



# Synthetic Variable Operations

Update date: November 13, 2019

Operator / Function	Description	Real Time Capability	Example	Result
<b>Mathematical Functions</b>				
<code>ceil({{raw}})</code>	Returns the rounded integer greater or equal to every value in a time series, as a integer. Ceil always rounds up to the nearest integer.	NO	<code>ceil([1.2, 2, 3.7])</code>	[2,2,4]
<code>floor({{raw}})</code>	Returns the rounded integer less or equal to every value in the time series, as a integer. Floor always round down to the nearest integer.	NO	<code>floor([1.2, 2, 3.7])</code>	[1,2,3]
<code>round({{raw}}, n)</code>	Returns the floating point value rounded to the "n" digits after the decimal point for every value in a time series. By default, n equals 2 in Ubidots.	NO	<code>round([1.22222, 2.9994332], 3)</code>	[1.222, 2.999]
<code>tan({{raw}})</code>	Returns the tangent of every value in radians in a time series.	NO	<code>tan([0, 90])</code>	[0, -1.99520041221]
<code>cos({{raw}})</code>	Returns the cosine of every value in radians in a time series.	NO	<code>cos([0, 90])</code>	[1, -0.44807361612]
<code>sin({{raw}})</code>	Returns the sine of every value in radians in a time series.	NO	<code>sin([0, 90])</code>	[0, 0.8939966636]
<code>arcsin({{raw}})</code>	Returns in radians the inverse sine of every value in the time series.	NO	<code>arcsin([0, 90])</code>	[0]
<code>arccos({{raw}})</code>	Returns in radians the inverse cosine of every value in the time series.	NO	<code>arccos([0, 90])</code>	[1.5707963]
<code>arctan({{raw}})</code>	Returns in radians the inverse tangent of every value in the time series.	NO	<code>arctan([0, 90])</code>	[0]
<code>arctan2({{raw-x}}, {{raw-y}})</code>	Returns in radians the trigonometric inverse tangent using as cartesian coordinates the input time series. Note: Will only perform the operation between values with the same timestamp.	NO	Assuming that the time series is sampled every minute <code>arctan2([1, 2], [0.1, 1])</code>	[1.471127, 1.1071]
<code>sinh({{raw}})</code>	Returns the hyperbolic sine of every value in the time series.	NO	<code>sinh([0, 90])</code>	[0, 6.1020exp38]
<code>cosh({{raw}})</code>	Returns the hyperbolic cosine of every value in the time series.	NO	<code>cosh([0, 90])</code>	[1, 6.1020exp38]
<code>tanh({{raw}})</code>	Returns the hyperbolic tangent of every value in the time series.	NO	<code>tanh([0, 90])</code>	[0, 1]
<code>arcsinh({{raw}})</code>	Returns in radians the inverse hyperbolic sine of every value in the time series.	NO	<code>arcsinh([0, 90])</code>	[0, 5.1929877136589]
<code>arccosh({{raw}})</code>	Returns in radians the inverse hyperbolic cosine of every value in the time series.	NO	<code>arccosh([0, 90])</code>	[5.1929877136589]
<code>arctanh({{raw}})</code>	Returns in radians the inverse hyperbolic tangent of every value in the time series.	NO	<code>arctanh([0, 90])</code>	[0]
<code>exp({{raw}})</code>	Returns the exponential of every value in the time series.	NO	<code>exp([-1, 0, 1, 2])</code>	[0.367, 1.0, 2.718, 7.389]
<code>log({{raw}}, base)</code>	Returns the logarithm of every value in the time series. By default the base is the Euler's number.	NO	<code>log([1, 2])</code>	[0, 0.6931471805599]
<code>abs({{raw}})</code>	Returns the absolute value of every data in the time series.	NO	<code>abs([-1, 0, 1, 2])</code>	[1, 0, 1, 2]
<code>sqrt({{raw}})</code>	Returns the square root value of every data in the time series	NO	<code>sqrt[1.4]</code>	[1,2]
<b>Data Range Functions</b>				
<code>max({{raw}}, "data_range")</code>	Calculates the maximum value of the variable in the specified data range.	NO	Assuming that values are sampled every second: <code>max([1, 2, 3, -1], "T")</code>	3
<code>min({{raw}}, "data_range")</code>	Calculates the minimum value of the variable in the specified data range.	NO	Assuming that values are sampled every hour: <code>min([1, 2, 3, -1], "4H")</code>	-1
<code>std({{raw}}, "data_range")</code>	Calculates the standard deviation of the variable in the specified data range.	NO	Assuming that values are sampled every minute: <code>std([1, 2], "T")</code>	[0.707]
<code>mean({{raw}}, "data_range")</code>	Calculates the mean value of the variable in the specified data range.	NO	Assuming that values are sampled every minute: <code>mean([1, 2, 3, 0, -1], "5T")</code>	1
<code>median({{raw}}, "data_range")</code>	Calculates the median value of the variable in the specified data range.	NO	Assuming that values are sampled every hour: <code>median([1, 2, 3, 0, -1], "D")</code>	1
<code>count({{raw}}, "data_range")</code>	Calculates the number of dots in the specified data range.	NO	Assuming that values are sampled every day: <code>count([1, 2, 3, 0, -1], "W")</code>	5
<code>last({{raw}}, "data_range")</code>	Calculates the last value of the variable in the specified data range.	NO	Assuming that values are sampled every month: <code>last([1, 2, 3, 0, -1], "4M")</code>	-1
<code>first({{raw}}, "data_range")</code>	Calculates the first value of the variable in the specified data range.	NO	Assuming that values are sampled every month: <code>frist([1, 2, 3, 0, -1], "4M")</code>	1
<code>sum({{raw}}, "data_range")</code>	Calculates the summation of the time series in the specified data range.	NO	Assuming that values are sampled every minute: <code>sum([1, 2, 3, 0, -1], "4T")</code>	5

Operator / Function	Description		Example	Result
<b>Available Data Ranges</b>				
"nT"	Returns a value representing a data range of every "n" number of MINUTE(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every second: max([1, 2, 3, 0, -1], "T")	3
"nH"	Returns a value representing a data range of every "n" number of HOUR(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every minute: min([1, 2, 3, 0, -1], "H")	-1
"nD"	Returns a value representing a data range of every "n" number of DAY(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every hour: mean([1, 2, 3, 0, -1], "D")	1
"nW"	Returns a value representing a data range of every "n" number of WEEK(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every day: count([1, 2, 3, 0, -1], "W")	5
"nM"	Returns a value representing a data range of every "n" number of MONTH(S); all ranges must be entered as strings; ie: with "quotes"		Assuming that values are sampled every week: last([1, 2, 3, 0, -1], "M")	-1
<b>Special Functions</b>				
where(condition, operation if fits, operation if not fits)	If-else statement. Variables attributes like context key or timestamp can be accessed using the dot, '.', operator	NO	Assuming that {{raw}} time series is equals to [-1, 2, 1] Step Function, unit(x): where({{raw}}>=0, 1, 0) Interval function: where({{raw}}<0, 0, where({{raw}}<1), 1, 2)	[0, 1, 1] [0, 2, 2]
fill_missing(x)	When performing operations between multiple variables timestamps, the function will enter the last value of a variable when an expression requires data from a timestamp that does not match the other timestamps within the expression.  Note: fill_missing() computes the operation between the whole time series once a new value arrives to any of the raw variables in the operation. Ubidots does not advise using this operation for real time applications.	NO	{{raw_1}} = [1, 2, 3], sampled every minute {{raw_2}} = [1, 5], sampled every 2 minutes fill_missing({{raw_1}} + {{raw_2}})	[2, 3, 8]
fill_missing(x, first_fill="ffill", last_fill=None, fill_value=None)	fill_missing recives additional parameter to fill gaps forward, backward or with a value. <b>"ffill"</b> : Fill gaps forward <b>"bfill"</b> : Fill gaps backward and <b>"fill_value = 0"</b> : Fill gaps with the number entered.	NO	{{raw_1}} = [1, 2, 3], sampled every minute {{raw_2}} = [1, 5], sampled every 2 minutes fill_missing({{raw_1}} + {{raw_2}}, first_fill="ffill") fill_missing({{raw_1}} + {{raw_2}}, first_fill="bfill") fill_missing({{raw_1}} + {{raw_2}}, fill_value=0)	[2, 3, 8] [2, 7, 8] [2, 2, 8]
shift({{raw}}, n)	Returns values of the variable by the specified number of (+/-) n steps in the time series. Note: N must be entered as an integer, not a string.	NO	shift([-2, -1, 0, 1, 2], 1) shift([-2, -1, 0, 1, 2], 2) actual value minus previous value: [0, 1, 3] - shift([0, 1, 3], -1)	[-1, 0, 1, 2] [0, 1, 2] [1, 2]
cumsum({{raw}})	Calculates the cumulative sum of the time series.	NO	cumsum[0,1,2,3]	6
rolling({{variable}}, {{aggregation}}	Calculates the moving agregation methos of the time series, using the specified parameters	NO	rolling({{variable}},mean,values,5T, min_periods=2)	
diff({{raw}}, steps)	Calculates the difference staring at the last element in a time series and the next separated by a specified number of steps.	NO	diff([14, 15, 17, 16])	[3, 1]