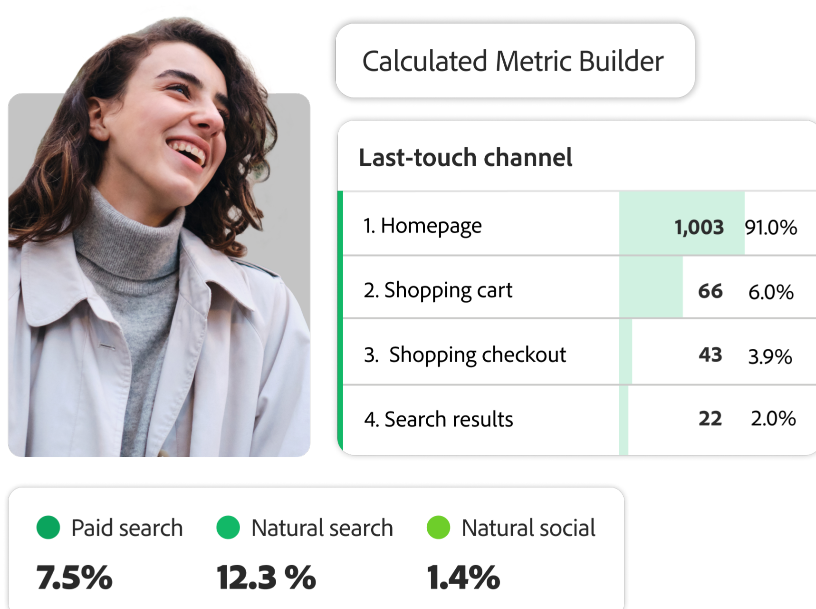
Advanced functions playbook.

How to use calculated metrics in Adobe Analytics Analysis Workspace.

How to use this playbook.

Calculated metrics in Adobe Analytics Analysis Workspace allow you to create powerful data combinations using a range of functions. Mastering these functions will enable you to answer complex questions and showcase your skills. This playbook offers essential knowledge to use these functions effectively. **You don't need to read the entire playbook at once—each section is independent, allowing you to reference specific functions as needed.** Bookmark this playbook and reference it anytime you have questions about a metric within Analysis Workspace.

While Adobe labels some functions as 'basic' or 'advanced,' the current interface doesn't clearly distinguish between them. Instead of focusing on these labels, this guide groups metrics based on their use and function. All examples use demo data, not actual customer behaviour.



The importance of rows when using functions.

Many functions depend on individual row data to work correctly. For instance, the mean function calculates the average for the entire dataset, but its value will change based on the dimension used (for example, day versus week). When creating a calculated metric, the function asks for the metric to average, but the granularity is determined by the table's dimension and row data. This is important to consider for most functions.

Day ↑ Page: 1 / 1 Rows: 25 1-15 of 15	Visits (Apr 6 – Apr 20) 253,588 out of 253,588		Mean Visits (Apr 6 – Apr 20) 16,970 out of 16,970	
1. Apr 6, 2024	18,963	7.5%	16,970	100.0%
2. Apr 7, 2024	26,692	10.5%	16,970	100.0%
3. Apr 8, 2024	15,573	6.1%	16,970	100.0%
4. Apr 9, 2024	19,335	7.6%	16,970	100.0%
5. Apr 10, 2024	13,023	5.1%	16,970	100.0%
6. Apr 11, 2024	13,272	5.2%	16,970	100.0%
7. Apr 12, 2024	13,205	5.2%	16,970	100.0%
8. Apr 13, 2024	12,923	5.1%	16,970	100.0%
9. Apr 14, 2024	13,173	5.2%	16,970	100.0%
10. Apr 15, 2024	15,145	6.0%	16,970	100.0%
11. Apr 16, 2024	18,238	7.2%	16,970	100.0%
12. Apr 17, 2024	19,190	7.6%	16,970	100.0%
13. Apr 18, 2024	27,430	10.8%	16,970	100.0%
14. Apr 19, 2024	13,083	5.2%	16,970	100.0%
15. Apr 20, 2024	15,306	6.0%	16,970	100.0%

Week ↑ Page: 1 / 1 Rows: 10 1-3 of 3	Visits (Apr 6 – Apr 20) 253,588 out of 253,588		Mean Visits (Apr 6 – Apr 20) 84,598 out of 84,598	
1. Mar 31, 2024	18,963	7.5%	84,598	100.0%
2. Apr 7, 2024	113,698	44.8%	84,598	100.0%
3. Apr 14, 2024	121,133	47.8%	84,598	100.0%

Note about rows with a value of 0.

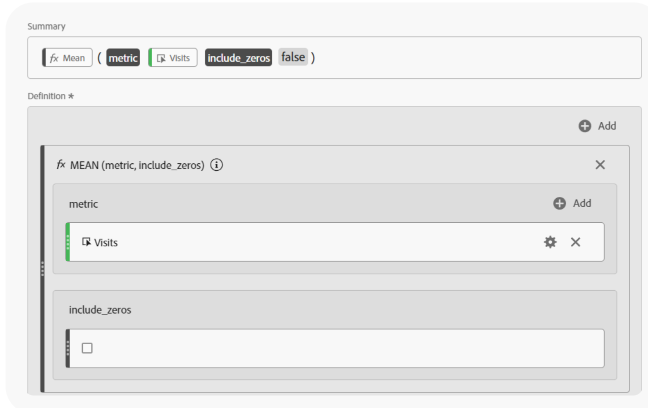
Typically, dimension values appear in your freeform table when they have data, but some rows may show totals of 0. Many functions allow you to 'include zeros' in the calculation. When you create a metric from a selection, this option is selected by default, but when you drag in a function from the menu, it is not. Whether to include rows with 0 depends on your specific use case.

Table of contents

Descriptive Metrics.

▪ I want to describe my data's averages…

Analysis Workspace has two of the three measures of central tendencies available for you – **MEAN** and **MEDIAN**. (It doesn't currently have an option for a mode). These measures of central tendencies are great for providing an overview or a summary number. If you're creating a dashboard and want to include some summary KPIs at the top using the summary number (or summary change) visualisations with **MEAN** or **MEDIAN** can be a great option.



Both functions can give you an idea of where the middle of your data lies. They can be slightly different though – the **MEAN** sums up all the values and divides by the count, whereas the **MEDIAN** takes the middle value. If your data has a lot of outliers on one side of the data, it's more likely to skew the **MEAN** than the **MEDIAN**, so the range of your data will determine what is better to use. But both of these are built out the same way. The function only requires one argument – the metric that you want averaged. It does give you the option to include zeros, meaning that if you have any rows with zero for that metric, it'll still include them in the calculation, although the default is to exclude them.

Once you have the metric in your table, select the dimension you want to use with the calculation. The metric will return an average for all of the rows in your table.

Mean + Median

Day ↑	Visits		Mean Visits		Median Visits	
Page: 1 / 1 Rows: 25 1-15 of 15	253,588	out of 253,588	16,970	out of 16,970	15,306	out of 15,306
1. Apr 6, 2024	18,963	7.5%	16,970	100.0%	15,306	100.0%
2. Apr 7, 2024	26,692	10.5%	16,970	100.0%	15,306	100.0%
3. Apr 8, 2024	15,573	6.1%	16,970	100.0%	15,306	100.0%
4. Apr 9, 2024	19,335	7.6%	16,970	100.0%	15,306	100.0%
5. Apr 10, 2024	13,023	5.1%	16,970	100.0%	15,306	100.0%
6. Apr 11, 2024	13,272	5.2%	16,970	100.0%	15,306	100.0%
7. Apr 12, 2024	13,205	5.2%	16,970	100.0%	15,306	100.0%
8. Apr 13, 2024	12,923	5.1%	16,970	100.0%	15,306	100.0%
9. Apr 14, 2024	13,173	5.2%	16,970	100.0%	15,306	100.0%
10. Apr 15, 2024	15,145	6.0%	16,970	100.0%	15,306	100.0%
11. Apr 16, 2024	18,238	7.2%	16,970	100.0%	15,306	100.0%
12. Apr 17, 2024	19,190	7.6%	16,970	100.0%	15,306	100.0%
13. Apr 18, 2024	27,430	10.8%	16,970	100.0%	15,306	100.0%
14. Apr 19, 2024	13,083	5.2%	16,970	100.0%	15,306	100.0%
15. Apr 20, 2024	15,306	6.0%	16,970	100.0%	15,306	100.0%

- **I want to keep a running total…**

It typically makes sense to use running totals, with time dimensions, such as days or weeks. There are two options here: a **CUMULATIVE** total summing all the rows or a **CUMULATIVE AVERAGE** creating an average as it moves down the rows. Both of these functions rely on the time series information. Regardless of what order the rows in your table are in, it will start its calculation at the earliest time point and move through time. So, the order of the rows in the table is less important than the actual dates or times being used.

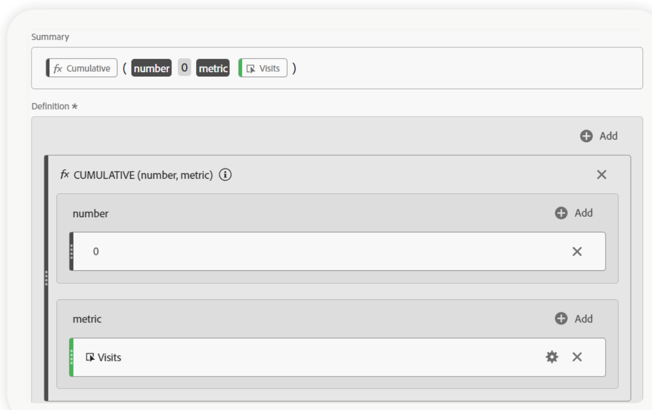
Cumulative - Ordered by Visits

Day Page: 1 / 1 Rows: 25 1-15 of 15	Visits 253,588 out of 253,588		Cumulative n=0 (Visits) 253,588 out of 253,588	
1. Apr 18, 2024	27,430	10.8%	226,162	89.2%
2. Apr 7, 2024	26,692	10.5%	45,655	18.0%
3. Apr 9, 2024	19,335	7.6%	80,563	31.8%
4. Apr 17, 2024	19,190	7.6%	198,732	78.4%
5. Apr 6, 2024	18,963	7.5%	18,963	7.5%
6. Apr 16, 2024	18,238	7.2%	179,542	70.8%
7. Apr 8, 2024	15,573	6.1%	61,228	24.1%
8. Apr 20, 2024	15,306	6.0%	254,551	100.4%
9. Apr 15, 2024	15,145	6.0%	161,304	63.6%
10. Apr 11, 2024	13,272	5.2%	106,858	42.1%
11. Apr 12, 2024	13,205	5.2%	120,063	47.3%
12. Apr 14, 2024	13,173	5.2%	146,159	57.6%
13. Apr 19, 2024	13,083	5.2%	239,245	94.3%
14. Apr 10, 2024	13,023	5.1%	93,586	36.9%
15. Apr 13, 2024	12,923	5.1%	132,986	52.4%

Cumulative - Ordered by Date

Day ↑ Page: 1 / 1 Rows: 25 1-15 of 15	Visits 253,588 out of 253,588		Cumulative n=0 (Visits) 253,588 out of 253,588	
1. Apr 6, 2024	18,963	7.5%	18,963	7.5%
2. Apr 7, 2024	26,692	10.5%	45,655	18.0%
3. Apr 8, 2024	15,573	6.1%	61,228	24.1%
4. Apr 9, 2024	19,335	7.6%	80,563	31.8%
5. Apr 10, 2024	13,023	5.1%	93,586	36.9%
6. Apr 11, 2024	13,272	5.2%	106,858	42.1%
7. Apr 12, 2024	13,205	5.2%	120,063	47.3%
8. Apr 13, 2024	12,923	5.1%	132,986	52.4%
9. Apr 14, 2024	13,173	5.2%	146,159	57.6%
10. Apr 15, 2024	15,145	6.0%	161,304	63.6%
11. Apr 16, 2024	18,238	7.2%	179,542	70.8%
12. Apr 17, 2024	19,190	7.6%	198,732	78.4%
13. Apr 18, 2024	27,430	10.8%	226,162	89.2%
14. Apr 19, 2024	13,083	5.2%	239,245	94.3%
15. Apr 20, 2024	15,306	6.0%	254,551	100.4%

One use for a **CUMULATIVE** metric is tracking performance compared to a specific goal. If you're running a campaign that is expected to generate a certain amount of revenue, you can use the **CUMULATIVE** function to see how the total is growing each day. You can even take this one step further and graph it against a 'static number' (your goal) to watch the progression over time.



To build either of these metrics, you need two conditions. The first is "number" which is a static number that you decide. If you put 0, the **CUMULATIVE** or **CUMULATIVE AVERAGE** function will include the current row and all rows above it. If you put a positive number (greater than 0), it will use that number of rows for the calculation. For example, if you put 5, it will use the current row and the four previous rows, totaling five rows. The second component of this function is the metric you want to use in the calculation.

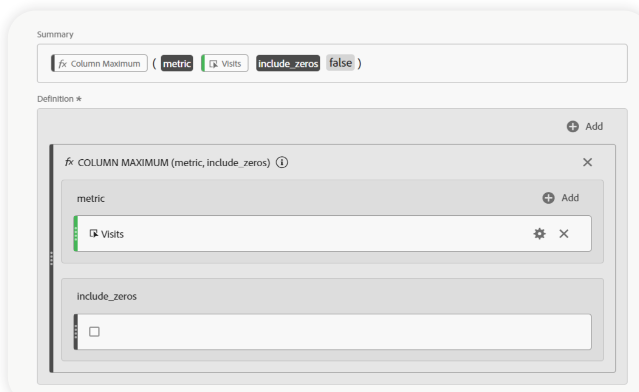
When you put the metric in the table it will start with the earliest time point and calculate the total or average. It will then move to the second time point, regardless of what row it is in, and calculate based on the first and second time points, and so on for each time point (depending on the number you've indicated in building the function).

Cumulative + Cumulative Average

Day (Page: 1 / 1 Rows: 25 1-15 of 15)	Visits	Cumulative n=0 (Visits)	Cumulative n=2 (Visits)	Cumulative Average n=0 (Visits)	Cumulative Average n=2 (Visits)
	253,588 out of 253,588	253,588 out of 253,588	253,588 out of 253,588	253,588 out of 253,588	253,588 out of 253,588
1. Apr 6, 2024	18,963 7.5%	18,963 7.5%	18,963 7.5%	18,963 7.5%	18,963 7.5%
2. Apr 7, 2024	26,692 10.5%	45,655 18.0%	45,655 18.0%	22,828 9.0%	22,828 9.0%
3. Apr 8, 2024	15,573 6.1%	61,228 24.1%	42,265 16.7%	20,409 8.0%	21,133 8.3%
4. Apr 9, 2024	19,335 7.6%	80,563 31.8%	34,908 13.8%	20,141 7.9%	17,454 6.9%
5. Apr 10, 2024	13,023 5.1%	93,586 36.9%	32,358 12.8%	18,717 7.4%	16,179 6.4%
6. Apr 11, 2024	13,272 5.2%	106,858 42.1%	26,295 10.4%	17,810 7.0%	13,148 5.2%
7. Apr 12, 2024	13,205 5.2%	120,063 47.3%	26,477 10.4%	17,152 6.8%	13,239 5.2%
8. Apr 13, 2024	12,923 5.1%	132,986 52.4%	26,128 10.3%	16,623 6.6%	13,064 5.2%
9. Apr 14, 2024	13,173 5.2%	146,159 57.6%	26,096 10.3%	16,240 6.4%	13,048 5.1%
10. Apr 15, 2024	15,145 6.0%	161,304 63.6%	28,318 11.2%	16,130 6.4%	14,159 5.6%
11. Apr 16, 2024	18,238 7.2%	179,542 70.8%	33,383 13.2%	16,322 6.4%	16,692 6.6%
12. Apr 17, 2024	19,190 7.6%	198,732 78.4%	37,428 14.8%	16,561 6.5%	18,714 7.4%
13. Apr 18, 2024	27,430 10.8%	226,162 89.2%	46,620 18.4%	17,397 6.9%	23,310 9.2%
14. Apr 19, 2024	13,083 5.2%	239,245 94.3%	40,513 16.0%	17,089 6.7%	20,257 8.0%
15. Apr 20, 2024	15,306 6.0%	254,551 100.4%	28,389 11.2%	16,970 6.7%	14,195 5.6%

▪ I want to find the biggest or smallest value in a single column…

The **COLUMN MAX** and **COLUMN MIN** metrics seem fairly self-explanatory – they look in the column to find either the largest or smallest value across all the rows. However, the value that it returns is the maximum or minimum for the entire dimension, not just the values that are in the table.



So, whether you have all values of a dimension for that time period, or only a portion of them, it will still consider the entire dimension for your selected time period. The data that this returns is going to be the same number across all of the values in your dimension. While this may look strange in a table, this is a great metric to use for a summary number visualisation or as a benchmark value in a line graph. This can be helpful if you're trying to compare multiple items. If, for example, a customer can pick one of three options on your site, then this can show you which option had the most or fewest interactions. To create either of these, the only item you need in the metric builder is your metric of interest. There is also an option to include rows that contain 0; the default is to exclude them.

Once in the table, it will look at all of the values for the selected dimension and return the specified value in each of the rows. If you are using a dimension, such as page name for example, and you want it to return the **COLUMN MAX** or **COLUMN MIN** for just the values in your table, the limitation needs to be done with a segment. Limiting dimension items with methods such as "display only selected rows" will not limit the functions calculation.

Column Max & Min

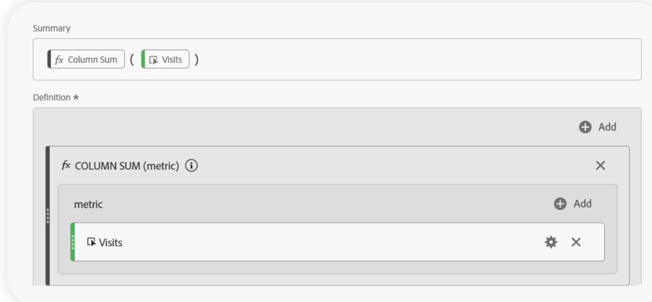
Day ↑ Page: 1 / 1 Rows: 25	Visits		Column Max (Visits)		Column Min (Visits)	
	253,588 out of 253,588		27,430 out of 27,430		12,923 out of 12,923	
1. Apr 6, 2024	18,963	7.5%	27,430	100.0%	12,923	100.0%
2. Apr 7, 2024	26,692	10.5%	27,430	100.0%	12,923	100.0%
3. Apr 8, 2024	15,573	6.1%	27,430	100.0%	12,923	100.0%
4. Apr 9, 2024	19,335	7.6%	27,430	100.0%	12,923	100.0%
5. Apr 10, 2024	13,023	5.1%	27,430	100.0%	12,923	100.0%
6. Apr 11, 2024	13,272	5.2%	27,430	100.0%	12,923	100.0%
7. Apr 12, 2024	13,205	5.2%	27,430	100.0%	12,923	100.0%
8. Apr 13, 2024	12,923	5.1%	27,430	100.0%	12,923	100.0%
9. Apr 14, 2024	13,173	5.2%	27,430	100.0%	12,923	100.0%
10. Apr 15, 2024	15,145	6.0%	27,430	100.0%	12,923	100.0%
11. Apr 16, 2024	18,238	7.2%	27,430	100.0%	12,923	100.0%
12. Apr 17, 2024	19,190	7.6%	27,430	100.0%	12,923	100.0%
13. Apr 18, 2024	27,430	10.8%	27,430	100.0%	12,923	100.0%
14. Apr 19, 2024	13,083	5.2%	27,430	100.0%	12,923	100.0%
15. Apr 20, 2024	15,306	6.0%	27,430	100.0%	12,923	100.0%

- **I want to add up all the values for a dimension…**

Due to the nature of the freeform table, there are instances where the grand totals for metrics are deduplicated. For example, looking at visits on different days, if a visit spans more than one day, it will deduplicate the grand total. When you want to know the grand total of all the values, regardless of duplication, you can use the **COLUMN SUM** function. This function works against the entirety of a dimension, regardless of what is showing in the table. If you have a dimension, such as pages, you can use the **COLUMN SUM** for the pageviews to see the grand total of all your pages together. This makes it very easy to put into a summary visualisation and present it as a KPI.

To build it, you only need to put in your metric of interest. Here there is no option to include or exclude rows with 0, as they won't impact a summation.

When you put the metric in your table, it will return the **COLUMN SUM** for all the values in your dimension, regardless of how many items from the dimension are in the table. Similar to the **MIN** and **MAX** functions, the **SUM** function doesn't change based on the individual row, and will return the same number on every row. If you want to limit the items in the dimension included in the calculation, you need to use a segment, as methods such as "display only selected rows" will not impact the function's calculation.

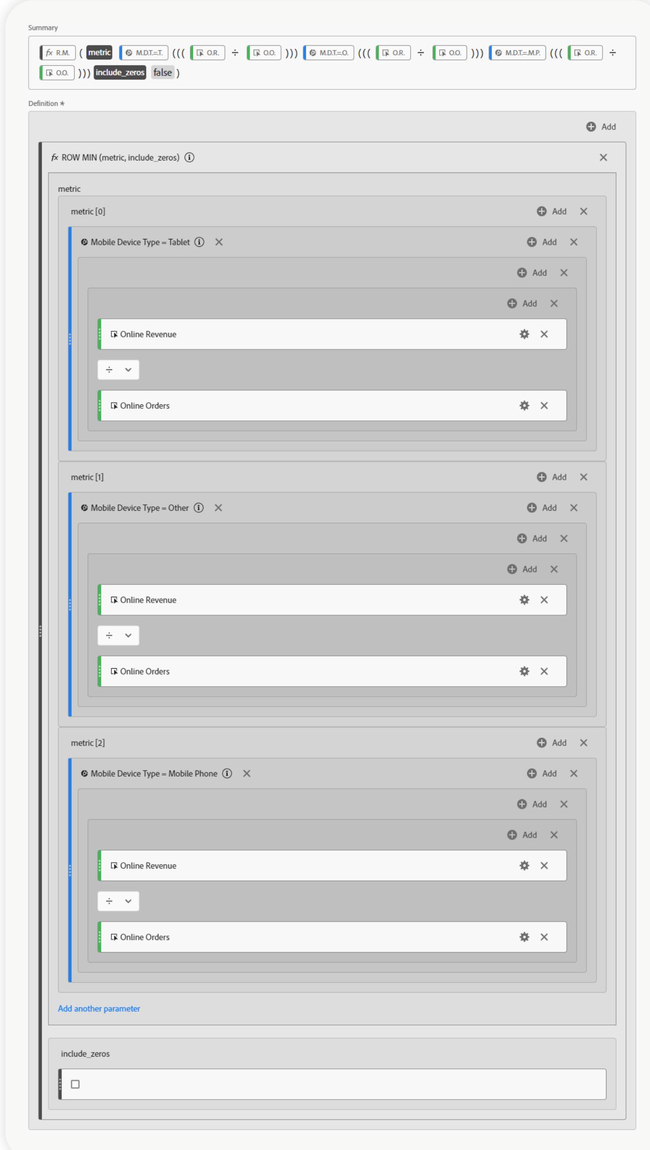


Column Sum

Day	Visits		Column Sum (Visits)	
Page: 1 / 1 Rows: 25 1-15 of 15	253,588 out of 253,588	Apr 6	254,551 out of 254,551	Apr 6
1. Apr 6, 2024	18,963	7.5%	254,551	100.0%
2. Apr 7, 2024	26,692	10.5%	254,551	100.0%
3. Apr 8, 2024	15,573	6.1%	254,551	100.0%
4. Apr 9, 2024	19,335	7.6%	254,551	100.0%
5. Apr 10, 2024	13,023	5.1%	254,551	100.0%
6. Apr 11, 2024	13,272	5.2%	254,551	100.0%
7. Apr 12, 2024	13,205	5.2%	254,551	100.0%
8. Apr 13, 2024	12,923	5.1%	254,551	100.0%
9. Apr 14, 2024	13,173	5.2%	254,551	100.0%
10. Apr 15, 2024	15,145	6.0%	254,551	100.0%
11. Apr 16, 2024	18,238	7.2%	254,551	100.0%
12. Apr 17, 2024	19,190	7.6%	254,551	100.0%
13. Apr 18, 2024	27,430	10.8%	254,551	100.0%
14. Apr 19, 2024	13,083	5.2%	254,551	100.0%
15. Apr 20, 2024	15,306	6.0%	254,551	100.0%

▪ I want to find the biggest or smallest value across multiple columns…

The **ROW MAX** and **ROW MIN** metrics aren't very well described in the current documentation. The purpose of these functions is to find the largest or smallest value, respectively, in each row across several columns. Say for example, you have an average order value (AOV) for three device types producing three columns, and you want to know for each day what is the highest or lowest AOV regardless of device type. You can use these functions to look at more than one column to find the indicated value. Note that you don't have to make the actual columns in your table, because you're going to build the logic for each column in the calculated metric. But adding the columns to your table will help you find which one the **MAX** or **MIN** value is coming from.



Building this is also not very intuitive. When you first bring the function into the metric builder, it asks for a single metric (metric [0]). However, this only represents one 'column.' If you select "add another parameter", it adds "metric[1]." Although the default is just a single metric, in order for this function to work properly, you need at least two metrics, as each metric represents a column. If you only have one column, then the **ROW MAX** or **ROW MIN** returned will be the same as the single column. So, for building this, you can bring in as many columns (that is, metrics) as needed (I've done up to 25 before), using either different metrics or the same metric but with different segments (such as device type).

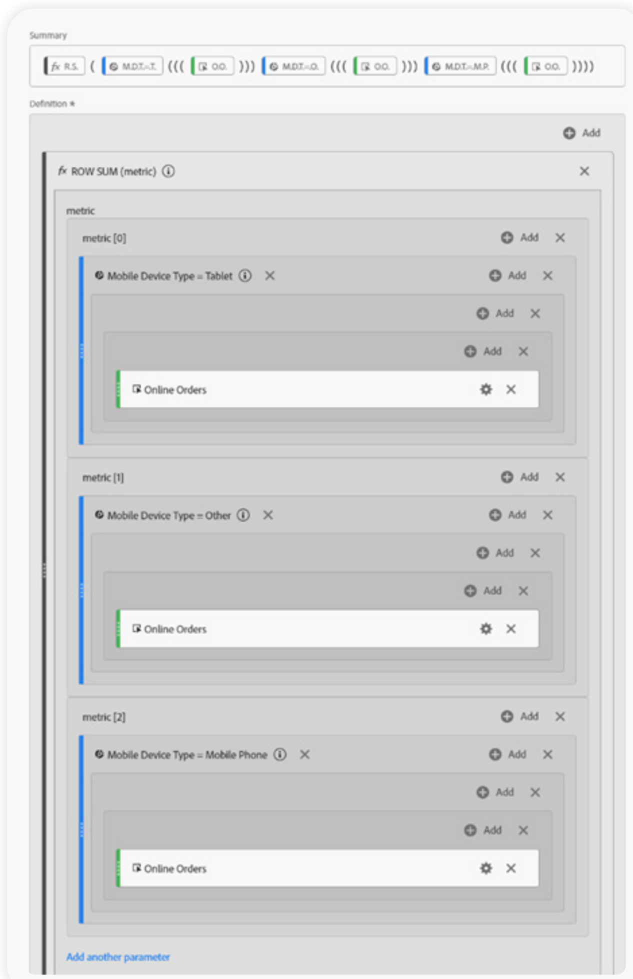
When putting this metric into your table, it will look at each row individually and consider the values for that dimension against each of the rows that you've included in the metric function. It will then return the appropriate result for each row.

Row Max & Min

	Avg. Order Value (AOV)			Row Min (AOV by Device Type)	Row Max (AOV by Device Type)
	Other	Mobile Phone	Tablet		
Day ↑ Page: 1 / 1 Rows: 25	$288.44 out of $288.44	$277.00 out of $277.00	$280.64 out of $280.64	277.00 out of 277.00	288.44 out of 288.44
1. Apr 6, 2024	$299.80 103.9%	$329.25 118.9%	$341.17 121.6%	299.80 108.2%	341.17 118.3%
2. Apr 7, 2024	$307.99 106.8%	$272.74 98.5%	$266.14 94.8%	266.14 96.1%	307.99 106.8%
3. Apr 8, 2024	$294.41 102.1%	$290.40 104.8%	$277.52 98.9%	277.52 100.2%	294.41 102.1%
4. Apr 9, 2024	$277.04 96.0%	$246.69 89.1%	$287.98 102.6%	246.69 89.1%	287.98 99.8%
5. Apr 10, 2024	$296.04 102.6%	$277.28 100.1%	$258.04 91.9%	258.04 93.2%	296.04 102.6%
6. Apr 11, 2024	$287.11 99.5%	$276.21 99.7%	$187.96 67.0%	187.96 67.9%	287.11 99.5%
7. Apr 12, 2024	$295.87 102.6%	$287.33 103.7%	$334.16 119.1%	287.33 103.7%	334.16 115.9%
8. Apr 13, 2024	$294.95 102.3%	$274.23 99.0%	$286.69 102.2%	274.23 99.0%	294.95 102.3%
9. Apr 14, 2024	$303.86 105.3%	$334.79 120.9%	$277.64 98.9%	277.64 100.2%	334.79 116.1%
10. Apr 15, 2024	$307.69 106.7%	$249.03 89.9%	$273.65 97.5%	249.03 89.9%	307.69 106.7%
11. Apr 16, 2024	$256.18 88.8%	$268.63 97.0%	$204.15 72.7%	204.15 73.7%	268.63 93.1%
12. Apr 17, 2024	$235.19 81.5%	$209.12 75.5%	$213.51 76.1%	209.12 75.5%	235.19 81.5%
13. Apr 18, 2024	$293.40 101.7%	$289.18 104.4%	$309.42 110.3%	289.18 104.4%	309.42 107.3%
14. Apr 19, 2024	$287.54 99.7%	$314.83 113.7%	$306.06 109.1%	287.54 103.8%	314.83 109.1%
15. Apr 20, 2024	$278.66 96.6%	$217.90 78.7%	$302.72 107.9%	217.90 78.7%	302.72 105.0%

I want to sum row values across multiple columns…

The **ROW SUM** function works similarly to the **ROW MAX** and **ROW MIN** functions. It allows you to put in as many metrics (that is, columns) as you want, and then calculates across those columns for each row. In the case of the **ROW SUM**, it takes the value in each column and adds them together, providing a total for each row of your table.



To build this metric, you need two or more metrics (you can technically build it with only one, but the sum would be the same as the values for the single column, so not very useful). You can use completely separate metrics, or you can use the same metric but with different segments (such as device type). This can be useful for when you want to see different customer segments but also a grand total for them combined. If you are using a dimension, such as device type, it's possible you'll only want to see your top two or three devices (which are probably mobile phones, tablets, and 'other', i.e. desktops), but there are likely lots of other devices with small amounts of traffic (like gaming systems, TVs, e-readers, and so on). Using a segment for each of the device types you want to see will let you sum across those, including only the values you specify.

When putting this metric in your table, it will create a single sum for all of the columns you indicated in the function. This will provide a total for each row individually.

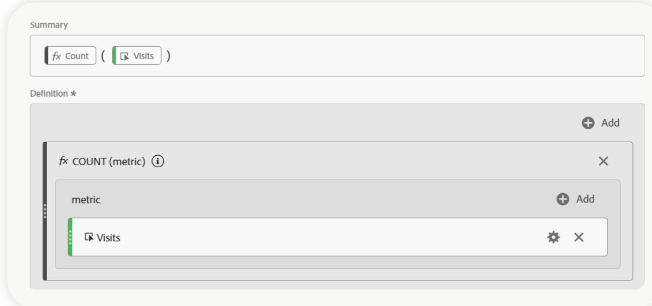
Row Sum

	Online Orders						Row Sum (Orders by Device Type)	
	Other		Mobile Phone		Tablet			
Day ↑ Page: 1 / 1 Rows: 25	Apr 13,557	out of 13,557	Apr 6 758	out of 758	Apr 6 456	out of 456	Apr 14,771	out of 14,771
1. Apr 6, 2024	1,035	7.6%	56	7.4%	35	7.7%	1,126	7.6%
2. Apr 7, 2024	1,719	12.7%	73	9.6%	65	14.3%	1,857	12.6%
3. Apr 8, 2024	753	5.6%	41	5.4%	33	7.2%	827	5.6%
4. Apr 9, 2024	778	5.7%	49	6.5%	28	6.1%	855	5.8%
5. Apr 10, 2024	747	5.5%	49	6.5%	31	6.8%	827	5.6%
6. Apr 11, 2024	715	5.3%	36	4.7%	8	1.8%	759	5.1%
7. Apr 12, 2024	762	5.6%	41	5.4%	18	3.9%	821	5.6%
8. Apr 13, 2024	740	5.5%	41	5.4%	20	4.4%	801	5.4%
9. Apr 14, 2024	721	5.3%	46	6.1%	28	6.1%	795	5.4%
10. Apr 15, 2024	697	5.1%	44	5.8%	17	3.7%	758	5.1%
11. Apr 16, 2024	701	5.2%	57	7.5%	28	6.1%	786	5.3%
12. Apr 17, 2024	1,095	8.1%	52	6.9%	26	5.7%	1,173	7.9%
13. Apr 18, 2024	1,622	12.0%	107	14.1%	65	14.3%	1,794	12.1%
14. Apr 19, 2024	735	5.4%	33	4.4%	32	7.0%	800	5.4%
15. Apr 20, 2024	737	5.4%	33	4.4%	22	4.8%	792	5.4%

- **I want to count how many non-zero values are in a dimension…**

The **COUNT** function is another great descriptive metric. This one allows you to count the number of items in a dimension that have a value greater than 0 for a given metric. This is listed as a **COUNT** of the non-zero rows in your table, but like the other column metrics, this counts all of the rows in the dimension regardless of whether they are in the table. It doesn't perform any calculations using the metric you specify; rather, it uses that metric as a type of test. If the value is not zero, it adds to the **COUNT**.

This has a few different use cases. First, it can be used as a part of calculations for other metrics where you need a row total. Second, it can be useful as a summary visualisation. If you're creating a dashboard with lots of graphs and visualisations, you might end up hiding the source table. So, if your users want to know how many unique items there are, rather than having to show the table, you can simply use the **COUNT** function. To build it, all you need is your metric of interest.



When you put it in your table it will look at what the value of that metric would be for each row, and count those above zero. The **COUNT** doesn't change based on the individual row and will return the same number on every row. If you want to limit the items in the dimension included in the calculation, you need to use a segment, as methods such as 'display only selected rows' will not impact the function's calculation.

Count - Page Dimension

Page Page: 1 / 831 › Rows: 10 1-10 of 8,305	Visits 209,760 out of 253,588	Count (Visits) 8,305 out of 8,305
1. home	88,867 42.4%	8,305 100.0%
2. category 5	41,702 19.9%	8,305 100.0%
3. category 2	40,603 19.4%	8,305 100.0%
4. category 4	39,224 18.7%	8,305 100.0%
5. category 3	38,961 18.6%	8,305 100.0%
6. articles	36,283 17.3%	8,305 100.0%
7. app: launch	34,906 16.6%	8,305 100.0%
8. category 1	34,350 16.4%	8,305 100.0%
9. forum	33,447 15.9%	8,305 100.0%
10. search results	32,804 15.6%	8,305 100.0%

Count - Page Dimension

Page Page: 1 / 1 Rows: 400 1-5 of 5	Visits Apr 6 249,357	Count (Visits) Apr 6 41,530
1. home	88,867 35.6%	8,306 20.0%
2. category 5	41,702 16.7%	8,306 20.0%
3. category 2	40,603 16.3%	8,306 20.0%
4. category 4	39,224 15.7%	8,306 20.0%
5. category 3	38,961 15.6%	8,306 20.0%

■ I want to count how many unique elements are in a dimension…

The **ROW COUNT** function is quite unique. It's one of only two functions (at the time of writing) that don't take any conditions at all when being built. This function counts the number of unique elements within a dimension, essentially, the number of rows. However, similar to the various column metrics, this counts all the unique dimensions in your time period regardless of whether or not they are in the table.

Row Count

	Row Count
Page 0-0 of 0	0 out of 0

When you put the metric in the table, it can't be your only metric. Unlike most of the other functions which work independently, this one does need at least one other metric – or more – to populate data. Here the **ROW COUNT** is counting the number of unique values in the dimension. If you have the entire dimension in the table, **ROW COUNT** will equal the number of rows. If you've limited the rows by dropping only part of a dimension or by using "display only selected rows" it will still count the entire dimension. It will only count a part of a dimension if you use a segment to limit the data.

Row Count

	Visits	Row Count
Page Page: 1 / 831 › Rows:	209,760 out of 253,588	8,305 out of 8,305
1. home	88,867 42.4%	8,305 100.0%
2. category 5	41,702 19.9%	8,305 100.0%
3. category 2	40,603 19.4%	8,305 100.0%
4. category 4	39,224 18.7%	8,305 100.0%
5. category 3	38,961 18.6%	8,305 100.0%
6. articles	36,283 17.3%	8,305 100.0%
7. app: launch	34,906 16.6%	8,305 100.0%
8. category 1	34,350 16.4%	8,305 100.0%
9. forum	33,447 15.9%	8,305 100.0%
10. search results	32,804 15.6%	8,305 100.0%

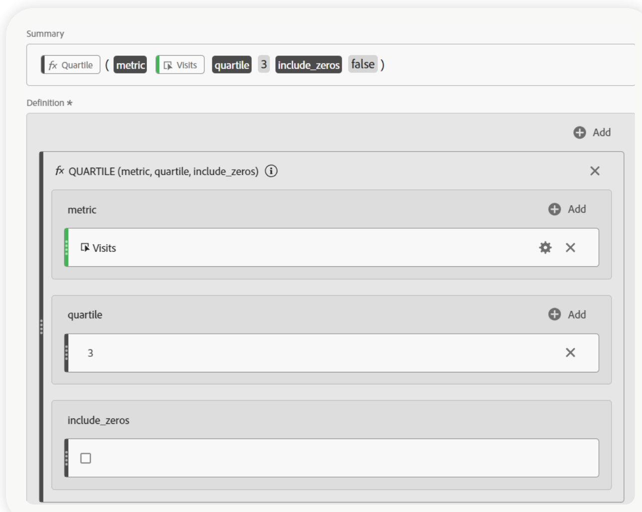
I want to identify a specific point in my distribution…

QUARTILE and **PERCENTILE** are two powerful functions to help you understand the distribution of your data. The number associated with a **QUARTILE** or a **PERCENTILE** represents how many values fall at or below that point. For example, a value that is at the 25th percentile means that, in the entire distribution, 25% of the values fall at or below that number.

A **QUARTILE**, like the name suggests, returns the values at the 0%, 25%, 50%, 75%, or 100% thresholds (the quarters). A **PERCENTILE** can be any point between 0% and 100%. With both of these functions the dimension you use for your rows is important, as it will use the values on each row to calculate where the quartiles or percentiles fall.

One use for these functions is to create thresholds or benchmarks for your values. If you have data from a particular time period, and you want to know if your present data is reaching at least a certain percentage of that amount, you can create a **PERCENTILE** or **QUARTILE** metric. Remember that this still needs to have a dimension with rows to count against. But you can then use a summary number visualisation to present the data in a dashboard.

To build the **QUARTILE**, all you need is your metric of interest and to indicate which **QUARTILE** you want to return. This will be indicated by a number between 0 and 4, with 0 being the lowest value (0%) and 4 being the highest value (100%). If you use a number greater than 4, it will just default to the 100% quartile. The 0% and 100% are functionally similar to the **MIN** and **MAX** functions that are available. For the other quartiles, 1 = 25%, 2 = 50%, and 3 = 75%.



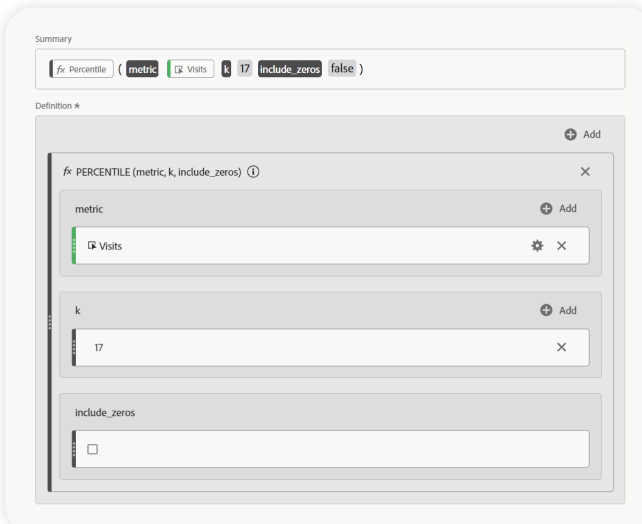
When you put the **QUARTILE** function in your table, the value returned is the number (0 to 4) associated with the quartile indicated in the metric. To determine this, it will look at the entire dimension and return the value based on that, not just the data in the table. This works the same as the various column functions – regardless of how many items from the dimension are present in the table, the entire dimension for that time period will be analysed. The only way to limit the items being considered is with a segment. Simply using 'display only selected rows' will not have an impact.

Quartiles

Day ↑	Visits	Quartile q=0 (Visit)	Quartile q=1 (Visit)	Quartile q=2 (Visit)	Quartile q=3 (Visit)	Quartile q=4 (Visit)	Quartile q=5 (Visit)
Page: 1 / 1 Rows: 25 1-15 of	Apr 6 253,588 out of 253,588	Apr 6 12,923 out of 12,923	Apr 6 13,173 out of 13,173	Apr 6 15,306 out of 15,306	Apr 6 19,190 out of 19,190	Apr 6 27,430 out of 27,430	Apr 6 27,430 out of 27,430
1. Apr 6, 2024	18,963 7.5%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
2. Apr 7, 2024	26,692 10.5%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
3. Apr 8, 2024	15,573 6.1%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
4. Apr 9, 2024	19,335 7.6%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
5. Apr 10, 2024	13,023 5.1%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
6. Apr 11, 2024	13,272 5.2%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
7. Apr 12, 2024	13,205 5.2%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
8. Apr 13, 2024	12,923 5.1%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
9. Apr 14, 2024	13,173 5.2%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
10. Apr 15, 2024	15,145 6.0%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
11. Apr 16, 2024	18,238 7.2%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
12. Apr 17, 2024	19,190 7.6%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
13. Apr 18, 2024	27,430 10.8%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
14. Apr 19, 2024	13,083 5.2%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%
15. Apr 20, 2024	15,306 6.0%	12,923 100.0%	13,173 100.0%	15,306 100.0%	19,190 100.0%	27,430 100.0%	27,430 100.0%

To build the **PERCENTILE** function, in addition to your metric of interest, you need to indicate the percentile that you want to return. Again, this number represents the amount of values that would be at or below that point. So '*k* = 17' would return the number that has 17% of the values below it. You can also use the **PERCENTILE** function to return a **QUARTILE** by putting in 25, 50, or 75.

When using **PERCENTILE** in your table, it works the same as **QUARTILE**. Regardless of how many items in the dimension are in your table, it will consider all the values in the dimension for your time period in its calculations. The number returned is the value associated with the *k* percentile indicated in the metric.

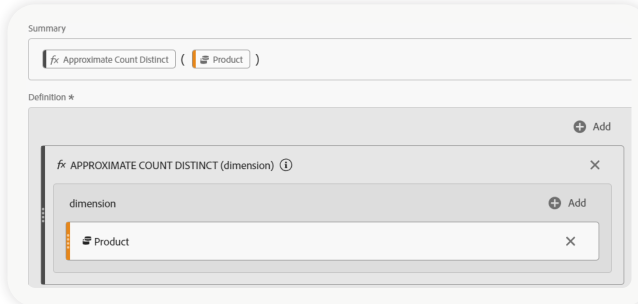


Percentiles

Day ↑	Visits	Percentile p=0 (Visits)	Percentile p=17 (Visits)	Percentile p=38 (Visits)	Percentile p=50 (Visits)	Percentile p=66 (Visits)	Percentile p=84 (Visits)	Percentile p=100 (Visits)
Page: 1 / 1 Rows: 25	Apr 6 253,588 out of 253,588	Apr 6 12,923 out of 12,923	Apr 6 13,066 out of 13,066	Apr 6 13,422 out of 13,422	Apr 6 15,306 out of 15,306	Apr 6 18,644 out of 18,644	Apr 6 22,572 out of 22,572	Apr 6 27,430 out of 27,430
1. Apr 6, 2024	18,963 7.5%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
2. Apr 7, 2024	26,692 10.5%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
3. Apr 8, 2024	15,573 6.1%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
4. Apr 9, 2024	19,335 7.6%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
5. Apr 10, 2024	13,023 5.1%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
6. Apr 11, 2024	13,272 5.2%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
7. Apr 12, 2024	13,205 5.2%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
8. Apr 13, 2024	12,923 5.1%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
9. Apr 14, 2024	13,173 5.2%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
10. Apr 15, 2024	15,145 6.0%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
11. Apr 16, 2024	18,238 7.2%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
12. Apr 17, 2024	19,190 7.6%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
13. Apr 18, 2024	27,430 10.8%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
14. Apr 19, 2024	13,083 5.2%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%
15. Apr 20, 2024	15,306 6.0%	12,923 100.0%	13,066 100.0%	13,422 100.0%	15,306 100.0%	18,644 100.0%	22,572 100.0%	27,430 100.0%

I want to count unique items in a dimension…

The **APPROXIMATE COUNT DISTINCT** function is a bit different than the others in that when you're building it, it doesn't require a metric. Instead, this function requires a dimension. The purpose of this function is to count the number of unique values in your dimension of interest against another dimension – for example, the unique number of products in each order.



One caveat with this is that it does say 'approximate', which means it might not be completely accurate. Adobe uses a 'HyperLogLog methodology' to do the calculation which is typically 95% accurate. Regardless, this is one of the most versatile and useful functions that Adobe Analytics has to offer. Whether it's counting the number of products against order numbers, the number of products against search terms, the number of unique pages per day, or so on, this can be very useful to deduplicate counts.

When building this metric, all that is required is a single dimension. It requires the entire dimension and doesn't allow for bringing in only specific items. It also won't allow you to use segments or metrics as conditions.

When putting this metric in your table, you do need to put it against another dimension. Ideally you want to use a different dimension than what you built the metric with (or else all of your rows are just going to be a series of 1's). This function will look at the dimension value in your row, count approximately how many unique values of the dimension in the function are associated with it, and return that number.

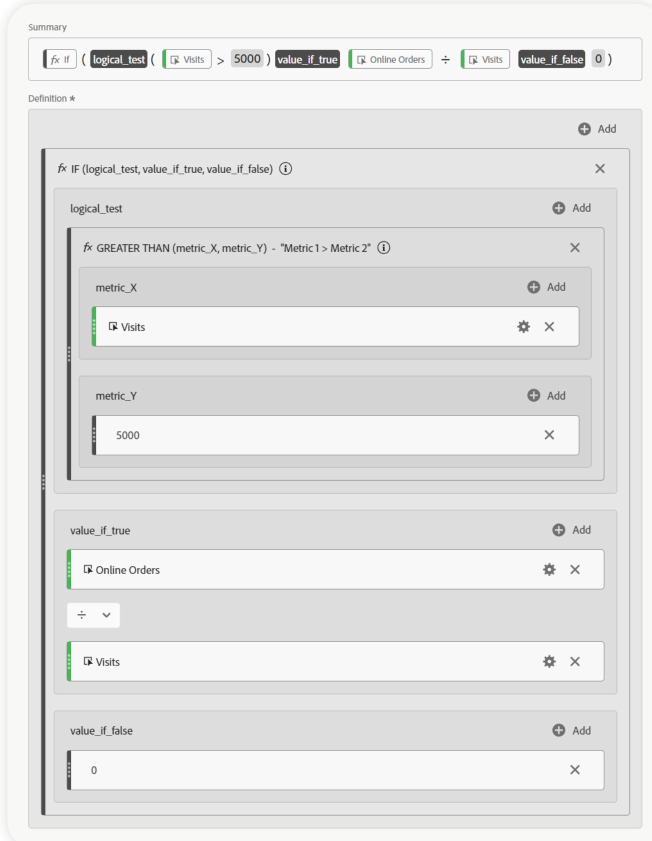
Distinct Products

	Approx. Count Distinct (Products)
Order ID Page: 1 / 4,902 › Rows: 5 1-5 of 24,509	**51** out of 51
1. orderID:1713439465492	9 17.8%
2. orderID:1713402101964	7 13.9%
3. orderID:1712389762992	7 13.9%
4. orderID:1712439624021	7 13.9%
5. orderID:1713564648873	7 13.9%

Logical Comparisons.

- **I want to only return values in a specific case…**

One of the questions I see a lot is how to filter values in a column to be above or below a certain threshold. While you can't remove them completely, you can essentially sort them using the **IF** function or one of the logical comparisons.



With just the **IF** function, it performs a logical test and then returns a value of true or false, depending on the outcome. While it is possible to use the presence or absence of a single metric as your logical test, it often makes sense to use one of many other comparison functions, such as **GREATER THAN, LESS THAN, EQUAL, NOT EQUAL**, and so on.

To build out our **IF** function, we need three components: our logical test, the value if it is true, and the value if it is false. Often for the logical test I end up nesting another function. For the value if true or the value if false, there are two general options. The first is to return a static number (such as true = 1 and false = 0). This is good if your goal is to simply sort your rows based on the logical test. The other option is to return another metric for true or false values. For example, I want to see which products have the best conversion rate, but I want to filter out products with low traffic. For those products that meet my logical test (visits **GREATER THAN** x), I can return the conversion rate if it's true, and return a 0 if it's false.

When you put this in your table, the comparison will evaluate each row individually and return the true or false value depending on the result of the logical test. In our example with conversion rate, we now have a column that only has conversion for products with visits over a specific threshold. I can then combine that with the setting "interpret 0 as no value", leaving me with a column that shows the conversion rate but only for specific articles. I can now sort by conversion rate to see how my top products are converting, without being impacted with low-traffic items.

Sorting before If Function

Product (Page: 1 / 6, Rows: 10)	Visits 101,739 out of 253,588		Conversion Rate 17.94% out of 7.20%
1. prd1042	133	0.1%	19.55%
2. prd1022	1,398	1.4%	15.31%
3. prd1045	2,484	2.4%	15.18%
4. prd1034	1,922	1.9%	15.14%
5. prd1032	2,490	2.4%	15.06%
6. prd1026	7,649	7.5%	14.92%
7. prd1005	1,209	1.2%	14.89%
8. prd1016	4,469	4.4%	14.79%
9. prd1002	5,325	5.2%	14.78%
10. prd1041	1,582	1.6%	14.66%

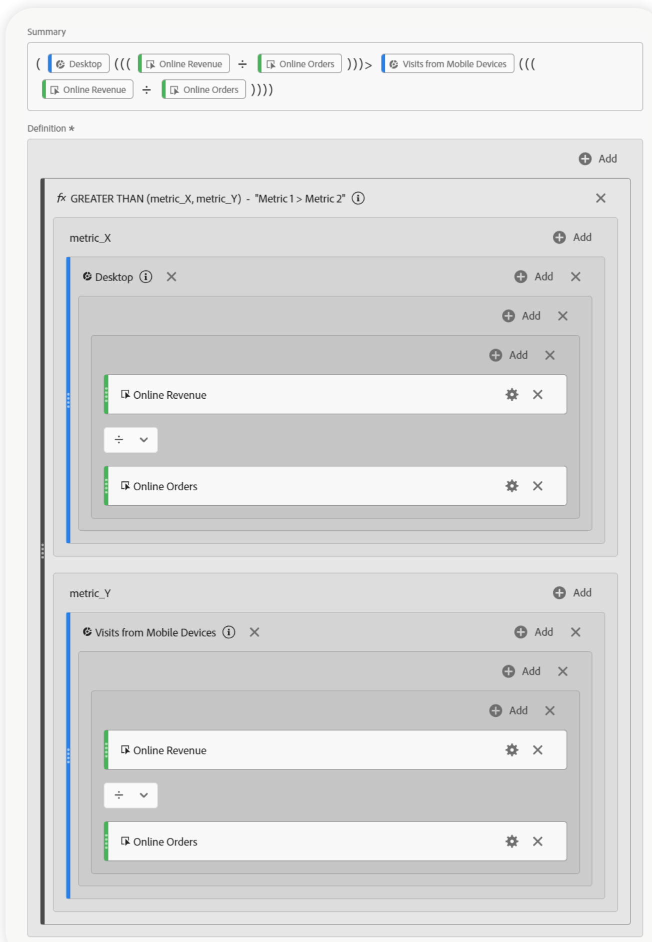
Sorting After If Function

Product (Page: 1 / 6, Rows: 10)	Visits 101,739 out of 253,588		Conversion Rate 17.94% out of 7.20%	Conversion (Visit > 5000) 17.94% out of 7.20%
1. prd1026	7,649	7.5%	14.92%	14.92%
2. prd1002	5,325	5.2%	14.78%	14.78%
3. prd1029	5,358	5.3%	14.43%	14.43%
4. prd1009	5,070	5.0%	14.26%	14.26%
5. prd1006	5,240	5.2%	14.16%	14.16%
6. prd1025	4,955	4.9%	14.29%	
7. prd1033	4,955	4.9%	13.91%	
8. prd1017	4,901	4.8%	14.38%	
9. prd1036	4,885	4.8%	14.02%	
10. prd1050	4,744	4.7%	13.62%	

I want to identify values above or below a threshold…

There are functions that can allow you to compare two values and return a true or false value based on the outcome. Functions such as **GREATER THAN**, **GREATER THAN OR EQUAL TO**, **LESS THAN**, and **LESS THAN OR EQUAL TO** give you the ability to compare two metrics to one another. It will always compare the first metric (metric X) to see if it is greater or less than the second metric (metric Y). It produces a Boolean result with true = 1 and false = 0.

To build any of these four functions you need two metrics. They can be events, calculated metrics, or even static numbers. This can be useful for comparing daily values to a specific benchmark or threshold, and identifying those results that meet the mark. The order of the two metrics is important, as the comparison is always directional.



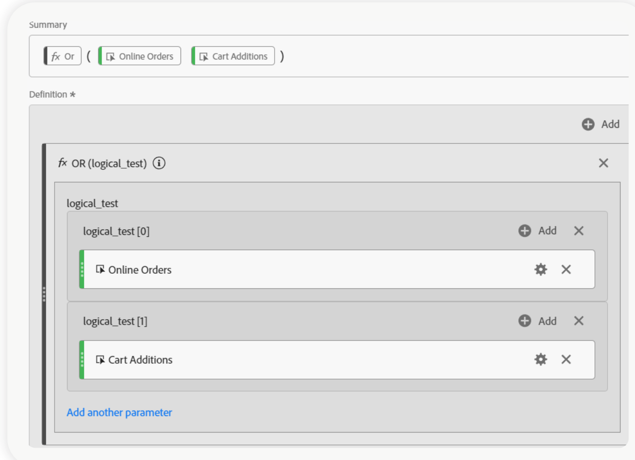
When you put the metric in your table, it will display the Boolean values in the column. If we want to identify days where desktop devices had the higher AOV, we could now sort based on our comparison column. We do also have the option here to select 'interpret 0 as no value', which will remove all the 0s and only leave behind the values associated with true, giving us a cleaner table. These functions are great for creating a column to sort your data. If you want to see the values of a dimension based on a comparison criteria, sorting by your **GREATER THAN** or **LESS THAN** column is an easy way to do that. Alternatively, you can use these nested in another function to return values based on different conditions.

Greater Than

	Avg. Order Value (AOV)		Desktop Greater than Mobile (AOV)
	Desktop	Visits from Mobile Devices	
Day ↑ Page: 1 / 1 Rows: 25	$288.44 out of $288.44	$285.74 out of $285.74	1 out of 1 (Apr 6)
1. Apr 6, 2024	$299.80 103.9%	$303.26 106.1%	
2. Apr 7, 2024	$307.99 106.8%	$295.89 103.6%	1 100.0%
3. Apr 8, 2024	$294.41 102.1%	$282.06 98.7%	1 100.0%
4. Apr 9, 2024	$277.04 96.0%	$252.16 88.2%	1 100.0%
5. Apr 10, 2024	$296.04 102.6%	$299.26 104.7%	
6. Apr 11, 2024	$287.11 99.5%	$286.91 100.4%	1 100.0%
7. Apr 12, 2024	$295.87 102.6%	$284.07 99.4%	1 100.0%
8. Apr 13, 2024	$294.95 102.3%	$291.27 101.9%	1 100.0%
9. Apr 14, 2024	$303.86 105.3%	$289.01 101.1%	1 100.0%
10. Apr 15, 2024	$307.69 106.7%	$288.10 100.8%	1 100.0%
11. Apr 16, 2024	$256.18 88.8%	$266.82 93.4%	
12. Apr 17, 2024	$235.19 81.5%	$249.45 87.3%	
13. Apr 18, 2024	$293.40 101.7%	$294.29 103.0%	
14. Apr 19, 2024	$287.54 99.7%	$302.03 105.7%	
15. Apr 20, 2024	$278.66 96.6%	$280.13 98.0%	

- **I want to return a true/false value based on one or more conditions…**

Being able to return a true/false value when one or more conditions are true can help you to identify specific parts of your data and sort it in your tables. The conditions can be evaluated with the **AND** and **OR** functions. The **AND** function evaluates one or more conditions and returns a true value only if all of the conditions are true (and returns false if even one condition is false); whereas the **OR** function returns true if at least one condition is true (and returns false only if all are false).



To build this, you need to add individual parameters. The builder for both the **AND** and **OR** functions defaults to only having one parameter. You could use just one, but the purpose is really to evaluate multiple conditions, so you can click "add another parameter" at the bottom. You can add as many parameters as you want. I've done these types of functions with about 25 parameters before.

When putting these in your table, it will evaluate each row against your conditions. In this case it is evaluating if online orders exist or if cart additions exist for that row of the dimension. For the **OR**, if at least one condition is true, it returns 1 (true), for the **AND**, if all conditions are true, it returns as 1 (true). You can also nest other functions in here, such as **GREATER THAN** or **LESS THAN**. Or you can nest these functions inside others, such as **IF**, to change the type of data it returns. For example, say you want to return a list of products that have both cart additions and orders. By entering both of those criteria in the **AND** function, it will return a 1 for those that meet all criteria, and a 0 for those that don't. This can make it easy to sort and identify products that are being seen (because if they're never seen they won't be in your report) but not interacted with.

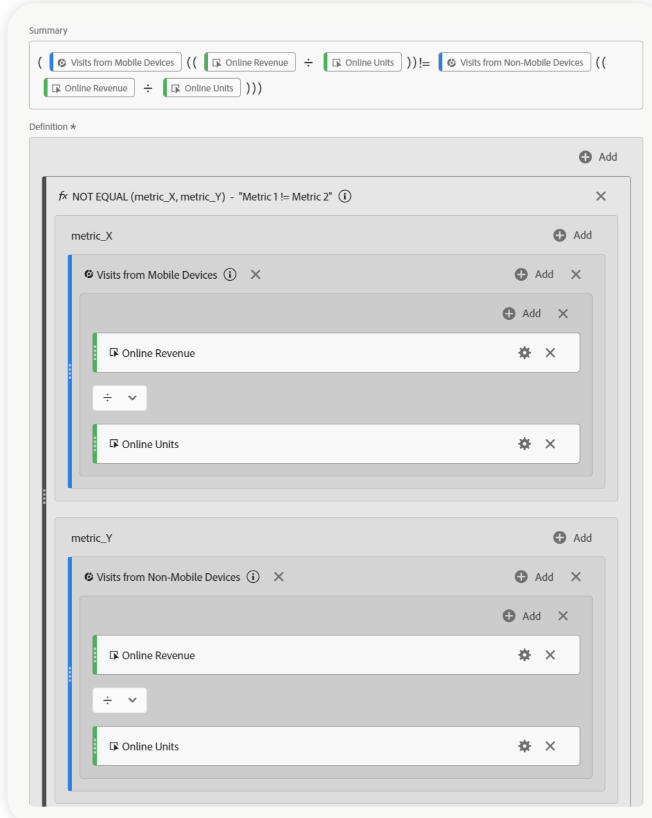
AND & OR

Product (Page: 1 / 6, Rows: 10)	Yesterday: Visits	Cart Additions	Online Orders	Cart Add OR Order	Cart Add AND Order
	12,851 out of 30,067	10,597 out of 10,597	2,302 out of 2,302	1 out of 1	1 out of 1
1. prd1001	8 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%
2. prd1046	10 0.1%	4 0.0%	1 0.0%	1 100.0%	1 100.0%
3. prd1013	15 0.1%	5 0.0%	0 0.0%	1 100.0%	0 0.0%
4. prd1042	19 0.1%	9 0.1%	6 0.3%	1 100.0%	1 100.0%
5. prd1014	20 0.2%	6 0.1%	1 0.0%	1 100.0%	1 100.0%
6. prd1037	117 0.9%	62 0.6%	11 0.5%	1 100.0%	1 100.0%
7. prd1008	119 0.9%	68 0.6%	18 0.8%	1 100.0%	1 100.0%
8. prd1005	154 1.2%	74 0.7%	18 0.8%	1 100.0%	1 100.0%
9. prd1047	170 1.3%	100 0.9%	23 1.0%	1 100.0%	1 100.0%
10. prd1049	171 1.3%	95 0.9%	24 1.0%	1 100.0%	1 100.0%

- **I want to return a true/false when exactly two conditions are the same....**

Similar to many other comparison functions, the **EQUAL** and **NOT EQUAL** functions will return a Boolean (true/false) value depending on the results of the conditions. When it compares the two values, it considers each row individually and returns a value based on that comparison. It does consider all of the decimal places of the metric, regardless of how many are visible. Meaning, if you have a calculated metric like average order value, even though you might display it as a currency and show two decimal places, the comparison will be for the entire number. Although at two decimal places two values might look the same due to rounding, it could still return as not being equal.

To build either of these you need exactly two conditions. These can be two different metrics, or the same metric but with different segments around them.



When you put either of these in a table, it will look at each row and return a true or false (1 or 0) depending on if it meets the conditions. The **EQUAL** and **NOT EQUAL** functions are mutually exclusive – if you put the same conditions in both functions, only one of them can be true at a time. The one you decide to use depends on how you want the results to appear. If you select 'interpret 0 as no value', then only the true values will remain, giving you an easy way to visually identify your data.

Equal & Not Equal

Product	Revenue Per Unit (RPU)		Equal RPU (Device Type)	Not Equal RPU (Device Type)
	Visits from Mobile Devices	Visits from Non-Mobile Devices		
Page: 1 / 1 Rows: 10	Apr 6 ↓ 1,095.83	Apr 6 1,095.83	Apr 6 6	Apr 6 4
1. prd1046	192.11 17.5%	192.11 17.5%	1 16.7%	0 0.0%
2. prd1024	190.77 17.4%	190.77 17.4%	1 16.7%	0 0.0%
3. prd1034	143.63 13.1%	143.63 13.1%	0 0.0%	1 25.0%
4. ï»¿prd1001	132.50 12.1%	132.50 12.1%	1 16.7%	0 0.0%
5. prd1028	113.43 10.4%	113.43 10.4%	0 0.0%	1 25.0%
6. prd1037	109.70 10.0%	109.70 10.0%	1 16.7%	0 0.0%
7. prd1013	80.40 7.3%	80.40 7.3%	1 16.7%	0 0.0%
8. prd1014	56.23 5.1%	56.23 5.1%	0 0.0%	1 25.0%
9. prd1022	51.11 4.7%	51.11 4.7%	1 16.7%	0 0.0%
10. prd1005	25.95 2.4%	25.95 2.4%	0 0.0%	1 25.0%

▪ I want to identify when a single condition is false…

The **NOT** function is used to check for the absence of a single condition. When placing a metric in the function, it evaluates if the metric is present for each row, and if it is absent, returns a true value (1). This function is another great way that you can create a column to sort your data based on a criterion, such as whether or not a product has ever been ordered.

To build this, you only need a single condition: the metric you want to evaluate.

When you place this in a table, it will identify the rows where your condition is not true. You can combine this with other logical comparisons by nesting them to add more conditions to your comparison.

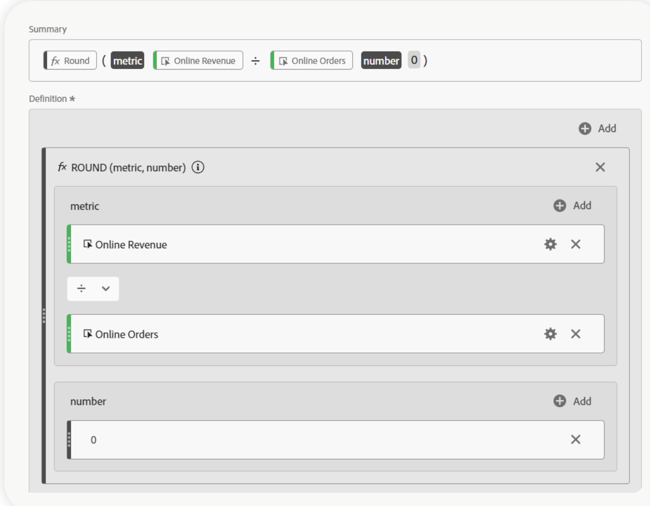
Not

	Yesterday					
	Visits		Online Orders		Not Orders	
Product Page: 1 / 6 Rows: 10	12	12,851 out of 30,067	12	2,302 out of 2,302	12 AM	0 out of 0
1. prd1001	8	0.1%	0	0.0%	1	1,000.0+%
2. prd1046	10	0.1%	1	0.0%	0	0.0%
3. prd1013	15	0.1%	0	0.0%	1	1,000.0+%
4. prd1042	19	0.1%	6	0.3%	0	0.0%
5. prd1014	20	0.2%	1	0.0%	0	0.0%
6. prd1037	117	0.9%	11	0.5%	0	0.0%
7. prd1008	119	0.9%	18	0.8%	0	0.0%
8. prd1005	154	1.2%	18	0.8%	0	0.0%
9. prd1047	170	1.3%	23	1.0%	0	0.0%
10. prd1049	171	1.3%	24	1.0%	0	0.0%

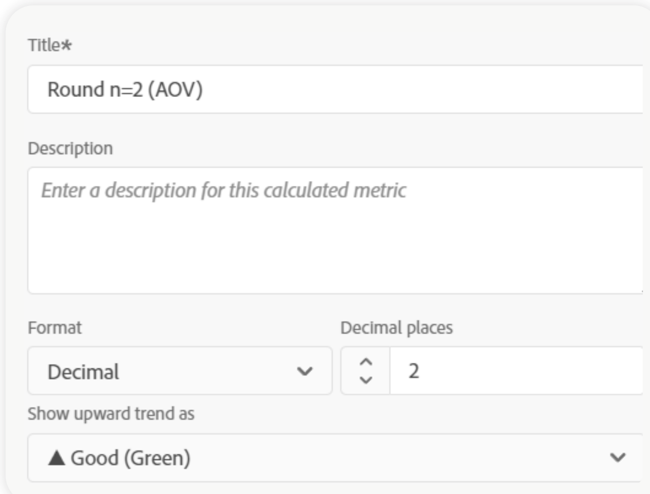
Data Transformations.

▪ I want to round my values to a certain decimal point…

There are three functions that modify how your numbers are presented, without changing the underlying data. The **ROUND**, **FLOOR**, and **CEILING** functions allow you to increase or decrease the number of decimal points associated with your number or round it to a whole number.



The **ROUND** function asks for the metric of interest and the number of decimal points you want. This has to be either 0 or a positive number. This function is one of the less useful ones, as the same result can be achieved by changing the number of decimal points associated with a metric when you create it.



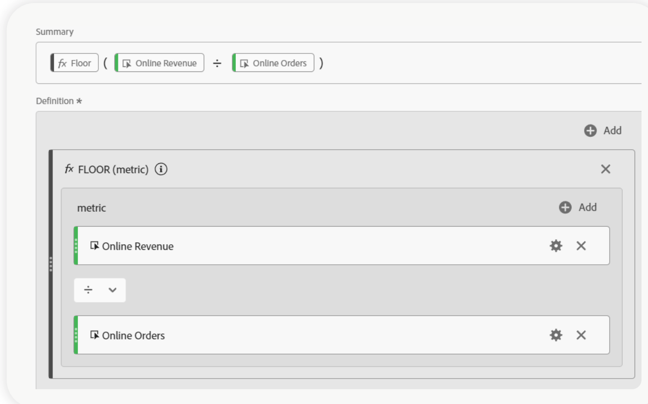
Once you have the metric created and put it in the table, it will return your metric of interest rounded to the specified number of decimal points. However, this is overwritten by the use of "decimals" in the metric settings. If you create a calculated metric using the function **ROUND**, and set it to two decimal points, but the selected decimal places are 0, your metric will be rounded to the lower amount. If your selected decimal points are greater than your rounding amount, the extra decimal places will be 0's.

Rounding

Day ↑	Avg. Order Value (AOV)		Round n=0 (AOV)		Round n=2 (AOV)		Round n=5 (AOV)		Round n=2, 4 decimals (AOV)	
Page: 1 / 3 › Rows: 5	$287.78 out of $287.78		288 out of 288		287.78 out of 287.78		287.77989 out of 287.77989		287.7800 out of 287.7800	
1. Apr 6, 2024	$300.70	104.5%	301	104.5%	300.70	104.5%	300.69768	104.5%	300.7000	104.5%
2. Apr 7, 2024	$304.94	106.0%	305	105.9%	304.94	106.0%	304.94011	106.0%	304.9400	106.0%
3. Apr 8, 2024	$291.42	101.3%	291	101.0%	291.42	101.3%	291.41863	101.3%	291.4200	101.3%
4. Apr 9, 2024	$270.99	94.2%	271	94.1%	270.99	94.2%	270.99032	94.2%	270.9900	94.2%
5. Apr 10, 2024	$296.91	103.2%	297	103.1%	296.91	103.2%	296.90984	103.2%	296.9100	103.2%

■ I want to round all my values up or down to a whole number…

The **FLOOR** and **CEILING** functions are a bit more useful than simple rounding. These round metrics to a whole number, **CEILING** always rounding up and **FLOOR** always rounding down. By default, these don't have any decimal places associated with them, but you can add decimals in the metric builder, although they will always be 0's. To build either of these metrics all you need is your metric of interest.



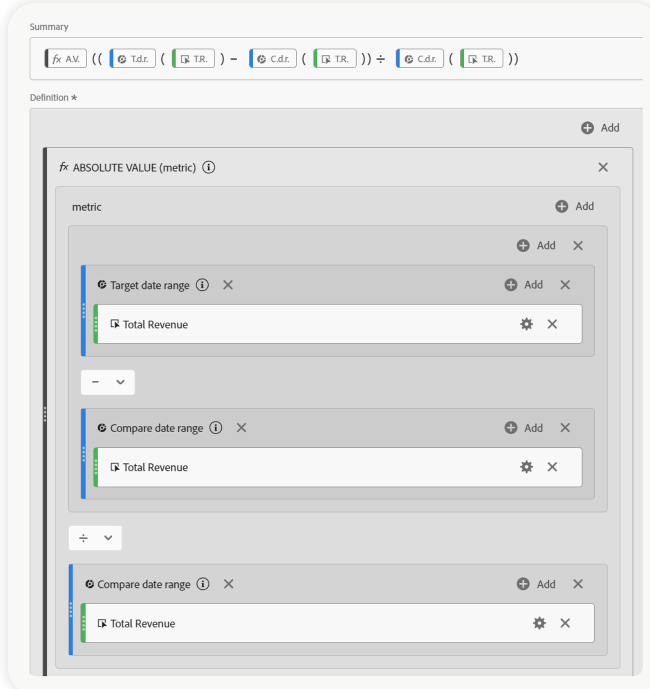
When putting the metric in the table, the resulting value will differ based on the information in each row. Using these functions as part of your calculated metrics are great for keeping the numbers clean in a table or in summary numbers. By default, these functions have no decimal places associated with them. Even if you change the metric settings to include decimal points, they will only return 0's in your table. There are times where always rounding up or down makes sense, such as when looking at an average amount of visits or orders, since you can't have half an order.

Floor & Ceiling

Day	Avg. Order Value (AOV)	Floor (AOV)	Ceiling (AOV)
Page: 1 / 1 Rows: 25	$287.78 out of $287.78	287 out of 287	288 out of 288
1. Apr 6, 2024	$300.70 104.5%	300 104.5%	301 104.5%
2. Apr 7, 2024	$304.94 106.0%	304 105.9%	305 105.9%
3. Apr 8, 2024	$291.42 101.3%	291 101.4%	292 101.4%
4. Apr 9, 2024	$270.99 94.2%	270 94.1%	271 94.1%
5. Apr 10, 2024	$296.91 103.2%	296 103.1%	297 103.1%
6. Apr 11, 2024	$287.06 99.8%	287 100.0%	288 100.0%
7. Apr 12, 2024	$293.48 102.0%	293 102.1%	294 102.1%
8. Apr 13, 2024	$294.04 102.2%	294 102.4%	295 102.4%
9. Apr 14, 2024	$299.73 104.2%	299 104.2%	300 104.2%
10. Apr 15, 2024	$302.49 105.1%	302 105.2%	303 105.2%
11. Apr 16, 2024	$259.12 90.0%	259 90.2%	260 90.3%
12. Apr 17, 2024	$238.57 82.9%	238 82.9%	239 83.0%
13. Apr 18, 2024	$293.64 102.0%	293 102.1%	294 102.1%
14. Apr 19, 2024	$291.41 101.3%	291 101.4%	292 101.4%
15. Apr 20, 2024	$279.03 97.0%	279 97.2%	280 97.2%

I want to see my numbers without a positive or negative indicator…

There can be instances (such as when doing statistical calculations) that it can be useful to see a number without any positive or negative value assigned to it. Using the **ABSOLUTE VALUE** function, it will return your metric of interest as an absolute value. While normally an **ABSOLUTE VALUE** refers to a number without a positive or negative indicator, in the case of Analysis Workspace, an **ABSOLUTE VALUE** will always have a positive attribution.



To build the metric all you need is the metric of interest. For this, it typically makes sense to use a metric that can return either positive or negative numbers. Some good examples of these would be comparison metrics, such as a year-over-year visit comp or the difference between two groups of customers.

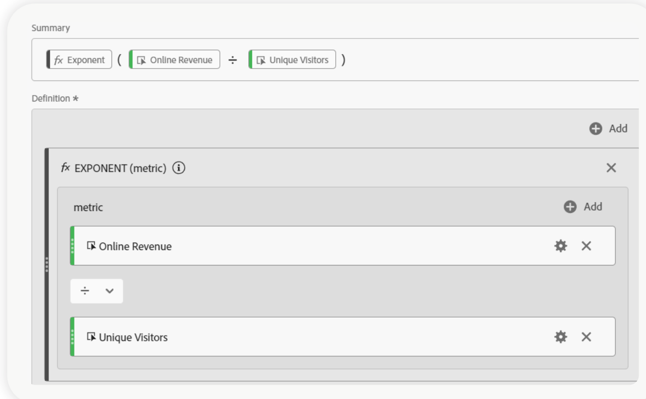
When placing this metric in your table, it doesn't change the actual numbers associated with your metric of interest, only their attribution. Any rows that previously had a negative value will now be treated as if they are positive (such as showing green in a conditional formatting). Any numbers that were already positive will not have any changes.

Absolute Value

Product Page: 1 / 1 Rows: 25	Revenue YoY Change Apr 6 ↓ 36.12%	Absolute Value (Revenue YoY Change) Apr 6 364.25%
1. prd1001	107.69%	107.69%
2. prd1013	47.21%	47.21%
3. prd1014	20.21%	20.21%
4. prd1019	7.41%	7.41%
5. prd1042	6.98%	6.98%
6. prd1005	6.33%	6.33%
7. prd1012	3.59%	3.59%
8. prd1007	0.77%	0.77%
9. prd1046	-12.66%	12.66%
10. prd1022	-14.91%	14.91%
11. prd1018	-15.05%	15.05%
12. prd1003	-18.66%	18.66%
13. prd1031	-21.40%	21.40%
14. prd1023	-23.03%	23.03%
15. prd1043	-24.33%	24.33%
16. prd1037	-34.02%	34.02%

I want to raise my data to an exponent…

The function **EXPONENT** allows you to multiply the constant *e* by a particular value and return the result. This is the inverse of the function **NATURAL LOG**. Using the exponent gets very large very fast, for example, a revenue per visitor of about $20 returns an exponent just shy of 1 billion. At some point Adobe Analytics starts converting large numbers into scientific notation before eventually just declaring a number 'infinity.' So, the exponent is best used with smaller numbers (if on its own), or as a part of a larger nested calculation.



To build this, you only need one condition: the metric that you want to use as the exponent against the constant *e*.

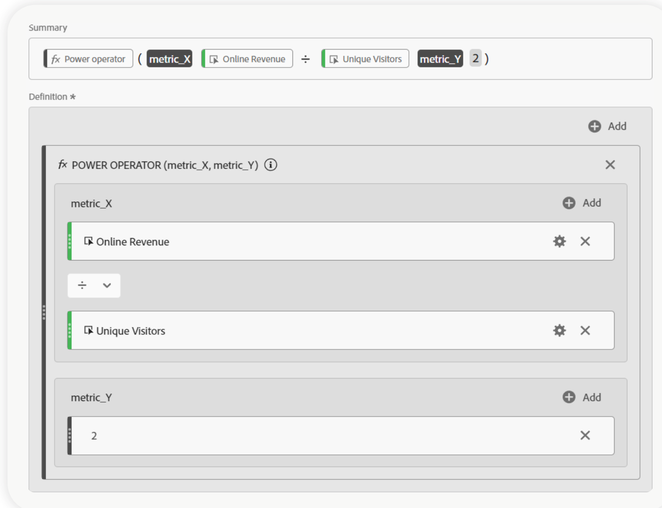
When putting this in your table, the value returned will be the exponent value associated with the metric value for that particular row. Used on its own, this function has limited use cases, but it can be used in conjunction with many of the statistical functions to provide insights and predictions.

Exponent

	Revenue / Visitor	Exponent (Revenue/Visitor)
Day ↑ Page: 1 / 3 › Rows: 5	$20.71 out of $20.71	985,266,535 out of 985,266,535
1. Apr 6, 2024	$22.18 107.1%	4,309,316,940 437.4%
2. Apr 7, 2024	$26.25 126.8%	252,145,878,534 1,000.0+%
3. Apr 8, 2024	$18.62 89.9%	121,996,665 12.4%
4. Apr 9, 2024	$14.41 69.6%	1,808,426 0.2%
5. Apr 10, 2024	$23.37 112.8%	14,091,735,597 1,000.0+%

▪ I want to raise my data to a specific power operator…

The **POWER OPERATOR** function allows you to raise your metric to a specified power. This can be squared, cubed, and so on, depending on your needs. As mentioned with the **EXPONENT** function, numbers can get very large quickly and Adobe will eventually convert to scientific notation for extremely large numbers before simply returning an 'infinity' value.



To build this, you need two conditions. The first is your metric of interest (metric X). The second is the power you want to raise metric X to (metric Y). You would use 2 to square your number, 3 to cube it, and so on. The space for metric Y will let you put any type of number in there – another metric, a static number or another function. While you can do anything, keep in mind what makes sense and what will be returned. Using a static number for your metric Y is often going to return the best results.

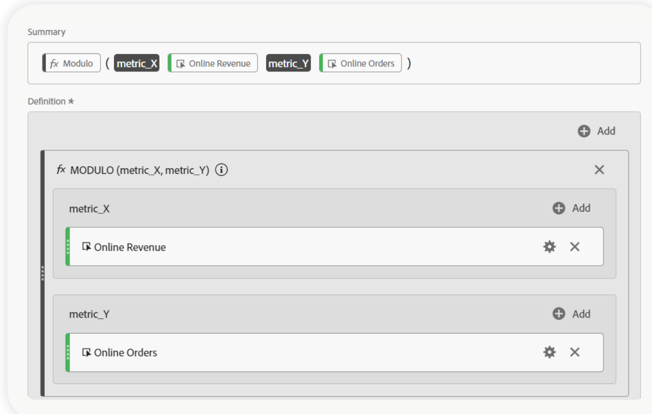
When putting this in your table it will return your metric to the specified power for each row individually. On its own this has limited use cases, but it can be used in conjunction with many of the statistical functions to provide insights and predictions.

● **Power Operator**

	Revenue / Visitor	Power Operator y=2 (Revenue/Visitor)
Day ↑ Page: 1 / 3 › Rows: 5	$20.71 out of $20.71	429 out of 429
1. Apr 6, 2024	$22.18 107.1%	492 114.8%
2. Apr 7, 2024	$26.25 126.8%	689 160.7%
3. Apr 8, 2024	$18.62 89.9%	347 80.8%
4. Apr 9, 2024	$14.41 69.6%	208 48.4%
5. Apr 10, 2024	$23.37 112.8%	546 127.3%

▪ I want to return the remainder of a number…

The **MODULO** function is one of the more interesting ones, in my opinion. A **MODULO** is a remainder left over after using Euclidean division. This is a function that isn't used too frequently, but can produce some interesting results, especially for various modelling purposes. Some potential uses would be to produce an even/odd alternate, limit numbers to a specific range, or look at the *n*th value in a repeating sequence.



Building the function is a lot easier than finding relevant use cases for it. To build the function all you need are two metrics. You could use static numbers instead of actual metrics too, depending on your use case.

When you put the metric in your table, it will return the remainder when metric X is divided by metric Y for each row.

Modulo

Day ↑	Online Revenue		Online Orders		Modulo (Revenue/Orders)	
Page: 1 / 1 Rows: 25	$5,251,408 out of $5,251,408		18,248 out of 18,248		14,232 out of 14,232	
1. Apr 6, 2024	$420,676	8.0%	1,399	7.7%	976	6.9%
2. Apr 7, 2024	$700,752	13.3%	2,298	12.6%	2,160	15.2%
3. Apr 8, 2024	$289,962	5.5%	995	5.5%	417	2.9%
4. Apr 9, 2024	$278,578	5.3%	1,028	5.6%	1,018	7.2%
5. Apr 10, 2024	$304,333	5.8%	1,025	5.6%	933	6.6%
6. Apr 11, 2024	$271,561	5.2%	946	5.2%	59	0.4%
7. Apr 12, 2024	$298,181	5.7%	1,016	5.6%	493	3.5%
8. Apr 13, 2024	$288,455	5.5%	981	5.4%	41	0.3%
9. Apr 14, 2024	$299,429	5.7%	999	5.5%	728	5.1%
10. Apr 15, 2024	$287,062	5.5%	949	5.2%	464	3.3%
11. Apr 16, 2024	$251,092	4.8%	969	5.3%	121	0.8%
12. Apr 17, 2024	$342,349	6.5%	1,435	7.9%	819	5.8%
13. Apr 18, 2024	$652,758	12.4%	2,223	12.2%	1,419	10.0%
14. Apr 19, 2024	$290,536	5.5%	997	5.5%	409	2.9%
15. Apr 20, 2024	$275,684	5.2%	988	5.4%	32	0.2%

▪ I want to return the root of a number…

The **SQUARE ROOT** and **CUBE ROOT** functions work in the opposite direction of the **POWER OPERATOR** function. Instead of making a number larger, this typically makes a number smaller. Unlike the **POWER OPERATOR**, these functions don't let you choose what power your number is raised to. The **SQUARE ROOT** will always raise the metric to 1/2, and the **CUBE ROOT** will always raise the metric to 1/3.

To build these metrics you only need one condition: the metric for which you want to return the root value.

When putting these functions in your table, it will return the root value for each row individually. On its own, these functions have limited use cases, but they can be used in conjunction with the various statistical functions for data modelling or to provide insights and predictions.

Roots

Day ↑ Page: 1 / 3 › Rows: 5	Online Orders		Square Root (Online Orders)		Cube Root (Online Orders)	
	18,248 out of 18,248		135.09 out of 135.09		26.33 out of 26.33	
1. Apr 6, 2024	1,399	7.7%	37.40	27.7%	11.18	42.5%
2. Apr 7, 2024	2,298	12.6%	47.94	35.5%	13.20	50.1%
3. Apr 8, 2024	995	5.5%	31.54	23.4%	9.98	37.9%
4. Apr 9, 2024	1,028	5.6%	32.06	23.7%	10.09	38.3%
5. Apr 10, 2024	1,025	5.6%	32.02	23.7%	10.08	38.3%

Statistical Analyses and Regressions.

▪ I want to calculate standard deviation/variance of a metric....

There are functions that calculate **VARIANCE** or **STANDARD DEVIATION** automatically. If you're into statistics (like myself), Adobe Analytics does provide the formulas they use in their official documentation here. But for using it in Analysis Workspace, you only need to know what it means, not the statistics behind it. The **STANDARD DEVIATION** refers to how much values differ from the average in a distribution, and the **VARIANCE** is the **STANDARD DEVIATION** squared. These metrics are typically used when you're conducting a statistical test to help determine which group is performing better.

To build either of these, the only required condition is your metric of interest. It does give you an option to include rows with a value of 0. Whether or not you want to include them is based on your specific use case.

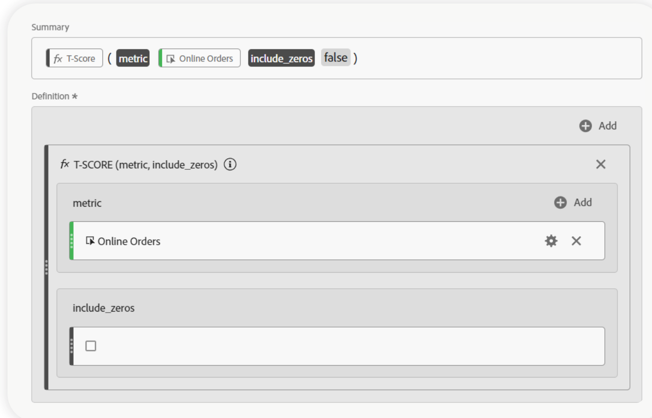
When you put the metric in your table, it will do its calculation based on the row values of the dimension. Like many of the other metrics, these calculate their values based on the entirety of a dimension, regardless of how many values are shown in the table (unless a segment is applied to filter the data).

Variance & Standard Deviation

	Visits		Variance (Visits)		Standard Deviation (Visits)	
Mobile Device Type Page: 1 / 1 Rows: 10	Apr 6	253,588 out of 253,588	Apr 6	4,697,248,720 out of 4,697,248,720	Apr 6	68,536 out of 68,536
1. Other	203,559	80.3%	4,697,248,720	100.0%	68,536	100.0%
2. Gaming Console	16,641	6.6%	4,697,248,720	100.0%	68,536	100.0%
3. Television	15,905	6.3%	4,697,248,720	100.0%	68,536	100.0%
4. Mobile Phone	6,495	2.6%	4,697,248,720	100.0%	68,536	100.0%
5. Set-top Box	6,068	2.4%	4,697,248,720	100.0%	68,536	100.0%
6. Tablet	4,715	1.9%	4,697,248,720	100.0%	68,536	100.0%
7. EReader	205	0.1%	4,697,248,720	100.0%	68,536	100.0%

- **I want to calculate a T-Score/Z-Score…**

To calculate the actual statistical test there are functions for both **T-SCORE** and **Z-SCORE**. In a real-world setting, a **T-SCORE** would be used with samples and a **Z-SCORE** would be used with entire populations. This results in them being calculated slightly differently due to the level of confidence in the data. However, in Analysis Workspace, both of these are calculated the same way, so they can be used interchangeably. The score returned is associated with how many standard deviations away from the mean the specified value is. These functions can be combined with others to help you perform a statistical test to find differences between groups.



To build it, you only need your metric of interest.

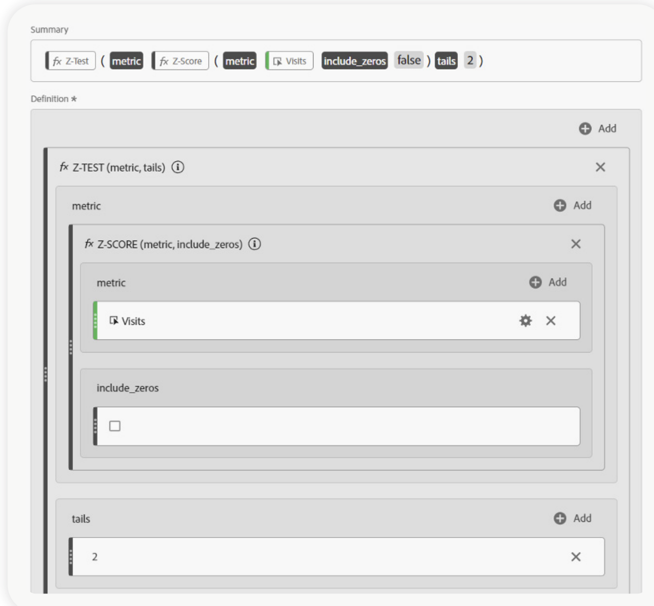
Using it in your table, it will do the calculation based on the rows of data, similar to the **VARIANCE** or **STANDARD DEVIATION**. Here, the resulting value is the **T-SCORE** (or **Z-SCORE**), which is a normalised value that would typically be looked up on a t-table and, in conjunction with the degrees of freedom, determine whether or not a result is statistically significant.

- T-Score & Z-Score

	Visits		T-Score (Visits)		Z-Score (Visits)	
Mobile Device Type Page: 1 / 1 Rows: 10 1-7 of 7	253,588 out of 253,588 (Apr 6)		3.17 out of 3.17 (Apr 6 – Apr 20)		3.17 out of 3.17 (Apr 6 – Apr 20)	
1. Other	203,559	80.3%	2.44	77.0%	2.44	77.0%
2. Gaming Console	16,641	6.6%	-0.29	-9.0%	-0.29	-9.0%
3. Television	15,905	6.3%	-0.30	-9.3%	-0.30	-9.3%
4. Mobile Phone	6,495	2.6%	-0.43	-13.7%	-0.43	-13.7%
5. Set-top Box	6,068	2.4%	-0.44	-13.9%	-0.44	-13.9%
6. Tablet	4,715	1.9%	-0.46	-14.5%	-0.46	-14.5%
7. EReader	205	0.1%	-0.53	-16.6%	-0.53	-16.6%

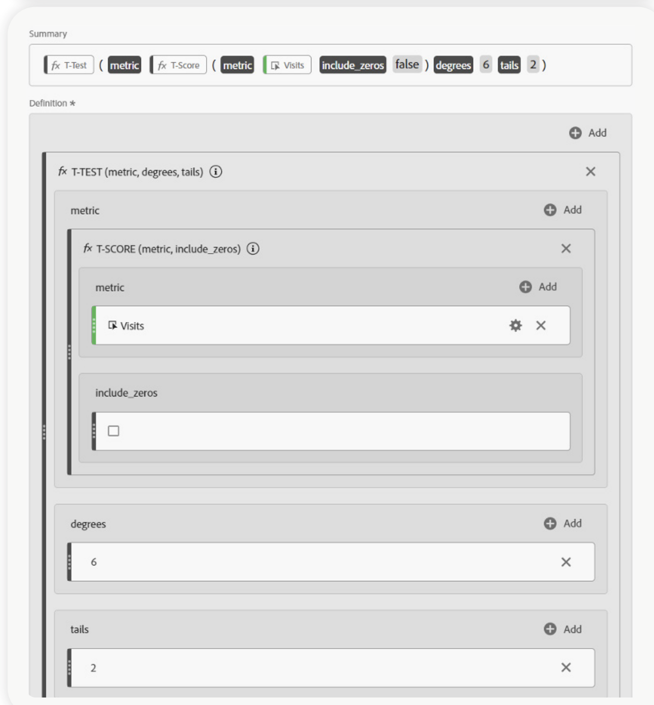
▪ I want to conduct a statistical test to determine if a value is significant…

There are often times when doing analyses that we want to know whether something is statistically significant, especially when doing an A/B test. While the 'Analytics for Target' (or A4T) panel does provide statistical calculations, it only does so for experiences run through Adobe Target. Even if you have a Target test, the A4T panel cannot be used with calculated metrics, so it can be fairly limited in its use. Fortunately, there are other functions that can perform the various tasks associated with calculated metrics. The **T-TEST/Z-TEST** build upon the other statistical functions, such as the **VARIANCE** or **STANDARD DEVIATION** and **T-SCORE** or **Z-SCORE**. In a real-world scenario, there are various factors that would dictate whether you use a **T-TEST** or a **Z-TEST**, including whether or not the population variance is known and the size of your sample.



Calculating the **T-SCORE** (or **Z-SCORE**) is a prerequisite for being able to determine if a test is significant or not. The **T-TEST** (or **Z-TEST**) returns the probability of seeing the test statistic (**T-SCORE/Z-SCORE**). The lower the result, the less likely the occurrence; the higher the result, the more likely the occurrence.

Unlike the **T-SCORE/Z-SCORE**, there are differences between the **T-TEST** and **Z-TEST**.



To build the **T-TEST**, it asks for three conditions: a metric, degrees of freedom, and tails. The metric you use will need to include the **T-SCORE** function nested within it. Within that should be your metric of interest. Degrees of freedom in this case is the number of rows in your table minus one. You also need to specify if your test is one-tailed or two-tailed. If you have a directional hypothesis use one-tailed, if your hypothesis could be greater or lower use two-tailed.

To build the **Z-TEST**, there are two required conditions: a metric and the tails. Because a **Z-TEST** is used when the population variance is known, you don't need to use degrees of freedom in the calculation. Your metric is again going to contain a nested function, the **Z-SCORE**, with your metric of interest.

When you put these in your table, it returns the probability of seeing your result. A lower value means that it is less likely to have occurred, and less likely to be a statistically significant difference. In your table it will calculate this probability for each row based on the set of data.

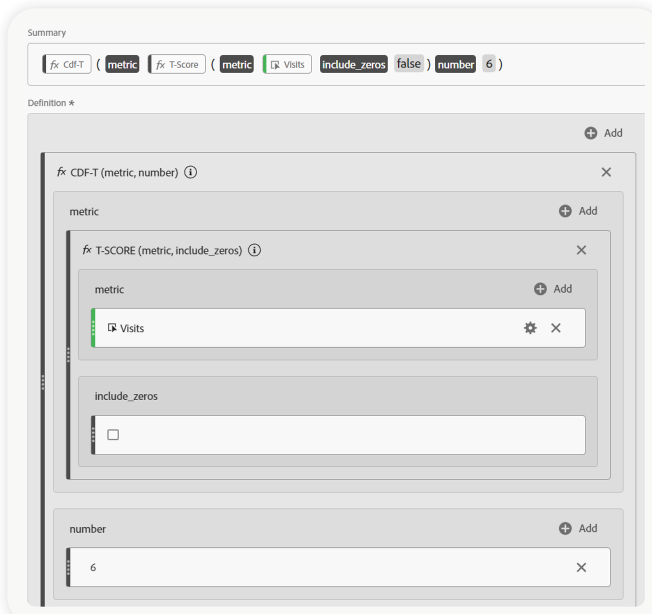
T-Test & Z-Test

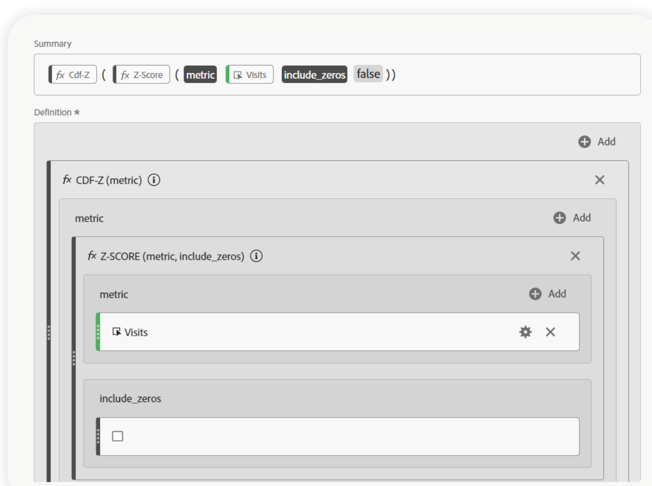
Mobile Device Type Page: 1 / 1 Rows: 10 1-7 of 7	Visits Apr 6	253,588 out of 253,588	T-Test (Visits) Apr 6	0.01928 out of 0.01928	Z-Test (Visits) Apr 6	0.00152 out of 0.00152
1. Other	203,559	80.3%	0.05037	261.2%	0.01463	964.3%
2. Gaming Console	16,641	6.6%	0.78466	1,000.0+%	0.77505	1,000.0+%
3. Television	15,905	6.3%	0.77684	1,000.0+%	0.76684	1,000.0+%
4. Mobile Phone	6,495	2.6%	0.67959	1,000.0+%	0.66443	1,000.0+%
5. Set-top Box	6,068	2.4%	0.67532	1,000.0+%	0.65991	1,000.0+%
6. Tablet	4,715	1.9%	0.66186	1,000.0+%	0.64567	1,000.0+%
7. EReader	205	0.1%	0.61802	1,000.0+%	0.59918	1,000.0+%

▪ I want to calculate the cumulative distribution function…

The cumulative distribution function, or **CDF-T/CDF-Z**, returns the percentage of values in the t-distribution or normal distribution, respectively, that have a score less than the specified value. Because they rely on different distributions, there is a slight difference in how they're calculated.

For the **CDF-T** function, you need to nest a **T-SCORE** for your metric of interest, and you need the number, which is your degrees of freedom (number of rows minus 1). For the **CDF-Z**, you only need to nest your **Z-SCORE** with the metric of interest, as the z distribution doesn't require degrees of freedom due to knowing the population variance.





When you put the metrics in your table, it will return the percent of values in the distribution that are less than or equal to the specified value. This can help you determine if a specific value is statistically significant.

Cdf-T & Cdf-Z

Mobile Device Type Page: 1 / 1 Rows: 10 1-7 of 7	Visits Apr 6	253,588 out of 253,588	CDF-T (Visits) Apr 6 Apr 20	0.99 out of 0.99	CDF-Z (Visits) Apr 6 Apr 20	1.00 out of 1.00
1. Other	203,559	80.3%	0.97	98.4%	0.99	99.3%
2. Gaming Console	16,641	6.6%	0.39	39.6%	0.39	38.8%
3. Television	15,905	6.3%	0.39	39.2%	0.38	38.4%
4. Mobile Phone	6,495	2.6%	0.34	34.3%	0.33	33.2%
5. Set-top Box	6,068	2.4%	0.34	34.1%	0.33	33.0%
6. Tablet	4,715	1.9%	0.33	33.4%	0.32	32.3%
7. EReader	205	0.1%	0.31	31.2%	0.30	30.0%

Statistical Test

Day Page: 1 / 1 Rows: 25	Online Orders Apr 6	18,248 out of 18,248	T-Score (Orders) Apr 6	39.18 out of 39.18	T-Test (Orders) Apr 6	0.00 out of 0.00	CDF-T (Orders) Apr 6	1.00 out of 1.00
1. Apr 6, 2024	1,399	7.7%	0.42	1.1%	0.68	1,000.0+%	0.66	65.9%
2. Apr 7, 2024	2,298	12.6%	2.49	6.3%	0.03	1,000.0+%	0.99	98.7%
3. Apr 8, 2024	995	5.5%	-0.51	-1.3%	0.62	1,000.0+%	0.31	30.9%
4. Apr 9, 2024	1,028	5.6%	-0.43	-1.1%	0.67	1,000.0+%	0.34	33.6%
5. Apr 10, 2024	1,025	5.6%	-0.44	-1.1%	0.67	1,000.0+%	0.33	33.3%
6. Apr 11, 2024	946	5.2%	-0.62	-1.6%	0.54	1,000.0+%	0.27	27.2%
7. Apr 12, 2024	1,016	5.6%	-0.46	-1.2%	0.65	1,000.0+%	0.33	32.6%
8. Apr 13, 2024	981	5.4%	-0.54	-1.4%	0.60	1,000.0+%	0.30	29.8%
9. Apr 14, 2024	999	5.5%	-0.50	-1.3%	0.62	1,000.0+%	0.31	31.2%
10. Apr 15, 2024	949	5.2%	-0.62	-1.6%	0.55	1,000.0+%	0.27	27.4%
11. Apr 16, 2024	969	5.3%	-0.57	-1.5%	0.58	1,000.0+%	0.29	28.9%
12. Apr 17, 2024	1,435	7.9%	0.50	1.3%	0.62	1,000.0+%	0.69	68.8%
13. Apr 18, 2024	2,223	12.2%	2.32	5.9%	0.04	1,000.0+%	0.98	98.2%
14. Apr 19, 2024	997	5.5%	-0.51	-1.3%	0.62	1,000.0+%	0.31	31.1%
15. Apr 20, 2024	988	5.4%	-0.53	-1.3%	0.61	1,000.0+%	0.30	30.4%

▪ I want to predict a dimension's values…

LINEAR REGRESSIONS are an incredibly powerful tool when it comes to predicting values. Regressions can be used for a variety of reasons. The most common is to see how two variables are related. For example, if you want to see if increases in traffic to your site are associated with an increase in orders, you can use a regression. This can help you predict future behaviours, or even help fill in gaps in your data. Once you have the regression formula, you can apply it to your metric to predict the value of another metric, such as predicting orders based on traffic amounts. One thing to keep in mind with regressions is that correlation does not imply causation. Just because two numbers go up together, it doesn't mean that one is causing the other. It's possible that there's a third variable somewhere that is impacting both numbers. Keep this in mind when interpreting your results and drawing conclusions.

Analysis Workspace provides options for six different types of regressions. The type of regression that you use will depend on how your data looks. It's possible that you might start with a standard linear regression but find that it doesn't fit your data. You may need to test a few different types of regression formulas until you find the one that fits your data the best.

Within each type of regression there are four functions: **CORRELATION COEFFICIENT**, **INTERCEPT**, **PREDICTED Y**, and **SLOPE**. Each of these will return a different part of the regression equation, Y = aX + b.

- **PREDICTED Y** = Y
- **SLOPE** = a
- **INTERCEPT** = b
- **CORRELATION COEFFICIENT** = Strength of the relationship between X and Y

The **PREDICTED Y** is the final result of the regression formula. In your table, for the given value of the X metric on a specific row, it will return what the predicted value is for the Y metric. This can be useful when you are missing data in a metric, and you want to estimate what it should be. The results of a regression are generally accurate, but there will be some differences between the predicted values and what the true value is due to natural variance.

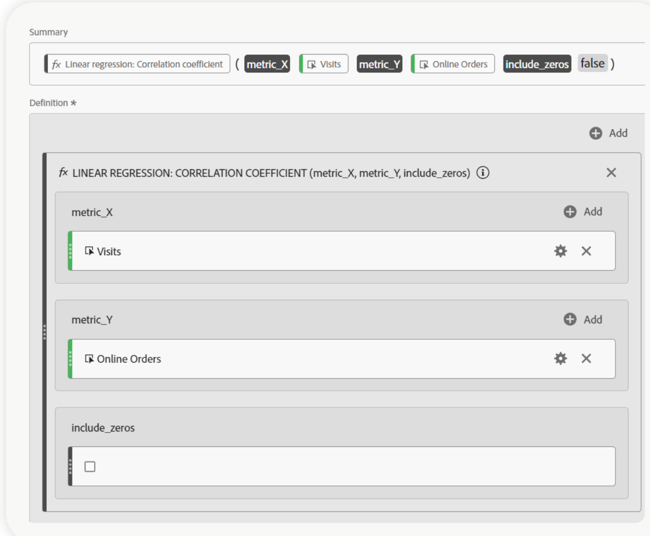
The **SLOPE** (the "a" in the above formula) is the actual correlation between the two variables and is used in the calculation to predict Y based on the value of X.

The **INTERCEPT** (the "b" in the above formula) is used to raise or lower the predicted values. If the metric X is 0, the predicted Y value would be equal to the intercept. Along with the slope, it is used to help predict the Y values.

The **CORRELATION COEFFICIENT** returns a value that indicates how strongly two metrics are associated with each other. It will return a value between –1 and +1. The further from zero the number is, the stronger the two metrics are related. If the result is positive that means when one metric increases, so does the other. If the result is negative that means when one metric increases the other decreases.

The metric builder is the same for all four functions. In each you need two metrics, an X and a Y. The metric that is identified as the Y will be predicted based on the values of the first metric, X. You also have the option to include rows where there are zeros, or to exclude them.

When placed in your table, the **CORRELATION COEFFICIENT**, **SLOPE**, and **INTERCEPT** will be the same down all of the rows because these are constants in the formula. The **PREDICTED Y** will differ on each row as it is the result of the calculated formula. In your table, based on the value of the X metric and the constants in the formula, it will return an estimated value for the Y metric.



Linear Regression

Day (Page: 1 / 1 Rows: 25)	Visits	Online Orders	Linear Regression Correlation Coefficient (x=Visits, y=Online Orders)	Linear Regression Predicted Y (x=Visits, y=Online Orders)	Linear Regression Slope (x=Visits, y=Online Orders)	Linear Regression Intercept (x=Visits, y=Online Orders)
	253,588 out of 253,588	18,248 out of 18,248	0.92 out of 0.92	21,681.08 out of 21,681.08	0.09 out of 0.09	-251.17 out of -251.17
1. Apr 6, 2024	18,963 7.5%	1,399 7.7%	0.92 100.0%	1,388.90 6.4%	0.09 100.0%	-251.17 100.0%
2. Apr 7, 2024	26,692 10.5%	2,298 12.6%	0.92 100.0%	2,057.36 9.5%	0.09 100.0%	-251.17 100.0%
3. Apr 8, 2024	15,573 6.1%	995 5.5%	0.92 100.0%	1,095.70 5.1%	0.09 100.0%	-251.17 100.0%
4. Apr 9, 2024	19,335 7.6%	1,028 5.6%	0.92 100.0%	1,421.07 6.6%	0.09 100.0%	-251.17 100.0%
5. Apr 10, 2024	13,023 5.1%	1,025 5.6%	0.92 100.0%	875.16 4.0%	0.09 100.0%	-251.17 100.0%
6. Apr 11, 2024	13,272 5.2%	946 5.2%	0.92 100.0%	896.70 4.1%	0.09 100.0%	-251.17 100.0%
7. Apr 12, 2024	13,205 5.2%	1,016 5.6%	0.92 100.0%	890.90 4.1%	0.09 100.0%	-251.17 100.0%
8. Apr 13, 2024	12,923 5.1%	981 5.4%	0.92 100.0%	866.51 4.0%	0.09 100.0%	-251.17 100.0%
9. Apr 14, 2024	13,173 5.2%	999 5.5%	0.92 100.0%	888.13 4.1%	0.09 100.0%	-251.17 100.0%
10. Apr 15, 2024	15,145 6.0%	949 5.2%	0.92 100.0%	1,058.69 4.9%	0.09 100.0%	-251.17 100.0%
11. Apr 16, 2024	18,238 7.2%	969 5.3%	0.92 100.0%	1,326.19 6.1%	0.09 100.0%	-251.17 100.0%
12. Apr 17, 2024	19,190 7.6%	1,435 7.9%	0.92 100.0%	1,408.53 6.5%	0.09 100.0%	-251.17 100.0%
13. Apr 18, 2024	27,430 10.8%	2,223 12.2%	0.92 100.0%	2,121.19 9.8%	0.09 100.0%	-251.17 100.0%
14. Apr 19, 2024	13,083 5.2%	997 5.5%	0.92 100.0%	880.35 4.1%	0.09 100.0%	-251.17 100.0%
15. Apr 20, 2024	15,306 6.0%	988 5.4%	0.92 100.0%	1,072.61 4.9%	0.09 100.0%	-251.17 100.0%

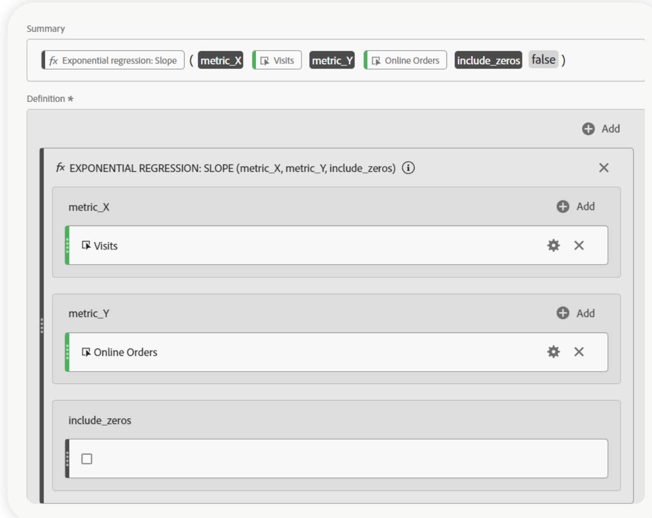
▪ I need a regression that has an increasing change over time….

Before reading this section about **EXPONENTIAL REGRESSIONS**, please read the above section about **LINEAR REGRESSIONS** to understand the use of the **SLOPE**, **INTERCEPT**, **PREDICTED Y**, and **CORRELATION COEFFICIENT**.

Analysis Workspace offers six different options for completing regressions. The **EXPONENTIAL REGRESSION** method is used when there is an increase or decrease in the change over time – essentially, when the growth or decay of a value starts slowly and then increases as time goes on. In order to determine if this is the right model to fit your data it is best to plot your X metric to determine its trend over time.

To build any of the four **EXPONENTIAL REGRESSION** functions, the process is the same as for **LINEAR REGRESSIONS**. You need to include your X metric and Y metric. The values of the X metric will be used to predict the values in your Y metric based on an exponential growth model.

When placed in your table, the **CORRELATION COEFFICIENT**, **SLOPE**, and **INTERCEPT** will be the same down all of the rows because these are constants in the formula. The **PREDICTED Y** will differ on each row as it is the result of the calculated formula. In your table, based on the value of the X metric and the constants in the formula, it will return an estimated value for the Y metric.



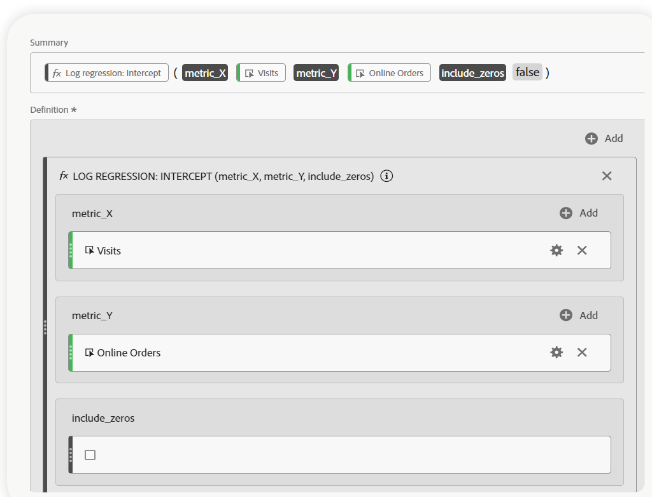
Exponential Regression

Day ↑ Page: 1 / 1 Rows: 25	Visits	Online Orders	Exponential Regression Correlation Coefficient (x=Visits, y=Online Orders)	Exponential Regression Predicted Y (x=Visits, y=Online Orders)	Exponential Regression Slope (x=Visits, y=Online Orders)	Exponential Regression Intercept (x=Visits, y=Online Orders)
	253,588 out of 253,588	18,248 out of 18,248	0.91 out of 0.91	907,833,375.67 out of 907,833,375.67	0.00 out of 0.00	437.99 out of 437.99
1. Apr 6, 2024	18,963 7.5%	1,399 7.7%	0.91 100.0%	1,299.61 0.0%	0.00 100.0%	437.99 100.0%
2. Apr 7, 2024	26,692 10.5%	2,298 12.6%	0.91 100.0%	2,024.56 0.0%	0.00 100.0%	437.99 100.0%
3. Apr 8, 2024	15,573 6.1%	995 5.5%	0.91 100.0%	1,069.97 0.0%	0.00 100.0%	437.99 100.0%
4. Apr 9, 2024	19,335 7.6%	1,028 5.6%	0.91 100.0%	1,327.63 0.0%	0.00 100.0%	437.99 100.0%
5. Apr 10, 2024	13,023 5.1%	1,025 5.6%	0.91 100.0%	924.39 0.0%	0.00 100.0%	437.99 100.0%
6. Apr 11, 2024	13,272 5.2%	946 5.2%	0.91 100.0%	937.68 0.0%	0.00 100.0%	437.99 100.0%
7. Apr 12, 2024	13,205 5.2%	1,016 5.6%	0.91 100.0%	934.09 0.0%	0.00 100.0%	437.99 100.0%
8. Apr 13, 2024	12,923 5.1%	981 5.4%	0.91 100.0%	919.10 0.0%	0.00 100.0%	437.99 100.0%
9. Apr 14, 2024	13,173 5.2%	999 5.5%	0.91 100.0%	932.38 0.0%	0.00 100.0%	437.99 100.0%
10. Apr 15, 2024	15,145 6.0%	949 5.2%	0.91 100.0%	1,044.02 0.0%	0.00 100.0%	437.99 100.0%
11. Apr 16, 2024	18,238 7.2%	969 5.3%	0.91 100.0%	1,246.67 0.0%	0.00 100.0%	437.99 100.0%
12. Apr 17, 2024	19,190 7.6%	1,435 7.9%	0.91 100.0%	1,316.64 0.0%	0.00 100.0%	437.99 100.0%
13. Apr 18, 2024	27,430 10.8%	2,223 12.2%	0.91 100.0%	2,112.10 0.0%	0.00 100.0%	437.99 100.0%
14. Apr 19, 2024	13,083 5.2%	997 5.5%	0.91 100.0%	927.58 0.0%	0.00 100.0%	437.99 100.0%
15. Apr 20, 2024	15,306 6.0%	988 5.4%	0.91 100.0%	1,053.71 0.0%	0.00 100.0%	437.99 100.0%

▪ I need a regression for binary data….

Before reading this section about **LOG REGRESSIONS**, please read the above section about **LINEAR REGRESSIONS** to understand the use of the **SLOPE**, **INTERCEPT**, **PREDICTED Y**, and **CORRELATION COEFFICIENT**.

Adobe offers six different options for completing regressions. The **LOG REGRESSION** is typically used when the outcome being predicted is binary in nature (for example, true or false, or yes or no).



These regressions can also be applied to ordinal data of more than two categories. **LOG REGRESSIONS** are often used to help determine how new data fits into an existing category.

To build any of the four **LOG REGRESSION** functions, the process is the same as for **LINEAR REGRESSIONS**. You need to include your X metric and Y metric. The values of the X metric will be used to predict the values in your Y metric based on a logarithmic model.

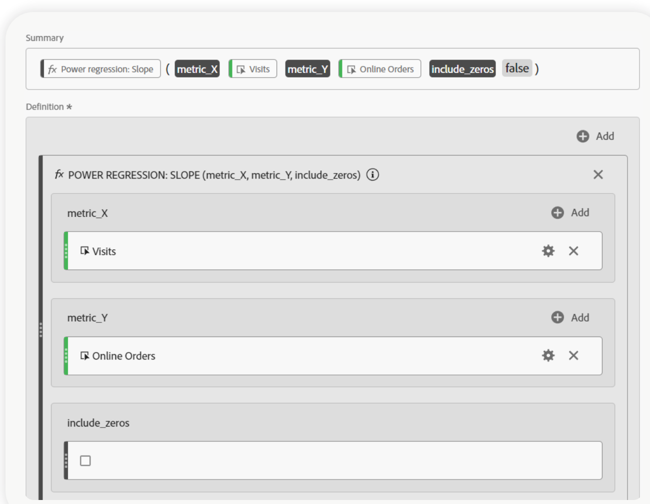
When placed in your table, the **CORRELATION COEFFICIENT**, **SLOPE**, and **INTERCEPT** will be the same down all of the rows because these are constants in the formula. The **PREDICTED Y** will differ on each row as it is the result of the calculated formula. In your table, based on the value of the X metric and the constants in the formula, it will return an estimated value for the Y metric.

Log Regression

	Visits		Desktop Greater than Mobile (AOV)		Online Orders		Log Regression Correlation Coefficient (x=Visits, y=Online Orders)		Log Regression Predicted Y (x=Visits, y=Online Orders)		Log Regression Slope (x=Visits, y=Online Orders)		Log Regression Intercept (x=Visits, y=Online Orders)	
Day ↑ Page: 1 / 1 Rows: 25	253,588 out of 253,588		1 out of 1		18,248 out of 18,248		0.87 out of 0.87		5,454.96 out of 5,454.96		1,548.97 out of 1,548.97		-13,819.54 out of -13,819.54	
1. Apr 6, 2024	18,963	7.5%	0	0.0%	1,399	7.7%	0.87	100.0%	1,438.15	26.4%	1,548.97	100.0%	-13,819.54	100.0%
2. Apr 7, 2024	26,692	10.5%	1	100.0%	2,298	12.6%	0.87	100.0%	1,967.70	36.1%	1,548.97	100.0%	-13,819.54	100.0%
3. Apr 8, 2024	15,573	6.1%	1	100.0%	995	5.5%	0.87	100.0%	1,133.08	20.8%	1,548.97	100.0%	-13,819.54	100.0%
4. Apr 9, 2024	19,335	7.6%	1	100.0%	1,028	5.6%	0.87	100.0%	1,468.24	26.9%	1,548.97	100.0%	-13,819.54	100.0%
5. Apr 10, 2024	13,023	5.1%	0	0.0%	1,025	5.6%	0.87	100.0%	856.09	15.7%	1,548.97	100.0%	-13,819.54	100.0%
6. Apr 11, 2024	13,272	5.2%	1	100.0%	946	5.2%	0.87	100.0%	885.43	16.2%	1,548.97	100.0%	-13,819.54	100.0%
7. Apr 12, 2024	13,205	5.2%	1	100.0%	1,016	5.6%	0.87	100.0%	877.59	16.1%	1,548.97	100.0%	-13,819.54	100.0%
8. Apr 13, 2024	12,923	5.1%	1	100.0%	981	5.4%	0.87	100.0%	844.15	15.5%	1,548.97	100.0%	-13,819.54	100.0%
9. Apr 14, 2024	13,173	5.2%	1	100.0%	999	5.5%	0.87	100.0%	873.83	16.0%	1,548.97	100.0%	-13,819.54	100.0%
10. Apr 15, 2024	15,145	6.0%	1	100.0%	949	5.2%	0.87	100.0%	1,089.91	20.0%	1,548.97	100.0%	-13,819.54	100.0%
11. Apr 16, 2024	18,238	7.2%	0	0.0%	969	5.3%	0.87	100.0%	1,377.77	25.3%	1,548.97	100.0%	-13,819.54	100.0%
12. Apr 17, 2024	19,190	7.6%	0	0.0%	1,435	7.9%	0.87	100.0%	1,456.58	26.7%	1,548.97	100.0%	-13,819.54	100.0%
13. Apr 18, 2024	27,430	10.8%	0	0.0%	2,223	12.2%	0.87	100.0%	2,009.95	36.8%	1,548.97	100.0%	-13,819.54	100.0%
14. Apr 19, 2024	13,083	5.2%	0	0.0%	997	5.5%	0.87	100.0%	863.21	15.8%	1,548.97	100.0%	-13,819.54	100.0%
15. Apr 20, 2024	15,306	6.0%	0	0.0%	988	5.4%	0.87	100.0%	1,106.29	20.3%	1,548.97	100.0%	-13,819.54	100.0%

▪ I need a regression where the rate of change relies on a variables exponent....

Before reading this section about **POWER REGRESSIONS**, please read the above section about **LINEAR REGRESSIONS** to understand the use of the **SLOPE**, **INTERCEPT**, **PREDICTED Y**, and **CORRELATION COEFFICIENT**.



Adobe offers six different options for completing regressions. The **POWER REGRESSION** method is used when your Y metric is proportional to your X metric when raised to a power exponent. This is one of the options to use when your data is non-linear, such as demonstrating an inverse or exponential relationship. In order to determine if this is the right model to fit your data, it is best to plot your metrics to determine their trend over time.

To build any of the four **POWER REGRESSION** functions, the process is the same as for **LINEAR REGRESSIONS**. You need to include your X metric and Y metric. The values of the X metric will be used to predict the values in your Y metric based on a power model.

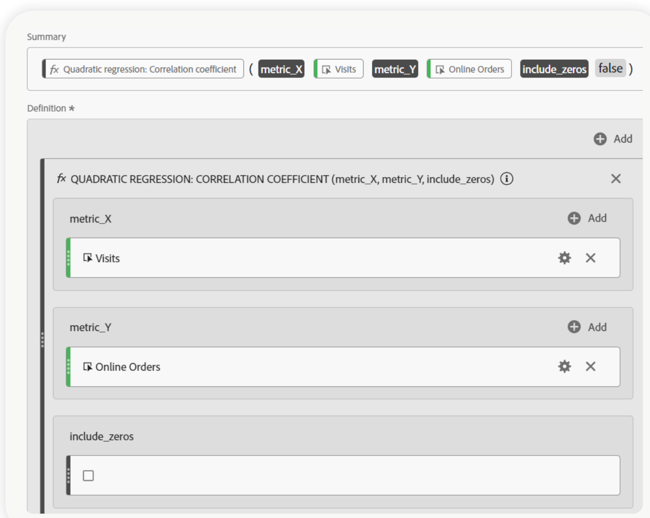
When placed in your table, the **CORRELATION COEFFICIENT**, **SLOPE**, and **INTERCEPT** will be the same down all of the rows because these are constants in the formula. The **PREDICTED Y** will differ on each row as it is the result of the calculated formula. In your table, based on the value of the X metric and the constants in the formula, it will return an estimated value for the Y metric.

Power Regression

Day ↑ Page: 1 / 1 Rows: 25	Visits 253,588 out of 253,588	Online Orders 18,248 out of 18,248	Power Regression Correlation Coefficient (x=Visits, y=Online Orders) 0.88 out of 0.88	Power Regression Predicted Y (x=Visits, y=Online Orders) 19,657.08 out of 19,657.08	Power Regression Slope (x=Visits, y=Online Orders) 1.03 out of 1.03	Power Regression Intercept (x=Visits, y=Online Orders) 0.05 out of 0.05
1. Apr 6, 2024	18,963 7.5%	1,399 7.7%	0.88 100.0%	1,344.16 6.8%	1.03 100.0%	0.05 100.0%
2. Apr 7, 2024	26,692 10.5%	2,298 12.6%	0.88 100.0%	1,914.46 9.7%	1.03 100.0%	0.05 100.0%
3. Apr 8, 2024	15,573 6.1%	995 5.5%	0.88 100.0%	1,096.39 5.6%	1.03 100.0%	0.05 100.0%
4. Apr 9, 2024	19,335 7.6%	1,028 5.6%	0.88 100.0%	1,371.45 7.0%	1.03 100.0%	0.05 100.0%
5. Apr 10, 2024	13,023 5.1%	1,025 5.6%	0.88 100.0%	911.22 4.6%	1.03 100.0%	0.05 100.0%
6. Apr 11, 2024	13,272 5.2%	946 5.2%	0.88 100.0%	929.25 4.7%	1.03 100.0%	0.05 100.0%
7. Apr 12, 2024	13,205 5.2%	1,016 5.6%	0.88 100.0%	924.40 4.7%	1.03 100.0%	0.05 100.0%
8. Apr 13, 2024	12,923 5.1%	981 5.4%	0.88 100.0%	903.99 4.6%	1.03 100.0%	0.05 100.0%
9. Apr 14, 2024	13,173 5.2%	999 5.5%	0.88 100.0%	922.08 4.7%	1.03 100.0%	0.05 100.0%
10. Apr 15, 2024	15,145 6.0%	949 5.2%	0.88 100.0%	1,065.23 5.4%	1.03 100.0%	0.05 100.0%
11. Apr 16, 2024	18,238 7.2%	969 5.3%	0.88 100.0%	1,291.03 6.6%	1.03 100.0%	0.05 100.0%
12. Apr 17, 2024	19,190 7.6%	1,435 7.9%	0.88 100.0%	1,360.81 6.9%	1.03 100.0%	0.05 100.0%
13. Apr 18, 2024	27,430 10.8%	2,223 12.2%	0.88 100.0%	1,969.24 10.0%	1.03 100.0%	0.05 100.0%
14. Apr 19, 2024	13,083 5.2%	997 5.5%	0.88 100.0%	915.57 4.7%	1.03 100.0%	0.05 100.0%
15. Apr 20, 2024	15,306 6.0%	988 5.4%	0.88 100.0%	1,076.95 5.5%	1.03 100.0%	0.05 100.0%

I need a regression for parabola-shaped data....

Before reading this section about **QUADRATIC REGRESSIONS**, please read the above section about **LINEAR REGRESSIONS** to understand the use of the **SLOPE**, **INTERCEPT**, **PREDICTED Y**, and **CORRELATION COEFFICIENT**.



Adobe offers six different options for completing regressions, the **QUADRATIC REGRESSION** method is used when your data has a U-shape like a parabola. Even if the data is a partial U-shape or an inverted U-shape, the **QUADRATIC REGRESSION** is still one of the best tools to use. It does require you to have at least three data points due to the non-linear nature (whereas linear data requires at least two points), although the more data points you have, the greater the accuracy of your model. In order to determine if this is the right model to fit your data, it is best to plot your X metric to determine its shape.

To build any of the four **QUADRATIC REGRESSION** functions, the process is the same as for **LINEAR REGRESSIONS**. You need to include your X metric and Y metric. The values of the X metric will be used to predict the values in your Y metric based on a quadratic model.

When placed in your table, the **CORRELATION COEFFICIENT**, **SLOPE**, and **INTERCEPT** will be the same down all of the rows because these are constants in the formula. The **PREDICTED Y** will differ on each row as it is the result of the calculated formula. In your table, based on the value of the X metric and the constants in the formula, it will return an estimated value for the Y metric.

Quadratic Regression

Day ↑ Page: 1 / 1 Rows: 25	Visits 253,588 out of 253,588		Online Orders 18,248 out of 18,248		Quadratic Regression Correlation Coefficient (x=Visits, y=Online Orders) 0.91 out of 0.91		Quadratic Regression Predicted Y (x=Visits, y=Online Orders) 87,633.93 out of 87,633.93		Quadratic Regression Slope (x=Visits, y=Online Orders) 0.00 out of 0.00		Quadratic Regression Intercept (x=Visits, y=Online Orders) 15.67 out of 15.67	
1. Apr 6, 2024	18,963	7.5%	1,399	7.7%	0.91	100.0%	1,342.22	1.5%	0.00	100.0%	15.67	100.0%
2. Apr 7, 2024	26,692	10.5%	2,298	12.6%	0.91	100.0%	2,041.34	2.3%	0.00	100.0%	15.67	100.0%
3. Apr 8, 2024	15,573	6.1%	995	5.5%	0.91	100.0%	1,081.65	1.2%	0.00	100.0%	15.67	100.0%
4. Apr 9, 2024	19,335	7.6%	1,028	5.6%	0.91	100.0%	1,372.52	1.6%	0.00	100.0%	15.67	100.0%
5. Apr 10, 2024	13,023	5.1%	1,025	5.6%	0.91	100.0%	904.16	1.0%	0.00	100.0%	15.67	100.0%
6. Apr 11, 2024	13,272	5.2%	946	5.2%	0.91	100.0%	920.79	1.1%	0.00	100.0%	15.67	100.0%
7. Apr 12, 2024	13,205	5.2%	1,016	5.6%	0.91	100.0%	916.30	1.0%	0.00	100.0%	15.67	100.0%
8. Apr 13, 2024	12,923	5.1%	981	5.4%	0.91	100.0%	897.52	1.0%	0.00	100.0%	15.67	100.0%
9. Apr 14, 2024	13,173	5.2%	999	5.5%	0.91	100.0%	914.16	1.0%	0.00	100.0%	15.67	100.0%
10. Apr 15, 2024	15,145	6.0%	949	5.2%	0.91	100.0%	1,050.75	1.2%	0.00	100.0%	15.67	100.0%
11. Apr 16, 2024	18,238	7.2%	969	5.3%	0.91	100.0%	1,284.13	1.5%	0.00	100.0%	15.67	100.0%
12. Apr 17, 2024	19,190	7.6%	1,435	7.9%	0.91	100.0%	1,360.67	1.6%	0.00	100.0%	15.67	100.0%
13. Apr 18, 2024	27,430	10.8%	2,223	12.2%	0.91	100.0%	2,115.74	2.4%	0.00	100.0%	15.67	100.0%
14. Apr 19, 2024	13,083	5.2%	997	5.5%	0.91	100.0%	908.15	1.0%	0.00	100.0%	15.67	100.0%
15. Apr 20, 2024	15,306	6.0%	988	5.4%	0.91	100.0%	1,062.32	1.2%	0.00	100.0%	15.67	100.0%

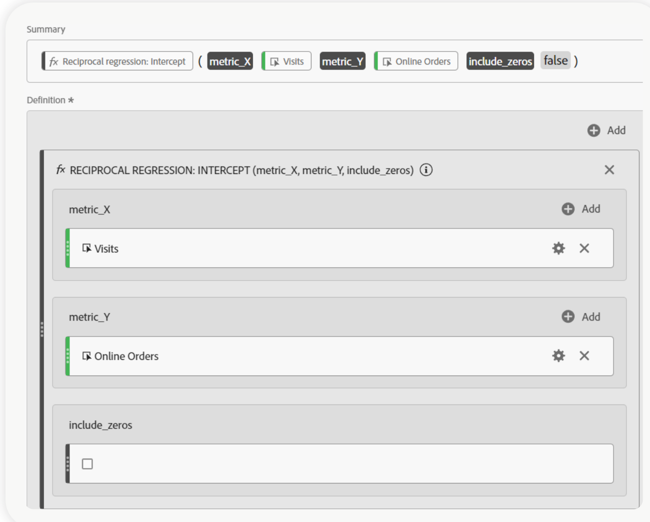
- **I need a regression for variables that change in opposite directions….**

Before reading this section about **RECIPROCAL REGRESSIONS**, please read the above section about **LINEAR REGRESSIONS** to understand the use of the **SLOPE**, **INTERCEPT**, **PREDICTED Y**, and **CORRELATION COEFFICIENT**.

Adobe offers six different options for completing regressions. The **RECIPROCAL REGRESSION** method is for when your Y metric is a reciprocal of your X metric, meaning one increases as the other decreases. As your X metric increases, the difference between it and your Y metric can decrease as it may have an asymptote (a limit it can't pass). The **RECIPROCAL REGRESSION** takes this decreasing difference into account in its calculations.

To build any of the four **RECIPROCAL REGRESSION** functions, the process is the same as for **LINEAR REGRESSIONS**. You need to include your X metric and Y metric. The values of the X metric will be used to predict the values in your Y metric based on a reciprocal model.

When placed in your table, the **CORRELATION COEFFICIENT**, **SLOPE**, and **INTERCEPT** will be the same down all of the rows because these are constants in the formula. The **PREDICTED Y** will differ on each row as it is the result of the calculated formula. In your table, based on the value of the X metric and the constants in the formula, it will return an estimated value for the Y metric.



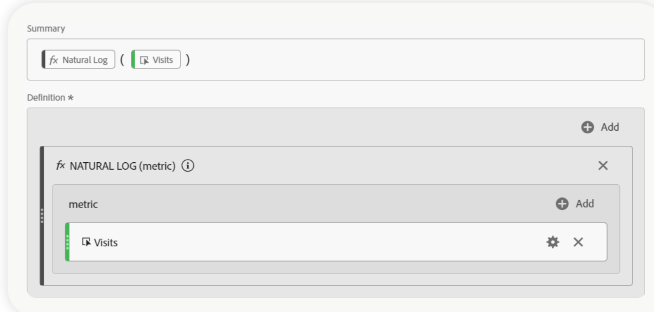
Reciprocal Regression

Day ↑ (Page: 1 / 1 Rows: 25)	Visits		Online Orders		Reciprocal Regression Correlation Coefficient (x=Visits, y=Online Orders)		Reciprocal Regression Predicted Y (x=Visits, y=Online Orders)		Reciprocal Regression Slope (x=Visits, y=Online Orders)		Reciprocal Regression Intercept (x=Visits, y=Online Orders)	
	253,588 out of 253,588		18,248 out of 18,248		-0.82 out of -0.82		2,734.90 out of 2,734.90		-25,900,938.00 out of -25,900,938.00		2,837.04 out of 2,837.04	
1. Apr 6, 2024	18,963	7.5%	1,399	7.7%	-0.82	100.0%	1,471.17	53.8%	-25,900,938.00	100.0%	2,837.04	100.0%
2. Apr 7, 2024	26,692	10.5%	2,298	12.6%	-0.82	100.0%	1,866.68	68.3%	-25,900,938.00	100.0%	2,837.04	100.0%
3. Apr 8, 2024	15,573	6.1%	995	5.5%	-0.82	100.0%	1,173.85	42.9%	-25,900,938.00	100.0%	2,837.04	100.0%
4. Apr 9, 2024	19,335	7.6%	1,028	5.6%	-0.82	100.0%	1,497.45	54.8%	-25,900,938.00	100.0%	2,837.04	100.0%
5. Apr 10, 2024	13,023	5.1%	1,025	5.6%	-0.82	100.0%	848.18	31.0%	-25,900,938.00	100.0%	2,837.04	100.0%
6. Apr 11, 2024	13,272	5.2%	946	5.2%	-0.82	100.0%	885.49	32.4%	-25,900,938.00	100.0%	2,837.04	100.0%
7. Apr 12, 2024	13,205	5.2%	1,016	5.6%	-0.82	100.0%	875.59	32.0%	-25,900,938.00	100.0%	2,837.04	100.0%
8. Apr 13, 2024	12,923	5.1%	981	5.4%	-0.82	100.0%	832.79	30.5%	-25,900,938.00	100.0%	2,837.04	100.0%
9. Apr 14, 2024	13,173	5.2%	999	5.5%	-0.82	100.0%	870.83	31.8%	-25,900,938.00	100.0%	2,837.04	100.0%
10. Apr 15, 2024	15,145	6.0%	949	5.2%	-0.82	100.0%	1,126.84	41.2%	-25,900,938.00	100.0%	2,837.04	100.0%
11. Apr 16, 2024	18,238	7.2%	969	5.3%	-0.82	100.0%	1,416.88	51.8%	-25,900,938.00	100.0%	2,837.04	100.0%
12. Apr 17, 2024	19,190	7.6%	1,435	7.9%	-0.82	100.0%	1,487.33	54.4%	-25,900,938.00	100.0%	2,837.04	100.0%
13. Apr 18, 2024	27,430	10.8%	2,223	12.2%	-0.82	100.0%	1,892.78	69.2%	-25,900,938.00	100.0%	2,837.04	100.0%
14. Apr 19, 2024	13,083	5.2%	997	5.5%	-0.82	100.0%	857.30	31.3%	-25,900,938.00	100.0%	2,837.04	100.0%
15. Apr 20, 2024	15,306	6.0%	988	5.4%	-0.82	100.0%	1,144.83	41.9%	-25,900,938.00	100.0%	2,837.04	100.0%

Data Modelling.

▪ I need to normalise my metrics…

Whether you're performing statistical tests such as regressions or just comparing metrics that are on different scales, there are two log-based options in the metric builder, **NATURAL LOG** and **LOG BASE 10**. The **NATURAL LOG** uses the constant e to return the logarithm, which is essentially the number of times e needs to be multiplied to return a number. This is the inverse of the function **EXPONENT**. The **LOG BASE** 10 is also called a common logarithm, it represents the number that 10 must be raised to a product of a given number. Both of these metrics are ways to normalise metrics that are very different so their change can be compared.



For example, if you want to see the change in visits and orders over time, you can use a log transformation of both metrics to compare them on the same scale. The two LOG functions are also important in some transformations for non-linear regression models.

To build the **NATURAL LOG** or **LOG BASE 10**, you only need one condition, the metric that you want to return the log value.

Natural Log

	Visits		Online Orders		Natural Log Visits		Natural Log Orders	
Day ↑ Page: 1 / 1 Rows: 25	253,588	out of 253,588	18,248	out of 18,248	12	out of 12	10	out of 10
1. Apr 6, 2024	18,963	7.5%	1,399	7.7%	10	79.2%	7	73.8%
2. Apr 7, 2024	26,692	10.5%	2,298	12.6%	10	81.9%	8	78.9%
3. Apr 8, 2024	15,573	6.1%	995	5.5%	10	77.6%	7	70.4%
4. Apr 9, 2024	19,335	7.6%	1,028	5.6%	10	79.3%	7	70.7%
5. Apr 10, 2024	13,023	5.1%	1,025	5.6%	9	76.1%	7	70.7%
6. Apr 11, 2024	13,272	5.2%	946	5.2%	9	76.3%	7	69.8%
7. Apr 12, 2024	13,205	5.2%	1,016	5.6%	9	76.3%	7	70.6%
8. Apr 13, 2024	12,923	5.1%	981	5.4%	9	76.1%	7	70.2%
9. Apr 14, 2024	13,173	5.2%	999	5.5%	9	76.2%	7	70.4%
10. Apr 15, 2024	15,145	6.0%	949	5.2%	10	77.4%	7	69.9%
11. Apr 16, 2024	18,238	7.2%	969	5.3%	10	78.8%	7	70.1%
12. Apr 17, 2024	19,190	7.6%	1,435	7.9%	10	79.3%	7	74.1%
13. Apr 18, 2024	27,430	10.8%	2,223	12.2%	10	82.1%	8	78.5%
14. Apr 19, 2024	13,083	5.2%	997	5.5%	9	76.2%	7	70.4%
15. Apr 20, 2024	15,306	6.0%	988	5.4%	10	77.4%	7	70.3%

When you put these in your table, the resulting value will be the log value of the metric. If you have multiple different metrics returned as logs, it makes it easier to compare changes in their values over time. For example, if you want to compare metrics that are vastly different, such as visits and orders, using log values can put them on the same scale so that you can visualise changes. While you can also use a line visualisation and select ‘normalisation’, this results in not having any scale, whereas with log values you can still see a numeric scale for the values.

- **I want some pi…**

There is a **PI** function available in the metric builder. It is (at the time of writing) one of only two metrics that don’t take any conditions. This function simply returns the value of mathematical **PI**. Although the documentation says it’s accurate to 15 decimal points, there isn’t a way to check that, as a calculated metric can have 10 decimal points at most.

To build this function, simply add it to your metric. No other arguments are needed.

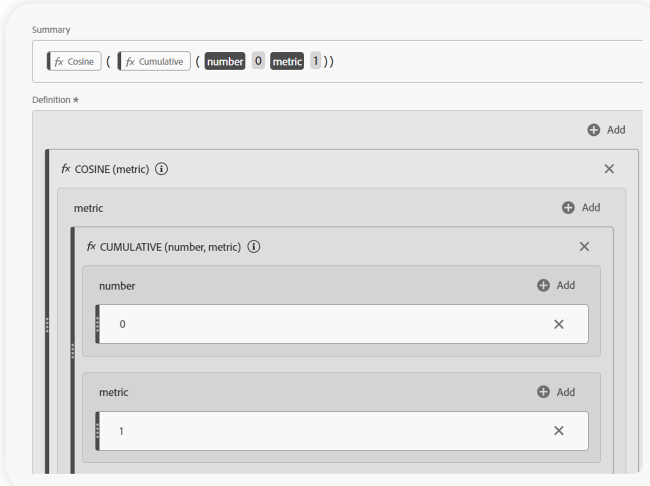
When you put it in your table, if there are no other calculations or functions in your metric, it will simply return the value of π. On its own, likely not useful. But there are several other trigonometry related functions that this can be used with.

• $\sqrt{-1}\ 2^3\ \Sigma\ \Pi$... it was delicious

Day ↑ Page: 1 / 1 Rows: 25	PI 3.1415926536 out of 3.1415926536	
1. Apr 6, 2024	3.1415926536	100.0%
2. Apr 7, 2024	3.1415926536	100.0%
3. Apr 8, 2024	3.1415926536	100.0%
4. Apr 9, 2024	3.1415926536	100.0%
5. Apr 10, 2024	3.1415926536	100.0%
6. Apr 11, 2024	3.1415926536	100.0%
7. Apr 12, 2024	3.1415926536	100.0%
8. Apr 13, 2024	3.1415926536	100.0%
9. Apr 14, 2024	3.1415926536	100.0%
10. Apr 15, 2024	3.1415926536	100.0%
11. Apr 16, 2024	3.1415926536	100.0%
12. Apr 17, 2024	3.1415926536	100.0%
13. Apr 18, 2024	3.1415926536	100.0%
14. Apr 19, 2024	3.1415926536	100.0%
15. Apr 20, 2024	3.1415926536	100.0%

■ I want to model how my data varies…

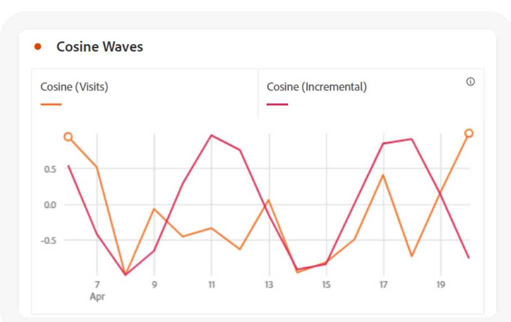
At first glance, the inclusion of these metrics can be a bit confusing. I'm sure for most people seeing **SINE**, **COSINE**, and **TANGENT** causes flashbacks to high school trigonometry, trying to find angles and side lengths. But these can also be applied to data modelling. Having a repeating sequence, such as a **SINE** or **COSINE** wave can provide a base pattern for events that happen in a cycle. **TANGENTS** can help with linear and non-linear models centred around a specific point.

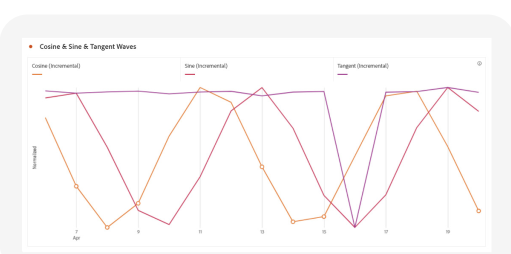


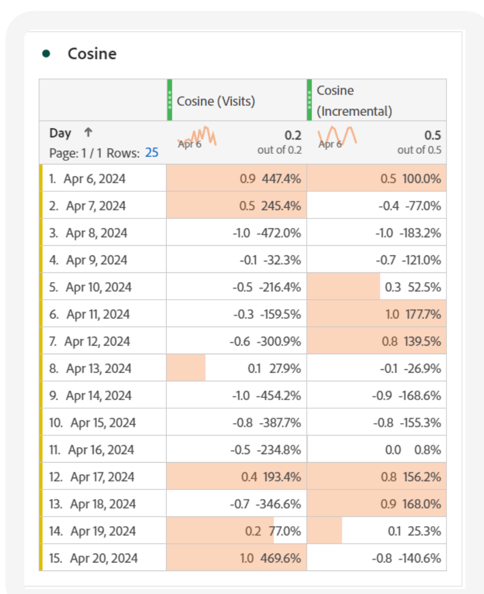
One way to go about modelling with these functions is to create a **MEAN** and **STANDARD DEVIATION** metric, and graph them to look at the variance in the data. Then you can use a SINE function to get values from –1 to +1 and multiply it by the standard deviation to amplify the natural sine wave. Depending on how deep into the data science you want to get, there are methods to layer the various waves on top of each other.

To build a **SINE**, **COSINE**, or **TANGENT** function, you only need one condition, the metric used to generate the wave. You can technically use any metric in generating the wave, such as visits. But you can also use a nested **CUMULATIVE** function with a static number of 1 to generate a traditional wave.

When you put these types of metrics in your table, the values on each of the rows will cycle through the wave. It's also good to visualise these with line graphs where you can see the waves.







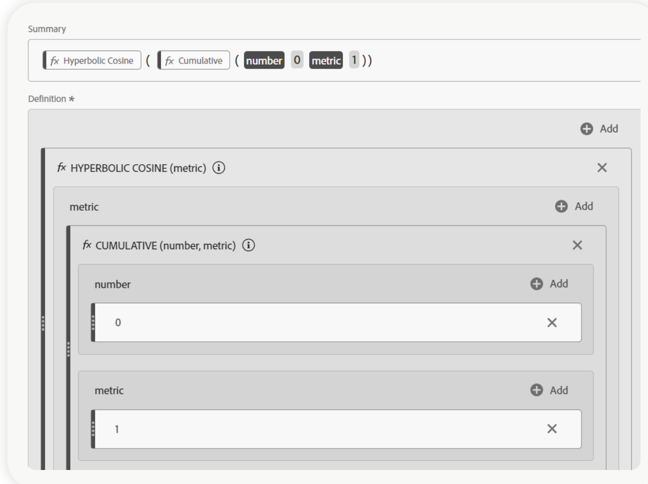
Cosine

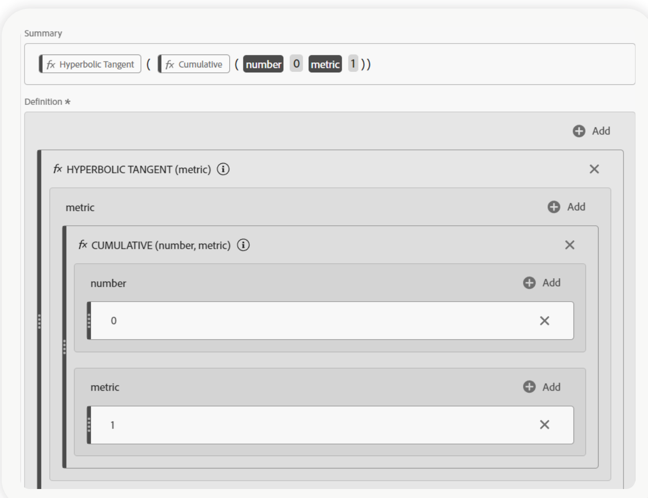
Day ↑ Page: 1 / 1 Rows: 25	Cosine (Visits) 0.2 out of 0.2	Cosine (Incremental) 0.5 out of 0.5
1. Apr 6, 2024	0.9 447.4%	0.5 100.0%
2. Apr 7, 2024	0.5 245.4%	-0.4 -77.0%
3. Apr 8, 2024	-1.0 -472.0%	-1.0 -183.2%
4. Apr 9, 2024	-0.1 -32.3%	-0.7 -121.0%
5. Apr 10, 2024	-0.5 -216.4%	0.3 52.5%
6. Apr 11, 2024	-0.3 -159.5%	1.0 177.7%
7. Apr 12, 2024	-0.6 -300.9%	0.8 139.5%
8. Apr 13, 2024	0.1 27.9%	-0.1 -26.9%
9. Apr 14, 2024	-1.0 -454.2%	-0.9 -168.6%
10. Apr 15, 2024	-0.8 -387.7%	-0.8 -155.3%
11. Apr 16, 2024	-0.5 -234.8%	0.0 0.8%
12. Apr 17, 2024	0.4 193.4%	0.8 156.2%
13. Apr 18, 2024	-0.7 -346.6%	0.9 168.0%
14. Apr 19, 2024	0.2 77.0%	0.1 25.3%
15. Apr 20, 2024	1.0 469.6%	-0.8 -140.6%

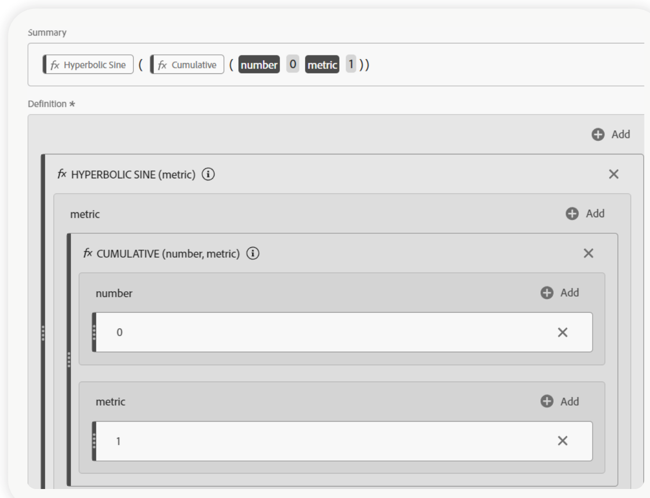
I want to make a hyperbolic or inverse model...

In addition to the basic **SINE**, **COSINE**, and **TANGENT** functions, there are also **HYPERBOLIC** versions of each function. Rather than using a circle to define the functions, the **HYPERBOLIC SINE**, **HYPERBOLIC COSINE**, and **HYPERBOLIC TANGENT** use a hyperbola curve instead.

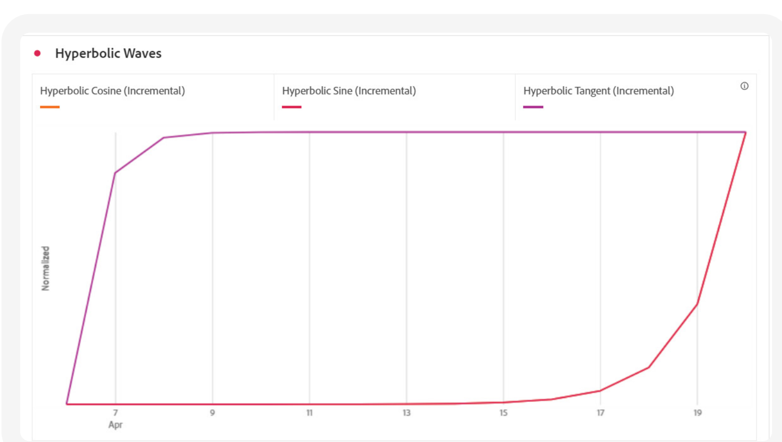
The way that you build the metrics is simply by adding in the value you want to use to generate the curves. Like we've seen before, we can use an actual metric (such as visits), or we can use the **CUMULATIVE** function with a value of 1 to generate a true curve.







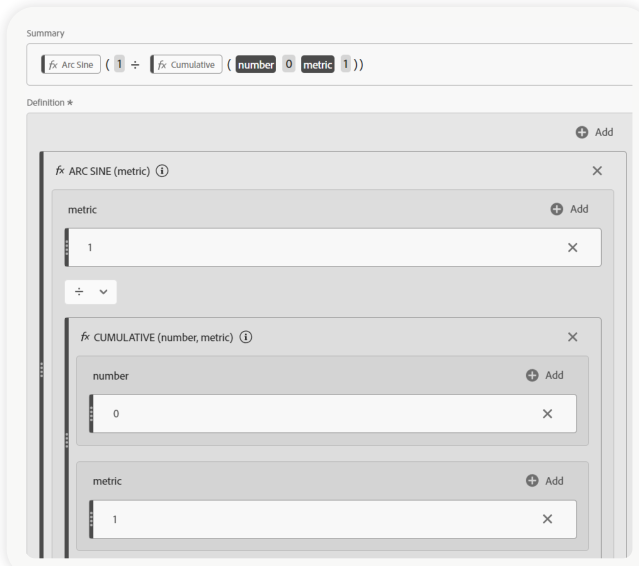
The direction of the curve for the **HYPERBOLIC TANGENT** is opposite to the direction of the **HYPERBOLIC SINE** and **HYPERBOLIC COSINE** curves. These can also get very large very fast, so you might need to adjust your scale (such as dividing by a factor of 100 or more). The curves are graphed below but are normalised so that they can be shown on the same scale.

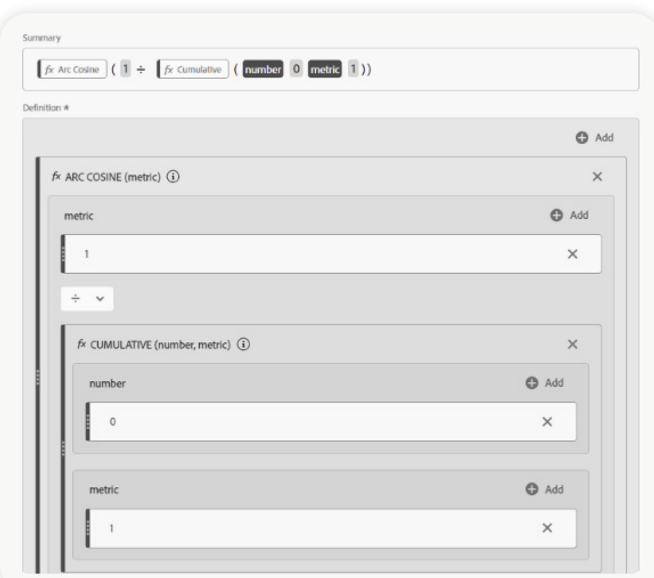


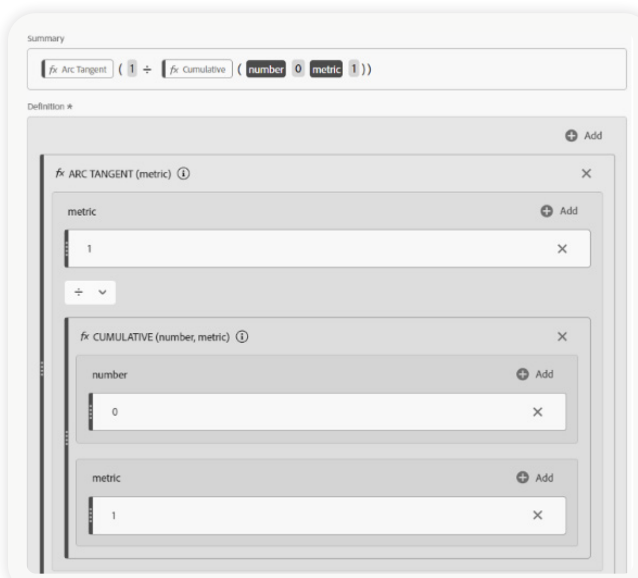
● Hyperbolic Sine - Cosine - Tangent

Day ↑ Page: 1 / 1 Rows: 25	Hyperbolic Cosine (Incremental) 1.54 out of 1.54		Hyperbolic Sine (Incremental) 3.63 out of 3.63		Hyperbolic Tangent (Incremental) 0.76 out of 0.76	
1. Apr 6, 2024	1.54	100.0%	3.63	100.0%	0.76	100.0%
2. Apr 7, 2024	3.76	243.8%	10.02	276.2%	0.96	126.6%
3. Apr 8, 2024	10.07	652.4%	27.29	752.4%	1.00	130.7%
4. Apr 9, 2024	27.31	1,000.0+%	74.20	1,000.0+%	1.00	131.2%
5. Apr 10, 2024	74.21	1,000.0+%	201.71	1,000.0+%	1.00	131.3%
6. Apr 11, 2024	201.72	1,000.0+%	548.32	1,000.0+%	1.00	131.3%
7. Apr 12, 2024	548.32	1,000.0+%	1,490.48	1,000.0+%	1.00	131.3%
8. Apr 13, 2024	1,490.48	1,000.0+%	4,051.54	1,000.0+%	1.00	131.3%
9. Apr 14, 2024	4,051.54	1,000.0+%	11,013.23	1,000.0+%	1.00	131.3%
10. Apr 15, 2024	11,013.23	1,000.0+%	29,937.07	1,000.0+%	1.00	131.3%
11. Apr 16, 2024	29,937.07	1,000.0+%	81,377.40	1,000.0+%	1.00	131.3%
12. Apr 17, 2024	81,377.40	1,000.0+%	221,206.70	1,000.0+%	1.00	131.3%
13. Apr 18, 2024	221,206.70	1,000.0+%	601,302.14	1,000.0+%	1.00	131.3%
14. Apr 19, 2024	601,302.14	1,000.0+%	1,634,508.69	1,000.0+%	1.00	131.3%
15. Apr 20, 2024	1,634,508.69	1,000.0+%	4,443,055.26	1,000.0+%	1.00	131.3%

Analysis Workspace also has functions for the **ARC SINE**, **ARC COSINE**, and **ARC TANGENT**. Each of these is the inverse of their non-arc functions. Depending on the type of data that you're trying to model, inverse functions may produce better results. Because these are inverse, you need to use the inverse of your metric, so taking 1 over the metric. Depending on what metric you're using, this can result in very small numbers, so when putting it in a table (or a graph), you might need to use extra decimal places or multiply the metric by a factor of 100 (or more) to get numbers that are distinguishable from one another.







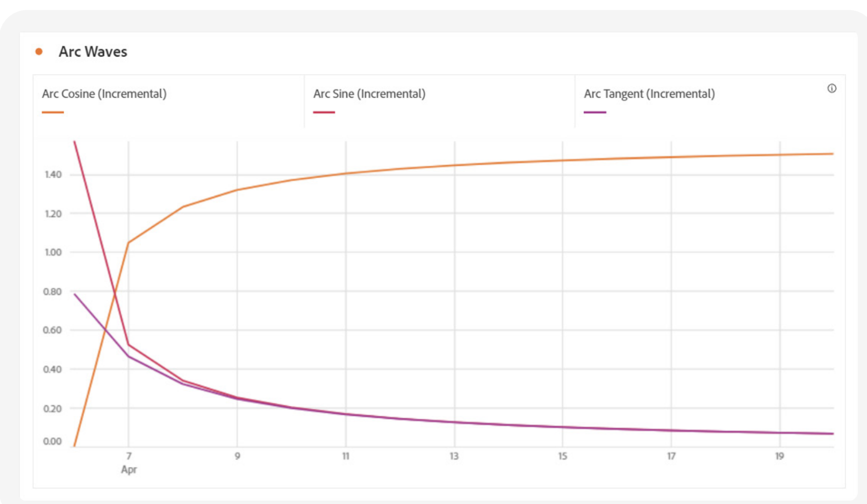
To build these functions, you need to bring in the metric you want to use to generate the curve. However, unlike the other trigonometry functions, with the **ARC SINE**, **ARC COSINE**, and **ARC TANGENT** you need to use the inverse of your metric.

The **ARC** curves are a bit easier to graph than the **HYPERBOLIC** ones, as they don't get as large as fast. Using the **CUMULATIVE** function, you can see the true waves. But again, you can replace that function with whatever metric you want to model.

Each of these can also be used to layer in different types of waves and create different data models. Most data analysts won't use these very frequently, if at all. But for those that are involved in data science, they can be quite useful.

Arc Sine - Cosine - Tangent

Day ↑ Page: 1 / 1 Rows: 25	Arc Cosine (Incremental) 0.00 out of 0.00		Arc Sine (Incremental) 1.57 out of 1.57		Arc Tangent (Incremental) 0.7853982 out of 0.7853982	
1. Apr 6, 2024	0.00	0.0%	1.57	100.0%	0.7853982	100.0%
2. Apr 7, 2024	1.05	1,000.0+%	0.52	33.3%	0.4636476	59.0%
3. Apr 8, 2024	1.23	1,000.0+%	0.34	21.6%	0.3217506	41.0%
4. Apr 9, 2024	1.32	1,000.0+%	0.25	16.1%	0.2449787	31.2%
5. Apr 10, 2024	1.37	1,000.0+%	0.20	12.8%	0.1973956	25.1%
6. Apr 11, 2024	1.40	1,000.0+%	0.17	10.7%	0.1651487	21.0%
7. Apr 12, 2024	1.43	1,000.0+%	0.14	9.1%	0.1418971	18.1%
8. Apr 13, 2024	1.45	1,000.0+%	0.13	8.0%	0.1243550	15.8%
9. Apr 14, 2024	1.46	1,000.0+%	0.11	7.1%	0.1106572	14.1%
10. Apr 15, 2024	1.47	1,000.0+%	0.10	6.4%	0.0996687	12.7%
11. Apr 16, 2024	1.48	1,000.0+%	0.09	5.8%	0.0906599	11.5%
12. Apr 17, 2024	1.49	1,000.0+%	0.08	5.3%	0.0831412	10.6%
13. Apr 18, 2024	1.49	1,000.0+%	0.08	4.9%	0.0767719	9.8%
14. Apr 19, 2024	1.50	1,000.0+%	0.07	4.6%	0.0713075	9.1%
15. Apr 20, 2024	1.50	1,000.0+%	0.07	4.2%	0.0665682	8.5%



Afterword.

There are an infinite number of ways that you can use calculated metrics and the various functions in them. For those that are very maths or statistics-minded, digging into these different functions and finding interesting use cases can be quite enjoyable (hence why I wrote this guide!). There are many, many possible use cases beyond what I've outlined here, so don't be afraid to experiment. In this guide I mostly focused on how to use functions relatively independently, but there are numerous ways that you can nest functions inside of each other to combine their capabilities and take your analyses to the next level. This is especially true for logical comparisons; the ways that you can change how your data is presented are infinite. Make sure you save this document and refer back to it when needed.

Happy Calculating!

This document was authored by Adobe Analytics Champion Mandy George