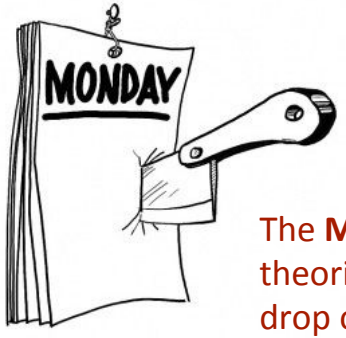
In Like a Bear, Out Like a Bull

Alok Kommajesula, Jessie Luk, David O'Morrissey, Jason Subbie

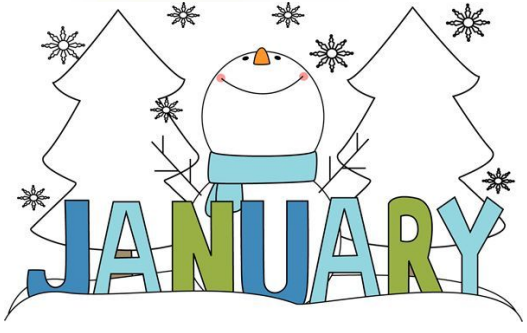
Overview

- *Best time to buy/sell stocks*
  - ***Day** of the week*
  - ***Month** of the year*
- *Trends within 5 industries:*
  - *Technology*
  - *Pharmaceuticals*
  - *Banking*
  - *Gaming*
  - *Food & Beverage*



The **Monday Effect** theorizes that stocks tend to drop on **Mondays.**

The **January Effect** is a seasonal increase in stock prices during the month of **January.**



Data

- ***Source:***
  - Alpha Vantage API, Yahoo Finance CSV
- **Parameters**:
  - Top 5 stocks within each of the 5 industries
  - 5 years (2013-2017) of daily data
  - S&P 500 = baseline

Daily Change

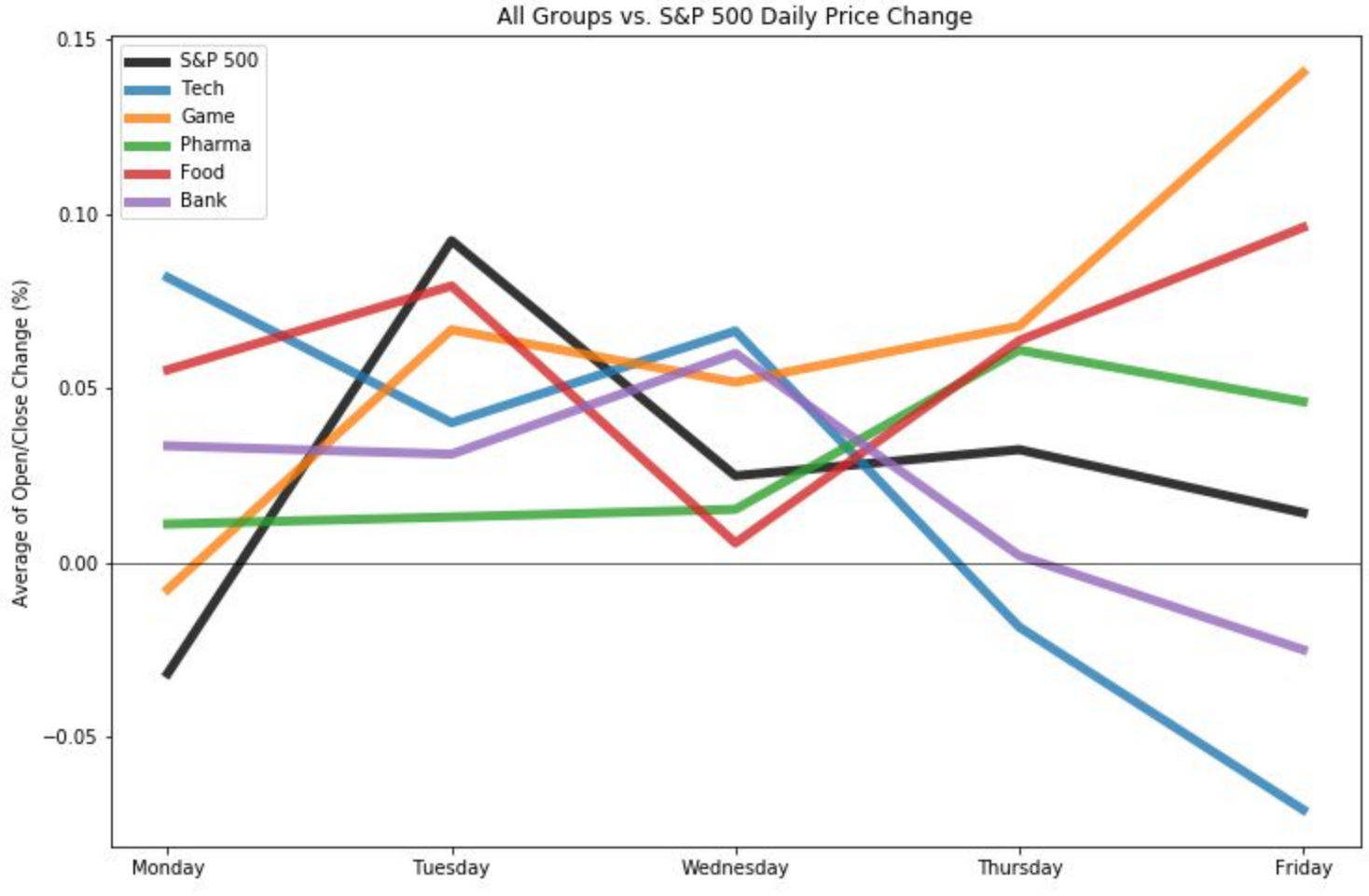
- Rate of Change
  - (New - Old) / Old
- Daily Rate of Change in Price
  - (Close - Open) / Open
- What does it tell you?

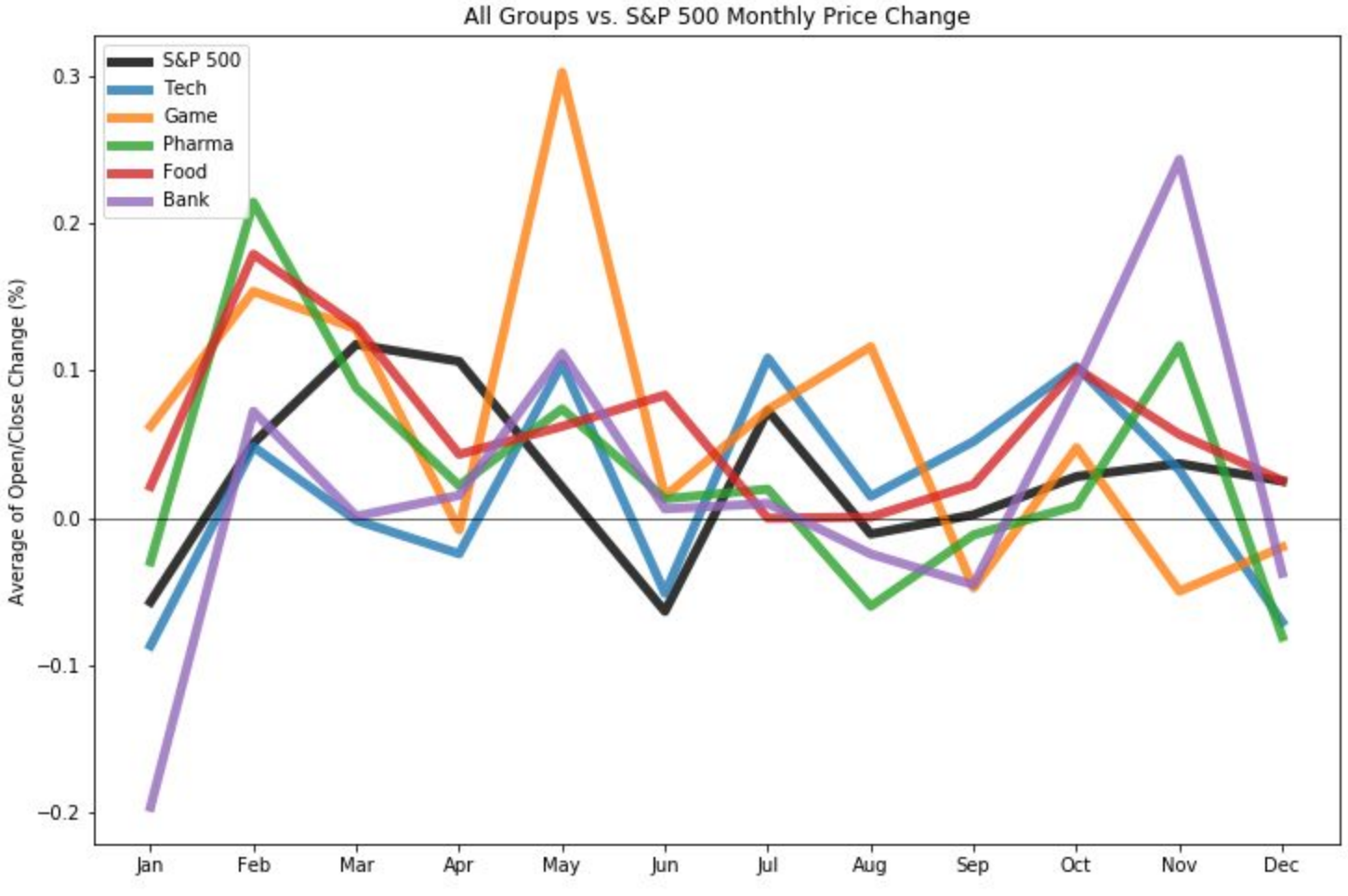
```
change = []

counter = 0
while counter < (len(df["index"]) -1) :
    o = df["1. open"][counter]
    c = df["4. close"][counter]
    calc = (float(c) - float(o)) / float(o)
    change.append(calc)
    counter += 1

change.append(0)
df["daily change"] = change
df.head()
```

Volatility

- Liability to Change Rapidly and Unpredictably
- Daily Rate of Volatility
  - (High - Low)/Low

```
Vol = []

x = 0

while x < len(master_df["Month"]):
    new = master_df["2. high"][x]
    old = master_df["3. low"][x]
    change = 0
    change = (new - old) /old
    Vol.append(change)
    x += 1

master_df["Volatility"] = Vol
master_df.head()
```

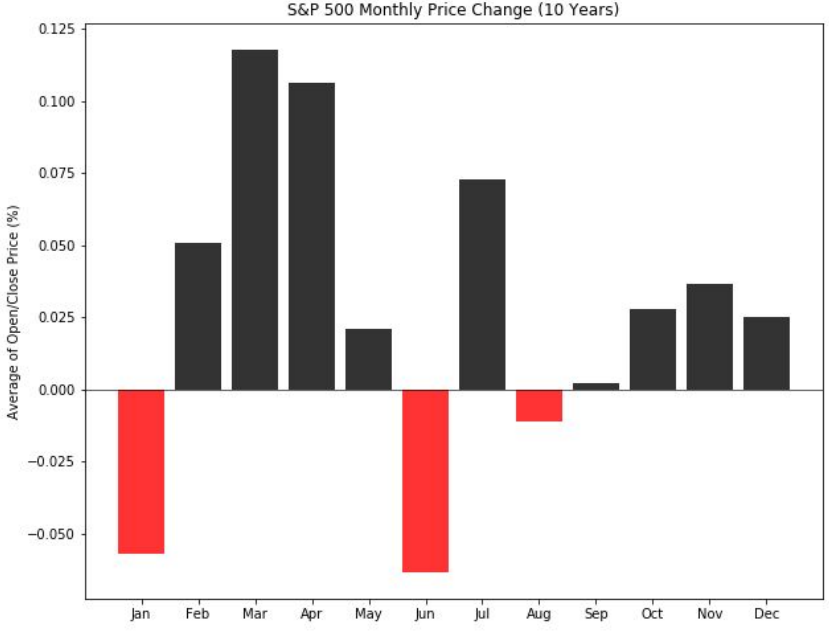

All Groups vs. S&P 500 Daily Price Change
S&P 500
Tech
Game
Pharma
Food
Bank
Average of Open/Close Change (%)
0.15
0.10
0.05
0.00
−0.05
Monday
Tuesday
Wednesday
Thursday
Friday

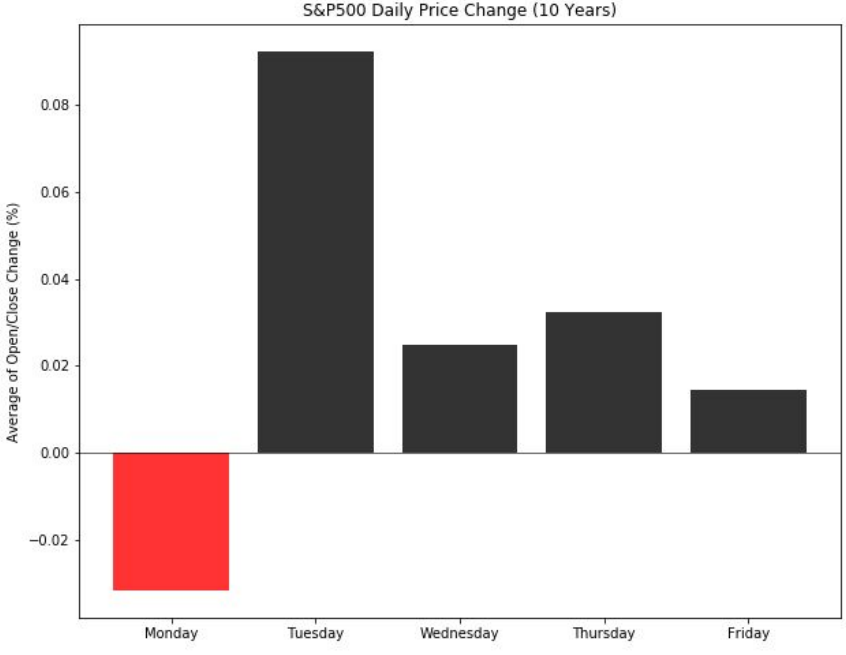

All Groups vs. S&P 500 Monthly Price Change
S&P 500
Tech
Game
Pharma
Food
Bank
Average of Open/Close Change (%)
0.3
0.2
0.1
0.0
−0.1
−0.2
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

S&P 500

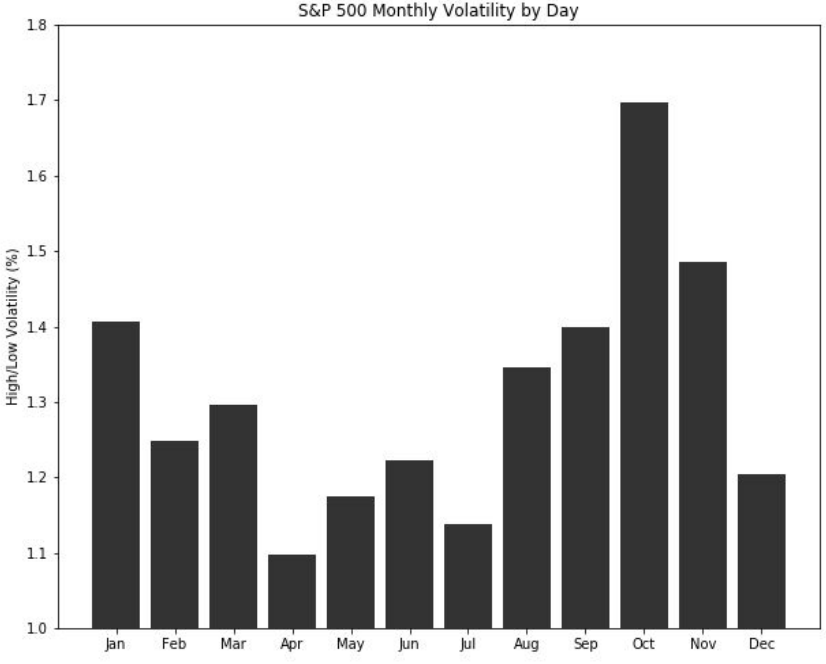
- Standard and Poors
- Why is it important?
  - Market Indicator

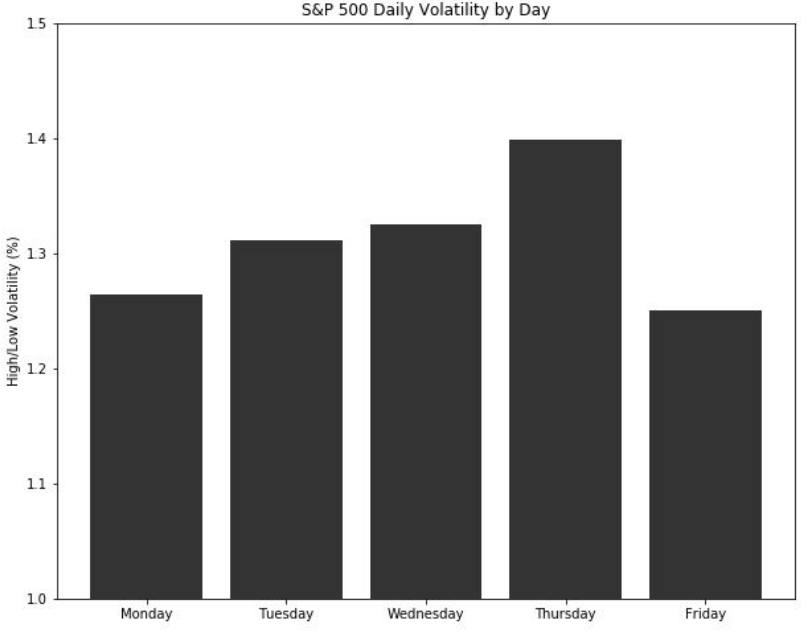
S&P 500


S&P 500 Monthly Price Change (10 Years)
Average of Open/Close Price (%)
0.125
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec

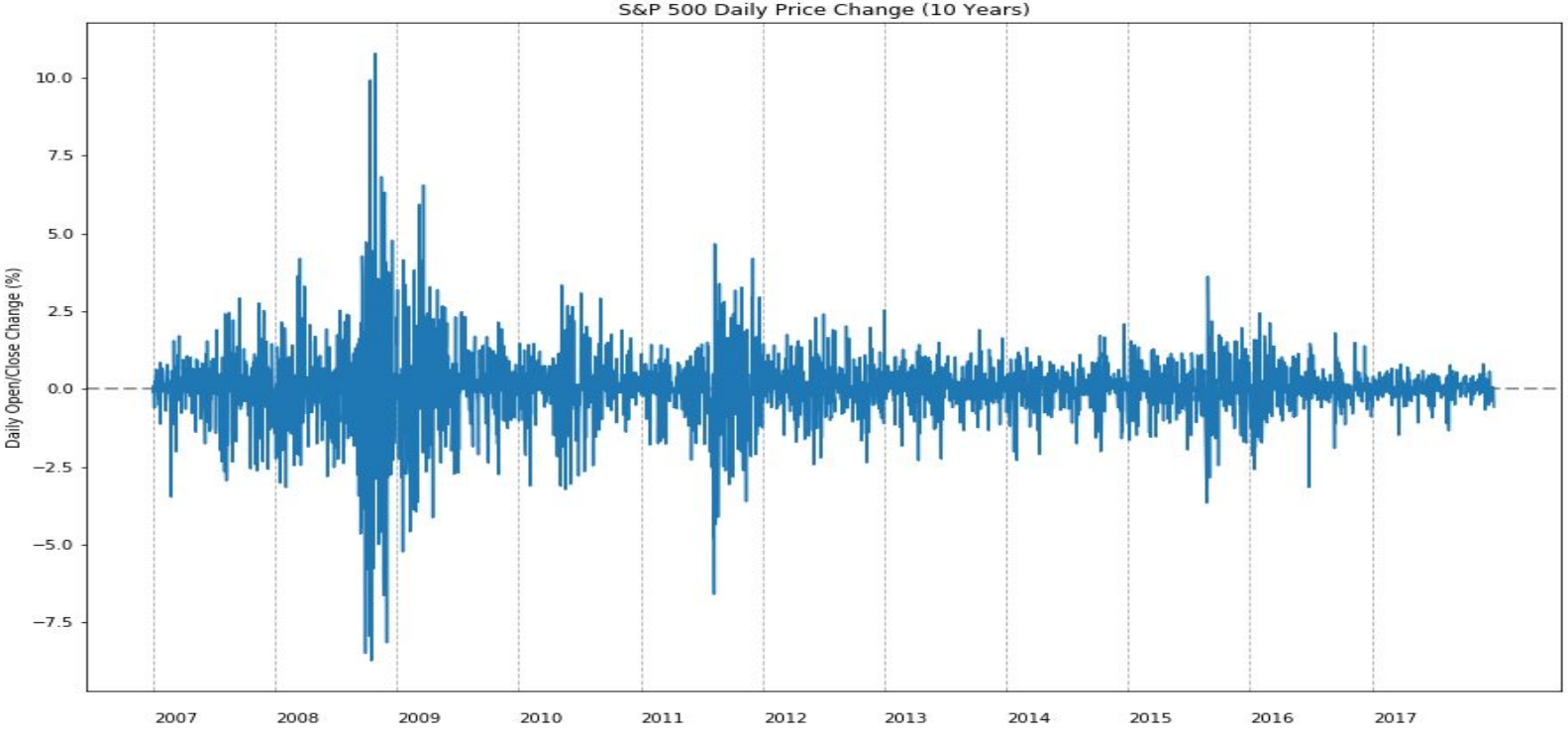

S&P500 Daily Price Change (10 Years)
Average of Open/Close Change (%)
0.08
0.06
0.04
0.02
0.00
−0.02
Monday
Tuesday
Wednesday
Thursday
Friday

S&P 500


S&P 500 Monthly Volatility by Day
High/Low Volatility (%)
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
1.8
Jan
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec


S&P 500 Daily Volatility by Day
High/Low Volatility (%)
1.0
1.1
1.2
1.3
1.4
1.5
Monday
Tuesday
Wednesday
Thursday
Friday

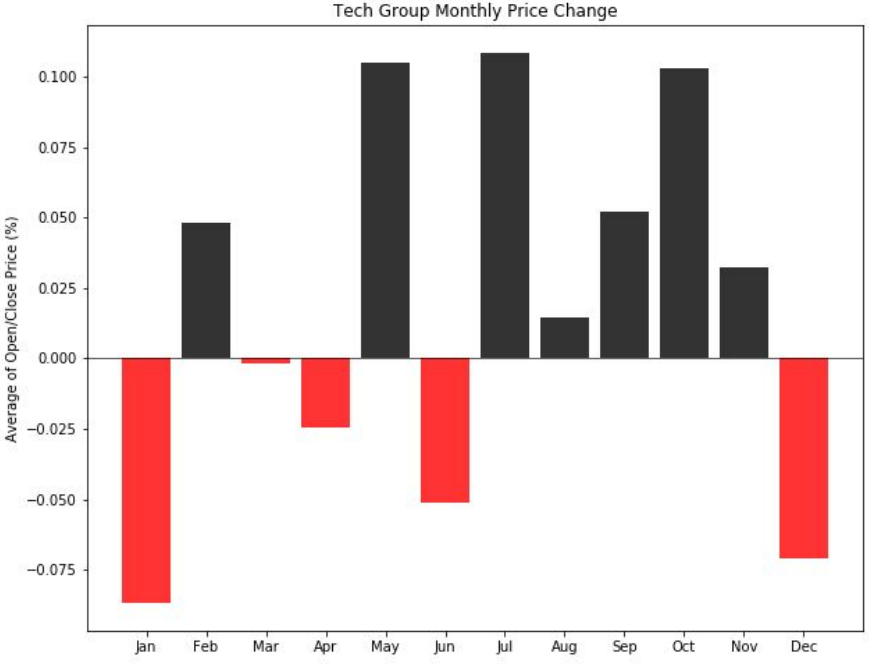
S&P 500

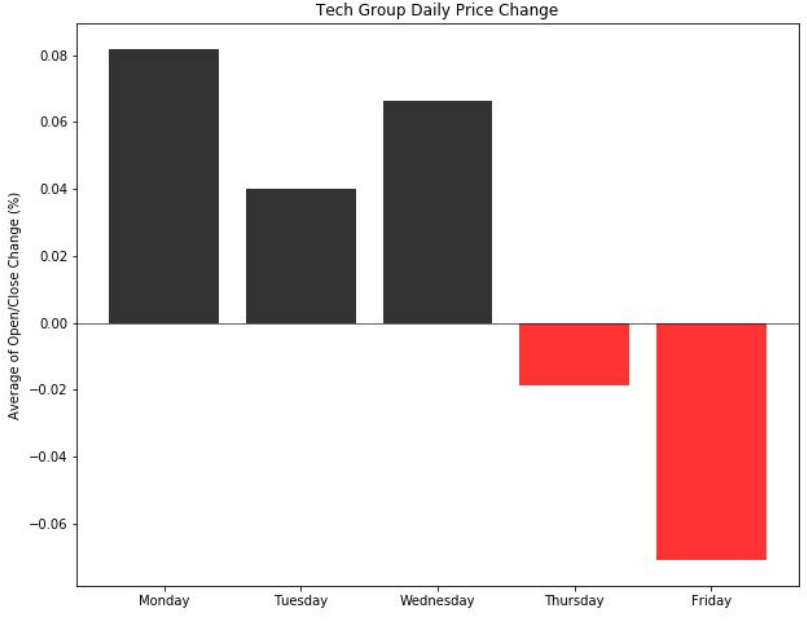

S&P 500 Daily Price Change (10 Years)
Daily Open/Close Change (%)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017

Technology

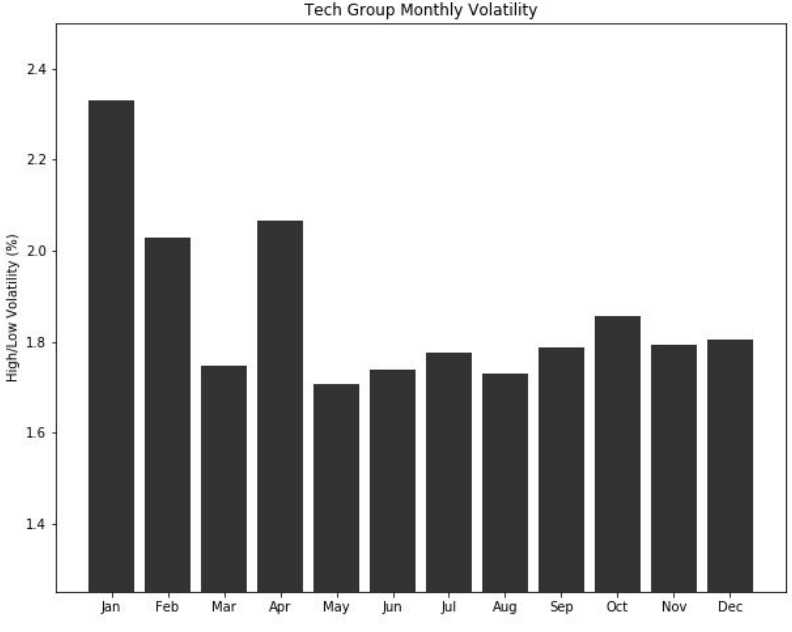
- Google
- Amazon
- Facebook
- Microsoft
- Apple

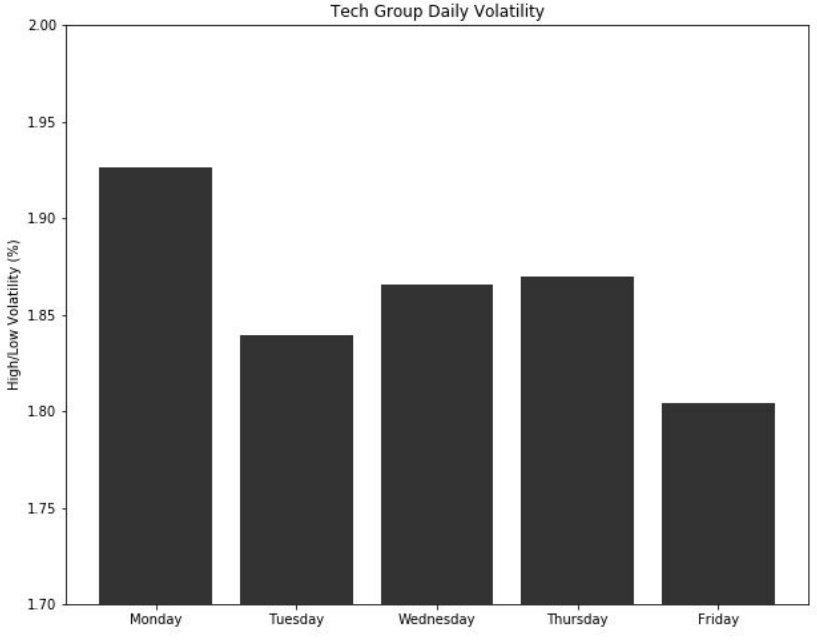
Technology (continued)



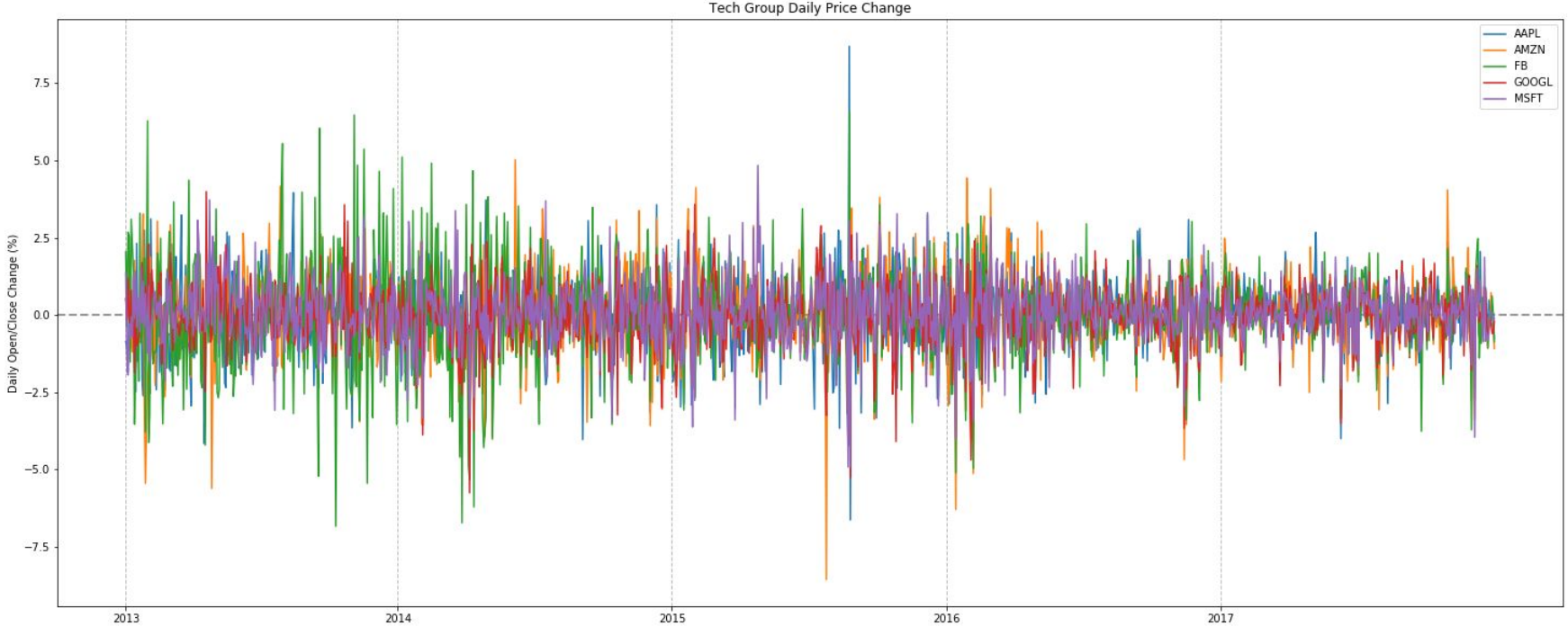


Technology (continued)





Technology (continued)



Sample API Call

```
import requests
import json
import pprint
import time
import pandas as pd
from datetime import datetime
import matplotlib.pyplot as plt
import numpy as np

url = "https://www.alphavantage.co/query"

function = "TIME_SERIES_DAILY"
symbol = "HSY"
api_key = "SMBQT4NW44MXXREK"
output = "full"

data = { "function": function,
        "symbol": symbol,
        "apikey": api_key,
        "outputsize": output}
page = requests.get(url, params = data)
pprint.pprint(page.json())
```

```
{'Meta Data': {'1. Information': 'Daily Prices (open, high, low, close) and '
                                 'Volumes',
               '2. Symbol': 'HSY',
               '3. Last Refreshed': '2018-01-11',
               '4. Output Size': 'Full size',
               '5. Time Zone': 'US/Eastern'},
 'Time Series (Daily)': {'2000-01-03': {'1. open': '47.0000',
                                        '2. high': '47.5000',
                                        '3. low': '46.1900',
                                        '4. close': '46.3800',
                                        '5. volume': '453200'},
                         '2000-01-04': {'1. open': '45.7500',
                                        '2. high': '46.0000',
                                        '3. low': '44.4400',
                                        '4. close': '44.6300',
                                        '5. volume': '530000'},
                         '2000-01-05': {'1. open': '44.7500',
                                        '2. high': '46.5000',
                                        '3. low': '44.3800',
```

Sample Code

```
month = []
day=[]
year = []
for index, row in df.iterrows():
    a = row["index"]
    datee = datetime.strptime(a, "%Y-%m-%d")
    monthdate = datee.strftime("%B")
    daydate = datee.strftime("%A")
    yeardate = datee.strftime("%Y")
    month.append(monthdate)
    year.append(yeardate)
    day.append(daydate)

df["Month"] = month
df["Day"] = day
df["Year"]= year
df["Company"] = symbol
df.head()

change = []

counter = 0
while counter < (len(df["index"]) -1) :
    open_num = df["1. open"][counter]
    close_num = df["4. close"][counter]
    calc = (float(close_num) - float(open_num)) / float(open_num)
    change.append(calc)
    counter += 1

change.append(0)
df["daily change"] = change
df.head()
```

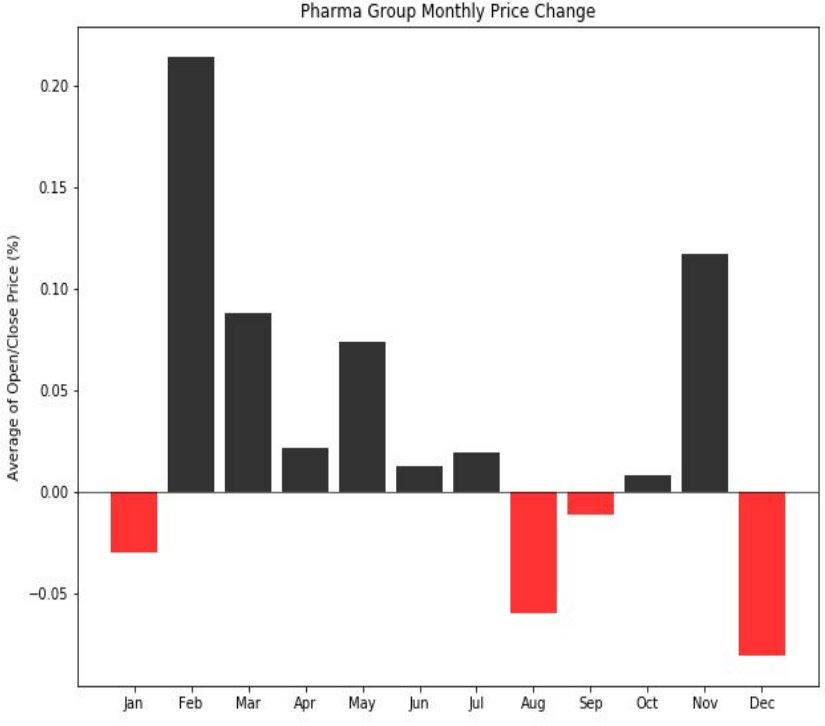
Sample Code

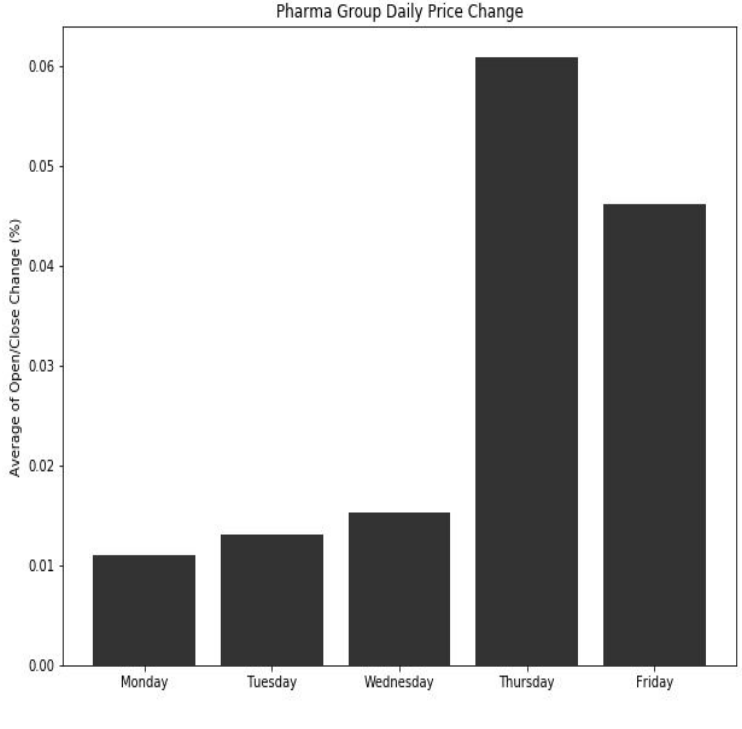
```
x_axis = np.arange(len(comp_one["Company"]))
y_axis = (invert_df["daily change"] * 100)
plt.figure(figsize=(25,10))

f = 0
year_label = invert_df["Year"].unique()
year_count = 0
for year in year_break:
    plt.axvline(x=(year), linestyle = "--", linewidth = 1, color = "black", alpha = .25, gid = year_label[year_count])
    f += 1
    year_count += 1

plt.xticks(year_break, invert_df["Year"].unique())
plt.axhline(y=0, linestyle='--', linewidth = 2, color = "black", alpha = .4)
plt.ylabel("Daily Open/Close Change (%)")
plt.title("Food Group Daily Price Change")

plt.ylim(-5,5)

zero = (invert_zero["daily change"] * 100)
one = (invert_one["daily change"] * 100)
two = (invert_two["daily change"] * 100)
three = (invert_three["daily change"] * 100)
four = (invert_four["daily change"] * 100)

zero_plot = plt.plot(x_axis, zero, label=companies[0])
one_plot = plt.plot(x_axis, one, label = companies[1])
two_plot = plt.plot(x_axis, two, label = companies[2])
three_plot = plt.plot(x_axis, three, label = companies[3])
four_plot = plt.plot(x_axis, four, label = companies[4])

plt.legend(handles=[zero_plot[0], one_plot[0], two_plot[0], three_plot[0], four_plot[0]])

plt.savefig("Figures/TechDailyVolLineCut")
plt.show()
```

Pharmaceuticals

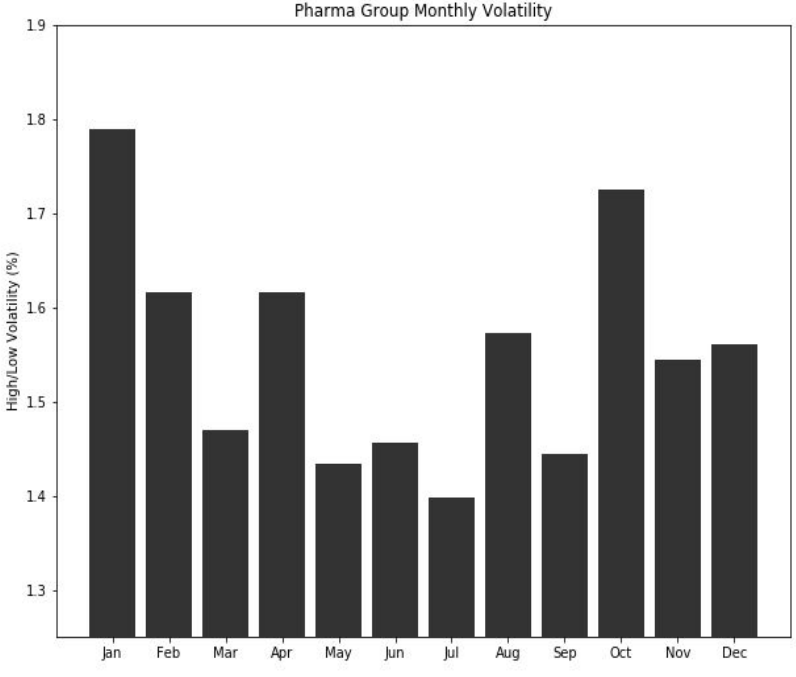
- Johnson & Johnson
- Pfizer
- Novartis
- Merck & CO
- Roche Holding AG

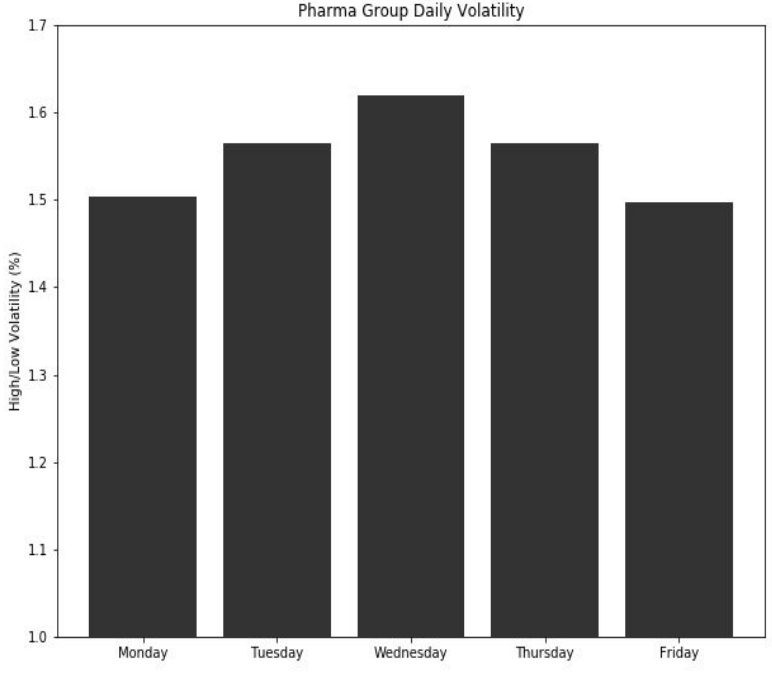
Pharmaceutical (continued)



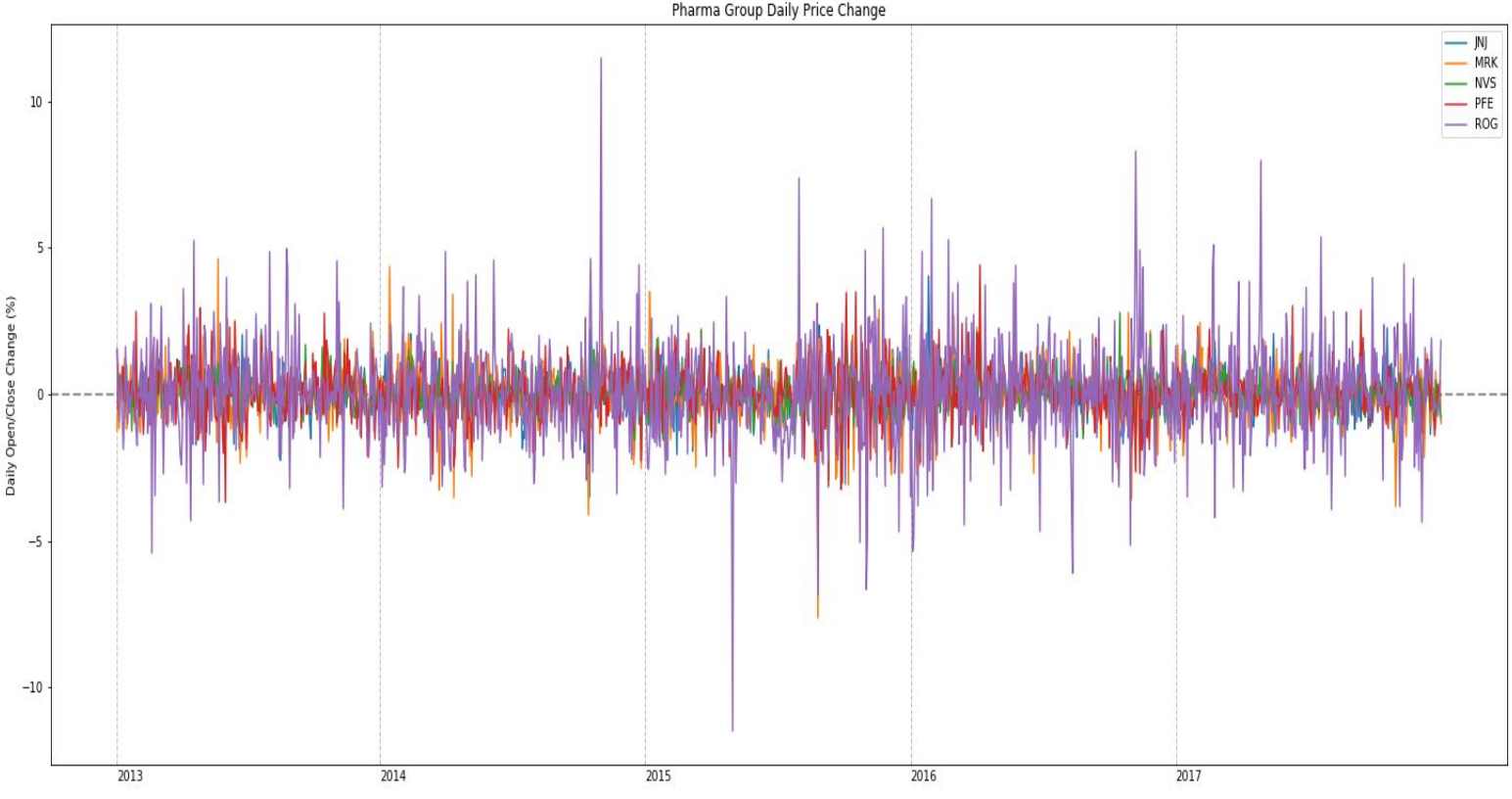


Pharmaceutical (continued)





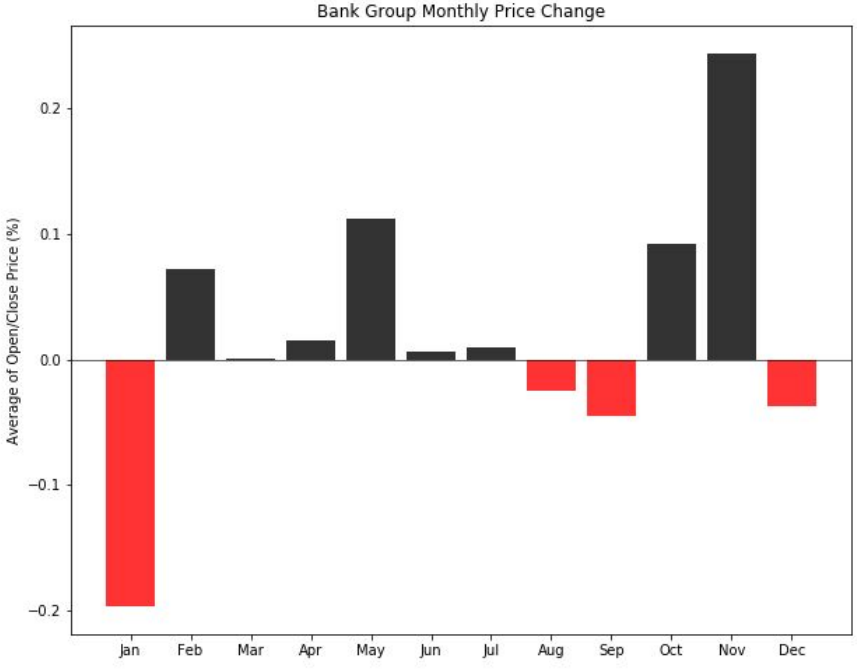
Pharmaceutical

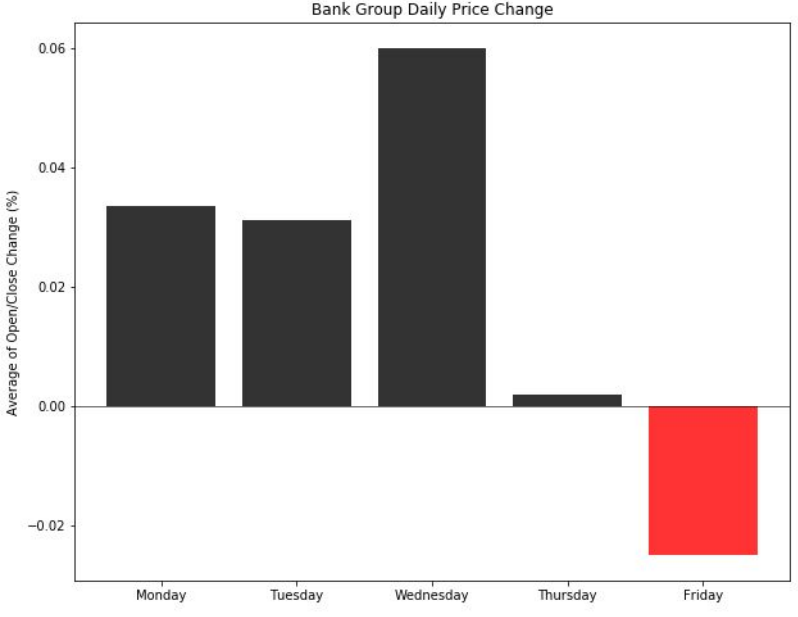


Banking

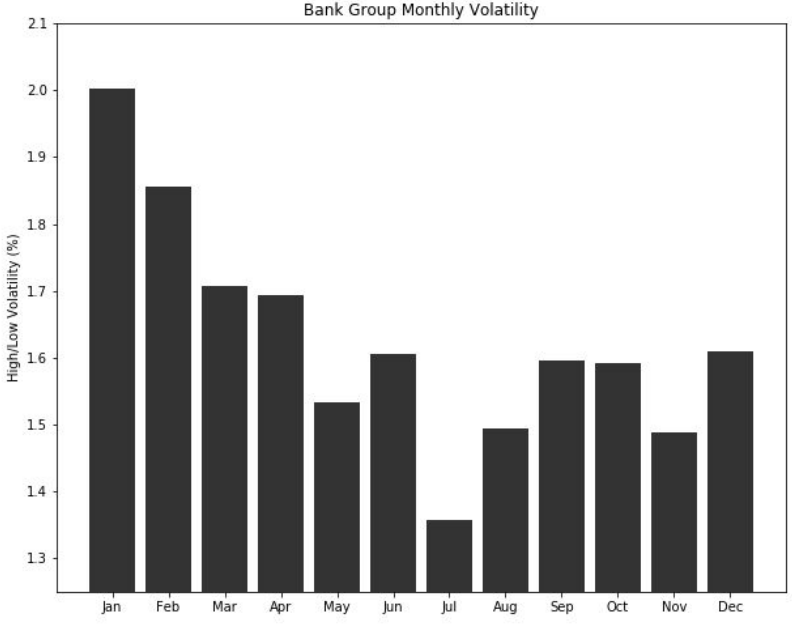
- JPMC
- Goldman Sachs
- Citigroup
- Bank of America
- Wells Fargo

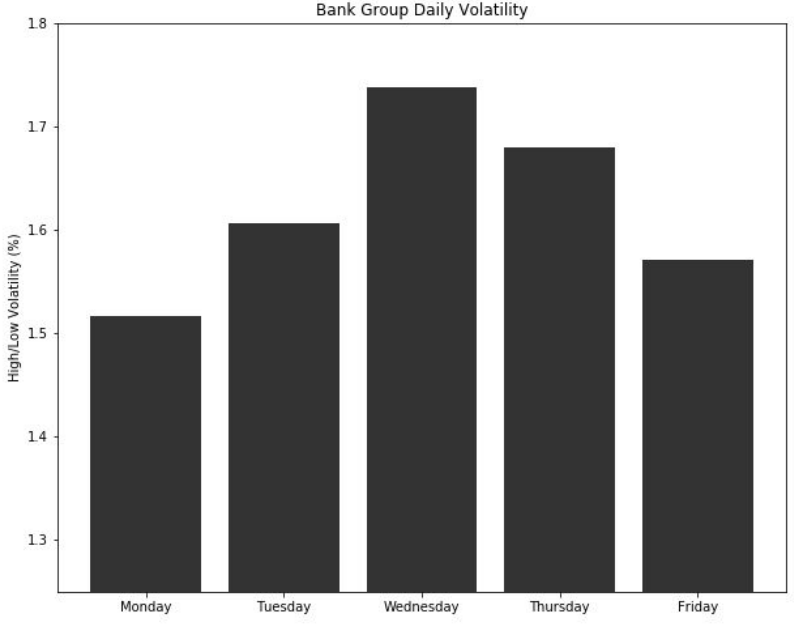
Banking (continued)



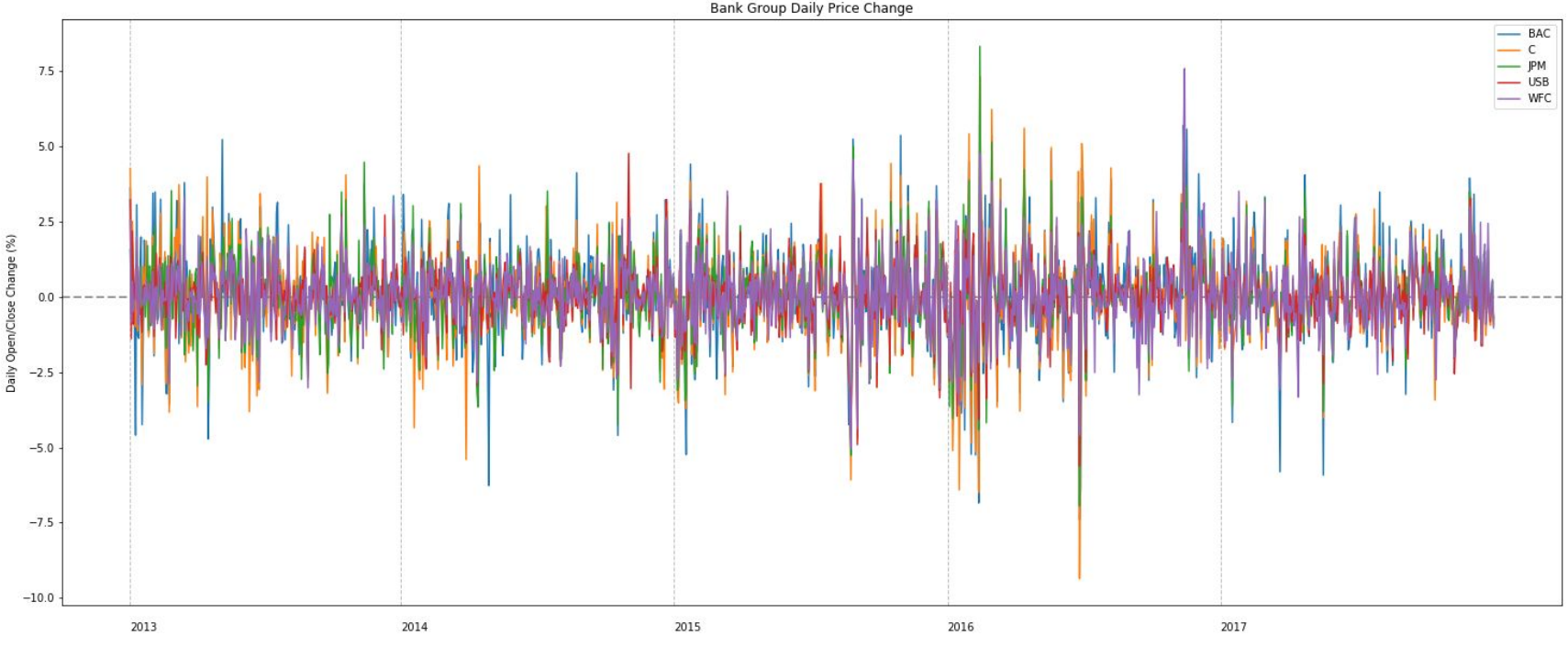


Banking (continued)





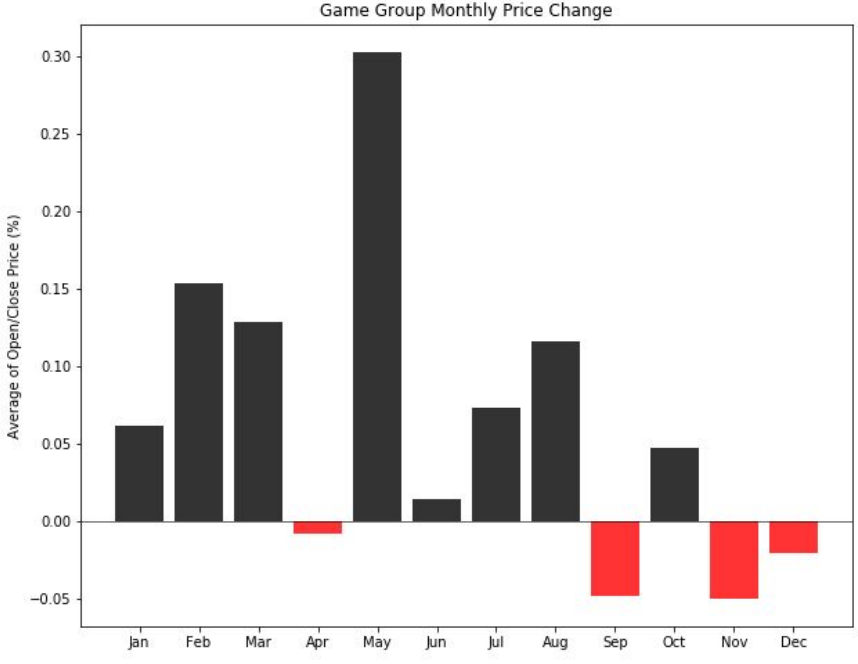
Banking (continued)

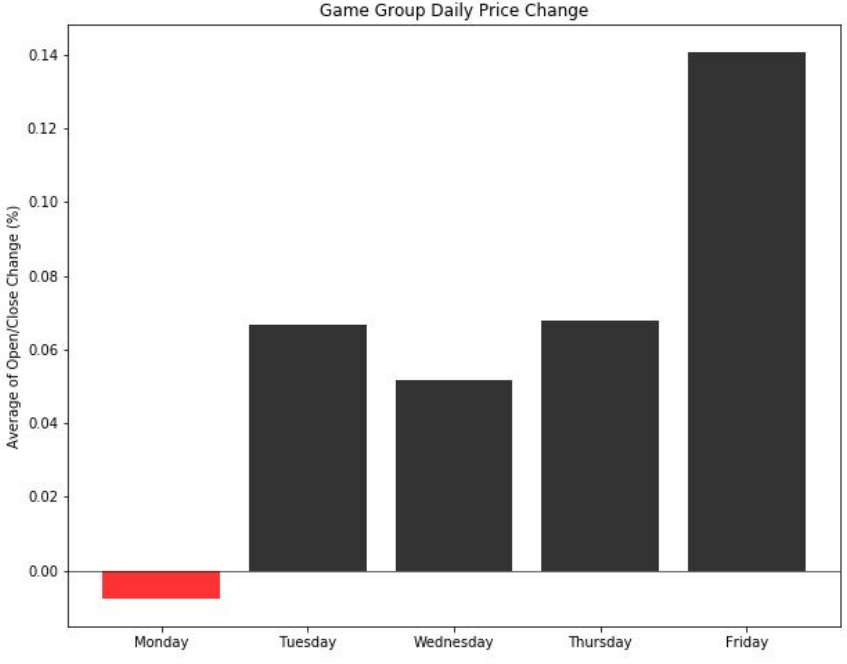


Gaming

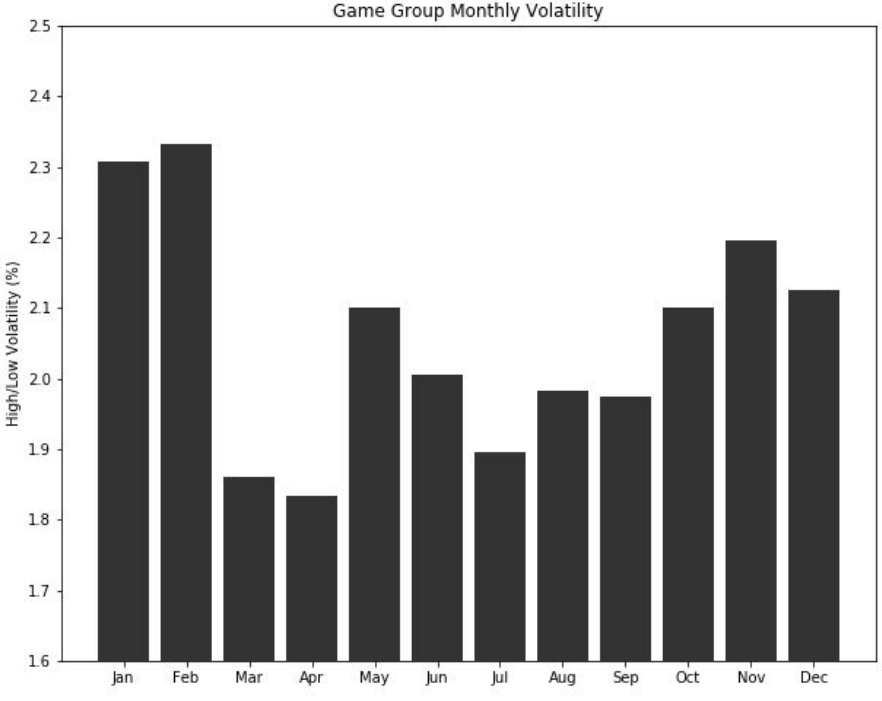
- EA
- Nintendo
- Take Two Interactive
- Microsoft
- Activision

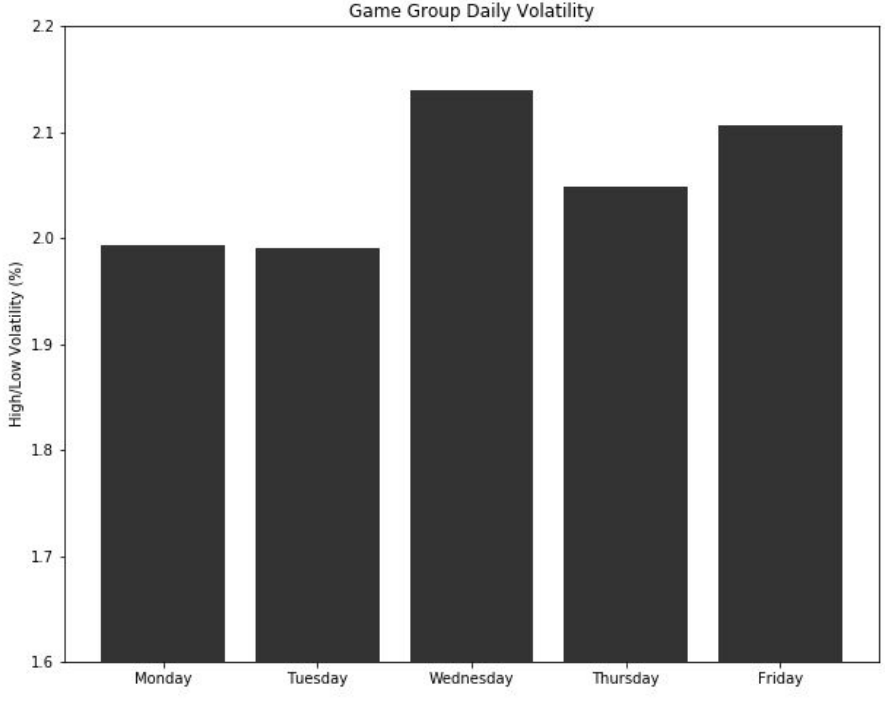
Gaming (continued)



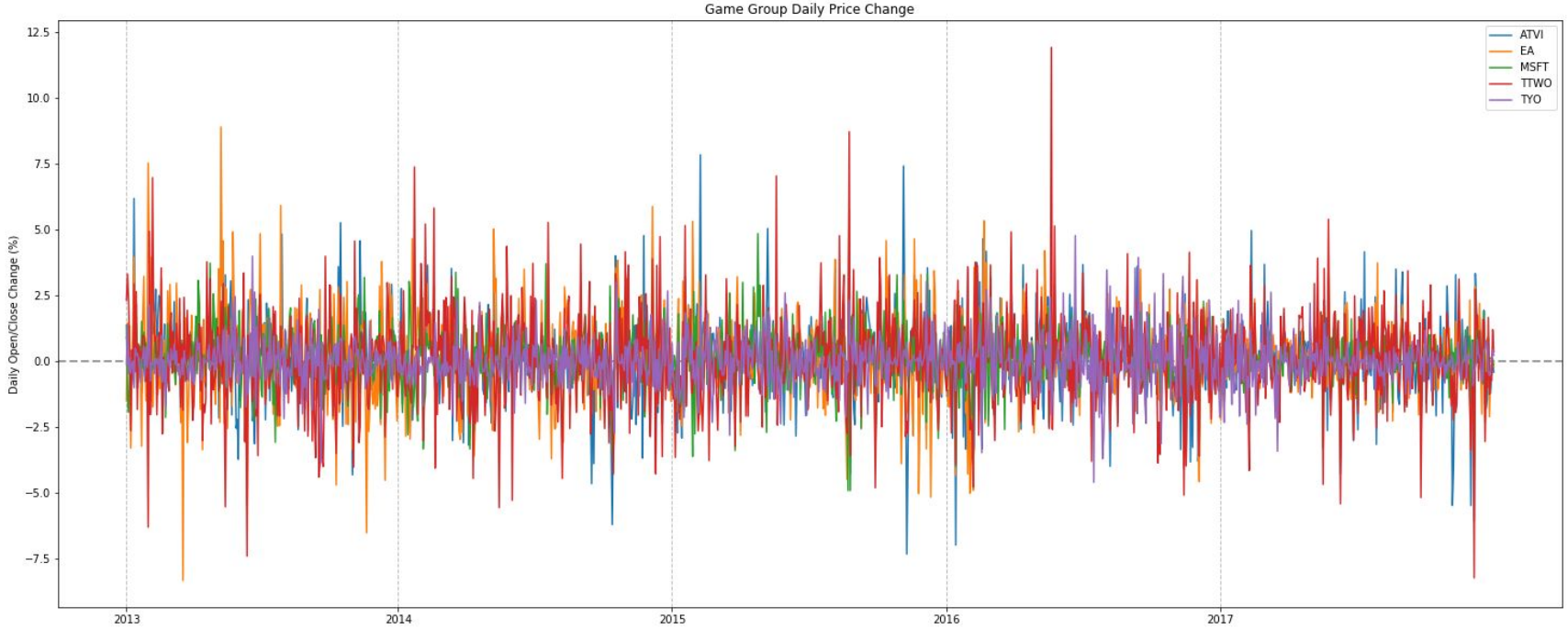


Gaming (continued)





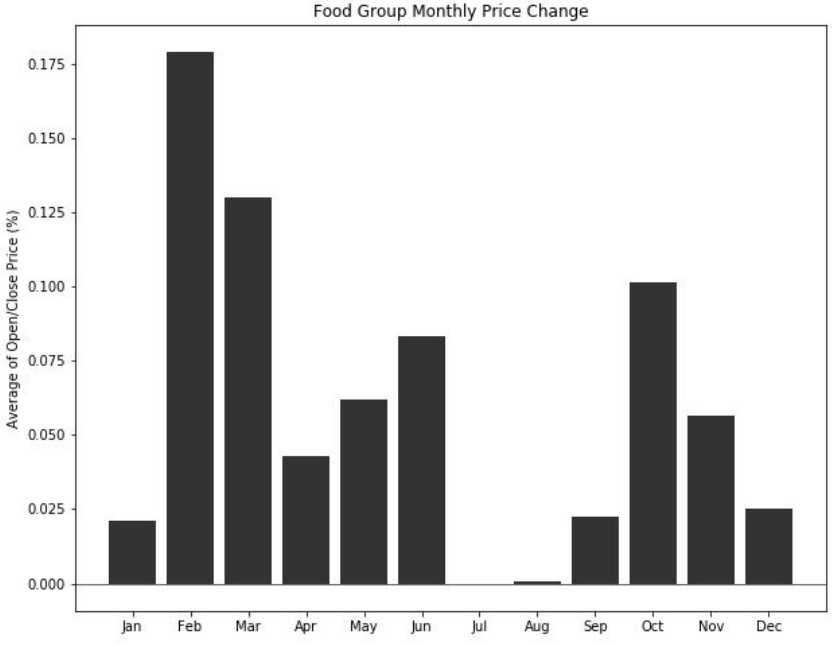
Gaming (continued)

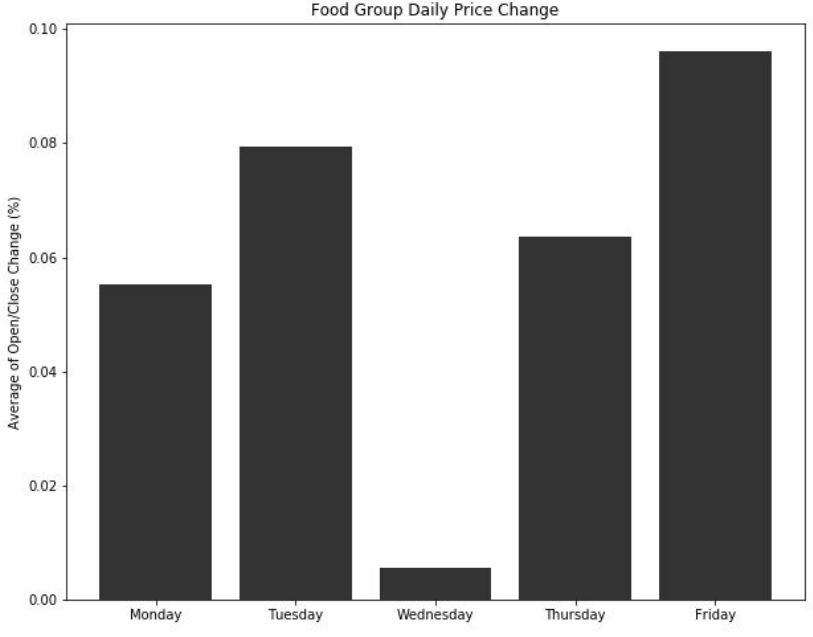


Food & Beverage

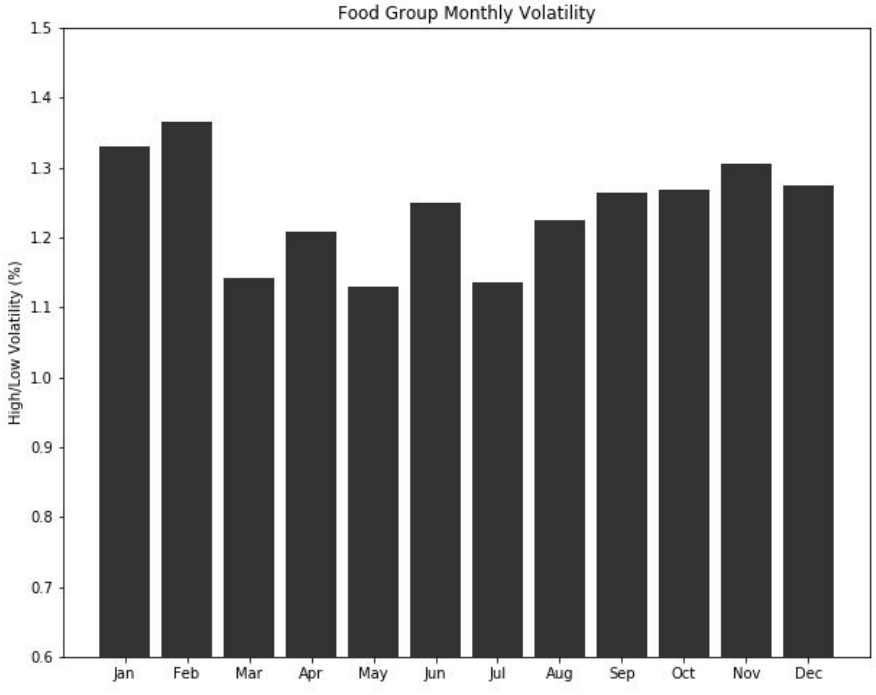
- PepsiCola
- Hershey
- CocaCola
- JBS
- Anheuser-Busch InBev

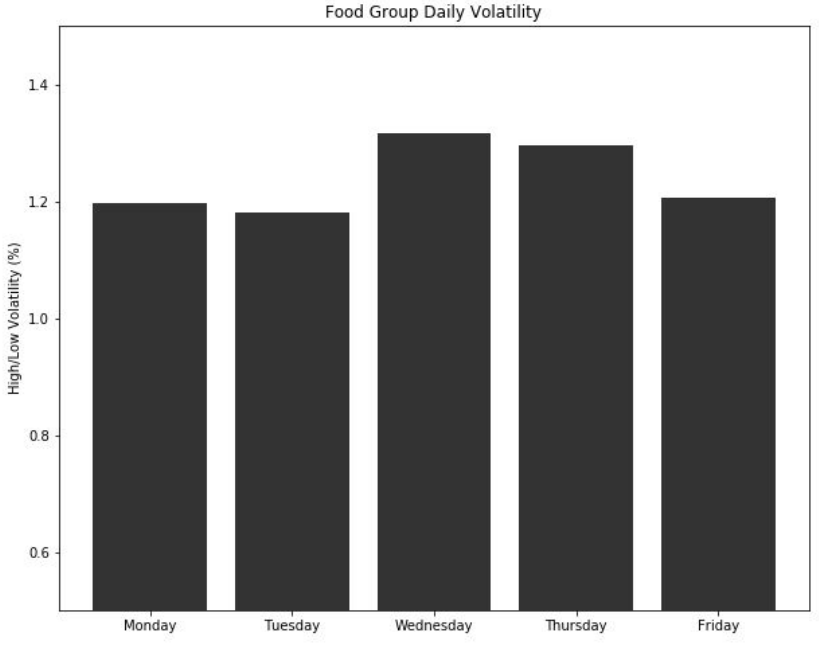
Food & Beverage



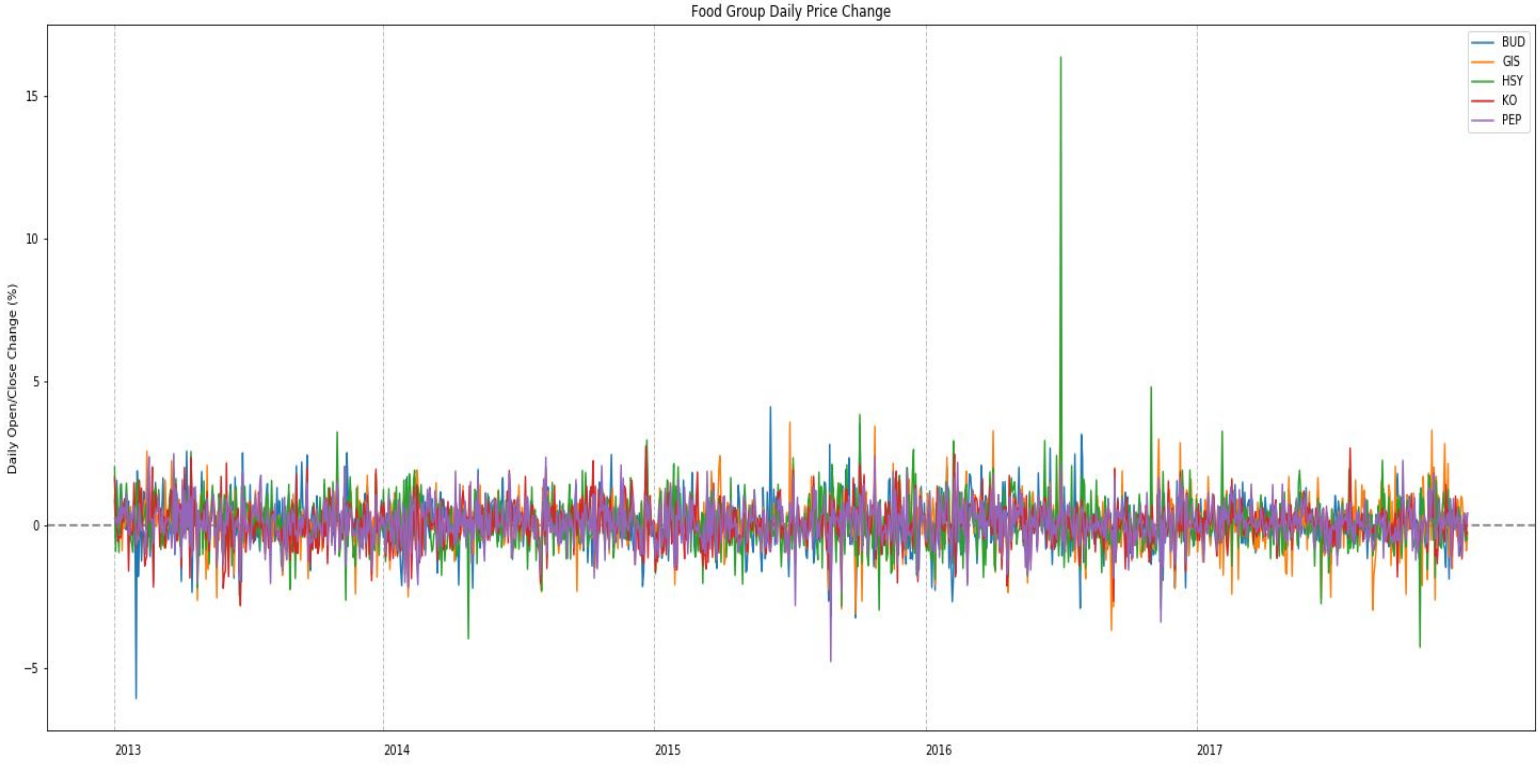


Food & Beverage





Food & Beverage



Conclusion

- Best Recommendation:
  - BUY Mondays in June
  - SELL Tuesday in March
- Not all industries are the same
  - Follow Monday:
    - Gaming
  - Follow January
    - Bank, Pharma, Tech
  - Sell in February
    - Food & Beverage
- Many other factors involved

Live Demonstration

- One, big name stock
- 10 year minimum
- NYSE or NASDAQ
  - McDonalds
  - 3M
  - Boeing
  - Home Depot
  - Chevron