Gathering Price Data Using Binance API

By

Jampala, Sai Krishna Anudeep

Goal

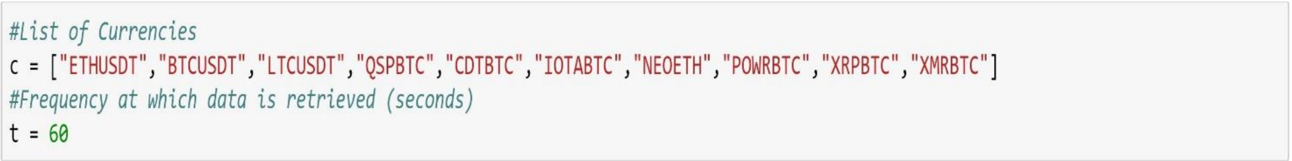
Using Binance API, we gather last traded price data for given 10 currencies and save them as a CSV files. We can choose the frequencies at which the data should be gathered and for which currencies we are gathering the data.

Note

- The code is ".ipynb file" (Jupyter notebook), it should be run on a Python 2.7 environment with Binance- Python package installed.
- Create a folder ticker_dump to store the output CSV files.
- Enter your API credentials in the beginning.
- I'm attaching the html format of the code, Jupyter notebook, ticker_dump folder and report.

Process

- We need a Binance account from which we can get access key and access secret, using these we connect to Binance API.
- We choose the currencies and the frequency at which data is gathered. We are creating "c", a list of currencies we need and "t" (time in seconds) frequency at which we gather data. We can add or remove elements from "c" and we can change "t" as per our requirement.

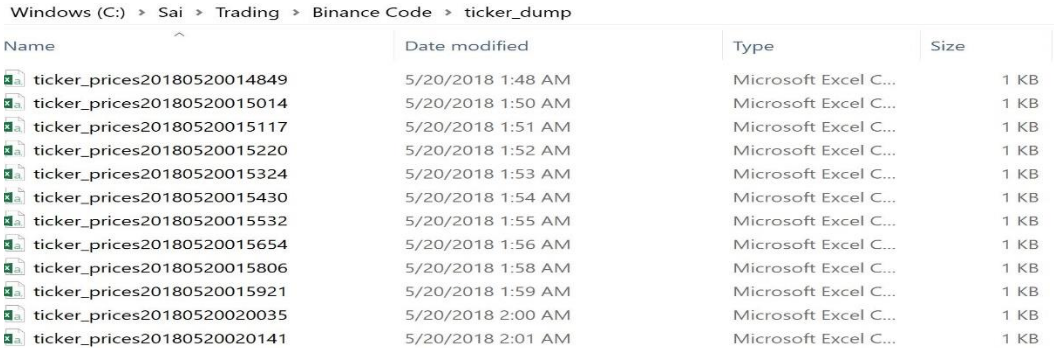


```
#List of Currencies
c = ["ETHUSDT","BTCUSDT","LTCUSDT","QSPBTC","CDTBTC","IOTABTC","NEOETH","POWRBTC","XRPBTC","XMRBTC"]
#Frequency at which data is retrieved (seconds)
t = 60
```

- Next we create a function "get_price_of_currency" to get price data from Binance API. It returns a Pandas series with Symbol, Price, time stamp. It uses Binace API function "get_symbol_ticker" to get the latest price details of the given currency.

- In case we use the incorrect symbol, it would return N/A and asks us to check the specific incorrect symbol.
- After specifying all the requirements, we retrieve data in CSV files in the folder called “ticker_dump”. For each time period, We create a new CSV file with symbols, prices and time stamp.

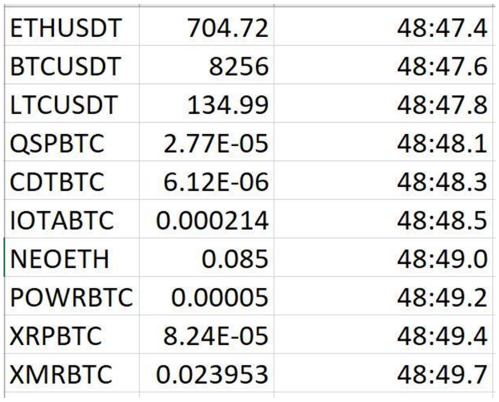
Ticker_dump folder:



Windows (C:) > Sai > Trading > Binance Code > ticker_dump

Name	Date modified	Type	Size
ticker_prices20180520014849	5/20/2018 1:48 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015014	5/20/2018 1:50 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015117	5/20/2018 1:51 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015220	5/20/2018 1:52 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015324	5/20/2018 1:53 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015430	5/20/2018 1:54 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015532	5/20/2018 1:55 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015654	5/20/2018 1:56 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015806	5/20/2018 1:58 AM	Microsoft Excel C...	1 KB
ticker_prices20180520015921	5/20/2018 1:59 AM	Microsoft Excel C...	1 KB
ticker_prices20180520020035	5/20/2018 2:00 AM	Microsoft Excel C...	1 KB
ticker_prices20180520020141	5/20/2018 2:01 AM	Microsoft Excel C...	1 KB

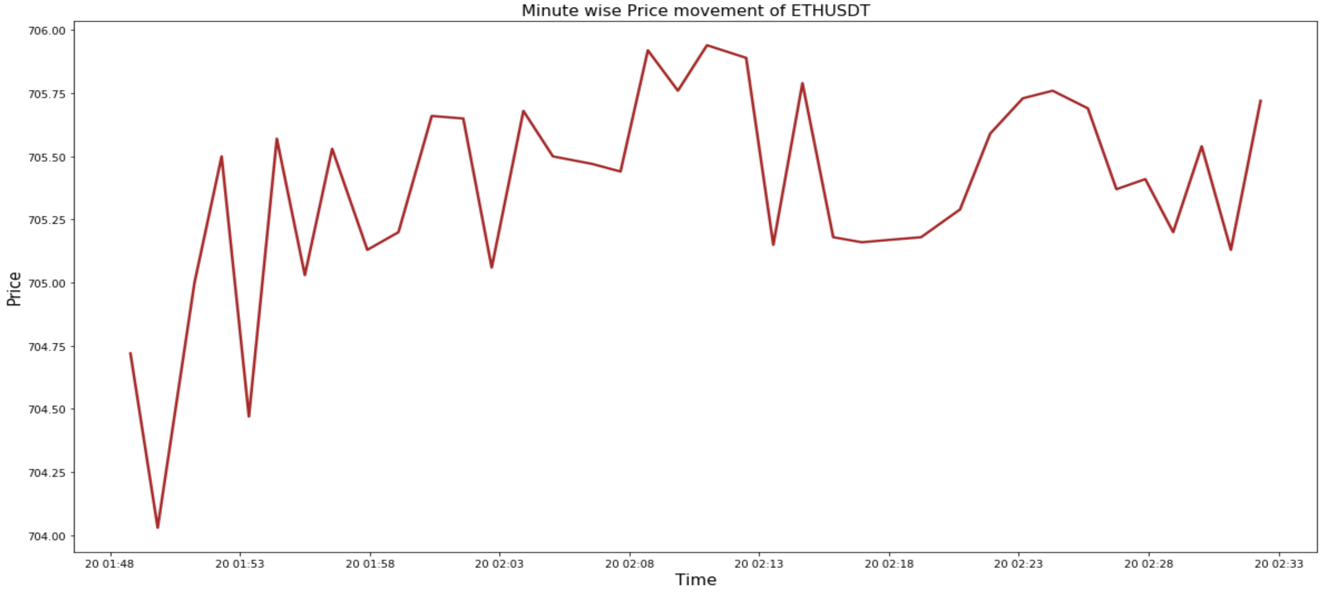
View of each csv file:



ETHUSDT	704.72	48:47.4
BTCUSDT	8256	48:47.6
LTCUSDT	134.99	48:47.8
QSPBTC	2.77E-05	48:48.1
CDTBTC	6.12E-06	48:48.3
IOTABTC	0.000214	48:48.5
NEOETH	0.085	48:49.0
POWRBTC	0.00005	48:49.2
XRPBTC	8.24E-05	48:49.4
XMRBTC	0.023953	48:49.7

- The code runs on an infinite loop. So, when we gathered the required amount of data, we must stop the code.
- Once we gathered the required data, we concatenate all the individual files in to a single file.
- We performed analysis on Ethereum. Here we plotted the minute wise prices of ETHUSDT (Y-axis price vs X- axis time).

- Minute wise Price movement of ETHUSDT



-- *The end* --