Define Metrics in Reports

Metrics are used to build reports. They are the numerical values displayed in reports. They represent aggregations of facts and attributes.

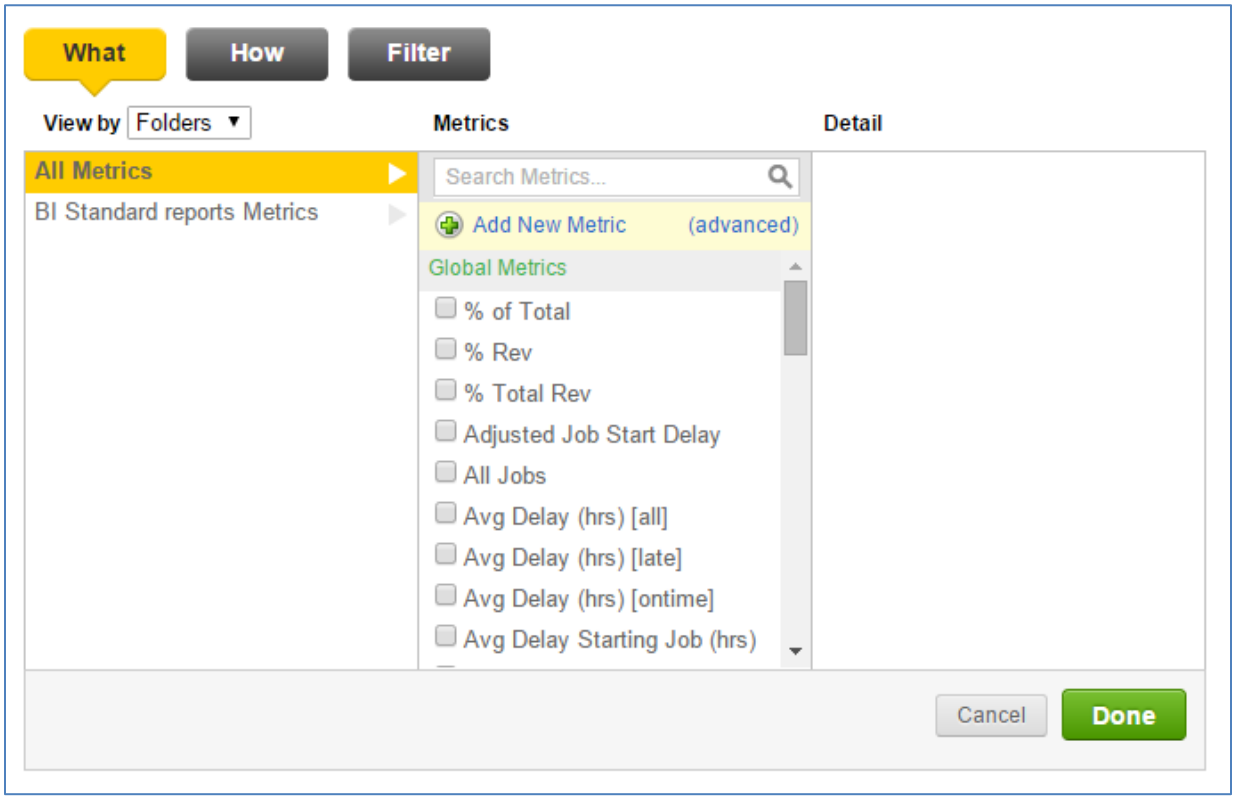
A metric is a simple or complex calculation applied to one or more numeric facts in your data. For example, metrics can be as simple as the sum of all revenue for the month of May. Or, you can build sophisticated metrics, which combine multiple metrics and integrated data filters, to create unprecedented reporting insights in your reports.

When metrics are created as part of creating a report, you can define the metric to be local to the report or be available globally.

- Local metrics can only be used in the report in which they were created.
- Global metrics are available to all FieldAware Editors and include the BI Dashboard Reporting metrics.

Select a Metric

Report Editor page → **What** button → **What** pane

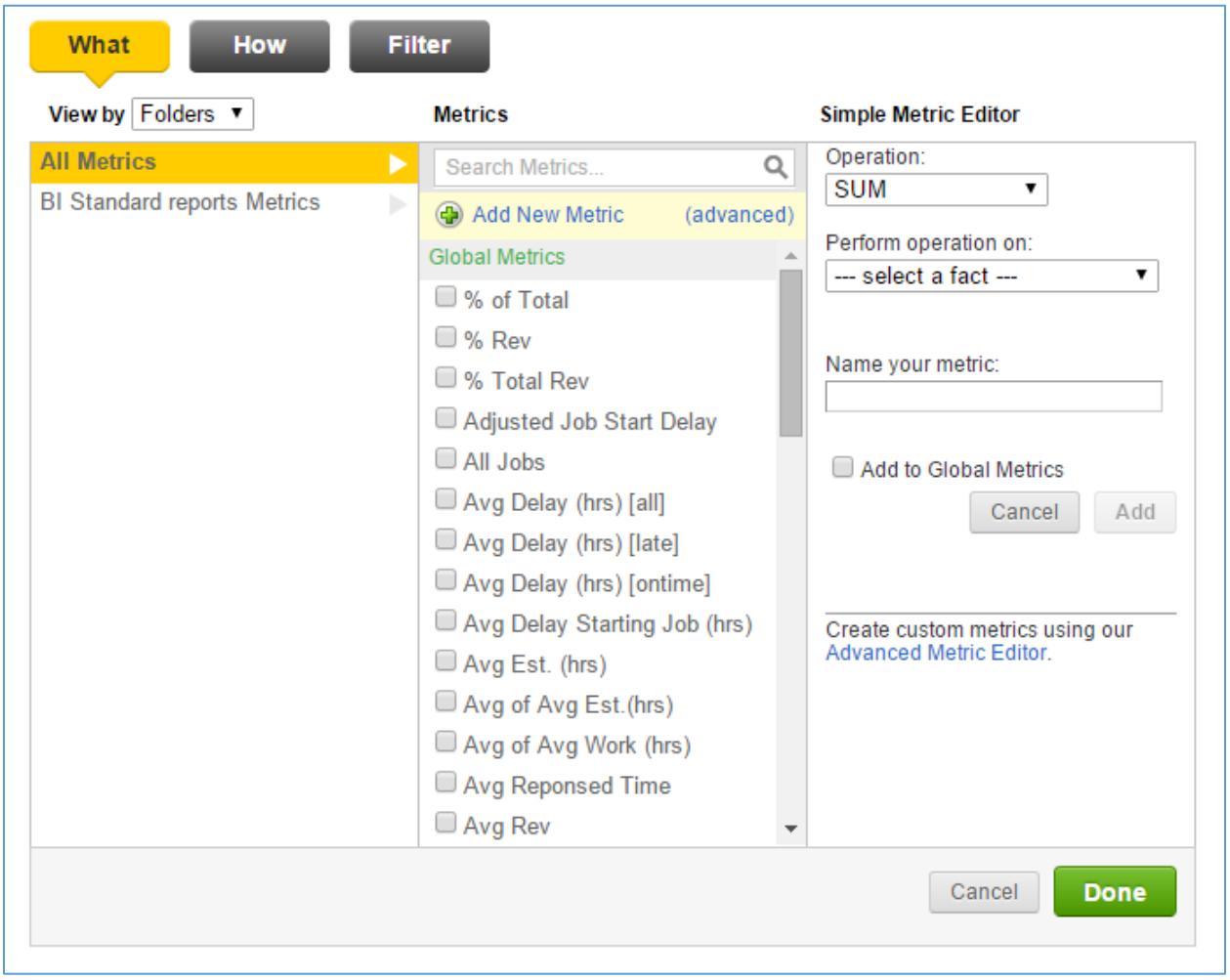


When you are creating or editing a report in the Report Editor, you can use the metrics available for use in other reports. To include a metric in your report, click the **What** button. Select a metric tag or folder. The available metrics are displayed in the Metrics column. Select one or more metrics to include, and click **Done**.

Create a Metric in the Simple Metric Editor

Through the **What** button, you can also create metrics through a simple wizard. Click the **Add New Metric** link. The Simple Metric Editor will display to enable you to quickly create a metric that performs a single operation on a single fact among the available datasets. Most metrics are summaries of individual business records, or fact values. Such metrics are formed by aggregating facts with functions like SUM, MAX, MIN, and AVG. Attributes can also be aggregated into metrics by taking a COUNT of the number of values.

Report Editor page → **What** button → **Add New Metric** → **Simple Metric Editor**



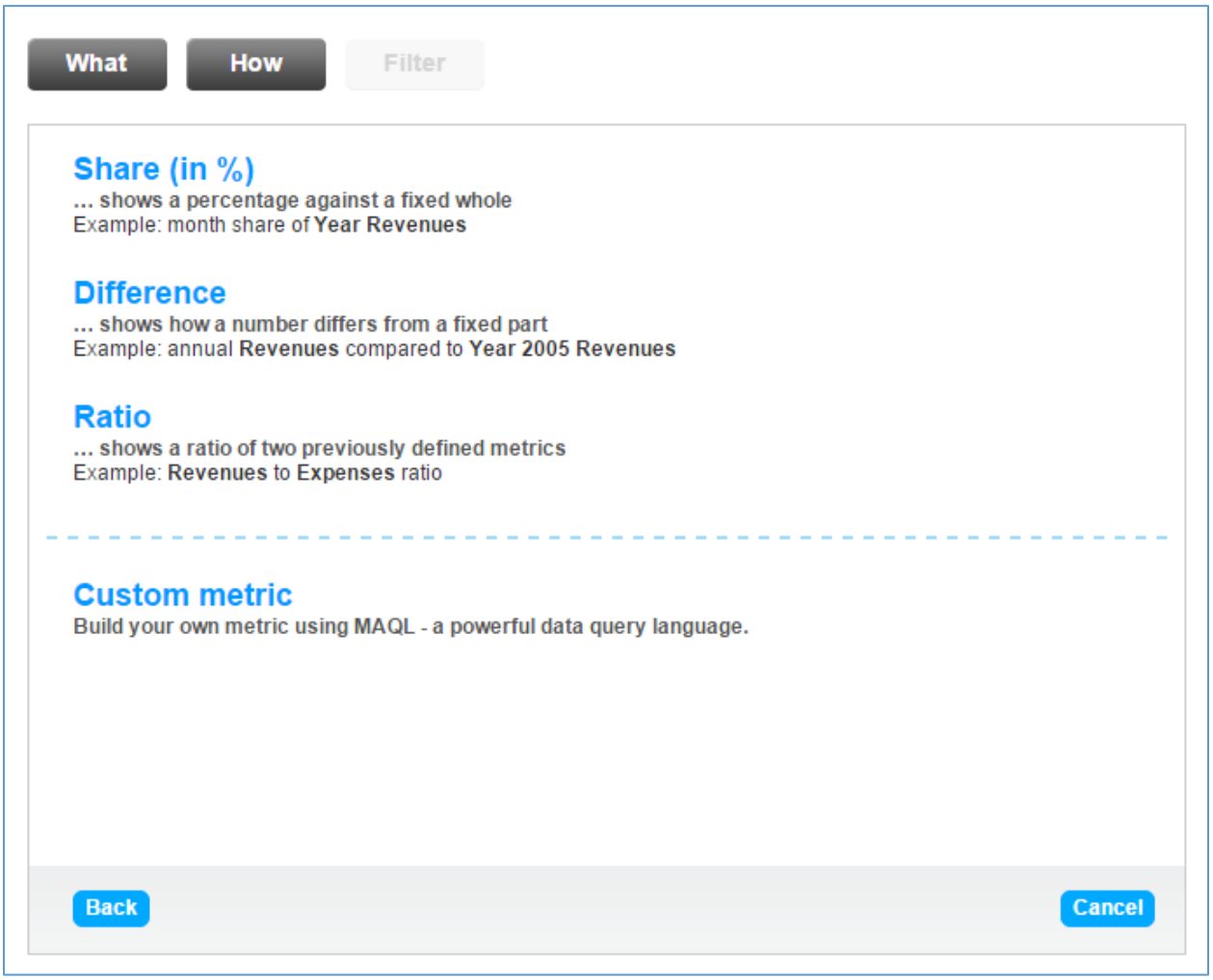
Field	Description
Operation	Metrics are “aggregations” because to summarize those records we apply aggregation operations like SUM, MIN, MAX, and AVG. These return the sum total, minimum value, maximum value, and average value of a fact’s values. A fifth aggregation operation, COUNT, returns the number of unique values belonging to some attribute
SUM	Returns the sum of all fact values
MAX	Returns the largest fact value.
MIN	Returns the smallest fact value.
AVG	Returns the average fact value.
COUNT	Returns the number of unique values belonging to some attribute.
Perform operation on:	Name of the fact or attribute
Name your metric:	Create a label or identifying name for the metric
Add to Global Metrics	Checkbox to include this metric in the global metrics available to all FieldAware Editors

Access the Advanced Metric Editor

Metrics can be much more sophisticated than a single operation applied to a single fact. You can create metrics that combine multiple metrics, perform a variety of arithmetic operations or logical comparisons, or utilize functions that are not available in the Simple Metric Editor.

Through the Advanced Metric Editor, you can create metrics using one of the available wizards or by creating your own metrics in MAQL (Multi-dimension Analytical Query Language). Using MAQL, you can build flexible and powerful metrics to deliver unprecedented insights to your users. MAQL includes a variety of functions, which can be applied to the facts and attributes of your reports.

Report Editor page → **What** button → **Add New Metric** → **Advanced Metric Editor**

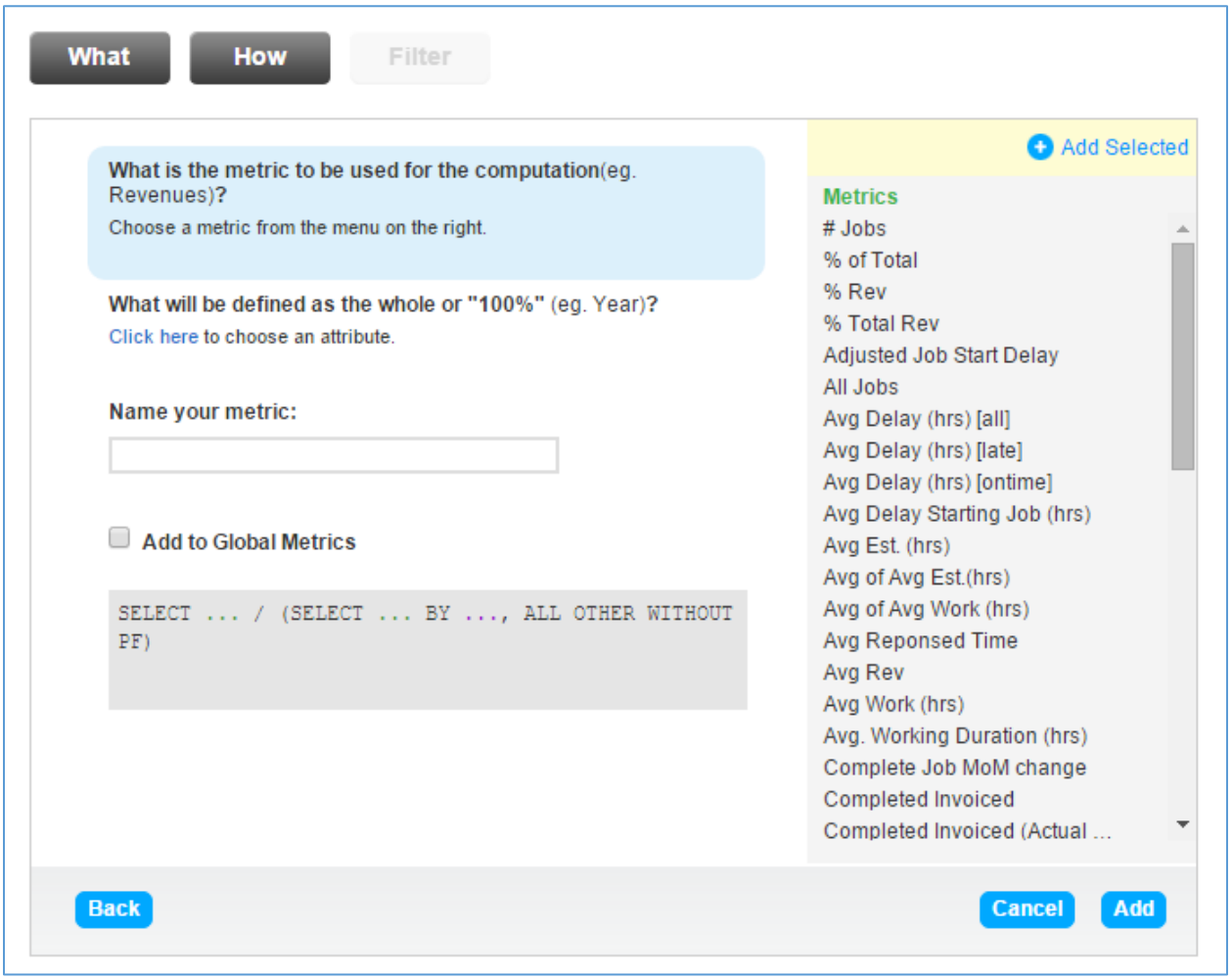


When the Metric Editor is first opened, you are presented with four options. The first three are metric wizards that provide a structure for creating specific types of advanced metrics with MAQL. Clicking **Custom metric** allows you complete flexibility to define metrics from scratch using MAQL.

In addition, you can access the **Advanced Metric Editor** from the **Manage** page by selecting **Metrics** and then **Create Metric**. Metrics created through the Manage page are global by default and can be used by other FieldAware Editors.

Create a Share Percentage Metric

Advanced Metric Editor → Share (in %) Wizard



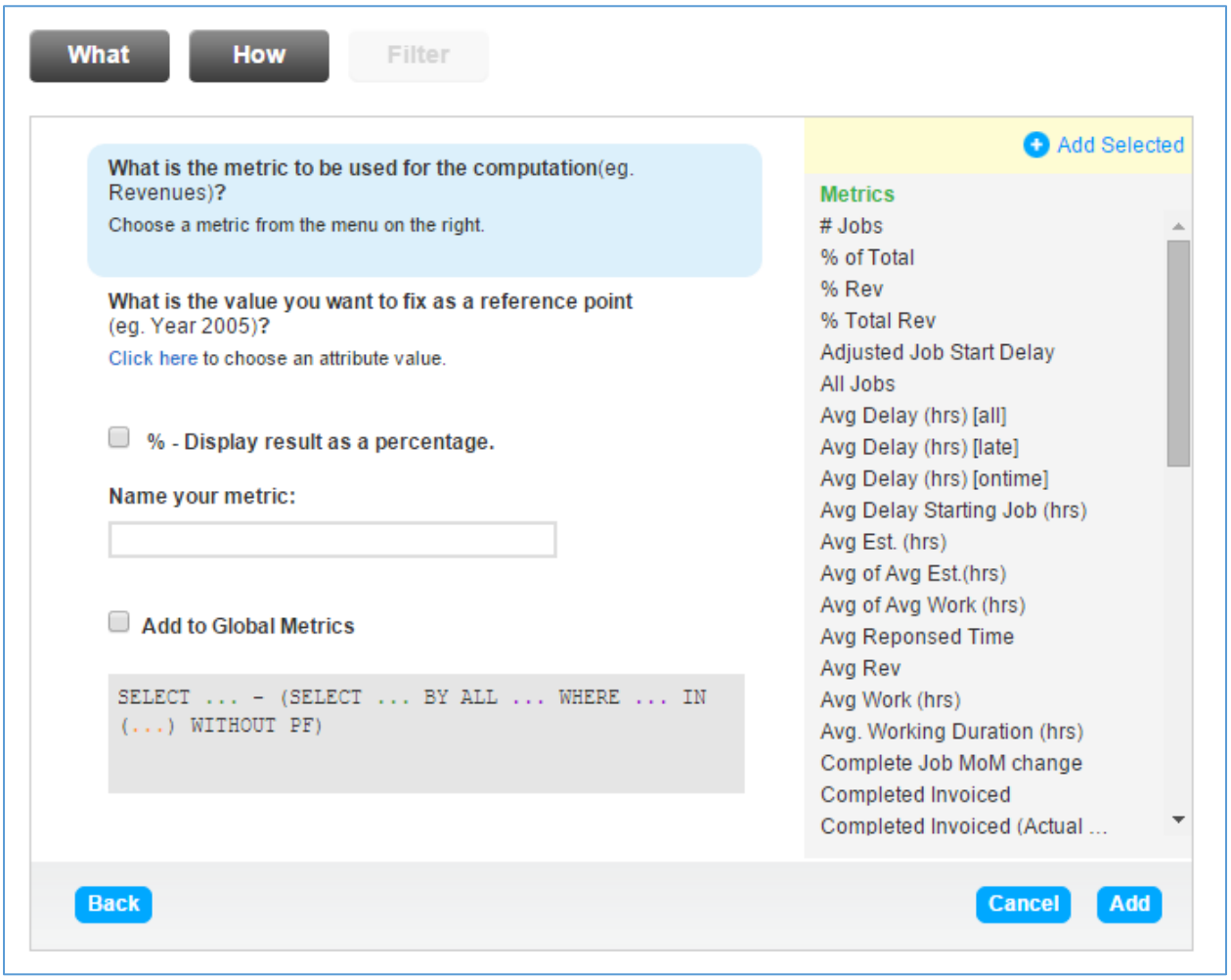
Use the Percentage Share Metric Wizard to create a new metric that uses a percentage to show the relationship of some part to a whole.

A share metric requires the following definitions:

- A metric (such as Sales) to serve as the "part"
- An attribute (such as Year) by which to aggregate this metric, to serve as the "whole"

Create a Difference Metric

Advanced Metric Editor → Difference Wizard



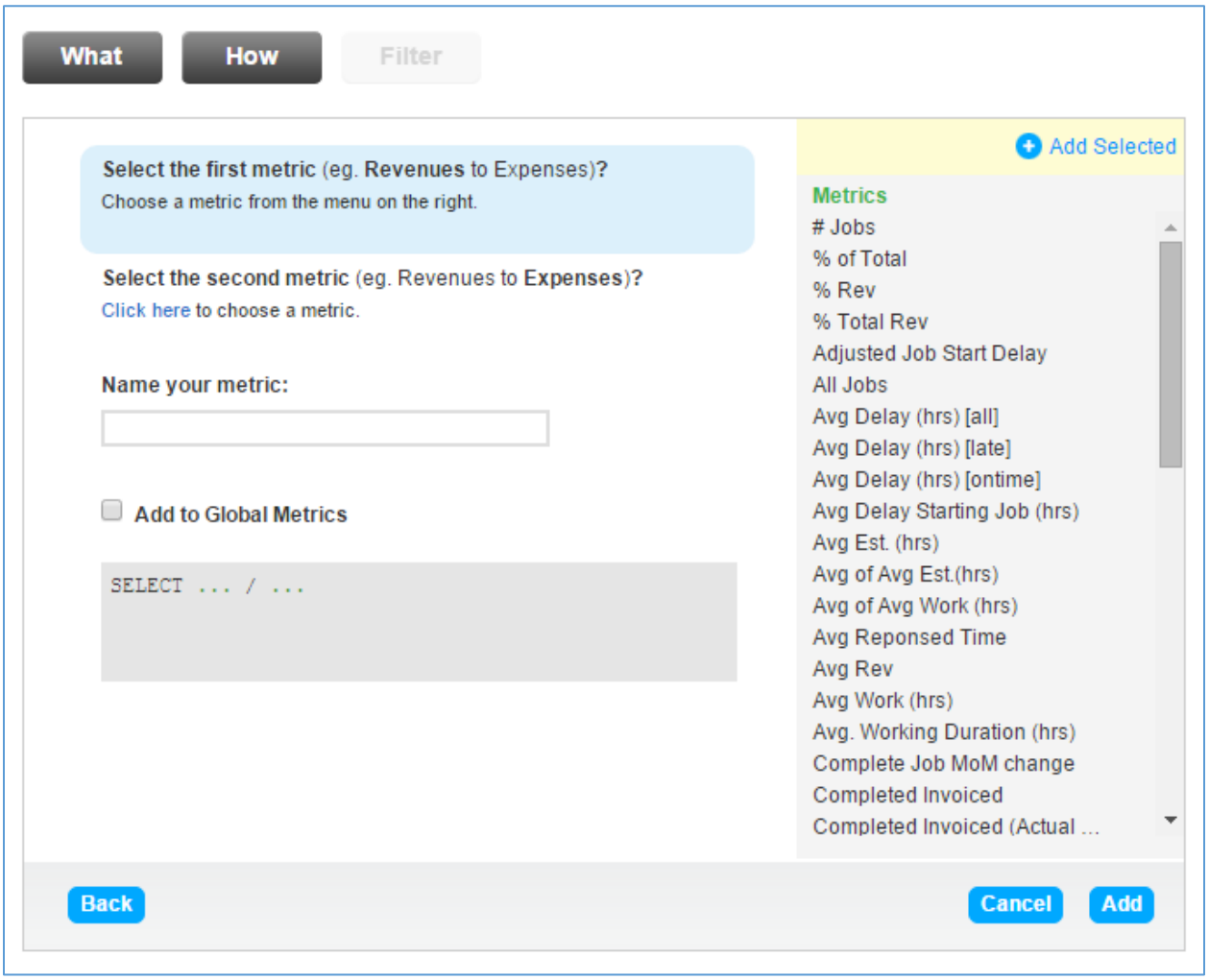
Use the Difference Metric Wizard to create a new metric that calculates the difference between an aggregated value and a fixed number. For example, you can create a metric that calculates the difference between year-to-date sales and a metric that represents the sales quota.

A difference metric requires the following definitions:

- A metric (such as Year-to-date Sales) to serve as the value that will be subtracted
- An attribute by which this metric will be aggregated (such as 2005) to serve as the fixed value that the above metric will be subtracted from

Create a Ratio Metric

Advanced Metric Editor → Ratio Wizard



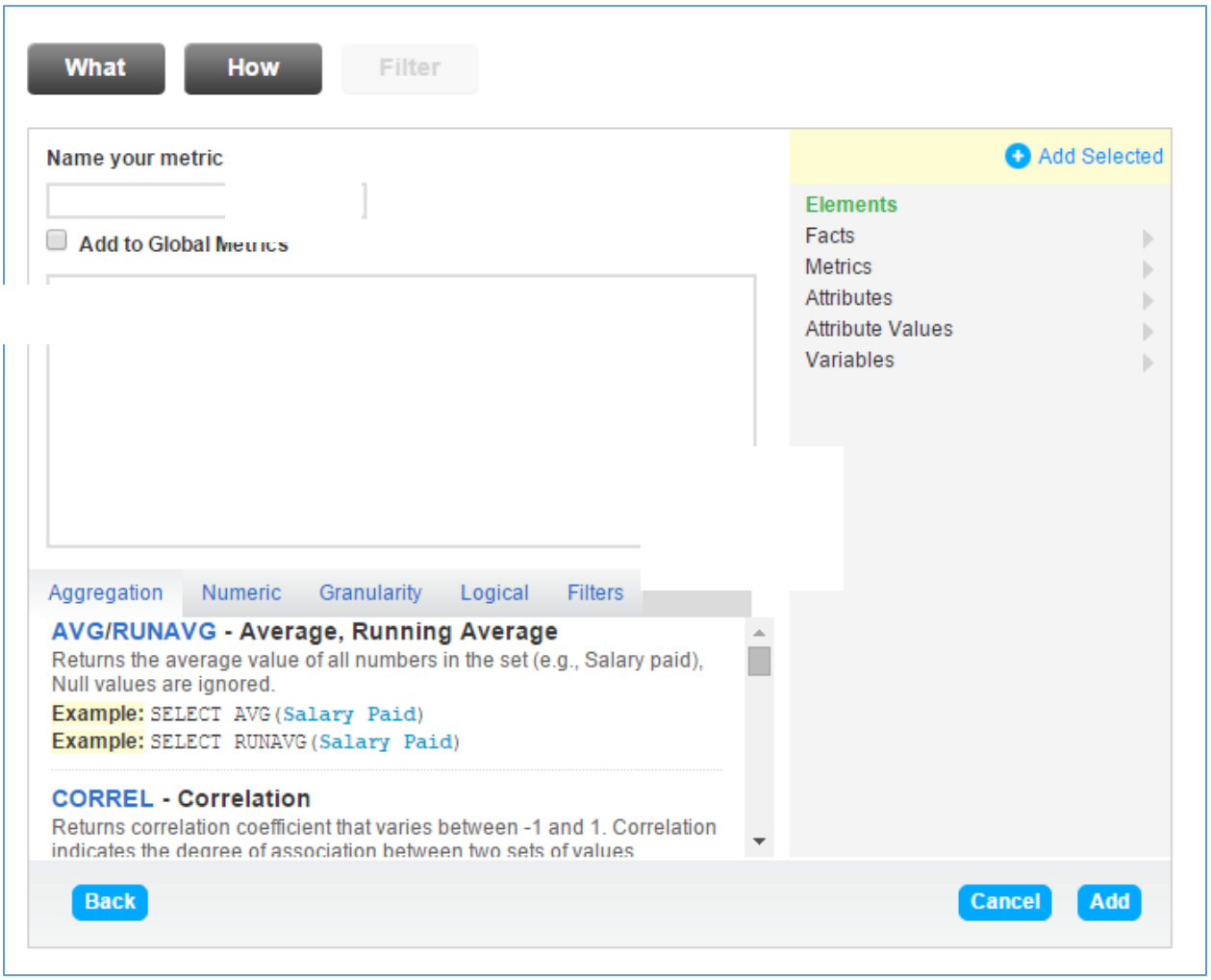
You can relate two metrics to one another with the Ratio Metric Wizard. For example, you can create a metric that calculates a Revenue to Expense ratio.

A ratio metric requires the following definitions:

- The first metric to be used in your ratio (for example, Revenues)
- The second metric to be used in your ratio (for example, Expenses)

Create a Custom Metric

Advanced Metric Editor → Custom Metric



Field	Description
Metric Name	Enter in a name for the metric
Element Sidebar	Lists available elements to include in your MAQL Definition. Elements in your MAQL Definitions are automatically color-coded so that you can keep track of which elements you've included in your metric and ensure your MAQL syntax is valid. Double click code snippets to insert them into the MAQL Definition. **Note:** While MAQL syntax can typically be typed directly into the MAQL Definition field, elements must be added using the sidebar.
MAQL Definition	Displays the MAQL Definition. All metric definitions begin with a SELECT keyword.
MAQL Syntax Reference	Select the appropriate tab to view a list of functions. Double click code snippets to insert syntax into the MAQL Definition.
Aggregation	Aggregation functions are simple mathematical functions that can be performed on your facts to create metrics. An example metric returns the total amount won in sales by summing individual sales figures, which are stored as a fact.

Numeric	Numeric functions perform mathematical operations on facts or metrics. These functions can be simple arithmetic operators or more sophisticated calculations.
Granularity Keywords	These keywords can be used to set the aggregation level for the MAQL expression.
Logical Expressions	Logical operators can be used to combine filters and apply conditions to your MAQL expressions.
Filters	Filters narrow the set of data from which a metric is computed by targeting attribute values you wish to include or exclude from consideration.

Write a Custom Metric with MAQL

Advanced Metric Editor → Custom Metric → MAQL Definition field

All MAQL Definitions begin with the SELECT command. The following syntax summary table will review how to write the basic functions in MAQL. Quick links to the syntax is also included in the Online MAQL Syntax Reference.

Field	Description
Aggregation Commands	
SUM | MAX | MIN | AVG	Use this command to make a request to the database and selecting the data you'd like returned. The basic aggregation functions of summary (SUM), maximum (MAX), minimum (MIN), and average (AVG) apply an operation to a fact in parentheses. • SELECT SUM (Payment)
COUNT	Use this command to make a request to the database and select the data to return, but also requires you to specify the data set where the count will take place. This information appears as a second parameter in the parentheses. • SELECT COUNT (Employees, Facts_of_Payroll)
Arithmetic Operations	
Add (+)	Predefined metric can be nested with a second metric and added together. • SELECT Expenses + Salary
Subtract (-)	Predefined metric can be nested with a second metric and subtracted. • SELECT All Jobs – Completed Jobs
Multiply (*)	Predefined metric can be nested with a second metric and multiplied. • SELECT Total Jobs * 0.03
Divide (/)	Predefined metric can be nested with a second metric and divided.

	• SELECT All Jobs / Job Type
Numeric Variables	
Variable	Assign variables to particular users to automatically tailor metric computations to each report viewer. This could be useful for a single report to display total commissions to sales reps with different commission rates. • SELECT Amount _Won * Commission_Rate_Variable
Mathematical Functions	
Absolute Value (ABS)	Extracts the absolute value of a number • SELECT ABS (-5)
Signum (SIGN)	Extracts the sign of a real number • SELECT SIGN (-3)
Square Root (SQRT)	Extracts the square root of a number • SELECT SQRT (33.6)
Running Total Functions	
Running Sums (RUNSUM)	Running totals represent the sum of all prior values and current value of a metric, broken down by time (date) attributes in a report.
Averages (RUNAVG)	Running totals represent the averages of all prior values and current value of a metric, broken down by time (date) attributes in a report.
Minimums (RUNMIN)	Running totals represent the minimums of all prior values and current value of a metric, broken down by time (date) attributes in a report.
Maximums (RUNMAX)	Running totals represent the minimums of all prior values and current value of a metric, broken down by time (date) attributes in a report.
Filtering	
WHERE Statements	Use the WHERE keyword to filter a metric based on certain conditions. Conditions can contain relational operators (**= <> < <= > >=**) or relational keywords (**IN**, **NOT IN**, **BETWEEN**, **NOT BETWEEN**, as well as time keywords such as **THIS**, **NEXT**, and **PREVIOUS**. Multiple conditions can be combined with logical expressions like **NOT**, **OR**, and **AND**. • SELECT ***Revenues*** WHERE ***Year*** **=** 2006 • SELECT ***Payment*** WHERE ***Date*** = **THIS – 1** • SELECT ***Revenues*** WHERE ***Year***=2006 **AND** ***Month***=5 You can also use filtered variables to define dynamic filter conditions that change depending on the user at hand. • SELECT ***#Leads*** WHERE ***Var_Industry***
Conditional Statements	
IF THEN ELSE	IF THEN ELSE conditional statements allow you to return one of two possible values or perform one of two possible computations, depending on whether some condition is met. • SELECT IF SUM(***Amount***) >= AVG(***Amount***) THEN 10 ELSE 0 END
CASE	Use CASE statements for complex conditionals that contain three or more conditions. • SELECT CASE WHEN SUM(***Amount***) > SUM(***Lost***) AND SUM(***Amount***) – SUM(***Lost***) > 100000 THEN 2, WHEN

	SUM(***Amount***) > SUM(***Lost***) AND SUM(***Amount***) – SUM(***Lost***) < 100000 THEN 1 ELSE 0 END
IFNULL	Use IFNULL to define a replacement value (second parameter) to be inserted in place of any null value returned by some metric expression (first parameter). • SELECT IFNULL(SUM(***Amount***), 0)
Time Transformations	
Time Over Time Comparisons	Time transformations alter the time period to which a metric value relates, which is useful for time over time comparisons (e.g., month over month, or quarter over quarter). A second parameter indicates the number of time periods from the present that the transformation should span. • SELECT ***Payment*** FOR Next(***Quarter***,3)
Ranking Functions	
RANK	Use the RANK function to sequentially rank all of a report's values or to rank within report subgroups, specified by the WITHIN keyword. Ranking can be carried out in ascending or descending order. • SELECT RANK(***Amount***) ASC WITHIN(***Year(Closed)***)
TOP(n) and BOTTOM(n)	TOP(n) and BOTTOM(n) are ranking filters that rank the top or bottom n or n% of values and then exclude all other report values from being displayed. • SELECT ***Amount*** WHERE TOP(5%) OF (***Amount***)
Overriding Report Attributes and Filters - MAQL keywords can customize how report-level attribute and filter configurations affect metrics.	
BY	The BY keyword sets a minimum level of granularity ("aggregation floor") by which a metric can be broken down—even if a report attribute would otherwise serve to break down the metric further. • SELECT ***Payment*** BY ***Year***
BY ALL, BY ALLIN ALL OTHER DIMENSIONS, BY ALLIN ALL OTHER DIMENSIONS EXCEPT FOR	Several additional formulations include the clauses BY ALL, BY ALL IN ALL OTHER DIMENSIONS, and BY ALL IN ALL OTHER DIMENSIONS EXCEPT FOR. • SELECT ***Payment***_BY ALL IN ALL OTHER DIMENSIONS EXCEPT FOR ***Date***
WITHOUT PARENT FILTER (WITHOUT PF)	To override all report level filters defined in the active report, use the WITHOUT PARENT FILTER clause (which can also be written "WITHOUT PF"). • SELECT ***Payment*** WITHOUT PARENT FILTER

Tips and Techniques

Consider the following tips and techniques as you define and create metrics.

- For most users, the Simple Metric Editor is the preferred method for creating metrics.
- After a metric has been added to the global metrics, you can only edit it through the **Manage** page.
- The Metric Editor supports copy and paste, which you can use to manipulate syntax within a metric's MAQL text field, or to duplicate or move syntax from one metric to another.
- While MAQL syntax can typically be typed directly into the MAQL Definition field, elements must be added using the Elements Sidebar.
- When you are creating a new metric, you should first create it as a local metric. You can test it extensively without impacting other FieldAware Editors. When you are satisfied, you can promote it to a global metric through the Metrics page or the Report Editor.
- After you promote a metric to be a global metric, you cannot move it back to being a local metric. Instead create a copy of the local metric and work with the duplicate.

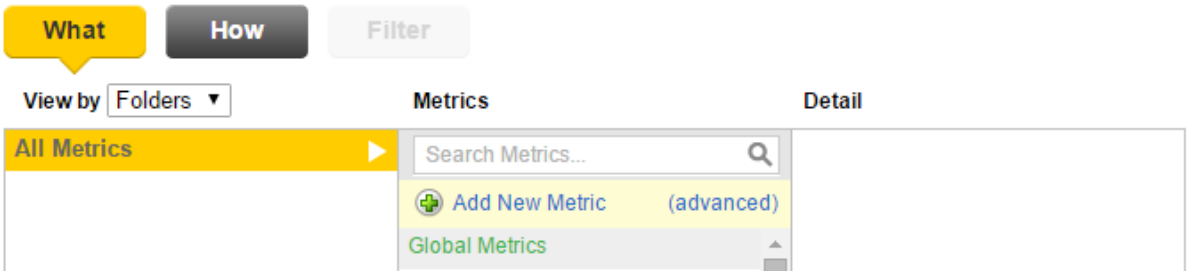
Summary Activity

Metrics are the numerical values displayed in reports. They represent aggregations of facts and attributes. Most metrics are summaries of individual business records, or *fact values*. Such metrics are formed by aggregating facts with functions like SUM, MAX, MIN, and AVG. Attributes can also be aggregated into metrics by taking a COUNT of the number of values they have.

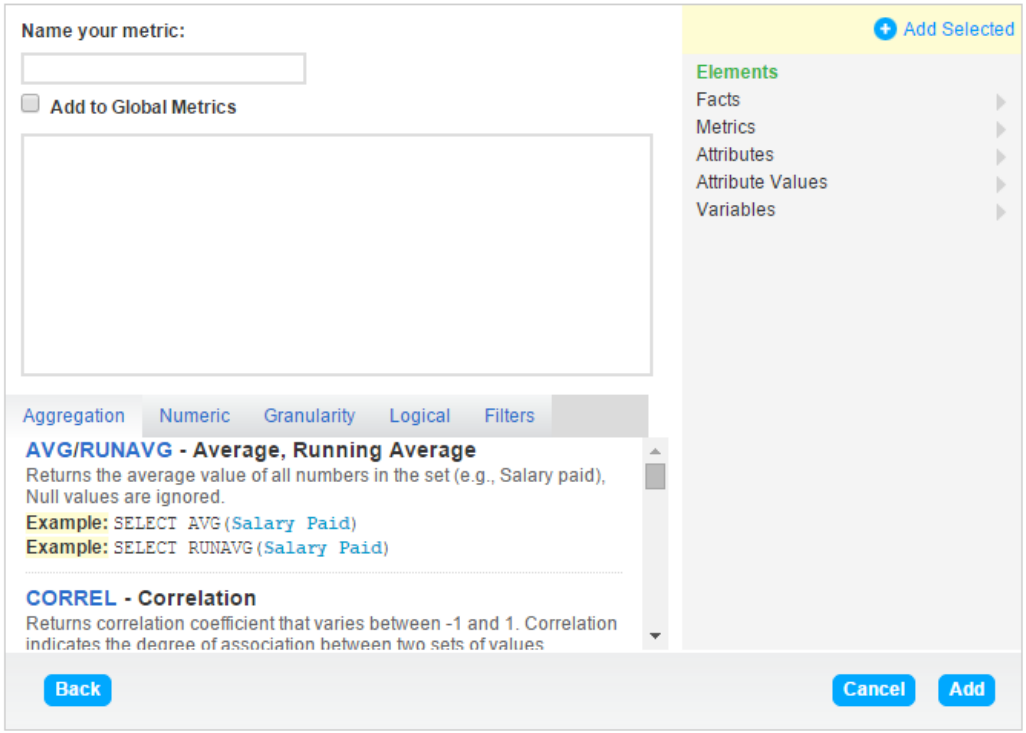
Activity 1: Create Metric using the SUM aggregation method by creating a metric to get the total number job that are in the completed state.

Step 1: Select **Reports** tab → **Report Directory** page → **Create Report** button → **Report Editor.**

Step 2: Select The **What** button → Click on **Advanced.**



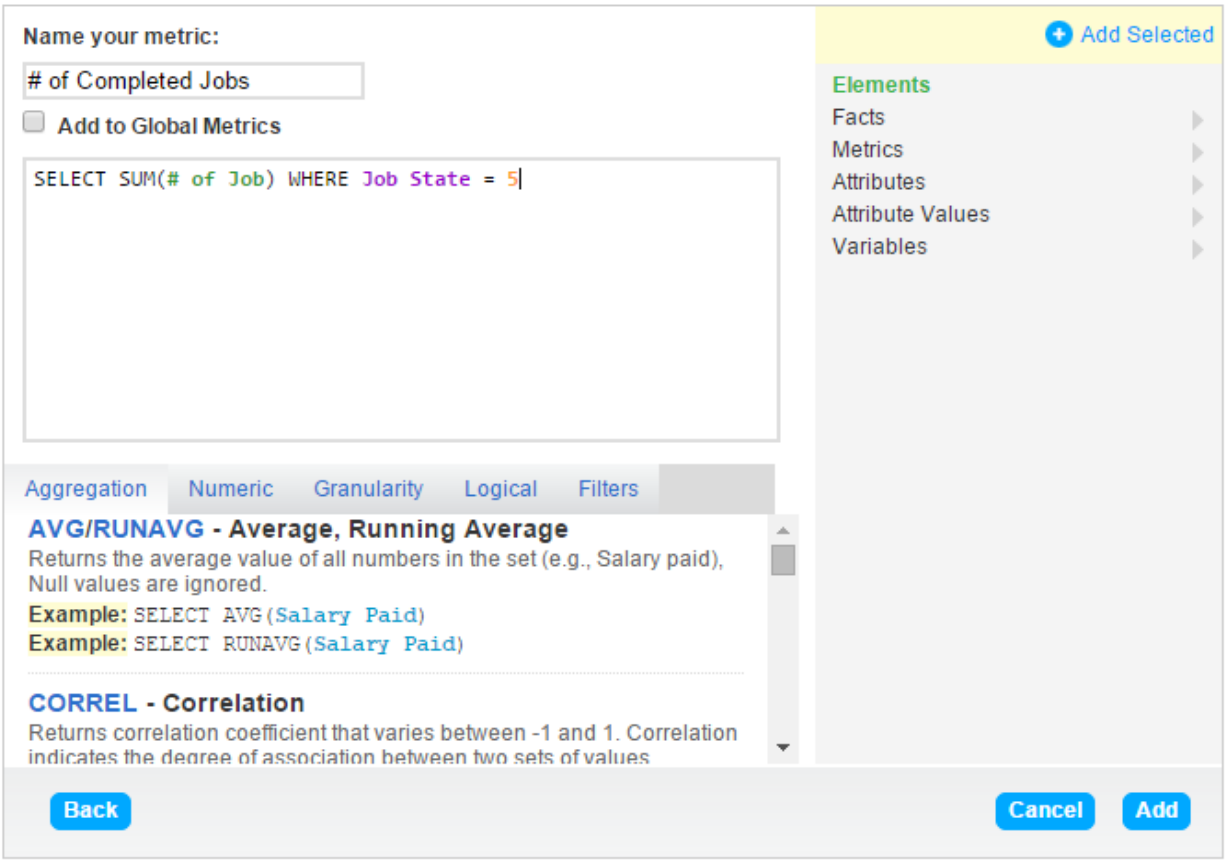
Step 3: Select the last option called **Custom metric.**



Step 4: You can name your metric to any or like this example "# of Completed Jobs" → Inside the text editor write you metric statement as follows "SELECT SUM()"→ Place the cursor inside the sum() parentheses then on the right side select the **Metrics** → Double click on the Metrics **# ofJob.**

Step 5: Add the Where statement after the sum () and select the Attributes on the right hand option →**Job** → Double click **Job State.**

Step 6: Add the equal sign follows the Attributes Value → **Job** → **Job State** → **5 (5 is the completed state)** → **Add** button at the bottom right → **Done.**



Step 7: Now that the metric completed you can see the sum of completed job. At this point the best way to demonstrate this metric would be in a headline report. So select the **Headline** icon and save your report.

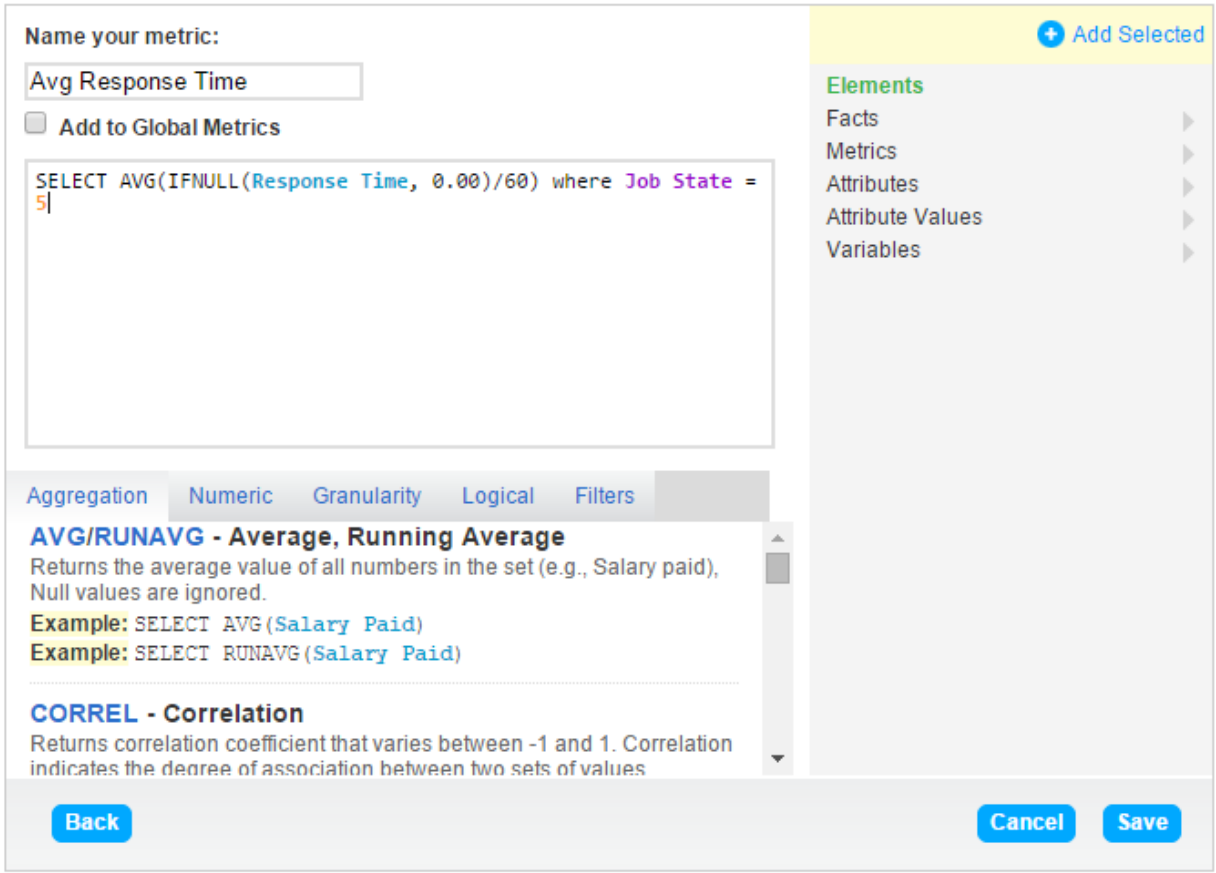
Activity 2: Create Metric using the AVG aggregation method by creating a metric to get the Avg Responsed Time by all job that are in the completed state.

Step 1: Select **Reports** tab → **Report Directory** page → **Create Report** button → **Report Editor.**

Step 2: Select The **What** button → Click on **Advanced.**

Step 3: Select the last option called **Custom metric.**

Step 4: You can name your Metric based your references, but it would be advised to have what aggregation you are using for later use → Write the aggregation statement as follow SELECT AVG(IFNULL(Response Time, 0.00)/60) where Job State = 5. In your statement you are saying to select the average response time of all job and if a single job is null replace value with 0.00. Then divide that average value by 60 to convert value from minute to hour. Last where all job is in the completed state. After that click **Save** and then **Done.**



Step 5: Select the Headline to highlight this value better → Click the **Create** to complete this report.

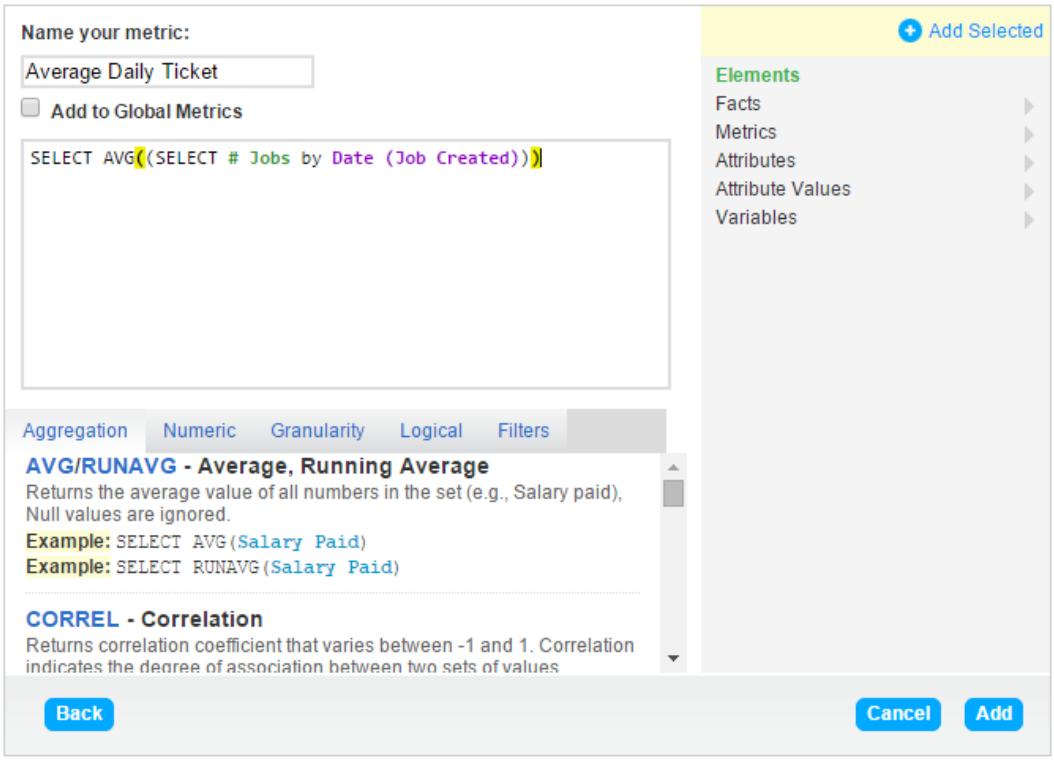
Activity 3: Create Metric using the AVG aggregation method by creating a metric to get the Avg Monthly Job per day.

Step 1: Select **Reports** tab → **Report Directory** page → **Create Report** button → **Report Editor.**

Step 2 : Select The **What** button → Click on **Advanced.**

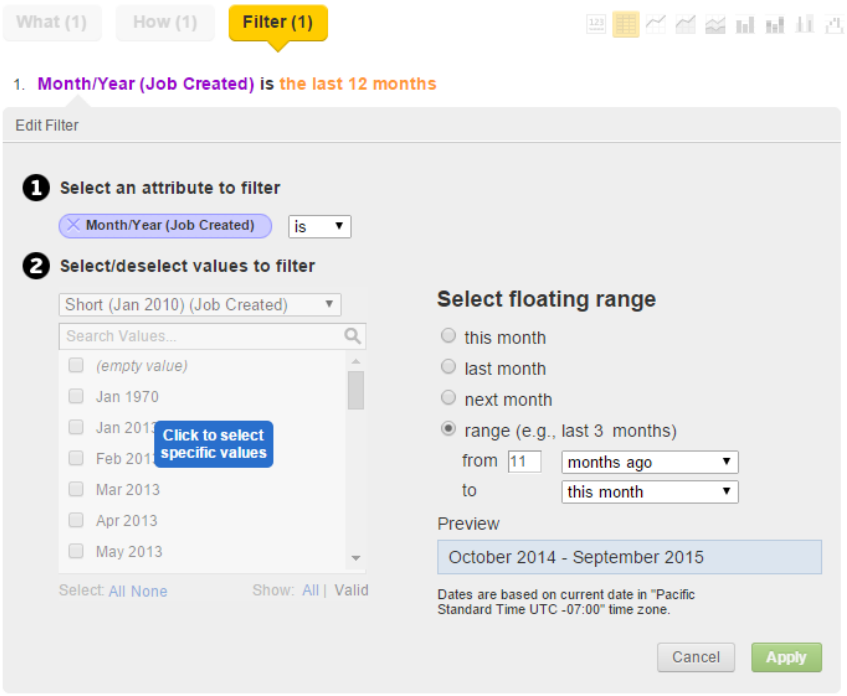
Step 3: Select the last option called **Custom metric.**

Step 4: In the Metric Editor please enter the following statement → SELECT AVG((SELECT COUNT(**Ticket Id**, **Ticket Id**) BY **Date (Created)**)) this statement is saying get the average of all job by the creation date → **Add** → **Done.**



Step 5: Select **How** → **Job** folder → **Date dimension (Job Created)** → **Month/Year** → **Done.**

Step 6: Then We are going to Filter this to just this year by → Select **Filter** → **Select from list value** → **Month/Year** option → Click on the right side and select the range option → Enter 11 month from this month → **Apply** → **Done.**



Step 7: Now we want to display this in a bar chart by selecting the Bar chart icon at the top. Then Click Create to complete this report.


Monthly Job Avg per day
Click here to add description
Saved
Options
What (1)
How (1)
Filter (1)
« Show Configuration
1. Month/Year (Job Created) is the last 12 months
Add Filter
Hide Filters
120.00
90.00
60.00
30.00
0.00
Average Daily Ticket
117.35
20.79
97.48
60.05
26.65
71.77
28.62
23.71
61.25
7.59
Oct 2014
Nov 2014
Dec 2014
Jan 2015
Feb 2015
Mar 2015
Apr 2015
May 2015
Jun 2015
Jul 2015
Month/Year (Job Created)