assignment1_thanachote.R

user

2021-02-09

```
setwd("/Users/user/Desktop/Arm/EE435 R")
#install.packages("quantmod")
#install.packages("fBasics")
#install.packages("sn")
#install.packages("PerformanceAnalytics")
#install.packages("car")
#install.packages("tseries")
#install.packages("forecast")
library(quantmod)
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(fBasics)
```

```
## Loading required package: timeDate
```

```
## Loading required package: timeSeries
```

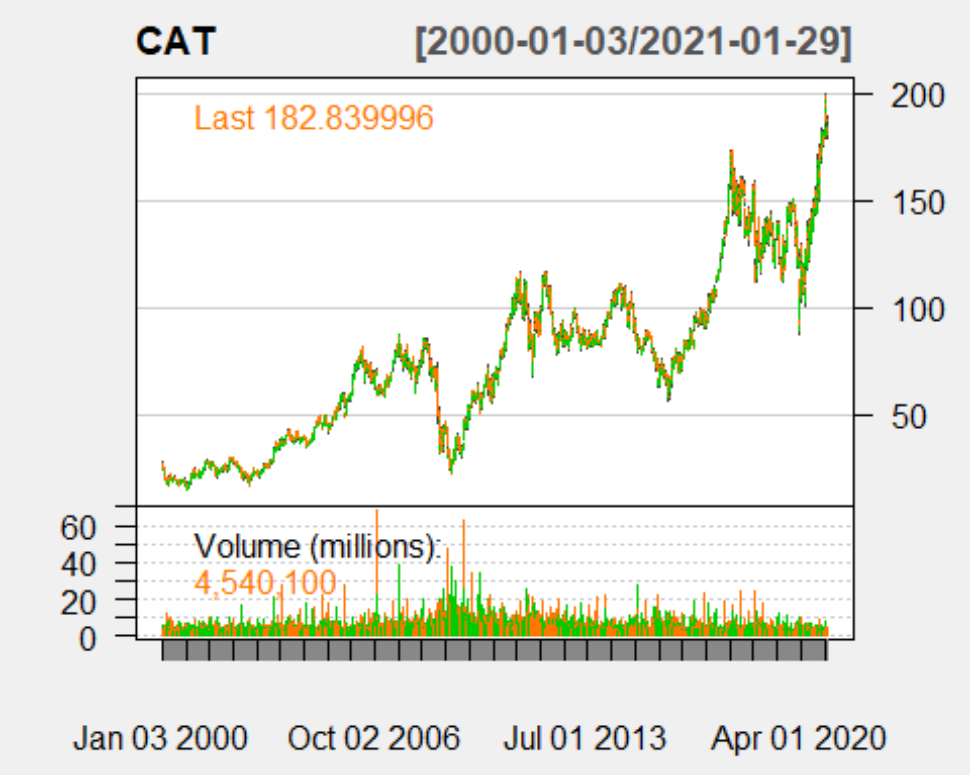
```
##
## Attaching package: 'timeSeries'
```

```
## The following object is masked from 'package:zoo':
##
##     time<-
```

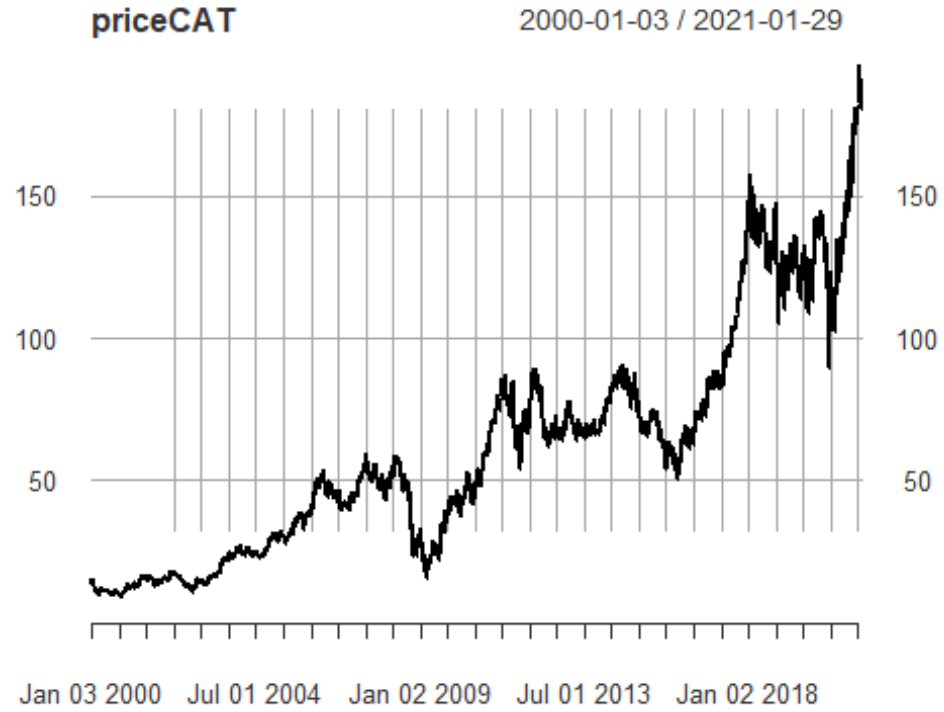
```
##
## Attaching package: 'fBasics'
```

```
## The following object is masked from 'package:TTR':
##
##     volatility

library(sn)

## Loading required package: stats4

##
## Attaching package: 'sn'

## The following object is masked from 'package:fBasics':
##
##     vech

## The following object is masked from 'package:stats':
##
##     sd

library(PerformanceAnalytics)

##
## Attaching package: 'PerformanceAnalytics'

## The following objects are masked from 'package:timeDate':
##
##     kurtosis, skewness

## The following object is masked from 'package:graphics':
##
##     legend

library(car)

## Loading required package: carData

##
## Attaching package: 'car'

## The following object is masked from 'package:fBasics':
##
##     densityPlot

library(tseries)
library(forecast)
getSymbols("CAT",from="2000-01-03",to="2021-01-31")

## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
```

```
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.

## [1] "CAT"

dim(CAT)

## [1] 5303    6

head(CAT)

##            CAT.Open CAT.High  CAT.Low CAT.Close CAT.Volume CAT.Adjusted
## 2000-01-03 23.84375 24.50000 23.84375  24.31250    5055000     13.75500
## 2000-01-04 24.31250 24.87500 24.00000  24.00000    6181400     13.57821
## 2000-01-05 24.00000 25.09375 24.00000  24.56250    6398600     13.89644
## 2000-01-06 25.28125 26.12500 25.28125  25.81250    5140600     14.60364
## 2000-01-07 26.37500 27.56250 26.37500  26.65625    6360200     15.08100
## 2000-01-10 26.65625 27.28125 25.75000  25.78125    3682200     14.58597

tail(CAT)

##            CAT.Open CAT.High CAT.Low CAT.Close CAT.Volume CAT.Adjusted
## 2021-01-22   190.85   192.82  189.31    191.94    2027000       191.94
## 2021-01-25   190.56   191.37  186.97    187.34    3837100       187.34
## 2021-01-26   187.60   189.41  186.83    187.21    2506000       187.21
## 2021-01-27   184.20   185.55  179.34    180.63    4095300       180.63
## 2021-01-28   182.26   187.61  181.31    184.34    3445400       184.34
## 2021-01-29   183.50   188.82  180.73    182.84    4540100       182.84

da=CAT
chartSeries(CAT,theme="white")
```

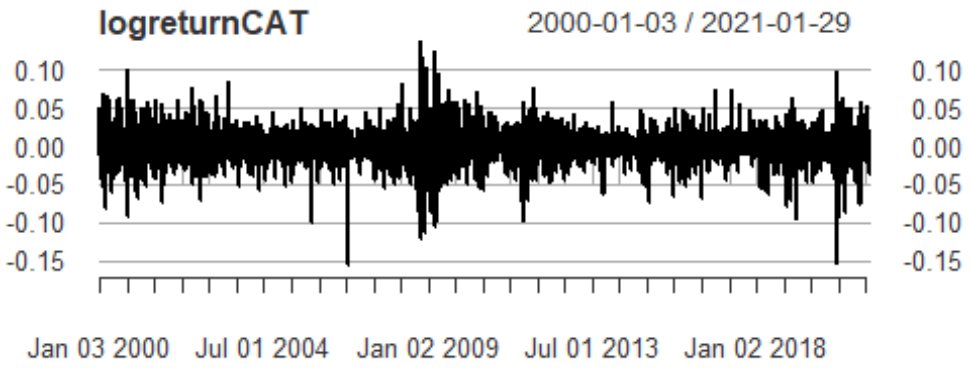


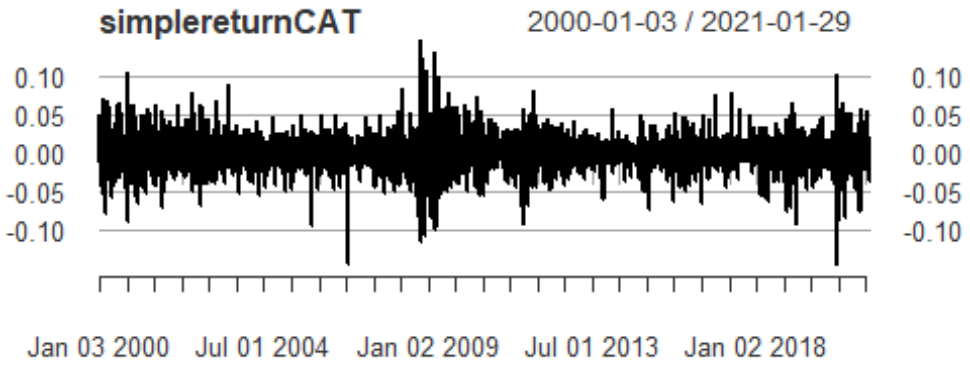

```
priceCAT=da[,6]
plot(priceCAT,type='l')
```



```
logpriceCAT=log(priceCAT)
plot(logpriceCAT,type='l')
```


```
logreturnCAT=diff(log(priceCAT))
simplereturnCAT <-exp(logreturnCAT)-1

par(mfrow=c(2,1))
plot(logreturnCAT,type='l')
plot(simplereturnCAT)
```






```
getSymbols("AOT.BK",from="2000-01-03",to="2021-01-31")
```

```
## [1] "AOT.BK"
```

```
dim(AOT.BK)
```

```
## [1] 4159    6
```

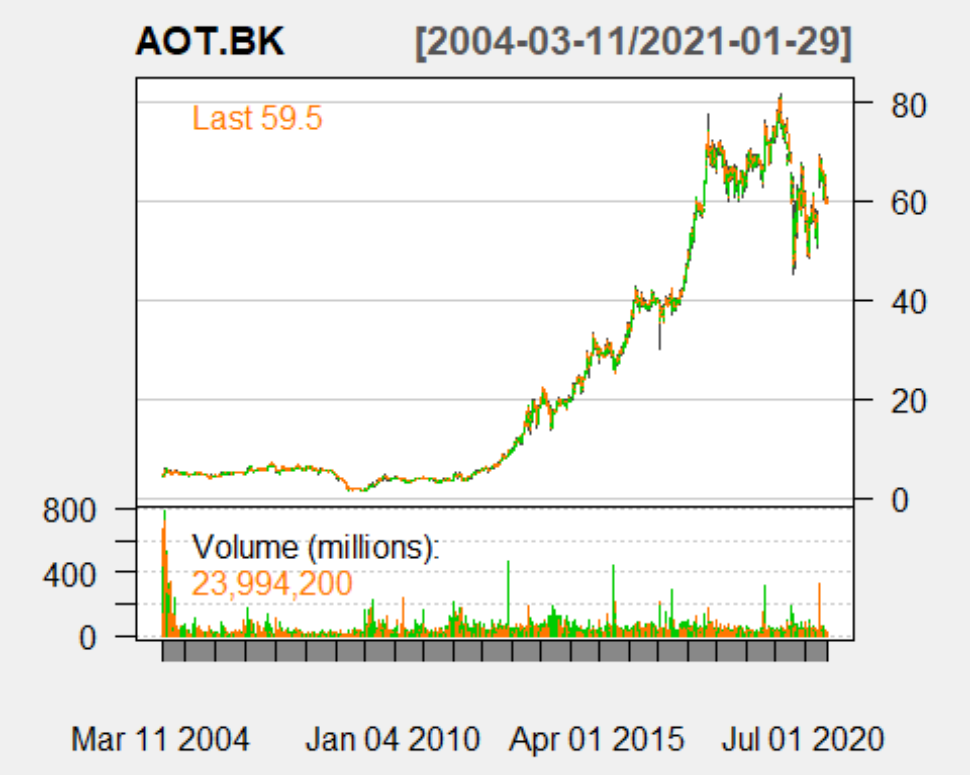
```
head(AOT.BK)
```

```
##            AOT.BK.Open AOT.BK.High AOT.BK.Low AOT.BK.Close AOT.BK.Volume
## 2004-03-11       4.500       4.800      4.425        4.800      17414690
## 2004-03-12       4.700       4.775      4.600        4.600     668350000
## 2004-03-15       4.625       4.650      4.550        4.550     346345000
## 2004-03-16       4.525       4.575      4.500        4.525     136384000
## 2004-03-17       4.550       4.600      4.550        4.600      88608000
## 2004-03-18       4.650       4.925      4.650        4.775     434665000
##            AOT.BK.Adjusted
## 2004-03-11        2.668538
## 2004-03-12        2.557348
## 2004-03-15        2.529552
## 2004-03-16        2.515653
## 2004-03-17        2.557348
## 2004-03-18        2.654639
```

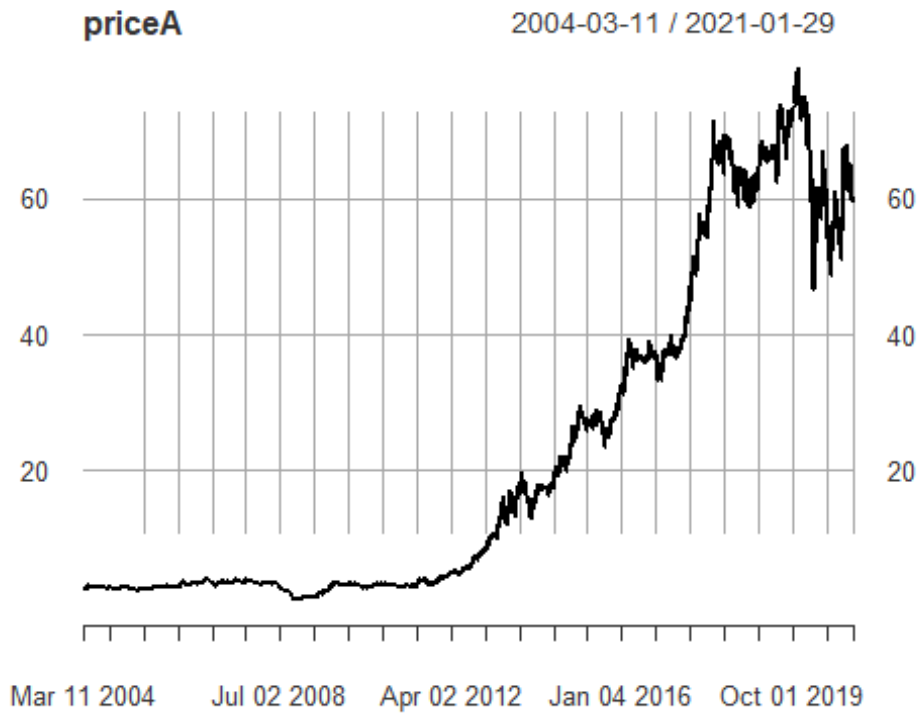
```
tail(AOT.BK)
```

```
##            AOT.BK.Open AOT.BK.High AOT.BK.Low AOT.BK.Close AOT.BK.Volume
## 2021-01-22       60.50       60.75        59.5         59.75       27812300
## 2021-01-25       59.75       60.50        59.5         59.75       10446500
## 2021-01-26       59.75       60.75        59.5         60.25       15439500
## 2021-01-27       60.25       60.75        60.0         60.25       12355000
## 2021-01-28       59.50       60.50        59.5         60.00       19675400
## 2021-01-29       60.25       60.50        59.5         59.50       23994200
##            AOT.BK.Adjusted
## 2021-01-22           59.75
## 2021-01-25           59.75
## 2021-01-26           60.25
## 2021-01-27           60.25
## 2021-01-28           60.00
## 2021-01-29           59.50
```

```
da=AOT.BK
chartSeries(AOT.BK,theme="white")
```

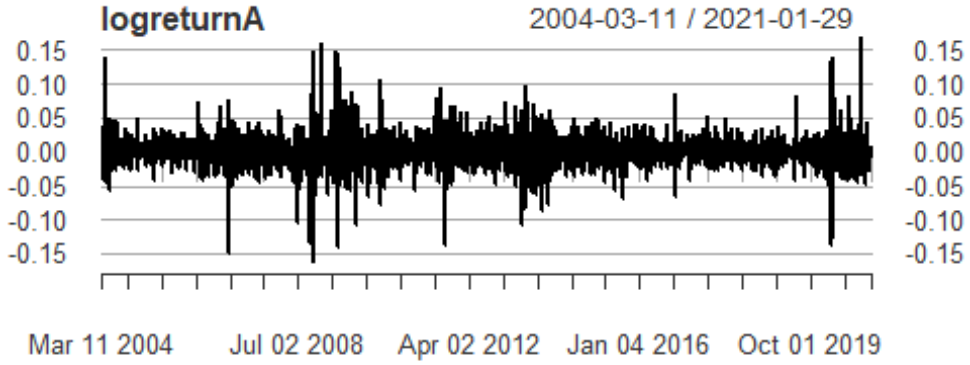


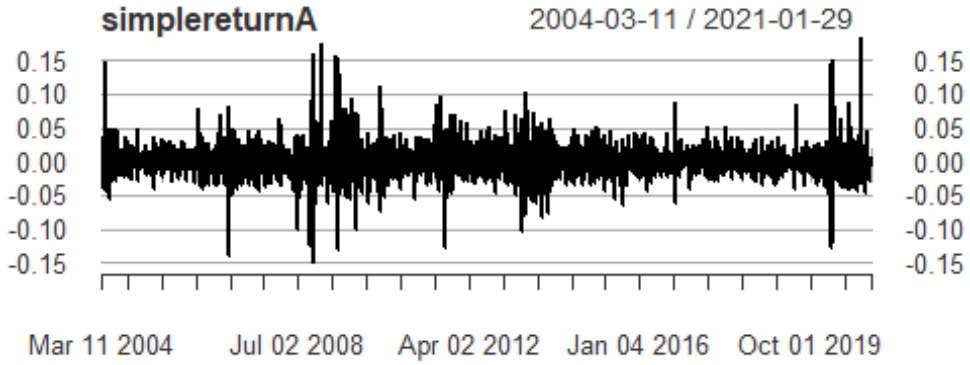

```
priceA=da[,6]
plot(priceA,type='l')
```



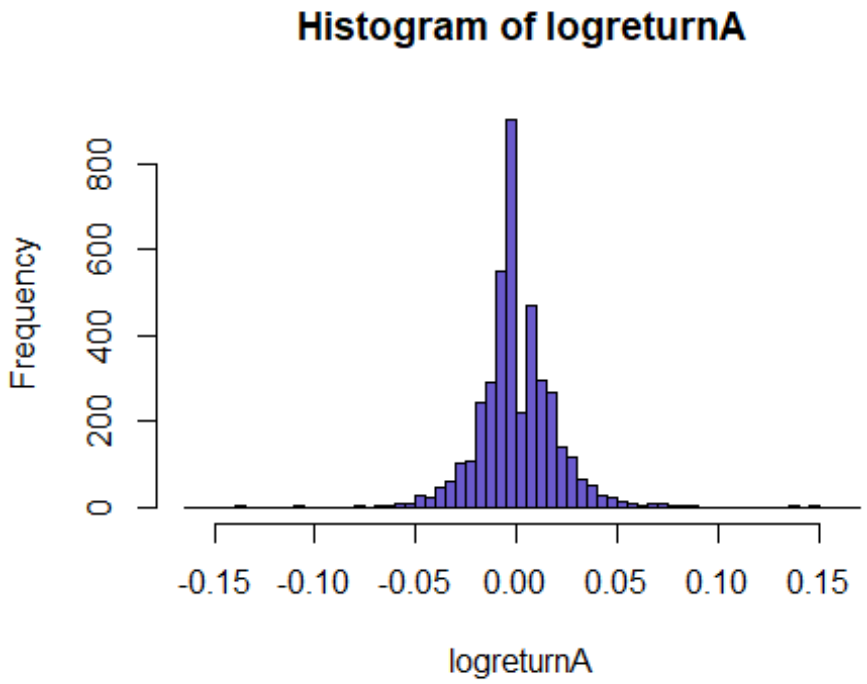

```
logpriceA=log(priceA)
plot(logpriceA,type='l')
```

```
logreturnA=diff(log(priceA))
simplereturnA <-exp(logreturnA)-1

par(mfrow=c(2,1))
plot(logreturnA,type='l')
plot(simplereturnA)
```

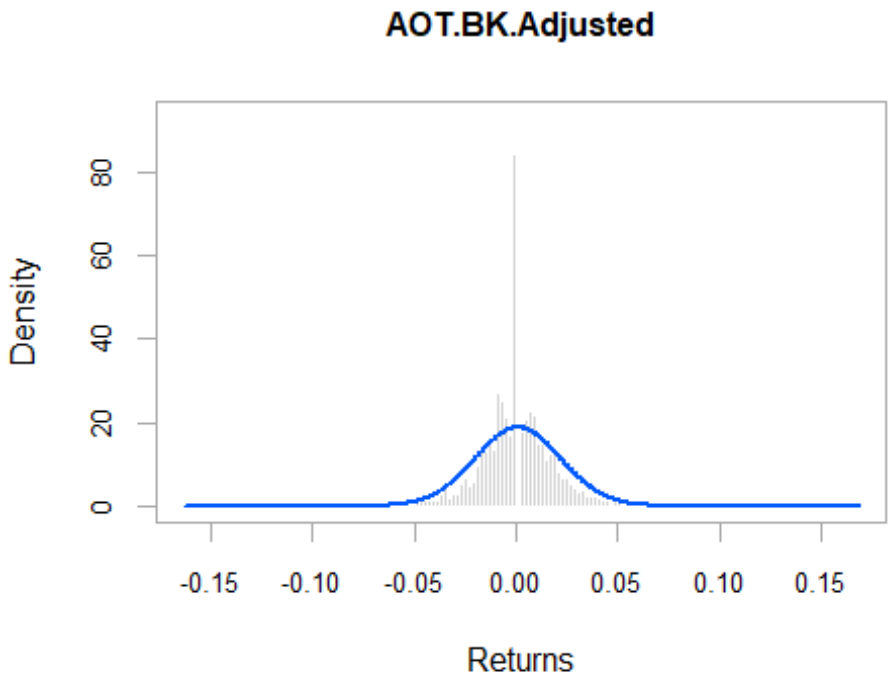





```
newlogreturnA <- logreturnA[2:nrow(logreturnA),]
newsimplereturnA <- simplereturnA[2:nrow(logreturnA),]

newlogreturnCAT <- logreturnCAT[2:nrow(logreturnCAT),]
newsimplereturnCAT <- simplereturnCAT[2:nrow(logreturnCAT),]

#statistics log
par(mfrow=c(1,1))
hist(logreturnA, breaks=100, col="slateblue")
```



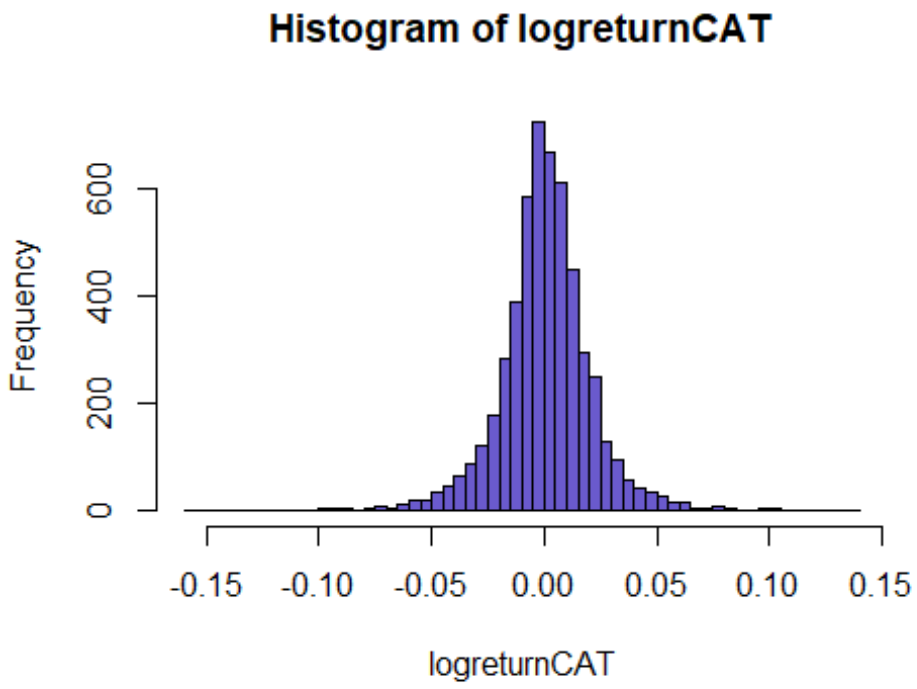

```
chart.Histogram(logreturnA,methods = c("add.normal"))
```



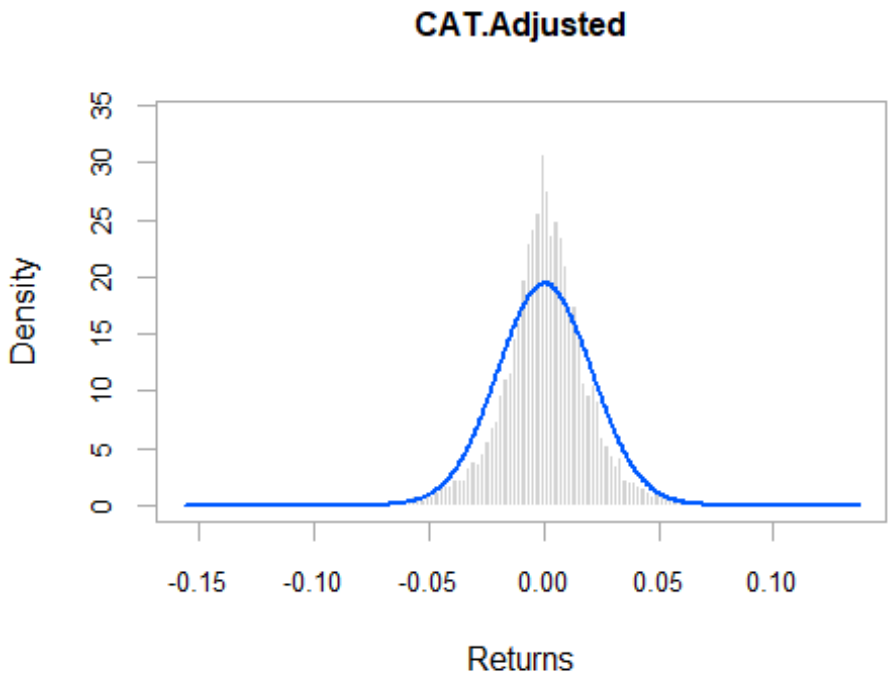

```
table.Stats(logreturnA)
```

```
##                  AOT.BK.Adjusted
## Observations           4158.0000
## NAs                       1.0000
## Minimum                  -0.1632
## Quartile 1               -0.0093
## Median                    0.0000
## Arithmetic Mean           0.0007
## Geometric Mean            0.0005
## Quartile 3                0.0102
## Maximum                   0.1684
## SE Mean                   0.0003
## LCL Mean (0.95)           0.0001
## UCL Mean (0.95)           0.0014
## Variance                  0.0004
## Stdev                     0.0212
## Skewness                  0.1746
## Kurtosis                  9.6096
```

```
par(mfrow=c(1,1))
hist(logreturnCAT, breaks=100, col="slateblue")
```



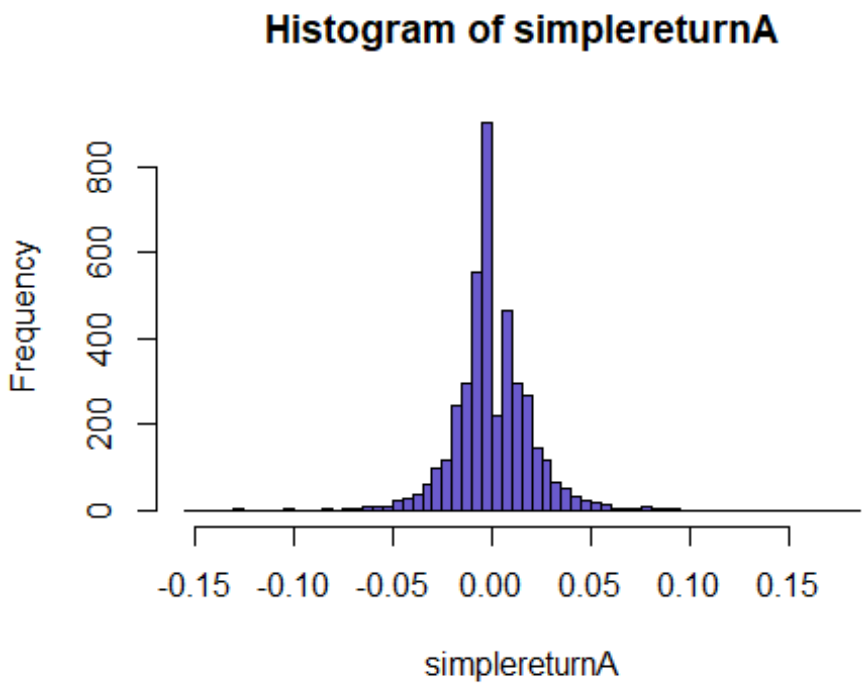

```
chart.Histogram(logreturnCAT,methods = c("add.normal"))
```



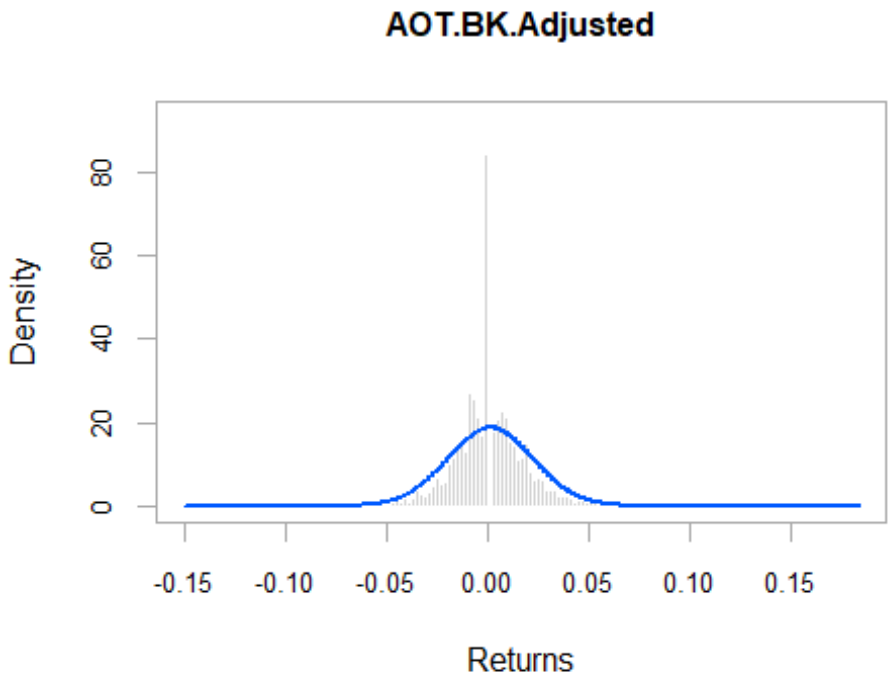

```
table.Stats(logreturnCAT)
```

```
##                  CAT.Adjusted
## Observations       5302.0000
## NAs                   1.0000
## Minimum              -0.1569
## Quartile 1           -0.0095
## Median                0.0005
## Arithmetic Mean       0.0005
## Geometric Mean        0.0003
## Quartile 3            0.0110
## Maximum               0.1373
## SE Mean               0.0003
## LCL Mean (0.95)      -0.0001
## UCL Mean (0.95)       0.0010
## Variance              0.0004
## Stdev                 0.0205
## Skewness             -0.1836
## Kurtosis              4.6982
```

```
#simple return

par(mfrow=c(1,1))
hist(simplereturnA, breaks=100, col="slateblue")
```



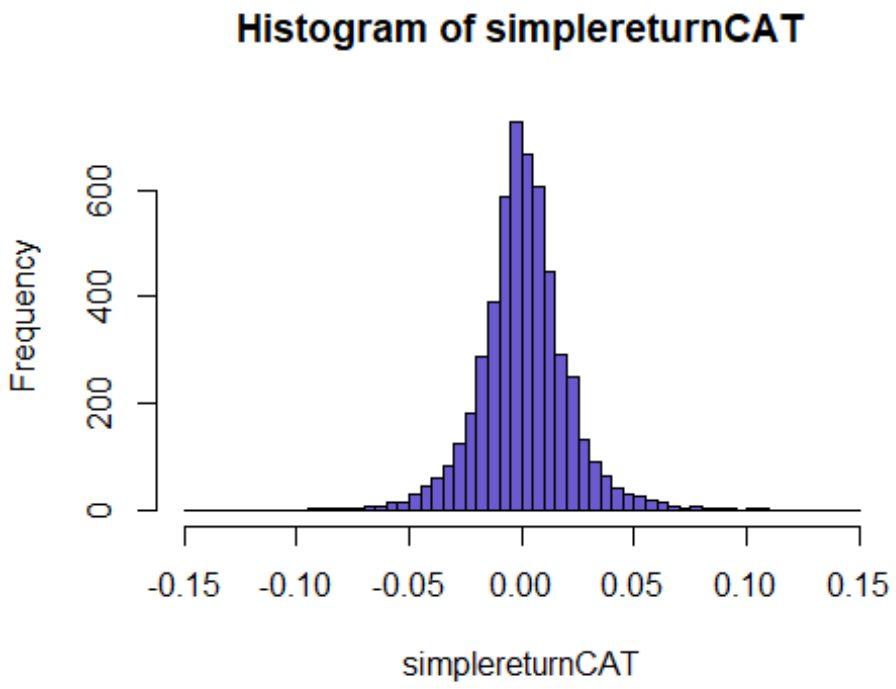

```
chart.Histogram(simplereturnA,methods = c("add.normal"))
```



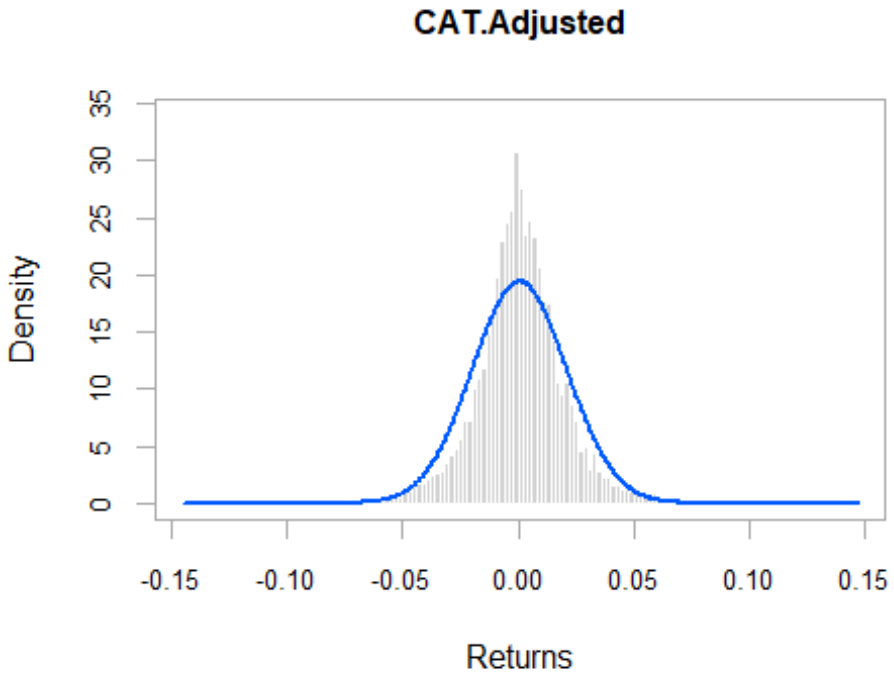

```
table.Stats(simplereturnA)
```

```
##                  AOT.BK.Adjusted
## Observations           4158.0000
## NAs                       1.0000
## Minimum                  -0.1505
## Quartile 1               -0.0093
## Median                    0.0000
## Arithmetic Mean           0.0010
## Geometric Mean            0.0007
## Quartile 3                0.0103
## Maximum                   0.1834
## SE Mean                   0.0003
## LCL Mean (0.95)           0.0003
## UCL Mean (0.95)           0.0016
## Variance                  0.0005
## Stdev                     0.0213
## Skewness                  0.5443
## Kurtosis                 10.0827
```

```
par(mfrow=c(1,1))
hist(simplereturnCAT, breaks=100, col="slateblue")
```



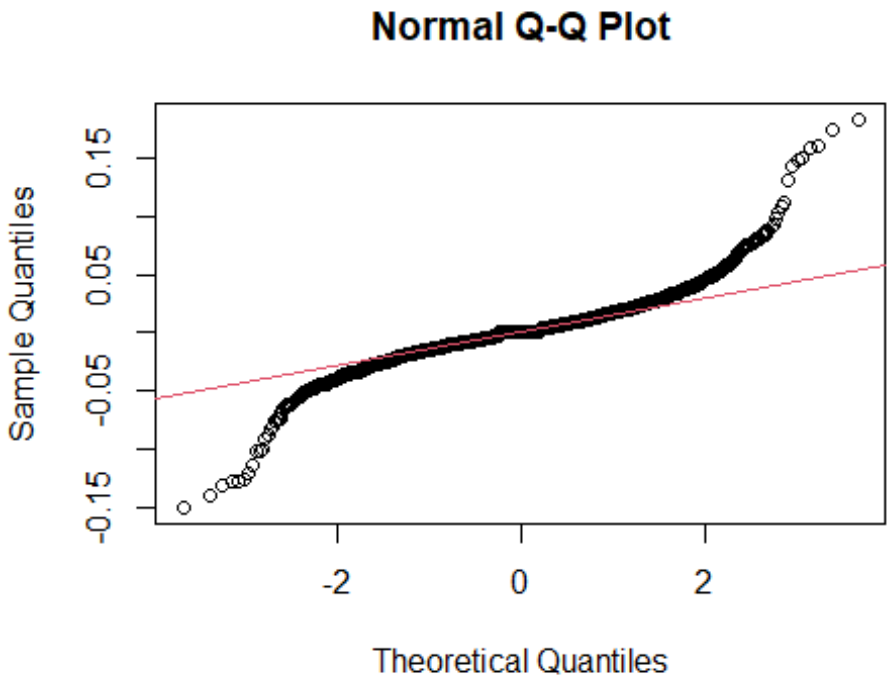

```
chart.Histogram(simplereturnCAT,methods = c("add.normal"))
```



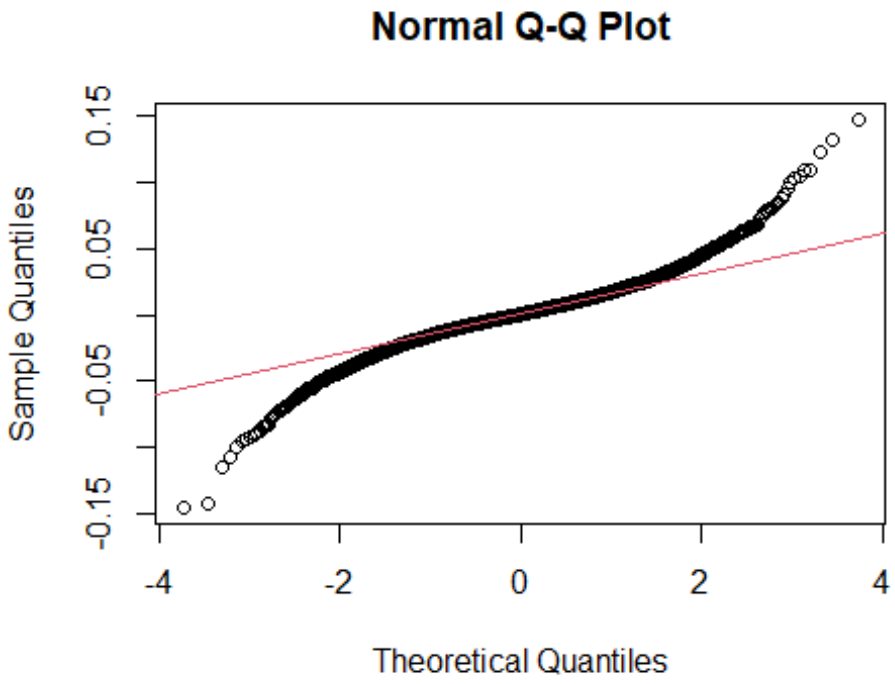

```
table.Stats(simplereturnCAT)
```

```
##                  CAT.Adjusted
## Observations       5302.0000
## NAs                   1.0000
## Minimum              -0.1452
## Quartile 1           -0.0095
## Median                0.0005
## Arithmetic Mean       0.0007
## Geometric Mean        0.0005
## Quartile 3            0.0110
## Maximum               0.1472
## SE Mean               0.0003
## LCL Mean (0.95)       0.0001
## UCL Mean (0.95)       0.0013
## Variance              0.0004
## Stdev                 0.0205
## Skewness              0.0197
## Kurtosis              4.5480
```

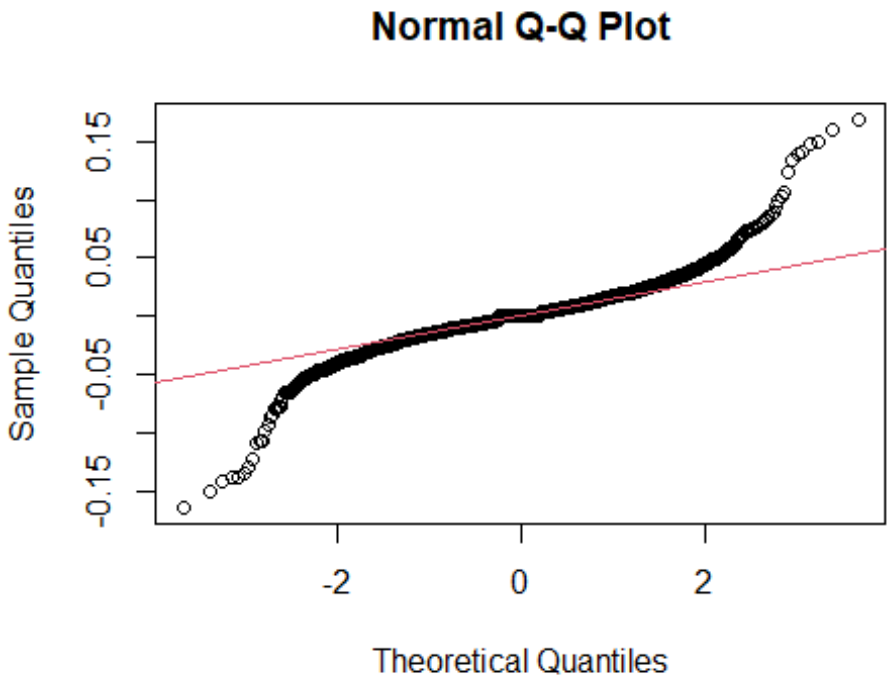
```
#QQ-plots and tests for normality

par(mfrow=c(1,1))
qqnorm(newsimplereturnA)
qqline(newsimplereturnA, col = 2)
```



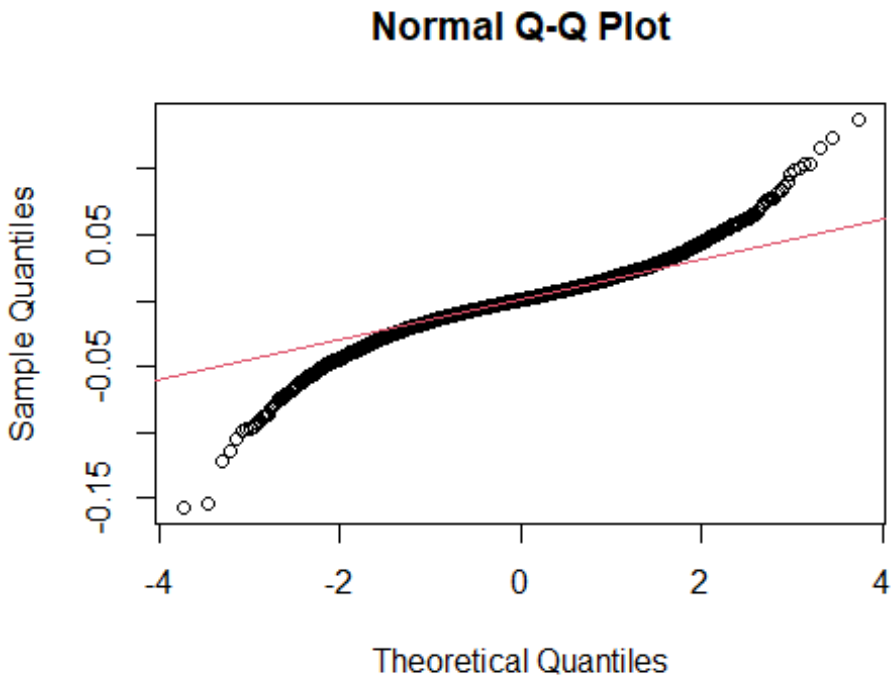

```
jarque.bera.test(newsimplereturnA)

##
##  Jarque Bera Test
##
## data:  newsimplereturnA
## X-squared = 17818, df = 2, p-value < 2.2e-16

par(mfrow=c(1,1))
qqnorm(newsimplereturnCAT)
qqline(newsimplereturnCAT, col = 2)
```




```
jarque.bera.test(newsimplereturnCAT)

##
##  Jarque Bera Test
##
## data:  newsimplereturnCAT
## X-squared = 4569.9, df = 2, p-value < 2.2e-16

par(mfrow=c(1,1))
qqnorm(newlogreturnA)
qqline(newlogreturnA, col = 2)
```




```
jarque.bera.test(newlogreturnA)

##
##  Jarque Bera Test
##
## data:  newlogreturnA
## X-squared = 16020, df = 2, p-value < 2.2e-16

par(mfrow=c(1,1))
qqnorm(newlogreturnCAT)
qqline(newlogreturnCAT, col = 2)
```




```
jarque.bera.test(newlogreturnCAT)
```

```
##
##  Jarque Bera Test
##
## data:  newlogreturnCAT
## X-squared = 4906.1, df = 2, p-value < 2.2e-16
```

```
#Test mean
t.test(newlogreturnCAT)
```

```
## Warning in tstat + c(-cint, cint): Recycling array of length 1 in array-
vector arithmetic is deprecated.
##   Use c() or as.vector() instead.

## Warning in cint * stderr: Recycling array of length 1 in vector-array
arithmetic is deprecated.
##   Use c() or as.vector() instead.

##
##  One Sample t-test
##
## data:  newlogreturnCAT
## t = 1.7296, df = 5301, p-value = 0.08377
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -6.513168e-05  1.041069e-03
## sample estimates:
```

```
##    mean of x
## 0.0004879685

t.test(newlogreturnA)

## Warning in tstat + c(-cint, cint): Recycling array of length 1 in array-
vector arithmetic is deprecated.
##   Use c() or as.vector() instead.

## Warning in tstat + c(-cint, cint): Recycling array of length 1 in vector-
array arithmetic is deprecated.
##   Use c() or as.vector() instead.

##
##  One Sample t-test
##
## data:  newlogreturnA
## t = 2.2696, df = 4157, p-value = 0.02328
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.0001016629 0.0013915769
## sample estimates:
##    mean of x
## 0.0007466199

#Test Skewness
T=length(newlogreturnCAT)
m3=skewness(newlogreturnCAT)
m3

## [1] -0.1836344

tst = m3/sqrt(6/T)
tst

## [1] -5.458812

pv = 2*pnorm(tst)
pv

## [1] 4.793299e-08

#6 Test excess kurtosis =0
K = kurtosis(newlogreturnA)
tst = K/sqrt(24/T)
tst

## [1] 142.8296

pv = 2*(1-pnorm(tst))
pv

## [1] 0
```