Module 5 : August 3rd, 2015

Main Discussion Points:

1. Bootstrap(Follow up from class on 31st July, 2015)
2. R walkthrough for forecasting risk and portfolio variability

Bootstrap:

Need for Bootstrap: Problems associated with the Sampling Distributions

- Sampling distribution is used to characterize uncertainty and to predict probability distribution of a random variable in future
- The objective behind sampling distribution is to carry out thought experiments which involves taking infinite samples with repetition

Evaluate the Standard Error → Standard Deviation of $P(t_n)$ where t is an estimator of underlying population

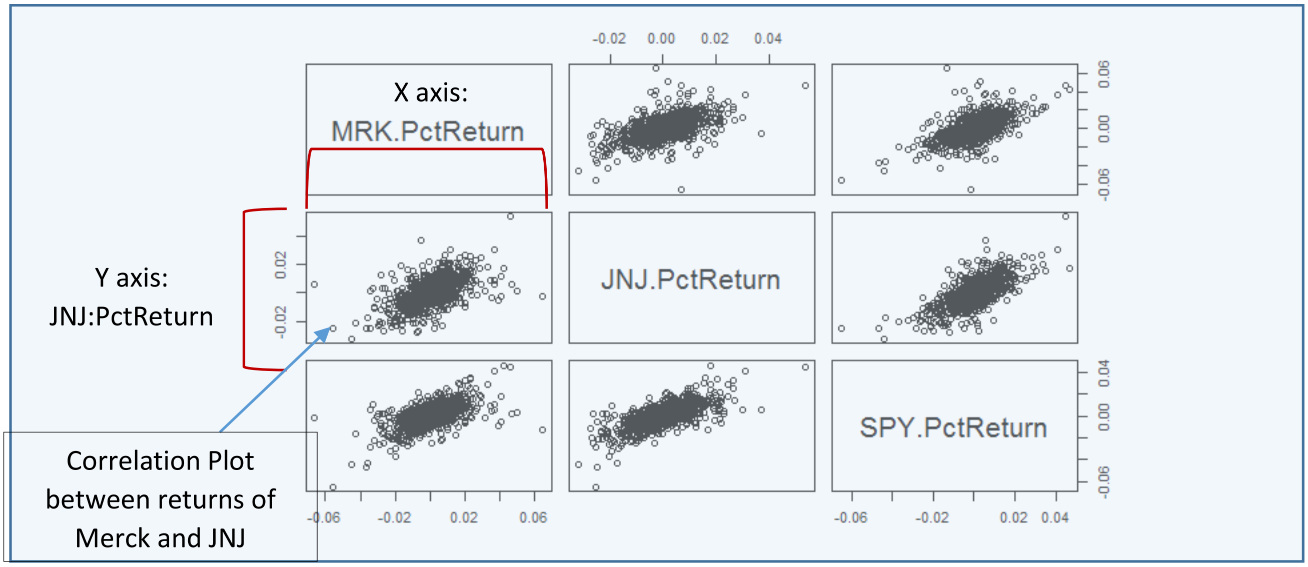
Example:

Sample 1: $x_1^1, x_2^1, \ldots\ldots, x_n^1 \longrightarrow t_n^{(1)}$
Sample 2: $x_1^2, x_2^2, \ldots\ldots, x_n^2 \longrightarrow t_n^{(2)}$
....
....
Sample 1000: $x_1^n, x_2^n, \ldots\ldots, x_n^n \longrightarrow t_n^{(1000)}$
....
∞
Sampling distribution = collection of $t_n^{(1)}, t_n^{(2)} \ldots\ldots t_n^{(1000)}, \ldots\ldots\infty$

- The whole point of carrying out a sampling distribution is to be able to say with a certain amount of confidence that an event will occur in a certain way
- A good analogy would be an assembly line where the manufacturer can say with 95% confidence that his smartphones are defect-free
- The idea in real life is to construct our error-bars just like the assembly line scenario, where we have a certain good level of confidence in the occurrence of event
- A good example of using probability distribution in real life is to use stock prices (obtained from Yahoo) to get returns. At the end we'll realize the issues with using this technique and how a bootstrap circumvents the problems of sampling distributions

Reading a pairplot in R

A pairplot represents the entire data (all the variables) in a form of a matrix of graphs.



Let X and Y be two Random Variables where:

X: daily returns on Johnson and Johnson
Y: daily returns on Merck

P(X,Y) is a joint probability of X and Y

f(X,Y) represents any function on the two random variables. In this example, let this function represent the value of a function where the total portfolio value is represented as a linear value of X and Y.

Let Z be a new Random Variable which stores the function 'f(X,Y)' for future predictions of portfolio value.

These relationships can be represented as a linear relationship as follows:

$$\boxed{Z \equiv f(X,Y) = w_x X + w_y Y}$$

According to classical probability theory,

$$\boxed{E(Z) = w_x E(X) + w_y E(Y)}$$

Where E is the expected value.

As the measure gets more complex, the parameters get more complex and increase in number.

Example for variance:

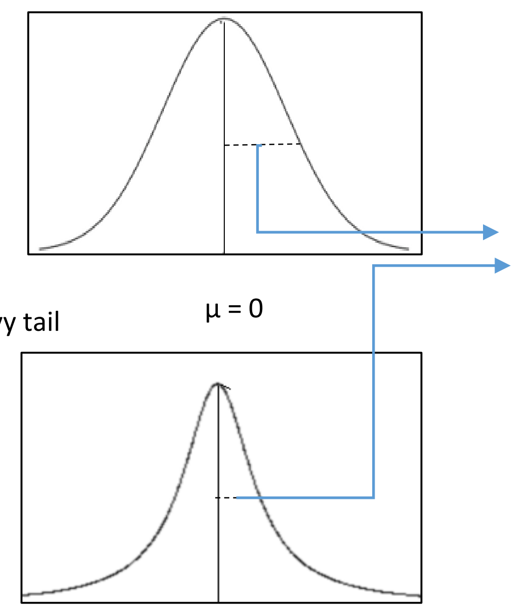
$$\boxed{Var(Z) = w_x^2 Var(X) + w_y^2 Var(Y) + \underbrace{2 w_x\, w_y Cov(X,Y)}_{\text{Variance inflation due to correlation}}}$$

This gets more unfeasible when you are talking about more than 2 Random Variables. Example: In this case, if we had more than 2 stocks.

Mathematically, this can be expressed as:

For P assets, number of covariance terms = $^{P}C_2 = P(P-1)/2$. Basically, it grows quadratically.

- For a large collection of assets (or in general Random Variables), the number of parameters increase and they are more prone to errors. This problem can be summarized as **Statistical Estimation Problem**
- Another problem with using sampling distribution is the **Conceptual Problem**
    - One thing to consider here is the fact the using only mean and variance can put us into trouble. This gives very deceiving information, and this kind of misinformation can have a big negative impact on decisions made by a place, as sensitive as a financial firm
    - Consider the two graphs below. Both have the same mean and standard deviation of 0 and 1 respectively, though they are widely different. The long tails are indicative of higher probability of having "bad days" i.e days of low returns in a financial setting. Had we not visualized this, we would have been deceived by the similar mean and variance and assumed that both stocks behave the same way.



S.D = 1

$\mu = 0$

Gaussian distribution with heavy tail

$\mu = 0$

A setting like a financial firm requires consistency and independence. Failure of independence renders a distribution to be "Non-Gaussian".

These problems make it necessary for us to use a bootstrap when we don't have the actual population to conduct experiments.

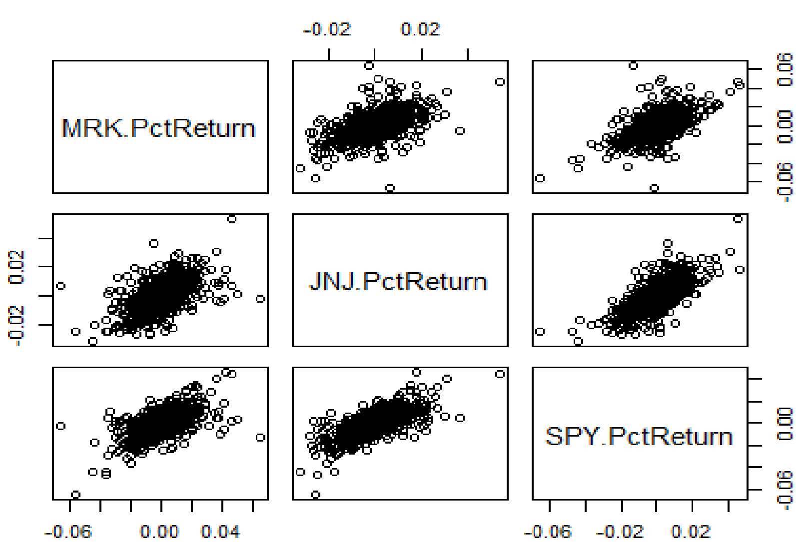
R walkthrough for forecasting risk and portfolio variability

```
library(mosaic)
library(fImport)
library(foreach)

# Import a few stocks
mystocks = c("MRK", "JNJ", "SPY")
myprices = yahooSeries(mystocks, from='2011-01-01', to='2015-07-30')
```

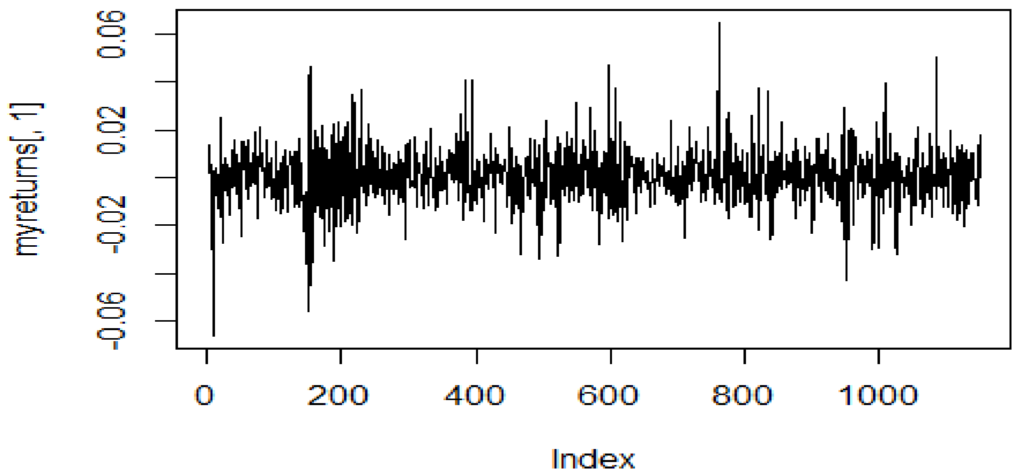
We are creating a function to calculate returns from a Yahoo series. The function takes time series and turns it into returns and interest rates as per the following formula:

$$rt = (P(t) - p(t-1))/P(t-1)$$

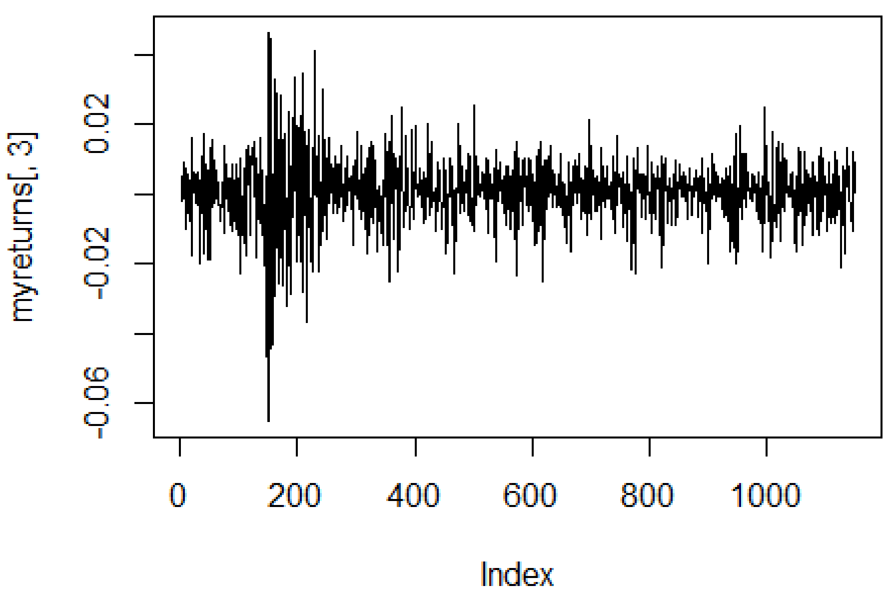
```
YahooPricesToReturns = function(series) {
    mycols = grep('Adj.Close', colnames(series))
    closingprice = series[,mycols]
    N = nrow(closingprice)
    percentreturn = as.data.frame(closingprice[2:N,]) / as.data.frame(closing
price[1:(N-1),]) - 1
    mynames = strsplit(colnames(percentreturn), '.', fixed=TRUE)
    mynames = lapply(mynames, function(x) return(paste0(x[1], ".PctReturn")))
    colnames(percentreturn) = mynames
    as.matrix(na.omit(percentreturn))
}

# Compute the returns from the closing prices
myreturns = YahooPricesToReturns(myprices)

# These returns can be viewed as draws from the joint distribution
pairs(myreturns)
```



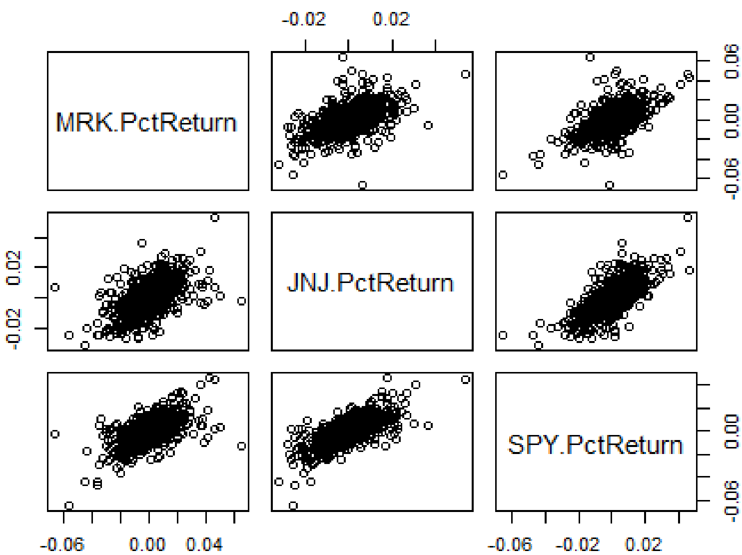
```
#Plot showing day on day returns i.e. connecting returns from Day1,day2…dayn
plot(myreturns[,1], type='l')
```



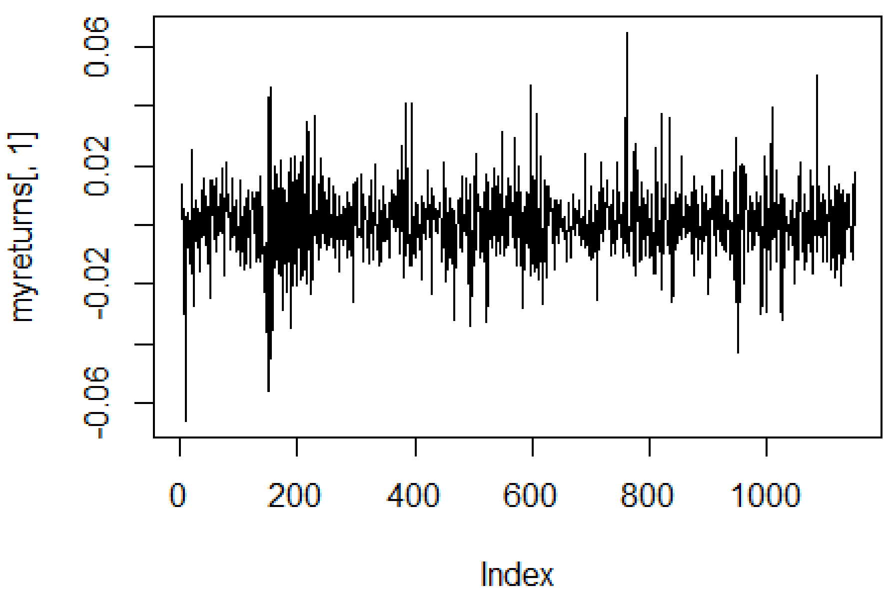

```
# Look at the market returns over time
plot(myreturns[,3], type='l')
```



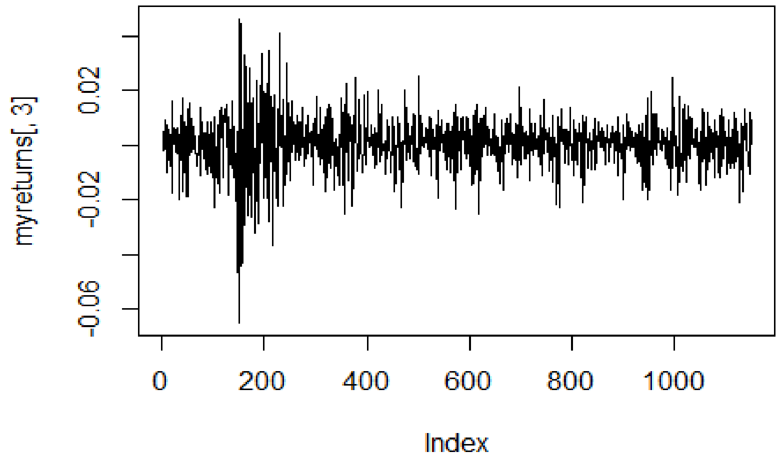
```r
# These returns can be viewed as draws from the joint distribution
pairs(myreturns)
```




```r
par(mfrow = c(1,1))
#Connecing dots between returns from day1, day2...dayn
plot(myreturns[,1], type='l')
```



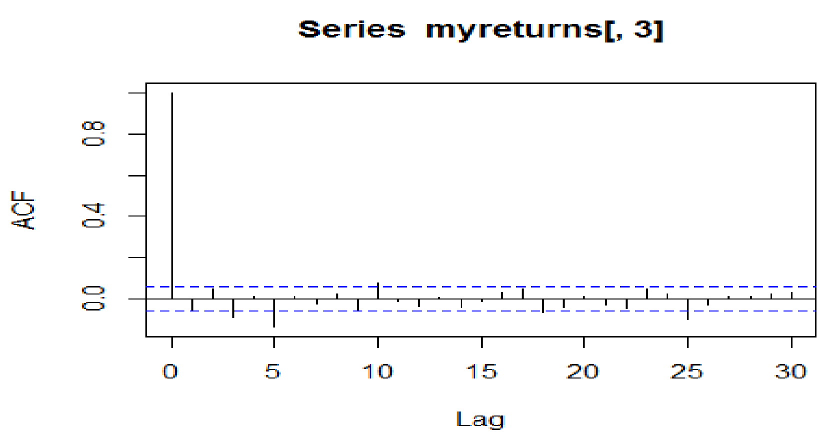
```r
# Look at the market returns over time
plot(myreturns[,3], type='l')
```



The autocorrelation plot is the correlation between elements of a series and elements of the same series separated by a given interval.

For example, ACF with lag 1 shows day 1 correlated to day 2.ACF with lag 2 shows show is Day 1 correlated with Day 3

```r
# An autocorrelation plot: nothing there
acf(myreturns[,3])
```

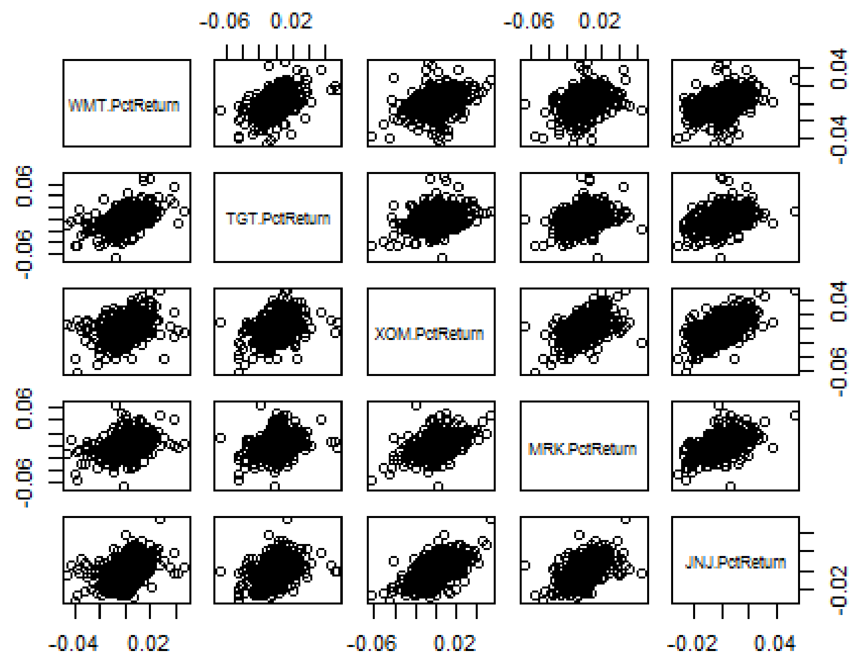


Looking at a portfolio with 5 assets:

Mean, Standard deviation and covariance are not fully sufficient to explain the uncertainty, hence we need to resample to surrogate from the population distribution.

Now, using bootstrap with more stocks:

```
mystocks = c("WMT", "TGT", "XOM", "MRK", "JNJ")
myprices = yahooSeries(mystocks, from='2011-01-01', to='2015-07-30')

# Compute the returns from the closing prices
myreturns = YahooPricesToReturns(myprices)
pairs(myreturns)
```



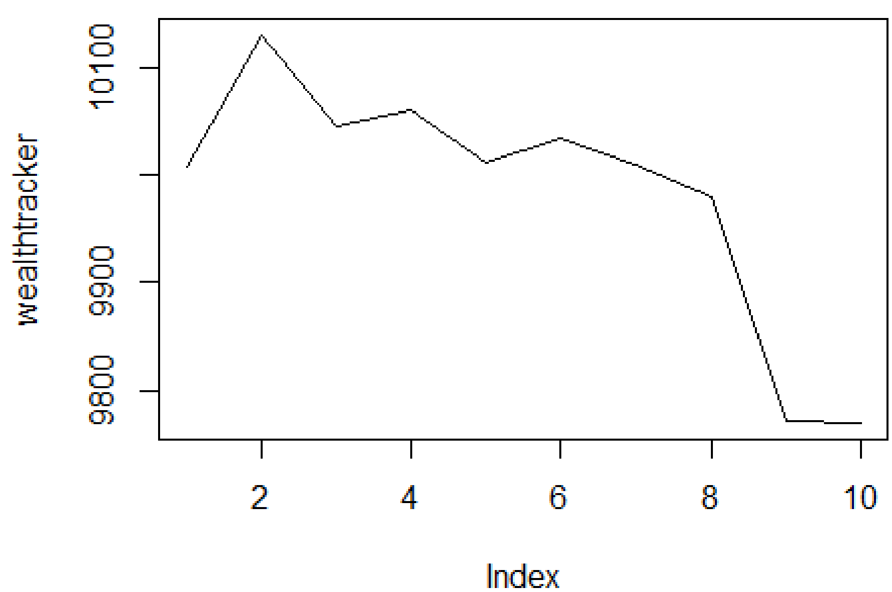

```
# Sample a random return from the empirical joint distribution
# This simulates a random day
return.today = resample(myreturns, 1, orig.ids=FALSE)
return.today

##            WMT.PctReturn TGT.PctReturn XOM.PctReturn MRK.PctReturn
## 2014-01-16     -0.011589   -0.01218323   0.001619801  -0.000380804
##            JNJ.PctReturn
## 2014-01-16  -0.001687806

# Update the value of your holdings
total_wealth = 10000
holdings = total_wealth*c(0.2,0.2,0.2,0.2,0.2)
holdings = holdings + holdings*return.today
```

```
# Compute your new total wealth
totalwealth = sum(holdings)
```

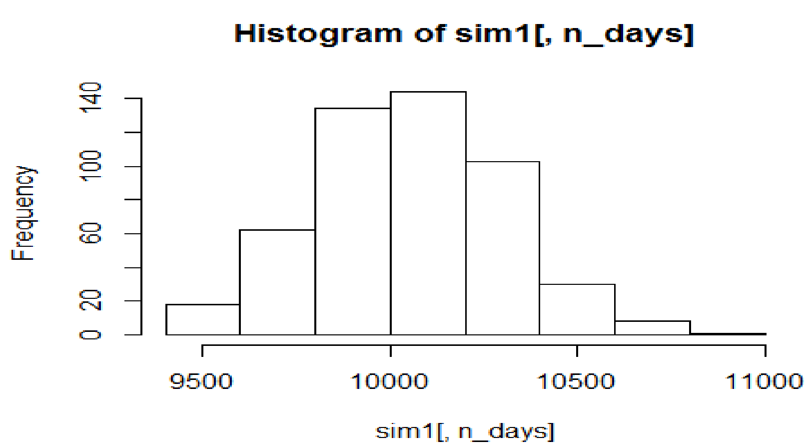
To calulate the 2 week wealth of an even split portfolio:

X_{it} = Returns of asset I for day t
Z= f(X_{it} for all i,t)
Z is the 2 week wealth of the even split portfolio
i is 1 to 5 days of a week and t is 1-10 weeks

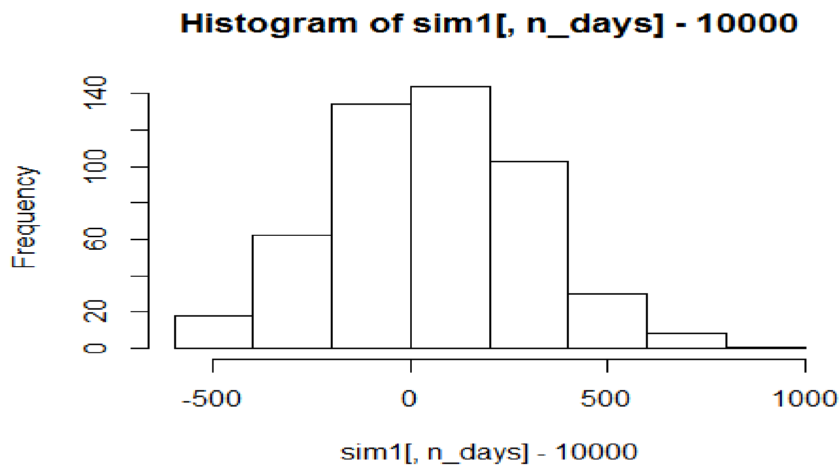
```
# Now loop over two trading weeks
totalwealth = 10000
weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
holdings = weights * totalwealth
n_days = 10
wealthtracker = rep(0, n_days) # Set up a placeholder to track total wealth
for(today in 1:n_days) {
    return.today = resample(myreturns, 1, orig.ids=FALSE)
    holdings = holdings + holdings*return.today
    totalwealth = sum(holdings)
    wealthtracker[today] = totalwealth
}
totalwealth

## [1] 9769.236

plot(wealthtracker, type='l')
```



```r
# Now simulate many different possible trading years!
sim1 = foreach(i=1:500, .combine='rbind') %do% {
    totalwealth = 10000
    weights = c(0.2, 0.2, 0.2, 0.2, 0.2)
    holdings = weights * totalwealth
    wealthtracker = rep(0, n_days) # Set up a placeholder to track total weal
th
    for(today in 1:n_days) {
        return.today = resample(myreturns, 1, orig.ids=FALSE)
        holdings = holdings + holdings*return.today
        totalwealth = sum(holdings)
        wealthtracker[today] = totalwealth
    }
    wealthtracker
}

head(sim1)

##                [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
## result.1 10046.146 10001.156  9951.340  9961.465  9795.615  9889.231
## result.2 10047.262 10103.131 10038.868 10040.794  9776.670  9801.897
## result.3 10007.174 10060.811  9971.881 10055.622 10030.898  9986.271
## result.4 10020.877 10063.325 10112.157 10142.452 10075.835  9958.943
## result.5 10035.782 10065.245 10031.402 10052.763 10057.246 10093.315
## result.6  9766.015  9821.347  9794.013  9751.894  9697.393  9742.036
##                [,7]      [,8]      [,9]     [,10]
## result.1  9862.999  9978.455 10018.600 10011.911
## result.2  9793.972  9847.421  9931.509 10027.679
## result.3  9873.178  9826.446  9821.400  9629.337
## result.4 10095.221 10088.700 10034.367 10120.707
## result.5 10105.491 10080.963 10112.248 10062.823
## result.6  9819.154  9892.850  9811.714  9857.316

hist(sim1[,n_days])
```



```r
# Profit/loss
hist(sim1[,n_days]- 10000)
```



Value of risk at level alpha is the alpha quantile of the profit and loss distribution of a portfolio.

Tail dependence describes the limiting proportion that one margin exceeds a certain threshold given that the other margin has already exceeded that threshold.

```r
# Calculate 5% value at risk
quantile(sim1[,n_days], 0.05) - 10000

##       5%
## -366.398
```

5% tail is at $9600 which is ~$400 loss from initial investment