

## Take-Home Exam Spring 2020

**DUE DATE:** Friday 11<sup>th</sup>, June 2021 Before mid-night.

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

Your Name: Achariyaporn T.

Your Student ID: 6104640336

**There are Three questions.**

### Instruction:

- You have to submit (1) The R codes (YourID\_Exam.R), (2) Your report with the empirical analysis.
- If you have any question related to this exam, please send me the e-mail at Wasin@econ.tu.ac.th with the Topic: EE435: Take-home exam questions.

### Question1. (20 Marks)

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

#### Toolbox 1

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

```

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

```

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

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

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

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

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

### Question2. (20 Marks)

Consider the price of the following cryptocurrencies.

Instruction:

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

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

The data are available from Yahoo Finance.

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

volatilities.

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

**Question3 (20 marks)**

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

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

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

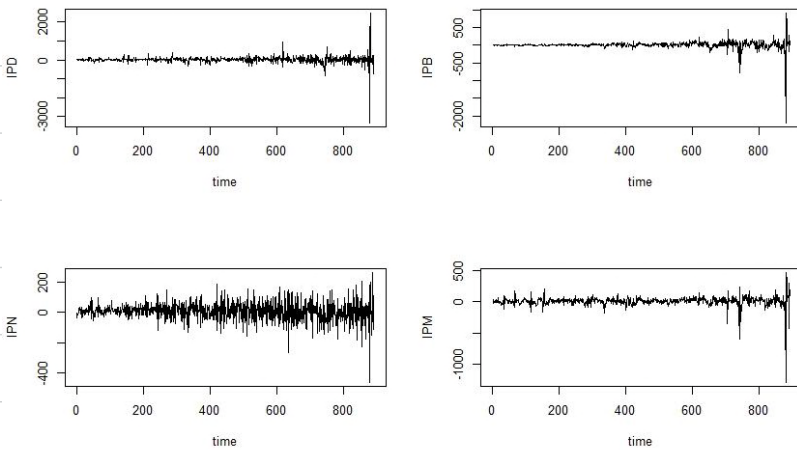
3.3 Obtain the forecast error variance decomposition. Interpret the results.

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

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

### Question 1

1.1 Construct the growth rate series  $z_t$  of the four industrial production index, i.e. take the first difference of the log data. Obtain time plots of  $z_t$ . Comment on the time plot.



To get this time plot I first install MTS package and uninstall it later in order to be able to use VAR package. From the four time plot for the index, IPN index seems to be the most volatile and fluctuate throughout the time. While the other 3 index do not volatile much.

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

```
> VARorder(Zt)
selected order: aic = 11
selected order: bic = 3
selected order: hq = 3
summary table:
      p   AIC   BIC   HQ   M(p) p-value
[1,]  0 35.0988 35.0988 35.0988  0.0000 0.0000
[2,]  1 34.8843 34.9703 34.9171 218.5604 0.0000
[3,]  2 34.7608 34.9330 34.8266 138.3796 0.0000
[4,]  3 34.6518 34.9100 34.7505 125.2874 0.0000
[5,]  4 34.6234 34.9676 34.7550  55.3489 0.0000
[6,]  5 34.6043 35.0346 34.7687  47.1416 0.0001
[7,]  6 34.5646 35.0810 34.7620  64.4401 0.0000
[8,]  7 34.5416 35.1440 34.7718  50.0189 0.0000
[9,]  8 34.5452 35.2337 34.8083  27.2595 0.0387
[10,] 9 34.5334 35.3079 34.8294  40.1391 0.0007
[11,] 10 34.5324 35.3929 34.8613  30.8851 0.0139
[12,] 11 34.5240 35.4706 34.8858  36.8738 0.0022
[13,] 12 34.5247 35.5574 34.9194  29.1783 0.0228
[14,] 13 34.5399 35.6586 34.9674  17.1065 0.3787
> m1=VAR(Zt,3)
```

→ Base on VARorder we select lag p = 3 base on BIC criteria.

```
## Estimation results for equation IPD:
## =====
## IPD = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.15653  0.05476  2.858  0.00436 **
## IPN.l1  0.08917  0.11855  0.752  0.45213
## IPB.l1  0.07629  0.10513  0.726  0.46823
## IPM.l1 -0.16784  0.13912 -1.206  0.22799
## IPD.l2 -0.21923  0.05372 -4.081  4.9e-05 ***
## IPN.l2  0.02163  0.12243  0.177  0.85981
## IPB.l2  0.14077  0.10388  1.355  0.17572
## IPM.l2 -0.24366  0.13953 -1.746  0.08112 .
## IPD.l3 -0.17652  0.05430 -3.251  0.00119 ***
## IPN.l3  0.11960  0.11878  1.007  0.31429
## IPB.l3  0.09700  0.10554  0.919  0.35831
## IPM.l3 -0.08901  0.13798 -0.645  0.51903
## const 12.15593  6.76687  1.796  0.07278 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Estimation results for equation IPN:
## =====
## IPN = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.046301  0.016989  2.725  0.00655 **
## IPN.l1 -0.261433  0.036777 -7.109 2.43e-12 ***
## IPB.l1 -0.019290  0.032613 -0.591  0.55436
## IPM.l1  0.056677  0.043161  1.313  0.18947
## IPD.l2 -0.018761  0.016666 -1.126  0.26061
## IPN.l2 -0.097276  0.037982 -2.561  0.01060 *
## IPB.l2  0.058199  0.032226  1.806  0.07127 .
## IPM.l2 -0.074456  0.043287 -1.720  0.08578 .
## IPD.l3 -0.002635  0.016845 -0.156  0.87575
## IPN.l3  0.007021  0.036850  0.191  0.84894
## IPB.l3 -0.003110  0.032741 -0.095  0.92435
## IPM.l3  0.015960  0.042805  0.373  0.70936
## const 10.837543  2.099282  5.162 3.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$\hat{IPD}_t = 12.15593 + 0.15653 IPD_{t-1} - 0.21923 IPD_{t-2} - 0.24366 IPM_{t-2} - 0.17652 IPD_{t-3}$$

(6.76687) (0.05476) (0.05372) (0.13953) (0.05430)

$$\hat{IPN}_t = 10.8375 + 0.0463 IPD_{t-1} - 0.2614 IPN_{t-1} - 0.0973 IPN_{t-2} + 0.0582 IPB_{t-2} - 0.0745 IPM_{t-2}$$

(2.0993) (0.01699) (0.0368) (0.03796) (0.0322) (0.0433)

```
## Estimation results for equation IPB:
## =====
## IPB = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.07486    0.03016    2.482  0.01325 * /
## IPN.l1  0.06210    0.06528    0.951  0.34172
## IPB.l1  0.13357    0.05789    2.307  0.02128 * /
## IPM.l1 -0.11307    0.07662   -1.476  0.14034
## IPD.l2 -0.15513    0.02958   -5.243  1.98e-07 *** /
## IPN.l2 -0.03545    0.06742   -0.526  0.59914
## IPB.l2  0.15661    0.05721    2.738  0.00631 ** /
## IPM.l2  0.04526    0.07684    0.589  0.55602
## IPD.l3 -0.12628    0.02990   -4.223  2.66e-05 *** /
## IPN.l3  0.07284    0.06541    1.114  0.26577
## IPB.l3  0.28938    0.05812    4.979  7.70e-07 *** /
## IPM.l3 -0.17262    0.07599   -2.272  0.02335 * /
## const  7.90970    3.72655    2.123  0.03407 * /
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Estimation results for equation IPM:
## =====
## IPM = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.125062   0.019407    6.444  1.92e-10 *** ✓
## IPN.l1  0.007548   0.042009    0.180  0.857444
## IPB.l1  0.023126   0.037254    0.621  0.534924
## IPM.l1 -0.024062   0.049302   -0.488  0.625630
## IPD.l2 -0.041265   0.019038   -2.168  0.030459 * ✓
## IPN.l2 -0.063918   0.043386   -1.473  0.141044
## IPB.l2  0.150896   0.036812    4.099  4.53e-05 *** ✓
## IPM.l2 -0.137071   0.049447   -2.772  0.005688 ** ✓
## IPD.l3 -0.074706   0.019242   -3.882  0.000111 *** ✓
## IPN.l3  0.059990   0.042093    1.425  0.154465
## IPB.l3  0.107740   0.037400    2.881  0.004064 ** ✓
## IPM.l3  0.122299   0.048896    2.501  0.012559 * ✓
## const  7.236032    2.397992    3.018  0.002622 ** ✓
## ---
```

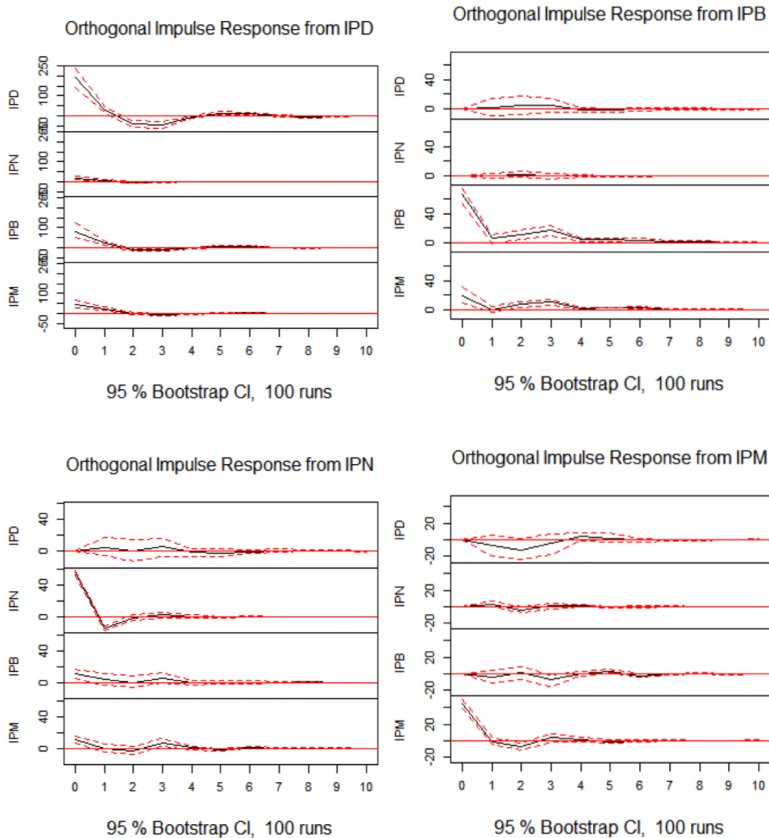
$$\hat{IPB}_t = 7.90970 + 0.07486 IPD_{t-1} + 0.13357 IPB_{t-1} - 0.15513 IPD_{t-2} + 0.15661 IPB_{t-2} - 0.12628 IPD_{t-3} + 0.28938 IPB_{t-3} - 0.17262 IPM_{t-3}$$

(3.72655) (0.03016) (0.05789) (0.02958) (0.05721) (0.02990) (0.05812) (0.07599)

$$\hat{IPM}_t = 7.2360 + 0.125062 IPD_{t-1} - 0.0413 IPD_{t-2} + 0.1509 IPB_{t-2} - 0.1371 IPM_{t-2} - 0.07471 IPD_{t-3} + 0.10774 IPB_{t-3} + 0.1223 IPM_{t-3}$$

(2.39799) (0.019407) (0.0190) (0.0368) (0.04945) (0.01924) (0.0374) (0.0489)

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



If we look at shock from IPD, it affects itself the most and the effect started to die down. Moving on to the effect from IPN shock, it also affects itself the most and then the effect then die down. And this also happens the same to IPB's and IPM's shocks in that they affect themselves the most and the effect from then die down. (approach zero)

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

## \$IPD					## \$IPB				
##	IPD	IPN	IPB	IPM	##	IPD	IPN	IPB	IPM
## [1,]	1.0000000	0.0000000	0.0000000	0.0000000	## [1,]	0.6068593	0.01159095	0.3815498	0.0000000
## [2,]	0.9979670	0.0004250436	8.491884e-05	0.001522999	## [2,]	0.6179030	0.01225658	0.3676075	0.002232922
## [3,]	0.9936178	0.0004100185	7.531241e-04	0.005219014	## [3,]	0.6157601	0.01194222	0.3700258	0.002271899
## [4,]	0.9918724	0.0013662524	1.463226e-03	0.005298160	## [4,]	0.6001250	0.01443847	0.3784401	0.006996476
## [5,]	0.9914744	0.0013822920	1.470936e-03	0.005672384	## [5,]	0.5993374	0.01441286	0.3792660	0.006983788
## [6,]	0.9912531	0.0015051074	1.523454e-03	0.005718372	## [6,]	0.5989509	0.01437936	0.3792067	0.007463045
## [7,]	0.9912299	0.0015270251	1.521433e-03	0.005721625	## [7,]	0.5988685	0.01432999	0.3788978	0.007903719
## [8,]	0.9912048	0.0015371771	1.524872e-03	0.005733157	## [8,]	0.5986028	0.01433067	0.3791060	0.007960548
## [9,]	0.9911955	0.0015443110	1.526934e-03	0.005733294	## [9,]	0.5984452	0.01438173	0.3792127	0.007960383
## [10,]	0.9911963	0.0015441492	1.527079e-03	0.005732502	## [10,]	0.5984010	0.01438011	0.3792368	0.007982087
## [11,]	0.9911939	0.0015446167	1.527208e-03	0.005734310	## [11,]	0.5983741	0.01437939	0.3792629	0.007983564
## [12,]	0.9911928	0.0015449778	1.527563e-03	0.005734640	## [12,]	0.5983702	0.01438034	0.3792659	0.007983608
## [13,]	0.9911926	0.0015450057	1.527571e-03	0.005734794	## [13,]	0.5983678	0.01438015	0.3792674	0.007984677
## [14,]	0.9911923	0.0015450843	1.527630e-03	0.005734969	## [14,]	0.5983635	0.01438017	0.3792706	0.007985700
## [15,]	0.9911923	0.0015451324	1.527641e-03	0.005734948	## [15,]	0.5983618	0.01438047	0.3792720	0.007985718
## [16,]	0.9911923	0.0015451331	1.527642e-03	0.005734957	## [16,]	0.5983613	0.01438048	0.3792725	0.007985718
## [17,]	0.9911923	0.0015451422	1.527644e-03	0.005734960	## [17,]	0.5983611	0.01438047	0.3792727	0.007985741
## [18,]	0.9911923	0.0015451430	1.527645e-03	0.005734959	## [18,]	0.5983610	0.01438047	0.3792728	0.007985749
## [19,]	0.9911923	0.0015451431	1.527645e-03	0.005734960	## [19,]	0.5983609	0.01438047	0.3792729	0.007985751
## [20,]	0.9911923	0.0015451435	1.527645e-03	0.005734961	## [20,]	0.5983609	0.01438047	0.3792729	0.007985756

## \$IPN					## \$IPM				
##	IPD	IPN	IPB	IPM	##	IPD	IPN	IPB	IPM
## [1,]	0.1063210	0.8936790	0.000000e+00	0.000000000	## [1,]	0.4527700	0.02703994	0.07800096	0.4421891
## [2,]	0.1058671	0.8923855	8.840260e-06	0.001738537	## [2,]	0.5174437	0.02385445	0.06892657	0.3897753
## [3,]	0.1071358	0.8845990	1.580340e-03	0.006684881	## [3,]	0.5060686	0.02493902	0.07814715	0.3908452
## [4,]	0.1081787	0.8834937	1.584939e-03	0.006742582	## [4,]	0.4921051	0.03337200	0.09638066	0.3781422
## [5,]	0.1083886	0.8828072	1.602995e-03	0.007201205	## [5,]	0.4922373	0.03336890	0.09683090	0.3755629
## [6,]	0.1085024	0.8824725	1.605243e-03	0.007419876	## [6,]	0.4915248	0.03347238	0.09808955	0.3769132
## [7,]	0.1086375	0.8823305	1.613333e-03	0.007418696	## [7,]	0.4916632	0.03348738	0.09938660	0.3754629
## [8,]	0.1086437	0.8822965	1.617338e-03	0.007442509	## [8,]	0.4916658	0.03349706	0.09954050	0.3752966
## [9,]	0.1086584	0.8822685	1.617297e-03	0.007455751	## [9,]	0.4915514	0.03348765	0.09974164	0.3752194
## [10,]	0.1086686	0.8822539	1.620517e-03	0.007456980	## [10,]	0.4914662	0.03348856	0.09989794	0.3751473
## [11,]	0.1086683	0.8822516	1.621029e-03	0.007459121	## [11,]	0.4914426	0.03349083	0.09993459	0.3751319
## [12,]	0.1086698	0.8822499	1.621125e-03	0.007459106	## [12,]	0.4914403	0.03348978	0.09995000	0.3751199
## [13,]	0.1086704	0.8822490	1.621290e-03	0.007459366	## [13,]	0.4914368	0.03348892	0.09996031	0.3751140
## [14,]	0.1086704	0.8822489	1.621391e-03	0.007459368	## [14,]	0.4914334	0.03348922	0.09996604	0.3751113
## [15,]	0.1086705	0.8822487	1.621392e-03	0.007459370	## [15,]	0.4914322	0.03348934	0.09996835	0.3751101
## [16,]	0.1086705	0.8822487	1.621403e-03	0.007459382	## [16,]	0.4914316	0.03348929	0.09996947	0.3751097
## [17,]	0.1086705	0.8822487	1.621409e-03	0.007459382	## [17,]	0.4914312	0.03348928	0.09997009	0.3751094
## [18,]	0.1086706	0.8822487	1.621409e-03	0.007459384	## [18,]	0.4914312	0.03348928	0.09997029	0.3751093
## [19,]	0.1086706	0.8822487	1.621410e-03	0.007459385	## [19,]	0.4914311	0.03348927	0.09997039	0.3751092
## [20,]	0.1086706	0.8822486	1.621411e-03	0.007459386	## [20,]	0.4914311	0.03348927	0.09997046	0.3751092

$\begin{bmatrix} IPD \\ IPN \\ IPB \\ IPM \end{bmatrix}$

meaning IPM, IPB and IPN cannot have effect on IPD.

IPB and IPM cannot affect IPN but IPD can.

IPM cannot affect IPB but IPD and IPN can

So IPD can affect other's shock but IPM cannot affect any variable's shock.

1.5 Compute 1-step to 6-step ahead predictions of  $z_t$  at the forecast origin  $h = 879$  (April 2020).

Obtain 95 % interval forecasts for each component series.

## \$IPD					## \$IPB				
##	fcst	lower	upper	CI	##	fcst	lower	upper	CI
## [1,]	28.43139444	-347.9652	404.8280	376.3966	## [1,]	112.208449	-95.07482	319.4917	207.2833
## [2,]	6.43027854	-375.0385	387.8991	381.4688	## [2,]	52.013612	-160.23137	264.2586	212.2450
## [3,]	66.47727431	-323.2906	456.2452	389.7679	## [3,]	35.928136	-179.11104	250.9673	215.0392
## [4,]	16.65753976	-383.2485	416.5636	399.9061	## [4,]	56.117462	-163.87413	276.1091	219.9916
## [5,]	-7.52832606	-407.6725	392.6158	400.1441	## [5,]	18.543060	-201.64964	238.7358	220.1927
## [6,]	0.02109574	-401.0963	401.1385	401.1174	## [6,]	8.129404	-212.52746	228.7863	220.6569

## \$IPN					## \$IPM				
##	fcst	lower	upper	CI	##	fcst	lower	upper	CI
## [1,]	-39.192404	-155.96167	77.57686	116.7693	## [1,]	-50.89055	-184.27506	82.49396	133.3845
## [2,]	6.141801	-114.42467	126.70827	120.5665	## [2,]	49.55739	-92.55370	191.66848	142.1111
## [3,]	27.675478	-93.47669	148.82765	121.1522	## [3,]	64.29124	-79.42182	208.00430	143.7131
## [4,]	6.739005	-114.58541	128.06342	121.3244	## [4,]	16.83084	-129.75184	163.41352	146.5827
## [5,]	3.738534	-117.63437	125.11143	121.3729	## [5,]	13.91250	-132.83552	160.66051	146.7480
## [6,]	11.920676	-109.48308	133.32443	121.4038	## [6,]	19.88719	-127.03735	166.81174	146.9245

## Question 2

6104640336 Achariyaporn T.

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

The data are available from Yahoo Finance.

Arch test

$H_0$ : there's no arch effect

$H_1$ : there's arch effect

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

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

So from the print out, I started from transforming price of Dogecoin into return using  $\text{diff}(\log(\text{as.numeric}(\text{adjusted close price of Dogecoin})))$ .

Then I started to do the mean equation by using `auto.arima` and I obtain ARMA(3,3) for mean equation as suggested by the program. And when I check the residual squared (check for arch effect) by using Ljung-Box test, we obtain the result in that p-value is less than 0.05 (alpha at 95% confidence level).

Then I propose the volatility equation which are Garch(1,1) and ARCH(1,0) for normal distribution and Student-t distribution according to model m3 to m6 from the print out. And according to BIC criteria, model m5 has the lowest BIC value, hence I will choose m5 to represent the

forecasting of Dogecoin and model m5 is ARMA(3,3) + Garch(1,1) with student-t

$$r_t = -0.0014 - 0.5060 r_{t-1} - 0.5331 a_{t-1} + 0.5749 a_{t-2} \rightarrow \text{mean eq.}$$

(0.0006) (0.1502) (0.1190) (0.1390)

$$\sigma_t^2 = 0.000177 + 0.6692 \sigma_{t-1}^2 + 0.71103 a_{t-1}^2 \quad \text{with } a_t \sim t_{2.393} \rightarrow \text{volatility eq.}$$

(0.00006) (0.2075) (0.02939)

```
## Coefficient(s):
##      mu      ar1      ar2      ar3      ma1
ma2
## -0.00142845  0.38830308 -0.50602012  0.10205168 -0.53308519
0.57494661
##      ma3      omega      alpha1      beta1      shape
## -0.18459409  0.00017662  0.66916538  0.711102963  2.39292439
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu -0.0014284  0.0006017 -2.374 0.017586 *
## ar1  0.3883031  0.2913734  1.333 0.182642
## ar2 -0.5060201  0.1501570 -3.370 0.000752 ***
## ar3  0.1020517  0.1807280  0.565 0.572298
## ma1 -0.5330852  0.2890136 -1.844 0.065111 .
## ma2  0.5749466  0.1390096  4.136 3.53e-05 ***
## ma3 -0.1845941  0.1827987 -1.010 0.312581
## omega 0.0001766  0.0000624  2.830 0.004651 **
## alpha1 0.6691654  0.2075135  3.225 0.001261 **
## beta1  0.7111029  0.0293852 24.197 < 2e-16 ***
## shape 2.3929244  0.1501627 15.936 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## Standardised Residuals Tests:
##      Jarque-Bera Test R Chi^2 11703.26 0
##      Shapiro-Wilk Test R W 0.8658672 0
##      Ljung-Box Test R Q(10) 56.74168 1.49041e-08
##      Ljung-Box Test R Q(15) 81.0624 4.463296e-11 } > 0.05
##      Ljung-Box Test R Q(20) 85.27692 4.90407e-10
##      Ljung-Box Test R^2 Q(10) 11.10941 0.3490581
##      Ljung-Box Test R^2 Q(15) 11.69615 0.7018674 } > 0.05
##      Ljung-Box Test R^2 Q(20) 14.32062 0.8138748
##      LM Arch Test R TR^2 11.58158 0.4798406
##
## Information Criterion Statistics:
##      AIC BIC SIC HQIC
## -3.516393 -3.489211 -3.516437 -3.506489
```

And for mean & volatility equation are adequate as Ljung-Box for R & R<sup>2</sup> is more than alpha = 0.05. And there's no arch effect leftover from LM arch test as we get 0.4798 (more than 0.05).

# Question 3

6104640336 Achariyaporn T.

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

```
## Estimation results for equation UKGDP:
## =====
## UKGDP = UKGDP.l1 + CANADAGDP.l1 + USGDP.l1 + UKGDP.l2 + CANADAGDP.l2 +
USGDP.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## UKGDP.l1      0.13130    0.09349   1.404 0.16282
## CANADAGDP.l1 -0.31837    0.14219  -2.239 0.02704 *
## USGDP.l1       0.40317    0.13747   2.933 0.00404 **
## UKGDP.l2       0.27087    0.09313   2.908 0.00435 **
## CANADAGDP.l2  0.12100    0.13823   0.875 0.38316
## USGDP.l2       0.00351    0.14283   0.025 0.98044
## const         0.44760    0.19167   2.335 0.02123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$\hat{UKGDP}_t = 0.4476 UKGDP_{t-1} - 0.31837 CANADAGDP_{t-1} + 0.40317 USGDP_{t-1} + 0.2709 UKGDP_{t-2}$$

(0.19167) (0.14219) (0.13747) (0.09313)

```
## Estimation results for equation CANADAGDP:
## =====
## CANADAGDP = UKGDP.l1 + CANADAGDP.l1 + USGDP.l1 + UKGDP.l2 + CANADAGDP.l2 +
USGDP.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## UKGDP.l1       0.11947    0.06266   1.907 0.059002 .
## CANADAGDP.l1  0.47321    0.09530   4.966 2.35e-06 ***
## USGDP.l1       0.31920    0.09213   3.465 0.000743 ***
## UKGDP.l2      -0.02657    0.06242  -0.426 0.671114
## CANADAGDP.l2  0.02889    0.09264   0.312 0.755747
## USGDP.l2      -0.27868    0.09573  -2.911 0.004313 **
## const         0.11003    0.12846   0.857 0.393463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$\hat{CANADAGDP}_t = 0.11947 UKGDP_{t-1} + 0.4732 CANADAGDP_{t-1} + 0.3192 USGDP_{t-1} - 0.2787 USGDP_{t-2}$$

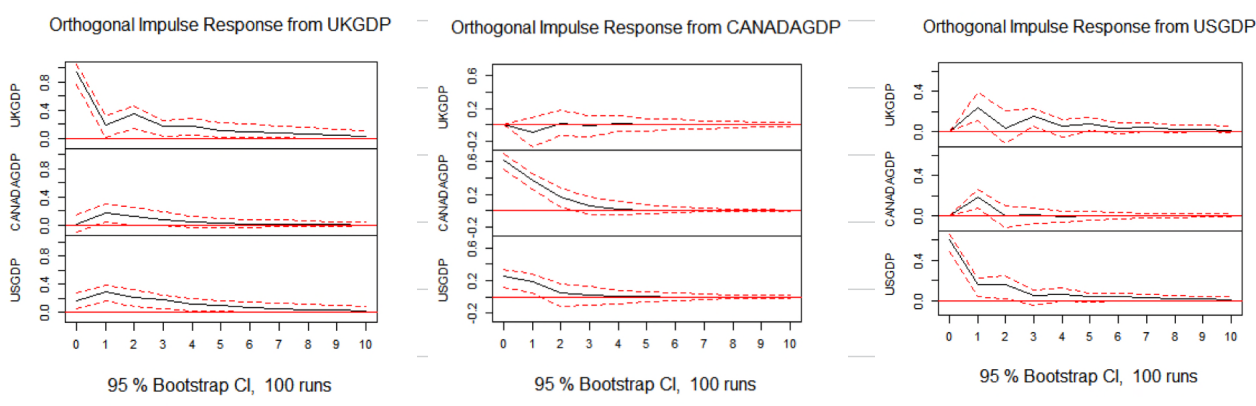
(0.06266) (0.09530) (0.09213) (0.09573)

```
## Estimation results for equation USGDP:
## =====
## USGDP = UKGDP.l1 + CANADAGDP.l1 + USGDP.l1 + UKGDP.l2 + CANADAGDP.l2 +
USGDP.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## UKGDP.l1       0.26552    0.06779   3.917 0.000151 ***
## CANADAGDP.l1  0.18335    0.10310   1.778 0.077936 .
## USGDP.l1       0.26106    0.09968   2.619 0.009986 **
## UKGDP.l2       0.05442    0.06753   0.806 0.421998
## CANADAGDP.l2 -0.06486    0.10023  -0.647 0.518799
## USGDP.l2       0.02657    0.10357   0.257 0.797946
## const         0.42320    0.13898   3.045 0.002874 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

$$\hat{USGDP}_t = 0.42320 USGDP_{t-1} + 0.26552 UKGDP_{t-1} + 0.18335 CANADAGDP_{t-1} + 0.26106 USGDP_{t-1}$$

(0.13898) (0.06779) (0.10310) (0.09968)

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



There 3 impulse response function are almost the same in that the shock from itself will have the most effect on its own variable the most and then the effect from shock will die down.

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

```
fevd(varfit,n.ahead = 20)
## $UKGDP
## UKGDP CANADAGDP USGDP
## [1,] 1.0000000 0.000000000 0.00000000
## [2,] 0.9297076 0.009441237 0.06085112
## [3,] 0.9359223 0.008671549 0.05540613
## [4,] 0.9171259 0.008354668 0.07451939
## [5,] 0.9171105 0.008295939 0.07459352
## [6,] 0.9133324 0.008173334 0.07849425
## [7,] 0.9128934 0.008134848 0.07897178
## [8,] 0.9120215 0.008097890 0.07988060
## [9,] 0.9117790 0.008081759 0.08013920
## [10,] 0.9115388 0.008070197 0.08039100
## [11,] 0.9114390 0.008064331 0.08049666
## [12,] 0.9113653 0.008060576 0.08057408
## [13,] 0.9113285 0.008058521 0.08061296
## [14,] 0.9113046 0.008057269 0.08063809
## [15,] 0.9112916 0.008056559 0.08065180
## [16,] 0.9112837 0.008056137 0.08066018
## [17,] 0.9112792 0.008055894 0.08066492
## [18,] 0.9112765 0.008055750 0.08066775
## [19,] 0.9112750 0.008055667 0.08066938
## [20,] 0.9112740 0.008055618 0.08067034
##
## $CANADAGDP
## UKGDP CANADAGDP USGDP
## [1,] 0.0003373506 0.9996626 0.00000000
## [2,] 0.0484624625 0.8896079 0.06192963
## [3,] 0.0694636089 0.8729466 0.05758981
## [4,] 0.0795270108 0.8637328 0.05674023
## [5,] 0.0832090324 0.8603302 0.05646073
## [6,] 0.0847049460 0.8588925 0.05640257
## [7,] 0.0853874952 0.8582191 0.05639341
## [8,] 0.0857234666 0.8578577 0.05641885
## [9,] 0.0859041554 0.8576590 0.05643684
## [10,] 0.0860025114 0.8575461 0.05645139
## [11,] 0.0860581801 0.8574817 0.05646014
## [12,] 0.0860896845 0.8574445 0.05646579
## [13,] 0.0861078461 0.8574230 0.05646912
## [14,] 0.0861183097 0.8574105 0.05647115
## [15,] 0.0861243897 0.8574033 0.05647233
## [16,] 0.0861279206 0.8573990 0.05647304
## [17,] 0.0861299788 0.8573966 0.05647345
## [18,] 0.0861311780 0.8573951 0.05647369
## [19,] 0.0861318779 0.8573943 0.05647383
## [20,] 0.0861322862 0.8573938 0.05647392
##
## $USGDP
## UKGDP CANADAGDP USGDP
## [1,] 0.06063856 0.1430828 0.7962787
## [2,] 0.18918578 0.1636079 0.6472063
## [3,] 0.23517742 0.1508355 0.6139871
## [4,] 0.26741869 0.1443755 0.5882058
## [5,] 0.28063551 0.1407135 0.5786510
## [6,] 0.28887541 0.1387469 0.5723777
## [7,] 0.29296398 0.1376151 0.5694209
## [8,] 0.29547802 0.1369707 0.5675512
## [9,] 0.29685250 0.1365947 0.5665528
## [10,] 0.29767814 0.1363771 0.5659447
## [11,] 0.29814676 0.1362500 0.5656032
## [12,] 0.29842405 0.1361761 0.5653999
## [13,] 0.29858396 0.1361329 0.5652832
## [14,] 0.29867792 0.1361077 0.5652144
## [15,] 0.29873248 0.1360930 0.5651746
## [16,] 0.29876444 0.1360844 0.5651512
## [17,] 0.29878305 0.1360793 0.5651376
## [18,] 0.29879394 0.1360764 0.5651296
## [19,] 0.29880029 0.1360747 0.5651250
## [20,] 0.29880400 0.1360737 0.5651223
##
```

$\begin{bmatrix} UK \\ Canada \\ US \end{bmatrix}$  → UK cannot be affected by Canada or US GDP → (100% from itself)  
 → Canada's GDP can be affected by UK's but not US. → (99.97 from itself  
 → US's GDP can be affect by both UK and Canada. the rest is from UK's

↳ About 60% of shock comes from itself, 14% are from Canada & 6% is from UK.

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

H0: there's unit root → I(1) → not cointegrate

H1: there's no unit root → I(0) → cointegrate

```
fit <- lm(UK-CANADA+US)
summary(fit)
##
## Call:
## lm(formula = UK ~ CANADA + US)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11745.9  -6772.7   -669.4   6486.1  12835.5
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.387e+04  1.326e+04  1.046  0.298
## CANADA      -1.324e-07  8.545e-08  -1.550  0.124
## US          3.013e+01  1.659e+00  18.165 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7111 on 124 degrees of freedom
## Multiple R-squared:  0.9958, Adjusted R-squared:  0.9957
## F-statistic: 1.461e+04 on 2 and 124 DF,  p-value: < 2.2e-16

fit2 <- lm(CANADA~UK+US)
summary(fit2)
##
## Call:
## lm(formula = CANADA ~ UK + US)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.317e+10 -5.414e+09 -9.450e+08  5.901e+09  1.584e+10
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.533e+11  1.670e+09  91.771 < 2e-16 ***
## UK          -1.435e+05  9.258e+04  -1.550  0.124
## US          2.328e+07  2.558e+06  9.098  1.9e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.402e+09 on 124 degrees of freedom
## Multiple R-squared:  0.9907, Adjusted R-squared:  0.9906
## F-statistic: 6623 on 2 and 124 DF,  p-value: < 2.2e-16

fit3 <- lm(US~UK+CANADA)
summary(fit3)
##
## Call:
## lm(formula = US ~ UK + CANADA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -379.53  -142.12   -50.88   176.28   403.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.473e+03  3.046e+02  -8.118 3.98e-13 ***
## UK          2.412e-02  1.328e-03  18.165 < 2e-16 ***
## CANADA      1.720e-08  1.890e-09  9.098 1.90e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.2 on 124 degrees of freedom
## Multiple R-squared:  0.9974, Adjusted R-squared:  0.9974
## F-statistic: 2.395e+04 on 2 and 124 DF,  p-value: < 2.2e-16
```

Since the p-value obtain when proposing co-integrating vector is less than 0.05 (alpha at 95% CI), so we do not reject H0 meaning the three variables are co-integrated.

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

```
VECM<-lm(diff.UK~diff.UK.L1+diff.CANADA.L1+diff.US.L1+error.L1)
summary(VECM)

##
## Call:
## lm(formula = diff.UK ~ diff.UK.L1 + diff.CANADA.L1 + diff.US.L1 +
##     error.L1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6024.8 -1070.3  168.1  1310.2  4773.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.134e+03  3.423e+02  3.312 0.00123 **
## diff.UK.L1   -4.232e-02  9.151e-02  -0.462 0.64461
## diff.CANADA.L1 -1.802e-07  9.775e-08  -1.844 0.06768 .
## diff.US.L1    2.116e+01  3.539e+00  5.980 2.36e-08 ***
## error.L1     -1.726e-02  2.690e-02  -0.642 0.52233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2061 on 120 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.2744, Adjusted R-squared:  0.2503
## F-statistic: 11.35 on 4 and 120 DF, p-value: 7.632e-08
```

$$\Delta UKGDP_t = 1134e+03 - 4.232e-02 \Delta UK_{t-1} - 1.802e-07 \Delta Canada_{t-1} + 2.116e+01 \Delta US_{t-1} - 1.726e-02 \hat{\epsilon}_t$$

(3.423e+02)      (9.151e-02)      (9.775e-08)      (3.539e+00)      (2.690e-02)

$$- 1.802e-07 \Delta Canada_{t-1} + 2.116e+01 \Delta US_{t-1}$$

(9.775e-08)      (3.539e+00)

$$- 1.726e-02 \hat{\epsilon}_t$$

(2.690e-02)

→ To adjust to equilibrium of UK's GDP, UK and Canada must have a negative change while U.S. need to have positive change.

```
VECM2<-lm(diff.CANADA~diff.UK.L1+diff.CANADA.L1+diff.US.L1+error2.L1)
summary(VECM2)

##
## Call:
## lm(formula = diff.CANADA ~ diff.UK.L1 + diff.CANADA.L1 + diff.US.L1 +
##     error2.L1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.364e+09 -1.278e+09 -1.858e+08  1.301e+09  5.505e+09
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.780e+08  3.139e+08  -0.886 0.377585
## diff.UK.L1   9.858e+04  8.273e+04  1.192 0.235756
## diff.CANADA.L1 3.338e-01  8.864e-02  3.766 0.000259 ***
## diff.US.L1    1.248e+07  3.200e+06  3.900 0.000159 ***
## error2.L1     -5.261e-02  2.365e-02  -2.225 0.027984 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.864e+09 on 120 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.4371, Adjusted R-squared:  0.4183
## F-statistic: 23.3 on 4 and 120 DF, p-value: 2.889e-14
```

$$\Delta CanadaGDP_t = -2.780e+08 + 9.858e+04 \Delta UK_{t-1} + 3.338e-01 \Delta Canada_{t-1} + 1.248e+07 \Delta US_{t-1} - 5.261e-02 \hat{\epsilon}_t$$

(3.3139e+08)      (8.273e+04)      (8.864e-02)      (3.200e+06)      (2.365e-02)

$$+ 3.338e-01 \Delta Canada_{t-1} + 1.248e+07 \Delta US_{t-1}$$

(8.864e-02)      (3.200e+06)

$$- 5.261e-02 \hat{\epsilon}_t$$

(2.365e-02)

→ For Canada's GDP to be at equilibrium, the GDP for UK, US and Canada itself need to have a positive change.

```
VECM3<-lm(diff.US~diff.UK.L1+diff.CANADA.L1+diff.US.L1+error3.L1)
summary(VECM3)

##
## Call:
## lm(formula = diff.US ~ diff.UK.L1 + diff.CANADA.L1 + diff.US.L1 +
##     error3.L1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -306.873 -21.905 -1.991  32.717  136.470
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.496e+01  9.532e+00  3.668 0.000366 ***
## diff.UK.L1   1.073e-02  2.557e-03  4.196 5.24e-05 ***
## diff.CANADA.L1 2.670e-09  2.693e-09  0.991 0.323595
## diff.US.L1    3.098e-01  9.779e-02  3.168 0.001947 **
## error3.L1     -2.078e-02  2.673e-02  -0.777 0.438401
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.94 on 120 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.3965, Adjusted R-squared:  0.3763
## F-statistic: 19.71 on 4 and 120 DF, p-value: 1.726e-12
```

$$\Delta USGDP_t = 3.496e+01 + 1.073e-02 \Delta UK_{t-1} + 2.670e-09 \Delta Canada_{t-1} + 3.098e-01 \Delta US_{t-1} - 2.078e-02 \hat{\epsilon}_t$$

(9.532e+00)      (2.557e-03)      (2.693e-09)      (9.779e-02)      (2.673e-02)

$$+ 2.670e-09 \Delta Canada_{t-1} + 3.098e-01 \Delta US_{t-1}$$

(2.693e-09)      (9.779e-02)

$$- 2.078e-02 \hat{\epsilon}_t$$

(2.673e-02)

→ For US's GDP to reach equilibrium, the GDP for UK, Canada and US should have a positive change.

## 6104640336.R

10.10

2021-06-02

```
cat(rep("\n",50))
library(quantmod)
## Loading required package: xts
## Loading required package: zoo
## Warning: package 'zoo' was built under R version 4.0.4
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
## Loading required package: TTR
## Registered S3 method overwritten by 'quantmod':
##   method           from
##   as.zoo.data.frame zoo
library(fBasics)
## Loading required package: timeDate
## Loading required package: timeSeries
##
## Attaching package: 'timeSeries'
## The following object is masked from 'package:zoo':
##
##   time<-
##
## Attaching package: 'fBasics'
## The following object is masked from 'package:TTR':
##
##   volatility
library(sn)
```

```
## Warning: package 'sn' was built under R version 4.0.5
## Loading required package: stats4
##
## Attaching package: 'sn'
## The following objects are masked from 'package:fBasics':
##
##   tr, vech
## The following object is masked from 'package:stats':
##
##   sd

library(PerformanceAnalytics)

##
## 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)

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

library(fGarch)

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

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
## Warning: package 'sandwich' was built under R version 4.0.5
## Loading required package: urca
## Loading required package: lmtest

#Question1
getSymbols("IPDCONGD",src = "FRED")

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

## [1] "IPDCONGD"

dim(IPDCONGD)

## [1] 892  1

tail(IPDCONGD)

##           IPDCONGD
## 2020-11-01 104.8970
## 2020-12-01 105.3158
## 2021-01-01 107.6347
## 2021-02-01 100.0964
## 2021-03-01 102.5666
## 2021-04-01 100.7979

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

## [1] "IPNCONGD"

dim(IPNCONGD)

## [1] 892  1

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

## [1] "IPBUSEQ"

```

```

dim(IPBUSEQ)
## [1] 892  1

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

## [1] "IPMAT"

dim(IPMAT)
## [1] 988  1

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

## [1] 892  4

colnames(IP) <- c("IPD","IPN","IPB","IPM")
#1.1
log_IP=log(IP)
Zt=diff(IP)*100
#1.2
m1= VAR(Zt,p=3)
summary(m1)

##
## VAR Estimation Results:
## =====
## Endogenous variables: IPD, IPN, IPB, IPM
## Deterministic variables: const
## Sample size: 888
## Log Likelihood: -20358.954
## Roots of the characteristic polynomial:
## 0.6866 0.6521 0.6521 0.6283 0.6283 0.5705 0.5705 0.5148 0.4082 0.3261
## 0.3261 0.03523
## Call:
## VAR(y = Zt, p = 3)
##
##
## Estimation results for equation IPD:
## =====
## IPD = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.15653    0.05476   2.858  0.00436 **
## IPN.l1  0.08917    0.11855   0.752  0.45213
## IPB.l1  0.07629    0.10513   0.726  0.46823
## IPM.l1 -0.16784    0.13912  -1.206  0.22799
## IPD.l2 -0.21923    0.05372  -4.081  4.9e-05 ***
## IPN.l2  0.02163    0.12243   0.177  0.85981

```

```

## IPB.l2  0.14077    0.10388    1.355    0.17572
## IPM.l2 -0.24366    0.13953   -1.746    0.08112 .
## IPD.l3 -0.17652    0.05430   -3.251    0.00119 **
## IPN.l3  0.11960    0.11878    1.007    0.31429
## IPB.l3  0.09700    0.10554    0.919    0.35831
## IPM.l3 -0.08901    0.13798   -0.645    0.51903
## const  12.15593    6.76687    1.796    0.07278 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 192 on 875 degrees of freedom
## Multiple R-Squared:  0.1226, Adjusted R-squared:  0.1106
## F-statistic: 10.19 on 12 and 875 DF,  p-value: < 2.2e-16
##
##
## Estimation results for equation IPN:
## =====
## IPN = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.046301  0.016989  2.725  0.00655 **
## IPN.l1 -0.261433  0.036777 -7.109 2.43e-12 ***
## IPB.l1 -0.019290  0.032613 -0.591  0.55436
## IPM.l1  0.056677  0.043161  1.313  0.18947
## IPD.l2 -0.018761  0.016666 -1.126  0.26061
## IPN.l2 -0.097276  0.037982 -2.561  0.01060 *
## IPB.l2  0.058199  0.032226  1.806  0.07127 .
## IPM.l2 -0.074456  0.043287 -1.720  0.08578 .
## IPD.l3 -0.002635  0.016845 -0.156  0.87575
## IPN.l3  0.007021  0.036850  0.191  0.84894
## IPB.l3 -0.003110  0.032741 -0.095  0.92435
## IPM.l3  0.015960  0.042805  0.373  0.70936
## const  10.837543  2.099282  5.162 3.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 59.58 on 875 degrees of freedom
## Multiple R-Squared:  0.07441, Adjusted R-squared:  0.06172
## F-statistic: 5.862 on 12 and 875 DF,  p-value: 7.431e-10
##
##
## Estimation results for equation IPB:
## =====
## IPB = IPD.l1 + IPN.l1 + IPB.l1 + IPM.l1 + IPD.l2 + IPN.l2 + IPB.l2 +
IPM.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)

```

```

## IPD.l1  0.07486    0.03016    2.482  0.01325  *
## IPN.l1  0.06210    0.06528    0.951  0.34172
## IPB.l1  0.13357    0.05789    2.307  0.02128  *
## IPM.l1 -0.11307    0.07662   -1.476  0.14034
## IPD.l2 -0.15513    0.02958   -5.243  1.98e-07  ***
## IPN.l2 -0.03545    0.06742   -0.526  0.59914
## IPB.l2  0.15661    0.05721    2.738  0.00631  **
## IPM.l2  0.04526    0.07684    0.589  0.55602
## IPD.l3 -0.12628    0.02990   -4.223  2.66e-05  ***
## IPN.l3  0.07284    0.06541    1.114  0.26577
## IPB.l3  0.28938    0.05812    4.979  7.70e-07  ***
## IPM.l3 -0.17262    0.07599   -2.272  0.02335  *
## const   7.90970    3.72655    2.123  0.03407  *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 105.8 on 875 degrees of freedom
## Multiple R-Squared:  0.1205, Adjusted R-squared:  0.1084
## F-statistic:  9.99 on 12 and 875 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.l2 + IPD.l3 + IPN.l3 + IPB.l3 + IPM.l3 + const
##
##      Estimate Std. Error t value Pr(>|t|)
## IPD.l1  0.125062  0.019407  6.444 1.92e-10 ***
## IPN.l1  0.007548  0.042009  0.180 0.857444
## IPB.l1  0.023126  0.037254  0.621 0.534924
## IPM.l1 -0.024062  0.049302 -0.488 0.625630
## IPD.l2 -0.041265  0.019038 -2.168 0.030459 *
## IPN.l2 -0.063918  0.043386 -1.473 0.141044
## IPB.l2  0.150896  0.036812  4.099 4.53e-05 ***
## IPM.l2 -0.137071  0.049447 -2.772 0.005688 **
## IPD.l3 -0.074706  0.019242 -3.882 0.000111 ***
## IPN.l3  0.059990  0.042093  1.425 0.154465
## IPB.l3  0.107740  0.037400  2.881 0.004064 **
## IPM.l3  0.122299  0.048896  2.501 0.012559 *
## const   7.236032  2.397992  3.018 0.002622 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 68.05 on 875 degrees of freedom
## Multiple R-Squared:  0.1786, Adjusted R-squared:  0.1673
## F-statistic: 15.86 on 12 and 875 DF, p-value: < 2.2e-16
##
##

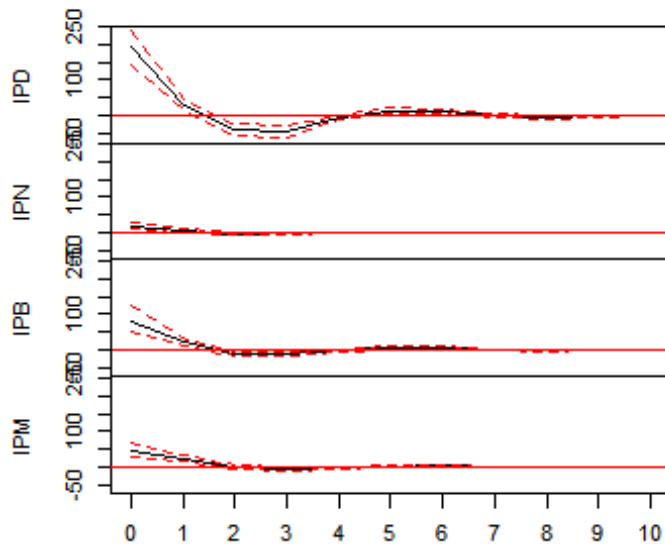
```

```
##
## Covariance matrix of residuals:
##      IPD  IPN  IPB  IPM
## IPD 36880 3731 15822 8794
## IPN  3731 3549  2242 1520
## IPB 15822 2242 11185 5142
## IPM  8794 1520  5142 4631
##
## Correlation matrix of residuals:
##      IPD  IPN  IPB  IPM
## IPD 1.0000 0.3261 0.7790 0.6729
## IPN 0.3261 1.0000 0.3558 0.3749
## IPB 0.7790 0.3558 1.0000 0.7144
## IPM 0.6729 0.3749 0.7144 1.0000
```

### #1.3

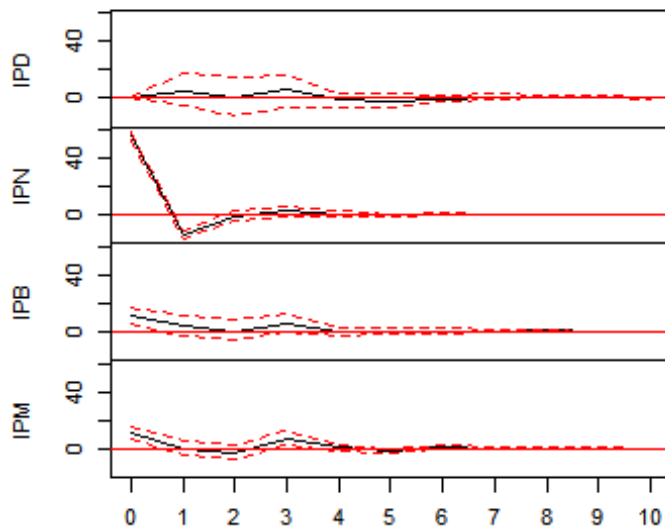
```
m1.impresp = irf(m1)
plot(m1.impresp)
```

### Orthogonal Impulse Response from IPD



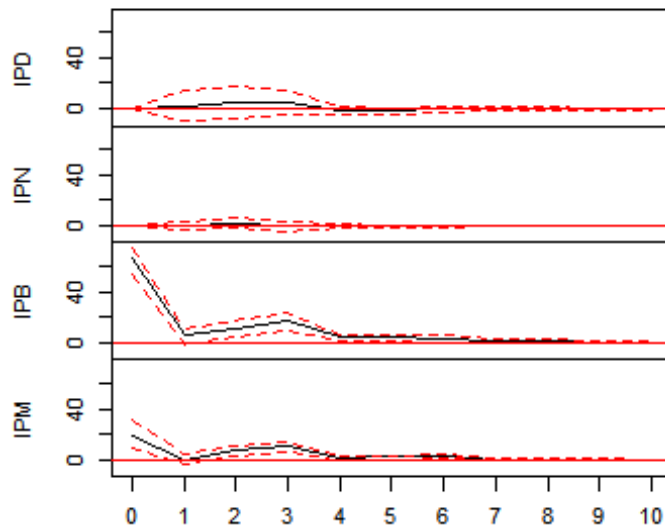
95 % Bootstrap CI, 100 runs

### Orthogonal Impulse Response from IPN



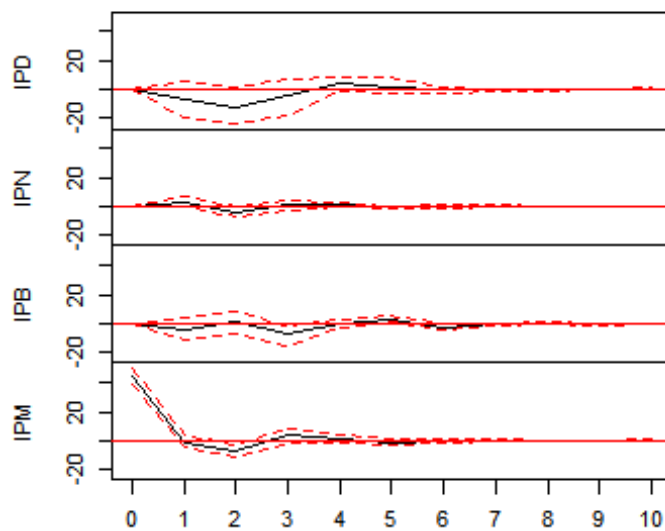
95 % Bootstrap CI, 100 runs

### Orthogonal Impulse Response from IPB



95 % Bootstrap CI, 100 runs

### Orthogonal Impulse Response from IPM



95 % Bootstrap CI, 100 runs

#1.4

```
fevd(m1, n.ahead = 20)
```

```

## $IPD
##          IPD          IPN          IPB          IPM
## [1,] 1.0000000 0.0000000000 0.000000e+00 0.000000000
## [2,] 0.9979670 0.0004250436 8.491884e-05 0.001522999
## [3,] 0.9936178 0.0004100185 7.531241e-04 0.005219014
## [4,] 0.9918724 0.0013662524 1.463226e-03 0.005298160
## [5,] 0.9914744 0.0013822920 1.470936e-03 0.005672384
## [6,] 0.9912531 0.0015051074 1.523454e-03 0.005718372
## [7,] 0.9912299 0.0015270251 1.521433e-03 0.005721625
## [8,] 0.9912048 0.0015371771 1.524872e-03 0.005733157
## [9,] 0.9911955 0.0015443110 1.526934e-03 0.005733294
## [10,] 0.9911963 0.0015441492 1.527079e-03 0.005732502
## [11,] 0.9911939 0.0015446167 1.527208e-03 0.005734310
## [12,] 0.9911928 0.0015449778 1.527563e-03 0.005734640
## [13,] 0.9911926 0.0015450057 1.527571e-03 0.005734794
## [14,] 0.9911923 0.0015450843 1.527630e-03 0.005734969
## [15,] 0.9911923 0.0015451324 1.527641e-03 0.005734948
## [16,] 0.9911923 0.0015451331 1.527642e-03 0.005734957
## [17,] 0.9911923 0.0015451422 1.527644e-03 0.005734960
## [18,] 0.9911923 0.0015451430 1.527645e-03 0.005734959
## [19,] 0.9911923 0.0015451431 1.527645e-03 0.005734960
## [20,] 0.9911923 0.0015451435 1.527645e-03 0.005734961

```

```

##
## $IPN
##          IPD          IPN          IPB          IPM
## [1,] 0.1063210 0.8936790 0.000000e+00 0.000000000
## [2,] 0.1058671 0.8923855 8.840260e-06 0.001738537
## [3,] 0.1071358 0.8845990 1.580340e-03 0.006684881
## [4,] 0.1081787 0.8834937 1.584939e-03 0.006742582
## [5,] 0.1083886 0.8828072 1.602995e-03 0.007201205
## [6,] 0.1085024 0.8824725 1.605243e-03 0.007419876
## [7,] 0.1086375 0.8823305 1.613333e-03 0.007418696
## [8,] 0.1086437 0.8822965 1.617338e-03 0.007442509
## [9,] 0.1086584 0.8822685 1.617297e-03 0.007455751
## [10,] 0.1086686 0.8822539 1.620517e-03 0.007456980
## [11,] 0.1086683 0.8822516 1.621029e-03 0.007459121
## [12,] 0.1086698 0.8822499 1.621125e-03 0.007459106
## [13,] 0.1086704 0.8822490 1.621290e-03 0.007459366
## [14,] 0.1086704 0.8822489 1.621391e-03 0.007459368
## [15,] 0.1086705 0.8822487 1.621392e-03 0.007459370
## [16,] 0.1086705 0.8822487 1.621403e-03 0.007459382
## [17,] 0.1086705 0.8822487 1.621409e-03 0.007459382
## [18,] 0.1086706 0.8822487 1.621409e-03 0.007459384
## [19,] 0.1086706 0.8822487 1.621410e-03 0.007459385
## [20,] 0.1086706 0.8822486 1.621411e-03 0.007459386

```

```

##
## $IPB
##          IPD          IPN          IPB          IPM
## [1,] 0.6068593 0.01159095 0.3815498 0.000000000
## [2,] 0.6179030 0.01225658 0.3676075 0.002232922

```

```

## [3,] 0.6157601 0.01194222 0.3700258 0.002271899
## [4,] 0.6001250 0.01443847 0.3784401 0.006996476
## [5,] 0.5993374 0.01441286 0.3792660 0.006983788
## [6,] 0.5989509 0.01437936 0.3792067 0.007463045
## [7,] 0.5988685 0.01432999 0.3788978 0.007903719
## [8,] 0.5986028 0.01433067 0.3791060 0.007960548
## [9,] 0.5984452 0.01438173 0.3792127 0.007960383
## [10,] 0.5984010 0.01438011 0.3792368 0.007982087
## [11,] 0.5983741 0.01437939 0.3792629 0.007983564
## [12,] 0.5983702 0.01438034 0.3792659 0.007983608
## [13,] 0.5983678 0.01438015 0.3792674 0.007984677
## [14,] 0.5983635 0.01438017 0.3792706 0.007985700
## [15,] 0.5983618 0.01438047 0.3792720 0.007985718
## [16,] 0.5983613 0.01438048 0.3792725 0.007985718
## [17,] 0.5983611 0.01438047 0.3792727 0.007985741
## [18,] 0.5983610 0.01438047 0.3792728 0.007985749
## [19,] 0.5983609 0.01438047 0.3792729 0.007985751
## [20,] 0.5983609 0.01438047 0.3792729 0.007985756

```

```
##
```

```
## $IPM
```

```

##          IPD          IPN          IPB          IPM
## [1,] 0.4527700 0.02703994 0.07800096 0.4421891
## [2,] 0.5174437 0.02385445 0.06892657 0.3897753
## [3,] 0.5060686 0.02493902 0.07814715 0.3908452
## [4,] 0.4921051 0.03337200 0.09638066 0.3781422
## [5,] 0.4922373 0.03336890 0.09683090 0.3775629
## [6,] 0.4915248 0.03347238 0.09808955 0.3769132
## [7,] 0.4916632 0.03348738 0.09938660 0.3754629
## [8,] 0.4916658 0.03349706 0.09954050 0.3752966
## [9,] 0.4915514 0.03348765 0.09974164 0.3752194
## [10,] 0.4914662 0.03348856 0.09989794 0.3751473
## [11,] 0.4914426 0.03349083 0.09993459 0.3751319
## [12,] 0.4914403 0.03348978 0.09995000 0.3751199
## [13,] 0.4914368 0.03348892 0.09996031 0.3751140
## [14,] 0.4914334 0.03348922 0.09996604 0.3751113
## [15,] 0.4914322 0.03348934 0.09996835 0.3751101
## [16,] 0.4914316 0.03348929 0.09996947 0.3751097
## [17,] 0.4914312 0.03348928 0.09997009 0.3751094
## [18,] 0.4914312 0.03348928 0.09997029 0.3751093
## [19,] 0.4914311 0.03348927 0.09997039 0.3751092
## [20,] 0.4914311 0.03348927 0.09997046 0.3751092

```

```
#1.5
```

```
predict(m1, n.ahead = 6, ci =0.95)
```

```
## $IPD
```

```

##          fcst          lower          upper          CI
## [1,] 28.43139444 -347.9652 404.8280 376.3966
## [2,] 6.43027854 -375.0385 387.8991 381.4688
## [3,] 66.47727431 -323.2906 456.2452 389.7679

```

```

## [4,] 16.65753976 -383.2485 416.5636 399.9061
## [5,] -7.52832606 -407.6725 392.6158 400.1441
## [6,] 0.02109574 -401.0963 401.1385 401.1174
##
## $IPN
##      fcst      lower      upper      CI
## [1,] -39.192404 -155.96167  77.57686 116.7693
## [2,]  6.141801 -114.42467 126.70827 120.5665
## [3,] 27.675478  -93.47669 148.82765 121.1522
## [4,]  6.739005 -114.58541 128.06342 121.3244
## [5,]  3.738534 -117.63437 125.11143 121.3729
## [6,] 11.920676 -109.48308 133.32443 121.4038
##
## $IPB
##      fcst      lower      upper      CI
## [1,] 112.208449  -95.07482 319.4917 207.2833
## [2,]  52.013612 -160.23137 264.2586 212.2450
## [3,]  35.928136 -179.11104 250.9673 215.0392
## [4,]  56.117462 -163.87413 276.1091 219.9916
## [5,]  18.543060 -201.64964 238.7358 220.1927
## [6,]   8.129404 -212.52746 228.7863 220.6569
##
## $IPM
##      fcst      lower      upper      CI
## [1,] -50.89055 -184.27506  82.49396 133.3845
## [2,]  49.55739  -92.55370 191.66848 142.1111
## [3,]  64.29124  -79.42182 208.00430 143.7131
## [4,]  16.83084 -129.75184 163.41352 146.5827
## [5,]  13.91250 -132.83552 160.66051 146.7480
## [6,]  19.88719 -127.03735 166.81174 146.9245

```

### *Question2*

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

```
## 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.
```

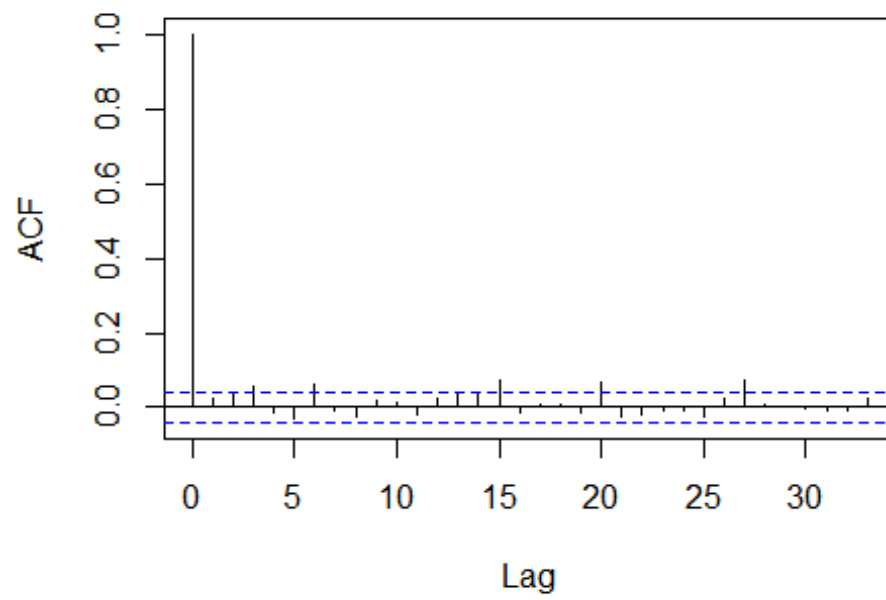
```
## [1] "DOGE-USD"
```

```
rate=diff(log(as.numeric(`DOGE-USD`[,6])))
rate=na.omit(rate)
```

### *#2.1*

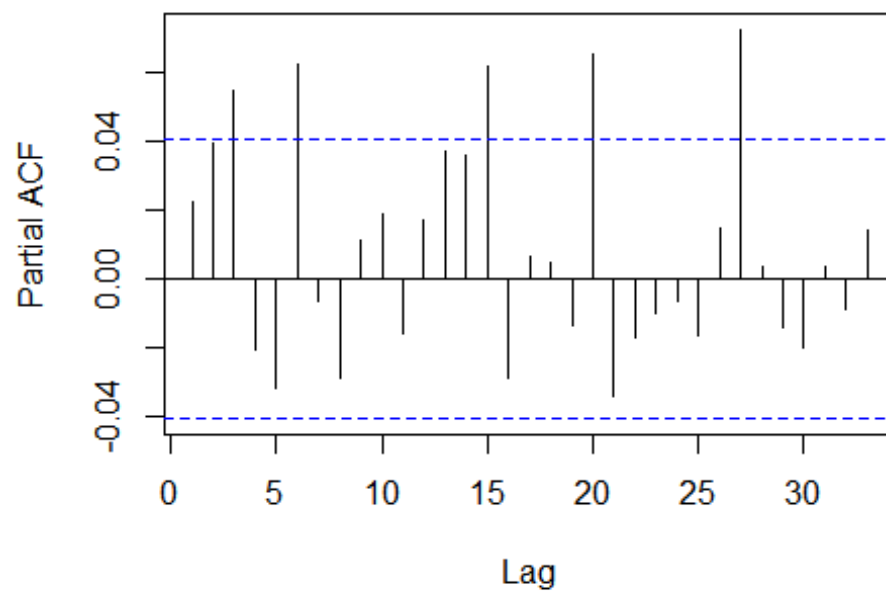
```
acf(rate)
```

### Series rate



```
pacf(rate)
```

### Series rate



```

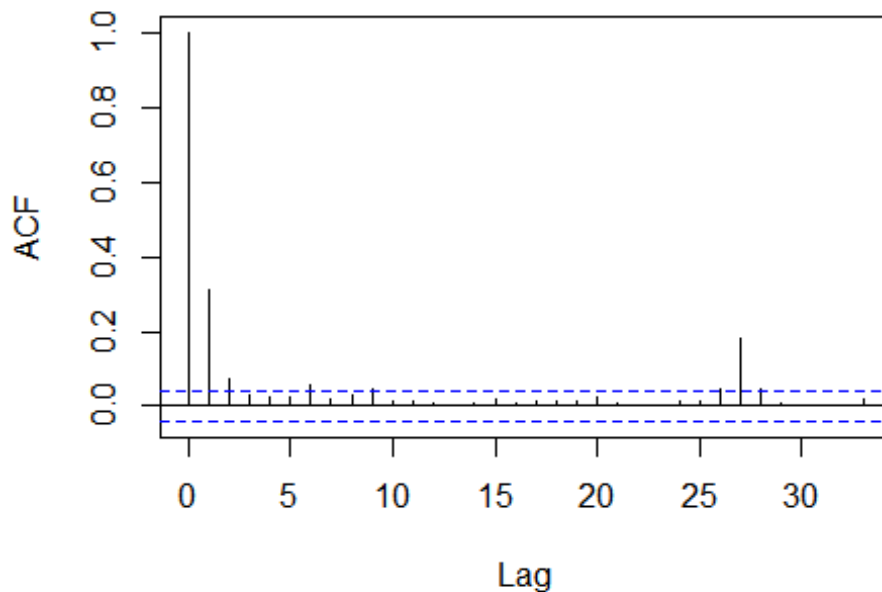
m2=auto.arima(rate)
summary(m2)

## Series: rate
## ARIMA(3,0,3) with non-zero mean
##
## Coefficients:
##          ar1      ar2      ar3      ma1      ma2      ma3      mean
##      -0.1365  -0.6177  0.2899  0.1575  0.6680  -0.2124  0.0032
## s.e.   0.2863   0.1375  0.2476  0.2920  0.1342   0.2587  0.0017
##
## sigma^2 estimated as 0.005466:  log likelihood=2763.68
## AIC=-5511.36  AICc=-5511.3  BIC=-5465.34
##
## Training set error measures:
##              ME      RMSE      MAE  MPE  MAPE      MASE
ACF1
## Training set 7.095178e-06 0.07382195 0.03969358 NaN  Inf 0.6871478
0.000674244

acf(m2$residuals^2)

```

### Series m2\$residuals^2



```

Box.test(m2$residuals^2,lag = 10,type = 'Ljung')

##
## Box-Ljung test
##

```

```

## data:  m2$residuals^2
## X-squared = 261.5, df = 10, p-value < 2.2e-16

m3= garchFit(~arma(3,3)+garch(1,1),data = rate,trace = F)

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

## 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 = rate, trace = F)
##
## Mean and Variance Equation:
## data ~ arma(3, 3) + garch(1, 1)
## <environment: 0x00000000137eb290>
## [data = rate]
##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##          mu          ar1          ar2          ar3          ma1
ma2
## -0.00099345 -0.05263201 -0.50537351  0.42283458  0.01227320
0.52563486
##          ma3          omega          alpha1          beta1
## -0.44071728  0.00012893  0.29654730  0.73900398
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##          Estimate Std. Error t value Pr(>|t|)
## mu          -9.935e-04  1.098e-03  -0.905  0.36546
## ar1         -5.263e-02  3.151e-01  -0.167  0.86736
## ar2         -5.054e-01  1.748e-01  -2.891  0.00384 **
## ar3          4.228e-01  2.851e-01   1.483  0.13802
## ma1          1.227e-02  3.090e-01   0.040  0.96832
## ma2          5.256e-01  1.640e-01   3.205  0.00135 **
## ma3         -4.407e-01  2.833e-01  -1.556  0.11976
## omega        1.289e-04  1.795e-05   7.182  6.85e-13 ***

```

```

## alpha1 2.965e-01 2.719e-02 10.906 < 2e-16 ***
## beta1 7.390e-01 1.904e-02 38.813 < 2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 3673.958 normalized: 1.578161
##
## Description:
## Wed Jun 02 17:27:19 2021 by user: 10.10
##
##
## Standardised Residuals Tests:
##
## Statistic p-Value
## Jarque-Bera Test R Chi^2 9736.563 0
## Shapiro-Wilk Test R W 0.8764044 0
## Ljung-Box Test R Q(10) 25.55084 0.004393644
## Ljung-Box Test R Q(15) 44.71646 8.487242e-05
## Ljung-Box Test R Q(20) 47.38148 0.0005191375
## Ljung-Box Test R^2 Q(10) 16.1789 0.09462443
## Ljung-Box Test R^2 Q(15) 16.74733 0.3341923
## Ljung-Box Test R^2 Q(20) 19.19326 0.509298
## LM Arch Test R TR^2 15.82561 0.1993548
##
## Information Criterion Statistics:
## AIC BIC SIC HQIC
## -3.147730 -3.123019 -3.147767 -3.138726

m4= garchFit(~arma(3,3)+garch(1,0),data = rate,trace = F)

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

## Warning in arima(.series$x, order = c(u, 0, v), include.mean =
include.mean): Using formula(x) is deprecated when x is a character vector of
length > 1.
## Consider formula(paste(x, collapse = " ")) instead.

summary(m4)

##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(3, 3) + garch(1, 0), data = rate, trace = F)
##
## Mean and Variance Equation:
## data ~ arma(3, 3) + garch(1, 0)
## <environment: 0x0000000020f3c360>
## [data = rate]

```

```

##
## Conditional Distribution:
## norm
##
## Coefficient(s):
##      mu      ar1      ar2      ar3      ma1
ma2
## -6.2596e-06 -1.9619e-01  2.5383e-01  2.7780e-01  8.4942e-02 -2.5553e-
01
##      ma3      omega      alpha1
## -3.3379e-01  1.6543e-03  1.0000e+00
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate  Std. Error  t value  Pr(>|t|)
## mu      -6.260e-06  4.549e-04  -0.014  0.9890
## ar1     -1.962e-01  3.089e-01  -0.635  0.5253
## ar2      2.538e-01  1.232e-01  2.060  0.0394 *
## ar3      2.778e-01  1.524e-01  1.823  0.0683 .
## ma1      8.494e-02  2.889e-01  0.294  0.7687
## ma2     -2.555e-01  1.452e-01  -1.760  0.0784 .
## ma3     -3.338e-01  1.379e-01  -2.420  0.0155 *
## omega    1.654e-03  7.841e-05  21.098  <2e-16 ***
## alpha1   1.000e+00  8.502e-02  11.762  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 3435.069 normalized: 1.475545
##
## Description:
## Wed Jun 02 17:27:20 2021 by user: 10.10
##
##
## Standardised Residuals Tests:
##
##      Statistic  p-Value
## Jarque-Bera Test  R  Chi^2  14587.88  0
## Shapiro-Wilk Test  R  W      0.854688  0
## Ljung-Box Test    R  Q(10)  92.80669  1.44329e-15
## Ljung-Box Test    R  Q(15)  125.0957  0
## Ljung-Box Test    R  Q(20)  130.7534  0
## Ljung-Box Test    R^2 Q(10)  42.00527  7.482514e-06
## Ljung-Box Test    R^2 Q(15)  60.86605  1.788975e-07
## Ljung-Box Test    R^2 Q(20)  75.41615  2.3218e-08
## LM Arch Test      R  TR^2  40.50904  5.91752e-05
##
## Information Criterion Statistics:

```

```

##           AIC           BIC           SIC           HQIC
## -2.943358 -2.921118 -2.943388 -2.935255

m5= garchFit(~arma(3,3)+garch(1,1),data = rate,cond.dist = 'std',trace = F)

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

## Warning in arima(.series$x, order = c(u, 0, v), include.mean =
include.mean): Using formula(x) is deprecated when x is a character vector of
length > 1.
## Consider formula(paste(x, collapse = " ")) instead.

summary(m5)

##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(3, 3) + garch(1, 1), data = rate, cond.dist =
"std",
## trace = F)
##
## Mean and Variance Equation:
## data ~ arma(3, 3) + garch(1, 1)
## <environment: 0x00000000200b85d8>
## [data = rate]
##
## Conditional Distribution:
## std
##
## Coefficient(s):
##           mu           ar1           ar2           ar3           ma1
ma2
## -0.00142845  0.38830308 -0.50602012  0.10205168 -0.53308519
0.57494661
##           ma3           omega           alpha1           beta1           shape
## -0.18459409  0.00017662  0.66916538  0.71102963  2.39292439
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##           Estimate Std. Error t value Pr(>|t|)
## mu          -0.0014284  0.0006017  -2.374 0.017586 *
## ar1           0.3883031  0.2913734   1.333 0.182642
## ar2          -0.5060201  0.1501570  -3.370 0.000752 ***
## ar3           0.1020517  0.1807280   0.565 0.572298
## ma1          -0.5330852  0.2890136  -1.844 0.065111 .
## ma2           0.5749466  0.1390096   4.136 3.53e-05 ***

```

```

## ma3      -0.1845941    0.1827987    -1.010  0.312581
## omega    0.0001766    0.0000624     2.830  0.004651 **
## alpha1   0.6691654    0.2075135     3.225  0.001261 **
## beta1    0.7110296    0.0293852    24.197 < 2e-16 ***
## shape    2.3929244    0.1501627    15.936 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 4104.081    normalized:  1.762922
##
## Description:
## Wed Jun 02 17:27:22 2021 by user: 10.10
##
##
## Standardised Residuals Tests:
##
##                               Statistic p-Value
## Jarque-Bera Test      R      Chi^2 11703.26 0
## Shapiro-Wilk Test     R      W      0.8658672 0
## Ljung-Box Test        R      Q(10) 56.74168 1.49041e-08
## Ljung-Box Test        R      Q(15) 81.0624 4.463296e-11
## Ljung-Box Test        R      Q(20) 85.27692 4.90407e-10
## Ljung-Box Test        R^2    Q(10) 11.10941 0.3490581
## Ljung-Box Test        R^2    Q(15) 11.69615 0.7018674
## Ljung-Box Test        R^2    Q(20) 14.32062 0.8138748
## LM Arch Test          R      TR^2  11.58158 0.4798406
##
## Information Criterion Statistics:
##           AIC           BIC           SIC           HQIC
## -3.516393 -3.489211 -3.516437 -3.506489

m6= garchFit(~arma(3,3)+garch(1,0),data = rate,cond.dist = 'std',trace = F)

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

## Warning in arima(.series$x, order = c(u, 0, v), include.mean =
include.mean): Using formula(x) is deprecated when x is a character vector of
length > 1.
## Consider formula(paste(x, collapse = " ")) instead.

summary(m6)

##
## Title:
## GARCH Modelling
##
## Call:
## garchFit(formula = ~arma(3, 3) + garch(1, 0), data = rate, cond.dist =
"std",
## trace = F)

```

```

##
## Mean and Variance Equation:
## data ~ arma(3, 3) + garch(1, 0)
## <environment: 0x00000002131d5f0>
## [data = rate]
##
## Conditional Distribution:
## std
##
## Coefficient(s):
##      mu      ar1      ar2      ar3      ma1      ma2
## -0.0013427  0.3423265 -0.4494836  0.2165952 -0.4823234  0.5045966
##      ma3      omega      alpha1      shape
## -0.2639528  0.0016711  1.0000000  2.6372226
##
## Std. Errors:
## based on Hessian
##
## Error Analysis:
##      Estimate Std. Error t value Pr(>|t|)
## mu      -0.0013427  0.0005965  -2.251  0.02440 *
## ar1      0.3423265  0.2245698   1.524  0.12742
## ar2     -0.4494836  0.1510429  -2.976  0.00292 **
## ar3      0.2165952  0.1056803   2.050  0.04041 *
## ma1     -0.4823234  0.2218631  -2.174  0.02971 *
## ma2      0.5045966  0.1686922   2.991  0.00278 **
## ma3     -0.2639528  0.1025971  -2.573  0.01009 *
## omega    0.0016711  0.0001946   8.588 < 2e-16 ***
## alpha1   1.0000000  0.1135881   8.804 < 2e-16 ***
## shape    2.6372226  0.0956395  27.575 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Log Likelihood:
## 3966.142    normalized:  1.703669
##
## Description:
## Wed Jun 02 17:27:24 2021 by user: 10.10
##
##
## Standardised Residuals Tests:
##
##      Statistic p-Value
## Jarque-Bera Test  R    Chi^2 23197.99 0
## Shapiro-Wilk Test R    W    0.8353825 0
## Ljung-Box Test   R    Q(10) 54.36079 4.156337e-08
## Ljung-Box Test   R    Q(15) 77.58016 1.929145e-10
## Ljung-Box Test   R    Q(20) 81.49889 2.182092e-09
## Ljung-Box Test   R^2  Q(10) 40.80123 1.223601e-05
## Ljung-Box Test   R^2  Q(15) 60.96747 1.718316e-07
## Ljung-Box Test   R^2  Q(20) 71.93533 8.771372e-08

```

```

## LM Arch Test      R      TR^2   45.90772  7.196928e-06
##
## Information Criterion Statistics:
##      AIC      BIC      SIC      HQIC
## -3.398747 -3.374036 -3.398784 -3.389743

#Question3
getSymbols("UKNGDP",from="1980-01-01", to="2011-04-1",src = "FRED")

## [1] "UKNGDP"

getSymbols("NAEXKP01CAQ189S",from="1980-01-01", to="2011-06-31",src = "FRED")

## [1] "NAEXKP01CAQ189S"

getSymbols("GDP",from="1980-01-01", to="2011-06-31",src = "FRED")

## [1] "GDP"

zt=cbind(UKNGDP$UKNGDP,NAEXKP01CAQ189S$NAEXKP01CAQ189S,GDP$GDP)
zt=zt[132:258,]
colnames(zt) <- c("UKGDP","CANADAGDP","USGDP")
log_zt=log(zt)
diff_log_zt=diff(log_zt)*100
diff_log_zt=diff_log_zt[2:127,]
#3.1
varfit= VAR(diff_log_zt,p=2)
summary(varfit)

##
## VAR Estimation Results:
## =====
## Endogenous variables: UKGDP, CANADAGDP, USGDP
## Deterministic variables: const
## Sample size: 124
## Log Likelihood: -387.34
## Roots of the characteristic polynomial:
## 0.7642 0.5113 0.4815 0.3415 0.3165 0.3165
## Call:
## VAR(y = diff_log_zt, p = 2)
##
##
## Estimation results for equation UKGDP:
## =====
## UKGDP = UKGDP.l1 + CANADAGDP.l1 + USGDP.l1 + UKGDP.l2 + CANADAGDP.l2 +
USGDP.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## UKGDP.l1      0.13130   0.09349   1.404  0.16282
## CANADAGDP.l1 -0.31837   0.14219  -2.239  0.02704 *
## USGDP.l1      0.40317   0.13747   2.933  0.00404 **
## UKGDP.l2      0.27087   0.09313   2.908  0.00435 **

```

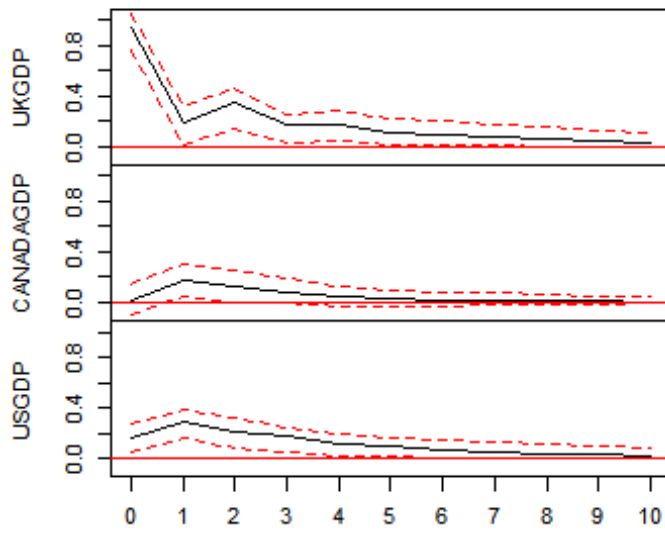
```

## CANADAGDP.l2  0.12100    0.13823    0.875    0.38316
## USGDP.l2      0.00351    0.14283    0.025    0.98044
## const        0.44760    0.19167    2.335    0.02123 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.9323 on 117 degrees of freedom
## Multiple R-Squared: 0.2855, Adjusted R-squared: 0.2488
## F-statistic: 7.791 on 6 and 117 DF, p-value: 4.599e-07
##
##
## Estimation results for equation CANADAGDP:
## =====
## CANADAGDP = UKGDP.l1 + CANADAGDP.l1 + USGDP.l1 + UKGDP.l2 + CANADAGDP.l2 +
USGDP.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## UKGDP.l1      0.11947    0.06266    1.907 0.059002 .
## CANADAGDP.l1  0.47321    0.09530    4.966 2.35e-06 ***
## USGDP.l1      0.31920    0.09213    3.465 0.000743 ***
## UKGDP.l2     -0.02657    0.06242   -0.426 0.671114
## CANADAGDP.l2  0.02889    0.09264    0.312 0.755747
## USGDP.l2     -0.27868    0.09573   -2.911 0.004313 **
## const         0.11003    0.12846    0.857 0.393463
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.6248 on 117 degrees of freedom
## Multiple R-Squared: 0.4148, Adjusted R-squared: 0.3847
## F-statistic: 13.82 on 6 and 117 DF, p-value: 7.957e-12
##
##
## Estimation results for equation USGDP:
## =====
## USGDP = UKGDP.l1 + CANADAGDP.l1 + USGDP.l1 + UKGDP.l2 + CANADAGDP.l2 +
USGDP.l2 + const
##
##           Estimate Std. Error t value Pr(>|t|)
## UKGDP.l1      0.26552    0.06779    3.917 0.000151 ***
## CANADAGDP.l1  0.18335    0.10310    1.778 0.077936 .
## USGDP.l1      0.26106    0.09968    2.619 0.009986 **
## UKGDP.l2      0.05442    0.06753    0.806 0.421998
## CANADAGDP.l2 -0.06486    0.10023   -0.647 0.518799
## USGDP.l2      0.02657    0.10357    0.257 0.797946
## const         0.42320    0.13898    3.045 0.002874 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##

```

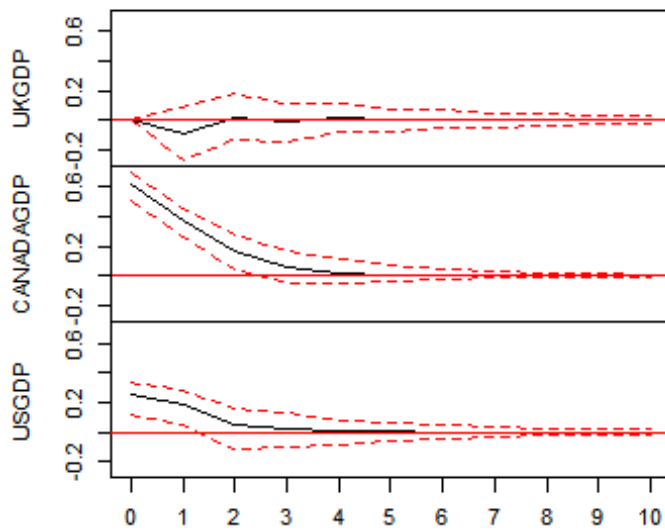
```
##
## Residual standard error: 0.676 on 117 degrees of freedom
## Multiple R-Squared: 0.388, Adjusted R-squared: 0.3566
## F-statistic: 12.36 on 6 and 117 DF, p-value: 9.544e-11
##
##
## Covariance matrix of residuals:
##          UKGDP CANADAGDP  USGDP
## UKGDP    0.8692    0.0107 0.1552
## CANADAGDP 0.0107    0.3904 0.1617
## USGDP     0.1552    0.1617 0.4570
##
## Correlation matrix of residuals:
##          UKGDP CANADAGDP  USGDP
## UKGDP    1.00000    0.01837 0.2462
## CANADAGDP 0.01837    1.00000 0.3827
## USGDP     0.24625    0.38272 1.0000
##
##3.2
impresp_3 = irf(varfit)
plot(impresp_3)
```

### Orthogonal Impulse Response from UKGDP



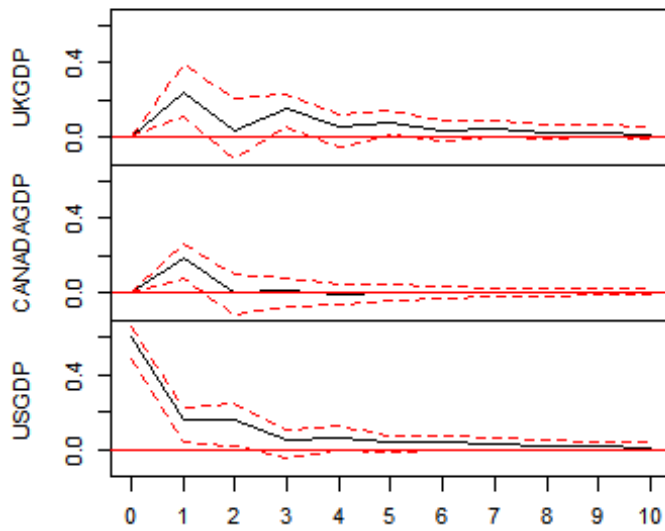
95 % Bootstrap CI, 100 runs

### Orthogonal Impulse Response from CANADAGDP



95 % Bootstrap CI, 100 runs

### Orthogonal Impulse Response from USGDP

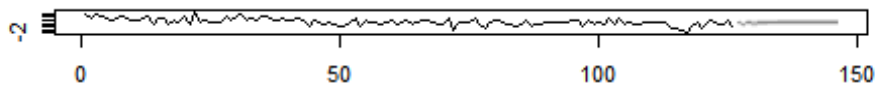


95 % Bootstrap CI, 100 runs

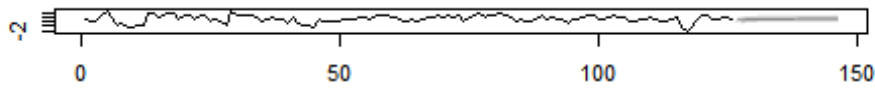
#3.3

```
varfit.prd= predict(varfit, n.ahead = 20, ci= 0.95)  
fanchart(varfit.prd)
```

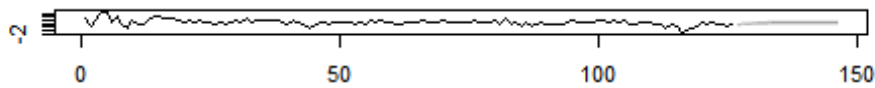
Fanchart for variable UKGDP



Fanchart for variable CANADAGDP



Fanchart for variable USGDP



```
fevd(varfit,n.ahead = 20)
```

```
## $UKGDP
##           UKGDP   CANADAGDP   USGDP
## [1,] 1.0000000 0.000000000 0.0000000
## [2,] 0.9297076 0.009441237 0.06085112
## [3,] 0.9359223 0.008671549 0.05540613
## [4,] 0.9171259 0.008354668 0.07451939
## [5,] 0.9171105 0.008295939 0.07459352
## [6,] 0.9133324 0.008173334 0.07849425
## [7,] 0.9128934 0.008134848 0.07897178
## [8,] 0.9120215 0.008097890 0.07988060
## [9,] 0.9117790 0.008081759 0.08013920
## [10,] 0.9115388 0.008070197 0.08039100
## [11,] 0.9114390 0.008064331 0.08049666
## [12,] 0.9113653 0.008060576 0.08057408
## [13,] 0.9113285 0.008058521 0.08061296
## [14,] 0.9113046 0.008057269 0.08063809
## [15,] 0.9112916 0.008056559 0.08065180
## [16,] 0.9112837 0.008056137 0.08066018
## [17,] 0.9112792 0.008055894 0.08066492
## [18,] 0.9112765 0.008055750 0.08066775
## [19,] 0.9112750 0.008055667 0.08066938
## [20,] 0.9112740 0.008055618 0.08067034
##
## $CANADAGDP
##           UKGDP   CANADAGDP   USGDP
## [1,] 0.0003373506 0.9996626 0.00000000
## [2,] 0.0484624625 0.8896079 0.06192963
## [3,] 0.0694636089 0.8729466 0.05758981
## [4,] 0.0795270108 0.8637328 0.05674023
## [5,] 0.0832090324 0.8603302 0.05646073
## [6,] 0.0847049460 0.8588925 0.05640257
## [7,] 0.0853874952 0.8582191 0.05639341
## [8,] 0.0857234666 0.8578577 0.05641885
## [9,] 0.0859041554 0.8576590 0.05643684
## [10,] 0.0860025114 0.8575461 0.05645139
## [11,] 0.0860581801 0.8574817 0.05646014
## [12,] 0.0860896845 0.8574445 0.05646579
## [13,] 0.0861078461 0.8574230 0.05646912
## [14,] 0.0861183097 0.8574105 0.05647115
## [15,] 0.0861243897 0.8574033 0.05647233
## [16,] 0.0861279206 0.8573990 0.05647304
## [17,] 0.0861299788 0.8573966 0.05647345
## [18,] 0.0861311780 0.8573951 0.05647369
## [19,] 0.0861318779 0.8573943 0.05647383
## [20,] 0.0861322862 0.8573938 0.05647392
##
## $USGDP
##           UKGDP   CANADAGDP   USGDP
```

```
## [1,] 0.06063856 0.1430828 0.7962787
## [2,] 0.18918578 0.1636079 0.6472063
## [3,] 0.23517742 0.1508355 0.6139871
## [4,] 0.26741869 0.1443755 0.5882058
## [5,] 0.28063551 0.1407135 0.5786510
## [6,] 0.28887541 0.1387469 0.5723777
## [7,] 0.29296398 0.1376151 0.5694209
## [8,] 0.29547802 0.1369707 0.5675512
## [9,] 0.29685250 0.1365947 0.5665528
## [10,] 0.29767814 0.1363771 0.5659447
## [11,] 0.29814676 0.1362500 0.5656032
## [12,] 0.29842405 0.1361761 0.5653999
## [13,] 0.29858396 0.1361329 0.5652832
## [14,] 0.29867792 0.1361077 0.5652144
## [15,] 0.29873248 0.1360930 0.5651746
## [16,] 0.29876444 0.1360844 0.5651512
## [17,] 0.29878305 0.1360793 0.5651376
## [18,] 0.29879394 0.1360764 0.5651296
## [19,] 0.29880029 0.1360747 0.5651250
## [20,] 0.29880400 0.1360737 0.5651223
```

### #3.4

```
UK=zt$UKGDP
```

```
CANADA=zt$CANADAGDP
```

```
US=zt$USGDP
```

```
fit <-lm(UK~CANADA+US)
```

```
summary(fit)
```

```
##
```

```
## Call:
```

```
## lm(formula = UK ~ CANADA + US)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -11745.9  -6772.7  -669.4   6486.1  12835.5
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.387e+04  1.326e+04   1.046   0.298
## CANADA      -1.324e-07  8.545e-08  -1.550   0.124
## US          3.013e+01  1.659e+00  18.165 <2e-16 ***
```

```
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
```

```
## Residual standard error: 7111 on 124 degrees of freedom
```

```
## Multiple R-squared:  0.9958, Adjusted R-squared:  0.9957
```

```
## F-statistic: 1.461e+04 on 2 and 124 DF,  p-value: < 2.2e-16
```

```
fit2 <-lm(CANADA~UK+US)
```

```
summary(fit2)
```

```
##
## Call:
## lm(formula = CANADA ~ UK + US)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.317e+10 -5.414e+09 -9.450e+08  5.901e+09  1.584e+10
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.533e+11  1.670e+09  91.771 < 2e-16 ***
## UK          -1.435e+05  9.258e+04  -1.550  0.124
## US           2.328e+07  2.558e+06   9.098  1.9e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.402e+09 on 124 degrees of freedom
## Multiple R-squared:  0.9907, Adjusted R-squared:  0.9906
## F-statistic: 6623 on 2 and 124 DF,  p-value: < 2.2e-16

fit3 <-lm(US~UK+CANADA)
summary(fit3)
```

```
##
## Call:
## lm(formula = US ~ UK + CANADA)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -379.53 -142.12  -50.88  176.28  403.55
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.473e+03  3.046e+02  -8.118 3.98e-13 ***
## UK           2.412e-02  1.328e-03  18.165 < 2e-16 ***
## CANADA       1.720e-08  1.890e-09   9.098 1.90e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 201.2 on 124 degrees of freedom
## Multiple R-squared:  0.9974, Adjusted R-squared:  0.9974
## F-statistic: 2.395e+04 on 2 and 124 DF,  p-value: < 2.2e-16
```

### #3.5

```
diff.UK=diff(UK)
diff.CANADA=diff(CANADA)
diff.US=diff(US)
diff.UK.L1=Lag(diff(UK),k=1)
diff.CANADA.L1=Lag(diff(CANADA),k=1)
diff.US.L1=Lag(diff(US),k=1)
```

```

error=residuals(fit)
error.L1= Lag(error,k=1)
error2=residuals(fit2)
error2.L1= Lag(error2,k=1)
error3=residuals(fit3)
error3.L1=Lag(error3,k=1)
VECM<-lm(diff.UK~diff.UK.L1+diff.CANADA.L1+diff.US.L1+error.L1)
summary(VECM)

##
## Call:
## lm(formula = diff.UK ~ diff.UK.L1 + diff.CANADA.L1 + diff.US.L1 +
##     error.L1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6024.8 -1070.3   168.1  1310.2  4773.9
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.134e+03  3.423e+02   3.312  0.00123 **
## diff.UK.L1    -4.232e-02  9.151e-02  -0.462  0.64461
## diff.CANADA.L1 -1.802e-07  9.775e-08  -1.844  0.06768 .
## diff.US.L1     2.116e+01  3.539e+00   5.980  2.36e-08 ***
## error.L1      -1.726e-02  2.690e-02  -0.642  0.52233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2061 on 120 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.2744, Adjusted R-squared:  0.2503
## F-statistic: 11.35 on 4 and 120 DF,  p-value: 7.632e-08

VECM2<-lm(diff.CANADA~diff.UK.L1+diff.CANADA.L1+diff.US.L1+error2.L1)
summary(VECM2)

##
## Call:
## lm(formula = diff.CANADA ~ diff.UK.L1 + diff.CANADA.L1 + diff.US.L1 +
##     error2.L1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.364e+09 -1.278e+09 -1.858e+08  1.301e+09  5.505e+09
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.780e+08  3.139e+08  -0.886  0.377585
## diff.UK.L1   9.858e+04  8.273e+04   1.192  0.235756
## diff.CANADA.L1 3.338e-01  8.864e-02   3.766  0.000259 ***

```

```

## diff.US.L1      1.248e+07  3.200e+06   3.900 0.000159 ***
## error2.L1      -5.261e-02  2.365e-02  -2.225 0.027984 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.864e+09 on 120 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.4371, Adjusted R-squared:  0.4183
## F-statistic: 23.3 on 4 and 120 DF,  p-value: 2.889e-14

VECM3<-lm(diff.US~diff.UK.L1+diff.CANADA.L1+diff.US.L1+error3.L1)
summary(VECM3)

##
## Call:
## lm(formula = diff.US ~ diff.UK.L1 + diff.CANADA.L1 + diff.US.L1 +
##     error3.L1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -306.873  -21.905   -1.991   32.717  136.470
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.496e+01  9.532e+00   3.668 0.000366 ***
## diff.UK.L1    1.073e-02  2.557e-03   4.196 5.24e-05 ***
## diff.CANADA.L1 2.670e-09  2.693e-09   0.991 0.323595
## diff.US.L1    3.098e-01  9.779e-02   3.168 0.001947 **
## error3.L1    -2.078e-02  2.673e-02  -0.777 0.438401
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 56.94 on 120 degrees of freedom
## (2 observations deleted due to missingness)
## Multiple R-squared:  0.3965, Adjusted R-squared:  0.3763
## F-statistic: 19.71 on 4 and 120 DF,  p-value: 1.726e-12

```