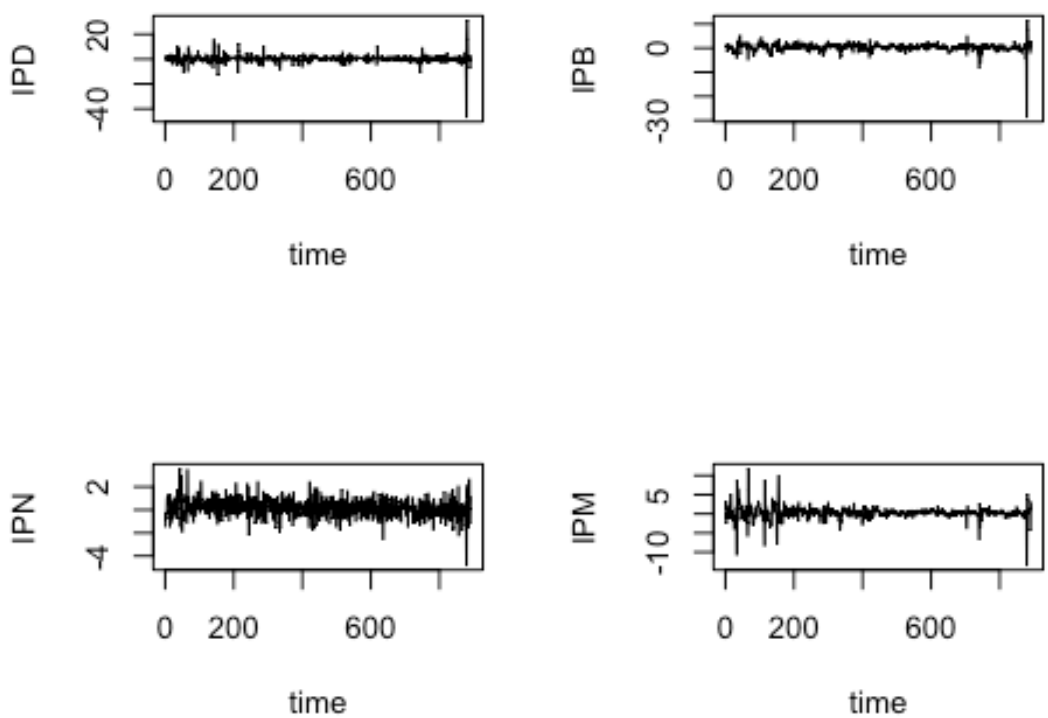
to_on

2021-06-10

Question 1

```
## Attaching package: 'MTS'

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

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.n
umeric(IPMAT[-c(1:96)]))
dim(IP)

## [1] 892   4

colnames(IP)<-c("IPD","IPN","IPB","IPM")
#1.1
IP=log(IP)
zt=diffM(IP)*100
MTSplot(zt)
```




```
#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,6)
```

```
## Constant term:
## Estimates:  0.2040297 0.1573752 0.1125976 0.1053311
## Std.Error:  0.1218999 0.03035209 0.06279805 0.05838232
## AR coefficient matrix
## AR( 1 )-matrix
##        [,1]    [,2]    [,3]    [,4]
## [1,] 0.0420  0.2551 -0.0533 0.36428
## [2,] 0.0333 -0.1817  0.0139 0.00113
## [3,] 0.0182  0.0786  0.0321 0.20610
## [4,] 0.0212  0.1377  0.0492 0.18413
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0503 0.1472 0.1004 0.0859
## [2,] 0.0125 0.0366 0.0250 0.0214
## [3,] 0.0259 0.0758 0.0517 0.0442
## [4,] 0.0241 0.0705 0.0481 0.0411
## AR( 2 )-matrix
##          [,1]    [,2]     [,3]     [,4]
## [1,] -0.23369  0.2624 -0.09357  0.12629
## [2,]  0.00238 -0.0443 -0.00387 -0.00966
## [3,] -0.13653  0.0929  0.12781  0.10189
## [4,]  0.00752  0.1451  0.06589 -0.10850
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0500 0.1497 0.0995 0.0856
## [2,] 0.0124 0.0373 0.0248 0.0213
## [3,] 0.0258 0.0771 0.0512 0.0441
## [4,] 0.0239 0.0717 0.0476 0.0410
## AR( 3 )-matrix
##          [,1]   [,2]    [,3]     [,4]
```

```
## [1,] -0.1374 0.1874 -0.0237  0.12870
## [2,]  0.0226 0.0283 -0.0339 -0.00202
## [3,] -0.1140 0.1101  0.1308  0.13163
## [4,] -0.0127 0.2006  0.0167  0.00225
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0513 0.1500 0.1005 0.0867
## [2,] 0.0128 0.0373 0.0250 0.0216
## [3,] 0.0264 0.0773 0.0518 0.0447
## [4,] 0.0246 0.0718 0.0481 0.0415
## AR( 4 )-matrix
##          [,1]   [,2]    [,3]     [,4]
## [1,] -0.04052 0.2162 -0.1612  0.20930
## [2,]  0.00597 0.0843 -0.0284 -0.00111
## [3,] -0.02123 0.1047 -0.0745  0.12391
## [4,]  0.04583 0.0957 -0.0388  0.01972
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0514 0.1501 0.0995 0.0864
## [2,] 0.0128 0.0374 0.0248 0.0215
## [3,] 0.0265 0.0773 0.0513 0.0445
## [4,] 0.0246 0.0719 0.0477 0.0414
## AR( 5 )-matrix
##          [,1]     [,2]     [,3]   [,4]
## [1,] -0.01366  0.08590 -0.24443  0.063
## [2,] -0.00685  0.04387 -0.00165  0.044
## [3,] -0.02941 -0.00504 -0.04715  0.108
## [4,]  0.05632  0.02542 -0.02394 -0.159
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0494 0.1508 0.0976 0.0847
## [2,] 0.0123 0.0375 0.0243 0.0211
## [3,] 0.0255 0.0777 0.0503 0.0436
## [4,] 0.0237 0.0722 0.0467 0.0406
## AR( 6 )-matrix
##        [,1]    [,2]    [,3]    [,4]
## [1,] 0.1364 -0.0678 -0.3126  0.0537
## [2,] 0.0198  0.0239  0.0240 -0.0172
## [3,] 0.0122 -0.1053  0.0270  0.0320
## [4,] 0.0521 -0.0438 -0.0266 -0.0106
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0494 0.1471 0.0977 0.0857
## [2,] 0.0123 0.0366 0.0243 0.0213
## [3,] 0.0254 0.0758 0.0503 0.0442
## [4,] 0.0237 0.0704 0.0468 0.0411
##
## Residuals cov-mtx:
##           [,1]      [,2]      [,3]      [,4]
## [1,] 9.5558929 0.8261288 3.5834514 2.2928640
```

```
## [2,] 0.8261288 0.5924362 0.4307313 0.3083878
## [3,] 3.5834514 0.4307313 2.5360441 1.2557605
## [4,] 2.2928640 0.3083878 1.2557605 2.1919322
##
## det(SSE) =  8.698098
## AIC =  2.378593
## BIC =  2.894939
## HQ  =  2.575936
```

```
#MTSdiag(m1)
m2=refVAR(m1,thres=1.645)
```

```
## Constant term:
## Estimates:  0.2985377 0.157489 0.1542269 0.1081254
## Std.Error:  0.1122942 0.02740226 0.0568169 0.05440154
## AR coefficient matrix
## AR( 1 )-matrix
##        [,1]   [,2] [,3]  [,4]
## [1,] 0.0000  0.226    0 0.417
## [2,] 0.0358 -0.166    0 0.000
## [3,] 0.0340  0.000    0 0.225
## [4,] 0.0395  0.145    0 0.200
## standard error
##         [,1]   [,2] [,3]   [,4]
## [1,] 0.00000 0.1370    0 0.0701
## [2,] 0.00836 0.0348    0 0.0000
## [3,] 0.01982 0.0000    0 0.0413
## [4,] 0.01832 0.0687    0 0.0381
## AR( 2 )-matrix
##        [,1]  [,2]   [,3]   [,4]
## [1,] -0.213 0.000 0.0000  0.000
## [2,]  0.000 0.000 0.0000  0.000
## [3,] -0.129 0.000 0.1374  0.105
## [4,]  0.000 0.141 0.0737 -0.100
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0337 0.0000 0.0000 0.0000
## [2,] 0.0000 0.0000 0.0000 0.0000
## [3,] 0.0251 0.0000 0.0484 0.0431
## [4,] 0.0000 0.0698 0.0360 0.0390
## AR( 3 )-matrix
##        [,1]  [,2]  [,3]  [,4]
## [1,] -0.131 0.000 0.000 0.138
## [2,]  0.000 0.000 0.000 0.000
## [3,] -0.113 0.000 0.149 0.134
## [4,]  0.000 0.185 0.000 0.000
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0379 0.0000 0.0000 0.0804
## [2,] 0.0000 0.0000 0.0000 0.0000
```

```
## [3,] 0.0245 0.0000 0.0478 0.0439
## [4,] 0.0000 0.0654 0.0000 0.0000
## AR( 4 )-matrix
##        [,1]   [,2]    [,3]  [,4]
## [1,] 0.0000 0.0000 -0.1674 0.187
## [2,] 0.0000 0.0616  0.0000 0.000
## [3,] 0.0000 0.0000 -0.0827 0.117
## [4,] 0.0443 0.0000  0.0000 0.000
## standard error
##        [,1]   [,2]   [,3]   [,4]
## [1,] 0.0000 0.0000 0.0746 0.0811
## [2,] 0.0000 0.0329 0.0000 0.0000
## [3,] 0.0000 0.0000 0.0389 0.0422
## [4,] 0.0163 0.0000 0.0000 0.0000
## AR( 5 )-matrix
##         [,1] [,2]   [,3]    [,4]
## [1,]  0.0000    0 -0.228  0.0000
## [2,]  0.0000    0  0.000  0.0305
## [3,] -0.0407    0  0.000  0.1049
## [4,]  0.0524    0  0.000 -0.1676
## standard error
##        [,1] [,2]   [,3]   [,4]
## [1,] 0.0000    0 0.0658 0.0000
## [2,] 0.0000    0 0.0000 0.0169
## [3,] 0.0198    0 0.0000 0.0417
## [4,] 0.0177    0 0.0000 0.0379
## AR( 6 )-matrix
##        [,1] [,2]   [,3] [,4]
## [1,] 0.1592    0 -0.326    0
## [2,] 0.0255    0  0.000    0
## [3,] 0.0000    0  0.000    0
## [4,] 0.0391    0  0.000    0
## standard error
##         [,1] [,2]   [,3] [,4]
## [1,] 0.04560    0 0.0869    0
## [2,] 0.00807    0 0.0000    0
## [3,] 0.00000    0 0.0000    0
## [4,] 0.01600    0 0.0000    0
## 
## Residuals cov-mtx:
##          [,1]      [,2]      [,3]      [,4]
## [1,] 9.672303 0.8232940 3.6137189 2.3002478
## [2,] 0.823294 0.6005646 0.4296673 0.3098108
## [3,] 3.613719 0.4296673 2.5704582 1.2639624
## [4,] 2.300248 0.3098108 1.2639624 2.2059227
## 
## det(SSE) =  9.263687 
## AIC =  2.3114 
## BIC =  2.515787 
## HQ  =  2.389515
```

```
#MTSdiag(m2)


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

varfit1=VAR(zt, p=6)
summary(varfit1)

##
## VAR Estimation Results:
## =========================
## Endogenous variables: IPD, IPN, IPB, IPM
## Deterministic variables: const
## Sample size: 885
## Log Likelihood: -5980.216
## Roots of the characteristic polynomial:
## 0.8157 0.8157 0.7766 0.7766 0.745 0.745 0.7311 0.7311 0.7043 0.6979 0.6979
0.6954 0.6954 0.6882 0.6882 0.6726 0.6726 0.6682 0.6335 0.6335 0.6274 0.6274
0.6055 0.09344
## Call:
## VAR(y = zt, p = 6)
##
##
## Estimation results for equation IPD:
## ====================================
## IPD = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + IPD.l4 + IPN.l4 + IPB.l4 + IPM.l4 + I
PD.l5 + IPN.l5 + IPB.l5 + IPM.l5 + IPD.l6 + IPN.l6 + IPB.l6 + IPM.l6 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.04198    0.05031   0.834  0.40426
## IPN.l1  0.25508    0.14719   1.733  0.08345 .
## IPB.l1 -0.05333    0.10037  -0.531  0.59528
## IPM.l1  0.36428    0.08589   4.241 2.46e-05 ***
## IPD.l2 -0.23369    0.04999  -4.675 3.42e-06 ***
## IPN.l2  0.26244    0.14966   1.754  0.07987 .
## IPB.l2 -0.09357    0.09945  -0.941  0.34702
## IPM.l2  0.12629    0.08559   1.476  0.14041
```

```
## IPD.l3 -0.13740    0.05127  -2.680  0.00750 **
## IPN.l3  0.18739    0.14996   1.250  0.21181
## IPB.l3 -0.02369    0.10051  -0.236  0.81374
## IPM.l3  0.12870    0.08668   1.485  0.13800
## IPD.l4 -0.04052    0.05137  -0.789  0.43041
## IPN.l4  0.21624    0.15008   1.441  0.15000
## IPB.l4 -0.16121    0.09950  -1.620  0.10557
## IPM.l4  0.20930    0.08643   2.422  0.01566 *
## IPD.l5 -0.01366    0.04943  -0.276  0.78237
## IPN.l5  0.08590    0.15079   0.570  0.56905
## IPB.l5 -0.24443    0.09756  -2.505  0.01242 *
## IPM.l5  0.06298    0.08469   0.744  0.45732
## IPD.l6  0.13638    0.04939   2.761  0.00588 **
## IPN.l6 -0.06783    0.14705  -0.461  0.64474
## IPB.l6 -0.31255    0.09766  -3.201  0.00142 **
## IPM.l6  0.05373    0.08573   0.627  0.53100
## const   0.20403    0.12190   1.674  0.09454 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 3.136 on 860 degrees of freedom
## Multiple R-Squared: 0.1296,  Adjusted R-squared: 0.1053
## F-statistic: 5.336 on 24 and 860 DF,  p-value: 6.753e-15
##
##
## Estimation results for equation IPN:
## ====================================
## IPN = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + IPD.l4 + IPN.l4 + IPB.l4 + IPM.l4 + I
PD.l5 + IPN.l5 + IPB.l5 + IPM.l5 + IPD.l6 + IPN.l6 + IPB.l6 + IPM.l6 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.033278   0.012527   2.656  0.00804 **
## IPN.l1 -0.181707   0.036648  -4.958 8.57e-07 ***
## IPB.l1  0.013927   0.024990   0.557  0.57748
## IPM.l1  0.001125   0.021386   0.053  0.95805
## IPD.l2  0.002378   0.012447   0.191  0.84850
## IPN.l2 -0.044255   0.037265  -1.188  0.23534
## IPB.l2 -0.003874   0.024762  -0.156  0.87571
## IPM.l2 -0.009664   0.021310  -0.454  0.65029
## IPD.l3  0.022594   0.012765   1.770  0.07708 .
## IPN.l3  0.028276   0.037340   0.757  0.44910
## IPB.l3 -0.033857   0.025026  -1.353  0.17645
## IPM.l3 -0.002022   0.021584  -0.094  0.92538
## IPD.l4  0.005968   0.012790   0.467  0.64090
## IPN.l4  0.084347   0.037369   2.257  0.02425 *
## IPB.l4 -0.028420   0.024776  -1.147  0.25166
## IPM.l4 -0.001115   0.021521  -0.052  0.95870
## IPD.l5 -0.006847   0.012308  -0.556  0.57818
```

```
## IPN.l5  0.043870   0.037546   1.168  0.24295
## IPB.l5 -0.001649   0.024293  -0.068  0.94591
## IPM.l5  0.043986   0.021087   2.086  0.03728 *
## IPD.l6  0.019753   0.012297   1.606  0.10858
## IPN.l6  0.023908   0.036615   0.653  0.51396
## IPB.l6  0.023983   0.024316   0.986  0.32425
## IPM.l6 -0.017243   0.021347  -0.808  0.41945
## const   0.157375   0.030352   5.185 2.69e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.7808 on 860 degrees of freedom
## Multiple R-Squared: 0.06574, Adjusted R-squared: 0.03967
## F-statistic: 2.522 on 24 and 860 DF,  p-value: 8.207e-05
##
##
## Estimation results for equation IPB:
## ====================================
## IPB = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + IPD.l4 + IPN.l4 + IPB.l4 + IPM.l4 + I
PD.l5 + IPN.l5 + IPB.l5 + IPM.l5 + IPD.l6 + IPN.l6 + IPB.l6 + IPM.l6 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.018176   0.025918   0.701  0.48332
## IPN.l1  0.078606   0.075825   1.037  0.30017
## IPB.l1  0.032099   0.051705   0.621  0.53489
## IPM.l1  0.206097   0.044246   4.658 3.70e-06 ***
## IPD.l2 -0.136527   0.025753  -5.301 1.46e-07 ***
## IPN.l2  0.092895   0.077101   1.205  0.22860
## IPB.l2  0.127808   0.051233   2.495  0.01280 *
## IPM.l2  0.101888   0.044090   2.311  0.02107 *
## IPD.l3 -0.114010   0.026411  -4.317 1.77e-05 ***
## IPN.l3  0.110077   0.077256   1.425  0.15457
## IPB.l3  0.130807   0.051778   2.526  0.01171 *
## IPM.l3  0.131626   0.044656   2.948  0.00329 **
## IPD.l4 -0.021233   0.026462  -0.802  0.42254
## IPN.l4  0.104686   0.077316   1.354  0.17609
## IPB.l4 -0.074467   0.051260  -1.453  0.14667
## IPM.l4  0.123911   0.044526   2.783  0.00551 **
## IPD.l5 -0.029409   0.025465  -1.155  0.24848
## IPN.l5 -0.005043   0.077681  -0.065  0.94826
## IPB.l5 -0.047148   0.050262  -0.938  0.34848
## IPM.l5  0.107847   0.043630   2.472  0.01363 *
## IPD.l6  0.012195   0.025442   0.479  0.63185
## IPN.l6 -0.105321   0.075756  -1.390  0.16481
## IPB.l6  0.027047   0.050309   0.538  0.59097
## IPM.l6  0.032039   0.044167   0.725  0.46840
## const   0.112598   0.062798   1.793  0.07332 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.615 on 860 degrees of freedom
## Multiple R-Squared: 0.1681,  Adjusted R-squared: 0.1448
## F-statistic: 7.238 on 24 and 860 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 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + IPD.l4 + IPN.l4 + IPB.l4 + IPM.l4 + I
PD.l5 + IPN.l5 + IPB.l5 + IPM.l5 + IPD.l6 + IPN.l6 + IPB.l6 + IPM.l6 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.021225   0.024096   0.881  0.37864
## IPN.l1  0.137728   0.070493   1.954  0.05105 .
## IPB.l1  0.049229   0.048069   1.024  0.30606
## IPM.l1  0.184129   0.041135   4.476 8.62e-06 ***
## IPD.l2  0.007522   0.023942   0.314  0.75346
## IPN.l2  0.145057   0.071680   2.024  0.04331 *
## IPB.l2  0.065891   0.047631   1.383  0.16691
## IPM.l2 -0.108503   0.040990  -2.647  0.00827 **
## IPD.l3 -0.012702   0.024554  -0.517  0.60508
## IPN.l3  0.200617   0.071824   2.793  0.00534 **
## IPB.l3  0.016737   0.048137   0.348  0.72816
## IPM.l3  0.002253   0.041516   0.054  0.95673
## IPD.l4  0.045827   0.024601   1.863  0.06283 .
## IPN.l4  0.095746   0.071880   1.332  0.18320
## IPB.l4 -0.038801   0.047656  -0.814  0.41576
## IPM.l4  0.019716   0.041395   0.476  0.63399
## IPD.l5  0.056317   0.023675   2.379  0.01759 *
## IPN.l5  0.025416   0.072219   0.352  0.72498
## IPB.l5 -0.023939   0.046727  -0.512  0.60857
## IPM.l5 -0.159069   0.040562  -3.922 9.49e-05 ***
## IPD.l6  0.052073   0.023653   2.201  0.02797 *
## IPN.l6 -0.043805   0.070429  -0.622  0.53412
## IPB.l6 -0.026646   0.046772  -0.570  0.56903
## IPM.l6 -0.010648   0.041061  -0.259  0.79546
## const   0.105331   0.058382   1.804  0.07156 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.502 on 860 degrees of freedom
## Multiple R-Squared: 0.1301,  Adjusted R-squared: 0.1058
## F-statistic: 5.358 on 24 and 860 DF,  p-value: 5.578e-15
##
##
##
```

```
## Covariance matrix of residuals:
##        IPD    IPN    IPB    IPM
## IPD 9.8337 0.8501 3.6876 2.3595
## IPN 0.8501 0.6097 0.4433 0.3174
## IPB 3.6876 0.4433 2.6098 1.2923
## IPM 2.3595 0.3174 1.2923 2.2557
##
## Correlation matrix of residuals:
##        IPD    IPN    IPB    IPM
## IPD 1.0000 0.3472 0.7279 0.5010
## IPN 0.3472 1.0000 0.3514 0.2706
## IPB 0.7279 0.3514 1.0000 0.5326
## IPM 0.5010 0.2706 0.5326 1.0000
```

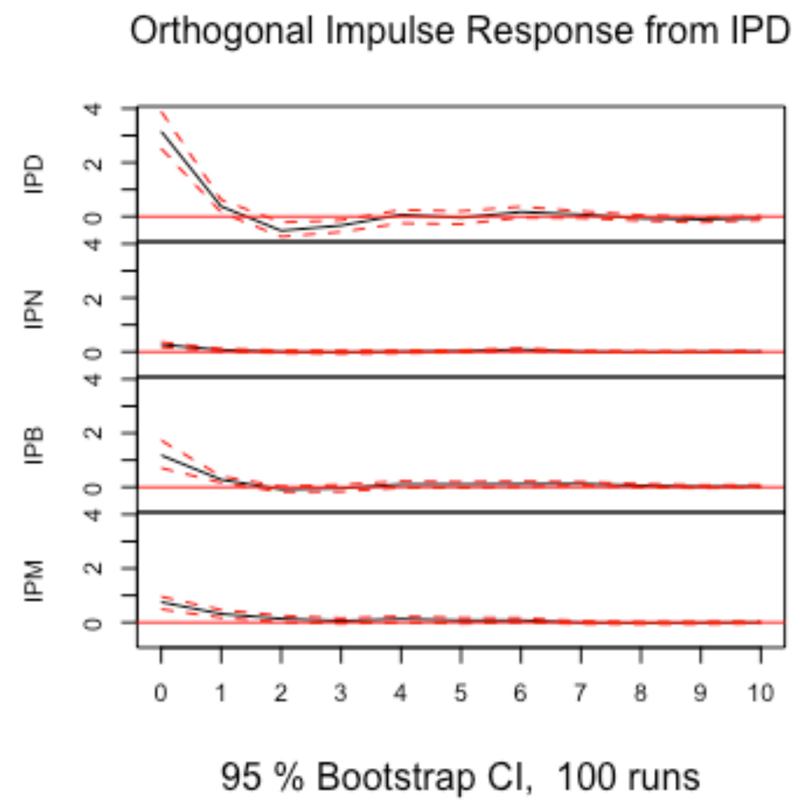
```
varfit2<-restrict(varfit1,thresh=1.645)
summary(varfit2)
```

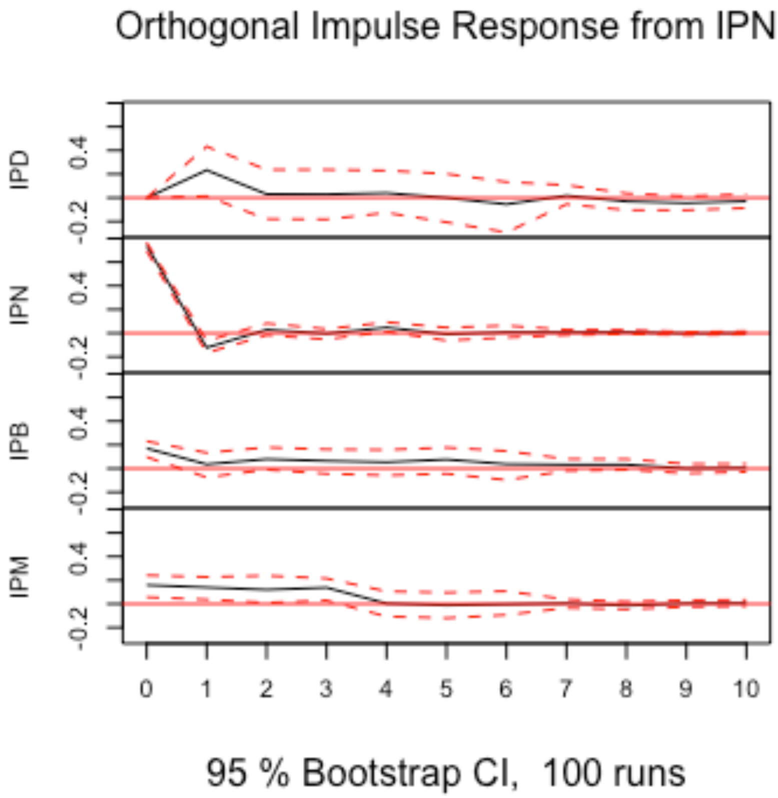
```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: IPD, IPN, IPB, IPM
## Deterministic variables: const
## Sample size: 885
## Log Likelihood: -6008.093
## Roots of the characteristic polynomial:
## 0.7922 0.7922 0.7889 0.7889 0.7355 0.7355 0.7336 0.7336 0.7264 0.7264 0.68
09 0.6809 0.573 0.573 0.5636 0.5636 0.5618 0.5618 0.5546 0.5546 0.2921     0
0     0
## Call:
## VAR(y = zt, p = 6)
##
##
## Estimation results for equation IPD:
## ====================================
## IPD = IPN.l1 + IPM.l1 + IPD.l2 + IPD.l3 + IPM.l3 + IPB.l4 + IPM.l4 + IPB.l
5 + IPD.l6 + IPB.l6 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## IPN.l1  0.22550    0.13698   1.646 0.100080
## IPM.l1  0.41737    0.07015   5.950 3.88e-09 ***
## IPD.l2 -0.21280    0.03365  -6.323 4.08e-10 ***
## IPD.l3 -0.13134    0.03791  -3.464 0.000558 ***
## IPM.l3  0.13768    0.08038   1.713 0.087078 .
## IPB.l4 -0.16739    0.07461  -2.243 0.025117 *
## IPM.l4  0.18714    0.08114   2.307 0.021315 *
## IPB.l5 -0.22831    0.06580  -3.470 0.000547 ***
## IPD.l6  0.15924    0.04560   3.492 0.000503 ***
## IPB.l6 -0.32615    0.08691  -3.753 0.000186 ***
## const   0.29854    0.11229   2.659 0.007992 **
## ---
```

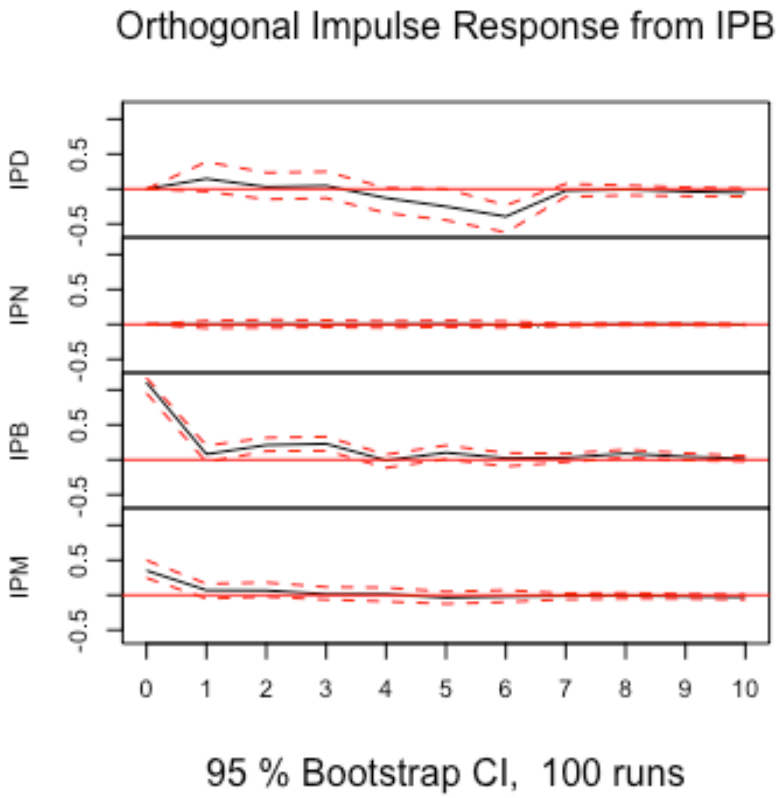
```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 3.13 on 874 degrees of freedom
## Multiple R-Squared: 0.1235,  Adjusted R-squared: 0.1124
## F-statistic: 11.19 on 11 and 874 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation IPN:
## ====================================
## IPN = IPD.l1 + IPN.l1 + IPN.l4 + IPM.l5 + IPD.l6 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.035802   0.008362   4.282 2.06e-05 ***
## IPN.l1 -0.166227   0.034823  -4.774 2.12e-06 ***
## IPN.l4  0.061584   0.032901   1.872  0.06157 .
## IPM.l5  0.030534   0.016868   1.810  0.07060 .
## IPD.l6  0.025487   0.008071   3.158  0.00164 **
## const   0.157489   0.027402   5.747 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.7776 on 879 degrees of freedom
## Multiple R-Squared: 0.09067, Adjusted R-squared: 0.08446
## F-statistic: 14.61 on 6 and 879 DF,  p-value: 6.039e-16
##
##
## Estimation results for equation IPB:
## ====================================
## IPB = IPD.l1 + IPM.l1 + IPD.l2 + IPB.l2 + IPM.l2 + IPD.l3 + IPB.l3 + IPM.l
3 + IPB.l4 + IPM.l4 + IPD.l5 + IPM.l5 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.03402    0.01982   1.717  0.08639 .
## IPM.l1  0.22486    0.04129   5.445 6.73e-08 ***
## IPD.l2 -0.12949    0.02506  -5.168 2.93e-07 ***
## IPB.l2  0.13741    0.04837   2.841  0.00460 **
## IPM.l2  0.10522    0.04305   2.444  0.01473 *
## IPD.l3 -0.11282    0.02455  -4.596 4.95e-06 ***
## IPB.l3  0.14926    0.04783   3.121  0.00186 **
## IPM.l3  0.13392    0.04393   3.048  0.00237 **
## IPB.l4 -0.08267    0.03886  -2.128  0.03365 *
## IPM.l4  0.11748    0.04220   2.784  0.00548 **
## IPD.l5 -0.04073    0.01977  -2.061  0.03961 *
## IPM.l5  0.10487    0.04167   2.517  0.01202 *
## const   0.15423    0.05682   2.714  0.00677 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

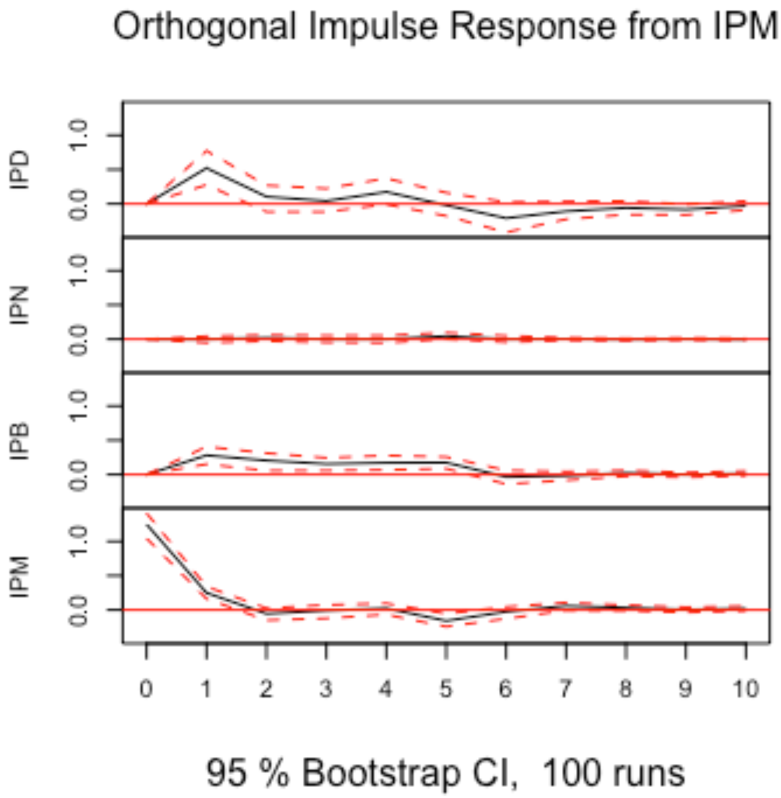
```
##
## Residual standard error: 1.615 on 872 degrees of freedom
## Multiple R-Squared: 0.1838,  Adjusted R-squared: 0.1716
## F-statistic:  15.1 on 13 and 872 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation IPM:
## ====================================
## IPM = IPD.l1 + IPN.l1 + IPM.l1 + IPN.l2 + IPB.l2 + IPM.l2 + IPN.l3 + IPD.l
4 + IPD.l5 + IPM.l5 + IPD.l6 + const
##
##        Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.03951    0.01832   2.157  0.03126 *
## IPN.l1  0.14507    0.06866   2.113  0.03489 *
## IPM.l1  0.19985    0.03806   5.251 1.90e-07 ***
## IPN.l2  0.14065    0.06977   2.016  0.04411 *
## IPB.l2  0.07372    0.03603   2.046  0.04108 *
## IPM.l2 -0.10017    0.03898  -2.570  0.01034 *
## IPN.l3  0.18455    0.06539   2.822  0.00488 **
## IPD.l4  0.04426    0.01628   2.718  0.00670 **
## IPD.l5  0.05241    0.01771   2.959  0.00317 **
## IPM.l5 -0.16758    0.03786  -4.426 1.08e-05 ***
## IPD.l6  0.03905    0.01600   2.441  0.01485 *
## const   0.10813    0.05440   1.988  0.04717 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 1.495 on 873 degrees of freedom
## Multiple R-Squared: 0.1432,  Adjusted R-squared: 0.1314
## F-statistic: 12.16 on 12 and 873 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##        IPD    IPN    IPB    IPM
## IPD 9.9535 0.8472 3.7188 2.3671
## IPN 0.8472 0.6180 0.4422 0.3188
## IPB 3.7188 0.4422 2.6452 1.3007
## IPM 2.3671 0.3188 1.3007 2.2700
##
## Correlation matrix of residuals:
##        IPD    IPN    IPB    IPM
## IPD 1.0000 0.3416 0.7247 0.4980
## IPN 0.3416 1.0000 0.3458 0.2692
## IPB 0.7247 0.3458 1.0000 0.5308
## IPM 0.4980 0.2692 0.5308 1.0000
```

```
impresp=irf(varfit2)
plot(impresp)
```


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


Orthogonal Impulse Response from IPN
IPD
IPN
IPB
IPM
-0.2
0.4
0
1
2
3
4
5
6
7
8
9
10
95 % Bootstrap CI, 100 runs


Orthogonal Impulse Response from IPB
IPD
IPN
IPB
IPM
0.5
-0.5
0
1
2
3
4
5
6
7
8
9
10
95 % Bootstrap CI, 100 runs




```
#1.4
fevd(varfit2,n.ahead = 20)

## $IPD
##              IPD         IPN         IPB        IPM
##  [1,] 1.0162791 0.000000000 0.000000000 0.00000000
##  [2,] 0.9824672 0.005279578 0.002094763 0.02642407
##  [3,] 0.9821815 0.005224575 0.002119878 0.02674088
##  [4,] 0.9821024 0.005246297 0.002295769 0.02662441
##  [5,] 0.9777281 0.005367907 0.003850941 0.02931819
##  [6,] 0.9720542 0.005335837 0.009677250 0.02918178
##  [7,] 0.9546833 0.005497666 0.023514152 0.03253440
##  [8,] 0.9536307 0.005517148 0.023527725 0.03355381
##  [9,] 0.9532684 0.005598469 0.023513264 0.03384959
## [10,] 0.9524733 0.005766881 0.023590530 0.03439857
## [11,] 0.9521017 0.005841247 0.023828008 0.03445752
## [12,] 0.9513875 0.005886896 0.024532248 0.03441912
## [13,] 0.9506049 0.005933864 0.025296225 0.03438856
## [14,] 0.9504630 0.005937495 0.025395872 0.03442671
## [15,] 0.9503586 0.005942617 0.025446425 0.03447517
## [16,] 0.9502519 0.005949491 0.025477382 0.03454385
## [17,] 0.9501491 0.005960875 0.025501861 0.03461060
## [18,] 0.9500726 0.005970263 0.025537864 0.03464160
```

```
## [19,] 0.9500432 0.005976508 0.025559279 0.03464327
## [20,] 0.9500347 0.005979302 0.025564934 0.03464336
##
## $IPN
##             IPD       IPN          IPB          IPM
##  [1,] 0.1192644 0.9028287 0.000000e+00 0.0000000000
##  [2,] 0.1230410 0.8989813 0.000000e+00 0.0000000000
##  [3,] 0.1228147 0.8986023 4.411844e-05 0.0005565253
##  [4,] 0.1233267 0.8980870 4.414190e-05 0.0005568213
##  [5,] 0.1229838 0.8984253 4.815606e-05 0.0005580230
##  [6,] 0.1236989 0.8945437 1.017092e-04 0.0036410180
##  [7,] 0.1333677 0.8847696 1.886358e-04 0.0036016444
##  [8,] 0.1333756 0.8846578 2.666894e-04 0.0036268538
##  [9,] 0.1334806 0.8845419 2.705794e-04 0.0036328385
## [10,] 0.1334950 0.8845124 2.740176e-04 0.0036443219
## [11,] 0.1335318 0.8843872 3.399585e-04 0.0036660101
## [12,] 0.1335199 0.8842869 4.480334e-04 0.0036691312
## [13,] 0.1334896 0.8840691 6.765002e-04 0.0036868716
## [14,] 0.1334886 0.8840617 6.827616e-04 0.0036889731
## [15,] 0.1334948 0.8840410 6.891013e-04 0.0036968741
## [16,] 0.1335076 0.8840086 7.018609e-04 0.0037035652
## [17,] 0.1335088 0.8839842 7.186509e-04 0.0037096221
## [18,] 0.1335096 0.8839629 7.346745e-04 0.0037140199
## [19,] 0.1335089 0.8839469 7.488110e-04 0.0037163215
## [20,] 0.1335108 0.8839409 7.514547e-04 0.0037178379
##
## $IPB
##             IPD        IPN       IPB        IPM
##  [1,] 0.5325818 0.01108068 0.4702910 0.00000000
##  [2,] 0.5293160 0.01090062 0.4453601 0.02845625
##  [3,] 0.5136922 0.01271750 0.4455026 0.04216594
##  [4,] 0.5002567 0.01375608 0.4508443 0.04924366
##  [5,] 0.4967367 0.01452222 0.4444657 0.05840314
##  [6,] 0.4910426 0.01615129 0.4396056 0.06735104
##  [7,] 0.4927291 0.01652144 0.4375295 0.06738367
##  [8,] 0.4948519 0.01677905 0.4354249 0.06712004
##  [9,] 0.4939130 0.01704939 0.4362062 0.06700722
## [10,] 0.4936640 0.01703781 0.4365126 0.06696083
## [11,] 0.4936918 0.01704095 0.4364431 0.06699960
## [12,] 0.4933146 0.01707793 0.4362472 0.06753642
## [13,] 0.4931959 0.01708218 0.4361324 0.06776605
## [14,] 0.4932051 0.01709971 0.4360667 0.06780539
## [15,] 0.4932007 0.01712719 0.4360418 0.06780754
## [16,] 0.4932093 0.01713123 0.4360306 0.06780621
## [17,] 0.4932157 0.01713148 0.4360178 0.06781230
## [18,] 0.4932175 0.01713146 0.4360146 0.06781371
## [19,] 0.4932175 0.01713145 0.4360139 0.06781442
## [20,] 0.4932158 0.01713140 0.4360125 0.06781757
##
## $IPM
```

```
##              IPD        IPN        IPB       IPM
##  [1,] 0.2517357 0.01127683 0.05524346 0.6968603
##  [2,] 0.2734694 0.01841036 0.05313009 0.6702015
##  [3,] 0.2766705 0.02394181 0.05405900 0.6605860
##  [4,] 0.2754787 0.03131310 0.05366630 0.6548557
##  [5,] 0.2801549 0.03110672 0.05338635 0.6506701
##  [6,] 0.2788247 0.03076806 0.05328942 0.6524390
##  [7,] 0.2803882 0.03069026 0.05332377 0.6509225
##  [8,] 0.2800315 0.03065320 0.05328888 0.6513507
##  [9,] 0.2800787 0.03070404 0.05325974 0.6512827
## [10,] 0.2801056 0.03070312 0.05338510 0.6511310
## [11,] 0.2799832 0.03070097 0.05367349 0.6509665
## [12,] 0.2798723 0.03068822 0.05402914 0.6507339
## [13,] 0.2797382 0.03068052 0.05420299 0.6507017
## [14,] 0.2796877 0.03067623 0.05422886 0.6507306
## [15,] 0.2796930 0.03068644 0.05423886 0.6507051
## [16,] 0.2797058 0.03069953 0.05425403 0.6506641
## [17,] 0.2797163 0.03070452 0.05427397 0.6506287
## [18,] 0.2797265 0.03070513 0.05430037 0.6505914
## [19,] 0.2797282 0.03070593 0.05432320 0.6505660
## [20,] 0.2797264 0.03070585 0.05433225 0.6505588
```

```
#1.5
detach("package:vars", unload = TRUE)
require(MTS)
```

```
## Loading required package: MTS
```

```
##
## Attaching package: 'MTS'
```

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

```
VARpred(m2,6,879)
```

```
## orig  879
## Forecasts at origin:  879
##           IPD       IPN     IPB     IPM
## [1,] -1.9893 -0.672682 -3.6293 -5.4249
## [2,] 10.9914  0.195877  0.1568 -2.9138
## [3,]  6.6928  0.429555 -1.4738 -1.3429
## [4,]  1.7880 -0.001829 -1.4463 -2.1403
## [5,]  2.0648 -0.661563 -2.9263 -1.0580
## [6,]  0.4245 -1.016649 -2.6670 -0.4851
## Standard Errors of predictions:
##       [,1]   [,2]  [,3]  [,4]
## [1,] 3.110 0.7750 1.603 1.485
## [2,] 3.185 0.7872 1.652 1.545
## [3,] 3.229 0.7880 1.681 1.557
```

IPM

1 step [-5.4249 - 1.96 (1.485)

2 step [-2.9138 - 1.96 (1.545)

3step [-1.3429 - 1.96 (1.557)

4step [-2.1403 - 1.96 (1.564)

5 step [-1.0580 - 1.96(1.569)

6 step [-0.4851 - 1.96(1.579)

```
## [4,] 3.246 0.7882 1.704 1.564
## [5,] 3.254 0.7896 1.717 1.569
## [6,] 3.264 0.7914 1.733 1.579
## Root mean square errors of predictions:
##       [,1]   [,2]  [,3]  [,4]
## [1,] 3.154 0.7859 1.626 1.506
## [2,] 3.435 0.8288 1.811 1.736
## [3,] 3.377 0.7906 1.779 1.601
## [4,] 3.306 0.7890 1.784 1.588
## [5,] 3.282 0.7946 1.759 1.588
## [6,] 3.298 0.7975 1.788 1.614
## Observations, predicted values,     errors, and MSE
##      time     obs    fcst     err     obs    fcst     err     obs    fcst
## case  880 18.0079 -1.9893 19.9972  1.7436 -0.6727  2.4163  9.8203 -3.6293
## case  881 31.6900 10.9914 20.6986  2.1309  0.1959  1.9350 11.7652  0.1568
## case  882 15.0623  6.6928  8.3695  1.0549  0.4296  0.6253  7.4632 -1.4738
## case  883 -0.9751  1.7880 -2.7630  1.2688 -0.0018  1.2706  2.5593 -1.4463
## case  884 -1.3476  2.0648 -3.4123 -1.5321 -0.6616 -0.8705 -0.1420 -2.9263
## case  885  0.8809  0.4245  0.4564  0.0960 -1.0166  1.1127  0.8422 -2.6670
##          err     obs    fcst    err
## case 13.4495 -1.0442 -5.4249 4.3807
## case 11.6084  5.1887 -2.9138 8.1025
## case  8.9369  3.6564 -1.3429 4.9993
## case  4.0056  0.6973 -2.1403 2.8376
## case  2.7842  0.0761 -1.0580 1.1341
## case  3.5092  1.1026 -0.4851 1.5877
```

Question 2

```
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"
```

```
rate=diff(log(as.numeric(`ETH-USD`[,6])))
ETHlogreturn=na.omit(rate)
#2.1
plot(`ETH-USD`,type='l')
```

Question 1

1.1 From the diagram, we can see that all of them may have relationship to each other because they all have high and low volatility in the same period.

1.2
$$\widehat{IPD}_t = \underset{(0.1122942)}{0.2985377} + \underset{(0.1370)}{0.226}\,IPN_{t-1} + \underset{(0.0701)}{0.417}\,IPM_{t-1} - \underset{(0.0337)}{0.213}\,\widehat{IPD}_{t-2} - \underset{(0.0379)}{0.131}\,IPD_{t-3} + \underset{(0.0804)}{0.138}\,IPM_{t-3} - \underset{(0.0746)}{0.1674}\,IPB_{t-4} + \underset{(0.0811)}{0.187}\,IPM_{t-4}$$
$$- \underset{(0.0658)}{0.228}\,IPB_{t-5} + \underset{(0.0456)}{0.1592}\,IPD_{t-6} - \underset{(0.0819)}{0.326}\,IPB_{t-6}$$

$$\widehat{IPN}_t = \underset{(0.02740226)}{0.157489} + \underset{(0.00836)}{0.0358}\,IPD_{t-1} - \underset{(0.0348)}{0.166}\,IPN_{t-1} + \underset{(0.0329)}{0.0616}\,IPN_{t-4} + \underset{(0.0169)}{0.0305}\,IPM_{t-5} + \underset{(0.00807)}{0.0255}\,IPD_{t-6}$$

$$\widehat{IPB}_t = \underset{(0.0568169)}{0.1542269} + \underset{(0.01982)}{0.0340}\,IPD_{t-1} + \underset{(0.0413)}{0.225}\,IPM_{t-1} - \underset{(0.0251)}{0.129}\,IPD_{t-2} + \underset{(0.0484)}{0.1374}\,IPB_{t-2} + \underset{(0.0431)}{0.105}\,IPM_{t-2} - \underset{(0.0245)}{0.113}\,IPD_{t-3} + \underset{(0.0478)}{0.149}\,IPB_{t-3}$$
$$+ \underset{(0.0439)}{0.134}\,IPM_{t-3} - \underset{(0.0389)}{0.0827}\,IPB_{t-4} + \underset{(0.0422)}{0.117}\,IPM_{t-4} - \underset{(0.0198)}{0.0407}\,IPD_{t-5} + \underset{(0.0417)}{0.1049}\,IPM_{t-5}$$

$$\widehat{IPM}_t = \underset{(0.05440154)}{0.1081254} + \underset{(0.01832)}{0.0395}\,IPD_{t-1} + \underset{(0.0687)}{0.145}\,IPN_{t-1} + \underset{(0.0381)}{0.2}\,IPM_{t-1} + \underset{(0.0698)}{0.141}\,IPN_{t-2} + \underset{(0.0360)}{0.0737}\,IPB_{t-2} - \underset{(0.039)}{0.1}\,IPM_{t-2} + \underset{(0.0654)}{0.185}\,IPN_{t-2}$$
$$+ \underset{(0.0163)}{0.0443}\,IPD_{t-4} + \underset{(0.0177)}{0.0524}\,IPD_{t-5} - \underset{(0.0379)}{0.1676}\,IPM_{t-5} + \underset{(0.016)}{0.0391}\,IPD_{t-6}$$

1.3 IPD = durable consumer good index IPB = business equivalent index

IPN = nondurable consumer goods index IPMat = material index

<u>shock from IPD</u>

- the shock of IPD have positive impact to durable consumer good index (IPD) for 1 period and have negative impact to itself in period 2 and 3 before die down in period 4.
- the shock of IPD also have positive impact to IPN, IPB and IPM with different magnitude and magnitude of effect to IPN is the lowest.

<u>shock from IPN</u>

- The shock from IPN have positive impact to IPD for 2 period before creating negative impact to IPD in period 6 and die down in period 7.
- The shock from IPN have positive impact to itself from initial until creating negative impact in period 1 and die down.
- The shock from IPN have slightly positive impact to IPB and IPM.

<u>Shock from IPB</u>

- The shock from IPB have negative impact to IPD from period 4 to 7.
- The shock from IPB doesn't have any impact to IPN
- The shock from IPB have positive Impact to itself and IPM from initial to period 4 and from initial to period 3, respectively.

<u>Shock from IPM</u>

- The shock from IPM have positive impact to IPD from period 1 to 5 and have negative impact to IPD until period 8.
- The shock from IPM does not have any impact to IPN.
- The shock from IPM have positive impact to IPB from period 1 to 6.
- The shock from IPM have positive impact to itself for first 2 period.

1.4 For IPD

Majority impact come from ε_{IPD} itself with more than 90% and around 0.5% come from ε_{IPN}, around 2% come from ε_{IPB} and around 2-3% come from ε_{IPM} that make us forecast error

For IPN

Majority impact come from ε_{IPN} itself around 88-90% and from ε_{IPD} around 12-13% and the rest come from ε_{IPB} and IPM that make us forecast error.

For IPB

Majority impact come from ε_{IPB} itself around 43-47% and from ε_{IPD} around 49-53% and the rest come from ε_{IPN} (1-2%) and ε_{IPM} (4-6%) that make us forecast error.

For IPM

Majority impact come from ε_{IPM} itself around 65-69% and from ε_{IPD} around 27-28%. The rest come from ε_{IPN} (1-3%) and ε_{IPB} (5%) that make us forecast error.

1.5 IPD

1 step	$[-1.9893 - 1.96(3.110),\ -1.9893 + 1.96(3.110)]$
2 step	$[10.9914 - 1.96(3.185),\ 10.9914 + 1.96(3.185)]$
3 step	$[6.6928 - 1.96(3.229),\ 6.6928 + 1.96(3.229)]$
4 step	$[1.7880 - 1.96(3.246),\ 1.7880 + 1.96(3.246)]$
5 step	$[2.0648 - 1.96(3.254),\ 2.0648 + 1.96(3.254)]$
6 step	$[0.4245 - 1.96(3.264),\ 0.4245 + 1.96(3.264)]$

IPN

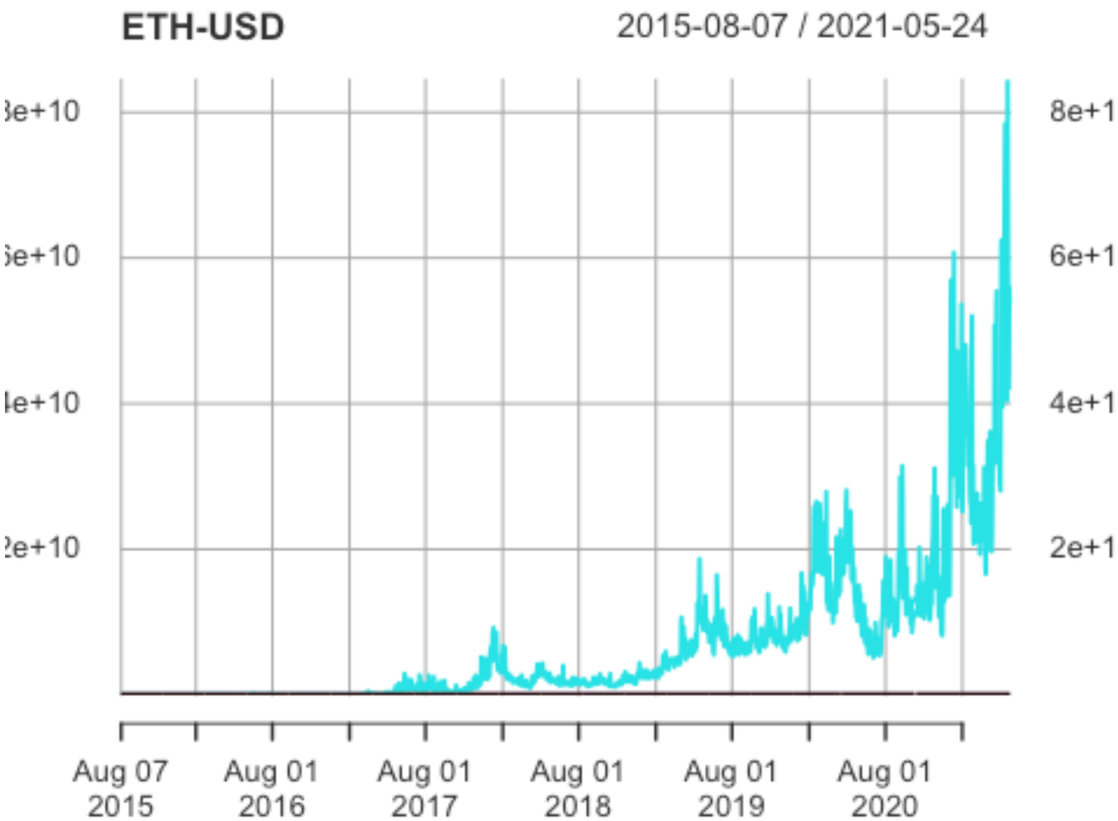
1 step	$[-0.672682 - 1.96(0.7750),\ -0.672682 + 1.96(0.7750)]$
2 step	$[0.195877 - 1.96(0.7872),\ 0.195877 + 1.96(0.7872)]$
3 step	$[0.429555 - 1.96(0.7880),\ 0.429555 + 1.96(0.7880)]$
4 step	$[-0.001829 - 1.96(0.7882),\ -0.001829 + 1.96(0.7882)]$
5 step	$[-0.661563 - 1.96(0.7896),\ -0.661563 + 1.96(0.7896)]$
6 step	$[-1.016649 - 1.96(0.7914),\ -1.016649 + 1.96(0.7914)]$

IPB

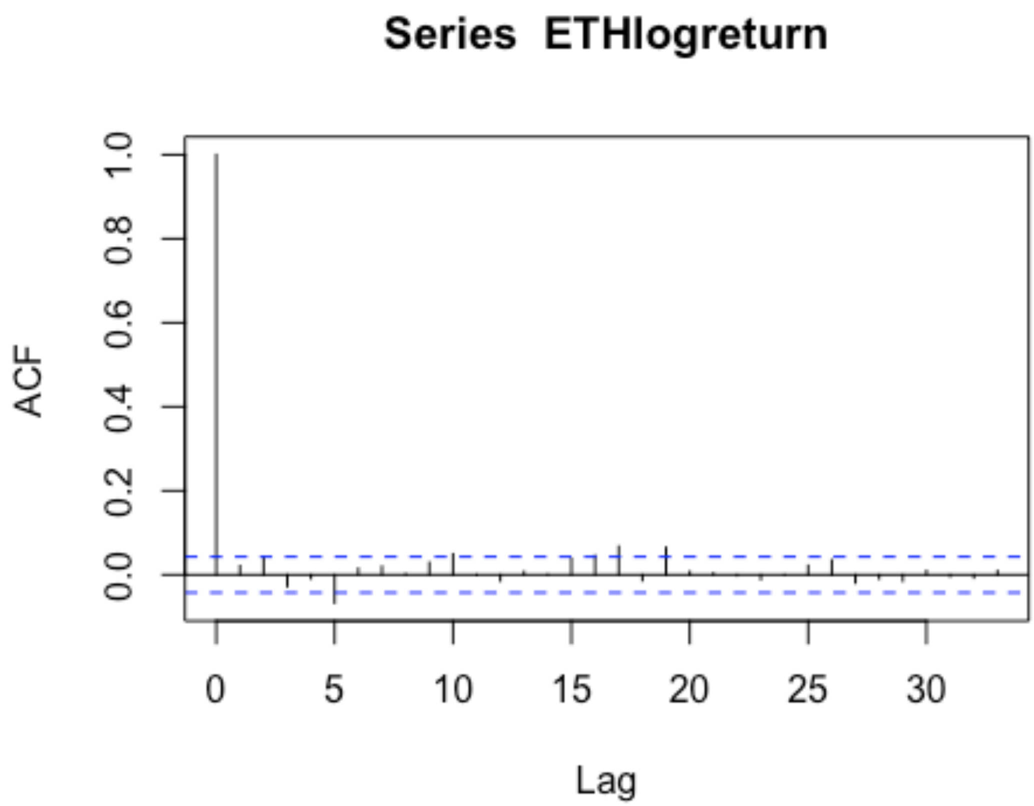
1 step	$[-3.6293 - 1.96(1.603),\ -3.6293 + 1.96(1.603)]$
2 step	$[0.1568 - 1.96(1.652),\ 0.1568 + 1.96(1.652)]$
3 step	$[-1.4738 - 1.96(1.681),\ -1.4738 + 1.96(1.681)]$
4 step	$[-1.4463 - 1.96(1.704),\ -1.4463 + 1.96(1.704)]$
5 step	$[-2.9263 - 1.96(1.717),\ -2.9263 + 1.96(1.717)]$
6 step	$[-2.6670 - 1.96(1.733),\ -2.6670 + 1.96(1.733)]$

IPM

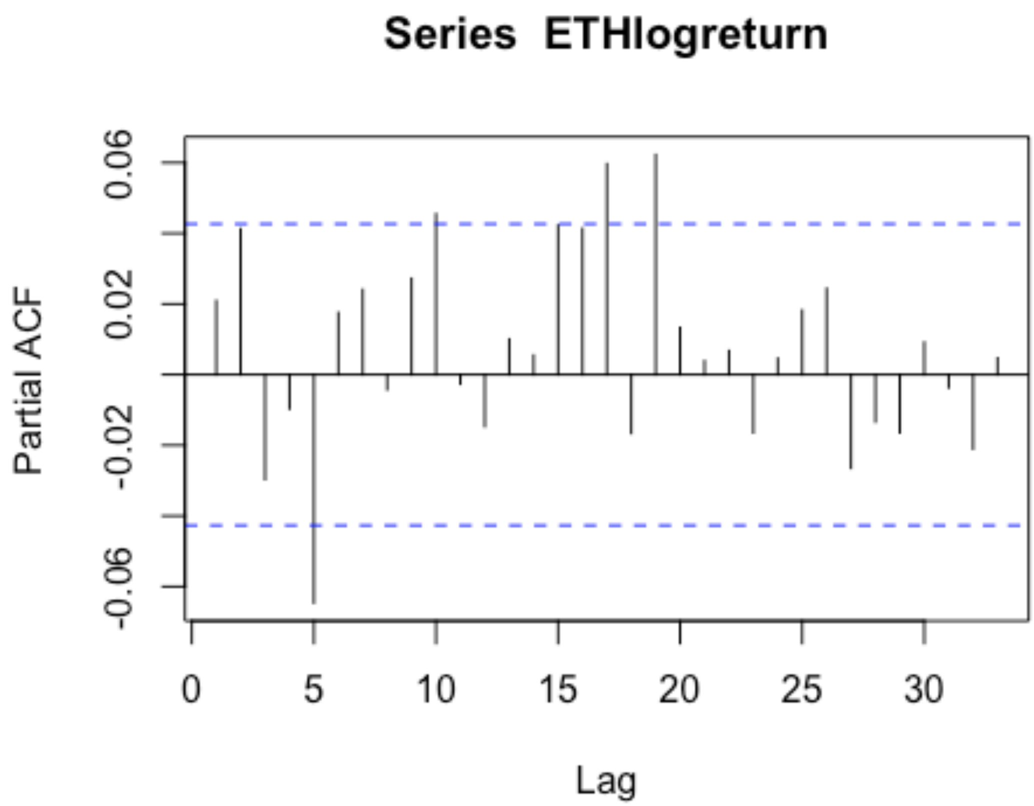
1 step	$[-5.4249 - 1.96(1.485),\ -5.4249 + 1.96(1.485)]$
2 step	$[-2.9138 - 1.96(1.545),\ -2.9138 + 1.96(1.545)]$
3 step	$[-1.3429 - 1.96(1.557),\ -1.3429 + 1.96(1.557)]$
4 step	$[-2.1403 - 1.96(1.564),\ -2.1403 + 1.96(1.564)]$
5 step	$[-1.0580 - 1.96(1.569),\ -1.0580 + 1.96(1.569)]$
6 step	$[-0.4851 - 1.96(1.579),\ -0.4851 + 1.96(1.579)]$



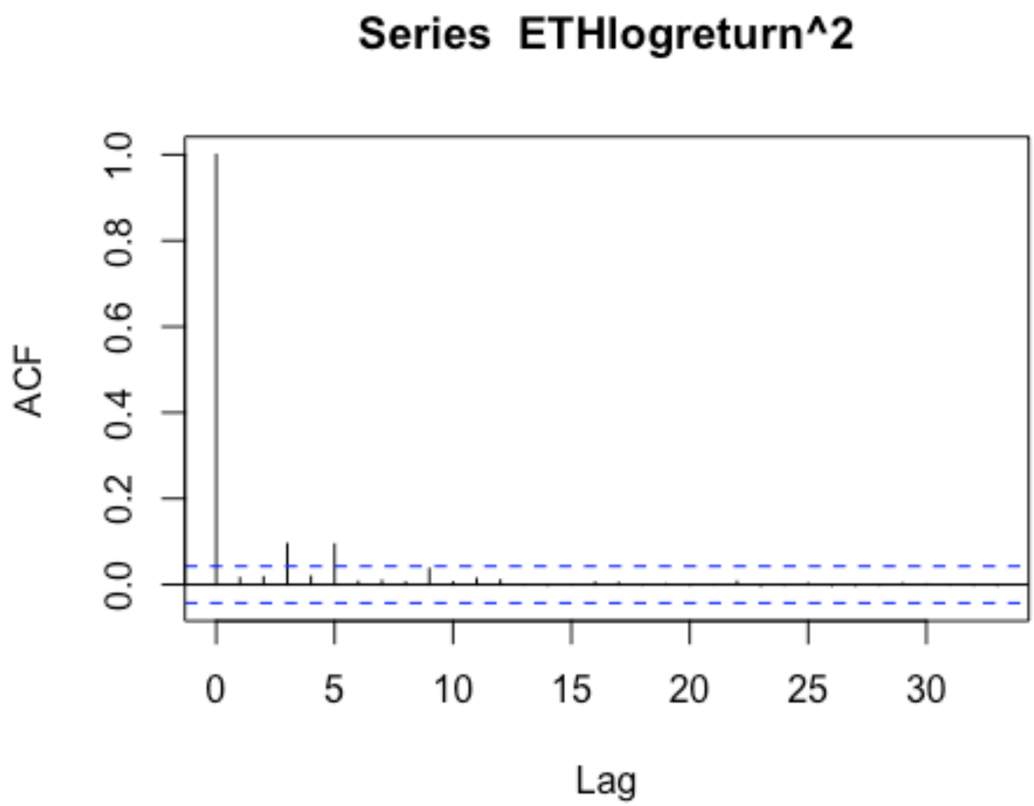

```
acf(ETHlogreturn)
```



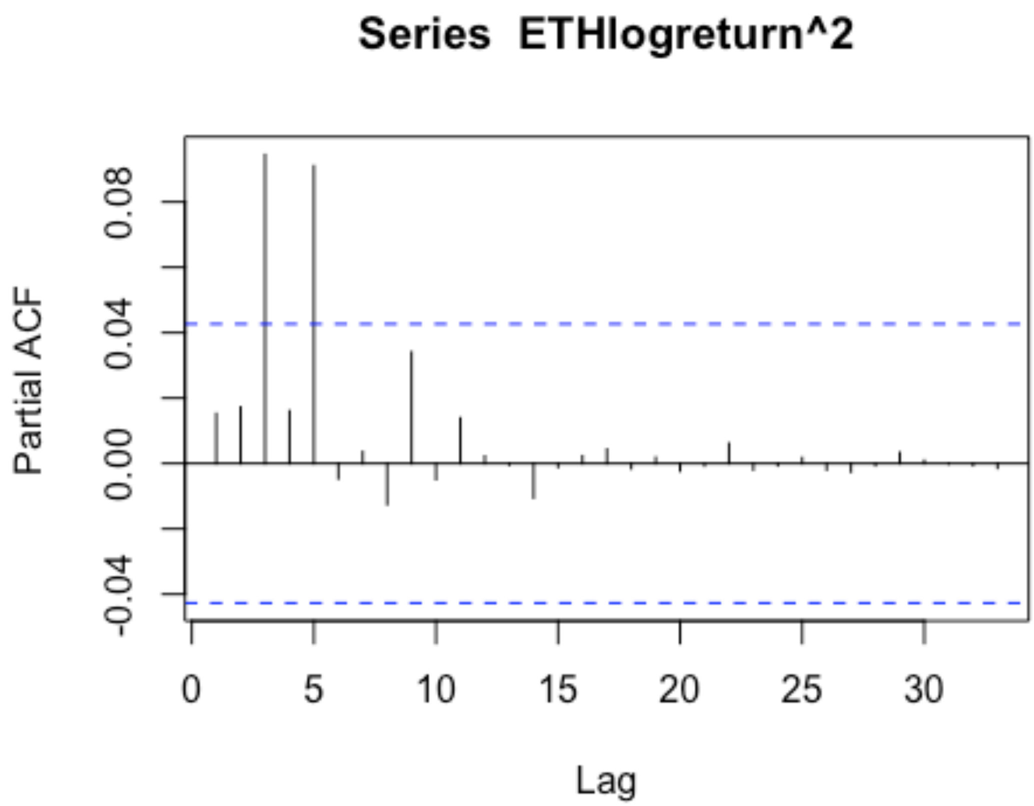

```
pacf(ETHlogreturn)
```




```
t.test(ETHlogreturn)

##
##  One Sample t-test
##
## data:  ETHlogreturn
## 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

acf(ETHlogreturn^2)
```




```
pacf(ETHlogreturn^2)
```




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

##
##  Box-Ljung test
##
## data:  ETHlogreturn^2
## X-squared = 42.837, df = 10, p-value = 5.319e-06

m1=garchFit(~garch(1,1),data=ETHlogreturn,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(m1)

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = ETHlogreturn, trace = F)
##
## Mean and Variance Equation:
```

```
##  data ~ garch(1, 1)
## <environment: 0x7fc979b6ecb0>
##  [data = ETHlogreturn]
##
## 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 16:37:06 2021 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  235016.8  0
##  Shapiro-Wilk Test  R    W      0.843771  0
##  Ljung-Box Test     R    Q(10)  18.23277  0.05116175
##  Ljung-Box Test     R    Q(15)  22.71407  0.09040232
##  Ljung-Box Test     R    Q(20)  37.32252  0.01070214
##  Ljung-Box Test     R^2  Q(10)  0.1873262 0.9999999
##  Ljung-Box Test     R^2  Q(15)  0.2821231 1
##  Ljung-Box Test     R^2  Q(20)  0.3662152 1
##  LM Arch Test       R    TR^2   8.72732   0.7260276
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -2.765593 -2.754873 -2.765600 -2.761667
```

```r
m2=garchFit(~garch(1,1),data=ETHlogreturn,cond.dist='std',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(m2)
```

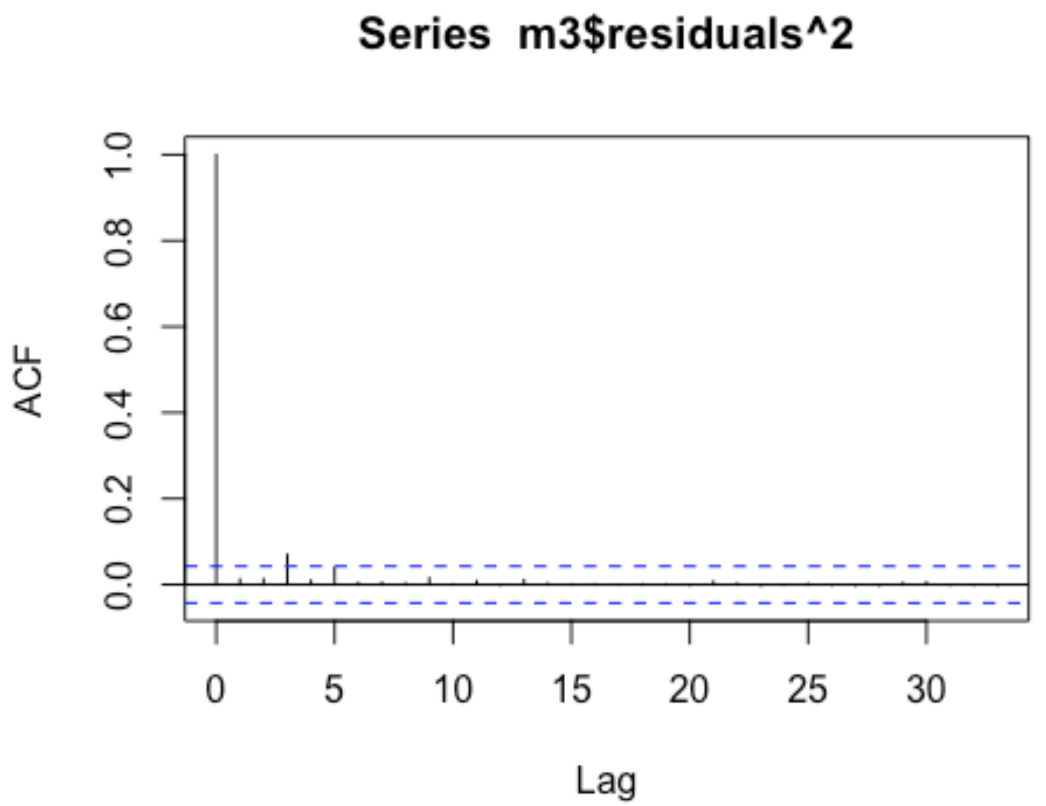
```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = ETHlogreturn, cond.dist = "std",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x7fc979394a28>
##  [data = ETHlogreturn]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         mu       omega      alpha1       beta1       shape
## 0.00091922  0.00027738  0.28888895  0.74732147  2.93594522
##
## Std. Errors:
##  based on Hessian
##
## 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 16:37:07 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
```

```
m3=auto.arima(ETHlogreturn)
summary(m3)
```

```
## Series: ETHlogreturn
## 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      MA
SE
## Training set -1.441276e-05 0.06774656 0.04212343 78.67127 204.4328 0.69458
22
##                       ACF1
## Training set -0.003782915
```

```
acf(m3$residuals^2)
```




```
Box.test(m3$residuals^2,lag=10,type = 'Ljung')

##
##  Box-Ljung test
##
## data:  m3$residuals^2
## X-squared = 15.141, df = 10, p-value = 0.127

m4=garchFit(~garch(1,2),data=ETHlogreturn,cond.dist='std',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 = ~garch(1, 2), data = ETHlogreturn, cond.dist = "std",
##     trace = F)
##
```

```
## Mean and Variance Equation:
##  data ~ garch(1, 2)
## <environment: 0x7fc979970c00>
##  [data = ETHlogreturn]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         mu       omega      alpha1       beta1       beta2       shape
## 0.00089697  0.00026597  0.30577762  0.67601667  0.06819523  2.91776683
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     8.970e-04   8.146e-04    1.101 0.270874
## omega  2.660e-04   7.951e-05    3.345 0.000823 ***
## alpha1 3.058e-01   7.412e-02    4.125  3.7e-05 ***
## beta1  6.760e-01   2.046e-01    3.304 0.000953 ***
## beta2  6.820e-02   1.692e-01    0.403 0.686871
## shape  2.918e+00   2.200e-01   13.263  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3289.918    normalized:  1.559203
##
## Description:
##  Wed Jun  9 16:37:09 2021 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  315496.4  0
##  Shapiro-Wilk Test  R    W      0.8280633 0
##  Ljung-Box Test     R    Q(10)  15.07738  0.1292662
##  Ljung-Box Test     R    Q(15)  19.14171  0.2073839
##  Ljung-Box Test     R    Q(20)  33.11988  0.03273253
##  Ljung-Box Test     R^2  Q(10)  20.51501  0.02474113
##  Ljung-Box Test     R^2  Q(15)  20.61074  0.1497339
##  Ljung-Box Test     R^2  Q(20)  20.68499  0.4158778
##  LM Arch Test       R    TR^2   10.2514   0.5939156
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.112719 -3.096640 -3.112735 -3.106831

m5=garchFit(~garch(5,0),data=ETHlogreturn,cond.dist='std',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(m5)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(5, 0), data = ETHlogreturn, cond.dist = "std",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(5, 0)
## <environment: 0x7fc977ac4a08>
##  [data = ETHlogreturn]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##         mu       omega      alpha1      alpha2      alpha3      alpha4
## 0.00068415  0.00151061  0.33633976  0.24404973  0.20796997  0.18127655
##     alpha5       shape
## 0.19545918  2.83419557
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     0.0006842   0.0008230    0.831 0.405829
## omega  0.0015106   0.0002658    5.684 1.31e-08 ***
## alpha1 0.3363398   0.0877557    3.833 0.000127 ***
## alpha2 0.2440497   0.0696305    3.505 0.000457 ***
## alpha3 0.2079700   0.0688542    3.020 0.002524 **
## alpha4 0.1812766   0.0607164    2.986 0.002830 **
## alpha5 0.1954592   0.0630001    3.103 0.001919 **
## shape  2.8341956   0.2144659   13.215  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3280.91    normalized:  1.554933
##
## Description:
##  Wed Jun  9 16:37:09 2021 by user:
##
```

```
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  147170.5  0
##  Shapiro-Wilk Test  R    W      0.8553714 0
##  Ljung-Box Test     R    Q(10)  15.93493  0.1015104
##  Ljung-Box Test     R    Q(15)  19.4331   0.1947742
##  Ljung-Box Test     R    Q(20)  34.17784  0.02494653
##  Ljung-Box Test     R^2  Q(10)  13.34737  0.2048933
##  Ljung-Box Test     R^2  Q(15)  13.4776   0.5654594
##  Ljung-Box Test     R^2  Q(20)  13.60017  0.8501761
##  LM Arch Test       R    TR^2   12.48429  0.4076102
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.102284 -3.080845 -3.102313 -3.094433
```

```
m6=garchFit(~garch(3,0),data=ETHlogreturn,cond.dist='std',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(m6)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(3, 0), data = ETHlogreturn, cond.dist = "std",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(3, 0)
## <environment: 0x7fc97900e6d8>
##  [data = ETHlogreturn]
##
## Conditional Distribution:
##  std
##
## Coefficient(s):
##        mu      omega     alpha1     alpha2     alpha3      shape
## 0.0010220  0.0022809  0.4012062  0.3017895  0.2775118  2.7320298
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     0.0010220   0.0008399    1.217 0.223713
```

```
## omega  0.0022809   0.0003987    5.722 1.06e-08 ***
## alpha1 0.4012062   0.1065742    3.765 0.000167 ***
## alpha2 0.3017895   0.0840833    3.589 0.000332 ***
## alpha3 0.2775118   0.0846087    3.280 0.001038 **
## shape  2.7320298   0.2015884   13.553  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3264.491    normalized:  1.547152
##
## Description:
##  Wed Jun  9 16:37:10 2021 by user:
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  215047.7  0
##  Shapiro-Wilk Test  R    W      0.8281301 0
##  Ljung-Box Test     R    Q(10)  11.60007  0.312713
##  Ljung-Box Test     R    Q(15)  15.70265  0.4020844
##  Ljung-Box Test     R    Q(20)  32.21015  0.0411099
##  Ljung-Box Test     R^2  Q(10)  2.426389  0.9919055
##  Ljung-Box Test     R^2  Q(15)  2.519227  0.9998664
##  Ljung-Box Test     R^2  Q(20)  2.886809  0.9999971
##  LM Arch Test       R    TR^2   10.49032  0.5730271
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.088617 -3.072538 -3.088633 -3.082729
```

```
predict(m6,10)
```

```
##    meanForecast meanError standardDeviation
## 1   0.001021973 0.1602262         0.1602262
## 2   0.001021973 0.1727443         0.1727443
## 3   0.001021973 0.1897456         0.1897456
## 4   0.001021973 0.1812615         0.1812615
## 5   0.001021973 0.1860361         0.1860361
## 6   0.001021973 0.1899298         0.1899298
## 7   0.001021973 0.1905687         0.1905687
## 8   0.001021973 0.1932418         0.1932418
## 9   0.001021973 0.1955342         0.1955342
## 10  0.001021973 0.1974039         0.1974039
```

Question 3

```
##
## Attaching package: 'MTS'
```

Question 2

To identify the optimal model for Ethereum. Firstly, we pull the data from Yahoo Finance. Then, we turn the series into log return and construct ACF. The reason that we need to change price to return is that it is not weakly stationary.

From ACF figure, we can see that there is no pattern in figure which can implied that AR, MA, ARMA model may not suitable for mean equation. The mean equation of this series may be $r_t = \mu + a_t$

Then, we check that μ is equal to zero or not. After, use t-test and obtain p-value equal to 0.03274 which less than $\alpha = 0.05$. So, we can reject $H_0: \mu = 0$ at 0.05 level of significant. That mean, mean of log return of ETH is not equal to zero at 95% CI.
⑤ most of the standardized residual test show that the model is adequate.

Next, we check ARCH effect in the series by using Ljung-Box test

```
##
##  Box-Ljung test
##
## data:  ETHlogreturn^2
## X-squared = 42.837, df = 10, p-value = 5.319e-06
```

$H_0: \rho_1 = \rho_2 = \dots = \rho_m$ or There is no ARCH effect

$H_1: \exists \rho_m \neq 0$; $m = 1, 2, 3, \dots, 10$ or There is ARCH effect

p-value = 5.319e-06

Since p-value is less than $\alpha = 0.05$. Then, we reject H_0 at 0.05 level of significant.

That is, there is ARCH effect in the residual term in the series.

After that, we construct PACF to find optimal lag

We try to construct 6 model and we choose model 6 that may be the most suitable model in this case.

Because ① the sum of parameter of variance eq. < 1 ② it change distribution to student-t ③ all of parameter is significant in variance eq.

④ BIC, AIC criteria have value very close to the model that get the highest value

The fitted model : mean eq. : $\hat{r}_t = \underset{(0.0008399)}{0.0010220} + a_t$

variance eq. : $\hat{\sigma}_t^2 = \underset{(0.0003987)}{0.0022809} + \underset{(0.1065742)}{0.4012062}\, a_{t-1}^2 + \underset{(0.0840833)}{0.3017895}\, a_{t-2}^2 + \underset{(0.0846087)}{0.2775118}\, a_{t-3}^2$

with $a_t \sim t_{2.7320298}$

Finally, we check the adequacy of the model from $\tilde{a}_t$ (standardized Residual term)

Checking distribution of model by Jarque-Bera test and get p-value < 0.05 which mean that the distribution is not normal which suitable to the model that we use student-t distribution.

```
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  215047.7  0
##  Shapiro-Wilk Test  R    W      0.8281301 0
##  Ljung-Box Test     R    Q(10)  11.60007  0.312713
##  Ljung-Box Test     R    Q(15)  15.70265  0.4020844
##  Ljung-Box Test     R    Q(20)  32.21015  0.0411099
##  Ljung-Box Test     R^2  Q(10)  2.426389  0.9919055
##  Ljung-Box Test     R^2  Q(15)  2.519227  0.9998664
##  Ljung-Box Test     R^2  Q(20)  2.886809  0.9999971
##  LM Arch Test       R    TR^2   10.49032  0.5730271
##
```

test linear dependency in $\tilde{a}_t$ by Ljung-Box test.
From the result, we get p-value > 0.05 for m = 10, 15.
Then, there is no serial correlation in $\tilde{a}_t$ or mean equation is adequate with 95% CI.

test non-linear dependent in $\tilde{a}_t$
According to Ljung-Box test of $\tilde{a}_t$ (m = 10, 15, 20), we get p-value > 0.05. Then, there is no serial correlation among $\tilde{a}_t^2$ and variance equation is adequate with 95% CI

test ARCH effect by LM ARCH test. Then, we get p-value > 0.05. Therefore there is no ARCH effect left in this model.

In conclusion, we can use this model to forecast the log return and its volatility

mean ± 1.96 (SD)

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

```
library(fUnitRoots)
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"
```

```
dim(CLVMNACSCAB1GQUK)
```

```
## [1] 183   1
```

```
tail(CLVMNACSCAB1GQUK)
```

```
##            CLVMNACSCAB1GQUK
## 2019-04-01         470665.5
## 2019-07-01         473006.0
## 2019-10-01         473087.6
## 2020-01-01         458804.7
## 2020-04-01         372766.5
## 2020-07-01         432422.1
```

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

```
## [1] "NAEXKP01CAQ189S"
```

```
dim(NAEXKP01CAQ189S)
```

```
## [1] 241   1
```

```
tail(NAEXKP01CAQ189S)
```

```
##            NAEXKP01CAQ189S
## 2019-10-01    528528250000
## 2020-01-01    518324000000
## 2020-04-01    459052500000
## 2020-07-01    499863000000
## 2020-10-01    511481250000
## 2021-01-01    519664950000
```

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

```
## [1] "GDPC1"
```

```
dim(GDPC1)

## [1] 297   1

tail(GDPC1)

##                 GDPC1
## 2019-10-01 19253.96
## 2020-01-01 19010.85
## 2020-04-01 17302.51
## 2020-07-01 18596.52
## 2020-10-01 18794.43
## 2021-01-01 19088.06

x=cbind(as.numeric(CLVMNACSCAB1GQUK[21:146,]),as.numeric(NAEXKP01CAQ189S[77:2
02,]),as.numeric(GDPC1[133:258,]))
zt=log(x)
#ccm(zt)
colnames(zt)<-c("UKrealgdp","Canadarealgdp","USArealgdp")
#3.1
VARorder(zt)

## selected order: aic =  2
## selected order: bic =  2
## selected order: hq =  2
## Summary table:
##          p      AIC      BIC       HQ      M(p) p-value
##  [1,]  0 -19.1870 -19.1870 -19.1870    0.0000  0.0000
##  [2,]  1 -30.8581 -30.6555 -30.7758 1281.8143  0.0000
##  [3,]  2 -31.7835 -31.3783 -31.6189  112.7073  0.0000
##  [4,]  3 -31.7746 -31.1669 -31.5277   13.7316  0.1322
##  [5,]  4 -31.7090 -30.8986 -31.3798    7.6844  0.5662
##  [6,]  5 -31.6656 -30.6526 -31.2540    9.5949  0.3843
##  [7,]  6 -31.5926 -30.3771 -31.0988    6.5361  0.6853
##  [8,]  7 -31.5344 -30.1163 -30.9583    7.6607  0.5687
##  [9,]  8 -31.4368 -29.8160 -30.7783    3.9549  0.9144
## [10,]  9 -31.4173 -29.5940 -30.6765   10.4266  0.3171
## [11,] 10 -31.4933 -29.4674 -30.6702   17.8361  0.0371
## [12,] 11 -31.4227 -29.1942 -30.5174    5.6773  0.7717
## [13,] 12 -31.4948 -29.0637 -30.5071   16.2242  0.0623
## [14,] 13 -31.4542 -28.8205 -30.3842    7.4155  0.5939

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

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

##
## VAR Estimation Results:
## =========================
## Endogenous variables: UKrealgdp, Canadarealgdp, USArealgdp
## Deterministic variables: const
## Sample size: 124
## Log Likelihood: 1414.364
## Roots of the characteristic polynomial:
## 0.9856 0.9563 0.8217 0.5202 0.5202 0.1523
## Call:
## VAR(y = zt, p = 2)
##
##
## Estimation results for equation UKrealgdp:
## ==========================================
## UKrealgdp = UKrealgdp.l1 + Canadarealgdp.l1 + USArealgdp.l1 + UKrealgdp.l2
+ Canadarealgdp.l2 + USArealgdp.l2 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## UKrealgdp.l1      1.40140    0.08166  17.161  < 2e-16 ***
## Canadarealgdp.l1  0.03900    0.08309   0.469    0.640
## USArealgdp.l1     0.11577    0.09334   1.240    0.217
## UKrealgdp.l2     -0.49628    0.08134  -6.101  1.4e-08 ***
## Canadarealgdp.l2 -0.03246    0.08465  -0.383    0.702
## USArealgdp.l2    -0.04250    0.09416  -0.451    0.653
## const             0.34393    0.44609   0.771    0.442
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005656 on 117 degrees of freedom
## Multiple R-Squared: 0.9995,  Adjusted R-squared: 0.9994
## F-statistic: 3.565e+04 on 6 and 117 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation Canadarealgdp:
## ==============================================
```

```
## Canadarealgdp = UKrealgdp.l1 + Canadarealgdp.l1 + USArealgdp.l1 + UKrealgd
p.l2 + Canadarealgdp.l2 + USArealgdp.l2 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## UKrealgdp.l1      0.28589    0.08432   3.391 0.000952 ***
## Canadarealgdp.l1  1.20185    0.08579  14.009  < 2e-16 ***
## USArealgdp.l1     0.35327    0.09638   3.665 0.000373 ***
## UKrealgdp.l2     -0.26496    0.08399  -3.155 0.002042 **
## Canadarealgdp.l2 -0.24763    0.08741  -2.833 0.005432 **
## USArealgdp.l2    -0.33122    0.09723  -3.407 0.000902 ***
## const             0.74342    0.46060   1.614 0.109218
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00584 on 117 degrees of freedom
## Multiple R-Squared: 0.9994,  Adjusted R-squared: 0.9994
## F-statistic: 3.394e+04 on 6 and 117 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation USArealgdp:
## ==========================================
## USArealgdp = UKrealgdp.l1 + Canadarealgdp.l1 + USArealgdp.l1 + UKrealgdp.l
2 + Canadarealgdp.l2 + USArealgdp.l2 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## UKrealgdp.l1      0.36850    0.08432   4.370  2.7e-05 ***
## Canadarealgdp.l1  0.17407    0.08580   2.029  0.04474 *
## USArealgdp.l1     1.04262    0.09639  10.817  < 2e-16 ***
## UKrealgdp.l2     -0.26915    0.08399  -3.204  0.00174 **
## Canadarealgdp.l2 -0.25543    0.08741  -2.922  0.00417 **
## USArealgdp.l2    -0.06237    0.09723  -0.641  0.52250
## const             1.08791    0.46063   2.362  0.01984 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00584 on 117 degrees of freedom
## Multiple R-Squared: 0.9996,  Adjusted R-squared: 0.9995
## F-statistic: 4.507e+04 on 6 and 117 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##               UKrealgdp Canadarealgdp USArealgdp
## UKrealgdp     3.199e-05     1.229e-06  5.799e-06
## Canadarealgdp 1.229e-06     3.411e-05  1.441e-05
## USArealgdp    5.799e-06     1.441e-05  3.411e-05
##
## Correlation matrix of residuals:
```

```
##               UKrealgdp Canadarealgdp USArealgdp
## UKrealgdp        1.0000        0.0372     0.1756
## Canadarealgdp    0.0372        1.0000     0.4225
## USArealgdp       0.1756        0.4225     1.0000
```

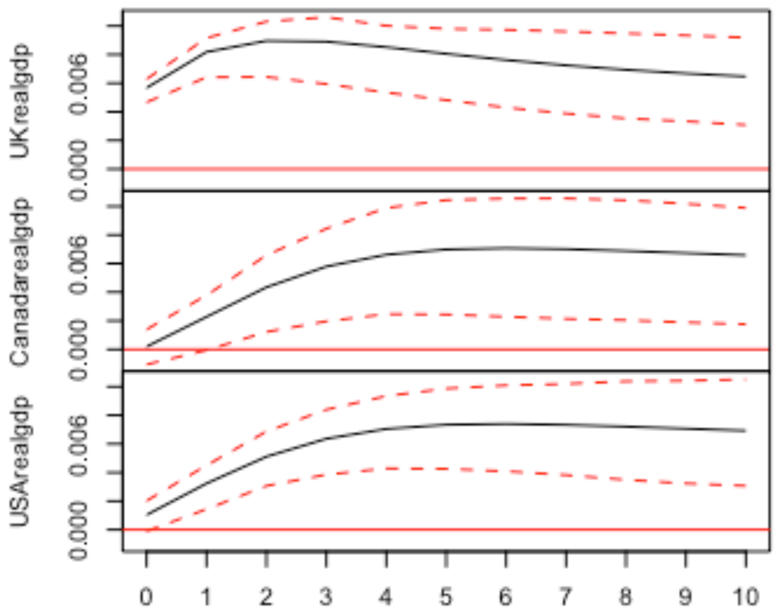
```
varfit1 <- restrict(varfit, thresh=1.645)
summary(varfit1)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: UKrealgdp, Canadarealgdp, USArealgdp
## Deterministic variables: const
## Sample size: 124
## Log Likelihood: 1411.917
## Roots of the characteristic polynomial:
##     1     1 0.816 0.5505 0.5505 0.2299
## Call:
## VAR(y = zt, p = 2)
##
##
## Estimation results for equation UKrealgdp:
## ==========================================
## UKrealgdp = UKrealgdp.l1 + USArealgdp.l1 + UKrealgdp.l2 + const
##
##               Estimate Std. Error t value Pr(>|t|)
## UKrealgdp.l1   1.42694    0.07494  19.040  < 2e-16 ***
## USArealgdp.l1  0.08187    0.02684   3.050  0.00282 **
## UKrealgdp.l2  -0.52537    0.06902  -7.611 6.82e-12 ***
## const          0.48200    0.14500   3.324  0.00118 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005604 on 120 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1
## F-statistic: 1.568e+08 on 4 and 120 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation Canadarealgdp:
## ==============================================
## Canadarealgdp = UKrealgdp.l1 + Canadarealgdp.l1 + USArealgdp.l1 + UKrealgd
p.l2 + Canadarealgdp.l2 + USArealgdp.l2
##
##                   Estimate Std. Error t value Pr(>|t|)
## UKrealgdp.l1       0.29148    0.08482   3.437 0.000814 ***
## Canadarealgdp.l1  1.22221    0.08543  14.306  < 2e-16 ***
## USArealgdp.l1      0.37334    0.09622   3.880 0.000173 ***
## UKrealgdp.l2      -0.28088    0.08397  -3.345 0.001104 **
## Canadarealgdp.l2 -0.22529    0.08689  -2.593 0.010722 *
```

```
## USArealgdp.l2    -0.37887    0.09326  -4.062 8.78e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00588 on 118 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1
## F-statistic: 4.186e+08 on 6 and 118 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation USArealgdp:
## ===========================================
## USArealgdp = UKrealgdp.l1 + Canadarealgdp.l1 + USArealgdp.l1 + UKrealgdp.l
2 + Canadarealgdp.l2 + const
##
##                   Estimate Std. Error t value Pr(>|t|)
## UKrealgdp.l1      0.38363    0.08076   4.750 5.77e-06 ***
## Canadarealgdp.l1  0.19941    0.07597   2.625 0.009817 **
## USArealgdp.l1     0.98379    0.02958  33.261  < 2e-16 ***
## UKrealgdp.l2     -0.28365    0.08069  -3.515 0.000624 ***
## Canadarealgdp.l2 -0.28569    0.07340  -3.892 0.000165 ***
## const             1.17762    0.43779   2.690 0.008183 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005826 on 118 degrees of freedom
## Multiple R-Squared:     1,   Adjusted R-squared:     1
## F-statistic: 5.264e+07 on 6 and 118 DF,  p-value: < 2.2e-16
##
##
##
## Covariance matrix of residuals:
##                UKrealgdp Canadarealgdp USArealgdp
## UKrealgdp     3.221e-05     1.088e-06  5.881e-06
## Canadarealgdp 1.088e-06     3.487e-05  1.432e-05
## USArealgdp    5.881e-06     1.432e-05  3.423e-05
##
## Correlation matrix of residuals:
##                UKrealgdp Canadarealgdp USArealgdp
## UKrealgdp       1.00000       0.03246     0.1771
## Canadarealgdp   0.03246       1.00000     0.4145
## USArealgdp      0.17712       0.41450     1.0000
```

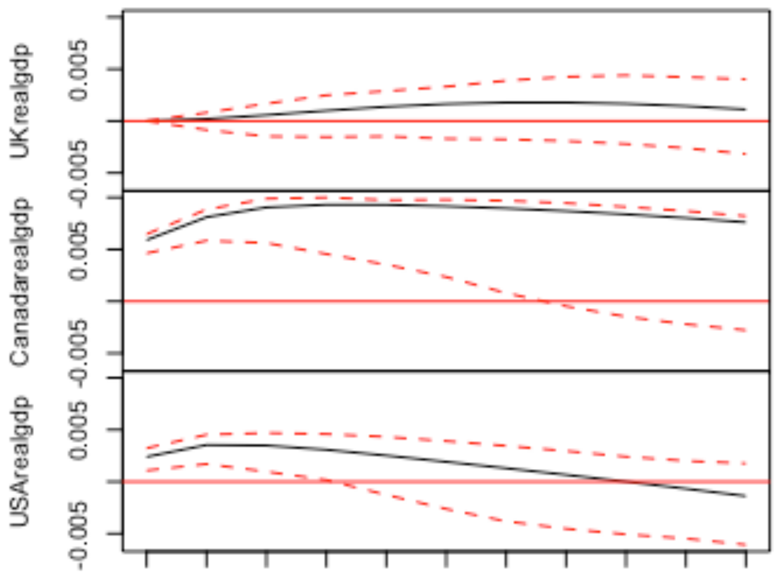
```
#3.2
impresp=irf(varfit1)
plot(impresp)
```

Orthogonal Impulse Response from UKrealgdp

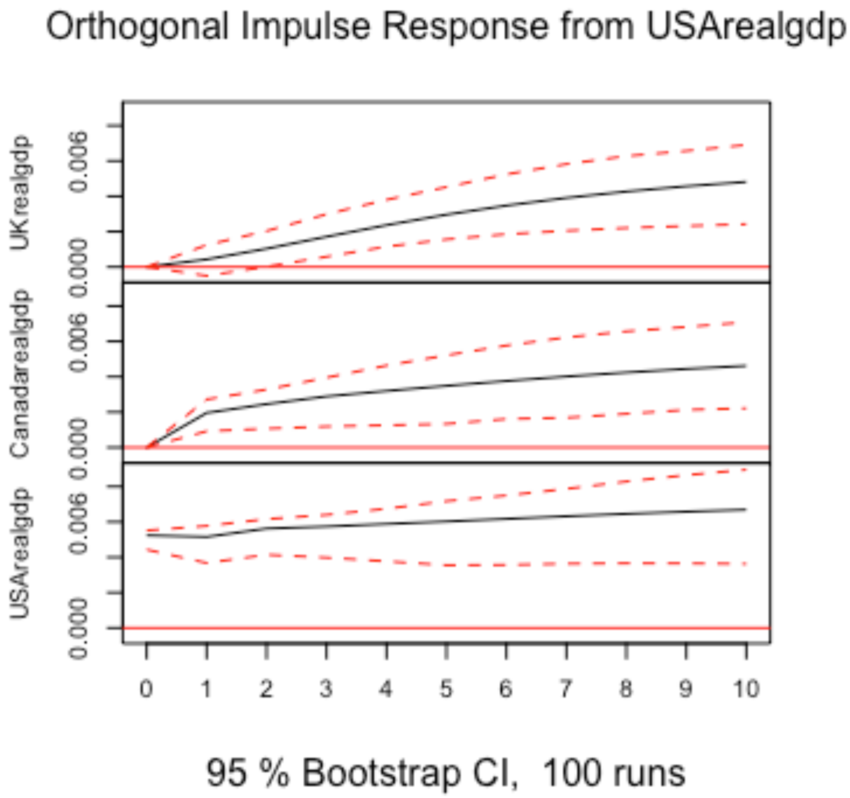


95 % Bootstrap CI, 100 runs

Orthogonal Impulse Response from Canadarealgdp



95 % Bootstrap CI, 100 runs




```
#3.3
fevd(varfit1, n.ahead=6)
```

```
## $UKrealgdp
##       UKrealgdp Canadarealgdp  USArealgdp
## [1,] 1.0256410   0.000000000 0.000000000
## [2,] 1.0231875   0.000395965 0.001897277
## [3,] 1.0161004   0.002048088 0.007092450
## [4,] 1.0034766   0.005229373 0.016201814
## [5,] 0.9857968   0.009598402 0.029100759
## [6,] 0.9644918   0.014393977 0.045148122
##
## $Canadarealgdp
##         UKrealgdp Canadarealgdp USArealgdp
## [1,] 0.001062768     1.0074842 0.00000000
## [2,] 0.048016806     0.9258963 0.03521163
## [3,] 0.112459302     0.8516191 0.04597886
## [4,] 0.169008504     0.7885718 0.05331976
## [5,] 0.211524052     0.7407981 0.05921652
## [6,] 0.241406398     0.7054123 0.06516548
##
## $USArealgdp
##        UKrealgdp Canadarealgdp USArealgdp
```

```
## [1,] 0.03163904     0.1686788  0.8082292
## [2,] 0.13920528     0.2194761  0.6512818
## [3,] 0.24806953     0.1995651  0.5639124
## [4,] 0.33469862     0.1702819  0.5078846
## [5,] 0.39615015     0.1428221  0.4748435
## [6,] 0.43791211     0.1198690  0.4566839
```

```
#3.4
logUKrealgdp=log(CLVMNACSCAB1GQUK[21:146,])
m1 <- adfTest(logUKrealgdp, lags=3, type=c("ct"), title=NULL, description = N
ULL)
```

```
## 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
```

```
##
## 0.7535667
```

```
logCanadarealgdp=log(NAEXKP01CAQ189S[77:202,])
m2 <- adfTest(logCanadarealgdp, 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
```

```
##
## 0.3721507
```

```
logUSArealgdp=log(GDPC1[133:258,])
m3 <- adfTest(logUSArealgdp, 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
```

```
m3@test$p.value
```

```
##
## 0.804053
```

```
m4 <- adfTest(diff(logUKrealgdp), lags=3, type=c("ct"), title=NULL, descripti
on = 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(logUKrealgdp), lags = 3, type = c("ct"), title = N
ULL, :
## p-value smaller than printed p-value

m4@test$p.value

##
## 0.01

m5 <- adfTest(diff(logCanadarealgdp), lags=3, type=c("ct"), title=NULL, descr
iption = 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(logCanadarealgdp), lags = 3, type = c("ct"), title
=
## NULL, : p-value smaller than printed p-value

m5@test$p.value

##
## 0.01

m6 <- adfTest(diff(logUSArealgdp), lags=3, type=c("ct"), title=NULL, descript
ion = 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(logUSArealgdp), lags = 3, type = c("ct"), title =
## NULL, : p-value smaller than printed p-value

m6@test$p.value

##
## 0.01

fit1 <- lm(logCanadarealgdp~logUKrealgdp+logUSArealgdp)
summary(fit1)

##
## Call:
## lm(formula = logCanadarealgdp ~ logUKrealgdp + logUSArealgdp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.031869 -0.016180 -0.000016  0.010834  0.060533
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)    15.3400     0.5305  28.919  < 2e-16 ***
## logUKrealgdp    0.6720     0.1138   5.906 3.19e-08 ***
## logUSArealgdp   0.2855     0.0978   2.919  0.00417 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.02058 on 123 degrees of freedom
## Multiple R-squared:  0.9928, Adjusted R-squared:  0.9927
## F-statistic:  8493 on 2 and 123 DF,  p-value: < 2.2e-16
```

```
error1=residuals(fit1)
m7 <- adfTest(error1, 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
```

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

```
##
## 0.02300044
```

```
fit2 <- lm(logUKrealgdp~logUSArealgdp+logCanadarealgdp)
summary(fit2)
```

```
##
## Call:
## lm(formula = logUKrealgdp ~ logUSArealgdp + logCanadarealgdp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.038761 -0.008042 -0.000668  0.010062  0.029801
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.43595    1.02802  -1.397    0.165
## logUSArealgdp     0.57420    0.04819  11.914  < 2e-16 ***
## logCanadarealgdp  0.32877    0.05566   5.906 3.19e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0144 on 123 degrees of freedom
## Multiple R-squared:  0.9964, Adjusted R-squared:  0.9964
## F-statistic: 1.717e+04 on 2 and 123 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
```

```
## Warning in adfTest(error2, lags = 3, type = c("nc"), title = NULL, descrip
tion =
## NULL): p-value smaller than printed p-value

m8@test$p.value

##
## 0.01

fit3 <- lm(logUSArealgdp~logCanadarealgdp+logUKrealgdp)
summary(fit3)

##
## Call:
## lm(formula = logUSArealgdp ~ logCanadarealgdp + logUKrealgdp)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.039741 -0.014332  0.001084  0.016284  0.033875
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -8.46852    1.07774  -7.858 1.67e-12 ***
## logCanadarealgdp   0.22695    0.07775   2.919  0.00417 **
## logUKrealgdp       0.93307    0.07831  11.914  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01835 on 123 degrees of freedom
## Multiple R-squared:  0.9957, Adjusted R-squared:  0.9956
## F-statistic: 1.429e+04 on 2 and 123 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

## Warning in adfTest(error3, lags = 3, type = c("nc"), title = NULL, descrip
tion =
## NULL): p-value smaller than printed p-value

m9@test$p.value

##
## 0.01

#3.5
diff.logUKrealgdp=diff(logUKrealgdp)
diff.logUSArealgdp=diff(logUSArealgdp)
diff.logCanadarealgdp=diff(logCanadarealgdp)
```

```r
diff.logUKrealgdp.L.1=Lag(diff(logUKrealgdp),k=1)
diff.logUSArealgdp.L.1=Lag(diff(logUSArealgdp),k=1)
diff.logCanadarealgdp.L.1=Lag(diff(logCanadarealgdp),k=1)
error1.L.1=Lag(error1,k=1)
error2.L.1=Lag(error2,k=1)
error3.L.1=Lag(error3,k=1)
fit11 <- lm(formula=diff.logUKrealgdp~diff.logUKrealgdp.L.1+diff.logUSArealgd
p.L.1+diff.logCanadarealgdp.L.1+error2.L.1)
summary(fit11)
```

```
##
## Call:
## lm(formula = diff.logUKrealgdp ~ diff.logUKrealgdp.L.1 + diff.logUSArealgd
p.L.1 +
##     diff.logCanadarealgdp.L.1 + error2.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0167184 -0.0028007 -0.0002558  0.0032301  0.0147306
##
## Coefficients:
##                            Estimate Std. Error t value Pr(>|t|)
## (Intercept)               0.0022858  0.0007188   3.180  0.00188 **
## diff.logUKrealgdp.L.1     0.5079756  0.0811516   6.260 6.33e-09 ***
## diff.logUSArealgdp.L.1    0.0854195  0.0893827   0.956  0.34118
## diff.logCanadarealgdp.L.1  0.0104570  0.0837193   0.125  0.90081
## error2.L.1               -0.0989680  0.0376738  -2.627  0.00975 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005671 on 119 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.3601, Adjusted R-squared:  0.3385
## F-statistic: 16.74 on 4 and 119 DF,  p-value: 6.55e-11
```

```r
fit22 <- lm(formula=diff.logUSArealgdp~diff.logUKrealgdp.L.1+diff.logUSArealg
dp.L.1+diff.logCanadarealgdp.L.1+error3.L.1)
summary(fit22)
```

```
##
## Call:
## lm(formula = diff.logUSArealgdp ~ diff.logUKrealgdp.L.1 + diff.logUSArealg
dp.L.1 +
##     diff.logCanadarealgdp.L.1 + error3.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0204776 -0.0032885  0.0000605  0.0035886  0.0157192
##
## Coefficients:
```

```
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.0029882  0.0007858   3.803 0.000227 ***
## diff.logUKrealgdp.L.1       0.3285073  0.0861542   3.813 0.000219 ***
## diff.logUSArealgdp.L.1      0.1598648  0.0983524   1.625 0.106717
## diff.logCanadarealgdp.L.1   0.1725787  0.0885850   1.948 0.053749 .
## error3.L.1                 -0.0215164  0.0316593  -0.680 0.498064
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.006131 on 119 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.3058, Adjusted R-squared:  0.2825
## F-statistic: 13.11 on 4 and 119 DF,  p-value: 7.094e-09
```

```
fit33 <- lm(formula=diff.logCanadarealgdp~diff.logUKrealgdp.L.1+diff.logUSAre
algdp.L.1+diff.logCanadarealgdp.L.1+error1.L.1)
summary(fit33)
```

```
##
## Call:
## lm(formula = diff.logCanadarealgdp ~ diff.logUKrealgdp.L.1 +
##     diff.logUSArealgdp.L.1 + diff.logCanadarealgdp.L.1 + error1.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0129234 -0.0035559 -0.0001957  0.0033786  0.0171672
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 0.0008393  0.0007464   1.124 0.263079
## diff.logUKrealgdp.L.1       0.2601149  0.0822372   3.163 0.001982 **
## diff.logUSArealgdp.L.1      0.3309112  0.0947854   3.491 0.000676 ***
## diff.logCanadarealgdp.L.1   0.2540305  0.0852632   2.979 0.003502 **
## error1.L.1                 -0.0464788  0.0278360  -1.670 0.097600 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005795 on 119 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.488,  Adjusted R-squared:  0.4708
## F-statistic: 28.36 on 4 and 119 DF,  p-value: < 2.2e-16
```

Question 3

3.1

$$UK\,realgdp_t = \underset{(0.145)}{0.482} + \underset{(0.07494)}{1.42694}\,UK\,realgdp_{t-1} + \underset{(0.02684)}{0.08187}\,USA\,realgdp - \underset{(0.06902)}{0.52537}\,UK\,realgdp_{t-2}$$

$$Canada\,realgdp_t = \underset{(0.08482)}{0.29148}\,UK\,realgdp_{t-1} + \underset{(0.08543)}{1.22221}\,canada\,realgdp_{t-1} + \underset{(0.09622)}{0.37334}\,USA\,realgdp_{t-1}$$
$$- \underset{(0.08797)}{0.28088}\,UK\,realgdp_{t-2} - \underset{(0.08689)}{0.22529}\,canada\,realgdp_{t-2} - \underset{(0.09326)}{0.37887}\,USA\,realgdp_{t-2}$$

$$USA\,realgdp_t = \underset{(0.43779)}{1.17762} + \underset{(0.08076)}{0.38363}\,UK\,realgdp_{t-1} + \underset{(0.07597)}{0.19941}\,Canada\,realgdp_{t-1} + \underset{(0.02958)}{0.98379}\,USA\,realgdp_{t-1} - \underset{(0.08069)}{0.28365}\,UK\,realgdp_{t-2}$$
$$- \underset{(0.07340)}{0.28569}\,Canada\,realgdp_{t-2}$$

3.2 Shock from UK realgdp

- The shock from UK realgdp have positive impact to itself for the whole period.
- The shock from UK real gdp have positive impact to Canada realgdp and USA realgdp from period 1 to the end of period.

Shock from Canada real gdp

- it have slightly positive impact to UK realgdp from period 1 to the end of period.
- it have significantly positive impact to Canada real gdp for the whole period.
- it have moderately positive impact to USA real gdp from initial to period 7 before generate negative impact.

Shock from USA real gdp

- it have significantly positive impact to UK realgdp, Canada real gdp and USA real gdp.

3.3 For UK real gdp

Majority impact come from $\varepsilon_{UK realgdp}$ itself with 96−100% that make us forecast error.

For Canada real gdp

Majority impact come from $\varepsilon_{canada realgdp}$ itself with 70−92% and $\varepsilon_{UK realgdp}$ with around 5−24%. The rest of impact come from $\varepsilon_{USA realgdp}$ (3-5%) that make us forecast error.

For USA real gdp

Majority impact come from $\varepsilon_{USA realgdp}$ itself with 45−80% and $\varepsilon_{UK realgdp}$ with around 3−43%. The rest of impact come from $\varepsilon_{canada realgdp}$ with around 11−21% that make us forecast error.

3.4 First we check each variable by performed Dickey-Fuller test

For log UK realgdp

H_0: there is unit-root in this series

H_1: no unit root

$\alpha = 0.05$

p-value = 0.7535667

Since p-value is higher than $\alpha = 0.05$.

We can not reject H_0 at 95% CI.

Therefore, there is unit-root in this series.

For logCanada real gdp

H_0: there is unit-root in this series

H_1: no unit root

$\alpha = 0.05$

p-value = 0.3721507

Since p-value is higher than $\alpha = 0.05$.

We can not reject H_0 at 95% CI.

Therefore, there is unit-root in this series.

For logUSA real gdp

H_0: there is unit-root in this series

H_1: no unit root

$\alpha = 0.05$

p-value = 0.804053

Since p-value is higher than $\alpha = 0.05$.

We can not reject H_0 at 95% CI.

Therefore, there is unit-root in this series.

After we use first difference to each series and test ADF again, we get p-value = 0.01 for all series. Then, we can conclude that the integrated order I(p) of each series is I(1)

Now we perform the model and find residual term. Next, we test unit-root in residual term by ADF. As a result, we get p-value = 0.02300044, 0.01, 0.01 which all reject H_0 and no unit-root in each residual term; I(0). This mean that log UK realgdp, log canada realgdp and log USA real gdp is cointegrate with CI(1,1)

3.5

$$\Delta \log \widehat{UKrealgdp} = \underset{(0.0007188)}{0.0022858} + \underset{(0.0811516)}{0.5079756}\,\Delta \log UKrealgdp_{t-1} + \underset{(0.0893827)}{0.0854195}\,\Delta \log USArealgdp_{t-1} + \underset{(0.0837193)}{0.010457}\,\Delta \log canadarealgdp_{t-1} - \underset{(0.0376738)}{0.098968}\,e_{t-1}$$

- If there is no change in other country real gdp and missmatch between actual log UKrealgdp and estimate model Change in log UK real gdp is 0.0022858
- All of the change in other countries real gdp in last period have positive impact to change in UK real gdp in this period. However, the actual real gdp of UK In last period exceed the expectation, so it will adjust to the normal by decrease in this period. This may because cash flow in to UK and make return go down, while return in other country go up, so people move out from UK in this period

$$\Delta \log \widehat{USArealgdp} = \underset{(0.0007858)}{0.0029882} + \underset{(0.0861542)}{0.3285073}\,\Delta \log UKrealgdp_{t-1} + \underset{(0.0983524)}{0.1598648}\,\Delta \log USArealgdp_{t-1} + \underset{(0.0885850)}{0.1725787}\,\Delta \log Canadarealgdp_{t-1} - \underset{(0.0316593)}{0.0215164}\,e_{t-1}$$

- If there is no change in other country real gdp and missmatch between actual log USArealgdp and estimate model Change in log USA real gdp is 0.0029882
- All of the change in other countries real gdp in last period have positive impact to change in USA real gdp in this period. However, the actual real gdp of USA In last period exceed the expectation, so it will adjust to the normal by decrease in this period. This may because cash flow in to USA and make return go down, while return in other country go up, so people move out from USA in this period

$$\Delta \log \widehat{Canadagdp} = \underset{(0.0007464)}{0.0008393} + \underset{(0.0822372)}{0.2601149}\,\Delta \log UKrealgdp_{t-1} + \underset{(0.0947554)}{0.3309112}\,\Delta \log USArealgdp_{t-1} + \underset{(0.0852632)}{0.2540305}\,\Delta \log canadarealgdp_{t-1} - \underset{(0.027836)}{0.0464788}\,e_{t-1}$$

- If there is no change in other country real gdp and missmatch between actual log canadarealgdp and estimate model Change in log Canada real gdp is 0.0008393
- All of the change in other countries real gdp in last period have positive impact to chan in Canada real gdp in this period. However, the actual real gdp of Canada In last period exceed the expectation, so it will adjust to the normal by decrease in this period. This may because cash flow in to Canada and make return go down, while return in other country go up, so people move out from Canada in this period