Time Series HW2

31/40

B082040005 高念慈

2023-03-03

7

1.

Consider the daily stock returns of American Express (AXP),
Caterpillar (CAT), and Starbucks (SBUX) from January 01, 2020 to
February 10, 2023. The price data can be obtained by using R package quantmod.

```
getSymbols("AXP",from="2020-01-01",to='2023-02-11')
```

```
## [1] "AXP"
```

```
getSymbols("CAT",from="2020-01-01",to='2023-02-11')
```

```
## [1] "CAT"
```

```
getSymbols("SBUX",from="2020-01-01",to='2023-02-11')
```

```
## [1] "SBUX"
```

d.Test the null hypothesis that the mean of the log returns of each stock is zero.
That is, perform three separate tests.
Use 5% significance level to draw your conclusion. ? -2
(You can use R function t.test() to answer the problem.)

取出 adjust price

```
AXP_adjust = AXP[,6]
CAT_adjust = CAT[,6]
SBUX_adjust = SBUX[,6]
```

- $r_t = ln(R_t + 1)$
- $r_t = ln(P_t) - ln(P_{t-1})$
- $R_t = e^{rt} - 1$
- simple returns : R_t
- log returns : r_t
- $R_t = \frac{Pt - Pt-1}{Pt-1}$

Log Return in percentages X -1

```
# Log Return : r_t = ln(P_t) - ln(P_{t-1})

AXP_log_returns = diff(log(AXP_adjust))*100
head(AXP_log_returns)
```

```
##            AXP.Adjusted
## 2020-01-02           NA
## 2020-01-03   -0.9982161
## 2020-01-06   -0.4343347
## 2020-01-07   -0.5253129
## 2020-01-08    1.7112188
## 2020-01-09    1.7920422
```

```
CAT_log_returns = diff(log(CAT_adjust))*100
head(CAT_log_returns)
```

```
##            CAT.Adjusted
## 2020-01-02           NA
## 2020-01-03  -1.39816009
## 2020-01-06  -0.06738172
## 2020-01-07  -1.33009612
## 2020-01-08   0.88417204
## 2020-01-09  -0.25086705
```

```
SBUX_log_returns = diff(log(SBUX_adjust))*100
head(SBUX_log_returns)
```

```
##            SBUX.Adjusted
## 2020-01-02            NA
## 2020-01-03    -0.5836709
## 2020-01-06    -0.7911632
## 2020-01-07    -0.3068224
## 2020-01-08     1.1542445
## 2020-01-09     1.8394303
```

t.test()

```
# AXP_log_returns
t.test(as.vector(AXP_log_returns))
```

```
##
##  One Sample t-test
##
## data:  as.vector(AXP_log_returns)
## t = 0.50358, df = 782, p-value = 0.6147
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1465192  0.2476335
## sample estimates:
##  mean of x
## 0.05055714
```

```
# p-value = 0.6147 not reject H0 · the mean of the log returns is zero
# 95 percent confidence interval : -0.1465192  0.2476335

# CAT_log_returns
t.test(as.vector(CAT_log_returns))
```

```
##
##  One Sample t-test
##
## data:  as.vector(CAT_log_returns)
## t = 0.92575, df = 782, p-value = 0.3549
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.08246386  0.22966229
## sample estimates:
##  mean of x
## 0.07359921
```

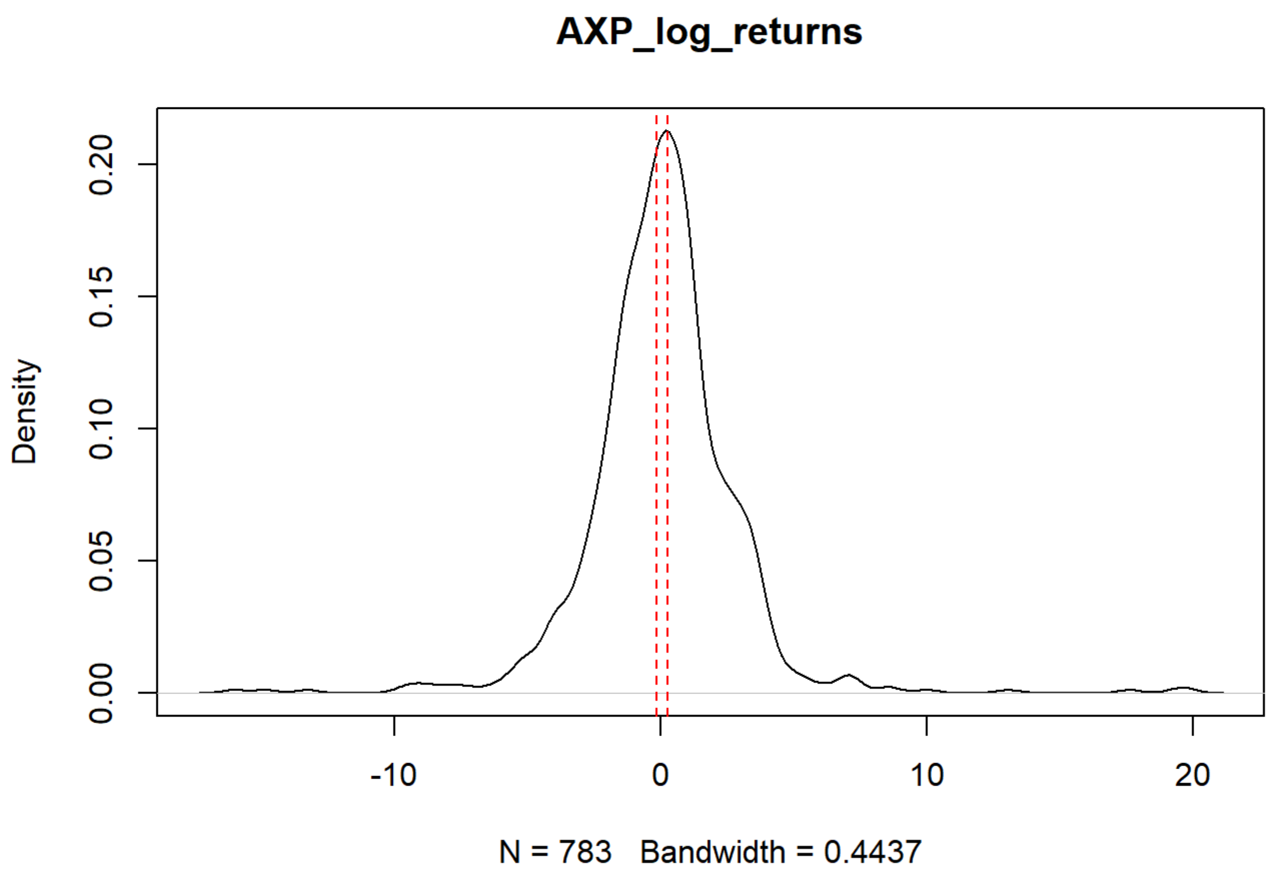
```
# p-value = 0.3549 not reject H0 · the mean of the log returns is zero
# 95 percent confidence interval : -0.08246387  0.22966233

# SBUX_log_returns
t.test(as.vector(SBUX_log_returns))
```

```
##
##  One Sample t-test
##
## data:  as.vector(SBUX_log_returns)
## t = 0.39353, df = 782, p-value = 0.694
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.1255052  0.1884439
## sample estimates:
##  mean of x
## 0.03146937
```

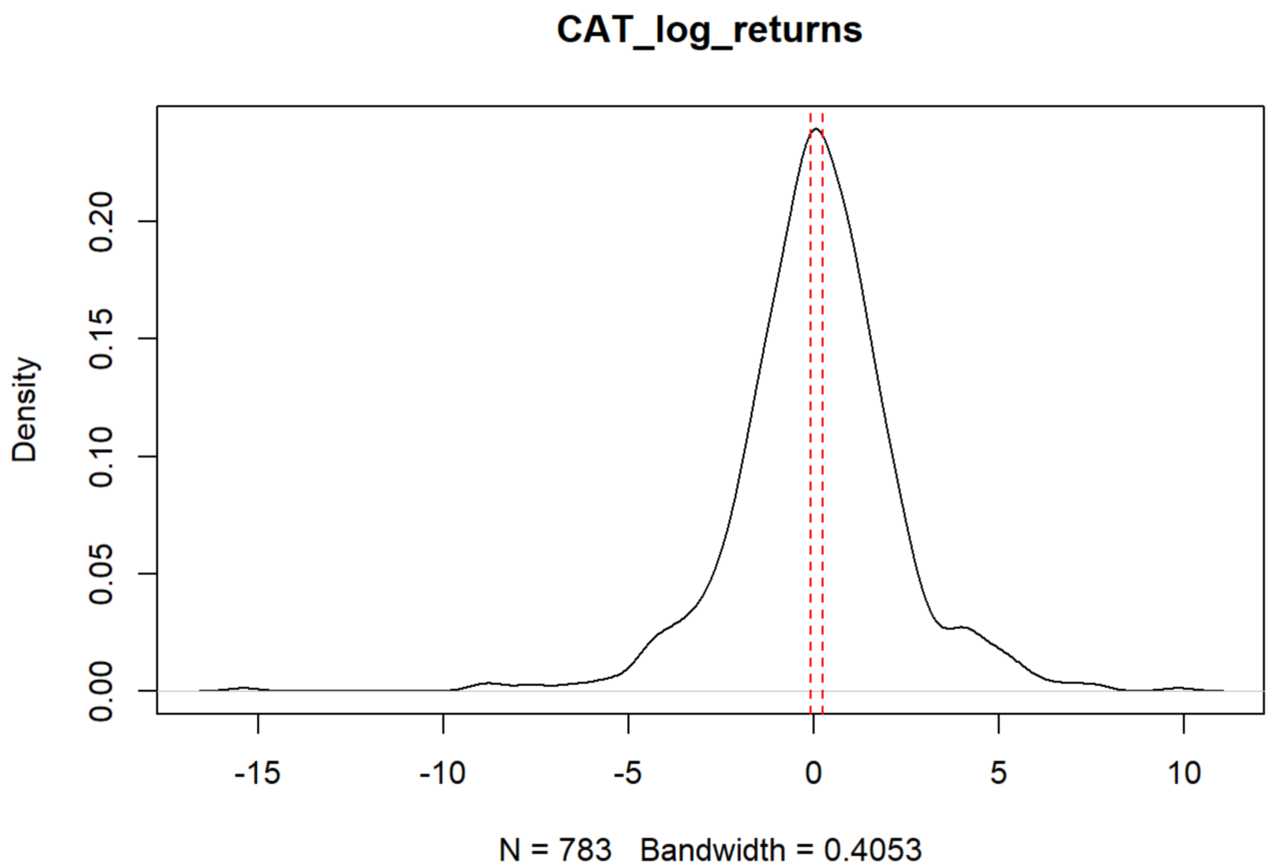
```
# p-value = 0.694 not reject H0 · the mean of the log returns is zero
# 95 percent confidence interval : -0.1255052  0.1884439
```

draw

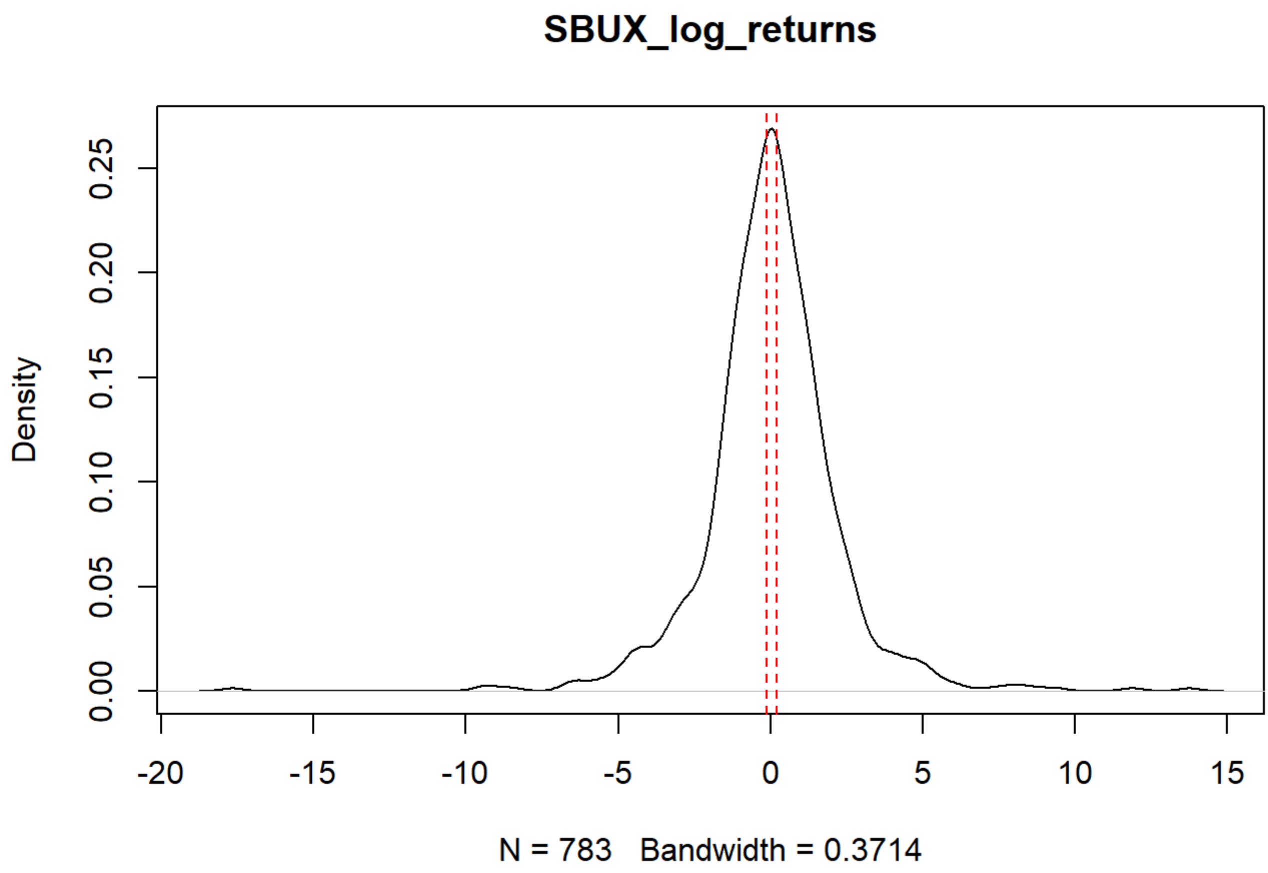
```
# AXP_log_returns
plot(density(as.vector(AXP_log_returns),na.rm = T),main = "AXP_log_returns")
abline(v = c(-0.1465192,0.2476335),col=c("red","red"),lty=2,lwd=1)
```




```
# CAT_log_returns
plot(density(as.vector(CAT_log_returns),na.rm = T),main = 'CAT_log_returns')
abline(v = c(-0.08246387,0.22966233),col=c("red","red"),lty=2,lwd=1)
```




```
# SBUX_log_returns
plot(density(as.vector(SBUX_log_returns),na.rm = T),main = "SBUX_log_returns")
abline(v = c(-0.1255052,0.1884439),col=c("red","red"),lty=2,lwd=1)
```



7

2.

Consider the monthly stock returns for General Motors (GM), CRSP value-weighted index (VW), CRSP equal weighted index (EW), and S&P composite index from January 1975 to December 2008.
The returns of the indexes include dividend distributions.
Data file is m-gm3dx7508.txt (date, gm, vw, ew, sp).

月度簡單回報

```
da = read.table("https://faculty.chicagobooth.edu/-/media/faculty/ruey-s-tsay/teaching/fts3/m
-gm3dx7508.txt",header = T)
head(da)
```

```
##       date       gm       vw       ew       sp
## 1 19750131 0.252033 0.141600 0.299260 0.122812
## 2 19750228 0.028571 0.058411 0.053918 0.059886
## 3 19750331 0.054487 0.030191 0.081497 0.021694
## 4 19750430 0.045593 0.046497 0.031093 0.047265
## 5 19750530 0.037209 0.055140 0.072876 0.044101
## 6 19750630 0.107955 0.051473 0.071792 0.044323
```

d.Test the null hypothesis that the mean of the log returns of each stock is zero.
That is, perform three separate tests.
Use 5% significance level to draw your conclusion. ? -2

- $r_t = ln(R_t + 1)$

Log Return in percentages

```
# Log Return : r_t = ln(R_t + 1)*100

a = da[,c("gm","vw","ew","sp")]
log_returns = log(a + 1)*100
head(log_returns)
```

```
##          gm        vw        ew        sp
## 1 22.476863 13.243079 26.179487 11.583625
## 2  2.817046  5.676873  5.251465  5.816136
## 3  5.305439  2.974422  7.834619  2.146203
## 4  4.458419  4.544840  3.061940  4.618200
## 5  3.653345  5.367346  7.034289  4.315623
## 6 10.251597  5.019204  6.933201  4.336883
```

t.test()

```
# "gm"
t.test(log_returns[,"gm"])
```

```
##
##  One Sample t-test
##
## data:  log_returns[, "gm"]
## t = 0.23206, df = 407, p-value = 0.8166
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.8231636  1.0435276
## sample estimates:
## mean of x
##  0.110182
```

```
# p-value = 0.8166 not reject H0 · the mean of the log returns is zero
# 95 percent confidence interval : -0.8231636  1.0435276

# "vw"
t.test(log_returns[,"vw"])
```

```
##
##  One Sample t-test
##
## data:  log_returns[, "vw"]
## t = 4.0064, df = 407, p-value = 7.332e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.4607213 1.3484127
## sample estimates:
## mean of x
##  0.904567
```

```
# p-value = 7.332e-05 reject H0 · the mean of the log returns is not zero
# 95 percent confidence interval : 0.4607213 1.3484127

# "ew"
t.test(log_returns[,"ew"])
```

```
##
##  One Sample t-test
##
## data:  log_returns[, "ew"]
## t = 4.1899, df = 407, p-value = 3.425e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.6194647 1.7145293
## sample estimates:
## mean of x
##  1.166997
```

```
# p-value = 3.425e-05 reject H0 · the mean of the log returns is not zero
# 95 percent confidence interval : 0.6194647 1.7145293

# "sp"
t.test(log_returns[,"sp"])
```

```
##
##  One Sample t-test
##
## data:  log_returns[, "sp"]
## t = 2.8996, df = 407, p-value = 0.003939
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.2035053 1.0603692
## sample estimates:
## mean of x
## 0.6319372
```

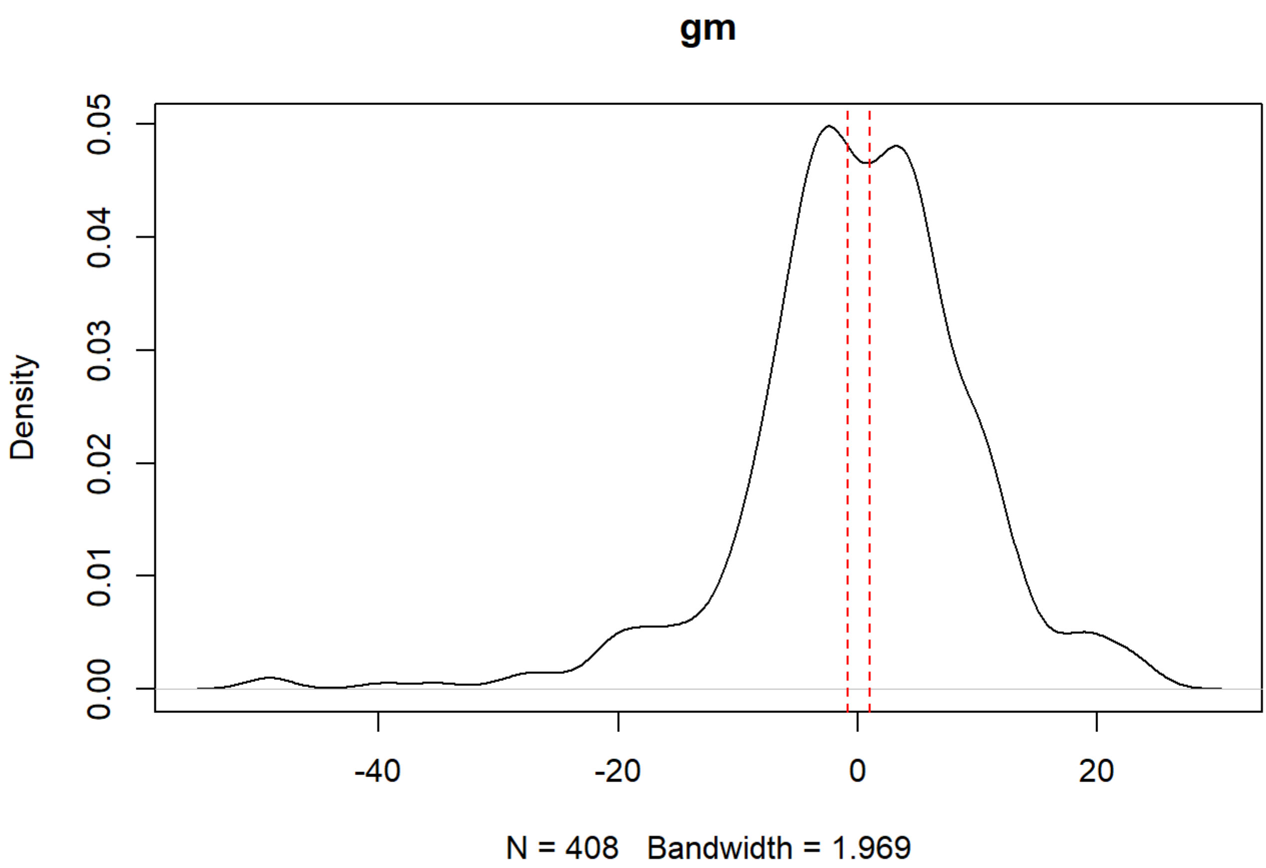
```
# p-value = 0.003939 reject H0 · the mean of the log returns is not zero
# 95 percent confidence interval : 0.2035053 1.0603692
```

```
apply(log_returns,2,t.test)   # t.test
```

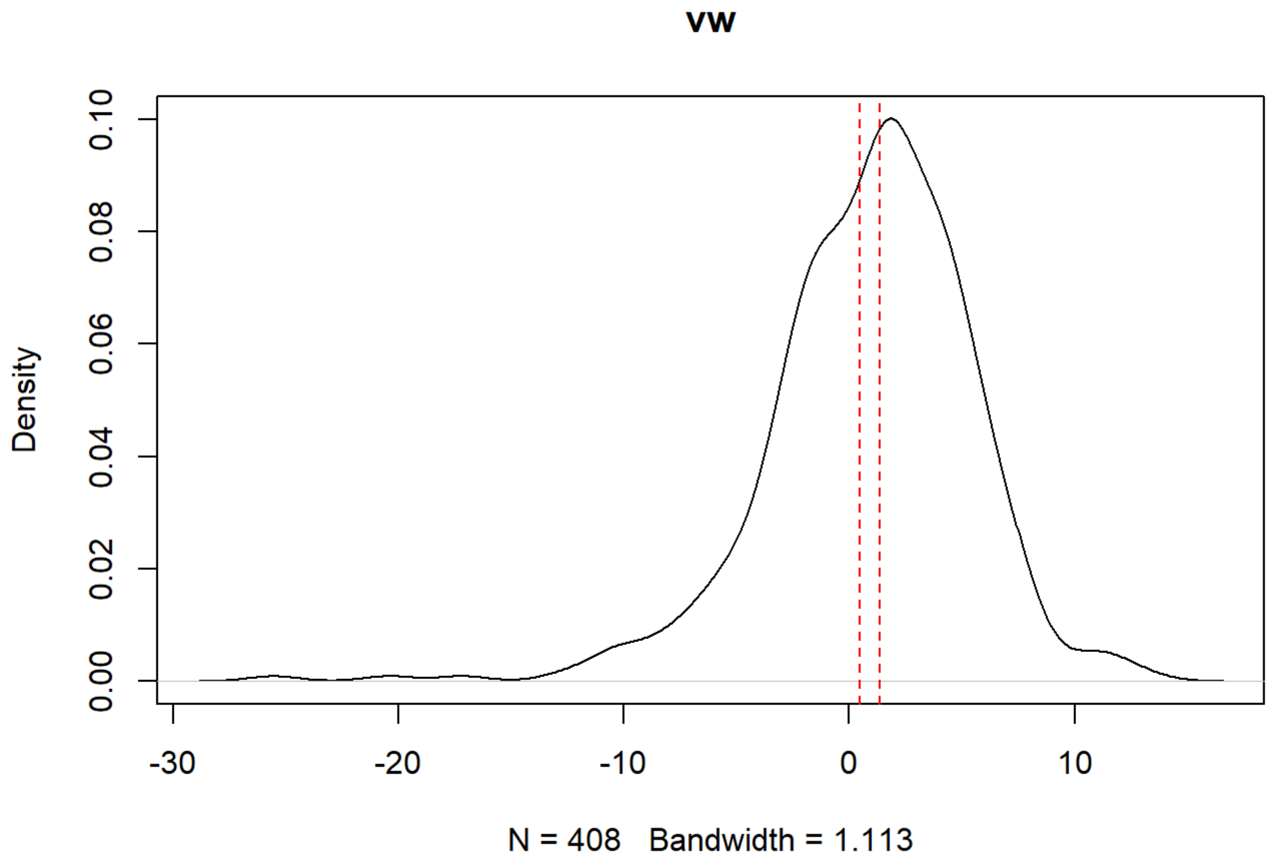
```
## $gm
##
##  One Sample t-test
##
## data:  newX[, i]
## t = 0.23206, df = 407, p-value = 0.8166
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.8231636  1.0435276
## sample estimates:
## mean of x
##  0.110182
##
##
## $vw
##
##  One Sample t-test
##
## data:  newX[, i]
## t = 4.0064, df = 407, p-value = 7.332e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.4607213 1.3484127
## sample estimates:
## mean of x
##  0.904567
##
##
## $ew
##
##  One Sample t-test
##
## data:  newX[, i]
## t = 4.1899, df = 407, p-value = 3.425e-05
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.6194647 1.7145293
## sample estimates:
## mean of x
##  1.166997
##
##
## $sp
##
##  One Sample t-test
##
## data:  newX[, i]
## t = 2.8996, df = 407, p-value = 0.003939
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.2035053 1.0603692
## sample estimates:
## mean of x
## 0.6319372
```

draw

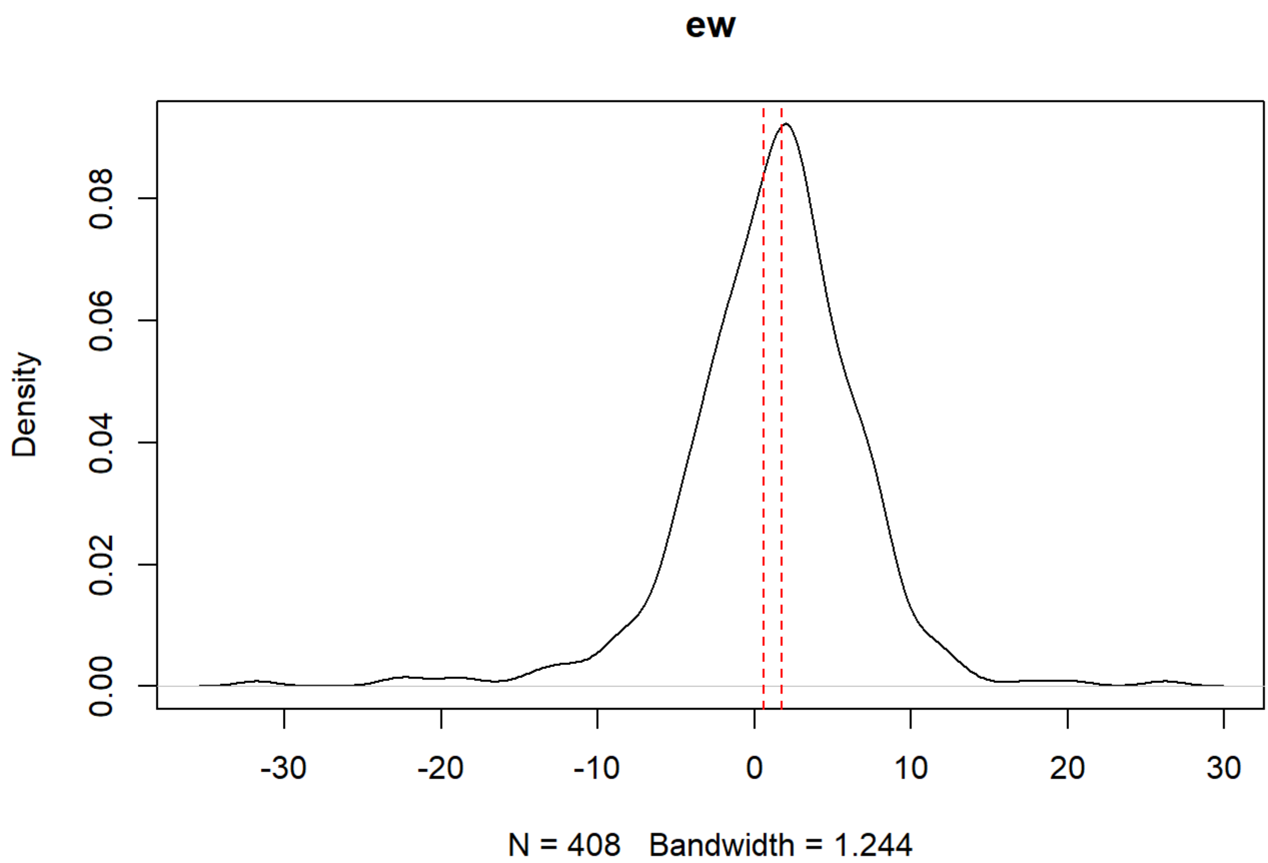
```r
# "gm"
plot(density(log_returns[,"gm"],na.rm = T),main = "gm")
abline(v = c(-0.8231636, 1.0435276),col=c("red","red"),lty=2,lwd=1)
```



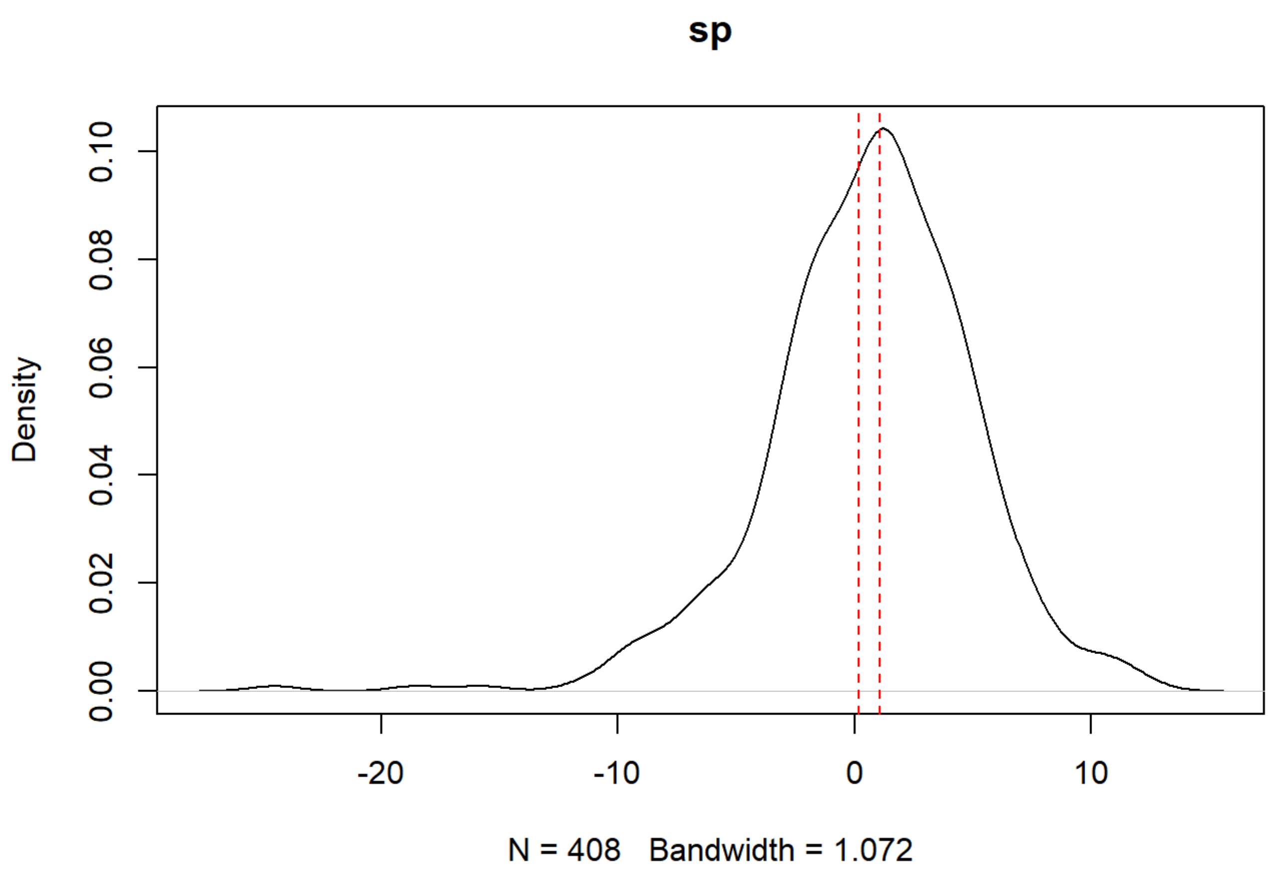

```r
# "vw"
plot(density(log_returns[,"vw"],na.rm = T),main = "vw")
abline(v = c(0.4607213, 1.3484127),col=c("red","red"),lty=2,lwd=1)
```




```
# "ew"
plot(density(log_returns[,"ew"],na.rm = T),main = "ew")
abline(v = c(0.6194647, 1.7145293),col=c("red","red"),lty=2,lwd=1)
```




```
# "sp"
plot(density(log_returns[,"sp"],na.rm = T),main = "sp")
abline(v = c(0.2035053, 1.0603692),col=c("red","red"),lty=2,lwd=1)
```



3. 9

Consider the daily log returns of American Express stock
from January 01, 2020 to February 10, 2023 as in Exercise 1.1.

- getSymbols(“AXP”,from=“2020-01-01”,to=‘2023-02-11’)
- AXP_adjust = AXP[,6]
- Daily Log Return in percentages
- $r_t = ln(P_t) - ln(P_{t-1})$
- AXP_log_returns = diff(log(AXP_adjust))*100 -1.
- head(AXP_log_returns)

Daily Log Return in percentages

```
head(AXP_log_returns)
```

```
##            AXP.Adjusted
## 2020-01-02           NA
## 2020-01-03   -0.9982161
## 2020-01-06   -0.4343347
## 2020-01-07   -0.5253129
## 2020-01-08    1.7112188
## 2020-01-09    1.7920422
```

Use the 5% significance level to perform the following tests:
a. Test the null hypothesis that the skewness of the returns is zero.

skewness is zero test

```
sk = skewness(AXP_log_returns, na.rm=TRUE) # 0.6272351

# Compute test statistic
t = nrow(AXP_log_returns)-1              # sum(is.na(AXP_log_returns)) : 1
tl = sk/sqrt(6/t)                        # 7.165321

# Compute p-value
pv = 2*(1 - pnorm(tl,lower.tail = TRUE))
pv                                       # 7.760459e-13
```

```
## [1] 7.758238e-13
## attr(,"method")
## [1] "moment"
```

```
# p-value = 7.760459e-13 · reject H0 · skewness of the returns is not zero
```

skewness.norm.test

```
skewness.norm.test(AXP_log_returns[-1,])
```

```
##
##  Skewness test for normality
##
## data:  AXP_log_returns[-1, ]
## T = 0.62844, p-value < 2.2e-16
```

b. Test the null hypothesis that the excess kurtosis of the returns is zero.

kurtosis is zero test

```
ku = kurtosis(AXP_log_returns,na.rm=TRUE)  # 11.04311

# Compute test statistic
k = nrow(AXP_log_returns)-1              # sum(is.na(AXP_log_returns)) : 1
kl = (ku-3)/sqrt(24/k)                   # 45.94085

# Compute p-value
pv = 2*(1 - pnorm(kl,lower.tail = TRUE))
pv                                       # 0
```

```
## [1] 0
## attr(,"method")
## [1] "excess"
```

```
# p-value = 0 · reject H0 · kurtosis of the returns is not zero
```

kurtosis.norm.test

```
kurtosis.norm.test(AXP_log_returns[-1,])
```

```
##
##  Kurtosis test for normality
##
## data:  AXP_log_returns[-1, ]
## T = 14.079, p-value < 2.2e-16
```

4.

8

Daily foreign exchange rates (spot rates) can be obtained from the
Federal Reserve Bank in Chicago. The data are the noon buying rates
in New York City certified by the Federal Reserve Bank of New York.

Consider the exchange rates between the U.S. dollar and
the Canadian dollar, euro, U.K. pound, and the Japanese yen
from January 4, 2000, to March 27, 2009. The data are also on the Web.
(d-caus.txt, d-usuk.txt, d-jpus.txt and d-useu.txt)

```
# the dollar-euro exchange rates
useu = read.table("https://faculty.chicagobooth.edu/-/media/faculty/ruey-s-tsay/teaching/fts
3/d-useu.txt",header = T)
head(useu)
```

```
##   Date Mon Day  Value
## 1 2000   1   4 1.0309
## 2 2000   1   5 1.0335
## 3 2000   1   6 1.0324
## 4 2000   1   7 1.0294
## 5 2000   1  10 1.0252
## 6 2000   1  11 1.0322
```

```
# the dollar-Japanese yen exchange rates
jpus = read.table("https://faculty.chicagobooth.edu/-/media/faculty/ruey-s-tsay/teaching/fts
3/d-jpus.txt",header = T)
head(jpus)
```

```
##   year mon day  value
## 1 2000   1   4 103.09
## 2 2000   1   5 103.77
## 3 2000   1   6 105.19
## 4 2000   1   7 105.17
## 5 2000   1  10 105.28
## 6 2000   1  11 106.09
```

- **Daily Log Return in percentages**

- $r_t = ln(P_t) - ln(P_{t-1})$

```
# Log Return : r_t = ln(P_t) - ln(P_{t-1})

useu_log_return = 100*(diff(log(useu[,"Value"])))
head(useu_log_return)
```

```
## [1]  0.2518893 -0.1064911 -0.2910081 -0.4088393  0.6804731 -0.3980008
```

```
jpus_log_return = 100*(diff(log(jpus[,"value"])))
head(jpus_log_return)
```

```
## [1]  0.65745185  1.35913271 -0.01901502  0.10453790  0.76643229 -0.31154144
```

c.

Discuss the **empirical characteristics**
of the log returns of the dollar-Japanese yen exchange rates.

```
basicStats(jpus_log_return)
```

```
##              jpus_log_return
## nobs             2322.000000
## NAs                 0.000000
## Minimum            -5.215648
## Maximum             2.708365
## 1. Quartile        -0.356996
## 3. Quartile         0.388778
## Mean               -0.002185
## Median              0.008289
## Sum                -5.073696
## SE Mean             0.013764
## LCL Mean           -0.029175
## UCL Mean            0.024805
## Variance            0.439876
## Stdev               0.663232
## Skewness           -0.670617
## Kurtosis            4.580879
```

```
# Minimum   -5.215648
# Maximum   2.708365

# 95% 的信心，日對數報酬率的平均為 0
# Mean  -0.002185
# LCL Mean  -0.029175
# UCL Mean  0.024805

# Variance  0.439876
# Stdev 0.663232

# Skewness  -0.670617           # 左偏
# Kurtosis  4.580879          # 厚尾
```

normal test

```
normtest::jb.norm.test(jpus_log_return)
```

```
##
##  Jarque-Bera test for normality
##
## data:  jpus_log_return
## JB = 2210.3, p-value < 2.2e-16
```

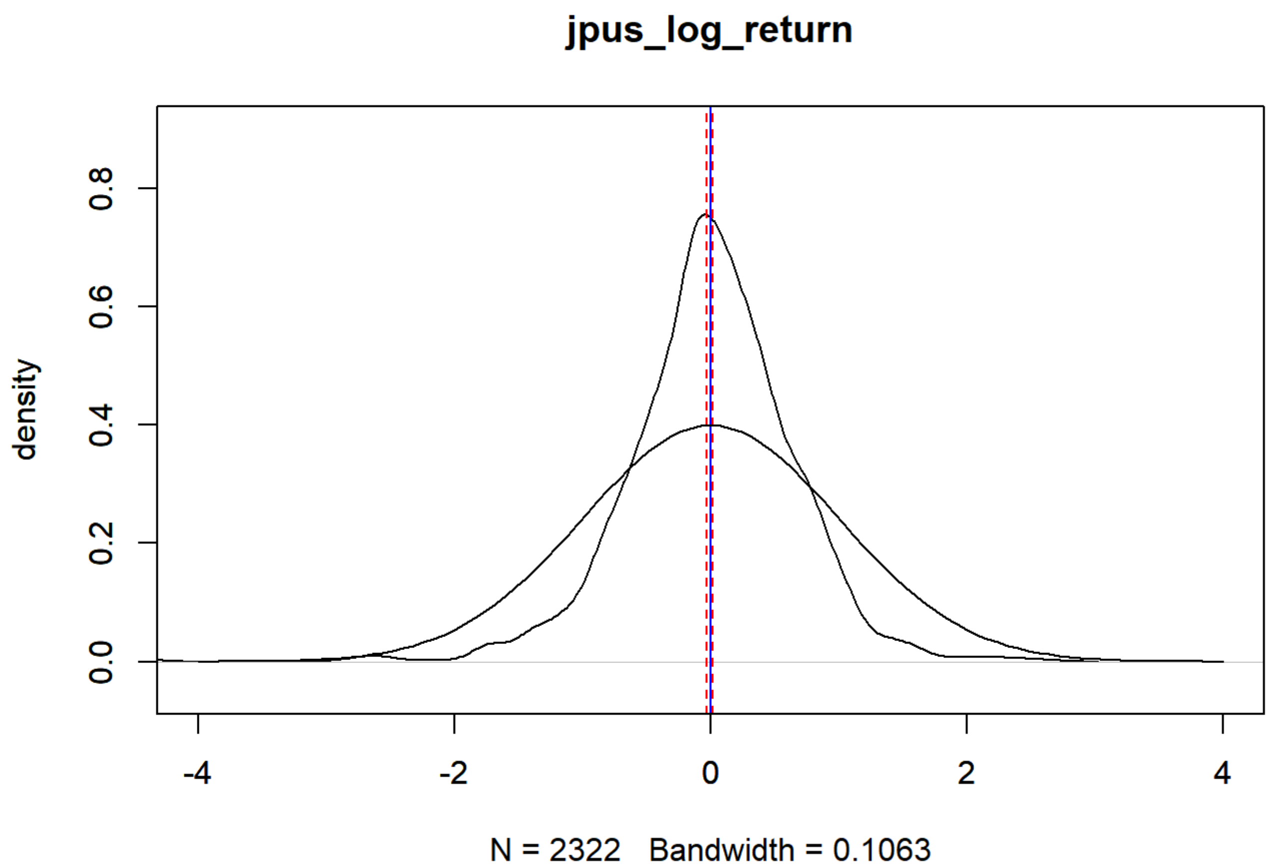
```
normalTest(jpus_log_return,method = "jb",na.rm = T)
```

```
##
## Title:
##  Jarque - Bera Normalality Test
##
## Test Results:
##   STATISTIC:
##     X-squared: 2210.311
##   P VALUE:
##     Asymptotic p Value: < 2.2e-16
##
## Description:
##  Fri Mar 03 03:06:59 2023 by user: user
```

```
# p-value < 2.2e-16 · reject H0 · the dist. is not normal
```

Plot normal

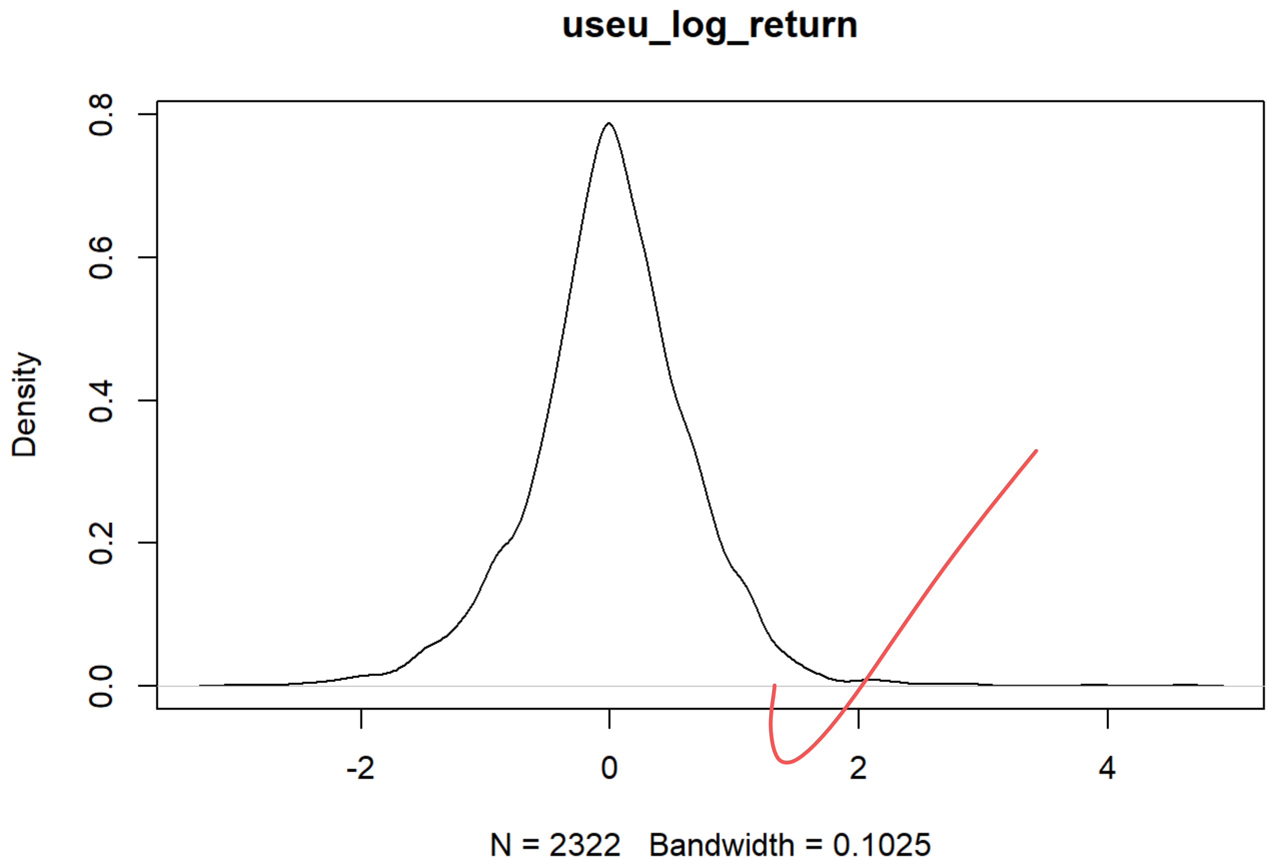
```
plot(density(jpus_log_return),main = "jpus_log_return",xlim=c(-4,4),ylim=c(-0.05,0.9),ylab =
"density")
abline(v = c(-0.030883, 0.019389),col=c("red","red"),lty=2,lwd=1)
abline(v = c(0),col="blue",lty=1,lwd=1)

par(new=TRUE)

x = rnorm(1000)
curve(dnorm(x),from = -4, to = 4,ylim=c(-0.05,0.9),ylab = "density",xlab = "")
```



d.

Use **kernel density estimation** to obtain **a density plot** of the daily log returns of the dollar-euro exchange rates

```
plot(density(useu_log_return),main = "useu_log_return")
```


useu_log_return
Density
0.0
0.2
0.4
0.6
0.8
-2
0
2
4
N = 2322   Bandwidth = 0.1025