Graded PS2

Shih-Chieh Lee

2023-03-15

Load all required packages

```
library(quantmod); library(xts); library(ggplot2); library(ggpubr)
library(tseries); library(urca); library(tsDyn); library(timetk); library(torch)
```

```
## Warning: i torch failed to start, restart your R session to try again.
## i You might need to reinstall torch using ‘install_torch()‘
## x /Library/Frameworks/R.framework/Versions/4.2/Resources/library/torch/lib/liblantern.dylib
##   -
##   dlopen(/Library/Frameworks/R.framework/Versions/4.2/Resources/library/torch/lib/liblantern.dylib,
##   10): Symbol not found:
##   __ZNSt3__113basic_filebufIcNS_11char_traitsIcEEE4openEPKcj Referenced from:
##   /Library/Frameworks/R.framework/Versions/4.2/Resources/library/torch/lib/liblantern.dylib
##   (which was built for Mac OS X 12.6) Expected in: /usr/lib/libc++.1.dylib
```

```
library(dplyr)
```

1. Data

For future analysis, we need to first read the data and convert it into class txs. The start date of the dataset should be 2012-01-01.

```
table <- read.csv(file = "Macro_data_can.csv", header =TRUE, sep = ",")
ind <- as.Date(table$Index, format="%Y-%m-%d")
table <- subset(table, select=-c(Index))
Macro_data_can <- xts(x=table, order.by=ind)
start_date <- as.Date("2012-01-01")
can.c <- subset(Macro_data_can, ind >= start_date)
```

2. Johansen Test

We now move to run Johansen Cointegration Test

```
cointegration <- ca.jo(can.c, type="trace", ecdet="trend", spec="transitory")
sc <- summary(cointegration)
sc
```

```
##
## ######################
## # Johansen-Procedure #
## ######################
##
## Test type: trace statistic , with linear trend in cointegration
##
## Eigenvalues (lambda):
## [1]  3.838184e-01  3.226279e-01  1.319213e-01  3.567509e-02 -4.163336e-17
##
## Values of teststatistic and critical values of test:
##
##            test 10pct  5pct  1pct
## r <= 3 |   4.58 10.49 12.25 16.26
## r <= 2 |  22.40 22.76 25.32 30.45
## r <= 1 |  71.48 39.06 42.44 48.45
## r = 0  | 132.50 59.14 62.99 70.05
##
## Eigenvectors, normalised to first column:
## (These are the cointegration relations)
##
##                      CPI.l1     GDP.l1 Unemployment.l1 Target.Rate.l1
## CPI.l1           1.00000000  1.0000000      1.00000000     1.00000000
## GDP.l1           0.04894894  0.5540824     -0.05722982     0.01142389
## Unemployment.l1  0.12367051 17.1120816     -1.65387571     1.32384943
## Target.Rate.l1  -2.46252004 -6.0651678     17.53964496    -2.14541871
## trend.l1        -0.28634256 -1.5975999     -0.17747221    -0.26734354
##                    trend.l1
## CPI.l1           1.00000000
## GDP.l1           0.05128275
## Unemployment.l1  3.80357466
## Target.Rate.l1   3.80857637
## trend.l1        -0.27947791
##
## Weights W:
## (This is the loading matrix)
##
##                     CPI.l1      GDP.l1 Unemployment.l1 Target.Rate.l1
## CPI.d           0.03479378 -0.01158538    -0.011958514    0.002824021
## GDP.d          -0.08133748 -0.43145371    -0.185854266    1.414027842
## Unemployment.d -0.01442374 -0.01181035     0.005804837   -0.025917671
## Target.Rate.d   0.03620897 -0.00151722     0.002180237    0.006137900
##                     trend.l1
## CPI.d          -3.810221e-14
## GDP.d          -2.090861e-12
## Unemployment.d -2.535914e-14
## Target.Rate.d  -4.195417e-15
```

The output shows that the value of the test statitstic is lowest when r = 3.Hence, we can conclude that the CPI level, the unemployment, the target rate, and the GDP growth rate are all cointegrated.

3. Time-series Validation:

```
vnames <- colnames(can.c)
dat <- as.matrix(na.omit(diff(can.c)))

for (lag in 1:3) {
  print("lag:")
  print(lag)

  for (m in 1:3) {

    TT <- nrow(dat)
    T1 <- floor(0.5*TT) # start at 50% of the sample size
    step <- 12 # forecast data horizon for MSE
    tseq <- seq(from=T1, to=TT, by=step)
    tseq <- tseq[-length(tseq)]
    MSE.t <- matrix(0,nrow=tseq[length(tseq)]+step-T1,ncol=length(vnames)) # initialize
    colnames(MSE.t) <- vnames

    for (j in tseq) {

      # VAR model
      if (m==1) {model <- lineVar(data=dat[1:j-1,], lag=lag, model="VAR", I="diff")
                 fcst <- predict(model, n.ahead=step)}

      # VAR model with one threshold
      if (m==2) {model <- TVAR(data=dat[1:j-1,], lag=lag, model="TAR", nthresh=1, trace=F)
                 fcst <- predict(model, n.ahead=step)}

      # VEC model
      if (m==3) {model <- lineVar(data=dat[1:j-1,], lag=lag, r=3, model="VEC")
                 fcst <- predict(model, n.ahead=step)}
      #Note: TVEC model is not implemented in R for more than 2 variables

      js <- j+step-1
      MSE.t[(j-T1+1):(js-T1+1),] <- (dat[j:js,]-fcst)^2
    }

    if (m==1) print("VAR")
    if (m==2) print("TVAR")
    if (m==3) print("VEC")

    MSE <- matrix(colMeans(MSE.t), nrow=1)
    colnames(MSE) <- vnames
    print(MSE)
    print(" ")
  }
}
```

```
## [1] "lag:"
## [1] 1
## [1] "VAR"
```

```
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2816295 4329.473     1.336816  0.03898917
## [1] " "
## [1] "TVAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2249946 1041.803    0.8564946  0.02347067
## [1] " "
## [1] "VEC"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.1613252 4127.366     1.176324  0.04717239
## [1] " "
## [1] "lag:"
## [1] 2
## [1] "VAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2266673 3631.451     1.241869  0.04523393
## [1] " "
## [1] "TVAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.3216202 943.7073    0.6365558  0.02091568
## [1] " "
## [1] "VEC"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2368877 2446.034    0.9640921  0.04847105
## [1] " "
## [1] "lag:"
## [1] 3
## [1] "VAR"
##           CPI      GDP Unemployment Target.Rate
## [1,] 0.284843 2507.723      1.32171  0.04162128
## [1] " "
## [1] "TVAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.3811514 1338.184    0.7212104  0.03413997
## [1] " "
## [1] "VEC"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2432071 2343.247     1.329919  0.04515792
## [1] " "
```

4. Comment

From the result, we are able to tell that VEC(1) has the lowest MSE for CPI, whereas TVAR(1) has the lowest MSE for GDP and Target Rate. TVAR(2) has the lowest MSE for Unemployment.