

Value at Risk

R markdown output file of VaR calculations

Jesse Keränen

20/11/2020

Prologue

In this document I calculate value at risk estimates for a portfolio formed of given assets. Some calculations could might have been done easier using some libraries, but I have tried, within my R skills, to avoid using too much libraries. Value at risk (VaR) is risk measurement used for example in finance. VaR tells investor what is the greatest loss investor will face with given probability.

Data collection

Download adjusted prices of the stocks and adjust data. Data is downloaded from Nasdaq's Quandl data base. For each asset adjusted closing price is selected, so that there is no need to take for example dividends into consideration.

```
library(ggplot2)
library(ggplot2)
library(pheatmap)
library(Quandl)
library(tidyverse)
library(data.table)

Quandl.api_key("bx1qdehfWXg6SNKnicQC")

names <- c("AAPL", "MSFT", "TSLA", "GOOGL", "AMZN")

# For monthly data use collapse = "monthly"
prices <- as.data.table(Quandl(c("WIKI/AAPL.11", "WIKI/MSFT.11",
  "WIKI/TSLA.11", "WIKI/GOOGL.11", "WIKI/AMZN.11"), start_date
  = "2010-07-01", end_date = "2015-02-01"))

colnames(prices) <- c("Date", "AAPL", "MSFT", "TSLA", "GOOGL", "AMZN")

# Tidying data table
prices <- melt(prices, id.vars = "Date", measure.vars = names,
  variable.name = "Company", value.name = "Adj_Close")

# Calculate price changes
prices[, Returns := Adj_Close/shift(Adj_Close) -1, by = Company]
```

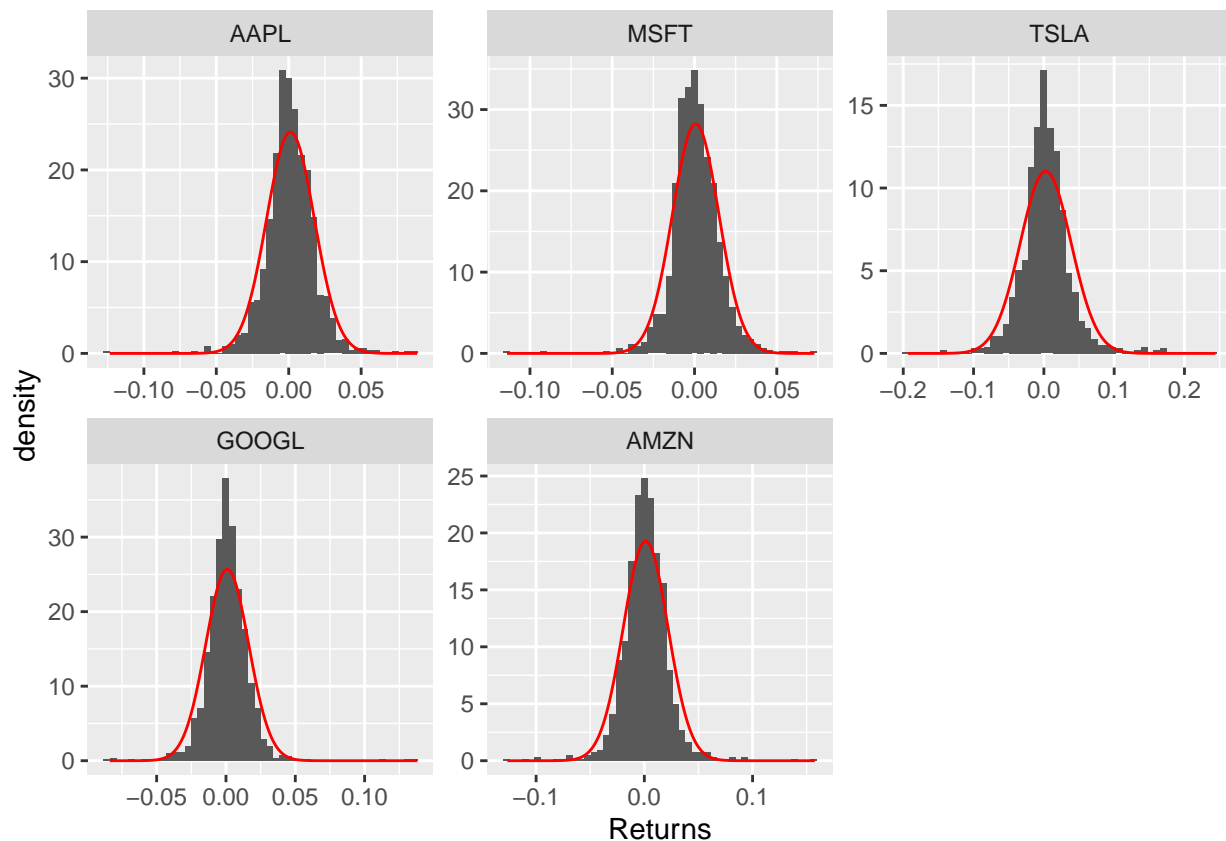
```
prices <- drop_na(prices)
```

Data visualisation

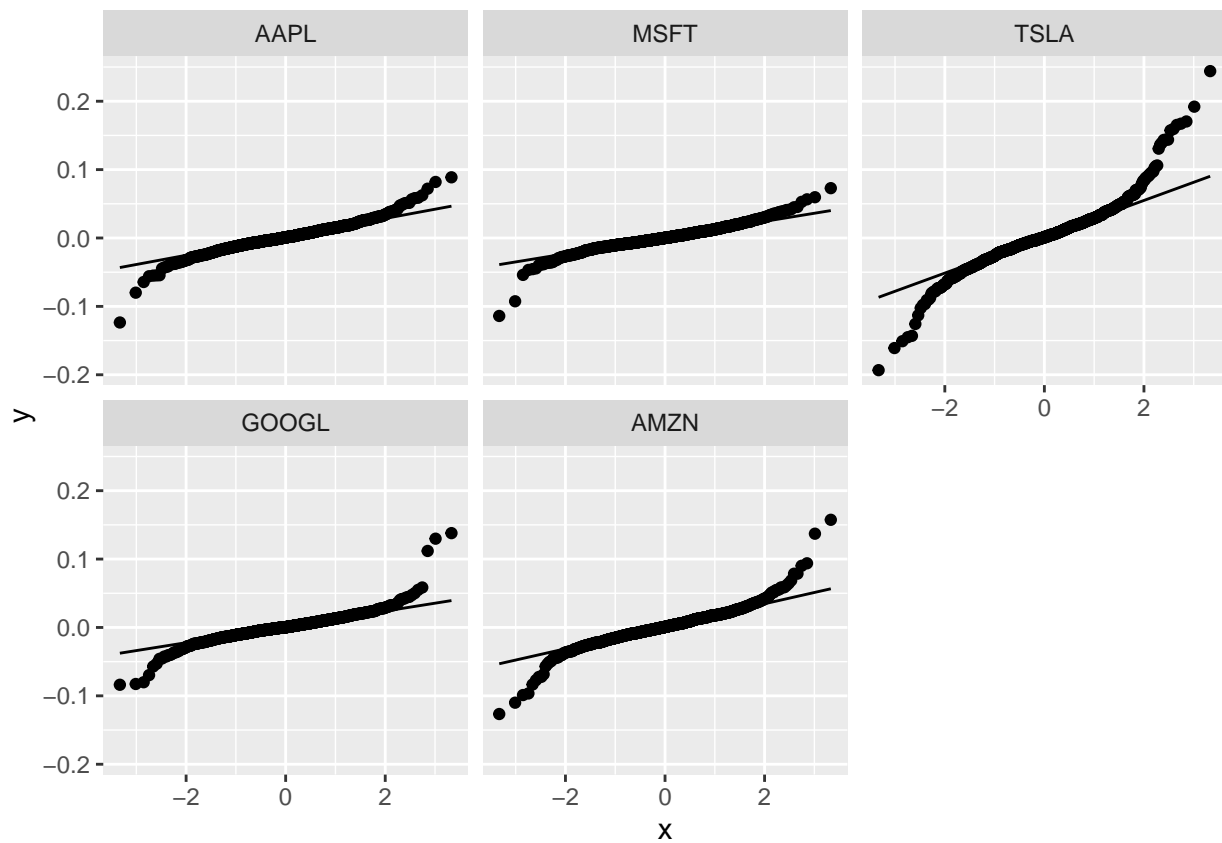
First densities of returns of each asset are plotted. Then normal distribution curve is plotted above to show how normally returns are distributed. R doesn't seem to offer method for plotting normal distribution curve when using `facet_wrap`, so this is done by using "ggh4x" library.

```
library(ggh4x)

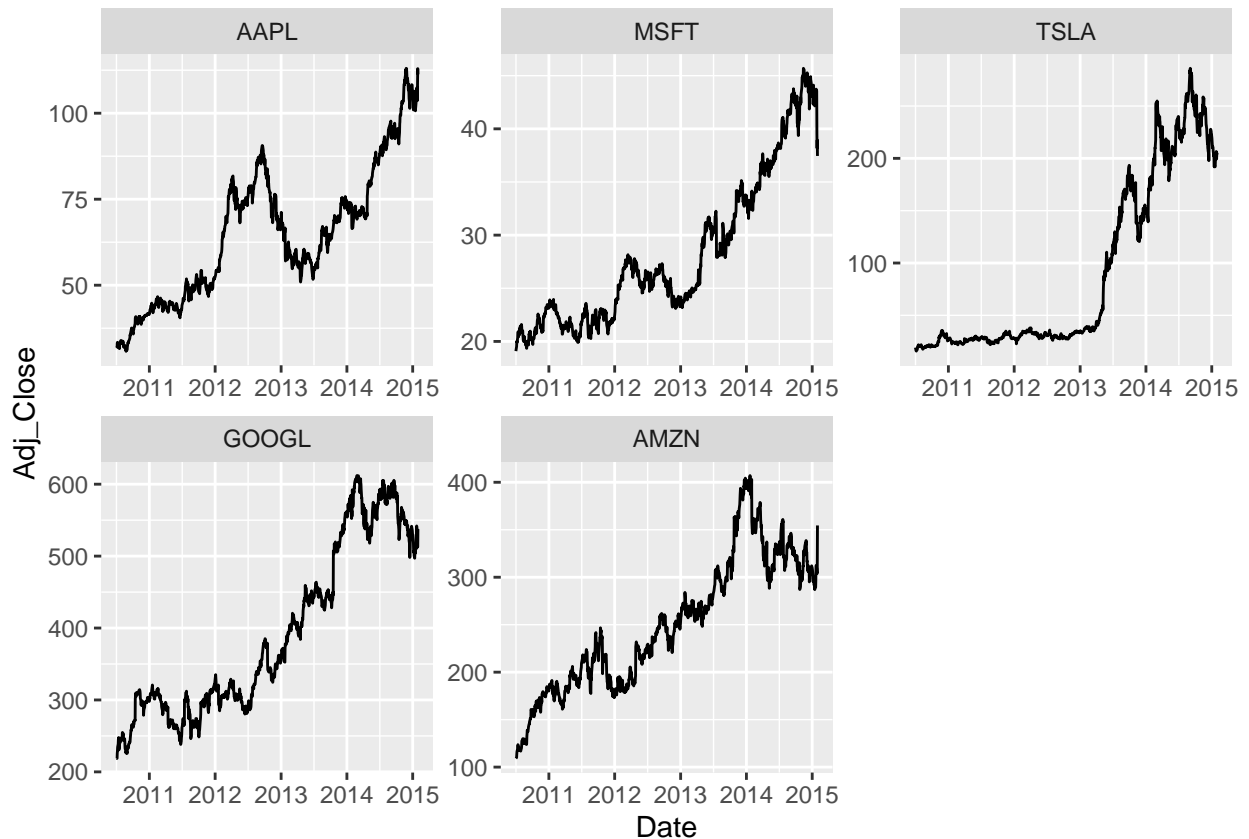
ggplot(prices, aes>Returns)) + geom_histogram(aes(y=..density..),
  bins = 50) + stat_theodensity(colour = "red") +
  facet_wrap(~Company, scales = "free")
```



```
ggplot(prices, aes(sample = Returns)) + geom_qq() + facet_wrap(~Company) +
  stat_qq_line()
```



```
ggplot(prices, aes(Date, Adj_Close)) + geom_line() + facet_wrap(~Company, scales = "free")
```



On y-axis one can see the actual price of the asset. Prices aren't scaled.

Risks, returns and correlations

First risk (standard deviation) and expected returns are calculated for each asset. Expected returns are simply calculated by taking arithmetic mean of the monthly returns. Then covariance and correlation matrices are also calculated. Correlation matrix is calculated for more intuitive visualization.

```
mean_std <- prices[, .(means = mean>Returns), stds = sd>Returns)), by= Company]
```

```
corr_matrix <- cor(as.data.table(split(prices[,4], prices$Company)))
corr_matrix
```

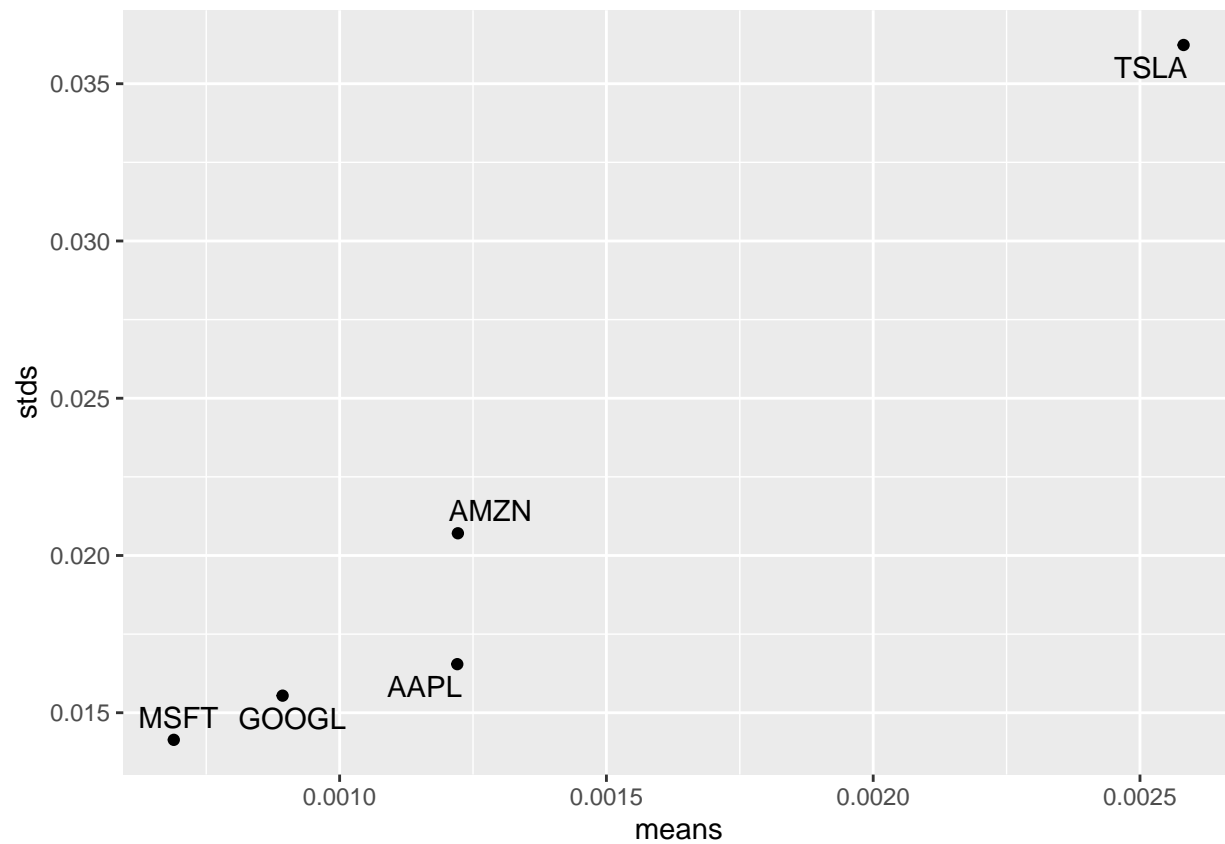
```
##           AAPL      MSFT      TSLA      GOOGL      AMZN
## AAPL  1.0000000  0.3074669  0.2027037  0.3803837  0.2844321
## MSFT  0.3074669  1.0000000  0.2120527  0.3928009  0.3105158
## TSLA  0.2027037  0.2120527  1.0000000  0.2399731  0.2635408
## GOOGL 0.3803837  0.3928009  0.2399731  1.0000000  0.4884545
## AMZN  0.2844321  0.3105158  0.2635408  0.4884545  1.0000000
```

```
cov_matrix <- cov(as.data.table(split(prices[,4], prices$Company)))
cov_matrix
```

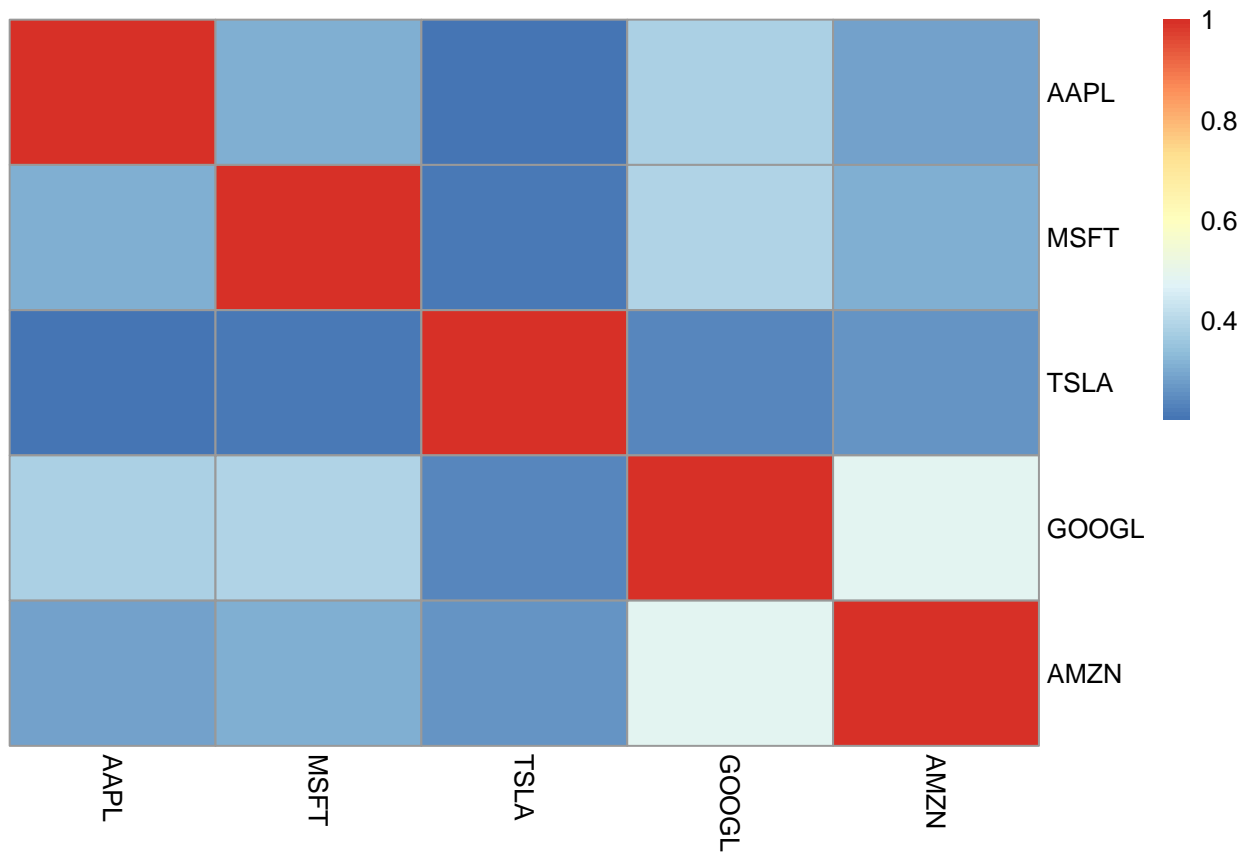
```
##           AAPL      MSFT      TSLA      GOOGL      AMZN
## AAPL  2.737378e-04  7.192477e-05  0.0001215105  9.782306e-05  9.744109e-05
## MSFT  7.192477e-05  1.999059e-04  0.0001086279  8.632512e-05  9.090600e-05
```

```
## TSLA 1.215105e-04 1.086279e-04 0.0013127126 1.351448e-04 1.977101e-04
## GOOGL 9.782306e-05 8.632512e-05 0.0001351448 2.416039e-04 1.572071e-04
## AMZN 9.744109e-05 9.090600e-05 0.0001977101 1.572071e-04 4.287383e-04
```

```
ggplot(mean_std, aes(means, stds, label=Company)) + geom_point() + geom_text_repel()
```



```
pheatmap(corr_matrix, cluster_cols = F, cluster_rows = F)
```



External variables

We need to define values for certain variables to be able to calculate value at risk. First we need to decide time period. Interpretation of VaR is; What is the biggest possible loss within this time period for given confidence level. When defining time period pay attention to frequency of your data. We also need to define risk free rate. We can download this from some data base or we can simply approximate it. Here we don't consider optimal portfolio allocation, so we take weights as given. I use equal weights. We have to also decide our confidence level i.e. how certain we want to be that we don't exceed specific amount of loss. Usually this is 95%

```
time <- 1
risk_free_rate <- 0.01/365
weights <- c(0.2, 0.2, 0.2, 0.2, 0.2)
conf_level = 0.05
```

Portfolio risk and return

Now we have all information we need to calculate expected return and standard deviation for whole portfolio. Portfolios expected return and standard deviation are calculated according to following equations:

Portfolio expected return :

$$E(R_p) = \sum_i w_i E(R_i)$$

Portfolio variance :

$$\sigma^2 = \sum_i w_i \sigma_i^2 + \sum_i \sum_{j \neq i} w_i w_j \sigma_i \sigma_j \rho_{ij}$$

```
port_ret <- weights %*% mean_std$means  
port_std <- sqrt(weights %*% cov_matrix %*% matrix(weights))
```

VaR calculations

Variance-Covariance

There is actually couple of different ways to calculate value at risk. First we are going to examine so called variance-covariance method. Now we need to expect that our portfolio returns are normally distributed. Parametric VaR can be calculated by:

$$VaR = \mu + z\sigma$$

Where μ is expected return of the portfolio, z is value of inverse cumulative distribution function on given confidence level and σ is standard deviation of the portfolio.

```
VaR1 <- qnorm(conf_level, port_ret, port_std)
```

In one day our losses shouldn't exceed -11,92% with probability of 95%

Monte Carlo simulation

Another way to calculate VaR is to use Monte Carlo method. Name Monte Carlo is used of large variety of methods and it doesn't specify too precisely what kind of method we are going to use. Nevertheless main idea of Monte Carlo method is to run many hypothetical trials and estimate results from the outcomes. Monte Carlo method requires more computing, but it is interesting method. We use Geometric Brownian motion:

$$S_t = S_0 \exp[(\mu - \frac{\sigma^2}{2} \Delta t + \sigma \sqrt{\Delta t} \epsilon_t)]$$

to estimate assets price after given time period. Idea of this model is that stock prices grow with certain drift. I use expected returns calculated above. In real life stock prices rarely move linearly. There is usually oscillation around the drift. That's why there is also random stochastic variable in the model. We can model this oscillation by generating random numbers between 0 and 1 and then calculating value of cumulative distribution function value of standard normal distribution with this value. Since we have several assets in our model, we need take into account possible correlation between assets when calculation this variable. We model this oscillation by taking Cholesky's decomposition of our correlation matrix and then multiplying it with vector of random numbers we created.

```
datalist = list()  
  
for (i in 1:15000){  
  random_num <- replicate(n=5, qnorm(runif(1)))  
  Cholesky_Decomposition <- chol(corr_matrix)  
  corr_rand <- random_num %*% Cholesky_Decomposition
```

```

simul_prices <- (100*exp((mean_std$means-0.5*(mean_std$stds^2))*
  time+mean_std$stds*sqrt(time)*corr_rand))
datalist[[i]] <- simul_prices
}
big_data = do.call(rbind, datalist)

tidy_data <- melt(big_data, measure.vars = names, variable.name =
  "Company", value.name = "Price")

```

```

## Warning in melt(big_data, measure.vars = names, variable.name = "Company", : The
## melt generic in data.table has been passed a matrix and will attempt to redirect
## to the relevant reshape2 method; please note that reshape2 is deprecated, and
## this redirection is now deprecated as well. To continue using melt methods from
## reshape2 while both libraries are attached, e.g. melt.list, you can prepend the
## namespace like reshape2::melt(big_data). In the next version, this warning will
## become an error.

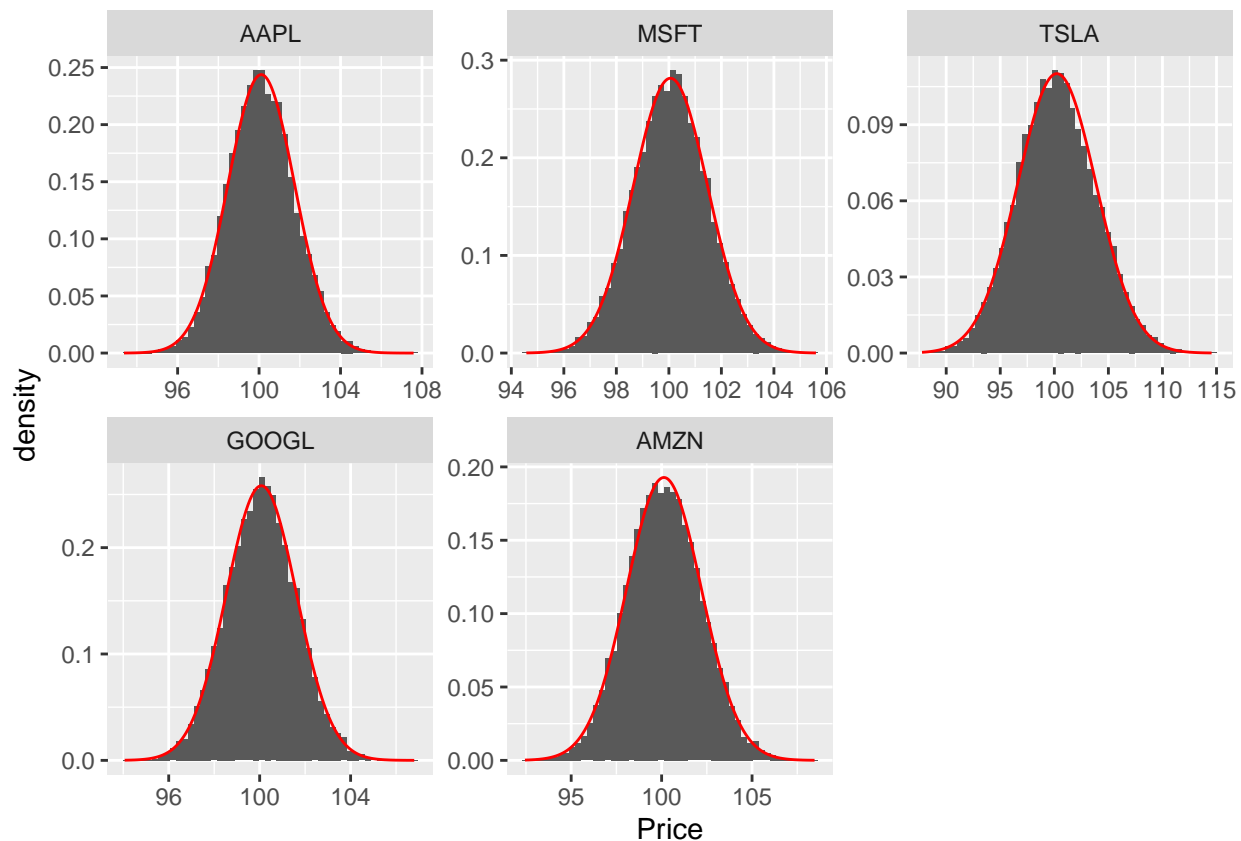
```

```

colnames(tidy_data) <- c("Iteration", "Company", "Price")

ggplot(tidy_data, aes(Price)) + geom_histogram(aes(y=..density..),
  bins = 50) + stat_theodensity(colour = "red") +
  facet_wrap(~Company, scales = "free")

```



Now that we have simulated n amount of correlated prices, we can calculate portfolio value at each outcome. If we consider that we bought one of each stock at the beginning, we can calculate portfolio value simply by adding all prices together.


```
port_val <- as.data.table(rowSums(big_data))
colnames(port_val) <- c("portfolio_value")
```

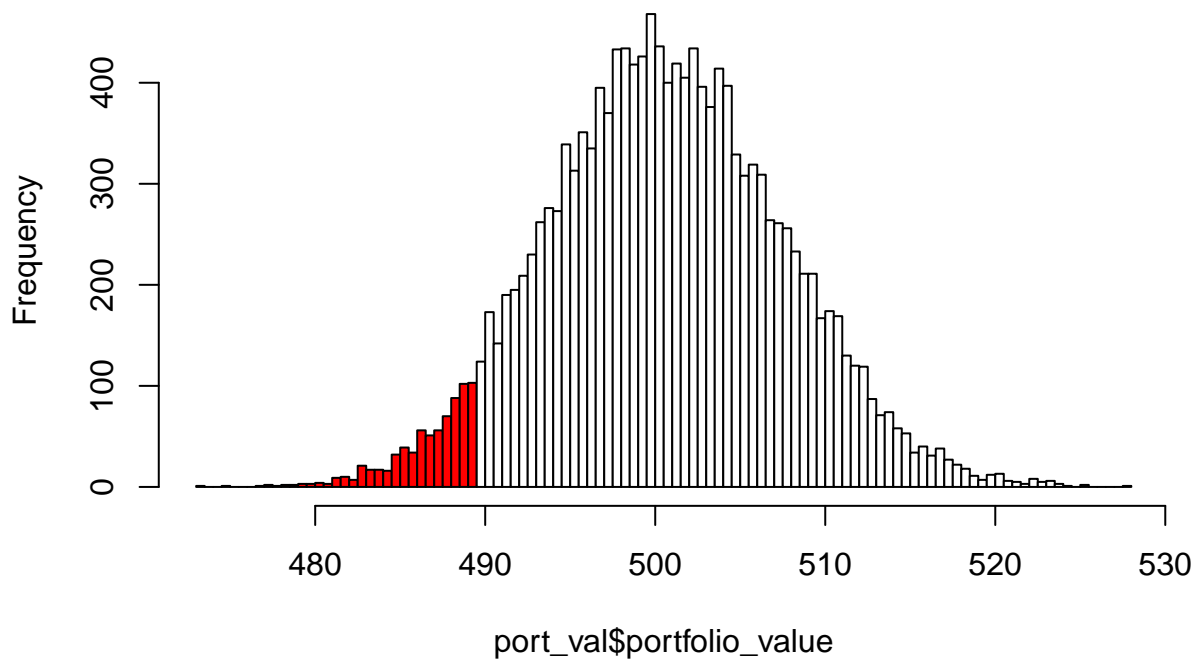
Actual value at risk.

```
Var2 <- quantile(port_val$portfolio_value, conf_level)
Var2
```

```
##      5%
## 489.4905
```

```
h <- hist(port_val$portfolio_value, breaks = 100, plot = FALSE)
cuts <- cut(h$breaks, c(-Inf, Var2, Inf))
plot(h, col=c("red", "white")[cuts])
```

Histogram of port_val\$portfolio_value



References:

<https://quant.stackexchange.com/questions/12592/is-there-a-step-by-step-guide-for-calculating-portfolio-var-using-monte-carlo-si>