

# herron-lookups-2022-03-24

March 24, 2022

## 1 Introduction

Occasionally, I do not know the answers to questions from the class. Instead of looking up answers and sharing them during class the next week, I will share these answers in a Jupyter Notebook on Canvas. This new approach has several advantages:

1. You do not need to wait a week to get these answers.
2. I am less likely to forget to share these answers.
3. All three sections can benefit from these questions.

Below are the answers to these questions from class this week.

```
[1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
[2]: import yfinance as yf
import requests_cache
session = requests_cache.CachedSession()
```

```
[3]: plt.rcParams['figure.dpi'] = 150
np.set_printoptions(precision=4, suppress=True)
pd.options.display.float_format = '{:,.4f}'.format
```

## 2 Monday - Section 03

### 2.1 Is there a `.lagged()` method we can use instead of `.shift()`?

I use `.shift()` with positive values to get *lags* and negative values to get *leads* or *forward* values. There is a DataCamp exercise in the time series course that generates a column named `lagged`, but I think that the `.shift()` method is the best way to lead and lag values.

```
[4]: df = yf.download(tickers='TSLA', session=session)
```

```
[*****100%*****] 1 of 1 completed
```

```
[5]: df['1-Day Lag of Adj Close'] = df['Adj Close'].shift(1)
```

```
[6]: df['1-Day Forward Adj Close'] = df['Adj Close'].shift(-1)
```

```
[7]: df.filter(regex='Adj Close')
```

```
[7]:
```

	Adj Close	1-Day Lag of Adj Close	1-Day Forward Adj Close
Date			
2010-06-29	4.7780	NaN	4.7660
2010-06-30	4.7660	4.7780	4.3920
2010-07-01	4.3920	4.7660	3.8400
2010-07-02	3.8400	4.3920	3.2220
2010-07-06	3.2220	3.8400	3.1600
...	...	...	...
2022-03-18	905.3900	871.6000	921.1600
2022-03-21	921.1600	905.3900	993.9800
2022-03-22	993.9800	921.1600	999.1100
2022-03-23	999.1100	993.9800	1013.9200
2022-03-24	1013.9200	999.1100	NaN

```
[2956 rows x 3 columns]
```

## 2.2 Where can we learn more about unit testing?

I asked a friend that works in data analytics at a hedge fund: “what should I teach students?” His response: “unit testing.” What is unit testing?

In computer programming, unit testing is a software testing method by which individual units of source code—sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures—are tested to determine whether they are fit for use.

He is correct that unit tests are important. However, he forgets that unit tests are an advanced skill and unnecessary until you know how to write code.

I typically write tests interactively with NumPy functions, like `numpy.allclose()`, which tests the similarity of two objects. When I write persistent tests, I typically use `numpy.testing.assert_close()`. These two functions are similar, but the former returns `False` if the test condition is not met, while the later raises an error.

If there is enough interest, we can discuss unit tests during class, but I consider them an advanced skill. At an interview, I think questions about converting daily returns to annualized Sharpe ratios and alphas are more likely than questions about unit tests. If you want to learn more about unit tests, this [StackOverflow answer](#) is a good start.

## 2.3 Can I use Python to dollar cost average (DCA) with Coinbase?

In most cases, the answer to, “can I do X with Python?” is “yes!” A student asked how to use this [GitHub repository](#) to DCA with Coinbase, a popular cryptocurrency exchange. Here is a simple example.

1. Open a Coinbase account at [coinbase.com](https://coinbase.com)
2. Follow the instructions in the repository above to create your advanced programming interface (API) key; *I suggest using the sandbox and not trading real assets until you*

*are comfortable with cryptocurrencies, Coinbase, Python, and the code in the repository above*

3. Download `config.json` and `main.py` from the repository above to the same folder as this notebook
4. Follow the instructions in the repository above to modify `config.json`
5. Add funds to your sandbox portfolio via Coinbase's website; the name of my sandbox portfolio is "Default Portfolio"
6. Run `main.py` by executing `%run main.py` in the cell below; you may need to `%pip install cbpro schedule`
7. You can also run `main.py` in a terminal window by executing `python main.py`
8. This code cell will run until your interrupt it

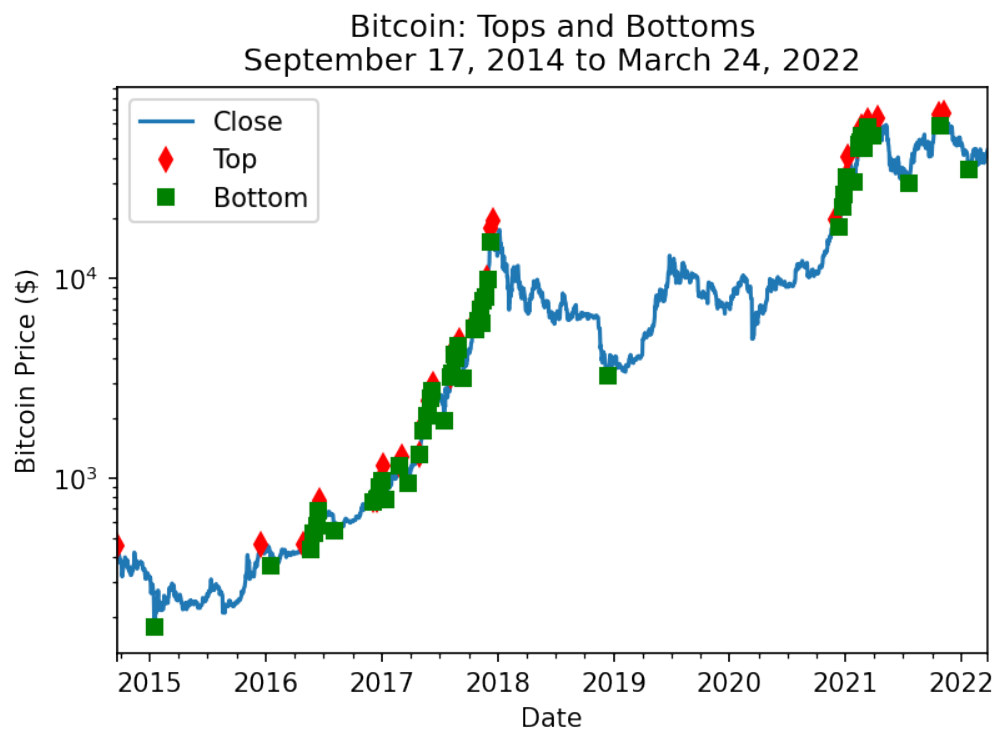
```
[8]: # %run main.py
```

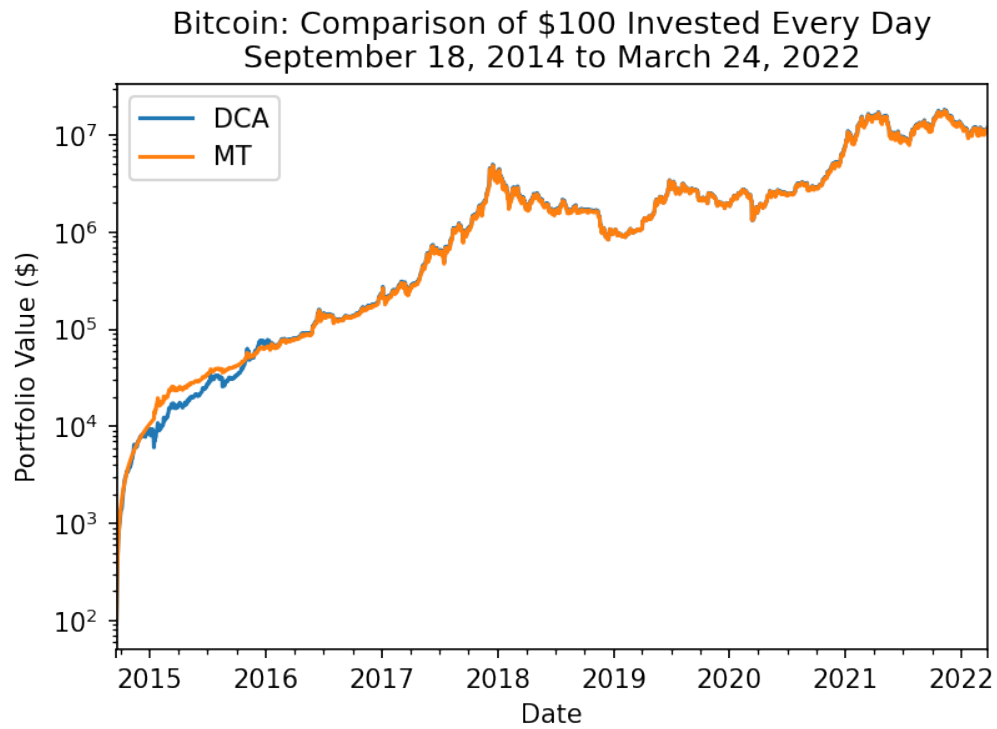
This script is a work in progress, but it will get the job done! Also, you can suggest improvements to the script and ask the author to incorporate them! We will not discuss this further. However, if there is enough interest, I would add it to future versions of the course (or add a standalone course).

### 3 Wednesday - Section 04

#### 3.1 Does market timing work in ~~emerging markets~~ Bitcoin?

No. Although Bitcoin's volatility is high, it is not high enough to help market timing outperform dollar-cost averaging.





## 4 Wednesday - Section 02

No lookups.