Spotfire Bootcamp

Intro to Data Science

Day 3

Introduction to Spotfire Data Science

Multivariate Data Analysis and Line Similarity

Relationships and Predictive Modeling

Automation API

R and TERR

Data Functions

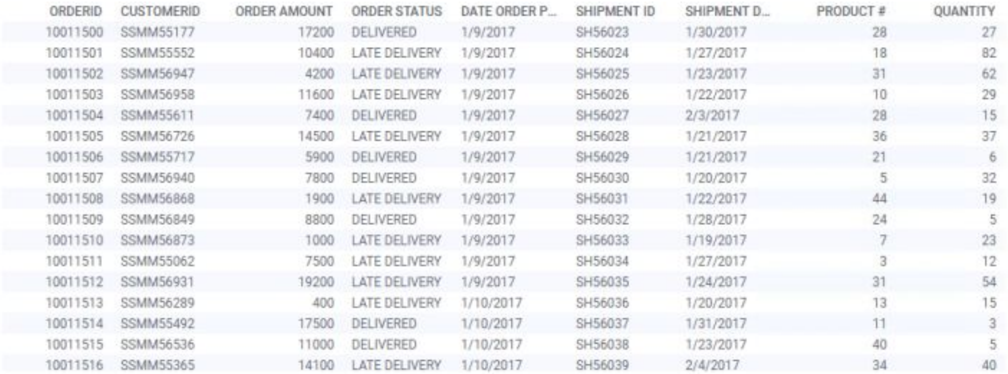
Multivariate Data Analysis and Line Similarity

Multivariate Data

Multivariate data analysis is a statistical approach which is used to analyze the data that derives from more than one variable.

The variables used to generate data are prototypes of real time situations, products and services or decision making involving more than one variable.

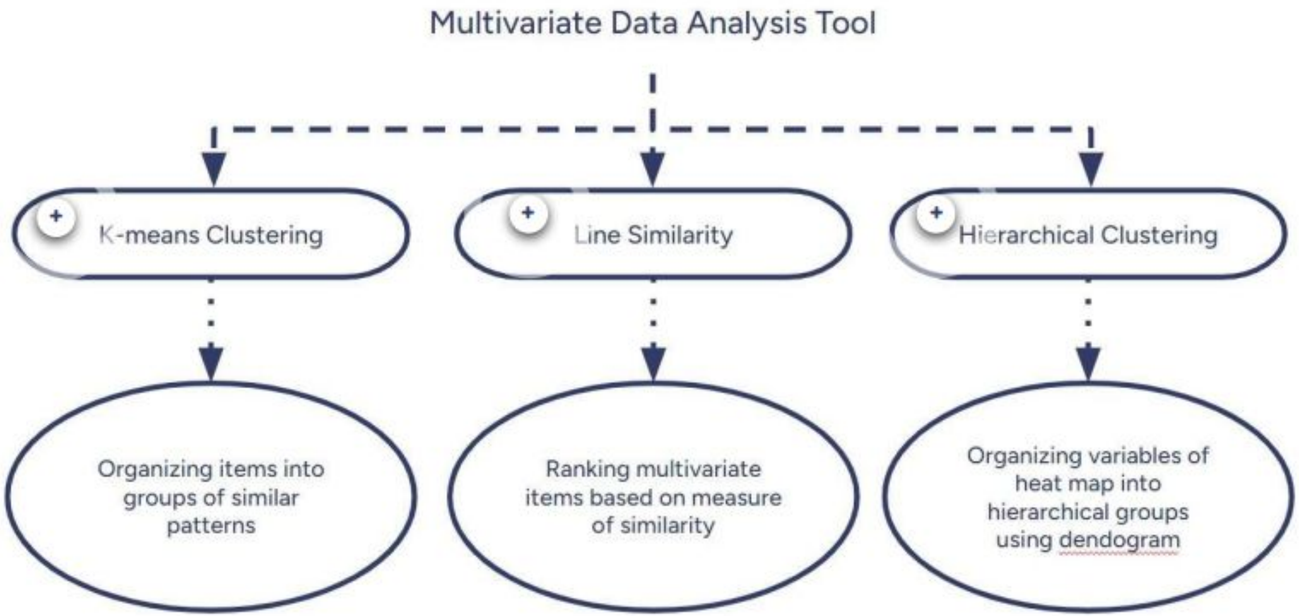
Multivariate analysis can be used to read and transform data stored in various databases to relevant data.



ORDERID	CUSTOMERID	ORDER AMOUNT	ORDER STATUS	DATE ORDER P...	SHIPMENT ID	SHIPMENT D...	PRODUCT #	QUANTITY
10011500	SSMM55177	17200	DELIVERED	1/9/2017	SH56023	1/30/2017	28	27
10011501	SSMM55552	10400	LATE DELIVERY	1/9/2017	SH56024	1/27/2017	18	82
10011502	SSMM56947	4200	LATE DELIVERY	1/9/2017	SH56025	1/23/2017	31	62
10011503	SSMM56958	11600	LATE DELIVERY	1/9/2017	SH56026	1/22/2017	10	29
10011504	SSMM55611	7400	DELIVERED	1/9/2017	SH56027	2/3/2017	28	15
10011505	SSMM56726	14500	LATE DELIVERY	1/9/2017	SH56028	1/21/2017	36	37
10011506	SSMM55717	5900	DELIVERED	1/9/2017	SH56029	1/21/2017	21	6
10011507	SSMM56940	7800	DELIVERED	1/9/2017	SH56030	1/20/2017	5	32
10011508	SSMM56868	1900	LATE DELIVERY	1/9/2017	SH56031	1/22/2017	44	19
10011509	SSMM56849	8800	DELIVERED	1/9/2017	SH56032	1/28/2017	24	5
10011510	SSMM56873	1000	LATE DELIVERY	1/9/2017	SH56033	1/19/2017	7	23
10011511	SSMM55062	7500	LATE DELIVERY	1/9/2017	SH56034	1/27/2017	3	12
10011512	SSMM56931	19200	LATE DELIVERY	1/9/2017	SH56035	1/24/2017	31	54
10011513	SSMM56289	400	LATE DELIVERY	1/10/2017	SH56036	1/20/2017	13	15
10011514	SSMM55492	17500	DELIVERED	1/10/2017	SH56037	1/31/2017	11	3
10011515	SSMM56536	11000	DELIVERED	1/10/2017	SH56038	1/23/2017	40	5
10011516	SSMM55365	14100	LATE DELIVERY	1/10/2017	SH56039	2/4/2017	34	40

Multivariate Data - Spotfire inbuilt tools

Spotfire has three inbuilt tools to perform analysis on multi-dimensional data tables.



Multivariate Data Analysis Tool

K-means Clustering

Organizing items into groups of similar patterns

Line Similarity

Ranking multivariate items based on measure of similarity

Hierarchical Clustering

Organizing variables of heat map into hierarchical groups using dendogram

Multivariate Data - Spotfire inbuilt tools

K-means Clustering is an algorithm for partitioning a data table into subsets (clusters), in such a way that the members of each cluster are relatively similar.

Line Similarity is used to compare the lines in a line chart to a selected master line.

Hierarchical Clustering arranges items in a hierarchy with a tree-like structure based on the distance or similarity between them.

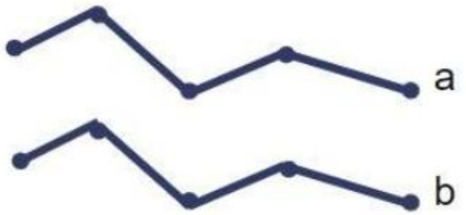
Multivariate Data - Line Similarity

Correlation Similarity

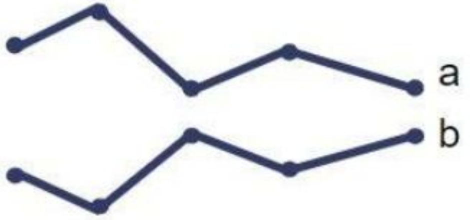
The correlation between two points, a and b, with k dimensions is calculated using Pearson's correlation.

It compares the shape of a line to the master line selected. It ranges from +1 to -1 where +1 is the highest correlation.

Complete opposite points have correlation -1.



a and b are identical, which means they have maximum correlation
Similarity = +1



a and b are perfectly mirrored, which means they have maximum negative correlation
Similarity = -1

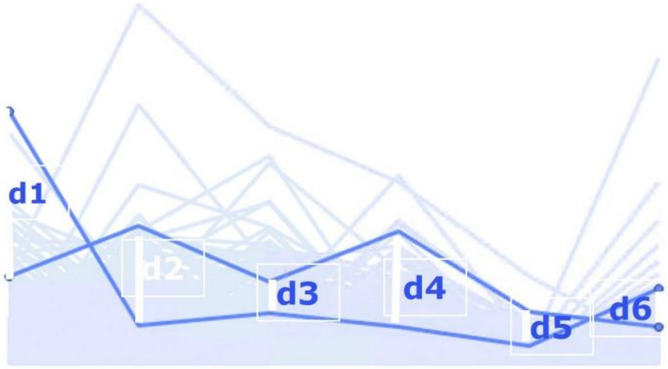
Multivariate Data - Euclidean Distance

In mathematics, the Euclidean distance or Euclidean metric is the "ordinary" straight-line distance between two points in Euclidean space.

With this distance, Euclidean space becomes a metric space. The Euclidean distance is always greater than or equal to zero.

Similarity = √∑d2
Where d is the distance between the dotted lines in the figure

The measurement would be zero for identical points and high for points that show little similarity.



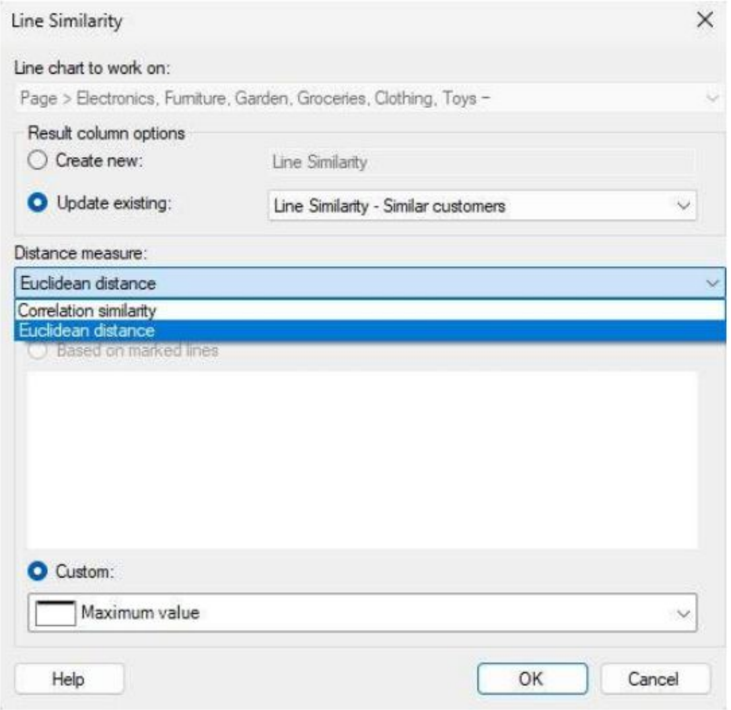
Multivariate Data - Line Similarity Comparison

The Line Similarity tool cannot be used unless you have created a suitable line chart to base the calculation on. All the values that should be included in the calculation are selected in y-axis; and x-axis is selected as “Column Names”.

You can create multiple lines by setting “Line by” or “Color by” variable. Mark one or more lines to use as the master line against which the search will be performed.

Go to the Tools menu and select Line Similarity. If you have multiple line charts in the page, select the Line chart to work on from the drop down list.

Create a new Line Similarity result column. Select a Distance measure to use in the calculation and select how to use master line.



Line Similarity

Line chart to work on: Page > Electronics, Furniture, Garden, Groceries, Clothing, Toys –

Result column options

○ Create new: Line Similarity

● Update existing: Line Similarity - Similar customers

Distance measure: Euclidean distance

Correlation similarity
Euclidean distance

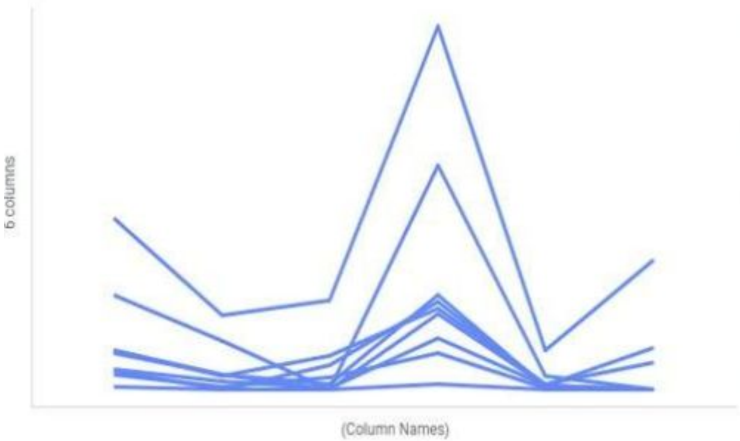
○ Based on marked lines

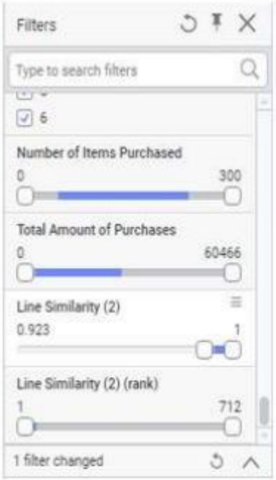
● Custom: Maximum value

Multivariate Data - Line Similarity Output

Two new columns are added to the data table and two new filters representing the columns are shown in the data and filters panels.

- Similarity column: represents the similarity to the master line for each row
- Rank column: lines most similar to the master line receives rank 1




Filters

Type to search filters

☑ 6

Number of Items Purchased
0 300

Total Amount of Purchases
0 60466

Line Similarity (2)
0.923 1

Line Similarity (2) (rank)
1 712

1 filter changed

Multivariate Data - Normalization and Empty Values

While performing multivariate data analysis, the variables used may have different data ranges. To reduce the difference, you can perform normalization on the data before performing the similarity calculation. You can perform normalization either using Add transformation option from the Data menu or you can use normalization settings in the Hierarchical Clustering tool itself.

The empty value replacement defines how empty values in the data set should be replaced in the clustering calculation.

- Constant value: replaces the value by a constant number.
- Column Average: returns the average of the corresponding column values.
- Row Average: replaces the value by the average value of the entire row.
- Row Interpolation: sets the missing value to the interpolated value between the two
- neighboring values in the row.

Multivariate Data - Normalization methods

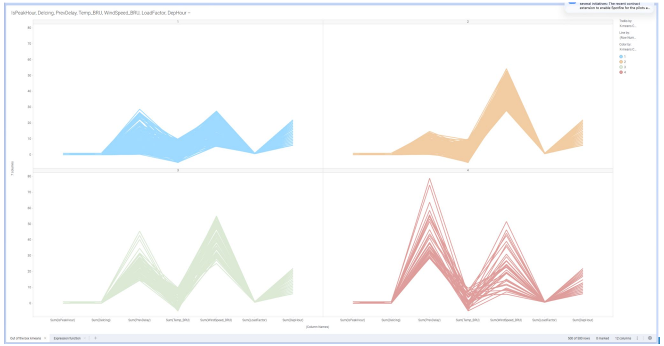
- Normalize by mean
- Normalize by trimmed mean
- Normalize by percentile
- Scale between 0 and 1
- Subtract the mean
- Subtract the median
- Normalization by signed ratio
- Normalization by log ratio
- Normalization by log ratio in standard deviation units
- Z-score calculation
- Normalize by standard deviation

Multivariate Data - Exercise

Load the data file “Exercise Random Forest Dataset.csv” into Spotfire
Create a line chart with a line per row and add your variables to the y-Axis, select (Column Names) on the X-Axis

Now use the built in K-Means clustering function to create 4 clusters of data

Analyze the result (is there a relationship between clusters and delays?


IsPeakHour, DeIcing, PrevDelay, Temp_BRU, WindSpeed_BRU, LoadFactor, DepHour –

several initiatives: The recent contract extension to enable Spotfire for the pilots a...

Trellis by: K-means C...

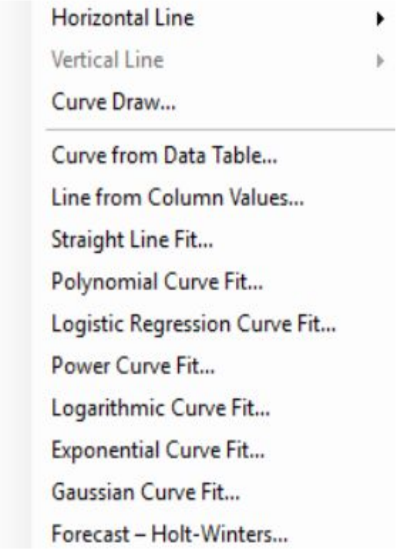
Line by: (Row Num...

Color by: K-means C...

500 of 500 rows 0 marked 12 columns

Relationships and Predictive Modeling

Reference Lines and Curves



Lines and curves

- Visualization Properties -> Lines and Curves
- Display visual relationship between variables

Lines

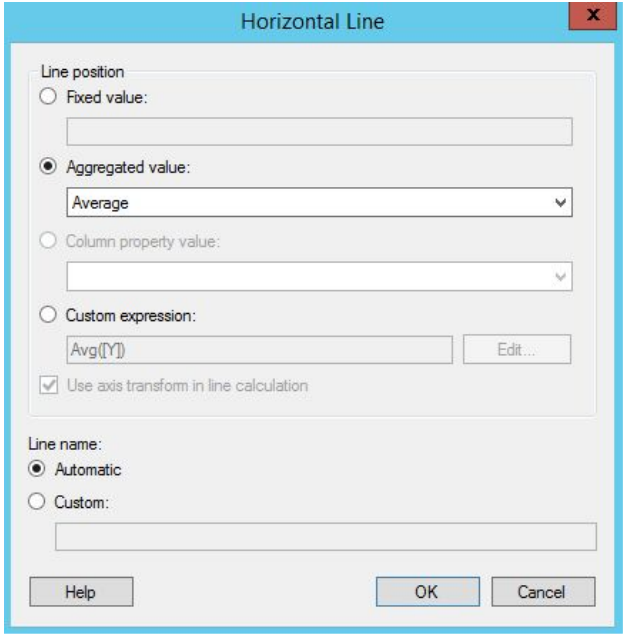
- Fixed: Do not change with filtering
- Calculated: Change with filtering

Curve fit

- Not a fixed curve but a line that changes with input data
- Define custom curves with fixed value, custom expressions, X and Y refer to the axes values

Horizontal/Vertical Line Dialog

- Lines & Curves -> Add -> Horizontal/Vertical Line
- Line position can be a fixed or aggregated value, property or custom expression
- Orientation of line switches with visualization



Line position

○ Fixed value:

● Aggregated value: Average

○ Column property value:

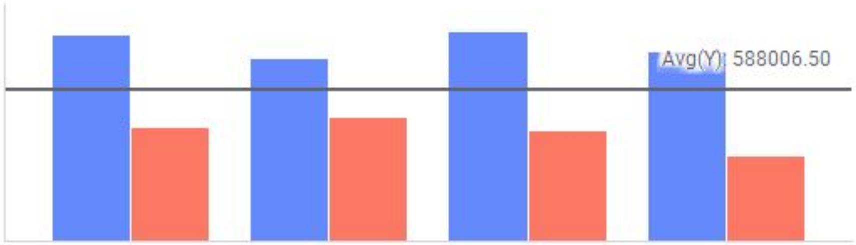
○ Custom expression: Avg([Y])

☑ Use axis transform in line calculation

Line name:

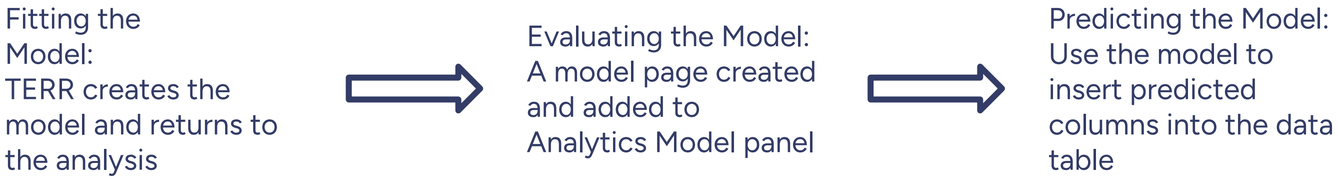
● Automatic

○ Custom:



Predictive Modeling

- Using existing column relationship to predict new column
- Uses regression/classification modeling
- TERR executes all models
- The tasks of predictive modeling includes



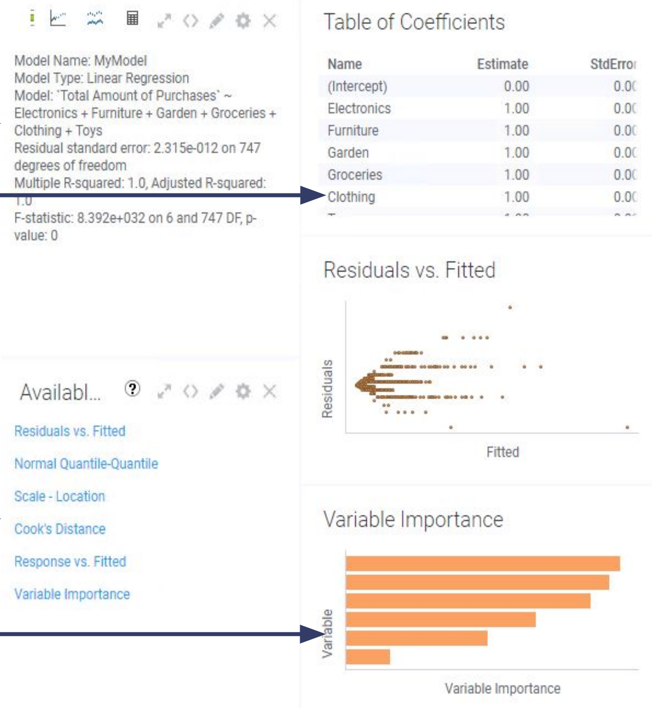
Fitting the Model:
TERR creates the model and returns to the analysis

Evaluating the Model:
A model page created and added to Analytics Model panel

Predicting the Model:
Use the model to insert predicted columns into the data table

Model and Evaluation Pages

- Created with a new Model
- Contains four sections
  - Model/Evaluation Summary
    - Displays the model name, type and model formula
  - Table of Coefficients
    - Provides model coefficients for the parametric models
    - Includes measure of variability
    - Not for Evaluation summary
  - Available Diagnostic Visualizations
    - Lists available diagnostics plots for the model
  - Visualization area to display diagnostic visualizations



Model Name: MyModel
Model Type: Linear Regression
Model: `Total Amount of Purchases` ~ Electronics + Furniture + Garden + Groceries + Clothing + Toys
Residual standard error: 2.315e-012 on 747 degrees of freedom
Multiple R-squared: 1.0, Adjusted R-squared: 1.0
F-statistic: 8.392e+032 on 6 and 747 DF, p-value: 0

Table of Coefficients

Name	Estimate	StdError
(Intercept)	0.00	0.0
Electronics	1.00	0.0
Furniture	1.00	0.0
Garden	1.00	0.0
Groceries	1.00	0.0
Clothing	1.00	0.0

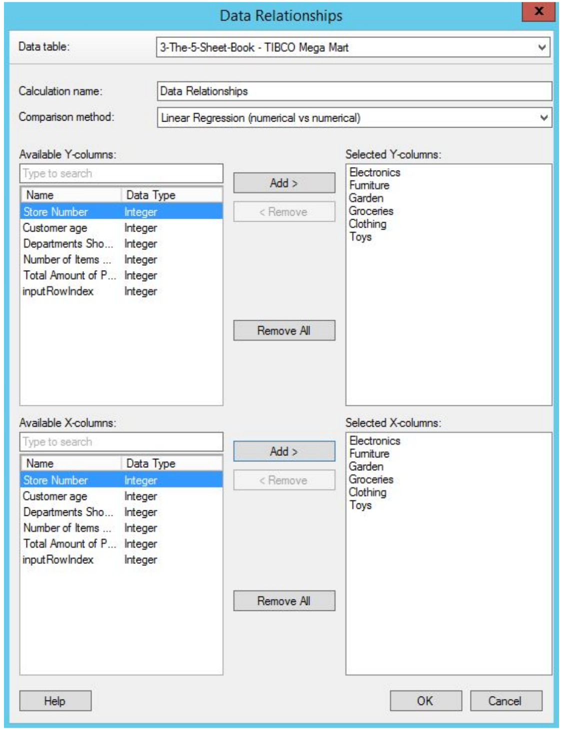
Data Relationships Tool

Investigate the relationship between different column pairs

Works on currently filtered data

For each combination of columns, calculates a p-value

Low p-value indicates strong connection



Data table: 3-The-5-Sheet-Book - TIBCO Mega Mart

Calculation name: Data Relationships

Comparison method: Linear Regression (numerical vs numerical)

Available Y-columns:

Type to search

Name	Data Type
Store Number	Integer
Customer age	Integer
Departments Sho...	Integer
Number of Items ...	Integer
Total Amount of P...	Integer
inputRowIndex	Integer

Selected Y-columns:

Electronics
Furniture
Garden
Groceries
Clothing
Toys

Available X-columns:

Type to search

Name	Data Type
Store Number	Integer
Customer age	Integer
Departments Sho...	Integer
Number of Items ...	Integer
Total Amount of P...	Integer
inputRowIndex	Integer

Selected X-columns:

Electronics
Furniture
Garden
Groceries
Clothing
Toys

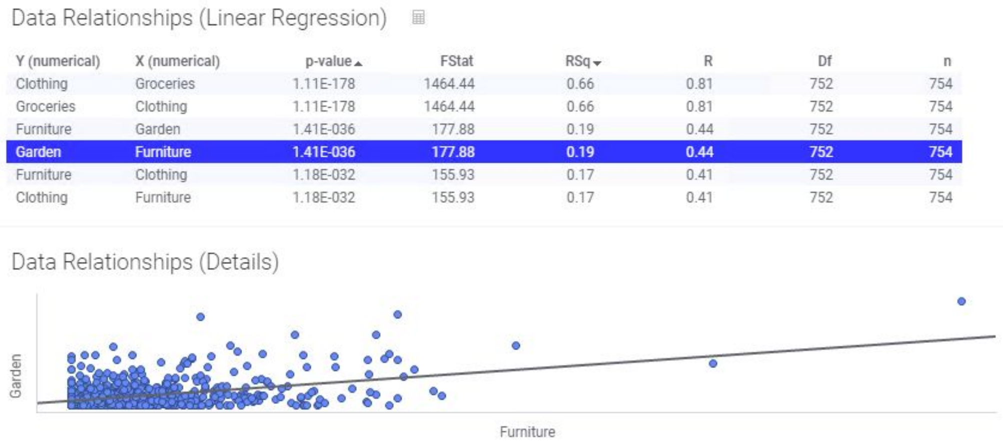
Data Relationships Tool

Input

- Select x and y values for comparison

Output

- Table of metrics is for pairwise comparison
- Visualization displays each comparison result



Data Relationships (Linear Regression)

Y (numerical)	X (numerical)	p-value ▲	FStat	RSq ▼	R	Df	n
Clothing	Groceries	1.11E-178	1464.44	0.66	0.81	752	754
Groceries	Clothing	1.11E-178	1464.44	0.66	0.81	752	754
Furniture	Garden	1.41E-036	177.88	0.19	0.44	752	754
Garden	Furniture	1.41E-036	177.88	0.19	0.44	752	754
Furniture	Clothing	1.18E-032	155.93	0.17	0.41	752	754
Clothing	Furniture	1.18E-032	155.93	0.17	0.41	752	754

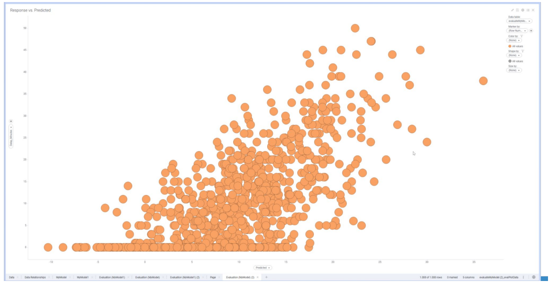
Data Relationships (Details)

Data Relationships - Exercise

1. Load the dataset ‘Exercise datarelations and predict.csv’
2. Go to tools data relationships and choose and regression model to check the relationships between delay_minutes and windspeed, visibility, precipitation and schedule hour
3. Select each of the X variables to the the relationship

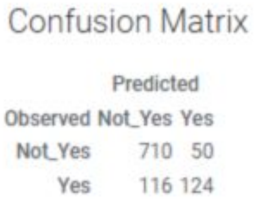
Predictive modelling - Exercise

1. Use the same dataset to predict the minutes of delay. Go to tools, regression modelling.
2. Choose linear regression
3. Response column should be delay_minutes
4. Choose the predictor columns
5. Analyse the results. Now remove features with a low importance
6. Check if the model has improved
7. Now we are going to validate the model. Add the predict_validation.csv file to the analysis and make sure it is loaded as a new table.
8. Go to the model page, select evaluate model and the choose the new dataset
9. Evaluate the result, specifically check the response vs predicted.
10. Finally we can add the predict_unlabeled dataset as a new table and choose predict from model to predict the delay status and see it confidence level.



Predictive modelling - Exercise

1. This time we will predict the column is_delayed.
2. Go to tools and choose classification modelling
3. Use logistic regression as the method
4. The response column is is_delayed and response_level is yes
5. Now again add feature columns that are related to the delay.
6. Analyse the result. Remove any columns with low contribution or directly related (delay_minutes)
7. Now we are going to validate the model. Add the predict_validation.csv file to the analysis and make sure it is loaded as a new table.
8. Go to the model page, select evaluate model and the choose the new dataset
9. Evaluate the result, specifically check the confusion matrix.
10. Finally we can add the predict_unlabeled dataset as a new table and choose predict from model to predict the delay status and see it confidence level.



Confusion Matrix

	Predicted	
Observed	Not_Yes	Yes
Not_Yes	710	50
Yes	116	124

Automation API

Spotfire Modules and Objects

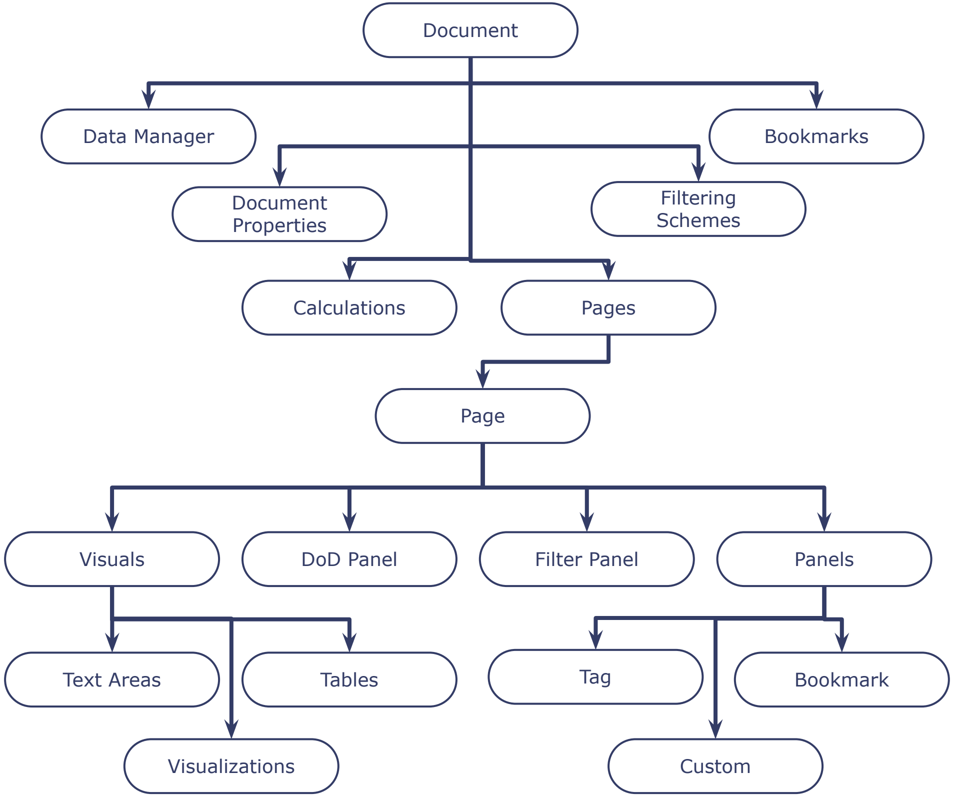
Import names that refers to objects from namespaces

Import gives a reference to entire object

Access members using the dotted notation

Use ‘from’, to access only a few objects from the module

Document model is a subtree of application model



Document Object

Set of properties

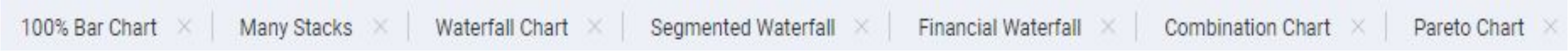
- To access the subtree of the model
  - Pages, properties, bookmarks, filter schemes, etc.
- To keep tracks of current state
  - Active page, active visualization

Document.ActivePageReference

Properties	Description
ActiveDataTableReference	Reference to the Active Data Table
ActiveFilteringSelectionReference	Reference to the Active Filtering Selection
ActiveMarkingSelectionReference	Reference to the Active Filtering Selection
ActivePageReference	Reference to the Active Page Reference
ActiveVisualReference	Reference to the Active Visual
Bookmarks	Access to the Bookmarks Collection
Data	Access to the Data Model
FilteringSchemes	Access to the FilteringSchemesCollection
Pages	Access to the Pages Collection(which includes all things on a Page, including visuals and Panels)
ScriptManager	Access to the scripts stored inside the Spotfire Analysis file

Pages Object

Collection of objects to access to the pages in a document

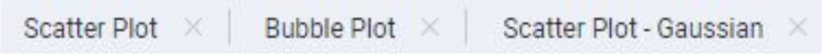


Methods to create, delete, move pages

Method	Description
AddNew()	Creates and adds a new page
Move()	Moves the page from the specified old index to the specified new index
MoveVisualTo()	Moves a visual from its current page to a new page
Remove()	Removes the specified page
RemoveAt()	Removes the page at the specified index

Page display is defined by the navigation mode

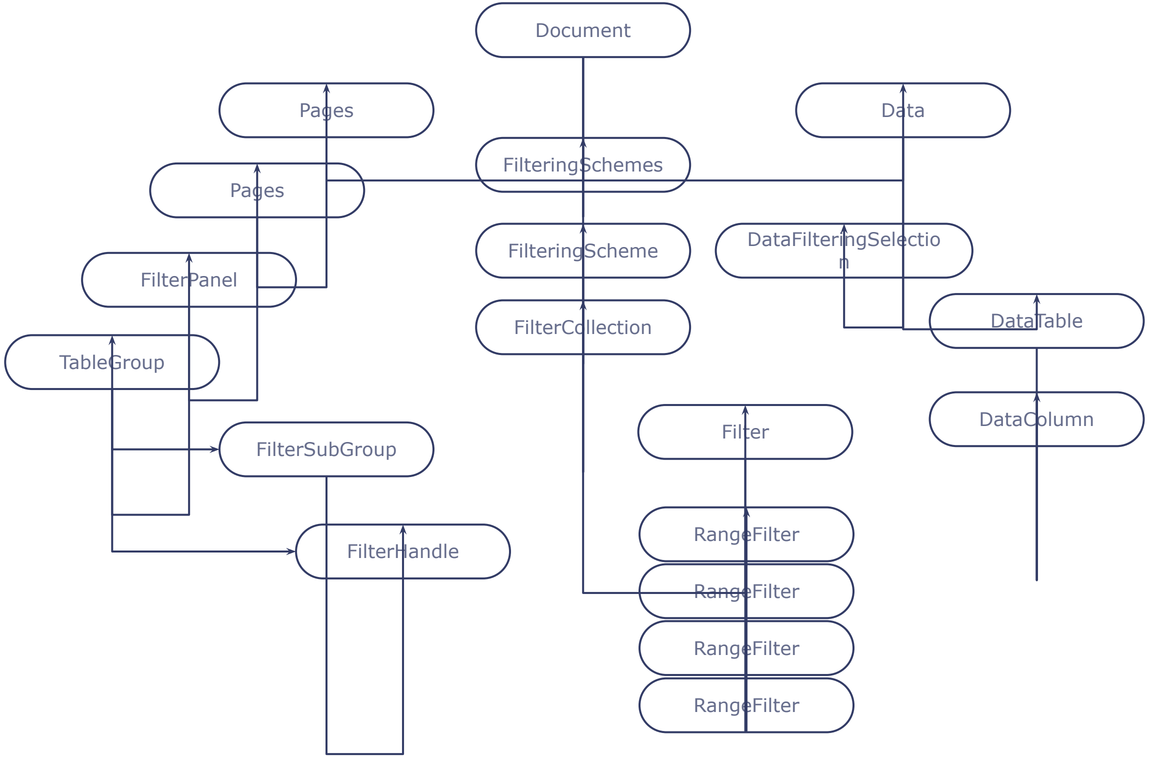
- Values defined by PageNavigationMode Enumeration: tabs, links or none
  - Documents.Pages.NavigationMode=PageNavigationMode.Links



Filter Object

Filter collection can be accessed through

- Page.FilterPanel
- FilterScheme
- Data



Plot and Common Plot Properties

- Creating plots is done using Visual.AddNew method

```
sPlot = Document.ActivePageReference.Visuals.AddNew[sPlot]()
sPlot.AutoConfigure()
sPlot.ApplyUserPreferences()
```

- Accessing the visuals on a page

```
for vis in Application.Document.ActivePageReference.Visuals:
# do something
# vis.Title
```

- Each plot has its own properties

Properties	
Data	Title
Description	Trellis
Details	TypeId
FittingModels	XJitter
Legend	YJitter

Axis and Axis Expressions

- Visualization have various axes to specify what data column to show

- Value for each axis is defined using expression property

- Simple Column Name

[plot].XAxis.Expression = Document.ActiveDataTableReference.Columns[0].NameEscapedForExpression

- Specifying Categorical and Continuous Axis

[plot].XAxis.Expression = “<[Items]>” [plot].XAxis.Expression = “[Items]”

- Multiple Columns

[plot].YAxis.Expression = “[At Bats], [League]”

- Use ‘Column Names’

[plot].XAxis.Expression = “<[Axis.Default.Names]>”

Visualization Properties

```
From Spotfire.Dxp.Application.Visuals import ScatterPlot, IndividualScalingMode, AxisRange,
AxisTransformType, TrellisMode

scatterPlot = sPlot.As[ScatterPlot]()


scatterPlot.YAxis.IndividualScaling = 1;
#scatterPlot.YAxis.IndividualScalingMode =  IndividualScalingMode.Trellis
scatterPlot.XAxis.TransformType = AxisTransformType.Log10
scatterplot.XAxis.Reversed = True
scatterPlot.Xaxis.ManualZoom = not (scatterPlot.XAxis.ManualZoom)
```

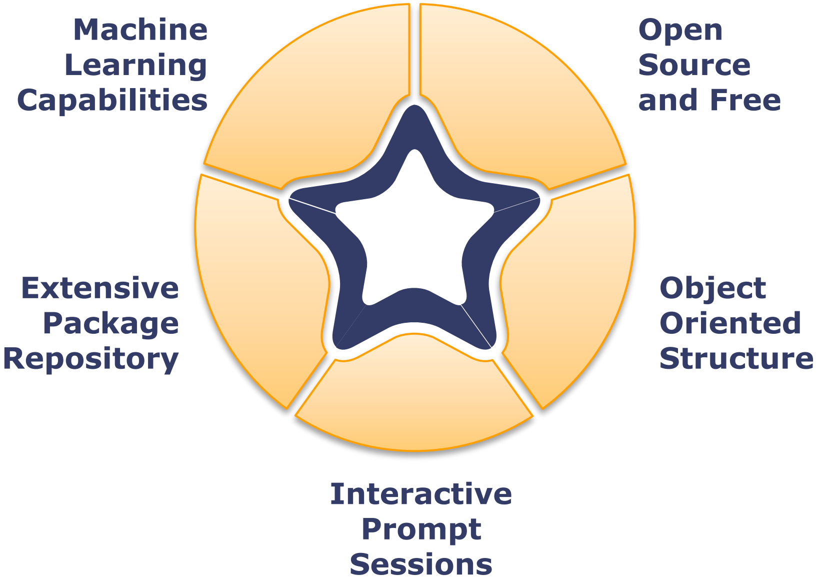
R and TERR

Introduce R

Statistical programming language

Advanced data programming capabilities

Used by statisticians world-wide

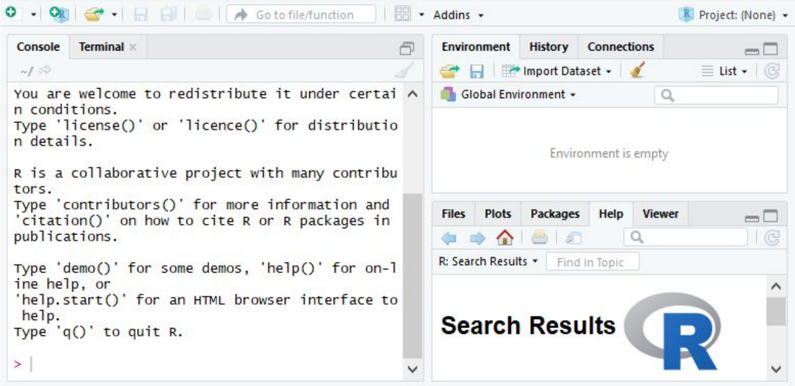

Machine Learning Capabilities
Open Source and Free
Extensive Package Repository
Object Oriented Structure
Interactive Prompt Sessions

R Environment

An integrated suite of software facilities for data calculation and graphical display

R console: command line interface

RStudio: IDE dedicated for R development

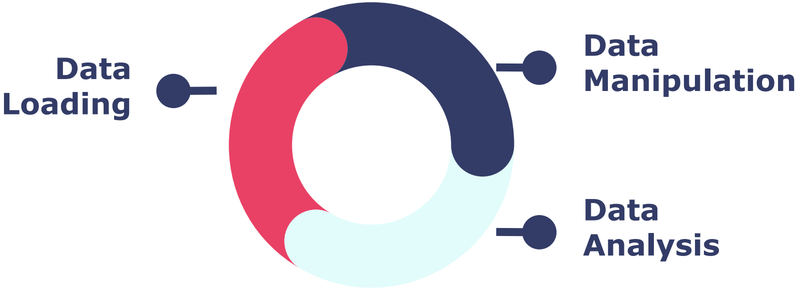


```
You are welcome to redistribute it under certai
n conditions.
Type 'license()' or 'licence()' for distributio
n details.

R is a collaborative project with many contribu
tors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in
publications.

Type 'demo()' for some demos, 'help()' for on-l
ine help, or
'help.start()' for an HTML browser interface to
 help.
Type 'q()' to quit R.

> |
```



Data Loading

Data Manipulation

Data Analysis

Limitations of R

Built as an academic tool for research and teaching

R is inefficient for enterprise usage

Restrictive open source license – GPL

No commercial-grade vendor support

Slower performance with big data and less memory efficient

Spotfire Enterprise Runtime for R

Enterprise-grade analytical engine compatible with R

Created with wide range of built-in methods

Designed for R language compatibility which is extensible through R community packages

Extends the reach of R in enterprise with better performance and commercial embeddable

Develop code in open source R and deploy on a commercially supported and robust platform

Leverage TERR Capabilities in Spotfire

Use in Spotfire Analyst

- In-built functions
- Expression functions
- Data functions etc.

Run on Local Engine or TERR

- Provided with Spotfire installation
- Separately install on the Spotfire server for better performance

Possible use-case scenarios of TERR

- Interactivity in ad-hoc analysis, for example, applying k-means clustering
- In complex guided analysis, for example, discover factors to predict causes of fraud using predictive modeling tools

TERR Tools

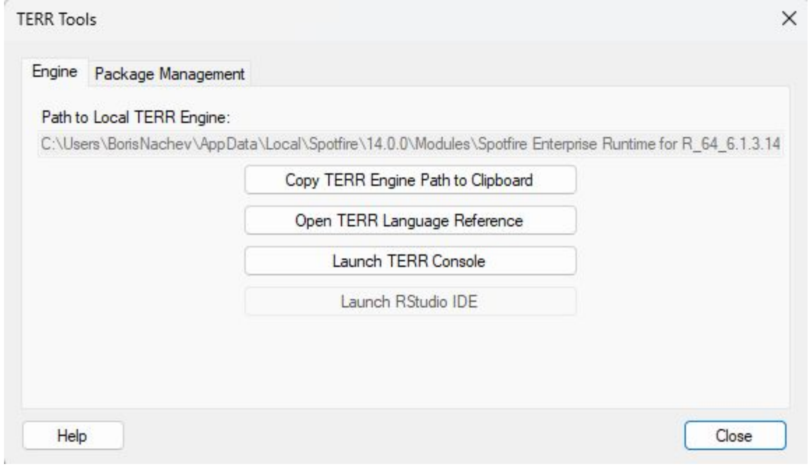
Must have TERR extension license to use TERR tools

Used to launch RStudio for script authoring

Launch TERR console

Accessing TERR language reference

Download and install CRAN packages



TERR Tools

Path to Local TERR Engine:
C:\Users\BorisNachev\AppData\Local\Spotfire\14.0.0\Modules\Spotfire Enterprise Runtime for R_64_6.1.3.14

TERR and TERR service

TERR

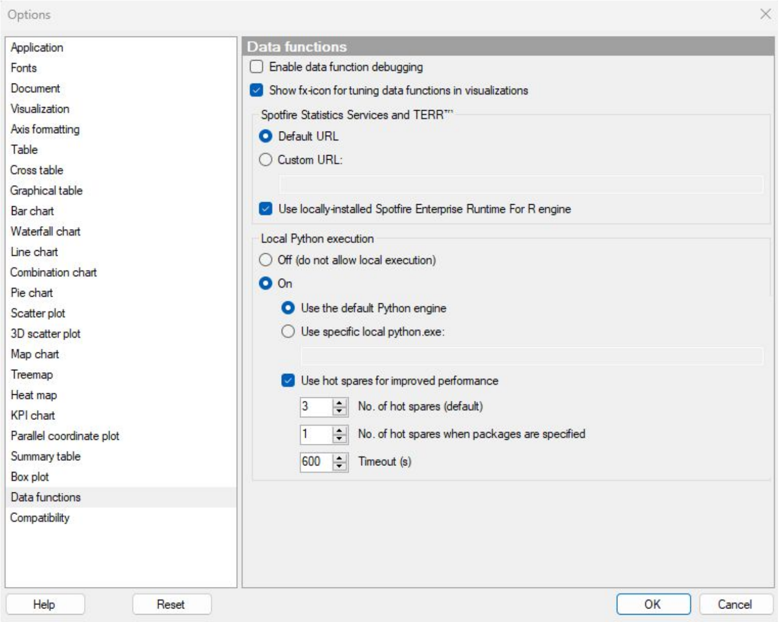
Can create analyses that use TERR data functions or TERR custom expressions to run locally

In Spotfire Analyst, from the menu, click Tools > Options. In the Options dialog, from the left list, select Data Functions.

TERR Service

Installed on a node for your Spotfire Server

Can share analysis created using TERR data functions with users of the Spotfire web client



Options

Data functions

☐ Enable data function debugging

☑ Show fx-icon for tuning data functions in visualizations

Spotfire Statistics Services and TERR™

● Default URL

○ Custom URL:

☑ Use locally-installed Spotfire Enterprise Runtime For R engine

Local Python execution

○ Off (do not allow local execution)

● On

● Use the default Python engine

○ Use specific local python.exe:

☑ Use hot spares for improved performance

3 No. of hot spares (default)

1 No. of hot spares when packages are specified

600 Timeout (s)

R Code in Spotfire Expressions

Calculated columns
Transformations
Custom expressions

In-line expressions
Expression functions

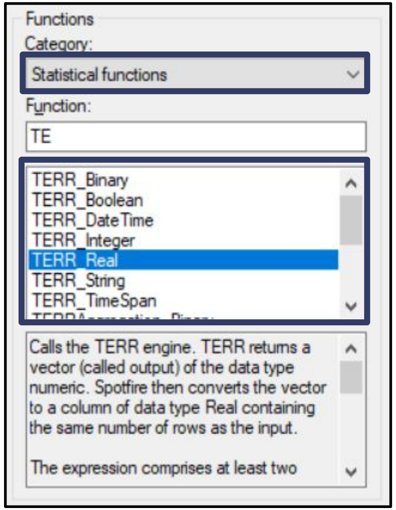
Expressions control various aspects

Use R code in addition to out-of-the-box functions

Built complex expressions using R
Leverage capabilities of TERR engine for calculation

In-line TERR Expressions

- To create an in-line TERR expression
  - Select appropriate TERR function
    - Category – statistical functions
    - Look for return data type
  - Insert function
- Uses – short, one-line R scripts
- Expressions executed on TERR engine
- Expression elements has two parts



Statistical functions

TE

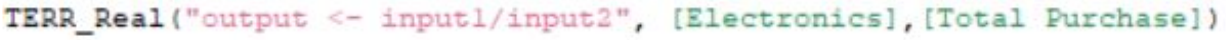
- TERR_Binary
- TERR_Boolean
- TERR_DateTime
- TERR_Integer
- TERR_Real
- TERR_String
- TERR_TimeSpan

Calls the TERR engine. TERR returns a vector (called output) of the data type numeric. Spotfire then converts the vector to a column of data type Real containing the same number of rows as the input.

The expression comprises at least two

R script enclosed in double quotes

Inputs to the R scripts as arguments



```
TERR_Real("output <- input1/input2", [Electronics],[Total Purchase])
```

TERR Expression Functions

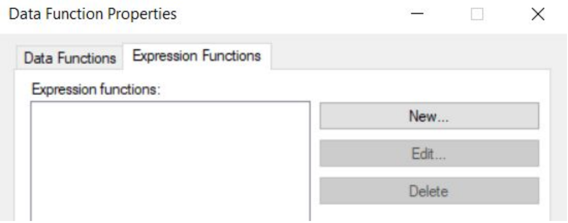
Create custom expression function for complex TERR scripts

TERR expression function is saved with the analysis file

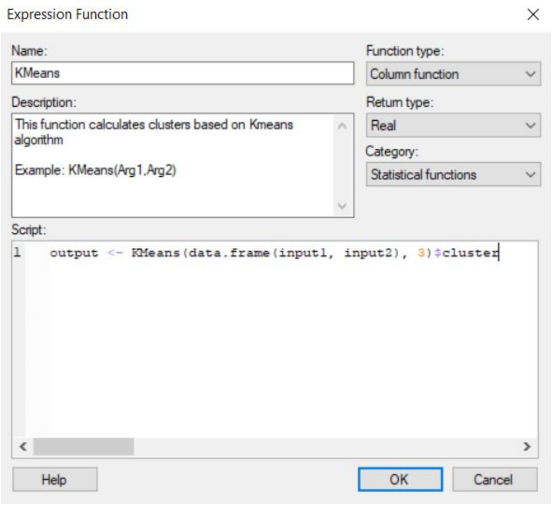
It is invoked as any other function in Spotfire

Advantage: use same R script at multiple locations

To insert/edit expression function, Data > Data Function Properties > Expression Functions tab



Data Function Properties



Expression Function

Name: KMeans

Function type: Column function

Description: This function calculates clusters based on Kmeans algorithm

Example: KMeans(Arg1,Arg2)

Return type: Real

Category: Statistical functions

Script:

```
output <- KMeans(data.frame(input1, input2), 3)$cluster
```

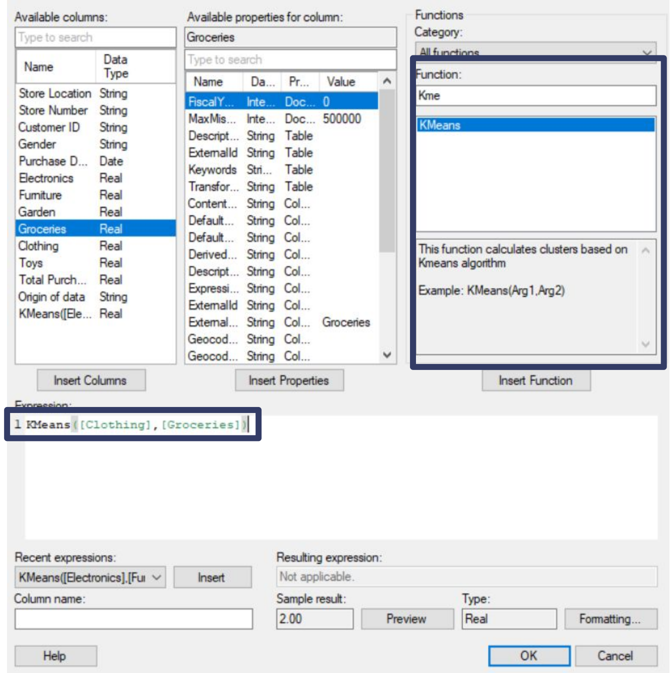
Insert Functions in Expressions

Open expression builder dialog to create a calculated column or custom expression

Search in functions section by typing in function name under specified category

Use Insert Function button provided under functions section to insert function

Check the description to understand what arguments to pass in the function



Available columns:

Type to search

Name	Data Type
Store Location	String
Store Number	String
Customer ID	String
Gender	String
Purchase D...	Date
Electronics	Real
Furniture	Real
Garden	Real
Groceries	Real
Clothing	Real
Toys	Real
Total Purch...	Real
Origin of data	String
KMeans([Ele...	Real

Available properties for column:

Groceries

Type to search

Name	Da...	Pr...	Value
FiscalY...	Inte...	Doc...	0
MaxMis...	Inte...	Doc...	500000
Descript...	String	Table	
ExternalId	String	Table	
Keywords	Stri...	Table	
Transfor...	String	Table	
Content...	String	Col...	
Default...	String	Col...	
Default...	String	Col...	
Derived...	String	Col...	
Descript...	String	Col...	
Expressi...	String	Col...	
ExternalId	String	Col...	
External...	String	Col...	Groceries
Geocod...	String	Col...	
Geocod...	String	Col...	

Functions

Category: All functions

Function: Kme

KMeans

This function calculates clusters based on Kmeans algorithm

Example: KMeans(Arg1,Arg2)

Expression:

```
KMeans([Clothing],[Groceries])
```

Recent expressions: KMeans([Electronics],[Fu

Column name:

Resulting expression: Not applicable.

Sample result: 2.00

Type: Real

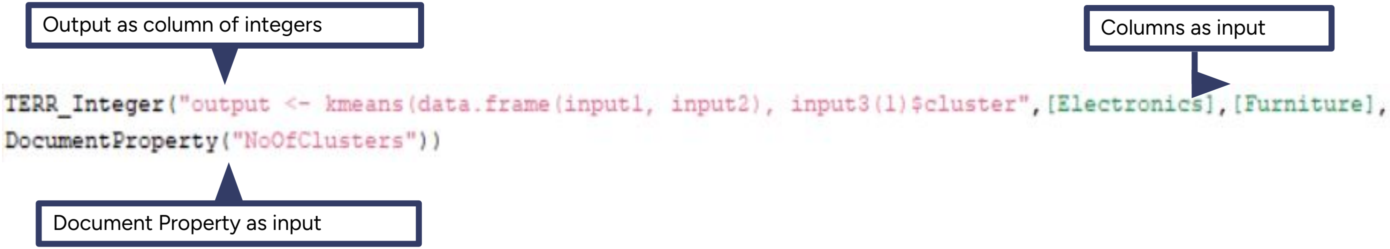
Input Output Considerations

Input

- Column, Document Property, Static value
- At least one input required
- Must be variables named input1, input2, etc.

Output

- Must be assigned to variable named output
- Single column with data type specified by function name, e.g. TERR_Integer



Output as column of integers

Columns as input

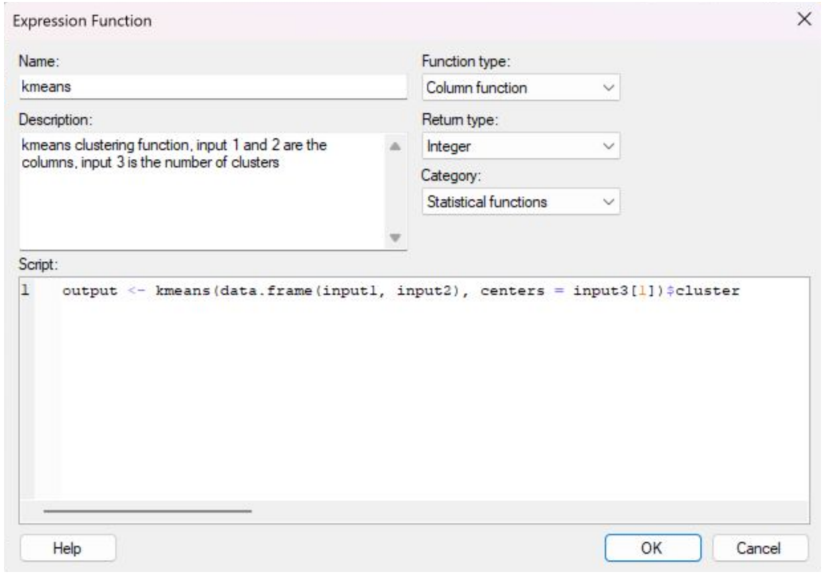
```
TERR_Integer("output <- kmeans(data.frame(input1, input2), input3(1)$cluster",[Electronics],[Furniture],
DocumentProperty("NoOfClusters"))
```

Document Property as input

Exercise - R expressions and expression functions

1. Load the data file “Exercise Random Forest Dataset.csv” into Spotfire
2. Create a scatterplot with a marker by row number
3. Choose a variable on both the x and y axis (eg. Windspeed and Previous Delay)
4. Now use the custom expression for the colour by and use: <TERR_Integer("output <- kmeans(data.frame(input1,input2),centers = 5)$cluster",[WindSpeed_BRU],[PrevDelay])>
5. See how this TERR Expression allows you to do Kmeans clustering.
6. For reusability and easy we can create a expression function

   Go to data data function properties, select the expression function tab and add a new expression function as shown on the screenshot to the right
7. Now add a new scatter plot and now use the newly created kmeans function on the colour by axis and provide the 3 inputs.



Expression Function

Name: kmeans

Function type: Column function

Description: kmeans clustering function, input 1 and 2 are the columns, input 3 is the number of clusters

Return type: Integer

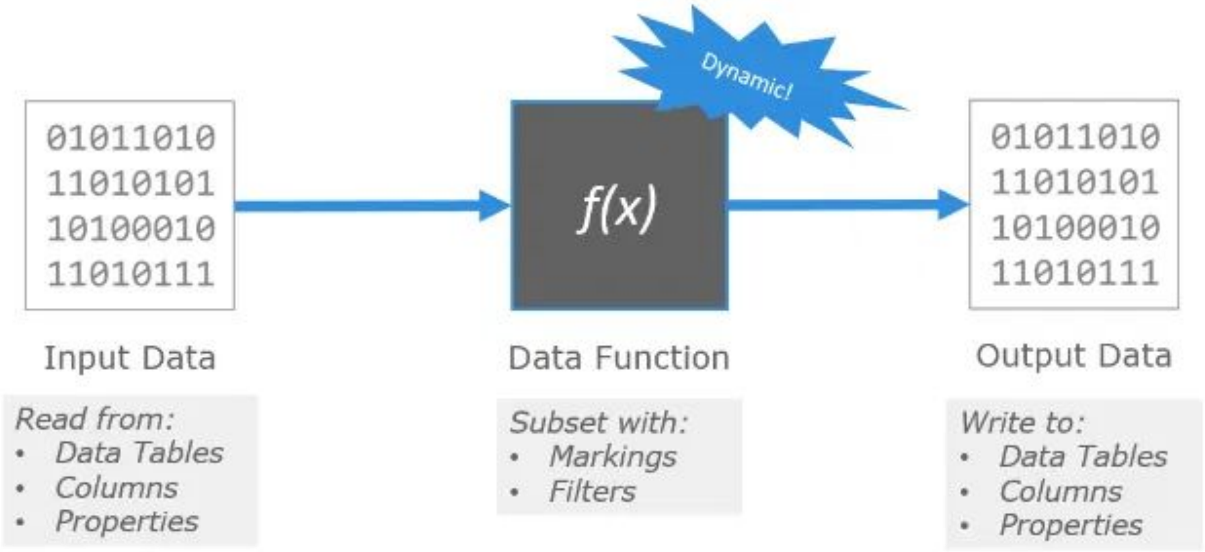
Category: Statistical functions

Script:

```
output <- kmeans(data.frame(input1, input2), centers = input3[1])$cluster
```

Data Functions

What is a Data Function?



Input Data

Read from:

- *Data Tables*
- *Columns*
- *Properties*

Data Function

Subset with:

- *Markings*
- *Filters*

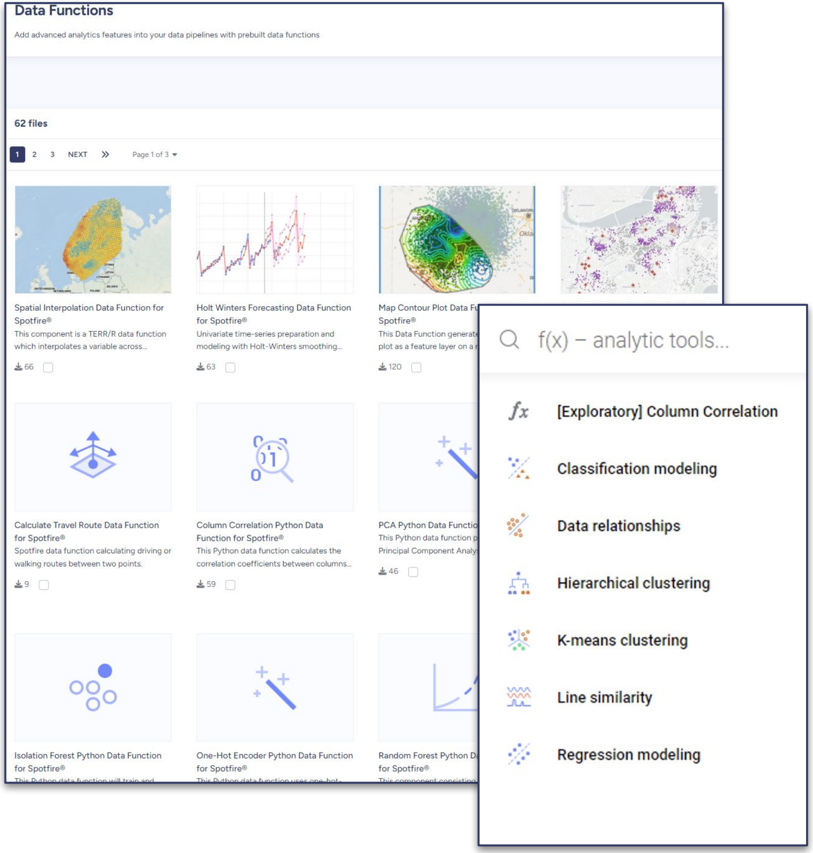
Output Data

Write to:

- *Data Tables*
- *Columns*
- *Properties*

Spotfire Data Functions

- Data Functions are scripts created using Python, R, TERR, Statistica or other languages that apply calculations to data tables
- Unique and differentiated **integration** of data science algorithms with data visualizations
- **Developers** can create data functions that can easily be re-used by others
- **Non-developers** use pre-built data functions–many are available for download from the community



Data Functions

Add advanced analytics features into your data pipelines with prebuilt data functions

62 files

1 2 3 NEXT » Page 1 of 3

Spatial Interpolation Data Function for Spotfire®
This component is a TERR/R data function which interpolates a variable across...
66 □

Holt Winters Forecasting Data Function for Spotfire®
Univariate time-series preparation and modeling with Holt-Winters smoothing...
63 □

Map Contour Plot Data Fu
Spotfire®
This Data Function generate
plot as a feature layer on a r
120 □

Calculate Travel Route Data Function for Spotfire®
Spotfire data function calculating driving or walking routes between two points.
9 □

Column Correlation Python Data Function for Spotfire®
This Python data function calculates the correlation coefficients between columns...
59 □

PCA Python Data Functio
This Python data function p
Principal Component Analy
46 □

Isolation Forest Python Data Function for Spotfire®
This Python data function will train and

One-Hot Encoder Python Data Function for Spotfire®
This Python data function uses one-hot-

Random Forest Python D
for Spotfire®
This component consisting

Register Data Functions

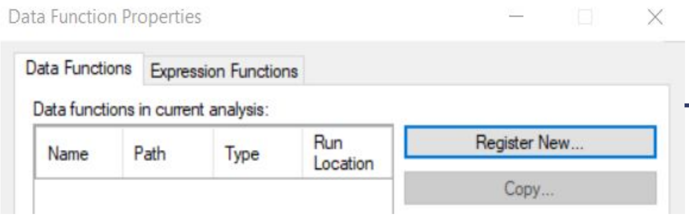
Define script, input and output

Open Register Data Functions dialog

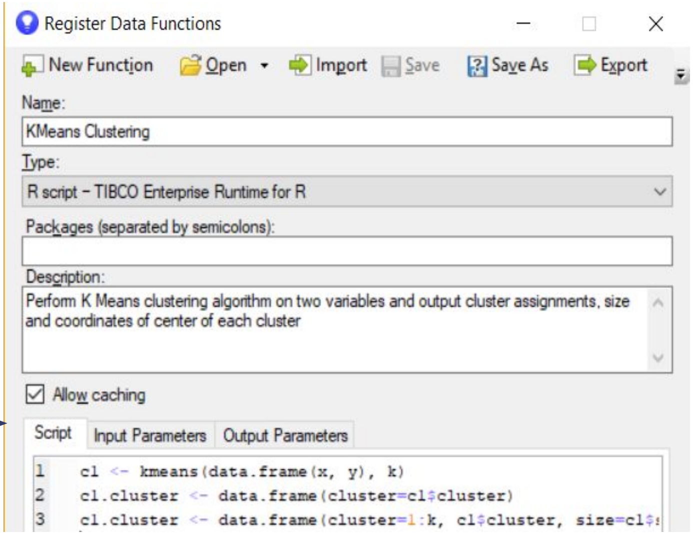
- Tools -> Register Data Functions
- Data Function Properties -> Register New

Script type

- R script: TERR



Data functions in current analysis:



Name:
KMeans Clustering

Type:
R script – TIBCO Enterprise Runtime for R

Packages (separated by semicolons):

Description:
Perform K Means clustering algorithm on two variables and output cluster assignments, size and coordinates of center of each cluster

☑ Allow caching

```
cl <- kmeans(data.frame(x, y), k)
cl.cluster <- data.frame(cluster=cl$cluster)
cl.cluster <- data.frame(cluster=1:k, cl$cluster, size=cl$s
```

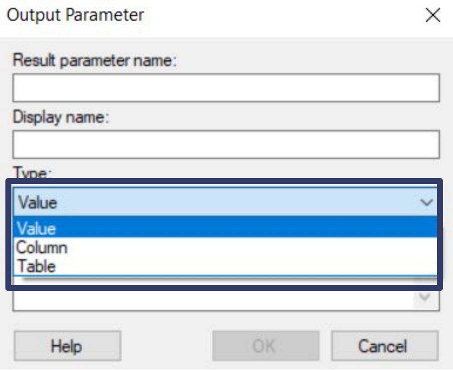
Input and Output Parameters

Define input/output parameters used in script

Names must match with parameters used in script

Set allowed data types for inputs

Type of output parameter: value, column and table

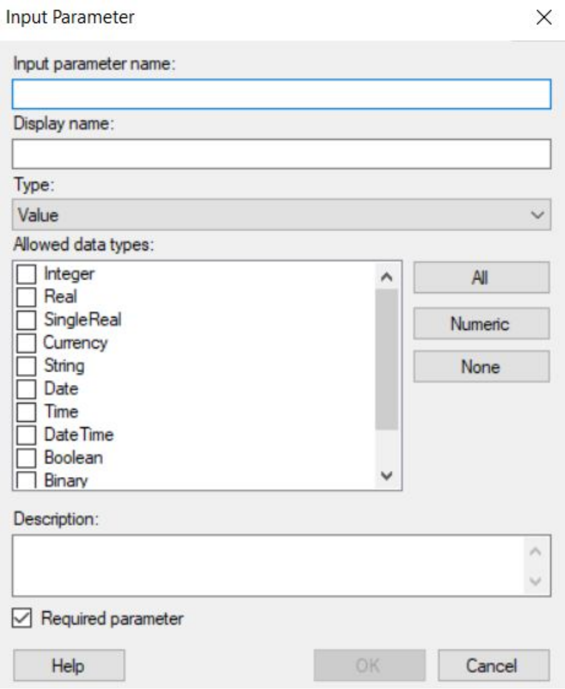


Output Parameter

Result parameter name:

Display name:

Type: Value



Input Parameter

Input parameter name:

Display name:

Type: Value

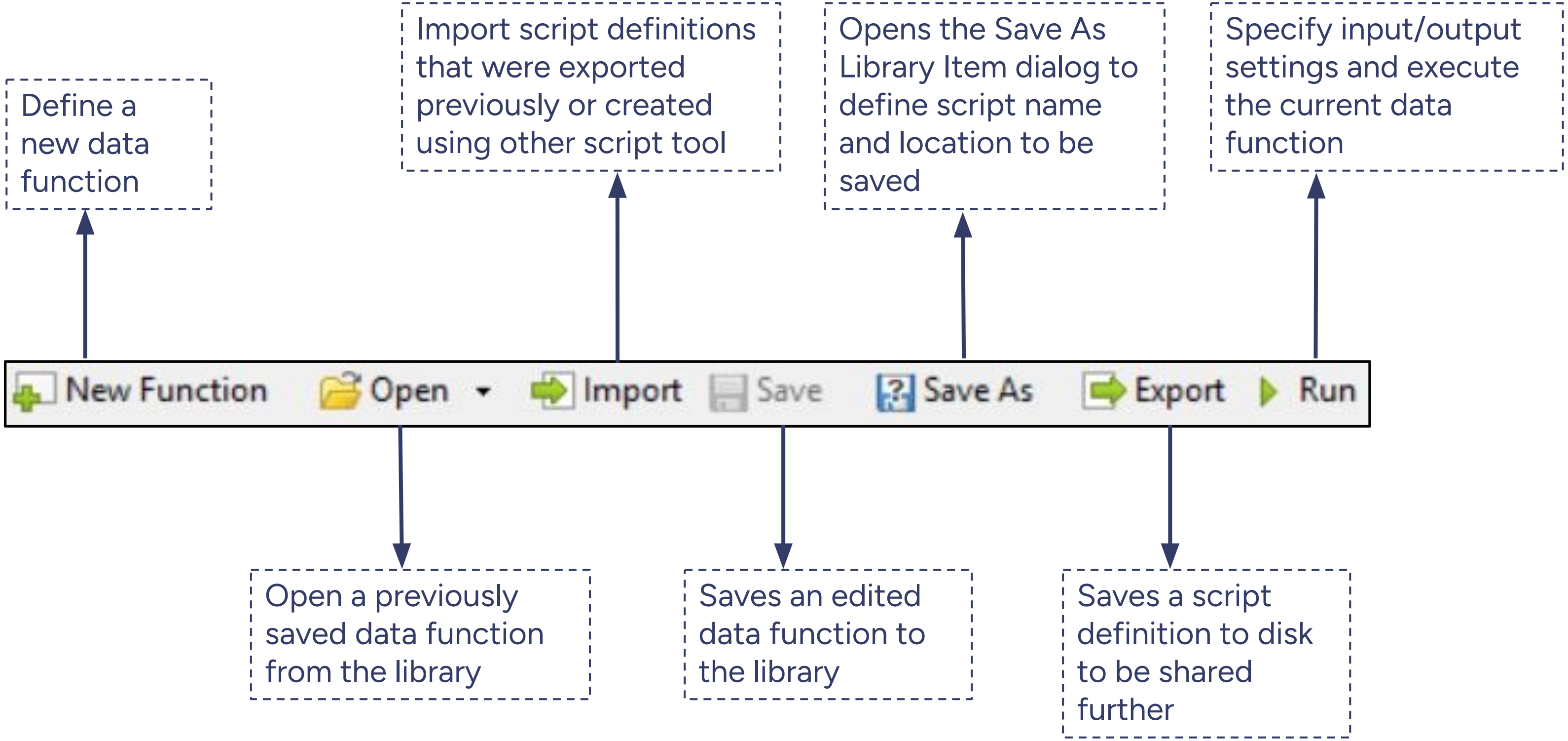
Allowed data types:

- □ Integer
- □ Real
- □ SingleReal
- □ Currency
- □ String
- □ Date
- □ Time
- □ DateTime
- □ Boolean
- □ Binary

Description:

☑ Required parameter

Save, Share and Reuse Data Functions



Define a new data function

Open a previously saved data function from the library

Import script definitions that were exported previously or created using other script tool

Saves an edited data function to the library

Opens the Save As Library Item dialog to define script name and location to be saved

Saves a script definition to disk to be shared further

Specify input/output settings and execute the current data function

Apply Data Functions

Insert data function

- From library
- From file (.sfd)

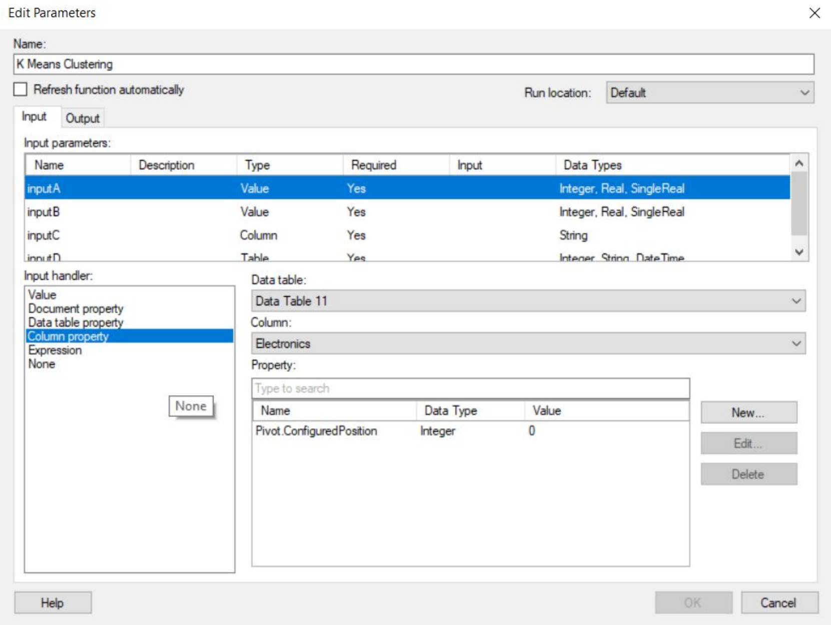
From library

- Search using keywords

Edit parameters

- Choose from available input handlers
- Input handlers vary based on selected type while registering

Can be configured to refresh automatically



Edit Parameters

Name: K Means Clustering

☐ Refresh function automatically

Run location: Default

Input parameters:

Name	Description	Type	Required	Input	Data Types
inputA		Value	Yes		Integer, Real, SingleReal
inputB		Value	Yes		Integer, Real, SingleReal
inputC		Column	Yes		String
inputD		Table	Yes		Integer, String, DateTime

Input handler:

- Value
- Document property
- Data table property
- Column property
- Expression
- None

Data table: Data Table 11

Column: Electronics

Property:

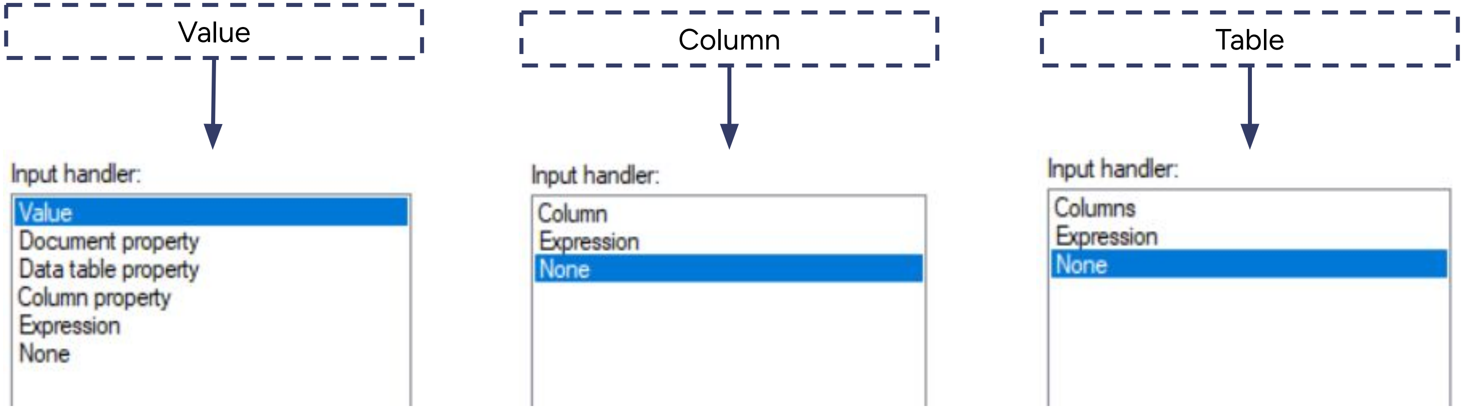
Name	Data Type	Value
Pivot.ConfiguredPosition	Integer	0

Input and Output Handlers

Input handler availability depends on type of selected input parameter

Map values from the analysis to inputs

Output handler defines how outputs are saved in analysis



Value

Input handler:

Value
Document property
Data table property
Column property
Expression
None

Column

Input handler:

Column
Expression
None

Table

Input handler:

Columns
Expression
None

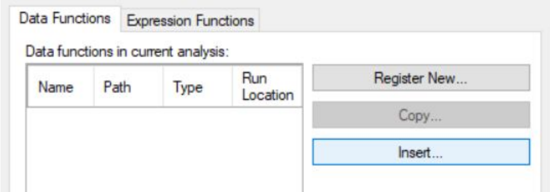
Modify Data Functions

Data -> Data Function Properties

- Edit script, parameters
- Copy, insert and delete
- Save to file, library, sync and refresh

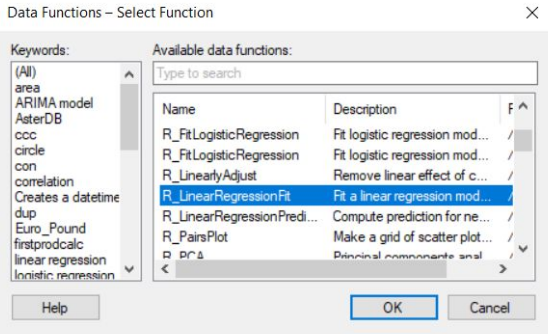
Insert data functions

- Select data function from available data functions list
- Select input and output parameters
    - ○ Limit data function by filtering or marking



Data functions in current analysis:

Name	Path	Type	Run Location



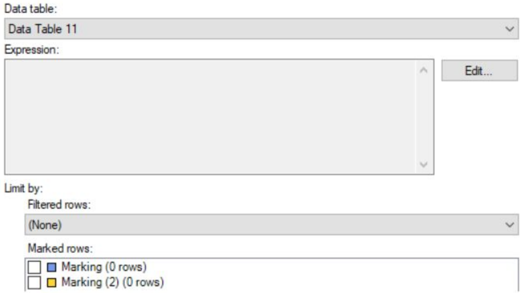
Keywords:

(All)
area
ARIMA model
AsterDB
ccc
circle
con
correlation
Creates a datetime
dup
Euro_Pound
firstprodcalc
linear regression
logistic regression

Available data functions:

Type to search

Name	Description
R_FitLogisticRegression	Fit logistic regression mod...
R_FitLogisticRegression	Fit logistic regression mod...
R_LinearlyAdjust	Remove linear effect of c...
R_LinearRegressionFit	Fit a linear regression mod...
R_LinearRegressionPredi...	Compute prediction for ne...
R_PairsPlot	Make a grid of scatter plot...
R_PCA	Principal components anal...



Data table: Data Table 11

Expression:

Limit by:

Filtered rows: (None)

Marked rows:

□ Marking (0 rows)
□ Marking (2) (0 rows)

Data Function- Exercise

1. Load the dataset “Exercise Random Forest Dataset.csv into Spotfire
2. Go to Tools => Python Tools and select the package management tab. Search for available packages and install “scikit-learn”
3. Add the datafunction (“[Modeling] Random Forest (Python).sfd”) e.g. by using drag & drop
4. The Fx flyout pops-up. Use the flyout to predict the IsDelayed field using a Classification method
5. Notice you will have to trust the datafunction. Trust the datafunction and select reload.
6. Check the data canvas to view the datafunction and it’s inputs and outputs.
7. Look at the additional tables that have been created.
8. Now create a visual that shows the top 5 features of most importance.
9. Also create a visual that shows the predicted vs the isDelayed field to check the accuracy

Data Function- Exercise

1. Open your analysis file
2. Create a data function
3. Create a data function using copilot

Some thoughts to add

Data functions on the community?

More on the data function flyout

Spotfire DSML asset