PoliStock Data Gathering

In this section we will be used to retrieve historical OPI (Open Price Index) data for the S&P 500 sectors as well as the S&P 500 Index as a whole. We prefer that data for each sector and stagnant unrelated columns. We also checked the data for null values and found none. We also checked for duplicate dates and found none. Then we concatenated our dataframes into one and stored it into the pickle file "sector_data.pkl".

Use Yahoo Finance yfinance open source library to pull stock data from four S&P 500 sectors: Energy (XLE), Finance (XLF), Technology (XLK), Healthcare (XLV) and the S&P 500 index

```
# Import yfinance and pandas
import yfinance as yf
import pandas as pd
# Define the ticker symbols for the sector ETFs and the S&P 500 index
tickers = ['XLE', 'XLF', 'XLK', 'XLV', '^GSPC']
# Download data for the last 10 years
data = yf.download(tickers, start='2010-01-01', end='2020-01-01')['Open']
# Display the first few rows of the data
print(data.head())
```

```
[*********************100%***********************]  5 of 5 completed
                  ^GSPC        XLE        XLF        XLK        XLV
Date
2010-01-04  1117.010010  72.660004  15.720000  26.960000  28.020000
2010-01-05  1118.480029  72.559998  15.830000  26.980000  28.080000
2010-01-06  1116.560059  73.019997  15.810000  26.957500  27.880000
2010-01-07  1112.349970  73.070000  15.750000  26.957500  27.790000
2010-01-08  1118.070029  73.908002  15.860000  26.957500  27.860000
```

```
# Import the pandas library
import pandas as pd
# Define the ticker symbols for the sector ETFs and the S&P 500 index
tickers = ['XLE', 'XLF', 'XLK', 'XLV', '^GSPC']
# Download data for the last 10 years
data = yf.download(tickers, start='2010-01-01', end='2020-01-01')['Open']
# Calculate the mean of each column
mean_values = data.mean()
# Display the mean values
print(mean_values)
```

```
[*********************100%***********************]  5 of 5 completed
^GSPC    2126.000000
XLE        56.000000
XLF        19.000000
XLK        40.000000
XLV        39.000000
dtype: float64
```

```
# Import the pandas library
import pandas as pd
# Define the ticker symbols for the sector ETFs and the S&P 500 index
tickers = ['XLE', 'XLF', 'XLK', 'XLV', '^GSPC']
# Download data for the last 10 years
data = yf.download(tickers, start='2010-01-01', end='2020-01-01')['Open']
# Calculate the mean of each column
mean_values = data.mean()
# Display the mean values
print(mean_values.to_frac_str())
```

```
[*********************100%***********************]  5 of 5 completed
^GSPC    2126
XLE        56
XLF        19
XLK        40
XLV        39
```

```
Energy Fuels      -0.000000
Energy Fuels      -0.000000
Energy Fuels      -0.000000
Energy Fuels      -0.000000
Energy Fuels      -0.000000
dtype: float64
```

```
sector_stdev_combined[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Sector Standard Deviation Before Election')
```

Sector Standard Deviation For All Presidential Terms

Calculate Standard Deviation For Biden Before, After The Election and Combined

```
sector_stdev_combined_biden_before_election = sector_stdev_combined[sector_stdev_combined.Date < '2020-11-03']
sector_stdev_combined_biden_after_election = sector_stdev_combined[sector_stdev_combined.Date > '2020-11-03']
```

```
sector_stdev_combined_biden_before_election[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Biden Standard Deviation Before Election')
```

Biden Standard Deviation Before Election

```
sector_stdev_combined_biden_after_election[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Biden Standard Deviation After Election')
```

Biden Standard Deviation After Election

```
combined_stdev = pd.concat([sector_stdev_combined_biden_after_election,sector_stdev_combined_biden_before_election], axis=0 ).sort_values(['Date', 'Date'])
combined_stdev.reset_index(drop=True,inplace=True)
combined_stdev.to_csv('combined_stdev.csv')
```

```
combined_stdev[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Biden Standard Deviation')
```

Biden Standard Deviation

Calculate Standard Deviation For Trump Before, After the Election and Combined

```
sector_stdev_trump_combined = pd.read_csv('sector_stdev_trump_combined.csv')
sector_stdev_trump_combined_before_election = sector_stdev_trump_combined[sector_stdev_trump_combined.Date < '2020-11-03']
sector_stdev_trump_combined_after_election = sector_stdev_trump_combined[sector_stdev_trump_combined.Date > '2020-11-03']
```

```
sector_stdev_trump_combined_before_election[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Trump Standard Deviation')
```

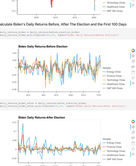
Trump Standard Deviation

Calculate Standard Deviation For Obama During First, Second Term and Combined

```
sector_stdev_obama_first = sector_stdev_combined[sector_stdev_combined.Date < '2013-01-20']
sector_stdev_obama_second = sector_stdev_combined[sector_stdev_combined.Date > '2013-01-20']
sector_stdev_obama_first.reset_index(drop=True,inplace=True)
sector_stdev_obama_second.reset_index(drop=True,inplace=True)
```

```
sector_stdev_obama_first[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Obama Standard Deviation - First Term')
```

Obama Standard Deviation - First Term




```
sector_stdev_obama_second[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Obama Standard Deviation - Second Term')
```

Obama Standard Deviation - Second Term

```
Bush_stdev_combined[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Bush Standard Deviation - First Term')
```

Bush Standard Deviation - First Term

```
Bush_stdev_combined[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Bush Standard Deviation - Second Term')
```

Bush Standard Deviation - Second Term

Calculate Sector Daily Returns

```
daily_returns_sector = daily_sector[daily_sector.columns[1:]].pct_change()
daily_returns_sector.plot(figsize=(15,7), title='Daily Returns')
```

Sector Daily Returns

Calculate Biden's Daily Returns Before, After The Election and the First 100 Days

```
daily_returns_biden_before = daily_returns_sector[daily_returns_sector.Date < '2020-11-03']
daily_returns_biden_after = daily_returns_sector[daily_returns_sector.Date > '2020-11-03']
```

```
daily_returns_biden_before[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Daily Returns Before Election')
```

Biden Daily Returns Before Election

```
daily_returns_biden_after[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Daily Returns After Election')
```

Biden Daily Returns After Election

```
daily_returns_biden = pd.concat([daily_returns_biden_after,daily_returns_biden_before], axis=0 ).sort_values(['Date', 'Date'])
daily_returns_biden.reset_index(drop=True,inplace=True)
daily_returns_biden.to_csv('daily_returns_biden.csv')
```

```
daily_returns_biden[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Daily Returns')
```

Biden Daily Returns

```
trump_mean = daily_returns[daily_returns.Date < '2020-11-03']
trump_mean.reset_index(drop=True,inplace=True)
trump_mean.to_csv('trump_mean.csv')
```

```
trump_mean[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Trump Mean Returns - Election')
```

Trump Mean Returns - Election

```
trump_100 = daily_returns[daily_returns.Date < '2021-02-03']
trump_100.reset_index(drop=True,inplace=True)
trump_100.to_csv('trump_100.csv')
```

```
trump_100[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Trump 100 Days')
```

Trump 100 Days

Calculate Obama's Daily Returns During the First & Second Election, The First 100 Days of First & Second Term

```
daily_returns_obama = daily_returns[daily_returns.Date < '2013-01-20']
daily_returns_obama.reset_index(drop=True,inplace=True)
daily_returns_obama.to_csv('daily_returns_obama.csv')
```

```
daily_returns_obama[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Obama Mean Returns - Election 1')
```

Obama Mean Returns - Election 1


```
obama_100 = daily_returns[daily_returns.Date < '2013-04-29']
obama_100.reset_index(drop=True,inplace=True)
obama_100.to_csv('obama_100.csv')
```

```
obama_100[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Obama First 100 Days - First Term')
```

Obama First 100 Days - First Term


```
daily_returns_obama = daily_returns[daily_returns.Date > '2013-01-20']
daily_returns_obama.reset_index(drop=True,inplace=True)
daily_returns_obama.to_csv('daily_returns_obama.csv')
```

```
daily_returns_obama[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Obama Mean Returns - Election 2')
```

Obama Mean Returns - Election 2


```
obama_100 = daily_returns[daily_returns.Date > '2013-04-29']
obama_100.reset_index(drop=True,inplace=True)
obama_100.to_csv('obama_100.csv')
```

```
obama_100[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Obama First 100 Days - Second Term')
```

Obama First 100 Days - Second Term

Calculate Bush's Daily Returns During the First & Second Election, The First 100 Days of First & Second Term

```
daily_returns_bush = daily_returns[daily_returns.Date < '2009-01-20']
daily_returns_bush.reset_index(drop=True,inplace=True)
daily_returns_bush.to_csv('daily_returns_bush.csv')
```

```
daily_returns_bush[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Bush Mean Returns - Election 1')
```

Bush Mean Returns - Election 1


```
bush_100 = daily_returns[daily_returns.Date < '2009-04-29']
bush_100.reset_index(drop=True,inplace=True)
bush_100.to_csv('bush_100.csv')
```

```
bush_100[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Bush First 100 Days - First Term')
```

Bush First 100 Days - First Term


```
daily_returns_bush = daily_returns[daily_returns.Date > '2009-01-20']
daily_returns_bush.reset_index(drop=True,inplace=True)
daily_returns_bush.to_csv('daily_returns_bush.csv')
```

```
daily_returns_bush[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Bush Mean Returns - Election 2')
```

Bush Mean Returns - Election 2


```
bush_100 = daily_returns[daily_returns.Date > '2009-04-29']
bush_100.reset_index(drop=True,inplace=True)
bush_100.to_csv('bush_100.csv')
```

```
bush_100[['S_P_Technology', 'S_P_Financial', 'S_P_Consumer_Cyclical', 'S_P_Consumer_Defensive', 'S_P_Energy']].plot(figsize=(15,7), title='Bush First 100 Days - Second Term')
```

Bush First 100 Days - Second Term