from Alteryx to KNIME®

Author: Corey Weisinger

Table of Contents

Introduction

KNIME Analytics Platform is a powerful tool for data analytics and data visualization. It provides a complete environment for data analysis which is fairly simple and intuitive to use. This, coupled with the fact that KNIME Analytics Platform is open source, has led a large number of professionals to use it. In addition, third-party software vendors develop KNIME extensions in order to integrate their tools into it. KNIME nodes are now available that reach beyond customer relationship management and business intelligence, extending into the field of finance, life sciences, biotechnology, pharmaceutical, and chemical industries. Thus, the archetypal KNIME user is no longer necessarily a data mining expert, although his/her goal is still the same: to understand data and to extract useful information.

This book was written for people who are familiar with Alteryx and now interested in finding out to transition to KNIME Analytics Platform. Consider this book a bit like a foreign language dictionary: We look at how the most commonly used tasks are spoken in “Alteryx” and then translate them into “KNIME”. Find out, for example, how to import and manipulate data, how to perform modeling and machine learning, which includes sections on regressions, clustering, neural networks and components to name just a few. The appendix contains a useful quick tool to node reference.

KNIME Interface

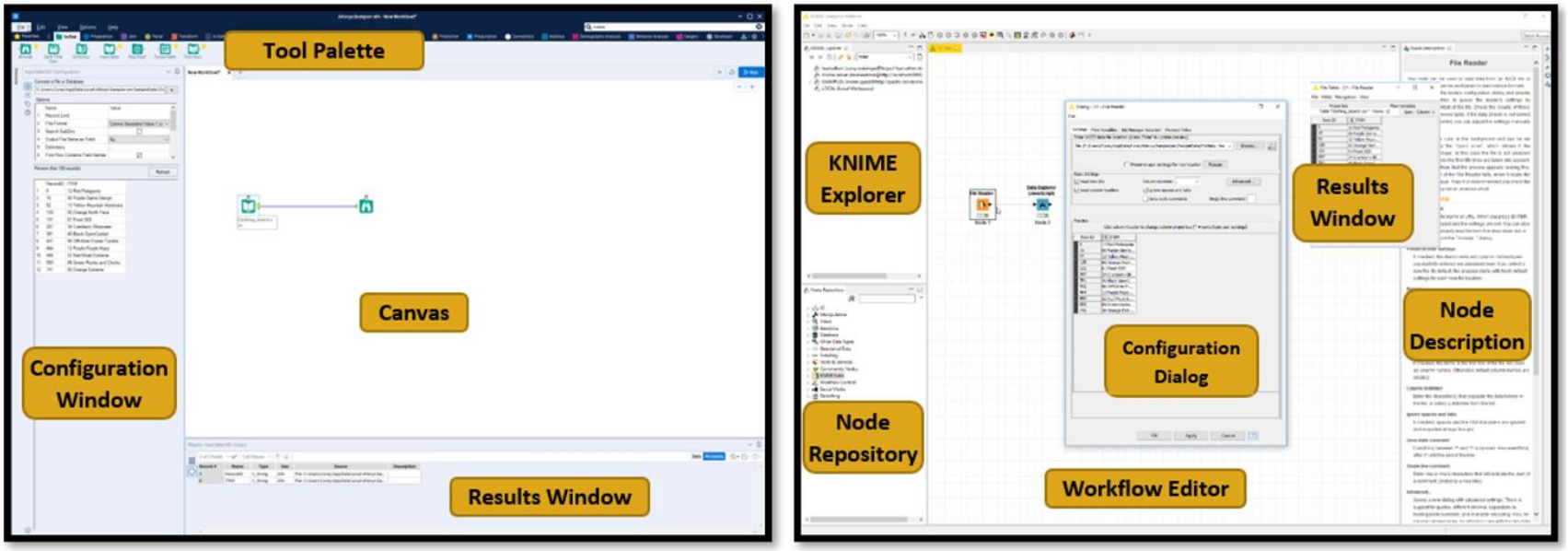


Figure 1 Left, Alteryx Interface. Right, KNIME Interface

KNIME Explorer
This is where you can browse files saved in KNIME workspaces; a workspace is just a directory, or folder, KNIME is connected to in order to store your KNIME workflows, node settings, and data produced by the workflow. For example, these files could be data source files like .csv or KNIME workflow files like knwf.

Node Repository
This is the equivalent of the Alteryx **Tool Palette**. In KNIME we call the tools "nodes" and they can be searched for from here and dragged into the workflow editor

Configuration Dialog
To configure a node in KNIME, you right click the node you wish to configure and select Configure. Unlike Alteryx, in KNIME the node configuration window is not always open.

Workflow Editor
This is the equivalent of the **Canvas** in Alteryx, it's where you drag & drop your nodes to build your workflow.

Results Window
Like the configuration window in Alteryx, the results window is not always open in KNIME, it can be accessed by right clicking a node and selecting the output you wish to view. Alternatively, the **Node Monitor** view can be enabled to show live data. We'll look at this in detail on the next page.

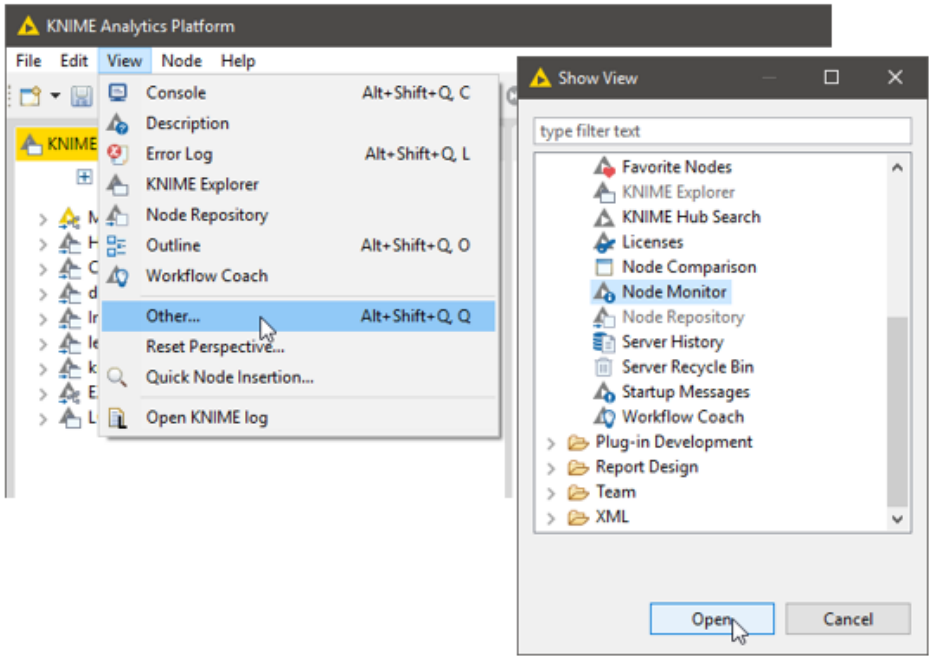


Figure 2 Where to find the Node Monitor

Node Description
This window provides a detailed description of what the selected node does. It includes an overview, information on the configuration options, and details on what each input and output port. A node's port is the equivalent of a tool's **anchor** in Alteryx.

Node Monitor
This optional view can be enabled by going to View > Other > Node Monitor and selecting
open. You can see where in Figure 2 to the left. Next, if you click the arrow in the **Node Monitor** view, you'll see a few different options here. Feel free to play around and see what each view displays but for now let's use the **Show Output Table** option (see Figure 3). This will give you an easy-to-see view of the output table of whichever node you have selected in your workflow, just like the normal results window in Alteryx.

The other views available allow you to see configuration settings, run time information, and **Flow Variables** that exist after the selected node. We'll cover what flow variables are later in this book but just keep the **Node Monitor** in mind if you're ever getting deep into their uses!

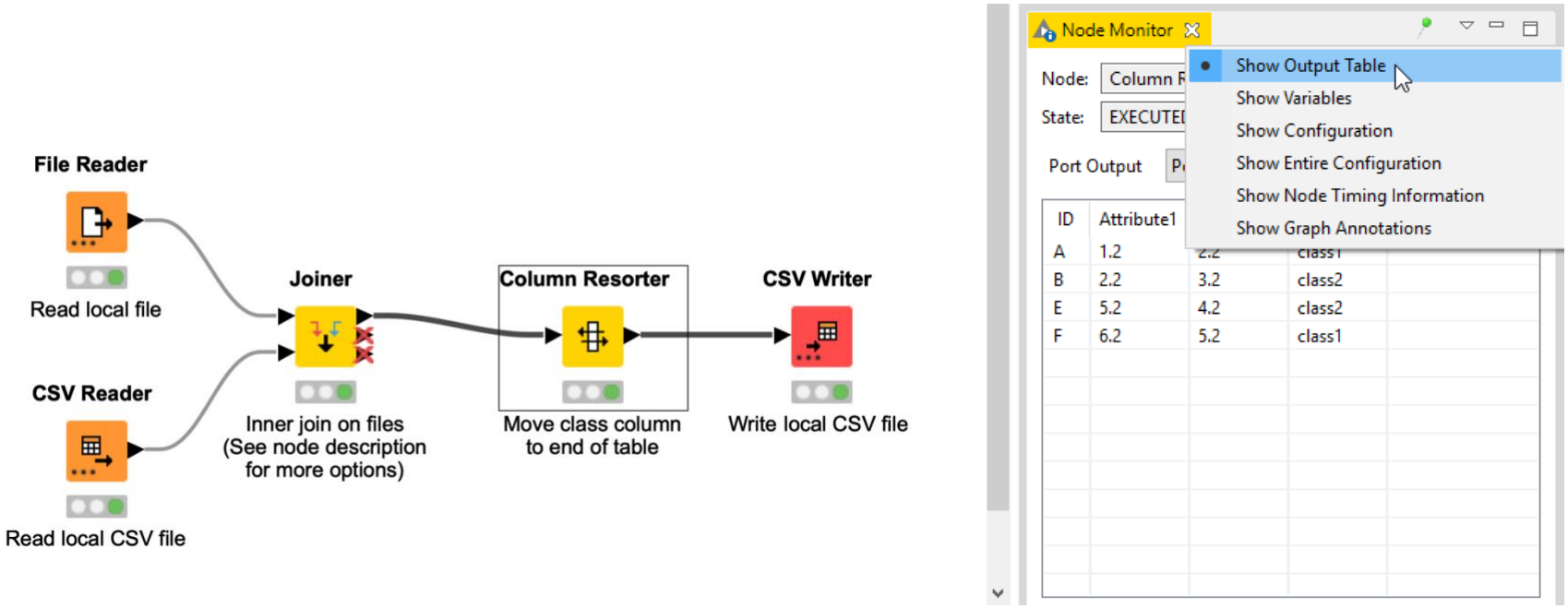


Figure 3 Node Monitor Show Output table setting and view

Node Interface

In KNIME, you build your workflow by dragging nodes from the Node Repository to the Workflow Editor, then connecting, configuring, and executing them. Like the tools in Alteryx, there are many nodes for many different tasks. The node's name is written above the node; below the node is an editable text description, which you can use to document in extra detail what each node is doing. Nodes have ports, these are the KNIME version of anchors. They are called input ports if they are on the left and output ports if they are on the right. They represent the data being fed into a node for processing and the data output from the node. Ports come in several varieties, the most common of these is the data port, represented by a black triangle. Another common type is the database connection port which is instead represented by a brown square in the same position. The node shown below, the File Reader, only has a port on the right, an output port. This is because no data is input into a file reader.

Traffic lights - identifying the status of a node

Unconfigured node:

If the traffic light below the node is red, the node has not yet been configured and it is not ready to be executed. A yellow triangle may show, detailing the error. In this case the node simply has not yet been configured.

Configured node:

Once the node is configured the traffic light turns yellow, this means the node is ready to be run and just needs to be executed. Some nodes may look like this when inserted into a workflow if they don't need specific configuration.

Executed node:

After a node has been executed its light turns green. At this point the data are available at the output port for viewing or further processing at the output port.

Importing Data

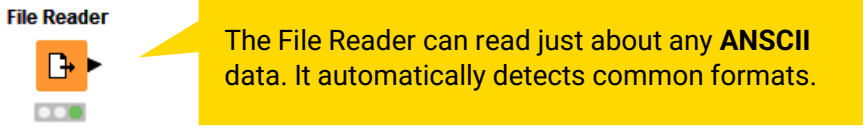
In Alteryx, all your data importing is done through varies configurations of the **Input Data** tool. In KNIME, a number of different nodes fill all the same roles. Here, we'll look at local files, databases, and other sources, to touch on a few of the most common options.

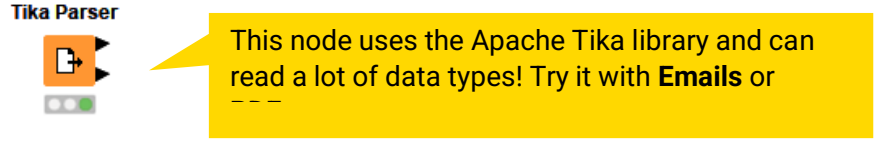
Local Files

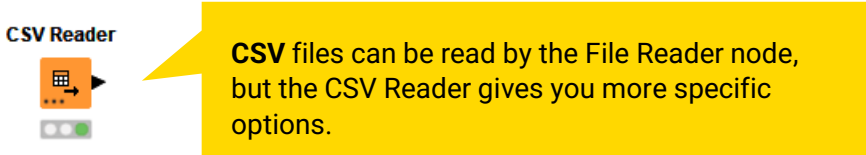
Figure 4 The Alteryx Input Data tool and the KNIME Reader nodes

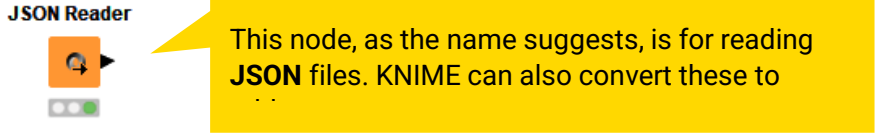
Local files, like Excel files, CSVs, PDFs, JSON, text files, and many others, are those typical files that just hang out on your hard drive. Similar to Alteryx, you can simply drag and drop the file you want to import into the Workflow Editor; KNIME automatically inserts the correct node needed to read it in.

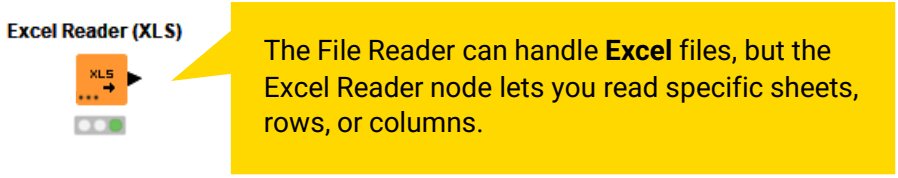
Let's look at each of the KNIME nodes one at a time, see what makes each one special. I'll give you a hint, it's the kind of files they can read and how they can be configured!

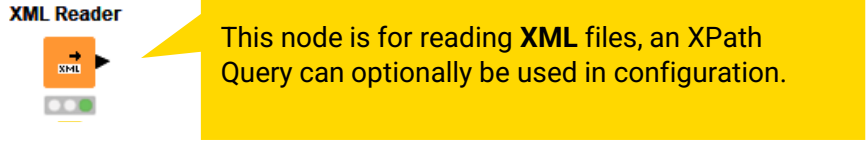






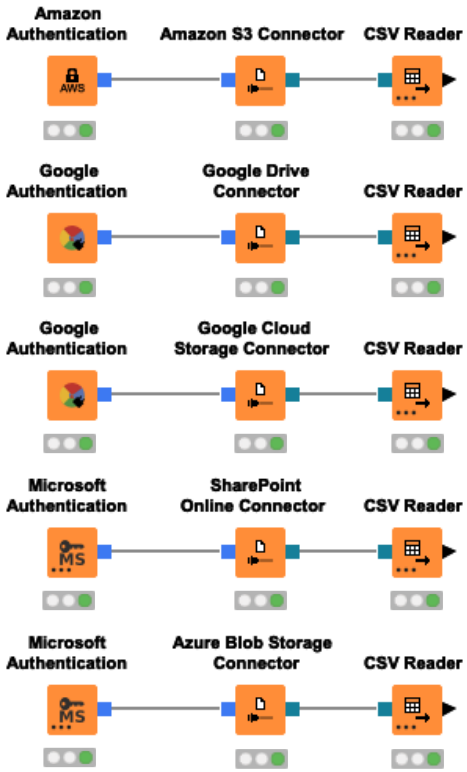






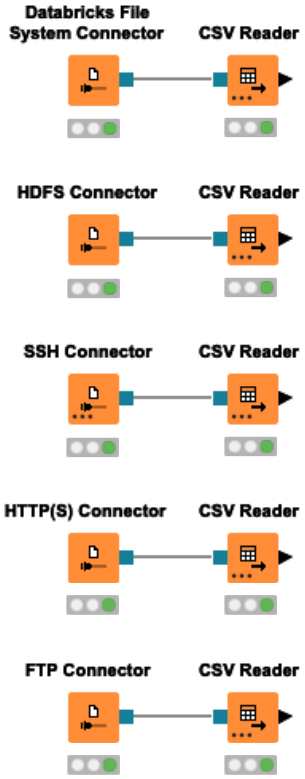
Remote File Connections

Many of these file reader nodes can even be used to connect to remote repositories such as Amazon S3 Buckets, Azure Blogs, Google Drives and more! On the compatible reader nodes, you'll see a set of three dots on the lower left corner of the node. In general, this is how we enable optional ports in KNIME. Click this icon and enable the optional port. Now we can connect the reader node to whichever remote file repository we want!



Some of these remote connections look like this. They require two nodes be connected before your reader. One for Authentication where you log into your system, and a second for establishing connection. These are typically systems we have multiple service integrations with.

Other "simpler" remote systems may only require one connector node where the authentication is integrated into the connector node. For example, grabbing a csv file from an FTP site is just two nodes as you see in the bottom example.



Databases

Now let's talk about connecting to databases; the first thing I want to point out is that you can't see any data ports on most of the KNIME nodes below. As a reminder, data ports are the black triangles on each side of a node that denote KNIME table data flowing into and out of the nodes.

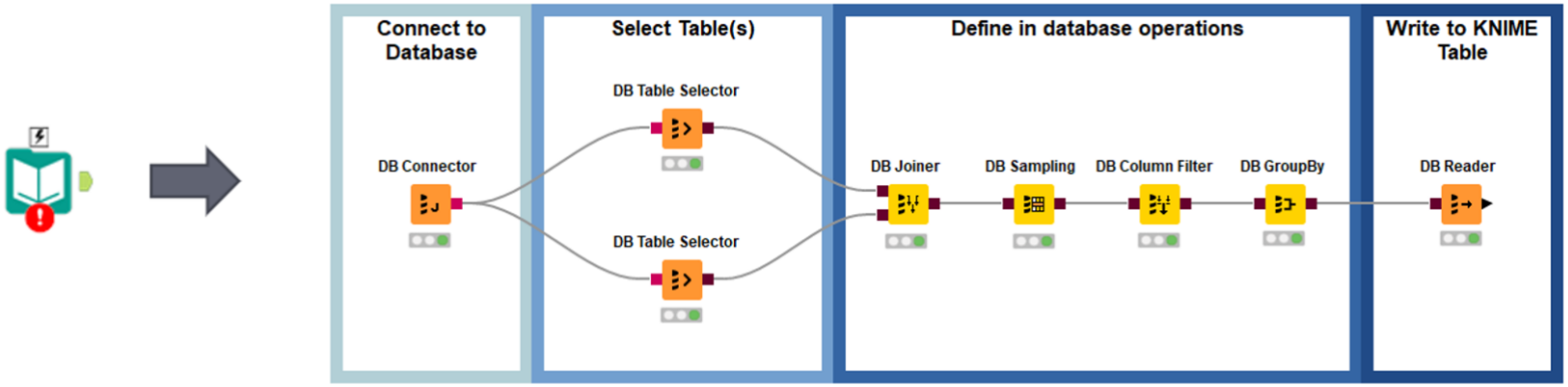


Figure 5 Alteryx database connector tool and the KNIME equivalents

In Alteryx this is like using the Connect In-DB tool and Data Stream Out tool:

Figure 6 Alteryx Connect In-DB and Data Stream Out tools

So how do we connect to the database in KNIME? This is done with the **Database Connector** node, be it a traditional format like MySQL, or a Hadoop based one like Impala or Hive. Once that connection is established, we can select a table in the **DB Table Selector** node. The **DB Connector** node at the far left (of the KNIME workflow in Figure 5) is a generic connector, it references a JDBC driver and connects to anything that supports one.

In **Error! Reference source not found.**, you can see the **Query** folder, opened in the Node Repository, so let's address that too while we're here. The Alteryx tools above would normally be used with several other of the In-Database tools, sorting, filtering, joining, i.e. standard processing primarily. The advantage of running the process in the database, is speed and the fact that it limits the data to be transferred. KNIME supports this functionality; in the adjacent figure on the right you can see some of the manipulations available for in-database processing with KNIME. All the nodes in the Query folder can be inserted into a workflow between a **DB Table Selector** node and a **DB Reader** node. What KNIME does here is use these nodes to generate SQL code automatically for you. This is handy if you're not an SQL expert or want to easily go back and modify your query.

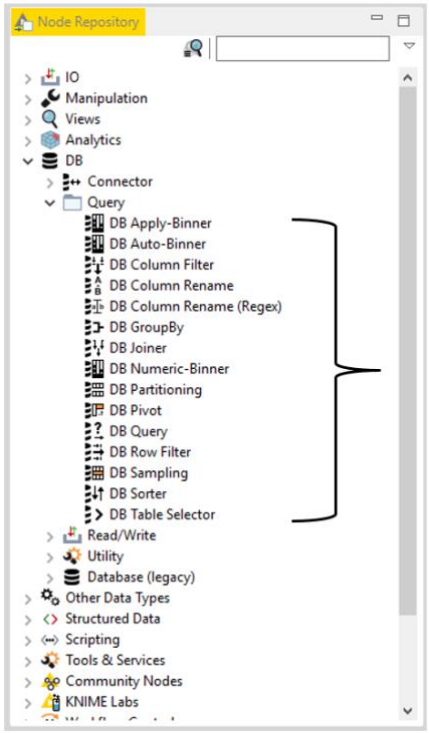


Figure 7 Node Repository with the Query folder expanded showing the DB Query nodes

Other Sources

Local files and databases aside, there are so many other data sources to access! We'll look at a few of those in this section. Two connection environments in this section are the Twitter nodes and the Google nodes. These can all be found in the **Social Media** section of the **Node Repository**. The **Twitter API Connector** requires you supply an API Key in the configuration window. The **Google Authentication** node is even easier to configure; in the node configuration dialog, click the Authenticate button. This opens the familiar Google account login screen in your web browser, where you can approve access to your files for KNIME to your files - and you're done!

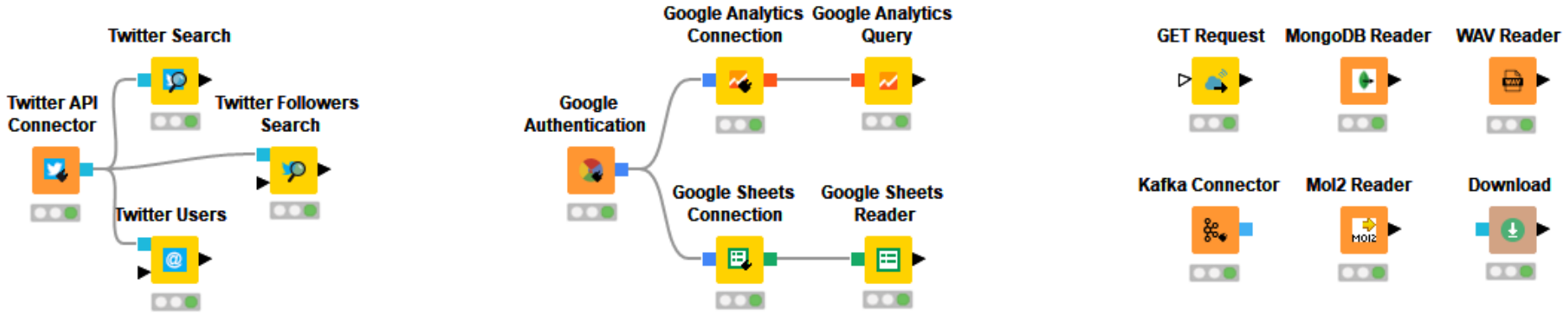


Figure 8 KNIME Nodes from the Social Media section of the Node Repository

For brevity, we'll just look at two of these nodes here. Feel free to explore other nodes yourself. Remember the node description in the KNIME Workbench or on the KNIME Hub tells you everything you need to know about how to configure the nodes. The **Get Request** and **Download** nodes, for example, are particularly great resources!

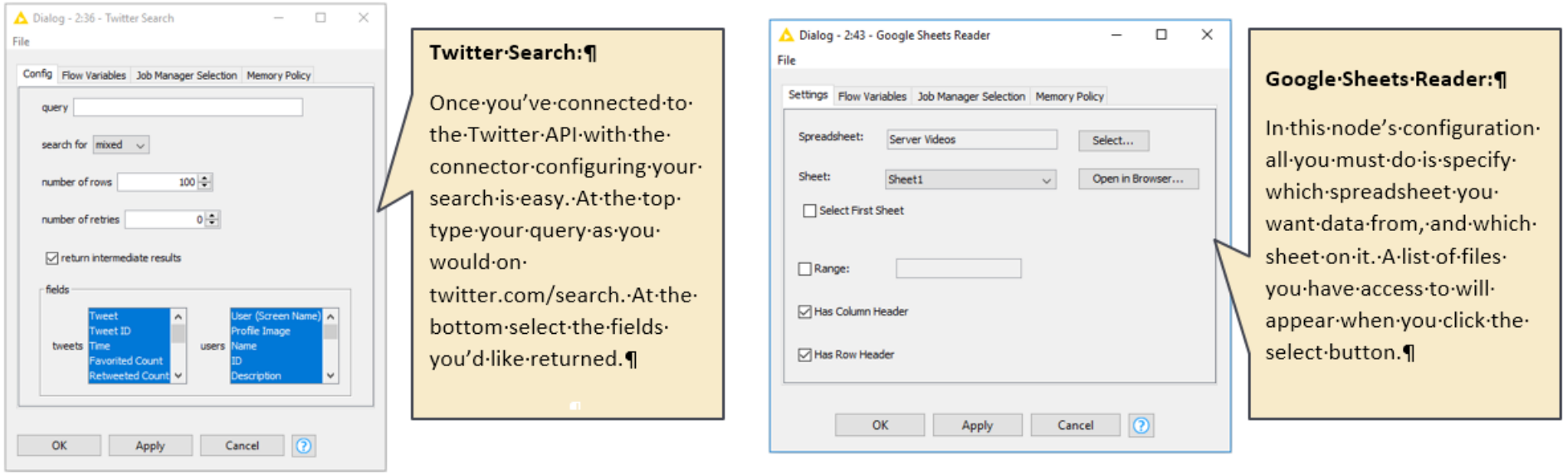


Figure 9 The configuration dialogs for the Twitter Search and the Google Sheets Reader nodes

Writing Data

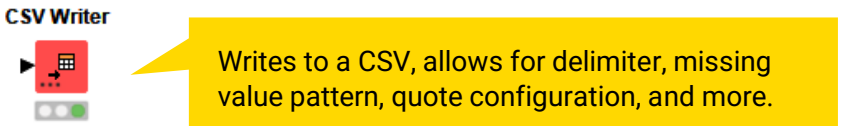
So far, we've had a look at the interface – the KNIME Workbench; and how to get data into KNIME. Now, let's have a look at getting data out of KNIME. Below you'll first find a list of the nodes for writing local files and, second, a section that looks at getting your data into databases. After we've finished here, we will dive into some of the tools for handling data in KNIME!

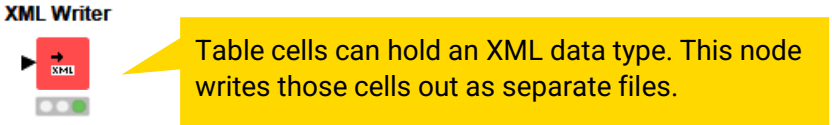
Keep in mind that the remote connections also work while writing data! Simply click the 3-dot icon and select add connection port. Then you'll be free to connect whichever remote connection you need! See page 8 for info on remote connections.

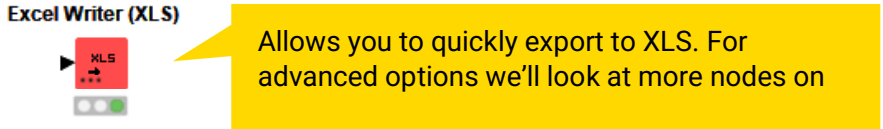
Local Files

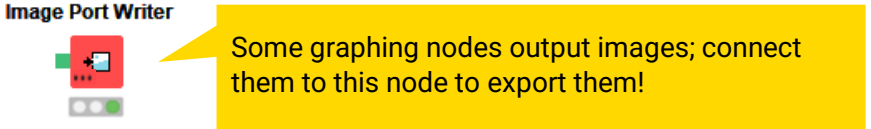
Figure 10 Alteryx tools for writing data and the equivalent KNIME nodes

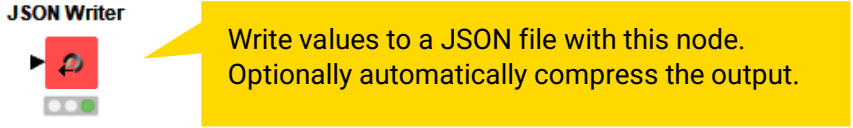
The nodes listed here are for writing local files, both standard data storage formats like CSV, Excel, and JSON, which, in Alteryx, you would write with the **Output Data** tool, and images and PDFs, which you'd write with the **Image** and **Render** tools in Alteryx. Again, the main difference here is that in Alteryx your output tool can be configured differently to perform different tasks and in KNIME we have separate nodes for these separate tasks.

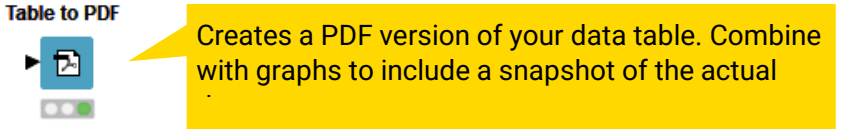












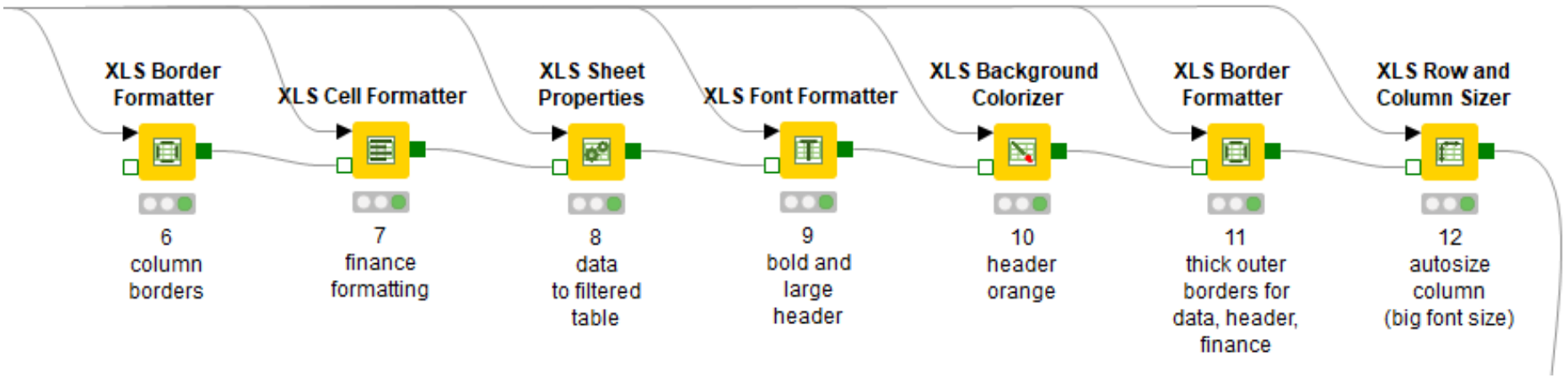


Figure 11 A string of XLS Formatter nodes

Above you see a string of XLS Formatter nodes, these are linked together and each change bits of the formatting in an Excel file you're preparing to write. The modular nature makes it easy to customize your formatting as much as you like, as well as add or remove parts. There is a variety of nodes for this purpose; if this is a major use case for you, check out the linked guide below for a full introduction to formatting Excel files in KNIME.

- https://www.knime.com/community/continental-nodes-for-knime-xls-formatter

Databases

Writing to databases is easy with KNIME and there's only one major difference between KNIME and Alteryx: when reading from a database in KNIME, the connection info is stored in a separate node - the **DB Connector** node. It supplies one of two inputs required by the **DB Writer** node, the other being the data port containing the table you want to write to your database.

Note: This exact same node can also be used to feed the **DB Table Selector** node when reading from a database, and, by swapping the database info in the connector you can easily transfer from a development to a production environment.

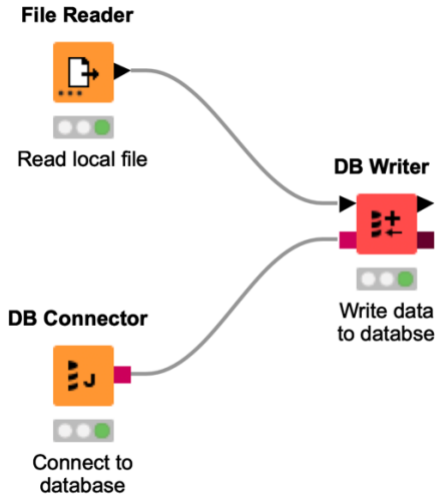


Figure 12 The File Reader, DB Connector, and DB Writer nodes

Figure 12 shows the three nodes you need to write to a database:

- **File Reader:** data you wish to append to your database, this could also be processed by your workflow before writing.
- **DB Connector:** supplies the information for connecting to the database, e.g. login credentials
- **DB Writer:** is where you specify the name of the table you want to write to as well as which columns you want to write to it

Manipulating Data

Filtering Data

Row filtering in KNIME is done with a few different nodes: the **Row Filter**, the **Rule-Based Row Filter**, and the **Row Splitter**, which is for collecting unmatched rows. For column filtering, the **Column Filter** node is your main stop! In Alteryx the **Select** tool has several purposes, filtering columns being just one. KNIME's **Rename Column** node fills in the other uses!

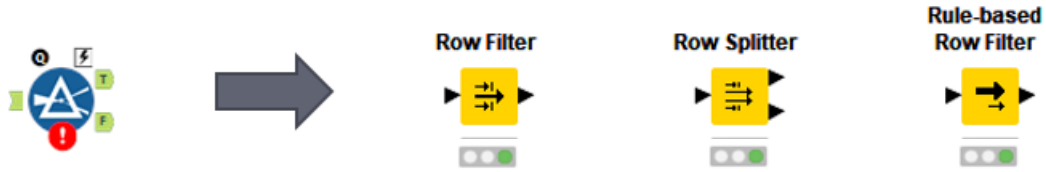
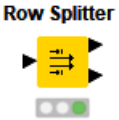
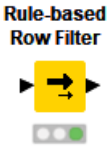


Figure 13 Alteryx Select tool is the same as the Row Filter, Row Splitter, and Rule-based Row Filter nodes in KNIME

Allows for quick filtering by way of string pattern matching, a numeric range, or missing values. This filtering can be performed on a column or the row ID itself and can be set as an include or exclude filter.



This node works just like the **Row Filter** above except that it exports both the included and excluded rows. You'll notice that it has two output data ports (represented by the black triangles on the right of the node). The top port is for included rows and the lower port is for excluded rows.



The **Rule-based Row Filter** node is akin to the custom filter option in Alteryx' Filter tool. You enter a set of rules, which are run through one by one. The first to match the row ends the process, e.g.:

$Score$ > 0.9 => TRUE
$Name$ = "Jon Smith" => TRUE

…to include all scores over 90%, and also, Jon's.

Sorting

Sorting data is an easy transition as both applications have one tool/node for this, and they're even named, well, similarly, **Sort** and **Sorter**! The KNIME **Sorter** node is configured just like the Alteryx tool: just set a list of columns to sort by and note if they should be ascending or descending. In KNIME you also have the option to move all missing cells to the end if desired by checking the box at the bottom of the configuration dialog.

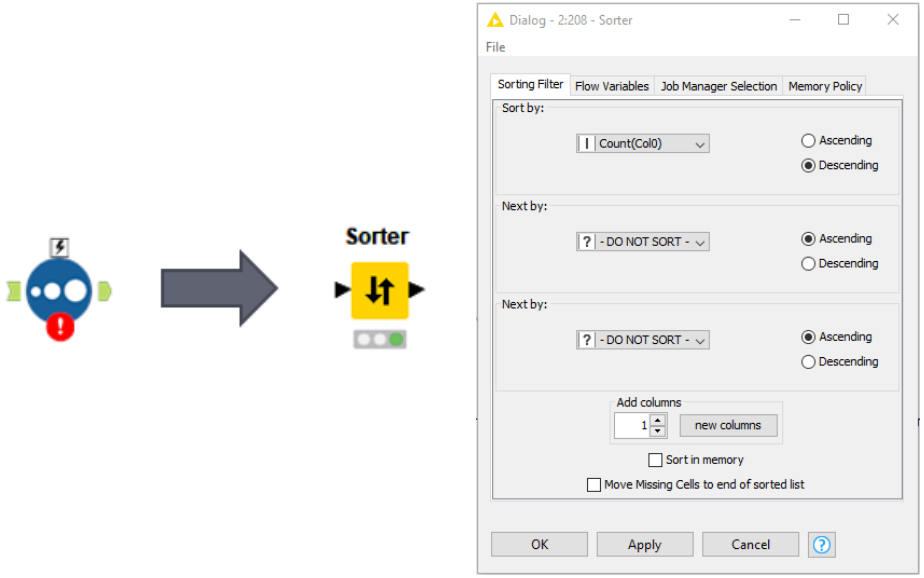


Figure 14 Alteryx Sort tool or the KNIME Sorter node and the KNIME Sorter node configuration dialog

You can also sort columns in KNIME, to do this simply using the **Column Resorter** node. You can sort alphabetically, or manually. This may be helpful when verticly combining tables with different column names or when combining multiple columns into a list data type with the the **Column Aggregator** node.

Aggregating Data

Basic aggregating of data is another one to one conversation between KNIME and Alteryx: the **Summarize** tool to the **GroupBy** node.

Figure 15 The Alteryx Summarize tool and the KNIME GroubBy node

In this case the configuration window looks quite a bit different in KNIME. To quickly summarize, in one tab you set the columns to use for creating groups:

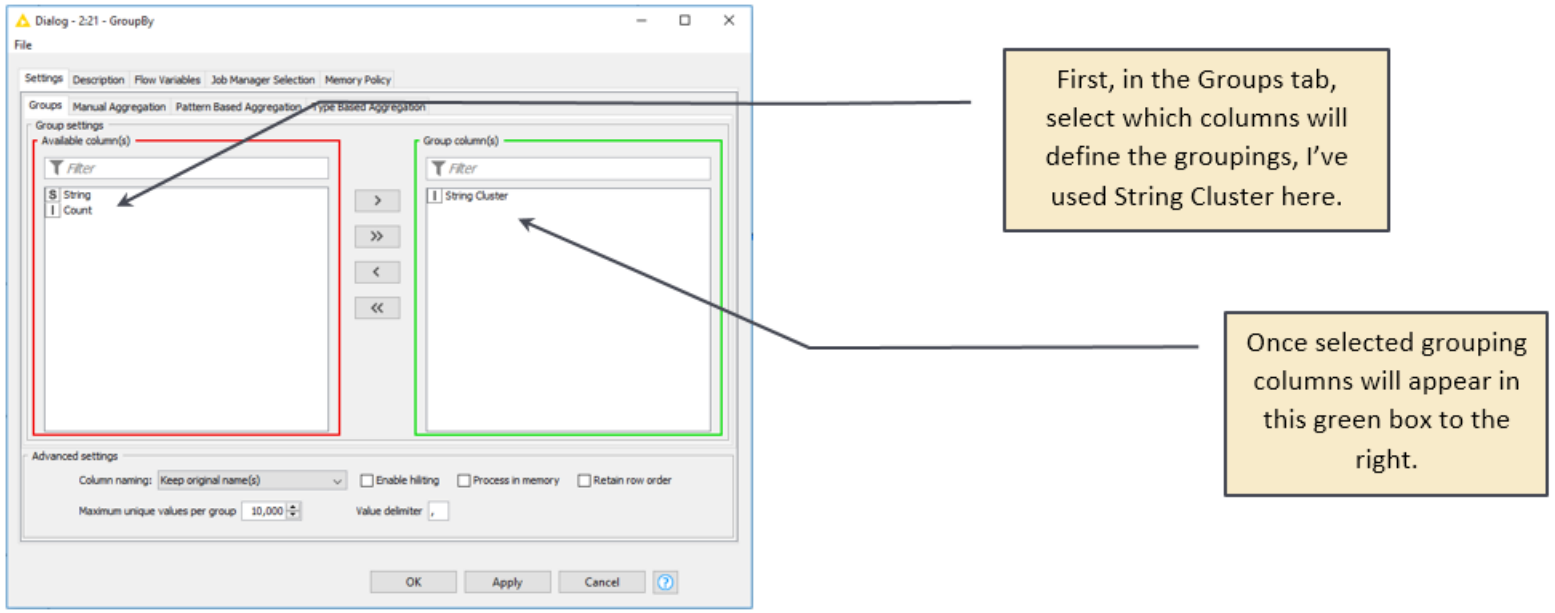


Figure 16 Groups tab in the Groupby configuration dialog

In the second tab (under Settings) you set which the method you want to use to aggregate the remaining columns:

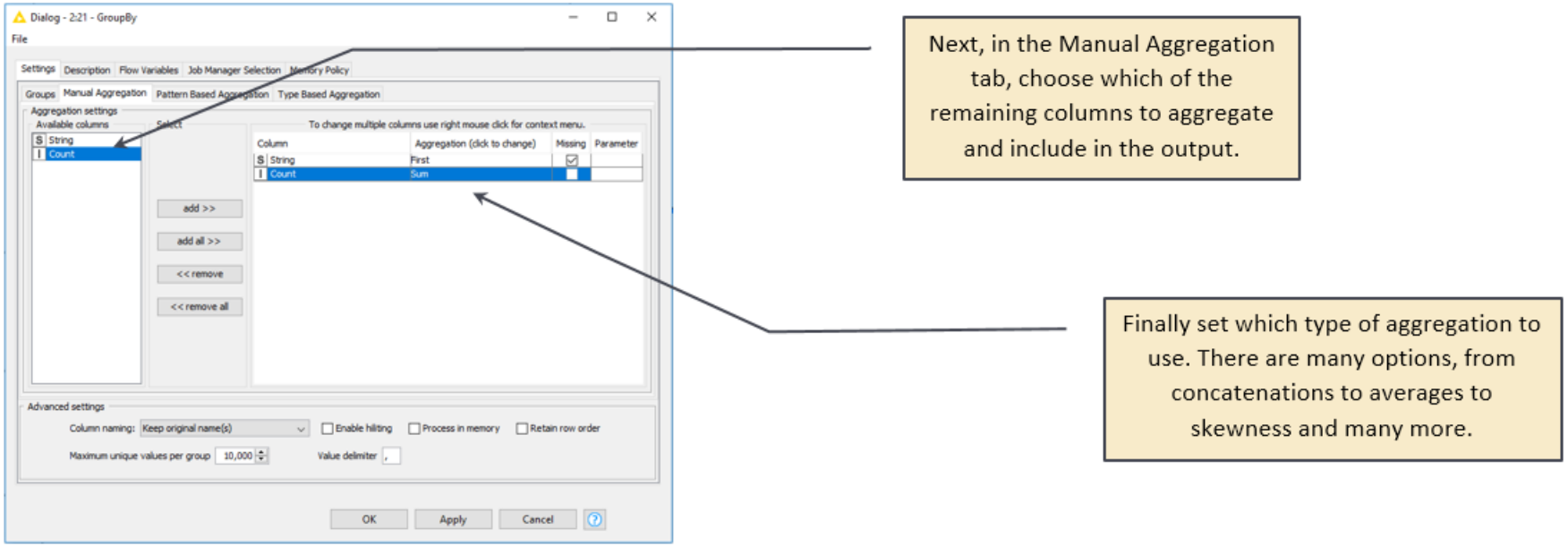


Figure 17 Manual Aggregation tab in the GroupBy configuration dialog

String Data

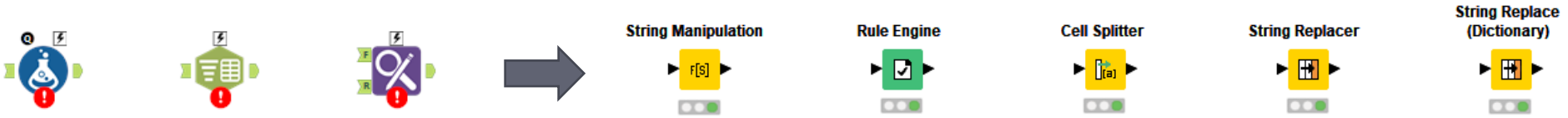


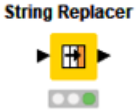
Figure 18 Alteryx Formula, Text to Columns & Find Replace tools and the KNIME String Manipulation, Rule Engine & Cell Splitter nodes

In this section, we touch on a few options for manipulating string data, namely the KNIME equivalents to the **Formula**, **Text to Columns**, and **Find Replace** tools in Alteryx. The **Formula** tool is most like the **String Manipulation** node, it is for writing basic string alteration instructions. The **Rule Engine** node is similar as well but can be use in more complicated ways as it allows for 'if then' type functionality. The **Text to Columns** tool can be replaced by the **Cell Splitter** node.

Use this node for things such as removing white space, removing punctuation, regex expressions, sub string creation, capitalizing and more.

The Rule Engine has a lot of the same functionality as the String Manipulation node. You can this for more control. For example, use this to reformat strings differently base on which source they are from.

This node will take one string column and split it into multiple columns based on a specified delimiter. A comma for example. Unlike the Alteryx equivalent, you do not need to specify the number of expected output columns. Rows with fewer than the max will simply have missing values in the right most columns.



Use the String Replacer for quick replacements or even removals inside strings. For example, configure this node to replace all instances of "two" with 2. This node also supports regular expressions.

You can direct this node to a text file formatted as detailed in the Node Description window. There's a little more setup here but with it you can easily replace a large set of strings. Otherwise, it functions as the String Replacer node above.

Numeric Data

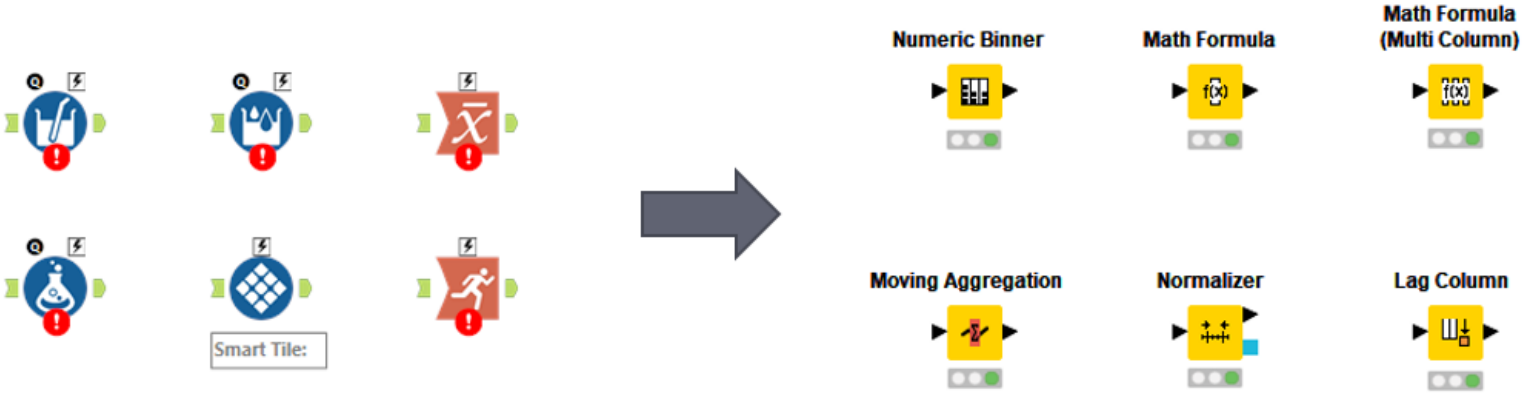
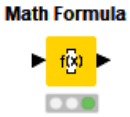


Figure 19 The different Alteryx tools and KNIME nodes for handling numeric data

There is a near endless variety of ways to manipulate numbers while preparing, analyzing, and modeling data. We'll touch on a few common examples and discuss how to get started with these manipulations in KNIME.

This node allows for custom defined numeric bins. Try the **Auto-Binner** for automated options like equal sized or equal interval bins.

The **Moving Aggregation** node will take the place of the running total node. It can be configured in many ways, check it out!



Like the **Formula** tool, the **Math Formula** node will allow you to alter numeric data with common math functions.

The **Normalizer** will stretch or compress your data to be within a given range, commonly 0 to 1.

This node functions just as the **Math Formula** node except it alter multiple columns at once!

KNIME does not have a **Multi-Row Formula** equivalent, but the **Lag Column** node will allow you to move values down to the next row. With a little work it can be combined with the **Math Formula** node to create a similar effect.

Multi-Row Calculations

Alteryx has a tool called the **Multi-Row Formula**. This is done in KNIME through a combination of the **Lag Column** node and the **Math Formula** node. The first creates a new column of shifted values. For example, if you want to reference the previous row in your formula you can use the **Lag Column** node with a value of 1 before applying the **Math Formula** node. This creates a new column for you to reference.

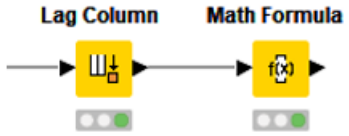
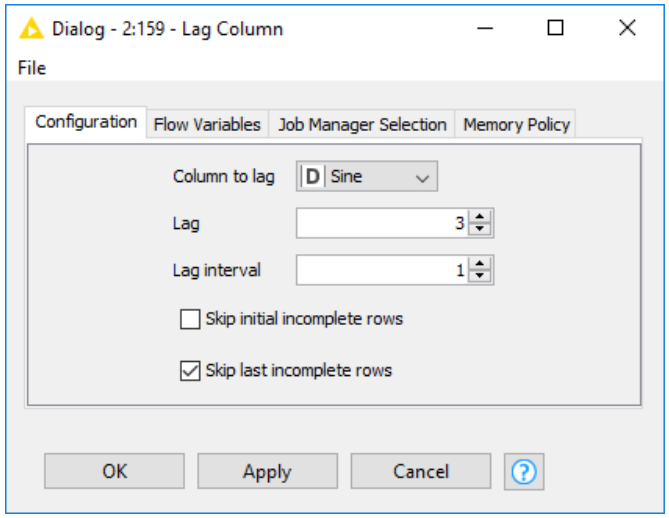


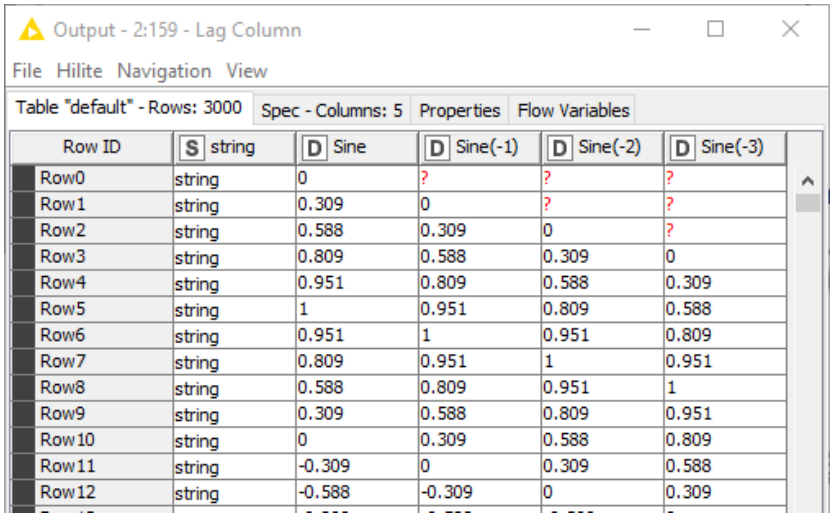
Figure 20 Lag Column and Math Formula KNIME nodes for recreating Alteryx' Multi-Row Formula tool

Lag Column Configuration

To use the lag column node, you first select the column to lag in the drop down menu shown on the left of

Figure 21. Next, you select the Lag and Lag Interval, this means you specify the number of lagged columns to create (Lag) and the number of rows to lag each time (Lag interval). I chose Lag = 3 and Lag Interval = 1, so I have created three columns, each lagged one from the last.





Row ID	S string	D Sine	D Sine(-1)	D Sine(-2)	D Sine(-3)
Row0	string	0	?	?	?
Row1	string	0.309	0	?	?
Row2	string	0.588	0.309	0	?
Row3	string	0.809	0.588	0.309	0
Row4	string	0.951	0.809	0.588	0.309
Row5	string	1	0.951	0.809	0.588
Row6	string	0.951	1	0.951	0.809
Row7	string	0.809	0.951	1	0.951
Row8	string	0.588	0.809	0.951	1
Row9	string	0.309	0.588	0.809	0.951
Row10	string	0	0.309	0.588	0.809
Row11	string	-0.309	0	0.309	0.588
Row12	string	-0.588	-0.309	0	0.309

Figure 21 The Lag Column configuration dialog and the Lag Column output

If you need to Lag multiple original columns simply apply a second Lag Column node to your workflow. After you've created the lagged values you need for your calculation you can call them just like you would any other value in your formula node of choice.

Missing Data

For the basic correction of missing data like that handled by the **Data Cleansing** tool in Alteryx, use the **Missing Value** node in KNIME. It has options such as removing rows with missing data, using the previous value, the max, average, moving average, and more. For more complicated corrections for missing data, such as altering the value differently based on another field, try the **Rule Engine**. This node has come up a lot in our introduction to KNIME, but it really does have a large variety of uses when you want to make decisions based on many fields.

Fig. 21: How missing data is handled in KNIME in comparison with Alteryx

Sampling Data

Whether you want to sample data to reduce execution time for analytics or constructing training sets for machine learning and modeling there are many options available in KNIME.

Figure 22 Sampling data with the Partitioning, Bootstrap Sampling, Equal Size Sampling, Row Sampling & Database Sampling nodes in KNIME

The **Partitioning** node allows you to split your data into two sets based on either a percentage or a number of records. There are a few options for how these partitions are drawn: from the top, linear sampling, random sampling, and stratified sampling. The node description defines these terms well, so don't forget to look them up on the KNIME Hub, if you're unsure. The **Bootstrap Sampling** node allows for the use of the bootstrapping technique for oversampling your data artificially, creating a larger dataset. Equal size sampling requires that you pick a nominal column to define the different classes; it then creates a sampled set with an equal number of records for each class. This can be helpful when training models based on counting algorithms like decision trees. Finally - remember there is a **Database Sampling** node. Performing sampling on the database end will save time when transferring data to KNIME for analysis.

Table Manipulations

I'd now like to hit on a few basic operations for manipulating tables as opposed to strictly manipulating the data within. he **Joiner** node combines tables horizontally, while the **Concatenate** node combines them vertically. **Pivoting** allows you to create additional columns in your table by effectively pivoting up some of the rows. **Unpivoting** is the inverse of this process and enables you to reduce the number of columns in a table by creating more rows to store the data.

Figure 23 Alteryx tools vs KNIME nodes for table manipulation: join, concatenate, pivot & unpivot

Use the **Concatenate** node to vertically combine tables. This node will match fields by name and can be configured to retain either a union or intersection of the columns in the two input tables.

The **Joiner** node in KNIME is going to replace your **Join** tool in Alteryx. There shouldn't be too much to get used to here simply select the fields you wish to match and the type of join: inner, left-outer, right-outer, full outer. The bottom two outputs also supply the anti-join rows. The values that did not match.

Configuring this node will be straight forward if you're familiar with pivot tables, just choose 3 things. The columns to be used as pivots the contents of which will become new columns. The columns to be used as groups, this will let you aggregate the rows as you pivot. And the aggregation methods for the fields you wish to retain.

Setting up the **Unpivoting** node I easy as well. Just select the columns you wish to rotate back down in to distinct rows, the value columns. And select the columns with values you wish to retain, the retained columns.

Documenting Your Workflow

KNIME offers several options to keep your workflow organized. Using them all in conjunction will keep your workflow clean, presentable, and easy for your coworkers to read. Pictured below from left to right are: a node with a comment, an annotation, and a named metanode with a comment.

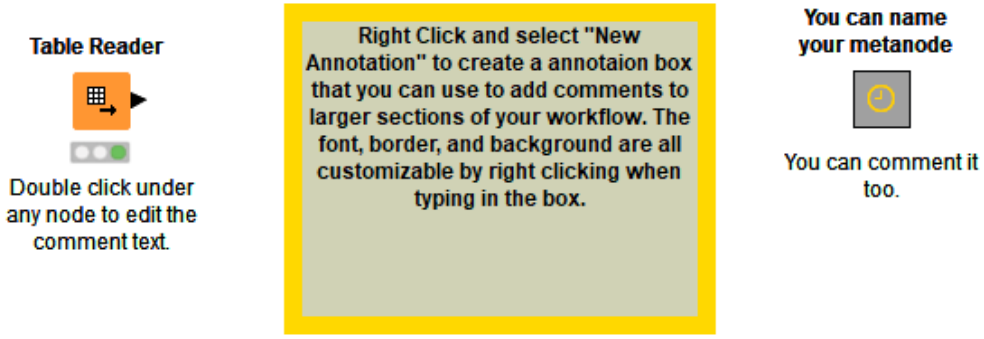


Figure 24 The different options in KNIME to document your workflow and keep it organized

Node Comments

By double clicking the text underneath a node you can edit the comment. Use this to note changes or just to give more detail on exactly what the node is doing in your workflow. You can comment nodes, metanodes, and components.

Workflow Annotations

Workflow annotations are colored boxes you can place over your workflow, as can be seen in many KNIME examples. A common use is to clearly separate sections of your workflow into data loading, ETL, modeling, and predicting. This makes it easy for colleagues to quickly identify the area they're looking for. You can customize the color of the border, the background, and text font / size.

Metanodes

Metanodes are like a subfolder inside a workflow. They are a container around a selection of nodes. To create a metanode simply highlight all the nodes you want to put inside and right click to select Collapse into Metanode. This won't affect how your workflow runs at all, it simply helps to structure the view visually. When you collapse your nodes into a metanode you can select what to name the metanode: this is the text that appears above the node. You can also comment your metanodes just like normal nodes by double clicking beneath them.

Modeling and Machine Learning

While Alteryx doubtlessly makes life easier for data engineers and data scientists alike, its predictive tools are not as extensive or a customizable as the functionality offered in KNIME Analytics Platform. In both environments you can connect to external tools to train your models, e.g. there are many R and Python libraries you can connect to, however, here we want to look at what can be done natively.

Learners, Predictors, and Scorers

In KNIME, building models mostly happens in the same framework regardless of the type of model you want to build. You'll start with some data, partition it into training and testing subsets, apply a Learner node, a Predictor node, and finally a Scorer node to view some statistics.

Now, of course, to generate a successful model for deployment, you'll want to make sure you've cleaned up your data and completed any feature engineering you might want to do first, but this is how training a model will look in KNIME. Pretty straightforward right?

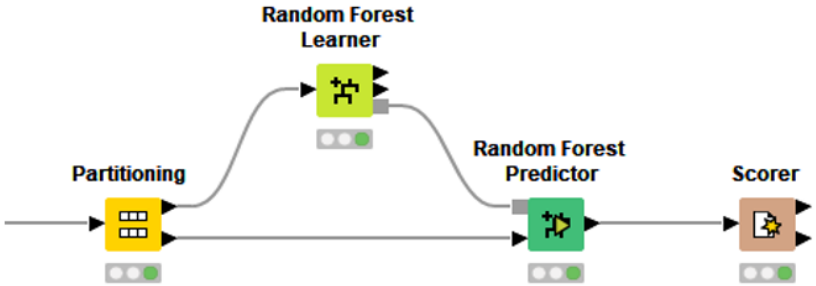


Figure 25 Part of a KNIME workflow in which a model is built using the Learner, Predictor, and Scorer nodes

Trees

Tree based models are incredibly versatile and come in many varieties. Some rivaling the predictive power of deep neural networks while requiring a fraction of the data and training time. These models aren't to be overlooked and KNIME supports the training and deployment of many.

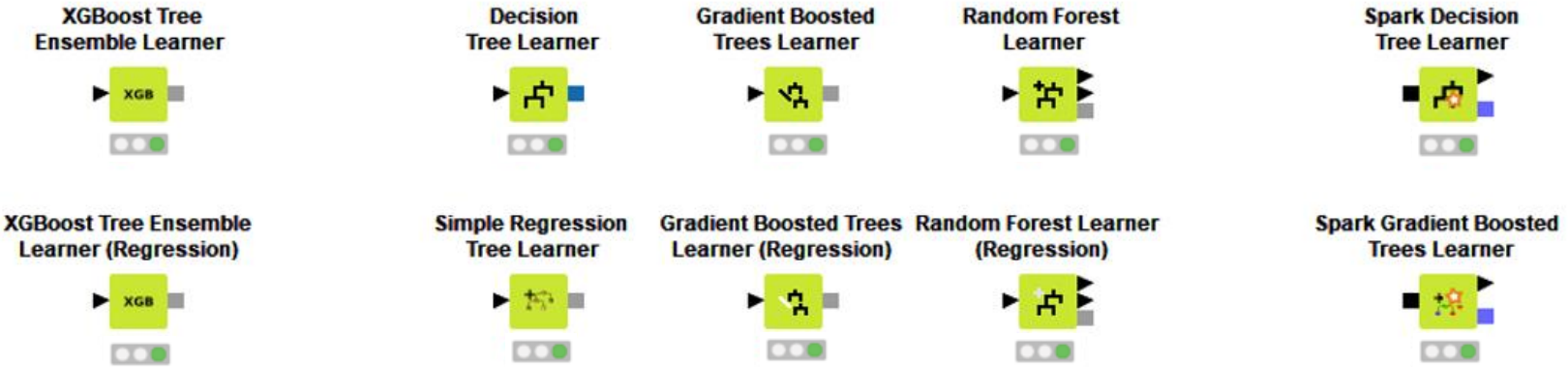


Figure 26 Nodes that support that training and deployment of tree-based models in KNIME

Both regression and classifications trees are supported as well as their ensembles such as random or boosted forests. In KNIME you can use KNIME specific implementations of these algorithms as well as those from several other popular open source tools, such as H20, XGBoost, and Spark. The customizations on these models is also quite robust with the ability to customize minimal tree node sizes, maximal tree depth, and more. The two primarily learning methods supported are Information Gain, and Gini Index.

Regressions

Regressions on aren't new by a long shot but are still amazing tools for modeling numeric data, or even for classification problems with the application of logistic regressions. KNIME supports many types, from linear to polynomial through to logistic and even trees and forests. All of these can be implemented right in KNIME Analytics Platform, but some can be deployed to Spark to take advantage of parallel computing. As with the tree-based algorithms in the prior section you also have access to H20 and XGBoost implementations for the algorithms in addition to the native KNIME nodes.

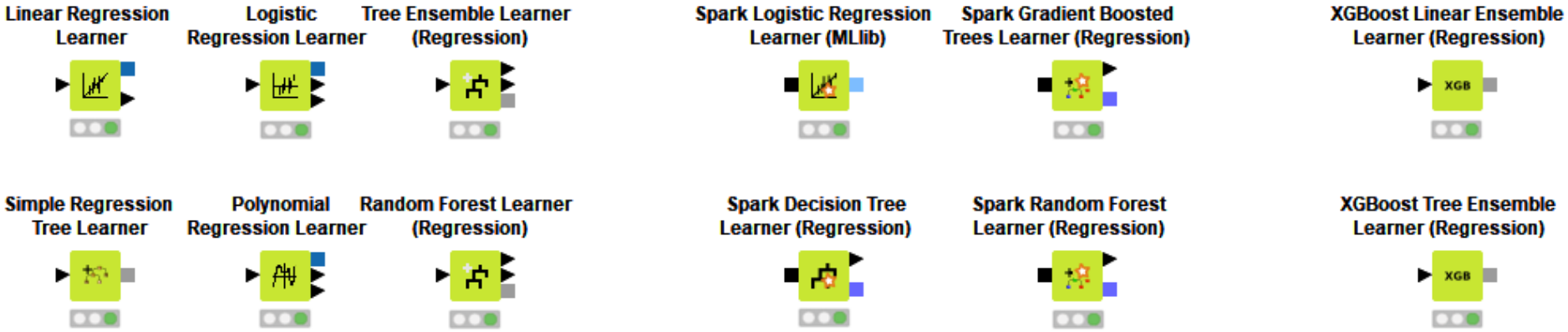


Figure 27 A veritable battalion of regression nodes in KNIME

Clustering

Clustering is an example of an unsupervised technique. This means you can use it without any prelabeled or classified data, as you would normally require for training decision trees, regressions, or neural networks. Clustering means that your data points are grouped according to some distance metric. There are many available in KNIME - from a simple Cartesian distance, something you may use to cluster gps locations on a map - to the Levenshtein distance for measuring the number of characters you would need to change to make two strings match. The latter is sometimes useful in automated correction of typos.

Let's look at how to set up hierarchical clustering. Unlike other techniques where you use a learner and a predictor, we'll require three steps here. First, we need to calculate distances using a distance node, note that there's a separate node for string and numeric distances: pick whichever suits your data. Second, we'll use those distances in the Hierarchical Clustering (DistMatrix) node to create the cluster tree. Then finally, the Hierarchical Cluster Assigner node assigns the actual cluster values to each row based on either a number of clusters or a maximum distance, which you can set in the configuration dialog, as shown in

Figure 28.

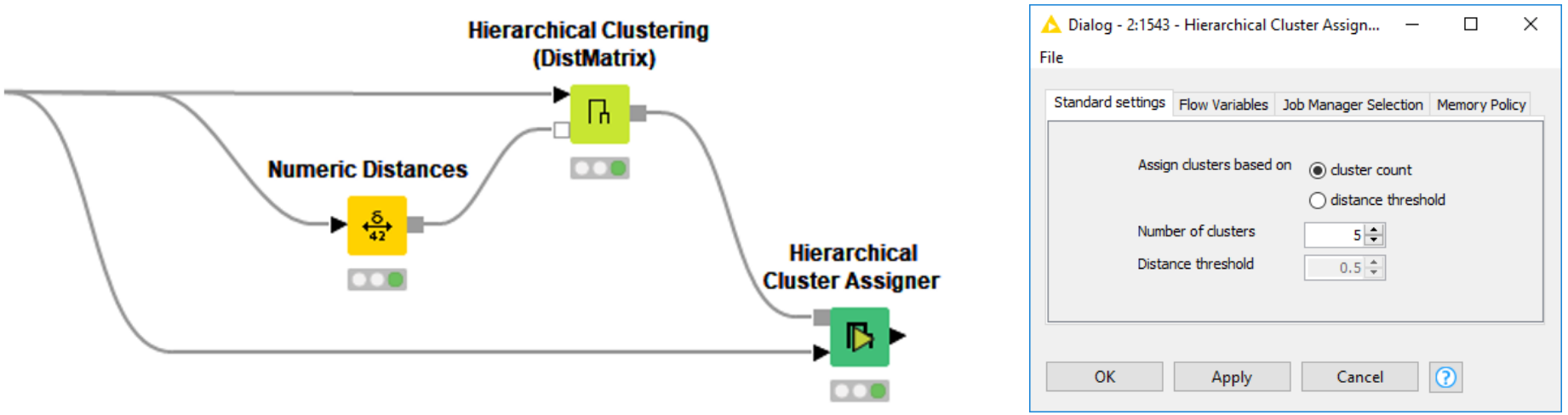


Figure 28 Hierarchical Clustering Example workflow and Cluster Assigner Configuration Dialog

Neural Networks

Neural networks in KNIME have undergone vast improvement. On the technical side, the deep learning extension for KNIME requires set-up of the Python Integration; it most notably implements Keras with a TensorFlow backend. Once you're set up you won't have to worry about all that too much. The Keras implementation works by using one node per network layer (there are also nodes for repeating and permuting the layers you want to include multiple times). The advantage of this is two-fold, you can highly customize each layer of your networks by selecting the layer type, number of nodes, and more - depending on the layer, and it gives you a nice visualization of the network you've assembled. Beyond powerful customization options for building your network, KNIME can also load pretrained models, append new layers to those models, and freeze layers to prevent them from being trained in the network trainer. Together, these nodes make applying transfer learning techniques in KNIME amazingly easy.

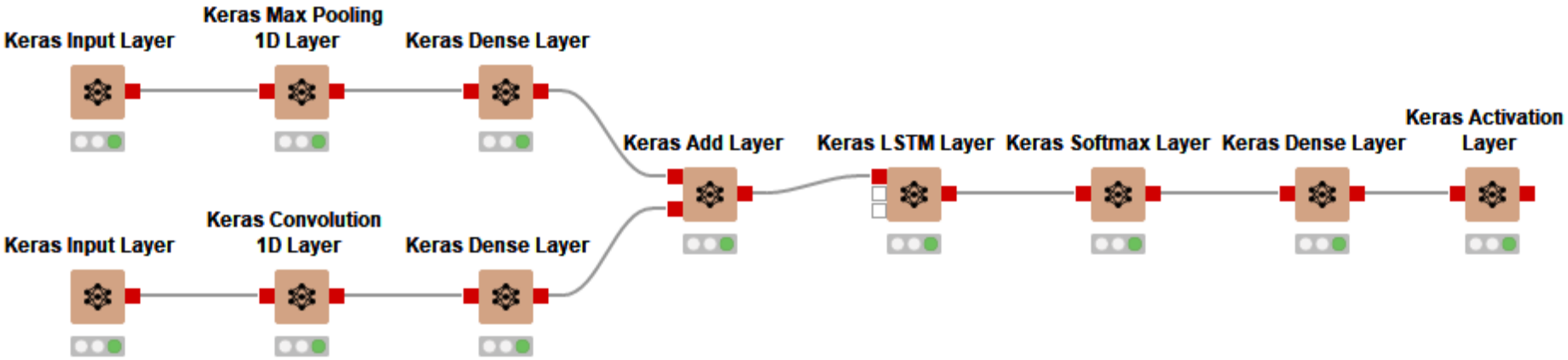


Figure 29 Keras implementation in KNIME Analytics Platform

Evaluation

Evaluating the success of your models is another important stage of the data science process. You'll want to understand how well your models perform in different scenarios and where they fail to properly pick the best model for deployment. Sometimes this means reviewing a confusion matrix with a **Scorer** node for statistics like accuracy, precision, and recall. Alternatively, visual evaluations might be best, such as **ROC Curves** or **Lift Charts**, when you need to present on your findings. Since version 4.0, KNIME Analytics Platform also supports several model interpretability tools such as LIME and Shapley. You can use these to better understand what is influencing the outputs of some of the "black box" models, this can be helpful if the goal is to better understand underlying causes. They're also great as dummy check to verify your model isn't just picking up on some incorrect correlation.

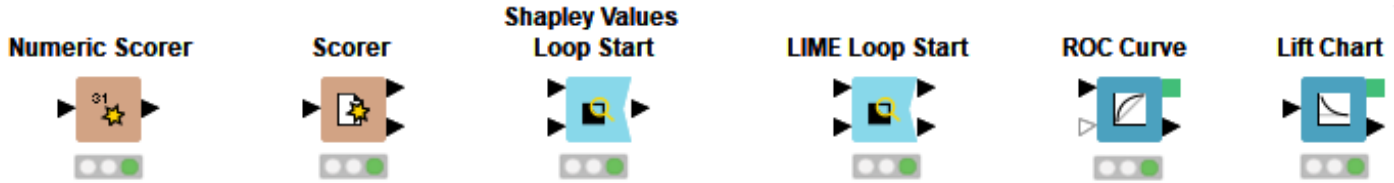


Figure 30 KNIME nodes for evaluating the success of your models, including the model interpretability tools, such as LIME and Shapley

Optimization

Sometimes building a model is as simple as loading in a dataset, choosing a learning technique, and clicking go. But then again, often enough it's not! Perhaps your random forest works amazingly with a max depth of five, but no good at all with seven. Perhaps your neural network is successful with five hidden layers, but not fifteen. There's a lot of work that goes into fine tuning your models and squeezing out every single ounce of predictive power you can. KNIME can help with this via its different functionality for feature and hyper-parameter tuning. KNIME supports feature selection techniques from correlation filters for removing highly correlated variables, to low variance filters for removing near constant variables, to forward and backward feature selection. There is also a loop for the tuning of hyperparameters, those being the features of your model, such as the number of hidden layers in a neural network, max depth of a node in a tree, or number.

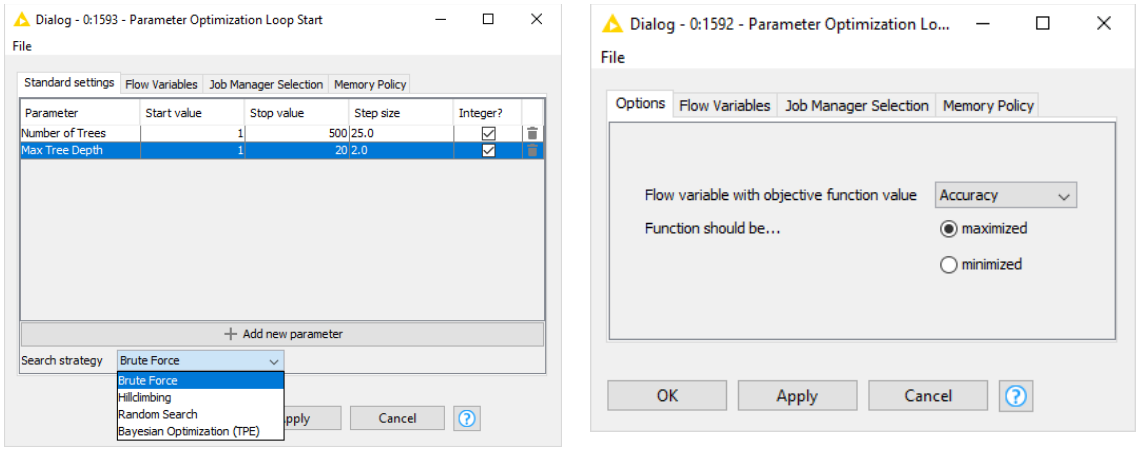


Figure 31 Parameter Optimization Loop Start Configuration Dialog (left) and Parameter Optimization Loop End Configuration Dialog (right)

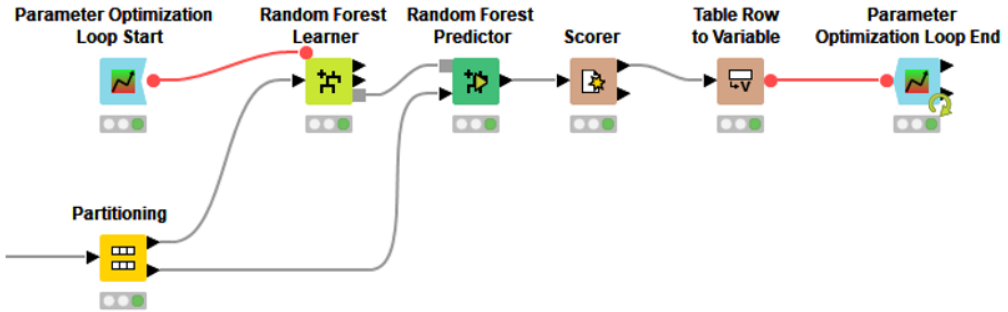


Figure 32 Parameter Operation example workflow, retrains Random Forest model with different numbers of trees and different maximum tree depths using a brute force method. Maximizes on accuracy.

Workflow Control

In this section we'll break our pattern and stop looking at directly related functionality and which Alteryx tools are synonymous with which nodes in KNIME. First, we will look at some options for workflow control in Alteryx and alternatives in KNIME so as to reassure you that the functionalities are available then we'll dive into Loops, Flow Variables, and the KNIME Web Portal. The goal here is to familiarize you with the abilities of these features as opposed to providing a complete guide to their use. Check out KNIME TV on YouTube or any of the content on the eLearning section of the KNIME website for more information on how to use these features.

KNIME Data Apps (Analytic Apps)

In Alteryx' words: "An analytic app is a workflow with a user interface. Create an analytic app to enable the app user to execute a workflow using their own data and parameter without having to build the workflow."

KNIME has very similar functionality through the **KNIME WebPortal**. The WebPortal is part of KNIME server and is accessed through a web browser. Building these WebPortal enabled Data Apps is easy and I'll summarize the steps, but first let's talk about **Widgets** and **Components**.

In Figure 33 you'll see one page of a Data App example called Guided Visualization; this is available on the KNIME Hub for you to try out. The view you see is a single component, and, as you continue to move through the WebPortal each new page is based on a new component in your workflow. In this way, the Data App user can move through your workflow at specific interaction points you have defined. This example allows a user to upload a data file of their choice and then create custom visualizations taken from the WebPortal. This is perfect for speeding up presentation design for a marketing or sales team!

These Data Apps are built in just the same way as you would build any workflow in KNIME Analytics Platform and then are deployed to KNIME Server for use.

Let's look at components next, to get a bit of an understanding of how these WebPortal pages are assembled.

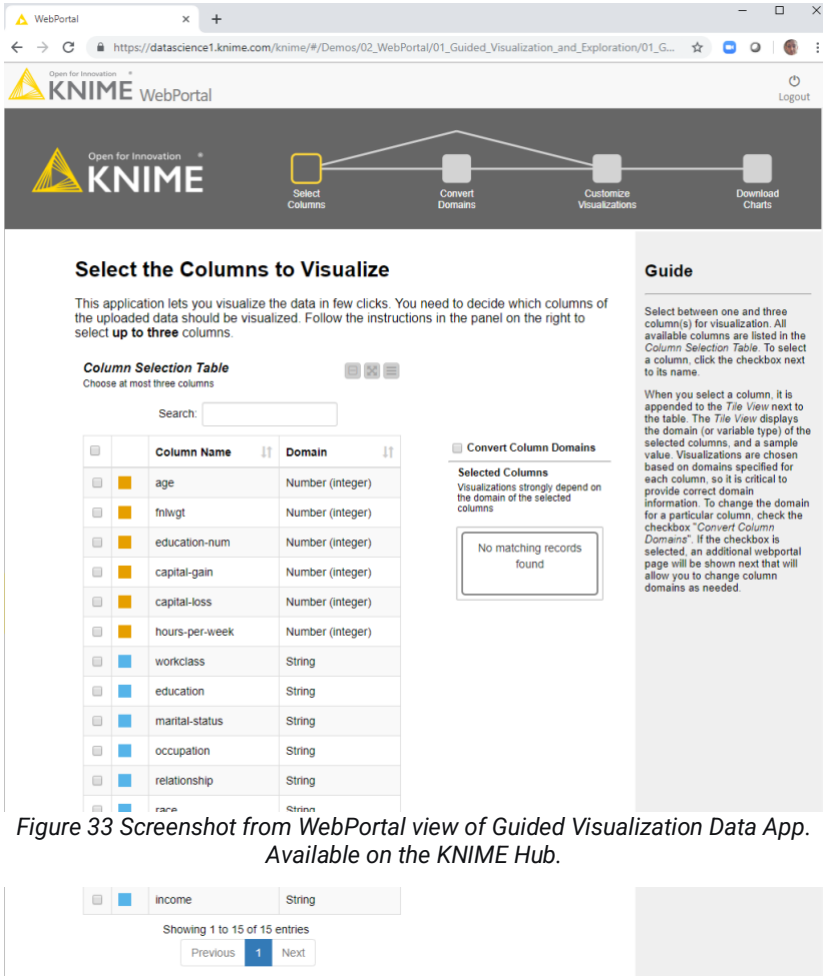


Figure 33 Screenshot from WebPortal view of Guided Visualization Data App. Available on the KNIME Hub.

Components (Macros)

Macros in Alteryx are your way to create what are, in effect, custom tools. You'll build a section of your workflow using special tools, which allow for interaction, and then wrap them up into a macro that can be used in another workflow.

Usually, you'll want to do this for common tasks that only change slightly each time. In KNIME we call macros 'components'.

These components can be saved either locally or to a Server for repeated use and sharing. Do this by right clicking on your component, expanding the component line, and then choosing the **Share** option.

To create a component intended for use **locally** the only major addition is a **Configuration** node, found in the **Workflow Abstraction** folder. These configuration nodes behave a lot like some of their widget counterparts with one exception. Instead of displaying in the WebPortal they display in the configuration dialog when you right click the finished component. Behaving just like a regular KNIME node. See Figure 34 bellow for an example.

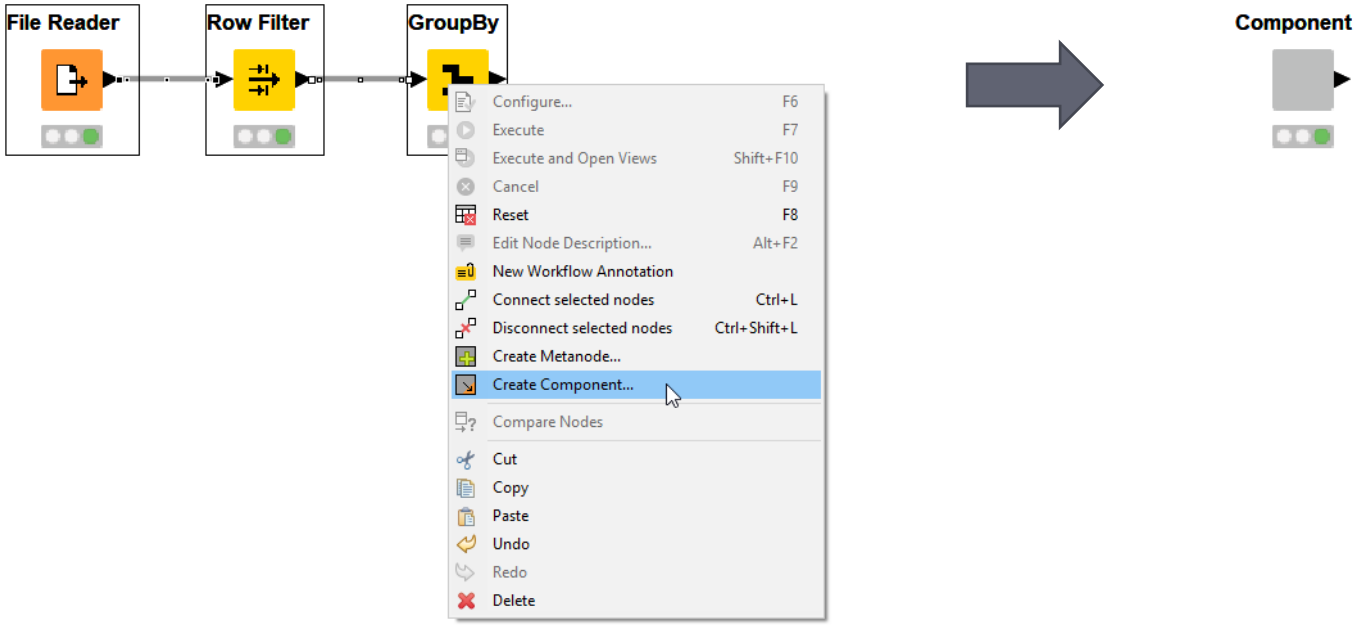


Figure 34 Right click and select Create Component… to condense a set of nodes into a Component.

Configuration nodes:

Directly above the widgets in the Node Repository you'll see the Configuration folder. Where **Widgets** are used to create interactive views for your component or WebPortal, **Configuration** nodes allow you to create configuration windows for the Components. The **Fast Fourier Transform (FFT)** component in Figure 35 has its configuration window next to it, this is the result of putting a **Column Selection Configuration**, **Single Selection Configuration,** and **Double Configuration** node inside the component for the user to interact with

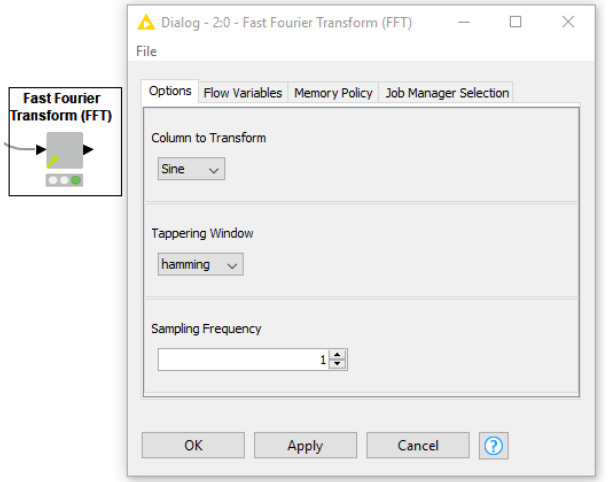


Figure 35 Fast Fourier Transform (FFT) component and its configurations

Widget nodes:

Widgets can be found under Workflow Abstraction > Widgets in the node repository. They come in a few different categories represented by the subfolders you see to the right: input, selection, filter, and output. Input enables users to provide Flow Variables for use in the workflow, this could be in the form of a dropdown selection for strings, a slider for numeric values, a calendar selector for Data & Time, etc. Input also contains the file upload node to allow the user to supply their own data. Selection allows the user to set things such as filter parameters, or column(s) for processing. Filter includes more interactive options for filtering, these can be tied to graphical displays for dynamic views. Finally, output allows for end outputs such as downloadable files, images, or text.

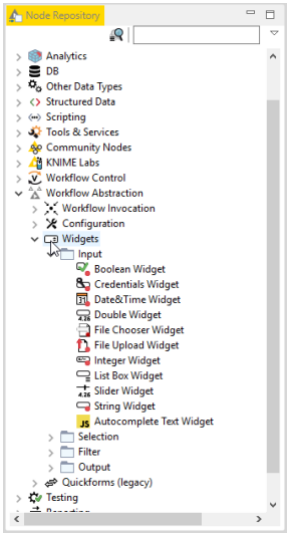


Figure 36 Widget location in Node Repository

Loops

Loops are sections of your workflow that will execute multiple times. This could be a set number of times, until every row in a table in processed, until some variable condition is met, or even indefinitely. A loop is built in KNIME by combining a loop start and loop end node, both of which come in several types. The nodes placed between this start and end nodes will be executed until your defined conditions are met.

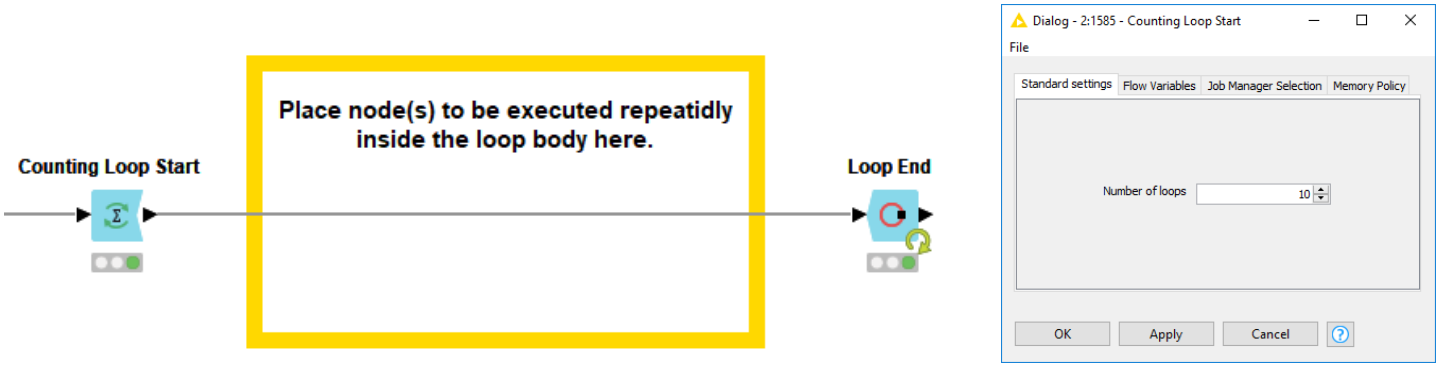


Figure 37 Counting Loop example and configuration window of Count Loop Start node

The image above using the **Counting Loop Start** node, this loop start variant simply loops a given number off time as you can set in its configuration window. Let's look at a couple other types of loops in KNIME to get you more familiar with what's possible. These are only three types, there are several more you can explore as well.

The **Group Loop** works a lot like the **Group By** node, which we looked at earlier in this booklet. You select a set of columns to use to group your data, but instead of setting a manual aggregation method for the data in that group you gain access to the groups one by one as you iterate through the **Group**

The **Recursive Loop** is special in that it is the only type of loop that can pass data back to the start to be used in the next iteration. It must be paired with a **Recursive Loop End** node where you'll declare what gets sent back to the next iteration.

The **Table Row to Variable Loop** doesn't supply data to each iteration like the others. It iterates over each row of the table providing the values inside that row as flow variables. A popular use of this node is to combine it with the **List Files** node and the **Excel Reader** node to easily read and concatenate an entire directory of files.

Flow Variables

Flow variables are used in KNIME to parametrize workflows when Node Settings need to be determined dynamically. Flow variables are carried along branches in a workflow via data links (black lines between nodes) and via explicit variable links (red lines between nodes). Flow variable ports can be enabled on any node via Right Click > Show Variable Ports.

Two of the most important applications for flow variables are the configuration of Loops and Components using Configuration Nodes. Let's look at a basic example and then an example using loops.

Example 1, Intro:

In the example to the right a **Flow Variable** is initially created by the **String Configuration** node. This node simply allows the user to type a string that will become the variable. In this workflow that string will represent the name of the excel file to be written. That flow variable is then passed into the **Create File Name** node which takes a string and a directory and creates a file name URL, we'll pass that into the **Excel Writer (XLS)** and use it to automatically set the location and name of the file we'll write. In the configuration window click on the **V** next to the output location to control it with a variable, then select the variable created, FilePath in this case, and it will dynamically control that setting. Perfect!

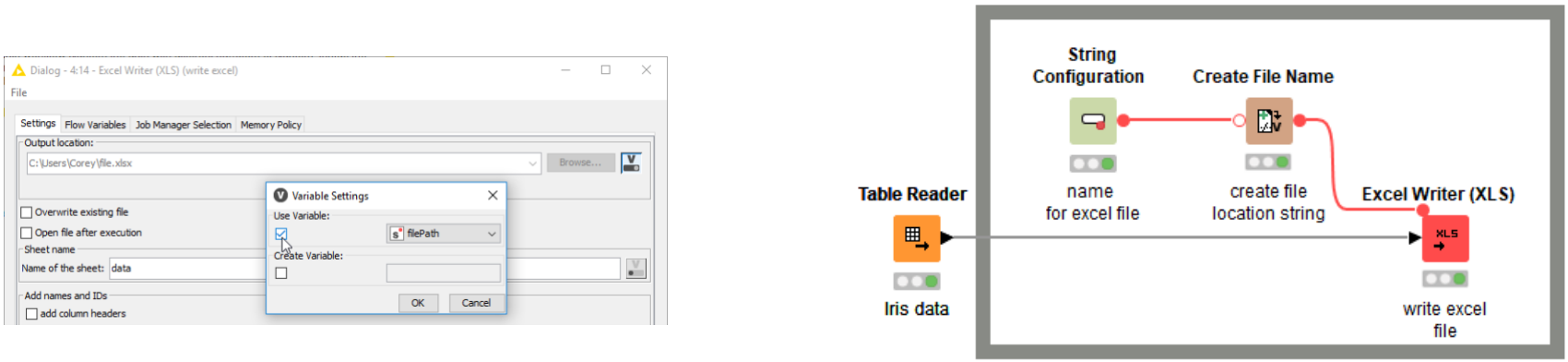


Figure 38 Example workflow using a flow variable to name an Excel file to be written

Example 2, Loops:

This example is available on the EXAMPLE server automatically available in KNIME Analytics Platform. Find it under Control Structures > 04_Loops > 01_Loop_over_a_set_of_parameter_for_k_means.

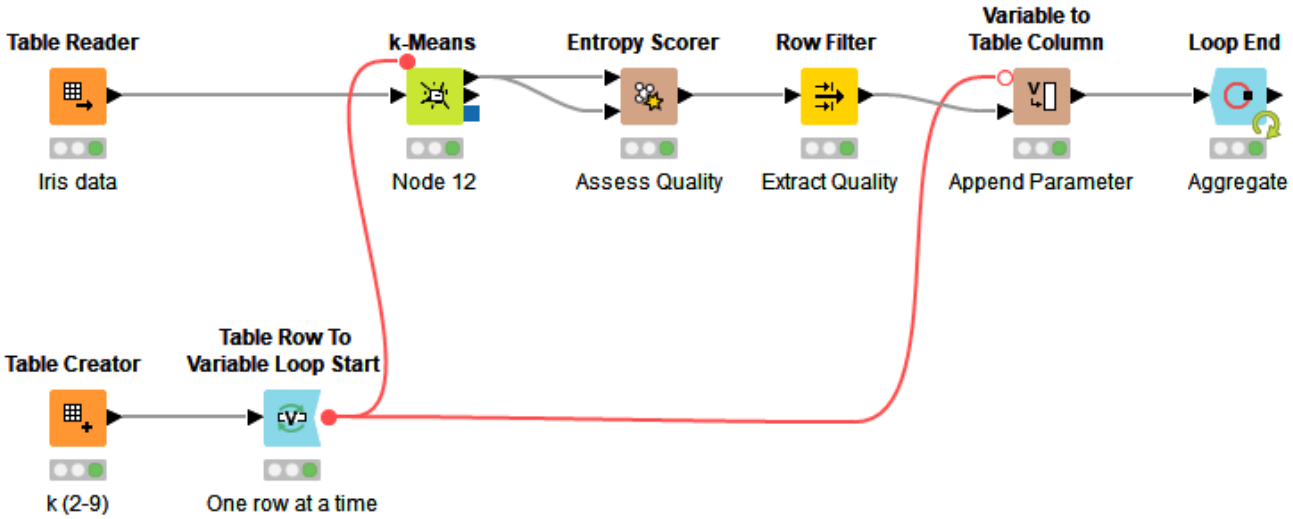


Figure 39 Example workflow using flow variables to control the number of clusters in a k-Means clustering node

You see in this workflow that instead of creating a **Flow Variable** with a configuration node manually the red variable line starts with a **Table Row to Variable Loop Start** node. We briefly touched on this node in the loop section as well but basically what it will do is convert each column in a table to a variable and iterate through them one row at a time allowing you to build a workflow that performs many similar tasks quickly and easily. In this case we're passing a variable into the **K-Means** clustering node to change how many clusters it creates and collecting that information, along with some info from the **Entropy Scorer** node at the end of the Loop to help us decide how to cluster our data.

Appendix

Available Data Types

Data Type	Alteryx	KNIME	Notes
Bool	X	X	
INT	X	X	
Decimal	X	X	Both having multiple options for precision
Complex Number		X	
String	X	X	Alteryx having multiple options for storage efficiency
Nominal		X	
Data and/or Time	X	X	Dates, Times, or Date / Times
Spatial Objects	X		Points, Lines, and Polygons
Network / Graph	X	X	
Audio		X	.wav format
Image		X	
Document		X	Includes text and meta data for text mining
Collection		X	List of values in single table cell

Quick Tool to Node Reference

Alteryx Tool	KNIME Node	Alternate Node	Alternate Node	Alternate Node
Browse	Data Explorer	Statistics		
Input Data	File Reader	Database Reader	Tika Parser	Hive/Impala Connectors
Output Data	Excel Writer	Table Writer	Database Writer	Hive/Impala Loaders
Text Input	Create Table	String Input	Numeric Input	
Data Cleansing	Missing Values	String Manipulation		
Filter	Row Filter	Rule Based Row Filter	Row Splitter	
Formula	String Manipulation	Math Formula	Rule Engine	Column Expression
Sample	Row Sampling	Bootstrap Sampling	Equal Size Sampling	Partitioning
Select	Column Filter	Column Rename	Column Sorter	Type to Type nodes
Sort	Sorter			
Join	Joiner	Cross Joiner		
Union	Concatenate	Concatenate (optional in)		
Text to Columns	Cell Splitter			
Summarize	Group by			
Tile	Auto-Binner	Numeric Binner	Binner (Dictionary)	CAIM Binner
Imputation	Missing Value	Rule Engine		
Render	PDF Writer	Image Port Writer		

Useful Links

FAQ

A collection of some of our most commonly asked questions, check out the forum if your answer isn't here!

https://www.knime.com/faq

KNIME Hub

The perfect place to search for nodes or example workflows when you're not quite sure what you need yet.

https://hub.knime.com/

Forum

Come here to engage in community discussion, submit feature requests, ask for help, or help others yourself!

https://forum.knime.com/

Blogs

A collection of blog posts covering data science with KNIME, a great space to learn what KNIME can really do.

https://www.knime.com/blog

Learning Hub

A central spot to access education material to get you started with KNIME

https://www.knime.com/learning-hub

KNIME TV

Our very own YouTube channel with everything from community news, to webinars, to mini lessons.

https://www.youtube.com/user/KNIMETV

KNIME Press

Information on all our available books, like this one!

https://www.knime.com/knimepress

Events and Courses

Information on all our upcoming events including courses, webinars, learnathons, and summits.

https://www.knime.com/learning/events