Krittaphong Kaewvichian

6104640609

Assignment-3.R

pardus

2021-03-02

```
#Krittaphong Kaewvichian ID 6104640609
setwd("/Users/pardus/Desktop/EE435/CSV")
cat(rep("\n",50))  #clear R Console
```

```
#update.packages(ask = FALSE, checkBuilt = TRUE)
#tinytex::tlmgr_update()
#tinytex::reinstall_tinytex()
#options(tinytex.verbose = TRUE)
#install.packages('plyr', repos = "http://cran.us.r-project.org")
#install.packages("tinytex")
#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
```

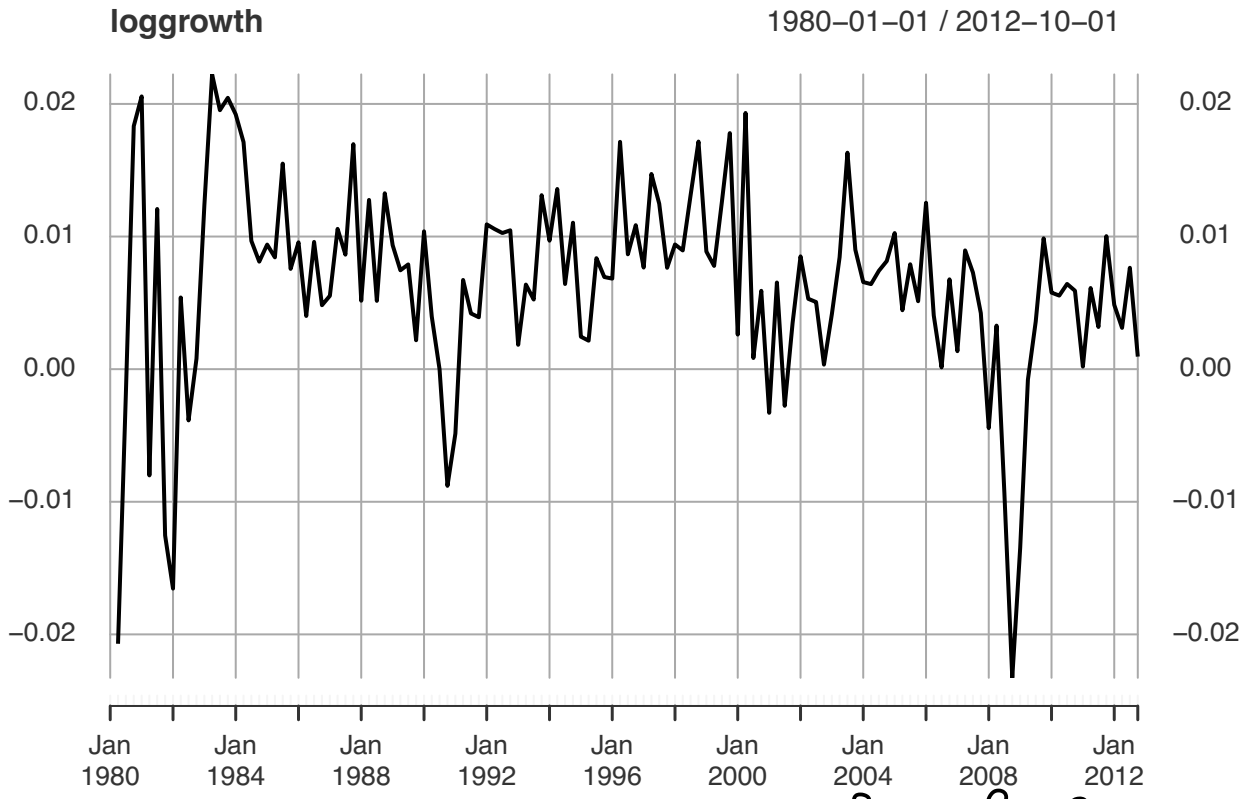
```
## Loading required package: timeSeries
```

```
##
## 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)
library(xts)
#1. QGDP ###################################################################

qgdpmaster <- read.zoo("qgdp.csv", header = TRUE, sep = ",",)
qgdp_xts <- as.xts(qgdpmaster)
da <- qgdp_xts[paste("1980", "2012", sep="/"),]
logda = log(da)
loggrowth = diff(log(da))
#a Plot
plot(loggrowth,theme="White")
```

I've cleaned the data before import it into R
txt → csv (Date)




```
#b Serial Correlation Testing


Box.test(loggrowth,lag=12, type='Ljung')
```

$H_0: \rho_1 = \rho_2 = \dots = \rho_{12} = 0$
$H_a: \rho_i \neq 0 \; ; \; i \in \{1, \dots, 12\}$
$\alpha = 0.05$
$p\text{-value}_{cal} < 0.05$
since p-value falls into rejection region at 0.05 level of confidence
We reject H_0. That is, some $\rho_i \neq 0$

```
##
##  Box-Ljung test
##
## data:  loggrowth
## X-squared = 43.083, df = 12, p-value = 2.186e-05
#c Mean Testing


t.test(loggrowth,mu=0)
```

```
## Warning in tstat + c(-cint, cint): Recycling array of length 1 in array-vector arithmetic is deprecat
##   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:  loggrowth
## t = 9.6124, df = 130, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.005088468 0.007725849
## sample estimates:
```

95% CI

$H_0: \mu = 0$
$H_a: \mu \neq 0$
$\alpha = 0.05$
$p\text{-value}_{cal} < 0.05$
since p-value falls into rejection region at 0.05 level of confidence
We reject H_0. That is, $\mu \neq 0$

```
##   mean of x
## 0.006407159
```

```
#2.AMZN ###################################################################
AMZNmaster <- read.zoo("AMZN3.csv", header = TRUE, sep = ",",)
amzn3_xts <- as.xts(AMZNmaster)
damzn3 <- amzn3_xts
#a Basic Statistics
basicStats(damzn3)
```

```
##                     amzn       vwrtn       ewrtn       sprtn
## nobs        1259.000000 1259.000000 1259.000000 1259.000000
## NAs            0.000000    0.000000    0.000000    0.000000
## Minimum       -0.127820   -0.089763   -0.078240   -0.090350
## Maximum        0.267951    0.114895    0.107422    0.115800
## 1. Quartile   -0.013195   -0.006328   -0.006365   -0.006301
## 3. Quartile    0.014571    0.007376    0.007780    0.006978
## Mean           0.001207    0.000210    0.000517    0.000114
## Median        -0.000451    0.000818    0.001210    0.000685
## Sum            1.519912    0.264749    0.651089    0.143876
## SE Mean        0.000820    0.000469    0.000434    0.000467
## LCL Mean      -0.000401   -0.000710   -0.000334   -0.000802
## UCL Mean       0.002816    0.001130    0.001368    0.001031
## Variance       0.000846    0.000277    0.000237    0.000275
## Stdev          0.029087    0.016641    0.015395    0.016573
## Skewness       1.031884   -0.113083   -0.168185   -0.018307
## Kurtosis       9.360219    6.298940    5.275824    7.154386
```

```
#b Transform the simple return to logreturn
s_amzn <- damzn3[,1]
s_vwrtn <- damzn3[,2]
s_ewrtn <- damzn3[,3]
s_prtn <- damzn3[,4]

logamzn = log(1+s_amzn)
logvwrtn = log(1+s_vwrtn)
logewrtn = log(1+s_ewrtn)
logprtn = log(1+s_prtn)

basicStats(logamzn)
```

```
##                     amzn
## nobs        1259.000000
## NAs            0.000000
## Minimum       -0.136759
## Maximum        0.237402
## 1. Quartile   -0.013283
## 3. Quartile    0.014466
## Mean           0.000791
## Median        -0.000451
## Sum            0.996211
## SE Mean        0.000810
## LCL Mean      -0.000798
## UCL Mean       0.002380
## Variance       0.000826
## Stdev          0.028736
```

```
## Skewness       0.627148
## Kurtosis       7.249289
```

```
basicStats(logvwrtn)
```

```
##                   vwrtn
## nobs        1259.000000
## NAs            0.000000
## Minimum       -0.094050
## Maximum        0.108760
## 1. Quartile   -0.006348
## 3. Quartile    0.007348
## Mean           0.000072
## Median         0.000818
## Sum            0.090160
## SE Mean        0.000470
## LCL Mean      -0.000850
## UCL Mean       0.000993
## Variance       0.000278
## Stdev          0.016673
## Skewness      -0.317276
## Kurtosis       6.194861
```

```
basicStats(logewrtn)
```

```
##                   ewrtn
## nobs        1259.000000
## NAs            0.000000
## Minimum       -0.081470
## Maximum        0.102035
## 1. Quartile   -0.006385
## 3. Quartile    0.007749
## Mean           0.000398
## Median         0.001209
## Sum            0.501599
## SE Mean        0.000435
## LCL Mean      -0.000454
## UCL Mean       0.001251
## Variance       0.000238
## Stdev          0.015420
## Skewness      -0.333442
## Kurtosis       5.193084
```

```
basicStats(logprtn)
```

```
##                   sprtn
## nobs        1259.000000
## NAs            0.000000
## Minimum       -0.094695
## Maximum        0.109572
## 1. Quartile   -0.006320
## 3. Quartile    0.006954
## Mean          -0.000023
## Median         0.000685
## Sum           -0.029133
## SE Mean        0.000468
```

```
## LCL Mean      -0.000941
## UCL Mean       0.000894
## Variance       0.000275
## Stdev          0.016594
## Skewness      -0.242090
## Kurtosis       6.952089
```

```
#c Test logamzn is zero
t.test(logamzn)
```

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

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

##
##  One Sample t-test
##
## data:  logamzn
## t = 0.97705, df = 1258, p-value = 0.3287
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0007975567  0.0023801007
## sample estimates:
##   mean of x
## 0.000791272
```

$H_0: \mu_{rt} = 0$

$H_a: \mu_{rt} \neq 0$

$\alpha = 0.05$

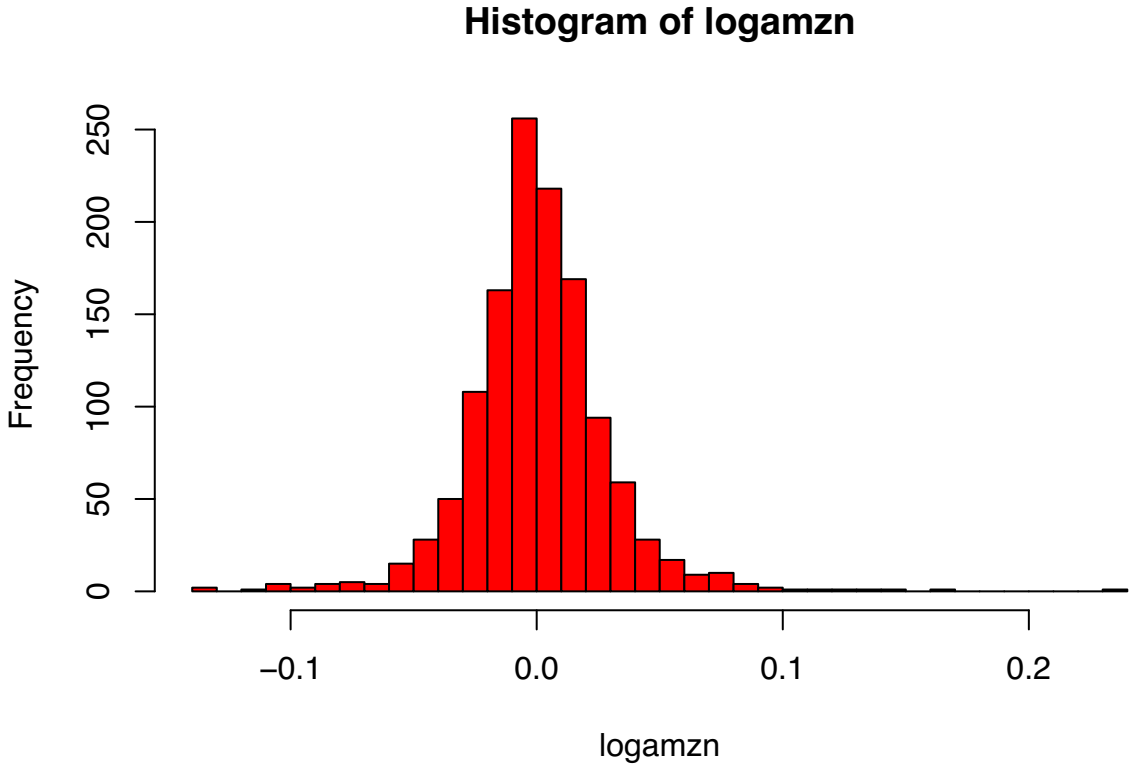
P-value does not into rejection region at 0.05 level of confidence

We do not reject. That is, some $\mu_{rt} = 0$ H_0

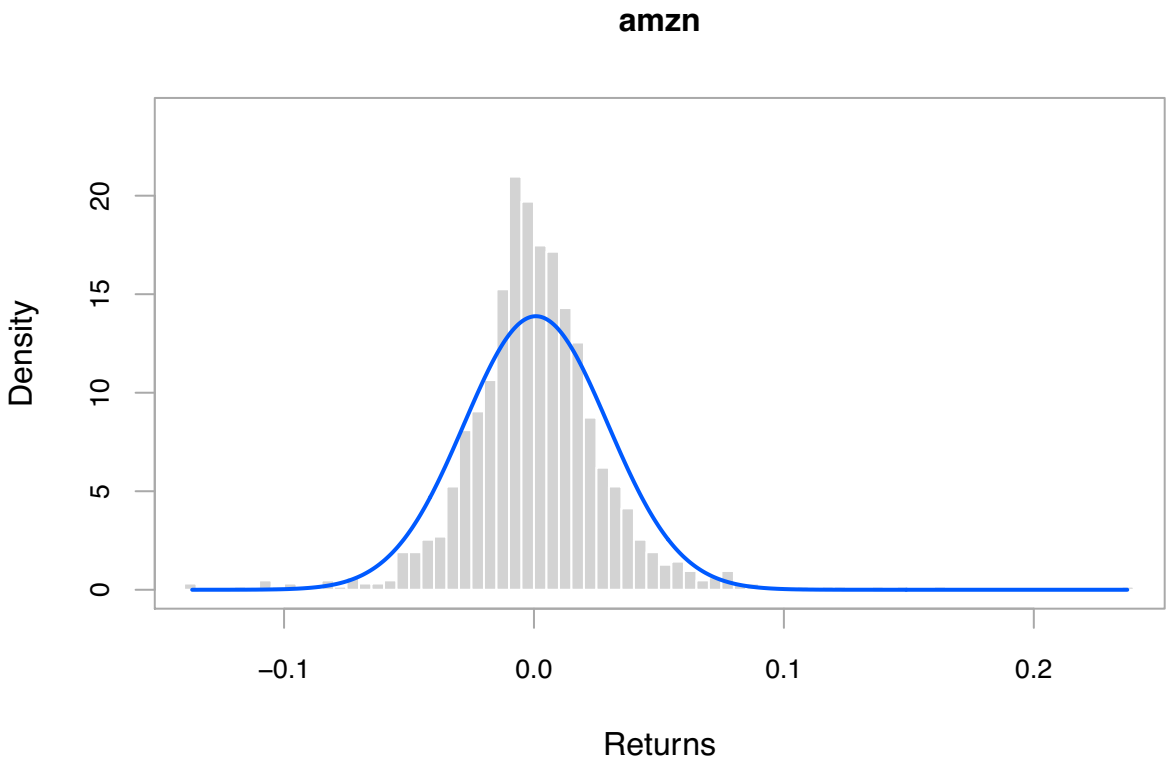
↓

① weakly stationarity

```
#d histogram (nclass=40)
par(mfrow=c(1,1))
hist(logamzn, breaks=40, col="red")
```



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




```
#3.Abbott#############################################################
ABTmaster <- read.zoo("Abt3dx.csv", header = TRUE, sep = ",",)
```

```
## Warning in zoo(rval3, ix): some methods for "zoo" objects do not work if the
## index entries in 'order.by' are not unique
```

```
abt_xts <- as.xts(ABTmaster)
daabt <- abt_xts
#a Basic Statistics
basicStats(daabt)
```

```
##                   RET     vwretd     ewretd     sprtrn
## nobs        492.000000 492.000000 492.000000 492.000000
## NAs           0.000000   0.000000   0.000000   0.000000
## Minimum      -0.234146  -0.225363  -0.272248  -0.217630
## Maximum       0.382326   0.165585   0.299260   0.163047
## 1. Quartile  -0.024230  -0.017196  -0.020694  -0.018948
## 3. Quartile   0.055840   0.039484   0.044462   0.035349
## Mean          0.014073   0.009020   0.011583   0.006386
## Median        0.014083   0.012626   0.014379   0.009091
## Sum           6.923863   4.437612   5.699074   3.142113
## SE Mean       0.002921   0.002089   0.002583   0.002019
## LCL Mean      0.008333   0.004915   0.006508   0.002419
## UCL Mean      0.019812   0.013124   0.016659   0.010354
## Variance      0.004198   0.002147   0.003283   0.002006
## Stdev         0.064795   0.046341   0.057299   0.044788
## Skewness      0.097058  -0.555341  -0.200025  -0.444303
## Kurtosis      2.469971   2.073657   3.288104   1.889777
```

```r
#b Transform the simple return to logreturn
s_RET <- daabt[,1]
s_VWRETD <- daabt[,2]
s_EWRETD <- daabt[,3]
s_SPRTRN <- daabt[,4]

logabt = log(1+s_RET)
logvwretd = log(1+s_VWRETD)
logewretd = log(1+s_EWRETD)
logsprtrn = log(1+s_SPRTRN)

basicStats(logabt)
```

```
##                    RET
## nobs        492.000000
## NAs           0.000000
## Minimum      -0.266764
## Maximum       0.323768
## 1. Quartile  -0.024528
## 3. Quartile   0.054337
## Mean          0.011924
## Median        0.013985
## Sum           5.866748
## SE Mean       0.002898
## LCL Mean      0.006230
## UCL Mean      0.017619
## Variance      0.004133
## Stdev         0.064287
## Skewness     -0.297125
## Kurtosis      2.005416
```

```r
basicStats(logvwretd)
```

```
##                  vwretd
## nobs        492.000000
## NAs           0.000000
## Minimum      -0.255361
## Maximum       0.153223
## 1. Quartile  -0.017345
## 3. Quartile   0.038724
## Mean          0.007903
## Median        0.012547
## Sum           3.888033
## SE Mean       0.002107
## LCL Mean      0.003762
## UCL Mean      0.012043
## Variance      0.002185
## Stdev         0.046745
## Skewness     -0.839542
## Kurtosis      3.003331
```

```r
basicStats(logewretd)
```

```
##                  ewretd
## nobs        492.000000
```

```
## NAs           0.000000
## Minimum      -0.317795
## Maximum       0.261795
## 1. Quartile  -0.020911
## 3. Quartile   0.043501
## Mean          0.009887
## Median        0.014277
## Sum           4.864341
## SE Mean       0.002592
## LCL Mean      0.004794
## UCL Mean      0.014979
## Variance      0.003305
## Stdev         0.057491
## Skewness     -0.662276
## Kurtosis      3.976081
```

```
basicStats(logsprtrn)
```

```
##                 sprtrn
## nobs        492.000000
## NAs           0.000000
## Minimum      -0.245428
## Maximum       0.151043
## 1. Quartile  -0.019130
## 3. Quartile   0.034738
## Mean          0.005360
## Median        0.009050
## Sum           2.636920
## SE Mean       0.002035
## LCL Mean      0.001361
## UCL Mean      0.009359
## Variance      0.002038
## Stdev         0.045145
## Skewness     -0.711215
## Kurtosis      2.659018
```

```
#c Test logamzn is zero
t.test(logabt)
```

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

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

##
##  One Sample t-test
##
## data:  logabt
## t = 4.1143, df = 491, p-value = 4.555e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.006229721 0.017618849
## sample estimates:
##  mean of x
## 0.01192429
```

$r_t = \log(\text{return})$

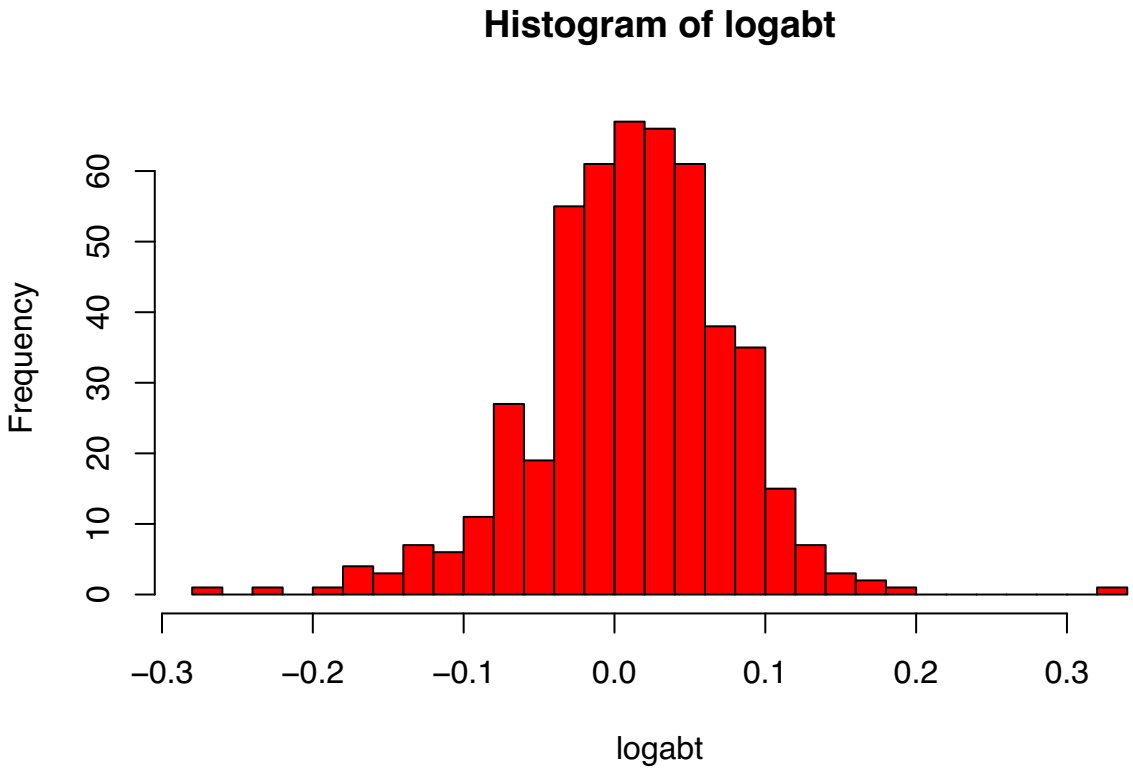
$H_0: r_t = 0$

$H_a: r_t \neq 0$

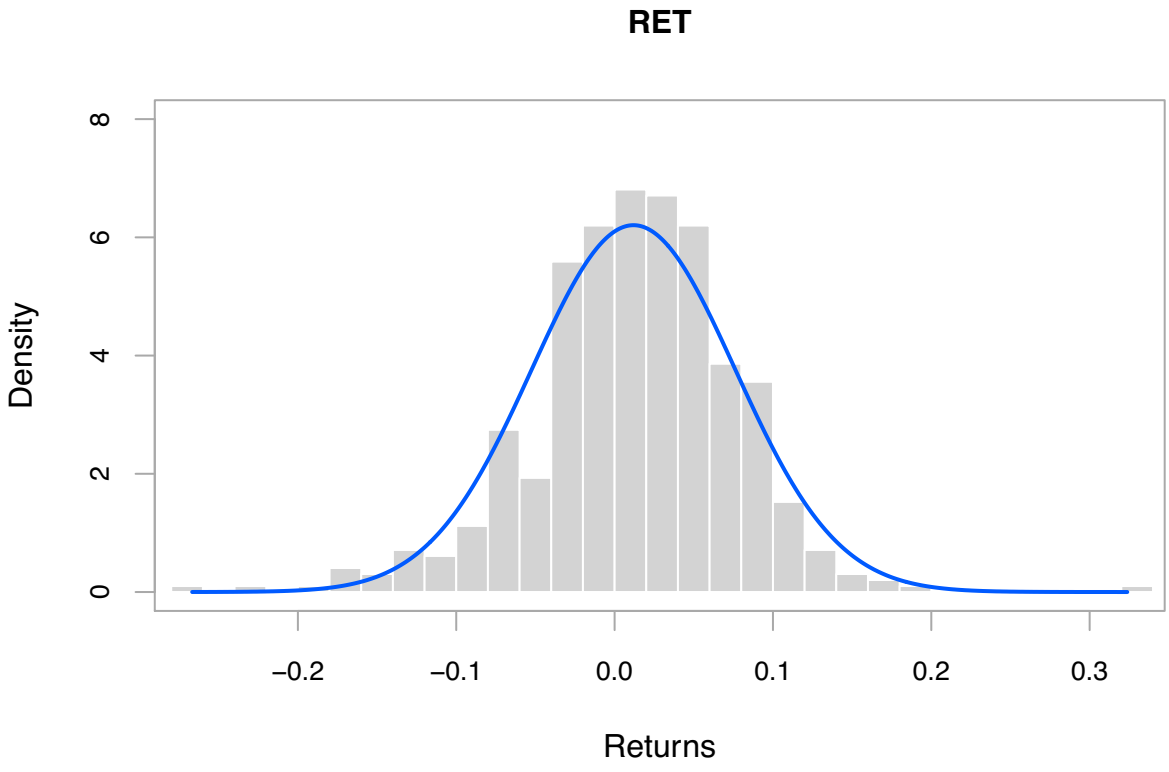
$\alpha = 0.05$, $p\text{-value}_{cal} < 0.05$

since p-value falls into rejection region at 0.05 level of confidence We reject H_0. That is, some $r_t \neq 0$

```r
#d histogram (nclass=40)
par(mfrow=c(1,1))
hist(logabt, breaks=40, col="red")
```




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



```
#4 VW problem ###############################################################
#a
s3=skewness(logamzn)
T=length(logamzn)
tst=s3/sqrt(6/T)
tst
```

$H_0: S(r_t) = 0$
$H_a: S(r_t) \neq 0$
$\alpha = 0.05$

```
## [1] 9.095459
```

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

since p-value falls into rejection region at 0.05 level of confidence
We reject H_0. That is, some $S(r_t) \neq 0$ ✓

```
## [1] 0
```

```
#b
k4=kurtosis(logamzn)
T=length(logamzn)
tst=k4/sqrt(24/T)
tst
```

$H_0: K(r_t) - 3 \leq 0$
$H_a: K(r_t) - 3 > 0$
$\alpha = 0.05$

```
## [1] 52.62331
```

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

since p-value falls into rejection region at 0.05 level of confidence
We reject H_0. That is, some $K(r_t) - 3 > 0$
$K(r_t) > 3$

```
## [1] 0 ✓
```

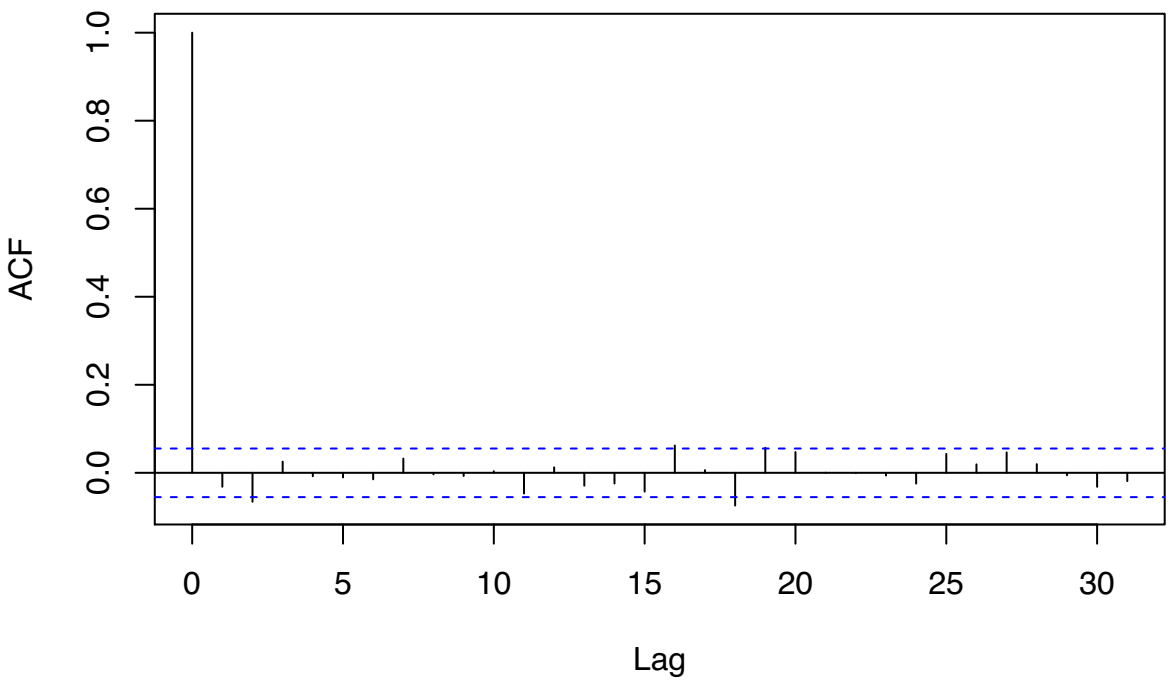
```
#c
t.test(logamzn)
```

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

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

##
##  One Sample t-test
##
## data:  logamzn
## t = 0.97705, df = 1258, p-value = 0.3287 ✓
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.0007975567  0.0023801007
## sample estimates:
##   mean of x
## 0.000791272
```

$H_0: \mu_{r_t} = 0$
$H_a: \mu_{r_t} \neq 0$
$\alpha = 0.05$

does not fall ∧ into rejection region at 0.05 level of confidence
since p-value
We ✓ H_0. That is, $\mu_{r_t} = 0$
do not

```
#b
m1=acf(logamzn)
```

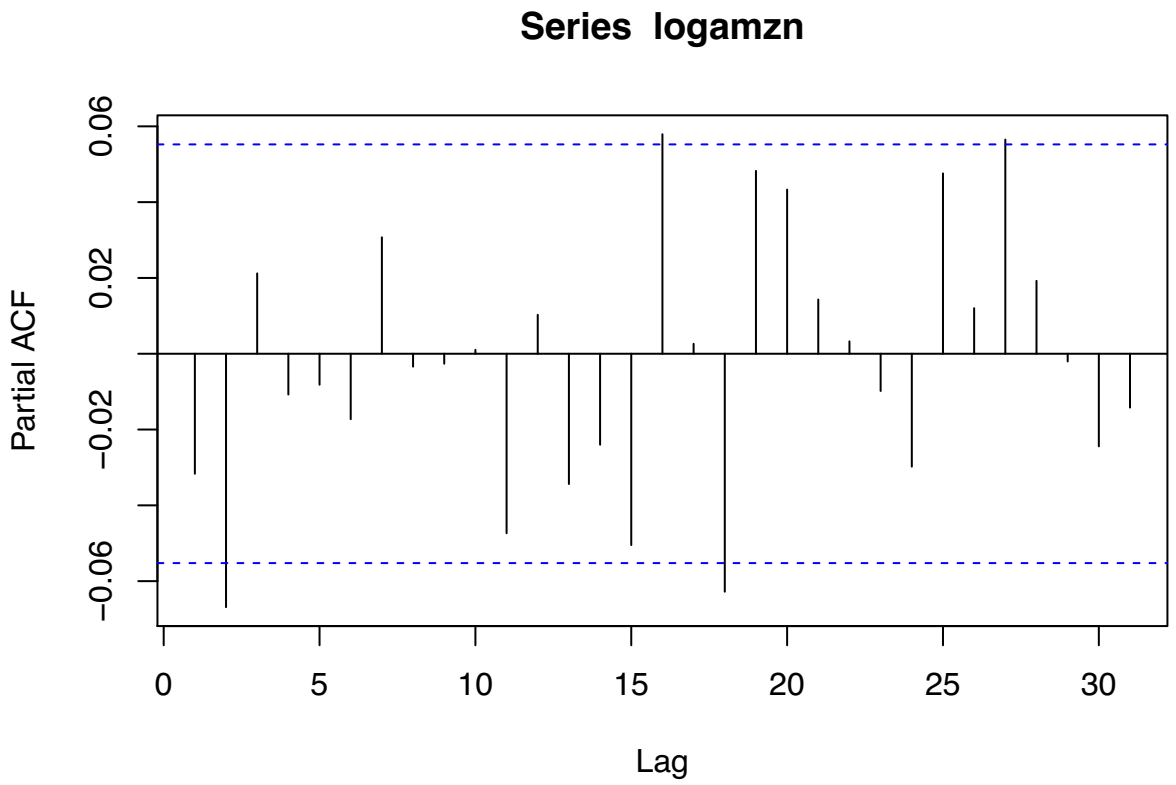



```
m1$acf
```

```
## , , 1
##
##                [,1]
##  [1,]  1.0000000000
##  [2,] -0.0316614625
##  [3,] -0.0657963636
##  [4,]  0.0254218662
##  [5,] -0.0078449975
##  [6,] -0.0106049170
##  [7,] -0.0148006161
##  [8,]  0.0323683041
##  [9,] -0.0036167194
## [10,] -0.0071029841
## [11,]  0.0036460656
## [12,] -0.0470650529
## [13,]  0.0124225728
## [14,] -0.0292344042
## [15,] -0.0243312999
## [16,] -0.0427496969
## [17,]  0.0620802986
## [18,]  0.0061473042
## [19,] -0.0744147656
## [20,]  0.0568749860
## [21,]  0.0471836320
## [22,]  0.0003109035
## [23,] -0.0006986516
## [24,] -0.0062326764
```

```
## [25,] -0.0244831099
## [26,]  0.0430134379
## [27,]  0.0192213377
## [28,]  0.0464068477
## [29,]  0.0197592080
## [30,] -0.0055967072
## [31,] -0.0317808897
## [32,] -0.0189335815
```

```
m2=pacf(logamzn)
```




```
m2$acf
```

```
## , , 1
##
##                [,1]
##  [1,] -0.031661462
##  [2,] -0.066865841
##  [3,]  0.021198224
##  [4,] -0.010767266
##  [5,] -0.008175617
##  [6,] -0.017271554
##  [7,]  0.030726846
##  [8,] -0.003403192
##  [9,] -0.002629988
## [10,]  0.001032256
## [11,] -0.047363651
## [12,]  0.010284023
## [13,] -0.034370971
## [14,] -0.024008840
```

```
## [15,] -0.050490587
## [16,]  0.057915450
## [17,]  0.002614565
## [18,] -0.062793034
## [19,]  0.048259706
## [20,]  0.043316077
## [21,]  0.014313942
## [22,]  0.003269419
## [23,] -0.009825781
## [24,] -0.029795895
## [25,]  0.047586551
## [26,]  0.012023445
## [27,]  0.056509305
## [28,]  0.019236094
## [29,] -0.002035994
## [30,] -0.024437113
## [31,] -0.014212519
```

```
Box.test(logamzn,lag=12,type='Ljung')
```

$H_0: \rho_1 = \rho_2 = \ldots = \rho_{12} = 0$

$H_a: \rho_i \neq 0 \; ; \; i \in \{1, \ldots 12\}$

$\alpha = 0.05$

since p-value does not into rejection region at 0.05 level of confidence We don't H_0. That is, some $\rho_i \neq 0$

We can use the past to forecast.

```
##
##  Box-Ljung test
##
## data:  logamzn
## X-squared = 12.488, df = 12, p-value = 0.4073
```

```
#6
#a
daex <- read.zoo("Exuseu.csv",header = TRUE, sep = ",")
```

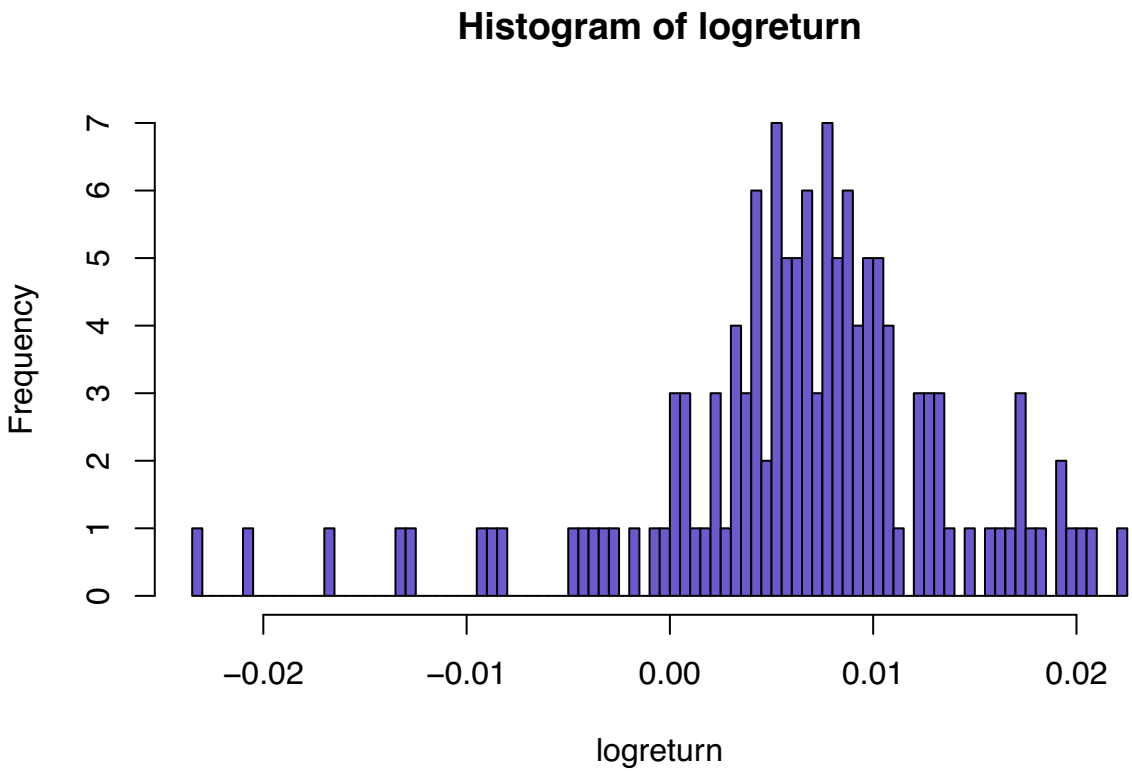
```
## Warning in zoo(rval3, ix): some methods for "zoo" objects do not work if the
## index entries in 'order.by' are not unique
```

```
daex_xts <- as.xts(daex)
srate=da[,1]
lograte=log(srate)
logreturn=diff(lograte)

#b
basicStats(logreturn)
```

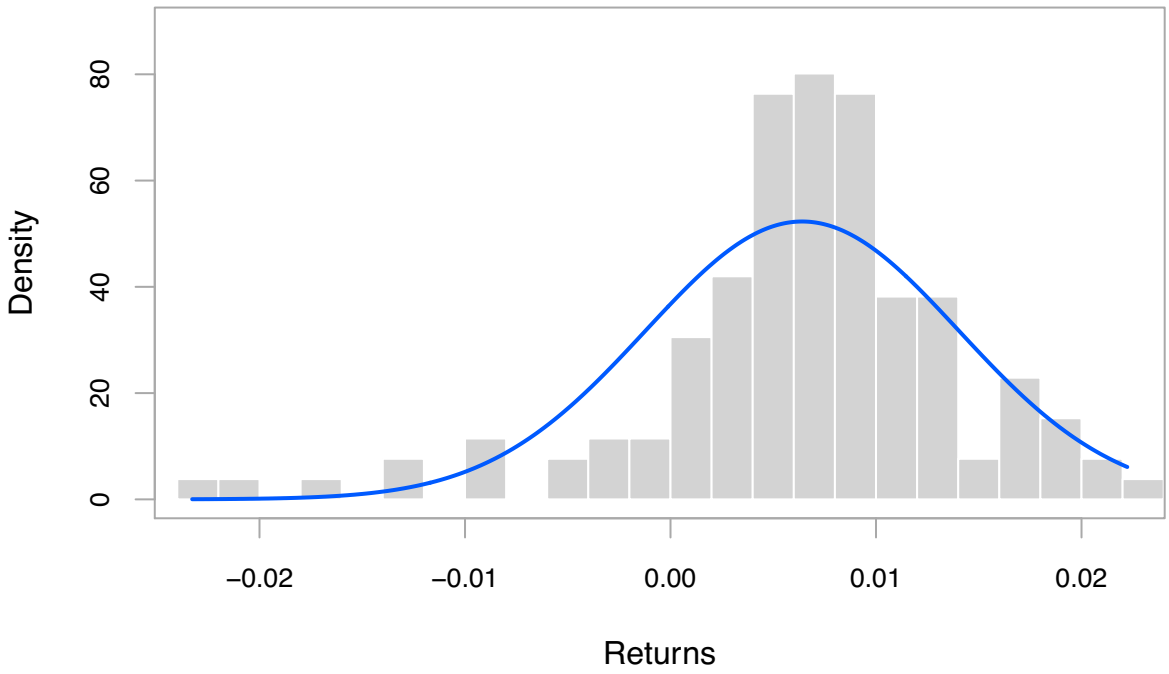
```
##                     x
## nobs        132.000000
## NAs           1.000000
## Minimum      -0.023276
## Maximum       0.022228
## 1. Quartile   0.003753
## 3. Quartile   0.010263
## Mean          0.006407
## Median        0.006948
## Sum           0.839338
## SE Mean       0.000667
## LCL Mean      0.005088
## UCL Mean      0.007726
## Variance      0.000058
## Stdev         0.007629
```

```
## Skewness      -1.060928
## Kurtosis       2.570377
```

```
#c
hist(logreturn, breaks=132, col="slateblue")
```




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



```r
#d
t.test(logreturn)
```

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

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

##
##  One Sample t-test
##
## data:  logreturn
## t = 9.6124, df = 130, p-value < 2.2e-16
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.005088468 0.007725849
## sample estimates:
##   mean of x
## 0.006407159
```

$H_0: \mu_{rf} = 0$

$H_a: \mu_{rf} \neq 0$

$\alpha = 0.05$

since p-value falls into rejection region at 0.05 level of confidence

We reject H_0. That is, $r_f \neq 0$