Solution Design: Mini Project 4

How Do Markets React to Republicans and Democrats?

Version 1.0

Project Instructions:

1. Create a csv file with a list of all presidents, their parties from 1920 onwards

2. Using Pandas load the .csv file into a Pandas dataframe.

3. Download data from an appropriate financial website such as Google Finance, Yahoo Finance, Quandl, CityFALCON, or another similar source.

4. Calculate yearly returns for both the downloaded indices from 1920 onwards

5. Segregate returns in terms of Presidency – i.e. stock market returns during Democratic and Republican years

6. Calculate measures of central tendency (mean return, median return, variance of returns) for each of the two groups.

7. Represent the findings through suitable comparative graphical studies

Python Libraries Used:

The following libraries are used and can be installed using pip, e.g. “pip install pandas”

pandas 0.22.0 - see https://pandas.pydata.org/pandas-docs/stable/ for more information.

Urllib3 1.22 – see https://urllib3.readthedocs.io/en/latest/ for more information.

fix\_yahoo\_finance – see https://pypi.org/project/fix-yahoo-finance/#description

BeautifulSoup from bs4 – see https://www.crummy.com/software/BeautifulSoup/bs4/doc/

Code Classes and Methods:

I created a Market object(class) that impplements the following methods. The \_\_init\_\_() method is the class constructor where all the necessary variable were declared and intialized either as an empty string or empty list and or empty pandas DataFrame.

1, **get\_dow\_data(***self***)**

This method retrieve the DJIA index from yahoo finance using the fix\_yahoo\_finance library with yf.pdr\_override(). The download index time series is returned as a pandas DataFrame

2, **get\_snp500\_data(***self***)**

This method retrieve the S&P 500 index from yahoo finance using the fix\_yahoo\_finance library with yf.pdr\_override(). The download index time series is returned as a pandas DataFrame

3, **calc\_daily\_returns(***self***, pandas.DataFrame dataset)**

This method reads the Adj Close column of the pandas DataFrame to calculate daily returns over the period of time the downloaded data spread through. The calculated daily return is then returned as a list.

4, **get\_years(***self***, pandas.DataFrame dataset)**

In other to be able to map returns to presidency the year of the traded was decided and this method reads in the downloaded DataFrame after reseting the index using pandas DataFrame reset\_index() method to make the Date column of the DataFrame available to be accessed as a column also assigning a sequential column with a zero base index as the DataFrame frame index. This method also returns a list.

5, **do\_yearly\_returns(***self***, pandas.DataFrame dataset)**

This method receives the pandas DataFrame with the daily returns and year columns now added to it to calculates yearly returns that is returned as a list

6, **get\_party(***self***, pandas.DataFrame dataset, string lookup)**

The method take as input DataFrame made from csv by reading the presidents csv file with pandas and a string lookup the name of a political party to returns a list of dictionary containing the party name and the year for which the party was in power.

7, **split\_returns\_by\_presidency(***self***, pandas.DataFrame rets, pandas.DataFrame presidency)**

The method takes two pandas DataFrame as input, the list of yearly returns loaded into pandas DataFrame and the dictionary of presidency also loaded into pandas DataFrame, the presidency DataFrame is that of a political party at a time and the method returns the dictionary of yearly returns for the presidency of the political party queried for.

Solution Steps

**Step 1:**

I wrote a utility python script to scrap the list of presidents, their period of presidency and their political party from wikipedia page http://en.wikipedia.org/wiki/List\_of\_Presidents\_of\_the\_United\_States using the following library urllib3, bs4 from where the BeautifulSoup module was imported and used and pandas which enabled me to load the list of dictionary I generated into a pandas DataFrame from where I wrote it into excel using pandas.DataFrame.to\_excel() function and converted the excel file into a csv file after removing some data and columns not needed.

**Step 2:**

I then loaded the csv into pandas DataFrame using the pandas.read\_csv() method

**Step 3:**

I then headed on to yahoo finance to download the DOW JONES INDUSTRIAL AVERAGE index and the S&P 500 index using fix\_yahoo\_finance from within my code specifying startdate as 1920 and end date as today.

**Step 4:**

Using the **do\_yearly\_returns()** method I wrote then calculated the yearly returns. Before this I had to calculate daily returns using **calc\_daily\_returns**() method

**Step 5:**

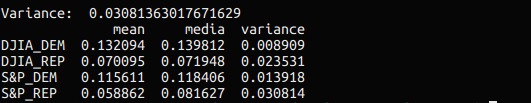
I then segregated the yearly returns by presidency of the two main political party under consideration using the **split\_returns\_by\_presidency()** method. I had to prepare the various input that this method will use to accomplish the purpose of the segregation process this is where the **get\_party()** and **get\_year()** methods comes in handy.

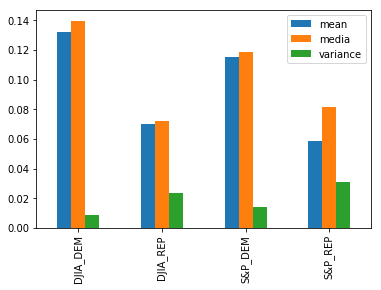
**Step 6:**

Now I have the datasets I need to calculate the measures of central tendency viz-a-viz mean return, median return, variance of returns for each of the two groups i.e. presidency by political party. I used the built-in method of the pandas DataFrame to find the measures. pandas.DataFrame.mean() from **mean,** pandad.DataFrame.median() for **median** and pandas.DataFrame.var() for variance.

**Step 7:**

I decided to use barchart for easy comparism of the measures of central tendencies calculated. So I made up a pandas DataFrame from them here is the DataFrame:



calling the pandas.DataFrame.plot.bar() function on the DataFrame displayed above gives the bar-chart below:

Using DEM – Democratics and REP – Republicans it could be observed from the bar-chart that yearly returns are higher during Democratics presidency than during Republicans presidency. The variance of the population during Republicans presidency is also higher than that of Democratics showing that during the presidency of Republicans the market is more volatile than that of the Democratics. The same showed pattern is noticed for both indexes under consideration and thus can be generalized to be the entire market reaction for both Republicans and Democratics

Tables:

