435-exam-1.R

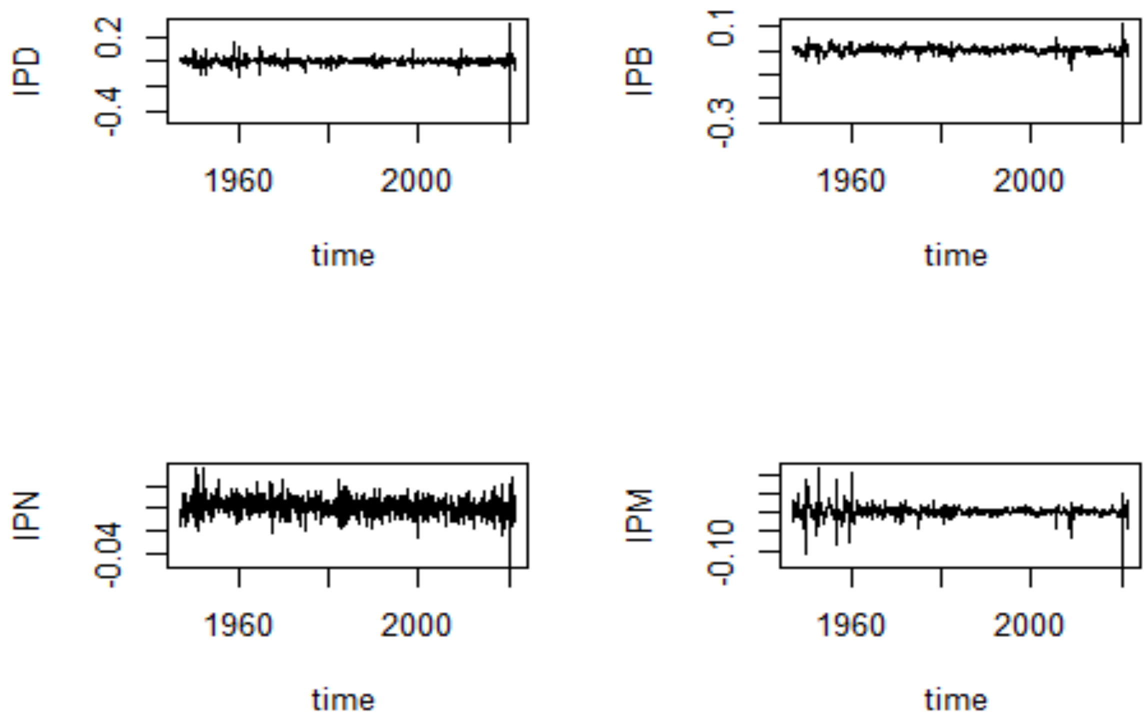
ASUS

2021-06-08

```
library(fBasics)

## Loading required package: timeDate

## Loading required package: timeSeries

library(timeDate)
library(timeSeries)
library(fGarch)

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

library(quantmod)

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: TTR

##
## Attaching package: 'TTR'

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

## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo

library(forecast)

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

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

## [1] "IPNCONGD"

dim(IPNCONGD)

## [1] 892   1

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

## [1] "IPBUSEQ"

dim(IPBUSEQ)

## [1] 892   1

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

## [1] "IPMAT"

dim(IPMAT)

## [1] 988   1

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

## [1] 892   4

colnames(IP) <- c("IPD","IPN","IPB","IPM")

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 1.1
IP=log(IP)
zt=diffM(IP)
tdx=c(1:891)/12+1947
MTSplot(zt,tdx)
```



1.1)

All 4 series portray a mean reverting pattern and constant mean, but they also portray a property of volatility clustering. All four series portray quite similar pattern of volatility, especially during pandemic shock of 2020 where all series experience a surge in volatility.

IPD - durable good faced overall low volatility as this take time for consumer to make a decision as it yield utility over longer period

IPN - non-durable good face high volatility as this fluctuate a lot with consumer demand

IPM - face high fluctuation during great depression

```r
#Question 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 -34.0260 -34.0260 -34.0260   0.0000  0.0000
##  [2,]  1 -34.1944 -34.1083 -34.1615 178.2331  0.0000
##  [3,]  2 -34.3289 -34.1567 -34.2631 147.9623  0.0000
##  [4,]  3 -34.3844 -34.1262 -34.2857  79.0713  0.0000
##  [5,]  4 -34.3851 -34.0408 -34.2535  31.4646  0.0117
##  [6,]  5 -34.4369 -34.0066 -34.2724  75.1412  0.0000
##  [7,]  6 -34.4582 -33.9418 -34.2608  48.7671  0.0000
##  [8,]  7 -34.4429 -33.8405 -34.2127  17.5535  0.3507
##  [9,]  8 -34.4246 -33.7361 -34.1614  14.8136  0.5383
## [10,]  9 -34.4189 -33.6444 -34.1229  25.4492  0.0623
## [11,] 10 -34.4146 -33.5540 -34.0857  26.3997  0.0487
## [12,] 11 -34.4087 -33.4620 -34.0469  24.9845  0.0701
## [13,] 12 -34.4155 -33.3829 -34.0209  35.4558  0.0034
## [14,] 13 -34.4147 -33.2960 -33.9871  28.9289  0.0244
```

```r
m1 <- VAR(zt, p=6)
```

```
## Constant term:
## Estimates:  0.002040297 0.001573752 0.001125976 0.001053311
## Std.Error:  0.001218999 0.0003035209 0.0006279805 0.0005838232
## 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,] 9.555893e-04 8.261288e-05 3.583451e-04 2.292864e-04
## [2,] 8.261288e-05 5.924362e-05 4.307313e-05 3.083878e-05
## [3,] 3.583451e-04 4.307313e-05 2.536044e-04 1.255761e-04
## [4,] 2.292864e-04 3.083878e-05 1.255761e-04 2.191932e-04
##
## det(SSE) =  8.698098e-16
## AIC =  -34.46277
## BIC =  -33.94642
## HQ  =  -34.26543
```

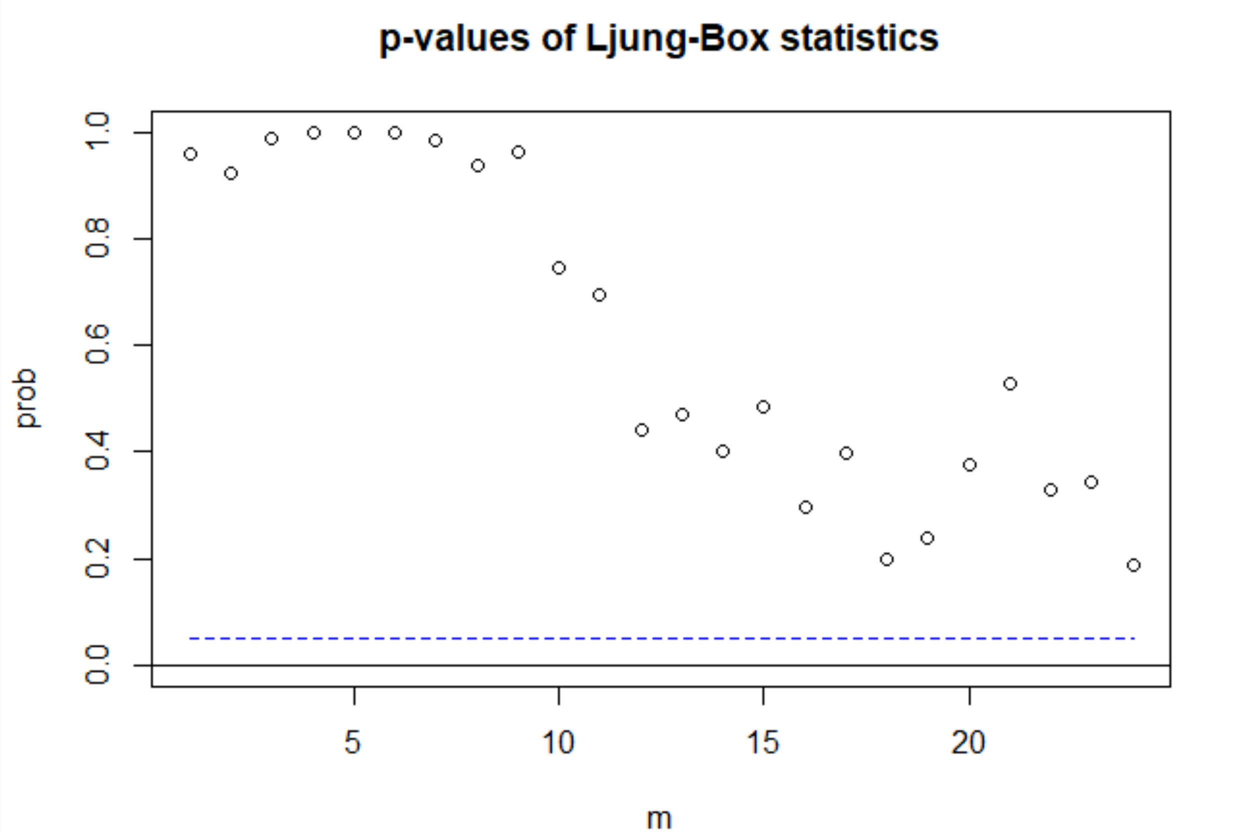
```
m2 <- refVAR(m1, thres=1.645)
```

```
## Constant term:
## Estimates:  0.002985377 0.00157489 0.001542269 0.001081254
## Std.Error:  0.001122942 0.0002740226 0.000568169 0.0005440154
## 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,] 0.0009672303 8.232940e-05 3.613719e-04 2.300248e-04
## [2,] 0.0000823294 6.005646e-05 4.296673e-05 3.098108e-05
## [3,] 0.0003613719 4.296673e-05 2.570458e-04 1.263962e-04
## [4,] 0.0002300248 3.098108e-05 1.263962e-04 2.205923e-04
##
```

```
## det(SSE) =  9.263687e-16
## AIC =  -34.52996
## BIC =  -34.32557
## HQ  =  -34.45185
```

MTSdiag(m2)



```r
#Question 1.3 & 1.2
detach("package:MTS", unload = TRUE)
require(vars)

varfit1 <- VAR(zt,p=6)

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: 10294.21
## 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
0.6809 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.l5 + IPD.l6 + IPB.l6 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPN.l1  0.225501   0.136982   1.646 0.100080
## IPM.l1  0.417370   0.070149   5.950 3.88e-09 ***
## IPD.l2 -0.212805   0.033654  -6.323 4.08e-10 ***
## IPD.l3 -0.131341   0.037913  -3.464 0.000558 ***
## IPM.l3  0.137685   0.080379   1.713 0.087078 .
## IPB.l4 -0.167389   0.074612  -2.243 0.025117 *
## IPM.l4  0.187141   0.081136   2.307 0.021315 *
## IPB.l5 -0.228314   0.065803  -3.470 0.000547 ***
## IPD.l6  0.159243   0.045596   3.492 0.000503 ***
## IPB.l6 -0.326152   0.086910  -3.753 0.000186 ***
## const   0.002985   0.001123   2.659 0.007992 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.0313 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   0.001575   0.000274   5.747 1.25e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.007776 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.l3 + IPB.l4 + IPM.l4 + IPD.l5 + IPM.l5 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.0340237  0.0198195   1.717  0.08639 .
## IPM.l1  0.2248565  0.0412946   5.445 6.73e-08 ***
## IPD.l2 -0.1294933  0.0250552  -5.168 2.93e-07 ***
## IPB.l2  0.1374148  0.0483678   2.841  0.00460 **
## IPM.l2  0.1052188  0.0430538   2.444  0.01473 *
## IPD.l3 -0.1128245  0.0245494  -4.596 4.95e-06 ***
## IPB.l3  0.1492588  0.0478251   3.121  0.00186 **
## IPM.l3  0.1339244  0.0439313   3.048  0.00237 **
## IPB.l4 -0.0826714  0.0388569  -2.128  0.03365 *
## IPM.l4  0.1174801  0.0421961   2.784  0.00548 **
## IPD.l5 -0.0407343  0.0197655  -2.061  0.03961 *
## IPM.l5  0.1048733  0.0416695   2.517  0.01202 *
## const   0.0015423  0.0005682   2.714  0.00677 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01615 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.l4 + IPD.l5 + IPM.l5 + IPD.l6 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.039514   0.018317   2.157  0.03126 *
## IPN.l1  0.145072   0.068659   2.113  0.03489 *
## IPM.l1  0.199846   0.038061   5.251 1.90e-07 ***
## IPN.l2  0.140646   0.069767   2.016  0.04411 *
## IPB.l2  0.073715   0.036034   2.046  0.04108 *
## IPM.l2 -0.100166   0.038978  -2.570  0.01034 *
## IPN.l3  0.184555   0.065390   2.822  0.00488 **
## IPD.l4  0.044260   0.016285   2.718  0.00670 **
## IPD.l5  0.052412   0.017712   2.959  0.00317 **
## IPM.l5 -0.167583   0.037860  -4.426 1.08e-05 ***
## IPD.l6  0.039052   0.015999   2.441  0.01485 *
## const   0.001081   0.000544   1.988  0.04717 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01495 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 9.953e-04 8.472e-05 3.719e-04 2.367e-04
## IPN 8.472e-05 6.180e-05 4.422e-05 3.188e-05
## IPB 3.719e-04 4.422e-05 2.645e-04 1.301e-04
## IPM 2.367e-04 3.188e-05 1.301e-04 2.270e-04
##
## 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)
```

Q 1.2) Ljung-Box of e_t : not rejecting H_0: No correlation among $\hat{e}_t$ for all lags → model adequate with 95% CI

All fitted model with threshold t=1.645

Fitted model for IPD:

IPD = 0.225501IPN.l1 + 0.417370IPM.l1 - 0.212805IPD.l2 - 0.131341IPD.l3 + 0.137685IPM.l3 - 0.167389IPB.l4 + 0.187141IPM.l4 - 0.228314IPB.l5 + 0.159243IPD.l6 - 0.326152IPB.l6 + 0.002985

```
##
##          Estimate Std. Error t value Pr(>|t|)
## IPN.l1  0.225501   0.136982   1.646 0.100080
## IPM.l1  0.417370   0.070149   5.950 3.88e-09 ***
## IPD.l2 -0.212805   0.033654  -6.323 4.08e-10 ***
## IPD.l3 -0.131341   0.037913  -3.464 0.000558 ***
## IPM.l3  0.137685   0.080379   1.713 0.087078 .
## IPB.l4 -0.167389   0.074612  -2.243 0.025117 *
## IPM.l4  0.187141   0.081136   2.307 0.021315 *
## IPB.l5 -0.228314   0.065803  -3.470 0.000547 ***
## IPD.l6  0.159243   0.045596   3.492 0.000503 ***
## IPB.l6 -0.326152   0.086910  -3.753 0.000186 ***
## const   0.002985   0.001123   2.659 0.007992 **
```

Fitted model for IPN:

IPN = 0.035802IPD.l1 - 0.166227IPN.l1 + 0.061584IPN.l4 + 0.030534IPM.l5 + 0.025487IPD.l6 + 0.001575

```
##          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   0.001575   0.000274   5.747 1.25e-08 ***
```

Fitted model for IPB:

IPB = 0.0340237IPD.l1 + 0.2248565IPM.l1 - 0.1294933IPD.l2 + 0.1374148IPB.l2 + 0.1052188IPM.l2 - 0.1128245IPD.l3 + 0.1492588IPB.l3 + 0.1339244IPM.l3 - 0.0826714IPB.l4 + 0.1174801IPM.l4 - 0.0407343IPD.l5 + 0.1048733IPM.l5 + 0.0015423

```
##
##           Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.0340237  0.0198195   1.717  0.08639 .
## IPM.l1  0.2248565  0.0412946   5.445 6.73e-08 ***
## IPD.l2 -0.1294933  0.0250552  -5.168 2.93e-07 ***
## IPB.l2  0.1374148  0.0483678   2.841  0.00460 **
## IPM.l2  0.1052188  0.0430538   2.444  0.01473 *
## IPD.l3 -0.1128245  0.0245494  -4.596 4.95e-06 ***
## IPB.l3  0.1492588  0.0478251   3.121  0.00186 **
```

```
## IPM.l3   0.1339244  0.0439313   3.048  0.00237 **
## IPB.l4 -0.0826714  0.0388569  -2.128  0.03365 *
## IPM.l4   0.1174801  0.0421961   2.784  0.00548 **
## IPD.l5 -0.0407343  0.0197655  -2.061  0.03961 *
## IPM.l5   0.1048733  0.0416695   2.517  0.01202 *
## const    0.0015423  0.0005682   2.714  0.00677 **
```

Fitted model for IPM:

IPM = 0.039514IPD.l1 + 0.145072IPN.l1 + 0.199846IPM.l1 + 0.140646IPN.l2 + 0.073715IPB.l2 - 0.100166IPM.l2 + 0.184555IPN.l3 + 0.044260IPD.l4 + 0.052412IPD.l5 - 0.167583IPM.l5 + 0.039052IPD.l6 + 0.001081

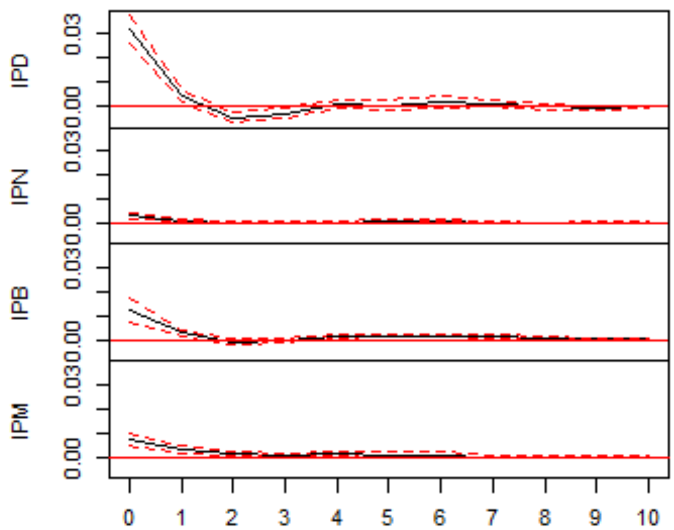
```
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1   0.039514   0.018317   2.157  0.03126 *
## IPN.l1   0.145072   0.068659   2.113  0.03489 *
## IPM.l1   0.199846   0.038061   5.251 1.90e-07 ***
## IPN.l2   0.140646   0.069767   2.016  0.04411 *
## IPB.l2   0.073715   0.036034   2.046  0.04108 *
## IPM.l2 -0.100166   0.038978  -2.570  0.01034 *
## IPN.l3   0.184555   0.065390   2.822  0.00488 **
## IPD.l4   0.044260   0.016285   2.718  0.00670 **
## IPD.l5   0.052412   0.017712   2.959  0.00317 **
## IPM.l5 -0.167583   0.037860  -4.426 1.08e-05 ***
## IPD.l6   0.039052   0.015999   2.441  0.01485 *
## const    0.001081   0.000544   1.988  0.04717 *
```

$$\begin{bmatrix}1&0&0&0\\ b_{21}&1&0&0\\ b_{31}&b_{32}&1&0\\ b_{41}&b_{42}&b_{43}&1\end{bmatrix}\begin{bmatrix}IPD\\ IPN\\ IPB\\ IPM\end{bmatrix}$$

Q1.3) Impact of IPD shock

- IPD: strong significant positive impact from shock itself to IPD, especially from concurrent period and the magnitude start to die down as lag increase

Orthogonal Impulse Response from IPD



95 % Bootstrap CI, 100 runs

Durable consumption shock transmit easily to all sector: business and nated

- IPN:

  there is concurrent effect/impact from shock in IPD to IPN. The impact is weakly positive significantly for all current and first few lag. The magnitude of impact die down as lag increase

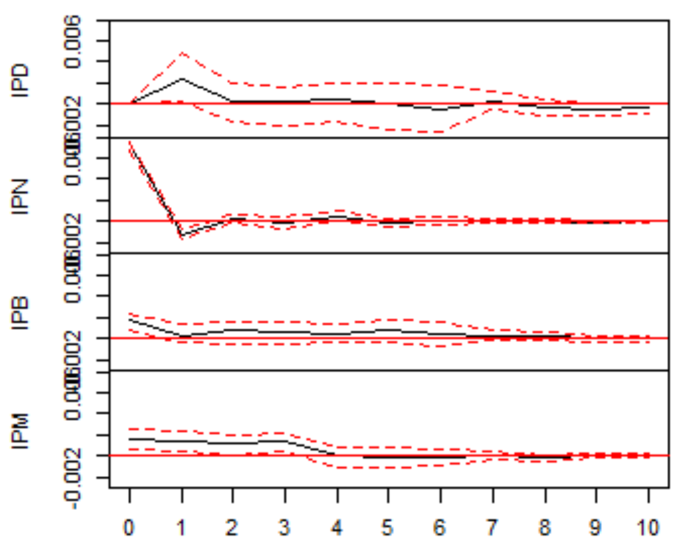
- IPB:

  there is concurrent effect/impact from shock in IPD to IPB. The impact is strongly positive significantly for all current and first few lags. The magnitude of impact die down as lag increase

- IPM:

  there is concurrent effect/impact from shock in IPD to IPM. The impact is weakly positive significantly for all current and first few lags. The magnitude of impact die down as lag increase

Orthogonal Impulse Response from IPN



95 % Bootstrap CI, 100 runs

Impact of IPN shock

- IPN: strong significant positive impact from shock itself to IPN, especially from concurrent period and the magnitude start to die down as lag increase

- IPD:

  there is no concurrent effect/impact from shock in IPN to IPD. The impact is weakly positive significantly for first few lag. The magnitude of impact die down as lag increase

- IPB:

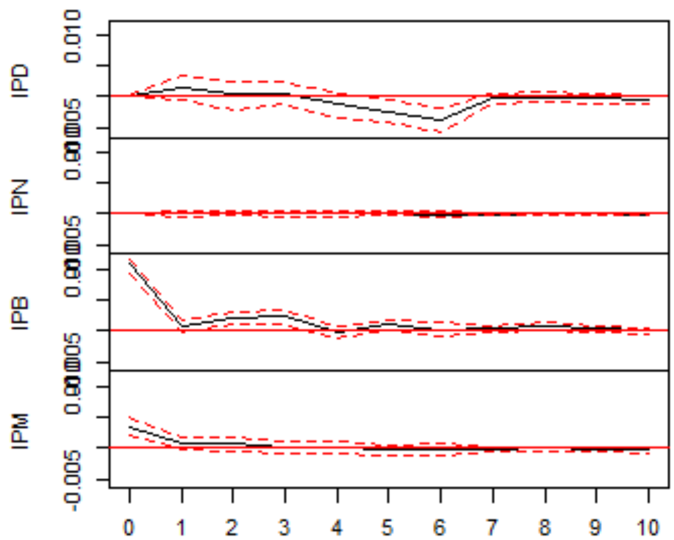
  there is concurrent effect/impact from shock in IPN to IPB. The impact is weakly positive significantly for current and first lag. The magnitude of impact die down as lag increase

- IPM:

  there is concurrent effect/impact from shock in IPN to IPM. The impact is weakly positive significantly for all current and first few lags. The magnitude of impact die down as lag increase

Business shock only affect the material in same direction

Orthogonal Impulse Response from IPB

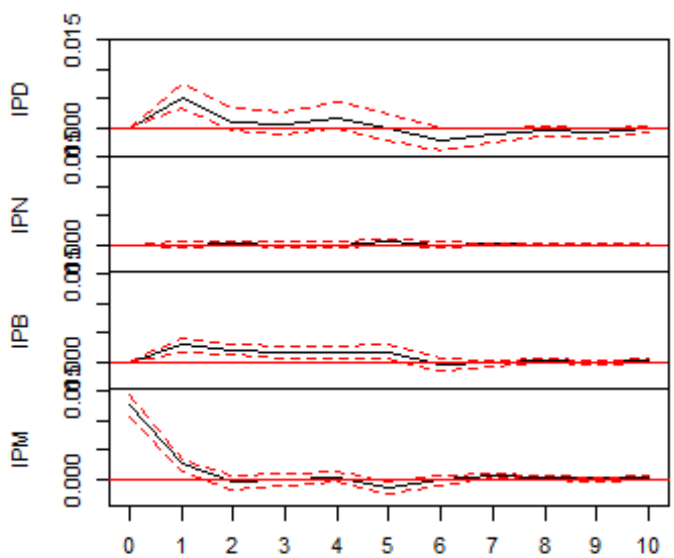


95 % Bootstrap CI, 100 runs

Impact of IPB shock

- IPB: strong significant positive impact from shock itself to IPB, especially from concurrent period and the magnitude start to die down as lag increase
- IPD: there is no concurrent effect / impact from shock in IPB to IPD. The impact is insignificant to IPD
- IPN: There is no impact from IPB shock passed to IPN
- IPM: there is concurrent effect / impact from shock in IPB to IPM. The impact is weekly positive significantly for all current and first lag. The magnitude of impact die down as lag increase

Orthogonal Impulse Response from IPM



95 % Bootstrap CI, 100 runs

Impact of IPM shock

- IPM: strong significant positive impact from shock itself to IPM, especially from concurrent period and the magnitude start to die down as lag increase
- IPD: there is no concurrent effect / impact from shock in IPM to IPD. The impact is weekly positive significant to IPD for first few lag. and the magnitude start to die down as lag increase
- IPN: There is no impact from IPB shock passed to IPN
- IPB: there is no concurrent effect / impact from shock in IPM to IPB. The impact is weekly positive significantly for first few lags. The magnitude of impact die down as lag increase

Material shock affect production → business → sale of durable good in the same direction

source of deviation from Actual future value of predicted future IPD
for 1-step ahead : all deviation come only from shock of IPD itself
for 2-step ahead : deviation come from IPD shock ~ 98.2 %
and IPN shock ~ 0.5 %

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

For further predict, the source of deviation come from IPN, IPM, IPB increasingly
and from IPD shock decreasingly

```
## $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
##
```

source of deviation from Actual future value of predicted future IPN
for 1-step ahead : deviation come from IPD shock ~ 10 %
and IPN shock ~ 90 %

For further predict, the source of deviation come from IPD, IPM, IPB increasingly
and from IPN shock decreasingly

source of deviation from Actual future value of predicted future IPB for 1-step ahead : deviation came from IPD shock ~ 53% and IPB shock ~ 47%.

For further predict, the source of deviation came from IPD, IPM, IPN increasingly and from IPB shock decreasingly

```
## $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
```

source of deviation from Actual future value of predicted future IPN for 1-step ahead : deviation came from IPD shock ~ 23% and IPM shock ~ 69%.

For further predict, the source of deviation came from IPD, IPM, IPB increasingly and from IPN shock decreasingly

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

m1 <- VAR(zt, p=6)

m2 <- refVAR(m1, thres=1.645)

VARpred(m2, h = 6, orig = 879)

## orig  879
## Forecasts at origin:  879
##             IPD        IPN       IPB       IPM
## [1,] -0.019893 -6.727e-03 -0.036293 -0.054249
## [2,]  0.109914  1.959e-03  0.001568 -0.029138
## [3,]  0.066928  4.296e-03 -0.014738 -0.013429
## [4,]  0.017880 -1.829e-05 -0.014463 -0.021403
## [5,]  0.020648 -6.616e-03 -0.029263 -0.010580
## [6,]  0.004245 -1.017e-02 -0.026670 -0.004851
## Standard Errors of predictions:
##         [,1]     [,2]    [,3]    [,4]
## [1,] 0.03110 0.007750 0.01603 0.01485
## [2,] 0.03185 0.007872 0.01652 0.01545
## [3,] 0.03229 0.007880 0.01681 0.01557
## [4,] 0.03246 0.007882 0.01704 0.01564
## [5,] 0.03254 0.007896 0.01717 0.01569
## [6,] 0.03264 0.007914 0.01733 0.01579
## Root mean square errors of predictions:
##         [,1]     [,2]    [,3]    [,4]
## [1,] 0.03154 0.007859 0.01626 0.01506
## [2,] 0.09531 0.019731 0.05446 0.05746
## [3,] 0.07630 0.009060 0.04397 0.03021
## [4,] 0.05450 0.008272 0.04063 0.02471
## [5,] 0.04406 0.010014 0.03178 0.02300
## [6,] 0.04635 0.010490 0.03532 0.02800
## Observations, predicted values,     errors, and MSE
##      time     obs    fcst     err     obs    fcst     err     obs    fcst
## case  880  0.1801 -0.0199  0.2000  0.0174 -0.0067  0.0242  0.0982 -0.0363
## case  881  0.3169  0.1099  0.2070  0.0213  0.0020  0.0194  0.1177  0.0016
## case  882  0.1506  0.0669  0.0837  0.0105  0.0043  0.0063  0.0746 -0.0147
## case  883 -0.0098  0.0179 -0.0276  0.0127  0.0000  0.0127  0.0256 -0.0145
## case  884 -0.0135  0.0206 -0.0341 -0.0153 -0.0066 -0.0087 -0.0014 -0.0293
## case  885  0.0088  0.0042  0.0046  0.0010 -0.0102  0.0111  0.0084 -0.0267
##         err     obs    fcst    err
## case 0.1345 -0.0104 -0.0542 0.0438
## case 0.1161  0.0519 -0.0291 0.0810
## case 0.0894  0.0366 -0.0134 0.0500
## case 0.0401  0.0070 -0.0214 0.0284
## case 0.0278  0.0008 -0.0106 0.0113
## case 0.0351  0.0110 -0.0049 0.0159
```

```
95% CI forecasting:

##            IPD

## [1,] (-0.019893-1.96(0.03110),-0.019893+1.96(0.03110))
## [2,]  (0.109914-1.96(0.03185),0.109914+1.96(0.03185))
## [3,]  (0.066928-1.96(0.03229), 0.066928+1.96(0.03229))
## [4,]  (0.017880-1.96(0.03229), 0.017880+1.96(0.03229))
## [5,]  (0.020648-1.96(0.03254), 0.020648+1.96(0.03254))
## [6,]  (0.004245-1.96(0.03264), 0.004245+1.96(0.03264))


##            IPN
## [1,]  (-6.727e-03-1.96(0.007750),-6.727e-03+1.96(0.007750))
## [2,]  (1.959e-03-1.96(0.007872), 1.959e-03+1.96(0.007872))
## [3,]  (4.296e-03-1.96(0.007880), 4.296e-03+1.96(0.007880))
## [4,]  (-1.829e-05-1.96(0.007882), -1.829e-05+1.96(0.007882))
## [5,]  (-6.616e-03-1.96(0.007896),  -6.616e-03+1.96(0.007896))
## [6,]  (-1.017e-02-1.96(0.007914), -1.017e-02+1.96(0.007914))

##            IPB
## [1,] (-0.036293-1.96(0.01603), -0.036293+1.96(0.01603))
## [2,] (0.001568-1.96(0.01652), 0.001568+1.96(0.01652))
## [3,] (-0.014738-1.96(0.01681),  -0.014738+1.96(0.01681))
## [4,] (-0.014463-1.96(0.01704), -0.014463+1.96(0.01704))
## [5,] (-0.029263-1.96(0.01717), -0.029263+1.96(0.01717))
## [6,] (-0.026670-1.96(0.01733), -0.026670+1.96(0.01733))

             IPM
## [1,] (-0.054249-1.96(0.01485), -0.054249+1.96(0.01485))
## [2,] (-0.029138-1.96(0.01545), -0.029138-1.96(0.01545))
## [3,] (-0.013429-1.96(0.01557), -0.013429-1.96(0.01557))
## [4,] (-0.021403-1.96(0.01564), -0.021403-1.96(0.01564))
## [5,] (-0.010580-1.96(0.01569), -0.010580-1.96(0.01569))
## [6,] (-0.004851-1.96(0.01579), -0.004851-1.96(0.01579))

## Standard Errors of predictions:
##          [,1]     [,2]    [,3]    [,4]
## [1,] 0.03110 0.007750 0.01603 0.01485
## [2,] 0.03185 0.007872 0.01652 0.01545
## [3,] 0.03229 0.007880 0.01681 0.01557
## [4,] 0.03246 0.007882 0.01704 0.01564
## [5,] 0.03254 0.007896 0.01717 0.01569
## [6,] 0.03264 0.007914 0.01733 0.01579
```

435-exam-2.R

ASUS

2021-06-08

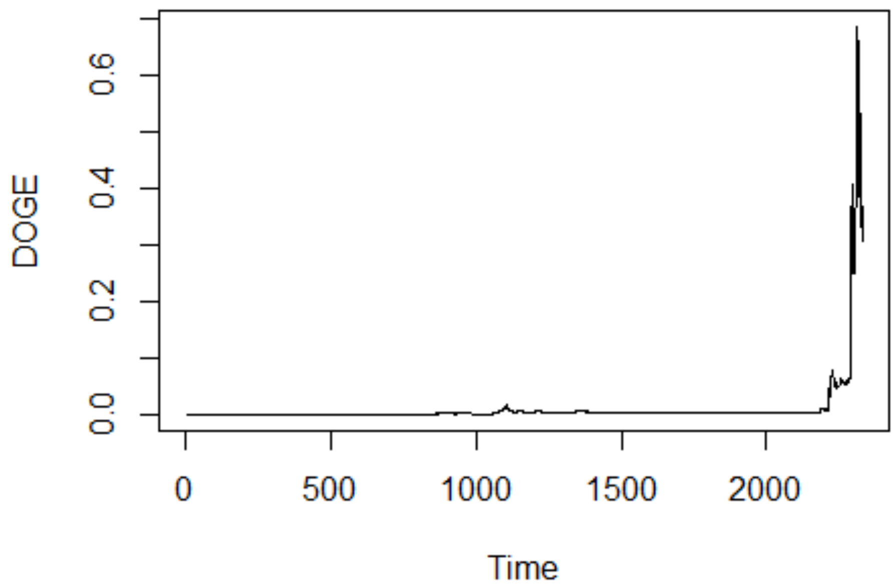
```
library(fBasics)

library(timeDate)
library(timeSeries)
library(fGarch)

library(quantmod)

library(forecast)
library(fUnitRoots)

getSymbols('DOGE-USD',from="2015-01-01",to="2021-05-24")

## [1] "DOGE-USD"

write.csv(`DOGE-USD`,"C:\\Users\\ASUS\\Desktop\\MyData.csv", row.names =
TRUE, col.names = TRUE)

sum(is.na(`DOGE-USD`))

## [1] 24

DOGE <- na.omit(`DOGE-USD`)
DOGE=DOGE[,6]

#1 Basic Statistics summary of DOGE price
basicStats(DOGE)

##                DOGE.USD.Adjusted
## nobs                 2332.000000
## NAs                     0.000000
## Minimum                 0.000087
## Maximum                 0.684777
## 1. Quartile             0.000226
## 3. Quartile             0.003080
## Mean                    0.010691
## Median                  0.002149
## Sum                    24.932087
## SE Mean                 0.001142
## LCL Mean                0.008452
## UCL Mean                0.012930
## Variance                0.003041
## Stdev                   0.055141
```

```
## Skewness                8.081347
## Kurtosis               70.417350
```

```
ts.plot(DOGE)
```

2.Plot of DOGE price over time

Showing a pattern of upward trend and non-constant mean and variance



3. Stationarity test of DOGE price series

```
m0 <- adfTest(DOGE, lags=2, type=c("c"), title=NULL)

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

m0@test$p.value

##
## 0.5023077 > 0.05. H0: the series is unit root is not rejected at 0.05
level of significance. That is, DOGE price series is non-stationary with 95%
CI
```

4. Transform into log return series (rtn)

```
logDOGE <- log(as.numeric(DOGE))
rtn <- diff(logDOGE)
```

```
5. Statistical summary of DOGE log return (rtn)
```

```
basicStats(rtn)
```

```
##                      rtn
## nobs        2331.000000
## NAs            0.000000
## Minimum       -0.515118
## Maximum        1.323469
## 1. Quartile   -0.021259
## 3. Quartile    0.018944
## Mean           0.003261
## Median         0.000000
## Sum            7.601345
## SE Mean        0.001537
## LCL Mean       0.000247
## UCL Mean       0.006275
## Variance       0.005506
## Stdev          0.074205
## Skewness       3.630047
## Kurtosis      55.832326
```

```
6. Test H0: mean of return is equal to zero
```

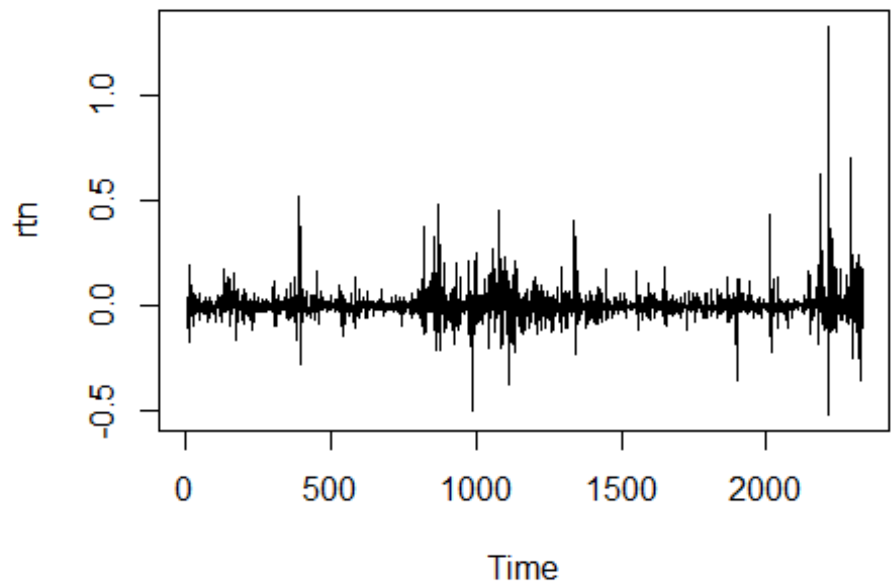
```
t.test(rtn)
```

```
##
##  One Sample t-test
##
## data:  rtn
## t = 2.1217, df = 2330, p-value = 0.03397
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  0.0002470455 0.0062749150
## sample estimates:
##  mean of x
## 0.00326098 < 0.05. H0 is not rejected at 0.05 level of significance. That
is, mean of log return rtn is not equal to zero with 95% CI
```

```
ts.plot(rtn)
```

7. Plot of DOGE log return series

Showing a mean reverting pattern implying constant mean, but with the hint of volatility clustering property (implying ARCH effect)

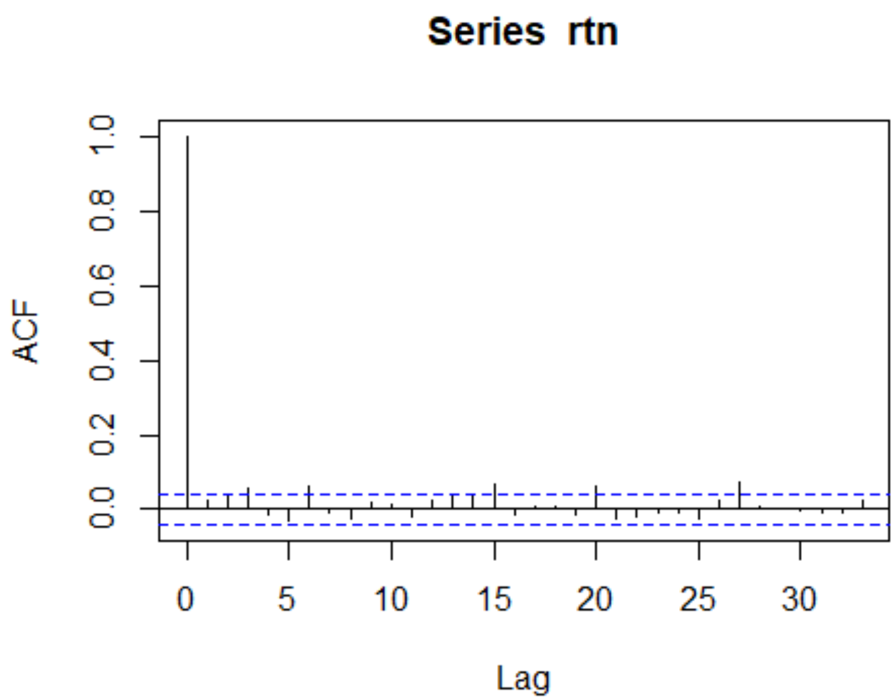


8. Test Stationarity of log return series. H0: The series is non-stationary

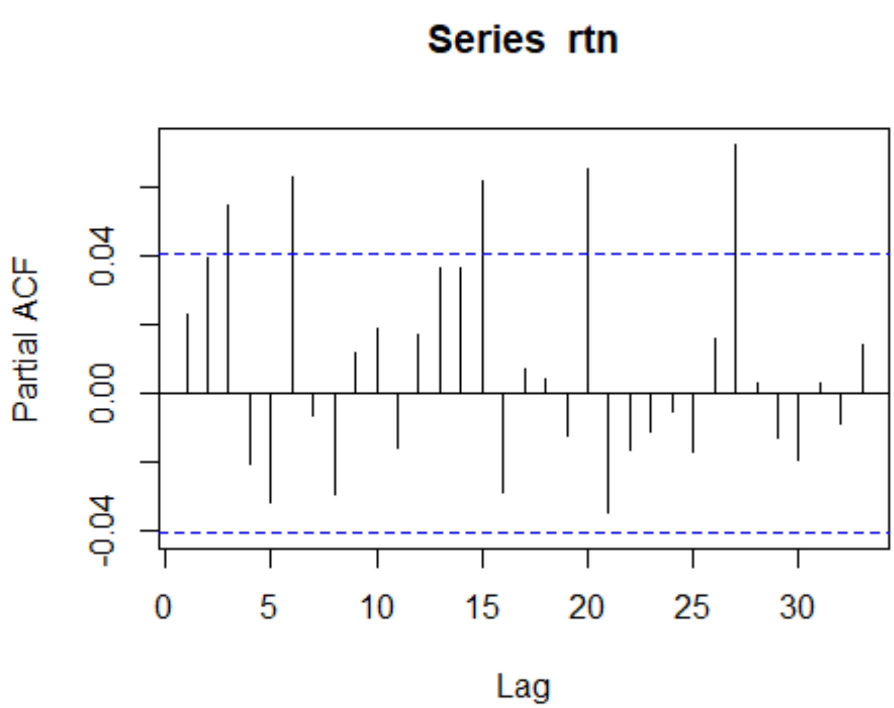
```
m00 <- adfTest(rtn, lags=2, type=c("nc"), title=NULL)

## Warning in adfTest(rtn, lags = 2, type = c("nc"), title = NULL): p-value
smaller
## than printed p-value

m00@test$p.value

##
## 0.01 < 0.05. H0 is rejected at 0.05 level of significance. That is, the
log return series of DOGE is stationary with 95% CI.
```

9. Testing existence of autocorrelation in the log return series

```
acf(rtn)
```




```
pacf(rtn)
```




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

```
##
##  Box-Ljung test
##
## data:  rtn
## X-squared = 27.126, df = 10, p-value = 0.002487
```

H0: There is no autocorrelation in the return series

p-value = 0.002487 < 0.05. H0 is rejected at 0.05 level of significance. That is, this rtn series has autocorrelation in the series. Also, ACF and PACF plotting portray a value of lag correlation that is significantly different from zero.

10. Finding mean equation, using optimal lag generator from auto.arima

```
auto.arima(rtn)
```

```
## Series: rtn
## ARIMA(3,0,3) with non-zero mean
##
## Coefficients:
##           ar1      ar2     ar3     ma1     ma2      ma3    mean
##       -0.1490  -0.6059  0.2796  0.1701  0.6565  -0.2002  0.0033
## s.e.   0.2876   0.1408  0.2526  0.2934  0.1376   0.2635  0.0017
##
## sigma^2 estimated as 0.005459:  log likelihood=2768.69
## AIC=-5521.38   AICc=-5521.31   BIC=-5475.34
```

```
m1=arima(rtn,order=c(3,0,3))
summary(m1)
```

```
##
## Call:
## arima(x = rtn, order = c(3, 0, 3))
##
## Coefficients:
##           ar1      ar2     ar3     ma1     ma2      ma3  intercept
##       -0.1490  -0.6059  0.2796  0.1701  0.6565  -0.2002     0.0033
## s.e.   0.2876   0.1408  0.2526  0.2934  0.1376   0.2635     0.0017
##
## sigma^2 estimated as 0.005443:  log likelihood = 2768.69,  aic = -5521.38
##
## Training set error measures:
##                          ME      RMSE        MAE MPE MAPE      MASE
ACF1
## Training set -1.502702e-05 0.0737762 0.03966208 NaN  Inf 0.6873583
0.0006249006
```

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

```
##
##  Box-Ljung test
```

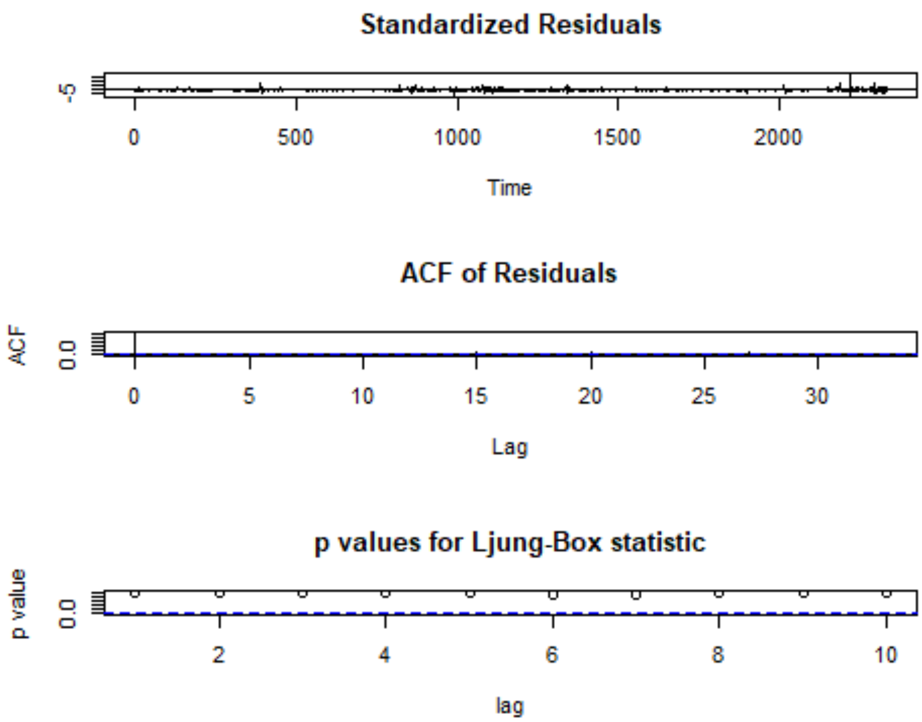
```
##
## data:  m1$residuals
## X-squared = 2.3542, df = 10, p-value = 0.9928
```

Mean Equation ARIMA(3,0,3) is yielded.

Checking Adequacy of model by Ljung Box testing the autocorrelation of residual series. H0: The model is adequate.

p-value = 0.9928 > 0.05. H0 is not rejected at 0.05 level of significance. That is, the model is adequate. Also, ACF of residual series show no value significantly from zero, too.

```
tsdiag(m1)
```



11. Testing the ARCH effect of the model

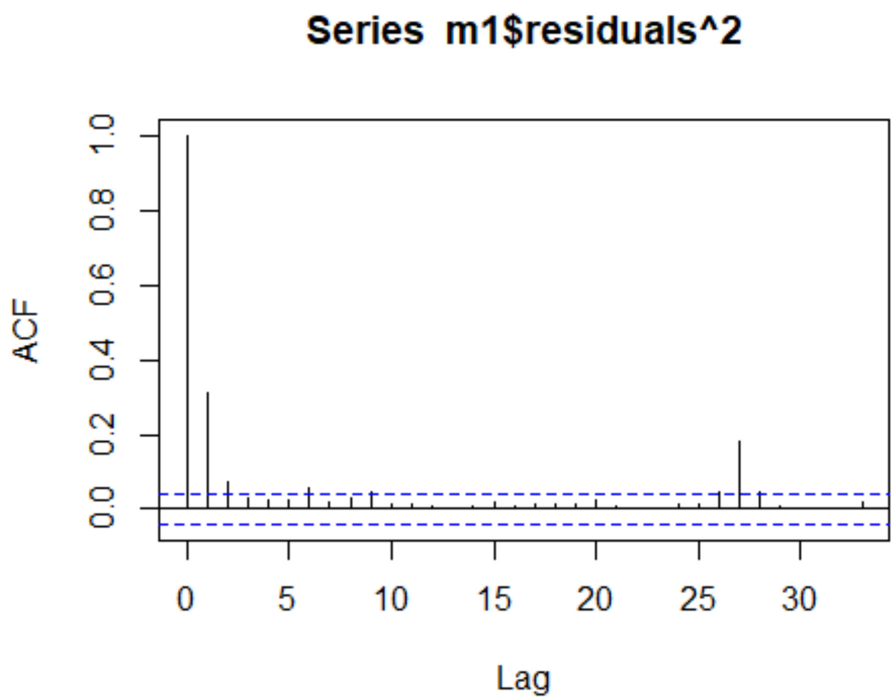
```
Box.test(m1$residuals^2,lag=10,type='Ljung')
```

```
##
##  Box-Ljung test
##
## data:  m1$residuals^2
## X-squared = 261.52, df = 10, p-value < 2.2e-16
```

Testing by Ljung Box checking the autocorrelation in residual squared series

H0: There is No ARCH effect in the series p-value < 2.2e-16 < 0.05. H0 is rejected at 0.05 level of significance. That is, there exists ARCH effect in the log return series. Also, the ACF of residual squared also show value of lag significantly different from zero.

```
acf(m1$residuals^2)
```

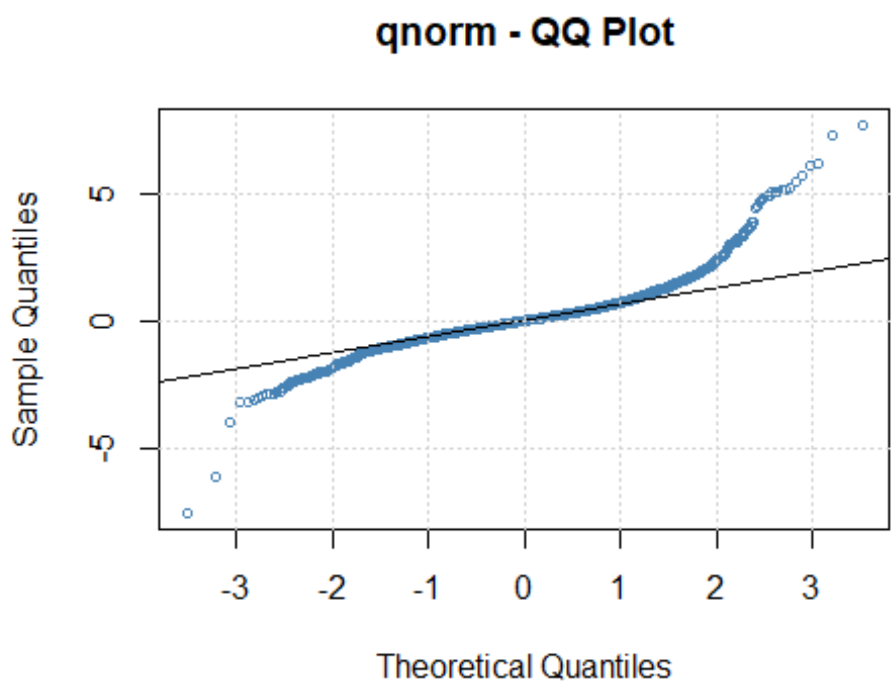


12. Finding complete mean and variance equation

```
m2=garchFit(~arma(3,3)+garch(1,1),data=rtn,trace=F)

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

summary(m2)

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(3, 3) + garch(1, 1), data = rtn, trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(3, 3) + garch(1, 1)
## <environment: 0x000000001e7a7140>
##  [data = rtn]
##
```

```
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1          ar2          ar3          ma1
ma2
## -0.00092389  -0.03084743  -0.50214058   0.43707698  -0.00892956
0.52408632
##         ma3        omega       alpha1        beta1
## -0.45470038   0.00012839   0.29576393   0.73951999
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -9.239e-04   1.050e-03   -0.880  0.37905
## ar1    -3.085e-02   3.133e-01   -0.098  0.92156
## ar2    -5.021e-01   1.705e-01   -2.944  0.00324 **
## ar3     4.371e-01   2.776e-01    1.574  0.11539
## ma1    -8.930e-03   3.085e-01   -0.029  0.97691
## ma2     5.241e-01   1.608e-01    3.260  0.00112 **
## ma3    -4.547e-01   2.773e-01   -1.640  0.10107
## omega   1.284e-04   1.786e-05    7.189 6.54e-13 ***
## alpha1  2.958e-01   2.714e-02   10.899  < 2e-16 ***
## beta1   7.395e-01   1.902e-02   38.876  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3681.162    normalized:  1.57922
##
## Description:
##  Tue Jun 08 23:16:02 2021 by user: ASUS
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  9755.223  0
##  Shapiro-Wilk Test  R    W      0.8765047 0
##  Ljung-Box Test     R    Q(10)  25.13979  0.005086737
##  Ljung-Box Test     R    Q(15)  44.6845   8.586157e-05
##  Ljung-Box Test     R    Q(20)  46.78782  0.0006276965
##  Ljung-Box Test     R^2  Q(10)  16.28194  0.09184104
##  Ljung-Box Test     R^2  Q(15)  16.88205  0.3259643
##  Ljung-Box Test     R^2  Q(20)  19.27288  0.5041575
##  LM Arch Test       R    TR^2   15.96671  0.1927654
##
## Information Criterion Statistics:
```

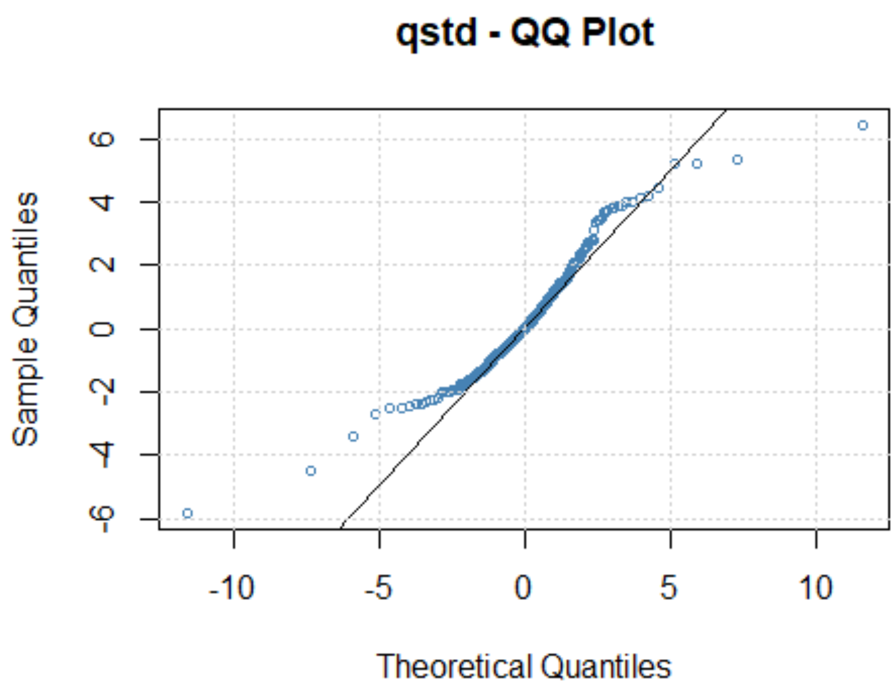
```
##        AIC       BIC       SIC      HQIC
## -3.149861 -3.125176 -3.149897 -3.140867

plot(m2,which=13)
```




```
m3=garchFit(~arma(3,3)+garch(1,1),data=rtn, cond.dist="std", trace=F)

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

summary(m3)

##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~arma(3, 3) + garch(1, 1), data = rtn, cond.dist =
"std",
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(3, 3) + garch(1, 1)
## <environment: 0x0000000012f885a8>
##  [data = rtn]
##
## Conditional Distribution:
```

```
##  std
##
## Coefficient(s):
##          mu          ar1          ar2          ar3          ma1
ma2
## -0.00144369   0.36658011  -0.53295424   0.13321300  -0.50926013
0.59706343
##         ma3        omega       alpha1        beta1        shape
## -0.21833229   0.00017723   0.65836016   0.71120459   2.39898766
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -1.444e-03   5.770e-04   -2.502 0.012344 *
## ar1     3.666e-01   2.102e-01    1.744 0.081227 .
## ar2    -5.330e-01   1.539e-01   -3.463 0.000535 ***
## ar3     1.332e-01   1.537e-01    0.867 0.386127
## ma1    -5.093e-01   2.073e-01   -2.457 0.014009 *
## ma2     5.971e-01   1.470e-01    4.060  4.9e-05 ***
## ma3    -2.183e-01   1.572e-01   -1.389 0.164873
## omega   1.772e-04   6.178e-05    2.869 0.004121 **
## alpha1  6.584e-01   2.013e-01    3.270 0.001075 **
## beta1   7.112e-01   2.906e-02   24.471  < 2e-16 ***
## shape   2.399e+00   1.502e-01   15.974  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  4110.433    normalized:  1.763377
##
## Description:
##  Tue Jun 08 23:16:03 2021 by user: ASUS
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  11608.29  0
##  Shapiro-Wilk Test  R    W      0.8662373 0
##  Ljung-Box Test     R    Q(10)  56.87998  1.403931e-08
##  Ljung-Box Test     R    Q(15)  81.52336  3.673906e-11
##  Ljung-Box Test     R    Q(20)  85.20093  5.054416e-10
##  Ljung-Box Test     R^2  Q(10)  11.26266  0.337424
##  Ljung-Box Test     R^2  Q(15)  11.85952  0.6896297
##  Ljung-Box Test     R^2  Q(20)  14.45463  0.8067175
##  LM Arch Test       R    TR^2   11.74366  0.4664801
##
## Information Criterion Statistics:
```

```
##       AIC       BIC       SIC      HQIC
## -3.517317 -3.490163 -3.517361 -3.507424
plot(m3,which=13)
```



By comparing m2 and m3,

For mean equation, testing linear dependence in standardized residual by Ljung Box Test H0: No linear dependence in standardized residual or Mean equation is appropriate. For both model, p-value < 0.05 for all lag considered. H0 is rejected at 0.05 level of significance for m2 and m3, showing no appropriate mean equation for both models.

For variance equation, testing non-linear dependence in standardized residual by Ljung Box test. H0: there is no ARCH effect. For both model, p-value > 0.05 for all lags considered. H0 is not rejected at 0.05 level of significance. That is, variance equation is appropriate for both models.

For distribution, m3 assuming standardized t-distribution is more appropriate as the Q-Q plot show less deviation from theoretical quantile, and the m2 model JB test is also rejected. H0: series is normally distributed is rejected.

Also, if compared AIC, m3 yield lower value of AIC assuring a more appropriate model to be adopted for forecasting

M3 model:

$$\hat{r}_t = \underset{(0.00058)}{(-1.444e\text{-}03)}^{*}(1 - 0.367 + 0.533 - 0.133) + \underset{(0.21)}{0.37}^{\cdot} r_{t-1} - \underset{(0.21)}{0.533}^{***} r_{t-2} + \underset{(0.154)}{0.133} r_{t-3} - \underset{(0.21)}{0.51}^{*} a_{t-1} + \underset{(0.15)}{0.6}^{***} a_{t-2} - \underset{(0.157)}{0.218} a_{t-3}$$

$$\sigma_t^2 = \underset{(6.2e\text{-}03)}{0.000177}^{***} + \underset{(0.2013)}{1.77e\text{-}04}^{***} a_{t-1}^2 + \underset{(0.03)}{0.7112}^{***} \sigma_{t-1}^2 \qquad t_{2.399}$$

435-exam-3.R

```
rm(list=ls())
library(fBasics)

library(timeDate)
library(timeSeries)
library(fGarch)

library(quantmod)

library(forecast)
library(fUnitRoots)

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

## [1] "NAEXKP01CAQ189S"

GDPca=NAEXKP01CAQ189S[77:202,]
logGDPca=log(GDPca)
head(GDPca)

##            NAEXKP01CAQ189S
## 1980-01-01    209975000000
## 1980-04-01    209908250000
## 1980-07-01    209667250000
## 1980-10-01    212561000000
## 1981-01-01    217245750000
## 1981-04-01    219687500000

tail(GDPca)

##            NAEXKP01CAQ189S
## 2010-01-01    430760250000
## 2010-04-01    433014250000
## 2010-07-01    436083000000
## 2010-10-01    440956250000
## 2011-01-01    444287000000
## 2011-04-01    445152500000

dim(GDPca)

## [1] 126   1

getSymbols("CLVMNACSCAB1GQUK",src="FRED")

## [1] "CLVMNACSCAB1GQUK"
```

```
GDPuk=CLVMNACSCAB1GQUK[21:146,]
logGDPuk=log(GDPuk)
head(GDPuk)
```

```
##            CLVMNACSCAB1GQUK
## 1980-01-01         202901.2
## 1980-04-01         198842.4
## 1980-07-01         198646.1
## 1980-10-01         196470.2
## 1981-01-01         196223.3
## 1981-04-01         196624.9
```

```
tail(GDPuk)
```

```
##            CLVMNACSCAB1GQUK
## 2010-01-01         396944.0
## 2010-04-01         401031.9
## 2010-07-01         403957.8
## 2010-10-01         404093.3
## 2011-01-01         405552.3
## 2011-04-01         405899.7
```

```
dim(GDPuk)
```

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

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

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

```
GDPus=GDPC1[133:258,]
logGDPus=log(GDPus)
head(GDPus)
```

```
##               GDPC1
## 1980-01-01 6837.641
## 1980-04-01 6696.753
## 1980-07-01 6688.794
## 1980-10-01 6813.535
## 1981-01-01 6947.042
## 1981-04-01 6895.559
```

```
tail(GDPus)
```

```
##               GDPC1
## 2010-01-01 15415.15
## 2010-04-01 15557.28
## 2010-07-01 15671.97
## 2010-10-01 15750.62
## 2011-01-01 15712.75
## 2011-04-01 15825.10
```

```
dim(GDPus)

## [1] 126   1

yt=cbind(as.numeric(GDPus),as.numeric(GDPuk),as.numeric(GDPca))
zt=log(yt)
colnames(zt) <- c("logGDPUS","logGDPUK","logGDPCA")


#Question1
require(MTS)

VARorder(zt)

## selected order: aic =  2
## selected order: bic =  2
## selected order: hq =  2

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

varfit1 <- VAR(zt,p=2)
summary(varfit1)

##
## VAR Estimation Results:
## =========================
## Endogenous variables: logGDPUS, logGDPUK, logGDPCA
## 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 logGDPUS:
## =========================================
## logGDPUS = logGDPUS.l1 + logGDPUK.l1 + logGDPCA.l1 + logGDPUS.l2 +
logGDPUK.l2 + logGDPCA.l2 + const
##
##              Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  1.04262    0.09639  10.817  < 2e-16 ***
## logGDPUK.l1  0.36850    0.08432   4.370  2.7e-05 ***
## logGDPCA.l1  0.17407    0.08580   2.029  0.04474 *
## logGDPUS.l2 -0.06237    0.09723  -0.641  0.52250
## logGDPUK.l2 -0.26915    0.08399  -3.204  0.00174 **
## logGDPCA.l2 -0.25543    0.08741  -2.922  0.00417 **
## 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
##
##
## Estimation results for equation logGDPUK:
## =========================================
## logGDPUK = logGDPUS.l1 + logGDPUK.l1 + logGDPCA.l1 + logGDPUS.l2 +
logGDPUK.l2 + logGDPCA.l2 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.11577    0.09334   1.240    0.217
## logGDPUK.l1  1.40140    0.08166  17.161  < 2e-16 ***
## logGDPCA.l1  0.03900    0.08309   0.469    0.640
## logGDPUS.l2 -0.04250    0.09416  -0.451    0.653
## logGDPUK.l2 -0.49628    0.08134  -6.101  1.4e-08 ***
## logGDPCA.l2 -0.03246    0.08465  -0.383    0.702
## 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 logGDPCA:
## =========================================
## logGDPCA = logGDPUS.l1 + logGDPUK.l1 + logGDPCA.l1 + logGDPUS.l2 +
logGDPUK.l2 + logGDPCA.l2 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.35327    0.09638   3.665 0.000373 ***
## logGDPUK.l1  0.28589    0.08432   3.391 0.000952 ***
## logGDPCA.l1  1.20185    0.08579  14.009  < 2e-16 ***
## logGDPUS.l2 -0.33122    0.09723  -3.407 0.000902 ***
## logGDPUK.l2 -0.26496    0.08399  -3.155 0.002042 **
## logGDPCA.l2 -0.24763    0.08741  -2.833 0.005432 **
## 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
##
```

```
##
##
## Covariance matrix of residuals:
##           logGDPUS  logGDPUK  logGDPCA
## logGDPUS 3.411e-05 5.799e-06 1.441e-05
## logGDPUK 5.799e-06 3.199e-05 1.229e-06
## logGDPCA 1.441e-05 1.229e-06 3.411e-05
##
## Correlation matrix of residuals:
##          logGDPUS logGDPUK logGDPCA
## logGDPUS   1.0000   0.1756   0.4225
## logGDPUK   0.1756   1.0000   0.0372
## logGDPCA   0.4225   0.0372   1.0000
```

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

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: logGDPUS, logGDPUK, logGDPCA
## 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 logGDPUS:
## =========================================
## logGDPUS = logGDPUS.l1 + logGDPUK.l1 + logGDPCA.l1 + logGDPUK.l2 +
logGDPCA.l2 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.98379    0.02958  33.261  < 2e-16 ***
## logGDPUK.l1  0.38363    0.08076   4.750 5.77e-06 ***
## logGDPCA.l1  0.19941    0.07597   2.625 0.009817 **
## logGDPUK.l2 -0.28365    0.08069  -3.515 0.000624 ***
## logGDPCA.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
##
```

```
##
## Estimation results for equation logGDPUK:
## =========================================
## logGDPUK = logGDPUS.l1 + logGDPUK.l1 + logGDPUK.l2 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.08187    0.02684   3.050  0.00282 **
## logGDPUK.l1  1.42694    0.07494  19.040  < 2e-16 ***
## logGDPUK.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 logGDPCA:
## =========================================
## logGDPCA = logGDPUS.l1 + logGDPUK.l1 + logGDPCA.l1 + logGDPUS.l2 +
logGDPUK.l2 + logGDPCA.l2
##
##             Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.37334    0.09622   3.880 0.000173 ***
## logGDPUK.l1  0.29148    0.08482   3.437 0.000814 ***
## logGDPCA.l1  1.22221    0.08543  14.306  < 2e-16 ***
## logGDPUS.l2 -0.37887    0.09326  -4.062 8.78e-05 ***
## logGDPUK.l2 -0.28088    0.08397  -3.345 0.001104 **
## logGDPCA.l2 -0.22529    0.08689  -2.593 0.010722 *
## ---
## 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
##
##
##
## Covariance matrix of residuals:
##           logGDPUS  logGDPUK  logGDPCA
## logGDPUS 3.423e-05 5.881e-06 1.432e-05
## logGDPUK 5.881e-06 3.221e-05 1.088e-06
## logGDPCA 1.432e-05 1.088e-06 3.487e-05
##
## Correlation matrix of residuals:
##          logGDPUS logGDPUK logGDPCA
## logGDPUS   1.0000  0.17712  0.41450
```

```
## logGDPUK   0.1771  1.00000  0.03246
## logGDPCA   0.4145  0.03246  1.00000
```

Q1.1

Final fitted model with threshold t=1.645:

Estimation results for equation logGDPUS:

```
## =========================================
## logGDPUS = 0.98379logGDPUS.l1 + 0.38363logGDPUK.l1 + 0.19941logGDPCA.l1 - 0.28365logGDPUK.l2 - 0.28569logGDPCA.l2 + 1.17762
##
##              Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.98379    0.02958  33.261  < 2e-16 ***
## logGDPUK.l1  0.38363    0.08076   4.750 5.77e-06 ***
## logGDPCA.l1  0.19941    0.07597   2.625 0.009817 **
## logGDPUK.l2 -0.28365    0.08069  -3.515 0.000624 ***
## logGDPCA.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
```

Estimation results for equation logGDPUK:

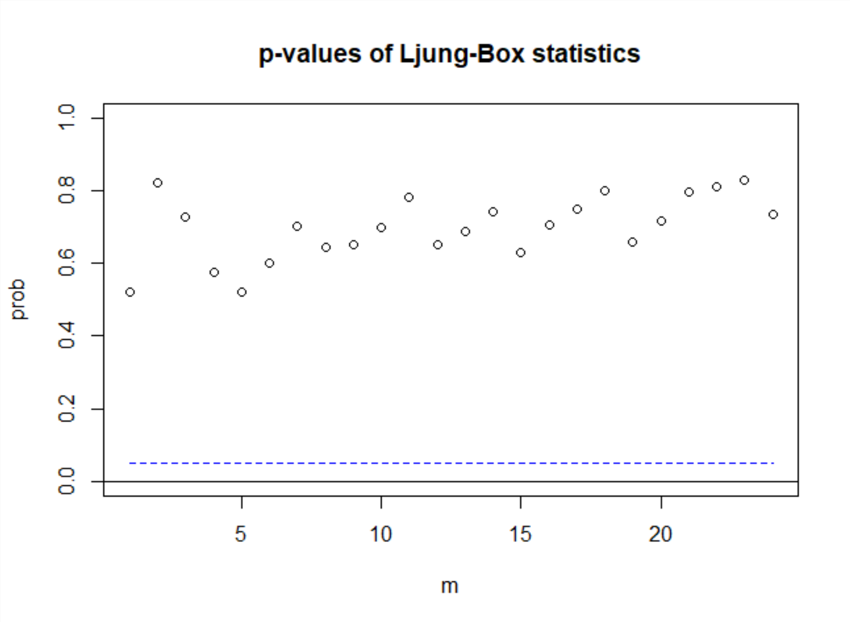
```
## =========================================
## logGDPUK = 0.08187logGDPUS.l1 + 1.42694logGDPUK.l1 - 0.52537logGDPUK.l2 + 0.48200
##
##              Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.08187    0.02684   3.050  0.00282 **
## logGDPUK.l1  1.42694    0.07494  19.040  < 2e-16 ***
## logGDPUK.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
```

Estimation results for equation logGDPCA:

```
## =========================================
## logGDPCA = 0.37334logGDPUS.l1 + 0.29148logGDPUK.l1 + 1.22221logGDPCA.l1 - 0.37887logGDPUS.l2 - 0.28088logGDPUK.l2 - 0.22529logGDPCA.l2
##
##              Estimate Std. Error t value Pr(>|t|)
## logGDPUS.l1  0.37334    0.09622   3.880 0.000173 ***
## logGDPUK.l1  0.29148    0.08482   3.437 0.000814 ***
## logGDPCA.l1  1.22221    0.08543  14.306  < 2e-16 ***
## logGDPUS.l2 -0.37887    0.09326  -4.062 8.78e-05 ***
## logGDPUK.l2 -0.28088    0.08397  -3.345 0.001104 **
## logGDPCA.l2 -0.22529    0.08689  -2.593 0.010722 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Model checking

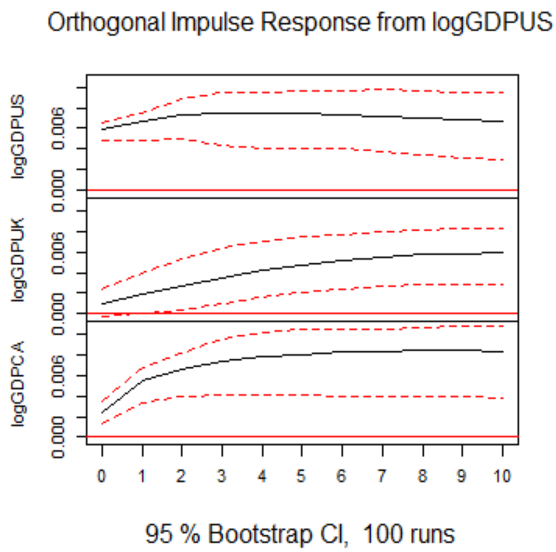
```
detach("package:vars", unload = TRUE)

m1 <- VAR(zt, p=2)

m2 <- refVAR(m1, thres=1.645)

MTSdiag(m2)
```



By testing the autocorrelation/CCM of residual, the Ljung box test of residual vestor: H0: the model is adequate. The p-value > 0.05 for all lags. H0 is not rejected at 0.05 level of significance. That is, model is adequate with 95% CI.

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

require(vars)
```

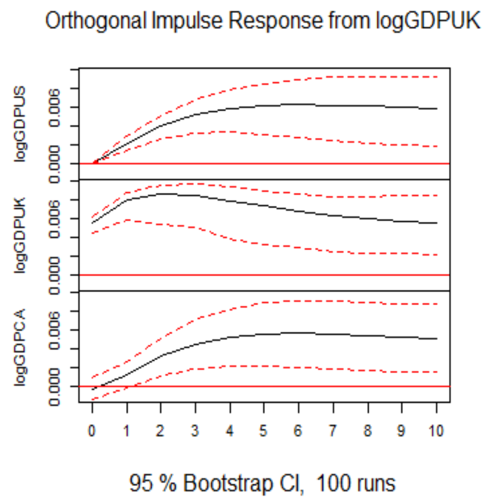
```
#Question 2
impresp <- irf(varfit2)
plot(impresp)
```



Impact of logGDPUS shock on:

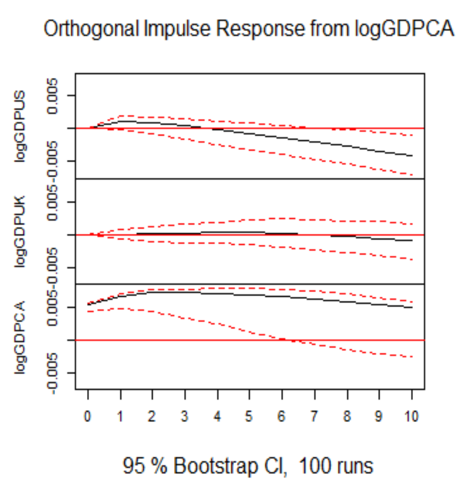
- logGDPUS: there is a significantly strong positive impact form shock of itself, especially from concurrent period. The positive effect prevails for a certain number of lags and the magnitude start to die down as lags increase.

- logGDPUK and logGDPCA: there is a significantly weak positive impact form shock of us gdp to both uk and Canada. The positive effect prevails since concurrent period (small size) and becomes increasingly stronger for a certain number of lags. Finally, the magnitude starts to die down as lags increase. This show that the shock in GDP even in large country like US takes time to transmit across countries.



Impact of logGDPUK shock on:

- logGDPUK: there is a <u>significantly strong positive</u> impact form shock of itself, especially from concurrent period. The positive effect prevails for a certain number of lags and the magnitude start to die down as lags increase.
- logGDPUS and logGDPCA: there is a <u>significantly weak positive</u> impact from shock of UK gdp to both US and Canada, but no concurrent effect. The positive effect becomes increasingly stronger for a certain number of lags. Finally, the magnitude starts to die down as lags increase.



Impact of logGDPCA shock:

Shock of Canada GDP has no impact that significantly different from zero, not being able to transmit to other countries' GDP (US and UK).

```r
#Question3
fevd(varfit2, n.ahead=6)
```

```
## $logGDPUS
##        logGDPUS   logGDPUK    logGDPCA
## [1,] 1.0085470 0.00000000 0.000000000
## [2,] 0.9432503 0.05288901 0.013823861
## [3,] 0.8720533 0.12743265 0.012061113
## [4,] 0.8095558 0.19473980 0.008569452
## [5,] 0.7622339 0.24530303 0.006278794
## [6,] 0.7273564 0.28082071 0.006287943
```

The source of deviation from actual future value of predicted future logGDP of US:

For first step ahead forecast: deviation comes solely from shock of US logGDP itself (~100%).

For next and further step of forecasting: source of deviation comes from all three countries' logGDP shock with UK and Canada become increasingly influential (UK effect is significantly stronger than that of Canada), while effect shock of US itself starts to decay.

Canada GDP has lowest contribution to deviation of US GDP.

```
## $logGDPUK
##         logGDPUS  logGDPUK     logGDPCA
## [1,] 0.03217530 0.9934657 0.000000e+00
## [2,] 0.04819675 0.9772840 0.000000e+00
## [3,] 0.06911440 0.9560830 4.349064e-05
## [4,] 0.09524004 0.9294937 1.739924e-04
## [5,] 0.12526061 0.8988964 3.389147e-04
## [6,] 0.15735132 0.8662398 4.427407e-04
```

The source of deviation from actual future value of predicted future logGDP of UK:

For first step ahead forecast: source of deviation comes from both shock of US and UK logGDP, with majority of deviation rooting from UK shock (~99%).

For next and further step of forecasting: source of deviation comes from all three countries' logGDP shock with US and Canada become increasingly influential (US effect is significantly stronger than that of Canada), while effect shock of UK itself starts to decay.

Canada GDP has lowest contribution to deviation of UK GDP.

```
## $logGDPCA
##       logGDPUS    logGDPUK  logGDPCA
## [1,] 0.1732748 0.001746268 0.8335260
## [2,] 0.3305722 0.016791287 0.6617613
## [3,] 0.3751111 0.056751020 0.5781952
## [4,] 0.3967958 0.096372443 0.5177318
## [5,] 0.4101177 0.127438451 0.4739825
## [6,] 0.4214840 0.149311083 0.4411891
```

The source of deviation from actual future value of predicted future logGDP of Canada:

For first-step ahead forecast: source of deviation comes from both shock of US and Canada logGDP, with majority of deviation rooting from Canada shock itself (~83%), yet US gdp shock still contribute a significant effect to deviation (~17%)

For next and further step of forecasting: source of deviation comes from all three countries' logGDP shock with US and UK become increasingly influential (US effect is significantly stronger than that of UK), while effect shock of Canada itself starts to decay.

UK GDP has lowest contribution to deviation of UK GDP.

```
#Question 4
m0 <- adfTest(logGDPus, lags=3, type=c("ct"), title=NULL)

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

m0@test$p.value

##
## 0.804053

m01 <- adfTest(diff(logGDPus), lags=3, type=c("ct"), title=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(logGDPus), lags = 3, type = c("ct"), title =
NULL): p-
## value smaller than printed p-value

m01@test$p.value

##
## 0.01
```

```
m1 <- adfTest(logGDPuk, lags=3, type=c("ct"), title=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

##
## 0.7535667

m11 <- adfTest(diff(logGDPuk), lags=3, type=c("ct"), title=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(logGDPuk), lags = 3, type = c("ct"), title =
NULL): p-
## value smaller than printed p-value

m11@test$p.value

##
## 0.01

m2 <- adfTest(logGDPca, lags=3, type=c("ct"), title=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

m21 <- adfTest(diff(logGDPca), lags=3, type=c("ct"), title=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(logGDPca), lags = 3, type = c("ct"), title =
NULL): p-
## value smaller than printed p-value

m21@test$p.value

##
## 0.01

fit <- lm(logGDPca~logGDPuk+logGDPus)
summary(fit)
```

```
##
## Call:
## lm(formula = logGDPca ~ logGDPuk + logGDPus)
##
## 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 ***
## logGDPuk      0.6720     0.1138   5.906 3.19e-08 ***
## logGDPus      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
```

```
error=residuals(fit)
m3 <- adfTest(error, lags=3, type=c("nc"), title=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
```

```
##
## 0.02300044
```

Ans

By using concept of long-run relationship, the spurious effect from non-stationary variable can be prevented. According to Engle and Granger, the concept of cointegration is used to propose a long-run relationship/equilibrium.

By checking the integration order of logGDP series of US, UK and Canada, the augmented dickey fuller with time trend and lag=3 show that all three logGDP series possess an integration order of 1: I(1), satisfying initial condition of same order.

After running linear regression model with logGDP of Canada as dependent variable, under hypothesis that it is the country with smallest economic power requiring more reliance (in terms of export and import control) upon other two countries, and logGDP of US and UK as independent variables; the residual of this model is tested by ADF and portray p-value < 0.05 rejecting an H0: residual series is non-stationary. That is, the residual is a series with integration order of 0: I(0).

This findings therefore satisfy long-run equilibrium requirement of Engle and Granger such that vector of zt is cointegrated of order 1,1: CI(1,1), proving the existence of long-run relationship and equilibrium

Hence, from linear regression model: lm(logGDPca ~ logGDPuk + logGDPus)

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  15.3400     0.5305  28.919  < 2e-16 ***
## logGDPuk      0.6720     0.1138   5.906 3.19e-08 ***
## logGDPus      0.2855     0.0978   2.919  0.00417 **
```

logGDPca - 15.3400 - 0.6720logGDPuk - 0.2855 logGDPus = 0 in Long-run equilibrium

$\beta^T X = \begin{bmatrix} 1 \\ -15.34 \\ -0.672 \\ -2.855 \end{bmatrix} [\text{logGDPca} \quad 1 \quad \text{logGDPuk} \quad \text{logGDPus}]$ with β as cointegrating vector.

```
#Question 5
diff.logGDPus=diff(logGDPus)
diff.logGDPuk=diff(logGDPuk)
diff.logGDPca=diff(logGDPca)
diff.logGDPus.L.1=Lag(diff(logGDPus),k=1)
diff.logGDPuk.L.1=Lag(diff(logGDPuk),k=1)
diff.logGDPca.L.1=Lag(diff(logGDPca),k=1)
error.L.1=Lag(error,k=1)


fit1 <-
lm(formula=diff.logGDPca~diff.logGDPus.L.1+diff.logGDPuk.L.1+diff.logGDPca.L.
1+error.L.1)
summary(fit1)

##
## Call:
## lm(formula = diff.logGDPca ~ diff.logGDPus.L.1 + diff.logGDPuk.L.1 +
##     diff.logGDPca.L.1 + error.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.logGDPus.L.1  0.3309112  0.0947854   3.491 0.000676 ***
## diff.logGDPuk.L.1  0.2601149  0.0822372   3.163 0.001982 **
```

```
## diff.logGDPca.L.1  0.2540305  0.0852632   2.979 0.003502 **
## error.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
```

```
fit2 <-
lm(formula=diff.logGDPuk~diff.logGDPus.L.1+diff.logGDPuk.L.1+diff.logGDPca.L.
1+error.L.1)
summary(fit2)
```

```
##
## Call:
## lm(formula = diff.logGDPuk ~ diff.logGDPus.L.1 + diff.logGDPuk.L.1 +
##     diff.logGDPca.L.1 + error.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0169337 -0.0030371 -0.0001463  0.0028363  0.0132674
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       0.0022056  0.0007509   2.937  0.00398 **
## diff.logGDPus.L.1 0.0923161  0.0953595   0.968  0.33497
## diff.logGDPuk.L.1 0.4662411  0.0827353   5.635 1.19e-07 ***
## diff.logGDPca.L.1 0.0543980  0.0857796   0.634  0.52719
## error.L.1         0.0092680  0.0280046   0.331  0.74127
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.00583 on 119 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.3236, Adjusted R-squared:  0.3008
## F-statistic: 14.23 on 4 and 119 DF,  p-value: 1.603e-09
```

```
fit3 <-
lm(formula=diff.logGDPus~diff.logGDPus.L.1+diff.logGDPuk.L.1+diff.logGDPca.L.
1+error.L.1)
summary(fit3)
```

```
##
## Call:
## lm(formula = diff.logGDPus ~ diff.logGDPus.L.1 + diff.logGDPuk.L.1 +
##     diff.logGDPca.L.1 + error.L.1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
```

```
## -0.0200562 -0.0031445  0.0001442  0.0035688  0.0171946
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.0034297  0.0007692   4.459 1.88e-05 ***
## diff.logGDPus.L.1   0.0787279  0.0976765   0.806 0.421847
## diff.logGDPuk.L.1   0.2952720  0.0847456   3.484 0.000692 ***
## diff.logGDPca.L.1   0.2158193  0.0878639   2.456 0.015482 *
## error.L.1          -0.0755575  0.0286851  -2.634 0.009559 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.005972 on 119 degrees of freedom
##   (2 observations deleted due to missingness)
## Multiple R-squared:  0.3415, Adjusted R-squared:  0.3194
## F-statistic: 15.43 on 4 and 119 DF,  p-value: 3.406e-10
```

$$logGDPca_{t-1} - 15.3400 - 0.6720\,logGDPuk_{t-1} - 0.2855\,logGDPus_{t-1} = \varepsilon_{CA_{t-1}}$$

Error Correction Model:

fit ①

$$\Delta \log GDP_{canada_t} = \underset{(0.0007)}{0.0008} + \underset{(0.095)}{0.331}^{***}\,\Delta \log GDP_{us_{t-1}} + \underset{(0.08)}{0.26}^{**}\,\Delta \log GDP_{uk_{t-1}} + \underset{(0.085)}{0.254}^{**}\,\Delta \log GDP_{canada_{t-1}} - \underset{(0.028)}{0.046}^{.}\,\varepsilon_{CA_{t-1}} + \varepsilon_{CA_t}$$

US term: go in same direction: increase in US gdp trend ↳ more import from canada

UK term: go in same direction: increase in UK gdp trend ↳ more import from canada

Canada term: increasing trend of its GDP

Error term: if $\varepsilon_{CA_{t-1}} > 0$ then canada GDP was too high in $t-1$ or US & UK GDP is too low in $t-1$:

Negative coefficient → reduction in current canada GDP

Possibly from current account adjustment of US & UK by lowering import to improve trade balance & increase their GDP result in lower export & GDP of Canada at t

fit 2:

$$\Delta \log GDP_{uk_t} = \underset{(0.0007)}{0.0022^{**}} + \underset{(0.467)}{0.092}\,\Delta \log GDP_{us_{t-1}} + \underset{(0.08)}{0.47^{***}}\,\Delta \log GDP_{uk_{t-1}} + \underset{(0.085)}{0.054}\,\Delta \log GDP_{canada_{t-1}} + \underset{(0.02)}{0.009}\,\varepsilon_{CA_{t-1}} + \varepsilon_{CA_t}$$

- 0.092 → insignificant
- 0.47 → only significant: GDP trend (+)
- 0.054 → insignificant
- 0.009 → insignificant: still positive → if $\varepsilon > 0$ adjustment to reduce Canada GDP by reducing import which help improve its GDP

fit 3:

$$\Delta \log GDP_{us_t} = \underset{(0.0007)}{0.0034^{***}} + \underset{(0.098)}{0.079}\,\Delta \log GDP_{us_{t-1}} + \underset{(0.08)}{0.3^{***}}\,\Delta \log GDP_{uk_{t-1}} + \underset{(0.088)}{0.216^{*}}\,\Delta \log GDP_{canada_{t-1}} - \underset{(0.03)}{0.08^{**}}\,\varepsilon_{CA_{t-1}} + \varepsilon_{CA_t}$$

- 0.079 → insig.
- 0.3 → (+) GDP trend world economy
- 0.216 → (+) GDP trend world economy
- 0.08 → negative coefficient; if $\varepsilon > 0$, US gdp ↓