#### Phần code:

### Câu 1:

library(mvtnorm)

set.seed(100)

# Define mean vector

mean\_vector <- c(0, 0, 1)

# Define covariance matrix

cov\_matrix <- matrix(c(1, 0.8, 0.6,

0.8, 1, 0.8,

0.6, 0.8, 1),

nrow = 3, byrow = TRUE)

# Generate random sample

random\_sample = rmvnorm(100, mean = mean\_vector, sigma = cov\_matrix)

colnames(random\_sample) = c("A", "B", "C")

head(random\_sample)

D = rnorm(100, mean = .5, sd = sqrt(0.5))

X = cbind(random\_sample, D)

X

##b Giá trị trung bình mẫu

colMeans(X)

# Ma trận hiệp phương sai

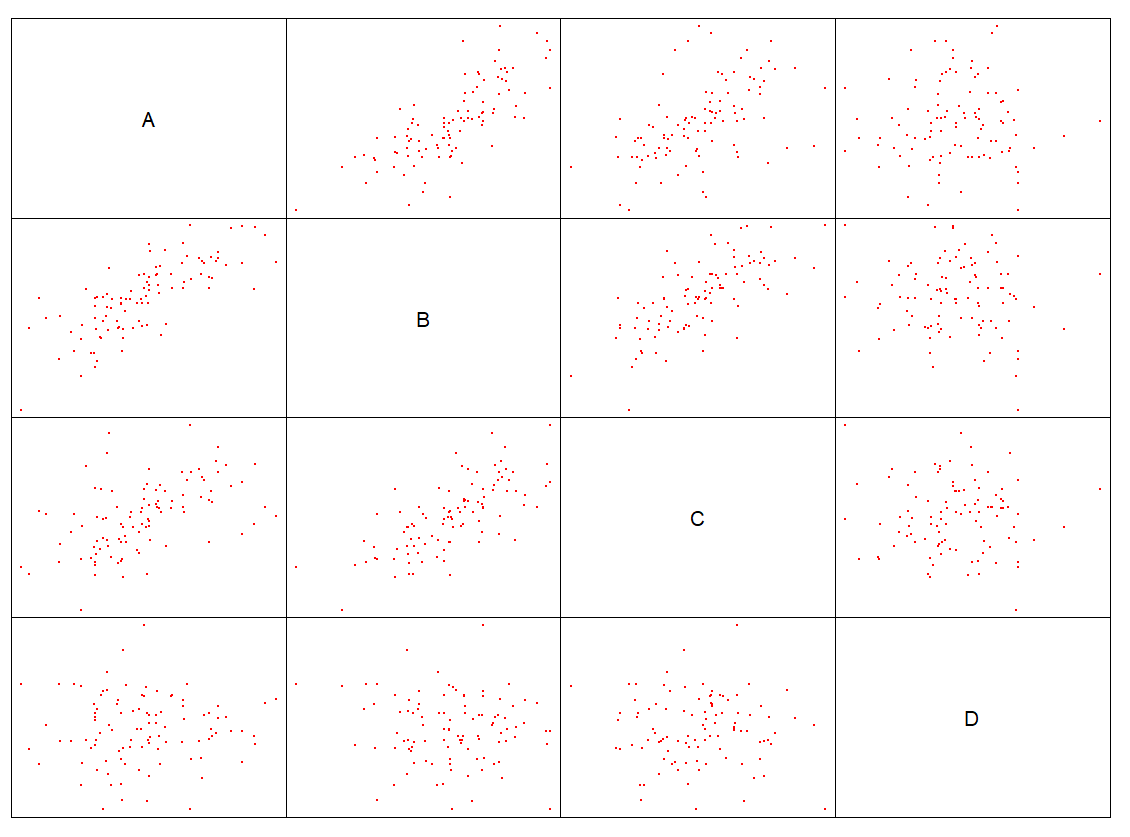
cov(X)

# Ma trận tương quan mẫu

cor(X)

## c. Biểu đồ tán xạ

pairs(X, lwd = 1, pch = 16, cex = .5, col = "red", gap = 0, xaxt = "n", yaxt = "n")



#Biểu đồ nhiệt của ma trận tương quan mẫu

library(reshape2)

library(ICSNP)

cor\_df = round(cor(X), 2)

cor\_df

melted\_cor = melt(cor\_df) # chuyển từ ma trận tương quan sang tương quan 2 biến theo cột

head(melted\_cor)

melted\_cor

dim(melted\_cor)

library(ggplot2)

# create corrrlation heatmap

ggplot(data = melted\_cor, aes(x=Var1, y=Var2, fill=value)) +

geom\_tile() + # khung có màu xanh nhiều mức

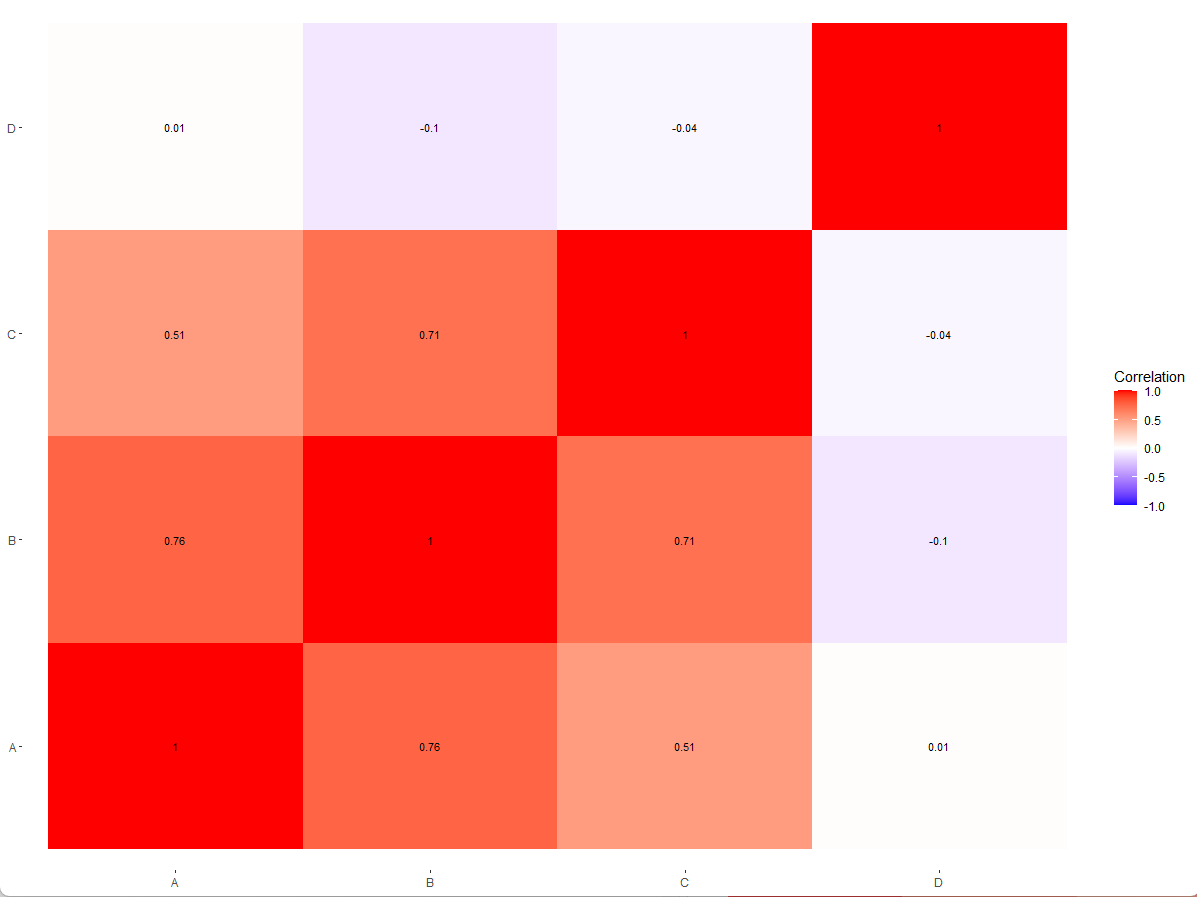
geom\_text(aes(Var2, Var1, label = value), size = 3) + # giá trị tương quan trong từng ô

scale\_fill\_gradient2(low = "blue", high = "red", limit = c(-1,1), name="Correlation") +

theme(axis.title.x = element\_blank(),

axis.title.y = element\_blank(),

panel.background = element\_blank())



## d. Kiểm tra tính chuẩn 4−chiều.

# Để kiểm tra 1 vecto nn X có tuân theo pp chuẩn k-chiều ko thì làm theo 2 bước:

## B1: Kiểm tra tính chuẩn 1-chiều của từng biến trong X

## Bt: H0: từng biến tuân theo pp chuẩn với H1: k tuân theo pp chuẩn

sapply(colnames(X), function(x) {

shapiro.test(X[,x])$p.value

})

# A B C D

# 0.3287910 0.5199584 0.9445196 0.5090947

# vì tất cả p\_value đều > 0.05 nên ta chấp nhận H0. Vậy tất cả các biến đều tuân theo pp chuẩn

## B2: Ktra tính chuẩn nhiều chiều shapiro.test(qnorm(pchisq(mah,4))

## B2 chính là ktra tính chuẩn của phân vị pp chuẩn tắc tương ứng với xs của pp khi bình phương xét trên dữ liệu

## là mah với 4 bậc tự do

# Mahalanobis distance: D2i = (Yi - Yh)' S^(-1)(Yi - Yh)

mah = mahalanobis(X, colMeans(X), var(X))

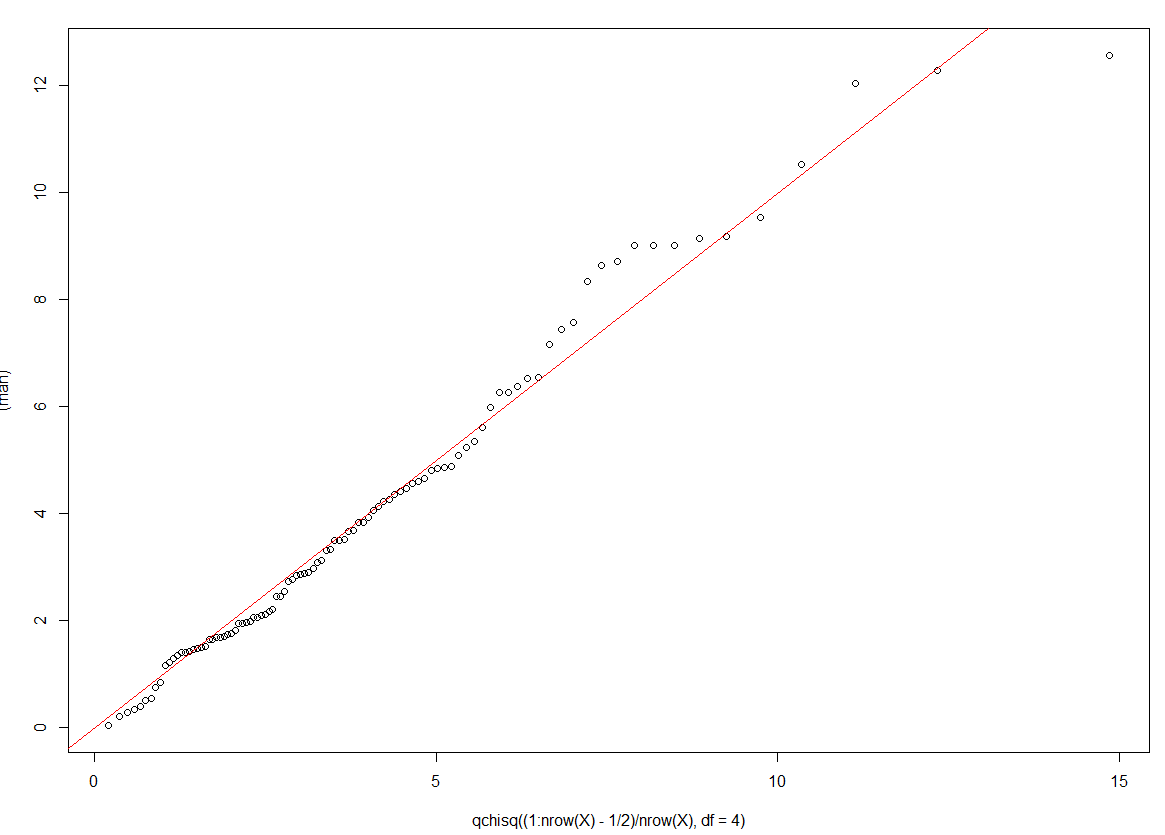
dim(X)

length(mah)

mah

qqplot(qchisq((1:nrow(X) - 1/2) / nrow(X), df = 4), (mah))

abline(a = 0, b = 1, col = "red")



shapiro.test(qnorm(pchisq(mah, 4))) # Kiểm tra tính chuẩn nhiều chiều

# p-value = 0.335 Tính chuẩn nhiều chiều đc chấp nhận

### Câu 2:

install.packages('ISLR')

library(ISLR)

data = Smarket

head(data)

## a. Tính giá trị trung bình mẫu, ma trận tương quan mẫu và ma trận hiệp phương sai mẫu của X.

X = data[, c("Lag1", "Lag2", "Lag3", "Lag4", "Lag5", "Volume", "Today")]

# Tính giá trị trung bình mẫu

mean\_values <- colMeans(X)

# Tính ma trận hiệp phương sai mẫu

covariance\_matrix <- cov(X)

# Tính ma trận tương quan mẫu

correlation\_matrix <- cor(X)

# Hiển thị kết quả

mean\_values

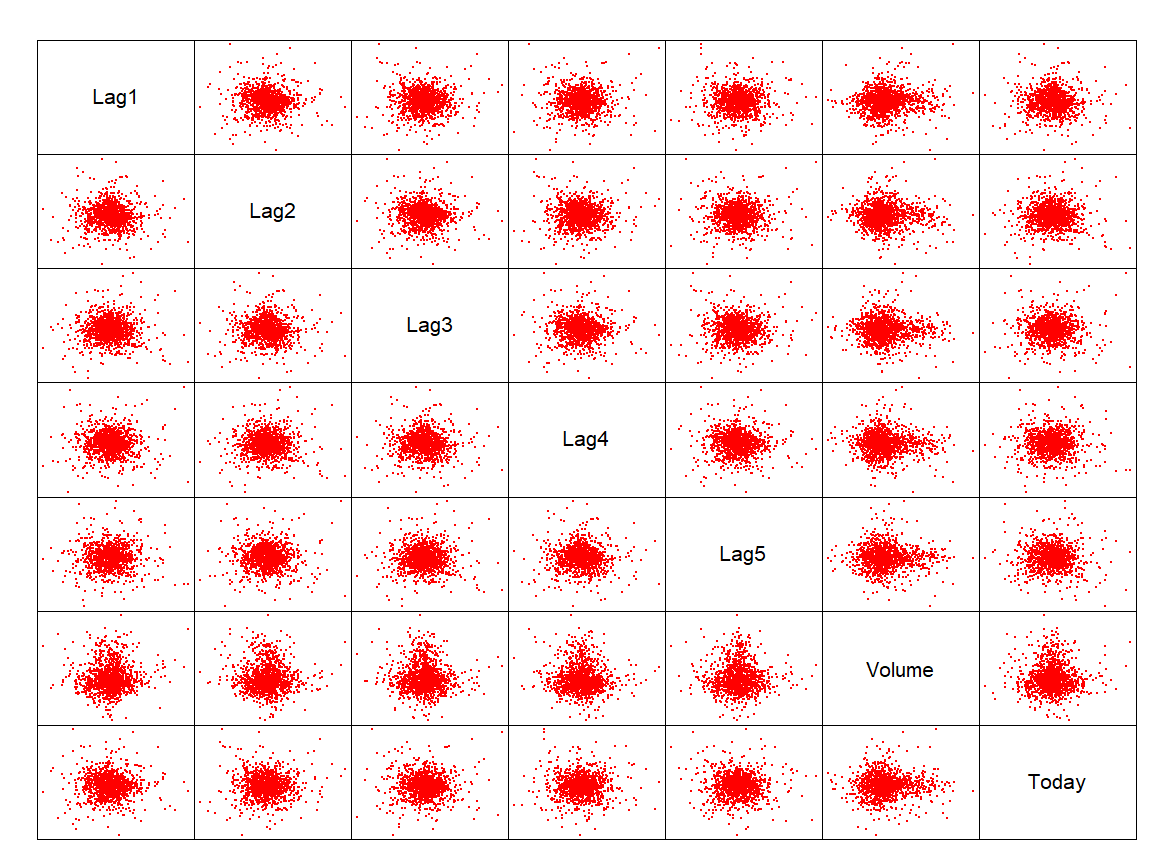
correlation\_matrix

covariance\_matrix

## b. Vẽ biểu đồ phân tán của các biến và biểu đồ nhiệt của ma trận tương quan mẫu của X.

# Biểu đồ tán xạ

pairs(X, lwd = 1, pch = 16, cex = .5, col = "red", gap = 0, xaxt = "n", yaxt = "n")



#Biểu đồ nhiệt của ma trận tương quan mẫu

library(reshape2)

library(ICSNP)

cor\_df = round(cor(X), 2)

cor\_df

melted\_cor = melt(cor\_df) # chuyển từ ma trận tương quan sang tương quan 2 biến theo cột

head(melted\_cor)

melted\_cor

dim(melted\_cor)

library(ggplot2)

# create corrrlation heatmap

ggplot(data = melted\_cor, aes(x=Var1, y=Var2, fill=value)) +

geom\_tile() + # khung có màu xanh nhiều mức

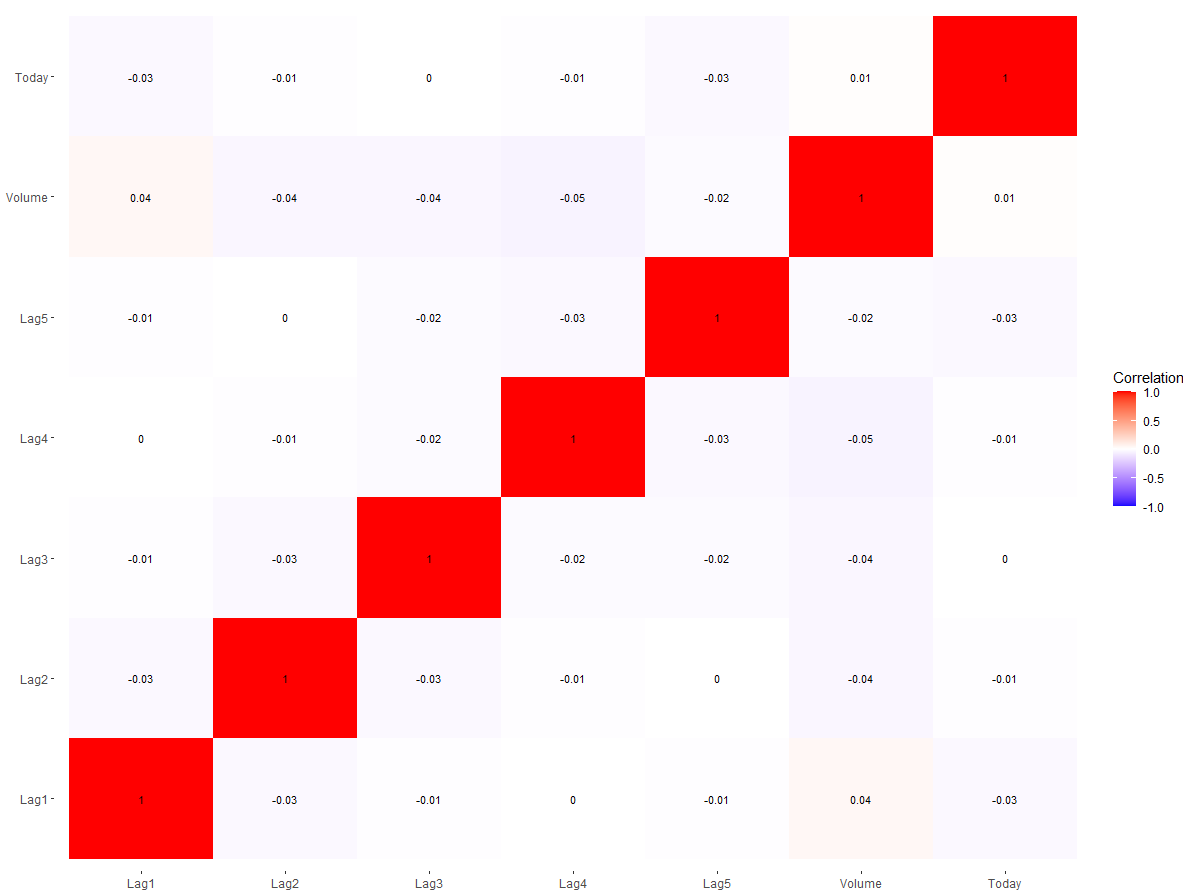
geom\_text(aes(Var2, Var1, label = value), size = 3) + # giá trị tương quan trong từng ô

scale\_fill\_gradient2(low = "blue", high = "red", limit = c(-1,1), name="Correlation") +

theme(axis.title.x = element\_blank(),

axis.title.y = element\_blank(),

panel.background = element\_blank())



## c. Kiểm tra tính chuẩn nhiều chiều của X.

# Để kiểm tra 1 vecto nn X có tuân theo pp chuẩn k-chiều ko thì làm theo 2 bước:

## B1: Kiểm tra tính chuẩn 1-chiều của từng biến trong X

## Bt: H0: từng biến tuân theo pp chuẩn với H1: k tuân theo pp chuẩn

sapply(colnames(X), function(x) {

shapiro.test(X[,x])$p.value

})

# Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

# 8.888786e-15 8.798395e-15 1.034889e-14 1.048548e-14 2.148901e-15 1.792894e-19 9.156928e-15

# vì tất cả p\_value đều < 0.05 nên ta bác bỏ H0. Vậy tất cả các biến đều ko tuân theo pp chuẩn

## B2: Ktra tính chuẩn nhiều chiều shapiro.test(qnorm(pchisq(mah,4))

## B2 chính là ktra tính chuẩn của phân vị pp chuẩn tắc tương ứng với xs của pp khi bình phương xét trên dữ liệu

## là mah với 4 bậc tự do

# Mahalanobis distance: D2i = (Yi - Yh)' S^(-1)(Yi - Yh)

mah = mahalanobis(X, colMeans(X), var(X))

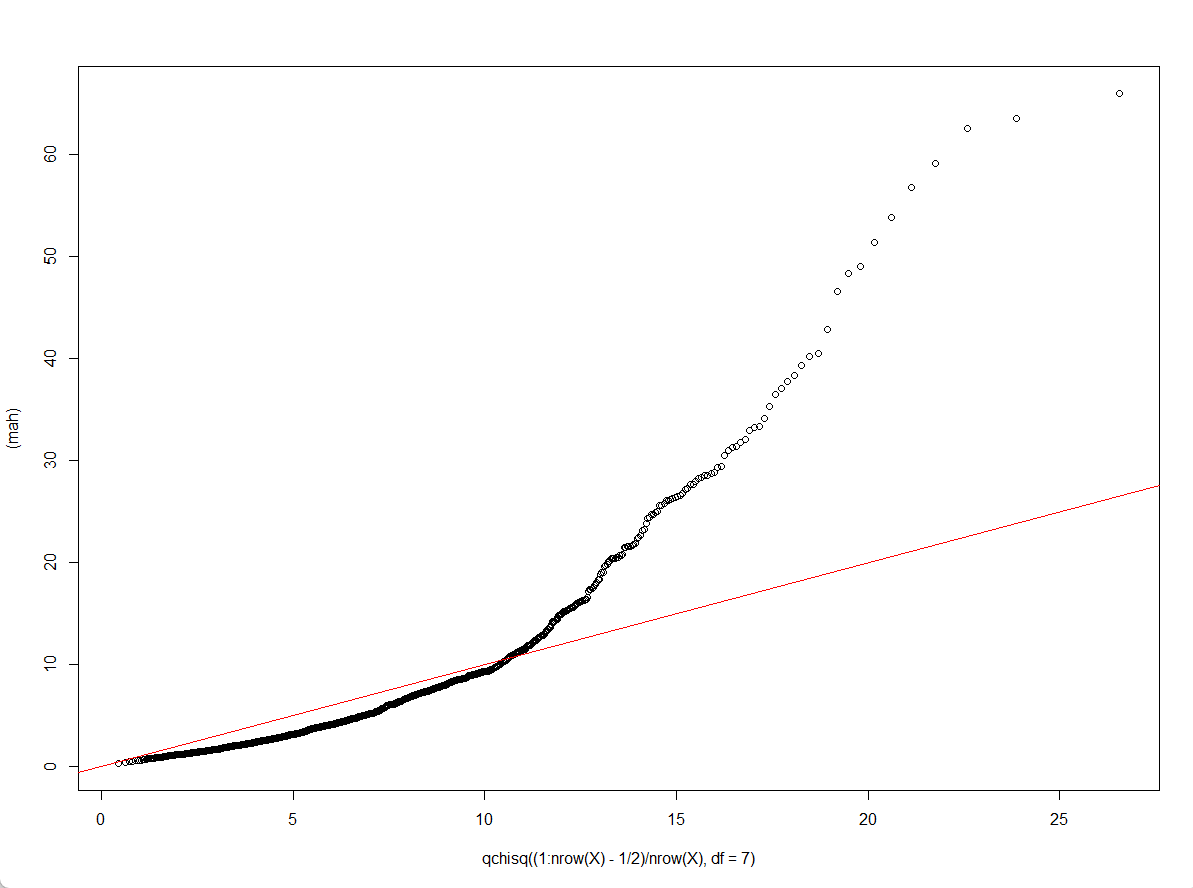
dim(X)

length(mah)

mah

qqplot(qchisq((1:nrow(X) - 1/2) / nrow(X), df = 7), (mah))

abline(a = 0, b = 1, col = "red")



shapiro.test(qnorm(pchisq(mah, 7))) # Kiểm tra tính chuẩn nhiều chiều

# p-value < 2.2e-16 Tính chuẩn ko nhiều chiều đc chấp nhận

## d. Có sự khác biệt giữa lợi nhuận trung bình một ngày trước và ba ngày trước ngày đang xét không?

# Tính lợi nhuận trung bình một ngày trước và ba ngày trước

mean\_Lag1 <- mean(data$Lag1)

mean\_Lag3 <- mean(data$Lag3)

# Hiển thị kết quả

mean\_Lag1

mean\_Lag3

# Kiểm tra sự khác biệt

difference <- mean\_Lag1 - mean\_Lag3

difference

## Vậy với difference = 0.0021184 có nghĩa là lợi nhuận trung bình một ngày trước lớn hơn so với ba ngày trước

## e. Lợi nhuận trung bình tại ngày đang xét và các ngày trước đó có thực sự khác 00 không?

# BT: H0: Lợi nhuận trung bình bằng 0.vs H1: Lợi nhuận trung bình không bằng 0.

# Lấy lợi nhuận tại ngày đang xét

profit\_today <- data$Today

# Kiểm định Wilcoxon để xem có sự khác biệt giữa lợi nhuận trung bình tại ngày đang xét và các ngày trước đó có khác 0 không

wilcox\_test\_result <- wilcox.test(profit\_today, mu = 0)

# Hiển thị kết quả

wilcox\_test\_result

## Với p-value = 0.8924 > 0.05 chưa đủ cơ sở bác bỏ H0.

## Vậy Do đó, không có đủ bằng chứng để kết luận rằng lợi nhuận trung bình tại ngày đang xét khác không (không bằng 0).

## f. Kiểm định xem có sự khác biệt giữa năm 2004 và 2005 về lợi nhuận trung bình tại ngày đang xét và các ngày trước đó không?

## BT: H0: Lợi nhuận trung bình tại ngày đang xét và các ngày trước đó giữa năm 2004 và 2005 không khác biệt => EX2004 = EX2005

## H1: Có sự khác biệt về lợi nhuận trung bình tại ngày đang xét và các ngày trước đó giữa năm 2004 và 2005 => EX2004 != EX2005

# Tạo hai nhóm dữ liệu cho năm 2004 và 2005

profit\_2004 <- data$Today[data$Year == 2004]

profit\_2005 <- data$Today[data$Year == 2005]

# Kiểm định Wilcoxon để xem có sự khác biệt giữa năm 2004 và 2005 về lợi nhuận trung bình tại ngày đang xét và các ngày trước đó không

wilcox\_test\_result\_2004\_2005 <- wilcox.test(profit\_2004, profit\_2005)

# Hiển thị kết quả

wilcox\_test\_result\_2004\_2005

## Với p-value = 0.7079 > 0.05 chưa đủ cơ sở bác bỏ H0.

## Vậy không có sự khác biệt đáng kể giữa lợi nhuận trung bình tại ngày đang xét và các ngày trước đó giữa năm 2004 và 2005.

###### Phần chạy chương trình:

> ### Câu 1:

> library(mvtnorm)

> set.seed(100)

> # Define mean vector

> mean\_vector <- c(0, 0, 1)

> # Define covariance matrix

> cov\_matrix <- matrix(c(1, 0.8, 0.6,

+ 0.8, 1, 0.8,

+ 0.6, 0.8, 1),

+ nrow = 3, byrow = TRUE)

> # Generate random sample

> random\_sample = rmvnorm(100, mean = mean\_vector, sigma = cov\_matrix)

> colnames(random\_sample) = c("A", "B", "C")

> head(random\_sample)

A B C

[1,] -0.40513150 -0.134055178 0.8625713

[2,] 0.90383122 0.594029399 1.5444986

[3,] -0.41583846 -0.003575464 0.4301784

[4,] -0.25477666 -0.036308741 1.0337095

[5,] 0.15975810 0.567255648 1.3653147

[6,] -0.06192353 -0.115750171 1.2797118

> D = rnorm(100, mean = .5, sd = sqrt(0.5))

> X = cbind(random\_sample, D)

> X

A B C D

[1,] -4.051315e-01 -0.134055178 0.862571277 -0.811997902

[2,] 9.038312e-01 0.594029399 1.544498624 -0.220234017

[3,] -4.158385e-01 -0.003575464 0.430178385 -0.239005136

[4,] -2.547767e-01 -0.036308741 1.033709451 0.039470084

[5,] 1.597581e-01 0.567255648 1.365314660 0.770597499

[6,] -6.192353e-02 -0.115750171 1.279711848 1.229663283

[7,] 4.852885e-02 1.312681411 1.349403256 0.592684003

[8,] 9.673955e-01 0.849066529 1.973305116 0.766714833

[9,] -1.071612e+00 -0.990930438 -0.012061988 -0.834801993

[10,] -2.166080e-01 -0.740405730 0.793594420 0.867850668

[11,] 6.142234e-01 1.329551805 1.584614297 1.107513027

[12,] -4.374567e-01 -0.697185130 0.492593291 -0.067445170

[13,] 5.935316e-01 0.855239330 2.151670266 0.152131383

[14,] 1.151447e+00 0.900567846 2.425301487 0.350832771

[15,] -1.427533e+00 -0.447297033 0.365877293 0.172131126

[16,] 1.331119e+00 0.799308170 2.329117101 0.711764033

[17,] -8.488934e-01 -1.689730163 -0.159328689 0.071865351

[18,] -9.866910e-01 0.210770523 2.312957606 0.189571542

[19,] -1.927760e+00 -0.725537159 -0.375563314 -0.009649779

[20,] 1.967141e+00 1.527945809 1.282454018 1.052112532

[21,] -3.508073e-01 -0.321219328 0.575266390 -0.364285500

[22,] 2.143407e+00 0.879250389 1.059476659 1.130169347

[23,] 6.258394e-01 0.398745730 1.178117092 0.679550564

[24,] -1.552514e-01 -0.115254921 0.230086657 0.453460758

[25,] -8.403370e-01 0.000752817 -0.409342490 0.642457998

[26,] -3.731143e-01 -0.764153085 0.789926291 2.251998835

[27,] 2.442322e-01 -0.892306100 1.173631393 0.833579600

[28,] -1.072890e+00 -1.910962449 -1.262675407 1.437763658

[29,] 6.096328e-01 0.255493181 1.297697108 0.972770523

[30,] 1.406614e+00 1.695464411 1.824038462 0.402847479

[31,] -1.151448e-01 0.557136480 0.672454439 -0.497849076

[32,] -6.369914e-01 0.103646688 2.641906505 1.348784891

[33,] -8.466774e-01 -0.497754965 -0.065625550 0.711488643

[34,] -8.342041e-01 -0.191538408 1.770429492 0.822601122

[35,] -4.185246e-01 -0.114944119 -0.043926001 0.809768239

[36,] -7.627752e-01 -0.948040715 0.256836869 0.347957095

[37,] 1.084999e+00 0.497986512 1.409847648 0.279296081

[38,] -3.804939e-01 -0.984563694 -0.445432840 0.010236334

[39,] 1.174322e-02 0.388640728 1.864986085 -1.194390258

[40,] 4.527836e-02 0.317863684 2.228453635 -0.620251549

[41,] -6.454755e-01 -0.218604964 0.474543263 1.743622802

[42,] -4.028176e-01 -1.295437308 0.006604825 -1.170813271

[43,] 5.717901e-02 0.506229982 0.937152607 0.778014635

[44,] -1.196182e-01 -0.567446010 0.154147109 0.900178502

[45,] 7.429866e-01 0.470711519 2.159862179 -0.238212670

[46,] 4.372895e-01 0.249831413 1.445180992 1.223450347

[47,] 5.389793e-02 0.183922337 0.820983308 0.137491799

[48,] -1.647050e+00 -0.484357801 1.121638722 0.533754926

[49,] -1.428949e+00 -1.490395852 -0.084585033 1.484912770

[50,] -2.462196e-01 0.181423453 1.165045198 0.194176935

[51,] 1.587623e+00 1.761298588 1.921878989 0.404421906

[52,] 4.350893e-05 0.053189902 0.802134755 1.417637325

[53,] -7.581264e-02 -0.017721942 1.162901714 0.449988824

[54,] 3.594930e-02 -0.114359659 0.985383028 0.202429530

[55,] 1.753299e-01 0.573487605 1.291707786 1.307049506

[56,] -8.254685e-01 -0.751414727 0.136184585 0.197975967

[57,] 6.825597e-01 1.031674255 1.954219497 1.486887443

[58,] 2.096162e-01 0.129488685 1.265316957 0.290564044

[59,] -8.060108e-01 -1.529214856 0.455227582 0.912002562

[60,] 7.284559e-01 1.771815172 3.335652672 -1.392379922

[61,] -8.123898e-01 0.016045192 1.050116497 -0.490429626

[62,] -2.927075e-02 0.579130859 1.749869799 2.836387657

[63,] -4.692838e-01 -0.727552802 -0.097069852 1.105833209

[64,] -7.056223e-01 -0.551500704 0.530914896 1.320962601

[65,] 1.189322e+00 0.964399867 2.784703449 0.697238163

[66,] 3.150636e-01 1.166019385 1.689434261 0.490419767

[67,] -7.059034e-01 0.029755207 0.998414699 -1.392485660

[68,] 1.657940e-01 0.755962398 1.716073742 0.585782273

[69,] 3.324759e-01 -0.638868709 0.333277561 0.146062479

[70,] -2.057666e+00 -2.730173435 -0.207072209 1.475579607

[71,] 1.040732e+00 0.507074903 0.414976507 0.212924081

[72,] -1.185714e+00 -1.292138734 1.131122583 1.487072944

[73,] -4.789474e-01 -0.558347138 1.288708213 1.022266350

[74,] -8.567989e-01 -1.345197650 0.288473231 1.020864863

[75,] 1.913462e-01 0.313513916 1.433797099 -0.018371463

[76,] -5.799782e-01 -0.241357635 0.056630421 -0.842368720

[77,] 1.040719e+00 0.255486074 1.468765685 0.815284278

[78,] 2.297541e-02 -0.651217079 -0.377353558 0.822625150

[79,] -6.337746e-01 -0.773426031 0.334036547 -0.579184395

[80,] 1.083009e+00 1.003610180 1.698601552 0.443019485

[81,] 6.176704e-02 1.150203453 0.482421664 0.259576195

[82,] -6.018006e-01 0.739925044 3.140016803 0.537958057

[83,] -9.128311e-01 -1.332923270 0.021261940 0.010940308

[84,] 1.791390e+00 0.220236999 0.858482093 0.285396701

[85,] 2.364423e-01 0.771440161 1.830075446 -0.345002537

[86,] -7.377696e-01 -0.971724438 1.740453393 1.221062181

[87,] -1.245778e+00 -0.824991584 0.680834289 0.161688846

[88,] -5.934073e-02 -0.685655928 0.871164519 0.033668700

[89,] 1.186432e+00 1.112219312 2.175920079 0.980040406

[90,] -5.626604e-01 -0.017313844 1.691520648 0.425502534

[91,] 8.796845e-01 0.978914808 2.226668506 0.179173157

[92,] -3.349351e-01 -0.017819800 0.431501691 1.448540353

[93,] -2.813757e-02 0.251574193 1.498041245 1.197177023

[94,] -1.758676e+00 -0.005827627 1.190509167 -0.345321249

[95,] 1.798972e+00 1.721737818 2.354333110 0.093513605

[96,] 4.086650e-01 0.575346229 1.256163464 1.194372751

[97,] 9.229551e-01 0.911923722 2.048130267 -0.671946917

[98,] -6.563028e-01 -0.429726059 1.027647000 -0.291709938

[99,] -1.057394e+00 -0.647312925 0.828599412 -0.327897877

[100,] 1.586745e+00 0.855706452 0.617856773 -0.305431118

> ##b Giá trị trung bình mẫu

> colMeans(X)

A B C D

-0.04635691 0.02601155 1.06545274 0.44113661

> # Ma trận hiệp phương sai

> cov(X)

A B C D

A 0.786666193 0.58607178 0.39070579 0.005553676

B 0.586071775 0.75756778 0.53491442 -0.064086192

C 0.390705795 0.53491442 0.74642114 -0.024396066

D 0.005553676 -0.06408619 -0.02439607 0.584130109

> # Ma trận tương quan mẫu

> cor(X)

A B C D

A 1.000000000 0.75918010 0.50987374 0.008192765

B 0.759180102 1.00000000 0.71134715 -0.096338287

C 0.509873737 0.71134715 1.00000000 -0.036946478

D 0.008192765 -0.09633829 -0.03694648 1.000000000

> ## c. Biểu đồ tán xạ

> pairs(X, lwd = 1, pch = 16, cex = .5, col = "red", gap = 0, xaxt = "n", yaxt = "n")

> #Biểu đồ nhiệt của ma trận tương quan mẫu

> library(reshape2)

> library(ICSNP)

> cor\_df = round(cor(X), 2)

> cor\_df

A B C D

A 1.00 0.76 0.51 0.01

B 0.76 1.00 0.71 -0.10

C 0.51 0.71 1.00 -0.04

D 0.01 -0.10 -0.04 1.00

> melted\_cor = melt(cor\_df) # chuyển từ ma trận tương quan sang tương quan 2 biến theo cột

> head(melted\_cor)

Var1 Var2 value

1 A A 1.00

2 B A 0.76

3 C A 0.51

4 D A 0.01

5 A B 0.76

6 B B 1.00

> melted\_cor

Var1 Var2 value

1 A A 1.00

2 B A 0.76

3 C A 0.51

4 D A 0.01

5 A B 0.76

6 B B 1.00

7 C B 0.71

8 D B -0.10

9 A C 0.51

10 B C 0.71

11 C C 1.00

12 D C -0.04

13 A D 0.01

14 B D -0.10

15 C D -0.04

16 D D 1.00

> dim(melted\_cor)

[1] 16 3

> library(ggplot2)

> # create corrrlation heatmap

> ggplot(data = melted\_cor, aes(x=Var1, y=Var2, fill=value)) +

+ geom\_tile() + # khung có màu xanh nhiều mức

+ geom\_text(aes(Var2, Var1, label = value), size = 3) + # giá trị tương quan trong từng ô

+ scale\_fill\_gradient2(low = "blue", high = "red", limit = c(-1,1), name="Correlation") +

+ theme(axis.title.x = element\_blank(),

+ axis.title.y = element\_blank(),

+ panel.background = element\_blank())

> ## d. Kiểm tra tính chuẩn 4−chiều.

> # Để kiểm tra 1 vecto nn X có tuân theo pp chuẩn k-chiều ko thì làm theo 2 bước:

> ## B1: Kiểm tra tính chuẩn 1-chiều của từng biến trong X

> ## Bt: H0: từng biến tuân theo pp chuẩn với H1: k tuân theo pp chuẩn

> sapply(colnames(X), function(x) {

+ shapiro.test(X[,x])$p.value

+ })

A B C D

0.3287910 0.5199584 0.9445196 0.5090947

> ## B2: Ktra tính chuẩn nhiều chiều shapiro.test(qnorm(pchisq(mah,4))

> ## B2 chính là ktra tính chuẩn của phân vị pp chuẩn tắc tương ứng với xs của pp khi bình phương xét trên dữ liệu

> ## là mah với 4 bậc tự do

> # Mahalanobis distance: D2i = (Yi - Yh)' S^(-1)(Yi - Yh)

> mah = mahalanobis(X, colMeans(X), var(X))

> dim(X)

[1] 100 4

> length(mah)

[1] 100

> mah

[1] 2.85378374 2.11993062 2.06905563 0.33708822 0.85407392 1.29620238 6.26964314 1.82809584 4.88399973

[10] 1.68448148 4.40975756 1.40413994 1.70229828 3.51783728 3.92638485 4.27590727 5.35345689 6.55148248

[19] 8.63806192 7.57345776 1.49147797 9.18662029 0.74851450 1.50460244 8.34068693 6.26551127 5.99393908

[28] 9.14065288 1.16617069 4.23180404 3.83100208 8.71550452 2.73368116 3.32327008 3.50101054 1.34780627

[37] 2.17965162 4.06025428 5.60921831 4.65944841 4.47781417 9.53985932 1.52689616 1.75121963 3.69276238

[46] 1.46397271 0.50391829 5.09925443 4.80177710 0.39839006 4.57374073 1.95548533 0.04768447 0.28055049

[55] 2.21442219 1.42533075 3.66298001 0.21684497 4.60977069 12.56681434 2.87072460 12.03810012 2.53963962

[64] 1.98032829 5.24587043 2.76376235 6.53775600 1.22625850 3.50237901 12.29071030 4.35248597 6.38381886

[73] 1.97320397 3.08202623 0.54005323 4.86534600 2.89146198 4.13059564 2.98189808 1.68292663 9.00833245

[82] 10.53320632 3.12812647 9.02104106 1.95811113 7.17021306 2.05355533 2.45283042 2.90921970 1.76029553

[91] 2.09744930 3.32905991 1.40854054 9.01201224 4.85425041 1.65384571 3.83594464 1.64763721 2.44675619

[100] 7.45077205

> qqplot(qchisq((1:nrow(X) - 1/2) / nrow(X), df = 4), (mah))

> abline(a = 0, b = 1, col = "red")

> shapiro.test(qnorm(pchisq(mah, 4))) # Kiểm tra tính chuẩn nhiều chiều

Shapiro-Wilk normality test

data: qnorm(pchisq(mah, 4))

W = 0.98532, p-value = 0.335

> library(ISLR)

> data = Smarket

> head(data)

Year Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today Direction

1 2001 0.381 -0.192 -2.624 -1.055 5.010 1.1913 0.959 Up

2 2001 0.959 0.381 -0.192 -2.624 -1.055 1.2965 1.032 Up

3 2001 1.032 0.959 0.381 -0.192 -2.624 1.4112 -0.623 Down

4 2001 -0.623 1.032 0.959 0.381 -0.192 1.2760 0.614 Up

5 2001 0.614 -0.623 1.032 0.959 0.381 1.2057 0.213 Up

6 2001 0.213 0.614 -0.623 1.032 0.959 1.3491 1.392 Up

> ## a. Tính giá trị trung bình mẫu, ma trận tương quan mẫu và ma trận hiệp phương sai mẫu của X.

> X = data[, c("Lag1", "Lag2", "Lag3", "Lag4", "Lag5", "Volume", "Today")]

> # Tính giá trị trung bình mẫu

> mean\_values <- colMeans(X)

> # Tính ma trận hiệp phương sai mẫu

> covariance\_matrix <- cov(X)

> # Tính ma trận tương quan mẫu

> correlation\_matrix <- cor(X)

> # Hiển thị kết quả

> mean\_values

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

0.0038344 0.0039192 0.0017160 0.0016360 0.0056096 1.4783050 0.0031384

> correlation\_matrix

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

Lag1 1.000000000 -0.026294328 -0.010803402 -0.002985911 -0.005674606 0.04090991 -0.026155045

Lag2 -0.026294328 1.000000000 -0.025896670 -0.010853533 -0.003557949 -0.04338321 -0.010250033

Lag3 -0.010803402 -0.025896670 1.000000000 -0.024051036 -0.018808338 -0.04182369 -0.002447647

Lag4 -0.002985911 -0.010853533 -0.024051036 1.000000000 -0.027083641 -0.04841425 -0.006899527

Lag5 -0.005674606 -0.003557949 -0.018808338 -0.027083641 1.000000000 -0.02200231 -0.034860083

Volume 0.040909908 -0.043383215 -0.041823686 -0.048414246 -0.022002315 1.00000000 0.014591823

Today -0.026155045 -0.010250033 -0.002447647 -0.006899527 -0.034860083 0.01459182 1.000000000

> covariance\_matrix

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

Lag1 1.291175062 -0.033950025 -0.013978596 -0.003863730 -0.007399459 0.016751517 -0.033771790

Lag2 -0.033950025 1.291132819 -0.033507330 -0.014044104 -0.004639348 -0.017763979 -0.013234781

Lag3 -0.013978596 -0.033507330 1.296644424 -0.031187581 -0.024577209 -0.017161918 -0.003167126

Lag4 -0.003863730 -0.014044104 -0.031187581 1.296805622 -0.035392903 -0.019867521 -0.008928177

Lag5 -0.007399459 -0.004639348 -0.024577209 -0.035392903 1.316871484 -0.009098570 -0.045457562

Volume 0.016751517 -0.017763979 -0.017161918 -0.019867521 -0.009098570 0.129857232 0.005975148

Today -0.033771790 -0.013234781 -0.003167126 -0.008928177 -0.045457562 0.005975148 1.291255147

> ## b. Vẽ biểu đồ phân tán của các biến và biểu đồ nhiệt của ma trận tương quan mẫu của X.

> # Biểu đồ tán xạ

> pairs(X, lwd = 1, pch = 16, cex = .5, col = "red", gap = 0, xaxt = "n", yaxt = "n")

> #Biểu đồ nhiệt của ma trận tương quan mẫu

> library(reshape2)

> library(ICSNP)

> cor\_df = round(cor(X), 2)

> cor\_df

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

Lag1 1.00 -0.03 -0.01 0.00 -0.01 0.04 -0.03

Lag2 -0.03 1.00 -0.03 -0.01 0.00 -0.04 -0.01

Lag3 -0.01 -0.03 1.00 -0.02 -0.02 -0.04 0.00

Lag4 0.00 -0.01 -0.02 1.00 -0.03 -0.05 -0.01

Lag5 -0.01 0.00 -0.02 -0.03 1.00 -0.02 -0.03

Volume 0.04 -0.04 -0.04 -0.05 -0.02 1.00 0.01

Today -0.03 -0.01 0.00 -0.01 -0.03 0.01 1.00

> melted\_cor = melt(cor\_df) # chuyển từ ma trận tương quan sang tương quan 2 biến theo cột

> head(melted\_cor)

Var1 Var2 value

1 Lag1 Lag1 1.00

2 Lag2 Lag1 -0.03

3 Lag3 Lag1 -0.01

4 Lag4 Lag1 0.00

5 Lag5 Lag1 -0.01

6 Volume Lag1 0.04

> melted\_cor

Var1 Var2 value

1 Lag1 Lag1 1.00

2 Lag2 Lag1 -0.03

3 Lag3 Lag1 -0.01

4 Lag4 Lag1 0.00

5 Lag5 Lag1 -0.01

6 Volume Lag1 0.04

7 Today Lag1 -0.03

8 Lag1 Lag2 -0.03

9 Lag2 Lag2 1.00

10 Lag3 Lag2 -0.03

11 Lag4 Lag2 -0.01

12 Lag5 Lag2 0.00

13 Volume Lag2 -0.04

14 Today Lag2 -0.01

15 Lag1 Lag3 -0.01

16 Lag2 Lag3 -0.03

17 Lag3 Lag3 1.00

18 Lag4 Lag3 -0.02

19 Lag5 Lag3 -0.02

20 Volume Lag3 -0.04

21 Today Lag3 0.00

22 Lag1 Lag4 0.00

23 Lag2 Lag4 -0.01

24 Lag3 Lag4 -0.02

25 Lag4 Lag4 1.00

26 Lag5 Lag4 -0.03

27 Volume Lag4 -0.05

28 Today Lag4 -0.01

29 Lag1 Lag5 -0.01

30 Lag2 Lag5 0.00

31 Lag3 Lag5 -0.02

32 Lag4 Lag5 -0.03

33 Lag5 Lag5 1.00

34 Volume Lag5 -0.02

35 Today Lag5 -0.03

36 Lag1 Volume 0.04

37 Lag2 Volume -0.04

38 Lag3 Volume -0.04

39 Lag4 Volume -0.05

40 Lag5 Volume -0.02

41 Volume Volume 1.00

42 Today Volume 0.01

43 Lag1 Today -0.03

44 Lag2 Today -0.01

45 Lag3 Today 0.00

46 Lag4 Today -0.01

47 Lag5 Today -0.03

48 Volume Today 0.01

49 Today Today 1.00

> dim(melted\_cor)

[1] 49 3

> library(ggplot2)

> # create corrrlation heatmap

> ggplot(data = melted\_cor, aes(x=Var1, y=Var2, fill=value)) +

+ geom\_tile() + # khung có màu xanh nhiều mức

+ geom\_text(aes(Var2, Var1, label = value), size = 3) + # giá trị tương quan trong từng ô

+ scale\_fill\_gradient2(low = "blue", high = "red", limit = c(-1,1), name="Correlation") +

+ theme(axis.title.x = element\_blank(),

+ axis.title.y = element\_blank(),

+ panel.background = element\_blank())

> # Để kiểm tra 1 vecto nn X có tuân theo pp chuẩn k-chiều ko thì làm theo 2 bước:

> ## B1: Kiểm tra tính chuẩn 1-chiều của từng biến trong X

> ## Bt: H0: từng biến tuân theo pp chuẩn với H1: k tuân theo pp chuẩn

> sapply(colnames(X), function(x) {

+ shapiro.test(X[,x])$p.value

+ })

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

8.888786e-15 8.798395e-15 1.034889e-14 1.048548e-14 2.148901e-15 1.792894e-19 9.156928e-15

> ## B2: Ktra tính chuẩn nhiều chiều shapiro.test(qnorm(pchisq(mah,4))

> ## B2 chính là ktra tính chuẩn của phân vị pp chuẩn tắc tương ứng với xs của pp khi bình phương xét trên dữ liệu

> ## là mah với 4 bậc tự do

> # Mahalanobis distance: D2i = (Yi - Yh)' S^(-1)(Yi - Yh)

> mah = mahalanobis(X, colMeans(X), var(X))

> dim(X)

[1] 1250 7

> length(mah)

[1] 1250

> mah

1 2 3 4 5 6 7 8 9 10 11

26.6130295 8.3763332 7.3383025 2.4825363 2.8163423 3.8525447 3.0352534 2.2409313 4.0081209 3.5067667 3.2758445

12 13 14 15 16 17 18 19 20 21 22

2.0025908 2.9753885 3.8071139 2.1020624 1.6077596 4.5078957 4.8191917 5.1282134 4.9185043 4.3584392 6.0161559

23 24 25 26 27 28 29 30 31 32 33

4.7546714 5.7972999 5.2414583 4.7288192 7.1837482 7.4769490 10.2837390 9.2698313 9.0380124 11.3003175 9.5547492

34 35 36 37 38 39 40 41 42 43 44

8.3602205 5.0237819 5.1274985 5.1227741 6.0763295 4.4667222 2.5940944 7.2683497 21.5988403 22.5298733 26.8173917

45 46 47 48 49 50 51 52 53 54 55

26.5160774 30.5594875 26.2092696 17.8108775 18.3690576 12.8613542 15.5683267 13.7929401 17.4454127 16.5608263 13.9755379

56 57 58 59 60 61 62 63 64 65 66

15.1829300 12.8336613 20.7479837 16.3718109 26.1394083 28.8858235 28.5791294 33.3410731 23.8320263 25.8076214 11.9774045

67 68 69 70 71 72 73 74 75 76 77

10.9650713 21.8420345 17.3563314 16.4178801 16.3187744 18.4052032 18.8698443 7.6286044 7.5395110 8.0674733 6.6468623

78 79 80 81 82 83 84 85 86 87 88

6.0882298 5.1139276 7.2420867 6.0597056 6.6940417 5.2174456 4.3205790 3.5580627 3.3108769 3.9023778 8.2984576

89 90 91 92 93 94 95 96 97 98 99

6.9515007 6.9597680 9.3317016 9.0636813 10.2027188 4.7987684 6.1792685 8.6119134 7.4206858 6.6425219 4.7365758

100 101 102 103 104 105 106 107 108 109 110

6.0833341 7.7333749 5.6690132 4.0666547 4.4383931 7.7702706 6.4606151 4.4376515 6.2401689 5.4803362 4.6005995

111 112 113 114 115 116 117 118 119 120 121

5.2426134 5.4104966 4.5438247 2.7493835 3.2549489 3.8794668 3.2275481 4.1496352 2.6154137 3.3678923 3.1716245

122 123 124 125 126 127 128 129 130 131 132

8.5836541 9.3687661 7.9323945 9.9887522 8.0125390 11.7620285 10.8651837 8.3286805 8.9514996 6.8645519 6.8798955

133 134 135 136 137 138 139 140 141 142 143

2.6734841 5.0456310 7.1734957 7.3360489 7.9081700 7.8454980 8.6582439 7.2872123 4.1819502 1.5285992 1.2224397

144 145 146 147 148 149 150 151 152 153 154

3.7341491 5.0257539 5.6118168 4.7020293 5.2399069 6.0308311 6.0898065 5.1612527 2.1043556 4.3946739 5.1538576

155 156 157 158 159 160 161 162 163 164 165

7.3723486 6.0402590 5.4652814 8.9201913 6.7921753 9.2343379 7.8258074 10.1821338 9.1063388 8.5548568 6.1891527

166 167 168 169 170 171 172 173 174 175 176

7.5665984 9.3864483 7.1862104 26.4357085 32.0961034 28.7411296 35.2963917 34.1510640 46.5722594 25.0683658 24.9468829

177 178 179 180 181 182 183 184 185 186 187

23.2033172 19.7290911 17.5620799 7.4101245 9.5831679 9.5813027 8.4342381 5.0552146 6.6217176 8.5502321 7.0973363

188 189 190 191 192 193 194 195 196 197 198

7.2363056 6.9102742 7.9642514 9.2905759 5.4188407 4.3280195 5.8828629 6.9490958 5.4465472 4.1038608 3.8538372

199 200 201 202 203 204 205 206 207 208 209

7.9880345 9.3271848 8.5728668 12.6625357 11.4364312 13.4091550 9.8923623 7.6424678 7.6495370 3.5516615 4.3559047

210 211 212 213 214 215 216 217 218 219 220

6.3239667 2.9646449 2.7941539 2.8126445 3.8009108 4.1631002 1.7070170 4.1542927 12.0809897 4.0467946 5.1790385

221 222 223 224 225 226 227 228 229 230 231

5.2210907 5.1711007 4.6569500 6.1634671 9.1720875 7.1924165 6.2622863 8.4619044 8.0889430 6.4066376 4.5807381

232 233 234 235 236 237 238 239 240 241 242

4.4877276 5.0180746 3.7175903 3.5984148 3.9849215 2.3168367 2.5695770 9.7553708 5.2155757 4.0923016 3.9334866

243 244 245 246 247 248 249 250 251 252 253

3.7252118 3.2161564 2.6665386 2.5501698 2.7216843 2.1160130 1.5588760 1.8773259 2.2697482 1.9927079 3.7381572

254 255 256 257 258 259 260 261 262 263 264

4.1057848 5.0488887 4.7533575 4.9906136 4.4637629 2.4959579 1.8968753 7.9710941 9.0067462 10.9556496 9.3309029

265 266 267 268 269 270 271 272 273 274 275

14.2496936 14.2375751 9.0187315 7.4570011 7.4236245 8.5834309 4.7761811 5.4754183 4.7364053 5.2456238 6.1506648

276 277 278 279 280 281 282 283 284 285 286

6.5487247 7.6446240 7.4996917 9.8647063 8.9996923 6.3530827 4.8838392 7.2381064 9.8059629 7.4940771 9.1467147

287 288 289 290 291 292 293 294 295 296 297

9.1899228 9.4564569 5.3974193 2.9551457 2.9905449 1.3992179 2.7818317 1.8623290 2.6707904 4.1936313 3.1798124

298 299 300 301 302 303 304 305 306 307 308

3.3174163 4.3318133 5.5471493 4.8671061 3.1404353 3.2441505 4.0901576 2.4555942 2.1376042 1.8819202 2.6960229

309 310 311 312 313 314 315 316 317 318 319

3.1737286 2.6216357 5.7232409 6.0153309 6.7672098 11.5092003 10.4497910 9.3060483 5.0671397 7.0329898 6.7871496

320 321 322 323 324 325 326 327 328 329 330

2.7751755 2.7882997 4.1899216 5.2492328 4.2092164 4.2711371 3.8579616 4.7593544 6.3326342 6.2075358 15.5959530

331 332 333 334 335 336 337 338 339 340 341

16.1239812 19.0152388 20.1657168 21.9185596 20.4508346 10.3775549 9.4059517 8.5705577 8.0781253 4.0917105 5.1970095

342 343 344 345 346 347 348 349 350 351 352

5.5699518 7.7273147 5.7372952 4.3564441 3.2953601 7.5871287 5.9774169 5.7670702 8.7968281 8.4794829 8.5266965

353 354 355 356 357 358 359 360 361 362 363

6.3327892 6.6420036 7.0883666 3.4627924 9.6972786 10.2778625 10.0180835 10.8198084 12.2782198 12.1734791 8.3046511

364 365 366 367 368 369 370 371 372 373 374

8.4213614 10.0075067 7.9825679 11.3697640 11.9117621 10.8605496 20.3830953 24.7459121 24.3077240 29.4000577 27.2745020

375 376 377 378 379 380 381 382 383 384 385

27.7236398 16.1997146 26.1324599 13.5049957 18.0479029 21.7170216 40.5706411 40.2380114 62.6275001 66.0298820 59.2145753

386 387 388 389 390 391 392 393 394 395 396

63.5417353 56.7647578 53.8874606 36.4791183 37.7516089 42.8807816 27.9966651 31.0198830 38.3559994 31.7988739 27.6779338

397 398 399 400 401 402 403 404 405 406 407

23.3068894 28.2359490 25.6168636 17.1834496 21.6054101 22.7070200 20.6991292 9.8045929 12.3446063 13.6513664 11.6971566

408 409 410 411 412 413 414 415 416 417 418

11.1148354 10.7931230 9.0825102 20.4200924 19.8779095 20.5243774 19.6126552 21.4702903 21.4993103 8.2347763 12.7912094

419 420 421 422 423 424 425 426 427 428 429

8.6121255 6.4171127 10.0243191 8.0824761 15.5958752 10.5038113 12.4385513 14.7921448 15.7823569 17.8609754 19.0677438

430 431 432 433 434 435 436 437 438 439 440

20.8412075 31.2970545 33.0059886 29.3116678 31.3905787 27.2183874 26.3263446 21.5769624 25.6536447 37.1449992 33.3180397

441 442 443 444 445 446 447 448 449 450 451

48.4161789 51.4496489 49.0743048 39.3434487 28.5740747 28.3258963 11.9375570 9.4480458 8.3103493 8.0629262 6.4178608

452 453 454 455 456 457 458 459 460 461 462

6.1624415 6.0847956 6.8320842 4.9002166 5.1017633 5.1191695 8.6253727 8.5308743 9.6584610 10.6294844 9.3527715

463 464 465 466 467 468 469 470 471 472 473

12.9878461 9.3526869 9.6085203 6.5552702 8.9373036 12.7301471 14.4265207 7.8755985 10.3553530 16.3432265 13.1231112

474 475 476 477 478 479 480 481 482 483 484

14.6487104 11.2013291 11.1177631 8.8688306 3.8201397 7.7829861 8.5568488 7.1985808 7.0423258 7.3456116 10.7316334

485 486 487 488 489 490 491 492 493 494 495

8.0092272 7.7838285 7.9988263 9.0965235 8.2630959 5.0816398 11.3876378 8.3482630 7.5991852 4.1466732 12.3417763

496 497 498 499 500 501 502 503 504 505 506

11.4863915 15.2369954 13.0037297 14.4843497 17.3731785 8.5217910 8.3723178 4.9634500 6.2914679 4.7813750 3.6208767

507 508 509 510 511 512 513 514 515 516 517

5.7314173 6.7204070 7.0019428 12.3880985 14.2836583 13.4664713 11.9417861 14.9231480 15.2685606 8.9904595 8.7871678

518 519 520 521 522 523 524 525 526 527 528

7.5328709 7.6143046 4.3850134 3.9728063 4.5987554 3.8794188 3.7672624 6.4136544 8.9916863 9.2061959 9.7957268

529 530 531 532 533 534 535 536 537 538 539

9.3555484 11.2303196 8.5136899 6.7225244 7.3049616 7.1928335 6.0581168 5.6065015 5.6954133 4.9669122 4.5255728

540 541 542 543 544 545 546 547 548 549 550

9.5455397 9.7626155 7.7662732 16.1170076 16.0167888 24.4734640 20.3773156 20.5549458 20.1466721 14.8981136 24.7845560

551 552 553 554 555 556 557 558 559 560 561

15.3059225 15.4846442 14.8178408 15.3540408 14.3329461 5.1997569 9.3562373 9.2930350 9.3737579 9.2329462 6.7569093

562 563 564 565 566 567 568 569 570 571 572

7.1331549 2.3209165 2.4162748 5.8460200 6.4049808 6.3381877 6.8451257 6.3005102 10.9369703 7.7042966 8.1352374

573 574 575 576 577 578 579 580 581 582 583

8.4660164 8.6756652 9.0597681 4.9748039 5.2407075 5.6141073 4.3820731 2.4951280 2.8285941 3.4326492 5.0143766

584 585 586 587 588 589 590 591 592 593 594

4.6592504 4.4101884 3.9082449 4.1132109 3.4552424 6.5077007 5.5568732 5.5421811 6.1150062 5.7085833 8.9140383

595 596 597 598 599 600 601 602 603 604 605

3.8736075 4.0603097 6.0082202 5.2833989 5.3978807 3.9114467 4.1840252 4.4372797 4.6043240 4.0131718 5.4057547

606 607 608 609 610 611 612 613 614 615 616

3.1726897 3.8862395 7.6595332 6.7004345 5.8039854 6.3932361 6.4348718 7.3975747 3.4854263 4.1385515 4.7380185

617 618 619 620 621 622 623 624 625 626 627

3.8163681 4.0371121 2.7080501 3.6527264 3.5894154 9.0644692 4.9567488 5.2186812 6.2085267 5.6110968 6.0901957

628 629 630 631 632 633 634 635 636 637 638

2.7396069 3.0173199 4.3424588 3.9457497 4.6485280 5.3056957 4.8658796 5.0141407 6.1208426 5.0636895 3.8545898

639 640 641 642 643 644 645 646 647 648 649

3.1566539 3.2266205 3.6290309 1.4628602 4.1049344 3.4927225 3.7592874 3.9037546 4.3446436 5.2828158 2.5491464

650 651 652 653 654 655 656 657 658 659 660

2.4449136 2.2156652 7.4626938 3.0973648 1.5799223 1.6220530 1.6127390 1.7421543 3.1195410 1.7261905 2.7042604

661 662 663 664 665 666 667 668 669 670 671

2.1503845 4.3156452 2.2780375 2.6223000 2.5032687 3.0112231 3.3976348 2.8378215 3.0221907 3.1539010 3.1956906

672 673 674 675 676 677 678 679 680 681 682

4.4021204 3.1411270 3.5188890 3.2059879 4.5317413 4.8652203 6.1374983 6.1001881 5.2162767 5.7417413 5.4521475

683 684 685 686 687 688 689 690 691 692 693

8.6600106 5.9802789 6.6760437 6.4052071 7.0505976 5.2954848 1.7101642 1.4885464 2.1156940 2.6244600 1.2469327

694 695 696 697 698 699 700 701 702 703 704

0.8493895 1.4515861 1.7098733 2.0642904 2.8866091 3.1264904 3.0891243 2.2311885 3.9398042 3.8857473 2.1707213

705 706 707 708 709 710 711 712 713 714 715

2.3271162 2.4815679 2.6039759 0.8491205 1.1263293 1.2251965 1.3651918 1.3575408 2.5267514 1.8597705 1.9531994

716 717 718 719 720 721 722 723 724 725 726

2.1261724 2.5103333 3.0445362 2.7617144 2.6770246 4.3711813 4.0103027 3.3929259 3.7815931 10.4332748 3.3076368

727 728 729 730 731 732 733 734 735 736 737

1.2778985 1.4012175 1.6751642 2.3977240 2.2718557 1.6028865 2.6103090 2.4946756 2.7252834 2.1884879 1.7104177

738 739 740 741 742 743 744 745 746 747 748

2.8100989 1.7983399 2.1083288 1.9942806 2.0524982 8.1832068 11.2121720 2.8425258 2.8942962 2.7996827 3.2384247

749 750 751 752 753 754 755 756 757 758 759

2.5530693 1.3522983 1.9842520 3.4085320 2.7873357 1.2293871 1.8547831 1.7010218 2.2309256 1.7593202 2.0217978

760 761 762 763 764 765 766 767 768 769 770

2.0969261 1.4101104 2.1658213 2.3946223 4.1966323 4.6577012 4.9365587 3.7169238 2.5958965 2.3422193 1.1317153

771 772 773 774 775 776 777 778 779 780 781

1.9569556 1.9043308 2.2229068 2.8883319 2.9680269 2.6978589 2.3673931 2.3764907 2.2639694 1.5358558 1.3552693

782 783 784 785 786 787 788 789 790 791 792

1.2015341 0.5725708 0.5476208 0.3415804 0.9410454 1.1265399 1.1263479 1.2409281 1.4334503 1.6200684 1.5603852

793 794 795 796 797 798 799 800 801 802 803

2.6576022 4.7765431 6.7062407 7.3856465 6.8073100 7.3554959 5.7641396 4.9615582 5.2602768 3.6594067 3.4196967

804 805 806 807 808 809 810 811 812 813 814

4.4319551 4.4566671 4.8324562 3.6272134 3.6727598 3.9078005 2.3284169 2.9732618 1.4500593 1.6536792 1.6446094

815 816 817 818 819 820 821 822 823 824 825

2.2007775 3.5643404 2.0830900 2.0405268 1.9108377 1.8610920 4.2754467 2.2964828 4.2347038 4.5764919 3.7862650

826 827 828 829 830 831 832 833 834 835 836

3.9176877 3.4056688 4.8527004 3.5385574 3.1966122 2.9391985 3.1170335 1.7768706 2.8628824 3.7259424 4.6449328

837 838 839 840 841 842 843 844 845 846 847

3.2141664 3.4805255 2.8433517 2.3643441 1.7076057 1.4532683 1.2955634 1.9868241 1.7726323 3.1196133 2.2711504

848 849 850 851 852 853 854 855 856 857 858

2.5328839 2.4757545 2.8931271 2.7718903 1.1398991 1.3753568 3.8248308 3.3454793 3.9233852 3.6246950 4.4583299

859 860 861 862 863 864 865 866 867 868 869

4.4064637 2.0776822 2.6564882 1.4871669 1.2255980 1.4963247 0.9678017 0.9296548 1.1598695 2.0558241 1.1260156

870 871 872 873 874 875 876 877 878 879 880

1.1315408 1.3429053 1.3553359 2.8391140 1.9503162 2.3742366 2.1622467 2.0462388 2.2573519 1.3470401 0.8693780

881 882 883 884 885 886 887 888 889 890 891

0.5473774 0.4522562 1.0519233 2.2067830 2.4629438 3.0015459 2.6899003 3.3655945 3.0006491 1.7548928 1.6777573

892 893 894 895 896 897 898 899 900 901 902

1.3276161 1.6315256 0.7385433 2.7569280 4.5466773 4.5279899 7.1798980 6.0244134 6.5403141 4.2321633 4.6238608

903 904 905 906 907 908 909 910 911 912 913

4.5526255 4.1829483 4.1164573 3.4370414 3.6768574 3.1178831 3.3033775 1.6111768 2.4350607 4.0634427 4.0614786

914 915 916 917 918 919 920 921 922 923 924

2.1126938 2.5768956 2.8599616 4.2758134 2.2942154 1.9738619 1.9183780 1.2493384 1.0258658 1.3682087 1.0293729

925 926 927 928 929 930 931 932 933 934 935

1.9143003 0.9020913 1.7518290 2.8479793 2.4553107 2.7063784 2.9936444 2.9271867 2.4424500 0.9144472 3.0516807

936 937 938 939 940 941 942 943 944 945 946

2.6161164 2.3637832 2.4040439 2.9662232 3.3283170 1.9211743 3.9749676 2.2290166 2.5313552 1.7593023 1.6714094

947 948 949 950 951 952 953 954 955 956 957

2.3584610 2.7050969 2.0628921 2.1076871 1.6939777 3.2273027 4.0501195 4.3448156 4.0616689 3.1800774 3.1466258

958 959 960 961 962 963 964 965 966 967 968

2.6772099 3.7349419 3.9588743 3.8154914 3.2391007 3.1909393 2.8545012 1.5343054 1.3422498 1.6943910 1.9629337

969 970 971 972 973 974 975 976 977 978 979

2.2610763 2.1392456 1.8490794 1.9036855 1.5943333 2.2653253 8.6986274 0.6626600 2.1018947 2.6127067 2.6590210

980 981 982 983 984 985 986 987 988 989 990

1.9975574 2.8719777 2.8861049 1.3623057 1.4829526 1.9778966 2.0760816 1.2789830 1.4853431 2.1734034 7.2379305

991 992 993 994 995 996 997 998 999 1000

1.2911862 1.2203184 1.2600239 3.3114333 3.4828414 3.1704676 2.9185778 3.7700937 4.6425487 1.9920711

[ reached getOption("max.print") -- omitted 250 entries ]

> qqplot(qchisq((1:nrow(X) - 1/2) / nrow(X), df = 7), (mah))

> abline(a = 0, b = 1, col = "red")

> shapiro.test(qnorm(pchisq(mah, 7))) # Kiểm tra tính chuẩn nhiều chiều

Shapiro-Wilk normality test

data: qnorm(pchisq(mah, 7))

W = 0.92943, p-value < 2.2e-16

> ## d. Có sự khác biệt giữa lợi nhuận trung bình một ngày trước và ba ngày trước ngày đang xét không?

> # Tính lợi nhuận trung bình một ngày trước và ba ngày trước

> mean\_Lag1 <- mean(data$Lag1)

> mean\_Lag3 <- mean(data$Lag3)

> # Hiển thị kết quả

> mean\_Lag1

[1] 0.0038344

> mean\_Lag3

[1] 0.001716

> # Kiểm tra sự khác biệt

> difference <- mean\_Lag1 - mean\_Lag3

> difference

[1] 0.0021184

> # Lấy lợi nhuận tại ngày đang xét

> profit\_today <- data$Today

> # Kiểm định Wilcoxon để xem có sự khác biệt giữa lợi nhuận trung bình tại ngày đang xét và các ngày trước đó có khác 0 không

> wilcox\_test\_result <- wilcox.test(profit\_today, mu = 0)

> # Hiển thị kết quả

> wilcox\_test\_result

Wilcoxon signed rank test with continuity correction

data: profit\_today

V = 392038, p-value = 0.8924

alternative hypothesis: true location is not equal to 0

> ## BT: H0: Lợi nhuận trung bình tại ngày đang xét và các ngày trước đó giữa năm 2004 và 2005 không khác biệt => EX2004 = EX2005

> ## H1: Có sự khác biệt về lợi nhuận trung bình tại ngày đang xét và các ngày trước đó giữa năm 2004 và 2005 => EX2004 != EX2005

> # Tạo hai nhóm dữ liệu cho năm 2004 và 2005

> profit\_2004 <- data$Today[data$Year == 2004]

> profit\_2005 <- data$Today[data$Year == 2005]

> # Kiểm định Wilcoxon để xem có sự khác biệt giữa năm 2004 và 2005 về lợi nhuận trung bình tại ngày đang xét và các ngày trước đó không

> wilcox\_test\_result\_2004\_2005 <- wilcox.test(profit\_2004, profit\_2005)

> # Hiển thị kết quả

> wilcox\_test\_result\_2004\_2005

Wilcoxon rank sum test with continuity correction

data: profit\_2004 and profit\_2005

W = 32365, p-value = 0.7079

alternative hypothesis: true location shift is not equal to 0

>

> ## Với p-value = 0.7079 > 0.05 chưa đủ cơ sở bác bỏ H0.

> ## Với p-value = 0.7079 > 0.05 chưa đủ cơ sở bác bỏ H0.

> ## Vậy không có sự khác biệt đáng kể giữa lợi nhuận trung bình tại ngày đang xét và các ngày trước đó giữa năm 2004 và 2005.