Option 1: Stock Price Forecast and Risk Management in Finance using Monte Carlo Simulation

Group 14

12/13/2019

Contents

Group Information

- Title: Option 1: Stock Price Forecast and Risk Management in Finance using Monte Carlo Simulation and Variance Reduction Method
- Group Number: 14
- Group Members:
  - **Junbin, Park (jpark318)**
  - Jayhwan, Lee (jayhwan2)
  - Seulhui, Lee (seulhui2)
  - Jaemin, Kim (jkim557)
  - Taehoon, Lee (taehoon2)

Peer-Review Group Information

- Title: Option 1: Monte Carlo Methods on S&P ETF and it's Characteristics
- Group Number: 13
- Group Members:
  - **Jia Li (jial2)**
  - Maanav Ahluwalia (maanava2)
  - Ajay Dugar (dugar3)
  - Anoop Sypereddi (anoops2)
  - Yuqiu Li (yuquili2)

Relevant Sources

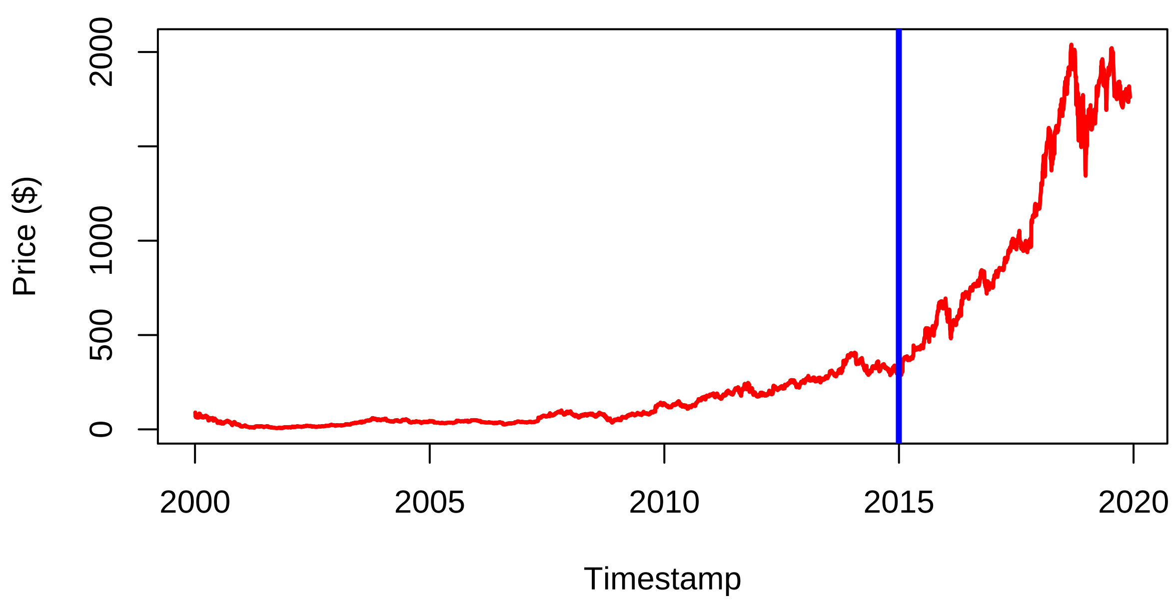
- Brodd, Tobias, and Adrian Djerf. "Monte Carlo Simulations of Stock Prices: Modelling the probability of future stock returns." (2018).
- Pažický, Martin. "Stock Price Simulation Using Bootstrap and Monte Carlo." Scientific Annals of Economics and Business64.2 (2017): 155-170.
- Siddiaui, Sehba Shahabuddin, and Vandana A. Patil. "Stock Market Valuation using Monte Carlo Simulation." 2018 International Conference on Current Trends towards Converging Technologies (ICCTCT). IEEE, 2018.

Abstract

Our group analyzes the underlying logics and theories behind the future stock price prediction. Our project is to predict the future stock price of Amazon using Monte Carlo simulation and random number generation. Our algorithm that utilizes Brownian Motion estimates expected level of return and volatility of the stock. The data were collected using API from Alpha Vantage. We modeled the simulations based on Markov Chain Monte Carlo random walk in stochastic process. The equation to forecast the next price follows Stochastic Differential Equation(SDE) that is incorporated into Brownian Motion that we implemented. As a result, the predictions we have made on Amazon correctly reflects our test data. It proves that the use of Monte Carlo simulation produces a valid prediction in real life application.

Introduction

In the tech-savviest era, statisticians began predicting the future stock price by means of computer. With the amount of computational power computers provide, it was made possible to run big and complicated codes to implement statistical methods that were just not possible in the past. Our group is going to analyze the underlying logics and theories behind the future stock price prediction. Our project aims to forecast future stock prices of Amazon using Monte Carlo. Amazon is an American multinational technology company that focuses on e-commerce. Hence, this analysis can demonstrate the real life application of Monte Carlo simulation by applying to Amazon. This simulation generates thousands of random walks based on stochastic model, Geometric Brownian Motion (GBM). Brownian motion is applied to the movement of the price of an asset by Louis Bachelier. It assumes that there are two parts to random movements: the driving force called the drift and the random amount of volatility that acts on the stock's price. Below graph displays historical data for Amazon stock price. In order to test whether results of analysis indeed reflect future price and correctly forecast risk, the original dataset was split into training and test data (blue line indicates split point), each containing data from 2000 to 2014 and 2015 to 12/4/2019.



Methods

The Simulation was designed to estimate the expected level of return and volatility of the stock. A return, in its simplest terms, is the money made or lost on an investment over some period of time. The volatility is a statistical measure of the dispersion of returns for a given security or market index. Subsections: - Data
The data are collected from the website Alpha Vantage. Since January, 2000 to December, 2014, our data present the close return of a day. We only take the data of the close return value of every business day into consideration. In order to show the correctness of our analysis, we are going to compare the training data (the data from 2000 to 2014) to the test data, which are the stock price since 2015 to present.
- Model
The method first estimates the expected level of return by the mean of the historical returns and the volatility of the stock by calculating the standard deviation of the historical returns. And then, the simulation is implemented by Markov Chain Monte Carlo of random walk. The MCMC approach to sampling from some density function f is to construct a Markov chain with stationary distribution f, and run the chain for a sufficiently long time until the chain converges (approximately) to its stationary distribution. The random walk is an example of MCMC, which allows to predict the next price based on the Stochastic Differential Equation (SDE). The next price is $S_{i-1}\ e(\mu - \sigma^2/2)dt + \sigma\sqrt{dt}\ N(0,1)$ We assume that the simulated prices follow a log-normal distribution.

The modeling method that we consider for our problem of interest is the Geometric Brownian Motion, which is a continuous-time stochastic process that satisfies the SDE, which is $dSt = \mu Stdt + \sigma StdBt$, where $St = a$ stochastic process and $B_t =$ Browninan Motion.

Each day, the price of an asset such as the stock is the previous day's price multiplied by the natural number (e) raised to the value of r, where r is the periodic rate of return. it is the rate that the asset is increased or decreased that day.
The rate of return on the asset is a random number. To model the movement and determine possible future values, we must use a formula that models random movements. Brownian motion is a formula used to model random movements in physics, and was first applied to the movement of the price of an asset by Louis Bachelier. The work was expanded on and eventually became the core of many areas in finance.
Brownian motion assumes that there are two parts to random movements. The first part is an overall constant driving force called the drift, and the second is the random component. Therefore, the rate that the asset changes in value each day can be broken down into two parts: The overall drift and a random stochastic component.
For the drift, we use the expected rate of return. In other words, we use the rate that we expect the price to change each day. The expected rate is the rate of change with the greatest odds of occurring. However it is not the rate that the price will change each day. The rate change is an unknown random number. The asset can increase or decrease in any random rate, but the Central Limit Theorem is statistics tells us that we can assume the periodic daily rates of return will be normally distributed. Which means that if we graph enough of the periodic daily returns, the graph is likely to follow a normal distribution. However, in reality though, this assumption is not technically true. It can only be adopted to modeling purposes.
This means that we can assume that the rates of the daily change in the future will also be normally distributed. In other words, the graph of possible future periodic daily rates of return will also follow a normal distribution. The stochastic process X_t follows the geometric brownian motion, and satisfies the following stochastic differential equation:

$$
\begin{aligned}
dX_t &= \mu X_t dt + \sigma X_t dB_t \ \rightarrow dX_t^2 = \sigma^2 X_t^2 dt \\
&Set\ up \\
Y_t &= lnX_t \\
dY_t &= d\ lnX_t \\
&= \frac{dX_t}{X_t} - \frac{1}{2}\frac{dX_t^2}{X_t^2} \\
d\ lnX_t &= \mu dt + \sigma dB_t - \frac{1}{2}\sigma^2 dt \\
d\ lnX_t &= (\mu - \frac{\sigma^2}{2})dt + \sigma dB_t \\
\int_0^T d\ lnX_t &= \int_0^T (\mu - \frac{\sigma^2}{2})dt + \int_0^T \sigma dB_t \\
ln\ X_T - ln\ X_0 &= (\mu - \frac{\sigma^2}{2})T + \sigma B_T \\
X_T &= X_0\ e^{(\mu - \frac{\sigma^2}{2})T + \sigma B_T} \\
X_{T+S} &= X_T\ e^{(\mu - \frac{\sigma^2}{2})T + \sigma(B_{T+S} - B_T)}
\end{aligned}
$$

Where B_t is a standard Brownian motion, which is a stochastic process, indexed by nonnegative real numbers t, following properties:
(1) $B_0 = 0$.
(2) With probability 1, the function $t \ \rightarrow \ B_t$ is continuous in t.
(3) The process $\{B_t\}_{t \geq 0}$ has stationary, independent increments.
(4) The increment $B_{t+s} - B_s$ has the $Normal\ (0, t)$ distribution.
Therefore,

$$X_{T+1} = X_T\ e^{(\mu - \frac{\sigma^2}{2})T + \sigma B_1}$$

Brownian motion applied through an asset, the drift which is the historical average of the periodic daily returns eroded by volatility at the rate of half the variance over time. But the drift has a rounded part that combines with the drift to give an actual rate of return that is normally distributed. In short, Brownian motion means that if we graph the future periodic daily returns, we assume that the graph will form a normal distribution bell-shaped curve using the drift as the mean and using the historical deviation as the future standard deviation

- Algorithm

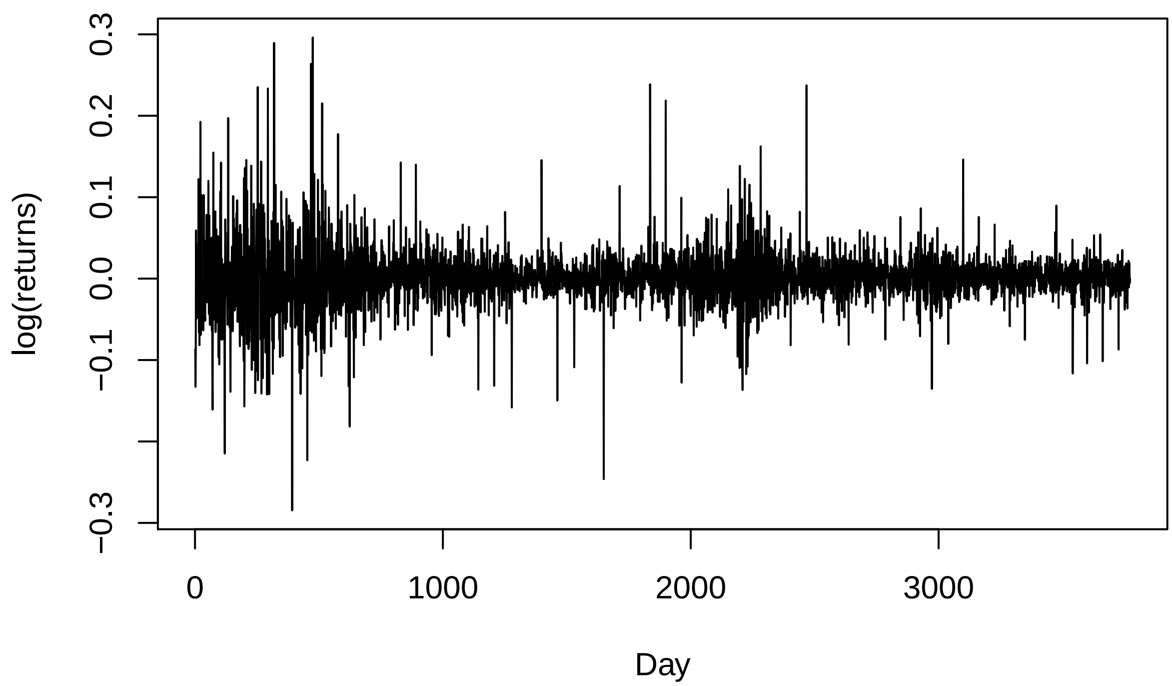
```
# Log return of the simulated price
log_returns=log(1+(diff(temp_train$close)/temp_train[-nrow(temp_train),2]))

# The GBM equation set up: the drift and the shock
for(i in 1:iterations){
  daily_returns[i,]= exp(drift+stdev*qnorm(runif(t_intervals)))
}

# Markov Chain MC of random walk
for(i in 2:t_intervals){
  for(j in 1:iterations){
    price_list[i,j]=price_list[i-1,j]*daily_returns[j,i]
  }
}
```

Results

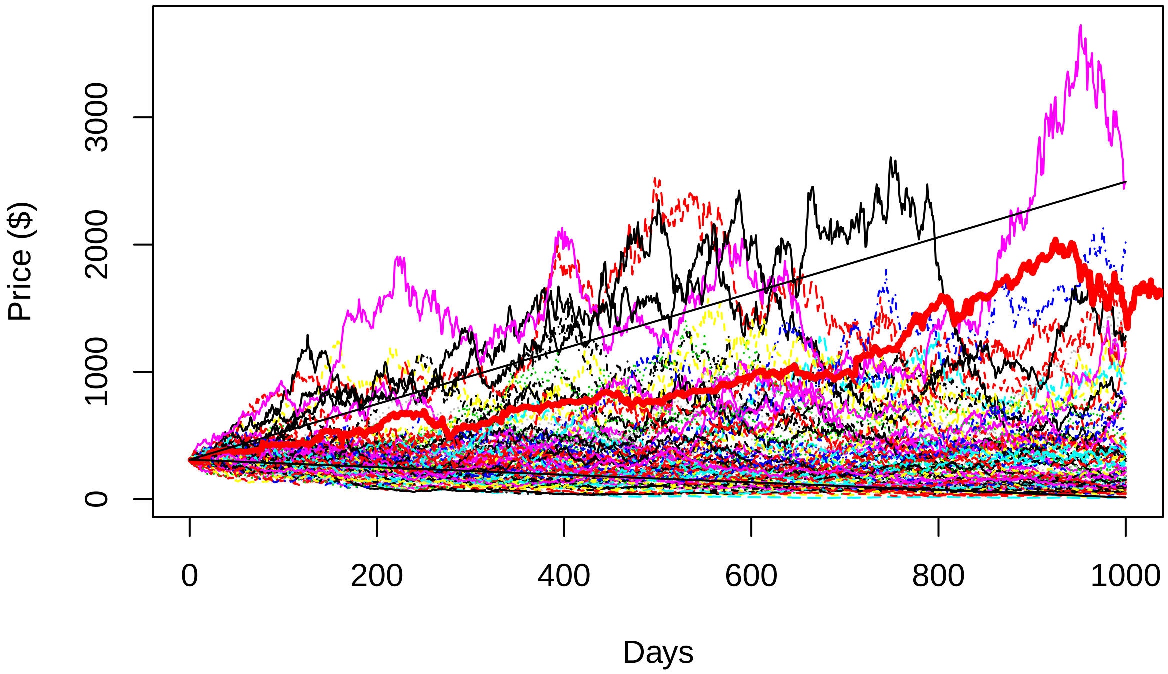
There were several outcomes produced while conducting our stock prediction analysis. In order to implement Brownina Motion, as mentioned above, we had to find daily return values of stock price. The daily return value includes Wiener Process, and I had to obtain the log return value of price differences between X_{t-1} and X_t. Below graph shows the resulting log return values. It can be observed that the log_returns indicate daily price fluctuation centered around price from the previous day. In early 2000s, the fluctations, or differences, in prices were greater compared to any other period in our time frame. As time passed, the increasing and decreasing interval of price reduced and displayed stable trend during later period of time.



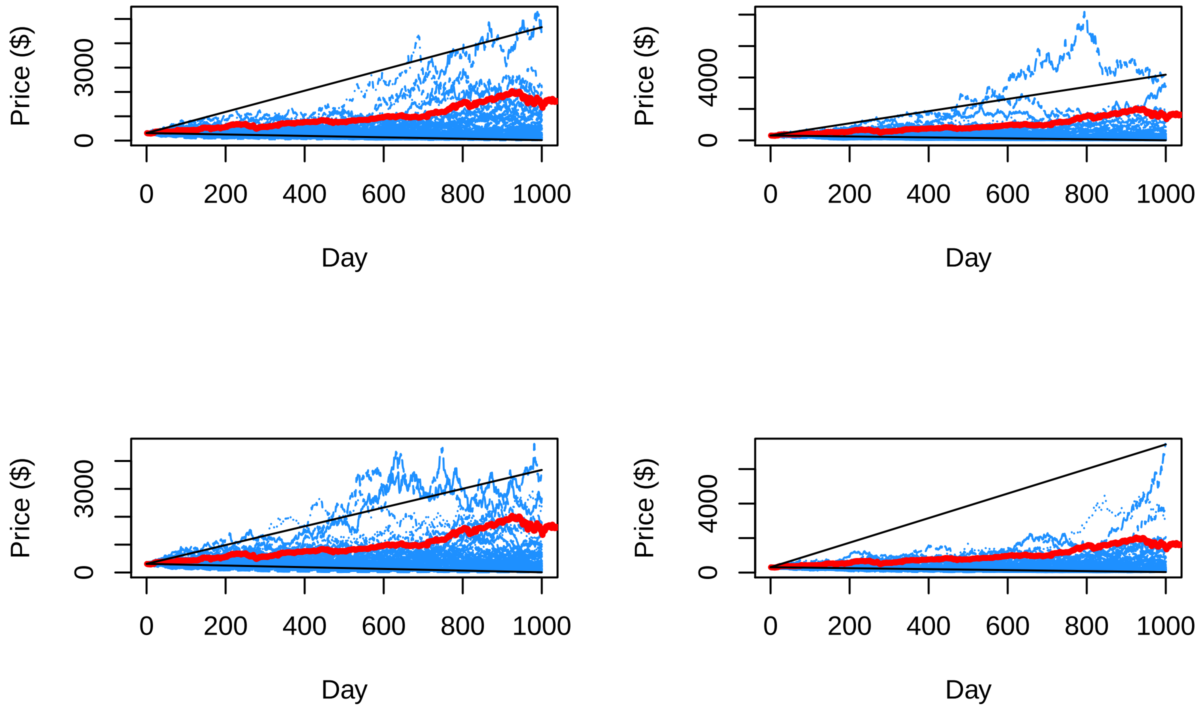
With the log return values, daily return values were calculated based on the Wiener Process in Brownian Motion that is expressed as the following equation:

$$e^{(\mu-\frac{\sigma^2}{2})T+\sigma B_1}$$

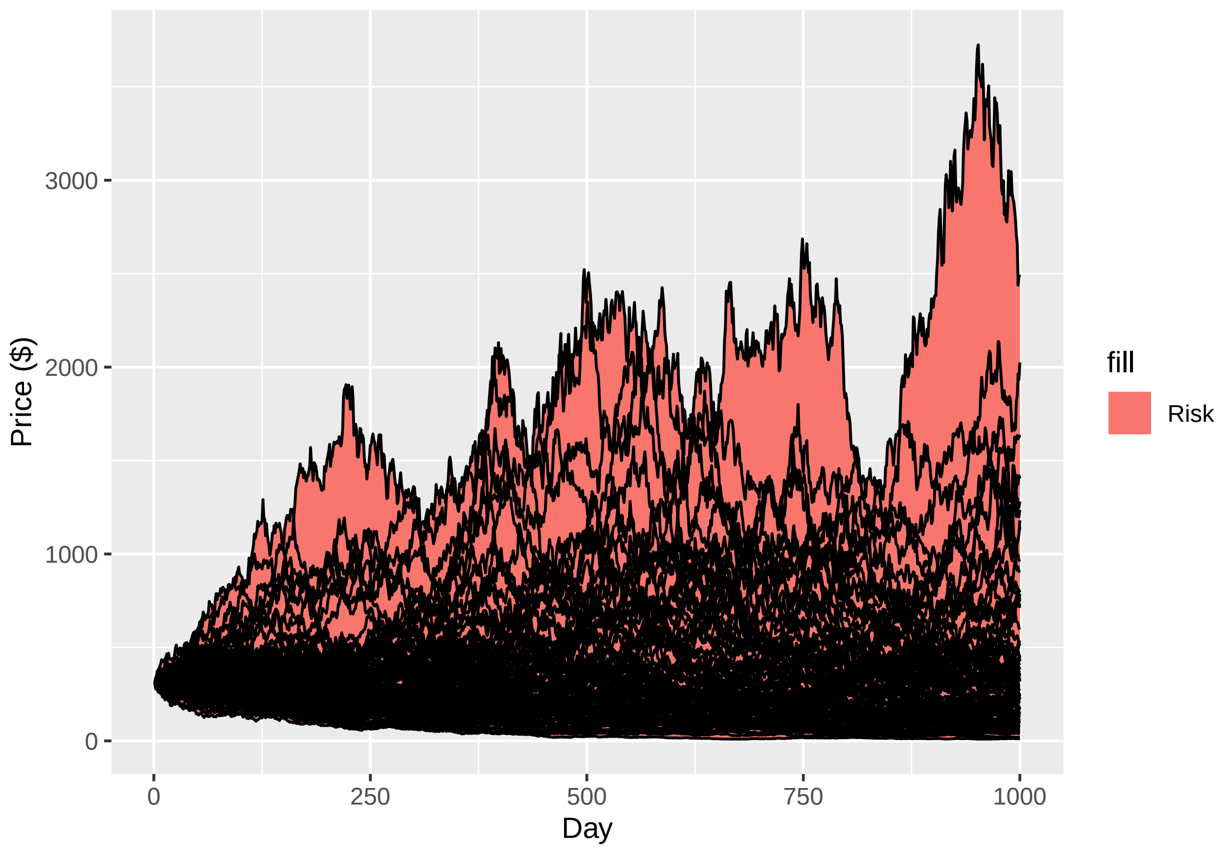
This daily return values were then multiplied with the prvious stock price for predicting X_t, that reflects the drift and volatility of past data. 100 simulations, or iterations, were performed over 1000 days after the last day(12/31/2014) of training dataset, where two linear line segments connect minimum and maximum values of the initial point (Day 1) to minimum and maximum values of the last day (Day 1000) in our time interval. The resulting simulations were then compared with the true value, the actual price of Amazon in test dataset.



To see how outcomes vary everytime the code is run due to the effect of random number generation within normal distribution in Monte Carlo simulation, results of multiple runs of Brownian Motion are tested. Blue lines indicate simulations, and bold red line indicates the actual stock price in test data.



This outcome displays the risk range of Amazon stock price forecast with ggplot function in R to combine simulation line graphs using geom_line() and risk area using geom_ribbon(). The graph shows simulations in black line and the range of maximum price and minimum price as risk, and it shows the variance among simulations of Brownian Motion random walk stochastic process.



Discussion

The results have disclosed that the prediction we made on the stock price of the training data matches the actual data (the test data). Under high level of uncertainty and randomness, it produces a sound outcome that can be applied to other stocks. The results of both Apple and Amazon explicitly represent in the graphs that the stock price is going to be appreciated. Some other data would help add more credibility to our methods. For example, the stock price of the other Big Four tech companies: Google and Facebook. Since we take only two of the Big Four companies, Apple and Amazon, due to the limited time, comparing all the Big Four companies can provide the higher accuracy if compared. In addition, comparing the data of the Big Four companies to the trend of the stocks listed in the S&P 500 also increases the accuracy of our prediction. S&P 500. S&P 500 is a stock market index that measures the stock performance of 500 large companies listed on stock exchanges in the US. There are some interesting observations from our analysis. First, the graphs show the results that we expected before simulation. Also, not all the simulations produce ideal results. During the simulation, some outlier trend emerges. It completely goes off the boundary that we expect as a prediction interval. In conclusion, the analysis we have performed has generated valid predictions using random number generation and Monte Carlo simulations. The methods we employ can lay the cornerstone of stock price prediction in real life.

Appendix

```
temp=amzn[20:nrow(amzn),c(1,5)] #20:nrow(amzn) 1/3/2000 (First day of 2000)

#Split train and test
#train
temp_train=amzn[20:3792, c(1,5)] #3792 12/31/2014 (Last day of 2014)
#test
temp_test=amzn[3793:nrow(amzn),c(1,5)]

log_returns=log(1+(diff(temp_train$close)/temp_train[-nrow(temp_train),2]))
log_returns=unlist(log_returns)

u=mean(log_returns)
v=var(log_returns)
drift=u-(0.5*v)
stdev=sd(log_returns)

#Set time interval and Number of Simulations
t_intervals=1000
iterations=100

daily_returns <- as.data.frame(matrix(0, ncol = t_intervals, nrow = iterations ))
for(i in 1:iterations){
  daily_returns[i,]= exp(drift+stdev*qnorm(runif(t_intervals)))
}

#Prediction
#Start with last price of train dataset
S0=temp_train[nrow(temp_train),2]
price_list <- as.data.frame(matrix(0, ncol = iterations, nrow = t_intervals))
price_list[1,]=S0
for(i in 2:t_intervals){
  for(j in 1:iterations){
    price_list[i,j]=price_list[i-1,j]*daily_returns[j,i]
  }
}

#plots
library(ggplot2)
library(MASS)
library(reshape2)
d=ggplot(price_list, aes(price_list))
ggplot_temp=cbind(price_list, id=c(1:nrow(price_list)))
meltR = melt(ggplot_temp, id="id")
x=c(1:1000)
y_mins=numeric(1000)
for(i in 1:1000){
  y_mins[i]=min(meltR$value[meltR$id==i])
}
y_maxs=numeric(1000)
for(i in 1:1000){
  y_maxs[i]=max(meltR$value[meltR$id==i])
}
```

```
ggplot(meltR, aes(x=id, y = value, group = variable, colour = variable)) +
  geom_line(aes(x=id, y = value, group = variable, colour = variable))

ggplot(meltR, aes(x=id, y = value, group = variable, colour = variable)) +
  geom_ribbon(aes(ymin= rep(y_mins,iterations),
                  ymax= rep(y_maxs,iterations),
                  fill="blue",alpha=0.5)) +
  geom_line(aes(x=id, y = value, group = variable, colour = variable))
```

References

http://web.math.ku.dk/~rolf/CT_FinOpt.pdf
https://uu.diva-portal.org/smash/get/diva2:1218088/FULLTEXT01.pdf
http://www.math.ucsd.edu/~politis/PAPER/BPIforARjspiFINAL.pdf
https://en.wikipedia.org/wiki/Wiener_process
https://en.wikipedia.org/wiki/Geometric_Brownian_motion
https://en.wikipedia.org/wiki/Autoregressive_conditional_heteroskedasticity
https://datascienceplus.com/how-to-apply-monte-carlo-simulation-to-forecast-stock-prices-using-python/
https://beta.vu.nl/nl/Images/werkstuk-dmouj_tcm235-91341.pdf
http://www.math.unl.edu/~sdunbar1/MathematicalFinance/Lessons/BrownianMotion/Definition/definition.pdf
https://ocw.mit.edu/courses/sloan-school-of-management/15-070j-advanced-stochastic-processes-fall-2013/lecture-notes/MIT15_070JF13_Lec6.pdf
https://www.stat.berkeley.edu/~aldous/Research/Ugrad/ZY1.pdf
https://iopscience.iop.org/article/10.1088/1742-6596/974/1/012047/pdf