Coinalyze custom metrics

Introduction

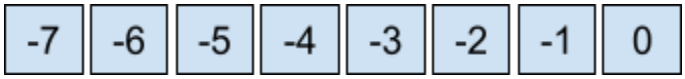
Coinalyze allows you to build your own metrics based on our data series: price, open interest, volume, buy volume, sell volume, liquidations, funding rate, predicted funding rate, long/short ratio, market cap and spot volume. Custom metrics are quite similar to Excel formulas. You create custom metrics by writing mathematical expressions/formulas. Once you have created a custom metric you can display it on the Coinalyze homepage listing as a table column and you can use it in the filter builder as well.

The key concepts of Coinalyze custom metrics formulas are: data series, data series timeframe/granularity and data series element(s) selection(referring/getting element(s) from the data series).

This is an expression for example fr_1h[0] Where:

- fr means the data series, in this case funding rate. See the “Series” section of this document for more details about the data series
- 1h means the timeframe
- [0] means the element selection. 0 index means the last element of the data series

Referring/accessing element(s) in the data series



The above image is a visual representation of a data series. The squares represent the series data-points/values. The 0 and the negative numbers represent the index for each data-point/value in the series. So [0] refers to the last value, [-1] refers to the previous value and so one.

You can select a range, you do that by using the “:” separator. For example [0:-4] refers to the last 5 elements in the series. [-2:-6] refers to the elements from the index -2 to -6 . The indexes are inclusive.

You can select two specific values from the series, you do that by using the “,” separator. For example [0,-2] refers to the elements with the index 0 and -2 This type of selection is used in the CHANGE() and PCHANGE() built-in functions. Those functions calculate the change, respectively percentage change, between two values in the data series.

Series

We provide the following data series, we will add even more in the near future.

- price - close price
- highprice - high price
- lowprice - low price
- mc - market capitalization
- vol - volume on the futures market
- volbuy - buy volume on the futures market
- volsell - sell volume on the futures market
- oi - open interest
- liq - liquidation (both longs and shorts)
- liqlong - longs liquidation
- liqshort - shorts liquidation
- fr - funding rate
- pfr - predicted funding rate
- lsr - long/short ratio
- volspot - volume on the spot market
- volspotbuy - buy volume on the spot market
- volspotsell - sell volume on the spot market

Timeframes

- 5m - 5 minute
- 1h - 1 hour timeframe
- 1d - 1 day timeframe

Series-timeframe

You combine the data series and timeframe using the "_" character. Examples:

- oi_1h open interest on 1 hour timeframe
- fr_1d daily funding rate
- price_1h hourly price
- pfr_5m predicted funding rate on 5 minute timeframe

Built-in functions

Functions you can use in your expression/formula

Math functions:

- SUM() calculates the sum over the specified range. Example SUM(vol_1h[0:-3]) It means: the volume over the last 4 hours. Approximately 4 hours, because the last interval/candlestick is not fully completed.

- AVG() calculates the average over a specified range. Example AVG(vol_1d[0:-10])

- MAX() calculates the maximum value over a specified range. Example MAX(fr_1h[0:-10])

- MIN() calculates the minimum value over a specified range. Example MIN(fr_1h[0:-10])

- CHANGE() calculates the absolute change between two specified values. Example CHANGE(oi_1d[0,-1]) Current/today open interest minus yesterday open interest. Basically it's a shorthand for oi_1d[0] - oi_1d[-1]

- PCHANGE() calculates the percentage change between two specified values. Example PCHANGE(oi_1d[0,-1]) Percentage change between current/today open interest and yesterday. Basically it's a shorthand for ((oi_1d[0] - oi_1d[-1]) / oi_1d[-1]) * 100

Technical indicator functions. These functions produce/generate series:

- SMA(series, length) calculates the simple moving average. Example SMA(price_1h, 10) Moving average over hourly price.

  Because the result is also a series you can use square brackets to access series element(s). Example SMA(price_1h, 10)[-1] You can also use the math functions. Example PCHANGE(SMA(price_1h, 10)[0, -20])

- CVD(timeframe, length, normalize) calculated the cumulative volume delta on future markets. Example CVD('1h', 20, false)

  The normalize parameter is a boolean parameter, by default it is false. If you omit the parameter, by default it is false. CVD('1h', 20) is similar to CVD('1h', 20, false)

  If the normalize parameter is true then the CVD values are normalized to the [0 100] interval. This parameter is useful when you want to calculate percentage change in CVD. Example PCHANGE(CVD('1h', 20, true)[0,-10])

- CVDSPOT(timeframe, length, normalize) calculates the cumulative volume delta on spot markets. Same parameters as CVD()

- RSI(series, length) calculates the Relative Strength Index. Example RSI(price_1h, 10)[0]

- ROC(series, length) calculates the Price Rate of Change. Example ROC(price_1h, 10)[-1]

Examples

- (SUM(liq_1h[0:-3]) / SUM(vol_1h[0:-3])) * 100 the ratio between liquidations and volume over the last 4 hours(approximately 4 hours)

- PCHANGE(pfr_5m[0, -11]) percentage change in predicted funding rate over the last hour

- (SUM(liqlong_1h[0:-23]) / SUM(liq_1h[0:-23])) * 100 24 hours longs liquidation/total liquidations ratio

- (SUM(liqshort_1h[0:-23]) / SUM(liq_1h[0:-23])) * 100 24 hours shorts liquidation/total liquidations ratio

- PCHANGE (mc_1d[0,-6]) percentage change in market cap over the last 7 days

- lsr_1h[0] display the current long/short ratio on 1 hour timeframe

- PCHANGE([SUM(vol_1h[-1:-4]), SUM(vol_1h[-5:-8])]) percentage change in volume between the previous 4 hours and the 4 hours before previous 4 hours

- CHANGE(vol_1d[-1,-2]) volume change between yesterday and the day before yesterday

- PCHANGE(cvdspot('1h', 20, true)[0,-10]) percentage change in spot cumulative volume delta

Observations

- Formulas are case-insensitive. E.g. SUM() and sum() are equivalent, OI_1H[0] and oi_1h[0] are equivalent. volbuy_1h[0] and volBuy_1h[0] are equivalent.

- Series length: for 5m timeframe = 300; for 1h timeframe = 90; for 1d timeframe = 90. E.g. The oi_5m series has a maximum of 300 values, the oi_1h series has a maximum of 90

values and the oi_1d has a maximum of 90 values. Be aware about that especially when you use the indicator functions(the length parameter).

- Series data is not updated in real time(yet), there is a 5 - 10 seconds delay.