```
# Finance Data Project
# In this project we will focus on exploratory data analysis of stock prices. Keep in mind, this project is just meant to practice your visualization and
```

```
import pandas as pd
import numpy as np
jpq install pandas
jpq install numpy
import cufflinks as cf
from plotly.offline import download_plotlyjs,init_notebook_mode,plot,iplot
jpq install pandas_datareader
%% pd.options.mode.chained_assignment = None
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.graph_objs as go
```

```
Requirement already satisfied: plotly in c:\users\lenovo\anaconda3\lib\site-packages (5.6.0)
Requirement already satisfied: six in c:\users\lenovo\anaconda3\lib\site-packages (from plotly) (1.16.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\lenovo\anaconda3\lib\site-packages (from plotly) (8.0.1)
Requirement already satisfied: cufflinks in c:\users\lenovo\anaconda3\lib\site-packages (0.17.3)
Requirement already satisfied: numpy>=1.9.2 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.19.5)
Requirement already satisfied: pandas>=0.19.2 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.3.5)
Requirement already satisfied: scipy>=0.9.0 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.10.0)
Requirement already satisfied: setuptools>=34.4.1 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (61.2.0)
Requirement already satisfied: plotly>=4.1.1 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (5.6.0)
Requirement already satisfied: ipywidgets>=7.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (8.0.4)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\lenovo\anaconda3\lib\site-packages (from plotly>=4.1.1 cufflinks) (8.0.1)
Requirement already satisfied: traitlets>=4.3.1 in c:\users\lenovo\anaconda3\lib\site-packages (from ipywidgets>=7.0.0 cufflinks) (5.1.1)
Requirement already satisfied: widgetsnbextension~=4.0.5 in c:\users\lenovo\anaconda3\lib\site-packages (from ipywidgets>=7.0.0 cufflinks) (4.0.5)
Requirement already satisfied: jupyterlab_widgets~=3.0.7 in c:\users\lenovo\anaconda3\lib\site-packages (from ipywidgets>=7.0.0 cufflinks) (3.0.7)
Requirement already satisfied: ipykernel>=4.5.1 in c:\users\lenovo\anaconda3\lib\site-packages (from ipywidgets>=7.0.0 cufflinks) (6.19.2)
Requirement already satisfied: ipython>=6.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from ipywidgets>=7.0.0 cufflinks) (7.31.1)
Requirement already satisfied: jupyter-client>=6.1.12 in c:\users\lenovo\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0 cufflinks) (6.1.
12)
Requirement already satisfied: tornado>=6.1 in c:\users\lenovo\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0 cufflinks) (6.1)
Requirement already satisfied: nest-asyncio in c:\users\lenovo\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0 cufflinks) (1.5.5)
Requirement already satisfied: matplotlib-inline>=0.1 in c:\users\lenovo\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0 cufflinks)
(0.1.6)
Requirement already satisfied: debugpy>=1.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from ipykernel>=4.5.1->ipywidgets>=7.0.0 cufflinks) (1.6.
6)
Requirement already satisfied: jupyter-core>=4.6.0 in c:\users\lenovo\anaconda3\lib\site-packages (from jupyter-client>=6.1.12->ipykernel>=4.5.1->ipywidgets>=7.
0.0 cufflinks) (4.9.2)
Requirement already satisfied: pickleshare in c:\users\lenovo\anaconda3\lib\site-packages (from ipython>=6.0.0->ipywidgets>=7.0.0 cufflinks) (0.7.5)
Requirement already satisfied: prompt-toolkit>=2.0,<3.0,!=3.0.0,!=3.0.1 in c:\users\lenovo\anaconda3\lib\site-packages (from ipython>=6.0.0->ipywidgets>=7.0.
0 cufflinks) (3.0.36)
Requirement already satisfied: jedi>=0.16 in c:\users\lenovo\anaconda3\lib\site-packages (from ipython>=6.0.0->ipywidgets>=7.0.0 cufflinks) (0.18.1)
Requirement already satisfied: decorator in c:\users\lenovo\anaconda3\lib\site-packages (from ipython>=6.0.0->ipywidgets>=7.0.0 cufflinks) (5.1.1)
Requirement already satisfied: pickle-mixin>=1.0 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.0.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas>=0.19.2 cufflinks) (2022.7)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas>=0.19.2 cufflinks) (2.8.2)
Requirement already satisfied: ipyfsm>=0.0.1 in c:\users\lenovo\anaconda3\lib\site-packages (from jupyter-core>=4.6.0->jupyter-client>=6.1.12->ipykernel>=4.5.1
->ipywidgets>=7.0.0 cufflinks) (0.0.1)
Requirement already satisfied: pywin32>=1.0 in c:\users\lenovo\anaconda3\lib\site-packages (from jupyter-core>=4.6.0->jupyter-client>=6.1.12->ipykernel>=4.5.1
->ipywidgets>=7.0.0 cufflinks) (302)
Requirement already satisfied: parso<0.8.0,>=0.7.0 in c:\users\lenovo\anaconda3\lib\site-packages (from jedi>=0.16->ipywidgets>=7.0.0 cufflinks) (0.8.0)
Requirement already satisfied: wcwidth in c:\users\lenovo\anaconda3\lib\site-packages (from prompt-toolkit>=2.0,<3.0,!=3.0.0,!=3.0.1->ipywidgets>=7.0.0 c
ufflinks) (0.2.5)
Requirement already satisfied: pexpect>4.3 ; sys_platform != "win32" in c:\users\lenovo\anaconda3\lib\site-packages (from ipyfsm>=0.0.1->ipywidgets>=7.
0.0->ipywidgets>=7.0.0 cufflinks) (4.8.0)
Requirement already satisfied: ptyprocess>=0.5 in c:\users\lenovo\anaconda3\lib\site-packages (from pexpect>4.3 ; sys_platform != "win32"->ipyfsm>=0.0.1
->ipywidgets>=7.0.0 cufflinks) (0.7.0)
Requirement already satisfied: ipywidgets>=7.0.0 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (8.0.4)
Requirement already satisfied: ipyfsm>=0.0.1 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (0.0.1)
Requirement already satisfied: pickle-mixin>=1.0 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.0.2)
Requirement already satisfied: numpy>=1.9.2 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.19.5)
Requirement already satisfied: plotly>=4.1.1 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (5.6.0)
Requirement already satisfied: setuptools>=34.4.1 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (61.2.0)
Requirement already satisfied: scipy>=0.9.0 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.10.0)
Requirement already satisfied: pandas>=0.19.2 in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.3.5)
Requirement already satisfied: six in c:\users\lenovo\anaconda3\lib\site-packages (from cufflinks) (1.16.0)
```

```
jpq install pandas_datareader
```

```
Requirement already satisfied: pandas_datareader in c:\users\lenovo\anaconda3\lib\site-packages (0.10.0)
Requirement already satisfied: lxml in c:\users\lenovo\anaconda3\lib\site-packages (from pandas_datareader) (4.8.0)
Requirement already satisfied: pandas>=0.23.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas_datareader) (1.3.5)
Requirement already satisfied: requests>=2.19.0 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas_datareader) (2.27.1)
Requirement already satisfied: numpy>=1.17.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas>=0.23.3->pandas_datareader) (1.19.5)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas>=0.23.3->pandas_datareader) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in c:\users\lenovo\anaconda3\lib\site-packages (from pandas>=0.23.3->pandas_datareader) (2022.7)
Requirement already satisfied: urllib3>=1.26.9 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.19.0->pandas_datareader) (1.26.9)
Requirement already satisfied: charset-normalizer~=2.0.12 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.19.0->pandas_datareader) (2.0.1
2)
Requirement already satisfied: idna>=2.10 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.19.0->pandas_datareader) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\lenovo\anaconda3\lib\site-packages (from requests>=2.19.0->pandas_datareader) (2021.10.8)
Requirement already satisfied: six>=1.5 in c:\users\lenovo\anaconda3\lib\site-packages (from python-dateutil>=2.7.3->pandas>=0.23.3->pandas_datareader) (1.1
6)
```

```
from pandas_datareader import data,wb
import datetime
```

```
%% Get the data
BAC = pd.read_csv('BAC.csv',parse_dates=True,index_col=0)
MS = pd.read_csv('MS.csv',parse_dates=True,index_col=0)
```

```
BAC = data.DataReader("BAC", "yahoo", start, end)
C = data.DataReader("C", "yahoo", start, end)
GS = data.DataReader("GS", "yahoo", start, end)
JPM = data.DataReader("JPM", "yahoo", start, end)
MS = data.DataReader("MS", "yahoo", start, end)
WFC = data.DataReader("WFC", "yahoo", start, end)
```

```
%% Use the head or tail of the listed variables to get an idea of the data
```

```
%% Create a dataframe for the Close prices for each bank's stock listed in alphabetical order. Call this list: tickers**
bank_stocks = pd.concat([BAC,C,GS,JPM,MS,WFC],axis=1,keys=tickers)
bank_stocks.head()
```

```
                  BAC                                                   C  ...                                                  WFC
                 High       Low      Open     Close    Volume  Adj Close  ...      High       Low      Open     Close    Volume  Adj Close
Date                                                                      ...                                                           
2006-01-03  47.080000  46.160000  46.919998  47.080002  16296700.0  33.584065  ...  490.700000  481.100000  490.000000  490.800004  ...  57.240000  56.325001  57.070313  56.204461  75.582000  72.259094
2006-01-04  47.240002  46.400001  47.000000  46.820002  17776962.0  33.373596  ...  491.000000  483.500000  488.600006  488.799988  ...  58.799001  56.349998  58.799001  56.349426  74.759002  72.084284
2006-01-05  46.830002  46.430000  46.580002  46.650000  14364600.0  33.252316  ...  487.799988  484.399994  486.200012  486.399994  ...  58.249001  56.599998  57.799016  56.599376  75.420002  72.695
2006-01-06  46.910000  46.349998  46.799999  46.837500  12599800.0  33.385937  ...  489.600000  480.799988  489.200012  488.799988  ...  58.770000  56.570000  58.000000  56.982027  76.740000  73.970588
2006-01-09  46.970001  46.369001  46.755001  46.909998  15619400.0  33.436495  ...  487.899994  483.000000  486.000000  486.899994  ...  58.610001  56.189998  58.419001  56.760557  75.260673  76.245673

5 rows x 36 columns
```

```
%% Use the column name levels (this is filled out for you):**
bank_stocks.columns.levels # Bank Tickers,'Stock Info'
bank_stocks.head()
```

```
Bank Tickers                  BAC                                                   C  ...                                                  WFC
                 High       Low      Open     Close    Volume  Adj Close  ...      High       Low      Open     Close    Volume  Adj Close
Date                                                                      ...                                                           
2006-01-03  47.080000  46.160000  46.919998  47.080002  16296700.0  33.584065  ...  490.700000  481.100000  490.000000  490.800004  ...  57.240000  56.325001  57.070313  56.204461  75.582000  72.259094
2006-01-04  47.240002  46.400001  47.000000  46.820002  17776962.0  33.373596  ...  491.000000  483.500000  488.600006  488.799988  ...  58.799001  56.349998  58.799001  56.349426  74.759002  72.084284
2006-01-05  46.830002  46.430000  46.580002  46.650000  14364600.0  33.252316  ...  487.799988  484.399994  486.200012  486.399994  ...  58.249001  56.599998  57.799016  56.599376  75.420002  72.695
2006-01-06  46.910000  46.349998  46.799999  46.837500  12599800.0  33.385937  ...  489.600000  480.799988  489.200012  488.799988  ...  58.770000  56.570000  58.000000  56.982027  76.740000  73.970588
2006-01-09  46.970001  46.369001  46.755001  46.909998  15619400.0  33.436495  ...  487.899994  483.000000  486.000000  486.899994  ...  58.610001  56.189998  58.419001  56.760557  75.260673  76.245673

5 rows x 36 columns
```

```
%% Take in the max Close price for each bank's stock throughout the time period**
bank_stocks.cross_section('Close',axis=1).max().to_frame()
```

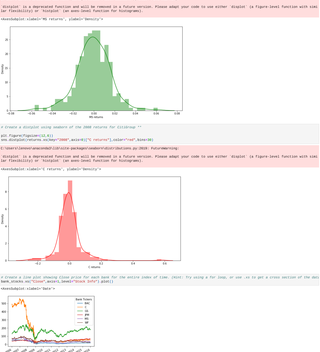
```
Bank Tickers
BAC    54.900002
C      564.000000
GS     247.919922
JPM    70.080002
MS     89.300003
WFC    58.984375
dtype: float64
```

```
%% Create a new empty DataFrame called returns. This dataframe will contain the returns for each bank's stock.

returns = pd.DataFrame()
```

```
%% Use to tickers:
for tick in tickers:
    returns[tick] = bank_stocks[tick]['Close'].pct_change()
returns.head()
```

```
                 BAC  Returns  C  Returns  GS  Returns  JPM  Returns  MS  Returns  WFC  Returns
Date                                                                                            
2006-01-03  0.000000  0.000000  0.000000  0.000000  0.000000  0.000000
2006-01-04  -0.005105  -0.014462  -0.013812  -0.014143  0.000000  0.003636
2006-01-05  -0.003199  -0.000000  -0.000000  -0.000000  0.000000  0.000000
2006-01-06  0.003851  0.000000  0.014286  0.007546  0.000000  0.000000
2006-01-09  0.001494  0.000000  0.010204  0.010102  0.000000  0.000000
```

```
%% Create a pairplot using seaborn of the returns dataframe. What stock stands out to you? Can you figure out why??

sns.pairplot(returns)
```




```
%% Using the returns dataframe, figure out on what dates each bank stock had the best and worst single day returns. You should notice that 4 of the banks share
```

```
returns.idxmax()
```

```
BAC returns    2009-04-09
C returns      2009-04-09
GS returns     2008-11-24
JPM returns    2009-01-21
MS returns     2008-10-13
WFC returns    2009-04-06
dtype: datetime64[ns]
```

```
returns.idxmin()
```

```
BAC returns    2009-01-20
C returns      2009-02-27
GS returns     2009-01-20
JPM returns    2009-01-20
MS returns     2008-10-09
WFC returns    2009-01-20
dtype: datetime64[ns]
```

```
%% Take a look at the standard deviation of the returns, which stock would you classify as the riskiest over the entire time period? Which would you classify a
```

```
returns.std(axis=0).sort_values()
```

```
BAC returns    0.015657
C returns      0.015836
JPM returns    0.016213
WFC returns    0.016957
GS returns     0.018481
MS returns     0.020405
dtype: float64
```

```
%% Create a distplot using seaborn of the 2008 returns for Citigroup **

plt.figure(figsize=(12,6))
sns.distplot(returns.ix['2008':,'C'],bins=50,color='green',kde=True)
```

```
C:\Users\lenovo\anaconda3\Lib\site-packages\seaborn\distributions.py:2619: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with sim
ilar flexibility) or `histplot` (an axes-level function for histograms).
```

```
```