Econ 8011: Time Series
Final Project Report

By Atikul Khalid and Karthikeya Nalluri
Date: 4/17/2023

How Disposable Personal Income and Median Sales Price of Houses Impact GDP: A Time Series Analysis

Gross Domestic Product (gdp) is perhaps the most dominant economic indicator of a country. GDP can be explained by many time series variables, housing and real estate is one of them. housing and real estate is an important economic sector, as we saw, 2008 economic recession was happened mostly because of the instability of this sector. Focusing on housing prices, in this class project, we try to explain gdp by unemployment rate, inflation (cpi), federal funds effective rate (ffe rate), disposable personal income (dpi), minimum hourly wages and median sales prices of houses using Vector Autoregressive Model (VAR) or Vector Error Correction Model (VECM). To do that, we collected following seasonally unadjusted quarterly data from the FRED website from 1963 1st quarter to 2022 4th quarter.

Variable 1: Gross Domestic Product (NA000334Q) [millions of dollars]
Source: U.S. Bureau of Economic Analysis, Gross Domestic Product [NA000334Q], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/NA000334Q, April 14, 2023

Variable 2: Unemployment Rate: Aged 15 and Over: All Persons for the United States (LRUNTTTTUSQ156N)
Source: Organization for Economic Co-operation and Development, Unemployment Rate: Aged 15 and Over: All Persons for the United States [LRUNTTTTUSQ156N], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/LRUNTTTTUSQ156N, April 15, 2023.

Variable 3: Consumer Price Index: Total All Items for the United States (CPALTT01USM659N) (Growth rate same period previous year, Not Seasonally Adjusted)
Source: Organization for Economic Co-operation and Development, Consumer Price Index: Total All Items for the United States [CPALTT01USQ659N], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/CPALTT01USQ659N, April 14, 2023.

Variable 4: Federal Funds Effective Rate (FEDFUNDS)
Source: Board of Governors of the Federal Reserve System (US), Federal Funds Effective Rate [FEDFUNDS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/FEDFUNDS, April 15, 2023.

Variable 5: Households and Nonprofit Organizations; Disposable Personal Income, Transactions (BOGZ1FU156012005Q) [millions of dollars]

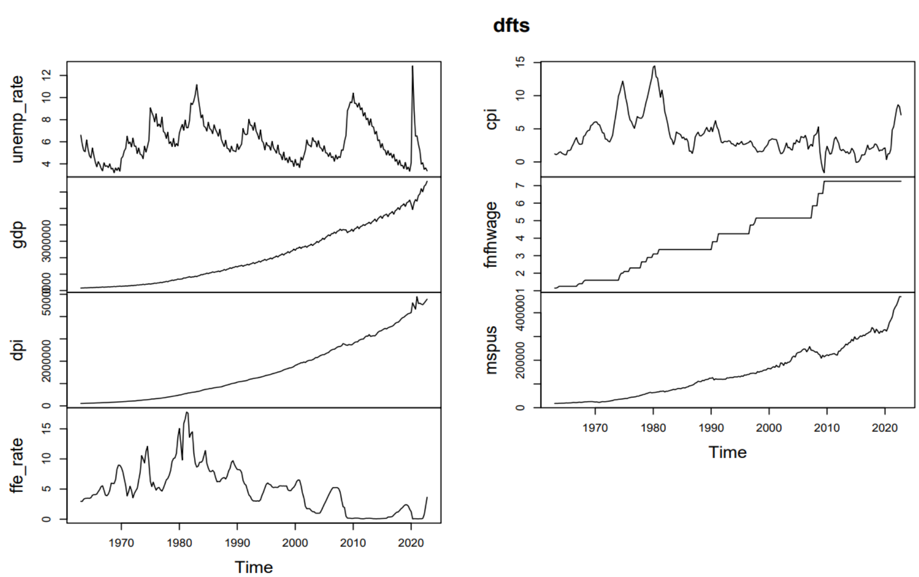
Source: Board of Governors of the Federal Reserve System (US), Households and Nonprofit Organizations; Disposable Personal Income, Transactions [BOGZ1FU156012005Q], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/BOGZ1FU156012005Q, April 15, 2023.

Variable 6: Federal Minimum Hourly Wage for Nonfarm Workers for the United States (FEDMINNFRWG) [dollars]
Source: U.S. Department of Labor, Federal Minimum Hourly Wage for Nonfarm Workers for the United States [FEDMINNFRWG], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/FEDMINNFRWG, April 13, 2023.

Variable 7: Median Sales Price of Houses Sold for the United States (MSPUS) [dollars]
Source: U.S. Census Bureau and U.S. Department of Housing and Urban Development, Median Sales Price of Houses Sold for the United States [MSPUS], retrieved from FRED, Federal Reserve Bank of St. Louis; https://fred.stlouisfed.org/series/MSPUS, April 15, 2023.

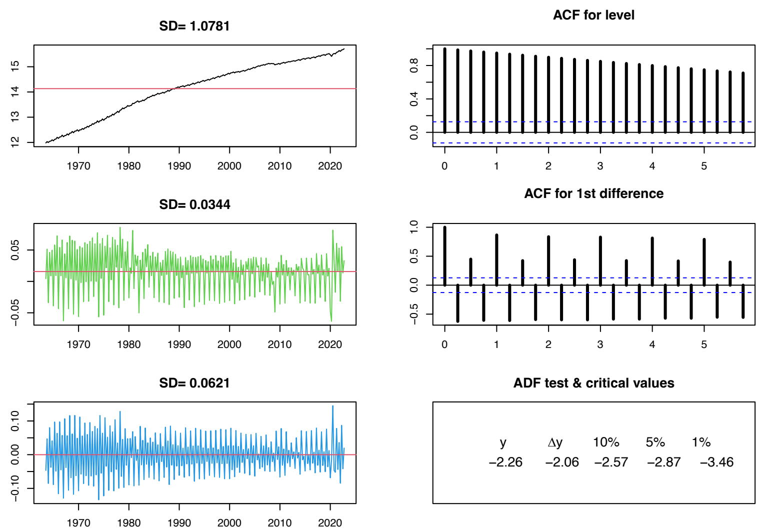
Variable Selection

The simple plotting of all the timeseries variables exhibits that gdp, dpi, Median Sales Prices of Houses (mspus) and federal minimum hourly wages for non-firm workers (fnfhwage) have similar trend over time. After doing necessary transformation (i.e., log, differentiation) of the variables, we tried many different combinations of these seven variables to have a better VAR or VECM Model including gdp and mspus every time. Finally, we got best model with gdp, mspus and dpi, and it is expected from the simple plotting as well, because seemingly these three variables have the most similar trend. Moving forward, in this paper we presented all the details of the models with these three variables and that is how our study turned out to be “How Disposable Personal Income and Median Sales Price of Houses Impact GDP”.

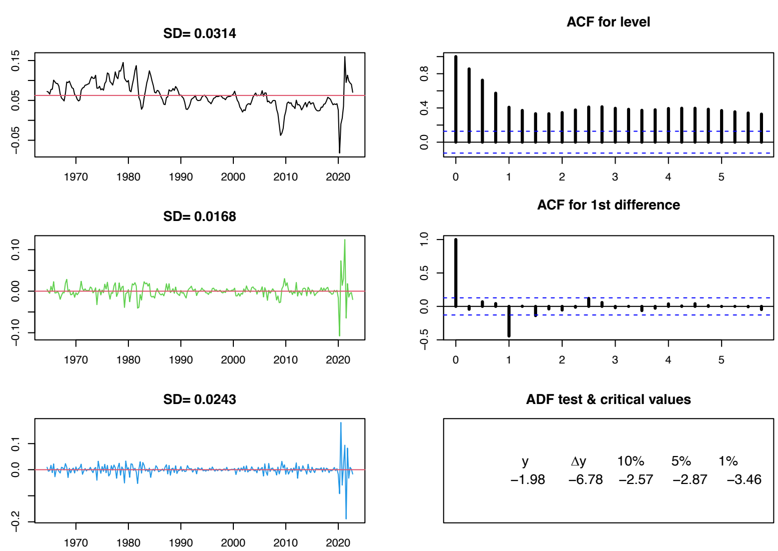


Checking stationarity

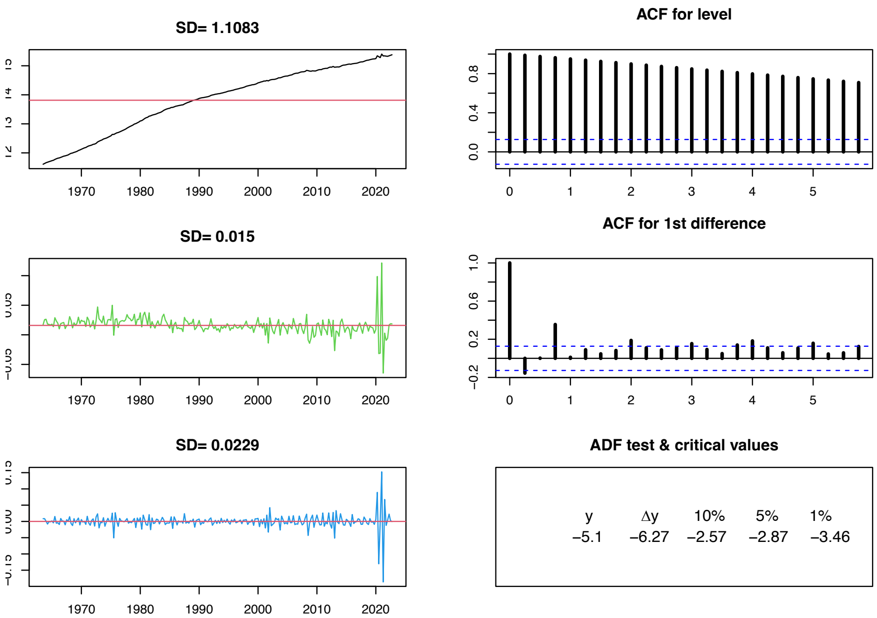
We transform all our considered three variables into log form and then checked stationarity using Intord function in R. Below is the intord output of log gdp, which is clearly I(2). We can see seasonality from the ACF graph of 1st difference.



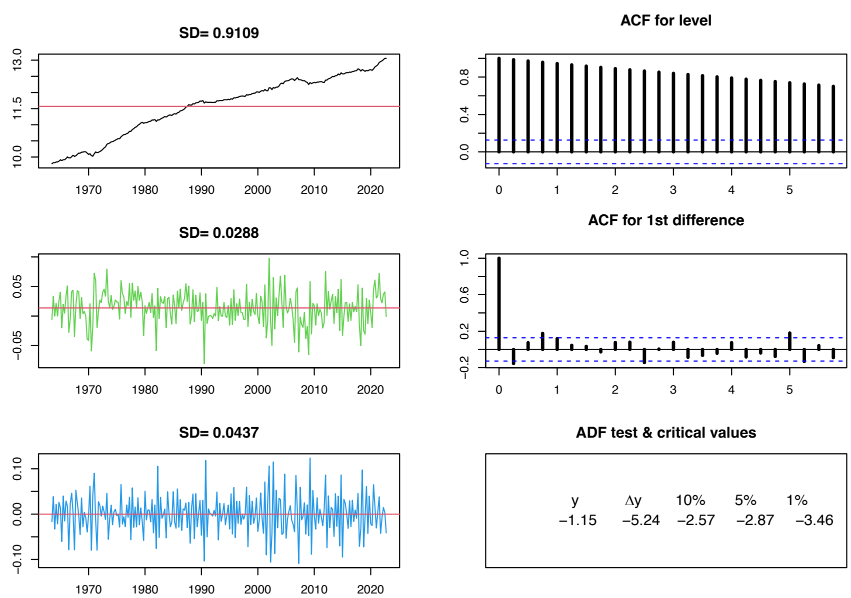
So, we seasonally differenced log gdp and check Intord output again. Now we see, the order of integration is I(1).



The below is the intord output of log dpi. we can observe that the variable is not stationary at level but stationary at first differencing. Hence the order of integration is I(1).



The below is the intord output of log mspus. We can observe that the log mspus is not stationary at level but stationary at first differencing. Hence, the order of integration is I(1).



Selecting lags

Now we have all the three variable in I(1) order of integration, which is a necessary condition to have a VAR or VECM model, because, if variables are stationary long run cointegration will not be found. To find the optimal number of lag, we used VARselect function and found two (2) as the number of lags according to schwartz criteria [SC(n) = 2]. Output of the function is given below.

```
> VARselect(ly, lag.max=12, type="const")
$selection
AIC(n)  HQ(n)  SC(n) FPE(n)
     9      5      2      9
```

Developing Model

When we used two as the number of lags, from both eigen and trace test statistic we found at least 2 combinations of eigenvectors of cointegration relations that are statistically significant. But using two lags, later we found serial correlation in the residuals from VAR, VECM (Johansen) and VECM (Engle-Granger) model. So, we tried different number of lags such as 4, 6 and 8 and found 8 is the appropriate number of lags, that produce no serial correlation in the residuals of all three models. When we used eight (8) as the number of lags, from both eigen and trace test statistic we found at least 3 combinations of eigenvectors of cointegration relations that are statistically significant.

Eigen test statistic summary

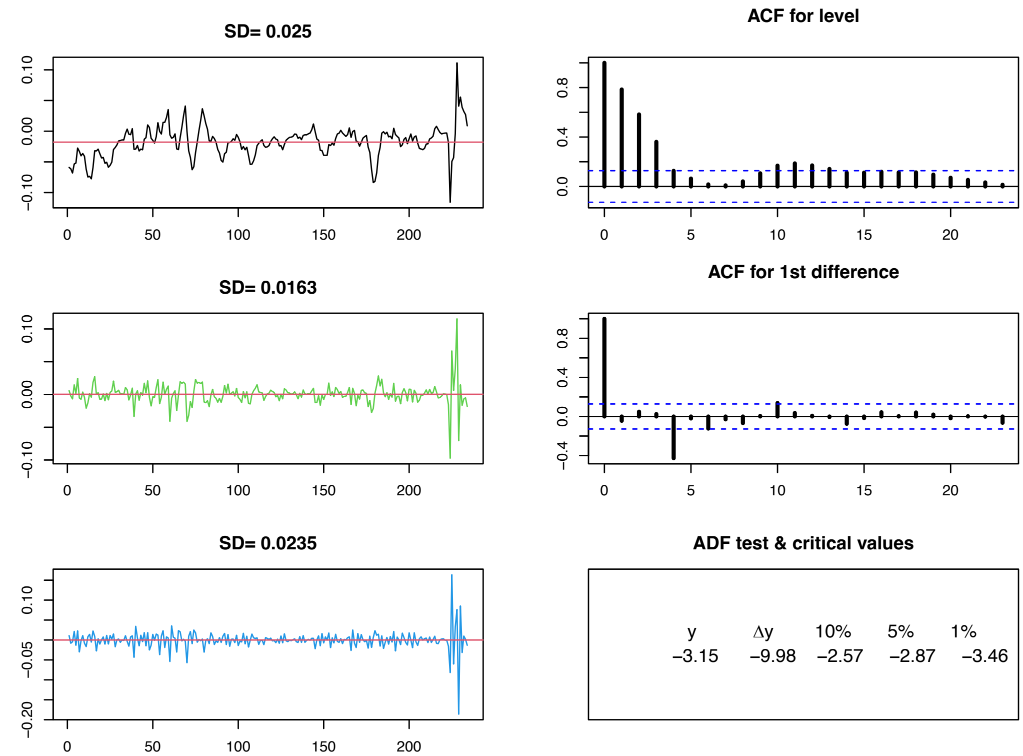
```
######################
# Johansen-Procedure #
######################

Test type: maximal eigenvalue statistic (lambda max) , without linear trend and constant
in cointegration

Eigenvalues (lambda):
[1]  0.2518881853142922456  0.0968136052311164563  0.0659429264957222677
[4] -0.0000000000000000111

Values of teststatistic and critical values of test:

         test 10pct  5pct 1pct
r <= 2 | 15.6  7.52  9.24 13.0
r <= 1 | 23.2 13.75 15.67 20.2
r = 0  | 66.2 19.77 22.00 26.8

Eigenvectors, normalised to first column:
(These are the cointegration relations)

          dlgdp.l8 ldpi.l8 lmspus.l8 constant
dlgdp.l8   1.00000  1.0000     1.000    1.000
ldpi.l8    0.04450  0.0918     2.066   -0.211
lmspus.l8 -0.05289 -0.0836    -2.468    0.460
constant  -0.00345 -0.3824    -0.113   -2.327

Weights W:
(This is the loading matrix)

         dlgdp.l8 ldpi.l8 lmspus.l8             constant
dlgdp.d    -0.441  -0.175  0.006063 -0.00000000000002223
ldpi.d      0.520  -0.129 -0.000419  0.00000000000000575
lmspus.d    0.059  -0.042  0.050355  0.00000000000008193
```

Trace test statistic summary

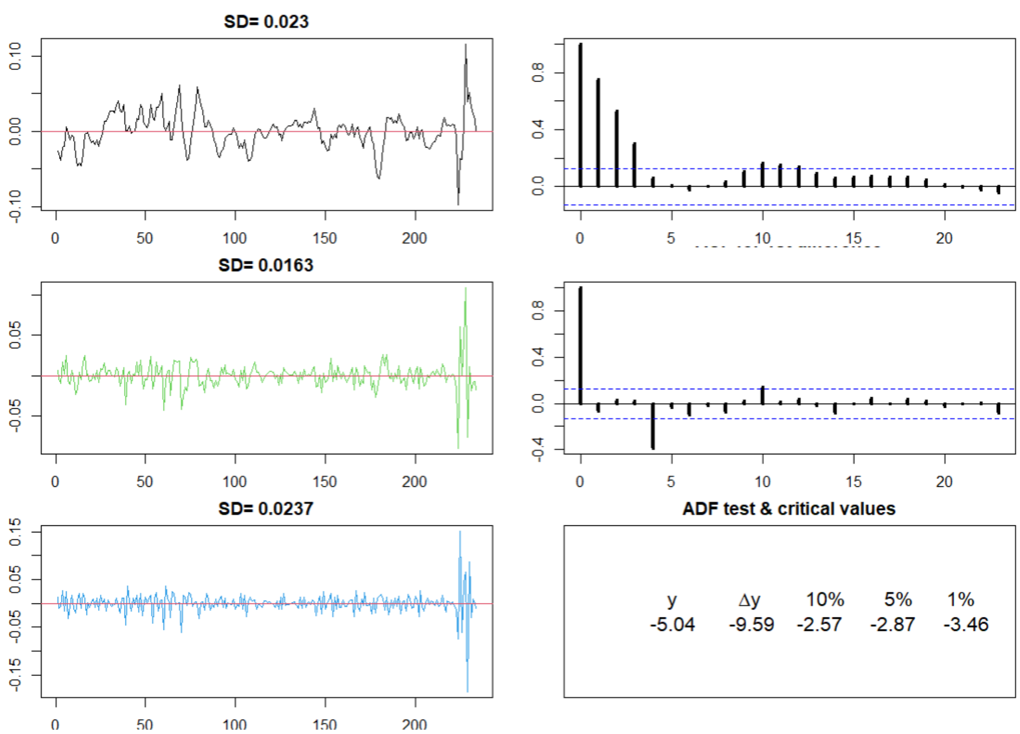
```
######################
# Johansen-Procedure #
######################

Test type: trace statistic , without linear trend and constant in cointegration

Eigenvalues (lambda):
[1]  0.2518881853142922456  0.0968136052311164563  0.0659429264957222677
[4] -0.0000000000000000111

Values of teststatistic and critical values of test:

          test 10pct  5pct 1pct
r <= 2 |  15.6  7.52  9.24 13.0
r <= 1 |  38.8 17.85 19.96 24.6
r = 0  | 104.9 32.00 34.91 41.1

Eigenvectors, normalised to first column:
(These are the cointegration relations)

          dlgdp.l8 ldpi.l8 lmspus.l8 constant
dlgdp.l8   1.00000  1.0000     1.000    1.000
ldpi.l8    0.04450  0.0918     2.066   -0.211
lmspus.l8 -0.05289 -0.0836    -2.468    0.460
constant  -0.00345 -0.3824    -0.113   -2.327

Weights W:
(This is the loading matrix)

         dlgdp.l8 ldpi.l8 lmspus.l8             constant
dlgdp.d    -0.441  -0.175  0.006063 -0.00000000000002223
ldpi.d      0.520  -0.129 -0.000419  0.00000000000000575
lmspus.d    0.059  -0.042  0.050355  0.00000000000008193
```

Among these 3 combinations of eigenvectors, the 2^{nd} combination produced best result regarding making residuals stationary.

Below is the intord output of ecm under Johansen method. According to standard deviation reduction, ecm is stationary and if we consider 5% significance level, ADF test also confirms ecm is stationary, that make us believe that, we can proceed with Johansen VECM.



Below is the intord output of ecm under Engle-Granger method. Standard deviation and ADF test both confirms ecm is stationary, so, we can proceed with Engle-Granger VECM as well.



So, we ran VAR, VECM (Johansen) and VECM (Engle-Granger) model using 8 lags and we found VECM (Johansen) is the best model though, according to AIC, BIC and log likelihood values VECM (Engle-Granger) is the best. We did not rely on VECM (Engle-Granger) because one exogenous term (error correction term) was positive and significant at 6% level.

Criteria	VAR	VECM (Johansen)	VECM (Engle-Granger)
AIC	-3611	-3618	-3623
BIC	-3324	-3320	-3325
Log Likelihood	1890	1896	1898

The following is the summary of VECM (Johansen) model.

```
Estimation results for equation ddlgdp:
=======================================
ddlgdp = ddlgdp.l1 + dldpi.l1 + dlmspus.l1 + ddlgdp.l2 + dldpi.l2 + dlmspus.l2 + ddlg
dp.l3 + dldpi.l3 + dlmspus.l3 + ddlgdp.l4 + dldpi.l4 + dlmspus.l4 + ddlgdp.l5 + dldp
i.l5 + dlmspus.l5 + ddlgdp.l6 + dldpi.l6 + dlmspus.l6 + ddlgdp.l7 + dldpi.l7 + dlmspu
s.l7 + ddlgdp.l8 + dldpi.l8 + dlmspus.l8 + const + sd1 + sd2 + sd3 + exo1

            Estimate Std. Error t value   Pr(>|t|)
ddlgdp.l1   0.137913   0.083686    1.65      0.101
dldpi.l1    0.330086   0.073933    4.46 0.00001346 ***
dlmspus.l1  0.050513   0.035689    1.42      0.159
ddlgdp.l2   0.026291   0.082590    0.32      0.751
dldpi.l2   -0.114843   0.076871   -1.49      0.137
dlmspus.l2  0.081900   0.036701    2.23      0.027 *
ddlgdp.l3  -0.015719   0.084901   -0.19      0.853
dldpi.l3   -0.024838   0.076916   -0.32      0.747
dlmspus.l3  0.066441   0.036971    1.80      0.074 .
ddlgdp.l4  -0.423321   0.082417   -5.14 0.00000067 ***
dldpi.l4    0.022897   0.081143    0.28      0.778
dlmspus.l4  0.020213   0.037422    0.54      0.590
ddlgdp.l5  -0.025114   0.074588   -0.34      0.737
dldpi.l5   -0.108627   0.084597   -1.28      0.201
dlmspus.l5  0.039964   0.037562    1.06      0.289
ddlgdp.l6  -0.058568   0.074700   -0.78      0.434
dldpi.l6   -0.026720   0.081076   -0.33      0.742
dlmspus.l6 -0.002539   0.037026   -0.07      0.945
ddlgdp.l7   0.099761   0.081639    1.22      0.223
dldpi.l7   -0.123784   0.083013   -1.49      0.138
dlmspus.l7 -0.069303   0.037073   -1.87      0.063 .
ddlgdp.l8  -0.403763   0.082016   -4.92 0.00000179 ***
dldpi.l8   -0.069408   0.092617   -0.75      0.455
dlmspus.l8  0.027799   0.037230    0.75      0.456
const      -0.003875   0.002962   -1.31      0.192
sd1        -0.005210   0.003006   -1.73      0.085 .
sd2        -0.000794   0.002950   -0.27      0.788
sd3        -0.000684   0.003029   -0.23      0.821
exo1       -0.163736   0.065281   -2.51      0.013 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


Residual standard error: 0.013 on 198 degrees of freedom
Multiple R-Squared: 0.489,      Adjusted R-squared: 0.417
F-statistic: 6.77 on 28 and 198 DF,  p-value: <0.0000000000000002
```

```
Estimation results for equation dldpi:
======================================
dldpi = ddlgdp.l1 + dldpi.l1 + dlmspus.l1 + ddlgdp.l2 + dldpi.l2 + dlmspus.l2 + ddlgd
p.l3 + dldpi.l3 + dlmspus.l3 + ddlgdp.l4 + dldpi.l4 + dlmspus.l4 + ddlgdp.l5 + dldpi.
l5 + dlmspus.l5 + ddlgdp.l6 + dldpi.l6 + dlmspus.l6 + ddlgdp.l7 + dldpi.l7 + dlmspus.
l7 + ddlgdp.l8 + dldpi.l8 + dlmspus.l8 + const + sd1 + sd2 + sd3 + exo1

           Estimate Std. Error t value  Pr(>|t|)
ddlgdp.l1   0.13332    0.08164    1.63   0.10406
dldpi.l1   -0.18294    0.07213   -2.54   0.01197 *
dlmspus.l1  0.00502    0.03482    0.14   0.88540
ddlgdp.l2   0.24229    0.08057    3.01   0.00298 **
dldpi.l2    0.04462    0.07499    0.60   0.55251
dlmspus.l2  0.00277    0.03580    0.08   0.93840
ddlgdp.l3  -0.14748    0.08283   -1.78   0.07652 .
dldpi.l3    0.35486    0.07504    4.73 0.0000043 ***
dlmspus.l3 -0.01156    0.03607   -0.32   0.74897
ddlgdp.l4   0.07317    0.08040    0.91   0.36390
dldpi.l4    0.12904    0.07916    1.63   0.10467
dlmspus.l4 -0.04382    0.03651   -1.20   0.23145
ddlgdp.l5  -0.00863    0.07276   -0.12   0.90567
dldpi.l5   -0.08769    0.08253   -1.06   0.28926
dlmspus.l5  0.02194    0.03664    0.60   0.55001
ddlgdp.l6   0.18287    0.07287    2.51   0.01290 *
dldpi.l6   -0.01534    0.07909   -0.19   0.84640
dlmspus.l6  0.06414    0.03612    1.78   0.07732 .
ddlgdp.l7  -0.09019    0.07964   -1.13   0.25883
dldpi.l7    0.05379    0.08098    0.66   0.50729
dlmspus.l7  0.14235    0.03617    3.94   0.00011 ***
ddlgdp.l8   0.06236    0.08001    0.78   0.43671
dldpi.l8    0.20637    0.09035    2.28   0.02343 *
dlmspus.l8 -0.01705    0.03632   -0.47   0.63927
const       0.00431    0.00289    1.49   0.13774
sd1         0.00237    0.00293    0.81   0.42023
sd2        -0.00451    0.00288   -1.57   0.11866
sd3        -0.00663    0.00296   -2.24   0.02590 *
exo1       -0.06905    0.06369   -1.08   0.27961
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.0127 on 198 degrees of freedom
Multiple R-Squared: 0.402,      Adjusted R-squared: 0.318
F-statistic: 4.76 on 28 and 198 DF,  p-value: 0.0000000000229
Estimation results for equation dlmspus:
========================================
dlmspus = ddlgdp.l1 + dldpi.l1 + dlmspus.l1 + ddlgdp.l2 + dldpi.l2 + dlmspus.l2 + ddl
gdp.l3 + dldpi.l3 + dlmspus.l3 + ddlgdp.l4 + dldpi.l4 + dlmspus.l4 + ddlgdp.l5 + dldp
i.l5 + dlmspus.l5 + ddlgdp.l6 + dldpi.l6 + dlmspus.l6 + ddlgdp.l7 + dldpi.l7 + dlmspu
s.l7 + ddlgdp.l8 + dldpi.l8 + dlmspus.l8 + const + sd1 + sd2 + sd3 + exo1

            Estimate Std. Error t value Pr(>|t|)
ddlgdp.l1   0.209628   0.174545    1.20   0.2312
dldpi.l1    0.000594   0.154203    0.00   0.9969
dlmspus.l1 -0.199315   0.074437   -2.68   0.0080 **
ddlgdp.l2   0.035355   0.172259    0.21   0.8376
dldpi.l2    0.214099   0.160330    1.34   0.1833
dlmspus.l2  0.023292   0.076548    0.30   0.7612
ddlgdp.l3  -0.114581   0.177079   -0.65   0.5183
dldpi.l3   -0.066513   0.160424   -0.41   0.6789
dlmspus.l3  0.283568   0.077111    3.68   0.0003 ***
ddlgdp.l4  -0.291408   0.171898   -1.70   0.0916 .
dldpi.l4   -0.078266   0.169241   -0.46   0.6443
dlmspus.l4  0.181064   0.078052    2.32   0.0214 *
ddlgdp.l5  -0.015488   0.155568   -0.10   0.9208
dldpi.l5    0.024607   0.176444    0.14   0.8892
dlmspus.l5  0.171951   0.078344    2.19   0.0293 *
ddlgdp.l6  -0.091163   0.155802   -0.59   0.5591
dldpi.l6    0.243643   0.169100    1.44   0.1512
dlmspus.l6 -0.000945   0.077225   -0.01   0.9903
ddlgdp.l7   0.022425   0.170275    0.13   0.8954
dldpi.l7   -0.007747   0.173141   -0.04   0.9644
dlmspus.l7 -0.082470   0.077323   -1.07   0.2875
ddlgdp.l8  -0.078386   0.171060   -0.46   0.6473
dldpi.l8    0.054613   0.193172    0.28   0.7777
dlmspus.l8 -0.088016   0.077650   -1.13   0.2584
const       0.004575   0.006177    0.74   0.4598
sd1         0.007663   0.006270    1.22   0.2231
sd2        -0.015970   0.006152   -2.60   0.0101 *
sd3         0.004779   0.006318    0.76   0.4503
exo1        0.050071   0.136156    0.37   0.7135
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1


Residual standard error: 0.0271 on 198 degrees of freedom
Multiple R-Squared: 0.245,      Adjusted R-squared: 0.138
F-statistic:  2.3 on 28 and 198 DF,  p-value: 0.00051
```

From the output, we can observe that no exogenous term is positive and significant, so the model is valid. From the output, we can further observe that ddlgdp (differenced seasonally differenced log gdp) is positively impacted by the 1st lag of dldpi (differenced log disposable personal income) and 2nd lag of dlmspus (differenced log median sales prices of houses); and negatively impacted by its own lag 4 and 8.

dldpi is positively impacted by its own lag (1, 3 and 8, overall positive), lag 2 and lag 6 of ddlgdp, and lag 7 of dlmspus. dlmspus is positively impacted by its own lag (1, 3 and 8, overall positive), lag 2 and lag 6 of ddlgdp, and lag 7 of dlmspus. On the other hand, dldpi is negatively impacted by the seasonal dummy 3. dlmspus is positively impacted by its own lag (1, 3, 4 and 5, overall positive). dlmspus is negatively impacted by the seasonal dummy 2.

Below is the plotting of fitted values and residuals from VECM (Johansen) model

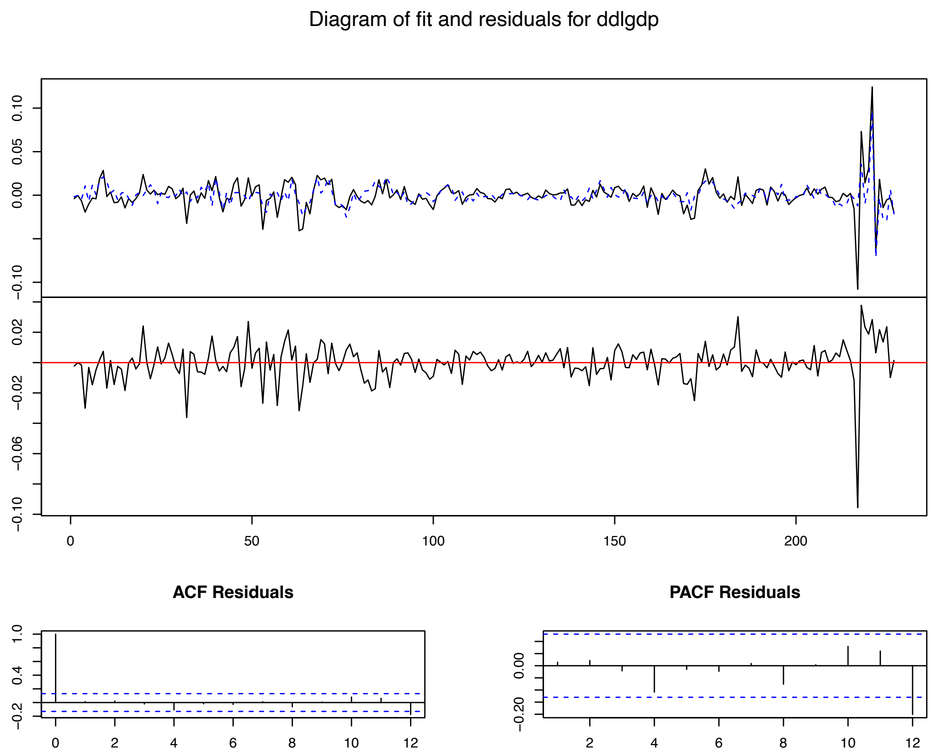


Diagram of fit and residuals for dldpi

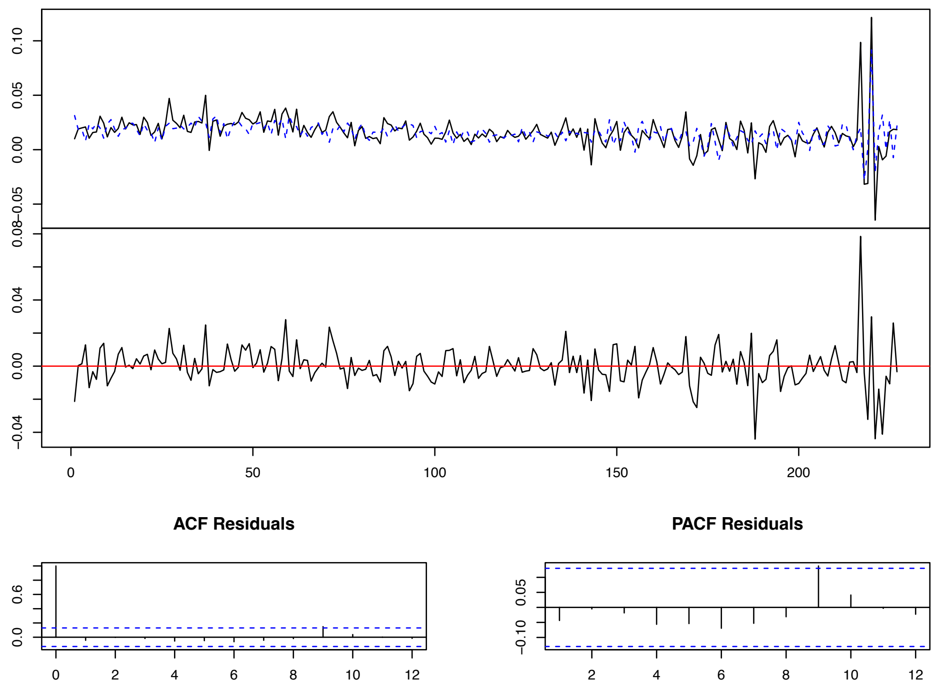
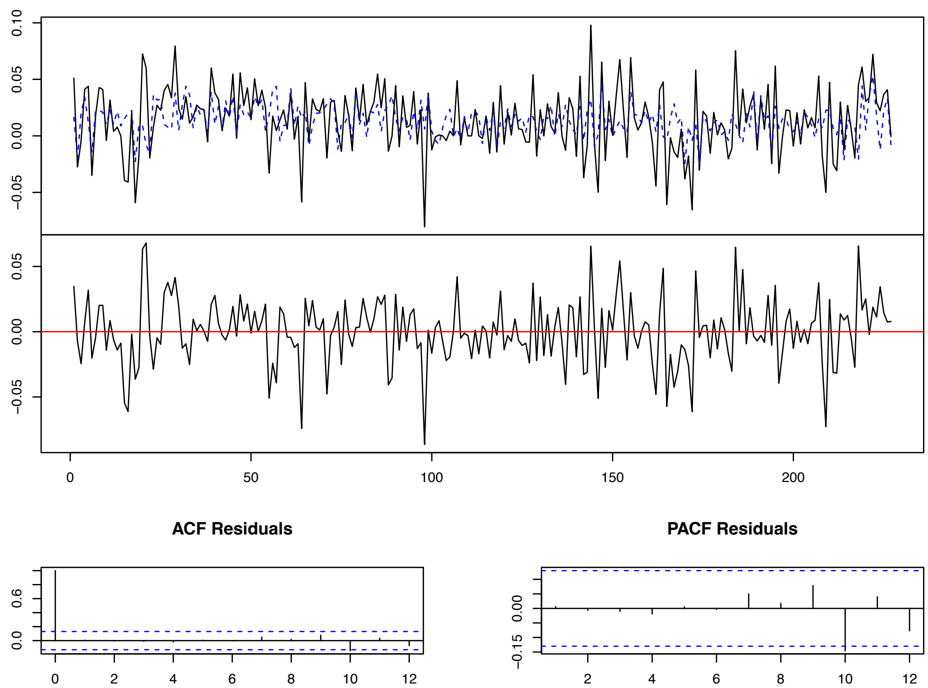


Diagram of fit and residuals for dlmspus



All three plots indicate good performance of the model, as fitted values go along the actual values and none of the residuals seem to be serially correlated.

Granger causality test

Deos lmspus granger cause dlgdp?

```
Model 1: z@Z0[, reg.number] ~ constant + sd1 + sd2 + sd3 + dlgdp.dl1 +
    ldpi.dl1 + lmspus.dl1 + dlgdp.dl2 + ldpi.dl2 + lmspus.dl2 +
    dlgdp.dl3 + ldpi.dl3 + lmspus.dl3 + dlgdp.dl4 + ldpi.dl4 +
    lmspus.dl4 + dlgdp.dl5 + ldpi.dl5 + lmspus.dl5 + dlgdp.dl6 +
    ldpi.dl6 + lmspus.dl6 + dlgdp.dl7 + ldpi.dl7 + lmspus.dl7 +
    dlgdp.l8 + ldpi.l8 + lmspus.l8 + trend.l8 - 1
Model 2: z@Z0[, reg.number] ~ sd1 + sd2 + sd3 + dlgdp.dl1 + ldpi.dl1 +
    dlgdp.dl2 + ldpi.dl2 + dlgdp.dl3 + ldpi.dl3 + dlgdp.dl4 +
    ldpi.dl4 + dlgdp.dl5 + ldpi.dl5 + dlgdp.dl6 + ldpi.dl6 +
    dlgdp.dl7 + ldpi.dl7 + dlgdp.l8 + ldpi.l8 + constant - 1
  Res.Df    RSS Df Sum of Sq    F  Pr(>F)
1    199 0.0332
2    208 0.0381 -9  -0.00492 3.28 0.00096 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

as p-value < 0.05, we reject the null hypothesis. Hence, there is Granger causality, so lmspus granger causes dlgdp.

Does ldpi granger cause dlgdp?

```
Model 1: z@Z0[, reg.number] ~ constant + sd1 + sd2 + sd3 + dlgdp.dl1 +
    ldpi.dl1 + lmspus.dl1 + dlgdp.dl2 + ldpi.dl2 + lmspus.dl2 +
    dlgdp.dl3 + ldpi.dl3 + lmspus.dl3 + dlgdp.dl4 + ldpi.dl4 +
    lmspus.dl4 + dlgdp.dl5 + ldpi.dl5 + lmspus.dl5 + dlgdp.dl6 +
    ldpi.dl6 + lmspus.dl6 + dlgdp.dl7 + ldpi.dl7 + lmspus.dl7 +
    dlgdp.l8 + ldpi.l8 + lmspus.l8 + trend.l8 - 1
Model 2: z@Z0[, reg.number] ~ sd1 + sd2 + sd3 + dlgdp.dl1 + lmspus.dl1 +
    dlgdp.dl2 + lmspus.dl2 + dlgdp.dl3 + lmspus.dl3 + dlgdp.dl4 +
    lmspus.dl4 + dlgdp.dl5 + lmspus.dl5 + dlgdp.dl6 + lmspus.dl6 +
    dlgdp.dl7 + lmspus.dl7 + dlgdp.l8 + lmspus.l8 + constant -
    1
  Res.Df    RSS Df Sum of Sq    F     Pr(>F)
1    199 0.0332
2    208 0.0421 -9  -0.00897 5.98 0.00000021 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

As p-value < 0.05, we reject the null hypothesis and hence, there is Granger causality. So ldpi granger causes dlgdp.

Does dlgdp granger cause ldpi?

```
Model 1: z@Z0[, reg.number] ~ constant + sd1 + sd2 + sd3 + dlgdp.dl1 +
    ldpi.dl1 + lmspus.dl1 + dlgdp.dl2 + ldpi.dl2 + lmspus.dl2 +
    dlgdp.dl3 + ldpi.dl3 + lmspus.dl3 + dlgdp.dl4 + ldpi.dl4 +
    lmspus.dl4 + dlgdp.dl5 + ldpi.dl5 + lmspus.dl5 + dlgdp.dl6 +
    ldpi.dl6 + lmspus.dl6 + dlgdp.dl7 + ldpi.dl7 + lmspus.dl7 +
    dlgdp.l8 + ldpi.l8 + lmspus.l8 + trend.l8 - 1
Model 2: z@Z0[, reg.number] ~ sd1 + sd2 + sd3 + ldpi.dl1 + lmspus.dl1 +
    ldpi.dl2 + lmspus.dl2 + ldpi.dl3 + lmspus.dl3 + ldpi.dl4 +
    lmspus.dl4 + ldpi.dl5 + lmspus.dl5 + ldpi.dl6 + lmspus.dl6 +
    ldpi.dl7 + lmspus.dl7 + ldpi.l8 + lmspus.l8 + constant -
    1
  Res.Df    RSS Df Sum of Sq    F   Pr(>F)
1    199 0.0277
2    208 0.0329 -9  -0.00525 4.19 0.000057 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

as p-value < 0.05 we reject the null hypothesis. Hence, there is Granger causality, so dlgdp granger causes ldpi.

Does lmspus granger cause ldpi?

```
Model 1: z@Z0[, reg.number] ~ constant + sd1 + sd2 + sd3 + dlgdp.dl1 +
    ldpi.dl1 + lmspus.dl1 + dlgdp.dl2 + ldpi.dl2 + lmspus.dl2 +
    dlgdp.dl3 + ldpi.dl3 + lmspus.dl3 + dlgdp.dl4 + ldpi.dl4 +
    lmspus.dl4 + dlgdp.dl5 + ldpi.dl5 + lmspus.dl5 + dlgdp.dl6 +
    ldpi.dl6 + lmspus.dl6 + dlgdp.dl7 + ldpi.dl7 + lmspus.dl7 +
    dlgdp.l8 + ldpi.l8 + lmspus.l8 + trend.l8 - 1
Model 2: z@Z0[, reg.number] ~ sd1 + sd2 + sd3 + ldpi.dl1 + dlgdp.dl1 +
    ldpi.dl2 + dlgdp.dl2 + ldpi.dl3 + dlgdp.dl3 + ldpi.dl4 +
    dlgdp.dl4 + ldpi.dl5 + dlgdp.dl5 + ldpi.dl6 + dlgdp.dl6 +
    ldpi.dl7 + dlgdp.dl7 + ldpi.l8 + dlgdp.l8 + constant - 1
  Res.Df    RSS Df Sum of Sq    F Pr(>F)
1    199 0.0277
2    208 0.0298 -9  -0.00211 1.69  0.094 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

as p-value > 0.05, we fail to reject the null hypothesis. Hence, there is no Granger causality and lmspus does not granger cause ldpi.

Does dlgdp granger cause lmspus?

```
Model 1: z@Z0[, reg.number] ~ constant + sd1 + sd2 + sd3 + dlgdp.dl1 +
    ldpi.dl1 + lmspus.dl1 + dlgdp.dl2 + ldpi.dl2 + lmspus.dl2 +
    dlgdp.dl3 + ldpi.dl3 + lmspus.dl3 + dlgdp.dl4 + ldpi.dl4 +
    lmspus.dl4 + dlgdp.dl5 + ldpi.dl5 + lmspus.dl5 + dlgdp.dl6 +
    ldpi.dl6 + lmspus.dl6 + dlgdp.dl7 + ldpi.dl7 + lmspus.dl7 +
    dlgdp.l8 + ldpi.l8 + lmspus.l8 + trend.l8 - 1
Model 2: z@Z0[, reg.number] ~ sd1 + sd2 + sd3 + lmspus.dl1 + ldpi.dl1 +
    lmspus.dl2 + ldpi.dl2 + lmspus.dl3 + ldpi.dl3 + lmspus.dl4 +
    ldpi.dl4 + lmspus.dl5 + ldpi.dl5 + lmspus.dl6 + ldpi.dl6 +
    lmspus.dl7 + ldpi.dl7 + lmspus.l8 + ldpi.l8 + constant -
    1
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1    199 0.136
2    208 0.144 -9  -0.00772 1.25   0.27
```

as p-value > 0.05, we fail to reject the null hypothesis and hence there is no Granger causality. So dlgdp does not granger cause lmspus.

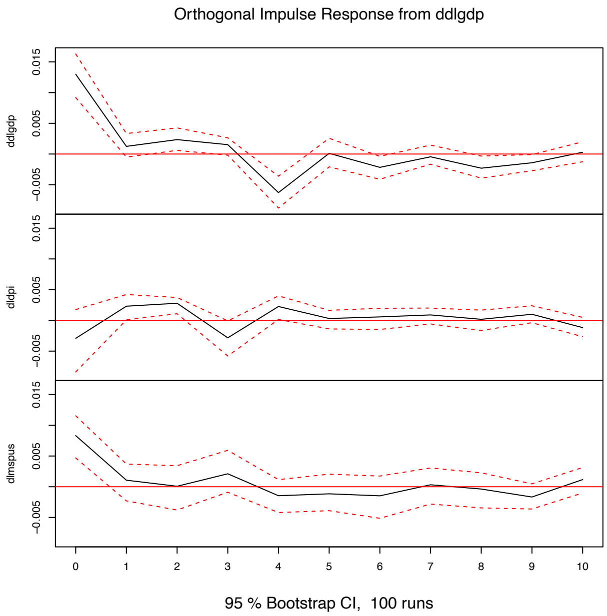
Does ldpi granger cause lmspus?

```
Model 1: z@Z0[, reg.number] ~ constant + sd1 + sd2 + sd3 + dlgdp.dl1 +
    ldpi.dl1 + lmspus.dl1 + dlgdp.dl2 + ldpi.dl2 + lmspus.dl2 +
    dlgdp.dl3 + ldpi.dl3 + lmspus.dl3 + dlgdp.dl4 + ldpi.dl4 +
    lmspus.dl4 + dlgdp.dl5 + ldpi.dl5 + lmspus.dl5 + dlgdp.dl6 +
    ldpi.dl6 + lmspus.dl6 + dlgdp.dl7 + ldpi.dl7 + lmspus.dl7 +
    dlgdp.l8 + ldpi.l8 + lmspus.l8 + trend.l8 - 1
Model 2: z@Z0[, reg.number] ~ sd1 + sd2 + sd3 + lmspus.dl1 + dlgdp.dl1 +
    lmspus.dl2 + dlgdp.dl2 + lmspus.dl3 + dlgdp.dl3 + lmspus.dl4 +
    dlgdp.dl4 + lmspus.dl5 + dlgdp.dl5 + lmspus.dl6 + dlgdp.dl6 +
    lmspus.dl7 + dlgdp.dl7 + lmspus.l8 + dlgdp.l8 + constant -
    1
  Res.Df   RSS Df Sum of Sq    F Pr(>F)
1    199 0.136
2    208 0.149 -9   -0.0128 2.07  0.033 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

as p-value < 0.05 we reject the null hypothesis. Hence, there is Granger causality, so, ldpi granger causes lmspus.

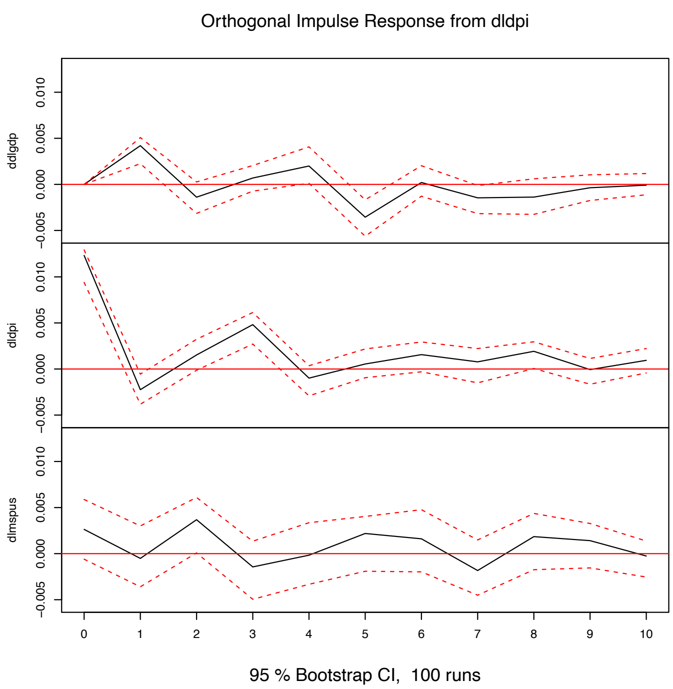
Examining Impulse Response

After checking for Granger causality, we formulated the Orthogonal Impulse Response for all the three variables. Below is the Orthogonal Impulse Response from ddlgdp.



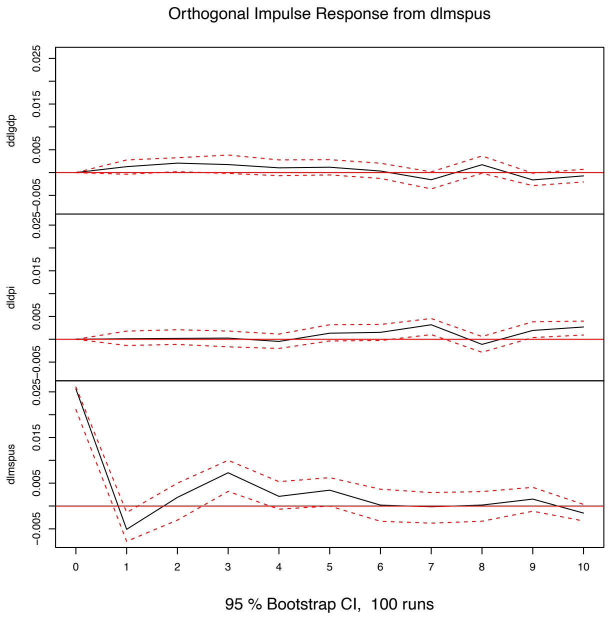
From the above, we can observe that, if there is a positive shock in differenced in seasonally differenced log gdp, its value will decrease in the next period from its statistically significant initial rise and then in the 4th quarter there will be negative shock. Shock on ddlgdp does not have significant impact on dldpi, except between quarter 1 & 2, there is a positive impact. dlmpsus will decrease and then merge to equilibrium. Shock on ddlgdp does not have significant impact on dlmspus as well, except statistically significant initial rise.

Below is the Orthogonal Impulse Response from dldpi.



From the above, we can observe that, if there is a positive shock in dldpi, its initial rise and rise in 3rd quarter is statistically significant. For ddlgdp rise in 1st quarter and fall in fifth quarter is statistically significant. For dlmpsus the impact of shock from dldpi is not significant.

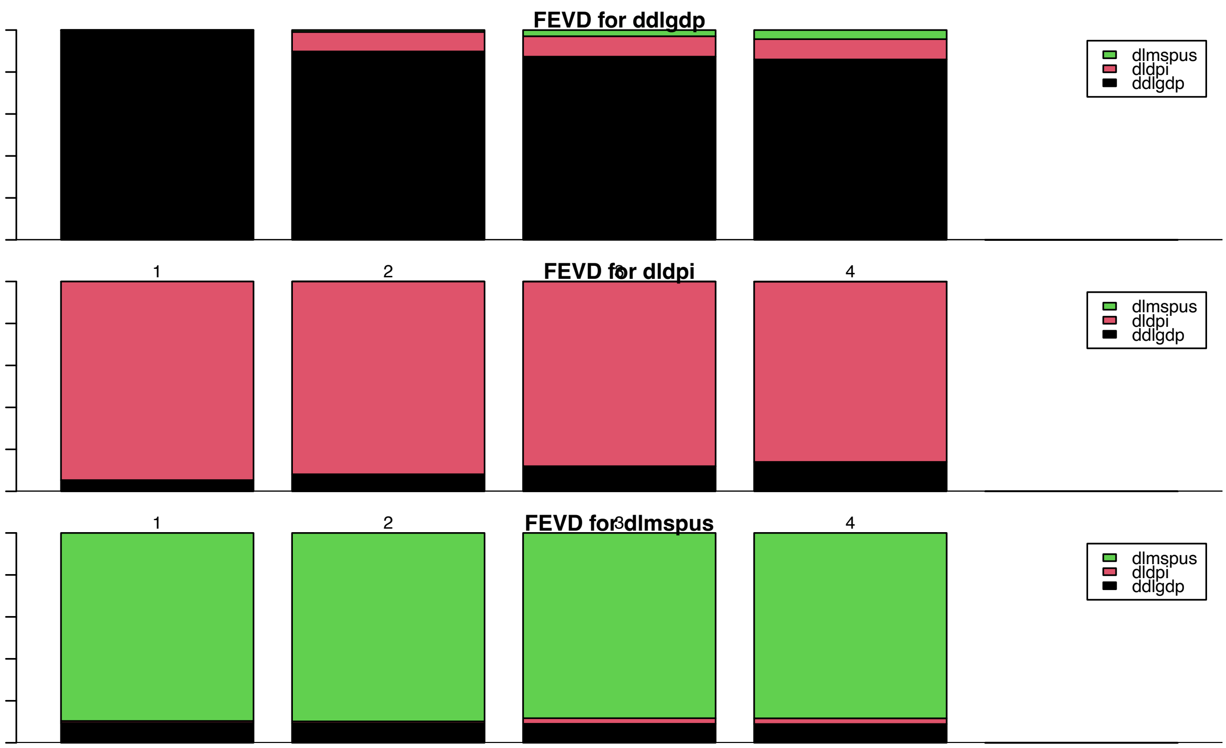
Below is the Orthogonal Impulse Response from dlmpsus.



From the above, we can observe that, if there is a positive shock in dlmspus, no significant impact on ddlgdp or dldpi but for its own, it has a significant rise in initial and in 3rd quarter.

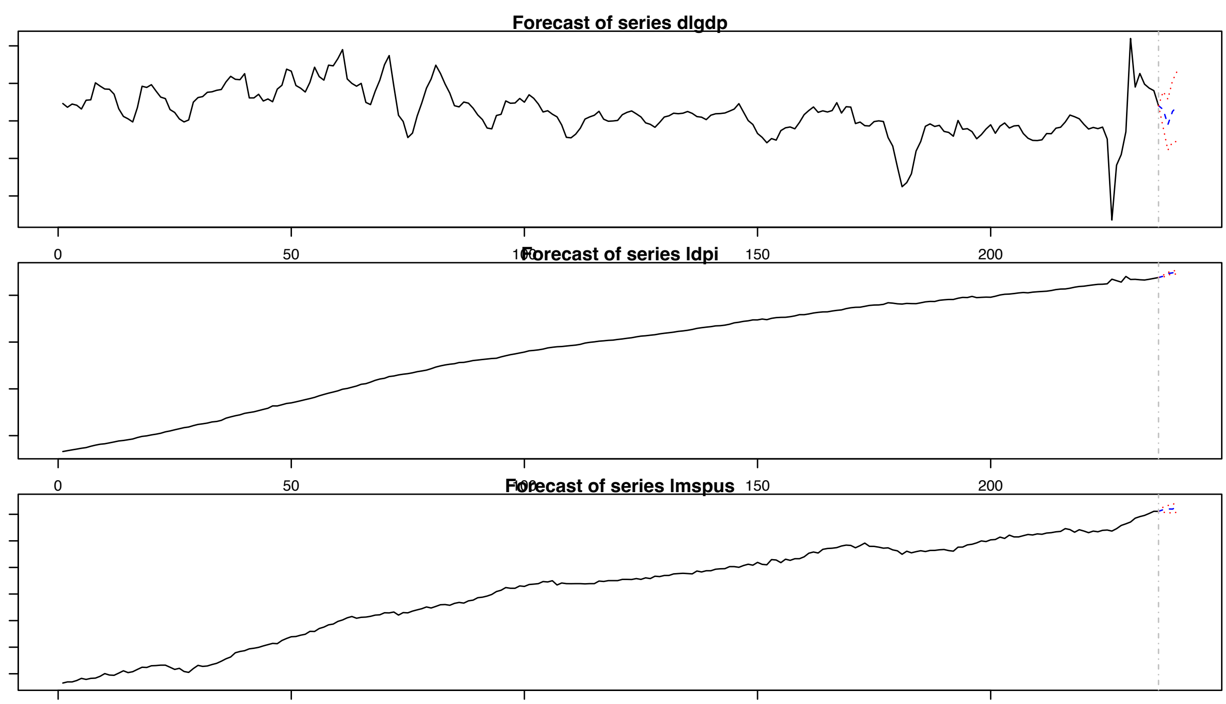
Variance Decompositions

Variance decompositions explains the contribution of each variable to predicting a variable. From the graph we can observe that the predictability of a variable by other variable increases over time. Variability of all three variables are mostly explained by itself and slightly by other variables.

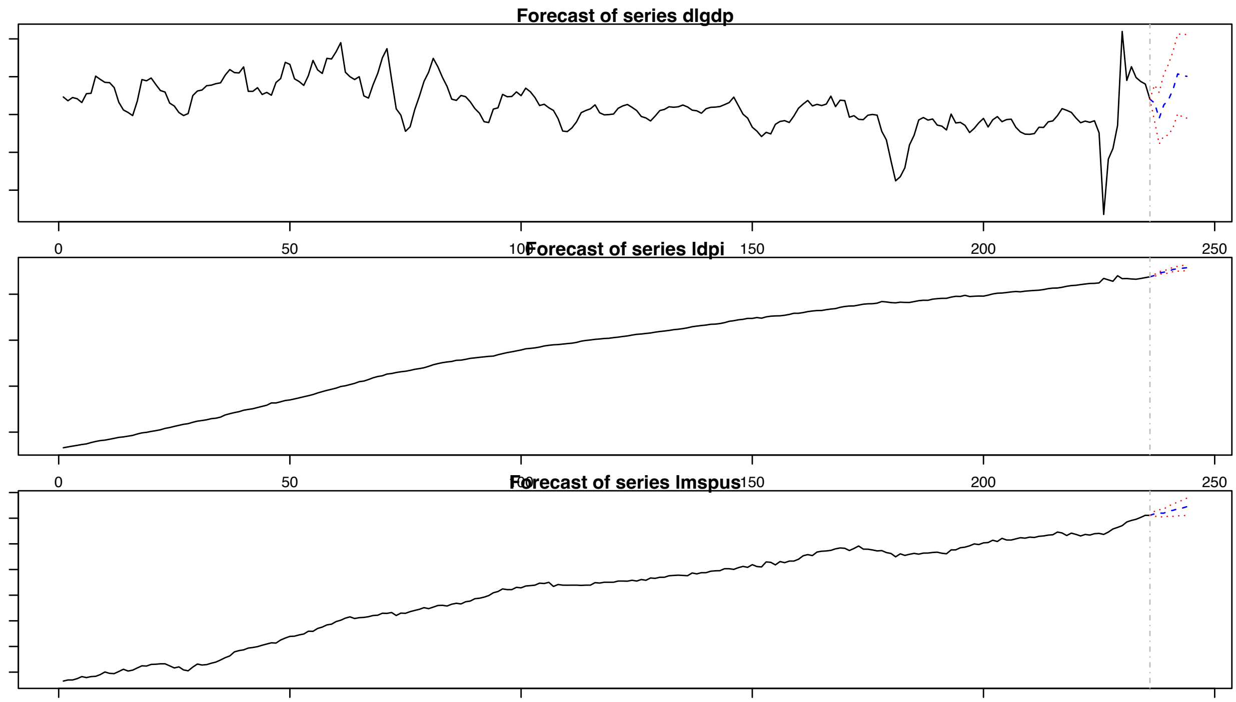


Forecasting

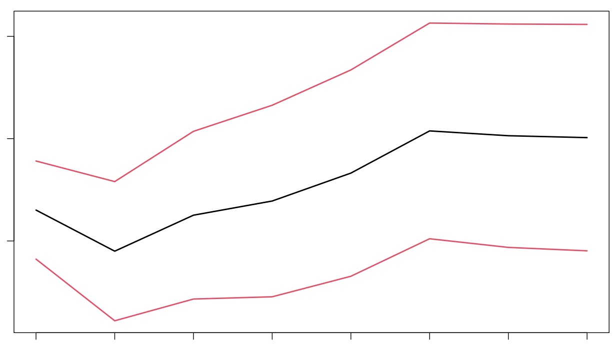
The following is the plotting of forecasting with VECM (Johansen) 4 period ahead.



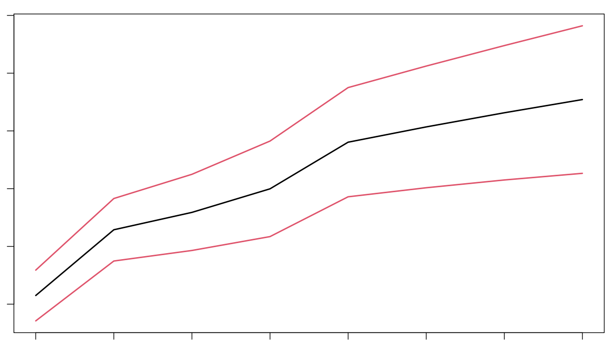
The following is the plotting of forecasting with VECM (Johansen) 8 period ahead.



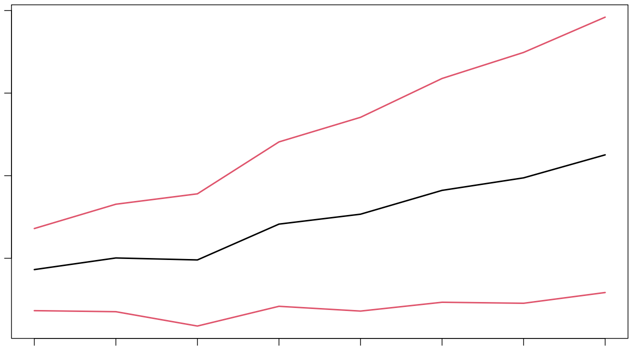
Below is the only forecasting plotting for dlgdp.



Below is the only forecasting plotting for ldpi.



Below is the only forecasting plotting for lmspus.



From the above plots of forecasting of variables for 4 and 8 periods, we can observe that dlgdp has large confidence interval with compared to ldpi and lmspus variables, but the volatility is nicely captured.

Serial Correlation Test

After forecasting, we now test for serial correlation for residuals. We use BoX-Ljung Q Statistic to test for serial correlation. The null hypothesis is 'no serial correlation'.

The following is the BoX-Ljung Q Statistic for ddlgdp

```
	Box-Ljung test

data:  resi
X-squared = 17, df = 20, p-value = 0.7

> blt = rep(0,20)
> for (i in 1:20){
+   b = Box.test(resi,lag = i, type="Ljung-Box")
+   blt[i]=b$p.value
+ }
> blt
 [1] 0.826 0.926 0.968 0.556 0.686 0.774 0.855 0.832 0.892 0.829 0.818 0.291 0.361 0.336 0.401 0.417 0.486 0.536
[19] 0.598 0.660
```

The following is the BoX-Ljung Q Statistic for dldpi

```
	Box-Ljung test

data:  resi
X-squared = 11, df = 20, p-value = 0.9

> blt = rep(0,20)
> for (i in 1:20){
+   b = Box.test(resi,lag = i, type="Ljung-Box")
+   blt[i]=b$p.value
+ }
> blt
 [1] 0.508 0.803 0.916 0.877 0.881 0.846 0.872 0.920 0.488 0.551 0.640 0.714 0.782 0.829 0.854 0.830 0.874
[18] 0.907 0.915 0.939
```

The following is the BoX-Ljung Q Statistic for dlmspus

```
	Box-Ljung test

data:  resi
X-squared = 18, df = 20, p-value = 0.6

> blt = rep(0,20)
> for (i in 1:20){
+   b = Box.test(resi,lag = i, type="Ljung-Box")
+   blt[i]=b$p.value
+ }
> blt
 [1] 0.925 0.991 0.998 0.998 1.000 1.000 0.998 0.999 0.987 0.701 0.756 0.723 0.773 0.687 0.742 0.747 0.782
[18] 0.703 0.756 0.564
```

We observe from the above three output that the p-value is greater than 0.05 and hence we fail to reject the null hypothesis and we can say that, considering the 20 lags for residuals, we do not observe serial correlation in the residuals of these three variables.

Volatility Test

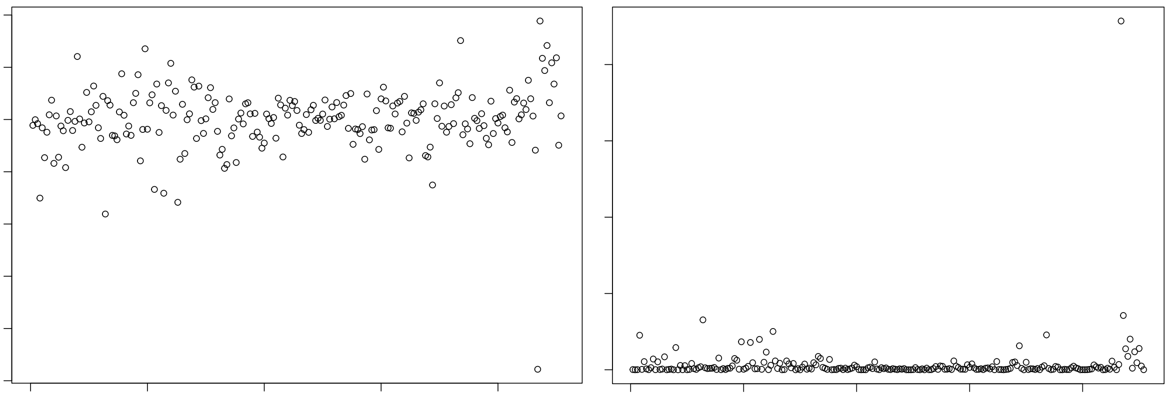
After testing for serial correlation, we now test for volatility using ARCH effects.
The null hypothesis for ARCH test is ‘no conditional heteroskedasticity.’

```
	ARCH (multivariate)

data:  Residuals of VAR object varf
Chi-squared = 168, df = 108, p-value = 0.0002
```

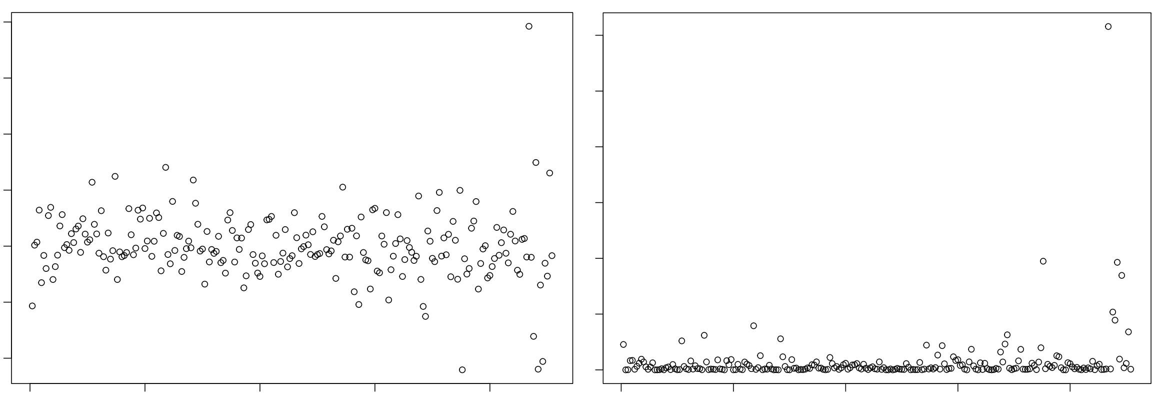
From the above output, we see that the p-value is less than 0.05 and hence, we reject the null hypothesis. This means that we have conditional heteroskedasticity in the model.

After the test for volatility, we plot the residuals and square residuals of the variables which seemingly confirms the volatility test result for ddlgdp and dldpi, but for dlmspus, it looks like the residuals of dlmspus does not suffer from conditional heteroskedasticity.

1. ddlgdp



2. dldpi



3. dlmpsus