VAR model

ewayagi

Tue 12, July 2022

Loading the libraries

```
library(AER)
library(tseries)
library(vars)
library(TSstudio)
library(forecast)
library(tidyverse)
library(mFilter)
library(dynlm)
library(fUnitRoots)
library(dplyr)
library(ggplot2)
library(plotly)
```

Loading the data

```
data("USMacroG")
#Assigning the data a new name
vardata = USMacroG
#convert the data to a data.frame
vardataDf = as.data.frame(vardata)
#confirm the class of the vardata
class(vardataDf)
```

```
## [1] "data.frame"
```

```
#call the variable names
variable.names(vardataDf)
```

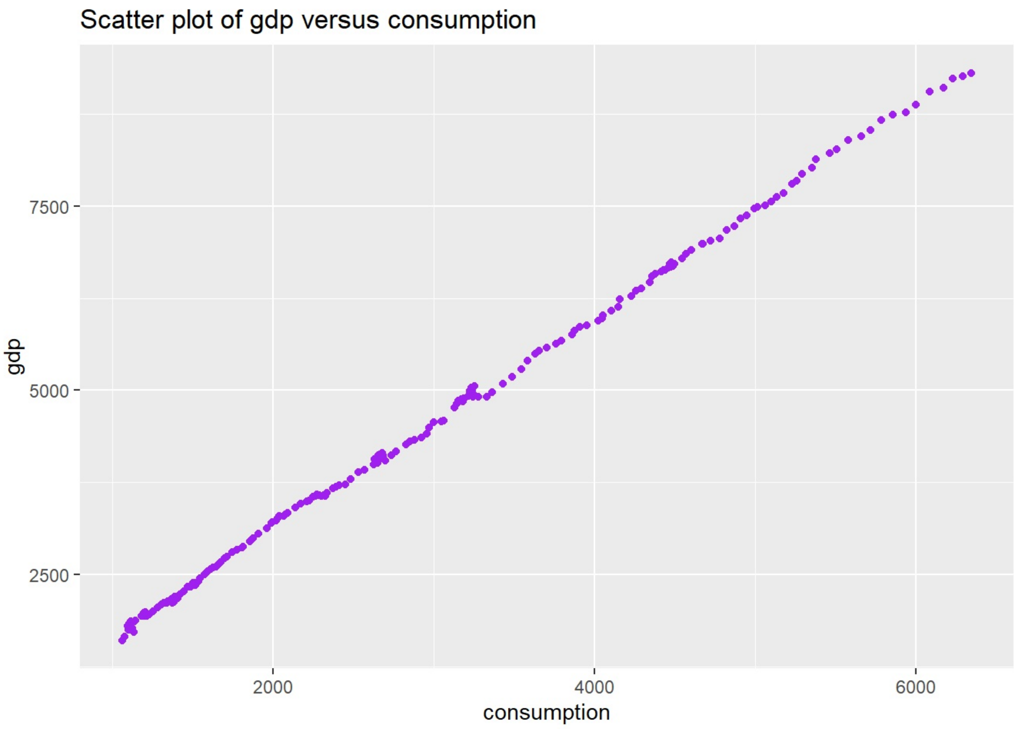
```
##  [1] "gdp"         "consumption" "invest"      "government"  "dpi"
##  [6] "cpi"         "m1"          "tbill"       "unemp"       "population"
## [11] "inflation"   "interest"
```

Defining the time series variables

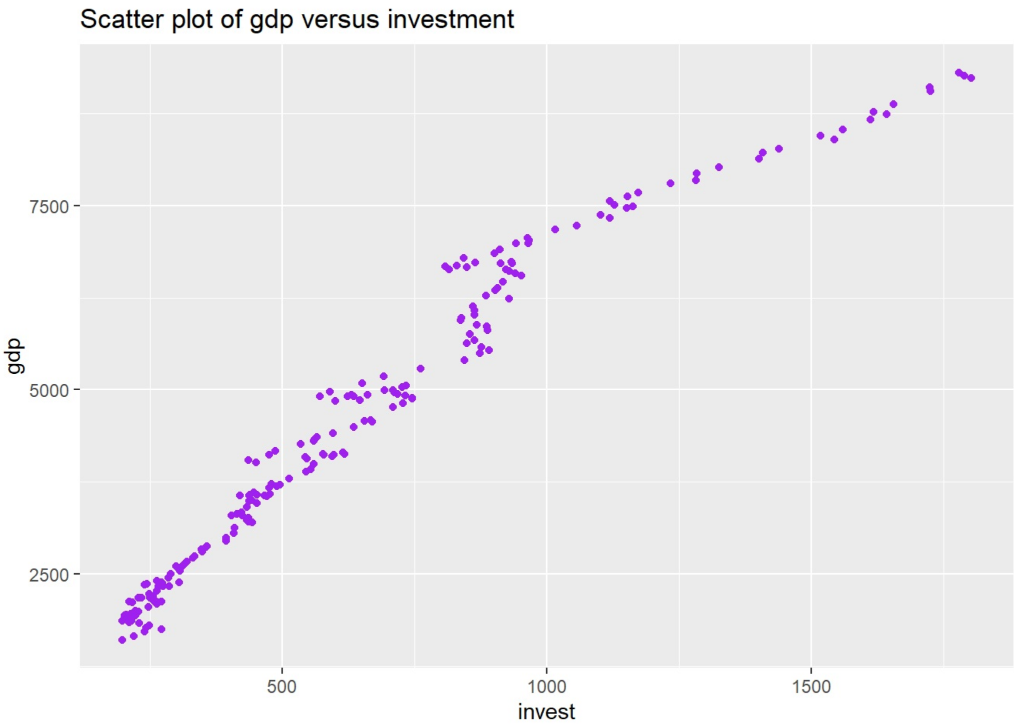
```
tsgdp = ts(vardataDf$gdp, start = c(1950,1,4), frequency = 4)
tsinvest = ts(vardataDf$invest, start = c(1950,1,4), frequency = 4)
tsconsumption = ts(vardataDf$consumption, start = c(1950,1,4), frequency = 4)
```

Visualizing the variables

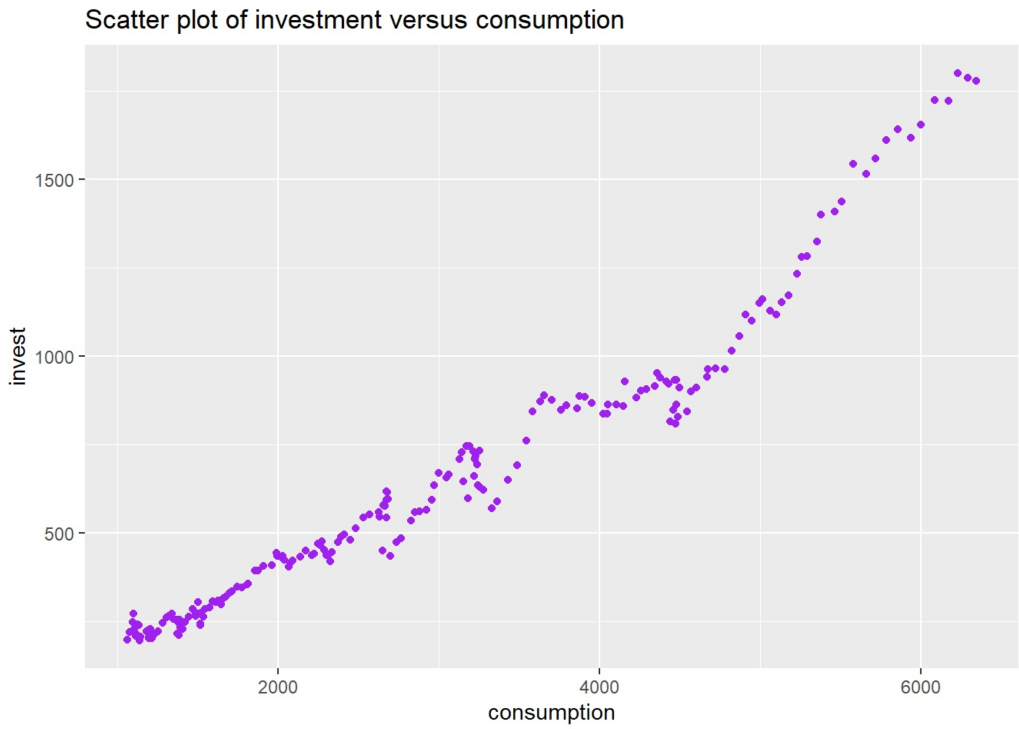
```
vardataDf %>%
  ggplot(aes(consumption,gdp))+
  geom_point(color="purple")+
  labs(title="Scatter plot of gdp versus consumption")
```




```
vardataDf %>%
  ggplot(aes(invest,gdp))+
  geom_point(color="purple")+
  labs(title="Scatter plot of gdp versus investment")
```

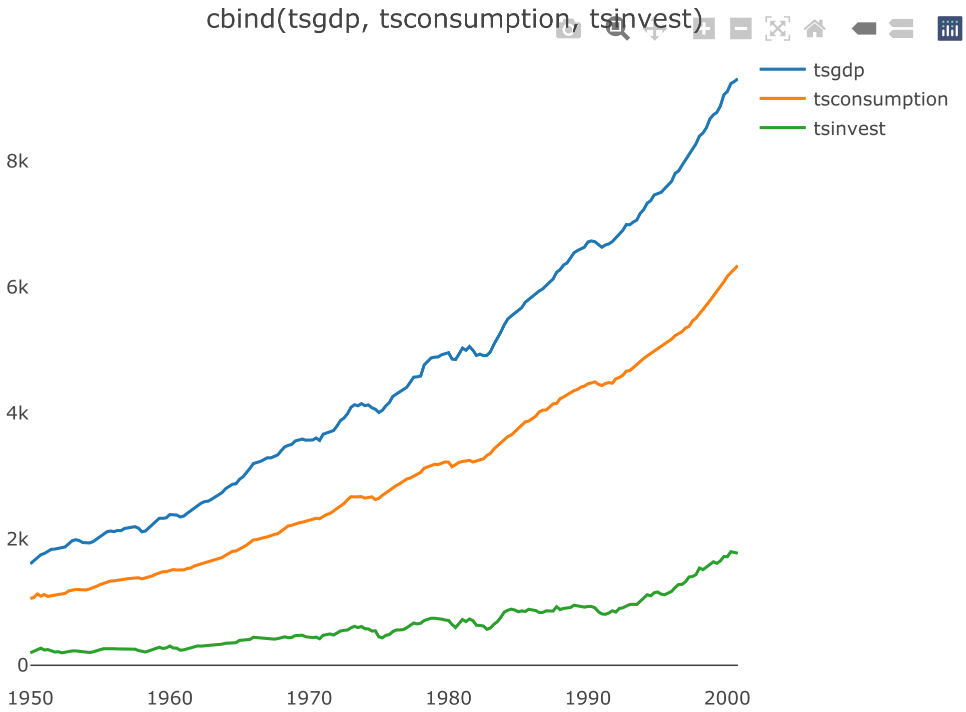



```
vardataDf %>%
  ggplot(aes(consumption,invest))+
  geom_point(color="purple")+
  labs(title="Scatter plot of investment versus consumption")
```



Testing for stationarity Visual plots

```
ts_plot(cbind(tsgdp, tsconsumption, tsinvest))
```




```
#OR
#plot.ts(cbind(tsgdp, tsconsumption, tsinvest))
```

Statistical tests Phillips-Perron Unit Root Test for Stationary

```
pp.test(tsgdp)
```

```
## Warning in pp.test(tsgdp): p-value greater than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  tsgdp
## Dickey-Fuller Z(alpha) = 1.6531, Truncation lag parameter = 4, p-value
## = 0.99
## alternative hypothesis: stationary
```

```
pp.test(tsinvest)
```

```
## Warning in pp.test(tsinvest): p-value greater than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  tsinvest
## Dickey-Fuller Z(alpha) = 0.27761, Truncation lag parameter = 4, p-value
## = 0.99
## alternative hypothesis: stationary
```

```
pp.test(tsconsumption)
```

```
## Warning in pp.test(tsconsumption): p-value greater than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  tsconsumption
## Dickey-Fuller Z(alpha) = 2.4416, Truncation lag parameter = 4, p-value
## = 0.99
## alternative hypothesis: stationary
```

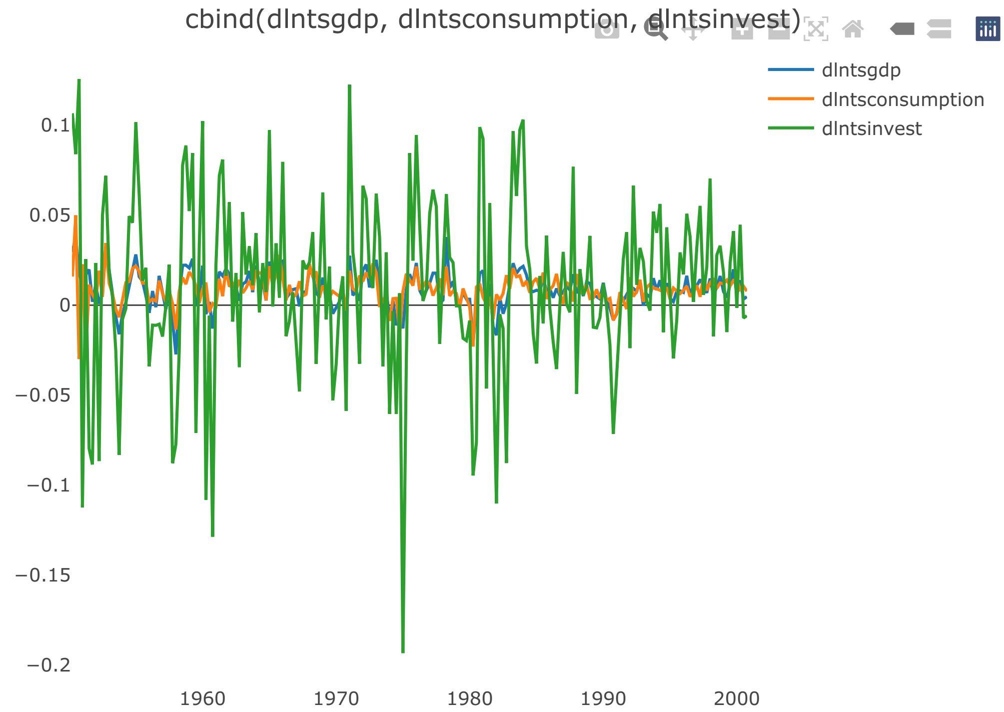
```
#All var are non-stationary
```

Log transformation and differencing for stationarity

```
dlntsgdp = diff(log(tsgdp))
dlntsconsumption = diff(log(tsconsumption))
dlntsinvest = diff(log(tsinvest))
```

Repeat the tests

```
ts_plot(cbind(dlntsgdp, dlntsconsumption, dlntsinvest))
```




```
#Phillips-Perron Unit Root Test

pp.test(dlntsgdp)
```

```
## Warning in pp.test(dlntsgdp): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  dlntsgdp
## Dickey-Fuller Z(alpha) = -129.68, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
pp.test(dlntsinvest)
```

```
## Warning in pp.test(dlntsinvest): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  dlntsinvest
## Dickey-Fuller Z(alpha) = -171.56, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
pp.test(dlntsconsumption)
```

```
## Warning in pp.test(dlntsconsumption): p-value smaller than printed p-value
```

```
##
##  Phillips-Perron Unit Root Test
##
## data:  dlntsconsumption
## Dickey-Fuller Z(alpha) = -226.99, Truncation lag parameter = 4, p-value
## = 0.01
## alternative hypothesis: stationary
```

```
#All variables are stationary
```

VAR model

```
#Determining the optimal lag
variables = cbind(dlntsgdp, dlntsinvest, dlntsconsumption)

#change the variable names
colnames(variables) = cbind("GDP" , "INVESTMENT" , "CONSUMPTION")

#selecting the lags
lags = VARselect(variables, lag.max = 10, type = "const")

#view the selection
lags$selection
```

```
## AIC(n)  HQ(n)  SC(n) FPE(n)
##      1      1      1      1
```

```
#the VAR model
varmodel = VAR(variables, p = 1, type = "const", season = NULL,
               exog = NULL, ic = "AIC")
summary(varmodel)
```

```
##
## VAR Estimation Results:
## =========================
## Endogenous variables: GDP, INVESTMENT, CONSUMPTION
## Deterministic variables: const
## Sample size: 202
## Log Likelihood: 1861.81
## Roots of the characteristic polynomial:
## 0.4066 0.3685 0.3685
## Call:
## VAR(y = variables, p = 1, type = "const", exogen = NULL, ic = "AIC")
##
##
## Estimation results for equation GDP:
## ====================================
```

```
## GDP = GDP.l1 + INVESTMENT.l1 + CONSUMPTION.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## GDP.l1         0.1189977  0.1434974   0.829  0.40795
## INVESTMENT.l1  0.0211950  0.0235252   0.901  0.36871
## CONSUMPTION.l1 0.2761822  0.1038485   2.659  0.00847 **
## const          0.0048385  0.0009911   4.882 2.16e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.00915 on 198 degrees of freedom
## Multiple R-Squared: 0.156,   Adjusted R-squared: 0.1432
## F-statistic:  12.2 on 3 and 198 DF,  p-value: 2.323e-07
##
##
## Estimation results for equation INVESTMENT:
## ===========================================
## INVESTMENT = GDP.l1 + INVESTMENT.l1 + CONSUMPTION.l1 + const
##
##                  Estimate Std. Error t value Pr(>|t|)
## GDP.l1         -1.754601   0.660566  -2.656  0.00855 **
## INVESTMENT.l1   0.261679   0.108294   2.416  0.01658 *
## CONSUMPTION.l1  3.694908   0.478049   7.729 5.33e-13 ***
## const          -0.009917   0.004562  -2.174  0.03091 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.04212 on 198 degrees of freedom
## Multiple R-Squared: 0.2865,  Adjusted R-squared: 0.2757
## F-statistic: 26.51 on 3 and 198 DF,  p-value: 1.858e-14
##
##
## Estimation results for equation CONSUMPTION:
## ============================================
## CONSUMPTION = GDP.l1 + INVESTMENT.l1 + CONSUMPTION.l1 + const
##
##                   Estimate Std. Error t value Pr(>|t|)
## GDP.l1          0.0122396  0.1328935   0.092   0.9267
## INVESTMENT.l1   0.0557503  0.0217868   2.559   0.0112 *
## CONSUMPTION.l1 -0.0550424  0.0961745  -0.572   0.5678
## const           0.0085539  0.0009179   9.320   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
##
## Residual standard error: 0.008474 on 198 degrees of freedom
## Multiple R-Squared: 0.09965, Adjusted R-squared: 0.086
## F-statistic: 7.304 on 3 and 198 DF,  p-value: 0.0001139
##
##
##
## Covariance matrix of residuals:
##                    GDP INVESTMENT CONSUMPTION
## GDP         8.373e-05  3.000e-04   4.772e-05
## INVESTMENT  3.000e-04  1.774e-03   9.295e-05
## CONSUMPTION 4.772e-05  9.295e-05   7.181e-05
##
## Correlation matrix of residuals:
##                GDP INVESTMENT CONSUMPTION
## GDP         1.0000     0.7783      0.6154
## INVESTMENT  0.7783     1.0000      0.2604
## CONSUMPTION 0.6154     0.2604      1.0000
```

Model diagnostic 1. Serial correlation

```
s_corr = serial.test(varmodel)
s_corr
```

```
##
##  Portmanteau Test (asymptotic)
##
## data:  Residuals of VAR object varmodel
## Chi-squared = 135.1, df = 135, p-value = 0.4814
```

```
#no serial correlation
```

2. Heteroscedasticity

```
HS = arch.test(varmodel)
HS
```

```
##
##  ARCH (multivariate)
##
## data:  Residuals of VAR object varmodel
## Chi-squared = 190.29, df = 180, p-value = 0.2852
```

```
#homoscedasticity
```

3. Normality

```
Norm = normality.test(varmodel)
Norm
```

```
## $JB
##
##  JB-Test (multivariate)
##
## data:  Residuals of VAR object varmodel
## Chi-squared = 1227.8, df = 6, p-value < 2.2e-16
##
##
## $Skewness
##
##  Skewness only (multivariate)
##
## data:  Residuals of VAR object varmodel
## Chi-squared = 149.06, df = 3, p-value < 2.2e-16
##
##
## $Kurtosis
##
##  Kurtosis only (multivariate)
##
## data:  Residuals of VAR object varmodel
## Chi-squared = 1078.7, df = 3, p-value < 2.2e-16
```

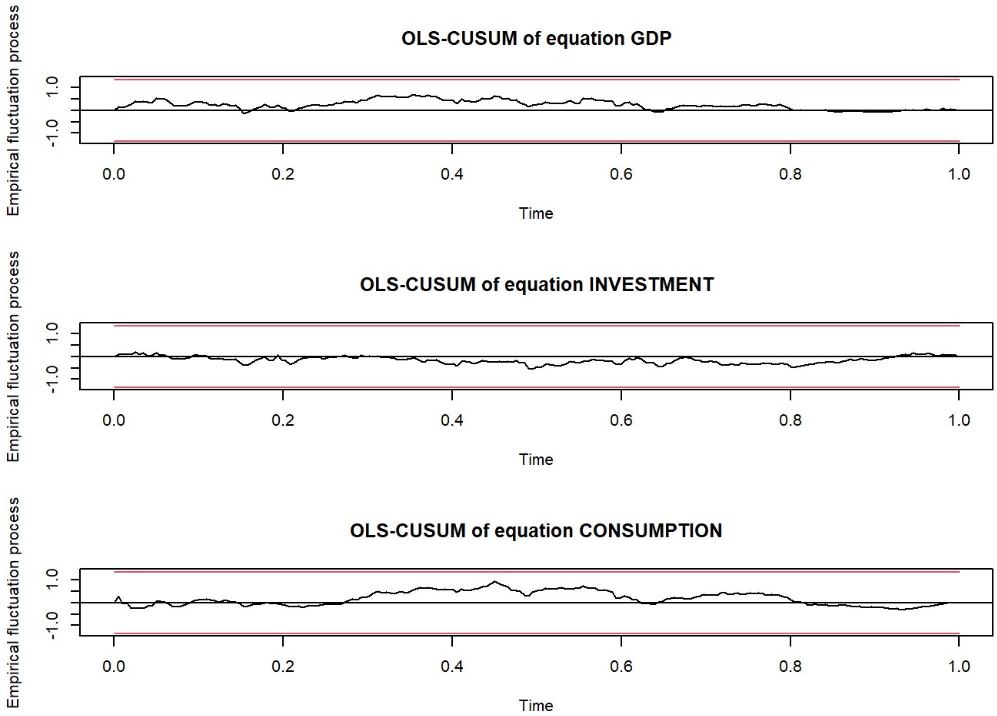
```
#non-normal
```

4. Structural break test

```
stability1 = stability(varmodel)
stability1
```

```
## $GDP
##
## Empirical Fluctuation Process: OLS-based CUSUM test
##
## Call: efp(formula = formula, data = data, type = type, h = h, dynamic = dynamic,
##     rescale = rescale)
##
##
## $INVESTMENT
##
## Empirical Fluctuation Process: OLS-based CUSUM test
##
## Call: efp(formula = formula, data = data, type = type, h = h, dynamic = dynamic,
##     rescale = rescale)
##
##
## $CONSUMPTION
##
## Empirical Fluctuation Process: OLS-based CUSUM test
##
## Call: efp(formula = formula, data = data, type = type, h = h, dynamic = dynamic,
##     rescale = rescale)
```

```
#plot
plot(stability1)
```




```
#Stable
```

Granger Causality

```
GrangerGDP<- causality(varmodel, cause = "GDP")
GrangerGDP
```

```
## $Granger
##
##  Granger causality H0: GDP do not Granger-cause INVESTMENT CONSUMPTION
##
## data:  VAR object varmodel
## F-Test = 3.8572, df1 = 2, df2 = 594, p-value = 0.02166
##
##
## $Instant
##
##  H0: No instantaneous causality between: GDP and INVESTMENT CONSUMPTION
##
## data:  VAR object varmodel
## Chi-squared = 89.055, df = 2, p-value < 2.2e-16
```

```
GrangerINVEST<- causality(varmodel, cause = "INVESTMENT")
GrangerINVEST
```

```
## $Granger
##
##  Granger causality H0: INVESTMENT do not Granger-cause GDP CONSUMPTION
##
## data:  VAR object varmodel
## F-Test = 3.6392, df1 = 2, df2 = 594, p-value = 0.02686
##
##
## $Instant
##
##  H0: No instantaneous causality between: INVESTMENT and GDP CONSUMPTION
##
## data:  VAR object varmodel
## Chi-squared = 81.952, df = 2, p-value < 2.2e-16
```

```
GrangerCONS<- causality(varmodel, cause = "CONSUMPTION")
GrangerCONS
```

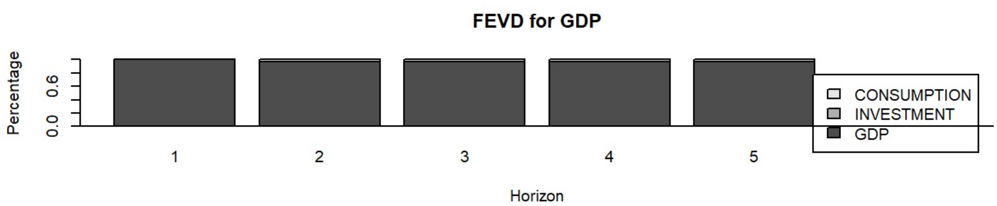
```
## $Granger
##
##  Granger causality H0: CONSUMPTION do not Granger-cause GDP INVESTMENT
##
## data:  VAR object varmodel
## F-Test = 44.158, df1 = 2, df2 = 594, p-value < 2.2e-16
##
##
## $Instant
##
##  H0: No instantaneous causality between: CONSUMPTION and GDP INVESTMENT
##
## data:  VAR object varmodel
## Chi-squared = 67.317, df = 2, p-value = 2.442e-15
```

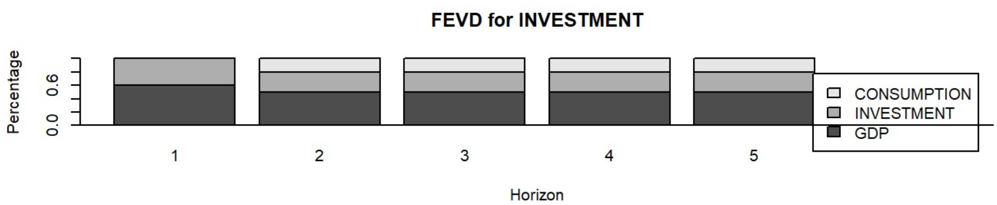
Variance decomposition

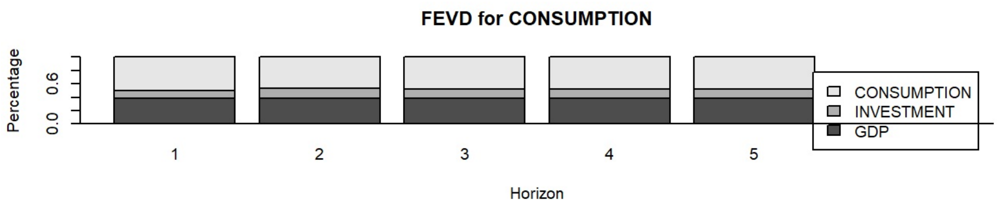
```
vard <- fevd(varmodel, n.ahead = 5)
vard
```

```
## $GDP
##            GDP  INVESTMENT CONSUMPTION
## [1,] 1.0000000 0.000000000  0.00000000
## [2,] 0.9710672 0.000666208  0.02826655
## [3,] 0.9670339 0.001810790  0.03115530
## [4,] 0.9650057 0.001852134  0.03314212
## [5,] 0.9647102 0.001935129  0.03335467
##
## $INVESTMENT
##            GDP INVESTMENT CONSUMPTION
## [1,] 0.6057964  0.3942036   0.0000000
## [2,] 0.5017052  0.2956175   0.2026773
## [3,] 0.4976116  0.3022800   0.2001084
## [4,] 0.4944859  0.3000419   0.2054722
## [5,] 0.4942798  0.3002895   0.2054308
##
## $CONSUMPTION
##            GDP INVESTMENT CONSUMPTION
## [1,] 0.3786564  0.1211585   0.5001851
## [2,] 0.3869532  0.1471543   0.4658926
## [3,] 0.3814076  0.1445370   0.4740554
## [4,] 0.3811394  0.1456173   0.4732434
## [5,] 0.3809031  0.1455227   0.4735742
```

```
plot(vard)
```







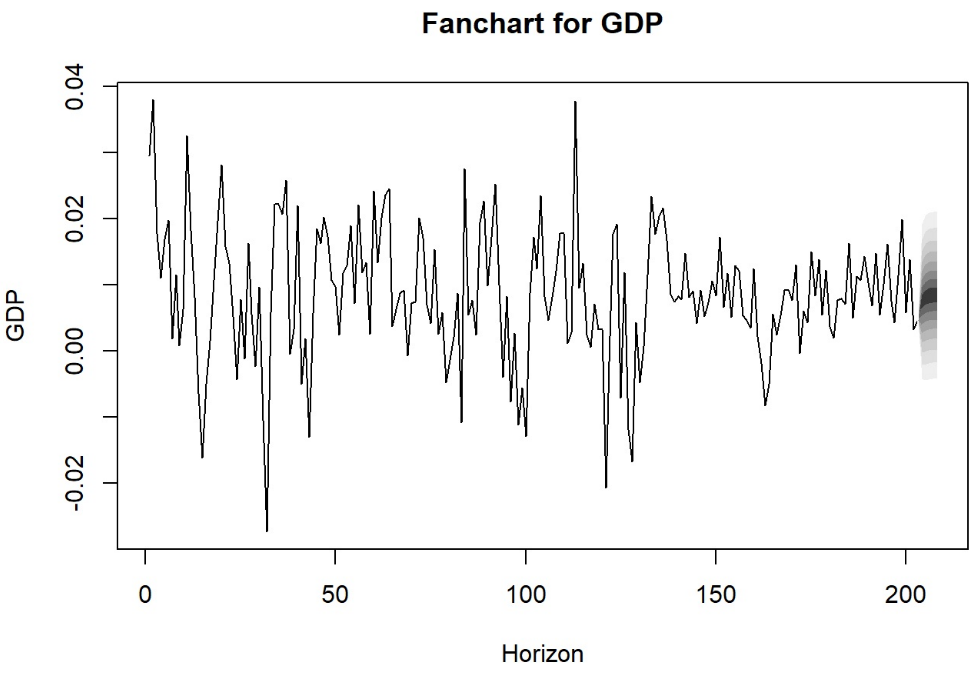
Forecast

```
forecast = predict(varmodel, n.ahead = 5, ci = 0.95)
forecast
```

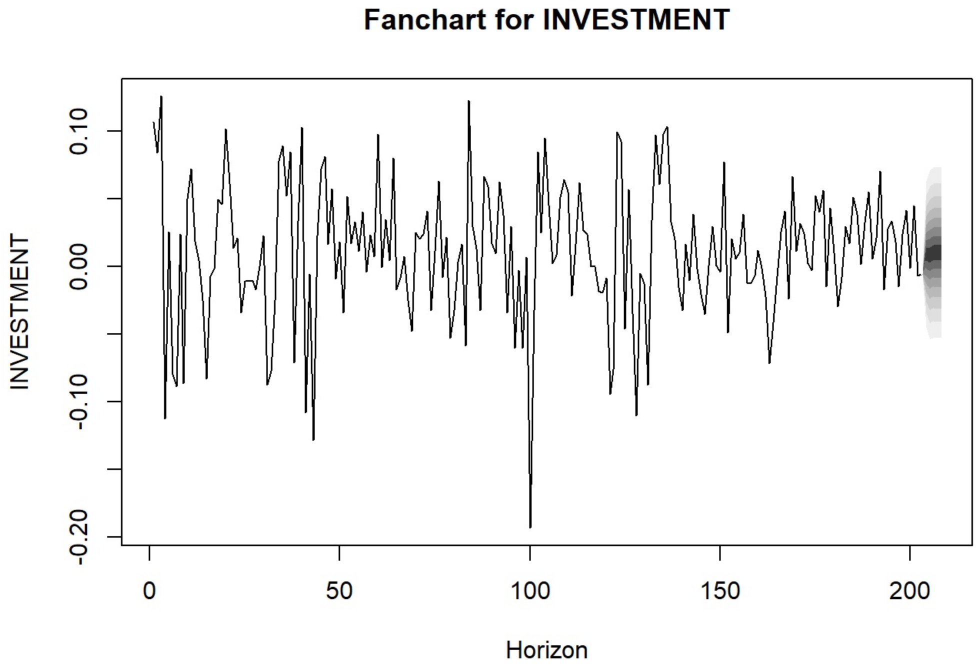
```
## $GDP
##             fcst       lower      upper         CI
## [1,] 0.007417702 -0.01051673 0.02535213 0.01793443
## [2,] 0.008080184 -0.01121603 0.02737640 0.01929621
## [3,] 0.008384249 -0.01107382 0.02784232 0.01945807
## [4,] 0.008441564 -0.01105108 0.02793421 0.01949264
## [5,] 0.008472906 -0.01102419 0.02797000 0.01949709
##
## $INVESTMENT
##             fcst       lower      upper         CI
## [1,] 0.008925118 -0.07363302 0.09148325 0.08255813
## [2,] 0.008431957 -0.08797623 0.10484015 0.09640819
## [3,] 0.010293623 -0.08700746 0.10759471 0.09730108
## [4,] 0.010002087 -0.08766833 0.10767250 0.09767041
## [5,] 0.010235969 -0.08745912 0.10793105 0.09769508
##
## $CONSUMPTION
##             fcst        lower      upper         CI
## [1,] 0.007856496 -0.008752656 0.02446565 0.01660915
## [2,] 0.008709861 -0.008525755 0.02594548 0.01723562
## [3,] 0.008643504 -0.008823114 0.02611012 0.01746662
## [4,] 0.008754667 -0.008727141 0.02623647 0.01748181
## [5,] 0.008732996 -0.008755436 0.02622143 0.01748843
```

```
#Forecasted Plots

fanchart(forecast, names = "GDP", main = "Fanchart for GDP",
         xlab = "Horizon", ylab = "GDP")
```




```
fanchart(forecast, names = "INVESTMENT", main = "Fanchart for INVESTMENT",
         xlab = "Horizon", ylab = "INVESTMENT")
```




```
fanchart(forecast, names = "CONSUMPTION", main = "Fanchart for CONSUMPTION",
         xlab = "Horizon", ylab = "CONSUMPTION")
```