**Data Project - Stock Market Analysis**

In this project we will be looking at data from the stock market, especially some technology stocks. Analysing the data, we try to answer the following questions:

1.) What was the change in price of the stock over time?  
2.) What was the daily return of the stock on average?  
3.) What was the moving average of the various stocks?  
4.) What was the correlation between different stocks' closing prices?  
4.) What was the correlation between different stocks' daily returns?  
5.) How much value do we put at risk by investing in a particular stock?  
6.) How can we attempt to predict future stock behavior?

We can use pandas to handle requesting stock information, and to analyze basic attributes of a stock.

* Along with the usual imports for visualization, time stamps, standard etc, we also import pandas\_datareader to read stock data from yahoo.
* Then, we create a list of tech stocks as tech\_list. And we also store the start and end dates in objects start and end.
* To get the yahoo finance data and storing it as a dataframe, we use a for loop. Using the globals() method to set the dataframe names. So each stock name in tech\_list will have a dataframe of its own.
* To get the summary stats for each stock:

<stock name>.describe()

* To get the general info, we use the <stock name>.info() method.
* Now, to answer the first question, we must plot out the volume and closing price of the stocks. We also plot the total volume of stock being traded each day over the past 5 years. This gives us the visualizations for the closing price and the volume traded each day.
* To answer the third question, we calculate the moving average for the stock. Pandas has a built in moving average calculator.It is the built in function .rolling\_mean()  and then, we plot all the additional moving averages.
* Now to analyze the risk of the stock, we take a closer look at the daily changes of the stock, and not just its absolute value. For that, we use use pandas to retrieve the daily returns for the stock.
* We use the distplot() method to create both a histogram and kde plot on the same figure to get an overall look at the average daily.
* To analyze the returns of all the stocks in our list, we build a DataFrame with all the ['Close'] columns for each of the stocks dataframes.
* Now as we have all the closing prices, we get the daily return for all the stocks using the pct\_change() method. This will help us compare the daily percentage return of two stocks to check how correlated they are.
* From this we notice that if two stocks are positively correlated with each other a linear relationship between its daily return values will be displayed.
* The Pearson product-moment correlation coefficient, will give us a sense of how correlated the daily percentage returns are.
* We use the built in seaborn method .pairplot() to display every possible combination of stocks in our technology stock ticker list.
* To investigate that individual comparison, we use sns.PairGrid() which gives us full control of the figure, including what kind of plots go in the diagonal, the upper triangle, and the lower triangle.
* Instead, we could also do a correlation plot, to get actual numerical values for the correlation between the stocks' daily return values. For this, we use the built in seaborn function .heatmap(). This gives us the strongest correlation of daily stock return. We can also see that all the technology comapnies are positively correlated.
* To find out the value which we put at risk by investing in a stock, we do risk analysis. One way is to use the information we've gathered on daily percentage returns is by comparing the expected return with the standard deviation of the daily returns.
* We define a value at risk parameter for our stocks. It is the amount of money we could expect to lose for a given confidence interval.
* One of the method we can use for estimating a value at risk, is “bootstrap” method. For this method we will calculate the empirical quantiles of the daily returns.
* To estimate the daily returns, we use the .distplot() method to plot the values to analyse it. The Nan values can’t be read by seaborn, so we use the .dropna() method.
* Then, we use the .quantile() method to get the risk value for the stock.
* From this we get the value at which the empirical quantile of daily returns is at.
* Another method to estimate daily returns, is the Monte Carlo method.
* The monte carlo method is running many trials with random market conditions, then we'll calculate portfolio losses for each trial. After this, we'll use the aggregation of all these simulations to establish how risky the stock is.
* We use Markov Process, also called geometric Brownian motion (GBM). It is given by the equation:

ΔS/S= μΔt + σϵ(Δt)^½

where,  S = stock price

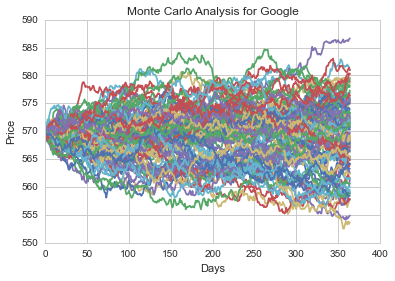
μ = the expected return

σ = the standard deviation of the returns

t = time

ϵ = the random variable

* We can mulitply both sides by the stock price (S) to rearrange the formula.
* We can see that the change in the stock price is the current stock price multiplied by two terms. The first term is "drift", which is the average daily return multiplied by the change of time. The second term is "shock", for each time period the stock will "drift" and then experience a "shock" which will randomly push the stock price up or down. By simulating this series of steps of drift and shock thousands of times, we can begin to do a simulation of where we might expect the stock price to be.
* Then, we create a function that takes in the starting price and number of days, and uses teh sigma and mu we already calculated form out daily returns. In this function, we define price, shock and drift arrays. And then, using a for loop we calculate the price, shock and drift. And then finally, return the price array.
* The final monte carlo plot looks like:



* Then, we create an array of simulations, which holds the end price data. We set the print options of numpy to only display 0-5 points from an array to suppress output, using .set\_printoptions() method.
* Then, using a  for loop, we set the simulation data point as the last stock price for that run. After we get the array of simulations, we can go ahead and plot a histogram and use quantile to define our risk for this stock.