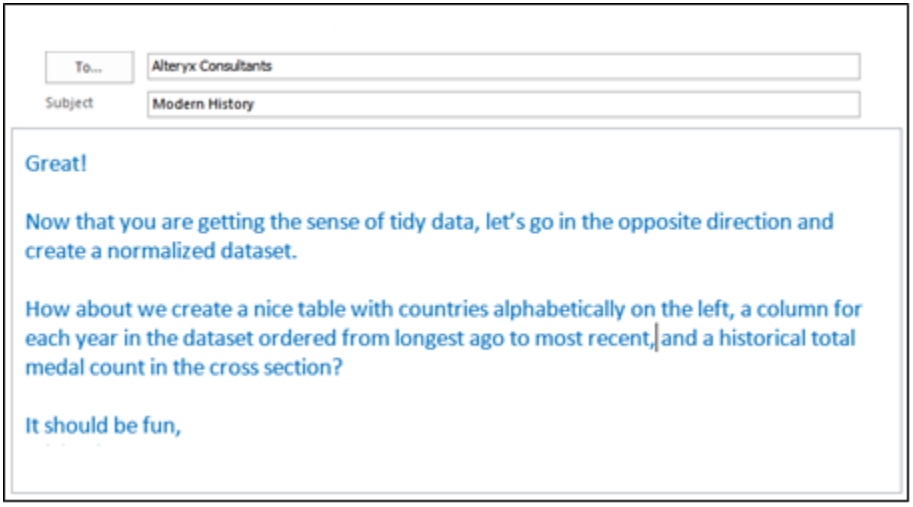


Figure 2-75 – Let's Tidy Things Up, Data stream after completion

2.21 Modern History



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

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

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

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

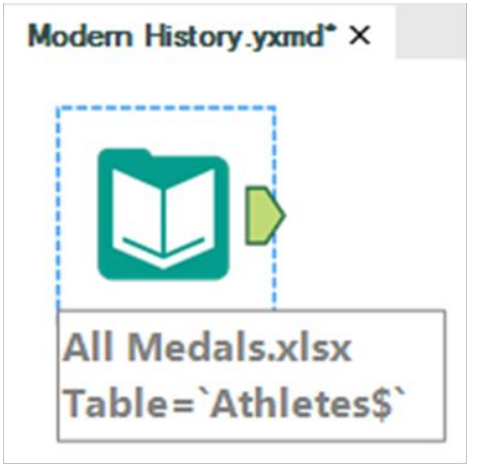


Figure 2-76 Modern history, Input Data

Best practice will once again bring in Browse and Select tools. But since we know from past experience what the data looks like and how it is read in, we will move directly into the next step.

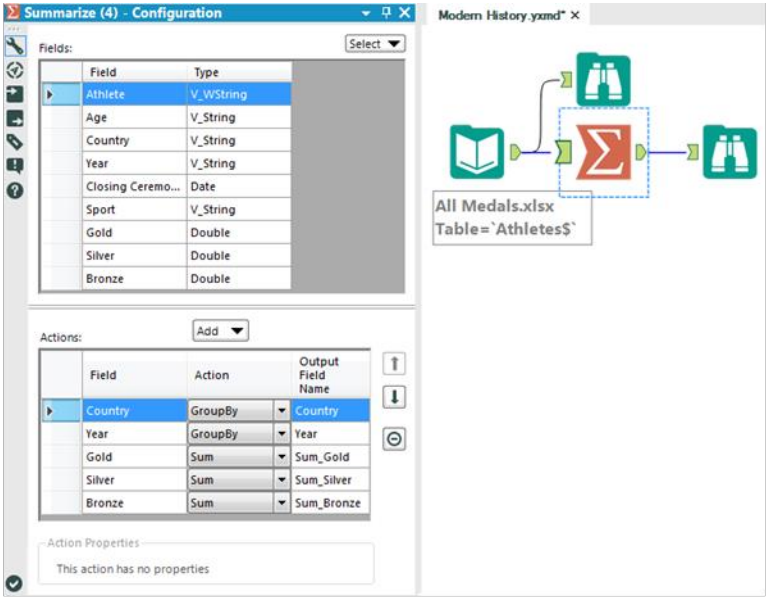


Figure 2-77 – Modern history, Summarize configuration

We know that this data is too granular for our desired result. So we can summarize it. Based on the email, we know the only information we will need in the end is the country, year, and something to do with the medals. So when using the Summary tool, we can group by the country and year fields and take the sum of each of the medal counts to take our first step down this path.

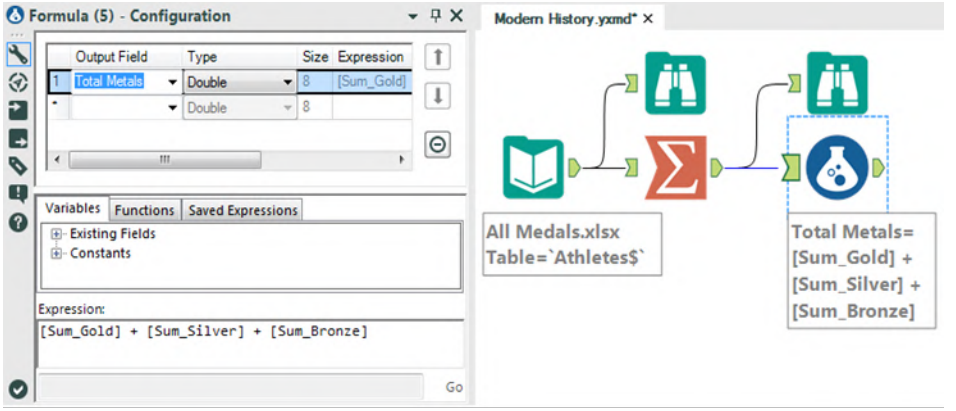


Figure 2-78– Modern history, Formula configuration

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

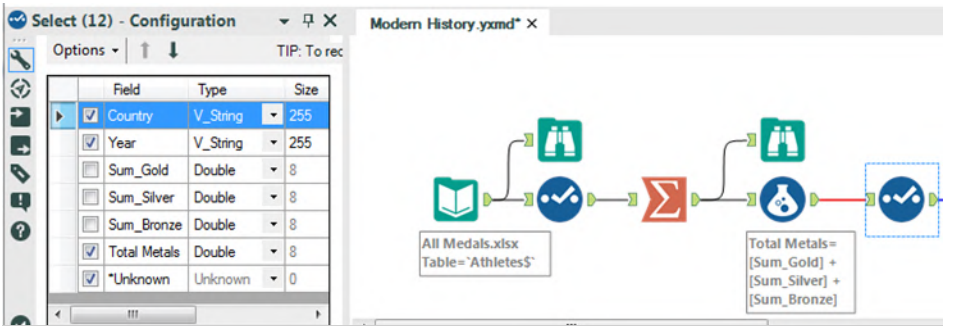


Figure 2-79 – Modern History Select Configuration

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

We know we need a historical medal count, which means we are going to need to take the running total along with the

country and year. But because Running Total is a tool where order matters, we need to sort the data.

We can sort the Country and Year in ascending order to help us in two places: Initially, this will help because we are creating the order for the Running Total, but it will also help us with the order of records and columns when we normalize the data set.

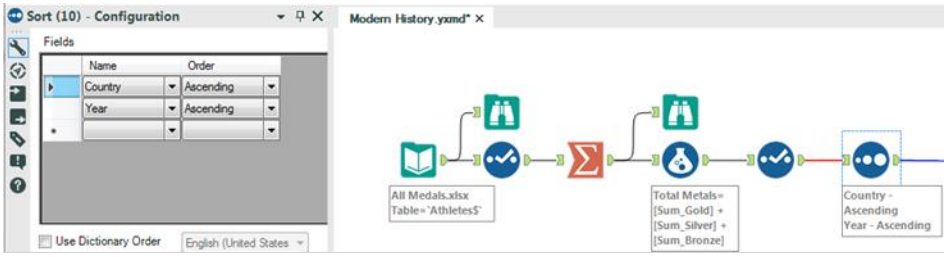


Figure 2-80 – Modern History Sort Configuration

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

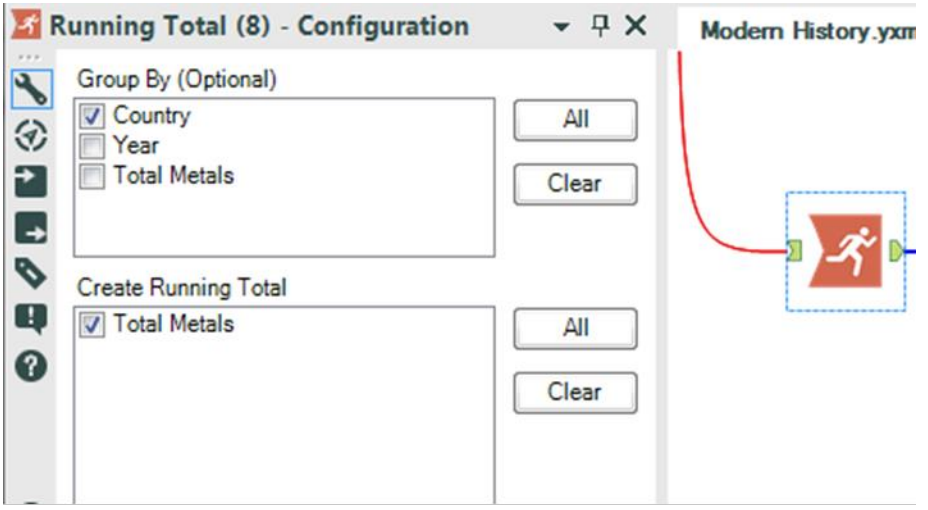


Figure 2-81 – Modern History Running Total Configuration

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

Browse (9) - Configuration — Modern History

Results - Browse (9)

4 of 4 Fields | Cell Viewer | 401 records displayed, 7764 byt

Record #	Country	Year	Total Metals	RunTot_Total Metals
1	Afghanistan	2008	1	1
2	Afghanistan	2012	1	2
3	Algeria	2000	5	5
4	Algeria	2008	2	7
5	Algeria	2012	1	8
6	Argentina	2000	20	20
7	Argentina	2004	49	69
8	Argentina	2008	51	120
9	Argentina	2012	21	141
10	Armenia	2000	1	1
11	Armenia	2008	6	7

Figure 2-82 – Modern History Browse Configuration after Total

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

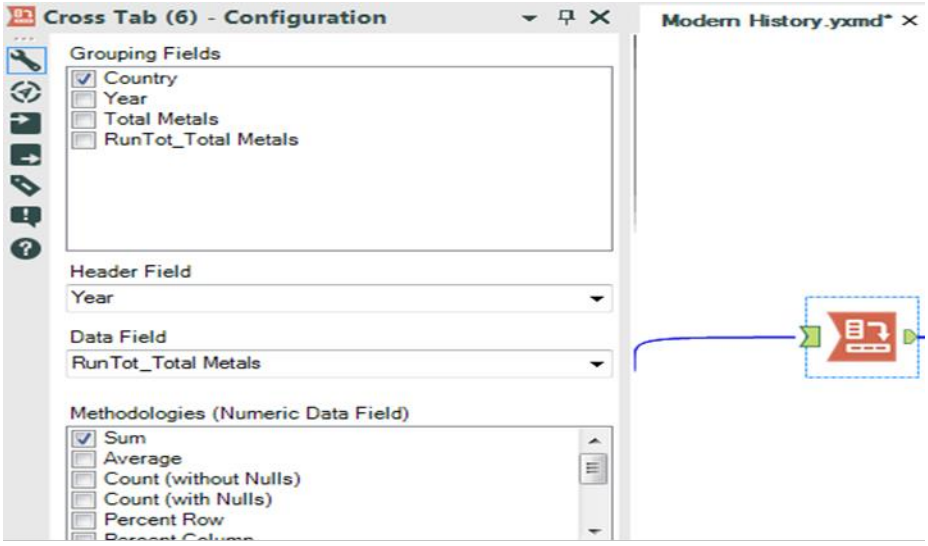


Figure 2-83 – Modern History Cross Tab Configuration

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

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

Browse (7) - Configuration | Modern History.yxmd*

Results - Browse (7)

8 of 8 Fields | Cell Viewer | 110 records displayed, 5181 bytes

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

Figure 2-84 – Modern History Browse Configuration Cross Tab

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

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

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

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

IF b1 THEN x ELSEIF b2 THEN y ELSE z ENDIF

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

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

IsNull(x)

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

The formulae that we need are:

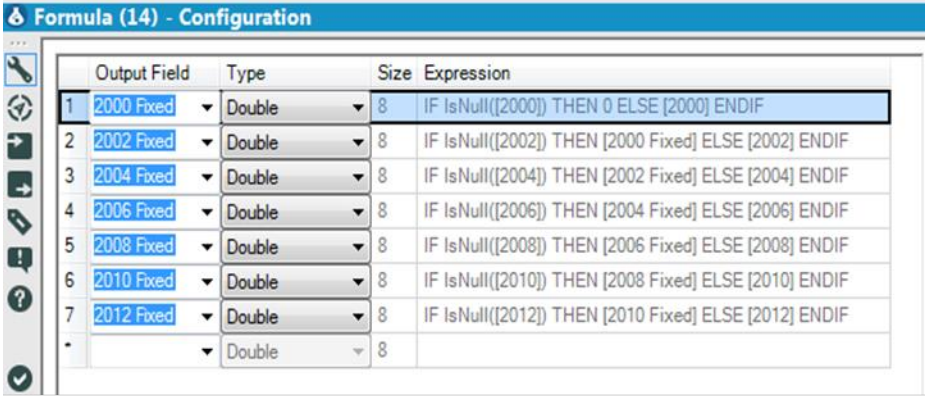
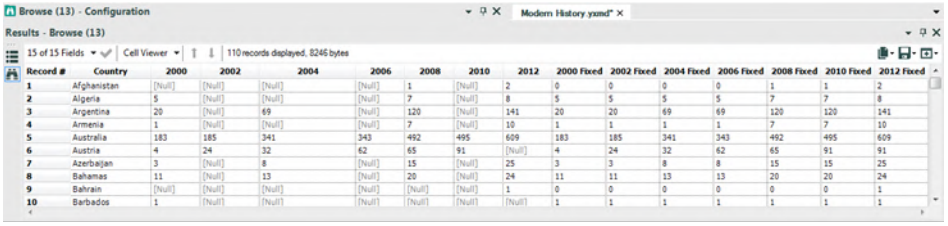


Figure 2-85 – Modern History Formula List

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

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



Record #	Country	2000	2002	2004	2006	2008	2010	2012	2000 Fixed	2002 Fixed	2004 Fixed	2006 Fixed	2008 Fixed	2010 Fixed	2012 Fixed
1	Afghanistan	[Null]	[Null]	[Null]	[Null]	1	[Null]	2	0	0	0	0	1	1	2
2	Algeria	5	[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	11	13	13	20	20	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

Figure 2-86 – Modern History Formula List

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

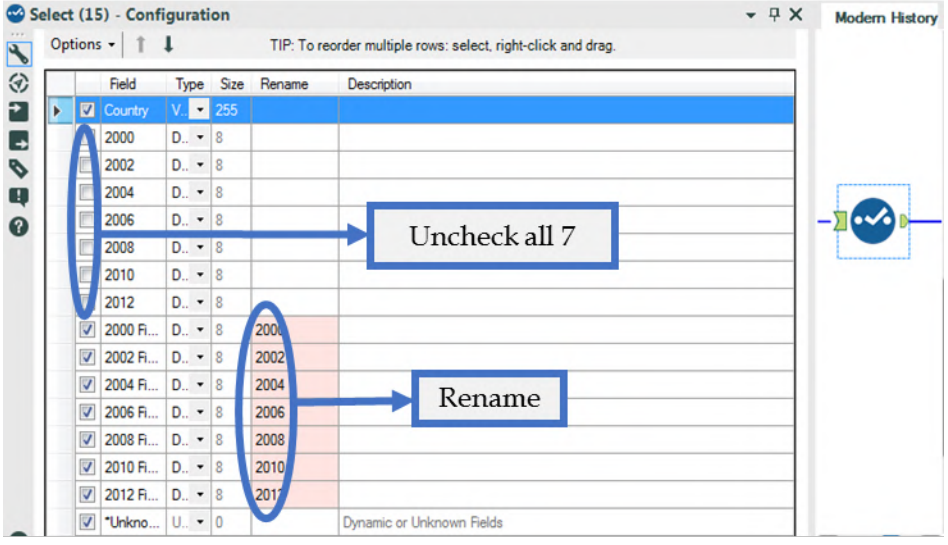


Figure 2-87 – Modern Select Configuration after Formula

Now the data has finished being prepped. We need to write it out, which we know because we were asked for a data set and not a specific answer. We should add a Browse tool and an

Output Data tool to end the data flow. Save the output as Historical Medal Count.csv.

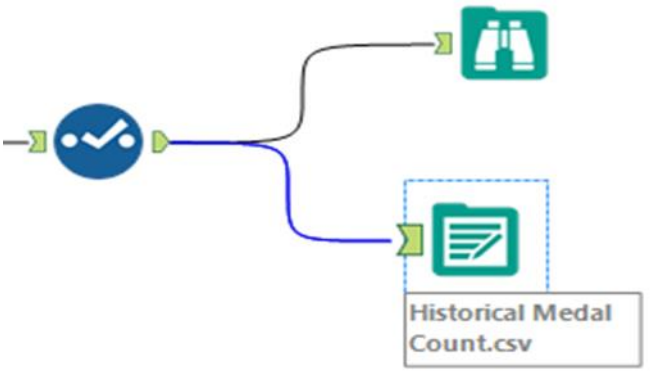


Figure 2-88 – Modern History Output

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

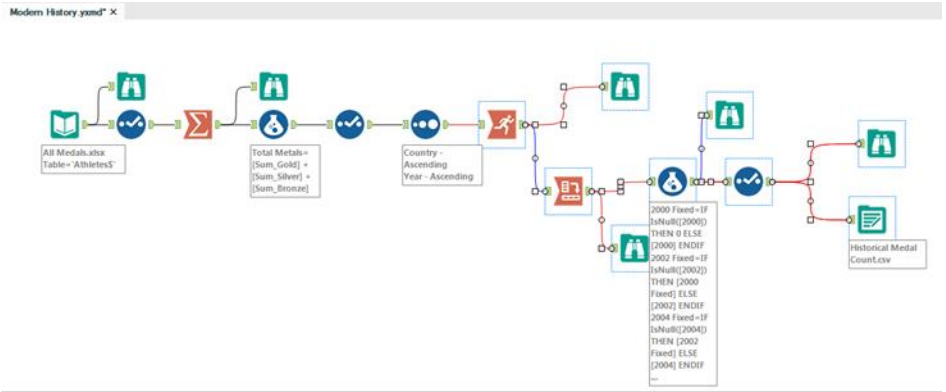
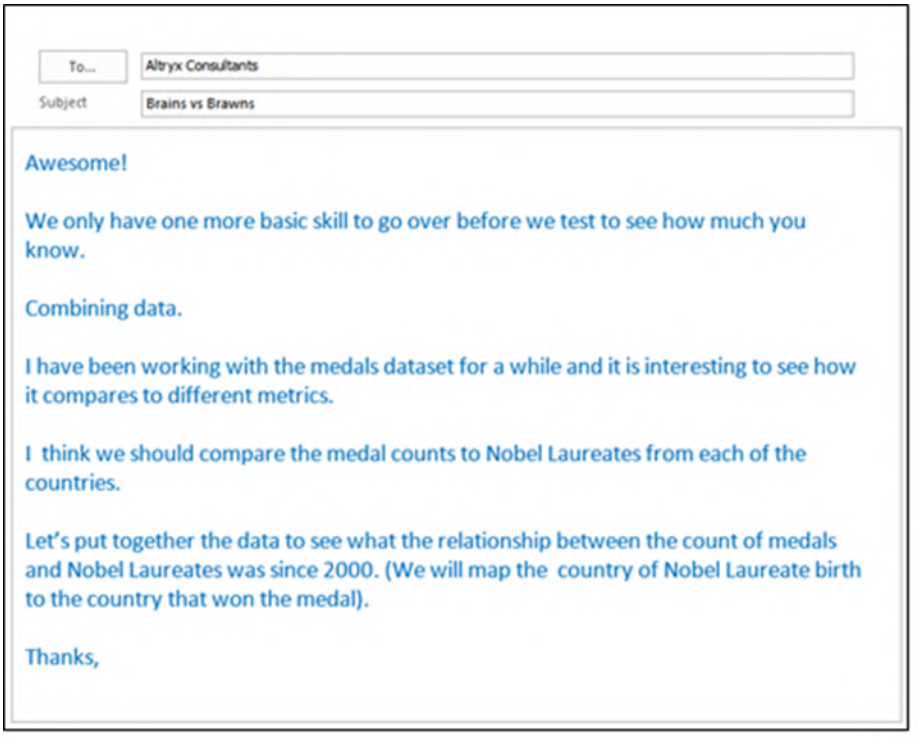


Figure 2-89 – Modern History Data Stream When Complete

2.22 Brains vs Brawns


To... Altryx Consultants

Subject Brains vs Brawns

Awesome!

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

Combining data.

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

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

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

Thanks,

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

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

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

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

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

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

The properties windows for each of these tools as well as the data stream that is produced are shown in the following figures.

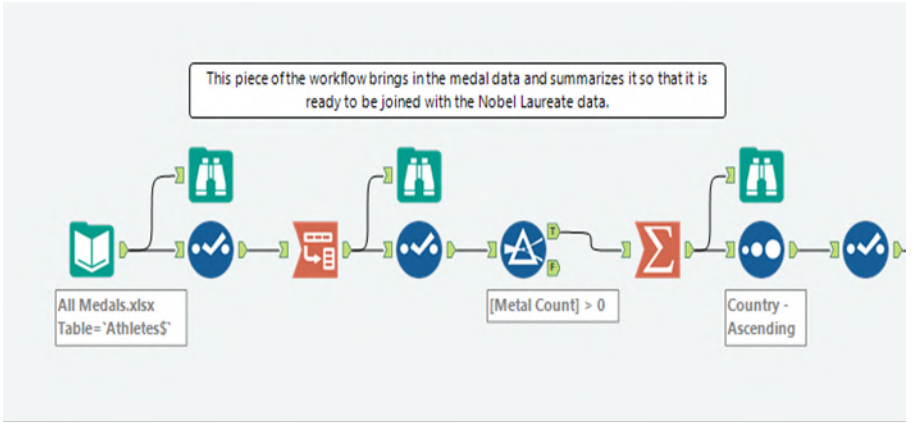
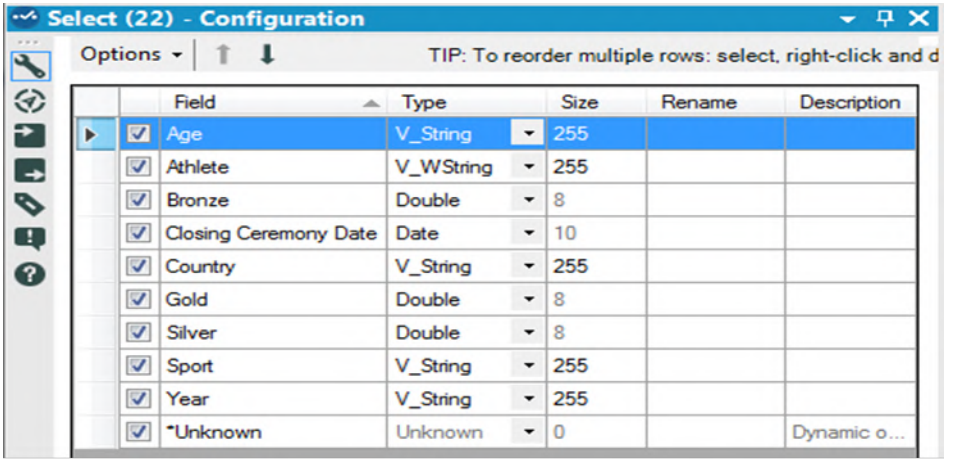


Figure 2-90 – Brains vs Brawns data stream



Field	Type	Size	Rename	Description
Age	V_String	255		
Athlete	V_WString	255		
Bronze	Double	8		
Closing Ceremony Date	Date	10		
Country	V_String	255		
Gold	Double	8		
Silver	Double	8		
Sport	V_String	255		
Year	V_String	255		
*Unknown	Unknown	0		Dynamic o...

Figure 2-91 – Initial Steps - Select

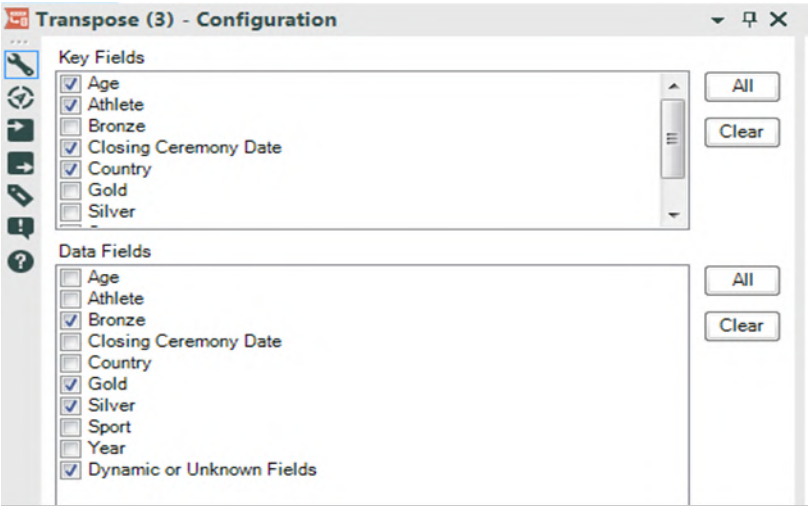


Figure 2-92 – Initial Steps - Transpose

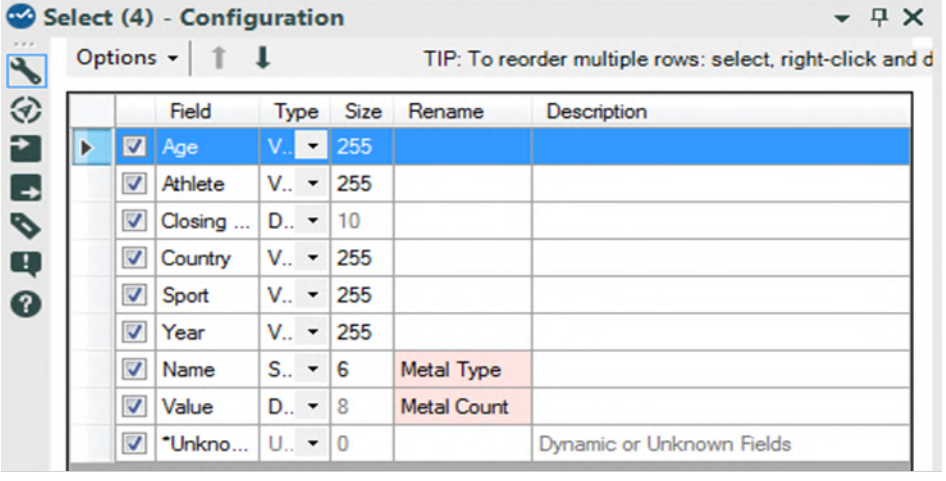


Figure 2-93 – Initial Steps - Select

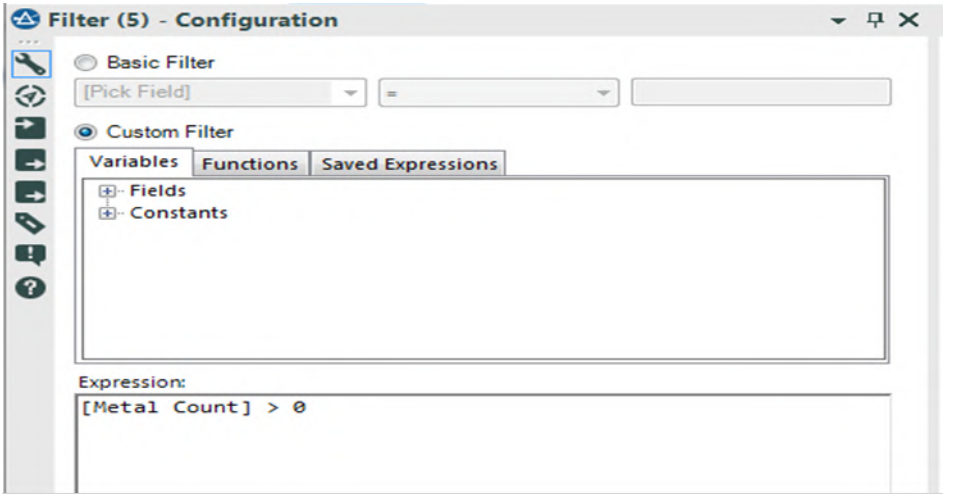


Figure 2-94 – Initial Steps - Filter

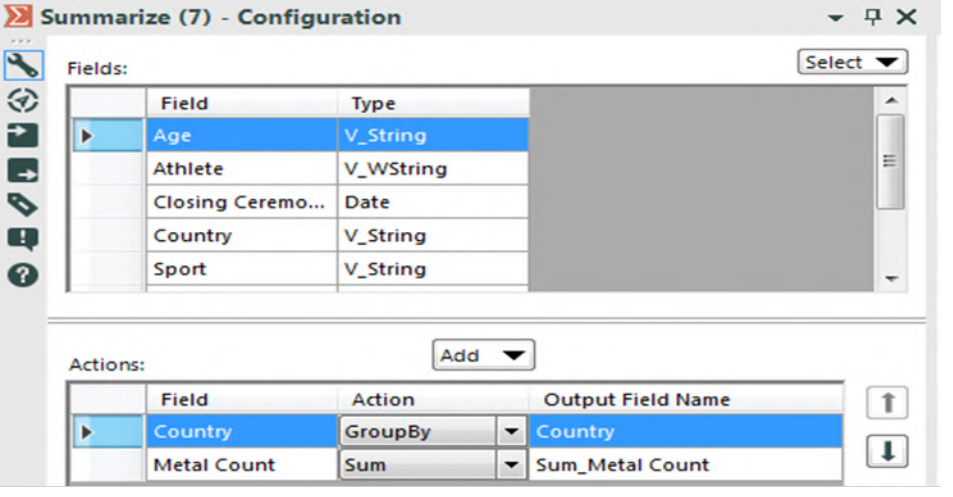


Figure 2-95 – Initial Steps - Summarize

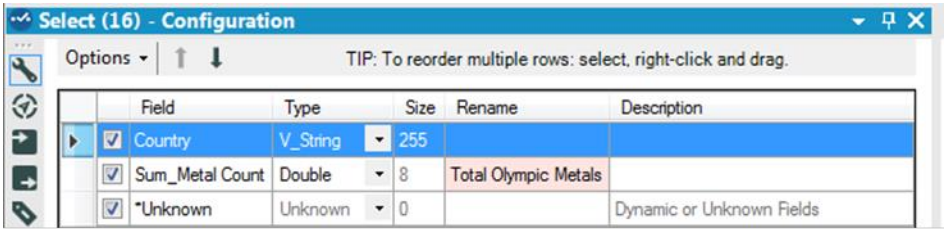


Figure 2-96 – Initial Steps - Sort

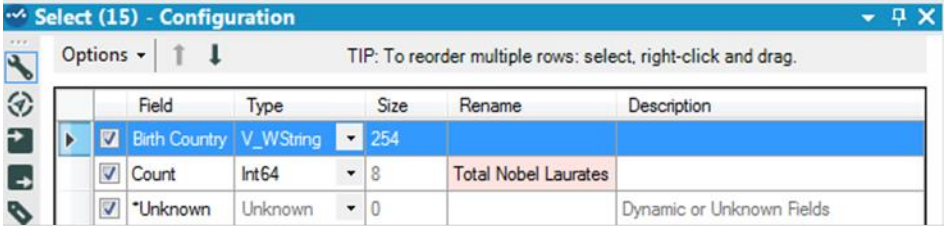


Figure 2-97 – Initial Steps - Select

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

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

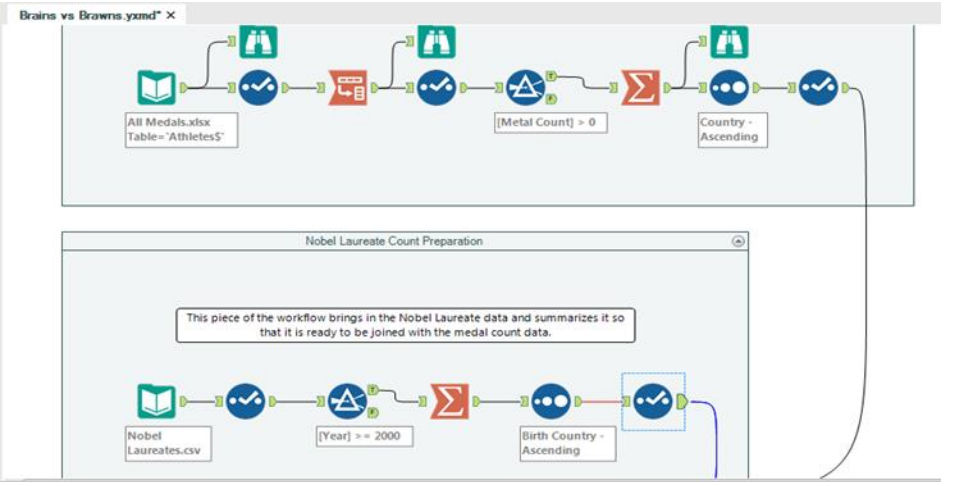


Figure 2-98 – Running unrelated analysis simultaneously

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

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

Browse (29) - Configuration | Brains vs Brawns.yxmd*

Results - Browse (29)

6 of 6 Fields | Cell Viewer | 943 records displayed, 35 KB

Record #	Birth Country	Category	Country	Name	Nobel Laureats	Year
1	Germany	Austria Chemistr	Germany	Richard Kuhn	1	1938
2	Germany	Chemistr	Germany	Fritz Haber	1	1918
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 Warcup Cornforth	1	1975
7	Austria	Chemistry	Austria	Friderik Pregl	1	1923
8	Austria	Chemistry	Austria	Richard AdolfZsigmondy*	1	1925
9	Austria	Chemistry	Austria	Richard Kuhn*	1	1938
10	Austria	Chemistry	Austria	Max F. Perutz	1	1962

Figure 2-99 – Nobel Laureates – Browse Configuration

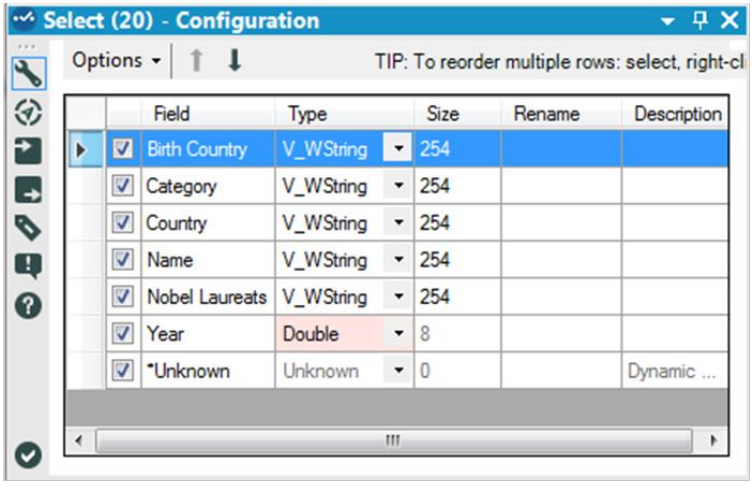


Figure 2-100 – Nobel Laureates – Select Configuration

Now that we know what the data structure is and that it parallels the medals file, we can start our preparation for the join.

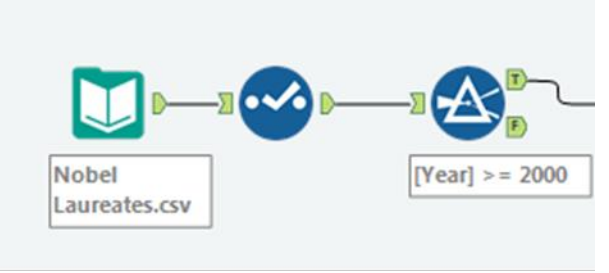


Figure 2-101 – Nobel Laureates – Preparing to join Medals

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

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

Results - Browse (29)

2 of 2 Fields

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 2-102 – Nobel Laureates – Browse after Summarize

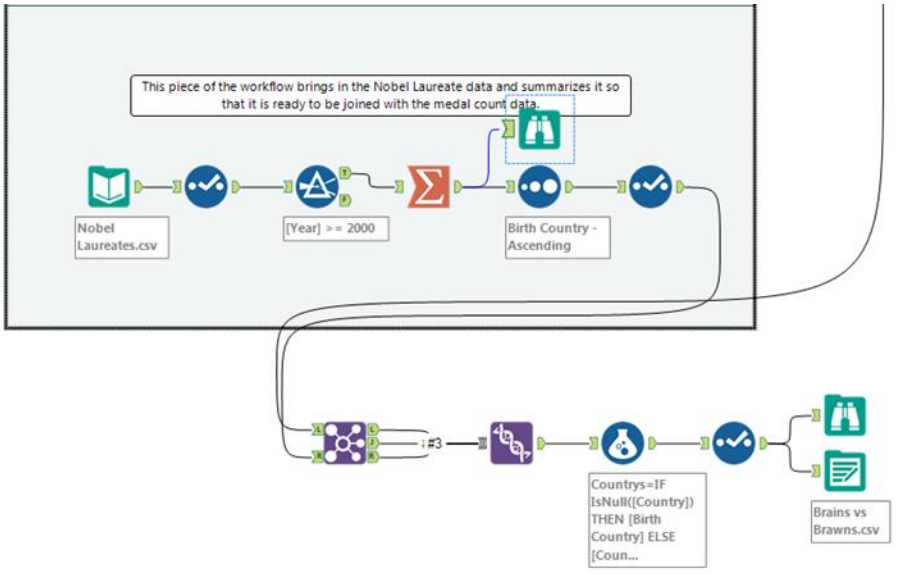


Figure 2-103 – Nobel Laureates – vis a vis All Medals

We see that we have a list of countries and a count of the number of Nobel Laureates. However, it is unclear what the number is because the field is called *Count*. We should rename it *Total Nobel Laureates*.

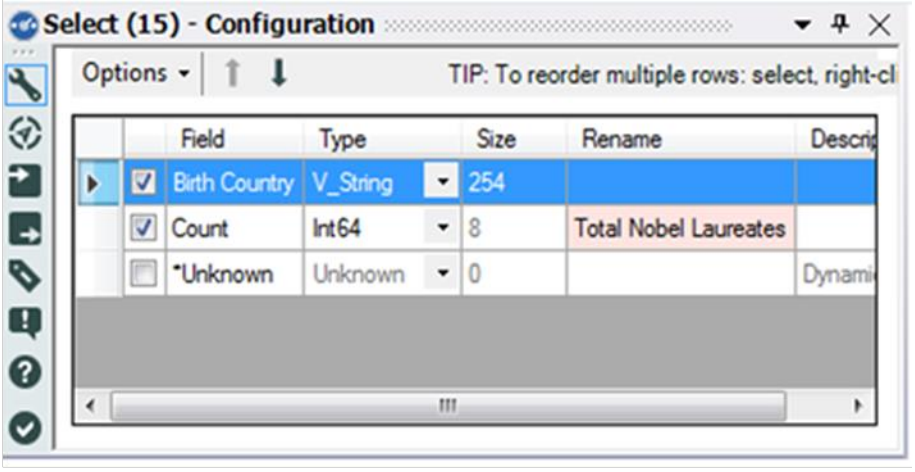


Figure 2-104 – Nobel Laureates – Select Configuration

We now have two data streams ready to be merged. We want to align the two datasets so that matching countries from each of the data steams share the same record, which means we want to join the data. Because we don't want to lose any data points if we have countries in one dataset but not the other, we will want to unite the three outputs from the join into a single data stream.

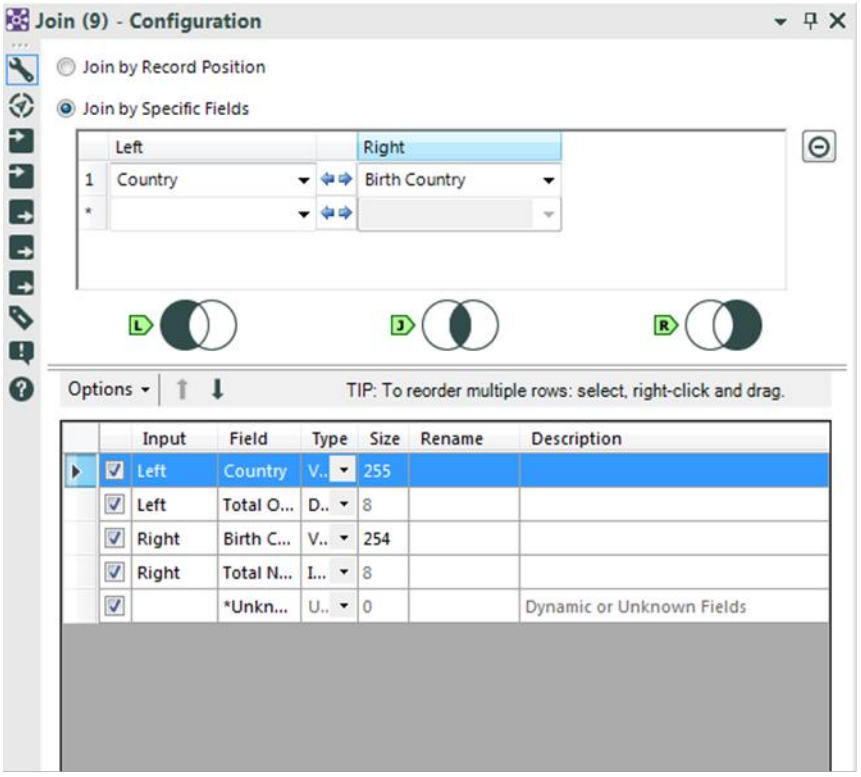


Figure 2-105 – Nobel Laureates – Join Configuration

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

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

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

Thus, we can see in the following image that we have relabeled the connections from *#1*, *#2*, and *#3* to *Left*, *Join*, and *Right*.

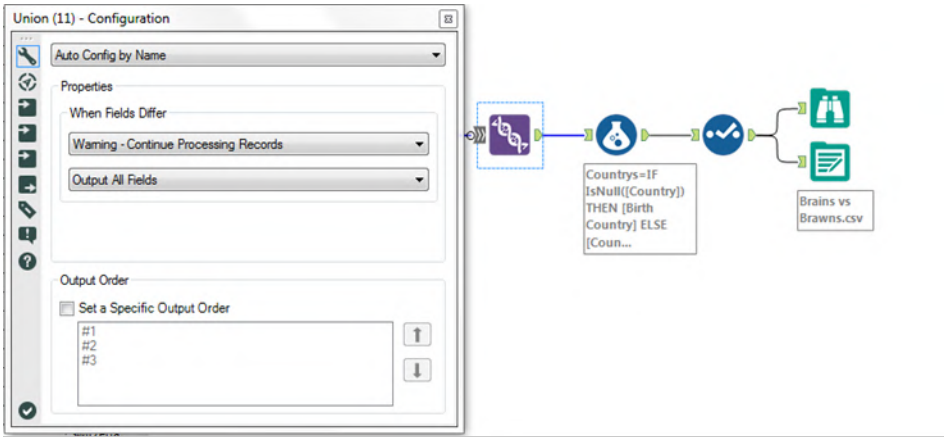


Figure 2-106 – Nobel Laureates – Union Configuration and Output Stream

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

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

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

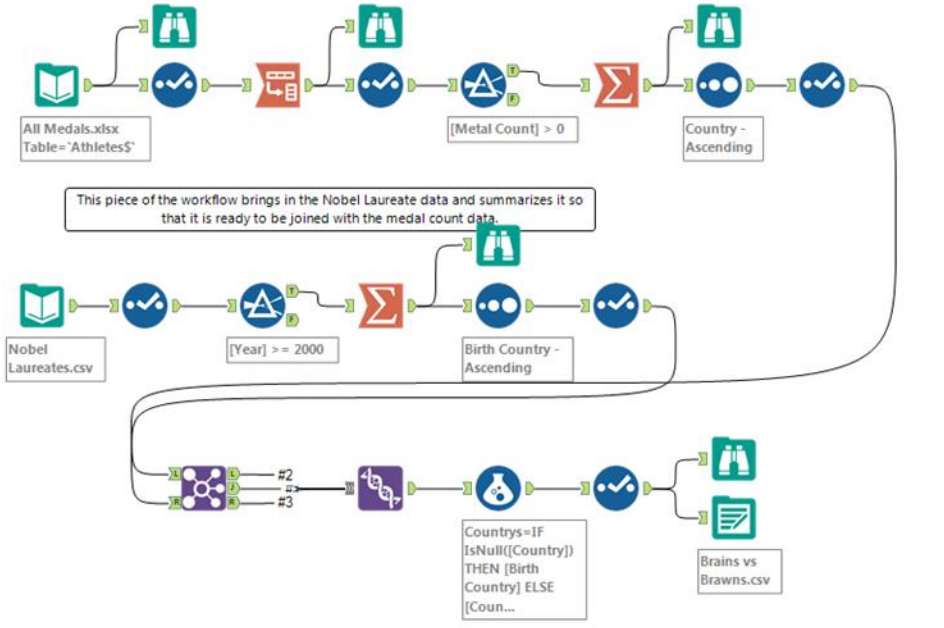


Figure 2-107 – Nobel Laureates – Comments and Containers

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

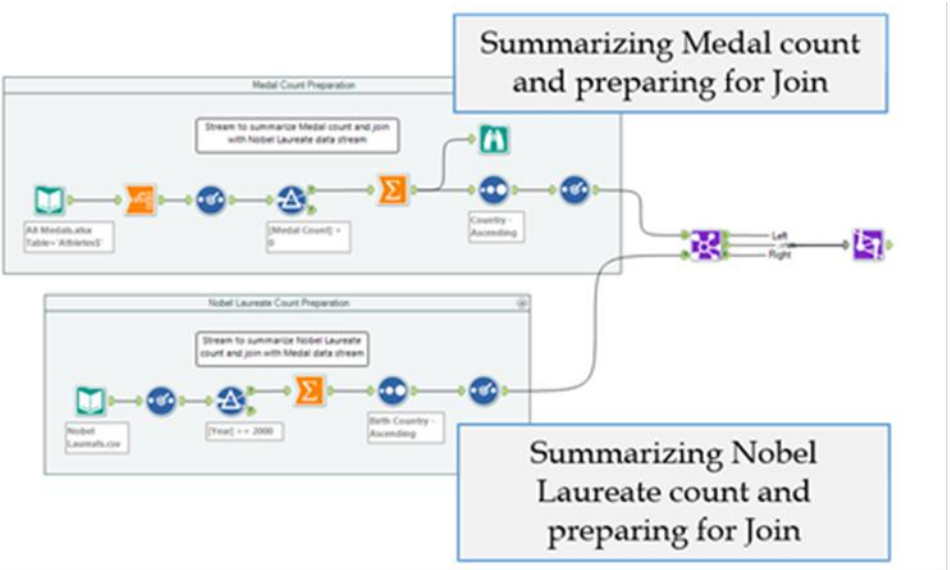


Figure 2-108 – Nobel Laureates –Output Stream with Comments

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

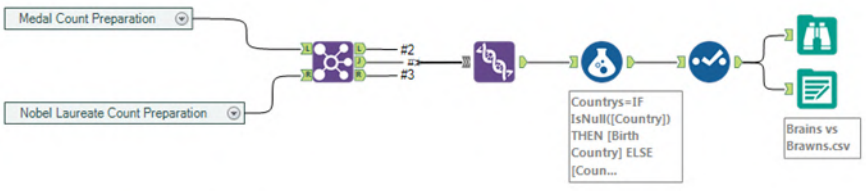
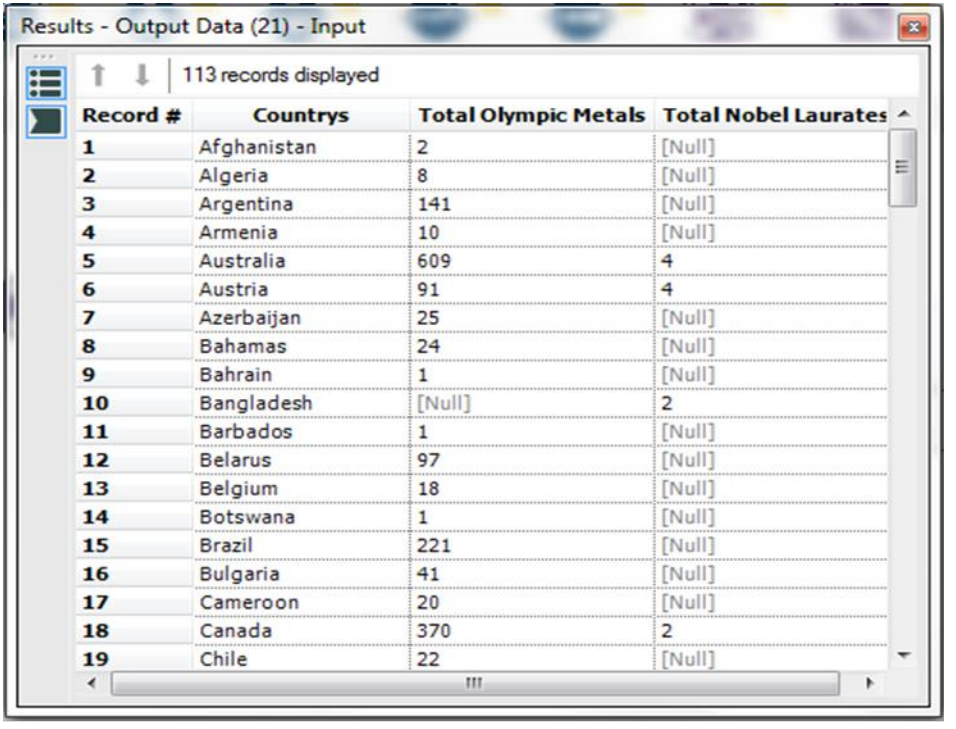


Figure 2-109 – Nobel Laureates –Simplified Output Stream

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


Results - Output Data (21) - Input

113 records displayed

Record #	Countrys	Total Olympic Metals	Total Nobel Laurates
1	Afghanistan	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]
16	Bulgaria	41	[Null]
17	Cameroon	20	[Null]
18	Canada	370	2
19	Chile	22	[Null]

Figure 2-110 – Nobel Laureates –Browse Configuration

We can observe from the Browse that the country names matched the names in both the Country and Birth Country fields. Let's create a conditional formula that allows us to convert the two columns with nulls into a single column that always has a Country name.

Add a formula tool to the end of the data stream with a formula called **Countries** with the formula: **IF IsNull ([Country]) THEN [Birth Country] ELSE [Country] ENDIF**. This will take the Country value unless it is null and the Birth Country if it is.

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

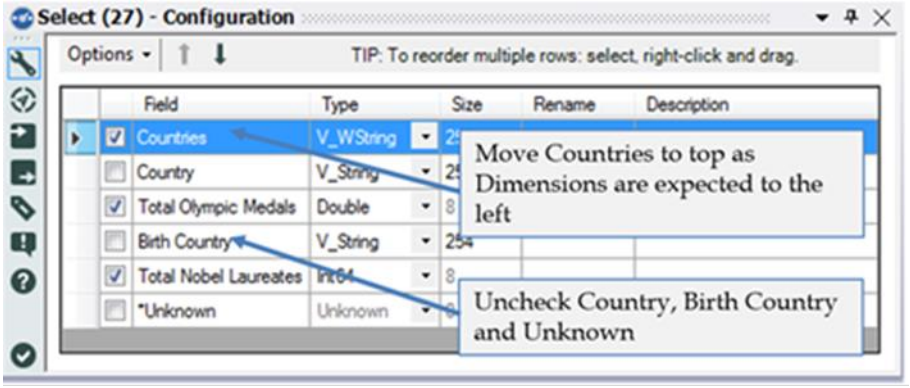


Figure 2-111 – Nobel Laureates –Select Configuration

The final workflow is as shown in the following figure.

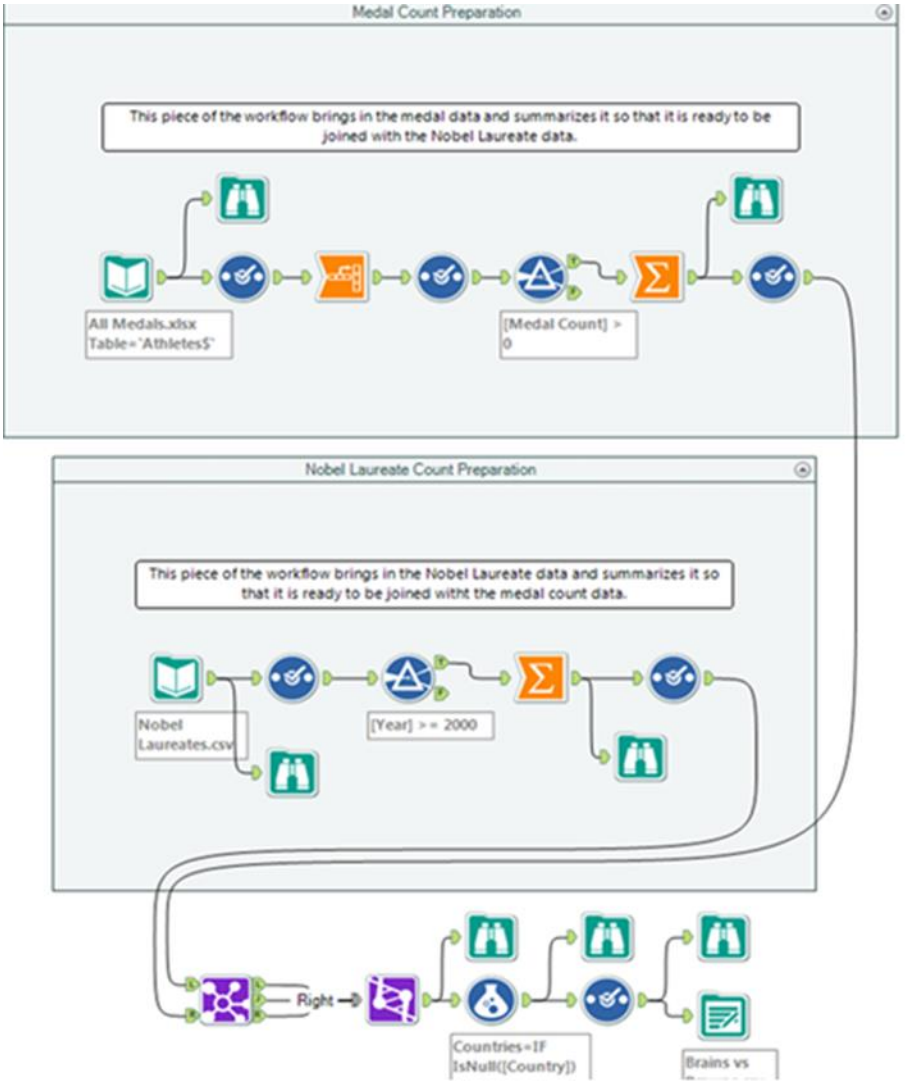


Figure 2-112 – Nobel Laureates –Complete Workflow

USEREADY

USEReady is a leading provider of advanced business intelligence, data management, and network security solutions. Our mission is to help businesses succeed by fast-tracking their business performance - through our industry, domain and technology expertise, agile global delivery practices, and our customer fanaticism. For more information, visit: www.useready.com.

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

- Laura Sellers,
Vice President,
Product Management,
Alteryx

Authored By

Abhilash Ramanathan
Anand Vadul
Ashish Tergaonkar
Jagdeesh Narayanan
Rahul Upadhye
Rima Upadhye
Vijai Narasimha
Vijay Gowtham

Published by

USEREADY

www.useready.com
contact@useready.com

ISBN 978-0-692-44795-6
90000>
9 780692 447956