Advanced Functions

Overview

top
Advanced Functions are used to transform results by applying post processing calculations to the initial query results. These functions are database independent as they are performed by Java code, rather than being part of the SQL query.

Available Functions

Yellowfin comes with a set of pre-defined functions. However, your administrator may add in additional functions specifically for your organisation or reporting needs. Expand the following to see the lists of functions available:

Accumulative Percentage	Displays a running percentage to total for the values in the field. A maximum of 100% will be displayed.
Accumulative Percentage with Cut-off	Displays a running percentage of total for the values in the field. A maximum % cut-off can be specified, or set to user prompt.
Accumulative Total	Displays out a running total for the values in the field.
Ascending Rank	Displays a rank based on the values in the field. The highest value returned will be displayed as a 1. Used where the preferable result is a higher value. Eg. Profit.
Bottom 10 Rank	Displays a rank based on the values in the field. The bottom 10 values (lowest) are returned.
Bottom N Rank	Displays a rank based on the values in the field. The bottom N values (lowest) are returned– user is prompted to define number to return.
Delta from Last	Displays the calculated change between consecutive rows.
Delta from Last N	Displays the calculated change between the current row and row - N.
Descending Rank	Displays a rank based on the values in the field. The lowest value returned will be displayed as a 1. Used where the preferable result is a lower value. Eg. Expenses.
Deviation From Previous	Displays the deviation from the previous value. The formula used is: `(current value - previous value) / previous value`.
Difference of Columns	Displays the result of the difference of two selected columns.
Division By Column	Displays the result of the division of two selected columns, where the current column is the denominator and the numerator is defined in the setup.
Division of Columns	Displays the result of the division of two selected columns, where the current column is the numerator and the denominator is defined in the setup.
Filter Percentage of Total	Displays the percentage of total for the values in the field, where the percentages are either above or below a specified threshold depending on the direction defined.
Grouped Percentage of Total	Returns top percentage of the specified field/column.
Multiplication of Columns	Displays the multiplication of two selected columns.
Natural Logarithm	Gives the base e logarithm of the values of a given field.

Null To Zero	Displays all NULL values found in the field with zero (0).
Percentage Against Absolute Maximum	Returns percentage of selected field according to an absolute maximum value.
Percentage Against Column	Creates a percentage ration of values in the selected column compared to another column.
Percentage Against Maximum	Returns the % of the attribute when compared to the maximum value of the attribute within the dataset.
Percentage Change Against Column	Displays the percentage of change of the selected field against a value in another column.
Percentage of Initial Value	Displays the percentage of a value compared to the initial value in the column
Percentage of Total	Returns the % of the attribute when compared to the total summed value of the attribute for the entire dataset.
Remove Values	Remove values below above or below a defined threshold.
Sum of Columns	Returns the sum of two selected columns.
Top 10 Rank	The top 10 values (highest) are returned.
Top N Rank	The top N values (highest) are returned – user is prompted to define number to return.
Top N With Ties	Returns top values for the selected field with provision for tied values. This means that if there are multiple records per ranking it will restrict it to N total rankings.
Top/Bottom N Percentage of Total	Display records that make up either the top or bottom N % of total.
Truncate Data Set	Removes N rows from either the top or bottom of the dataset.

Date Extrapolation	This extends the date range displayed in the table by a defined number of periods. Both the number of periods, and the units can be defined.
Date Period Extractor	This extracts a date or timestamp component from a given date.
Days Between Date	The days between the date selected and another date column on the report.
Days to Now	The days between the date selected and the current date. (age in days)
Months Between Date	The months between the date selected and another date column on the report.
Months to Now	The months between the date selected and the current date. (age in months)
Weeks Days Between	The week days between the date selected and another date column on the report.
Years Between Date	The years between the date selected and another date column on the report.
Years to Now	The years between the date selected and the current date. (age in years)

Decile	Decile divides the rows returned into 10 equal parts, and assigns a value of 1 to 10, based upon its rank to the highest value. Deciles are used as a measure of dispersion.
Deviation	The number of deviations from the mean.
Linear Regression	A linear trendline is a best-fit straight line that is used with simple linear data sets. Your data is linear if the pattern in its data points resembles a line. A linear trendline usually shows that something is increasing or decreasing at a steady rate.
Mean	The arithmetic mean (or simply the mean) of a list of numbers is the sum of all the members of the list divided by the number of items in the list.
Median	The median is described as the number separating the higher half of a sample, a population, or a probability distribution, from the lower half.
Mode	The mode is the value that occurs the most frequently in a data set

Moving Average	A moving average trendline smoothes out fluctuations in data to show a pattern or trend more clearly. A moving average uses a specific number of data points (set by the Period option), averages them, and uses the average value as a point in the line. If Period is set to 2, for example, then the average of the first two data points is used as the first point in the moving average trendline. The average of the second and third data points is used as the second point in the trendline, and so on.
Moving Total	The total over the last N periods.
Naïve Forecasting	A naive forecasting model is a special case of the moving average forecasting model where the number of periods used for smoothing is 1. Therefore, the forecast for a period, t, is simply the observed value for the previous period, t-1. Due to the simplistic nature of the naive forecasting model, it can only be used to forecast up to one period in the future. It is not at all useful as a medium-long range forecasting tool.
Polynomial Regression	A polynomial trendline is a curved line that is used when data fluctuates. It is useful, for example, for analysing gains and losses over a large data set. The order of the polynomial can be determined by the number of fluctuations in the data or by how many bends (hills and valleys) appear in the curve. An Order 2 polynomial trendline generally has only one hill or valley. Order 3 generally has one or two hills or valleys. Order 4 generally has up to three.
Quartile	Quartile divides the rows returned into 4 equal parts, and assigns a value of 1 to 4, based upon its rank to the highest value. Quartiles are used as a measure of dispersion.
Standard Deviation	The standard deviation is a measure of the dispersion of a set of values. It can apply to a probability distribution, a random variable, a population or a multiset.
Standard Deviation from Mean	This function calculates how many standard deviations is each value away from the selected field's mean. Also includes options to remove values based a specified threshold.
Standard Score	The standard score indicates how many standard deviations an observation is above or below the mean. It allows comparison of observations from different normal distributions, which is done frequently in research.
Stepped Regression	Implements a step function against a given column.
Trend	Displays a trend metric against an extended period.
Triple Exponential Smoothing	Returns triple exponential smoothing result based in the input data set.
Variance	Returns the difference between the data sets.
Weighted Moving Average	Returns a moving average that is weighted so that the more recent the value, the more weight is applied to it.

Concatenate	Joins two columns into one text string.

Data Conversion allows you to adjust results once they've returned from the database. For example you may wish to convert a currency value which is stored in the database from a full currency value to a (‘000) where the value is divided by 1000. This transformation can be achieved using the data conversion.

Custom Advanced Function Example - R Integration

We have included an example of a custom advanced function, which enables the use of R within a report.

Writing R-Scripts

For Yellowfin to understand and execute R script a slightly different structure will be used in the script discussed below.

Consider a sample script called <`R_file_name`>`.R`. The input parameters passed from Yellowfin will be available in a file called <`R_file_name`>`.R.input.csv` . Post processing, the R-script should write the results (only one column) back to <`R_file_name`>`.R.result.csv` .

Below is a sample R-Script for Neural Networks. Copy this script and make sure it runs without any errors in R.

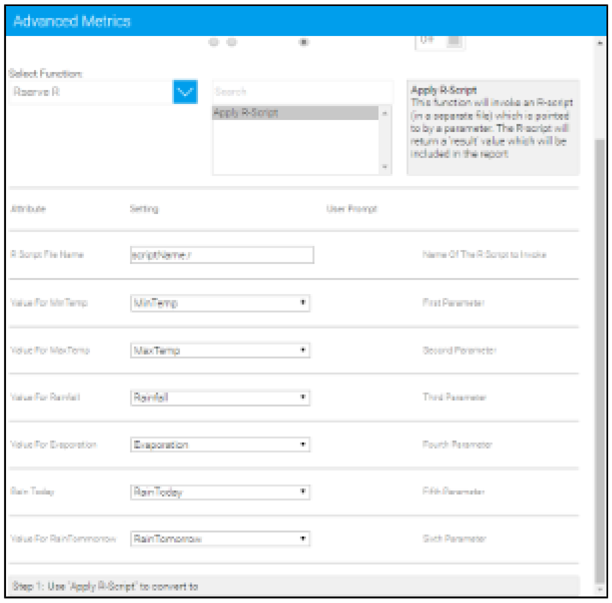
Sample R-Script : neural-net-script.R

```
setwd("C:/R/R-3.2.3/bin/x64")
library(rattle)   # To access the weather dataset and utility commands.
library(magrittr) # For the %>% and %<>% operators.
building <- TRUE
scoring  <- ! building
# A pre-defined value is used to reset the random seed so that results are repeatable.
crv$seed <- 42
# Load the data.
rPATH  <- Sys.getenv("RSCRIPT_PATH")
rINPUT <- paste0(rPATH ,"/neural-net-script.r.input.csv")
rOUTPUT <- paste0(rPATH ,"/neural-net-script.r.result.csv")
dataset <- read.csv(file=rINPUT, header=FALSE, sep=",")
# Note the user selections.
# Build the training/validate/test datasets.
set.seed(crv$seed)
crs$nobs <- nrow(dataset) # 366 observations
crs$sample <- crs$train <- sample(nrow(dataset), 0.7*crs$nobs) # 256 observations
crs$validate <- sample(setdiff(seq_len(nrow(dataset)), crs$train), 0.15*crs$nobs) # 54 observations
crs$test <- setdiff(setdiff(seq_len(nrow(dataset)), crs$train), crs$validate) # 56 observations
# The following variable selections have been noted.
crs$input <- c("V1", "V2", "V3", "V4","V5")
crs$target  <- "V6"
#============================================================
# Neural Network
#============================================================

# Build a neural network model using the nnet package.
library(nnet, quietly=TRUE)
# Build the NNet model.
set.seed(199)
crs$nnet <- nnet(as.factor(V6) ~ .,data=dataset[crs$sample,c(crs$input, crs$target)],size=10, skip=TRUE,
MaxNWts=10000, trace=FALSE, maxit=100)
#============================================================
# Score a dataset.
#============================================================
# Obtain probability scores for the Neural Net model on weather.csv [validate].
#crs$pr <- predict(crs$nnet, newdata=dataset[crs$validate, c(crs$input)], type="class")
#crs$pr <- predict(crs$nnet, newdata=dataset[crs$validate, c(crs$input)], type="class")
crs$pr <- predict(crs$nnet, newdata=dataset, type="class")
write.table(crs$pr, file=rOUTPUT, row.names=FALSE, col.names = FALSE)
```

Calling R-Script

Once installed, you will be able to use this function through the Advanced Function menu.



Select Function: Rserve R

Apply R-Script
This function will invoke an R-script (in a separate file) which is pointed to by a parameter. The R-script will return a 'result' value which will be included in the report

Attribute	Setting	User Prompt
R Script File Name	scriptName.r	Name Of The R-Script to Invoke
Value For MinTemp	MinTemp	First Parameter
Value For MaxTemp	MaxTemp	Second Parameter
Value For Rainfall	Rainfall	Third Parameter
Value For Evaporation	Evaporation	Fourth Parameter
Rain Today	RainToday	Fifth Parameter
Value For RainTomorrow	RainTomorrow	Sixth Parameter

Step 1: Use 'Apply R-Script' to convert to

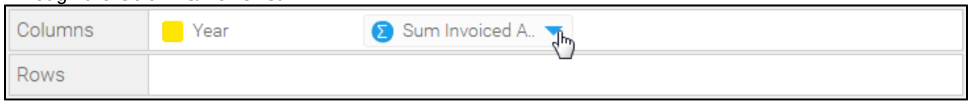
Applying a Function

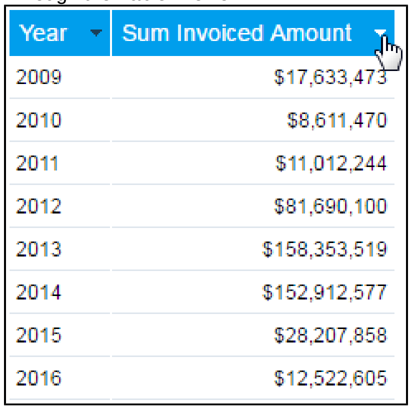
top

Report Builder - Data Step

To apply an advanced function, first you will need to ensure the field you wish to apply it to, has been added to your table. Often when applying functions, you may find you need two copies of the field in your table; one to display the original values (such as sales figures) and the second to have a function applied to it (for example, displaying the top 10 rank of sales figures).

To apply a function to a field in your table from the Data step of the report builder, complete the following:

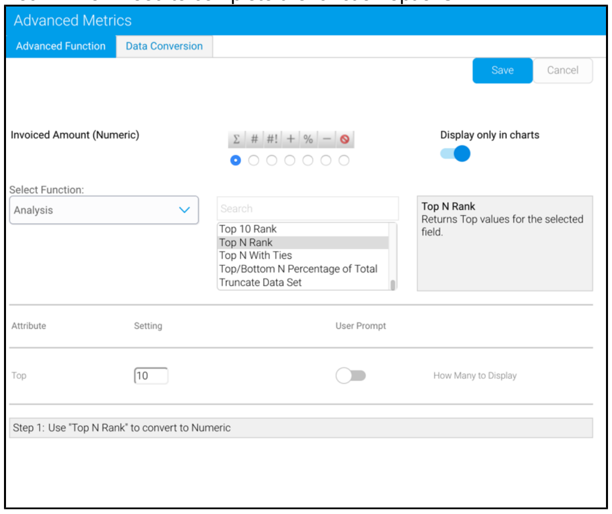
1. Open the field's drop down menu, in either of these locations:
   a. Through the Columns/Rows list
   

   b. Through the Table Preview
   

Year	Sum Invoiced Amount
2009	$17,633,473
2010	$8,611,470
2011	$11,012,244
2012	$81,690,100
2013	$158,353,519
2014	$152,912,577
2015	$28,207,858
2016	$12,522,605

2. Select the **Advanced Function** option from the list, then select Edit to open the Advanced Function display.

3. You will now need to complete the function options:



Invoiced Amount (Numeric)

Display only in charts

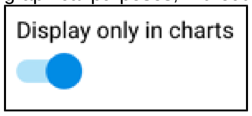
Analysis

Top N Rank
Returns Top values for the selected field.

Top 10 How Many to Display

Step 1: Use "Top N Rank" to convert to Numeric

   a. Apply the appropriate aggregation to the field. This ensures the function is applied on top of any aggregations necessary.
   b. Select which function type you wish to use, from **Analysis**, **Statistical**, and **Text**.
   c. Select the name of the function you wish to use from the list. Once selected, you will see a description of the function displayed next to it. Some functions require extra parameters, which will need to be defined once the function has been selected.
   d. You also have the option to set the function to only display on the Charts page. This hides the field on the report table, but you will still be able to see and edit it through the Column/Rows list. This allows you to create additional copies of a field to apply functions for graphical purposes, without cluttering your table with extra fields.

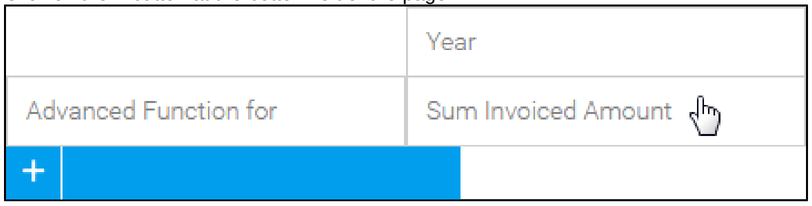


4. When defined, click **Save** to apply the function.

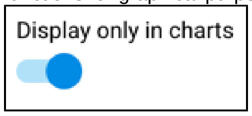
Report Builder - Charts Step

To apply a function to a field in your table from the Charts step of the report builder, complete the following:

1. Click on the + button at the bottom left of the page



2. Select **Advanced Function for** and choose the field you wish to use from the list, opening the Advanced Function display.
3. You will now need to complete the function options:
   a. Apply the appropriate aggregation to the field. This ensures the function is applied on top of any aggregations necessary.
   b. Select which function type you wish to use, from **Analysis**, **Statistical**, and **Text**.
   c. Select the name of the function you wish to use from the list. Once selected, you will see a description of the function displayed next to it. Some functions require extra parameters, which will need to be defined once the function has been selected.
   d. You also have the option to set the function to only display on the Charts page. This hides the field on the report table, but you will still be able to see and edit it through the Column/Row list on the Data step. This allows you to create additional copies of a field to apply functions for graphical purposes, without cluttering your table with extra fields.

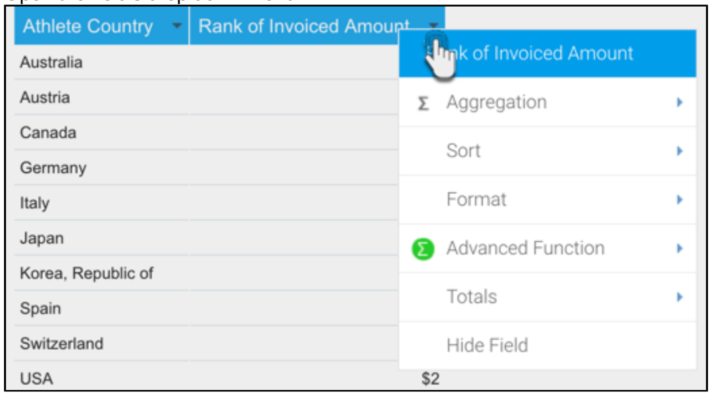


4. When defined, click **Save** to apply the function.

Report Builder - Design Step

To apply a function to a field in your table from the Design step of the report builder, complete the following:

1. Open the field's drop down menu.



2. Now select the **Advanced Function** option from the list, and then **Edit**, to open the Advanced Function display.
3. You will now need to complete the function options:
    a. Apply the appropriate aggregation to the field. This ensures the function is applied on top of any aggregations necessary.
    b. Select which function type you wish to use, from **Analysis**, **Statistical**, and **Text**.
    c. Select the name of the function you wish to use from the list. Once selected, you will see a description of the function displayed next to it. Some functions require extra parameters, which will need to be defined once the function has been selected.
    d. You also have the option to set the function to only display on the Charts page. This hides it on the report table, but you will still be able to see and edit the report field through the Column/Row list on the Data step. This allows you to create additional copies of a field to apply functions to for graphical purposes, without cluttering your table with extra fields.
4. When defined, click **Save** to apply the function.

Applying Data Conversion

top

Data in a field can be converted by completing the following:

1. Open the Advanced Function display, as outlined in the sections above.
2. Choose the ‘Data Conversion’ tab. This will provide you with the interface to use to select the conversion you wish to apply.
3. Choose the aggregation appropriate for you conversion.
4. Click **Add** button to select and apply a conversion – this will present you with a list of possible conversions for the data type you have selected. By default there is a java date converter and a Numeric divide converter (This lets you divide a value by 1000’s etc).
5. Follow the on screen instructions for the converter and click **Save**.
6. Note that you can add multiple converters to a data type if required by clicking the add link and creating a new type.

top