

TIME SERIES ANALYSIS OF AAPL STOCK

Leon Zhu

UCLA Stats 418 – Tools in Data Science

Spring 2019



Attaining the dataset – Web Scrap

The screenshot shows the Yahoo Finance website for Apple Inc. (AAPL) with the 'Historical Data' tab selected. The time period is set to 'Jan 01, 2018 - Dec 31, 2018' and the frequency is 'Daily'. The current price is \$208.48, down 3.27 (-1.54%). The browser's developer tools are open, showing the HTML structure of the historical data table. The table has a class of 'W(100%) M(0)' and a data-test of 'historical-prices'. The HTML structure is as follows:

```
<table class="W(100%) M(0)" data-test="historical-prices">
  <thead>...</thead>
  <tbody>
    <tr class="BdT Bdc($c-fuji-grey-c) Ta(end) Fz(s) Whs(nw)">
      <td class="Py(10px) Ta(start) Pend(10px)">
        <span>Dec 31, 2018</span>
      </td>
      <td class="Py(10px) Pstart(10px)">
        <span>158.53</span> == $0
      </td>
      <td class="Py(10px) Pstart(10px)">
        <span>159.36</span>
      </td>
      <td class="Py(10px) Pstart(10px)">
        <span>156.48</span>
      </td>
    </tr>
  </tbody>
</table>
```

The table data is as follows:

Date	Open	High	Low	Close*	Adj Close**	Volume
Dec 31, 2018	158.53	159.36	156.48	157.74	157.07	35,003,500
Dec 28, 2018	157.50	158.52	154.55	156.23	155.56	42,291,400

```
def web_scrap_data(url):
    web_raw = requests.get(url).text

    web_soup = BeautifulSoup(web_raw, 'html.parser')

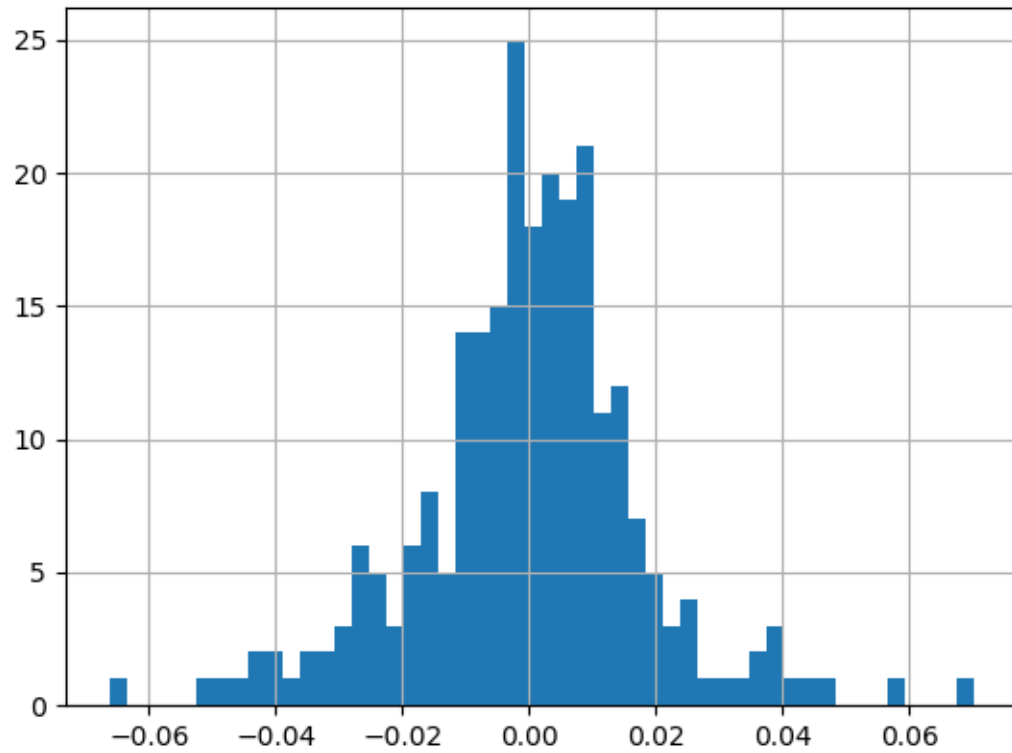
    web_tables = web_soup.find_all('table')
    web_trs = web_tables[0].find_all('tr')

    # Represents each row of the table
    cleaned_data = []
    # List to temporary hold each column index in a
    # So they can be appended to a proper row when f
    temp = []
    # Loop to go through every row in table
    # HTML only loads up to 102 even though there are
    for row in range(1, len(web_trs)):
        temp = [] # Clear the temp row after each i
        web_tds = web_trs[row].find_all('td')

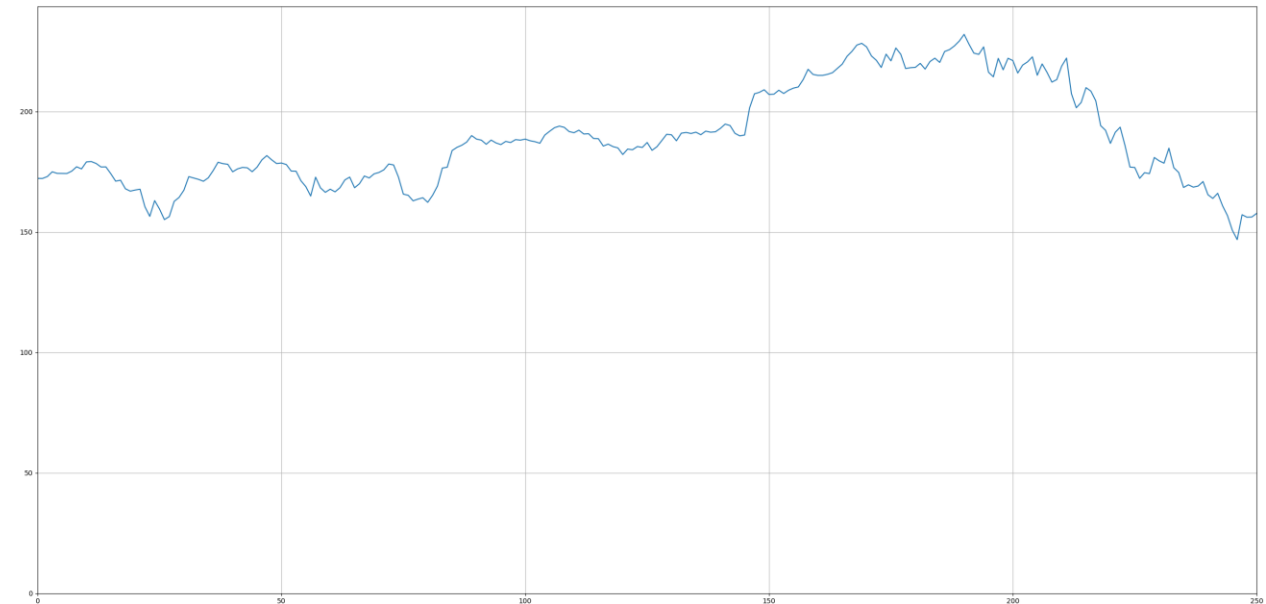
        # Because dividend is displayed as an entire
        if len(web_tds) == 7:
            # should go from 0 to 6 (7 columns)
            for row2 in range(0, len(web_tds)):
                temp.append(web_tds[row2].text)
            cleaned_data.append(temp)

    data df = pd.DataFrame(cleaned_data)
```

Exploratory Analysis



Histogram of the daily percent change



Proposal

- ◆ Conduct Time series analysis on Apple stock price using the daily closing price
- ◆ Product: Given input historical stock price data, produce a forecast of the short term price with a 95% confidence interval