Take-Home Exam Spring 2020

DUE DATE: Friday 11th, June 2021 Before mid-night.

I pledge to the Honor Code and to obey all rules for taking and performing Take-home exam as specified by the course instructor.

Your Name: Janyaporn J.

Your Student ID: 6104640435

There are Three questions.

Instruction:

- You have to submit (1) The R codes (YourID_Exam.R), (2) Your report with the empirical analysis.

- If you have any question related to this exam, please send me the e-mail at Wasin@econ.tu.ac.th with the Topic: EE435: Take-home exam questions.

Question1. (20 Marks)

Consider four components of U.S. monthly industrial production index from January 1947 -April 2021. The four components are durable consumer goods (IPD- CONGD), nondurable consumer goods (IPNCONGD), business equivalent (IPBUSEQ), and materials (IPMAT). The original data are available from the Federal Reserve Bank of St. Louis and are seasonally adjusted. Note that IPMAT starts at January 1939. You can download these data set by using the below codes:

Toolbox 1

```
getSymbols("IPDCONGD",src="FRED")
dim(IPDCONGD)
tail(IPDCONGD)
getSymbols("IPNCONGD",src="FRED")
dim(IPNCONGD)
getSymbols("IPBUSEQ",src="FRED")
dim(IPBUSEQ)
getSymbols("IPMAT",src="FRED")
```

```
dim(IPMAT)
IP = cbind(as.numeric(IPDCONGD),as.numeric(IPNCONGD),as.numeric(IPBUSEQ),as.numeric(IPMAT[-c
    (1:96)]))
dim(IP)
colnames(IP) <- c("IPD","IPN","IPB","IPM")
```

√1.1 Construct the growth rate series $\mathbf{z}_t$ of the four industrial production index, i.e. take the first difference of the log data. Obtain time plots of $\mathbf{z}_t$. Comment on the time plot.

√1.2 Build a VAR model for $\mathbf{z}_t$, including simplification (or refine) the model by removing insignificant parameters with threshold of t-ratio 1.645 and perform model checking. Write down the final fitted model.

√1.3 Obtain the plots of impulse response function of the fitted model. Interpret the plots.

√1.4 Obtain the forecast error variance decomposition. Interpret the results.

√1.5 Compute 1-step to 6-step ahead predictions of $\mathbf{z}_t$ at the forecast origin h = 879 (April 2020). Obtain 95 % interval forecasts for each component series.

Question2. (20 Marks)

Consider the price of the following cryptocurrencies.

Instruction:

If your last student id number is the odd number, you have to use the series of Ethereum from January 01, 2015 to May 24, 2021.

If your last student id number is the even number, you have to use the series of Dogecoin from January 01, 2015 to May 24, 2021.

The data are available from Yahoo Finance.

2.1 propose the research paper to identify the optimal model for forecasting the log returns and its volatilities.

Remaked: You may start from explaning the basic statistics of the prices and log returns of the cryptocurrencies. The score will be considered from the completness of the report.

UK = 21-146
CA = 77-202
US = 133 - 258

Question3 (20 marks)

1 2 3

Consider the quarterly real GDP of United Kingdom, Canada, and the United States from the first quarter of 1980 to the second quarter of 2011. The data are available from the Federal Reserve Bank of St. Louis (FRED). Let $\mathbf{z}_t$ be the log GDP series. You can learn how to download the data from the Federal Reserve Bank of St. Louis .

3.1 Build a VAR model for $\mathbf{z}_t$, including simplification (or refine) the model by removing insignificant parameters with threshold of t-ratio 1.645 and perform model checking. Write down the final fitted model.

3.2 Obtain the plots of impulse response function of the fitted model. Interpret the plots.

3.3 Obtain the forecast error variance decomposition. Interpret the results.

3.4 Propose your co-integrating vector, and check it by using the Engle-Granger cointegration test.

3.5 Build an ECM-VAR model (aka. VECM). Write down the fitted model and interpret the result.

TAKEHOME-Q1.R

mew

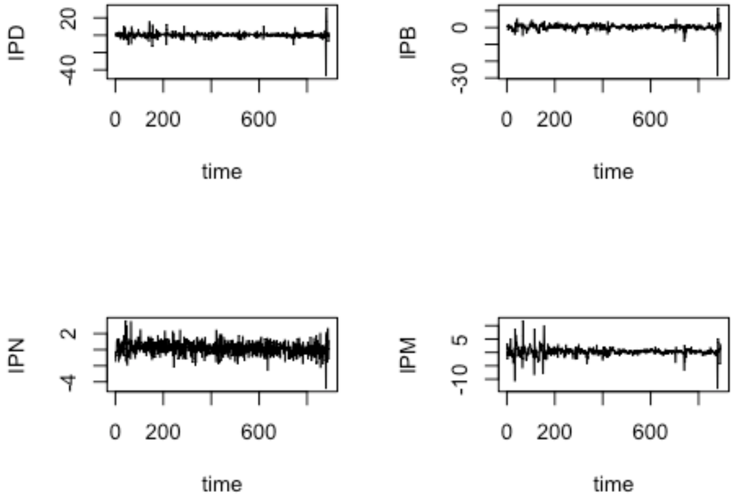
2021-06-07

```
#EE435
setwd("/Users/mew/Desktop")
cat(rep("\n",50))  #clear R Console
#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)
#1
getSymbols("IPDCONGD",src = "FRED")
## '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] "IPDCONGD"
dim(IPDCONGD)
## [1] 892   1
tail(IPDCONGD)
##            IPDCONGD
## 2020-11-01 104.8970
## 2020-12-01 105.3158
## 2021-01-01 107.6347
## 2021-02-01 100.0964
## 2021-03-01 102.5666
## 2021-04-01 100.7979
getSymbols("IPNCONGD",src = "FRED")
## [1] "IPNCONGD"
dim(IPNCONGD)
## [1] 892   1
getSymbols("IPBUSEQ",src = "FRED")
## [1] "IPBUSEQ"
dim(IPBUSEQ)
## [1] 892   1
getSymbols("IPMAT",src = "FRED")
## [1] "IPMAT"
dim(IPMAT)
```

```
## [1] 988   1
IP=cbind(as.numeric(IPDCONGD),as.numeric(IPNCONGD),as.numeric(IPBUSEQ),as.numeric(IPMAT[-c(1:96)]))
dim(IP)
## [1] 892   4
colnames(IP) <- c("IPD","IPN","IPB","IPM")
#1.1
require(MTS)
## Loading required package: MTS
##
## Attaching package: 'MTS'
## The following object is masked from 'package:TTR':
##
##     VMA
x=log(IP)
zt=diff(x)*100
MTSplot(zt)
```

From the graph, nondurable goods (IPN) has the largest fluctation since it is non-durable not long-lasting, it can cause excess demand and excess supply in different period of time. while durable goods are less volatile. Moreover, there are huge decline growth for all data in the lastest period due to Covid-19 situation. And IPB is mean clustering




```
#1.2
VARorder(zt)
## selected order: aic =  6
## selected order: bic =  2
## selected order: hq =  3
## Summary table:
##        p    AIC    BIC     HQ     M(p) p-value
##  [1,]  0 2.8153 2.8153 2.8153   0.0000  0.0000
##  [2,]  1 2.6470 2.7330 2.6798 178.2331  0.0000
##  [3,]  2 2.5125 2.6846 2.5783 147.9623  0.0000
```

```
## [4,]  3 2.4570 2.7151 2.5556  79.0713  0.0000
## [5,]  4 2.4563 2.8005 2.5879  31.4646  0.0117
## [6,]  5 2.4045 2.8348 2.5689  75.1412  0.0000
## [7,]  6 2.3832 2.8995 2.5805  48.7671  0.0000
## [8,]  7 2.3984 3.0008 2.6287  17.5535  0.3507
## [9,]  8 2.4168 3.1053 2.6799  14.8136  0.5383
## [10,]  9 2.4224 3.1970 2.7184  25.4492  0.0623
## [11,] 10 2.4268 3.2874 2.7557  26.3997  0.0487
## [12,] 11 2.4327 3.3793 2.7945  24.9845  0.0701
## [13,] 12 2.4258 3.4585 2.8205  35.4558  0.0034
## [14,] 13 2.4266 3.5454 2.8542  28.9289  0.0244
m1=VAR(zt,2)
```
↖ from BIC

```
## Constant term:
## Estimates:  0.1724828 0.1811122 0.1757223 0.1275675
## Std.Error:  0.1133758 0.02803105 0.05863395 0.05433982
## AR coefficient matrix
## AR( 1 )-matrix
##         [,1]    [,2]     [,3]   [,4]
## [1,] 0.11876  0.2193 -0.10721 0.3384
## [2,] 0.02955 -0.1678  0.00916 0.0162
## [3,] 0.00339  0.0539  0.12272 0.1800
## [4,] 0.01713  0.1435  0.02771 0.2112
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0473 0.1467 0.0928 0.0821
## [2,] 0.0117 0.0363 0.0230 0.0203
## [3,] 0.0245 0.0759 0.0480 0.0425
## [4,] 0.0227 0.0703 0.0445 0.0394
## AR( 2 )-matrix
##           [,1]    [,2]     [,3]     [,4]
## [1,] -0.157177  0.1887 -0.19773  0.10530
## [2,]  0.000419 -0.0449 -0.00132  0.00082
## [3,] -0.142158  0.0250  0.18277  0.10314
## [4,]  0.010632  0.1338  0.02122 -0.06006
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0471 0.1461 0.0924 0.0825
## [2,] 0.0116 0.0361 0.0229 0.0204
## [3,] 0.0243 0.0756 0.0478 0.0427
## [4,] 0.0226 0.0700 0.0443 0.0395
##
## Residuals cov-mtx:
##            [,1]      [,2]      [,3]      [,4]
## [1,] 10.0294942 0.8338041 3.6930427 2.3401480
## [2,]  0.8338041 0.6130805 0.4406453 0.3140822
## [3,]  3.6930427 0.4406453 2.6824829 1.2567987
## [4,]  2.3401480 0.3140822 1.2567987 2.3039604
##
## det(SSE) =  11.45125
## AIC =  2.509928
## BIC =  2.682044
## HQ  =  2.575709
m2=refVAR(m1,thres = 1.645)
```

```
## Constant term:
## Estimates:  0.2152919 0.1773043 0.1842434 0.124001
## Std.Error:  0.109649 0.02686661 0.05687657 0.05346031
## AR coefficient matrix
## AR( 1 )-matrix
##        [,1]   [,2]  [,3]  [,4]
## [1,] 0.1023  0.000 0.000 0.347
## [2,] 0.0348 -0.155 0.000 0.000
## [3,] 0.0000  0.000 0.134 0.185
## [4,] 0.0000  0.178 0.000 0.232
## standard error
##         [,1]   [,2]   [,3]   [,4]
## [1,] 0.03733 0.0000 0.0000 0.0783
## [2,] 0.00844 0.0351 0.0000 0.0000
## [3,] 0.00000 0.0000 0.0378 0.0413
## [4,] 0.00000 0.0672 0.0000 0.0337
## AR( 2 )-matrix
##        [,1]  [,2]   [,3]  [,4]
## [1,] -0.129 0.000 -0.175 0.000
## [2,]  0.000 0.000  0.000 0.000
## [3,] -0.139 0.000  0.182 0.102
## [4,]  0.000 0.144  0.000 0.000
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0454 0.0000 0.0860 0.0000
## [2,] 0.0000 0.0000 0.0000 0.0000
## [3,] 0.0238 0.0000 0.0464 0.0425
## [4,] 0.0000 0.0654 0.0000 0.0000
##
## Residuals cov-mtx:
##           [,1]     [,2]     [,3]     [,4]
## [1,] 10.096231 0.827661 3.701401 2.322786
## [2,]  0.827661 0.614838 0.439733 0.313845
## [3,]  3.701401 0.439733 2.684327 1.256237
## [4,]  2.322786 0.313845 1.256237 2.315673
##
## det(SSE) =  11.75217
## AIC =  2.495463
## BIC =  2.570764
## HQ  =  2.524243
```

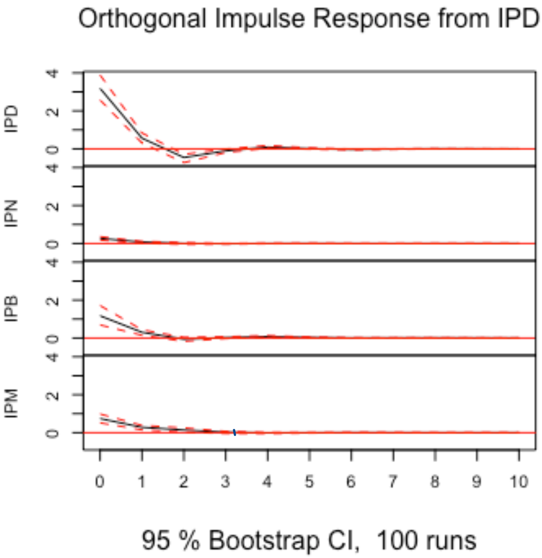
$$\begin{bmatrix} \widehat{IPD}_t \\ \widehat{IPN}_t \\ \widehat{IPB}_t \\ \widehat{IPM}_t \end{bmatrix} = \begin{bmatrix} \substack{0.2153 \\ (0.11)} \\ \substack{0.1773 \\ (0.027)} \\ \substack{0.1842 \\ (0.057)} \\ \substack{0.124 \\ (0.053)} \end{bmatrix} + \begin{bmatrix} \substack{0.1023 \\ (0.373)} & \substack{0 \\ (0)} & \substack{0 \\ (0)} & \substack{0.347 \\ (0.0783)} \\ \substack{0.0348 \\ (0.0084)} & \substack{-0.155 \\ (0.0351)} & \substack{0 \\ (0)} & \substack{0 \\ (0)} \\ \substack{0 \\ (0)} & \substack{0 \\ (0)} & \substack{0.134 \\ (0.038)} & \substack{0.185 \\ (0.0413)} \\ \substack{0 \\ (0)} & \substack{0.178 \\ (0.0672)} & \substack{0 \\ (0)} & \substack{0.232 \\ (0.0337)} \end{bmatrix} \begin{bmatrix} IPD_{t-1} \\ IPN_{t-1} \\ IPB_{t-1} \\ IPM_{t-1} \end{bmatrix}$$

$$+ \begin{bmatrix} \substack{-0.129 \\ (0.0454)} & \substack{0 \\ (0)} & \substack{-0.175 \\ (0.086)} & \substack{0 \\ (0)} \\ \substack{0 \\ (0)} & \substack{0 \\ (0)} & \substack{0 \\ (0)} & \substack{0 \\ (0)} \\ \substack{-0.139 \\ (0.0238)} & \substack{0 \\ (0)} & \substack{0.182 \\ (0.0464)} & \substack{0.102 \\ (0.0425)} \\ \substack{0 \\ (0)} & \substack{0.144 \\ (0.0654)} & \substack{0 \\ (0)} & \substack{0 \\ (0)} \end{bmatrix} \begin{bmatrix} IPD_{t-2} \\ IPN_{t-2} \\ IPB_{t-2} \\ IPM_{t-2} \end{bmatrix}$$

$\therefore$ model m_2 has lower AIC and BIC than m_1

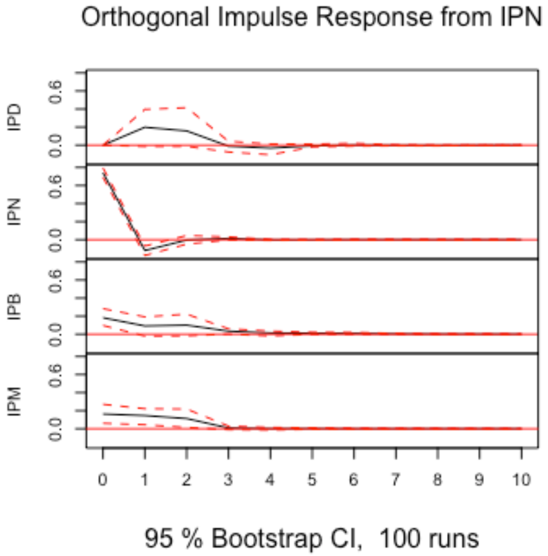
```
#1.3 uninstall MTS
require(vars)
## Loading required package: vars
## Loading required package: MASS
## Loading required package: strucchange
## Loading required package: sandwich
## Loading required package: urca
## Loading required package: lmtest
##
## Attaching package: 'vars'
## The following object is masked from 'package:MTS':
##
##     VAR
```

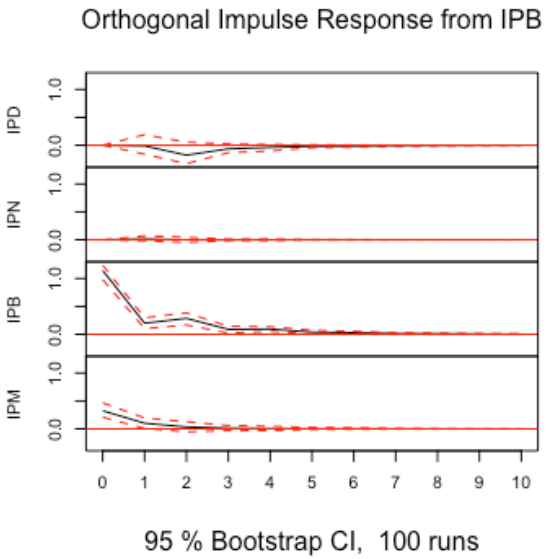
```
varfit=VAR(zt,p=2)
impresp=irf(varfit)
plot(impresp)
```



- shock from IPD has positive impact to all, however, IPD has very silently effect on IPN
- After 3 periods, the shocks will convert to steady state

- shock from IPN has positive impact to all, and the largest shock is shock from itself
- After 4 periods, the shocks will convert to steady state

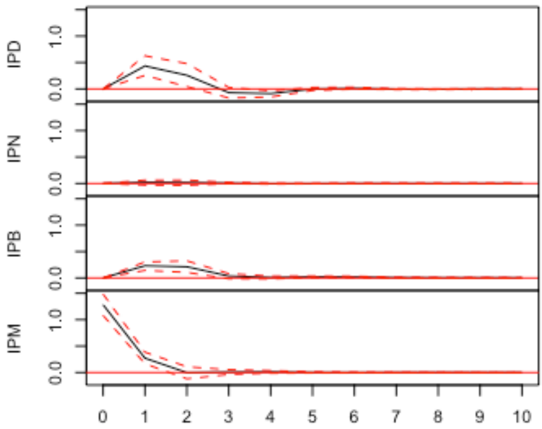




- there is an unclear shock from IPB to IPD and IPN (it can be positive or negative)
- there is a positive impact from IPB to IPM and itself
- After 4 periods, the shocks will convert to steady state

- there is a positive shock from IPM to IPD, IPM, and itself
- IPM has very silently effect on IPN
- After 4 periods, the shocks will convert to steady state

Orthogonal Impulse Response from IPM



95 % Bootstrap CI, 100 runs

#1.4

```
fevd(varfit,n.ahead = 10)
## $IPD
##          IPD         IPN          IPB        IPM
##  [1,] 1.0000000 0.000000000 0.000000e+00 0.00000000
##  [2,] 0.9786677 0.003677477 1.547554e-05 0.01763932
##  [3,] 0.9681556 0.005825149 2.865465e-03 0.02315381
##  [4,] 0.9673749 0.005826039 3.233717e-03 0.02356535
##  [5,] 0.9664951 0.005913986 3.365580e-03 0.02422538
##  [6,] 0.9664566 0.005918078 3.400740e-03 0.02422457
##  [7,] 0.9664239 0.005917334 3.427742e-03 0.02423098
##  [8,] 0.9664153 0.005917546 3.434368e-03 0.02423283
##  [9,] 0.9664095 0.005918007 3.436407e-03 0.02423607
## [10,] 0.9664087 0.005918072 3.437077e-03 0.02423614
##
## $IPN
##          IPD       IPN          IPB          IPM
##  [1,] 0.1130659 0.8869341 0.0000000000 0.0000000000
##  [2,] 0.1176617 0.8812807 0.0003848728 0.0006727082
##  [3,] 0.1176060 0.8808902 0.0003857537 0.0011180226
##  [4,] 0.1178793 0.8805521 0.0003980169 0.0011706388
##  [5,] 0.1178761 0.8805347 0.0003990007 0.0011902249
##  [6,] 0.1178942 0.8805099 0.0003989982 0.0011968518
##  [7,] 0.1178942 0.8805093 0.0003991068 0.0011974314
##  [8,] 0.1178956 0.8805076 0.0003991836 0.0011976850
##  [9,] 0.1178956 0.8805075 0.0003991979 0.0011977477
## [10,] 0.1178956 0.8805074 0.0003991997 0.0011977775
```

In the first period, IPD has pure shock from itself (100%) and 0% from other factors.
For next 10 period, shock from itself reduce from 100% to 96.64%. 0.59% from IPN, 0.344% from IPB, and 2.42% from IPM

In the first period, IPN will be affected from IPD 11.3% and from itself 88.7%. (summation = 100%)
For next 10 period, IPN will be affected from IPD 11.79%, from itself 88%, from IPB 0.4% and from IPM 0.12%.

```
##
## $IPB
##            IPD        IPN       IPB        IPM
##  [1,] 0.5069355 0.01224107 0.4808235 0.00000000
##  [2,] 0.5048449 0.01432290 0.4624710 0.01836117
##  [3,] 0.4828424 0.01696945 0.4679075 0.03228060
##  [4,] 0.4812822 0.01727732 0.4688451 0.03259533
##  [5,] 0.4808006 0.01725670 0.4694796 0.03246315
##  [6,] 0.4806182 0.01727671 0.4695372 0.03256786
##  [7,] 0.4804322 0.01729458 0.4696161 0.03265708
##  [8,] 0.4803961 0.01729859 0.4696392 0.03266607
##  [9,] 0.4803859 0.01729914 0.4696485 0.03266650
## [10,] 0.4803820 0.01729948 0.4696507 0.03266778
```

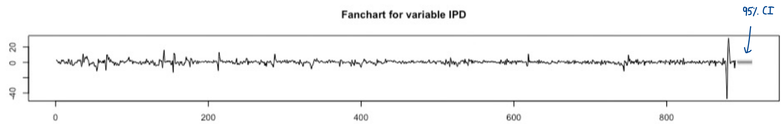
In the first period, IPB will be affected from IPD 50.7/ , from IPN 1.2 %, from itself 48.1% and no effect from IPM
For next 10 period, IPB will be affected from IPD 48%, from IPN 1.7% , from itself 47% and 3.3% from IPM

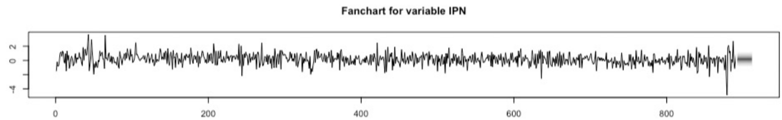
```
##
## $IPM
##            IPD        IPN        IPB       IPM
##  [1,] 0.2369914 0.01140501 0.04501385 0.7065897
##  [2,] 0.2512518 0.01903374 0.04569678 0.6840177
##  [3,] 0.2554440 0.02378324 0.04554842 0.6752243
##  [4,] 0.2555947 0.02379966 0.04558903 0.6750166
##  [5,] 0.2556231 0.02379681 0.04559513 0.6749849
##  [6,] 0.2556249 0.02379765 0.04559845 0.6749790
##  [7,] 0.2556289 0.02379741 0.04560048 0.6749732
##  [8,] 0.2556296 0.02379737 0.04560115 0.6749719
##  [9,] 0.2556295 0.02379740 0.04560134 0.6749718
## [10,] 0.2556295 0.02379741 0.04560140 0.6749717
```

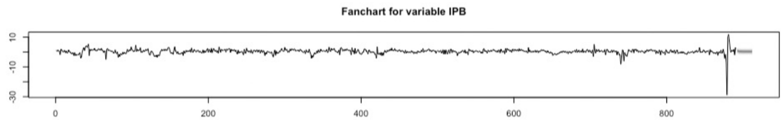
In the first period, IPM will be affected from IPD 23.7%, from IPN 1.1%, from IPB 4.5% and from itself 70.7%.
For next 10 period, IPM will be affected from IPD 25.6%, from IPN 2.4%, from IPB 4.56% and fom itself 67.5%.

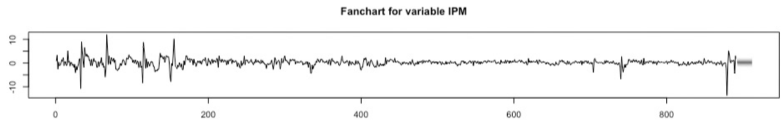
```
#1.5
predict(varfit,6)
## $IPD
##              fcst     lower    upper       CI
##  [1,] -0.06150473 -6.300249 6.177239 6.238744
##  [2,]  0.77907327 -5.624736 7.182883 6.403809
##  [3,]  0.18461230 -6.318302 6.687526 6.502914
##  [4,]  0.05341044 -6.455561 6.562382 6.508971
##  [5,]  0.18430627 -6.330032 6.698645 6.514339
##  [6,]  0.24660374 -6.267897 6.761105 6.514501
##  [7,]  0.22682501 -6.288099 6.741749 6.514924
##  [8,]  0.22019697 -6.294785 6.735179 6.514982
##  [9,]  0.22827211 -6.286733 6.743277 6.515005
## [10,]  0.23291860 -6.282089 6.747927 6.515008
##
## $IPN
##              fcst     lower    upper       CI
##  [1,] -0.05426783 -1.596737 1.488201 1.542469
##  [2,]  0.15209749 -1.415492 1.719687 1.567589
##  [3,]  0.19159580 -1.376365 1.759557 1.567961
##  [4,]  0.15594667 -1.412572 1.724465 1.568519
##  [5,]  0.15469714 -1.413842 1.723237 1.568540
##  [6,]  0.16062462 -1.407939 1.729188 1.568563
##  [7,]  0.16124108 -1.407323 1.729805 1.568564
##  [8,]  0.16017874 -1.408387 1.728744 1.568566
##  [9,]  0.16004112 -1.408525 1.728607 1.568566
## [10,]  0.16033330 -1.408232 1.728899 1.568566
```

```
##
## $IPB
##          fcst     lower    upper       CI
##  [1,] 0.9027450 -2.323714 4.129204 3.226459
##  [2,] 0.7194082 -2.619747 4.058564 3.339155
##  [3,] 0.5411836 -2.877592 3.959959 3.418775
##  [4,] 0.3508809 -3.073996 3.775757 3.424877
##  [5,] 0.3737427 -3.059243 3.806729 3.432986
##  [6,] 0.3594820 -3.075094 3.794058 3.434576
##  [7,] 0.3408222 -3.094427 3.776072 3.435250
##  [8,] 0.3263874 -3.109014 3.761789 3.435402
##  [9,] 0.3243662 -3.111117 3.759850 3.435484
## [10,] 0.3223762 -3.113132 3.757885 3.435508
##
## $IPM
##          fcst     lower    upper       CI
##  [1,] 0.3852687 -2.604896 3.375434 2.990165
##  [2,] 0.3011154 -2.805045 3.407276 3.106160
##  [3,] 0.2343799 -2.891949 3.360708 3.126329
##  [4,] 0.2485357 -2.878334 3.375405 3.126869
##  [5,] 0.2380792 -2.888995 3.365153 3.127074
##  [6,] 0.2275148 -2.899574 3.354604 3.127089
##  [7,] 0.2291433 -2.897961 3.356248 3.127104
##  [8,] 0.2305070 -2.896601 3.357615 3.127108
##  [9,] 0.2295074 -2.897601 3.356616 3.127109
## [10,] 0.2287581 -2.898351 3.355867 3.127109
```









TAKEHOME-Q2.R

mew

2021-06-09

```
#2
setwd("/Users/mew/Desktop")
cat(rep("\n",50))  #clear R Console
#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(fGarch)
getSymbols("ETH-USD",from="2015-01-01",to="2021-05-24")
```

→ download data from yahoo finance

ETH - USD = Ethereum

```
## '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.
## Warning: ETH-USD contains missing values. Some functions will not work if
## objects contain missing values in the middle of the series. Consider using
## na.omit(), na.approx(), na.fill(), etc to remove or replace them.
## [1] "ETH-USD"
head(`ETH-USD`)
##            ETH-USD.Open ETH-USD.High ETH-USD.Low ETH-USD.Close ETH-USD.Volume
## 2015-08-07     2.831620     3.536610    2.521120      2.772120         164329
## 2015-08-08     2.793760     2.798810    0.714725      0.753325         674188
## 2015-08-09     0.706136     0.879810    0.629191      0.701897         532170
## 2015-08-10     0.713989     0.729854    0.636546      0.708448         405283
## 2015-08-11     0.708087     1.131410    0.663235      1.067860        1463100
## 2015-08-12     1.058750     1.289940    0.883608      1.217440        2150620
##            ETH-USD.Adjusted
## 2015-08-07         2.772120
## 2015-08-08         0.753325
## 2015-08-09         0.701897
## 2015-08-10         0.708448
## 2015-08-11         1.067860
## 2015-08-12         1.217440
tail(`ETH-USD`)
##            ETH-USD.Open ETH-USD.High ETH-USD.Low ETH-USD.Close ETH-USD.Volume
## 2021-05-19     3382.657     3437.936    1952.460      2460.679    84482912776
```

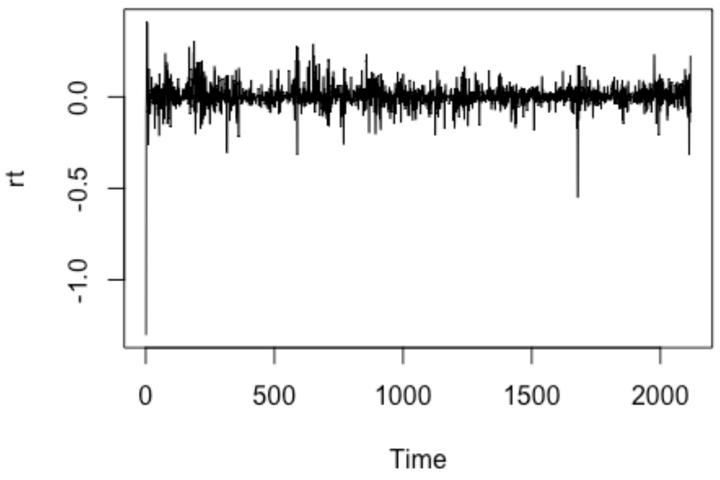
```
## 2021-05-20     2439.639     2993.145    2170.229     2784.294    67610826680
## 2021-05-21     2772.341     2938.205    2113.347     2430.621    53774070802
## 2021-05-22     2436.015     2483.983    2168.124     2295.706    42089937660
## 2021-05-23     2298.367     2384.412    1737.469     2109.580    56005721977
## 2021-05-24     2099.936     2672.596    2090.640     2643.591    53697121740
##            ETH-USD.Adjusted
## 2021-05-19        2460.679
## 2021-05-20        2784.294
## 2021-05-21        2430.621
## 2021-05-22        2295.706
## 2021-05-23        2109.580
## 2021-05-24        2643.591
dim(`ETH-USD`)
## [1] 2118    6
```

calculate log return

```
rt=diff(log(as.numeric(`ETH-USD`[,6])))
```

→ make it to be weak stationarity

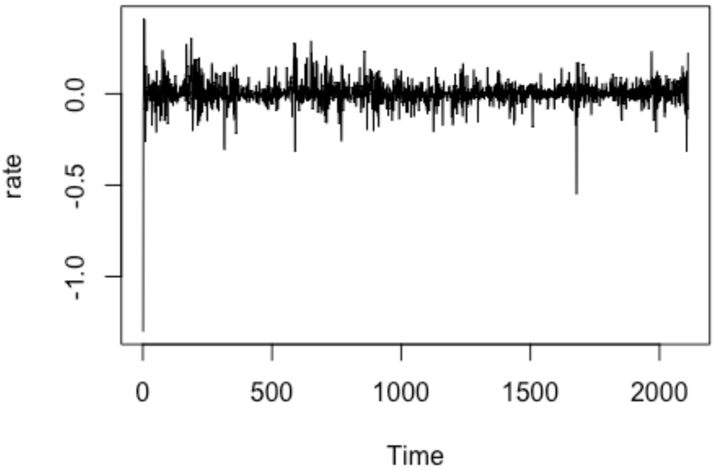
```
ts.plot(rt)
```



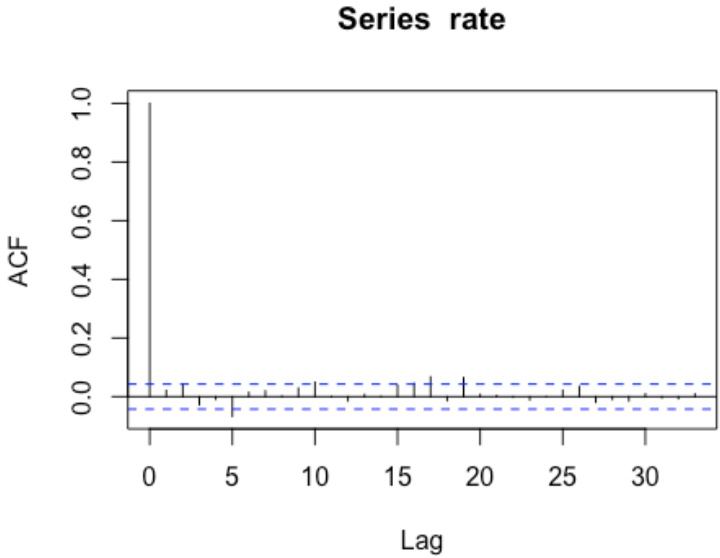

```
rate=na.omit(rt)
```

→ fix missing value

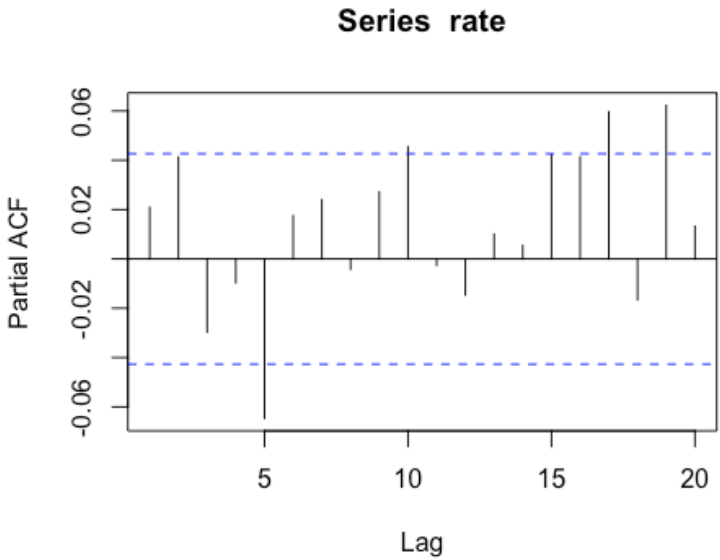
```
ts.plot(rate)
```




```
#
acf(rate)
```




```
pacf(rate , lag.max = 20)
```



```
t.test(rate)
##
##  One Sample t-test
##
## data:  rate
## t = 2.1366, df = 2109, p-value = 0.03274
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.0002613857 0.0061011761
## sample estimates:
##   mean of x
## 0.003181281
```

→ check mean

$H_0: E(r_t) = 0$

$H_1: E(r_t) \neq 0$

since p-value < 0.05, we reject H_0

So, mean of rate is not equal to zero at 95% CI

```
Box.test(rate, lag=10, type = 'Ljung' )
##
##  Box-Ljung test
##
## data:  rate
## X-squared = 24.206, df = 10, p-value = 0.007073
#
acf(rate^2)
```

→ check serial correlation

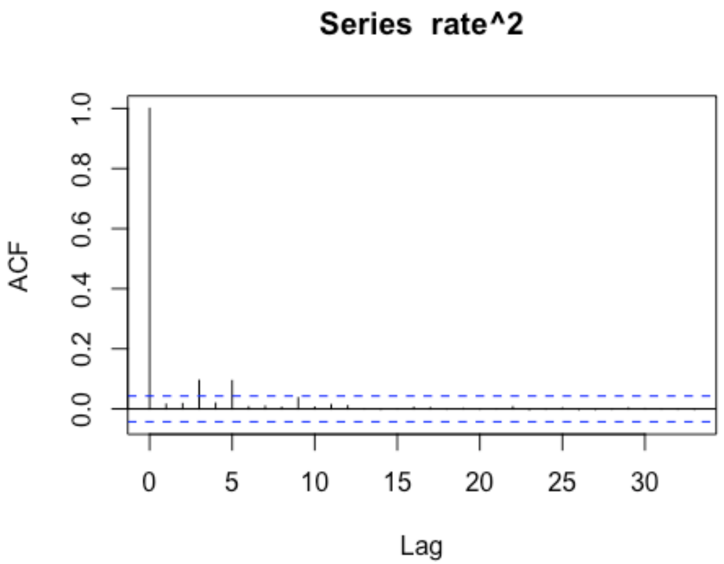
$H_0: \rho_1 = \rho_2 = \dots = 0$

$H_1: \exists \rho_i \neq 0$

since p-value < 0.05, we reject H_0

∴ There is serial correlation at 95% CI

(PACF also support this conclusion)




```
Box.test(rate^2, lag=10, type = 'Ljung' )
##
##  Box-Ljung test
##
## data:  rate^2
## X-squared = 42.837, df = 10, p-value = 5.319e-06
#
```

→ Check ARCH effect

H_0: There is no ARCH effect

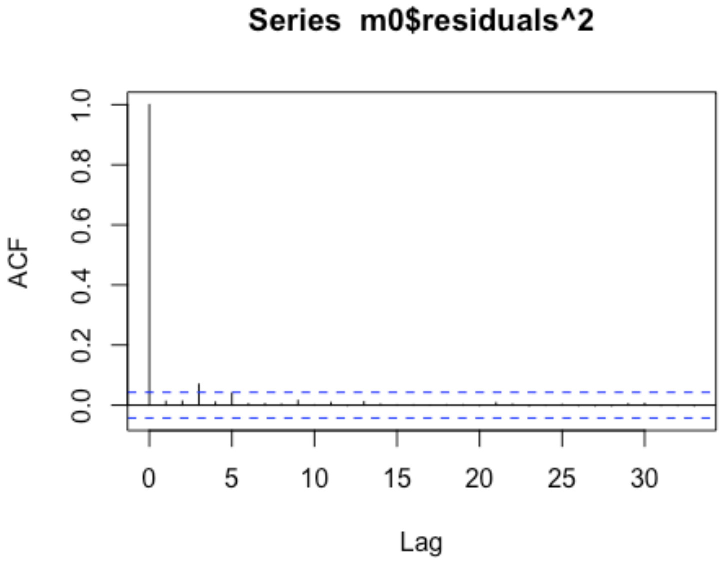
H_1: There is an ARCH effect

since p-value close to zero, there is an ARCH effect at 95% CI

I will select the model between

GARCH(1,1) and ARMA(5,5) + GARCH(1,1)

```
m0=auto.arima(rate)
summary(m0)
## Series: rate
## ARIMA(5,0,5) with non-zero mean
##
## Coefficients:
##          ar1      ar2     ar3      ar4      ar5     ma1     ma2      ma3
##      -0.1825  -0.2961  0.0234  -0.4489  -0.7065  0.2044  0.3502  -0.0420
## s.e.  0.1245   0.0794  0.0954   0.0786   0.1271  0.1312  0.0842   0.1018
##          ma4     ma5    mean
##       0.4456  0.6186  0.0032
## s.e.  0.0803  0.1386  0.0015
##
## sigma^2 estimated as 0.004614:  log likelihood=2685.9
## AIC=-5347.79   AICc=-5347.64   BIC=-5279.94
##
## Training set error measures:
##                         ME       RMSE        MAE      MPE     MAPE      MASE
## Training set -1.441276e-05 0.06774656 0.04212343 78.67127 204.4328 0.6945822
##                     ACF1
## Training set -0.003782915
acf(m0$residuals^2)
```

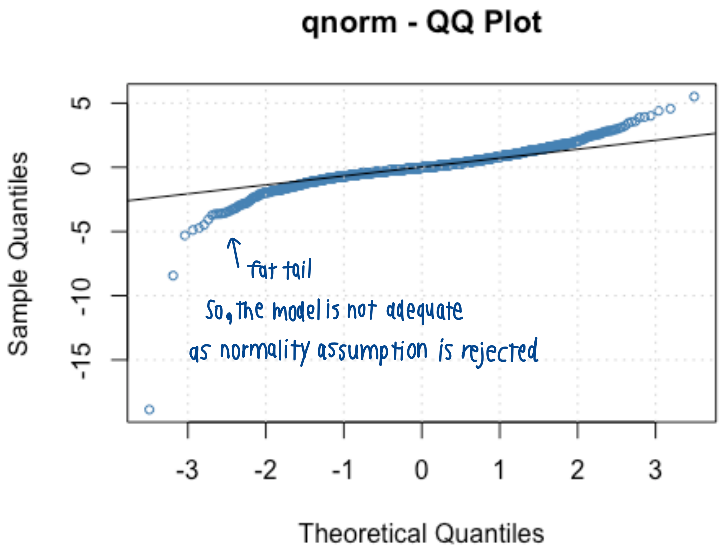



```
Box.test(m0$residuals^2, lag=10, type = 'Ljung' )
##
##  Box-Ljung test
##
```

```
## data:  m0$residuals^2
## X-squared = 15.141, df = 10, p-value = 0.127
predict(m0,5)
## $pred
## Time Series:
## Start = 2111
## End = 2115
## Frequency = 1
## [1]  0.001553993  0.016634581 -0.002056067  0.005118405 -0.030268561
##
## $se
## Time Series:
## Start = 2111
## End = 2115
## Frequency = 1
## [1] 0.06792384 0.06794016 0.06802548 0.06806533 0.06806972
#
m1=garchFit(~garch(1,1), data = rate, trace = FALSE)
```
→ plain vanila model

```
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
summary(m1)
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = rate, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fe0c242dd18>
##  [data = rate]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu       omega      alpha1       beta1
## 0.00181437  0.00030506  0.22581488  0.73102249
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     1.814e-03   1.145e-03    1.584    0.113
## omega  3.051e-04   5.238e-05    5.824 5.74e-09 ***
## alpha1 2.258e-01   3.316e-02    6.810 9.73e-12 ***
## beta1  7.310e-01   3.061e-02   23.879  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
```

```
##  2921.7    normalized:  1.384692
##
## Description:
##  Wed Jun  9 23:59:43 2021 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value          p-value < 0.05, reject H0
##  Jarque-Bera Test   R    Chi^2  235016.8  0             - not normal distribution
##  Shapiro-Wilk Test  R    W      0.843771  0                   for lag 10 and 15
##  Ljung-Box Test     R    Q(10)  18.23277  0.05116175    } since p-value > 0.05, we cannot reject H0
##  Ljung-Box Test     R    Q(15)  22.71407  0.09040232    }   ∴ mean equation is adequate
##  Ljung-Box Test     R    Q(20)  37.32252  0.01070214    - not adequate
##  Ljung-Box Test     R^2  Q(10)  0.1873262 0.9999999     } since p-value > 0.05, we cannot reject H0
##  Ljung-Box Test     R^2  Q(15)  0.2821231 1             }   ∴ volatility equation is adequate
##  Ljung-Box Test     R^2  Q(20)  0.3662152 1             }
##  LM Arch Test       R    TR^2   8.72732   0.7260276     no arch effect at 95% C.I.
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -2.765593 -2.754873 -2.765600 -2.761667
predict(m1,5)
##   meanForecast meanError standardDeviation
## 1  0.001814371 0.1487592         0.1487592
## 2  0.001814371 0.1465578         0.1465578
## 3  0.001814371 0.1444200         0.1444200
## 4  0.001814371 0.1423445         0.1423445
## 5  0.001814371 0.1403298         0.1403298
plot(m1, which = 13)
```



standardized student-t

```
#
m2=garchFit(~garch(1,1), data = rate, cond.dist = "std", trace = FALSE)
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
summary(m2)
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = rate, cond.dist = "std",
##     trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fe0c21f4d18>
##  [data = rate]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         mu       omega      alpha1       beta1       shape
## 0.00091922  0.00027738  0.28888895  0.74732147  2.93594522
##
## Std. Errors:
##  based on Hessian
```

mean eq : $\hat{r}_t = 0.0009192$
(0.0008166)

$a_t = \sigma_t \varepsilon_t$, $\varepsilon_t \overset{iid}{\sim} t_{2.936}$

volatility eq: $\sigma_t^2 = \underset{(0.0000723)}{0.0002774} + \underset{(0.056)}{0.289} a_{t-1}^2 + \underset{(0.03)}{0.747} \sigma_{t-1}^2$

```
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     0.0009192   0.0008166    1.126 0.260301
## omega  0.0002774   0.0000723    3.837 0.000125 ***
## alpha1 0.2888889   0.0563659    5.125 2.97e-07 ***
## beta1  0.7473215   0.0303087   24.657  < 2e-16 ***
## shape  2.9359452   0.2251239   13.041  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3290.666    normalized:  1.559557
##
## Description:
##  Wed Jun  9 23:59:43 2021 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  330248.5  0
##  Shapiro-Wilk Test  R    W      0.8305545 0
##  Ljung-Box Test     R    Q(10)  17.83585  0.05779657
##  Ljung-Box Test     R    Q(15)  22.21607  0.1022622
##  Ljung-Box Test     R    Q(20)  35.66518  0.01683461
##  Ljung-Box Test     R^2  Q(10)  0.1809104 1
##  Ljung-Box Test     R^2  Q(15)  0.2710454 1
##  Ljung-Box Test     R^2  Q(20)  0.3454955 1
##  LM Arch Test       R    TR^2   9.82396   0.6314013
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.114375 -3.100976 -3.114387 -3.109469
predict(m2,5)
##   meanForecast meanError standardDeviation
## 1 0.0009192249 0.1708079         0.1708079
## 2 0.0009192249 0.1746687         0.1746687
## 3 0.0009192249 0.1785813         0.1785813
## 4 0.0009192249 0.1825472         0.1825472
## 5 0.0009192249 0.1865677         0.1865677
plot(m2, which = 13)
```

p-value < 0.05, reject H_0
– not normal distribution (Jarque-Bera, Shapiro-Wilk)

for lag 10 and 15
since p-value > 0.05, we cannot reject H_0
∴ mean equation is adequate

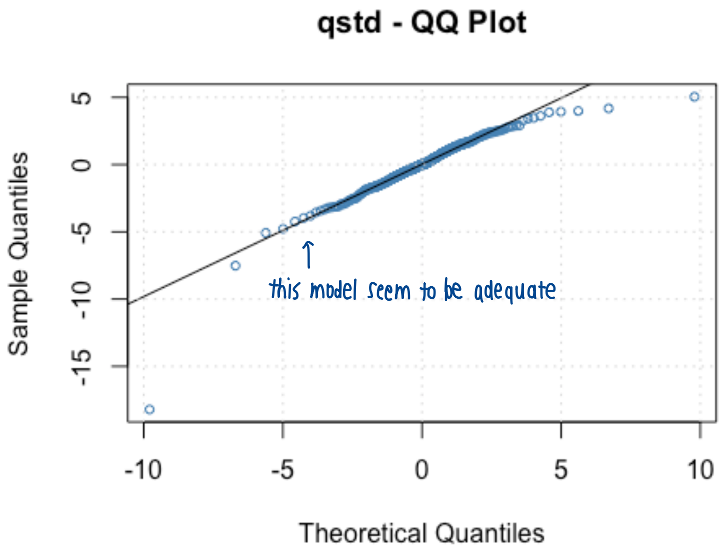
– not adequate (Q(20))

since p-value > 0.05, we cannot reject H_0
∴ volatility equation is adequate

no arch effect at 95% CI.

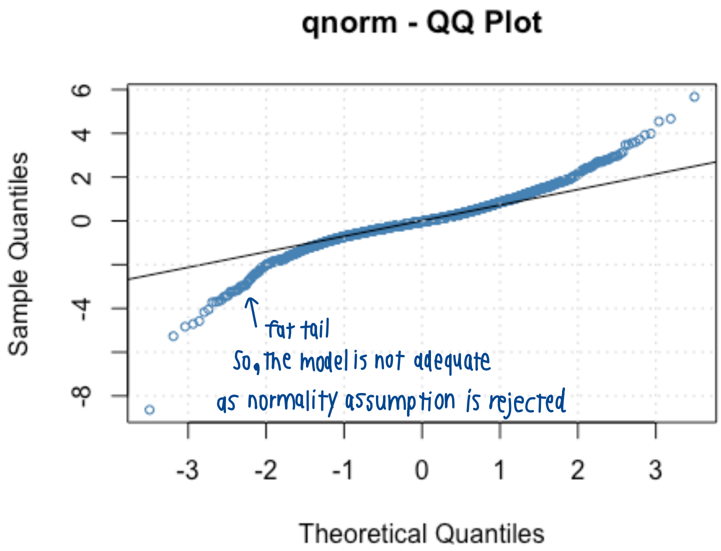
lowest (AIC, BIC)

From all models (m1, m2, m3), I prefer m2 since this model has lowest AIC and BIC (-3.11, -3.1)
and mean equation (for lag 10 and 15) and volatility equation are adequate
Also, when we plot QQ plot, the normality assumption is quite accepted




```
#
m3=garchFit(~arma(2,2)+garch(1,1), data = rate, trace = FALSE)
## Warning: Using formula(x) is deprecated when x is a character vector of length > 1.
##   Consider formula(paste(x, collapse = " ")) instead.
summary(m3)
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(2, 2) + garch(1, 1), data = rate, trace = FALSE)
##
## Mean and Variance Equation:
##  data ~ arma(2, 2) + garch(1, 1)
## <environment: 0x7fe0c12c5508>
##  [data = rate]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1          ar2          ma1          ma2        omega
##  0.00049040   0.97263538  -0.18963496  -0.95202725   0.19803471   0.00026091
##      alpha1        beta1
##  0.17598351   0.76892698
##
## Std. Errors:
```

```
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu      4.904e-04   3.469e-04    1.414    0.157
## ar1     9.726e-01   1.014e-01    9.589  < 2e-16 ***
## ar2    -1.896e-01   2.392e-02   -7.928 2.22e-15 ***
## ma1    -9.520e-01   1.035e-01   -9.202  < 2e-16 ***
## ma2     1.980e-01   3.392e-02    5.838 5.27e-09 ***
## omega   2.609e-04   4.896e-05    5.329 9.87e-08 ***
## alpha1  1.760e-01   2.275e-02    7.735 1.04e-14 ***
## beta1   7.689e-01   2.764e-02   27.815  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3103.401    normalized:  1.470806
##
## Description:
##  Wed Jun  9 23:59:44 2021 by user:
##
##
## Standardised Residuals Tests:                         p-value < 0.05, reject H0
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  2919.98   0 - not normal distribution
##  Shapiro-Wilk Test  R    W      0.9395041 0          for lag 10 and 15
##  Ljung-Box Test     R    Q(10)  13.78415  0.183067   } since p-value > 0.05, we cannot reject H0
##  Ljung-Box Test     R    Q(15)  19.16574  0.2063214  }   ∴ mean equation is adequate
##  Ljung-Box Test     R    Q(20)  34.48549  0.02302208 - not adequate
##  Ljung-Box Test     R^2  Q(10)  8.06864   0.6221323  } since p-value > 0.05, we cannot reject H0
##  Ljung-Box Test     R^2  Q(15)  10.3484   0.7972712  }   ∴ volatility equation is adequate
##  Ljung-Box Test     R^2  Q(20)  13.24189  0.8667661  }
##  LM Arch Test       R    TR^2   8.414405  0.7519658  → no arch effect at 95% CI.
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -2.934029 -2.912590 -2.934058 -2.926179
predict(m3,5)
## Warning in a_vec[i] <- ar[1:min(u2, i - 1)] * a_vec[(i - 1):(i - u2)] + : number
## of items to replace is not a multiple of replacement length
## Warning in a_vec[i] <- ar[1:min(u2, i - 1)] * a_vec[(i - 1):(i - u2)] + : number
## of items to replace is not a multiple of replacement length
##    meanForecast meanError standardDeviation
## 1 -0.0003402842 0.1399466         0.1399466
## 2  0.0032882424 0.1370232         0.1369928
## 3  0.0037531905 0.1342307         0.1341420
## 4  0.0035173200 0.1315352         0.1313914
## 5  0.0031997336 0.1289341         0.1287383
plot(m3, which = 13)
```


qnorm - QQ Plot
Sample Quantiles
6
4
2
0
-4
-8
-3
-2
-1
0
1
2
3
Theoretical Quantiles
fat tail
So, the model is not adequate
as normality assumption is rejected

TAKEHOME-Q3.R

mew

2021-06-09

```
#EE435
setwd("/Users/mew/Desktop")
cat(rep("\n",50))  #clear R Console
#3
library(fGarch)
## Loading required package: timeDate
## Loading required package: timeSeries
## Loading required package: fBasics
library(quantmod)
## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following object is masked from 'package:timeSeries':
##
##     time<-
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
## Loading required package: TTR
##
## Attaching package: 'TTR'
## The following object is masked from 'package:fBasics':
##
##     volatility
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
library(fUnitRoots)
require(vars)
## Loading required package: vars
## Loading required package: MASS
## Loading required package: strucchange
## Loading required package: sandwich
## Loading required package: urca
##
## Attaching package: 'urca'
## The following objects are masked from 'package:fUnitRoots':
##
##     punitroot, qunitroot, unitrootTable
## Loading required package: lmtest
getSymbols("CLVMNACSCAB1GQUK",src = "FRED")
```

```
## '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] "CLVMNACSCAB1GQUK"
getSymbols("NAEXKP01CAQ189S",src = "FRED")
## [1] "NAEXKP01CAQ189S"
getSymbols("GDPC1",src = "FRED")
## [1] "GDPC1"
uk=CLVMNACSCAB1GQUK[21:146]
ca=NAEXKP01CAQ189S[77:202]
us=GDPC1[133:258]
da=cbind(as.numeric(uk),as.numeric(ca),as.numeric(us))
dim(da)
## [1] 126   3
x=log(da)
zt=diff(x)*100
colnames(zt)<-c("uk","ca","us")
#3.1
varfit=VAR(zt,p=2)
summary(varfit)
##
## VAR Estimation Results:
## =========================
## Endogenous variables: uk, ca, us
## Deterministic variables: const
## Sample size: 123
## Log Likelihood: -301.231
## Roots of the characteristic polynomial:
## 0.6857 0.3994 0.3785 0.2519 0.2519 0.1096
## Call:
## VAR(y = zt, p = 2)
##
##
## Estimation results for equation uk:
## ===================================
## uk = uk.l1 + ca.l1 + us.l1 + uk.l2 + ca.l2 + us.l2 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## uk.l1  0.439964   0.092395   4.762 5.59e-06 ***
## ca.l1 -0.021313   0.100053  -0.213   0.8317
## us.l1  0.103185   0.096564   1.069   0.2875
## uk.l2  0.097239   0.095563   1.018   0.3110
## ca.l2  0.078684   0.086610   0.908   0.3655
## us.l2 -0.007348   0.098589  -0.075   0.9407
## const  0.170522   0.080520   2.118   0.0363 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
##
## Residual standard error: 0.58 on 116 degrees of freedom
## Multiple R-Squared: 0.3422,  Adjusted R-squared: 0.3082
## F-statistic: 10.06 on 6 and 116 DF,  p-value: 6.207e-09
##
##
## Estimation results for equation ca:
## ===================================
## ca = uk.l1 + ca.l1 + us.l1 + uk.l2 + ca.l2 + us.l2 + const
##
##       Estimate Std. Error t value Pr(>|t|)
## uk.l1  0.43162    0.08925   4.836 4.11e-06 ***
## ca.l1  0.31596    0.09665   3.269  0.00142 **
## us.l1  0.47460    0.09328   5.088 1.41e-06 ***
## uk.l2 -0.19750    0.09231  -2.139  0.03449 *
## ca.l2 -0.10933    0.08366  -1.307  0.19388
## us.l2 -0.05804    0.09523  -0.609  0.54342
## const  0.05626    0.07778   0.723  0.47091
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.5603 on 116 degrees of freedom
## Multiple R-Squared: 0.5303,  Adjusted R-squared: 0.506
## F-statistic: 21.83 on 6 and 116 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation us:
## ===================================
## us = uk.l1 + ca.l1 + us.l1 + uk.l2 + ca.l2 + us.l2 + const
##
##       Estimate Std. Error t value Pr(>|t|)
## uk.l1  0.44572    0.09648   4.620    1e-05 ***
## ca.l1  0.21841    0.10448   2.090 0.038760 *
## us.l1  0.21377    0.10084   2.120 0.036143 *
## uk.l2 -0.20892    0.09979  -2.094 0.038477 *
## ca.l2 -0.08623    0.09044  -0.953 0.342379
## us.l2  0.04290    0.10295   0.417 0.677661
## const  0.30042    0.08408   3.573 0.000515 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.6057 on 116 degrees of freedom
## Multiple R-Squared: 0.3327,  Adjusted R-squared: 0.2982
## F-statistic:  9.64 on 6 and 116 DF,  p-value: 1.353e-08
##
##
##
## Covariance matrix of residuals:
##         uk      ca      us
## uk 0.33645 0.01376 0.05948
## ca 0.01376 0.31394 0.13031
```

```
## us 0.05948 0.13031 0.36688
##
## Correlation matrix of residuals:
##         uk      ca      us
## uk 1.00000 0.04235 0.1693
## ca 0.04235 1.00000 0.3840
## us 0.16929 0.38396 1.0000
m0=restrict(varfit,thres = 1.645)
```

refinement

```
summary(m0)
##
## VAR Estimation Results:
## =========================
## Endogenous variables: uk, ca, us
## Deterministic variables: const
## Sample size: 123
## Log Likelihood: -304.623
## Roots of the characteristic polynomial:
## 0.6261 0.4882 0.189 0.01898     0     0
```

} less than 1 = OK

```
## Call:
## VAR(y = zt, p = 2)
##
##
## Estimation results for equation uk:
## ====================================
## uk = uk.l1 + us.l1 + const
```

$$\underset{(0.076)}{\hat{UK}_t = 0.193} + \underset{(0.08)}{0.4944\,UK_{t-1}} + \underset{(0.078)}{0.146\,US_{t-1}}$$

```
##
##       Estimate Std. Error t value Pr(>|t|)
## uk.l1  0.49439    0.08153   6.064 1.58e-08 ***
## us.l1  0.14633    0.07837   1.867   0.0643 .
## const  0.19302    0.07624   2.532   0.0126 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.5758 on 120 degrees of freedom
## Multiple R-Squared: 0.6055,  Adjusted R-squared: 0.5956
## F-statistic: 61.39 on 3 and 120 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation ca:
## ====================================
## ca = uk.l1 + ca.l1 + us.l1 + uk.l2
```

$$\hat{CA}_t = \underset{(0.08527)}{0.242\,CA_{t-1}} + \underset{(0.086)}{0.422\,UK_{t-1}} - \underset{(0.091)}{0.213\,UK_{t-2}} + \underset{(0.0886)}{0.475\,US_{t-1}}$$

```
##
##       Estimate Std. Error t value Pr(>|t|)
## uk.l1  0.42204    0.08608   4.903 3.02e-06 ***
## ca.l1  0.24215    0.08527   2.840  0.00531 **
## us.l1  0.47472    0.08858   5.360 4.15e-07 ***
## uk.l2 -0.21270    0.09108  -2.335  0.02120 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.5612 on 119 degrees of freedom
```

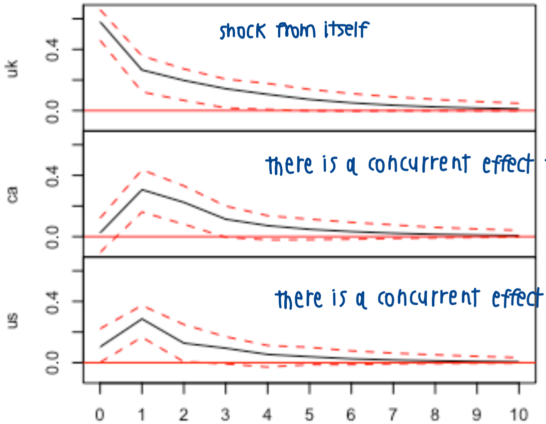
```
## Multiple R-Squared: 0.6968,  Adjusted R-squared: 0.6867
## F-statistic: 68.38 on 4 and 119 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation us:
## =====================================
## us = uk.l1 + ca.l1 + us.l1 + uk.l2 + const
```

$$\hat{US}_t = \underset{(0.08)}{0.294} + \underset{(0.1)}{0.2077} US_{t-1} + \underset{(0.092)}{0.201} CA_{t-1} + \underset{(0.0955)}{0.44} UK_{t-1} - \underset{(0.098)}{0.2077} UK_{t-2}$$

```
##
##       Estimate Std. Error t value Pr(>|t|)
## uk.l1  0.44128    0.09552   4.620 9.87e-06 ***
## ca.l1  0.20128    0.09186   2.191 0.030405 *
## us.l1  0.20774    0.10017   2.074 0.040266 *
## uk.l2 -0.20774    0.09811  -2.117 0.036324 *
## const  0.29404    0.08041   3.657 0.000383 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.6029 on 118 degrees of freedom
## Multiple R-Squared: 0.6543,  Adjusted R-squared: 0.6397
## F-statistic: 44.67 on 5 and 118 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##          uk       ca      us
## uk 0.342999 0.009193 0.05688
## ca 0.009193 0.322952 0.13385
## us 0.056882 0.133849 0.36976
##
## Correlation matrix of residuals:
##         uk      ca     us
## uk 1.00000 0.02762 0.1597
## ca 0.02762 1.00000 0.3873
## us 0.15973 0.38734 1.0000
#3.2
impresp=irf(varfit)
plot(impresp)
```

- there is a positive shock from UK to US and itself
- there is an unclear shock from UK to CA at first period
- After 7 periods, the shocks will convert to steady state

Orthogonal Impulse Response from uk



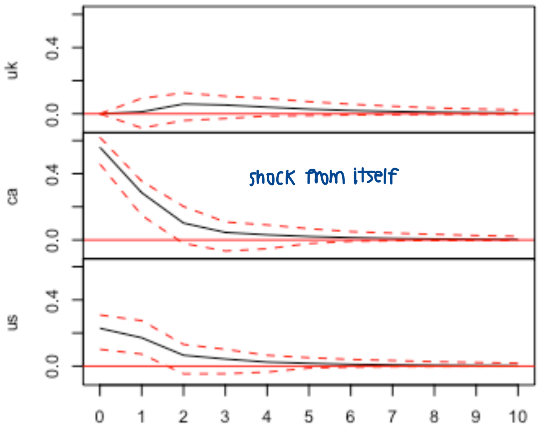
shock from itself

there is a concurrent effect from UK to CA

there is a concurrent effect from UK to US

95 % Bootstrap CI, 100 runs

Orthogonal Impulse Response from ca



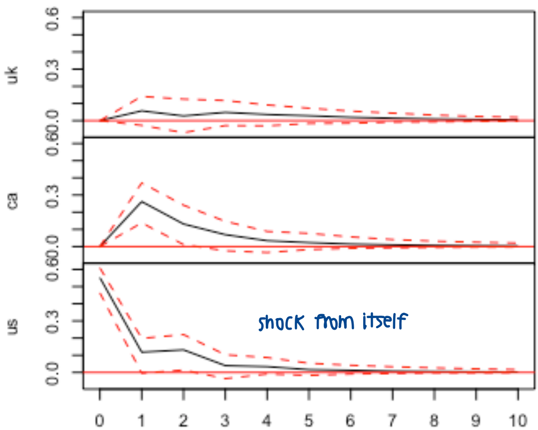
shock from itself

95 % Bootstrap CI, 100 runs

- CA has very silently effect on UK
- There are unclear shock from CA to all (it can be positive or negative)
- After 6 periods, the shocks will convert to steady state

- There are unclear shock from US to all (It can be positive or negative)
- US also has very silently effect on UK (US and CA are near to each other while UK is far)
- After 6 periods, the shocks will convert to steady state

Orthogonal Impulse Response from us



95 % Bootstrap CI, 100 runs

#3.3

```
fevd(varfit,n.ahead = 10)
## $uk
##            uk           ca          us
##  [1,] 1.0000000 0.0000000000 0.000000000
##  [2,] 0.9917741 0.0003302867 0.007895577
##  [3,] 0.9831839 0.0079945859 0.008821524
##  [4,] 0.9733932 0.0134131371 0.013193615
##  [5,] 0.9683015 0.0162768099 0.015421697
##  [6,] 0.9655791 0.0176056384 0.016815226
##  [7,] 0.9642757 0.0182591342 0.017465161
##  [8,] 0.9636307 0.0185718331 0.017797499
##  [9,] 0.9633234 0.0187217557 0.017954804
## [10,] 0.9631764 0.0187926457 0.018030905
##
## $ca
##             uk        ca        us
##  [1,] 0.001793841 0.9982062 0.0000000
##  [2,] 0.169447475 0.7077207 0.1228319
##  [3,] 0.227551425 0.6375417 0.1349068
##  [4,] 0.240448850 0.6214460 0.1381052
##  [5,] 0.245568387 0.6160935 0.1383381
##  [6,] 0.247803380 0.6137494 0.1384472
##  [7,] 0.248885923 0.6126648 0.1384493
##  [8,] 0.249381671 0.6121682 0.1384502
##  [9,] 0.249614023 0.6119390 0.1384470
## [10,] 0.249722155 0.6118324 0.1384454
```

In the first period, UK has pure shock from itself(100%) and 0% from other factors.
For next 10 period, shock from itself reduce from 100% to 96.3% and 1.9% from CA and 1.8% from US

In the first period, CA has largest shock from Itself which is 99.8% and 0.2% from UK (no shock from US)
For next 10 period, CA will be affected from UK 25%, from US 13.8% and from itself 61.2%

```
##
## $us
##               uk        ca        us
##  [1,] 0.02865948 0.1422264 0.8291141
##  [2,] 0.18733842 0.1656778 0.6469838
##  [3,] 0.20462496 0.1619768 0.6333983
##  [4,] 0.21641962 0.1616916 0.6218888
##  [5,] 0.21987718 0.1614144 0.6187084
##  [6,] 0.22179850 0.1613191 0.6168825
##  [7,] 0.22259101 0.1612593 0.6161497
##  [8,] 0.22298040 0.1612321 0.6157875
##  [9,] 0.22315563 0.1612186 0.6156258
## [10,] 0.22323907 0.1612123 0.6155486
```

In the first period, US has largest shock from itself which is 82.9%, from UK which is 2.9% and from CA 14.2%. For next 10 period, US will be affected from UK 22.3%, from CA 16.1% and from itself 61.6%

```
#3.4
Zt=as.data.frame(zt)
a=Zt$uk
b=Zt$ca
c=Zt$us
```

data already get first diff → already I(1)

```
m1=adfTest(a, lags = 2, type = c("ct"), title = NULL, description = NULL)
## Warning in adfTest(a, lags = 2, type = c("ct"), title = NULL, description =
## NULL): p-value smaller than printed p-value
m1@test$p.value
##
## 0.01
```

since Pvalue < 0.05, we reject Ho ∴ there is no unit root test (weak stantionarity) a ~ I(1)

```
m2=adfTest(b, lags = 2, type = c("ct"), title = NULL, description = NULL)
## Warning in adfTest(b, lags = 2, type = c("ct"), title = NULL, description =
## NULL): p-value smaller than printed p-value
m2@test$p.value
##
## 0.01
```

since Pvalue < 0.05, we reject Ho ∴ there is no unit root test (weak stantionarity) b ~ I(1)

```
m3=adfTest(c, lags = 2, type = c("ct"), title = NULL, description = NULL)
## Warning in adfTest(c, lags = 2, type = c("ct"), title = NULL, description =
## NULL): p-value smaller than printed p-value
m3@test$p.value
##
## 0.01
```

since Pvalue < 0.05, we reject Ho ∴ there is no unit root test (weak stantionarity) C ~ I(1)

```
fit <- lm(a~b+c)
```

→ linear model

```
summary(fit)
##
## Call:
## lm(formula = a ~ b + c)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.18235 -0.29935  0.00594  0.34910  1.66506
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.23483    0.07919   2.966 0.003635 **
## b            0.12092    0.09260   1.306 0.194086
## c            0.36821    0.09661   3.811 0.000218 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.6495 on 122 degrees of freedom
## Multiple R-squared:  0.2258, Adjusted R-squared:  0.2131
## F-statistic: 17.79 on 2 and 122 DF,  p-value: 1.661e-07
error <- residuals(fit)
m4=adfTest(error, lags = 2, type = c("nc"), title = NULL, description = NULL)
## Warning in adfTest(error, lags = 2, type = c("nc"), title = NULL, description =
## NULL): p-value smaller than printed p-value
m4@test$p.value
##
## 0.01
```

since Pvalue < 0.05, we reject H_0

so, $I(0)$

after check ADF of a, b, c → $I(1)$

then check $CI(a,b)$ by OLS → error

check ADF

∴ $I(0)$ so, all data are $CI(1,1)$

the models are adequate

```
#3.5
a.L.1=lag(a,k=1)
b.L.1=lag(b,k=1)
c.L.1=lag(c,k=1)
error.L.1=lag(error,k=1)
VECM <- lm(a~a.L.1+b.L.1+c.L.1+error.L.1)
summary(VECM)
## Warning in summary.lm(VECM): essentially perfect fit: summary may be unreliable
##
## Call:
## lm(formula = a ~ a.L.1 + b.L.1 + c.L.1 + error.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.526e-16 -6.710e-17 -3.890e-17 -1.390e-17  5.475e-15
##
## Coefficients: (1 not defined because of singularities)
##               Estimate Std. Error    t value Pr(>|t|)
## (Intercept)  3.972e-16  6.386e-17  6.220e+00 7.37e-09 ***
## a.L.1        1.000e+00  7.051e-17  1.418e+16  < 2e-16 ***
## b.L.1       -3.375e-17  7.263e-17 -4.650e-01    0.643
## c.L.1       -1.036e-17  7.960e-17 -1.300e-01    0.897
## error.L.1           NA         NA         NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.059e-16 on 121 degrees of freedom
## Multiple R-squared:      1,  Adjusted R-squared:      1
## F-statistic: 8.659e+31 on 3 and 121 DF,  p-value: < 2.2e-16
```

$$\Delta \hat{UK}_t = \underset{(6.4e-17)}{3.97e-16} + \underset{(7.05e-17)}{\Delta UK_{t-1}} - \underset{(7.26e-17)}{(3.375e-17)\Delta CA_{t-1}} - \underset{(7.96e-17)}{(1.036e-17)\Delta US_{t-1}} + \phi$$

Both CA and US from last period have negative impact on UK with a very little amount since UK is highly depends on itself and donot depends on other countries since UK is not a member of EUROPE anymore (independent from anyone)

And, we have no error term