Minitest1_TanapatInpringananta.R

ryu_r

2021-02-11

```
#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
```

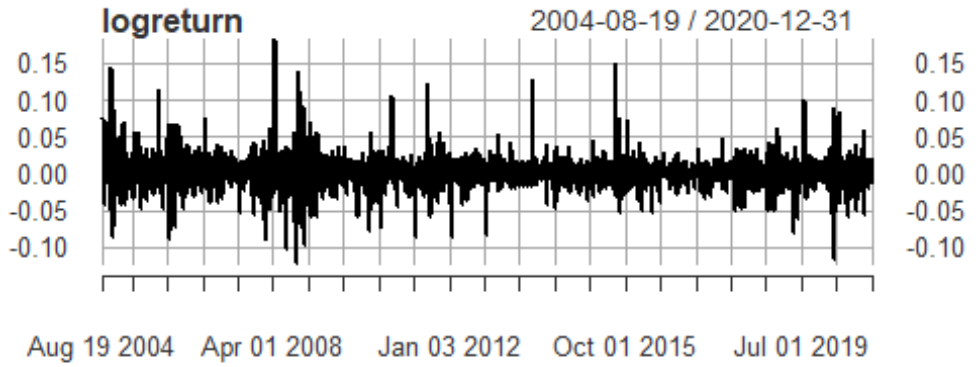
```
##
## Attaching package: 'car'
```

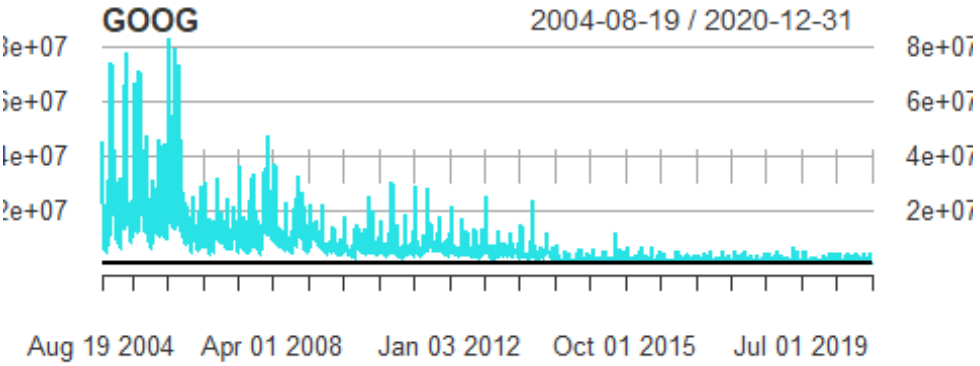
```
## The following object is masked from 'package:fBasics':
##
##     densityPlot
```

```
library(tseries)
library(forecast)

#GOOG
getSymbols("GOOG",from="2004-08-19",to="2021-01-01")
```

```
## '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] "GOOG"

da=GOOG
GOOG_price=da[,6]
logprice=log(GOOG_price)
logreturn=diff(log(GOOG_price))
par(mfrow=c(2,1))
plot(logreturn,type='l')
plot(GOOG)
```

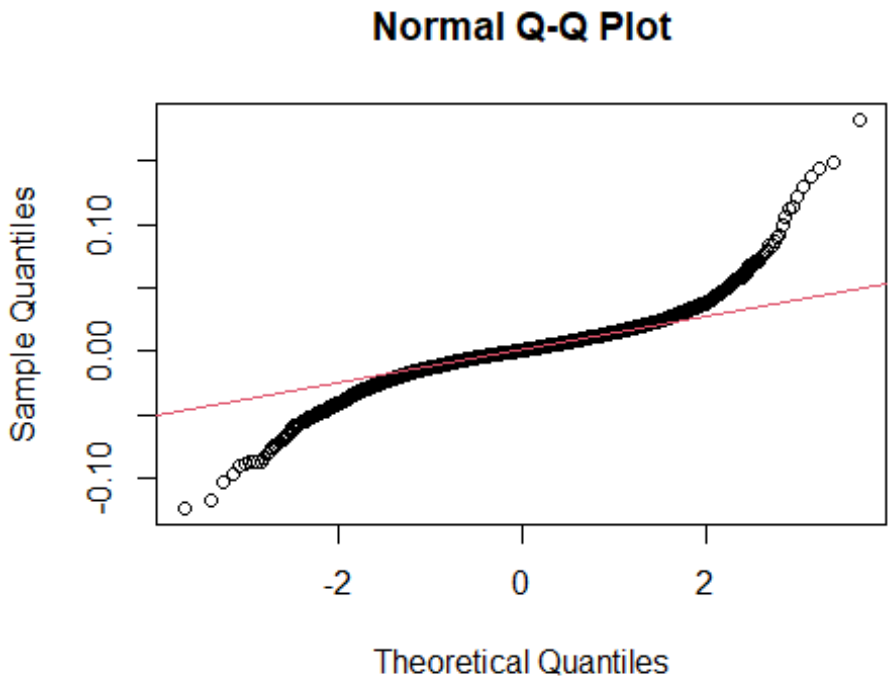





```
#2 GOOG's  log return mean, variance, standard deviation, skewness, excess
kurtosis, minimum and maximum
table.Stats(logreturn)

##                  GOOG.Adjusted
## Observations         4121.0000
## NAs                     1.0000
## Minimum                -0.1234
## Quartile 1             -0.0074
## Median                  0.0007
## Arithmetic Mean         0.0009
## Geometric Mean          0.0007
```

```
## Quartile 3              0.0100
## Maximum                 0.1823
## SE Mean                 0.0003
## LCL Mean (0.95)         0.0003
## UCL Mean (0.95)         0.0014
## Variance                0.0004
## Stdev                   0.0191
## Skewness                0.4528
## Kurtosis                9.1391
```

```
#3
newlogreturn <- logreturn[2:nrow(logreturn),]
par(mfrow=c(1,1))
qqnorm(newlogreturn)
qqline(newlogreturn, col = 2)
```




```
jarque.bera.test(newlogreturn)
```

```
##
##  Jarque Bera Test
##
## data:  newlogreturn
## X-squared = 14482, df = 2, p-value < 2.2e-16
```

```
#since the calculated p-value=2.2e-16 which is less than alpha=0.05. Thus,
GOOG's log return data is not normally distributed
```

```r
#4 Test for GOOG's mean=0 given H0 :E(r) = 0.8, H1 : E(r) =/ 0.8
t.test(newlogreturn)
```

```
## 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:  newlogreturn
## t = 2.8957, df = 4120, p-value = 0.003803
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.0002787253 0.0014474425
## sample estimates:
##    mean of x
## 0.0008630839
```

```r
#The calculated t = 2.8957 is greater than critical value = 1.96. Therefore,
we can reject null hypothesis : E(r) = 0.8 at 95% confidence interval

#5
table.Stats(logreturn)
```

```
##                 GOOG.Adjusted
## Observations        4121.0000
## NAs                    1.0000
## Minimum               -0.1234
## Quartile 1            -0.0074
## Median                 0.0007
## Arithmetic Mean        0.0009
## Geometric Mean         0.0007
## Quartile 3             0.0100
## Maximum                0.1823
## SE Mean                0.0003
## LCL Mean (0.95)        0.0003
## UCL Mean (0.95)        0.0014
## Variance               0.0004
## Stdev                  0.0191
## Skewness               0.4528
## Kurtosis               9.1391
```

```r
qnorm(.95)
```

```
## [1] 1.644854
```

```r
qnorm(0.05)
```

```
## [1] -1.644854
```

```
#7 Test Skewness = 0
T1=length(newlogreturn)
GOOG_s3=skewness(newlogreturn)
GOOG_s3
```

```
## [1] 0.4527864
```

```
tst = GOOG_s3/sqrt(6/T1)
tst
```

```
## [1] 11.8664
```

```
pv1 = 2*pnorm(tst)
pv1
```

```
## [1] 2
```

```
# Since the calculated s* = 11.8664 is greater than 1.96 (critical  value) we
can reject null hypothesis = H0:m3 is less than or equal to 0 at 95%
confidence interval
```

```
#9 Test excess kurtosis = 0
GOOG_k4 = kurtosis(newlogreturn)
GOOG_k4
```

```
## [1] 9.139095
```

```
tst3 = GOOG_k4/sqrt(24/T1)
tst3
```

```
## [1] 119.7564
```

```
pv3 = 2*(1-pnorm(tst3))
pv3
```

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

```
# since the calculated p-value = 0 which is less than alpha = 0.05. we can
reject null hypothesis : k = 3 at 95% confidence interval
```