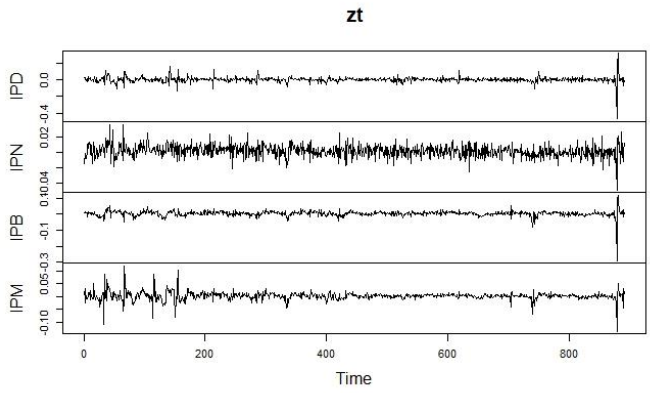
Take Home Exam

Question 1

Question 1.1 Obtain time plot of the zt series



The above figure shows the growth rate series of four industrial production indices which consist of durable goods (IPD), non-durable goods (IPN), business equivalent (IPB) and material (IPM). It can be seen from the figure that all the growth rate data were fluctuating around its constant mean and the data seem to become more volatile in the recent period.

Question 1.2 Build VAR model

VAR model with 2 lags:

IPD_t = 0.0017 + 0.1187IPD_{t-1}-0.1572IPD_{t-2}+0.2192IPN_{t-1}+0.1887IPN_{t-2}
(0.0011) (0.0473) (0.0471) (0.1467) (0.1461)
-0.1072IPB_{t-1}-0.1977IPB_{t-2}+0.3384IPM_{t-1}+0.1053IPM_{t-2}
(0.0928) (0.0924) (0.0821) (0.0825)

IPN_t = 0.0018 + 0.0295IPD_{t-1}-0.00042IPD_{t-2}-0.1678IPN_{t-1}-0.0449IPN_{t-2}
(0.00028) (0.0117) (0.0116) (0.0363) (0.0361)
+0.0092IPB_{t-1}-0.00132IPB_{t-2}+0.0162IPM_{t-1}+0.00082IPM_{t-2}
(0.0.02295) (0.0204) (0.03) (0.0204)

IPB_t = 0.0018 + 0.0034IPD_{t-1}-0.1422IPD_{t-2}+0.0539IPN_{t-1}+0.025IPN_{t-2}
(0.00058) (0.0245) (0.0243) (0.0759) (0.0756)
+0.1222IPB_{t-1}+0.1828IPB_{t-2}+0.1800IPM_{t-1}+0.1031IPM_{t-2}
(0.0480) (0.0478) (0.0425) (0.0427)

IPM_t= 0.0013+ 0.0171IPD_{t-1}+0.0106IPD_{t-2}+0.1435IPN_{t-1}+0.1338IPN_{t-2}
(0.00054) (0.0227) (0.0226) (0.0703) (0.0700)
+0.0277IPB_{t-1}+0.0212IPB_{t-2}+0.2112IPM_{t-1}-0.0601IPM_{t-2}
(0.0445) (0.0443) (0.0394) (0.0395)

Refined VAR model:

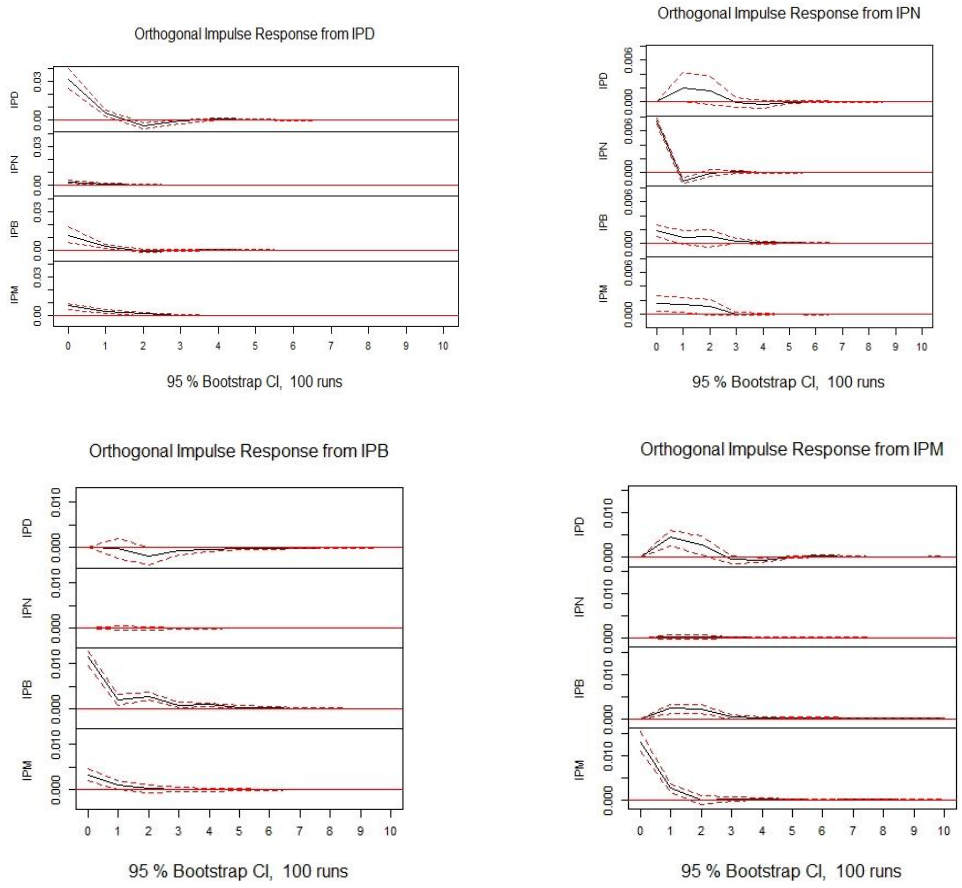
IPD_t = 0.1187IPD_{t-1}-0.1572IPD_{t-2}-0.1977IPB_{t-2}+0.3384IPM_{t-1}
(0.0473) (0.0471) (0.0924) (0.0821)

IPN_t = 0.0018 + 0.0295IPD_{t-1}-0.1678IPN_{t-1}
(0.00028) (0.0117) (0.0363)

IPB_t = 0.0018 -0.1422IPD_{t-2}+0.1222IPB_{t-1}+0.1828IPB_{t-2}+0.1800IPM_{t-1}+0.1031IPM_{t-2}
(0.00058) (0.0243) (0.0480) (0.0478) (0.0425) (0.0427)

IPM_t= 0.0013+0.1435IPN_{t-1}+0.2112IPM_{t-1}
(0.00054) (0.0703) (0.0394)

Question 1.3 Impulse response function



From the figures, it shows that IPN or non-durable goods index does not get much impact from the shocks from others. The shocks from IPD create positive impact on IPD, IPB, and IPM. The shocks from IPN also creates positive impact on IPB and IPM while for IPD, there is no concurrent impact, but the positive impact will show in the next period. The shocks from IPB will affect IPD around 2 periods after the shock occurs but it is immediately affecting IPB and IPM positively. Lastly, the shock from IPM will also affect IPD and IPB positively in the next period after the shock occurs.

Question 1.4 Forecast error variance decomposition

According to forecast error variance decomposition, it can be seen that the model assumes that the IPD will not be affected by other variables in the first period. Plus, in other period, IPD does not get much effect from other three industrial production indices. For IPN, in the first period, it is consisting of 88.69 percent effect from its own shock and 11.3 percent from IPD. The period after IPB and IPM become more influential to IPN but the number is

very low. For IPB, in the first period, it is affected by 50.69 percent from IPD, 1.2 percent from IPN and 48 percent from its own which means that the effect from change in IPD is more influential than IPB itself. Lastly, for IPM, the variance is consisting of 70.66 percent of IPM itself, 23.69 percent of IPD, 1.14 percent of IPN and 4.5 percent of IPB in the first period but the numbers do not change much in the other period. So, it can be seen that the shock from IPD has large impact on IPM.

Question 1.5 Forecasting (95 percent interval)

IPD

Period 1: (-0.063,0.0618)
Period 2: (-0.0562, 0.0718)
Period 3: (-0.0632, 0.0669)
Period 4: (-0.0646, 0.0656)
Period 5: (-0.0633, 0.06699)
Period 6: (-0.0627, 0.0676)

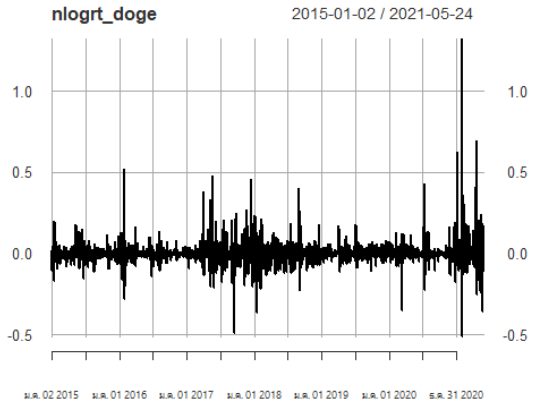
IPN
Period 1: (-0.016, 0.01488)
Period 2: (-0.01415, 0.0172)
Period 3: (-0.0138, 0.0176)
Period 4: (-0.01413, 0.01725)
Period 5: (-0.01414, 0.01723)
Period 6: (-0.01408, 0.01729)

IPB
Period 1: (-0.0232, 0.0413)
Period 2: (-0.0262, 0.0406)
Period 3: (-0.0288, 0.0396)
Period 4: (-0.0307, 0.0378)
Period 5: (-0.0306, 0.0381)
Period 6: (-0.0308, 0.038)

IPM
Period 1: (-0.026, 0.0338)
Period 2: (-0.0281, 0.0341)
Period 3: (-0.0289, 0.0336)
Period 4: (-0.0288, 0.0338)
Period 5: (-0.0289, 0.0337)
Period 6: (-0.029, 0.03355)

Question 2 Dogecoin

After obtaining the data of Dogecoin price from Yahoo finance, it can be seen that the price series of Dogecoin is not weakly stationarity since the price is increasing sharply in the recent period. So, I transformed the price data to log return of the series denoted as nlogrt_doge. The log return series are weakly stationarity according to the figure. You can see that the return was fluctuating around the constant mean.



The mean of the series is not equal to zero. The t-test of nlogrt_doge shows that the mean of the series is not equal to zero since we need to reject H0: True mean is equal to zero (p-value is less than 0.05). In addition, there is serial correlation in the log return series. The result of Ljung-box test shows that we need to reject H0: There is no serial correlation in the data since p-value falls into rejection region. Hence, we need to propose ARMA model to predict the mean of the log return.

According to the Ljung-box test of nlogrt_doge^2 which is the squared of log return series, it shows that there is ARCH effect in the log return of Dogecoin since we need to reject H0: There is no ARCH effect (p-value is less than 0.05). Therefore, we need to use ARCH and GARCH model to explain the variance of the series instead of using the constant term.

Consequently, I have proposed several models of ARMA and GARCH model and then select the most suitable one. I have checked the adequacy of the model by looking at the calculated p-value of standardized residuals tests as shown in the summary of the model.

Model	AIC	BIC	Adequacy
M2(ARMA(1,1)-GARCH(1,1))	-3.150958	-3.136147	Inadequate
M3(GARCH(1,1))	-3.148906	-3.139032	Inadequate
M4(GARCH(3,3))	-3.167287	-3.147539	Inadequate
M5(ARMA(6,6)-GARCH(3,3))	-3.175755	-3.126386	Adequate for lag 10 and 20

Therefore, I will select the model m5 to represent the mean and variance equation of the log return of Dogecoin since the model is the most suitable one out of the 4 models. It has the lowest number of AIC and the model is adequate for 10 lags and 20 lags.

Fitted model

Mean equation: $R_t = -0.3178R_{t-1}+0.2924\ R_{t-2}+0.3164\ R_{t-3}+0.1999\ R_{t-4}-0.7937\ R_{t-5}-0.3155\ R_{t-6}$
(0.0012) (0.2262) (0.0577) (0.0504) (0.0489) (0.1782)
$+0.2455a_{t-1}-0.2446a_{t-2}-0.2605a_{t-3}-0.1763a_{t-4}+0.7807a_{t-5}+0.2413a_{t-6}$
(0.2384) (0.0602) (0.0726) (0.0507) (0.0531) (0.1888)

Variance equation: $\sigma_t^2= 0.0002+0.453\ a_{t-1}^2+(1.000e-08)\ a_{t-2}^2+(1.000e-08)\ a_{t-3}^2+0.2666\ \sigma_{t-1}^2$
(2.452e-05) (0.043) (NA) (0.04652) (NA)
$+0.05267\ \sigma_{t-2}^2+0.2710\ \sigma_{t-3}^2$
(NA) (0.06005)

Question 3

Question 3.1

Suppose uk_gdp is the GDP of UK, can_gdp is the GDP of Canada and us_gdp is the GDP of the US.

Final fitted model

$UK\_gdp_t = 0.0023 + 0.4614\ UK\_gdp_{t-1}$
$\quad\quad (0.0007) \quad (0.0811)$

$Can\_gdp_t = 0.2846\ UK\_gdp_{t-1} + 0.2238\ Can\_gdp_{t-1} + 0.3732\ US\_gdp_{t-1}$
$\quad\quad (0.0815) \quad\quad (0.0839) \quad\quad (0.092)$

$US\_gdp_t = 0.00306 + 0.335\ UK\_gdp_t$
$\quad\quad (0.0008) \quad (0.0854)$

Question 3.2 Impulse response function

According to impulse response function plot, it can be seen that the shocks from uk_gdp is positively affected can_gdp and us_gdp in the first period. The shocks from can_gdp and us_gdp barely affect the GDP of the UK. Canada's GDP is positively affecting Canadian GDP and US GDP significantly. For the shocks from the US, it will start to affect Canada's GDP in the first period.

Question 3.3 Forecast error variance decomposition

According to the forecast error variance decomposition from the R program, it can be seen that the model assumes that uk_gdp is not affected by the shocks from other variables in the first period and the UK GDP is barely affected by other countries. For Canada's GDP, it can be seen that majority of the variance consist of the shock from Canada's GDP itself. For the US GDP, the variance consists of 80 percent from the shock of itself, 17 percent from the shock of Canada's GDP and 0.2 percent from the shock of UK in the first period. While as the time passed by, the shock from the UK and Canada become more influential to the US GDP.

Question 3.4

$\beta_1 uk\_gdp + \beta_2 can\_gdp + \beta_3\ us\_gdp = e_t$

All of the three data series of GDP are I(1) as the calculated p-value of the ADF Test of uk2, can2 and us2 (diff of the original data) is less than 0.05. However, error of fit model is not I(0) using ADF test. So, the data is not cointegrated.

Question 3.5 VECM

$\Delta uk\_gdp = 0.0499 + 0.6088\ \Delta uk\_gdp_{t-1} - (1.046e\text{-}07)\ \Delta can\_gdp_{t-1}$
$\quad\quad (00198) \quad (8.202e\text{-}02) \quad\quad (7.722e\text{-}08)$
$+5.312\ \Delta us\_gdp_{t-1} - 0.1169\ error_{t-1}$
$\quad\quad (2.429e+00) \quad\quad (3.706e\text{-}02)$

The difference of UK GDP and US GDP in the previous period has positive impact on the UK GDP now while Canada's GDP in the previous period has negative impact on the UK GDP. For the error term from the previous period, it is also negatively affecting the UK GDP now.

$$\Delta can\_gdp = \underset{(2.105e+08)}{(1.944e+08)} + \underset{(8.717e+04)}{(3.712e+05)}\,\Delta uk\_gdp_{t-1} - \underset{(8.207e-02)}{(1.046e-07)}\,\Delta can\_gdp_{t-1} + \underset{(2.582e+06)}{(1.044e+07)}\,\Delta us\_gdp_{t-1} - \underset{(3.938e+04)}{(3.009e+04)}\,error_{t-1}$$

$$\Delta us\_gdp = \underset{(8.006e+00)}{0.0332} + \underset{(3.316e-03)}{(1.691e-02)}\,\Delta uk\_gdp_{t-1} + \underset{(3.122e-09)}{(2.375e-09)}\,\Delta can\_gdp_{t-1} + \underset{(9.820e-02)}{(1.151e-01)}\,\Delta us\_gdp_{t-1} + \underset{(1.498e-03)}{(1.044e-03)}\,error_{t-1}$$

Take Home Exam-6104640138.R

Jidapa

2021-06-10

```
setwd("D:/uni year 3/EE435")
cat(rep("\n", 50)) #clear R console

#install.packages("quantmod")
#install.packages("fBasics")
#install.packages("sn")
#install.packages("PerformanceAnalytics")
#install.packages("car")
#install.packages("tseries")
#install.packages("forecast")
#install.packages("fGarch")
#install.packages("vars")
library(fGarch)

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

## Loading required package: timeDate

## Loading required package: timeSeries

## Loading required package: fBasics

library(quantmod)

## Loading required package: xts

## Loading required package: zoo

##
## Attaching package: 'zoo'

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

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

## Loading required package: TTR

##
## Attaching package: 'TTR'

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

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

library(fBasics)
library(sn)

## Loading required package: stats4

##
## Attaching package: 'sn'

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

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

library(PerformanceAnalytics)

##
## Attaching package: 'PerformanceAnalytics'

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

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

library(car)

## Loading required package: carData

##
## Attaching package: 'car'

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

library(tseries)
library(forecast)
require(quantmod)
require(vars)

## Loading required package: vars

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

## Loading required package: MASS
```

```
## Loading required package: strucchange

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

## Loading required package: sandwich

## Loading required package: urca

## Loading required package: lmtest
```

```
##Question 1
#Load data from FRED
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"
```

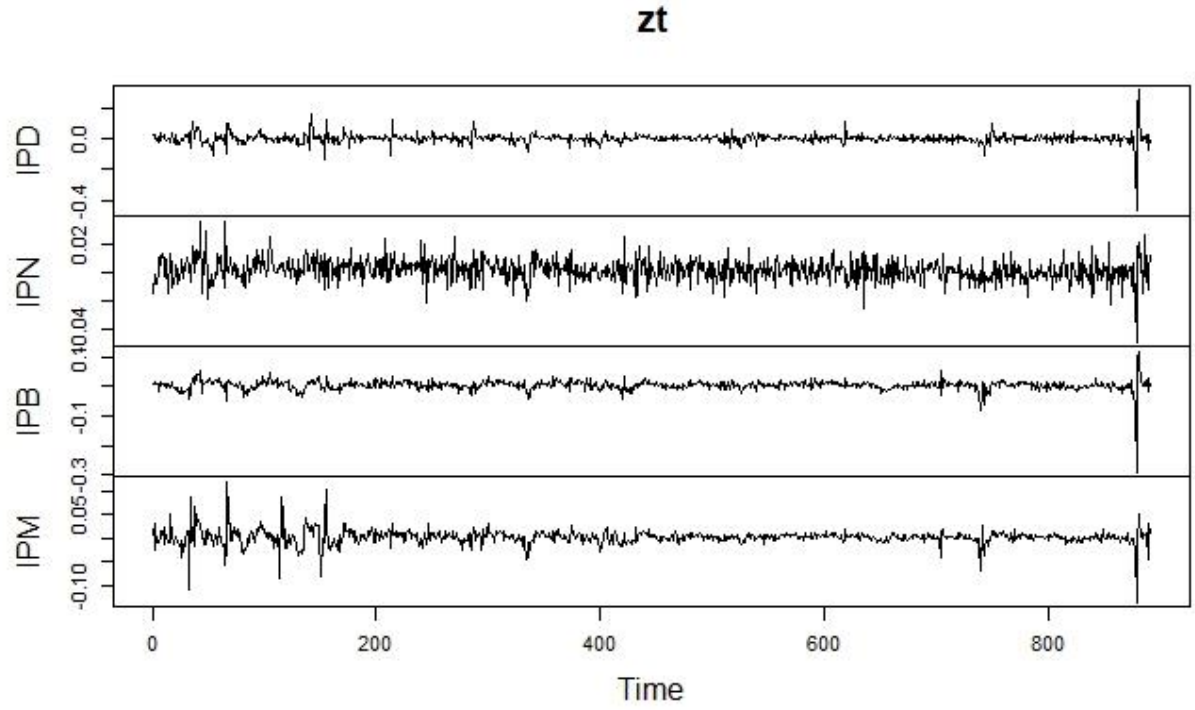
```
dim(IPMAT)
```

```
## [1] 988   1
```

```
#Combine data
IP = cbind(as.numeric(IPDCONGD), as.numeric(IPNCONGD), as.numeric(IPBUSEQ), a
s.numeric(IPMAT[-c(1:96)]))
dim(IP)
```

```
## [1] 892   4
```

```
colnames(IP) <- c("IPD", "IPN", "IPB", "IPM")
zt = diff(log(IP))
#1.1 Plot time series data
plot.ts(zt)
```




```
#1.2 VAR model
m1 = VAR(zt, p = 2)
summary(m1)
```

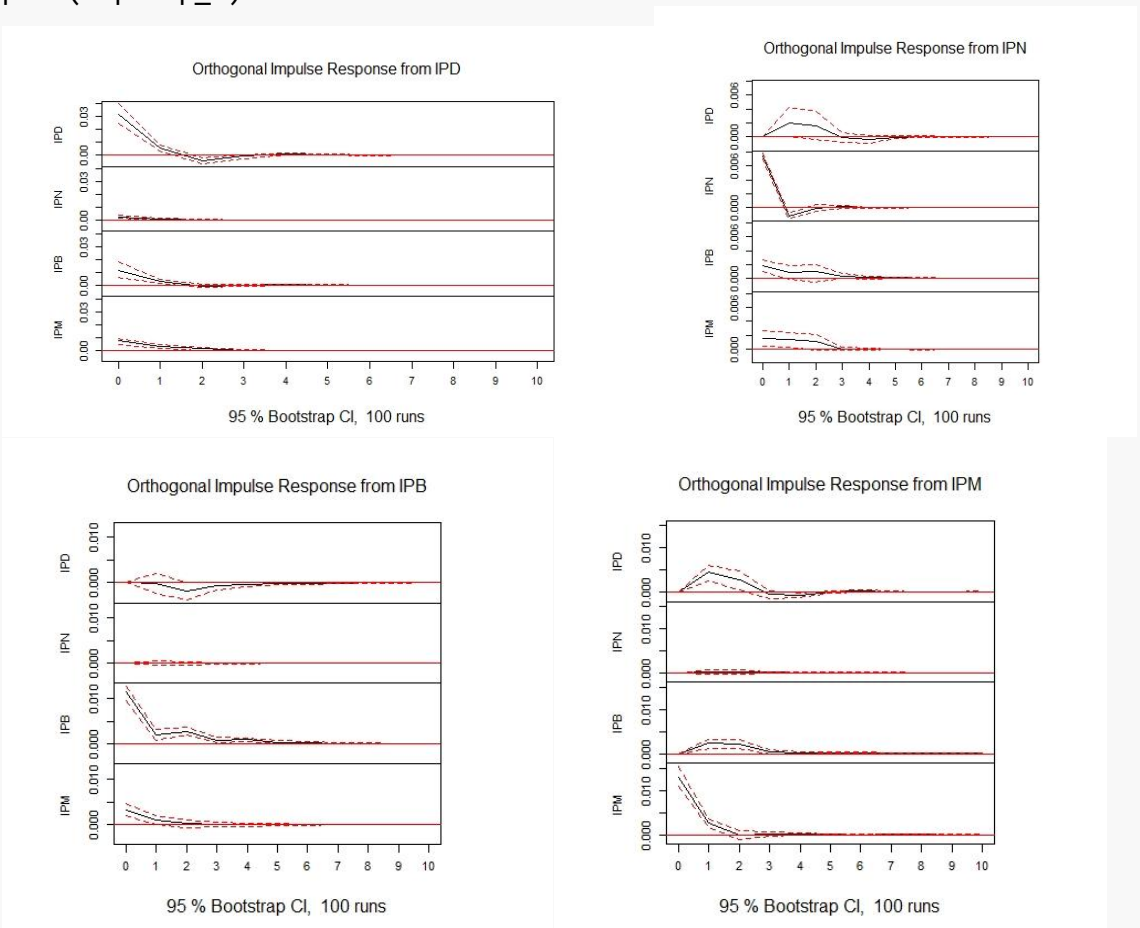
```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: IPD, IPN, IPB, IPM
## Deterministic variables: const
## Sample size: 889
```

```
## Log Likelihood: 10246.505
## Roots of the characteristic polynomial:
## 0.5908 0.4882 0.4882 0.4313 0.241 0.241 0.1895 0.1895
## Call:
## VAR(y = zt, p = 2)
##
##
## Estimation results for equation IPD:
## ====================================
## IPD = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##         Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.118760   0.047315   2.510 0.012251 *
## IPN.l1  0.219294   0.146724   1.495 0.135376
## IPB.l1 -0.107211   0.092840  -1.155 0.248490
## IPM.l1  0.338377   0.082137   4.120 4.15e-05 ***
## IPD.l2 -0.157177   0.047061  -3.340 0.000873 ***
## IPN.l2  0.188680   0.146146   1.291 0.197029
## IPB.l2 -0.197735   0.092422  -2.139 0.032671 *
## IPM.l2  0.105296   0.082487   1.277 0.202111
## const   0.001725   0.001134   1.521 0.128534
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.03183 on 880 degrees of freedom
## Multiple R-Squared: 0.08298, Adjusted R-squared: 0.07465
## F-statistic: 9.954 on 8 and 880 DF,  p-value: 2.479e-13
##
##
## Estimation results for equation IPN:
## ====================================
## IPN = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.0295464  0.0116981   2.526   0.0117 *
## IPN.l1 -0.1678360  0.0362760  -4.627 4.27e-06 ***
## IPB.l1  0.0091590  0.0229539   0.399   0.6900
## IPM.l1  0.0161758  0.0203075   0.797   0.4259
## IPD.l2  0.0004192  0.0116354   0.036   0.9713
## IPN.l2 -0.0449204  0.0361332  -1.243   0.2141
## IPB.l2 -0.0013243  0.0228504  -0.058   0.9538
## IPM.l2  0.0008203  0.0203941   0.040   0.9679
## const   0.0018111  0.0002803   6.461 1.72e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.00787 on 880 degrees of freedom
## Multiple R-Squared: 0.03302, Adjusted R-squared: 0.02423
## F-statistic: 3.756 on 8 and 880 DF,  p-value: 0.000244
##
##
## Estimation results for equation IPB:
## ====================================
## IPB = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.0033940  0.0244694   0.139 0.889715
## IPN.l1  0.0539316  0.0758804   0.711 0.477431
## IPB.l1  0.1227247  0.0480138   2.556 0.010754 *
## IPM.l1  0.1800147  0.0424782   4.238  2.5e-05 ***
## IPD.l2 -0.1421577  0.0243383  -5.841  7.3e-09 ***
## IPN.l2  0.0249723  0.0755816   0.330 0.741175
## IPB.l2  0.1827721  0.0477972   3.824 0.000141 ***
## IPM.l2  0.1031364  0.0426593   2.418 0.015822 *
## const   0.0017572  0.0005863   2.997 0.002804 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01646 on 880 degrees of freedom
## Multiple R-Squared: 0.1179,  Adjusted R-squared: 0.1098
## F-statistic:  14.7 on 8 and 880 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation IPM:
## ====================================
## IPM = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 + IPM.l
2 + const
##
##          Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.0171285  0.0226773   0.755   0.4503
## IPN.l1  0.1434833  0.0703232   2.040   0.0416 *
## IPB.l1  0.0277149  0.0444974   0.623   0.5335
## IPM.l1  0.2112295  0.0393673   5.366 1.03e-07 ***
## IPD.l2  0.0106319  0.0225559   0.471   0.6375
## IPN.l2  0.1337699  0.0700463   1.910   0.0565 .
## IPB.l2  0.0212195  0.0442968   0.479   0.6320
## IPM.l2 -0.0600639  0.0395351  -1.519   0.1291
## const   0.0012757  0.0005434   2.348   0.0191 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.01526 on 880 degrees of freedom
## Multiple R-Squared: 0.08573, Adjusted R-squared: 0.07742
```

```
## F-statistic: 10.31 on 8 and 880 DF,  p-value: 7.286e-14
##
##
##
## Covariance matrix of residuals:
##           IPD       IPN       IPB       IPM
## IPD 1.013e-03 8.423e-05 3.731e-04 2.364e-04
## IPN 8.423e-05 6.194e-05 4.452e-05 3.173e-05
## IPB 3.731e-04 4.452e-05 2.710e-04 1.270e-04
## IPM 2.364e-04 3.173e-05 1.270e-04 2.328e-04
##
## Correlation matrix of residuals:
##        IPD    IPN    IPB    IPM
## IPD 1.0000 0.3363 0.7120 0.4868
## IPN 0.3363 1.0000 0.3436 0.2643
## IPB 0.7120 0.3436 1.0000 0.5055
## IPM 0.4868 0.2643 0.5055 1.0000
```

```
#1.3 Impulse response function
impresp_1 = irf(m1)
plot(impresp_1)
```




```
#1.4 Forecast error variance decomposition
fevd(m1, n.ahead = 20)
```

```
## $IPD
##             IPD         IPN          IPB        IPM
##  [1,] 1.0000000 0.000000000 0.000000e+00 0.00000000
##  [2,] 0.9786677 0.003677477 1.547554e-05 0.01763932
##  [3,] 0.9681556 0.005825149 2.865465e-03 0.02315381
##  [4,] 0.9673749 0.005826039 3.233717e-03 0.02356535
##  [5,] 0.9664951 0.005913986 3.365580e-03 0.02422538
##  [6,] 0.9664566 0.005918078 3.400740e-03 0.02422457
##  [7,] 0.9664239 0.005917334 3.427742e-03 0.02423098
##  [8,] 0.9664153 0.005917546 3.434368e-03 0.02423283
##  [9,] 0.9664095 0.005918007 3.436407e-03 0.02423607
## [10,] 0.9664087 0.005918072 3.437077e-03 0.02423614
## [11,] 0.9664084 0.005918071 3.437398e-03 0.02423613
## [12,] 0.9664083 0.005918075 3.437499e-03 0.02423614
## [13,] 0.9664082 0.005918079 3.437531e-03 0.02423616
## [14,] 0.9664082 0.005918080 3.437542e-03 0.02423617
## [15,] 0.9664082 0.005918080 3.437546e-03 0.02423617
## [16,] 0.9664082 0.005918080 3.437547e-03 0.02423617
## [17,] 0.9664082 0.005918080 3.437548e-03 0.02423617
## [18,] 0.9664082 0.005918080 3.437548e-03 0.02423617
## [19,] 0.9664082 0.005918080 3.437548e-03 0.02423617
## [20,] 0.9664082 0.005918080 3.437548e-03 0.02423617
##
## $IPN
##             IPD       IPN          IPB          IPM
##  [1,] 0.1130659 0.8869341 0.0000000000 0.0000000000
##  [2,] 0.1176617 0.8812807 0.0003848728 0.0006727082
##  [3,] 0.1176060 0.8808902 0.0003857537 0.0011180226
##  [4,] 0.1178793 0.8805521 0.0003980169 0.0011706388
##  [5,] 0.1178761 0.8805347 0.0003990007 0.0011902249
##  [6,] 0.1178942 0.8805099 0.0003989982 0.0011968518
##  [7,] 0.1178942 0.8805093 0.0003991068 0.0011974314
##  [8,] 0.1178956 0.8805076 0.0003991836 0.0011976850
##  [9,] 0.1178956 0.8805075 0.0003991979 0.0011977477
## [10,] 0.1178956 0.8805074 0.0003991997 0.0011977775
## [11,] 0.1178956 0.8805074 0.0003992011 0.0011977783
## [12,] 0.1178956 0.8805074 0.0003992018 0.0011977789
## [13,] 0.1178956 0.8805074 0.0003992021 0.0011977792
## [14,] 0.1178956 0.8805074 0.0003992021 0.0011977793
## [15,] 0.1178956 0.8805074 0.0003992021 0.0011977793
## [16,] 0.1178956 0.8805074 0.0003992021 0.0011977793
## [17,] 0.1178956 0.8805074 0.0003992021 0.0011977793
## [18,] 0.1178956 0.8805074 0.0003992021 0.0011977793
## [19,] 0.1178956 0.8805074 0.0003992021 0.0011977793
## [20,] 0.1178956 0.8805074 0.0003992021 0.0011977793
##
## $IPB
##             IPD        IPN       IPB        IPM
##  [1,] 0.5069355 0.01224107 0.4808235 0.00000000
##  [2,] 0.5048449 0.01432290 0.4624710 0.01836117
```

```
##  [3,] 0.4828424 0.01696945 0.4679075 0.03228060
##  [4,] 0.4812822 0.01727732 0.4688451 0.03259533
##  [5,] 0.4808006 0.01725670 0.4694796 0.03246315
##  [6,] 0.4806182 0.01727671 0.4695372 0.03256786
##  [7,] 0.4804322 0.01729458 0.4696161 0.03265708
##  [8,] 0.4803961 0.01729859 0.4696392 0.03266607
##  [9,] 0.4803859 0.01729914 0.4696485 0.03266650
## [10,] 0.4803820 0.01729948 0.4696507 0.03266778
## [11,] 0.4803799 0.01729967 0.4696519 0.03266859
## [12,] 0.4803793 0.01729973 0.4696522 0.03266874
## [13,] 0.4803791 0.01729974 0.4696524 0.03266877
## [14,] 0.4803790 0.01729975 0.4696524 0.03266879
## [15,] 0.4803790 0.01729975 0.4696524 0.03266880
## [16,] 0.4803790 0.01729975 0.4696524 0.03266880
## [17,] 0.4803790 0.01729975 0.4696524 0.03266880
## [18,] 0.4803790 0.01729975 0.4696524 0.03266880
## [19,] 0.4803790 0.01729975 0.4696524 0.03266880
## [20,] 0.4803790 0.01729975 0.4696524 0.03266880
##
## $IPM
##             IPD        IPN        IPB       IPM
##  [1,] 0.2369914 0.01140501 0.04501385 0.7065897
##  [2,] 0.2512518 0.01903374 0.04569678 0.6840177
##  [3,] 0.2554440 0.02378324 0.04554842 0.6752243
##  [4,] 0.2555947 0.02379966 0.04558903 0.6750166
##  [5,] 0.2556231 0.02379681 0.04559513 0.6749849
##  [6,] 0.2556249 0.02379765 0.04559845 0.6749790
##  [7,] 0.2556289 0.02379741 0.04560048 0.6749732
##  [8,] 0.2556296 0.02379737 0.04560115 0.6749719
##  [9,] 0.2556295 0.02379740 0.04560134 0.6749718
## [10,] 0.2556295 0.02379741 0.04560140 0.6749717
## [11,] 0.2556295 0.02379741 0.04560143 0.6749717
## [12,] 0.2556295 0.02379741 0.04560144 0.6749717
## [13,] 0.2556295 0.02379741 0.04560145 0.6749717
## [14,] 0.2556295 0.02379741 0.04560145 0.6749717
## [15,] 0.2556295 0.02379741 0.04560145 0.6749717
## [16,] 0.2556295 0.02379741 0.04560145 0.6749717
## [17,] 0.2556295 0.02379741 0.04560145 0.6749717
## [18,] 0.2556295 0.02379741 0.04560145 0.6749717
## [19,] 0.2556295 0.02379741 0.04560145 0.6749717
## [20,] 0.2556295 0.02379741 0.04560145 0.6749717
```

```
#1.5 Prediction
m1.prd <- predict(m1, n.ahead = 6, ci = 0.95)
m1.prd
```

```
## $IPD
##                fcst       lower      upper         CI
## [1,] -0.0006150473 -0.06300249 0.06177239 0.06238744
## [2,]  0.0077907327 -0.05624736 0.07182883 0.06403809
```

```
## [3,]  0.0018461230 -0.06318302 0.06687526 0.06502914
## [4,]  0.0005341044 -0.06455561 0.06562382 0.06508971
## [5,]  0.0018430627 -0.06330032 0.06698645 0.06514339
## [6,]  0.0024660374 -0.06267897 0.06761105 0.06514501
##
## $IPN
##                fcst       lower      upper         CI
## [1,] -0.0005426783 -0.01596737 0.01488201 0.01542469
## [2,]  0.0015209749 -0.01415492 0.01719687 0.01567589
## [3,]  0.0019159580 -0.01376365 0.01759557 0.01567961
## [4,]  0.0015594667 -0.01412572 0.01724465 0.01568519
## [5,]  0.0015469714 -0.01413842 0.01723237 0.01568540
## [6,]  0.0016062462 -0.01407939 0.01729188 0.01568563
##
## $IPB
##              fcst       lower      upper         CI
## [1,] 0.009027450 -0.02323714 0.04129204 0.03226459
## [2,] 0.007194082 -0.02619747 0.04058564 0.03339155
## [3,] 0.005411836 -0.02877592 0.03959959 0.03418775
## [4,] 0.003508809 -0.03073996 0.03775757 0.03424877
## [5,] 0.003737427 -0.03059243 0.03806729 0.03432986
## [6,] 0.003594820 -0.03075094 0.03794058 0.03434576
##
## $IPM
##              fcst       lower      upper         CI
## [1,] 0.003852687 -0.02604896 0.03375434 0.02990165
## [2,] 0.003011154 -0.02805045 0.03407276 0.03106160
## [3,] 0.002343799 -0.02891949 0.03360708 0.03126329
## [4,] 0.002485357 -0.02878334 0.03375405 0.03126869
## [5,] 0.002380792 -0.02888995 0.03365153 0.03127074
## [6,] 0.002275148 -0.02899574 0.03354604 0.03127089
```

```
#Question 2
getSymbols("DOGE-USD",src="yahoo",from = "2015-01-01", to = "2021-05-24")
```

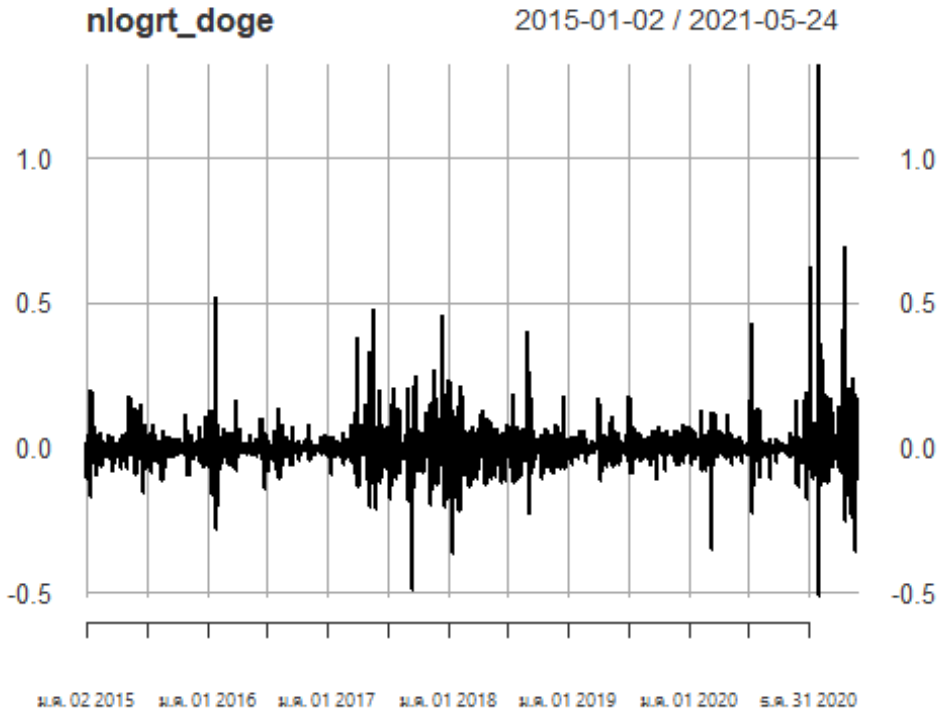
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
## Warning: DOGE-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.
```

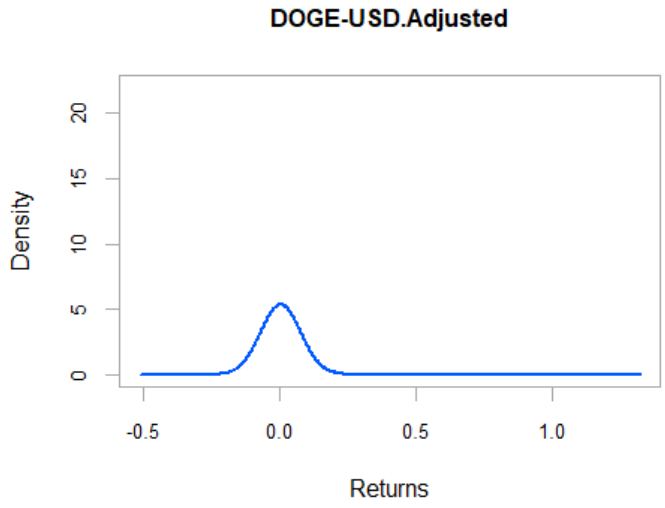
```
## [1] "DOGE-USD"
```

```
da1 = na.omit(`DOGE-USD`)
doge_price = da1[,6]
plot(doge_price)
```

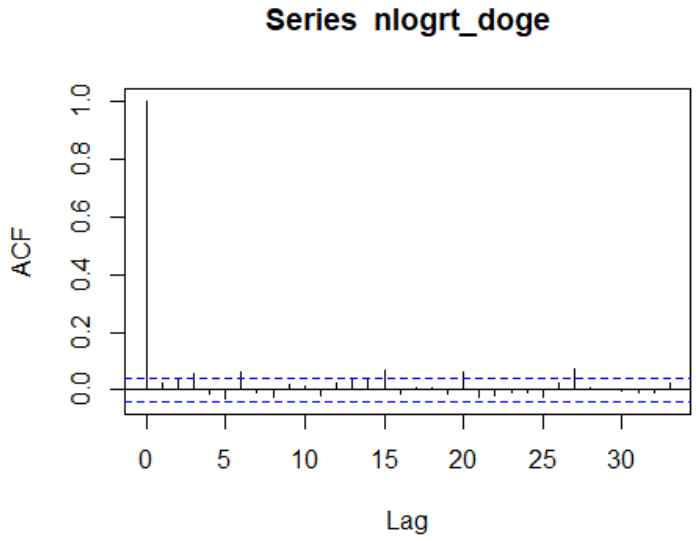

```
log_dogeprice = log(doge_price)
logrt_doge = diff(log_dogeprice)
nlogrt_doge = logrt_doge[2:nrow(logrt_doge),]
plot(nlogrt_doge)
```



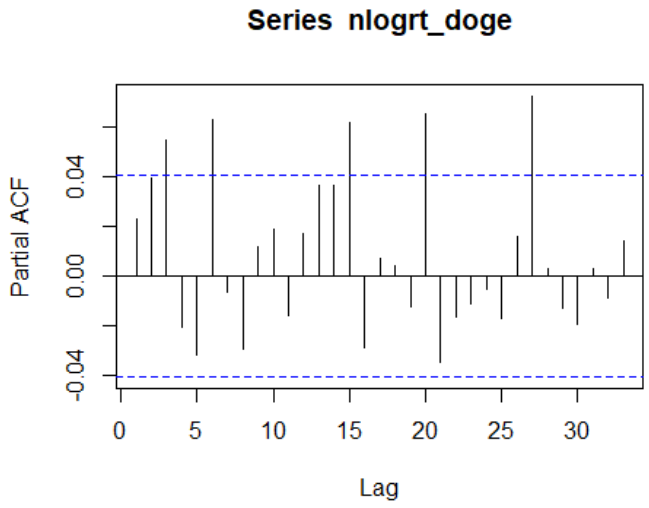
```
chart.Histogram(nlogrt_doge, methods = c("add.normal"))
```



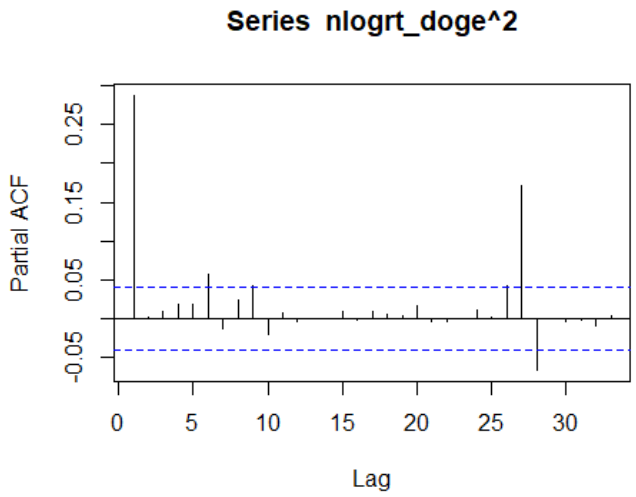

```
acf(nlogrt_doge)
```




```
pacf(nlogrt_doge)
```




```
pacf(nlogrt_doge^2)
```




```
Box.test(nlogrt_doge^2, lag = 10, type = "Ljung")

##
##  Box-Ljung test
##
## data:  nlogrt_doge^2
## X-squared = 238.03, df = 10, p-value < 2.2e-16

table.Stats(nlogrt_doge)

##                  DOGE-USD.Adjusted
## Observations              2331.0000
## NAs                          0.0000
## Minimum                     -0.5151
## Quartile 1                  -0.0213
```

```
## Median                       0.0000
## Arithmetic Mean              0.0033
## Geometric Mean               0.0007
## Quartile 3                   0.0189
## Maximum                      1.3235
## SE Mean                      0.0015
## LCL Mean (0.95)              0.0002
## UCL Mean (0.95)              0.0063
## Variance                     0.0055
## Stdev                        0.0742
## Skewness                     3.6324
## Kurtosis                    55.8828
```

```
t.test(nlogrt_doge)
```

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

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

```
##
##  One Sample t-test
##
## data:  nlogrt_doge
## 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
```

```
Box.test(nlogrt_doge, lag = 15, type = "Ljung")
```

```
##
##  Box-Ljung test
##
## data:  nlogrt_doge
## X-squared = 46.728, df = 15, p-value = 4.07e-05
```

```
m2 = garchFit(~arma(1,1)+garch(1,1),data = nlogrt_doge, 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 = ~arma(1, 1) + garch(1, 1), data = nlogrt_doge,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(1, 1) + garch(1, 1)
## <environment: 0x0000000020fb2ee8>
##  [data = nlogrt_doge]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1          ma1        omega       alpha1        bet
a1
## -0.00142729  -0.62734232   0.56568900   0.00013459   0.30047548   0.734592
27
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -1.427e-03   1.230e-03   -1.161  0.24583
## ar1    -6.273e-01   2.035e-01   -3.083  0.00205 **
## ma1     5.657e-01   2.146e-01    2.636  0.00840 **
## omega   1.346e-04   1.873e-05    7.188 6.59e-13 ***
## alpha1  3.005e-01   2.715e-02   11.068  < 2e-16 ***
## beta1   7.346e-01   1.927e-02   38.126  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3678.442    normalized:  1.578053
##
## Description:
##  Fri Jun 11 16:32:01 2021 by user: Jidapa
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  10169.55  0
##  Shapiro-Wilk Test  R    W      0.8740751 0
##  Ljung-Box Test     R    Q(10)  26.2674   0.003396709
##  Ljung-Box Test     R    Q(15)  46.13667  5.057359e-05
##  Ljung-Box Test     R    Q(20)  48.83952  0.0003239674
```

```
##  Ljung-Box Test     R^2  Q(10)  16.40989  0.08848471
##  Ljung-Box Test     R^2  Q(15)  17.02452  0.3174007
##  Ljung-Box Test     R^2  Q(20)  19.35229  0.499044
##  LM Arch Test       R    TR^2   16.17184  0.1834915
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.150958 -3.136147 -3.150972 -3.145562
```

```
m3 = garchFit(~garch(1,1),data = nlogrt_doge, 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(m3)
```

```
##
## Title:
##  GARCH Modelling
##
## Call:
##  garchFit(formula = ~garch(1, 1), data = nlogrt_doge, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(1, 1)
## <environment: 0x000000001fa99178>
##  [data = nlogrt_doge]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu        omega       alpha1        beta1
## -0.00081557   0.00013016   0.28884085   0.74264416
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     -8.156e-04   7.817e-04   -1.043    0.297
## omega   1.302e-04   1.801e-05    7.225    5e-13 ***
## alpha1  2.888e-01   2.560e-02   11.285   <2e-16 ***
## beta1   7.426e-01   1.834e-02   40.495   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3674.05    normalized:  1.576169
##
```

```
## Description:
##  Fri Jun 11 16:32:01 2021 by user: Jidapa
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  10080.55  0
##  Shapiro-Wilk Test  R    W      0.8762516 0
##  Ljung-Box Test     R    Q(10)  20.04457  0.02883397
##  Ljung-Box Test     R    Q(15)  37.57527  0.001042214
##  Ljung-Box Test     R    Q(20)  40.33482  0.00453035
##  Ljung-Box Test     R^2  Q(10)  16.08876  0.09711968
##  Ljung-Box Test     R^2  Q(15)  16.57086  0.3451582
##  Ljung-Box Test     R^2  Q(20)  19.07062  0.5172398
##  LM Arch Test       R    TR^2   15.769    0.2020467
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.148906 -3.139032 -3.148912 -3.145309

m4 = garchFit(~garch(3,3), data = nlogrt_doge, trace = F)

## Warning in sqrt(diag(fit$cvar)): NaNs produced

## 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(3, 3), data = nlogrt_doge, trace = F)
##
## Mean and Variance Equation:
##  data ~ garch(3, 3)
## <environment: 0x0000000021165030>
##  [data = nlogrt_doge]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu        omega       alpha1       alpha2       alpha3        bet
a1
## -0.00054428   0.00019389   0.42671649   0.00000001   0.00000001   0.234854
33
##       beta2        beta3
```

```
##  0.09159923   0.28125563
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##         Estimate  Std. Error  t value Pr(>|t|)
## mu     -5.443e-04   7.675e-04   -0.709    0.478
## omega   1.939e-04   2.162e-05    8.968  < 2e-16 ***
## alpha1  4.267e-01   4.001e-02   10.665  < 2e-16 ***
## alpha2  1.000e-08          NA       NA       NA
## alpha3  1.000e-08   6.029e-02    0.000    1.000
## beta1   2.349e-01          NA       NA       NA
## beta2   9.160e-02          NA       NA       NA
## beta3   2.813e-01   6.296e-02    4.467 7.92e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3699.473    normalized:  1.587076
##
## Description:
##  Fri Jun 11 16:32:01 2021 by user: Jidapa
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  8564.509  0
##  Shapiro-Wilk Test  R    W      0.883077  0
##  Ljung-Box Test     R    Q(10)  21.19589  0.01976828
##  Ljung-Box Test     R    Q(15)  39.44757  0.0005492046
##  Ljung-Box Test     R    Q(20)  42.55431  0.002340269
##  Ljung-Box Test     R^2  Q(10)  5.728417  0.837541
##  Ljung-Box Test     R^2  Q(15)  6.285072  0.974551
##  Ljung-Box Test     R^2  Q(20)  9.653935  0.9740218
##  LM Arch Test       R    TR^2   5.520496  0.9383025
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.167287 -3.147539 -3.167311 -3.160092
```

```
m5 = garchFit(~arma(6,6)+garch(3,3), data = nlogrt_doge, trace = F)
```

```
## Warning in arima(.series$x, order = c(u, 0, v), include.mean = include.mea
n):
## possible convergence problem: optim gave code = 1
```

```
## Warning in sqrt(diag(fit$cvar)): NaNs produced
```

```
## 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 = ~arma(6, 6) + garch(3, 3), data = nlogrt_doge,
##     trace = F)
##
## Mean and Variance Equation:
##  data ~ arma(6, 6) + garch(3, 3)
## <environment: 0x0000000018717428>
##  [data = nlogrt_doge]
##
## Conditional Distribution:
##  norm
##
## Coefficient(s):
##          mu          ar1          ar2          ar3          ar4          a
r5
## -0.00106457  -0.31776780   0.29236980   0.31640890   0.19993482  -0.793727
82
##         ar6          ma1          ma2          ma3          ma4          m
a5
## -0.31553448   0.24550592  -0.24463262  -0.26051625  -0.17631977   0.780684
09
##         ma6        omega       alpha1       alpha2       alpha3        bet
a1
##  0.24126880   0.00019573   0.45323666   0.00000001   0.00000001   0.266606
90
##       beta2        beta3
##  0.05266579   0.27098636
##
## Std. Errors:
##  based on Hessian
##
## Error Analysis:
##          Estimate  Std. Error  t value Pr(>|t|)
## mu     -1.065e-03   1.214e-03   -0.877 0.380582
## ar1    -3.178e-01   2.262e-01   -1.405 0.160100
## ar2     2.924e-01   5.777e-02    5.061 4.18e-07 ***
## ar3     3.164e-01   7.966e-02    3.972 7.13e-05 ***
## ar4     1.999e-01   5.048e-02    3.961 7.46e-05 ***
## ar5    -7.937e-01   4.895e-02  -16.216  < 2e-16 ***
## ar6    -3.155e-01   1.782e-01   -1.771 0.076587 .
```

```
## ma1     2.455e-01   2.384e-01    1.030 0.303091
## ma2    -2.446e-01   6.024e-02   -4.061 4.89e-05 ***
## ma3    -2.605e-01   7.262e-02   -3.587 0.000334 ***
## ma4    -1.763e-01   5.069e-02   -3.478 0.000505 ***
## ma5     7.807e-01   5.313e-02   14.695  < 2e-16 ***
## ma6     2.413e-01   1.888e-01    1.278 0.201360
## omega   1.957e-04   2.452e-05    7.981 1.55e-15 ***
## alpha1  4.532e-01   4.330e-02   10.467  < 2e-16 ***
## alpha2  1.000e-08          NA       NA       NA
## alpha3  1.000e-08   4.652e-02    0.000 1.000000
## beta1   2.666e-01          NA       NA       NA
## beta2   5.267e-02          NA       NA       NA
## beta3   2.710e-01   6.005e-02    4.512 6.41e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
##  3721.343    normalized:  1.596458
##
## Description:
##  Fri Jun 11 16:32:07 2021 by user: Jidapa
##
##
## Standardised Residuals Tests:
##                                 Statistic p-Value
##  Jarque-Bera Test   R    Chi^2  7821.927  0
##  Shapiro-Wilk Test  R    W      0.8876106 0
##  Ljung-Box Test     R    Q(10)  16.86437  0.07741961
##  Ljung-Box Test     R    Q(15)  25.55087  0.04301705
##  Ljung-Box Test     R    Q(20)  27.40361  0.1242859
##  Ljung-Box Test     R^2  Q(10)  5.10862   0.8838049
##  Ljung-Box Test     R^2  Q(15)  6.098591  0.9780436
##  Ljung-Box Test     R^2  Q(20)  9.138104  0.9812518
##  LM Arch Test       R    TR^2   5.11456   0.9540426
##
## Information Criterion Statistics:
##       AIC       BIC       SIC      HQIC
## -3.175755 -3.126386 -3.175901 -3.157768
```

```
#Question 3
getSymbols("CLVMNACSCAB1GQUK", src = "FRED")
```

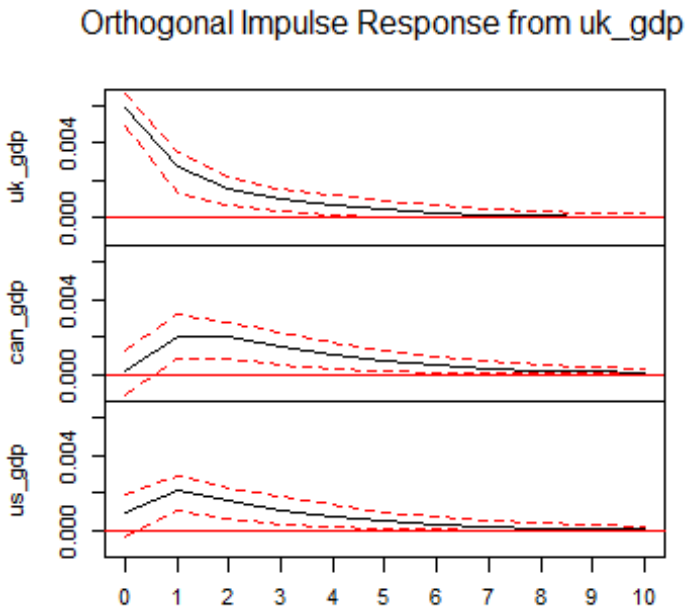
```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

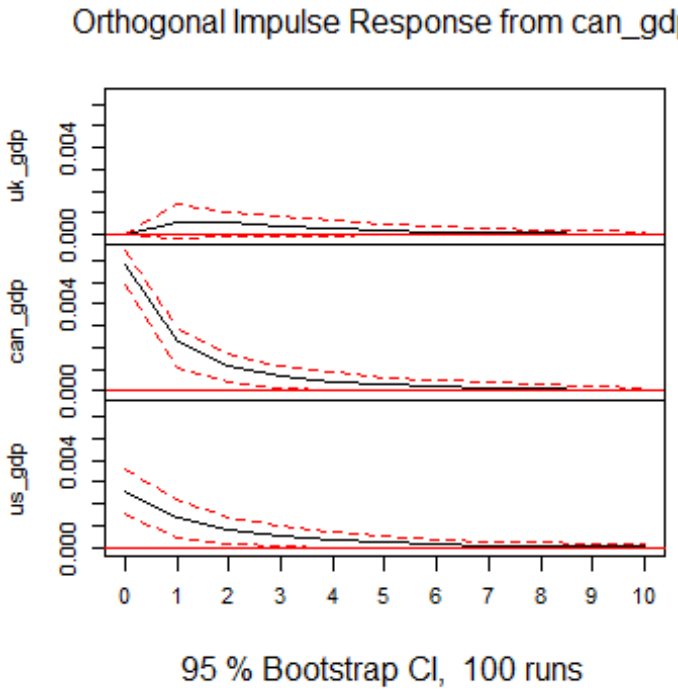
```
## [1] "CLVMNACSCAB1GQUK"

uk_gdp = CLVMNACSCAB1GQUK[c(21:146)]
getSymbols("NAEXKP01CAQ189S", src = "FRED")

## [1] "NAEXKP01CAQ189S"

can_gdp = NAEXKP01CAQ189S[c(77:202)]
getSymbols("GDPC1", src = "FRED")

## [1] "GDPC1"

us_gdp = GDPC1[c(133:258)]
gdp = cbind(as.numeric(uk_gdp), as.numeric(can_gdp), as.numeric(us_gdp))
colnames(gdp) <- c("uk_gdp", "can_gdp", "us_gdp")
head(gdp)

##        uk_gdp      can_gdp   us_gdp
## [1,] 202901.2 209975000000 6837.641
## [2,] 198842.4 209908250000 6696.753
## [3,] 198646.1 209667250000 6688.794
## [4,] 196470.2 212561000000 6813.535
## [5,] 196223.3 217245750000 6947.042
## [6,] 196624.9 219687500000 6895.559

zt2 = diff(log(gdp))
head(zt2)

##             uk_gdp       can_gdp       us_gdp
## [1,] -0.0202066099 -0.0003179455 -0.020820007
## [2,] -0.0009877016 -0.0011487803 -0.001189193
## [3,] -0.0110140837  0.0137072558  0.018477486
## [4,] -0.0012574694  0.0218001952  0.019404882
## [5,]  0.0020445563  0.0111768808 -0.007438376
## [6,]  0.0107141768 -0.0090180657  0.011904108

#3.1
varfit = VAR(zt2, p = 1)
summary(varfit)

##
## VAR Estimation Results:
## ========================
## Endogenous variables: uk_gdp, can_gdp, us_gdp
## Deterministic variables: const
## Sample size: 124
## Log Likelihood: 1399.942
## Roots of the characteristic polynomial:
## 0.6618 0.2389 0.06806
## Call:
## VAR(y = zt2, p = 1)
##
```

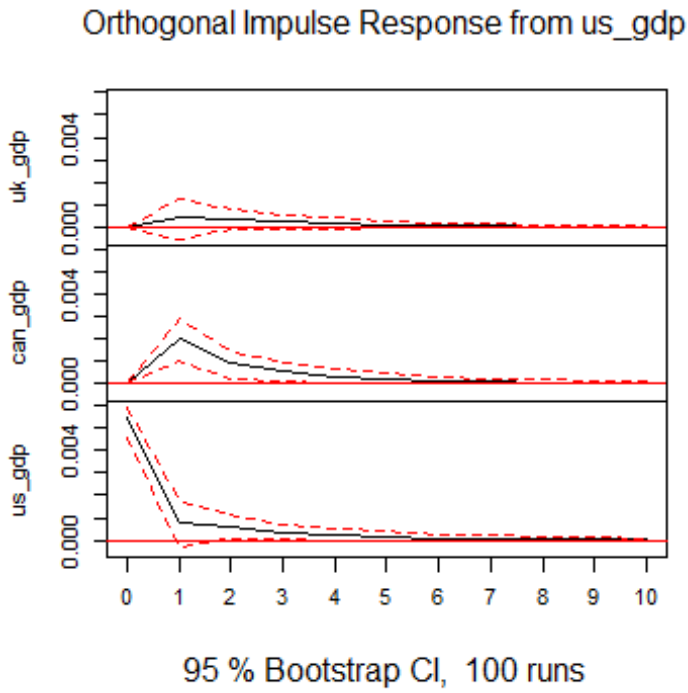
```
##
## Estimation results for equation uk_gdp:
## ======================================
## uk_gdp = uk_gdp.l1 + can_gdp.l1 + us_gdp.l1 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## uk_gdp.l1  0.4613685  0.0811121   5.688 9.23e-08 ***
## can_gdp.l1 0.0604288  0.0835100   0.724  0.47071
## us_gdp.l1  0.0838852  0.0915521   0.916  0.36137
## const      0.0022500  0.0007361   3.057  0.00276 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005808 on 120 degrees of freedom
## Multiple R-Squared: 0.3229,  Adjusted R-squared: 0.306
## F-statistic: 19.08 on 3 and 120 DF,  p-value: 3.497e-10
##
##
## Estimation results for equation can_gdp:
## ======================================
## can_gdp = uk_gdp.l1 + can_gdp.l1 + us_gdp.l1 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## uk_gdp.l1  0.2845506  0.0815253   3.490 0.000676 ***
## can_gdp.l1 0.2237861  0.0839354   2.666 0.008730 **
## us_gdp.l1  0.3731921  0.0920184   4.056 8.93e-05 ***
## const      0.0006166  0.0007398   0.833 0.406273
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.005838 on 120 degrees of freedom
## Multiple R-Squared: 0.476,   Adjusted R-squared: 0.4629
## F-statistic: 36.34 on 3 and 120 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation us_gdp:
## ======================================
## us_gdp = uk_gdp.l1 + can_gdp.l1 + us_gdp.l1 + const
##
##             Estimate Std. Error t value Pr(>|t|)
## uk_gdp.l1  0.3349956  0.0854314   3.921 0.000147 ***
## can_gdp.l1 0.1666529  0.0879570   1.895 0.060537 .
## us_gdp.l1  0.1474612  0.0964273   1.529 0.128835
## const      0.0030676  0.0007753   3.957 0.000129 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
```

```
## Residual standard error: 0.006118 on 120 degrees of freedom
## Multiple R-Squared: 0.3031,  Adjusted R-squared: 0.2857
## F-statistic:  17.4 on 3 and 120 DF,  p-value: 1.916e-09
##
##
##
## Covariance matrix of residuals:
##            uk_gdp   can_gdp    us_gdp
## uk_gdp  3.374e-05 1.158e-06 5.286e-06
## can_gdp 1.158e-06 3.408e-05 1.502e-05
## us_gdp  5.286e-06 1.502e-05 3.742e-05
##
## Correlation matrix of residuals:
##          uk_gdp can_gdp us_gdp
## uk_gdp  1.00000 0.03416 0.1488
## can_gdp 0.03416 1.00000 0.4206
## us_gdp  0.14877 0.42058 1.0000
```

```
#3.2
impresp_2 = irf(varfit)
plot(impresp_2)
```


Orthogonal Impulse Response from uk_gdp
uk_gdp
can_gdp
us_gdp
0.000
0.004
0 1 2 3 4 5 6 7 8 9 10
95 % Bootstrap CI, 100 runs


Orthogonal Impulse Response from can_gdp
uk_gdp
can_gdp
us_gdp
0.000
0.004
0 1 2 3 4 5 6 7 8 9 10
95 % Bootstrap CI, 100 runs




```
#3.3
fevd(varfit)
```

```
## $uk_gdp
##          uk_gdp     can_gdp      us_gdp
##  [1,] 1.0000000 0.000000000 0.000000000
##  [2,] 0.9873062 0.007637889 0.005055939
##  [3,] 0.9786801 0.012950673 0.008369257
##  [4,] 0.9744027 0.015593097 0.010004195
##  [5,] 0.9724591 0.016795246 0.010745679
##  [6,] 0.9715988 0.017327541 0.011073663
##  [7,] 0.9712210 0.017561317 0.011217656
##  [8,] 0.9710555 0.017663752 0.011280742
##  [9,] 0.9709830 0.017708613 0.011308369
## [10,] 0.9709513 0.017728258 0.011320467
##
## $can_gdp
##            uk_gdp   can_gdp     us_gdp
##  [1,] 0.001166889 0.9988331 0.00000000
##  [2,] 0.088174586 0.8235085 0.08831691
##  [3,] 0.154625783 0.7526102 0.09276398
##  [4,] 0.185677815 0.7212522 0.09307002
##  [5,] 0.199371622 0.7076932 0.09293518
##  [6,] 0.205342976 0.7018169 0.09284008
##  [7,] 0.207947312 0.6992597 0.09279296
##  [8,] 0.209084903 0.6981436 0.09277151
##  [9,] 0.209582404 0.6976556 0.09276199
```

```
## [10,] 0.209800130 0.6974421 0.09275780
##
## $us_gdp
##           uk_gdp   can_gdp    us_gdp
##  [1,] 0.02213163 0.1728444 0.8050239
##  [2,] 0.11933560 0.1867431 0.6939213
##  [3,] 0.16282199 0.1854647 0.6517133
##  [4,] 0.18210896 0.1841691 0.6337220
##  [5,] 0.19049060 0.1834997 0.6260097
##  [6,] 0.19413918 0.1831926 0.6226683
##  [7,] 0.19573110 0.1830561 0.6212128
##  [8,] 0.19642689 0.1829961 0.6205771
##  [9,] 0.19673130 0.1829697 0.6202990
## [10,] 0.19686456 0.1829582 0.6201772
```

```
#3.4
m6 = adfTest(uk_gdp, lag = 3, type = c("ct"))
```

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

```
m6
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -1.8921
##   P VALUE:
##     0.6214
##
## Description:
##  Fri Jun 11 21:25:41 2021 by user: Jidapa
```

```
m7= adfTest(can_gdp, lag = 3, type = c("ct"))
```

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

```
m7
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
```

```
##   PARAMETER:
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -2.2593
##   P VALUE:
##     0.4688
##
## Description:
##  Fri Jun 11 21:25:41 2021 by user: Jidapa
```

```
m8 = adfTest(us_gdp, lag = 3, type = c("ct"))
```

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

```
m8
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -2.5164
##   P VALUE:
##     0.3619
##
## Description:
##  Fri Jun 11 21:25:41 2021 by user: Jidapa
```

```
uk2 = diff(uk_gdp)
uk2 = uk2[c(2:nrow(uk_gdp))]
m9 = adfTest(uk2, lag = 3, type = c("ct"))
```

```
## 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(uk2, lag = 3, type = c("ct")): p-value smaller than pri
nted
## p-value
```

```
m9
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
```

```
##   PARAMETER:
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -4.8655
##   P VALUE:
##     0.01
##
## Description:
##  Fri Jun 11 21:25:41 2021 by user: Jidapa
```

```
can2 = diff(can_gdp)
can2 = can2[c(2:nrow(can_gdp))]
m10 = adfTest(can2, lag = 3, type = c("ct"))
```

```
## 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(can2, lag = 3, type = c("ct")): p-value smaller than pr
inted
## p-value
```

```
m10
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -4.4814
##   P VALUE:
##     0.01
##
## Description:
##  Fri Jun 11 21:25:41 2021 by user: Jidapa
```

```
us2 = diff(us_gdp)
us2 = us2[c(2:nrow(us_gdp))]
m11 = adfTest(us2, lag = 3, type = c("ct"))
```

```
## 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(us2, lag = 3, type = c("ct")): p-value smaller than pri
nted
## p-value
```

```
m11
```

```
##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -4.0632
##   P VALUE:
##     0.01


fit <- lm(uk_gdp~can_gdp+us_gdp)
summary(fit)

##
## Call:
## lm(formula = uk_gdp ~ can_gdp + us_gdp)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8698.8 -2583.5   -18.8  3228.3  8552.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.145e+04  2.537e+03  12.394  < 2e-16 ***
## can_gdp     2.463e-07  5.792e-08   4.252 4.15e-05 ***
## us_gdp      1.727e+01  1.459e+00  11.839  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4130 on 123 degrees of freedom
## Multiple R-squared:  0.9967, Adjusted R-squared:  0.9966
## F-statistic: 1.833e+04 on 2 and 123 DF,  p-value: < 2.2e-16

error = residuals(fit)
m12 = adfTest(error, lags = 3 , type = c("ct"))

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

m12

##
## Title:
##  Augmented Dickey-Fuller Test
##
## Test Results:
##   PARAMETER:
```

```
##     Lag Order: 3
##   STATISTIC:
##     Dickey-Fuller: -2.9168
##   P VALUE:
##     0.1955
##


#3.5 VECM
uk.l1 = Lag(uk2, k=1)
can.l1 = Lag(can2, k=1)
us.l1 = Lag(us2, k=1)
error.l1 = Lag(error, k=1)
error.l1 = error.l1[2:126]
fit2 <- lm(uk2~uk.l1+can.l1+us.l1+error.l1)
summary(fit2)

##
## Call:
## lm(formula = uk2 ~ uk.l1 + can.l1 + us.l1 + error.l1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5308.7  -822.4    21.3   991.8  4078.8
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.988e+02  1.980e+02   2.519  0.01311 *
## uk.l1        6.088e-01  8.202e-02   7.422 1.89e-11 ***
## can.l1      -1.046e-07  7.722e-08  -1.355  0.17809
## us.l1        5.312e+00  2.429e+00   2.187  0.03073 *
## error.l1    -1.169e-01  3.706e-02  -3.155  0.00203 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1590 on 119 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.4798, Adjusted R-squared:  0.4623
## F-statistic: 27.44 on 4 and 119 DF,  p-value: 3.839e-16

fit3<- lm(can2~uk.l1+can.l1+us.l1+error.l1)
summary(fit3)

##
## Call:
## lm(formula = can2 ~ uk.l1 + can.l1 + us.l1 + error.l1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -3.365e+09 -9.490e+08 -1.804e+08  1.125e+09  4.086e+09
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.944e+08  2.105e+08   0.924   0.3575
## uk.l1        3.712e+05  8.717e+04   4.258 4.14e-05 ***
## can.l1       1.855e-01  8.207e-02   2.260   0.0256 *
## us.l1        1.044e+07  2.582e+06   4.044 9.36e-05 ***
## error.l1    -3.009e+04  3.938e+04  -0.764   0.4464
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.69e+09 on 119 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.5385, Adjusted R-squared:  0.523
## F-statistic: 34.71 on 4 and 119 DF,  p-value: < 2.2e-16
```

```
fit4 <- lm(us2~uk.l1+can.l1+us.l1+error.l1)
summary(fit4)
```

```
##
## Call:
## lm(formula = us2 ~ uk.l1 + can.l1 + us.l1 + error.l1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -259.700  -36.490    1.381   46.848  140.981
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3.320e+01  8.006e+00   4.146 6.36e-05 ***
## uk.l1       1.691e-02  3.316e-03   5.099 1.30e-06 ***
## can.l1      2.375e-09  3.122e-09   0.761    0.448
## us.l1       1.151e-01  9.820e-02   1.172    0.243
## error.l1    1.044e-03  1.498e-03   0.697    0.487
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 64.29 on 119 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.3625, Adjusted R-squared:  0.3411
## F-statistic: 16.92 on 4 and 119 DF,  p-value: 5.246e-11
```