CSC425 Week1 Practice

1.Set the working directory

```
setwd('~/Desktop')
```

2. library

```
library(fBasics)
```

```
## Loading required package: timeDate
## Loading required package: timeSeries
##
## Rmetrics Package fBasics
## Analysing Markets and calculating Basic Statistics
## Copyright (C) 2005-2014 Rmetrics Association Zurich
## Educational Software for Financial Engineering and Computational Science
## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.
## https://www.rmetrics.org --- Mail to: info@rmetrics.org
```

3. import data

```
dat = read.table("ibm.txt", header = T)
head(dat)
```

```
##       date       ibm     sprtn
## 1 19670331  0.048837  0.039410
## 2 19670428  0.100887  0.042239
## 3 19670531 -0.035234 -0.052441
## 4 19670630  0.067024  0.017512
## 5 19670731  0.020603  0.045344
## 6 19670831 -0.013589 -0.011715
```

```
#first two cols of the data
dat[1,]
```

```
##       date      ibm   sprtn
## 1 19670331 0.048837 0.03941
```

4.Get IBM simpole returns

```
ibm = dat[,2]
head(ibm)
```

```
## [1]  0.048837  0.100887 -0.035234  0.067024  0.020603 -0.013589
```

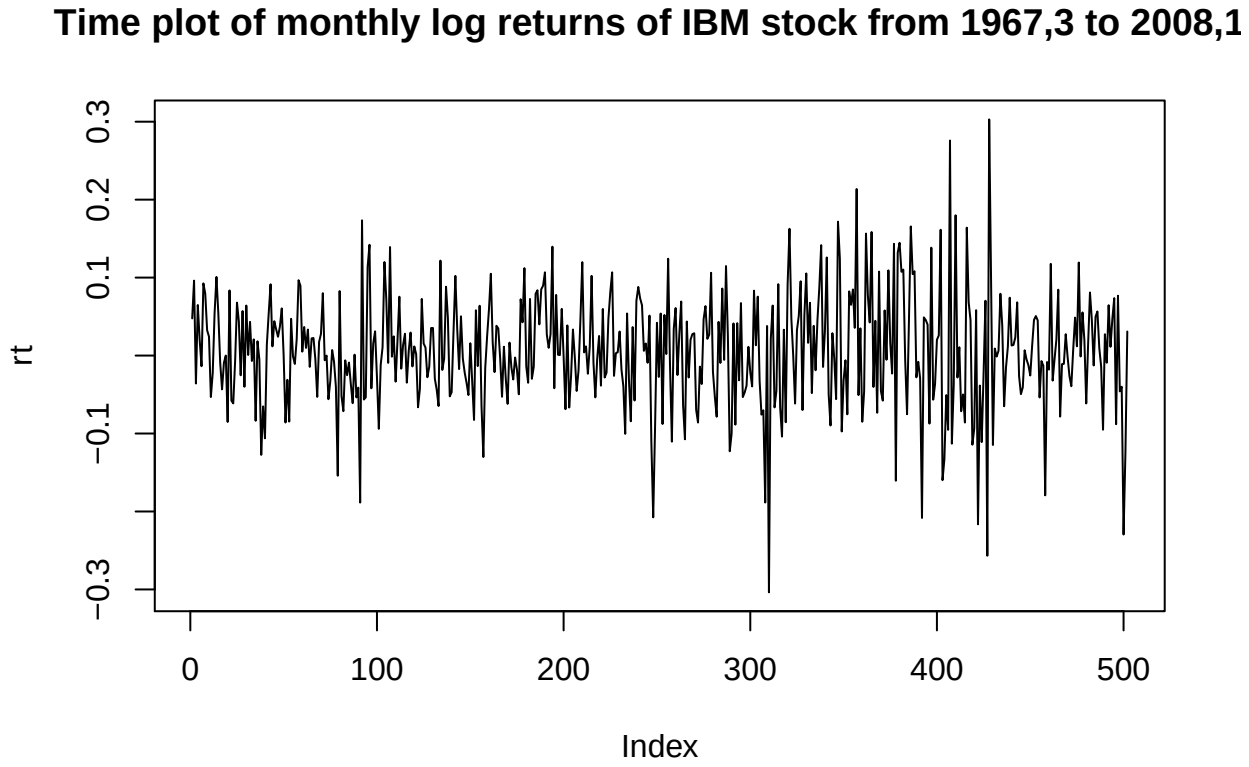
5.Transform into log returns

```
rt = log(ibm + 1)
head(rt)
```

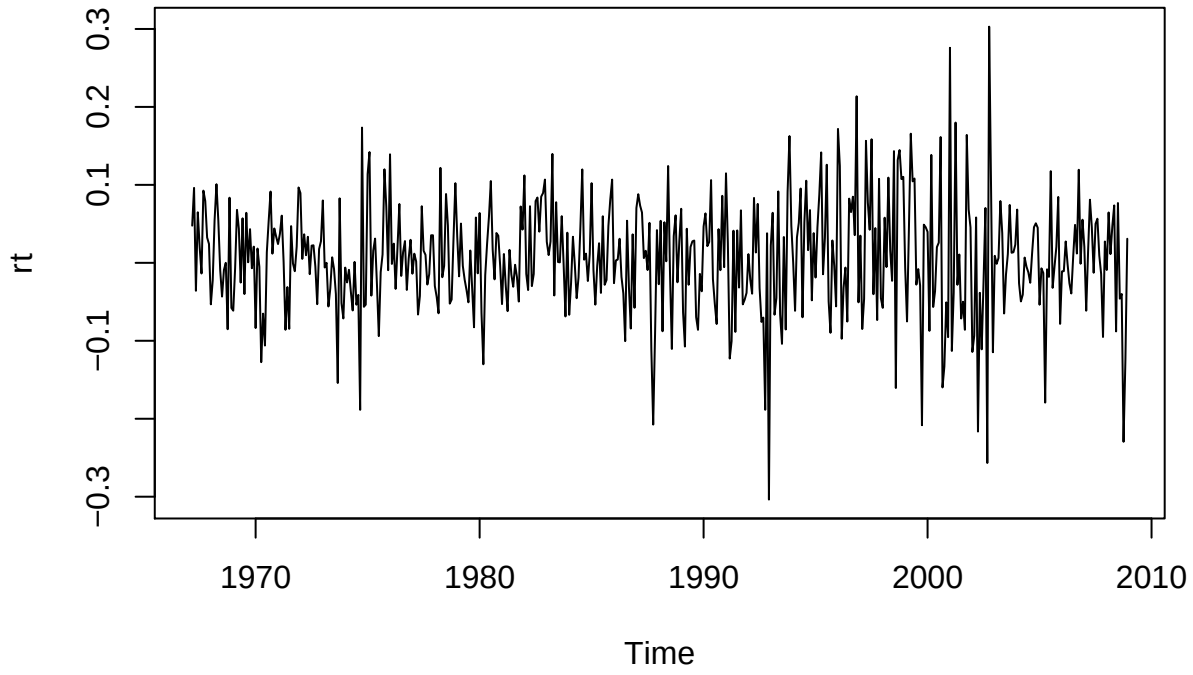
```
## [1]  0.04768193  0.09611622 -0.03586969  0.06487347  0.02039363 -0.01368218
```

6.Plot

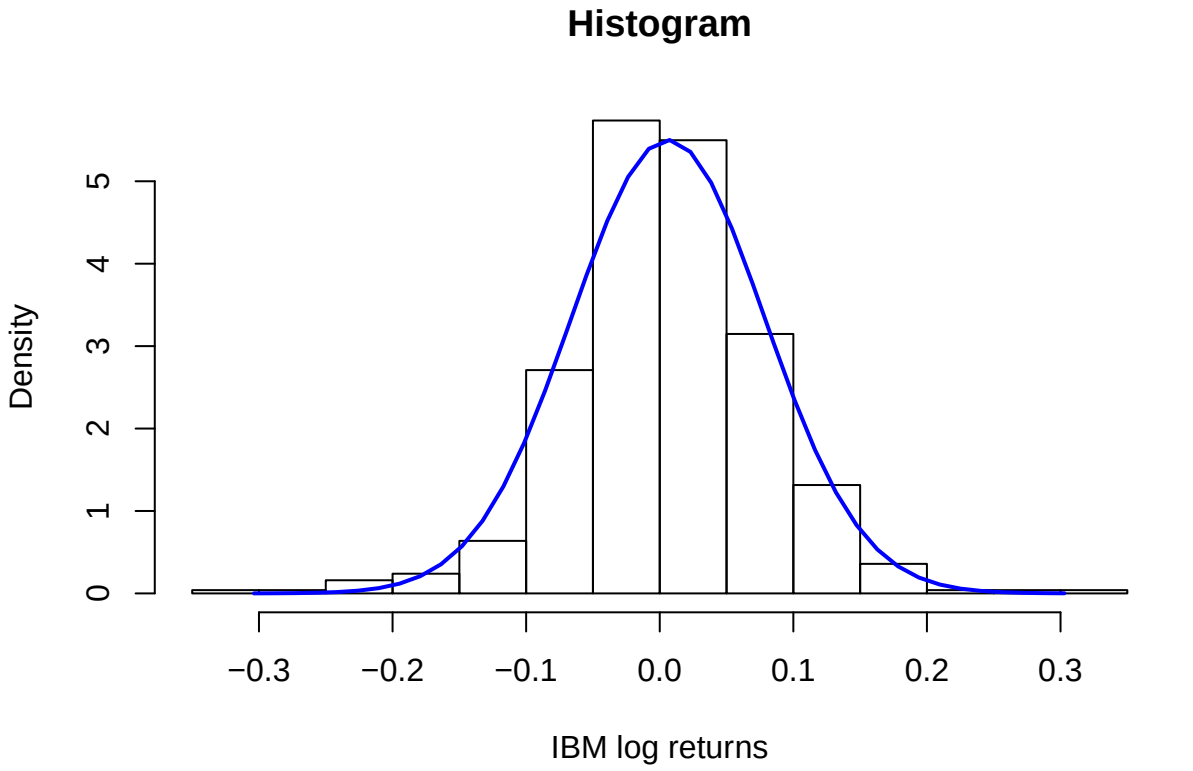
```
# Time plot of log returns with caption
plot(rt, type = 'l')
title(main = 'Time plot of monthly log returns of IBM stock from 1967,3 to 2008,12')
```




```
# Create Time Series object
rt = ts(rt, frequency = 12, start = c(1967, 3))
# Creates Time plot with time on X-axis
plot.ts(rt)
```

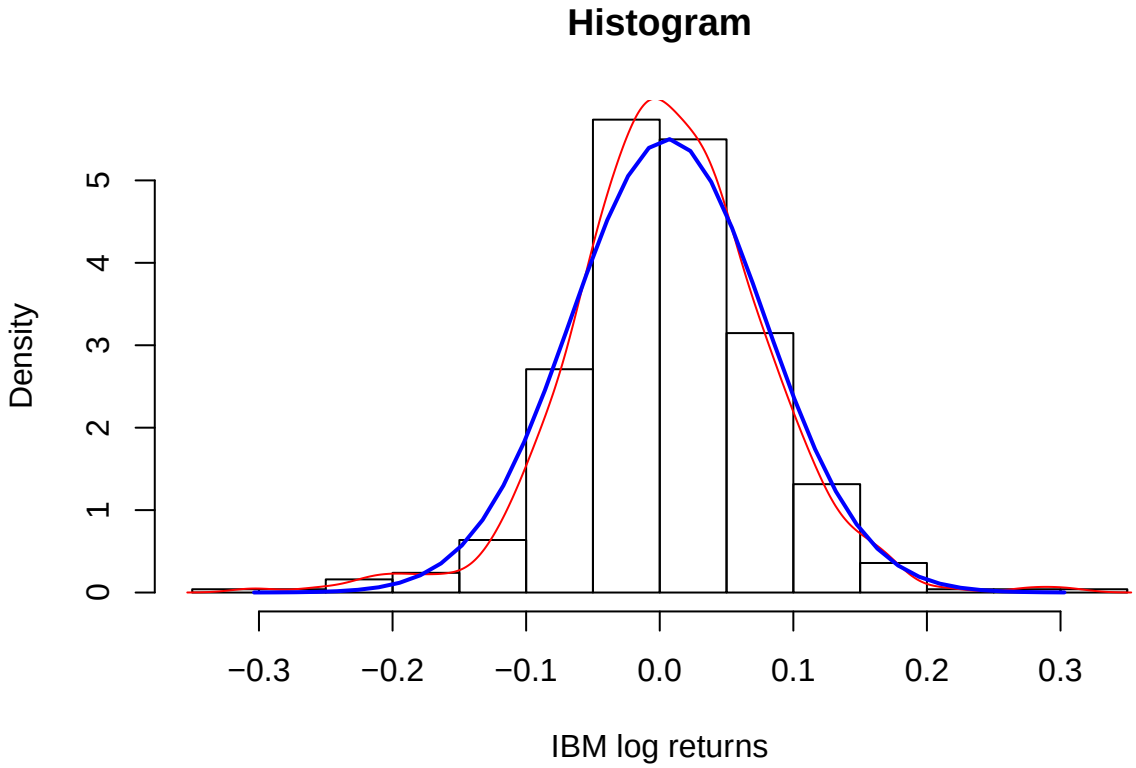



```
# Histogram of IBM returns
hist(rt, xlab = "IBM log returns", freq = FALSE, main = "Histogram")
xfit <- seq(min(rt), max(rt), length = 40)
yfit <- dnorm(xfit, mean = mean(rt), sd = sd(rt))
lines(xfit, yfit, col = "blue", lwd = 2)
```

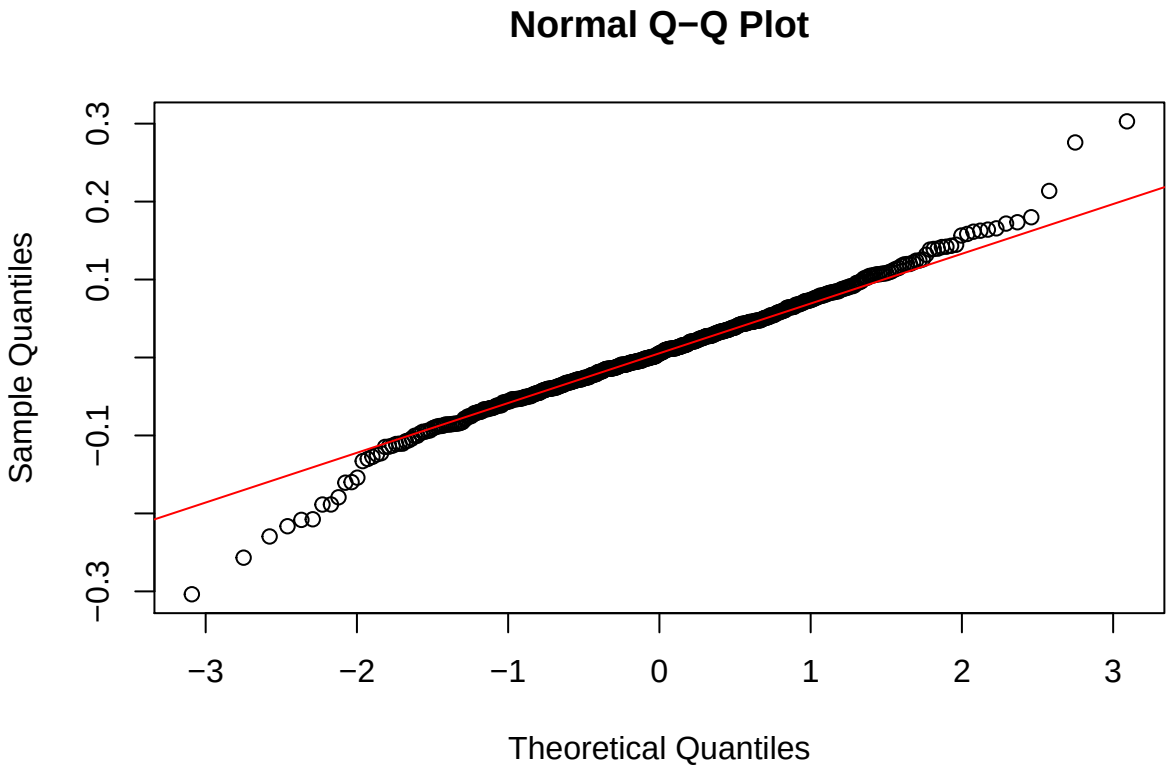



```
# Histogram with kernel density curve (approximated returns density)
hist(rt, xlab = "IBM log returns", freq = FALSE, main ="Histogram")
```

```r
lines(density(rt), col = "red")
# Add normal density curve to the histogram if xfit and yfit are already defined
lines(xfit, yfit, col = "blue", lwd = 2)
```




```r
# Normal quantile plot with diagonal line
qqnorm(rt)
qqline(rt, col = 2)
```



7. Summary Statistics

```
# Compute sample mean
mean(rt)
```

```
## [1] 0.006208082
```

```
# Compute sample variance
var(rt)
```

```
## [1] 0.005258775
```

```
# Compute standard deviation
stdev(rt)
```

```
## [1] 0.07251741
```

```
# Compute sample skewness
skewness(rt)
```

```
## [1] -0.1353432
## attr(,"method")
## [1] "moment"
```

```
# Compute basic statistics (better option)
basicStats(rt)
```

```
##                     rt
## nobs        502.000000
## NAs           0.000000
## Minimum      -0.303683
## Maximum       0.302915
```

```
## 1. Quartile  -0.037641
## 3. Quartile   0.048443
## Mean          0.006208
## Median        0.005260
## Sum           3.116457
## SE Mean       0.003237
## LCL Mean     -0.000151
## UCL Mean      0.012567
## Variance      0.005259
## Stdev         0.072517
## Skewness     -0.135343
## Kurtosis      1.693092
```

```
# Compute summary statistics
summary(rt)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.303700 -0.037640  0.005260  0.006208  0.048440  0.302900
```

```
# Perfomr t-test for mean being zero.
t.test(rt)
```

```
##
##  One Sample t-test
##
## data:  rt
## t = 1.9181, df = 501, p-value = 0.05567
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0001509194  0.0125670842
## sample estimates:
##   mean of x
## 0.006208082
```

```
#Perform one-sided test
t.test(rt, alternative = c("greater"))
```

```
##
##  One Sample t-test
##
## data:  rt
## t = 1.9181, df = 501, p-value = 0.02783
## alternative hypothesis: true mean is greater than 0
## 95 percent confidence interval:
##  0.0008744697          Inf
## sample estimates:
##   mean of x
## 0.006208082
```

8.Normality Test

```
# Perform Jarque-Bera normality test
normalTest(rt, method = c("jb"))
```

```
##
## Title:
```

```
##  Jarque - Bera Normalality Test
##
## Test Results:
##   STATISTIC:
##     X-squared: 62.8363
##   P VALUE:
##     Asymptotic p Value: 2.265e-14
##
## Description:
##  Thu Sep 14 17:10:19 2017 by user:
```

```
# Perform Shapiro-Wilk normality test
normalTest(rt, method = c("sw"))
```

```
##
## Title:
##  Shapiro - Wilk Normality Test
##
## Test Results:
##   STATISTIC:
##     W: 0.9828
##   P VALUE:
##     1.172e-05
##
## Description:
##  Thu Sep 14 17:10:19 2017 by user:
```

9.Skewness Test

```
s3 = skewness(rt)
T = length(rt)
tst = s3 / sqrt(6/T)
pv = 2 * pnorm(-abs(tst))
pv
```

```
## [1] 0.2157246
## attr(,"method")
## [1] "moment"
```

10. Kurtosis (FAT-TAIL) TEST

```
k4 = kurtosis(rt)
tst = k4/sqrt(24/T)
pv = 2*pnorm(-abs(tst))
pv
```

```
## [1] 9.686087e-15
## attr(,"method")
## [1] "excess"
```