ubidots Synthetic Variable Operations

Operator / Function	Description	Real Time Capability	Example	Result
T				
`ceil({{raw}})`	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.	YES	`ceil([1.2, 2, 3.7])`	`[2,2,4]`
`floor({{raw}})`	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.	YES	`floor([1.2, 2, 3.7])`	`[1,2,3]`
`round({{raw}}, n)`	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.	YES	`round([1.22222, 2.9994332], 3)`	`[1.222, 2.999]`
Trigonomics				
`tan({{raw}})`	Returns the tangent of every value in radians in a time series.	YES	`tan([0, 90])`	`[0, -1.99520041221]`
`cos({{raw}})`	Returns the cosine of every value in radians in a time series.	YES	`cos([0, 90])`	`[1, -0.44807361612]`
`sin({{raw}})`	Returns the sine of every value in radians in a time series.	YES	`sin([0, 90])`	`[0, 0.8939966636]`
`arcsin({{raw}})`	Returns in radians the inverse sine of every value in the time series.	YES	`arcsin([0, 90])`	`[0]`
`arccos({{raw}})`	Returns in radians the inverse cosine of every value in the time series.	YES	`arccos([0, 90])`	`[1.5707963]`
`arctan({{raw}})`	Returns in radians the inverse tangent of every value in the time series.	YES	`arctan([0, 90])`	`[0]`
`arctan2({{raw_x}}, {{raw_y}})`	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.	YES	Assuming that the time series is sampled every minute `arctan2([1, 2], [0.1, 1])`	`[1.471127, 1.1071]`
Mathematics				
`sinh({{raw}})`	Returns the hyperbolic sine of every value in the time series.	YES	`sinh([0, 90])`	`[0, 6.1020exp38]`
`cosh({{raw}})`	Returns the hyperbolic cosine of every value in the time series.	YES	`cosh([0, 90])`	`[1, 6.1020exp38]`
`tanh({{raw}})`	Returns the hyperbolic tangent of every value in the time series.	YES	`tanh([0, 90])`	`[0, 1]`
`exp({{raw}})`	Returns the exponential of every value in the time series.	YES	`exp([-1, 0, 1, 2])`	`[0.36787944117144233, 1.0, 2.718281828459045, 7.38905609893065]`
Data Range Functions				
`max({{raw}}, "data_range")`	Calculates the maximum value of the variable in the specified data range.	NO	Assuming that values are sampled every minute: `max([1, 2, 3, 0, -1], '5T')`	`3`
`min({{raw}}, "data_range")`	Calculates the minimum value of the variable in the specified data range.	NO	Assuming that values are sampled every hour: `min([1, 2, 3, 0, -1], '5H')`	`-1`
`mean({{raw}}, "data_range")`	Calculates the mean value of the variable in the specified data range.	NO	Assuming that values are sampled every day: `mean([1, 2, 3, 0, -1], '5D')`	`1`
`count({{raw}}, "data_range")`	Calculates the number of dots in the specified data range.	NO	Assuming that values are sampled every week: `count([1, 2, 3, 0, -1], '5W')`	`5`
`last({{raw}}, "data_range")`	Calculates the last value of the variable in the specified data range.	NO	Assuming that values are sampled every month: `last([1, 2, 3, 0, -1], '5M')`	`-1`
`sum({{raw}}, "data_range")`	Calculates the summation of the time series in the specified data range.	NO	Assuming that values are sampled every minute: `sum([1, 2, 3, 0, -1], '5T')`	`5`

Operator / Function	Description		Example	Result
Available Data Ranges				
`"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 minute: `max([1, 2, 3, 0, -1], '5T')`	`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 hour: `min([1, 2, 3, 0, -1], '5H')`	`-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 day: `mean([1, 2, 3, 0, -1], '5D')`	`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 week: `count([1, 2, 3, 0, -1], '5W')`	`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 month: `last([1, 2, 3, 0, -1], '5M')`	`-1`
Special Functions				

`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(f(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]`

`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]`