CDTT

Ta Tien Hai

2022-03-12

Import data and package

library(vars);

## Warning: package 'vars' was built under R version 4.1.3

## Loading required package: MASS

## Loading required package: strucchange

## Warning: package 'strucchange' was built under R version 4.1.3

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 4.1.3

##   
## Attaching package: 'zoo'

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

## Loading required package: sandwich

## Warning: package 'sandwich' was built under R version 4.1.3

## Loading required package: urca

## Warning: package 'urca' was built under R version 4.1.3

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 4.1.3

library(VARtests);

## Warning: package 'VARtests' was built under R version 4.1.3

library(urca);  
library(forecast)

## Warning: package 'forecast' was built under R version 4.1.3

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

library(tseries);

## Warning: package 'tseries' was built under R version 4.1.3

library(urca);  
library(FitAR);

## Warning: package 'FitAR' was built under R version 4.1.3

## Loading required package: lattice

## Loading required package: leaps

## Warning: package 'leaps' was built under R version 4.1.3

## Loading required package: ltsa

## Loading required package: bestglm

## Warning: package 'bestglm' was built under R version 4.1.3

##   
## Attaching package: 'FitAR'

## The following object is masked from 'package:forecast':  
##   
## BoxCox

library(zoo);  
library(tsDyn);

## Warning: package 'tsDyn' was built under R version 4.1.3

library(tidyverse);

## Warning: package 'tidyverse' was built under R version 4.1.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.8  
## v tidyr 1.2.0 v stringr 1.4.0  
## v readr 2.1.2 v forcats 0.5.1

## Warning: package 'ggplot2' was built under R version 4.1.3

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'readr' was built under R version 4.1.3

## Warning: package 'purrr' was built under R version 4.1.3

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.3

## Warning: package 'forcats' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x stringr::boundary() masks strucchange::boundary()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()  
## x dplyr::select() masks MASS::select()

library(GGally)

## Warning: package 'GGally' was built under R version 4.1.3

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

library(readxl)

## Warning: package 'readxl' was built under R version 4.1.3

macrodat <- read\_excel("macrodat.xlsx")  
cpi=ts(macrodat$cpi,start = c(2012,4),end = c(2021,12),frequency = 12)  
m2=ts(macrodat$m2,start = c(2012,4),end = c(2021,12),frequency = 12)  
exc=ts(macrodat$exc,start = c(2012,4),end = c(2021,12),frequency = 12)  
bot=ts(macrodat$bot,start = c(2012,4),end = c(2021,12),frequency = 12)  
vnindex=ts(macrodat$vnindex,start = c(2012,4),end = c(2021,12),frequency = 12)

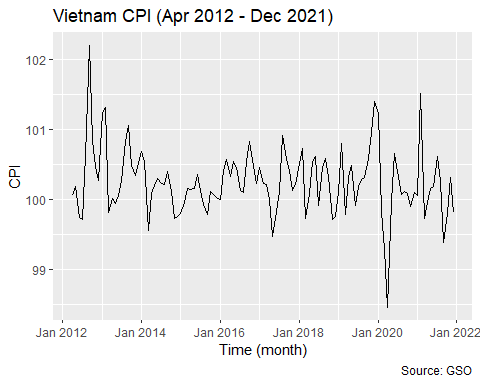
EDA

str(macrodat)

## tibble [117 x 7] (S3: tbl\_df/tbl/data.frame)  
## $ Month : chr [1:117] "M4" "M5" "M6" "M7" ...  
## $ Year : num [1:117] 2012 2012 2012 2012 2012 ...  
## $ cpi : num [1:117] 100 100.2 99.7 99.7 100.6 ...  
## $ m2 : num [1:117] 3184846 3247187 3306645 3338824 3371651 ...  
## $ exc : num [1:117] 20830 20820 20880 20850 20790 ...  
## $ bot : num [1:117] 4387160 4181169 4947935 4830356 5035163 ...  
## $ vnindex: num [1:117] 474 429 422 414 396 ...

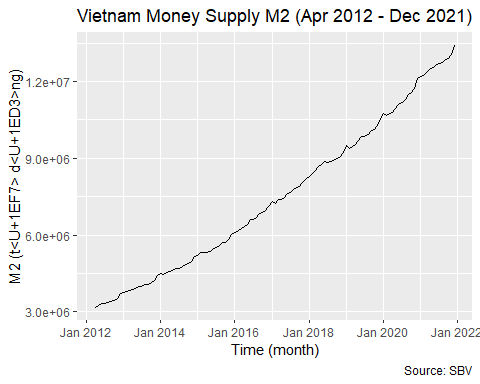
# Create timeline  
year = as.character(macrodat$Year)  
mon = as.character(macrodat$Month) %>% str\_sub(2,3)  
timeline=str\_c(year,mon,sep="-") %>% as.yearmon()  
visdat = data.frame(timeline,vnindex,m2,exc,bot,cpi)  
# Visualize  
## CPI  
ggplot(visdat,aes(x=timeline, y=cpi))+  
 geom\_line()+  
 labs(x="Time (month)", y="CPI",  
 title="Vietnam CPI (Apr 2012 - Dec 2021)",  
 caption="Source: GSO")

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



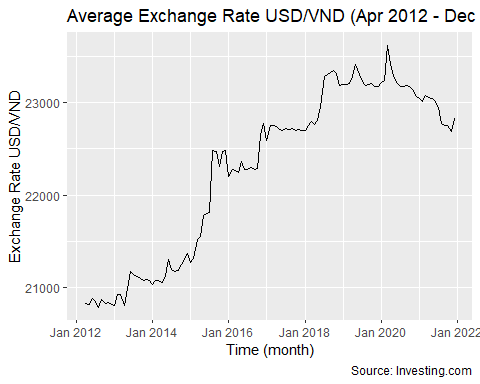
## m2  
ggplot(visdat,aes(x=timeline, y=m2))+  
 geom\_line()+  
 labs(x="Time (month)", y="M2 (tỷ đồng)",  
 title="Vietnam Money Supply M2 (Apr 2012 - Dec 2021)",  
 caption="Source: SBV")

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



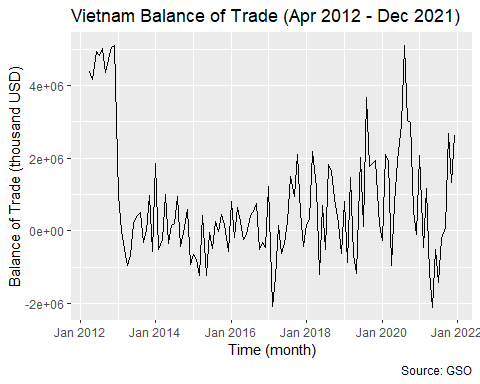
## exc  
ggplot(visdat,aes(x=timeline, y=exc))+  
 geom\_line()+  
 labs(x="Time (month)", y="Exchange Rate USD/VND",  
 title="Average Exchange Rate USD/VND (Apr 2012 - Dec 2021)",  
 caption="Source: Investing.com")

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



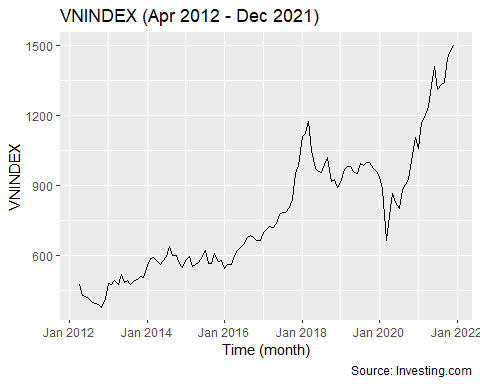
## bot  
ggplot(visdat,aes(x=timeline, y=bot))+  
 geom\_line()+  
 labs(x="Time (month)", y="Balance of Trade (thousand USD)",  
 title="Vietnam Balance of Trade (Apr 2012 - Dec 2021)",  
 caption="Source: GSO")

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



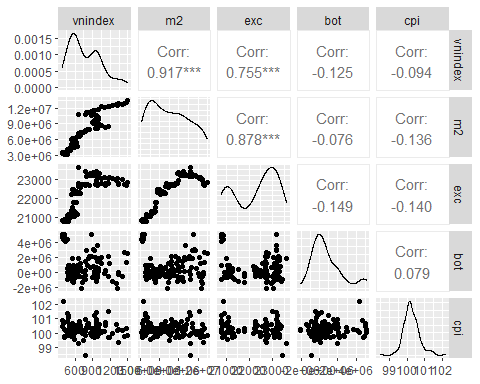
## VNINDEX  
ggplot(visdat,aes(x=timeline, y=vnindex))+  
 geom\_line()+  
 labs(x="Time (month)", y="VNINDEX",  
 title="VNINDEX (Apr 2012 - Dec 2021)",  
 caption="Source: Investing.com")

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.



# Pair plot  
ggpairs(visdat[,-1])

## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.  
## Don't know how to automatically pick scale for object of type ts. Defaulting to continuous.

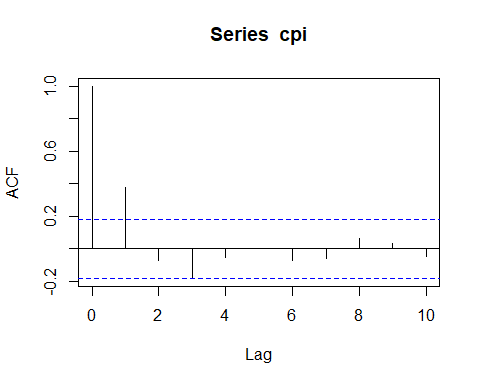


Stationary

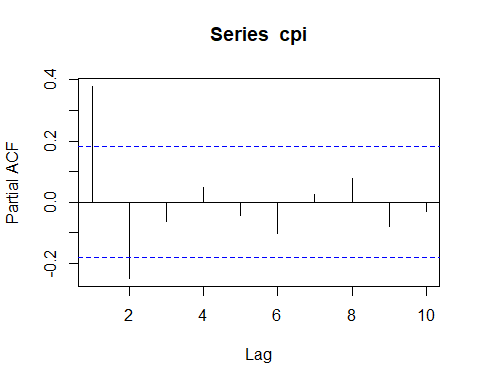
# normalize data  
norm=function(dat,rng=c(0,1)){  
 min=rng[1];max=rng[2]  
 std\_dat=(dat-min(dat))/(max(dat)-min(dat))  
 scaled\_dat=std\_dat\*(rng[2]-rng[1])+rng[1]  
 return(scaled\_dat=as.vector(scaled\_dat))  
}  
cpi=norm(cpi); m2=norm(m2); exc=norm(exc); bot=norm(bot); vnindex=norm(vnindex)  
  
# cpi I(0)  
summary(ur.df(cpi, type = c("drift"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.38366 -0.05638 -0.00521 0.06408 0.43529   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.37156 0.05020 7.401 2.67e-11 \*\*\*  
## z.lag.1 -0.77465 0.10199 -7.595 9.99e-12 \*\*\*  
## z.diff.lag 0.25368 0.09179 2.764 0.00668 \*\*   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1162 on 112 degrees of freedom  
## Multiple R-squared: 0.3516, Adjusted R-squared: 0.34   
## F-statistic: 30.36 on 2 and 112 DF, p-value: 2.91e-11  
##   
##   
## Value of test-statistic is: -7.5952 28.8458   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.46 -2.88 -2.57  
## phi1 6.52 4.63 3.81

acf(cpi, lag.max = 10, plot=TRUE,na.action = na.contiguous)



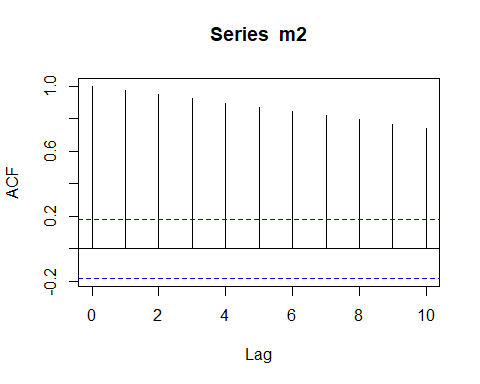
pacf(cpi, lag.max = 10, plot=TRUE,na.action = na.contiguous)



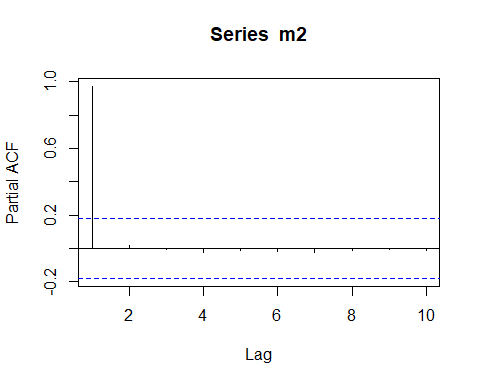
# m2 I(0)  
summary(ur.df (m2, type = c("trend"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0198668 -0.0043572 -0.0000553 0.0031429 0.0265849   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)  
## (Intercept) 0.0035702 0.0022525 1.585 0.116  
## z.lag.1 -0.0195896 0.0214992 -0.911 0.364  
## tt 0.0002380 0.0001844 1.290 0.200  
## z.diff.lag -0.0754856 0.0993363 -0.760 0.449  
##   
## Residual standard error: 0.007142 on 111 degrees of freedom  
## Multiple R-squared: 0.1011, Adjusted R-squared: 0.07684   
## F-statistic: 4.163 on 3 and 111 DF, p-value: 0.00778  
##   
##   
## Value of test-statistic is: -0.9112 26.7586 6.2413   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -3.99 -3.43 -3.13  
## phi2 6.22 4.75 4.07  
## phi3 8.43 6.49 5.47

acf(m2, lag.max = 10, plot=TRUE,na.action = na.contiguous)



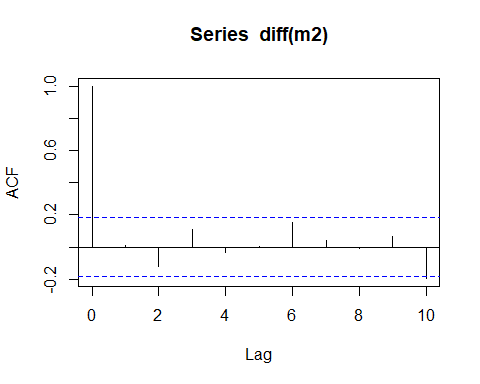
pacf(m2, lag.max = 10, plot=TRUE,na.action = na.contiguous)



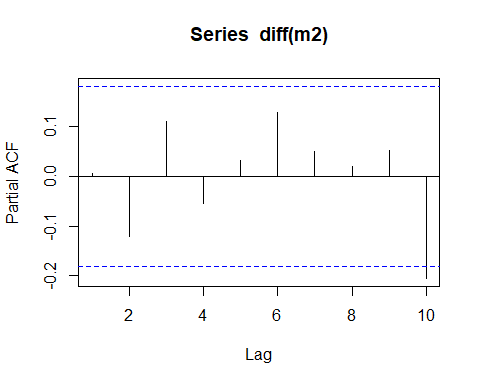
# m2 I(1)  
summary(ur.df(diff(m2), type = c("drift"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.0187786 -0.0053093 -0.0002507 0.0040430 0.0283381   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.009780 0.001384 7.068 1.47e-10 \*\*\*  
## z.lag.1 -1.133372 0.142535 -7.952 1.68e-12 \*\*\*  
## z.diff.lag 0.137408 0.100104 1.373 0.173   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.007465 on 111 degrees of freedom  
## Multiple R-squared: 0.4759, Adjusted R-squared: 0.4664   
## F-statistic: 50.39 on 2 and 111 DF, p-value: 2.687e-16  
##   
##   
## Value of test-statistic is: -7.9515 31.6966   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.46 -2.88 -2.57  
## phi1 6.52 4.63 3.81

acf(diff(m2), lag.max = 10, plot=TRUE,na.action = na.contiguous)



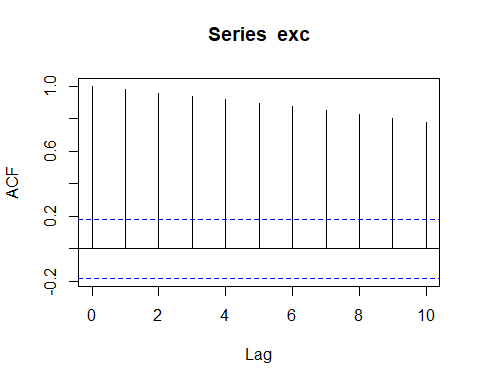
pacf(diff(m2), lag.max = 10, plot=TRUE,na.action = na.contiguous)



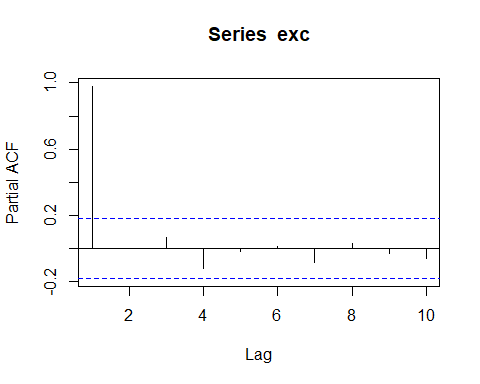
# exc I(0)  
summary(ur.df(exc, type = c("trend"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.103529 -0.018203 -0.005328 0.009184 0.226815   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.576e-02 7.961e-03 1.979 0.0503 .  
## z.lag.1 -2.437e-02 3.174e-02 -0.768 0.4442   
## tt 5.798e-05 3.051e-04 0.190 0.8496   
## z.diff.lag -1.150e-02 9.899e-02 -0.116 0.9078   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.04112 on 111 degrees of freedom  
## Multiple R-squared: 0.0217, Adjusted R-squared: -0.004744   
## F-statistic: 0.8206 on 3 and 111 DF, p-value: 0.4852  
##   
##   
## Value of test-statistic is: -0.7679 1.6817 1.219   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -3.99 -3.43 -3.13  
## phi2 6.22 4.75 4.07  
## phi3 8.43 6.49 5.47

acf(exc, lag.max = 10, plot=TRUE,na.action = na.contiguous)



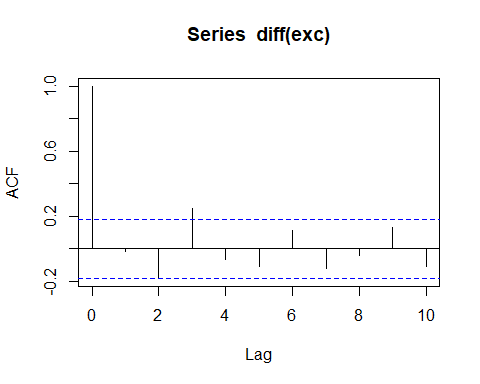
pacf(exc, lag.max = 10, plot=TRUE,na.action = na.contiguous)



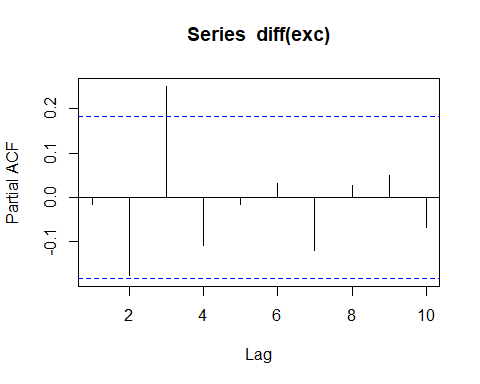
# exc I(1)  
summary(ur.df(diff(exc), type = c("drift"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.096869 -0.018054 -0.007589 0.011317 0.229966   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.007190 0.003909 1.839 0.0685 .   
## z.lag.1 -1.193842 0.133314 -8.955 9.02e-15 \*\*\*  
## z.diff.lag 0.178515 0.093960 1.900 0.0600 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.04089 on 111 degrees of freedom  
## Multiple R-squared: 0.5204, Adjusted R-squared: 0.5118   
## F-statistic: 60.23 on 2 and 111 DF, p-value: < 2.2e-16  
##   
##   
## Value of test-statistic is: -8.9551 40.0979   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.46 -2.88 -2.57  
## phi1 6.52 4.63 3.81

acf(diff(exc), lag.max = 10, plot=TRUE,na.action = na.contiguous)



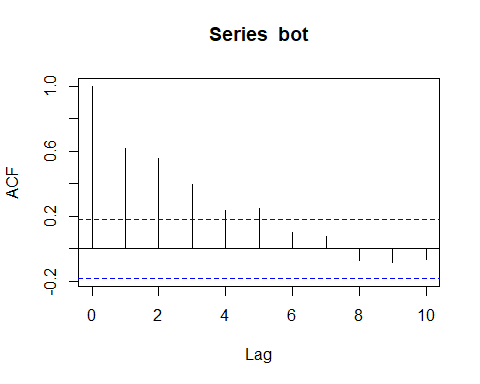
pacf(diff(exc), lag.max = 10, plot=TRUE,na.action = na.contiguous)



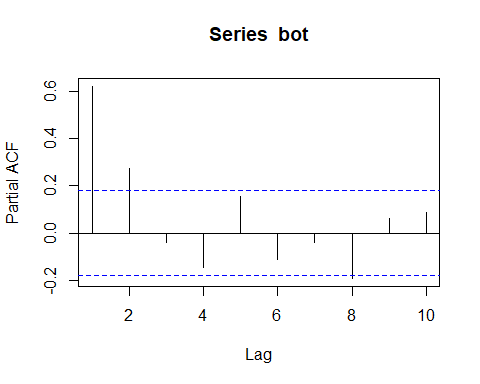
# bot I(0)  
summary(ur.df(bot, type = c("drift"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.39563 -0.11858 0.00138 0.11543 0.43304   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.10687 0.03352 3.188 0.001855 \*\*   
## z.lag.1 -0.28300 0.07602 -3.723 0.000310 \*\*\*  
## z.diff.lag -0.29864 0.08833 -3.381 0.000996 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1677 on 112 degrees of freedom  
## Multiple R-squared: 0.2748, Adjusted R-squared: 0.2619   
## F-statistic: 21.22 on 2 and 112 DF, p-value: 1.53e-08  
##   
##   
## Value of test-statistic is: -3.7227 6.9539   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.46 -2.88 -2.57  
## phi1 6.52 4.63 3.81

acf(bot, lag.max = 10, plot=TRUE,na.action = na.contiguous)



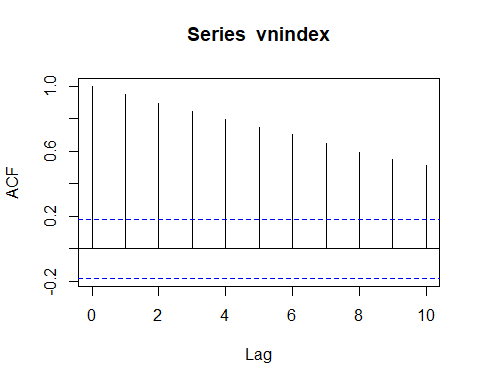
pacf(bot, lag.max = 10, plot=TRUE,na.action = na.contiguous)



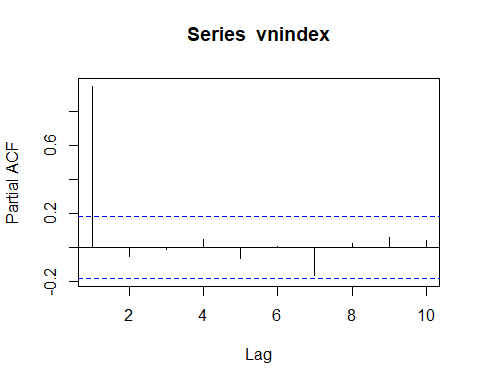
# vnindex I(0)  
summary(ur.df(vnindex, type = c("trend"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression trend   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + tt + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.21367 -0.01360 0.00121 0.01857 0.10948   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) -0.0023941 0.0081076 -0.295 0.7683   
## z.lag.1 -0.0671157 0.0407863 -1.646 0.1027   
## tt 0.0005700 0.0002842 2.006 0.0473 \*  
## z.diff.lag 0.0806433 0.0970890 0.831 0.4080   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.04223 on 111 degrees of freedom  
## Multiple R-squared: 0.03902, Adjusted R-squared: 0.01305   
## F-statistic: 1.502 on 3 and 111 DF, p-value: 0.2179  
##   
##   
## Value of test-statistic is: -1.6455 2.6929 2.0964   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau3 -3.99 -3.43 -3.13  
## phi2 6.22 4.75 4.07  
## phi3 8.43 6.49 5.47

acf(vnindex, lag.max = 10, plot=TRUE,na.action = na.contiguous)



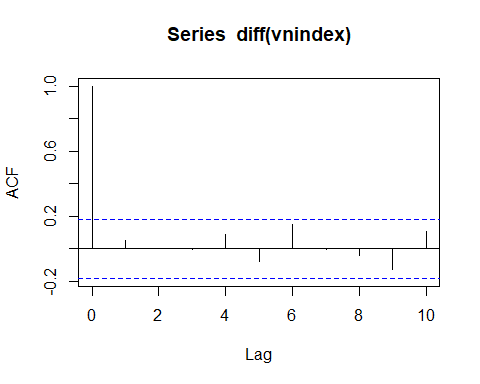
pacf(vnindex, lag.max = 10, plot=TRUE,na.action = na.contiguous)



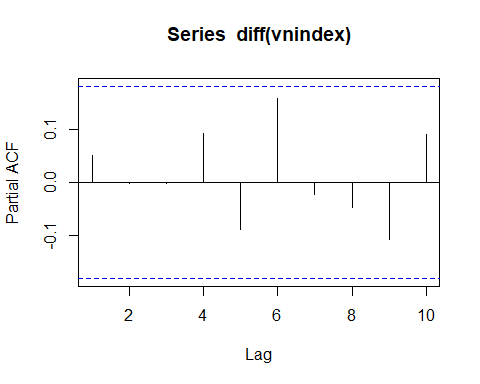
# vnindex I(1)  
summary(ur.df(diff(vnindex), type = c("drift"), selectlags="AIC"))

##   
## ###############################################   
## # Augmented Dickey-Fuller Test Unit Root Test #   
## ###############################################   
##   
## Test regression drift   
##   
##   
## Call:  
## lm(formula = z.diff ~ z.lag.1 + 1 + z.diff.lag)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.201720 -0.017032 0.002153 0.017373 0.103183   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 0.008031 0.004158 1.931 0.056 .   
## z.lag.1 -0.952383 0.130446 -7.301 4.59e-11 \*\*\*  
## z.diff.lag 0.001590 0.094494 0.017 0.987   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.04301 on 111 degrees of freedom  
## Multiple R-squared: 0.4756, Adjusted R-squared: 0.4661   
## F-statistic: 50.33 on 2 and 111 DF, p-value: 2.767e-16  
##   
##   
## Value of test-statistic is: -7.3009 26.6596   
##   
## Critical values for test statistics:   
## 1pct 5pct 10pct  
## tau2 -3.46 -2.88 -2.57  
## phi1 6.52 4.63 3.81

acf(diff(vnindex), lag.max = 10, plot=TRUE,na.action = na.contiguous)



pacf(diff(vnindex), lag.max = 10, plot=TRUE,na.action = na.contiguous)



Estimate VECM model

Series2=cbind(vnindex,m2,exc,bot,cpi)  
  
#chon lag 1  
VARselect(Series2, lag.max = 10, type = "const")

## $selection  
## AIC(n) HQ(n) SC(n) FPE(n)   
## 1 1 1 1   
##   
## $criteria  
## 1 2 3 4 5  
## AIC(n) -3.044479e+01 -3.039760e+01 -3.023900e+01 -3.012973e+01 -3.011951e+01  
## HQ(n) -3.014100e+01 -2.984065e+01 -2.942888e+01 -2.906645e+01 -2.880307e+01  
## SC(n) -2.969540e+01 -2.902372e+01 -2.824062e+01 -2.750686e+01 -2.687215e+01  
## FPE(n) 6.001375e-14 6.310615e-14 7.452055e-14 8.433917e-14 8.725724e-14  
## 6 7 8 9 10  
## AIC(n) -3.001492e+01 -2.991178e+01 -2.979251e+01 -2.965535e+01 -2.952281e+01  
## HQ(n) -2.844532e+01 -2.808902e+01 -2.771659e+01 -2.732627e+01 -2.694057e+01  
## SC(n) -2.614307e+01 -2.541543e+01 -2.467167e+01 -2.391001e+01 -2.315299e+01  
## FPE(n) 1.004097e-13 1.171358e-13 1.414974e-13 1.780713e-13 2.294824e-13

#Kiem tra so quan he dong tich hop (r=1)  
ctest1tr=ca.jo(Series2, type="trace", ecdet="trend", K=2)  
summary(ctest1tr)

##   
## ######################   
## # Johansen-Procedure #   
## ######################   
##   
## Test type: trace statistic , with linear trend in cointegration   
##   
## Eigenvalues (lambda):  
## [1] 4.112670e-01 2.373204e-01 1.147994e-01 7.138133e-02 4.520565e-02  
## [6] -7.981030e-17  
##   
## Values of teststatistic and critical values of test:  
##   
## test 10pct 5pct 1pct  
## r <= 4 | 5.32 10.49 12.25 16.26  
## r <= 3 | 13.84 22.76 25.32 30.45  
## r <= 2 | 27.86 39.06 42.44 48.45  
## r <= 1 | 59.02 59.14 62.99 70.05  
## r = 0 | 119.94 83.20 87.31 96.58  
##   
## Eigenvectors, normalised to first column:  
## (These are the cointegration relations)  
##   
## vnindex.l2 m2.l2 exc.l2 bot.l2 cpi.l2  
## vnindex.l2 1.00000000 1.0000000 1.000000000 1.00000000 1.00000000  
## m2.l2 1.97039598 -111.5924591 -1.127845469 -7.55630580 0.91543950  
## exc.l2 1.43493822 -21.1659790 0.430737527 -0.97892342 1.18579892  
## bot.l2 1.21015088 13.5647421 1.043995045 -0.07722296 0.03438854  
## cpi.l2 -8.40048243 0.2239253 0.438640000 -0.05225062 0.01236926  
## trend.l2 -0.04180168 1.1902186 -0.005710257 0.06438425 -0.02287368  
## trend.l2  
## vnindex.l2 1.000000000  
## m2.l2 1.224016698  
## exc.l2 -0.433426091  
## bot.l2 0.009999249  
## cpi.l2 -0.195544694  
## trend.l2 -0.012386694  
##   
## Weights W:  
## (This is the loading matrix)  
##   
## vnindex.l2 m2.l2 exc.l2 bot.l2 cpi.l2  
## vnindex.d 0.0056607849 0.0009600875 -0.017733990 -0.0323421014 -0.044061501  
## m2.d 0.0004877069 0.0010351449 -0.002436233 -0.0002957167 0.003772761  
## exc.d -0.0071174859 0.0011241048 0.004941196 0.0464816326 -0.022413236  
## bot.d -0.0181314974 -0.0114793340 -0.162441348 0.0388815104 0.088249259  
## cpi.d 0.0992472164 -0.0001169616 0.006035753 0.0248705774 -0.007482268  
## trend.l2  
## vnindex.d 9.688884e-16  
## m2.d -4.575939e-16  
## exc.d -2.775953e-15  
## bot.d 5.072911e-15  
## cpi.d 5.716047e-15

#Uoc luong VECM  
VECM1= VECM(Series2, lag=1, r=1, include = c("both"), LRinclude = c("const"), estim ="ML")

## Warning in lineVar(data, lag, r = r, include = include, model = "VECM", : When `LRinclude` is either 'const' or 'both', `include` can only be `none`.  
## Setting include='none'.

summary(VECM1)

## #############  
## ###Model VECM   
## #############  
## Full sample size: 117 End sample size: 115  
## Number of variables: 5 Number of estimated slope parameters 30  
## AIC -3437.003 BIC -3343.676 SSR 5.770085  
## Cointegrating vector (estimated by ML):  
## vnindex m2 exc bot cpi const  
## r1 1 -4.518557 0.1847605 0.2212277 -8.024218 1.435647  
##   
##   
## ECT vnindex -1 m2 -1   
## Equation vnindex -0.0005(0.0016) 0.0411(0.0999) 0.6675(0.5819)   
## Equation m2 -0.0020(0.0003)\*\*\* 0.0254(0.0172) 0.0217(0.1004)   
## Equation exc -0.0041(0.0015)\*\* 0.0249(0.0952) -1.3237(0.5546)\*   
## Equation bot -0.0064(0.0067) 0.3757(0.4145) -3.4908(2.4149)   
## Equation cpi 0.0174(0.0050)\*\*\* 0.4311(0.3103) 5.8285(1.8079)\*\*   
## exc -1 bot -1 cpi -1   
## Equation vnindex -0.0058(0.1048) -0.0295(0.0213) -0.0111(0.0294)   
## Equation m2 0.0039(0.0181) -0.0080(0.0037)\* -0.0108(0.0051)\*   
## Equation exc 0.0006(0.0999) 0.0186(0.0203) 0.0072(0.0281)   
## Equation bot 0.1185(0.4350) -0.4093(0.0885)\*\*\* 0.0285(0.1222)   
## Equation cpi -0.3947(0.3257) 0.0524(0.0662) -0.1030(0.0915)

#Bien doi thanh VAR  
VECM2VAR=vec2var(ctest1tr,r=1)  
  
#Kiem dinh phan du (ok)  
VECM2VARresiduals = resid(VECM2VAR)  
Box.test(VECM2VARresiduals[,1])

##   
## Box-Pierce test  
##   
## data: VECM2VARresiduals[, 1]  
## X-squared = 0.010366, df = 1, p-value = 0.9189

Box.test(VECM2VARresiduals[,2])

##   
## Box-Pierce test  
##   
## data: VECM2VARresiduals[, 2]  
## X-squared = 0.0022496, df = 1, p-value = 0.9622

Box.test(VECM2VARresiduals[,3])

##   
## Box-Pierce test  
##   
## data: VECM2VARresiduals[, 3]  
## X-squared = 0.0058478, df = 1, p-value = 0.939

Box.test(VECM2VARresiduals[,4])

##   
## Box-Pierce test  
##   
## data: VECM2VARresiduals[, 4]  
## X-squared = 0.081807, df = 1, p-value = 0.7749

Box.test(VECM2VARresiduals[,5])

##   
## Box-Pierce test  
##   
## data: VECM2VARresiduals[, 5]  
## X-squared = 0.76908, df = 1, p-value = 0.3805

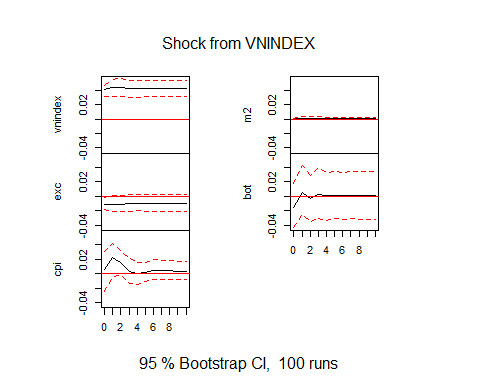
#Serial Correlation  
serial.test(VECM2VAR, lags.pt = 8, type="PT.asymptotic")

##   
## Portmanteau Test (asymptotic)  
##   
## data: Residuals of VAR object VECM2VAR  
## Chi-squared = 171.59, df = 155, p-value = 0.1715

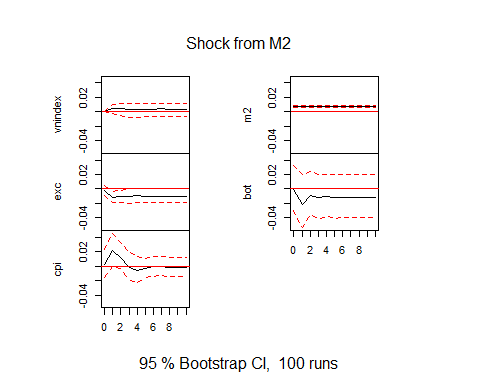
# Heteroscedasticity   
arch.test(VECM2VAR,lags.multi=12,multivariate.only=TRUE)

##   
## ARCH (multivariate)  
##   
## data: Residuals of VAR object VECM2VAR  
## Chi-squared = 1545, df = 2700, p-value = 1

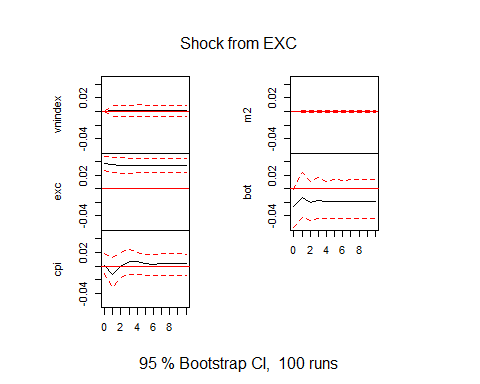
#Calculate IRF  
VECM2VAR %>% irf(impulse="vnindex", n.ahead=10) %>% plot(main="Shock from VNINDEX")



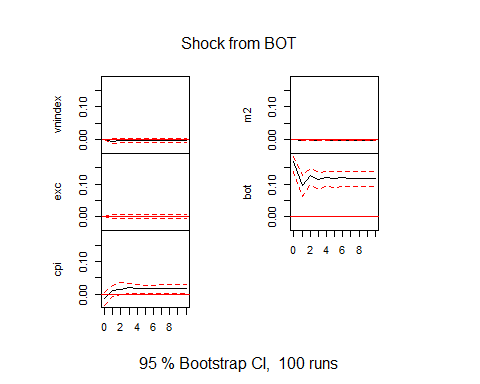
VECM2VAR %>% irf(impulse="m2", n.ahead=10) %>% plot(main="Shock from M2")



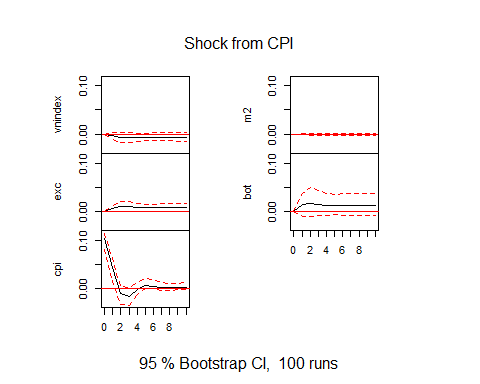
VECM2VAR %>% irf(impulse="exc", n.ahead=10) %>% plot(main="Shock from EXC")



VECM2VAR %>% irf(impulse="bot", n.ahead=10) %>% plot(main="Shock from BOT")



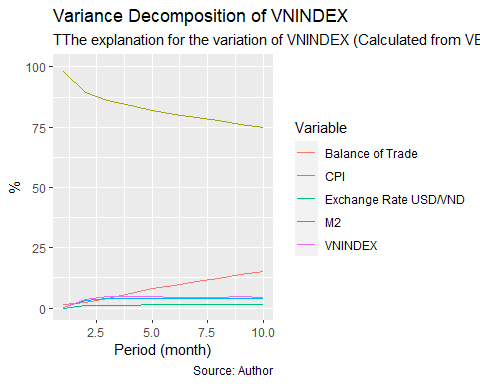
VECM2VAR %>% irf(impulse="cpi", n.ahead=10) %>% plot(main="Shock from CPI")



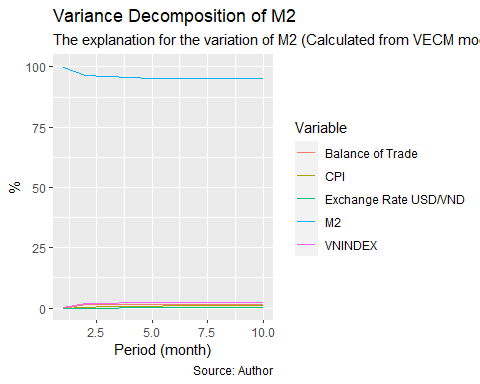
# Variance Decomposition  
VD1=fevd(VECM2VAR, n.hat=4) %>% unname()  
period=rep(1:10,5) %>% sort()  
Variable=rep(c("VNINDEX","M2","Exchange Rate USD/VND","Balance of Trade","CPI"),10)  
VD1\_cpi=VD1[1] %>% as.data.frame() %>% pivot\_longer(cols = 1:5,names\_to = "var",values\_to = "per") %>% data.frame(period,Variable)  
VD1\_m2=VD1[2] %>% as.data.frame() %>% pivot\_longer(cols = 1:5,names\_to = "var",values\_to = "per") %>% data.frame(period,Variable)  
VD1\_exc=VD1[3] %>% as.data.frame() %>% pivot\_longer(cols = 1:5,names\_to = "var",values\_to = "per") %>% data.frame(period,Variable)  
VD1\_bot=VD1[4] %>% as.data.frame() %>% pivot\_longer(cols = 1:5,names\_to = "var",values\_to = "per") %>% data.frame(period,Variable)  
VD1\_vnindex=VD1[5] %>% as.data.frame() %>% pivot\_longer(cols = 1:5,names\_to = "var",values\_to = "per") %>% data.frame(period,Variable)  
fevd2=fevd(VECM2VAR)  
fevd2

## $vnindex  
## vnindex m2 exc bot cpi  
## [1,] 1.0000000 0.000000000 0.0000000000 0.000000000 0.000000000  
## [2,] 0.9862402 0.005643205 0.0002104653 0.004641122 0.003264972  
## [3,] 0.9767186 0.007524848 0.0003747359 0.003938436 0.011443412  
## [4,] 0.9713105 0.007353339 0.0006945370 0.003934488 0.016707133  
## [5,] 0.9696049 0.007108890 0.0008758325 0.003842195 0.018568159  
## [6,] 0.9690603 0.006997223 0.0009632175 0.003873847 0.019105446  
## [7,] 0.9686328 0.007006080 0.0010057816 0.003878131 0.019477158  
## [8,] 0.9681495 0.007022602 0.0010405279 0.003880993 0.019906360  
## [9,] 0.9677219 0.007022816 0.0010721555 0.003876222 0.020306944  
## [10,] 0.9673974 0.007013344 0.0010991688 0.003874021 0.020616082  
##   
## $m2  
## vnindex m2 exc bot cpi  
## [1,] 0.0007029609 0.9992970 0.0000000000 0.00000000 0.000000000  
## [2,] 0.0178416194 0.9633271 0.0003154157 0.01599530 0.002520575  
## [3,] 0.0199005085 0.9592027 0.0003704557 0.01446999 0.006056322  
## [4,] 0.0212363062 0.9546556 0.0005550877 0.01552312 0.008029910  
## [5,] 0.0216505312 0.9534464 0.0006302338 0.01558824 0.008684642  
## [6,] 0.0221093960 0.9524267 0.0006731657 0.01589997 0.008890773  
## [7,] 0.0224497295 0.9517789 0.0006931272 0.01602710 0.009051186  
## [8,] 0.0227211024 0.9511936 0.0007110438 0.01614490 0.009229324  
## [9,] 0.0229140409 0.9507510 0.0007265094 0.01622053 0.009387874  
## [10,] 0.0230649604 0.9503997 0.0007396179 0.01628748 0.009508239  
##   
## $exc  
## vnindex m2 exc bot cpi  
## [1,] 0.08688462 0.001637939 0.9114774 0.000000000 0.00000000  
## [2,] 0.08014788 0.045556893 0.8629512 0.001038030 0.01030596  
## [3,] 0.07930888 0.055177959 0.8386597 0.001270120 0.02558335  
## [4,] 0.07690604 0.057902923 0.8306689 0.001338993 0.03318317  
## [5,] 0.07542989 0.059241815 0.8285522 0.001456339 0.03531973  
## [6,] 0.07474008 0.060562839 0.8272538 0.001558085 0.03588524  
## [7,] 0.07440454 0.061738386 0.8257752 0.001631333 0.03645054  
## [8,] 0.07413433 0.062621632 0.8244198 0.001677755 0.03714650  
## [9,] 0.07388047 0.063253351 0.8233929 0.001711840 0.03776147  
## [10,] 0.07366233 0.063738140 0.8226435 0.001739930 0.03821611  
##   
## $bot  
## vnindex m2 exc bot cpi  
## [1,] 0.007890908 2.413155e-05 0.02369923 0.9683857 0.000000000  
## [2,] 0.006718432 1.139403e-02 0.02141111 0.9560205 0.004455929  
## [3,] 0.004852959 9.590207e-03 0.02238467 0.9552628 0.007909400  
## [4,] 0.004032299 9.926405e-03 0.02259778 0.9541556 0.009287916  
## [5,] 0.003326745 9.555071e-03 0.02273659 0.9549781 0.009403537  
## [6,] 0.002857276 9.652138e-03 0.02267925 0.9553264 0.009484884  
## [7,] 0.002493281 9.662217e-03 0.02265719 0.9555524 0.009634910  
## [8,] 0.002216651 9.691304e-03 0.02265494 0.9556146 0.009822457  
## [9,] 0.001995408 9.685529e-03 0.02266604 0.9556879 0.009965122  
## [10,] 0.001815700 9.684406e-03 0.02267192 0.9557623 0.010065720  
##   
## $cpi  
## vnindex m2 exc bot cpi  
## [1,] 0.002547264 0.0006718183 4.176108e-05 0.01664308 0.9800961  
## [2,] 0.036014716 0.0334564770 1.015741e-02 0.02445894 0.8959125  
## [3,] 0.049193115 0.0416120557 9.668841e-03 0.03894090 0.8605851  
## [4,] 0.047420182 0.0397688324 1.154101e-02 0.06261271 0.8386573  
## [5,] 0.046300544 0.0408274195 1.359572e-02 0.08015213 0.8191242  
## [6,] 0.045630881 0.0405784749 1.420383e-02 0.09545206 0.8041348  
## [7,] 0.045791678 0.0398399107 1.443924e-02 0.10957263 0.7903565  
## [8,] 0.046045886 0.0391148393 1.475088e-02 0.12402361 0.7760648  
## [9,] 0.046064117 0.0384671116 1.521055e-02 0.13824320 0.7620150  
## [10,] 0.045957106 0.0378924466 1.569661e-02 0.15195371 0.7485001

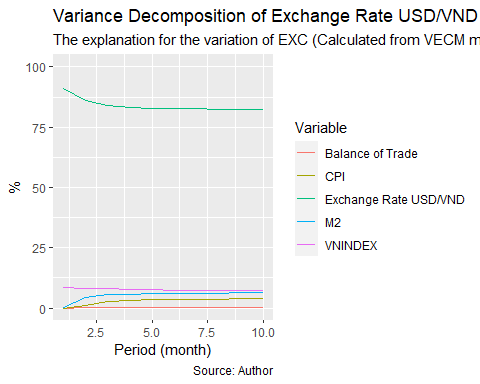
## Explain for VNINDEX  
ggplot(data = VD1\_vnindex, aes(x=period,y=per\*100,color=Variable))+  
 geom\_line()+  
 labs(x="Period (month)",y="%",title="Variance Decomposition of VNINDEX",  
 caption="Source: Author",subtitle="TThe explanation for the variation of VNINDEX (Calculated from VECM model)")+  
 ylim(0,100)



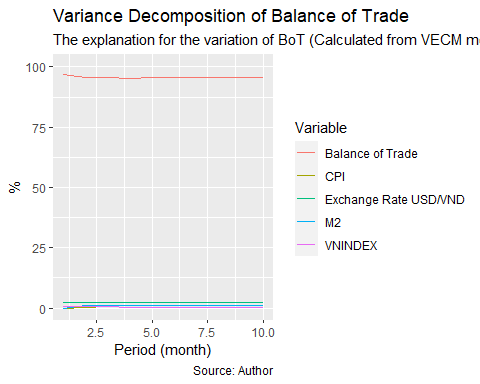
## Explain for M2  
ggplot(data = VD1\_m2, aes(x=period,y=per\*100,color=Variable))+  
 geom\_line()+  
 labs(x="Period (month)",y="%",title="Variance Decomposition of M2",  
 caption="Source: Author",subtitle="The explanation for the variation of M2 (Calculated from VECM model)")+  
 ylim(0,100)



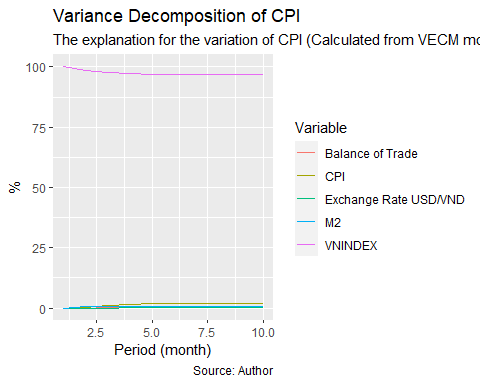
## Explain for EXC  
ggplot(data = VD1\_exc, aes(x=period,y=per\*100,color=Variable))+  
 geom\_line()+  
 labs(x="Period (month)",y="%",title="Variance Decomposition of Exchange Rate USD/VND",  
 caption="Source: Author",subtitle="The explanation for the variation of EXC (Calculated from VECM model)")+  
 ylim(0,100)



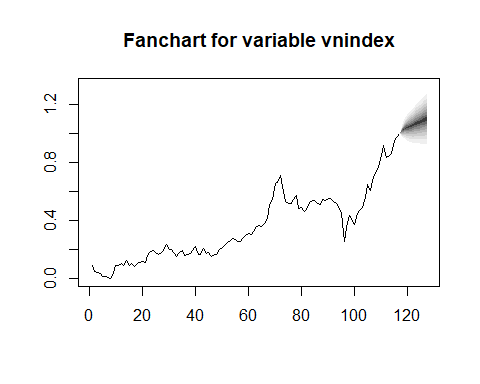
## Explain for BOT  
ggplot(data = VD1\_bot, aes(x=period,y=per\*100,color=Variable))+  
 geom\_line()+  
 labs(x="Period (month)",y="%",title="Variance Decomposition of Balance of Trade",  
 caption="Source: Author",subtitle="The explanation for the variation of BoT (Calculated from VECM model)")+  
 ylim(0,100)



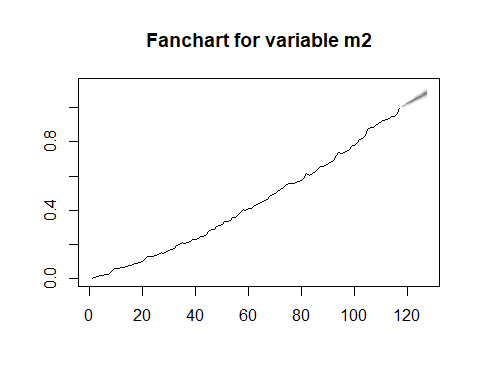
## Explain for CPI  
ggplot(data = VD1\_cpi, aes(x=period,y=per\*100,color=Variable))+  
 geom\_line()+  
 labs(x="Period (month)",y="%",title="Variance Decomposition of CPI",  
 caption="Source: Author",subtitle="The explanation for the variation of CPI (Calculated from VECM model)")+  
 ylim(0,100)



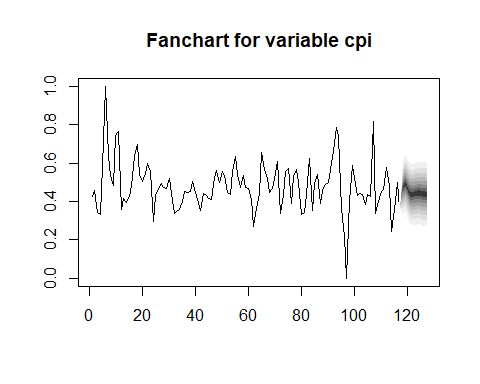
FevdSeries=predict(VECM2VAR, n.head=10, ci=0.95)  
fanchart(FevdSeries, names="vnindex")



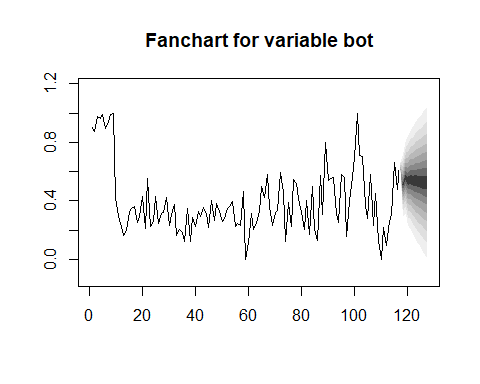
fanchart(FevdSeries, names="m2")



fanchart(FevdSeries, names="cpi")



fanchart(FevdSeries, names="bot")



fanchart(FevdSeries, names="exc")

