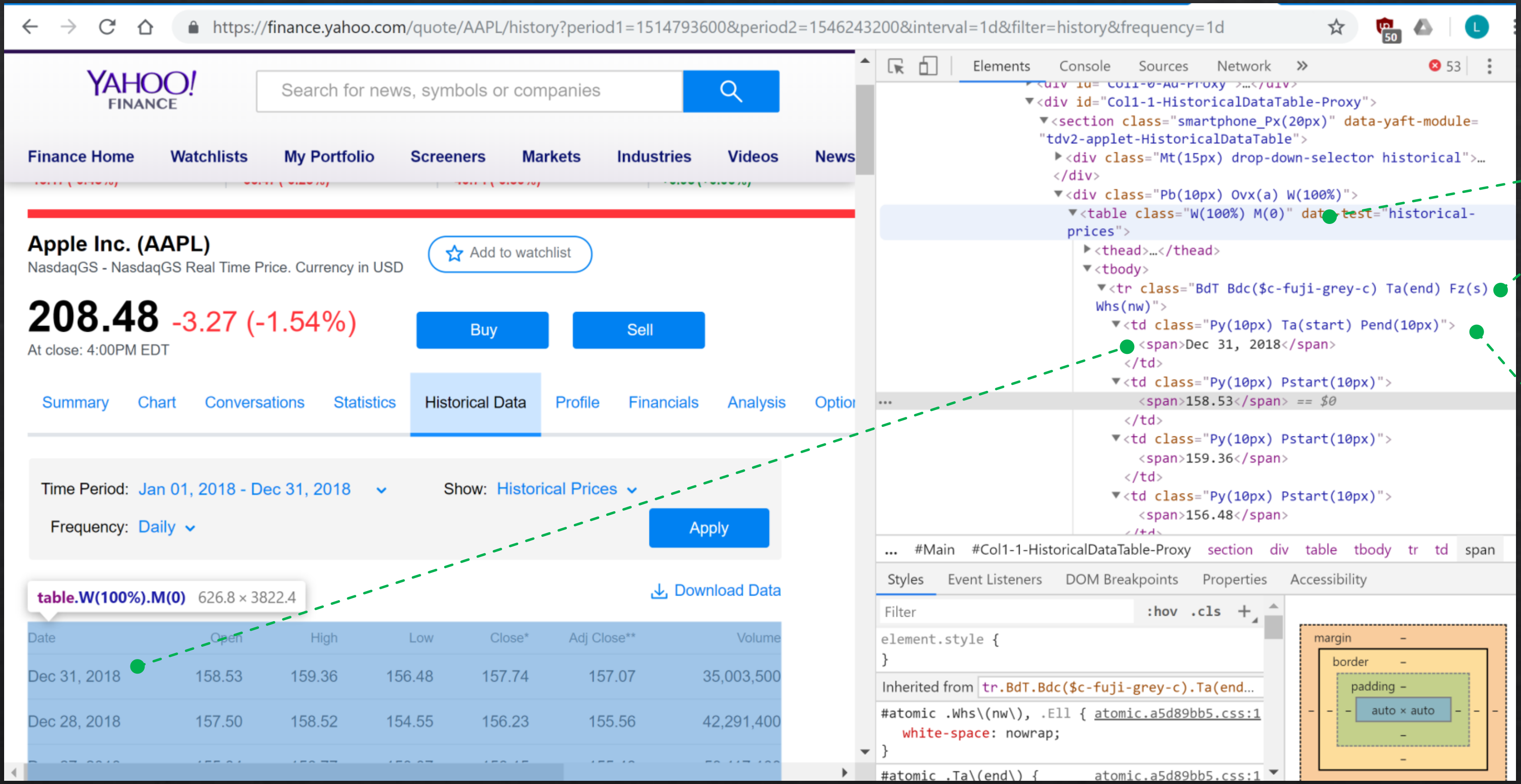
TIME SERIES ANALYSIS OF AAPL STOCK

Leon Zhu

UCLA Stats 418 – Tools in Data Science

Spring 2019

Attaining the dataset – Web Scrap


https://finance.yahoo.com/quote/AAPL/history?period1=1514793600&period2=1546243200&interval=1d&filter=history&frequency=1d

Apple Inc. (AAPL)
NasdaqGS - NasdaqGS Real Time Price. Currency in USD

208.48 -3.27 (-1.54%)
At close: 4:00PM EDT

Time Period: Jan 01, 2018 - Dec 31, 2018 Show: Historical Prices
Frequency: Daily

table.W(100%).M(0) 626.8 × 3822.4

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

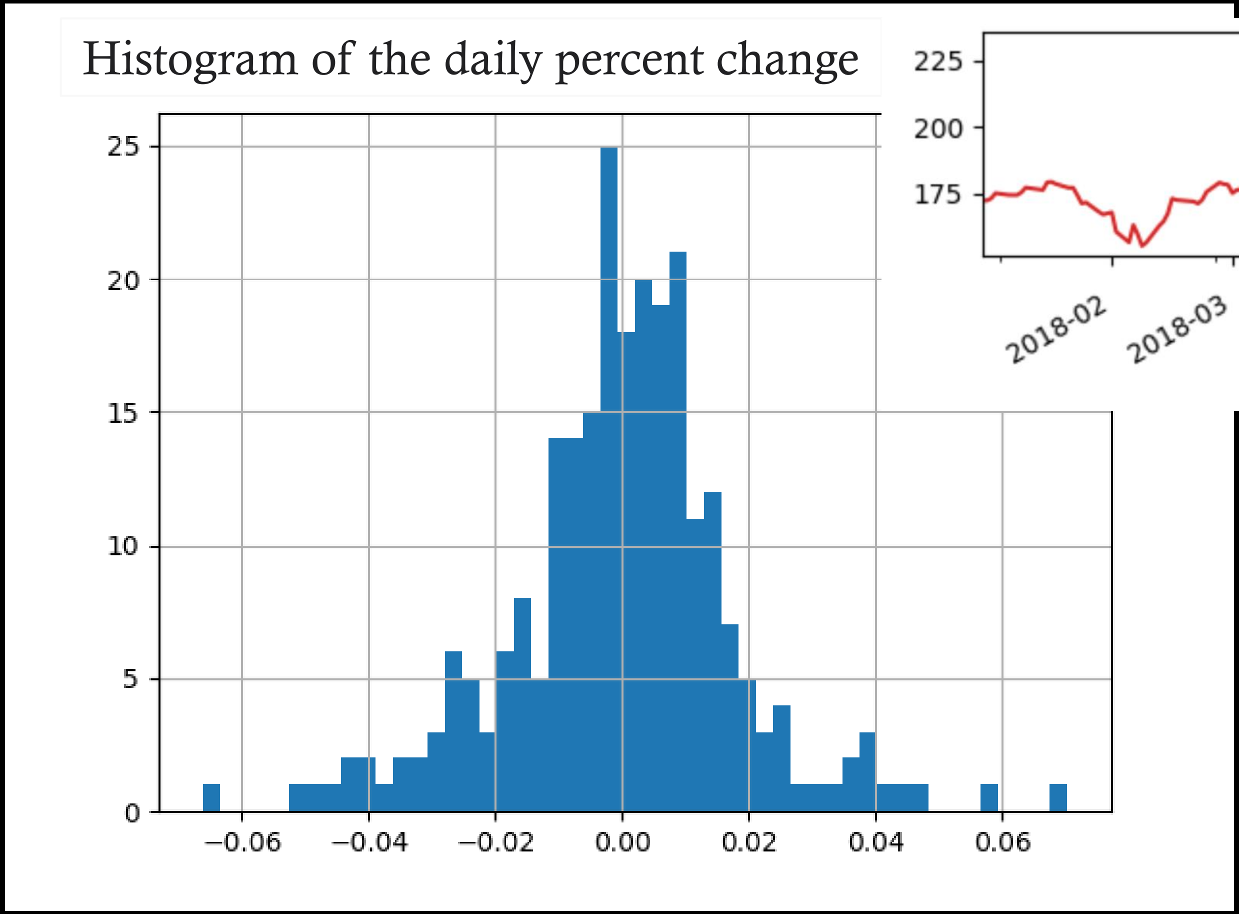
```
▼<div id="Col1-1-HistoricalDataTable-Proxy">
  ▼<section class="smartphone_Px(20px)" data-yaft-module=
  "tdv2-applet-HistoricalDataTable">
    ▶<div class="Mt(15px) drop-down-selector historical">…
    </div>
    ▼<div class="Pb(10px) Ovx(a) W(100%)">
      ▼<table class="W(100%) M(0)" data-test="historical-
      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>
```

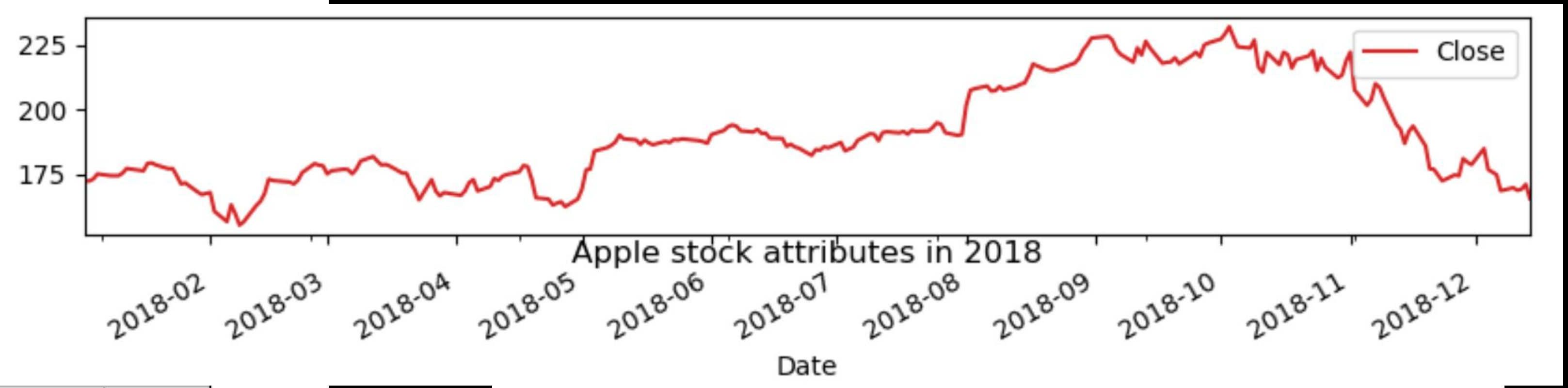
… #Main #Col1-1-HistoricalDataTable-Proxy section div table tbody tr td span

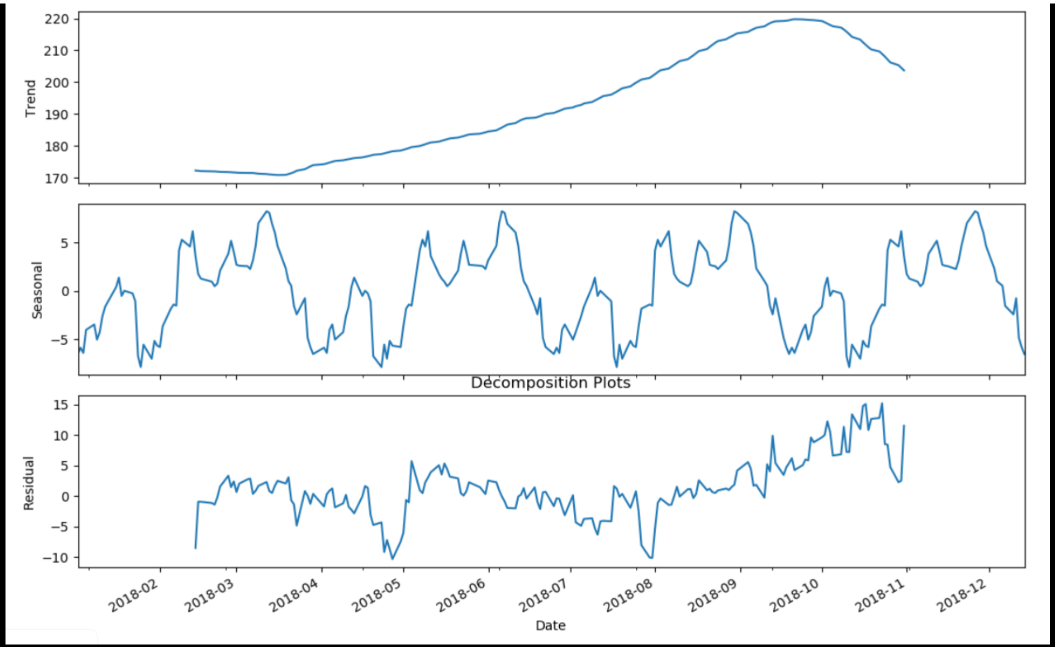
```
element.style {
}
Inherited from tr.BdT.Bdc($c-fuji-grey-c).Ta(end…
#atomic .Whs\(nw\), .Ell {          atomic.a5d89bb5.css:1
  white-space: nowrap;
}
#atomic .Ta\(end\) {               atomic.a5d89bb5.css:1
```

```
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 ar
    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)
```

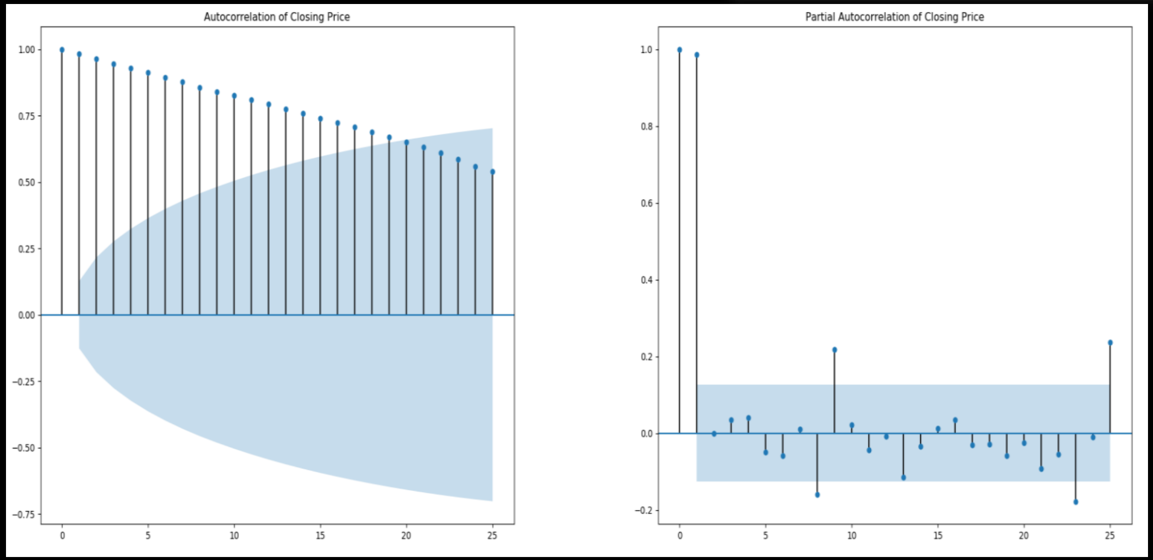
Exploratory Analysis

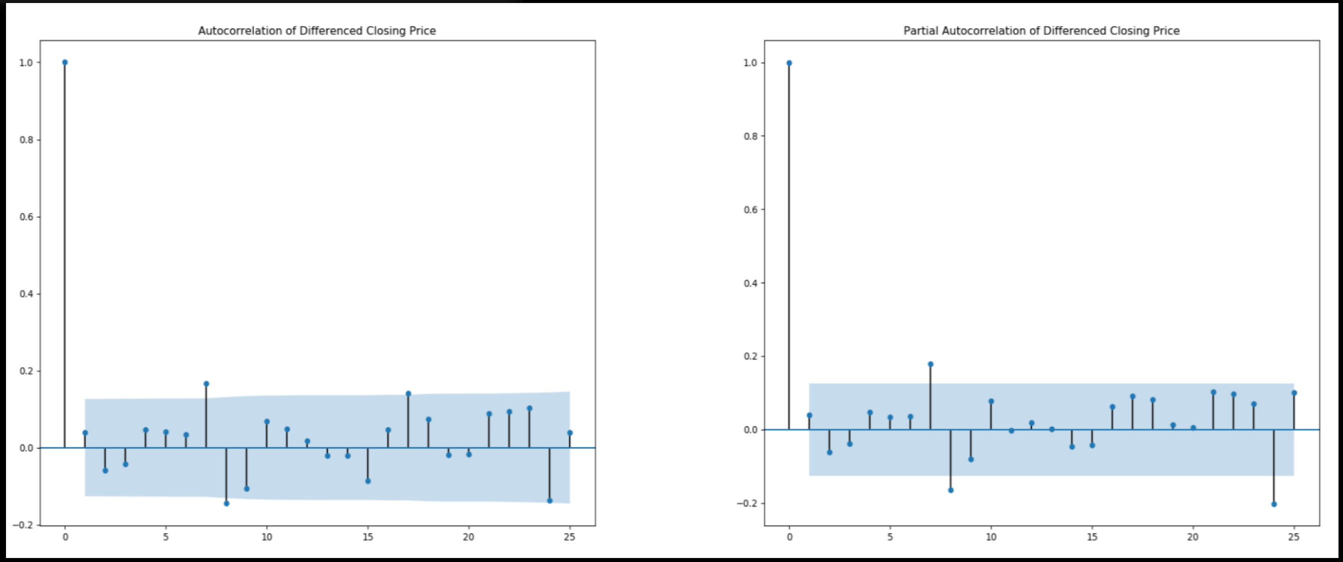


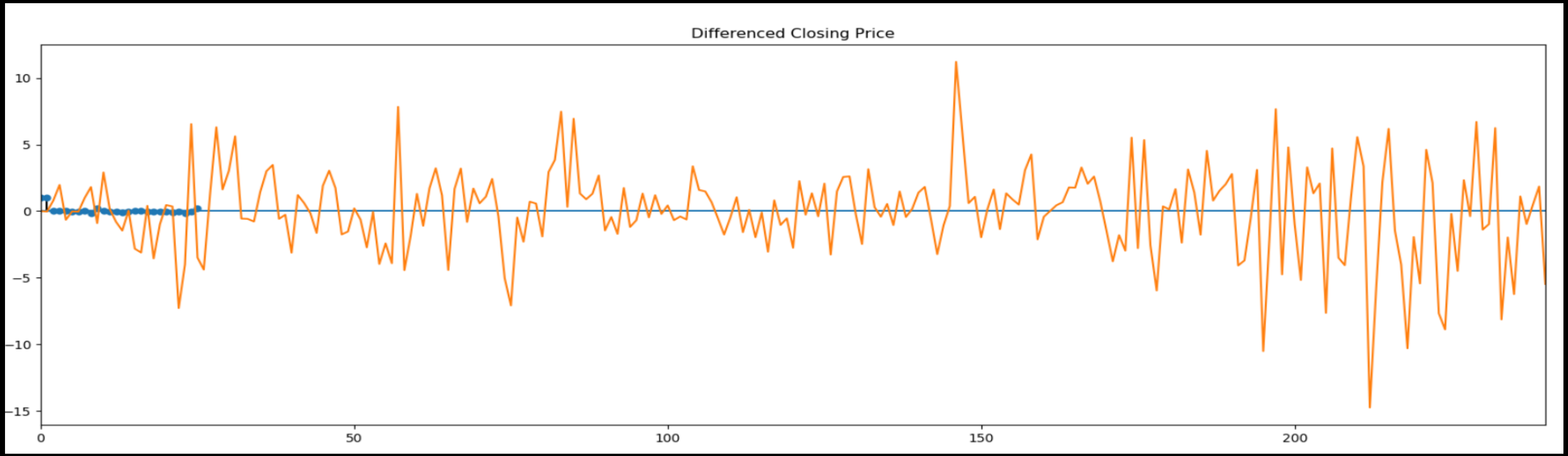




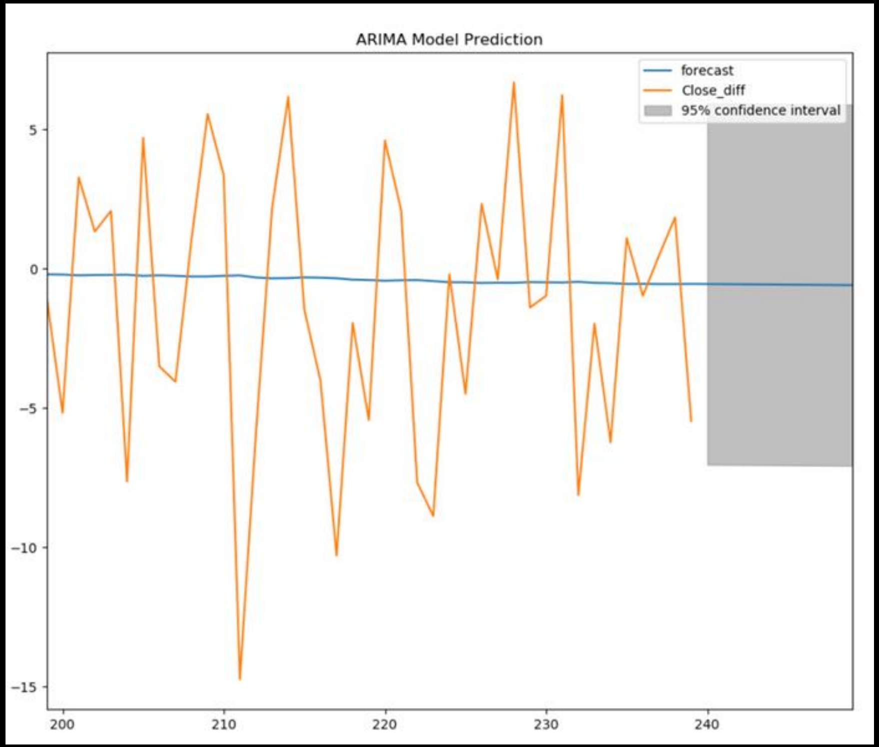
Exploratory Analysis

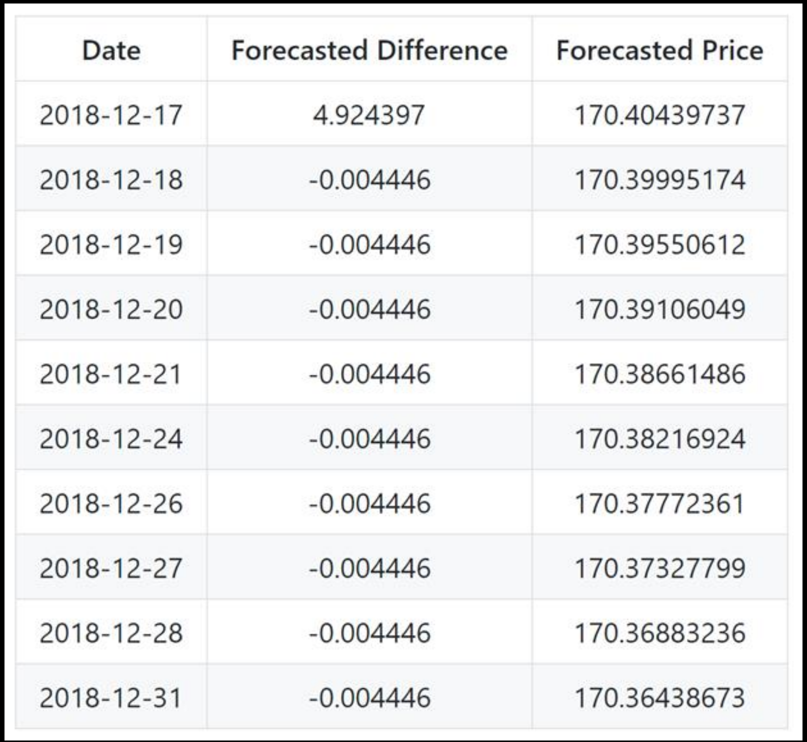






Model Analysis



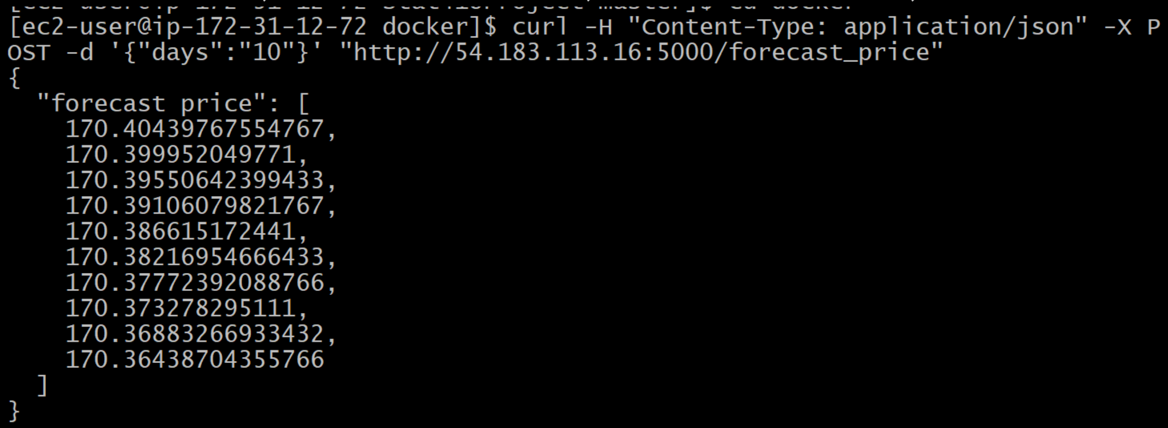


Date	Forecasted Difference	Forecasted Price
2018-12-17	4.924397	170.40439737
2018-12-18	-0.004446	170.39995174
2018-12-19	-0.004446	170.39550612
2018-12-20	-0.004446	170.39106049
2018-12-21	-0.004446	170.38661486
2018-12-24	-0.004446	170.38216924
2018-12-26	-0.004446	170.37772361
2018-12-27	-0.004446	170.37327799
2018-12-28	-0.004446	170.36883236
2018-12-31	-0.004446	170.36438673

RMSE: 5.0224385057219685

Deployment

- Docker & Flask & AWS
- Input: Desired number of days to forecast (from 2018-12-14)
- Output: List of forecasted price for desired amount of days
- curl -H "Content-Type: application/json" -X POST -d '{"days":"10"}' http://13.57.212.119:5000/forecast_price
- { "forecast price": [170.40439737, 170.39995174, 170.39550612, 170.39106049, 170.38661486, 170.38216924, 170.37772361, 170.37327799, 170.36883236, 170.36438673] }



```
[ec2-user@ip-172-31-12-72 docker]$ curl -H "Content-Type: application/json" -X P
OST -d '{"days":"10"}' "http://54.183.113.16:5000/forecast_price"
{
  "forecast price": [
    170.40439767554767,
    170.399952049771,
    170.39550642399433,
    170.39106079821767,
    170.386615172441,
    170.38216954666433,
    170.37772392088766,
    170.373278295111,
    170.36883266933432,
    170.36438704355766
  ]
}
```

UCLA Department of Statistics