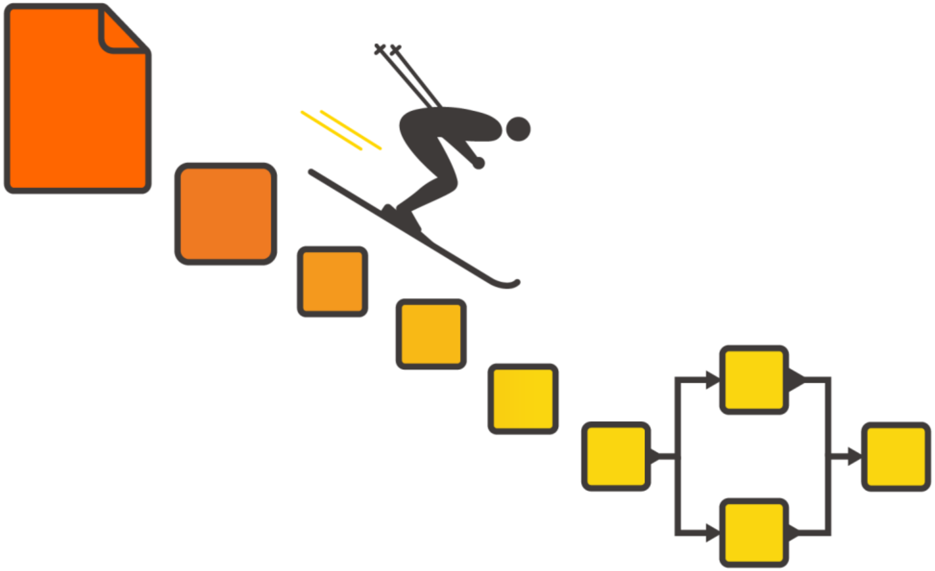
KNIME Transition Booklets

COREY **WEISINGER**

From Alteryx to KNIME

KNIME v4.7



Open for Innovation
KNIME

This book has been updated for **KNIME 4.7**.

For information regarding permissions and sales, write to:

KNIME Press
Talacker 50
8001 Zurich
Switzerland
knimepress@knime.com

www.knime.com

Preface

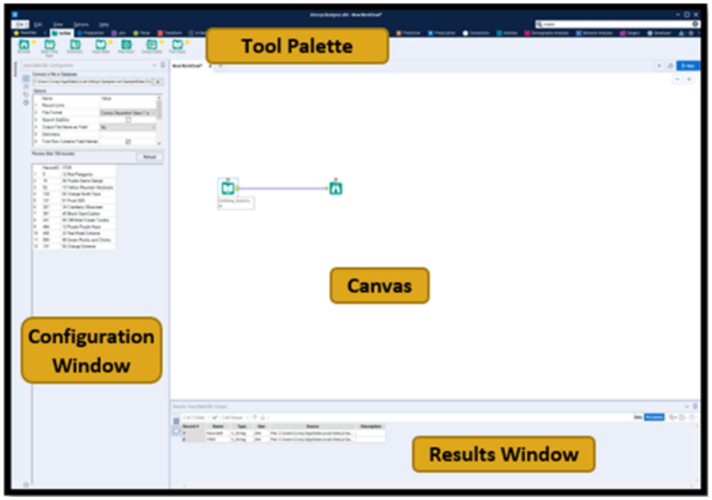
KNIME Analytics Platform is a powerful tool for data analytics and data visualization. It provides a complete environment for data analysis which is fairly simple and intuitive to use. This, coupled with the fact that KNIME Analytics Platform is open source, has led a large number of professionals to 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.

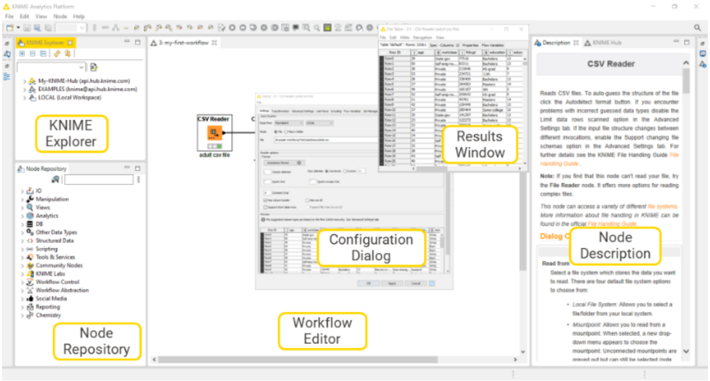
Table of Contents

KNIME Analytics Platform Interface

Alteryx



KNIME Analytics Platform



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.

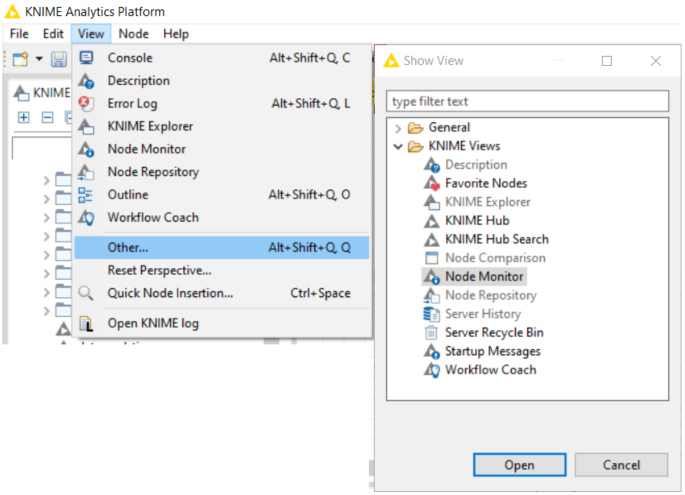
Results Window

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.

Node Description

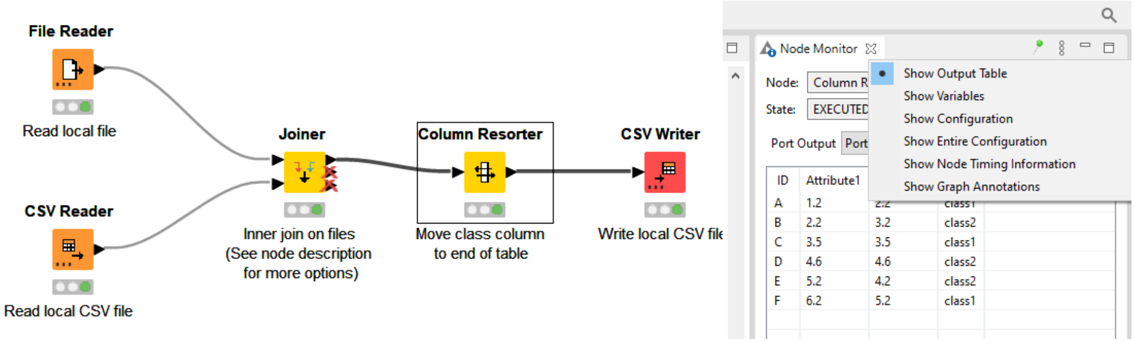
This window provides a detailed description of what the selected node does. It includes an overview, information on the configuration options, and details on what each input and output port. A node’s port is the equivalent of a tool’s anchor in Alteryx.

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!



Node: Column R

State: EXECUTED

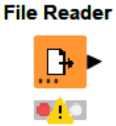
Port Output Port

ID	Attribute1		
A	1.2	2.2	class1
B	2.2	3.2	class2
C	3.5	3.5	class1
D	4.6	4.6	class2
E	5.2	4.2	class2
F	6.2	5.2	class1

Node Interface

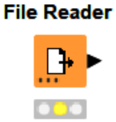
In KNIME Analytics Platform, you build your workflow by dragging nodes from the *Node Repository* to the *Workflow Editor*, then connecting, configuring, and executing them. Like the tools in Alteryx, there are many nodes for many different tasks. The node's name is written above the node. Below the 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. The input port represents the data being fed into a node for processing and the output port eh 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 represented by a brown square instead. The node shown below, the *File Reader* node, only has a port on the right, the output port. This is because no data is input into a *File Reader* node.

Traffic Lights Identifying the Status of a Node



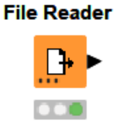
Unconfigured node:

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.

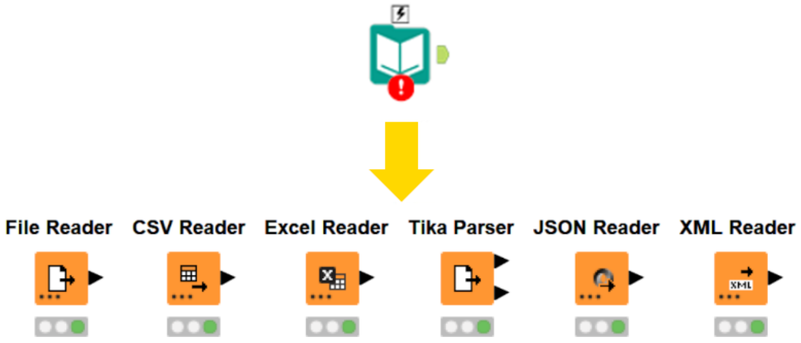
Importing Data

In Alteryx, all your data importing is done through varies configurations of the *Input Data* tool. In KNIME Analytics Platform, 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

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



File Reader node:

The *File Reader* node can read just about any ANSCII data. It automatically detects common formats.

CSV Reader node:

Although CSV files can be read by the *File Reader* node, the *CSV Reader* node gives you more specific options.

Excel Reader node:

The *File Reader* node can also handle Excel files, but the *Excel Reader* node lets you read specific sheets, rows, or columns.

Tika Parser node:

This node uses the *Apache Tika* library and can read a lot of data types! Try it with emails or PDFs.

JSON Reader node:

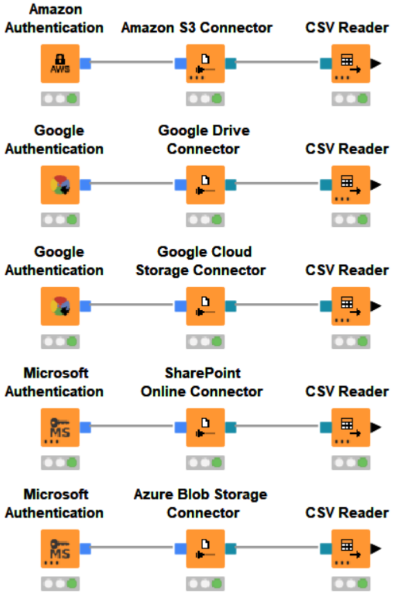
This node, as the name suggests, is for reading JSON files. KNIME can also convert these into tables.

XML Reader node:

This node is for reading XML files. Optionally, an XPath Query can be specified in the configuration window.

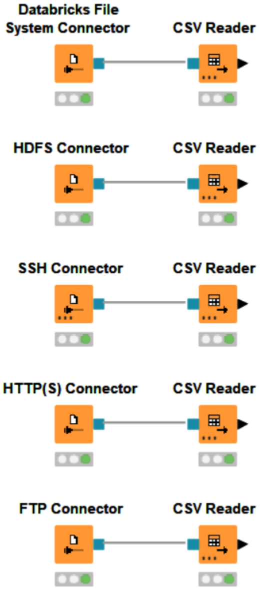
Remote File Connections

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!



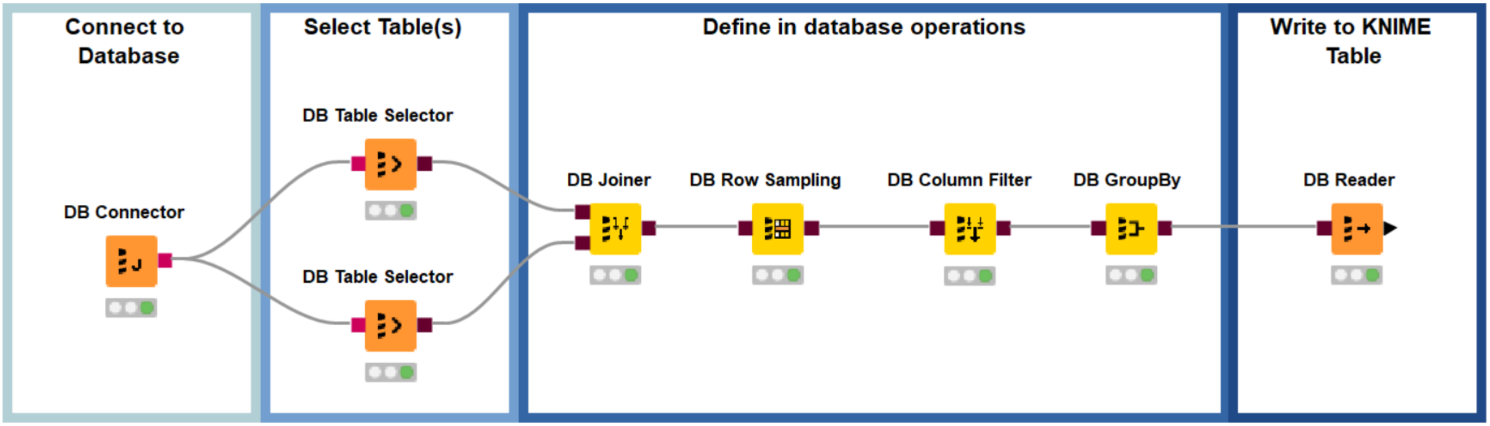
In the figure on the left. some of these remote connections are displayed. They require two nodes to be connected before the actual reader node. One for authentication, where you log into your system, and a second for establishing the connection. These are typically systems we have multiple service integrations with.

Other "simpler" remote systems may only require one connector node (see figure on the right) 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 righthand example.

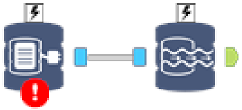


Databases

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.

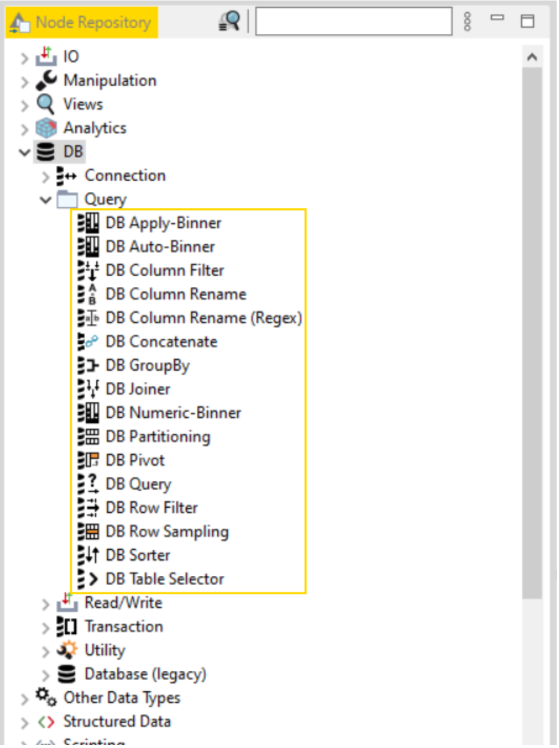


In Alteryx this is like using the *Connect In-DB* tool and *Data Stream Out* tool.



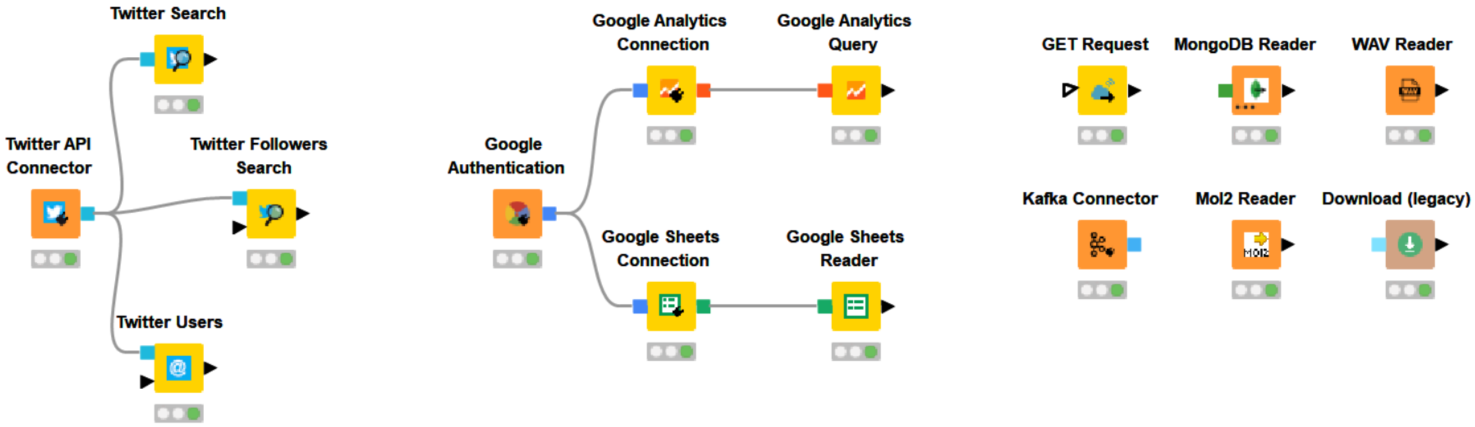
So how do we connect to the database in KNIME? This is done with the *Database Connector* node, be it a traditional format like *MySQL*, or a *Hadoop* based one like *Impala* or *Hive*. Once that connection is established, we can select a table in the *DB Table Selector* node. The *DB Connector* node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

Next, you can see the *Query* folder, opened in the *Node Repository* (see figure on the right), 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, like 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 Analytics Platform. All the nodes in the *Query* folder can be inserted into a workflow between a *DB Table Selector* node and a *DB Reader* node. KNIME is using 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.



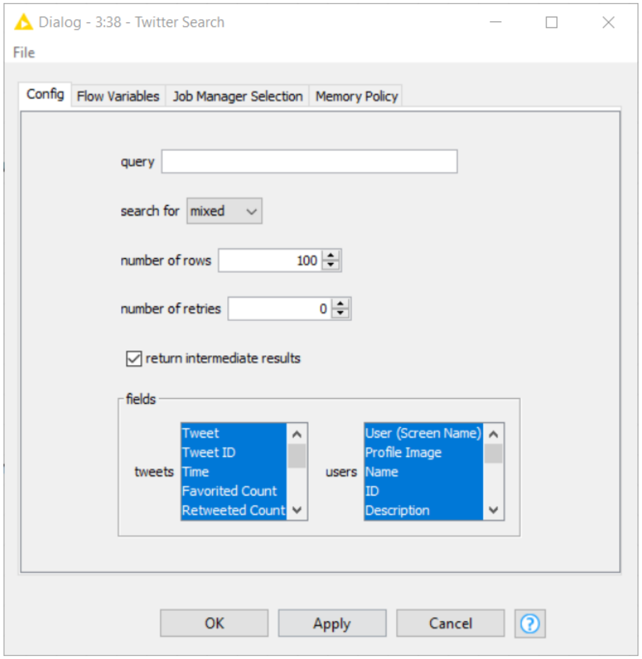
Other Sources

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* node 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!



For brevity, we'll just look at two of these nodes here. Feel free to explore other nodes yourself.

Note. The node description in the *KNIME Workbench* or on the *KNIME Community Hub* tells you everything you need to know about how to configure the nodes. The *Get Request* and *Download (legacy)* nodes, for example, are particularly great resources!



Dialog - 3:38 - Twitter Search

query

search for: mixed

number of rows: 100

number of retries: 0

☑ return intermediate results

fields

tweets: Tweet, Tweet ID, Time, Favorited Count, Retweeted Count

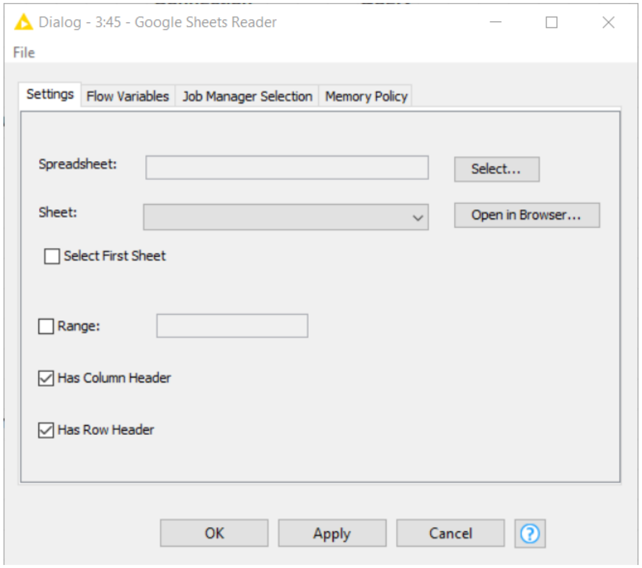
users: User (Screen Name), Profile Image, Name, ID, Description

Twitter Search node:

Once you've connected to the Twitter API with the connector configuration, your search is easy. At the top, type your query as you would on twitter.com/search. At the bottom select the fields you'd like to get returned (see figure on the left).

Google Sheets Reader node:

In this node's configuration window, all you must do is specify which spreadsheet you want data from and which sheet on it. A list of files you have access to will appear when you click the select button (see figure on the right).



Dialog - 3:45 - Google Sheets Reader

Spreadsheet:

Sheet:

□ Select First Sheet

□ Range:

☑ Has Column Header

☑ Has Row Header

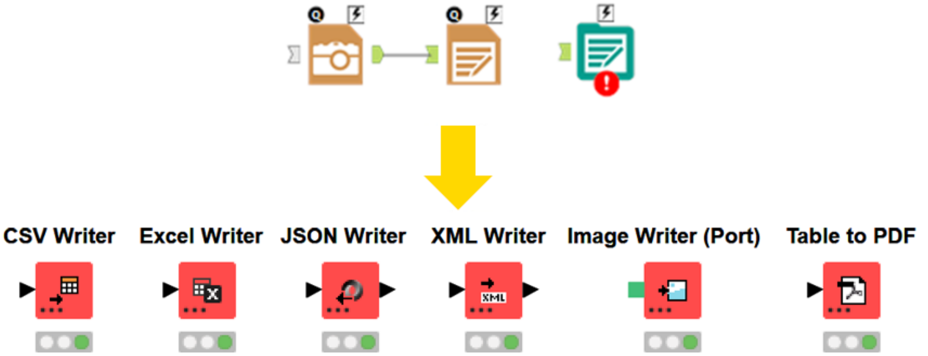
Writing Data

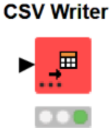
So far, we've had a look at the interface – the *KNIME Workbench*; and how to get data into KNIME. Now, let's have a look at getting data out of KNIME. Below you'll first find a list of the nodes for writing local files and, second, a section that looks at getting your data into databases. After we've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

Note. The remote connections also work while writing data! Simply click the 3-dot icon and select "Add File System Connection port". Then you'll be free to connect to whichever remote connection you need! See page 6 for more information on remote connections.

Local Files

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 would 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.





CSV Writer node:

This node writes to a CSV file, allowing for delimiter, missing value pattern, quote configuration, and more.

Excel Writer node:

This node allows you to quickly export to XLS files. For advanced options we'll look at more nodes on the next page.

JSON Writer node:

Write values to a JSON file with this node. Optionally, you can specify to automatically compress the output.

Image Writer (Port) node:

Some graphic nodes output images. Connect them to this node if you want to export them.

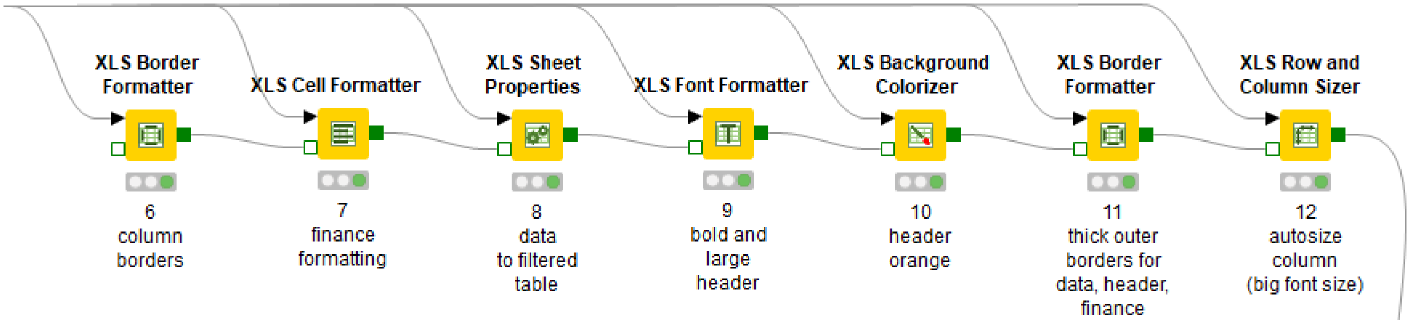
XML Writer node:

Table cells can hold an XML data type. This node writes those cells out as separate files.

Table to PDF node:

This node creates a PDF version of your data table. Combine with graphs to include a snapshot of the actual data.

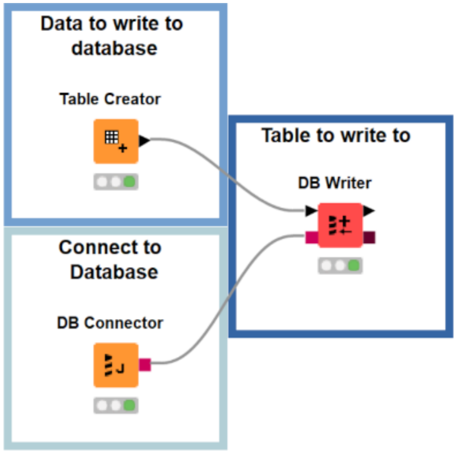
Below, you see a string of XLS Formatter nodes. These are linked together and each node change bits of the formatting in an Excel file you're preparing to write. The modular nature makes it easy to customize your formatting as much as you like and allows to add or remove parts. There is a variety of 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 Analytics Platform 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.

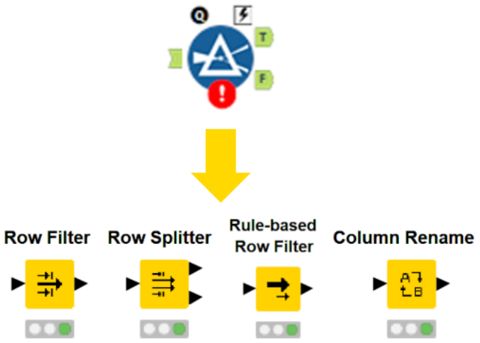


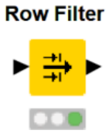
- ***File Reader node/Table Creator node:*** The data you wish to append to your database, this could also be processed by your workflow before writing.
- ***DB Connector node:*** This supplies the information for connecting to the database, e.g., login credentials.
- ***DB Writer node:*** This is where 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

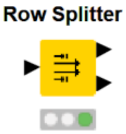
Row filtering in KNIME is done with a few different nodes: the *Row Filter* node, the *Rule-Based Row Filter* node, and the *Row Splitter* node, which is for collecting unmatched rows. For column filtering, the *Column Filter* node is your main stop! In Alteryx the *Select* tool has several purposes,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





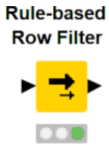
Row Filter node:

Allows for quick filtering by means 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.



Row Splitter node:

This node works just like the *Row Filter* node 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.



Rule-based Row Filter node:

The *Rule-based Row Filter* node is similar to the expression builder capabilities in the *Select* node in SPSS Modeler. You enter a set of rules, which are run through one by one. The first rule to match the row ends the process, e.g.:

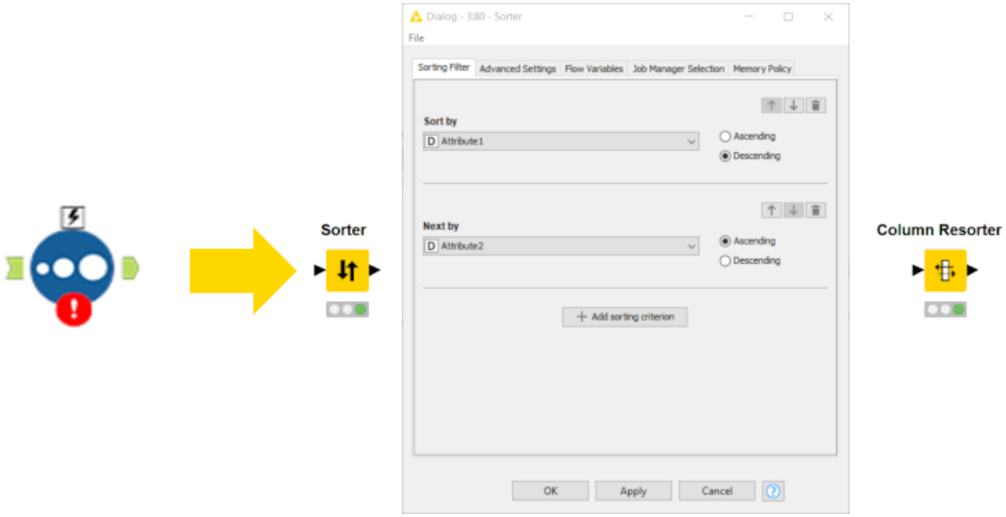
```
$Score$ > 0.9 => TRUE

$Name$ = “Jon Smith” => TRUE
```

… returns all scores over 90%, and also, all Jon Smiths.

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 in the *Advanced Settings* tab of the configuration dialog.



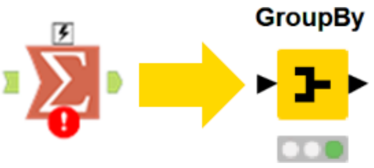
Sort by
D Attribute1
○ Ascending
● Descending

Next by
D Attribute2
● Ascending
○ Descending

Column Resorter node:

You can also sort columns in KNIME Analytics Platform 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.

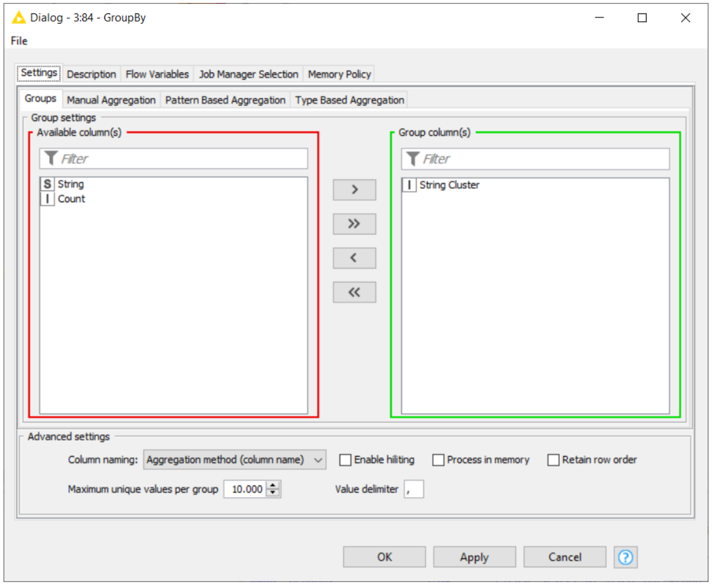
Aggregating Data



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

In this case, the configuration window looks quite a bit different in KNIME. First, in the *Groups* tab of the configuration dialog (left), select which columns will define the groupings. I've used the *String Cluster* column here. Once selected, the grouping column(s) will appear in the green box to the right.

Next, in the *Manual Aggregation* tab (right), choose which of the remaining columns to aggregate and to include in the output. Finally, set which type of aggregation to use. There are many options, from concatenations to average to skewness and many more.



Group settings

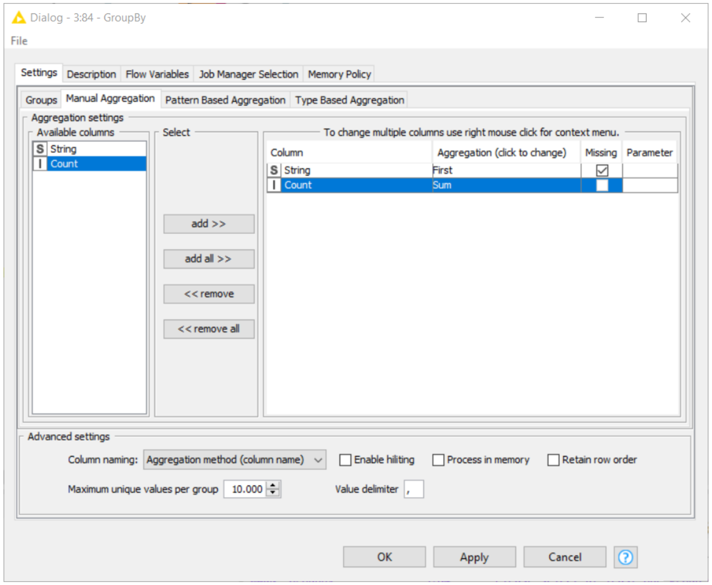
Available column(s): S String, I Count

Group column(s): I String Cluster

Advanced settings

Column naming: Aggregation method (column name)　☐ Enable hiliting　☐ Process in memory　☐ Retain row order

Maximum unique values per group 10.000　Value delimiter ,



Aggregation settings

Available columns: S String, I Count

To change multiple columns use right mouse click for context menu.

Column	Aggregation (click to change)	Missing	Parameter
S String	First	☑	
I Count	Sum	☐	

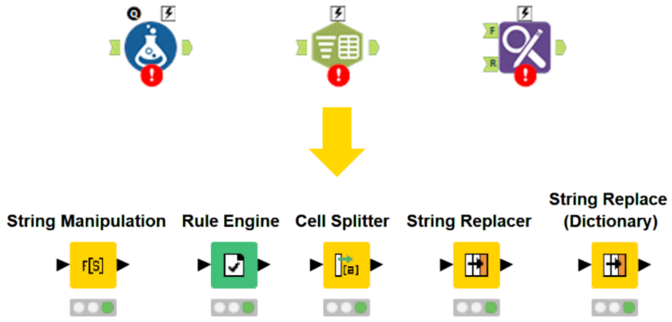
Advanced settings

Column naming: Aggregation method (column name)　☐ Enable hiliting　☐ Process in memory　☐ Retain row order

Maximum unique values per group 10.000　Value delimiter ,

String Data

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 used 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.

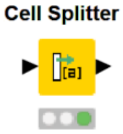


String Manipulation node:

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

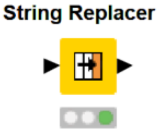
Rule Engine node:

The *Rule Engine* node has some of the same functions as the *String Manipulation* node but allows for more control by taking a list of user-defined rules and matches them to each input row. For example, you can reformat strings differently depending on their source.



Cell Splitter node:

This node will take one string column and split it into multiple columns based on a specified delimiter, for example, a comma. This feature isn't available in SPSS Modeler.



String Replacer node:

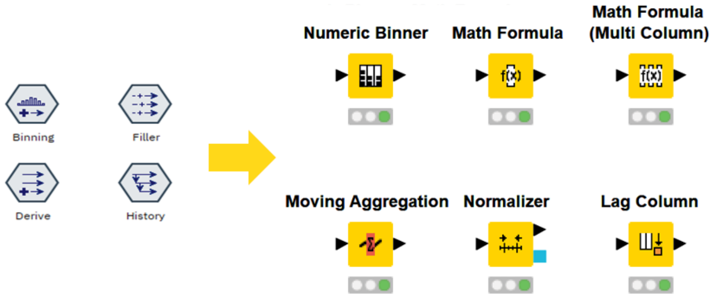
Use the *String Replacer* node for quick replacements or even removals inside strings. For example, configure this node to replace all instances of "two" with "2". This node also supports regular expressions.

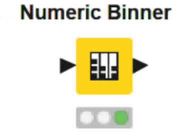
String Replace (Dictionary) node:

You can direct this node to a text file (dictionary) formatted as detailed in the node’s *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.

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.



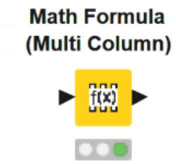


Numeric Binner node:

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

Math Formula node:

Like the *Filler/Derive* nodes in SPSS Modeler, the *Math Formula* node will allow you to alter numeric data with common math functions.

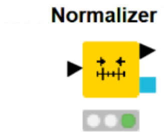


Math Formula (Multi Column) node:

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

Moving Aggregation node:

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



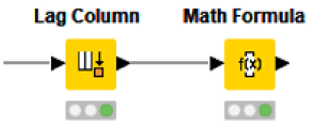
Normalizer node:

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

Lag Column node:

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

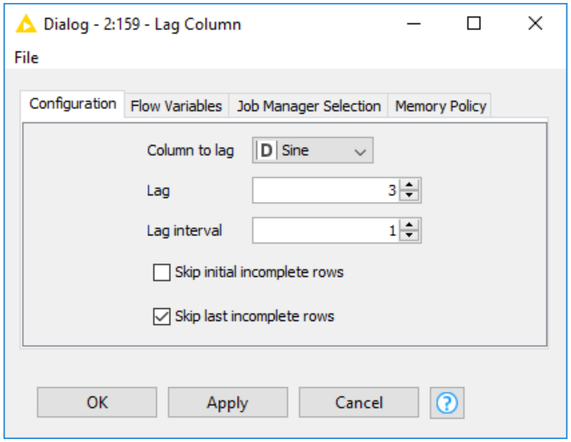


Alteryx has a tool called the *Multi-Row Formula*. This is done in KNIME Analytics Platform through a combination of the *Lag Column* node and the *Math Formula* node. The first creates a new column of shifted values. For example, if you want to reference the previous row in your formula you can use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

Lag Column Noe Configuration

To use the *Lag Column* node, you first select the column to lag in the drop-down menu (see figure on the left). 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 (see figure on the right).

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.



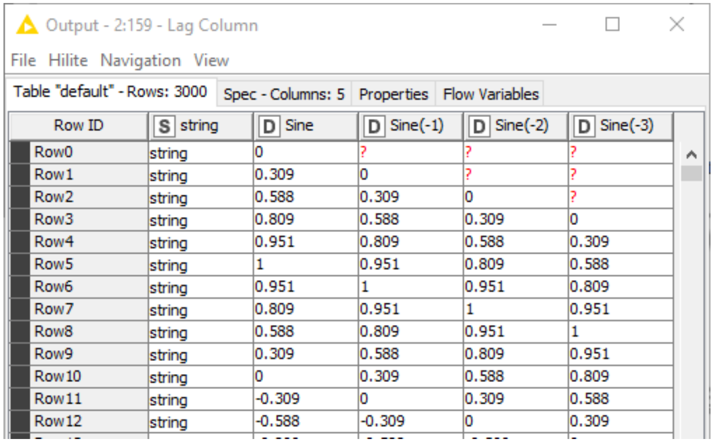
Column to lag: D Sine

Lag: 3

Lag interval: 1

□ Skip initial incomplete rows

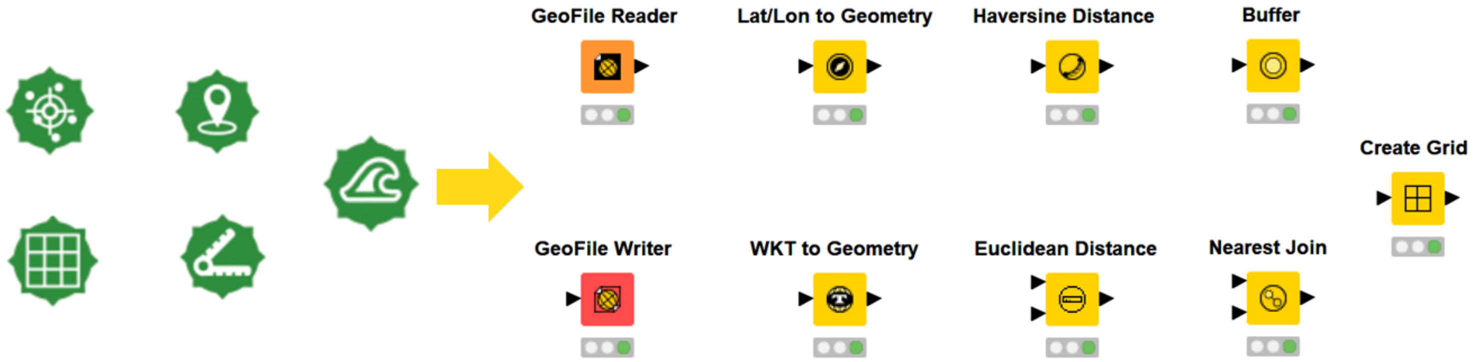
☑ Skip last incomplete rows

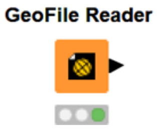

Output - 2:159 - Lag Column
File Hilite Navigation View
Table "default" - Rows: 3000
Spec - Columns: 5
Properties
Flow Variables

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

Geospatial Data

One of the newest additions to the KNIME Analytics platform is the Geospatial Analytics extension, developed in collaboration with the Center for Geographic Analysis at Harvard University. This actively growing collection of nodes provides a wide variety of manipulations, calculations, and visualizations for geospatial data. With this extension, geo files can be read, created from numeric Lat/Lon coordinates, or even from WKT representations. Beyond the standard distance and area calculations, shapes can be clipped, bounded, and even modeled with a Geographically Weighted Regression (GWR) model. Check the KNIME Community Hub for a full list of nodes and features!





GeoFile Reader node:

The first node you use in spatial analytics may be the *GeoFile Reader* node. You can use it to access *.shp*, *.gpkg*, or *.geojson* files. There is also an equivalent *GeoFile Writer* node.

Lat/Lon to Geometry node:

If your spatial data is stored as a Latitude and Longitude pair instead of a geo file, you can convert it to the spatial data type using the *Lat/Lon to Geometry* node.

WKT to Geometry node:

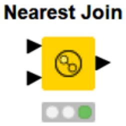
If your spatial data is stored in a WKT string instead of a geo file, you can use the *WKT to Geometry* node to convert it to the spatial data type.

Haversine Distance node:

Alteryx uses one tool for distance calculations, however, in KNIME Analytics Platform there are multiple nodes representing the different types of distances. The *Haversine Distance* node is one of them. Alternatively, you can use the *Euclidean Distance* node.

Buffer node:

The *Buffer* node creates a geometry column containing a polygon that represents the space within a specified distance from a point or other spatial object.



Nearest Join node:

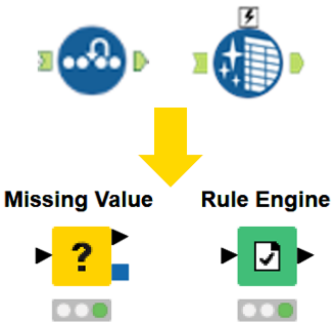
The *Nearest Join* node takes two tables as input. Depending on the merge setting, either every record from the top table is assigned its nearest record from the bottom or vice versa.

Create Grid node:

Use the *Create Grid* node to divide a polygon region into sectors. Define the size of a grid box by setting the length in meters when configuring the node.

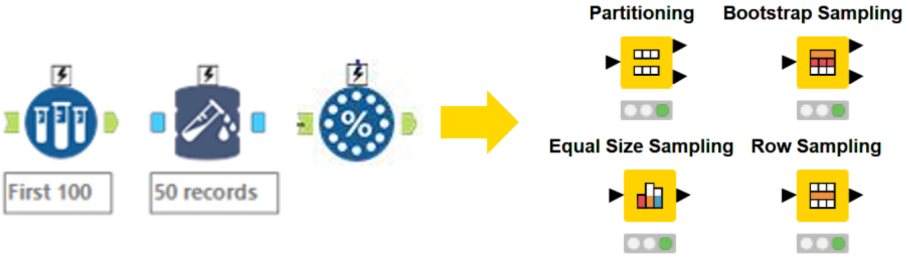
Missing Data

For the basic correction of missing data like that handled by the *Data Cleansing* tool in Alteryx, use the *Missing Value* node in KNIME Analytics Platform. 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* node. 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.



Sampling Data

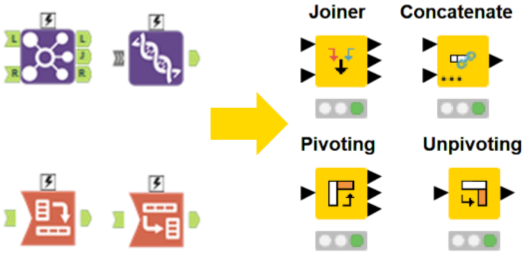
Whether you want to sample data to reduce execution time for analytics or constructing training sets for machine learning and modeling there are many options available in KNIME Analytics Platform.



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 Community 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. The *Equal Size Sampling* node 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 Analytics Platform for analysis.

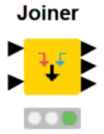
Table Manipulations

I would now like to hit on a few basic operations for manipulating tables as opposed to strictly manipulating the data within. The *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.



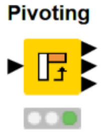
Concatenate node:

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.



Joiner node:

The *Joiner* node in KNIME is going to replace your *Merge* node in SPSS Modeler. 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.



Pivoting node:

Configuring this node is straight forward if you're familiar with pivot tables. You simply specify three things: 1) The columns used as pivots. Their contents will become new columns. 2) The columns to be used as groups. This will let you aggregate the rows as you pivot. 3) The aggregation methods for the fields you wish to retain.

Unpivoting node:

Setting up the *Unpivoting* node I easy as well. Just select the columns you wish to rotate back down into distinct rows (= the value columns) and 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. In the figure to the right there is a node with a comment, an annotation, and a named metanode with a comment.

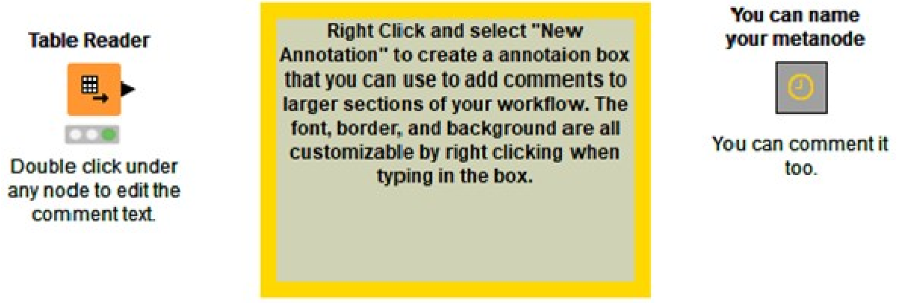


Table Reader

Double click under any node to edit the comment text.

Right Click and select "New Annotation" to create a annotaion box that you can use to add comments to larger sections of your workflow. The font, border, and background are all customizable by right clicking when typing in the box.

You can name your metanode

You can comment it too.

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

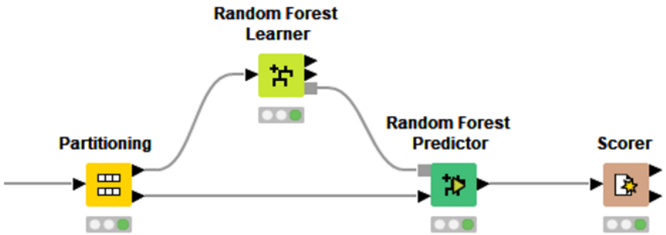
Workflow annotations are colored boxes you can place over your workflow, as can be seen in many KNIME examples. A common use is to clearly separate sections of your workflow into data loading, ETL, modeling, and predicting. This makes it easy for colleagues to quickly identify the area they're looking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

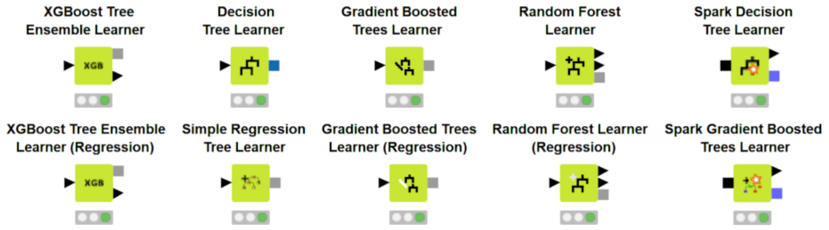
In KNIME Analytics Platform, building models mostly happens in the same framework regardless of the type of model you want to build. You'll start with some data, partition it into training and testing subsets, apply a *Learner* node, a Predictor node, and finally a *Scorer* node to view some statistics.

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?

Trees

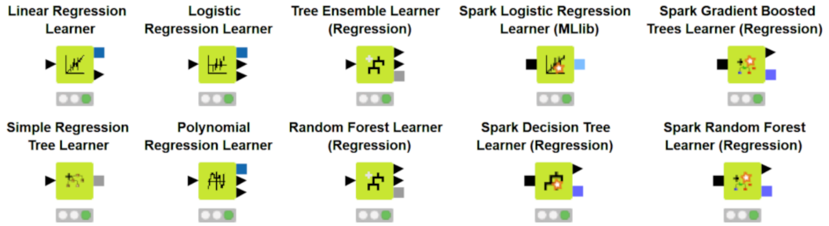
Tree based models are incredibly versatile and come in many varieties. Some rivaling the predictive power of deep neural networks while requiring a fraction of the data and training time. These models aren't to be overlooked and KNIME supports the training and deployment of many.

Both regression and classifications trees are supported as well as their ensembles such as random or boosted forests. In KNIME Analytics Platform 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 customization 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*.


XGBoost Tree Ensemble Learner
XGB
Decision Tree Learner
Gradient Boosted Trees Learner
Random Forest Learner
Spark Decision Tree Learner
XGBoost Tree Ensemble Learner (Regression)
XGB
Simple Regression Tree Learner
Gradient Boosted Trees Learner (Regression)
Random Forest Learner (Regression)
Spark Gradient Boosted Trees Learner

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.


Linear Regression Learner
Logistic Regression Learner
Tree Ensemble Learner (Regression)
Spark Logistic Regression Learner (MLlib)
Spark Gradient Boosted Trees Learner (Regression)
Simple Regression Tree Learner
Polynomial Regression Learner
Random Forest Learner (Regression)
Spark Decision Tree Learner (Regression)
Spark Random Forest Learner (Regression)

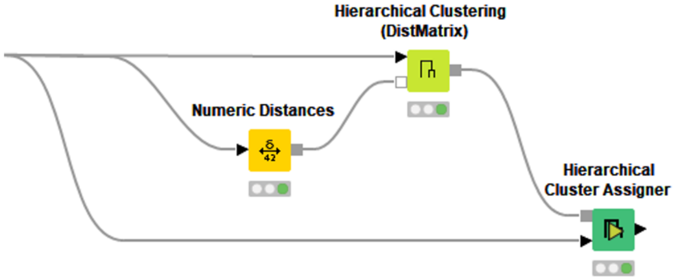
Clustering

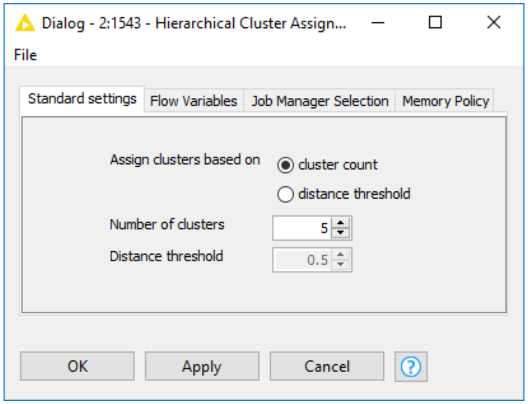
Clustering is an example of an unsupervised technique. This means you can use it without any prelabeled or classified data, as you would normally require for training decision trees, regressions, or neural networks. Clustering means that your data points are grouped according to some distance metric. 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. There is 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 the figure below.





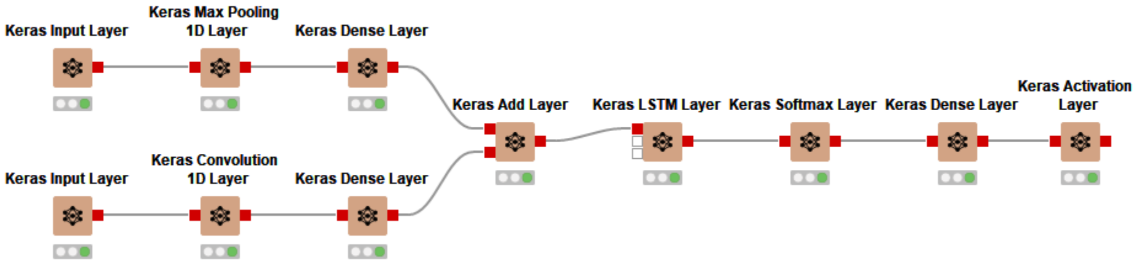
Assign clusters based on ● cluster count ○ distance threshold

Number of clusters 5

Distance threshold 0.5

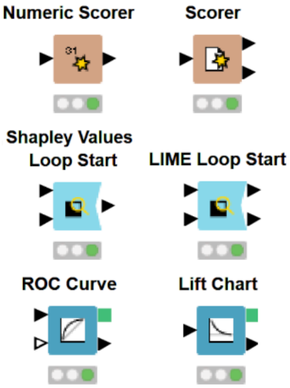
Neural Networks

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.



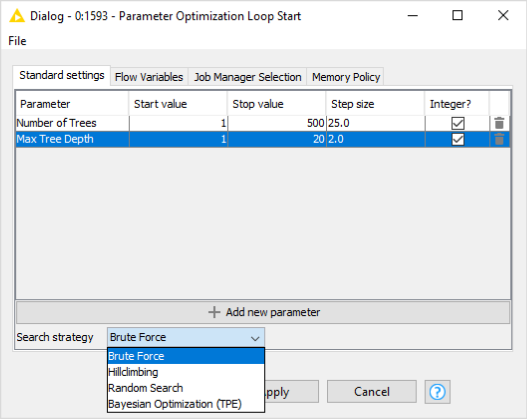
Evaluation

Evaluating the success of your models is another important stage of the data science process. You will 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.



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.



Dialog - 0:1593 - Parameter Optimization Loop Start

File

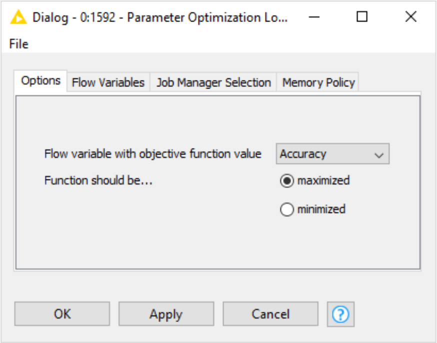
Standard settings | Flow Variables | Job Manager Selection | Memory Policy

Parameter	Start value	Stop value	Step size	Integer?
Number of Trees	1	500	25.0	☑
Max Tree Depth	1	20	2.0	☑

+ Add new parameter

Search strategy: Brute Force

- Brute Force
- Hillclimbing
- Random Search
- Bayesian Optimization (TPE)



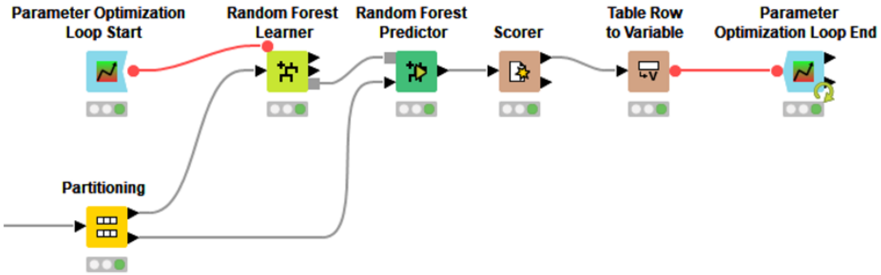
Dialog - 0:1592 - Parameter Optimization Lo...

File

Options | Flow Variables | Job Manager Selection | Memory Policy

Flow variable with objective function value: Accuracy

Function should be...
● maximized
○ minimized



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 data apps, loops, and flow variables. 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 e-learning section of the KNIME website for more information on how to use these features.

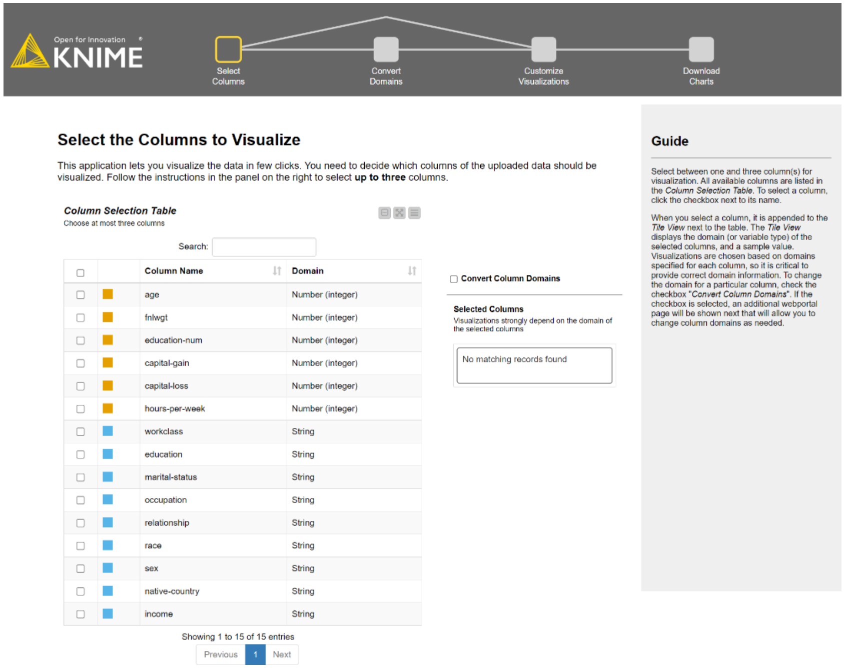
Data Apps (Analytic Apps)

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 a very similar functionality through the KNIME Business Hub, which we call data app. You can build such a data app by using workflows and making them accessible to business users by deploying them to the KNIME Business Hub. Business users can then interact with the data app from the web browser.

Building a data app is easy, and I'll summarize the steps below, but first let's talk about *widgets* and *components*.

In the figure on the right you see one page of a data app example called "Guided Visualization" which is available on the KNIME Community Hub for you to try



Select the Columns to Visualize

This application lets you visualize the data in few clicks. You need to decide which columns of the uploaded data should be visualized. Follow the instructions in the panel on the right to select **up to three** columns.

Column Selection Table

Choose at most three columns

Search:

□		Column Name	Domain
□	🟧	age	Number (integer)
□	🟧	fnlwgt	Number (integer)
□	🟧	education-num	Number (integer)
□	🟧	capital-gain	Number (integer)
□	🟧	capital-loss	Number (integer)
□	🟧	hours-per-week	Number (integer)
□	🟦	workclass	String
□	🟦	education	String
□	🟦	marital-status	String
□	🟦	occupation	String
□	🟦	relationship	String
□	🟦	race	String
□	🟦	sex	String
□	🟦	native-country	String
□	🟦	income	String

Showing 1 to 15 of 15 entries

Previous 1 Next

□ **Convert Column Domains**

Selected Columns

Visualizations strongly depend on the domain of the selected columns

No matching records found

Guide

Select between one and three column(s) for visualization. All available columns are listed in the *Column Selection Table*. To select a column, click the checkbox next to its name.

When you select a column, it is appended to the *Tile View* next to the table. The *Tile View* displays the domain (or variable type) of the selected columns, and a sample value. Visualizations are chosen based on domains specified for each column, so it is critical to provide correct domain information. To change the domain for a particular column, check the checkbox "*Convert Column Domains*". If the checkbox is selected, an additional webportal page will be shown next that will allow you to change column domains as needed.

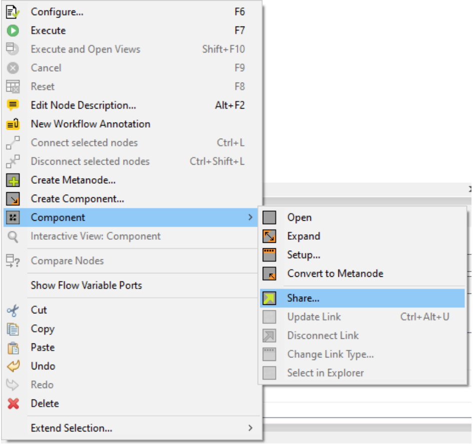
out. The view you see is a single component, and, as you continue to move through the data app, each new page is based on a separate component in your workflow. This allows the user to move through the entire workflowat specific interaction points that you have specified. The above shown example allows a user to upload a data file of their choice and then create custom visualizations. 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 deployed to the KNIME Business Hub for use.

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

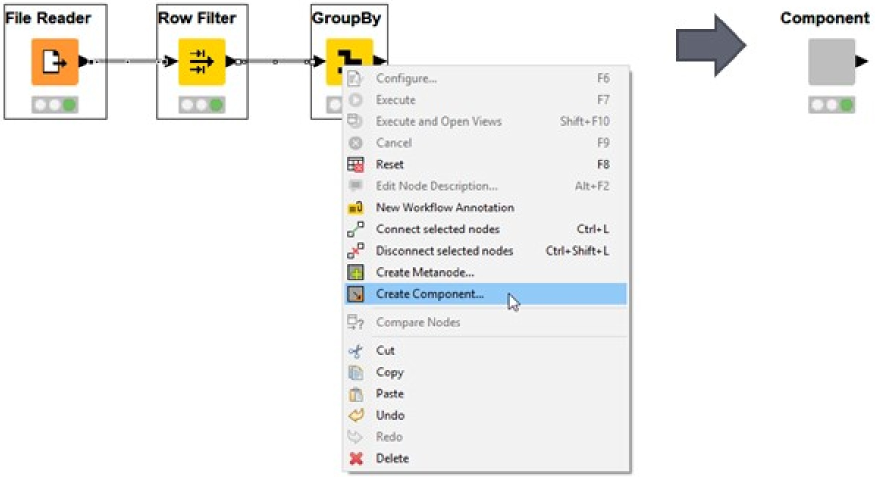
Components (Macros)

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.

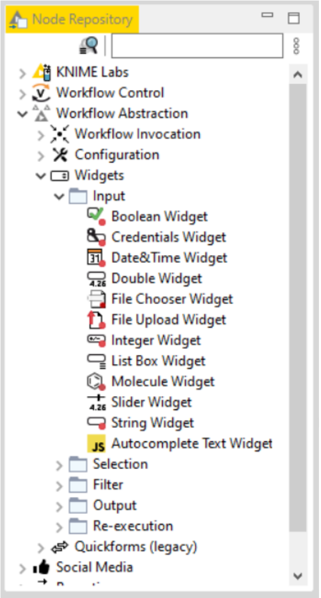


In KNIME Analytics Platform we call macros "components". These components can be saved either locally or to a server for repeated use and sharing. Do this by right clicking on your component, expanding the component line, and then choosing the "Share…" option (see figure on the right).

To create a component intended for use *locally* the only major addition is a (or multiple) configuration node(s), found in the *Workflow Abstraction* folder in the *Node Repository*. These configuration nodes behave a lot like some of their widget counterparts with one exception. Instead of displaying in the data app, the configuration nodes display in the configuration window of the components. This makes components behave just like a regular KNIME node.



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 Date & Time, etc. *Input* also contains the *File Upload Widget* node to allow the user to supply their own data.

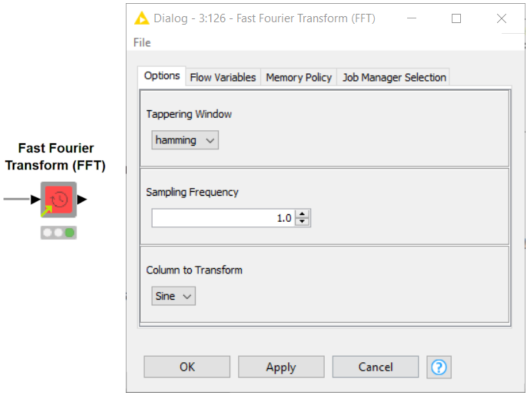
Selection allows the user to set things such as filter parameters, or column(s) for processing.

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.

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 data apps, configuration nodes allow you to create configuration windows for the components. The *Fast Fourier Transform (FFT)* component in the figure on the right has its configuration window next to it. This is the result of putting a *Column Selection Configuration* node, a *Single Selection Configuration* node, and a *Double Configuration* node inside the component for the user to interact with.


Fast Fourier Transform (FFT)
Dialog - 3:126 - Fast Fourier Transform (FFT)
File
Options
Flow Variables
Memory Policy
Job Manager Selection
OK
Apply
Cancel

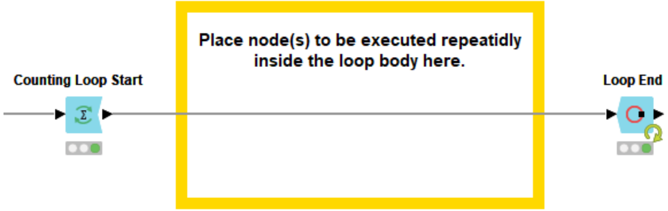
Tappering Window: hamming

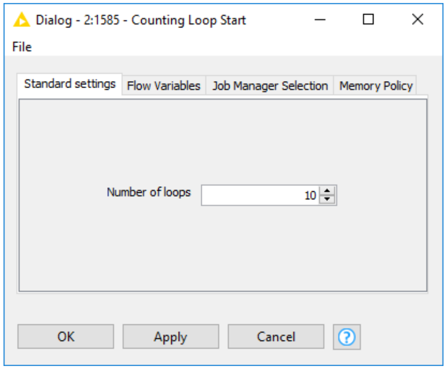
Sampling Frequency: 1.0

Column to Transform: Sine

Loops

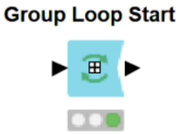
Loops are sections of your workflow that will execute multiple times. This could be a set number of times, until every row in a table is processed, until some variable condition is met, or even indefinitely. In KNIME Analytics Platform, a loop is built by combining a loop start and loop end node, both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


Counting Loop Start
Place node(s) to be executed repeatidly inside the loop body here.
Loop End



Number of loops: 10

The image above is using the *Counting Loop Start* node which simply loops a given number of times which you can define in the configuration window. Let's look at a couple other types of loops in KNIME to get you more familiar with what's possible. Below are only three types but there are several more available that you can explore as well.



Group Loop Start node:

The *Group Loop Start* node works a lot like the *GroupBy* node, which we looked at earlier. 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 groups.

Recursive Loop Start node:

The *Recursive Loop Start* node is special as it's 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.

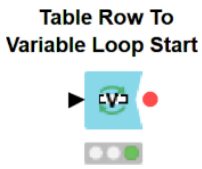


Table Row to Variable Loop Start node:

The *Table Row to Variable Loop Start* node doesn't supply data to each iteration like the others. Instead, 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.

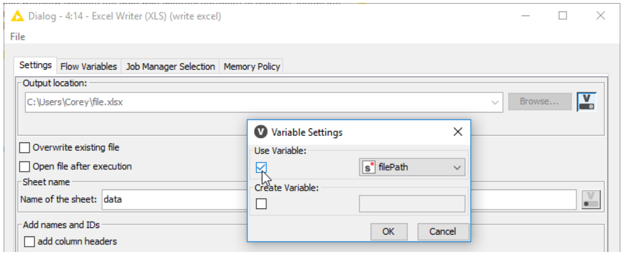
Flow Variables

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 workflow below, 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. This 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 then pass the file path on to the *Excel Writer (XLS)* and use it to automatically set the location and name of the file we write. In the configuration window click on the **V** next to the output location (see image on the left) to control it with a variable, then select the variable created, *filePath* in this case, and the setting will be dynamically controlled. Perfect!



Dialog - 4:14 - Excel Writer (XLS) (write excel)

Output location: C:\Users\Corey\file.xlsx

□ Overwrite existing file

□ Open file after execution

Sheet name

Name of the sheet: data

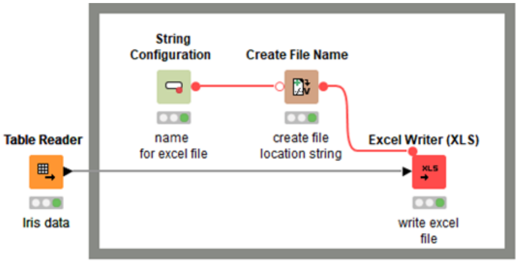
Add names and IDs

□ add column headers

Variable Settings

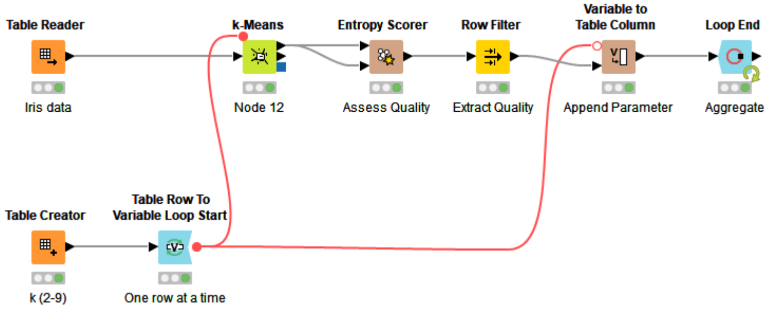
Use Variable: ☑ filePath

Create Variable: □



Example 2, Loops:

This example workflow is available in the Examples space on the KNIME Community Hub or in *KNIME Examples*, a repository of workflows and components ready to be executed on some data directly available in your KNIME Explorer, under *06_Control Structures > 04_Loops > 01_Loop_over_a_set_of_parameter_for_k_means*.



In this workflow, you see that instead of manually creating a flow variable with a configuration node, the red variable line starts at the *Table Row to Variable Loop Start* node (we briefly addressed this node in the section on loops). It converts each column of a table to a variable and then iterates through them one row at a time. This allows 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 that allows for different values for the parameter k (= the number of clusters) and at the end of the loop we're collecting that information along with some information from the *Entropy Scorer* node. That helps us decide how to best cluster our data.

Appendix

Available Data Types

Data Type	Alteryx	KNIME Analytics Platform	Notes
Bool	✓	✓	
INT	✓	✓	
Decimal	✓	✓	Both having multiple options for precision
Complex Number	✗	✓	
String	✓	✓	Alteryx having multiple options for storage efficiency
Nominal	✗	✓	
Date and/or Time	✓	✓	Dates, Times, or Date / Times
Spatial Objects	✓	✓	Points, Lines, and Polygons
Network / Graph	✓	✓	
Audio	✗	✓	*.wav* format

Image	✘	✓	
Document	✘	✓	Includes text and meta data for text mining
Collection	✘	✓	List of values in single table cell

Quick Tool to Node Reference

Alteryx	KNIME Analytics Platform
Browse	Data Explorer node *(Alternative: Statistics node)*
Input Data	File Reader node *(Alternative: DB Reader node, Tika Parser node, or Hive/Impala Connector nodes)*
Output Data	Excel Writer node *(Alternative: Table Writer node or DB Writer node)*
Text Input	Create Table node *(Alternative: String Input (legacy) node)*
Data Cleansing	Missing Value node *(Alternative: String Manipulation node)*
Filter	Row Filter node *(Alternative: Rule-based Row Filter node or Row Splitter node)*
Formula	String Manipulation node *(Alternative: Math Formula node, Rule Engine node, or Column Expression node)*

Sample	Row Sampling node *(Alternative: Bootstrap Sampling node, Equal Size Sampling node, or Partitioning node)*
Select	Column Filter node *(Alternative: Column Rename node or Column Sorter node)*
Sort	Sorter node
Join	Joiner node *(Alternative: Cross Joiner node)*
Union	Concatenate node
Text to Columns	Cell Splitter node
Summarize	GroupBy node
Tile	Auto-Binner node *(Alternative: Numeric Binner node, Binner (Dictionary) node, or CAIM Binner node)*
Imputation	Missing Value node *(Alternative: Rule Engine node)*
Render	Table to PDF node *(Alternative: Image Writer (Port) node)*

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 Community 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

- *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/events

From Alteryx to KNIME

This guide will help you transition from Alteryx to KNIME. It maps the most commonly used Alteryx functions and techniques to their KNIME equivalents: from importing data, to manipulating data, to documenting your workflow, through to modeling and machine learning.

Corey Weisinger studied Mathematics at Michigan State University and works as a Data Scientist with KNIME where he focuses on Time Series Analysis, Forecasting, and Signal Analytics. He is a co-creator and instructor of the KNIME Time Series Analysis course, author of the e-book: Alteryx to KNIME, creator of the KNIME Time Series Analysis components, and Co-Author of the Codeless Time Series Analysis Book.