Apple share Value at Risk(vaR) Estimation using Monte Carlo Simulation

Hans Peter Ndeffo

Professor - Jonathan Natov

May 25, 2020

Abstract

In this project we aim at calculating Apple VaR(Value at Risk) for 1 share using Monte Carlo Simulation with the following:

- Apple yearly expected return μ is set to **37.7%**.
- The standart deviation over a year σ is 2.16.
- The risk free rate for Apple is **0.71%**
- The time period for vaR is **t =** $\frac{7}{252}$ which is a week in trading days.
- The 95 % confidence vaR can be calculated using the quantile function which is used in R to determine ***nth*** percentile.

Introduction

We live in an era where technology can more than ever, help us to compute and train complex model to predict stock prices. There are different techniques such as recurent neural network which focus on understading trends and behavior of stocks. However, the results you will obtain, depend on the stock(apple, microsoft), time(can be in seconds, days or months) and many other factors. In this analysis, we will focus our attention only to have a rough estimate of the closing prices using an interval of 10 days in the past and in the future.

Step 1: load the required libraries

```
# Load libraries
# ggplot2 to plot graphs

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

Step 2: Defining the variables

Expected return exp

```
exp_return = 0.377
```

Standart deviation std is

```
std = 2.16
```

Time period

```
t = 7/252
```

Risk free rate

```
r = 0.0071
```

Step 3: Defining the function use to simulate stock prices over a time period t

Function “model” that simulates stock prices St over t period Accepts: - n = number of steps use over time period t

- model.run = number of simulation run
- model.S0 = initial stock price input
- model.exp_return = expected return input into the function for the stock
- model.t = time period
- model.std = standart deviation

The model follows the equation $S_{t+1} = S_t + S_t \mu \delta t + S_t \sigma \epsilon \sqrt(\delta t) = S_t(1 + \mu \delta t + \sigma \epsilon \sqrt(\delta t)$

```
model = function(n, model.run, model.t, model.S0, model.exp_return,

  # Reset St
  St = c()
  # deltaT
  deltaT <<- model.t/max(n, model.run)
  # Steps
  steps <<- seq(from = deltaT, to= model.t, by= deltaT)
  seeds = sample(1:100000, model.run, replace = FALSE)

  for(k in 1:model.run){

    # Reset current St
    cur_St = c(model.S0)
    # current seed
    set.seed(seeds[k])
    # Generating n sample using standart normal distribution
    sample = rnorm(n, mean= 0, sd= 1)

    for(i in 1:n- 1){
      cur_St = c(cur_St, cur_St[length(cur_St)]+
                 cur_St[length(cur_St)]*((model.exp_return-r)*deltaT+
    }
    St = c(St, cur_St[length(cur_St)])
  }
```

```
  return (St)
}
```

Step 4: Run the model with the parameters from step 2

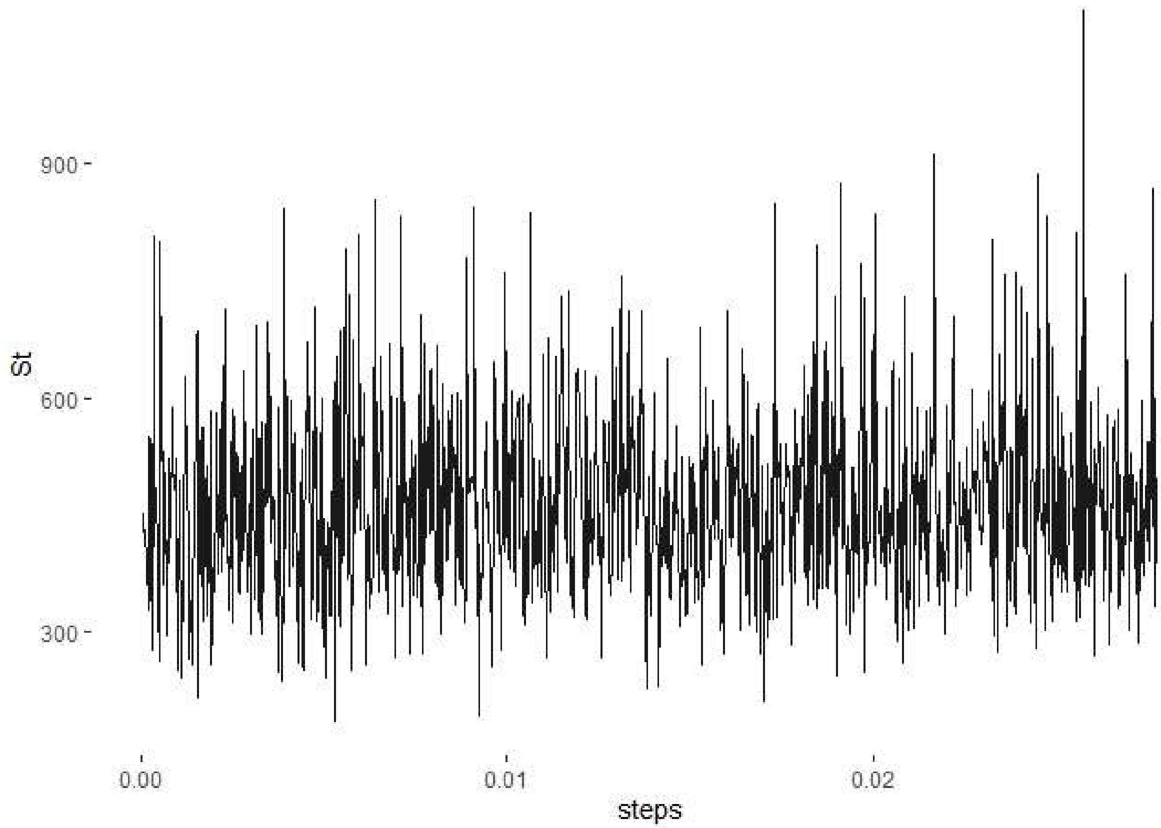
```
St = model(n = 500, model.t= t, model.run  = 1000, model.S0 = 460, m
```

Plot 1000 stock prices simulated by the model at the end of each 7/252 trading days period with timesteps = $\frac{7/252}{500}$

```
model_result <- as.data.frame(steps)
model_result <- cbind(model_result, St)

ggplot(model_result, aes(x = steps, y = St))+
  geom_line(stat = "summary", colour = "black", alpha= 0.9)+
  theme(panel.background = element_rect("white"))
```

```
## No summary function supplied, defaulting to `mean_se()`
```



Step 5: Computing vaR with 95% confidence

```
# S0 is the initial stock price
S0 = 460
vaR = S0 - quantile(sort(St),0.05)
vaR
```

```
##       5%
## 161.1418
```

Conclusion

In this analysis, we found with **95% confidence** that vaR is 157.95 USD over a week period. So if you buy one apple share you could be losing up to 157.95 USD over a 7 trading days period.