Advanced candlesticks for machine learning (i): tick bars

In this article we will learn how to build tick bars, we will thoroughly analyze their statistical properties such as normality of returns or autocorrelation and we will explore in which scenarios these bars can be a good substitute for traditional time-based candlesticks. In order to illustrate the applicability of tick bars in the forecasting of cryptocurrency markets, we will base our analysis on a whole dataset comprising 16 cryptocurrency trading pairs including the most popular cryptoassets such as Bitcoin, Ethereum or Litecoin

Gerard Martínez · Follow
Published in Towards Data Science
9 min read · Apr 24, 2019

1. — Introduction

In a previous article we explored why traditional time-based candlesticks are not the most suitable price data format if we are planning to train a machine learning (ML)

algorithm. Namely: (1) time-based candlesticks over-sample low activity periods and under-sample high activity periods, (2) markets are increasingly controlled by trading algorithms that no longer follow any human-related daylight cycle, (3) the use of time-based candlesticks is ubiquitous among traders and trading bots, which increases the competition and, as we will see in this article, (4) time-based candlesticks offer poorer statistical properties. Find the link to the article below in case you missed the update.

Financial Machine Learning practitioners have been using the wrong candlesticks: here's why

In this article we will explore why traditional time-based candlesticks are an inefficient method to aggregate price...

towardsdatascience.com

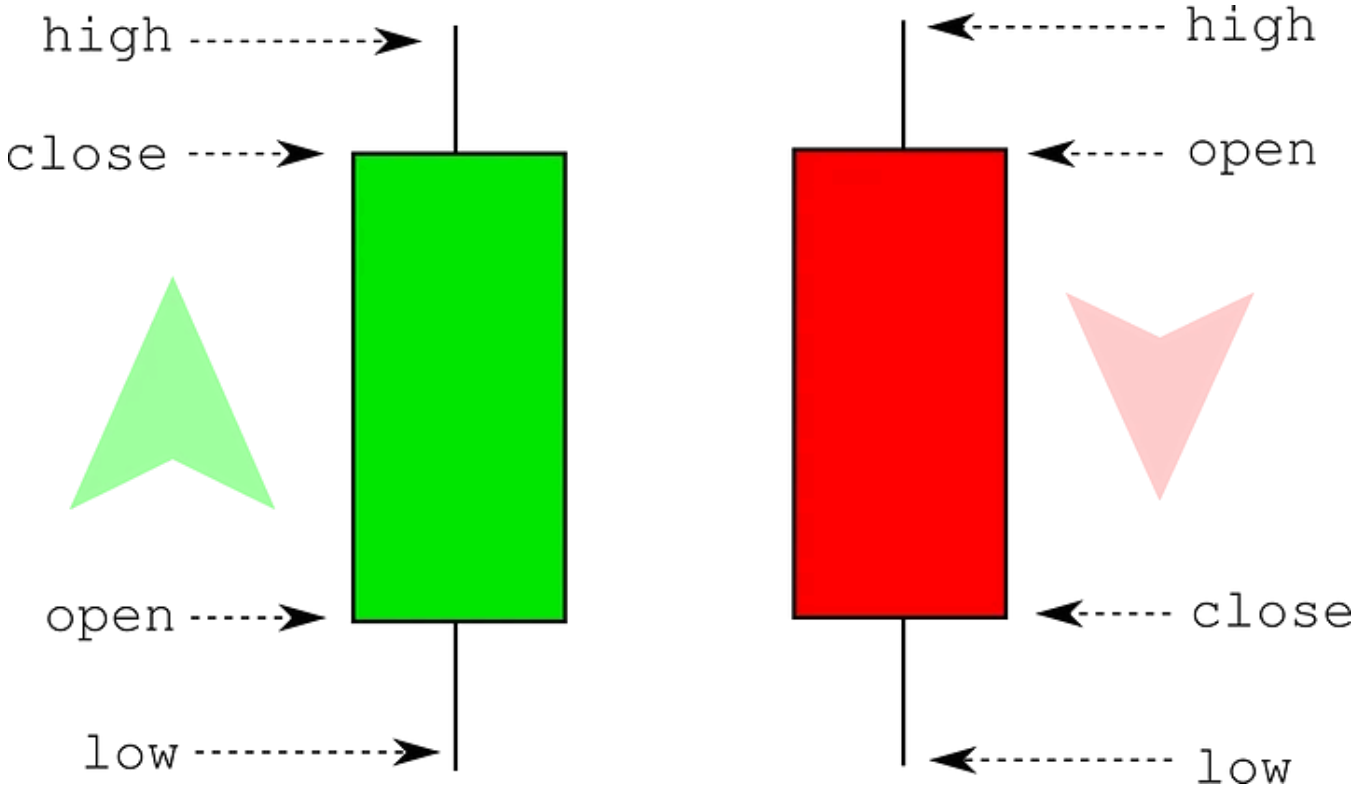
In this article we will explore one of the alternative bars that were proposed: tick bars. Let's dig into them.

2. — Building tick bars

There are at least two main definitions of what a tick is. Quoting investopedia:

> *A tick is a measure of the minimum upward or downward movement in the price of a security. A tick can also refer to the change in the price of a security from trade to trade.*

In the case of tick bars, the definition we care about is the second one: in the scope of tick bars, a tick is essentially a trade and the price at which the trade was made in the exchange. A tick bar or a tick candle is simply the aggregation of a predefined number of ticks. For instance, if we want to generate 100-tick bars we must keep a store of all trades and every time we "receive" 100 trades from the exchange we build a bar or candlestick. Candlesticks are then built by calculating the Open, High, Low, Close and Volume values (they are usually shortened to OHLCV).



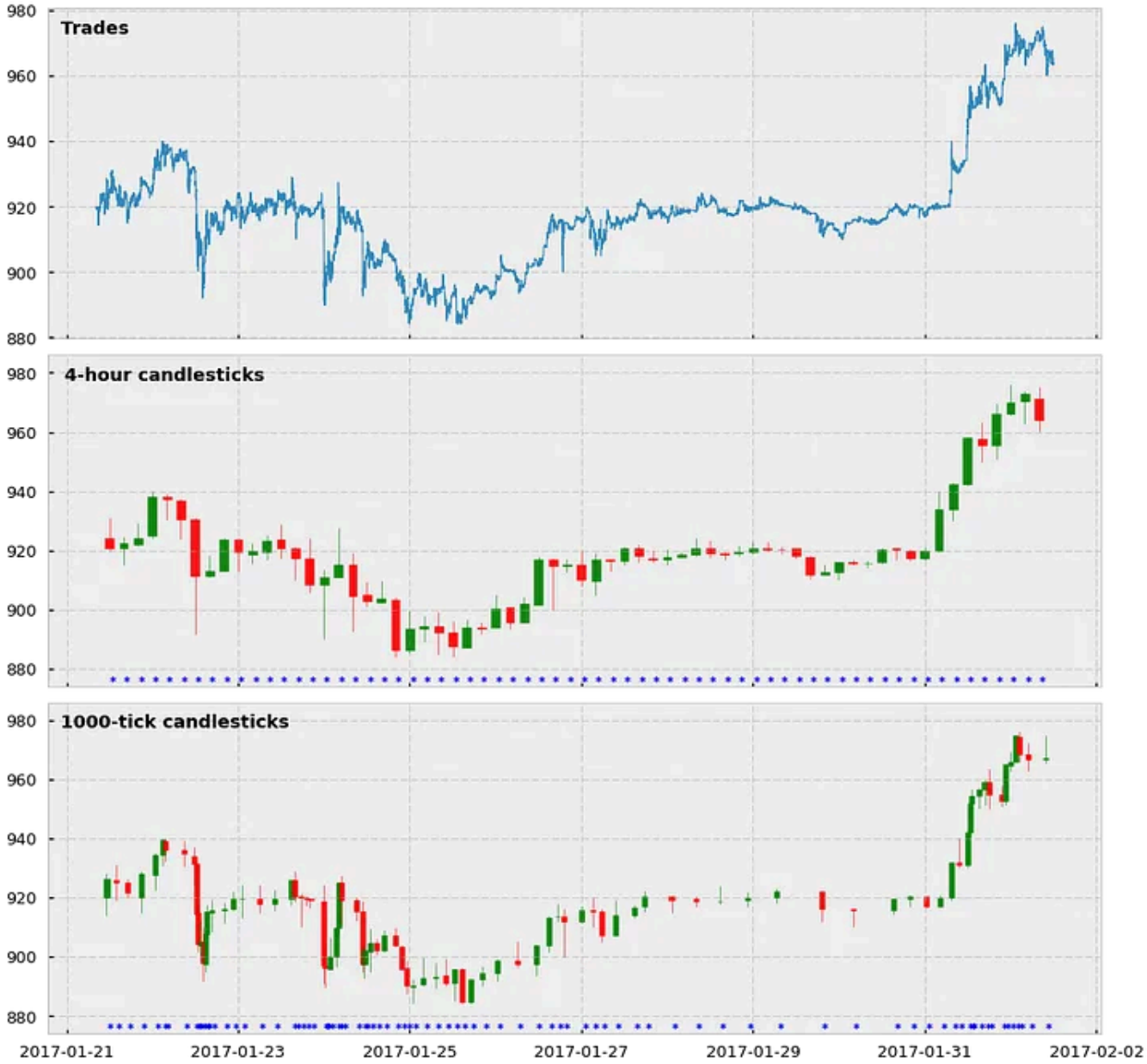
The open and close values correspond to the price of the first and last trade, respectively. The high and the low are the max and min price of all the trades in the candle (may overlap with the open and close). Finally the volume is the sum of all exchanged assets (for instance in the ETH-USD pair, the volume is measured as the number of Ethereums exchanged during the candle). The convention is that when a candle closes at a higher price than its open price we color them green (or keep it empty) while if the close price is lower than the open price we then color them red (or filled with black).

Here's a very simple but fast Python implementation to generate tick candlesticks:

```python
# expects a numpy array with trades
# each trade is composed of: [time, price, quantity]
def generate_tickbars(ticks, frequency=1000):
    times = ticks[:,0]
    prices = ticks[:,1]
    volumes = ticks[:,2]
    res = np.zeros(shape=(len(range(frequency, len(prices), frequency)), 6))
    it = 0
    for i in range(frequency, len(prices), frequency):
        res[it][0] = times[i-1]                         # time
        res[it][1] = prices[i-frequency]                # open
        res[it][2] = np.max(prices[i-frequency:i])      # high
        res[it][3] = np.min(prices[i-frequency:i])      # low
        res[it][4] = prices[i-1]                        # close
        res[it][5] = np.sum(volumes[i-frequency:i])     # volume
        it += 1
    return res
```

tickbar_generator.py hosted with ❤️ by GitHub view raw

And here a visualization of how the tick bars look in comparison to standard time-based candlesticks. In this case we show 4-hour and 1000-tick bars for the BTC-USD trading pair, as well as the price of all trades comprised between 21–01–2017 and 02–20–2017. Notice that for the candlesticks we also show an asterisk every time we sample a bar.



Two main observations about these plots:

1. Yes, tick candlesticks look very ugly. They are chaotic, overlapped and hard to understand but remember they are not supposed to be human-friendly: they are supposed to be machine-friendly.

2. The actual reason why they are ugly is because they are doing very well their job. Look at the asterisks, see how in periods in which the price changes a lot there are more asterisks (and more bars) cluttered together? And the opposite: when the price does not change much, tick bar sampling is much lower. We are essentially creating a system in which we are synchronizing the arrival of information to the market (higher activity and price volatility) with the sampling of candlesticks. We are finally sampling more in periods of high activity and sampling less in periods of low activity. Hooray!

3. — Statistical properties

So what about their statistical properties? Are they any better than their traditional time-based counterparts?

We'll be looking at two different properties: (1) serial correlation and (2) normality of returns for each of the 15 cryptocurrency pairs offered in CryptoDatum.io, including all historical bars of the Bitfinex exchange, and for each of the time-based and tick candle-stick sizes:

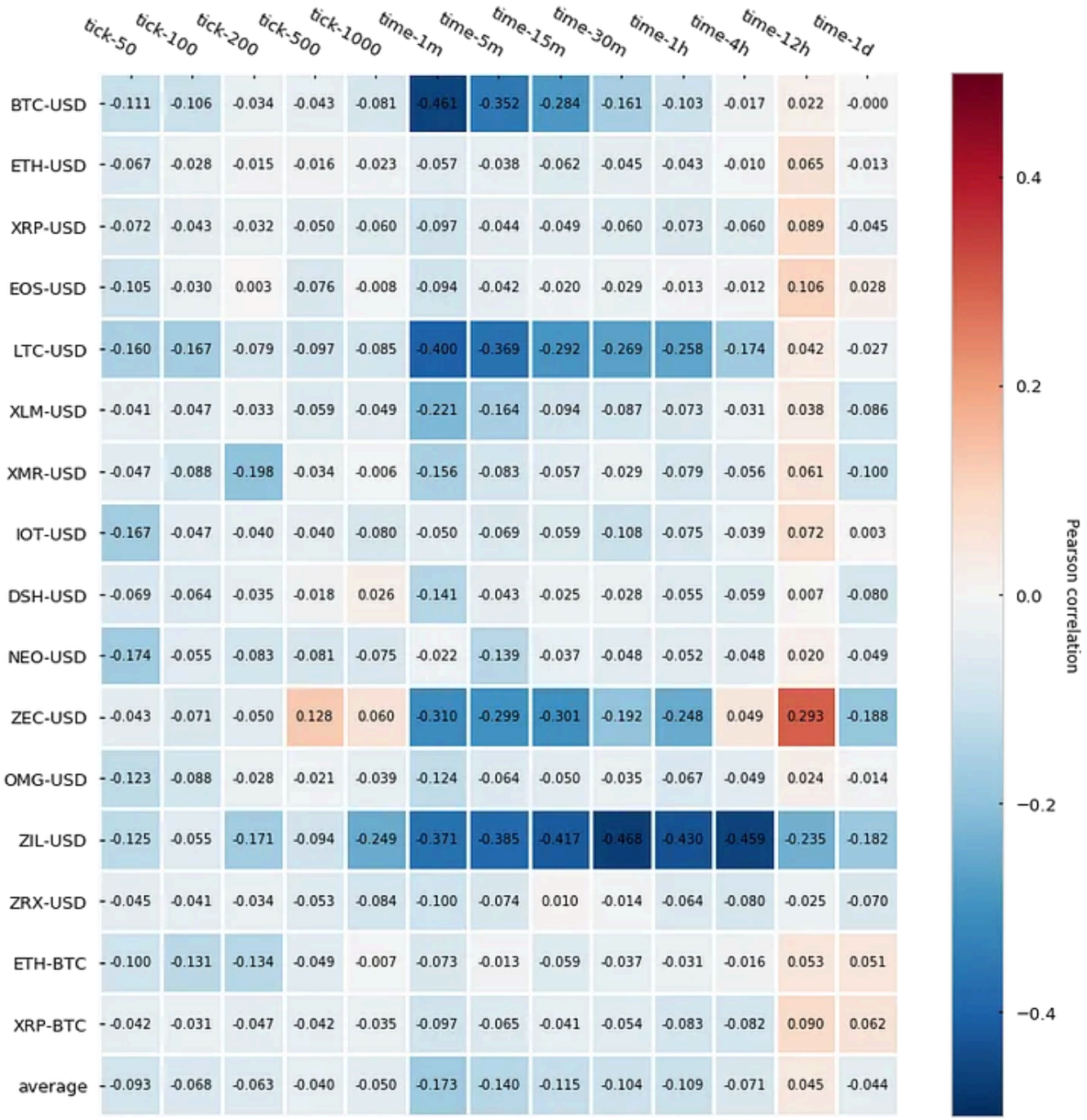
- **Time-based bar sizes**: 1-min, 5-min, 15-min, 30-min, 1-hour, 4-hour, 12-hour, 1-day.
- **Tick bar sizes**: 50, 100, 200, 500, 1000

3.1 — Serial correlation (a.k.a. auto-correlation)

Serial correlation measures how much each value of a time series is correlated with the following one (for lag=1) or between any value i and any other value i+n (lag=n). In our case, we will calculate the serial correlation of the log returns, which are calculated as the first-difference of the log of candles close prices.

Ideally, each data point of our series should be an independent observation. If serial correlation exists it means that they are not independent (they depend on each other at lag=1 or higher) and this will have consequences when building regressive models because the errors we will observe in our regression will be smaller or bigger than the true errors, which will mislead our interpretations and predictions. You can see a very visual explanation of the problem here.

In order to measure the serial correlation we will calculate the Pearson correlation of the series respect the shifted self (lag=1, a.k.a. first-order correlation). Here are the results:



Pearson auto-correlation

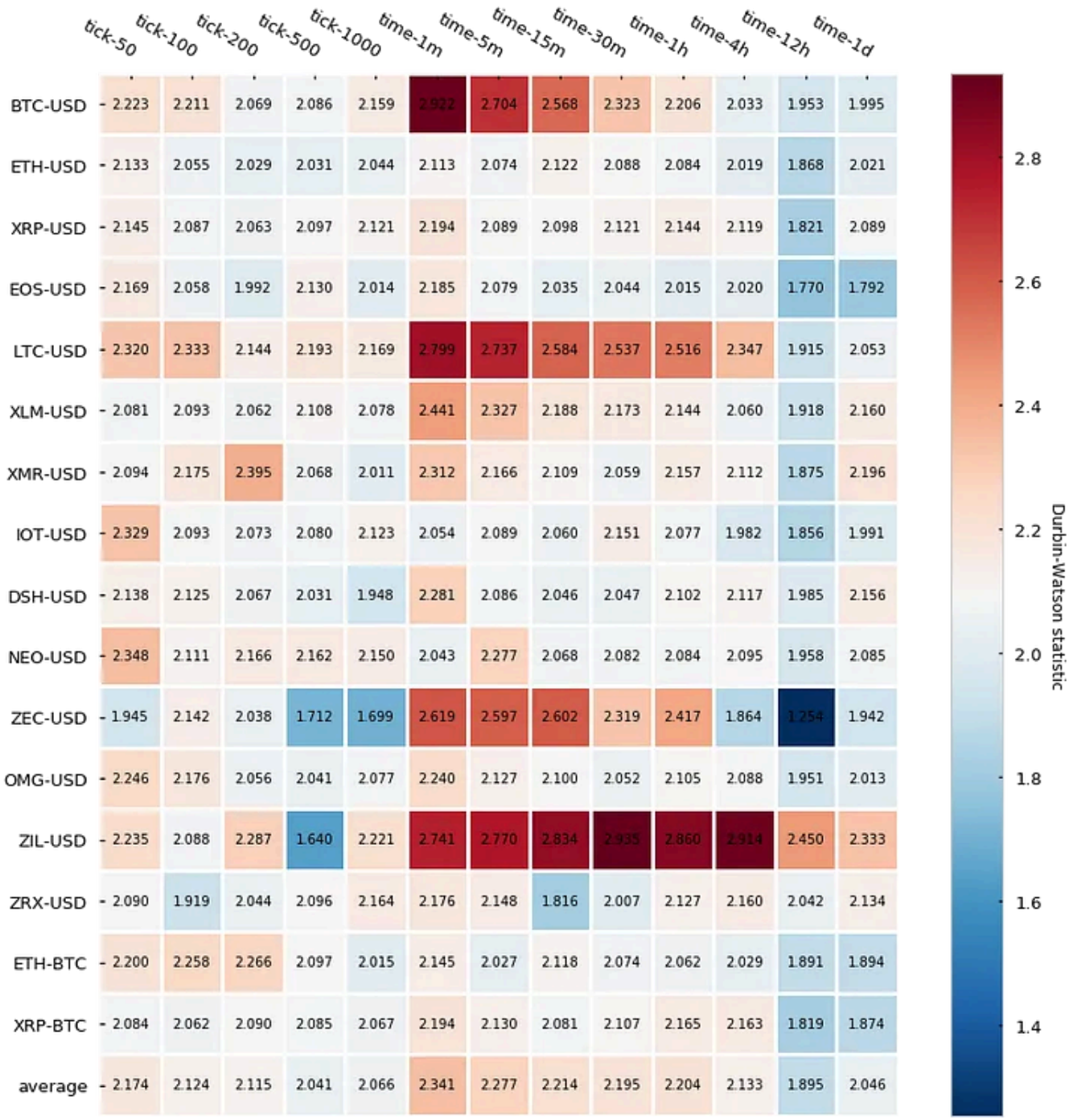
It turns out that tick bars (labelled as tick-*) have generally lower auto-correlation than time-based candlesticks (labelled as time-*) — this is, the Pearson auto-correlation is closer to 0. The difference seems to be less evident for the bigger time bars (4h, 12h, 1d) but interestingly even the smallest tick bars (50-tick and 100-tick) yield a very low auto-correlation, which does not hold true for the smaller time-bars (1-min, 5-min).

Finally, is interesting to see how several cryptocurrencies (BTC, LTC, ZEC and ZIL) express fairly strong negative auto-correlation in few of the time-bars. Roberto Pedace comments here about negative auto-correlations:

> *A*utocorrelation, *also known as* serial correlation, *may exist in a regression model when the order of the observations in the data is relevant or important. In other words, with time-series (and sometimes panel or logitudinal) data, autocorrelation is a concern. […]* No autocorrelation *refers to a situation in which no identifiable relationship exists between the values of the error term. […] Although unlikely, negative autocorrelation is also possible.* Negative autocorrelation *occurs when an error of a given sign tends to be followed by an error of the opposite sign. For instance, positive errors are usually followed by negative errors and negative errors are usually followed by positive errors.*

We will perform an additional statistic test called the Durbin-Watson (DB) test that also diagnoses the existence of serial correlation. The DB statistic lies in the 0–4 range and its interpretation is the following:

Essentially, the closest to 2, the lowest the serial correlation is. Here are the results:



Durbin-Watson test

Results are in line with the Pearson autocorrelation test, which gives strength to the narrative that tick bars display a slightly lower autocorrelation than time-based candlesticks.

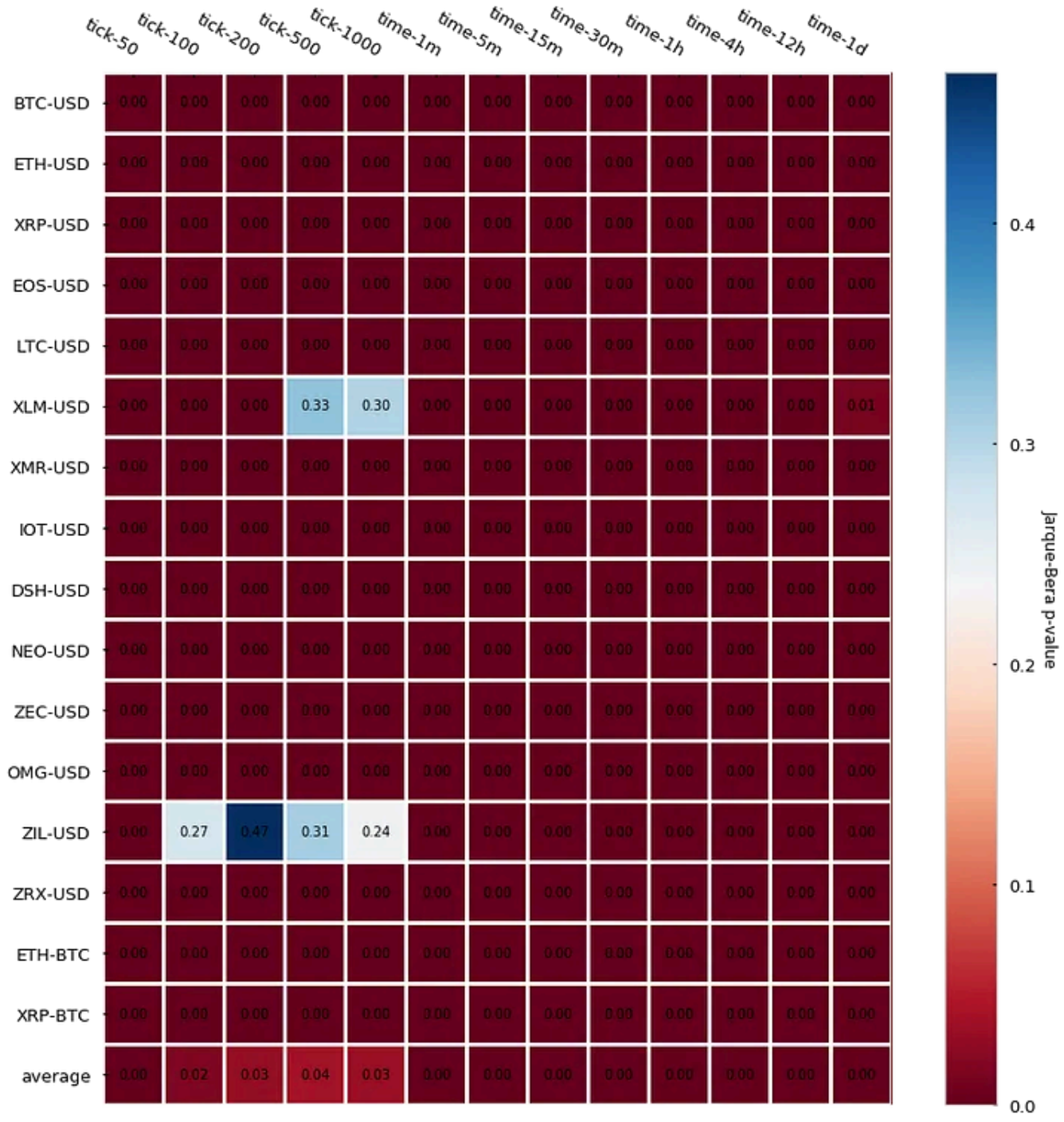
3.2 — Normality of returns

Another statistic we can look at is the normality of returns, this is whether the distribution of our log returns follows or not a normal (a.k.a Gaussian) distribution.

There are several tests we can run to check the normality — we will perform 2 of them: the **Jarque-Bera test**, which tests whether the data has the skewness and kurtosis matching a normal distribution, and the **Shapiro-Wilk test**, which is one of the most classical tests to check if a sample follows a Gaussian distribution.

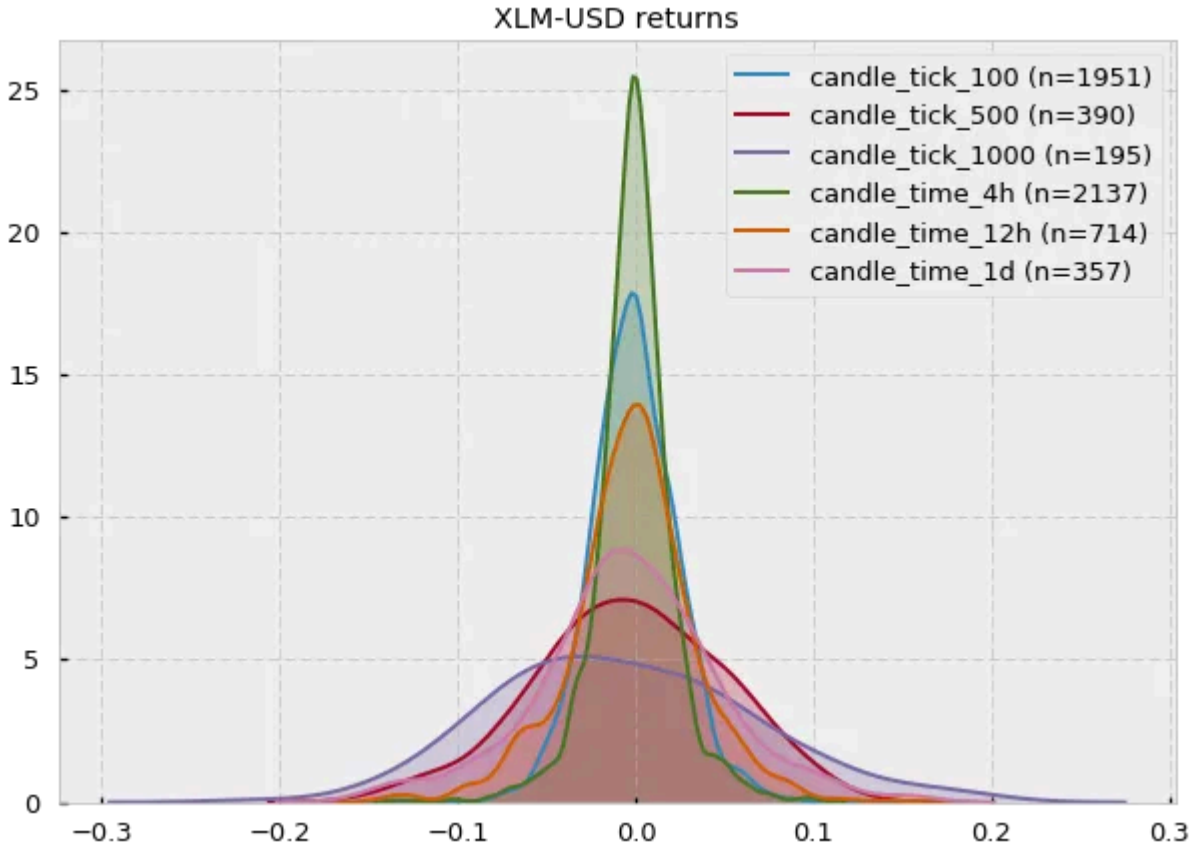
In both cases, the null hypothesis is that the sample follows a normality. If the null hypothesis is rejected (p-value lower than the significance level — usually < 0.05) there is compelling evidence that the sample does not follow a normal distribution.

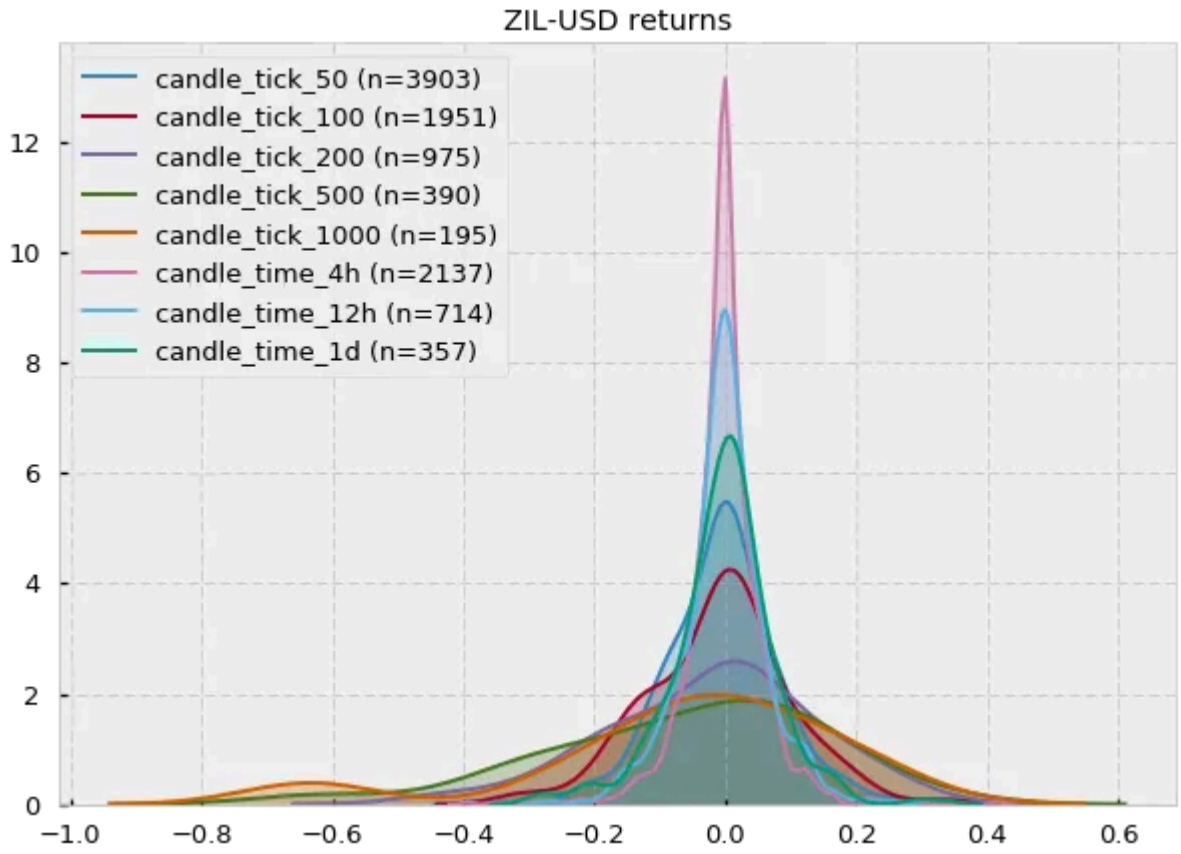
Let's look at the p-values for the Jarque-Bera first:



Jarque-Bera p-value

The results are almost unanimous: log returns do not follow a Gaussian distribution (most p-values < 0.05) . Two cryptocurrency pairs (Stellar and Zilliqa) seem to actually follow a Gaussian if we set our significance level at 0.05. Let's take a look at their distributions (kernel density estimates):



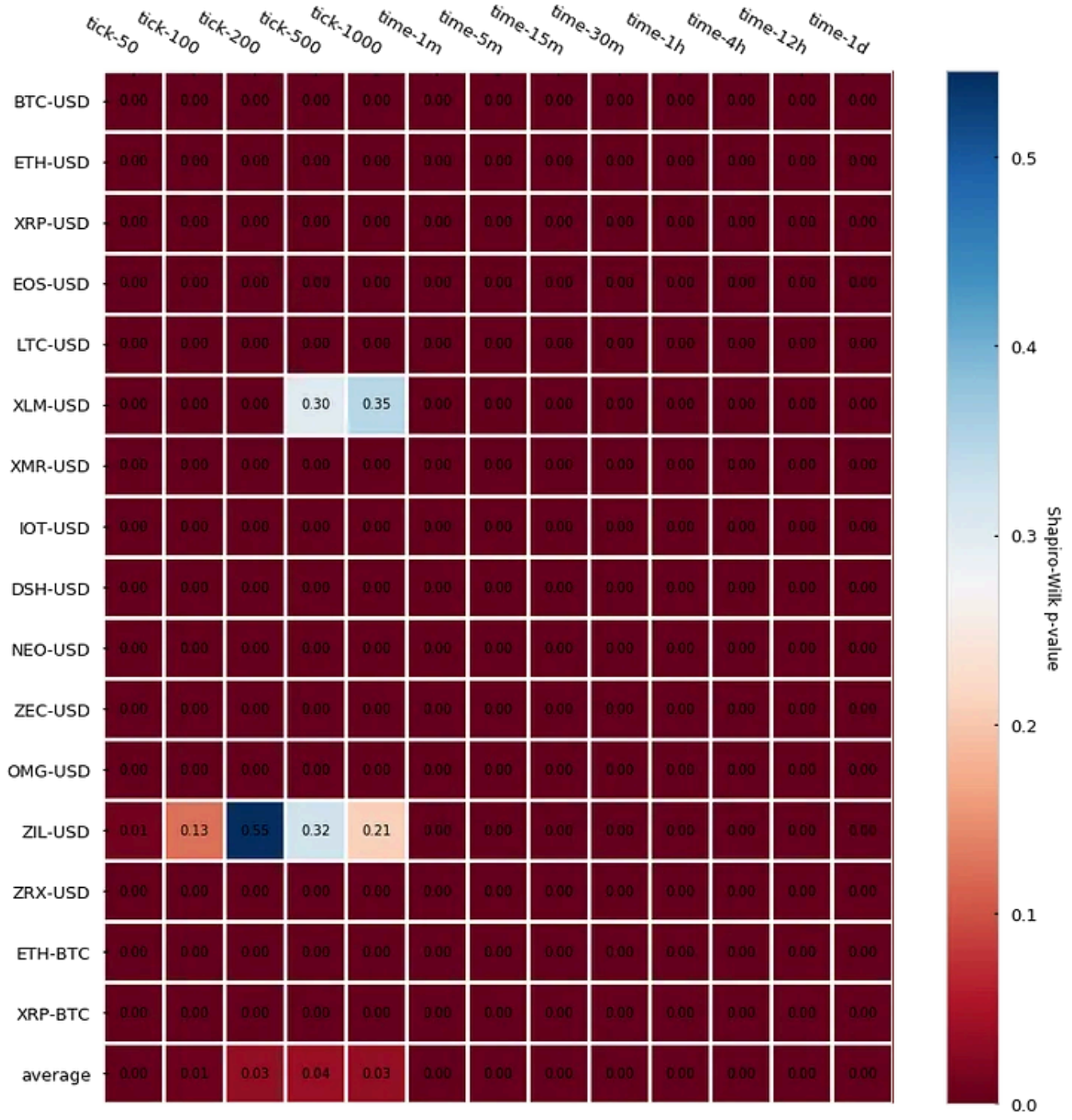


Fair enough, some of them could look Gaussian (at least visually). However, notice that the number of samples (n) is very small (e.g. for XLM-USD candle_tick_1000 n=195) so I suspect that one of the reasons may just be the lack of sampling, which provides Jarque-Bera not enough evidence to reject the null hypothesis of normality.

In fact, a quick look at the CryptoDatum.io database shows that XLM-USD and ZIL-USD trading pairs were just released May and July last year (2018), respectively, and they seem to have quite low volume…

Mystery solved? :)

Let's now run the Shapiro-Wilk test to see if it agrees with the previous results:



Shapiro-Wilk p-value

Damn, Shapiro, didn't they teach you not to copy during a test at school? Non-normality of returns seems to be the rule regardless of bar type.

4. What did we learn?

1. Tick candlesticks are generated by aggregating a predefined number of ticks and calculating the associated OHLCV values.

2. Tick bars look ugly in a chart but they do their job well: they sample more during high activity periods and sample less during low activity periods.

3. Log returns from tick candlesticks display lower serial correlation when compared to time-based candlesticks, even at small sizes (50, 100-tick bars).

4. Log returns from both tick and time-based bars do not follow a normal distribution.

Thanks for reading. In the following weeks we will be talking about other type of advanced bars such as volume bars, dollar bars and (my favorite) imbalance bars. So, if you liked this article: stay tuned!

This project is part of our research at CryptoDatum.io, a cryptocurrency data API that aims to provide plug-and-play datasets to train machine learning algorithms. If you liked the data we showed in this article, get your free API key and play with it yourself at https://cryptodatum.io

Cryptocurrency | Towards Data Science | Python | Finance | Machine Learning

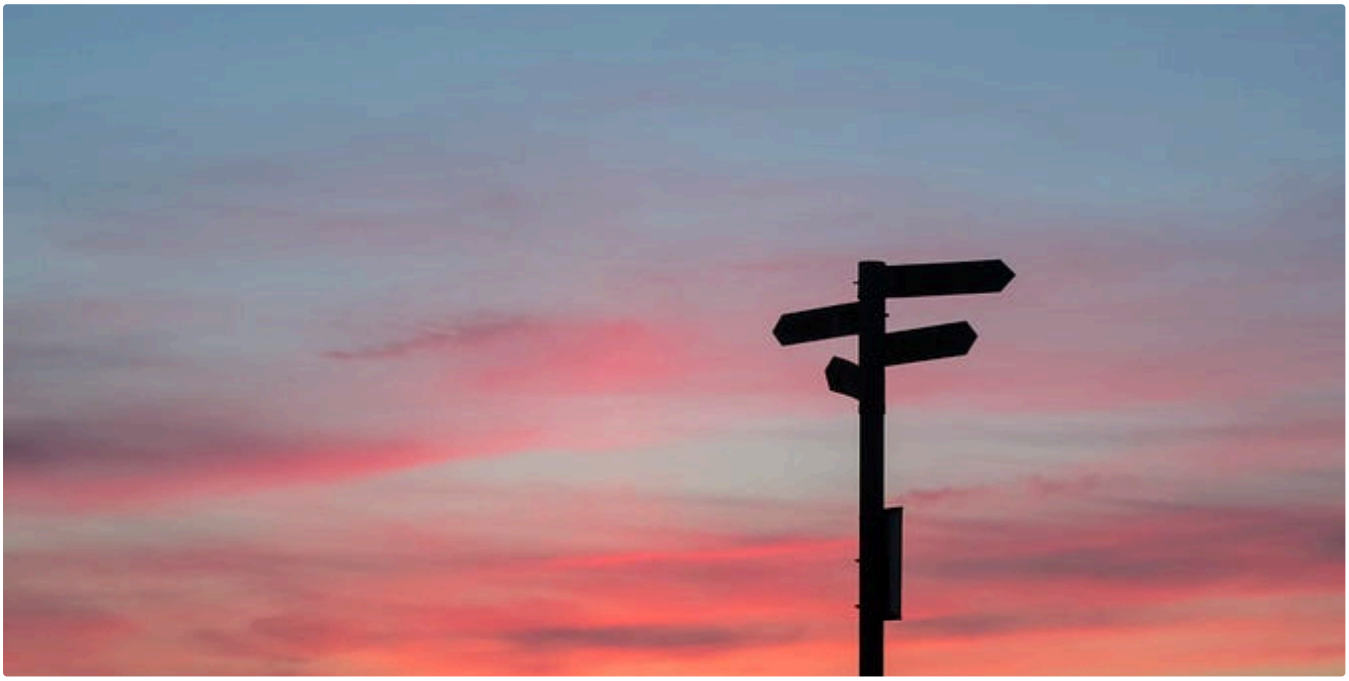
Follow

Written by Gerard Martínez

1.4K Followers · Writer for Towards Data Science

Trading strategy developer — Founder of CryptoDatum.io

More from Gerard Martínez and Towards Data Science



Gerard Martínez in Towards Data Science

Information-driven bars for financial machine learning: imbalance bars

In previous articles we talked about tick bars, volume bars and dollar bars, alternative types of bars which allow market...

8 min read · May 20, 2019

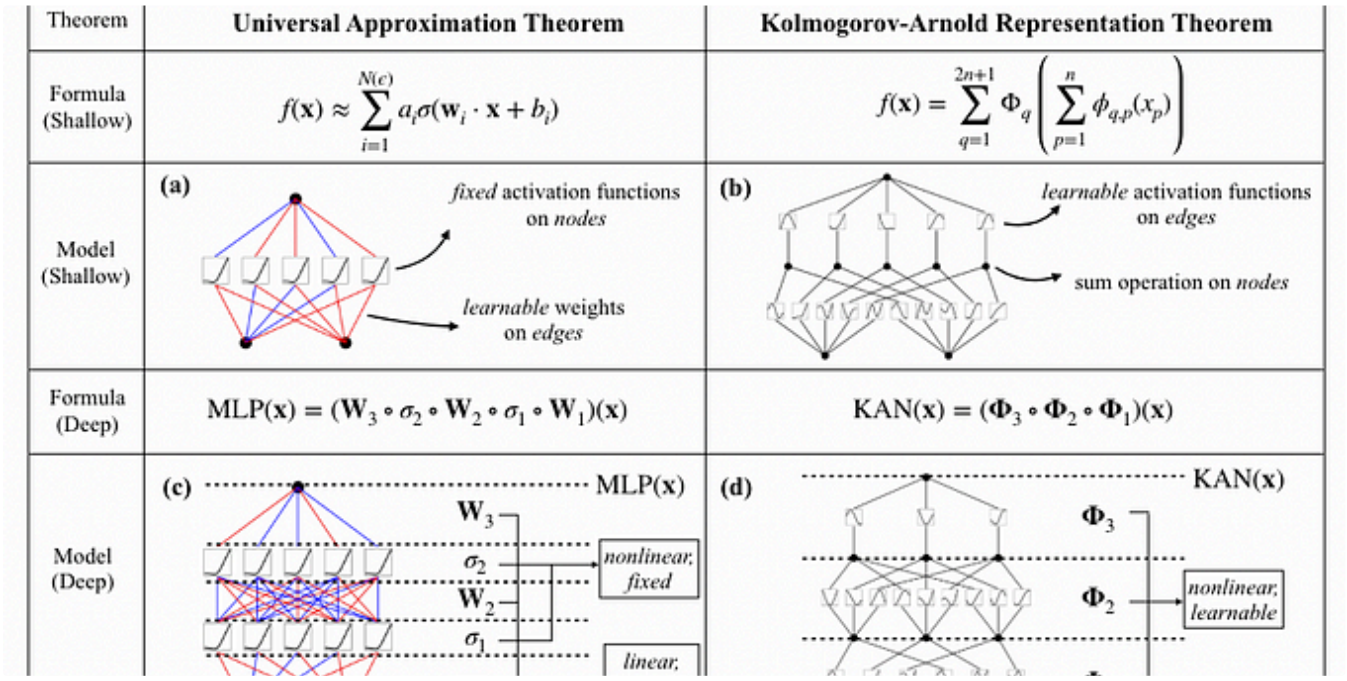
372 4

Torsten Walbaum in Towards Data Science

What 10 Years at Uber, Meta and Startups Taught Me About Data Analytics

Advice for Data Scientists and Managers

9 min read · May 30, 2024



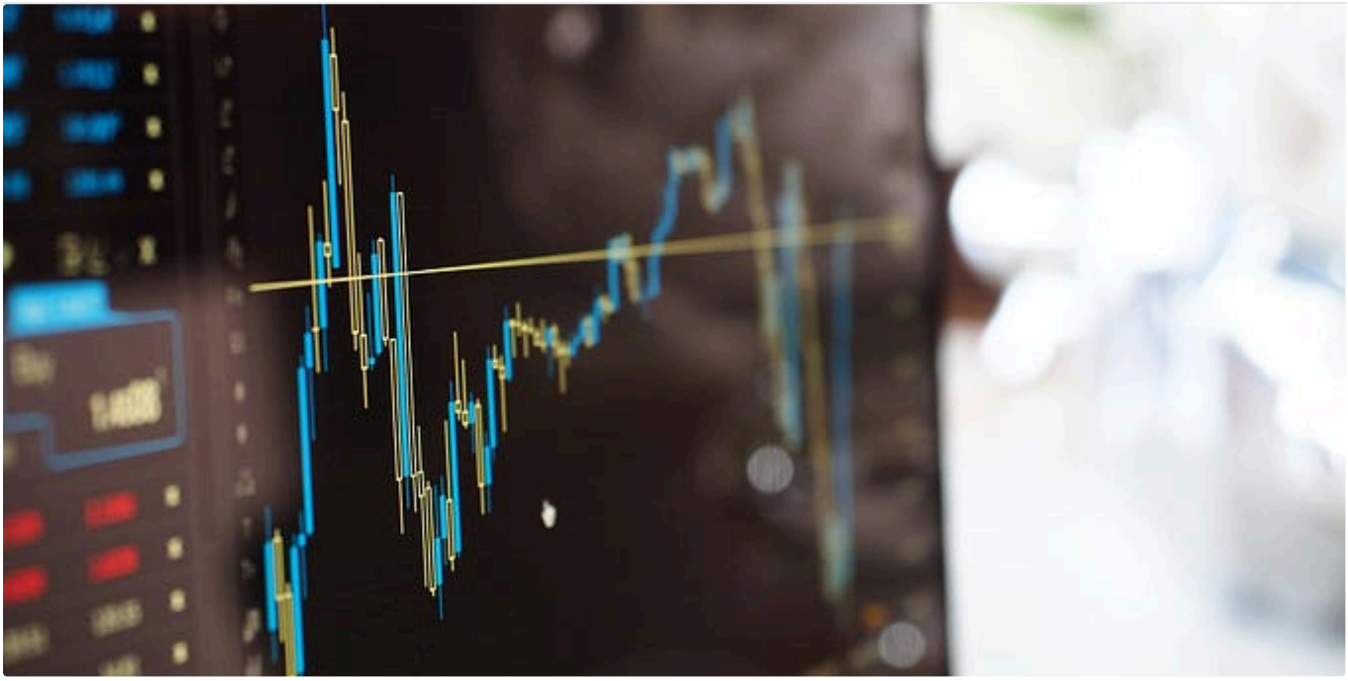
Theo Wolf in Towards Data Science

Kolmogorov-Arnold Networks: the latest advance in Neural Networks, simply explained

The new type of network that is making waves in the ML world.

✨ · 9 min read · May 12, 2024

2.1K 18



Gerard Martínez in Towards Data Science

Autoencoders for the compression of stock market data

A Pythonic exploration of diverse neural-network autoencoders to reduce the dimensionality of Bitcoin price time series

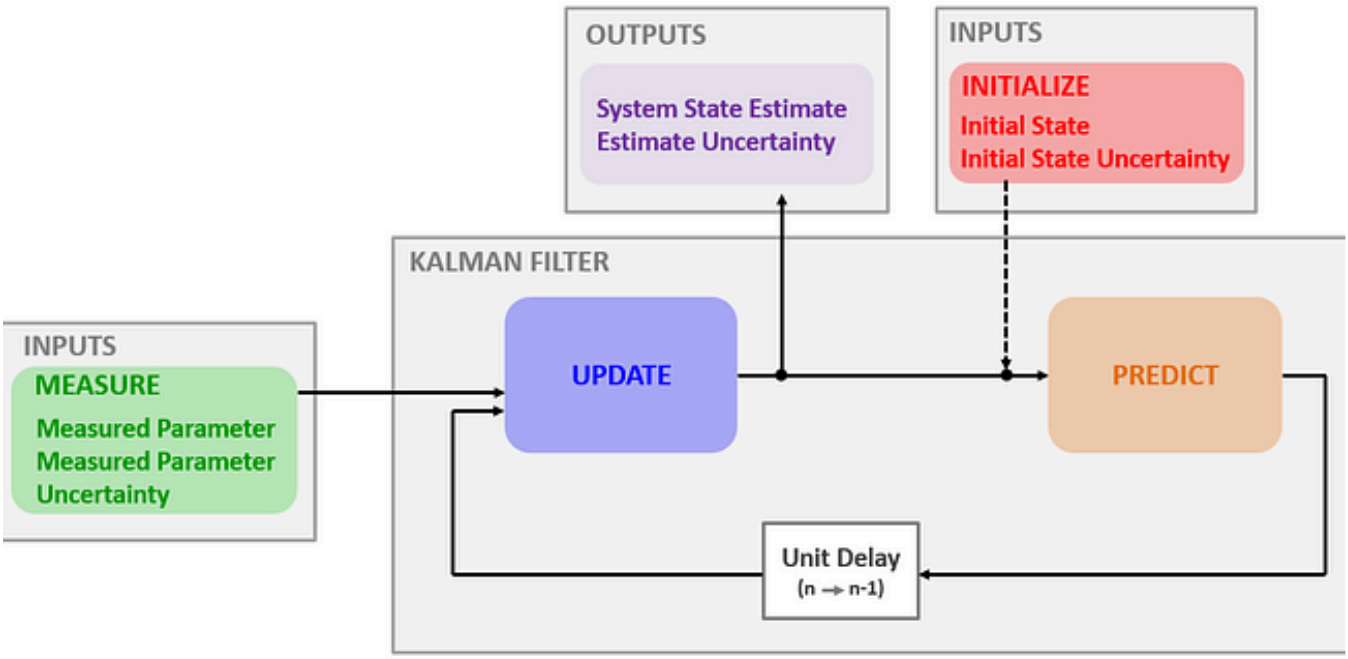
8 min read · Jan 18, 2019

473 6

See all from Gerard Martínez

See all from Towards Data Science

Recommended from Medium



Sausan

The Magic of Kalman Filters: How They Transform Data into Accurate Predictions

In the world of data analysis and signal processing, Kalman filters are nothing short of magical. These mathematical marvels can sift...

4 min read · Jun 15, 2024

Fisher Lok in InsiderFinance Wire

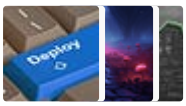
Trading from First Principle 3: Building a Machine Learning-Based Trading Strategy

This article proposed a standardized approach to produce a trading strategy based on machine learning models. A systematic approach is...

13 min read · Feb 13, 2024

177

Lists

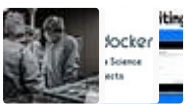


Predictive Modeling w/ Python

20 stories · 1316 saves

Practical Guides to Machine Learning

10 stories · 1580 saves



Coding & Development

11 stories · 664 saves

Natural Language Processing

1537 stories · 1071 saves

PulsePointFX

Relative Strength Index Strategy using Python 🐍

Learn how to create a Relative Strength Index (RSI) trading strategy using Python programming language

6 min read · Jan 10, 2024

33

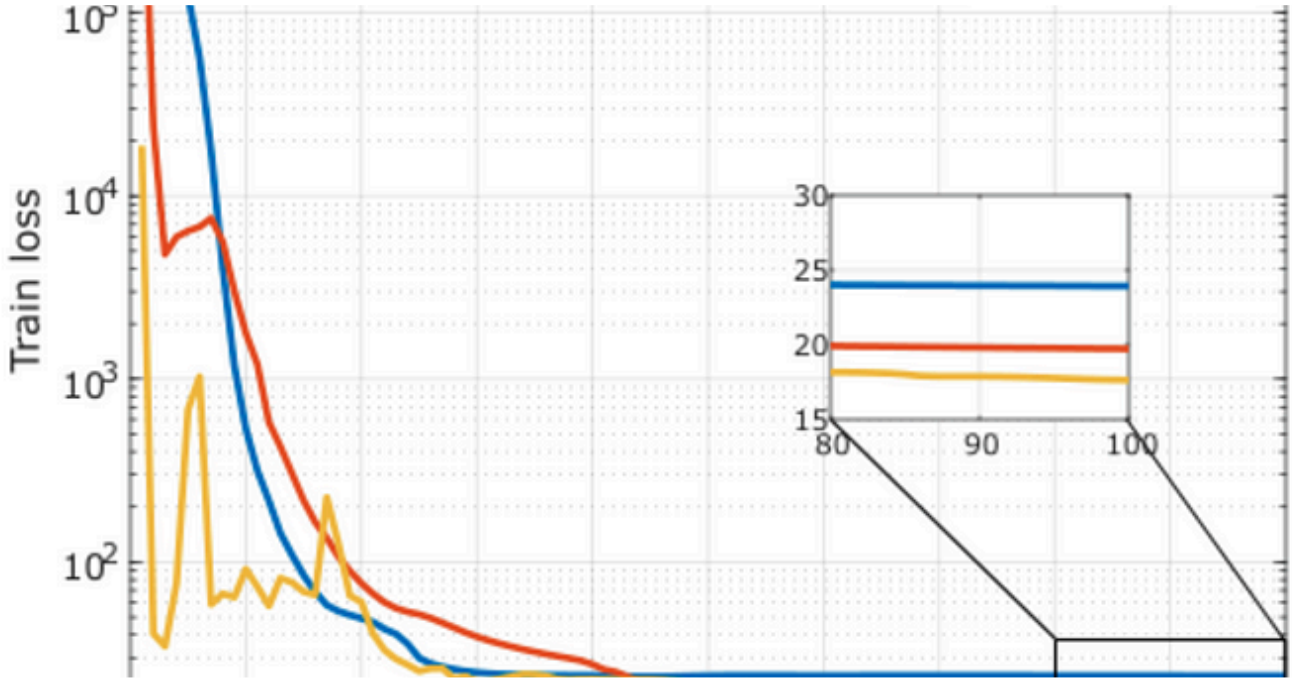
Cristian Velasquez

Riding the Waves of Stock Prices with Wavelet Transform Signals in Python

Towards Unlocking Market Signals for Clearer Trading Insights

✦ · 9 min read · Sep 11, 2023

437 3



Sercan Bugra Gultekin

(2) Stock Price Prediction with ML in Python: GRU (Gated Recurrent Unit) Model

In our first series, I made a forecast with the LSTM model, now we will change our model and try the same experiment with GRU (Gated...

✦ · 3 min read · May 31, 2024

1

 Adam in Call For Atlas

Unsupervised Learning as Signals for Pairs Trading and StatArb

In this article, we will leverage clustering algorithms to detect pairs in a universe of tradable securities. We will train three...

23 min read · Mar 6, 2024

202 2