6004640477_Exam.R

Goatchasorn

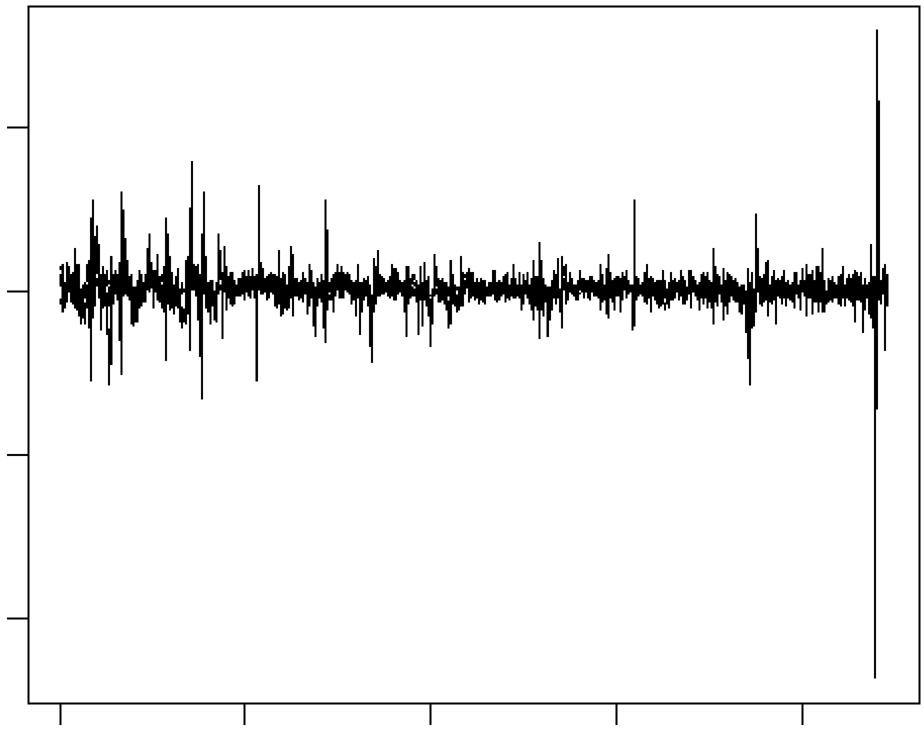
2021-06-10

```
setwd("/Users/Goatchasorn/Documents/BE/4TH YEAR 2ND SEMESTER/EE435/Take home
exam")
cat(rep("\n",50))

library(quantmod)

## Warning: package 'quantmod' was built under R version 4.0.5

## 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)

## Warning: package 'fBasics' was built under R version 4.0.4

## Loading required package: timeDate

## Warning: package 'timeDate' was built under R version 4.0.4

## Loading required package: timeSeries

## Warning: package 'timeSeries' was built under R version 4.0.4

##
## Attaching package: 'timeSeries'

## The following object is masked from 'package:zoo':
##
##     time<-
```

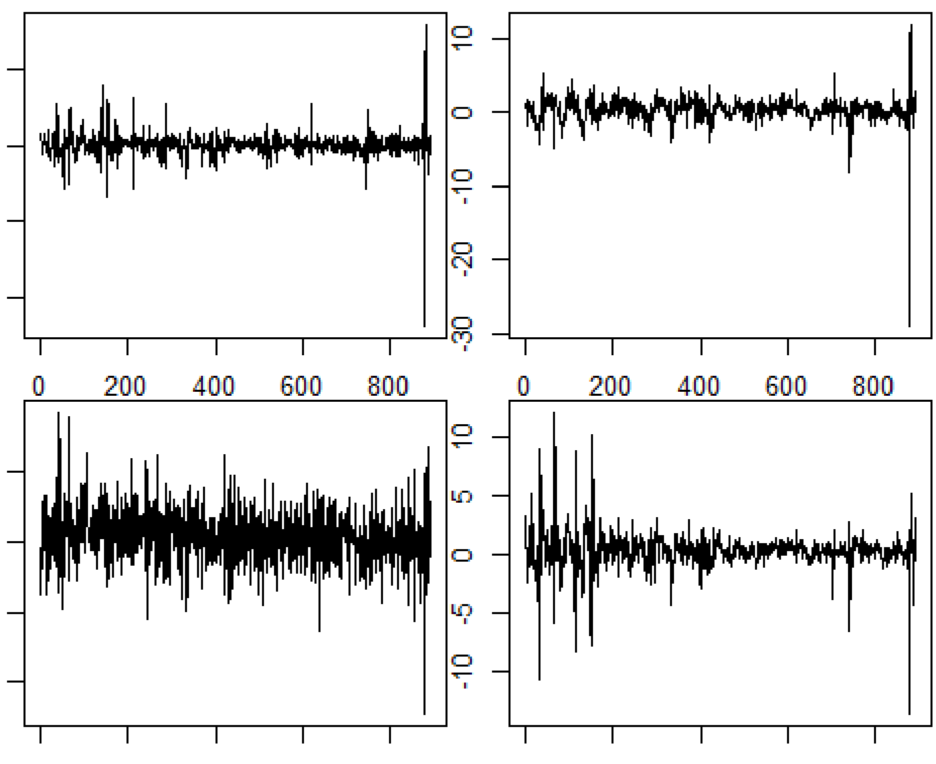
```
##
## Attaching package: 'fBasics'

## The following object is masked from 'package:TTR':
##
##     volatility

library(sn)

## Warning: package 'sn' was built under R version 4.0.4

## Loading required package: stats4

##
## Attaching package: 'sn'

## The following objects are masked from 'package:fBasics':
##
##     tr, vech

## The following object is masked from 'package:stats':
##
##     sd

library(PerformanceAnalytics)

## Warning: package 'PerformanceAnalytics' was built under R version 4.0.4

##
## Attaching package: 'PerformanceAnalytics'

## The following objects are masked from 'package:timeDate':
##
##     kurtosis, skewness

## The following object is masked from 'package:graphics':
##
##     legend

library(car)

## Warning: package 'car' was built under R version 4.0.4

## Loading required package: carData

##
## Attaching package: 'car'

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

library(tseries)
```

```
## Warning: package 'tseries' was built under R version 4.0.4

library(forecast)

## Warning: package 'forecast' was built under R version 4.0.4

library(fGarch)

## Warning: package 'fGarch' was built under R version 4.0.5

library(fUnitRoots)

## Warning: package 'fUnitRoots' was built under R version 4.0.4

library(fGarch)
library(readxl)
#install.packages("Quandl")
library(Quandl)

## Warning: package 'Quandl' was built under R version 4.0.5

#install.packages('MTS')
require(MTS)

## Loading required package: MTS

## Warning: package 'MTS' was built under R version 4.0.5

##
## Attaching package: 'MTS'

## The following object is masked from 'package:TTR':
##
##     VMA

QUESTION 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

#1.1
IP = cbind(as.numeric(IPDCONGD),as.numeric(IPNCONGD),as.numeric(IPBUSEQ),as.n
umeric(IPMAT[-c(1:96)]))
dim(IP)

## [1] 892   4

colnames(IP)=c("IPD","IPN","IPB","IPM")

zt=diffM(log(IP))*100

par(mar=c(1,1,1,1))
ts.plot(zt)
```



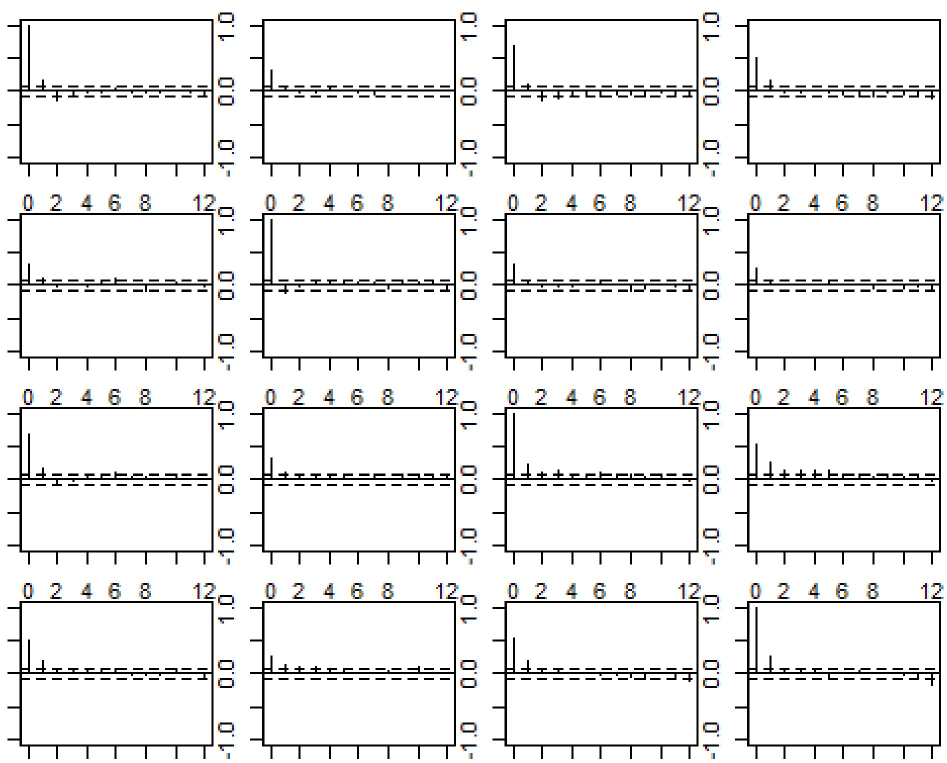
```
MTSplot(zt)
```



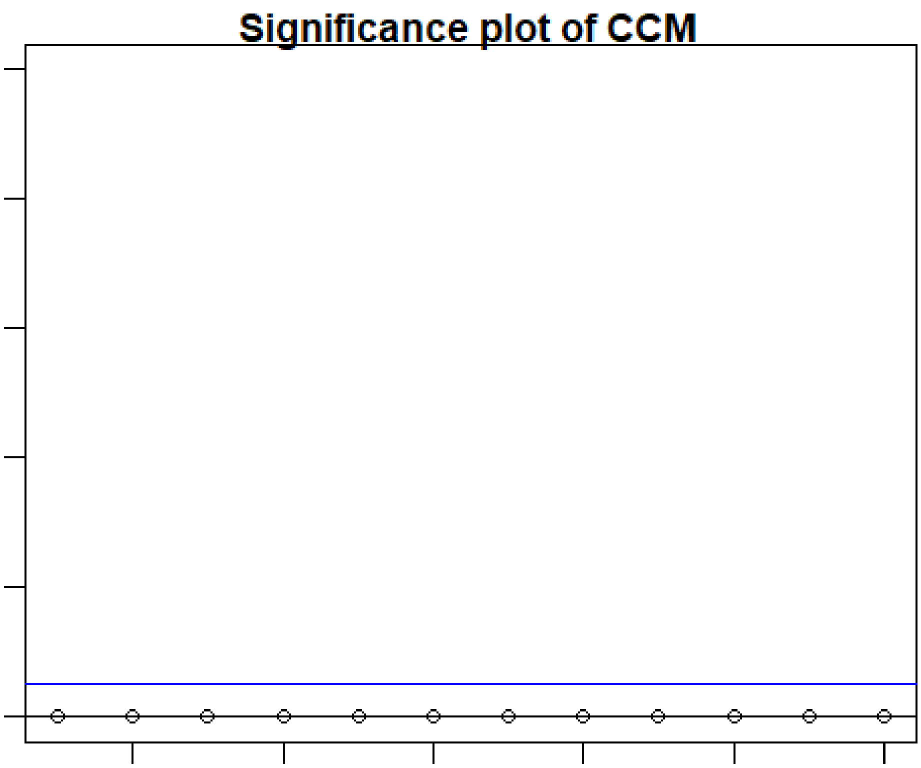

```
ccm(zt)
```

```
## [1] "Covariance matrix:"
##        IPD   IPN  IPB   IPM
## IPD 10.938 0.858 4.01 2.588
## IPN  0.858 0.637 0.46 0.321
## IPB  4.012 0.460 3.04 1.449
## IPM  2.588 0.321 1.45 2.528
## CCM at lag:  0
##       [,1]  [,2]  [,3]  [,4]
## [1,] 1.000 0.325 0.696 0.492
## [2,] 0.325 1.000 0.331 0.253
## [3,] 0.696 0.331 1.000 0.523
## [4,] 0.492 0.253 0.523 1.000
## Simplified matrix:
## CCM at lag:  1
## + + + +
## + - . .
## + + + +
## + + + +
## CCM at lag:  2
## - . - .
## . . . .
## . . + +
## + + + .
## CCM at lag:  3
## - . - .
## . . . .
## . . + +
## . + . .
## CCM at lag:  4
## . . . .
## . . . .
## . + + +
## . + . .
## CCM at lag:  5
## . . - .
## . . . +
## . . + +
## . . . .
## CCM at lag:  6
## . . . .
## + . + .
## + . + +
## . . . .
## CCM at lag:  7
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  8
## . . . .
```

```
## - . - .
## . . . .
## . . . .
## CCM at lag:  9
## . . - .
## . . . .
## . + . +
## . . - .
## CCM at lag:  10
## . . . .
## . . . .
## + + . .
## + + . .
## CCM at lag:  11
## . . . .
## . + . .
## . . . .
## . . - .
## CCM at lag:  12
## . . - -
## . . . .
## . + . .
## . . - -
```



```
## Hit Enter for p-value plot of individual ccm:
```




```
adf=adf.test(diffM(log(as.numeric(IP))))

## Warning in adf.test(diffM(log(as.numeric(IP)))): p-value smaller than prin
ted p-
## value

adf

##
##  Augmented Dickey-Fuller Test
##
## data:  diffM(log(as.numeric(IP)))
## Dickey-Fuller = -14.776, Lag order = 15, p-value = 0.01
## alternative hypothesis: stationary

#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)
```

```
## 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
```

```
m1a = 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
```

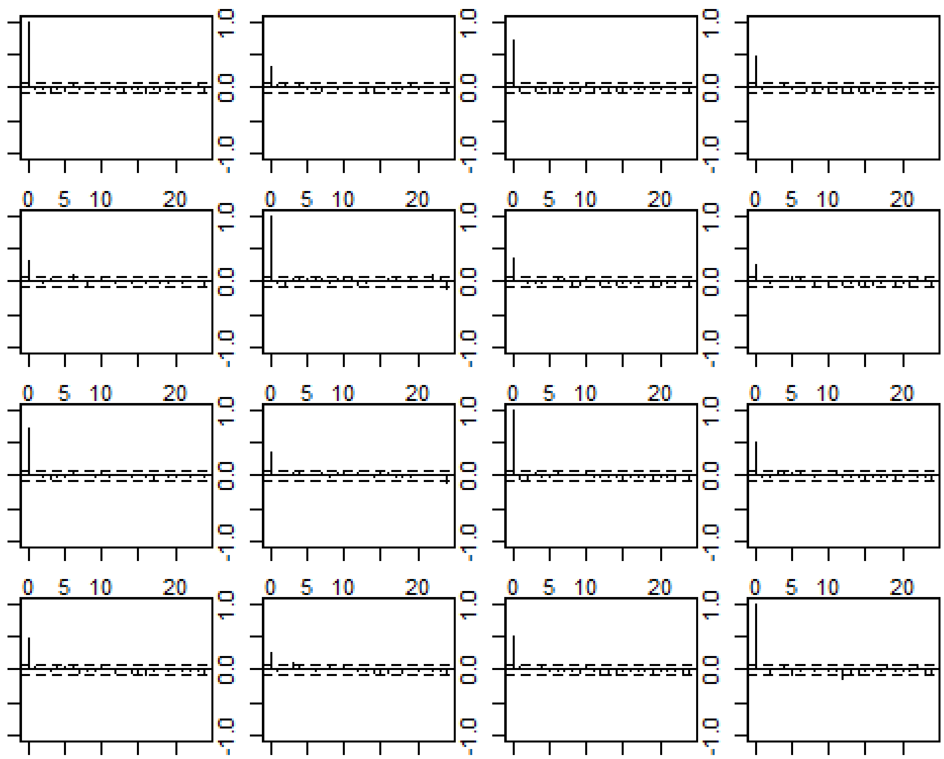
```
par("mar")
```

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

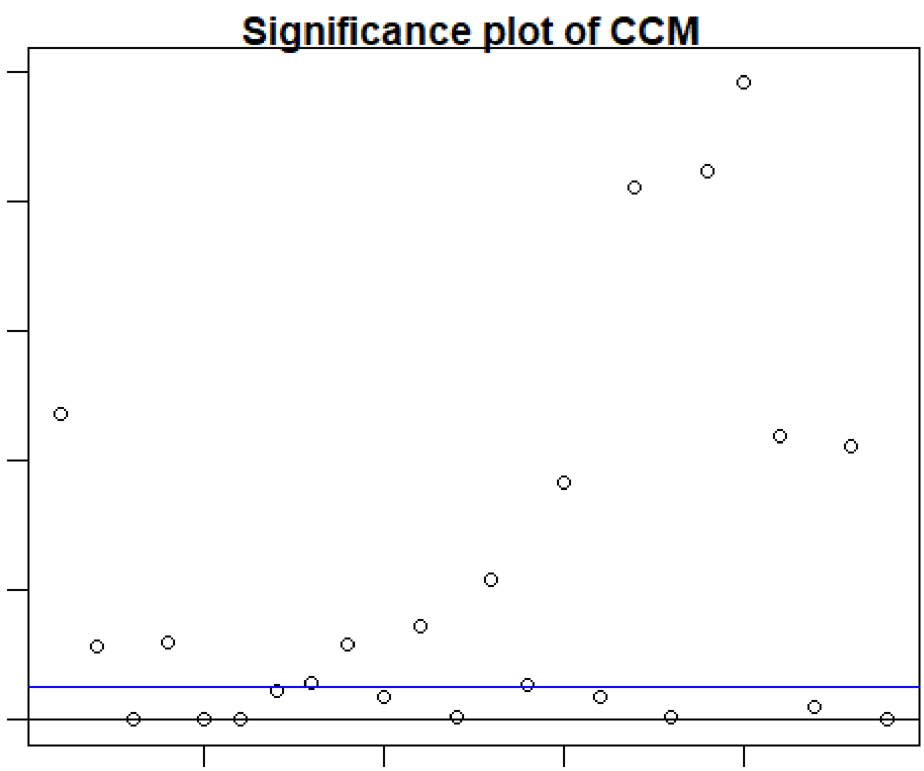
```
par(mar=c(1,1,1,1))
MTSdiag(m1a)

## [1] "Covariance matrix:"
##        IPD   IPN  IPB   IPM
## IPD 10.108 0.829 3.71 2.325
## IPN  0.829 0.616 0.44 0.314
## IPB  3.706 0.440 2.69 1.258
## IPM  2.325 0.314 1.26 2.318
## CCM at lag:  0
##       [,1]  [,2]  [,3]  [,4]
## [1,] 1.000 0.332 0.711 0.480
## [2,] 0.332 1.000 0.342 0.263
## [3,] 0.711 0.342 1.000 0.504
## [4,] 0.480 0.263 0.504 1.000
## Simplified matrix:
## CCM at lag:  1
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  2
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  3
## - . . .
## . . . .
## . . . .
## . + . .
## CCM at lag:  4
## . . . .
## . + . .
## . . . +
## + . . +
## CCM at lag:  5
## . . - .
## . . . +
## . . . .
## . . . -
## CCM at lag:  6
## . . . .
## + . + .
## + . + .
## . . . .
## CCM at lag:  7
## . . . .
## . . . .
## . . . .
```

```
## . . . .
## CCM at lag:  8
## . . . .
## - . - .
## . . . .
## . . . .
## CCM at lag:  9
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  10
## . . . .
## . . . .
## . . . .
## + + + .
## CCM at lag:  11
## . . . .
## . + . .
## . . . .
## . . . .
## CCM at lag:  12
## . . . -
## . . . .
## . . . .
## . . - -
## CCM at lag:  13
## . . . .
## . . . .
## . . . .
## . . . -
## CCM at lag:  14
## . - . .
## . . . .
## . . . .
## . . - -
## CCM at lag:  15
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  16
## - . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  17
## . . . .
## . . . .
## . . . .
```

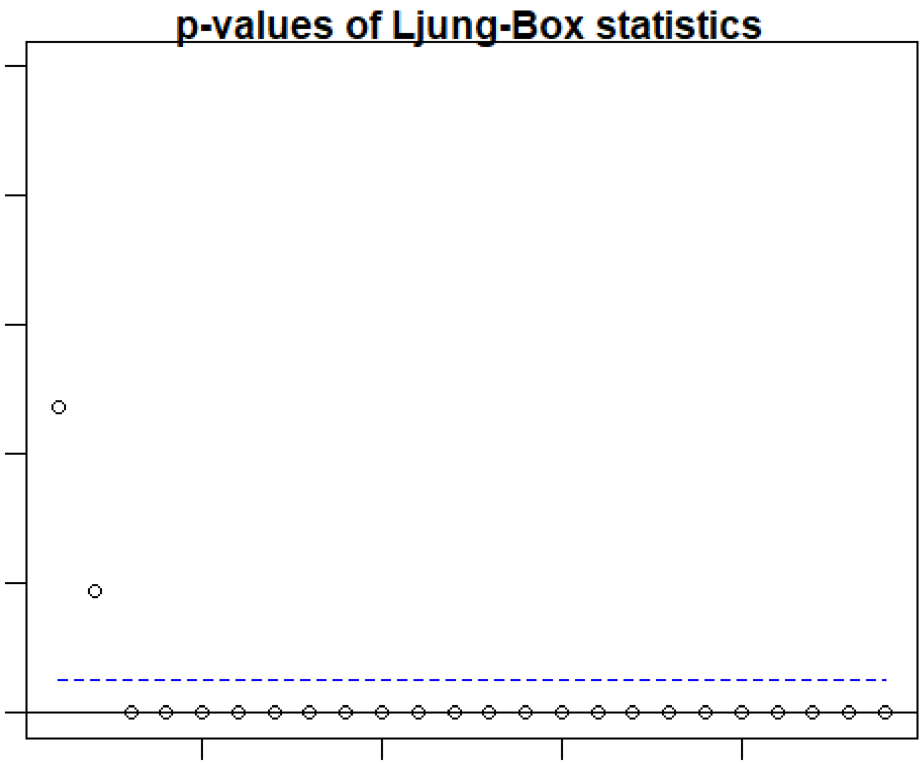
```
## . . . .
## CCM at lag:  18
## . . . .
## . . . .
## . . . .
## . . . +
## CCM at lag:  19
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  20
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  21
## . . . .
## . . . .
## . . . .
## . . . .
## CCM at lag:  22
## . . . .
## . + . .
## . . . .
## . . . .
## CCM at lag:  23
## . . . .
## . + . .
## . . . .
## . . . .
## CCM at lag:  24
## . - . .
## . - . .
## . - - .
## . . . -
```



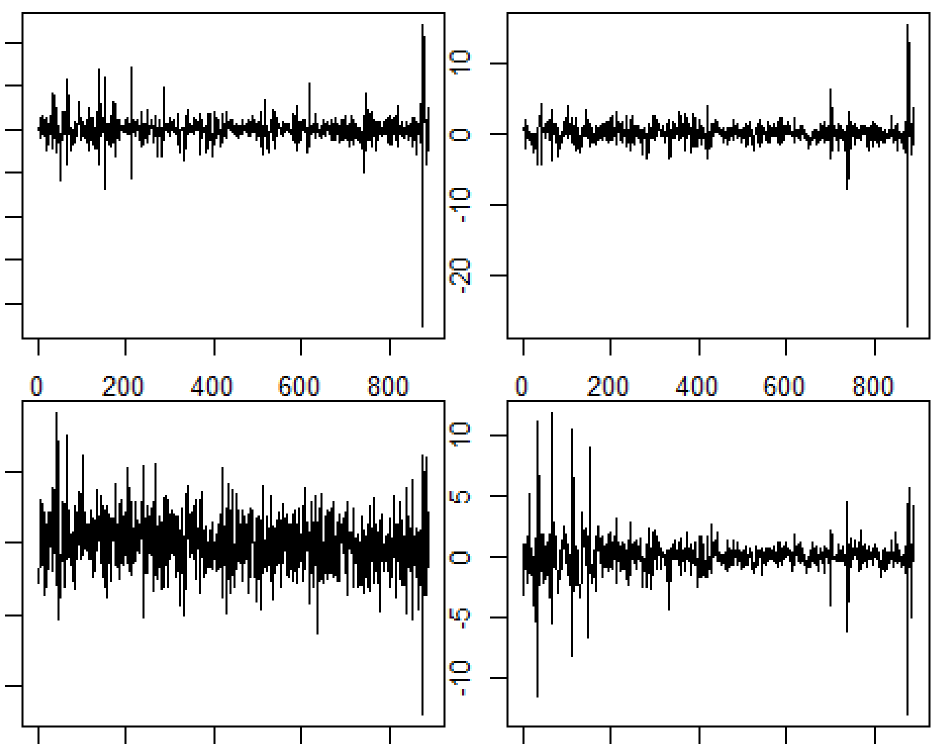
```
## Hit Enter for p-value plot of individual ccm:
```



```
## Hit Enter to compute MQ-statistics:
##
## Ljung-Box Statistics:
##          m       Q(m)     df    p-value
##  [1,]   1.0      15.7    16.0     0.47
##  [2,]   2.0      38.8    32.0     0.19
##  [3,]   3.0      92.1    48.0     0.00
##  [4,]   4.0     114.8    64.0     0.00
##  [5,]   5.0     166.4    80.0     0.00
##  [6,]   6.0     212.5    96.0     0.00
##  [7,]   7.0     239.3   112.0     0.00
##  [8,]   8.0     265.2   128.0     0.00
##  [9,]   9.0     288.1   144.0     0.00
## [10,]  10.0     315.7   160.0     0.00
## [11,]  11.0     337.7   176.0     0.00
## [12,]  12.0     371.5   192.0     0.00
## [13,]  13.0     391.6   208.0     0.00
## [14,]  14.0     417.7   224.0     0.00
## [15,]  15.0     435.0   240.0     0.00
## [16,]  16.0     462.6   256.0     0.00
## [17,]  17.0     473.4   272.0     0.00
## [18,]  18.0     507.9   288.0     0.00
## [19,]  19.0     518.3   304.0     0.00
## [20,]  20.0     524.4   320.0     0.00
## [21,]  21.0     540.6   336.0     0.00
## [22,]  22.0     570.3   352.0     0.00
## [23,]  23.0     586.8   368.0     0.00
## [24,]  24.0     623.8   384.0     0.00
```




```
## Hit Enter to obtain residual plots:
```



$IPD_t = 0.2142919 + 0.1023IPD_{t-1} + 0.347IPM_{t-1} - 0.129IPD_{t-2} - 0.175IPB_{t-2}$

(0.109649) (0.03733) (0.0783) (-0.129) (-0.175)

$IPN_t = 0.1773043 + 0.0348IPD_{t-1} - 0.155IPN_{t-1}$

(0.02686661) (0.00844) (0.0351)

$IPB_t = 0.1842434 + 0.134IPB_{t-1} - 0.185IPM_{t-1} - 0.139IPD_{t-2} + 0.182IPB_{t-2} + 0.102IPM_{t-2}$

(0.05687657) (0.0378) (0.0413) (0.0238) (0.0464) (0.0425)

$IPM_t = 0.124001 + 0.178IPN_{t-1} + 0.232IPM_{t-1} + 0.144IPN_{t-2}$

(0.05346031)(0.0672) (0.0337) (0.0654)

```
#1.3
detach("package:MTS", unload = TRUE)
require(vars)

## Loading required package: vars

## Warning: package 'vars' was built under R version 4.0.5

## Loading required package: MASS

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.0.5

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.0.5

## Loading required package: urca

## Warning: package 'urca' was built under R version 4.0.4

##
## Attaching package: 'urca'

## The following objects are masked from 'package:fUnitRoots':
##
##     punitroot, qunitroot, unitrootTable

## Loading required package: lmtest

varfit=VAR(zt,p=2)
summary(varfit)

##
## VAR Estimation Results:
```

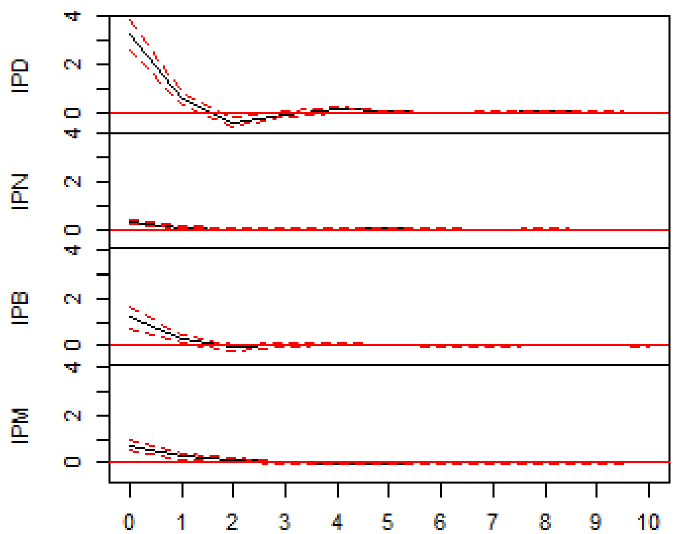
```
## =========================
## Endogenous variables: IPD, IPN, IPB, IPM
## Deterministic variables: const
## Sample size: 889
## Log Likelihood: -6129.48
## Roots of the characteristic polynomial:
## 0.5908 0.4882 0.4882 0.4313 0.241 0.241 0.1895 0.1895
## Call:
## VAR(y = zt, p = 2)
##
##
## Estimation results for equation IPD:
## ====================================
## IPD = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.11876    0.04731   2.510 0.012251 *
## IPN.l1  0.21929    0.14672   1.495 0.135376
## IPB.l1 -0.10721    0.09284  -1.155 0.248490
## IPM.l1  0.33838    0.08214   4.120 4.15e-05 ***
## IPD.l2 -0.15718    0.04706  -3.340 0.000873 ***
## IPN.l2  0.18868    0.14615   1.291 0.197029
## IPB.l2 -0.19773    0.09242  -2.139 0.032671 *
## IPM.l2  0.10530    0.08249   1.277 0.202111
## const   0.17248    0.11338   1.521 0.128534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 3.183 on 880 degrees of freedom
## Multiple R-Squared: 0.08298, Adjusted R-squared: 0.07465
## F-statistic: 9.954 on 8 and 880 DF,  p-value: 2.479e-13
##
##
## Estimation results for equation IPN:
## ====================================
## IPN = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.0295464  0.0116981   2.526   0.0117 *
## IPN.l1 -0.1678360  0.0362760  -4.627 4.27e-06 ***
## IPB.l1  0.0091590  0.0229539   0.399   0.6900
## IPM.l1  0.0161758  0.0203075   0.797   0.4259
## IPD.l2  0.0004192  0.0116354   0.036   0.9713
## IPN.l2 -0.0449204  0.0361332  -1.243   0.2141
## IPB.l2 -0.0013243  0.0228504  -0.058   0.9538
## IPM.l2  0.0008203  0.0203941   0.040   0.9679
## const   0.1811122  0.0280311   6.461 1.72e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.787 on 880 degrees of freedom
## Multiple R-Squared: 0.03302, Adjusted R-squared: 0.02423
## F-statistic: 3.756 on 8 and 880 DF,  p-value: 0.000244
##
##
## Estimation results for equation IPB:
## ====================================
## IPB = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.003394   0.024469   0.139 0.889715
## IPN.l1  0.053932   0.075880   0.711 0.477431
## IPB.l1  0.122725   0.048014   2.556 0.010754 *
## IPM.l1  0.180015   0.042478   4.238  2.5e-05 ***
## IPD.l2 -0.142158   0.024338  -5.841  7.3e-09 ***
## IPN.l2  0.024972   0.075582   0.330 0.741175
## IPB.l2  0.182772   0.047797   3.824 0.000141 ***
## IPM.l2  0.103136   0.042659   2.418 0.015822 *
## const   0.175722   0.058634   2.997 0.002804 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.646 on 880 degrees of freedom
## Multiple R-Squared: 0.1179,  Adjusted R-squared: 0.1098
## F-statistic:  14.7 on 8 and 880 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation IPM:
## ====================================
## IPM = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.01713    0.02268   0.755   0.4503
## IPN.l1  0.14348    0.07032   2.040   0.0416 *
## IPB.l1  0.02771    0.04450   0.623   0.5335
## IPM.l1  0.21123    0.03937   5.366 1.03e-07 ***
## IPD.l2  0.01063    0.02256   0.471   0.6375
## IPN.l2  0.13377    0.07005   1.910   0.0565 .
## IPB.l2  0.02122    0.04430   0.479   0.6320
## IPM.l2 -0.06006    0.03954  -1.519   0.1291
## const   0.12757    0.05434   2.348   0.0191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
##
## Residual standard error: 1.526 on 880 degrees of freedom
## Multiple R-Squared: 0.08573, Adjusted R-squared: 0.07742
## F-statistic: 10.31 on 8 and 880 DF,  p-value: 7.286e-14
##
##
##
## Covariance matrix of residuals:
##         IPD    IPN    IPB    IPM
## IPD 10.1321 0.8423 3.7308 2.3641
## IPN  0.8423 0.6194 0.4452 0.3173
## IPB  3.7308 0.4452 2.7099 1.2697
## IPM  2.3641 0.3173 1.2697 2.3275
##
## Correlation matrix of residuals:
##        IPD    IPN    IPB    IPM
## IPD 1.0000 0.3363 0.7120 0.4868
## IPN 0.3363 1.0000 0.3436 0.2643
## IPB 0.7120 0.3436 1.0000 0.5055
## IPM 0.4868 0.2643 0.5055 1.0000
```

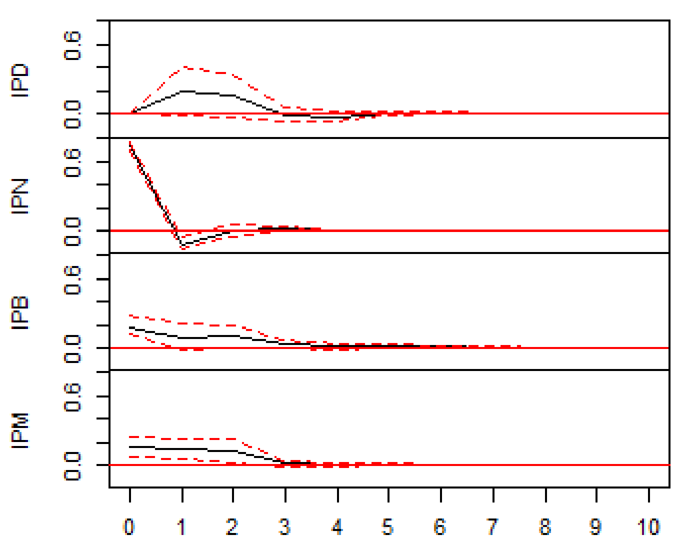
```
impresp=irf(varfit)
plot(impresp)
```

Orthogonal Impulse Response from IPD

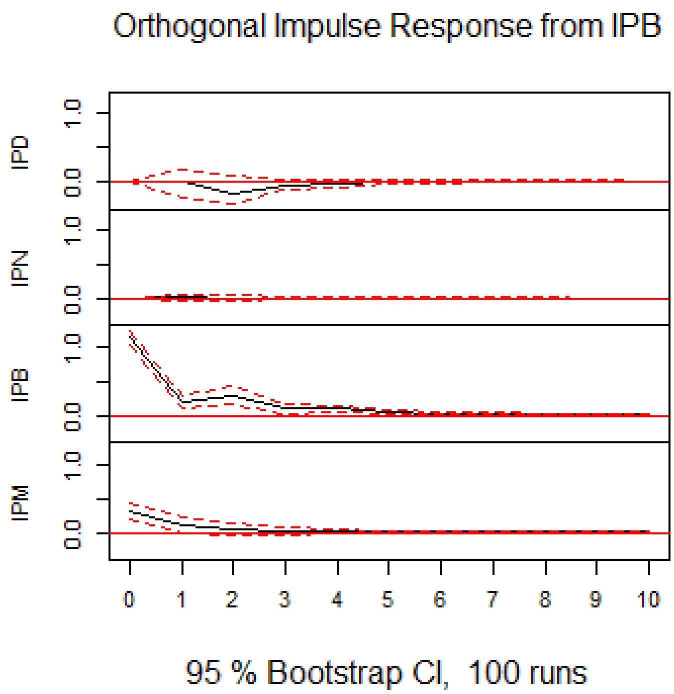


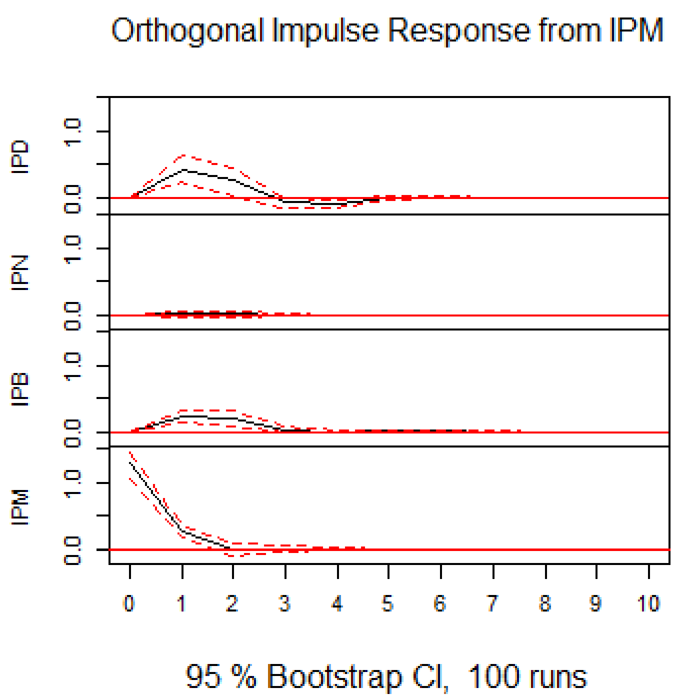
95 % Bootstrap CI, 100 runs

Orthogonal Impulse Response from IPN



95 % Bootstrap CI, 100 runs


Orthogonal Impulse Response from IPB
IPD
IPN
IPB
IPM
1.0
0.0
0 1 2 3 4 5 6 7 8 9 10
95 % Bootstrap CI, 100 runs


Orthogonal Impulse Response from IPM
IPD
IPN
IPB
IPM
1.0
0.0
0 1 2 3 4 5 6 7 8 9 10
95 % Bootstrap CI, 100 runs

The shock from IPD on other variables are mostly similar. There are positive impacts in the first and second lag, then gradually converge to zero.

The shock from IPN affects IPD greatly on the first 2 periods. It also impacts IPB and IPM on these 2 periods with a positive effect, then converge to zero at the start of lag 3. The impact of IPN on itself is highly positive in the first lag, negative in the second lag, and becomes zero afterwards.

The shock from IPB to IPD is uncertain as it is positive and negative in the consecutive lags. The effect of IPB to IPD is insignificant. The effect to itself is positive but with different magnitude in the first 3 lags. And the effect of IPB to IPM is positive in the first and second lags.

The shock from IPM to IPD is uncertain, with positive value on the first three lags, and negative for lag 4-5. The effect on IPN is insignificant. The effect on IPB is positive until lag 3. While the effect on itself shows strong declining trend in the first 2 periods.

```
#1.4
fevd(varfit,n.ahead=6)

## $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
##
## $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
##
## $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
```

```
##
## $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
```

The forecast error variance decomposition for IPD is such that: the shock of IPD to itself is from 96.6% to 100%. IPN can be explained by IPD by 0.36 to 0.59%. IPB can be explained by 1.55e-05% to 3.4e-03%. And IPM can be explained by 1.76% to 2.42%.

The forecast error variance decomposition for IPN is such that: the IPD can be influenced by IPN by 11.3% to 11.7%. IPN can be explained by itself by 88.05% to 88.69%. IPB can be influenced by 0.038% to 0.039%. And IPM can be influenced by 0.006% to 0.11%.

The forecast error variance decomposition for IPB is such that: the IPD can be explained by 48% to 50%. IPN can be explained by 12% to 17%. IPB can be explained by itself by 47% to 48%. And IPM can be explained by 1.8% to 3.2%.

The forecast error variance decomposition for IPM is such that: the IPD can be explained by 23% to 25%. IPN can be explained by 1-2%. IPB can be explained by 4.5 to 4.6%. And IPM can be explained by itself by 67 to 70%.

```
#1.5

varfit.prd=predict(varfit,n.ahead=6,ci=0.95)

detach("package:vars", unload = TRUE)
require(MTS)

## Loading required package: MTS

## Warning: package 'MTS' was built under R version 4.0.5

##
## Attaching package: 'MTS'

## The following object is masked from 'package:TTR':
##
##     VMA

VARpred(m1a,6,orig=879)

## orig  879
## Forecasts at origin:  879
##           IPD      IPN      IPB     IPM
```

```
## [1,] -5.91054 -0.71202 -5.44726 -4.2032
## [2,]  9.30575  0.08195 -1.38707 -1.6795
## [3,]  2.30107  0.48882 -0.90869 -0.3538
## [4,] -0.62803  0.18156 -1.72370  0.1407
## [5,]  0.06237  0.12722 -0.54210  0.2592
## [6,]  0.69424  0.15972 -0.05163  0.2330
## Standard Errors of predictions:
##       [,1]   [,2]  [,3]  [,4]
## [1,] 3.177 0.7841 1.638 1.522
## [2,] 3.263 0.7956 1.695 1.577
## [3,] 3.305 0.7961 1.736 1.586
## [4,] 3.309 0.7963 1.740 1.587
## [5,] 3.311 0.7963 1.744 1.587
## [6,] 3.311 0.7963 1.745 1.587
## Root mean square errors of predictions:
##       [,1]   [,2]  [,3]  [,4]
## [1,] 3.194 0.7881 1.647 1.530
## [2,] 3.286 0.7987 1.710 1.591
## [3,] 3.317 0.7962 1.748 1.589
## [4,] 3.310 0.7964 1.741 1.587
## [5,] 3.311 0.7963 1.745 1.587
## [6,] 3.311 0.7963 1.745 1.587
## Observations, predicted values,     errors, and MSE
##      time     obs    fcst     err     obs    fcst     err     obs    fcst
## case  880 18.0079 -5.9105 23.9184  1.7436 -0.7120  2.4556  9.8203 -5.4473
## case  881 31.6900  9.3057 22.3843  2.1309  0.0819  2.0490 11.7652 -1.3871
## case  882 15.0623  2.3011 12.7613  1.0549  0.4888  0.5660  7.4632 -0.9087
## case  883 -0.9751 -0.6280 -0.3470  1.2688  0.1816  1.0872  2.5593 -1.7237
## case  884 -1.3476  0.0624 -1.4099 -1.5321  0.1272 -1.6593 -0.1420 -0.5421
## case  885  0.8809  0.6942  0.1866  0.0960  0.1597 -0.0637  0.8422 -0.0516
##          err     obs    fcst     err
## case 15.2675 -1.0442 -4.2032  3.1590
## case 13.1523  5.1887 -1.6795  6.8682
## case  8.3719  3.6564 -0.3538  4.0102
## case  4.2830  0.6973  0.1407  0.5566
## case  0.4001  0.0761  0.2592 -0.1831
## case  0.8938  1.1026  0.2330  0.8697
```

QUESTION 2

```
library(quantmod)
getSymbols('ETH-USD', from = "2015-01-01", to = "2021-05-24" )

## '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"
```

```
`ETH-USD` = na.omit(`ETH-USD`)
prices = Ad(`ETH-USD`)

colnames(prices) = "adj_price"

rt = diff(log((prices)))
rt = na.omit(rt)

t.test(rt)
```

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

Step1: Test for normality: Employed Jarque-Bera test to test.

H_0 : Follows normal distribution.

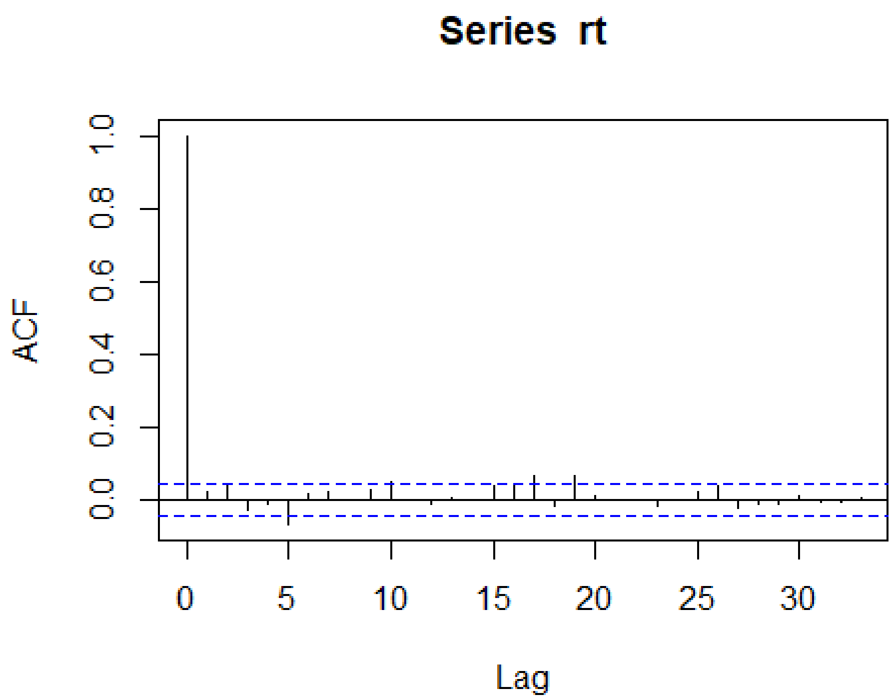
H_a : Does not follow normal distribution

According to the JB test, we can reject the null hypothesis at 95% significance level. So, the data does not follow normal distribution.

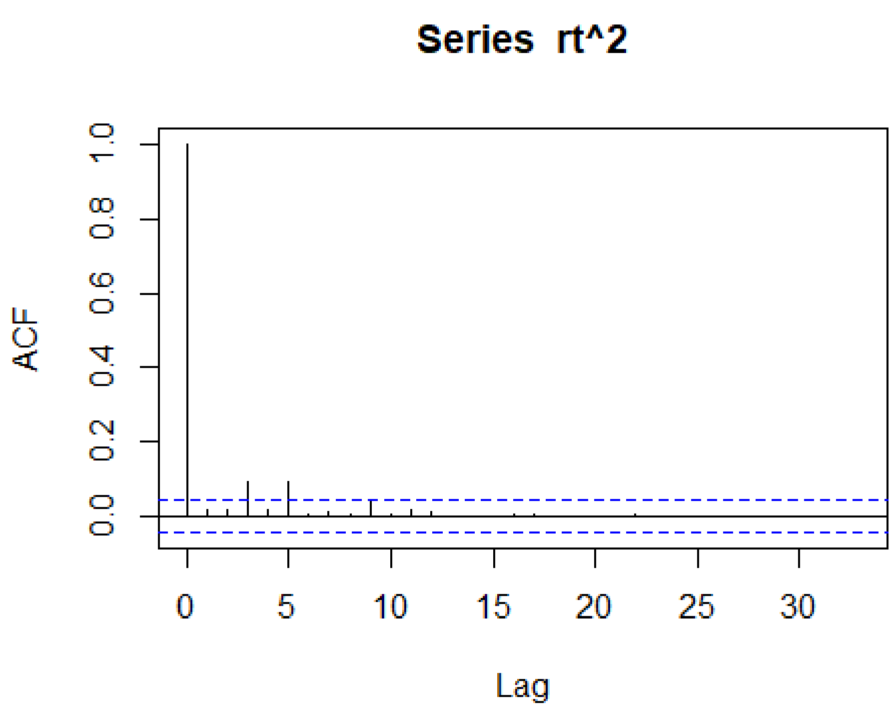
```
## Warning in cint * stderr: Recycling array of length 1 in vector-array arit
hmetic is deprecated.
##   Use c() or as.vector() instead.
```

```
##
##  One Sample t-test
##
## data:  rt
## t = 2.1827, df = 2112, p-value = 0.02917
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.0003296413 0.0061637626
## sample estimates:
##   mean of x
## 0.003246702
```

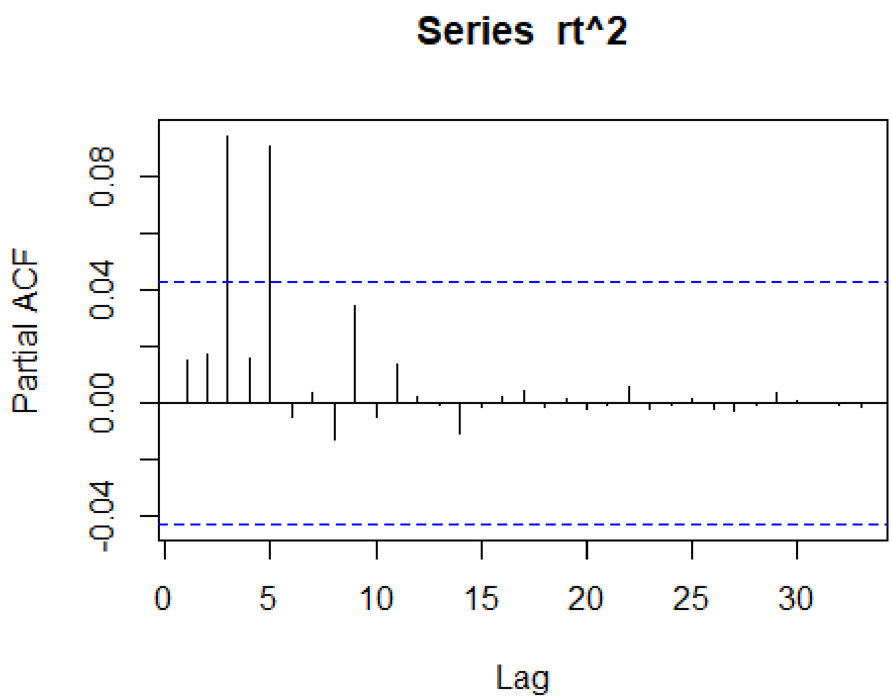
```
acf(rt)
```



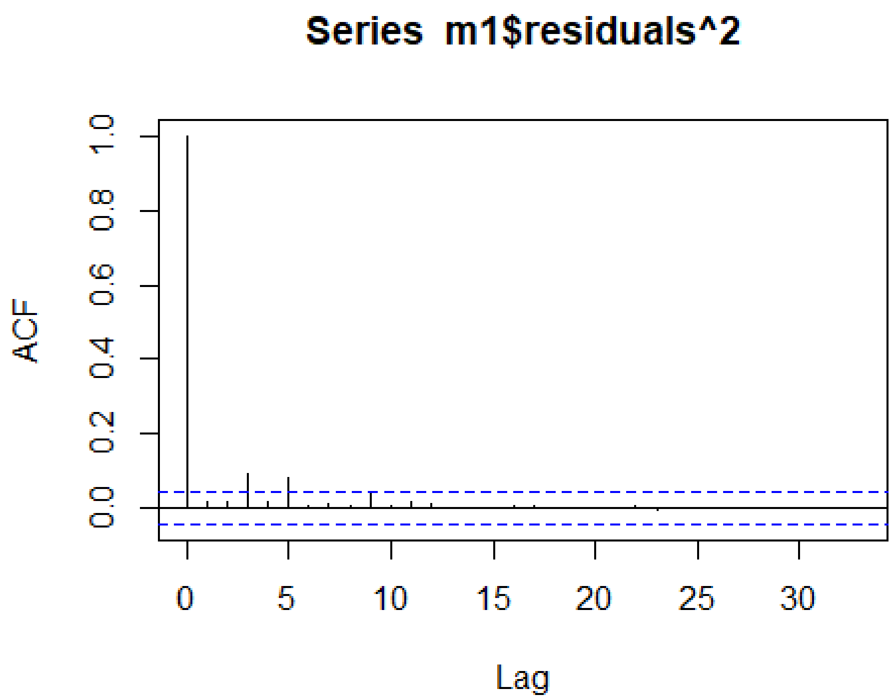

```
acf(rt^2)
```




```
pacf(rt^2)
```

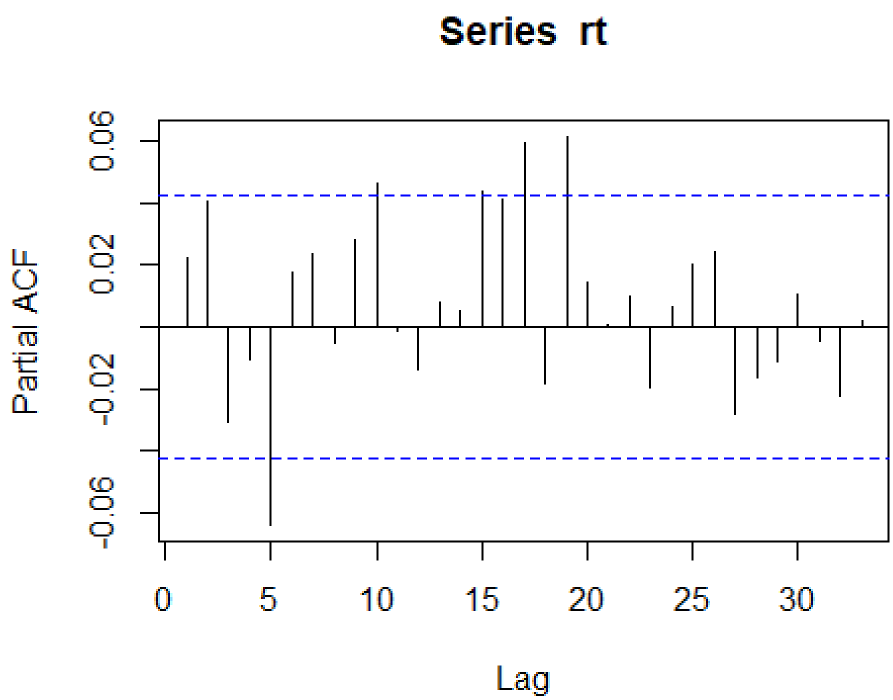



```
Box.test(rt,lag=10,type='Ljung')

##
##  Box-Ljung test
##
## data:  rt
## X-squared = 24.432, df = 10, p-value = 0.006532

Box.test(rt^2,lag=10,type='Ljung')

##
##  Box-Ljung test
##
## data:  rt^2
## X-squared = 42.898, df = 10, p-value = 5.186e-06

pacf(rt)

m1=arima(rt,order=c(0,0,2))
acf(m1$residuals^2)
```




```
Box.test(m1$residuals^2,lag=10,type='Ljung')

##
##  Box-Ljung test
##
## data:  m1$residuals^2
## X-squared = 37.583, df = 10, p-value = 4.485e-05
```

```
#THERE IS AN ARCH EFFECT
```



#Using Ljung box test for serial correlation in the log return series

H_0 : there is no serial correlation.

H_a : there is serial correlation in the return series

And according to the pattern of the ACF, there is no evidence of exponential decay. So, AR model is not applicable in this case. MA model will be used with the optimal lag of 1.

H_0 : there is no ARCH effect in the log return series.

H_a : there is an ARCH effect

```
m4=garchFit(~arma(0,2)+garch(1,1),data=rt,trace=F)

## Warning: Using formula(x) is deprecated when x is a character vector of le
ngth > 1.
##   Consider formula(paste(x, collapse = " ")) instead.

summary(m4)
```

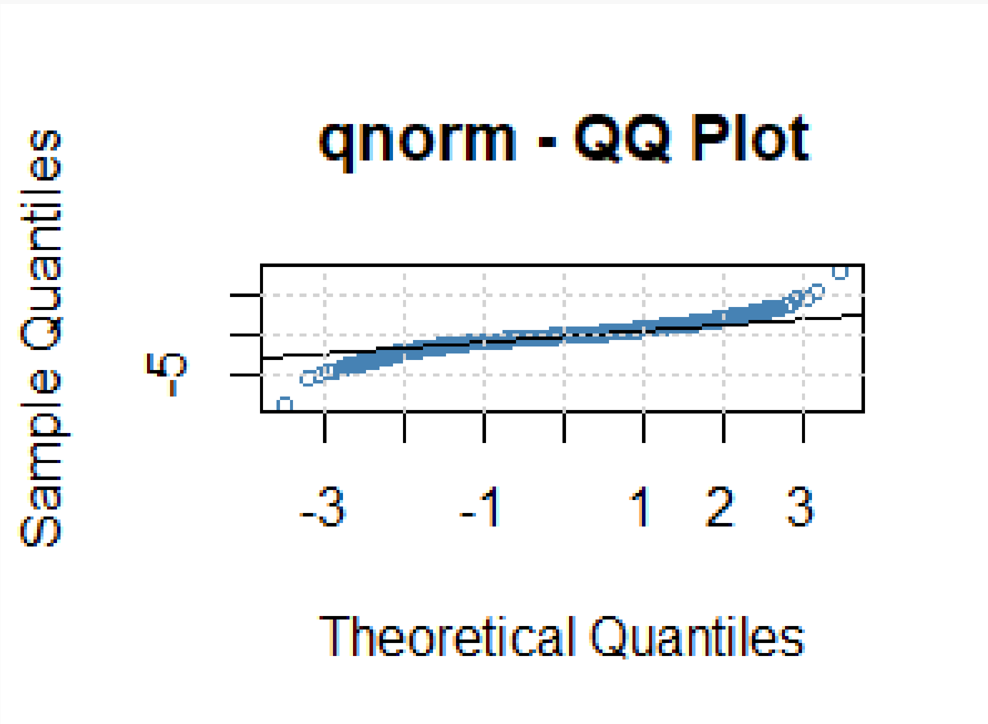
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(0, 2) + garch(1, 1), data = rt, trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(0, 2) + garch(1, 1)
## <environment: 0x00000000216063c8>
##  [data = rt]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##         mu         ma1         ma2       omega      alpha1       beta1
## 0.00247594  0.01843321  0.06251420  0.00023344  0.15220661  0.79564733
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     2.476e-03   1.191e-03    2.079   0.0376 *
## ma1    1.843e-02   2.546e-02    0.724   0.4690
## ma2    6.251e-02   2.544e-02    2.457   0.0140 *
## omega  2.334e-04   4.526e-05    5.157 2.50e-07 ***
## alpha1 1.522e-01   1.930e-02    7.888 3.11e-15 ***
## beta1  7.956e-01   2.450e-02   32.481  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3084.753    normalized:  1.459893
##
## Description:
##  Fri Jun 11 13:39:33 2021 by user: Orachai
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  4551.905  0
##  Shapiro-Wilk Test  R    W      0.9310086 0
##  Ljung-Box Test     R    Q(10)  17.31726  0.06763196
##  Ljung-Box Test     R    Q(15)  23.88379  0.06709331
##  Ljung-Box Test     R    Q(20)  39.20348  0.006289433
##  Ljung-Box Test     R^2  Q(10)  5.516288  0.8541355
##  Ljung-Box Test     R^2  Q(15)  7.238377  0.9506941
##  Ljung-Box Test     R^2  Q(20)  9.006211  0.9828352
```

```
##   LM Arch Test       R    TR^2   7.882436  0.7942488
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -2.914106 -2.898046 -2.914122 -2.908226
```

```
#plot(m4)
```

```
predict(m4,6)
```

```
##   meanForecast meanError standardDeviation
## 1  0.001798651 0.1343389         0.1343389
## 2  0.016772333 0.1317021         0.1316788
## 3  0.002475942 0.1294024         0.1291068
## 4  0.002475942 0.1269103         0.1266207
## 5  0.002475942 0.1245021         0.1242183
## 6  0.002475942 0.1221757         0.1218975
```



Fitted model:

$$r_t = 0.0022755190794 + 0.01843321\varepsilon_{t-1} + 0.06251420\varepsilon_{t-2}$$

(1.191e-03) (2.546e-02) (2.544e-02)

$$\sigma^2_t = 2.334e\text{-}04 + 0.1522a^2_{t-1} + 0.7956e\text{-}01\sigma^2_{t-1}$$

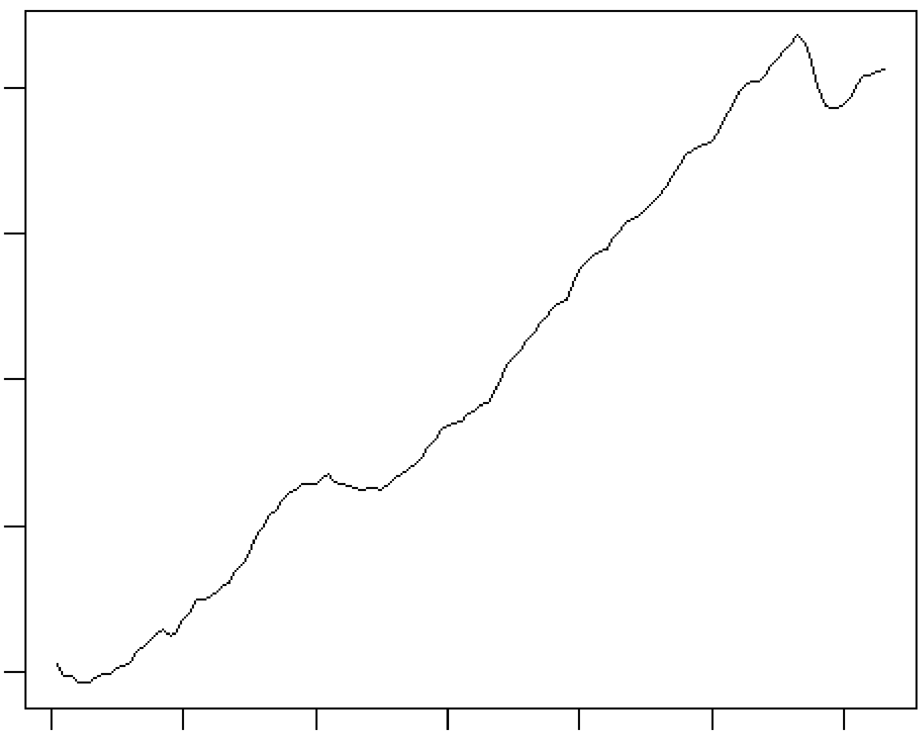
(4.526e-02 (1.93e-02) (2.45e-02)

According to the Ljung box test, H_0 is not rejected at 0.05 level of significance. This implies that the mean equation is adequate. The volatility model is also adequate as there is no linear dependency in α^2_t.

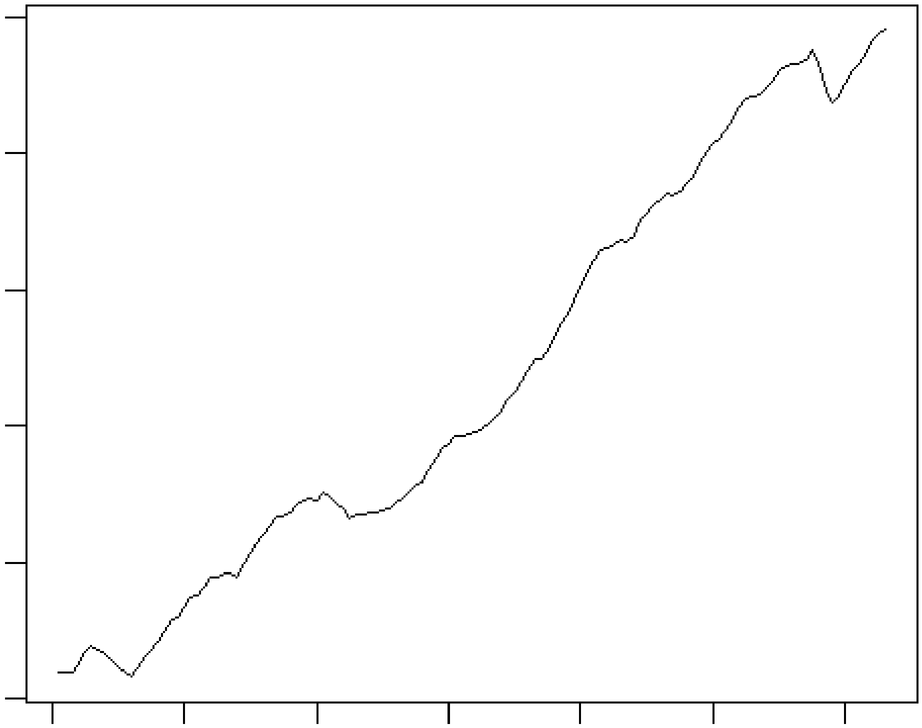
QUESTION 3

3.1

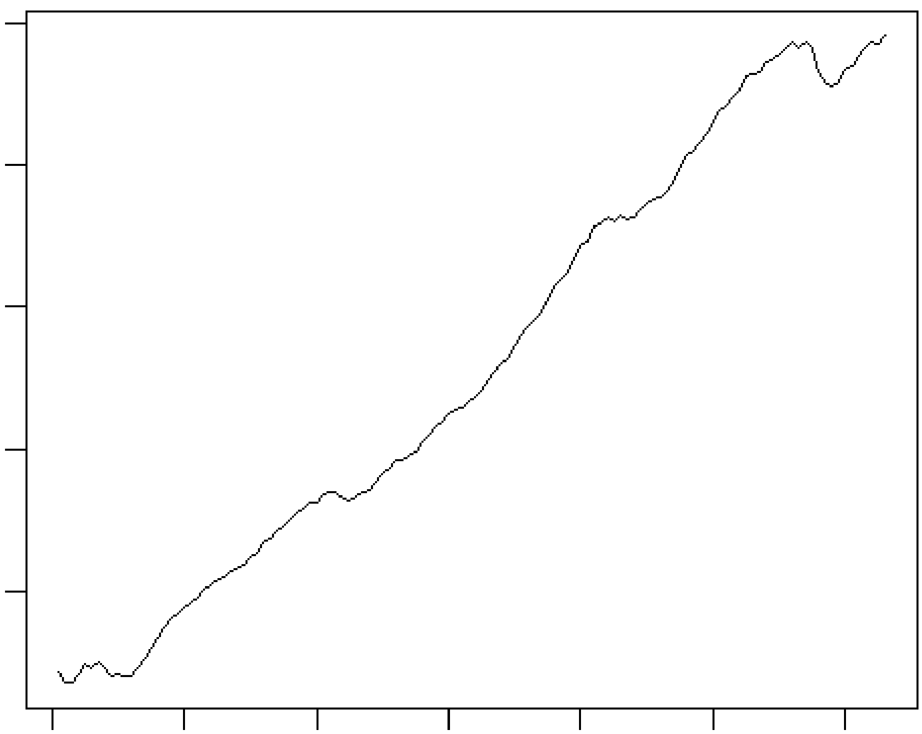
```
require(MTS)
getSymbols("CLVMNACSCAB1GQUK",src="FRED")

## [1] "CLVMNACSCAB1GQUK"

getSymbols("NAEXKP01CAQ189S",src="FRED")

## [1] "NAEXKP01CAQ189S"

getSymbols("GDPC1",src="FRED")

## [1] "GDPC1"

UK=CLVMNACSCAB1GQUK[21:146]
CAN=NAEXKP01CAQ189S[77:202]
USA=GDPC1[133:258]
ts.plot(UK)
```



```
ts.plot(CAN)
```



```
ts.plot(USA)
```



```
data=cbind(as.numeric(UK),as.numeric(CAN),as.numeric(USA))
dim(data)
```

```
## [1] 126   3
```

```
library(urca)
adfuk=ur.df(log(UK),type="drift",lags=2)
summary(adfuk)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.016595 -0.003106  0.000196  0.003234  0.014568
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.043159   0.028129   1.534   0.1276
## z.lag.1     -0.003241   0.002228  -1.454   0.1485
## z.diff.lag1  0.455787   0.089956   5.067  1.5e-06 ***
## z.diff.lag2  0.148802   0.084779   1.755   0.0818 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005745 on 119 degrees of freedom
## Multiple R-squared:  0.3382, Adjusted R-squared:  0.3215
## F-statistic: 20.27 on 3 and 119 DF,  p-value: 1.119e-10
##
##
## Value of test-statistic is: -1.4542 6.2991
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
adfcan=ur.df(log(CAN),type="drift",lags=2)
summary(adfcan)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
```

```
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0183888 -0.0038802  0.0008965  0.0039069  0.0234761
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.055837   0.068794   0.812    0.419
## z.lag.1     -0.001996   0.002600  -0.768    0.444
## z.diff.lag1  0.563594   0.091410   6.166 9.94e-09 ***
## z.diff.lag2 -0.054727   0.091177  -0.600    0.549
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006786 on 119 degrees of freedom
## Multiple R-squared:  0.2932, Adjusted R-squared:  0.2754
## F-statistic: 16.46 on 3 and 119 DF,  p-value: 5.215e-09
##
##
## Value of test-statistic is: -0.7679 7.2188
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
adfus=ur.df(log(USA),type="drift",lags=2)
summary(adfus)
```

```
##
## ###############################################
## # Augmented Dickey-Fuller Test Unit Root Test #
## ###############################################
##
## Test regression drift
##
##
## Call:
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.023089 -0.003436  0.000480  0.004300  0.015611
```

```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.039425   0.020495   1.924 0.056787 .
## z.lag.1     -0.003828   0.002193  -1.745 0.083483 .
## z.diff.lag1  0.338684   0.090849   3.728 0.000297 ***
## z.diff.lag2  0.117070   0.085463   1.370 0.173318
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006516 on 119 degrees of freedom
## Multiple R-squared:  0.2078, Adjusted R-squared:  0.1878
## F-statistic:  10.4 on 3 and 119 DF,  p-value: 3.938e-06
##
##
## Value of test-statistic is: -1.7455 10.4568
##
## Critical values for test statistics:
##       1pct  5pct 10pct
## tau2 -3.46 -2.88 -2.57
## phi1  6.52  4.63  3.81
```

```
zt1=diffM(log(data))
colnames(zt1)=c("UK","CANADA","US")
VARorder(zt1)
```

```
## selected order: aic =  1
## selected order: bic =  1
## selected order: hq =  1
## Summary table:
##         p     AIC      BIC       HQ     M(p) p-value
##  [1,]  0 -30.7048 -30.7048 -30.7048   0.0000  0.0000
##  [2,]  1 -31.6569 -31.4533 -31.5742 117.8338  0.0000
##  [3,]  2 -31.6544 -31.2472 -31.4890  14.7882  0.0969
##  [4,]  3 -31.5615 -30.9506 -31.3133   5.1821  0.8182
##  [5,]  4 -31.5481 -30.7335 -31.2172  12.8638  0.1689
##  [6,]  5 -31.4529 -30.4347 -31.0392   4.6592  0.8629
##  [7,]  6 -31.3979 -30.1761 -30.9015   8.2336  0.5108
##  [8,]  7 -31.2818 -29.8563 -30.7027   2.4941  0.9810
##  [9,]  8 -31.3040 -29.6749 -30.6421  14.3771  0.1095
## [10,]  9 -31.2757 -29.4430 -30.5312   9.6683  0.3780
## [11,] 10 -31.1946 -29.1582 -30.3674   5.0621  0.8289
## [12,] 11 -31.2660 -29.0259 -30.3560  16.6884  0.0538
## [13,] 12 -31.3153 -28.8716 -30.3226  14.4031  0.1087
## [14,] 13 -31.2559 -28.6086 -30.1804   6.0477  0.7351
```

```
m3=VAR(zt1,3)
```

```
## Constant term:
## Estimates:  0.002098151 0.0001718731 0.002484129
## Std.Error:  0.0008372519 0.0008328323 0.0008887753
```

```
## AR coefficient matrix
## AR( 1 )-matrix
##       [,1]    [,2]   [,3]
## [1,] 0.486 0.00664 0.0796
## [2,] 0.416 0.27745 0.4695
## [3,] 0.423 0.16377 0.2098
## standard error
##        [,1]  [,2]   [,3]
## [1,] 0.0916 0.101 0.0949
## [2,] 0.0911 0.101 0.0944
## [3,] 0.0973 0.107 0.1007
## AR( 2 )-matrix
##         [,1]   [,2]    [,3]
## [1,]  0.0507  0.114 -0.0546
## [2,] -0.1365 -0.129  0.0173
## [3,] -0.1299 -0.109  0.1465
## standard error
##       [,1]  [,2]  [,3]
## [1,] 0.107 0.102 0.105
## [2,] 0.107 0.102 0.104
## [3,] 0.114 0.109 0.111
## AR( 3 )-matrix
##          [,1]    [,2]   [,3]
## [1,] -0.04293 -0.2013 0.1839
## [2,] -0.01870 -0.0532 0.0503
## [3,] -0.00226 -0.0832 0.0555
## standard error
##        [,1]   [,2]   [,3]
## [1,] 0.0961 0.0856 0.0963
## [2,] 0.0956 0.0852 0.0958
## [3,] 0.1020 0.0909 0.1022
##
## Residuals cov-mtx:
##              [,1]         [,2]         [,3]
## [1,] 2.911274e-05 1.864267e-06 6.452746e-06
## [2,] 1.864267e-06 2.880619e-05 1.092736e-05
## [3,] 6.452746e-06 1.092736e-05 3.280611e-05
##
## det(SSE) =  2.298528e-14
## AIC =  -30.97192
## BIC =  -30.36101
## HQ  =  -30.72374
```

```
m3a = refVAR(m3,thres=1.645)
```

```
## Constant term:
## Estimates:  0.002365646 0 0.002491684
## Std.Error:  0.0007471589 0 0.0007999906
## AR coefficient matrix
## AR( 1 )-matrix
```

```
##       [,1]  [,2] [,3]
## [1,] 0.530 0.000 0.00
## [2,] 0.356 0.251 0.46
## [3,] 0.367 0.153 0.19
## standard error
##        [,1]   [,2]   [,3]
## [1,] 0.0765 0.0000 0.0000
## [2,] 0.0765 0.0879 0.0866
## [3,] 0.0853 0.0873 0.0979
## AR( 2 )-matrix
##      [,1]   [,2] [,3]
## [1,]    0  0.000    0
## [2,]    0 -0.135    0
## [3,]    0  0.000    0
## standard error
##      [,1]   [,2] [,3]
## [1,]    0 0.0000    0
## [2,]    0 0.0736    0
## [3,]    0 0.0000    0
## AR( 3 )-matrix
##      [,1]   [,2]  [,3]
## [1,]    0 -0.186 0.235
## [2,]    0  0.000 0.000
## [3,]    0  0.000 0.000
## standard error
##      [,1]   [,2]   [,3]
## [1,]    0 0.0813 0.0849
## [2,]    0 0.0000 0.0000
## [3,]    0 0.0000 0.0000
##
## Residuals cov-mtx:
##              [,1]         [,2]         [,3]
## [1,] 3.002134e-05 1.757092e-06 6.013706e-06
## [2,] 1.757092e-06 2.950146e-05 1.164017e-05
## [3,] 6.013706e-06 1.164017e-05 3.424233e-05
##
## det(SSE) =  2.533318e-14
## AIC =  -31.14666
## BIC =  -30.9204
## HQ  =  -31.05474
```

Fitted VAR model

$UK = 0.002365 + 0.530UK_{t-1} - 0.186CAN_{t-3} + 0.235USA_{t-3}$

(0.000747) (0.0765) (0.0813) (0.0849)

$CAN = 0.356UK_{t-1} + 0.251CAN_{t-1} + 0.46USA_{t-1} - 0{,}135CAN_{t-2}$

(0.0765) (0.0879) (0.0866) (0.0736)

$USA = 0.00249 + 0.367UK_{t-3} + 0.153CAN_{t-1} + 0.19USA_{t-1}$

(0.000799)(0.0853) (0.0873) (0.0979)

Model checking:

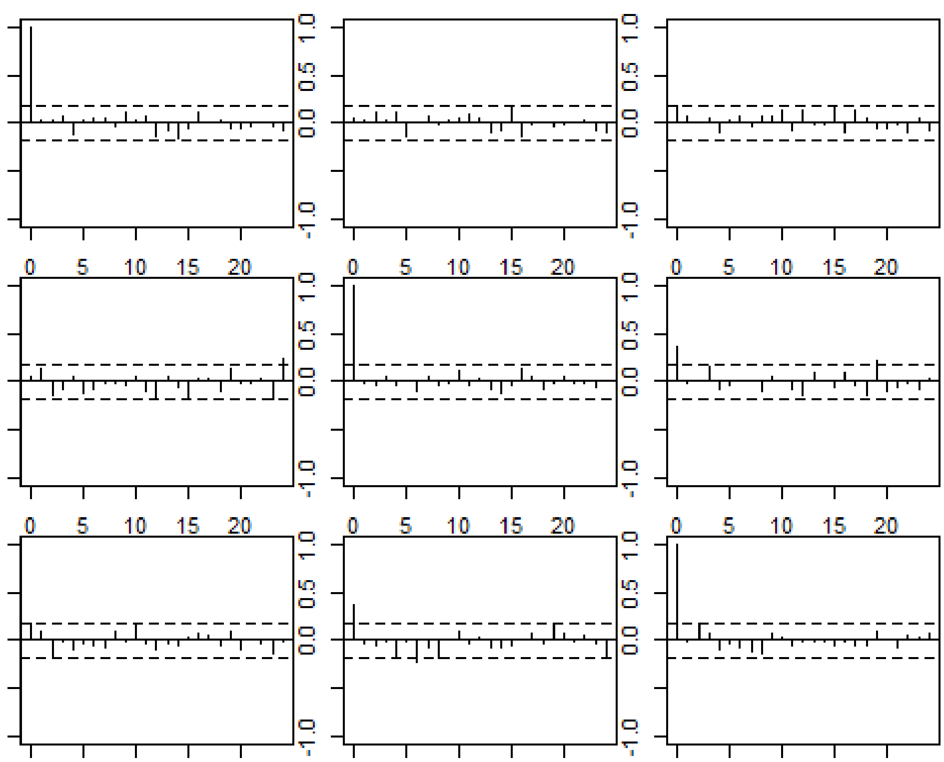
According to the Ljung box statistics, we can not reject the null hypothesis, which means that the model is appropriate.

```
par(mar=c(1,1,1,1))
MTSdiag(m3a)
```

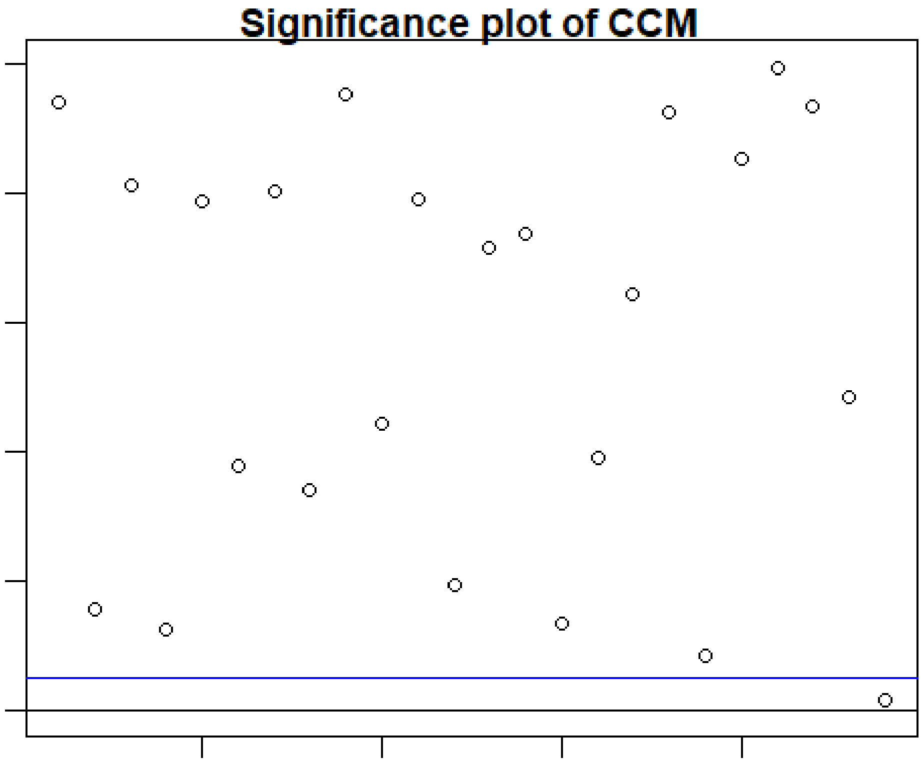
```
## [1] "Covariance matrix:"
##              UK   CANADA       US
## UK     3.03e-05 1.77e-06 6.06e-06
## CANADA 1.77e-06 2.97e-05 1.17e-05
## US     6.06e-06 1.17e-05 3.45e-05
## CCM at lag:  0
##       [,1]  [,2]  [,3]
## [1,] 1.000 0.059 0.188
## [2,] 0.059 1.000 0.366
## [3,] 0.188 0.366 1.000
## Simplified matrix:
## CCM at lag:  1
## . . .
## . . .
## . . .
## CCM at lag:  2
## . . .
## . . .
## . . .
## CCM at lag:  3
## . . .
## . . .
## . . .
## CCM at lag:  4
## . . .
## . . .
## . . .
## CCM at lag:  5
## . . .
## . . .
## . . .
## CCM at lag:  6
## . . .
## . . .
## . - .
## CCM at lag:  7
## . . .
## . . .
```

```
## . . .
## CCM at lag:  8
## . . .
## . . .
## . . .
## CCM at lag:  9
## . . .
## . . .
## . . .
## CCM at lag:  10
## . . .
## . . .
## . . .
## CCM at lag:  11
## . . .
## . . .
## . . .
## CCM at lag:  12
## . . .
## . . .
## . . .
## CCM at lag:  13
## . . .
## . . .
## . . .
## CCM at lag:  14
## . . .
## . . .
## . . .
## CCM at lag:  15
## . . .
## . . .
## . . .
## CCM at lag:  16
## . . .
## . . .
## . . .
## CCM at lag:  17
## . . .
## . . .
## . . .
## CCM at lag:  18
## . . .
## . . .
## . . .
## CCM at lag:  19
## . . .
## . . +
## . . .
## CCM at lag:  20
```

```
## . . .
## . . .
## . . .
## CCM at lag:  21
## . . .
## . . .
## . . .
## CCM at lag:  22
## . . .
## . . .
## . . .
## CCM at lag:  23
## . . .
## . . .
## . . .
## CCM at lag:  24
## . . .
## + . .
## . . .
```

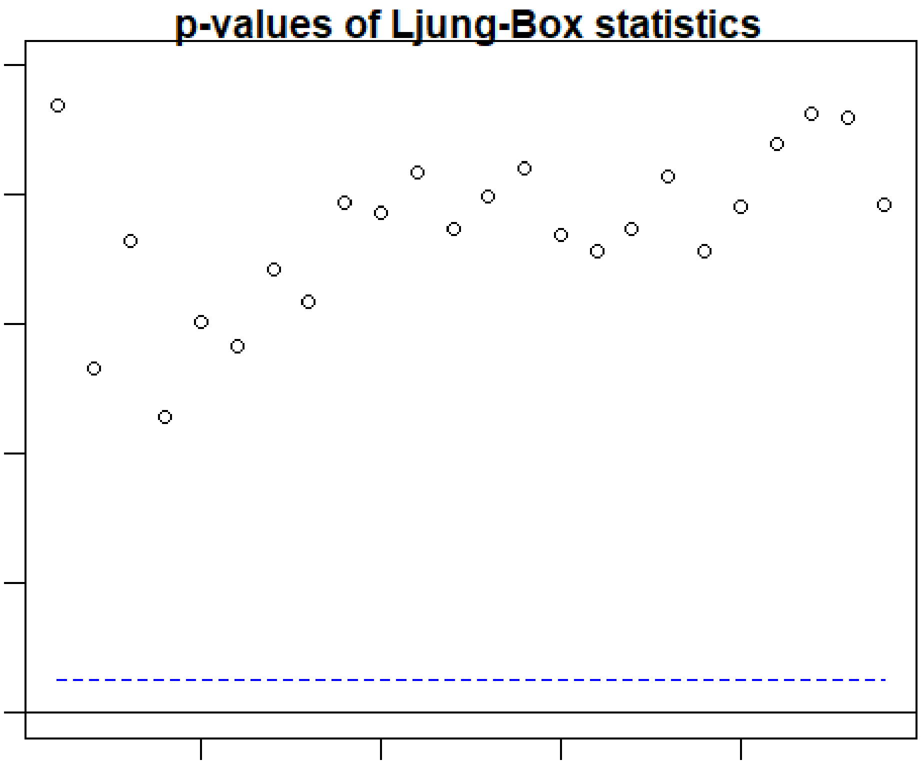


```
## Hit Enter for p-value plot of individual ccm:
```

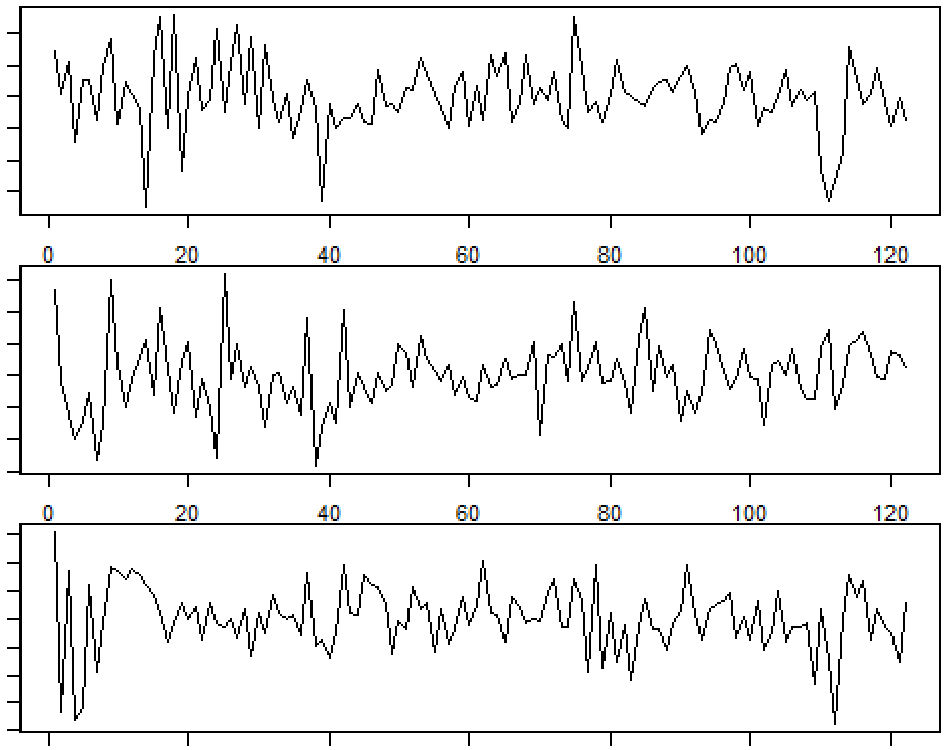



```
## Hit Enter to compute MQ-statistics:
##
## Ljung-Box Statistics:
##           m       Q(m)     df    p-value
##  [1,]   1.00      3.54    9.00      0.94
##  [2,]   2.00     16.86   18.00      0.53
##  [3,]   3.00     22.18   27.00      0.73
##  [4,]   4.00     36.25   36.00      0.46
##  [5,]   5.00     41.87   45.00      0.61
##  [6,]   6.00     51.64   54.00      0.57
##  [7,]   7.00     57.05   63.00      0.69
##  [8,]   8.00     67.29   72.00      0.63
##  [9,]   9.00     70.57   81.00      0.79
## [10,]  10.00     79.65   90.00      0.77
## [11,]  11.00     85.21   99.00      0.84
## [12,]  12.00     97.88  108.00      0.75
## [13,]  13.00    104.07  117.00      0.80
## [14,]  14.00    110.15  126.00      0.84
## [15,]  15.00    124.15  135.00      0.74
## [16,]  16.00    133.94  144.00      0.71
## [17,]  17.00    140.99  153.00      0.75
## [18,]  18.00    144.91  162.00      0.83
## [19,]  19.00    160.14  171.00      0.71
## [20,]  20.00    164.92  180.00      0.78
## [21,]  21.00    166.45  189.00      0.88
## [22,]  22.00    170.03  198.00      0.93
```

```
## [23,]  23.00    178.88  207.00     0.92
## [24,]  24.00    199.21  216.00     0.79
```




```
## Hit Enter to obtain residual plots:
```

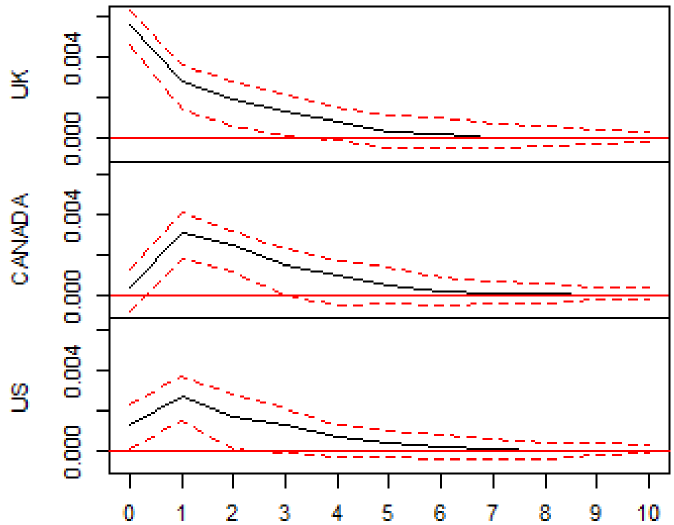



```
#3.2
detach("package:MTS", unload = TRUE)
require(vars)

## Loading required package: vars

## Warning: package 'vars' was built under R version 4.0.5

varfit1=VAR(zt1,p=3)
summary(varfit1)

##
## VAR Estimation Results:
## =========================
## Endogenous variables: UK, CANADA, US
## Deterministic variables: const
## Sample size: 122
## Log Likelihood: 1396.308
## Roots of the characteristic polynomial:
## 0.5499 0.5499 0.5403 0.4365 0.4277 0.4277 0.2784 0.2784 0.02472
## Call:
## VAR(y = zt1, p = 3)
##
##
## Estimation results for equation UK:
## ===================================
## UK = UK.l1 + CANADA.l1 + US.l1 + UK.l2 + CANADA.l2 + US.l2 + UK.l3 + CANAD
```

```
A.l3 + US.l3 + const
##
##              Estimate Std. Error t value Pr(>|t|)
## UK.l1      0.4864358  0.0916259   5.309 5.64e-07 ***
## CANADA.l1  0.0066398  0.1011028   0.066   0.9478
## US.l1      0.0795650  0.0948968   0.838   0.4036
## UK.l2      0.0506837  0.1071296   0.473   0.6371
## CANADA.l2  0.1137187  0.1023646   1.111   0.2690
## US.l2     -0.0545881  0.1049639  -0.520   0.6040
## UK.l3     -0.0429276  0.0961119  -0.447   0.6560
## CANADA.l3 -0.2013135  0.0856433  -2.351   0.0205 *
## US.l3      0.1838910  0.0962684   1.910   0.0587 .
## const      0.0020982  0.0008373   2.506   0.0136 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005631 on 112 degrees of freedom
## Multiple R-Squared: 0.3712,  Adjusted R-squared: 0.3206
## F-statistic: 7.345 on 9 and 112 DF,  p-value: 2.347e-08
##
##
## Estimation results for equation CANADA:
## ========================================
## CANADA = UK.l1 + CANADA.l1 + US.l1 + UK.l2 + CANADA.l2 + US.l2 + UK.l3 + C
ANADA.l3 + US.l3 + const
##
##              Estimate Std. Error t value Pr(>|t|)
## UK.l1      0.4162191  0.0911422   4.567 1.28e-05 ***
## CANADA.l1  0.2774491  0.1005691   2.759  0.00678 **
## US.l1      0.4694936  0.0943958   4.974 2.39e-06 ***
## UK.l2     -0.1365129  0.1065641  -1.281  0.20283
## CANADA.l2 -0.1294277  0.1018242  -1.271  0.20633
## US.l2      0.0172531  0.1044098   0.165  0.86905
## UK.l3     -0.0186975  0.0956046  -0.196  0.84530
## CANADA.l3 -0.0532470  0.0851912  -0.625  0.53322
## US.l3      0.0503340  0.0957602   0.526  0.60019
## const      0.0001719  0.0008328   0.206  0.83688
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005602 on 112 degrees of freedom
## Multiple R-Squared: 0.5433,  Adjusted R-squared: 0.5066
## F-statistic:  14.8 on 9 and 112 DF,  p-value: 1.405e-15
##
##
## Estimation results for equation US:
## ====================================
## US = UK.l1 + CANADA.l1 + US.l1 + UK.l2 + CANADA.l2 + US.l2 + UK.l3 + CANAD
```

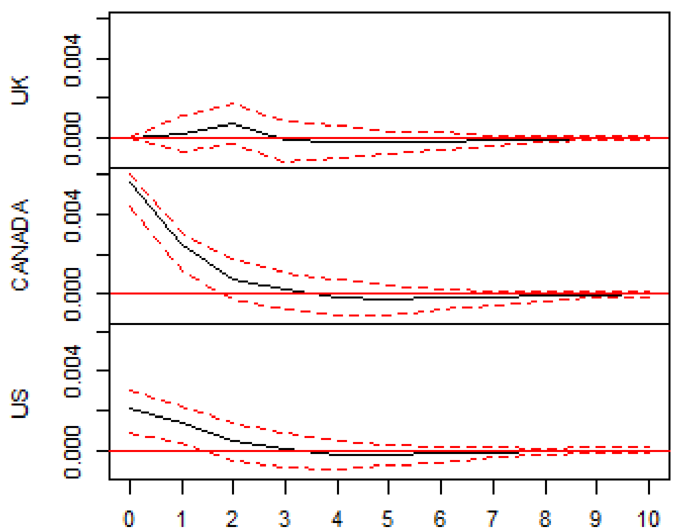
```
A.l3 + US.l3 + const
##
##              Estimate Std. Error t value Pr(>|t|)
## UK.l1      0.4228757  0.0972645   4.348 3.04e-05 ***
## CANADA.l1  0.1637679  0.1073245   1.526  0.12985
## US.l1      0.2098451  0.1007366   2.083  0.03952 *
## UK.l2     -0.1299347  0.1137222  -1.143  0.25566
## CANADA.l2 -0.1094521  0.1086639  -1.007  0.31598
## US.l2      0.1465159  0.1114232   1.315  0.19121
## UK.l3     -0.0022597  0.1020265  -0.022  0.98237
## CANADA.l3 -0.0831504  0.0909137  -0.915  0.36236
## US.l3      0.0554715  0.1021927   0.543  0.58834
## const      0.0024841  0.0008888   2.795  0.00611 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005978 on 112 degrees of freedom
## Multiple R-Squared: 0.3591,  Adjusted R-squared: 0.3076
## F-statistic: 6.973 on 9 and 112 DF,  p-value: 6.084e-08
##
##
##
## Covariance matrix of residuals:
##                UK    CANADA        US
## UK     3.171e-05 2.031e-06 7.029e-06
## CANADA 2.031e-06 3.138e-05 1.190e-05
## US     7.029e-06 1.190e-05 3.574e-05
##
## Correlation matrix of residuals:
##              UK  CANADA     US
## UK     1.00000 0.06438 0.2088
## CANADA 0.06438 1.00000 0.3555
## US     0.20880 0.35546 1.0000

impresp1=irf(varfit1)
plot(impresp1)
```

Orthogonal Impulse Response from UK

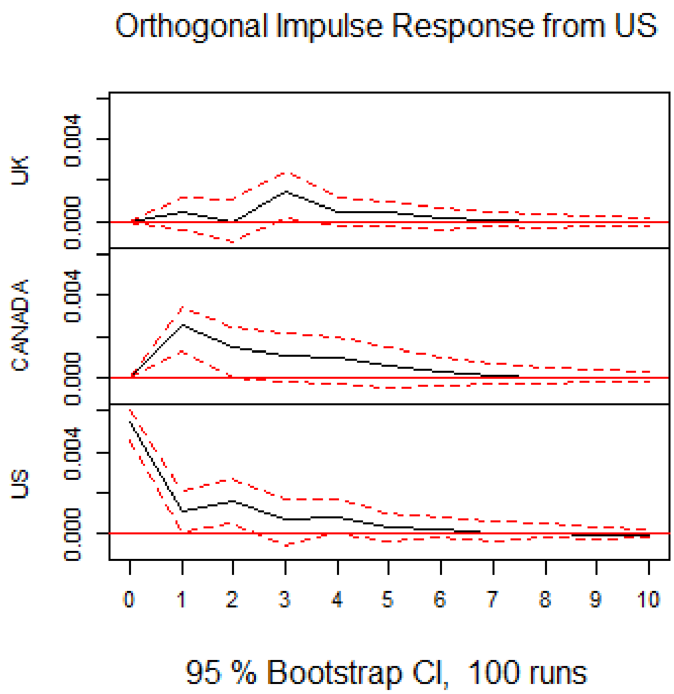


95 % Bootstrap CI, 100 runs

Orthogonal Impulse Response from CANADA



95 % Bootstrap CI, 100 runs



Impulse response function implication:

The shock from UK to itself is highly positive in the first lag, but the impact decreases gradually and becomes 0 in lag 7. The shocks to CAN and USA are positive with an increasing trend in the first lag, then continue to be positive but with a decreasing trend afterwards.

The shock from CAN to itself is highly positive in the first lag and gradually converges to 0. The impact on UK is almost insignificant. And the impact to the US is positive in the first 2 lags and becomes 0 afterwards.

The shock from US to itself is positive but with different magnitudes during the first 2 lags. Afterwards, it is still positive but gradually converges to 0. The impact to UK is almost nothing in the first 2 lags but becomes positive during lag 3 and 4. The impact to CAN is positive in lag 1 and then converges to 0.

```
#3.3
fevd(varfit1,n.ahead=5)

## $UK
##             UK     CANADA          US
## [1,] 1.0000000 0.00000000 0.000000000
## [2,] 0.9942559 0.00100098 0.004743151
## [3,] 0.9821887 0.01349583 0.004315496
## [4,] 0.9416841 0.01275012 0.045565776
## [5,] 0.9364240 0.01352974 0.050046298
##
```

```
## $CANADA
##                UK    CANADA        US
## [1,] 0.004144263 0.9958557 0.0000000
## [2,] 0.174089540 0.7023489 0.1235616
## [3,] 0.248482219 0.6085936 0.1429242
## [4,] 0.268139518 0.5776740 0.1541865
## [5,] 0.273377375 0.5624522 0.1641704
##
## $US
##               UK    CANADA        US
## [1,] 0.04359648 0.1174657 0.8389378
## [2,] 0.19191966 0.1301279 0.6779524
## [3,] 0.22480988 0.1199232 0.6552670
## [4,] 0.24583129 0.1151987 0.6389700
## [5,] 0.24872785 0.1136564 0.6376157
```

The forecast error variance decomposition for UK is that: the UK can be explained by itself by 93-100%. It can explain CAN by 0.1-1.3%, while it can explain US by 0.47-5%.

The forecast error variance decomposition for CAN is that: the UK can be explained by CAN by 0.4-27%. It can explain itself by 56-99%, while it can explain US by 12-16%.

The forecast error variance decomposition for US is that: the UK can be explained by US by 4.3-24%. It can explain CAN by 11.36-11.74-1.3%, while it can explain itself by 63-83%.

```
#3.4

m1=adfTest(UK,lags = 3, type = c("ct"), title = NULL,description = NULL)

## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used

m1@test$p.value

##
## 0.6213688

# cannot reject p-value so this represents unitroot
m1@test$parameter

## Lag Order
##         3

m1@test$lm

##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)
##
## Coefficients:
## (Intercept)      y.lag.1           tt  y.diff.lag1  y.diff.lag2  y.diff.la
```

```
g3
##  6103.04318     -0.02962     55.11977      0.64058      0.05568     -0.058
22

m2=adfTest(CAN,lags = 3, type = c("ct"), title = NULL,description = NULL)

## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used

m2@test$p.value

##
## 0.4687572

# cannot reject p-value so this represents unit root
m2@test$parameter

## Lag Order
##         3

m2@test$lm

##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)
##
## Coefficients:
## (Intercept)      y.lag.1           tt   y.diff.lag1  y.diff.lag2  y.diff.la
g3
##   6.956e+09   -3.406e-02    7.396e+07    6.363e-01   -1.307e-01    3.749e-
02

m3US=adfTest(USA,lags = 3, type = c("ct"), title = NULL,description = NULL)

## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used

m3US@test$p.value

##
## 0.3619283

# cannot reject p-value so this represents unit root
m3US@test$parameter

## Lag Order
##         3

m3US@test$lm
```

```
##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + tt + y.diff.lag)
##
## Coefficients:
## (Intercept)      y.lag.1           tt  y.diff.lag1  y.diff.lag2  y.diff.la
g3
##  313.642280    -0.047879     3.974854     0.356687     0.225453     0.0074
15

m4=adfTest(diff(UK),lags = 3, type = c("c"), title = NULL,description = NULL)

## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used

## Warning in adfTest(diff(UK), lags = 3, type = c("c"), title = NULL, descri
ption
## = NULL): p-value smaller than printed p-value

m4@test$p.value

##
## 0.01

m4@test$parameter

## Lag Order
##         3

m4@test$lm

##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)
##
## Coefficients:
## (Intercept)      y.lag.1  y.diff.lag1  y.diff.lag2  y.diff.lag3
##   757.00998     -0.44023      0.08259      0.12488      0.09996

m5=adfTest(diff(CAN),lags = 3, type = c("c"), title = NULL,description = NULL
)

## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used

## Warning in adfTest(diff(CAN), lags = 3, type = c("c"), title = NULL, descr
iption
## = NULL): p-value smaller than printed p-value

m5@test$p.value
```

```
##
## 0.01

m5@test$parameter

## Lag Order
##         3

m5@test$lm

##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)
##
## Coefficients:
## (Intercept)      y.lag.1  y.diff.lag1  y.diff.lag2  y.diff.lag3
##   8.719e+08   -4.711e-01    1.076e-01   -2.458e-02   -2.190e-02

m6=adfTest(diff(USA),lags = 3, type = c("c"), title = NULL,description = NULL
)

## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used

## Warning in adfTest(diff(USA), lags = 3, type = c("c"), title = NULL, descr
iption
## = NULL): p-value smaller than printed p-value

m6@test$p.value

##
## 0.01

m6@test$parameter

## Lag Order
##         3

m6@test$lm

##
## Call:
## lm(formula = y.diff ~ y.lag.1 + 1 + y.diff.lag)
##
## Coefficients:
## (Intercept)      y.lag.1  y.diff.lag1  y.diff.lag2  y.diff.lag3
##   33.765317    -0.465235    -0.194398     0.012604     0.002964

fit=lm(UK~USA)
summary(fit)
```

```
##
## Call:
## lm(formula = UK ~ USA)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -7069  -3451  -1108   3421  10096
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.039e+04  1.516e+03   26.65   <2e-16 ***
## USA         2.345e+01  1.307e-01  179.45   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4405 on 124 degrees of freedom
## Multiple R-squared:  0.9962, Adjusted R-squared:  0.9961
## F-statistic: 3.22e+04 on 1 and 124 DF,  p-value: < 2.2e-16
```

```
error=residuals(fit)
m7=adfTest(error, lags = 3, type =c("nc"), title = NULL, description = NULL)
```

```
## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used
```

```
## Warning in adfTest(error, lags = 3, type = c("nc"), title = NULL, descript
ion =
## NULL): p-value smaller than printed p-value
```

```
m7@test$p.value
```

```
##
## 0.01
```

```
fit2=lm(USA~CAN)
summary(fit2)
```

```
##
## Call:
## lm(formula = USA ~ CAN)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -510.72 -191.96  -44.38  185.44  505.35
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.357e+03  9.772e+01  -13.89   <2e-16 ***
## CAN          3.957e-08  2.995e-10  132.12   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254.2 on 124 degrees of freedom
## Multiple R-squared:  0.9929, Adjusted R-squared:  0.9929
## F-statistic: 1.746e+04 on 1 and 124 DF,  p-value: < 2.2e-16
```

```
error2=residuals(fit2)
m8=adfTest(error2, lags = 3, type =c("nc"), title = NULL, description = NULL)
```

```
## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used
```

```
m8@test$p.value
```

```
##
## 0.1990854
```

```
fit3=lm(CAN~UK)
summary(fit3)
```

```
##
## Call:
## lm(formula = CAN ~ UK)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.011e+10 -4.791e+09 -1.348e+09  3.626e+09  1.794e+10
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -6.288e+09  2.534e+09  -2.482   0.0144 *
## UK           1.068e+06  8.142e+03 131.175   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.448e+09 on 124 degrees of freedom
## Multiple R-squared:  0.9928, Adjusted R-squared:  0.9928
## F-statistic: 1.721e+04 on 1 and 124 DF,  p-value: < 2.2e-16
```

```
error3=residuals(fit3)
m9=adfTest(error3, lags = 3, type =c("nc"), title = NULL, description = NULL)
```

```
## Warning in if (class(x) == "timeSeries") x = series(x): the condition has
length
## > 1 and only the first element will be used
```

```
m9@test$p.value
```

```
##
## 0.06847904
```

Cointegrating vectors: 1

There are only 1 cointegrating vector, which is between UK and USA. According to the estimation re sult using Engle-Granger, the two series are cointegrated at 0.05 level of significance.

H_0 : e_t has unit root (no cointegration)

H_a : e_t has no unit root (cointegration)

```
#3.5

diff.uk=diff(UK)
diff.uk=diff.uk[2:126]
diff.canada=diff(CAN)
diff.canada=diff.canada[2:126]
diff.us=diff(USA)
diff.us=diff.us[2:126]

diff.uk.L.1=Lag(diff(UK),k=1)
diff.uk.L.1=diff.uk.L.1[2:126]
diff.canada.L.1=Lag(diff(CAN),k=1)
diff.canada.L.1=diff.canada.L.1[2:126]
diff.us.L.1=Lag(diff(USA),k=1)
diff.us.L.1=diff.us.L.1[2:126]

error.L.1 =Lag(error,k=1)
error.L.1=error.L.1[2:126]

error2.L.1=Lag(error2,k=1)
error2.L.1=error2.L.1[2:126]

error3.L.1=Lag(error3,k=1)
error3.L.1=error3.L.1[2:126]

fit4=lm(diff.uk~diff.uk.L.1+diff.us.L.1+diff.canada.L.1+error.L.1)
summary(fit4)

##
## Call:
## lm(formula = diff.uk ~ diff.uk.L.1 + diff.us.L.1 + diff.canada.L.1 +
##     error.L.1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5285.7  -764.1    47.0  1096.9  3870.3
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      5.526e+02  1.986e+02   2.782  0.00628 **
```

```
## diff.uk.L.1      5.978e-01  8.106e-02   7.375  2.4e-11 ***
## diff.us.L.1      4.372e+00  2.449e+00   1.785  0.07676 .
## diff.canada.L.1 -8.982e-08  7.627e-08  -1.178  0.24128
## error.L.1       -1.088e-01  3.398e-02  -3.201  0.00176 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1589 on 119 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4809, Adjusted R-squared:  0.4635
## F-statistic: 27.57 on 4 and 119 DF,  p-value: 3.365e-16
```

```
fit5=lm(diff.us~diff.us.L.1+diff.uk.L.1+diff.canada.L.1+error2.L.1)
summary(fit5)
```

```
##
## Call:
## lm(formula = diff.us ~ diff.us.L.1 + diff.uk.L.1 + diff.canada.L.1 +
##     error2.L.1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -255.487  -35.905    5.009   41.511  140.438
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.696e+01  8.170e+00   4.524 1.44e-05 ***
## diff.us.L.1     5.037e-02  1.031e-01   0.489   0.6259
## diff.uk.L.1     1.708e-02  3.176e-03   5.379 3.80e-07 ***
## diff.canada.L.1 2.607e-09  3.040e-09   0.858   0.3928
## error2.L.1      4.592e-02  2.506e-02   1.832   0.0695 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 63.53 on 119 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.3775, Adjusted R-squared:  0.3565
## F-statistic: 18.04 on 4 and 119 DF,  p-value: 1.328e-11
```

```
fit6=lm(diff.canada~diff.canada.L.1+diff.uk.L.1+diff.us.L.1+error3.L.1)
summary(fit6)
```

```
##
## Call:
## lm(formula = diff.canada ~ diff.canada.L.1 + diff.uk.L.1 + diff.us.L.1 +
##     error3.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.577e+09 -1.010e+09 -2.078e+08  1.016e+09  3.946e+09
##
```

```
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       2.196e+08  2.141e+08   1.026 0.307070
## diff.canada.L.1   2.128e-01  8.301e-02   2.564 0.011592 *
## diff.uk.L.1       3.400e+05  8.686e+04   3.914 0.000152 ***
## diff.us.L.1       1.002e+07  2.668e+06   3.755 0.000270 ***
## error3.L.1       -1.806e-02  2.680e-02  -0.674 0.501761
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.691e+09 on 119 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.538,  Adjusted R-squared:  0.5225
## F-statistic: 34.64 on 4 and 119 DF,  p-value: < 2.2e-16
```

Fitted model:

$\Delta UK_t = 5.526e{+}02 + 5.978e{-}01\Delta UK_{t-1} + 4.372e{+}00\Delta USA_{t-1} - 8.982e{-}08\Delta CAN_{t-1} - 1.088e{-}01e_{t-1}$

(1.986e+02) (8.106e-02) (2.449e+00) (7.627e-08) (3.398e-02)

$\Delta CAN_t = 3.696e{+}01 + 5.037e{-}02\Delta US_{t-1} + 1.708e{-}02\Delta UK_{t-1} + 2.607e{-}09\Delta CAN_{t-1} + 4.592e{-}02e_{t-1}$

(8.170e+00) (1.031e-01) (3.176e-03) (3.04e-09) (2.506e-02)

$\Delta USA_t = 2.1996e{+}08 + 2.128e{-}01\Delta CAN_{t-1} + 3.4e{+}05\Delta UK_{t-1} + 1.002e{+}01\Delta USA_{t-1} - 1.806e{-}02e_{t-1}$

(2.141e+08) (8.301e-02) (8.686e+04) (2.668e+06) (2.680e-02)

Implications:

UK: The higher GDP of USA has positive value towards the UK, but the increase in GDP of Canada ha s negative value on the UK'S GDP. The shock from last period is going to decrease the value of GDP.

CAN: The higher GDP of USA and UK has positive value towards Canada, but the shock from last per iod has positive influence toward the GDP of Canada.

USA: The higher GDP of CAN and UK has positive value towards the USA, but the shock from last pe riod has negative influence toward the GDP of USA.