Project on Daily stock price of Facebook from 31-12-2014 to 05-02-2018

In [1]:
```
#Hello Everyone
#This is Umang!
#I have two datasets of historical stock data of Microsoft and Facebook
#I am going to analyse the data and view basic trends using python
```

In [2]:
```
#Pandas is used for data structure for time series data
import pandas as pd
import numpy as np
import scipy
from scipy import stats
from scipy.stats import norm
from math import sqrt
```

In [3]:
```
import matplotlib. pyplot as plt
%matplotlib inline
```

Analyzing Facebook Data

In [4]:
```
#Importing Facebook data, that we uploaded in our jupiter notebook
fb = pd.read_csv(r"C:\Users\umang\Desktop\facebook.csv")
```

In [5]:
```
#To obtain first five rows and check the heads of our data
fb.head()
```

Out[5]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume
0	31-12-2014	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500
1	02-01-2015	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000
2	05-01-2015	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800
3	06-01-2015	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100
4	07-01-2015	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200

In [6]:
```
#This data shows Opening price, closing price, highest price, lowest price, Adjust close with
#Lets attract some more basic details of our data in order to study it
fb.shape
```

Out[6]:
```
(780, 8)
```

In [7]:
```
fb.columns
```

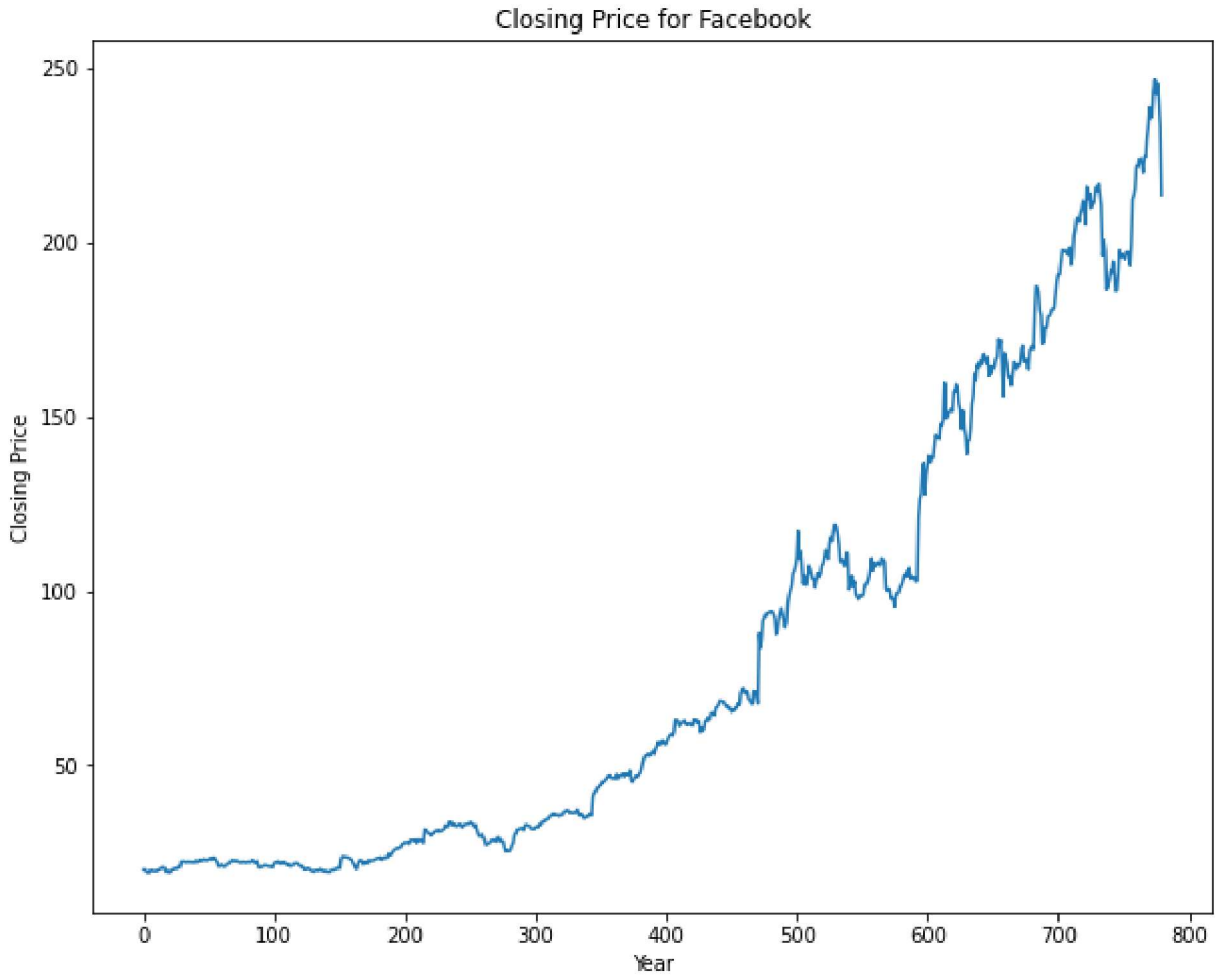
Out[7]:
```
Index(['Date', 'Year', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'], dtype='objec
t')
```

In [8]:
```
fb.describe()
```

Out[8]:

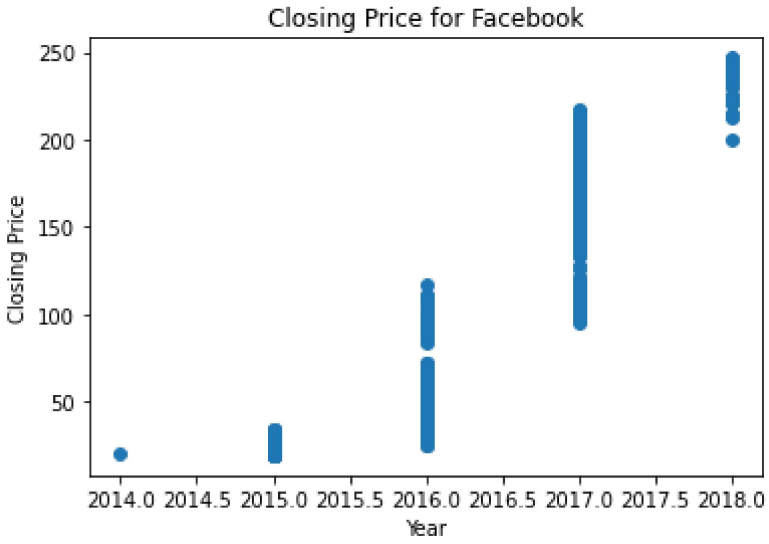
	Year	Open	High	Low	Close	Adj Close	Volume
count	780.000000	780.000000	780.000000	780.000000	780.000000	780.000000	7.800000e+02
mean	2016.057692	80.212705	81.285654	79.022397	80.264897	79.914215	1.204453e+07
std	0.877916	64.226121	65.048907	63.190963	64.198375	64.327846	8.221848e+06
min	2014.000000	19.250000	19.500000	18.940001	19.139999	18.576082	1.311200e+06
25%	2015.000000	25.525000	26.085000	24.845000	25.475000	25.134512	7.215200e+06
50%	2016.000000	53.379999	54.034999	52.930000	53.420000	53.035403	9.728700e+06
75%	2017.000000	113.322502	115.779999	110.297499	113.702501	113.261238	1.408885e+07
max	2018.000000	245.770004	249.270004	244.449997	246.850006	246.850006	9.232320e+07

In [9]:
```
#Plotting the closing price
plt.figure(figsize=(10,8))
fb['Close'].plot()
plt.xlabel("Year")
plt.ylabel("Closing Price")
plt.title("Closing Price for Facebook")
plt.show()
```



In [10]:
```
#We can notice from the graph that the growth of closing stock price was low till 2016. But i
```

In [11]:
```
#Plotting year wise scatter plot
plt.scatter(fb['Year'],fb['Close'])
plt.xlabel("Year")
plt.ylabel("Closing Price")
plt.title("Closing Price for Facebook")
plt.show()
```

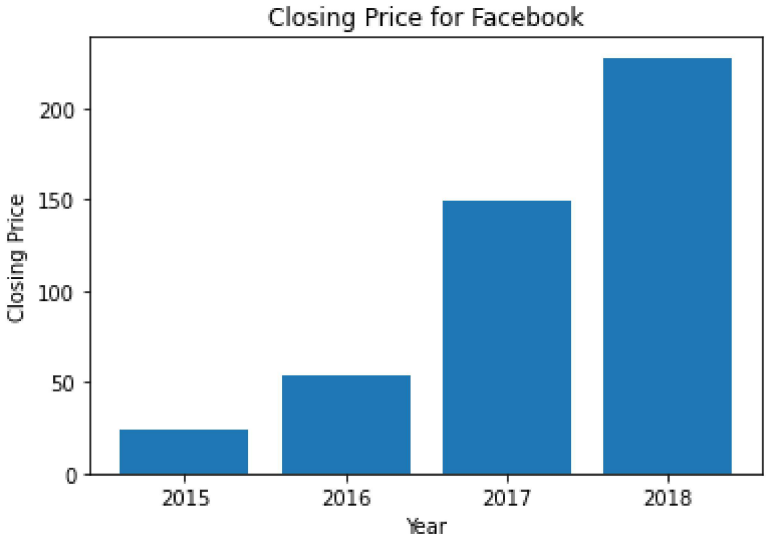


In [12]:
```
fb_2015 = fb.loc[fb.Year==2015]
mean_fb_2015 = fb_2015['Close'].mean()


fb_2016 = fb.loc[fb.Year==2016]
mean_fb_2016 = fb_2016['Close'].mean()


fb_2017 = fb.loc[fb.Year==2017]
mean_fb_2017 = fb_2017['Close'].mean()


fb_2018 = fb.loc[fb.Year==2018]
mean_fb_2018 = fb_2018['Close'].mean()
```

In [13]:
```
X = ["2015","2016","2017","2018"]
Y = [mean_fb_2015, mean_fb_2016, mean_fb_2017, mean_fb_2018]
```

In [14]:
```
plt.bar(X,Y)
plt.xlabel("Year")
plt.ylabel("Closing Price")
plt.title("Closing Price for Facebook")
```

Out[14]:
```
Text(0.5, 1.0, 'Closing Price for Facebook')
```



In [15]:
```
#Changing the date into datetime format
fb['Date'] = pd.to_datetime(fb.Date)
fb.head()
```

Out[15]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200

In [16]:
```
#creating a new column in dataframe for year
fb['Year'] = pd.DatetimeIndex(fb.Date).year
fb.head()
```

Out[16]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200

In [17]:
```
fb['Price1'] = fb['Close'].shift(-1)
fb.head()
```

Out[17]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500	20.129999
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000	19.790001
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800	19.190001
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100	19.139999
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200	19.860001

In [18]:
```
#creating new column of price difference (close price of tomorow - close price of today)
fb['PriceDiff']= fb['Price1'] - fb['Close']
fb.head()
```

Out[18]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500	20.129999	0.080000
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000	19.790001	-0.339998
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800	19.190001	-0.600000
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100	19.139999	-0.050002
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200	19.860001	0.720002

In [19]:
```
#creating a column of daily return
fb['Return'] = fb['PriceDiff']/fb['Close']
fb.head()
```

Out[19]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Retur
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500	20.129999	0.080000	0.00399
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000	19.790001	-0.339998	-0.01689
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800	19.190001	-0.600000	-0.03031
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100	19.139999	-0.050002	-0.00260
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200	19.860001	0.720002	0.03761

In [20]:
```
#Finding the direction of the price difference
fb['Direction'] = [1 if fb.loc[ei, 'PriceDiff']>0 else -1 for ei in fb.index]
fb.head()
```

Out[20]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Retur
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500	20.129999	0.080000	0.00399
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000	19.790001	-0.339998	-0.01689
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800	19.190001	-0.600000	-0.03031
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100	19.139999	-0.050002	-0.00260
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200	19.860001	0.720002	0.03761

In [21]:
```
#calculate moving average of close price over three days
fb['Aveerage3']= (fb['Close'] + fb['Close'].shift(1) + fb['Close'].shift(2))/3
fb.head()
```

Out[21]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Retur
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500	20.129999	0.080000	0.00399
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000	19.790001	-0.339998	-0.01689
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800	19.190001	-0.600000	-0.03031
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100	19.139999	-0.050002	-0.00260
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200	19.860001	0.720002	0.03761

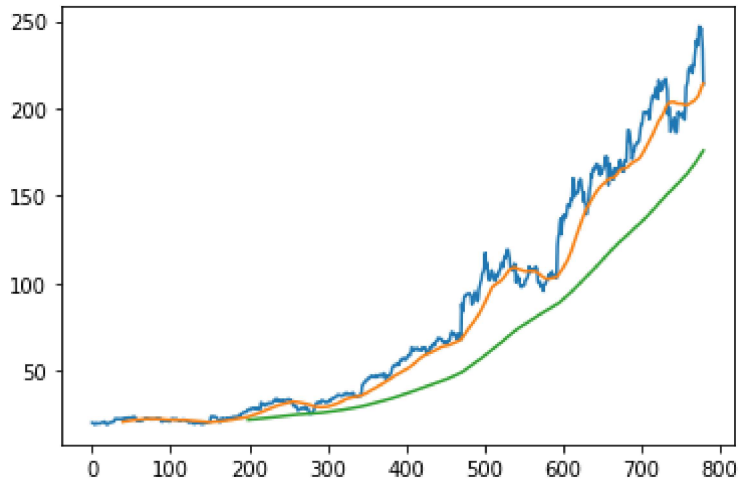
In [22]:
```
fb['MA40'] = fb['Close'].rolling(40).mean()
fb['MA200'] = fb['Close'].rolling(200).mean()
fb.head()
```

Out[22]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Retur
0	2014-12-31	2014	20.400000	20.510000	19.990000	20.049999	19.459270	4157500	20.129999	0.080000	0.00399
1	2015-02-01	2015	20.129999	20.280001	19.809999	20.129999	19.536913	2842000	19.790001	-0.339998	-0.01689
2	2015-05-01	2015	20.129999	20.190001	19.700001	19.790001	19.206934	4948800	19.190001	-0.600000	-0.03031
3	2015-06-01	2015	19.820000	19.840000	19.170000	19.190001	18.624611	4944100	19.139999	-0.050002	-0.00260
4	2015-07-01	2015	19.330000	19.500000	19.080000	19.139999	18.576082	8045200	19.860001	0.720002	0.03761

In [23]:
```
#MA40 is a fast signal for recent changes(short term) and MA200 is a slow signal for long ter
#when MA40>MA200 Traders beleive it that stock price will go upwards for a while
#ie when MA40>MA200 we buy and hold one share
fb['Close'].plot()
fb['MA40'].plot()
fb['MA200'].plot()
```

Out[23]:
```
<AxesSubplot:>
```



In [24]:
```
fb['MA40']= fb['Close'].rolling(40).mean()
fb['MA200']= fb['Close'].rolling(200).mean()
fb = fb.dropna()
```

In [25]:
```
#creating a column of shares
fb['Shares']=[1 if fb.loc[ei, 'MA40']> fb.loc[ei, 'MA200'] else 0 for ei in fb.index]
fb.head()
```

Out[25]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Re
199	2015-10-15	2015	27.410000	27.990000	27.129999	27.430000	26.961004	13151200	27.860001	0.430001	0.01
200	2015-10-16	2015	27.520000	28.120001	27.450001	27.860001	27.383654	11209900	27.809999	-0.050002	-0.00
201	2015-10-19	2015	27.950001	28.080000	27.490000	27.809999	27.334509	7367800	27.770000	-0.039999	-0.00
202	2015-10-20	2015	27.690001	27.900000	27.480000	27.770000	27.295193	6610700	27.410000	-0.360000	-0.01
203	2015-10-21	2015	27.940001	28.110001	27.360001	27.410000	26.941347	9657000	28.400000	0.990000	0.03

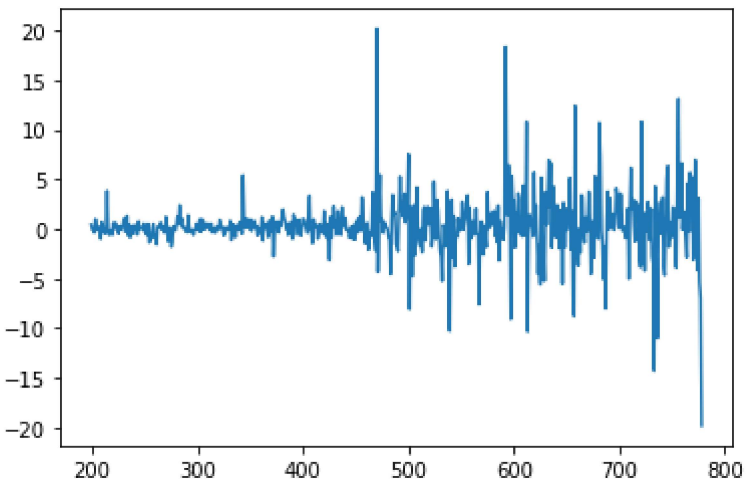
In [26]:
```
#we already have a new column of Price1 ie close price for tomorrow
#Now we create a column for profit
fb['Profit']= [fb.loc[ei,'Price1']-fb.loc[ei,'Close']
               if fb.loc[ei,'Shares']==1
               else 0 for ei in fb.index]
fb.head()
```

Out[26]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Re
199	2015-10-15	2015	27.410000	27.990000	27.129999	27.430000	26.961004	13151200	27.860001	0.430001	0.01
200	2015-10-16	2015	27.520000	28.120001	27.450001	27.860001	27.383654	11209900	27.809999	-0.050002	-0.00
201	2015-10-19	2015	27.950001	28.080000	27.490000	27.809999	27.334509	7367800	27.770000	-0.039999	-0.00
202	2015-10-20	2015	27.690001	27.900000	27.480000	27.770000	27.295193	6610700	27.410000	-0.360000	-0.01
203	2015-10-21	2015	27.940001	28.110001	27.360001	27.410000	26.941347	9657000	28.400000	0.990000	0.03

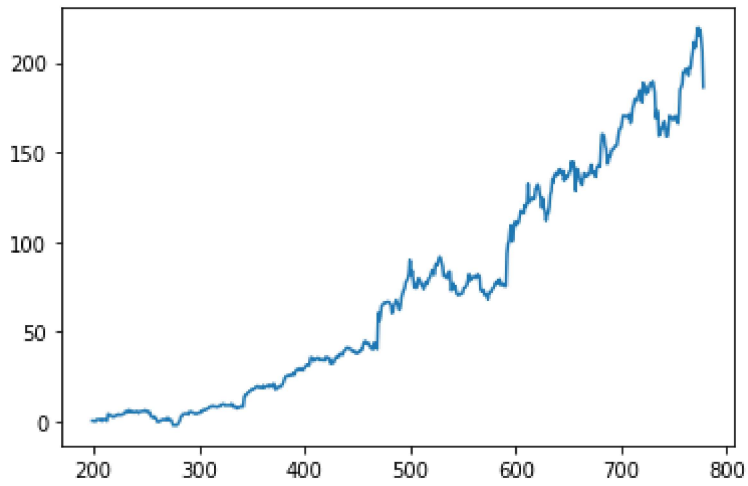
In [27]:
```
#Plotting the proft
fb['Profit'].plot()
```

Out[27]:
```
<AxesSubplot:>
```



In [28]:
```
#Finding the cumulative profit or wealth
fb['Wealth']= fb['Profit'].cumsum()
fb['Wealth'].plot()
```

Out[28]: <AxesSubplot:>



In [29]:
```
print("Total money win is ",fb.loc[fb.index[-2],'Wealth'])
```

```
Total money win is  206.090004
```

In [30]:
```
print("Total money spent is ",fb.loc[fb.index[0],'Close'])
```

```
Total money spent is  27.43
```

Estimating the Average stock return with 90% Confidence Interval

We use Log return

In [31]:
```
fb['Log_Return'] = np.log(fb['Close'].shift(-1)) - np.log(fb['Close'])
fb.head()
```

Out[31]:

	Date	Year	Open	High	Low	Close	Adj Close	Volume	Price1	PriceDiff	Re
199	2015-10-15	2015	27.410000	27.990000	27.129999	27.430000	26.961004	13151200	27.860001	0.430001	0.01
200	2015-10-16	2015	27.520000	28.120001	27.450001	27.860001	27.383654	11209900	27.809999	-0.050002	-0.00
201	2015-10-19	2015	27.950001	28.080000	27.490000	27.809999	27.334509	7367800	27.770000	-0.039999	-0.00
202	2015-10-20	2015	27.690001	27.900000	27.480000	27.770000	27.295193	6610700	27.410000	-0.360000	-0.01
203	2015-10-21	2015	27.940001	28.110001	27.360001	27.410000	26.941347	9657000	28.400000	0.990000	0.03

In [32]:
```
#Getting Sample size, Sample Mean, Sample standard deviation
sample_size = fb['Log_Return'].shape[0]
sample_mean = fb['Log_Return'].mean()
sample_std = fb['Log_Return'].std(ddof=1) / sample_size**5
z_score = 1.645

print("Sample Size =",sample_size)
print("Sample Mean =",sample_mean)
print("Sample Standard Deviation =",sample_std)
print("Z-Score =",z_score)
```

```
Sample Size = 580
Sample Mean = 0.003698843590199483
Sample Standard Deviation = 3.977703036945102e-16
Z-Score = 1.645
```

In [33]:
```
# Let Margin of Error be z_score ×sqrt(sample_mean×(1-sample_mean)/sample_size)
margin_of_error =z_score * sqrt(sample_mean*(1 - sample_mean)/sample_size)
margin_of_error
```

Out[33]:
```
0.004146486027229849
```

In [34]:
```
interval_left = sample_mean - margin_of_error
interval_right = sample_mean + margin_of_error
```

In [35]:
```
#90% confidence interval tells that there is 90% chance that average stock return lies betwee
print("90% confidence interval is ", (interval_left, interval_right))
```

```
90% confidence interval is  (-0.0004476424370303661 5, 0.007845329617429331)
```