Graded PS2 Alexander Sanderson

ECO374

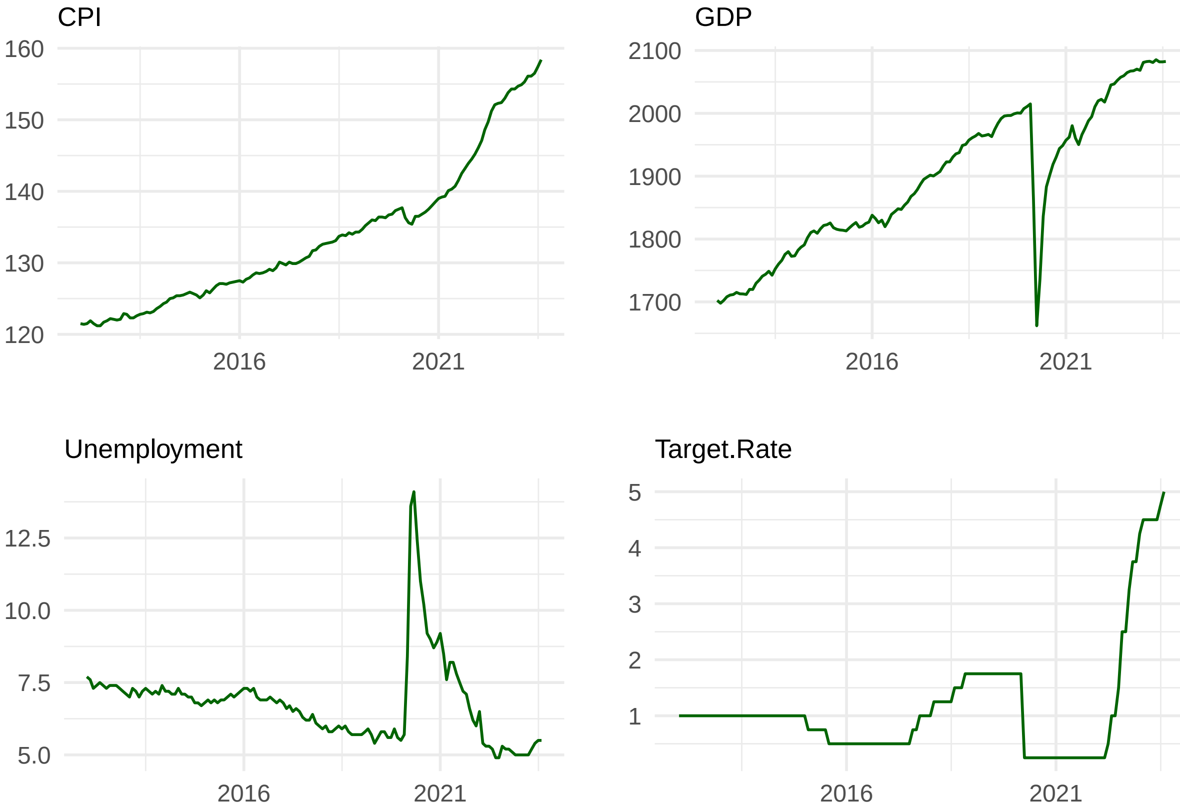
```
if (!require("quantmod")) install.packages("quantmod")
if (!require("xts")) install.packages("xts")
if (!require("ggplot2")) install.packages("ggplot2")
if (!require("ggpubr")) install.packages("ggpubr")
if (!require("tseries")) install.packages("tseries")
if (!require("urca")) install.packages("urca") # Johansen cointegration test
if (!require("tsDyn")) install.packages("tsDyn")
if (!require("timetk")) install.packages("timetk")
if (!require("torch")) install.packages("torch")
if (!require("dplyr")) install.packages("dplyr")
library(quantmod); library(xts); library(ggplot2); library(ggpubr); library(tseries)
library(urca); library(tsDyn); library(timetk); library(torch); library(dplyr)
```

#1 Data

```
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)
Macro_data <- na.omit(Macro_data_can) # balance the panel
Macro_data <- window(Macro_data, start="2012-01-01") # starting date for data
dat <- as.matrix(na.omit(diff(Macro_data)))
vnames <- names(Macro_data)
```

Data Plots

```
plot_data_column = function (column) {
  ggplot(Macro_data) + geom_line(aes(x=index(Macro_data), y=Macro_data[,column]), color="darkgreen")+
    labs(x="", y="", title=paste(column)) +
    theme_minimal() + theme(plot.title = element_text(size=10)) +
    scale_x_date(date_breaks="5 years", date_labels = "%Y")
}
g <- lapply(names(Macro_data), plot_data_column)
ggarrange(g[[1]],g[[2]],g[[3]],g[[4]],ncol=2, nrow=2)
```



#2 Model Selection

```
cointegration <- ca.jo(Macro_data, type="trace", ecdet="trend", spec="transitory")
sc <- summary(cointegration)
sc@test.name
```

```
## [1] "Johansen-Procedure"
```

```
cbind(sc@teststat, sc@cval)
```

```
##                       10pct  5pct  1pct
## r <= 3 |   6.325107 10.49 12.25 16.26
## r <= 2 |  20.514165 22.76 25.32 30.45
## r <= 1 |  56.807481 39.06 42.44 48.45
## r = 0  | 125.353707 59.14 62.99 70.05
```

```
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
```

```r
    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)}

      # TVAR model
      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=2, 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,] 0.177714 1424.555     0.888739  0.06977464
## [1] " "
## [1] "TVAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2494868 1420.974    0.7768526  0.06754752
## [1] " "
## [1] "VEC"
##            CPI     GDP Unemployment Target.Rate
## [1,] 0.1758298 1625.58    0.9688655  0.07239228
## [1] " "
## [1] "lag:"
## [1] 2
## [1] "VAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.1685855 1482.122    0.9357732  0.07099891
## [1] " "
## [1] "TVAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2616673 1423.564    0.7756026  0.06766769
## [1] " "
```

```
## [1] "VEC"
##          CPI      GDP Unemployment Target.Rate
## [1,] 0.17426 1527.344    0.9707654  0.06896058
## [1] " "
## [1] "lag:"
## [1] 3
## [1] "VAR"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.1683077 1494.915    0.8755374  0.07111237
## [1] " "
## [1] "TVAR"
##           CPI      GDP Unemployment Target.Rate
## [1,] 0.259967 1428.992    0.7747945  0.06828228
## [1] " "
## [1] "VEC"
##            CPI      GDP Unemployment Target.Rate
## [1,] 0.2875353 1523.031     0.813772  0.07826657
## [1] " "
```

The VAR model with 3 lags has the lowest time series validation MSE for CPI. The TVAR model with 1 lag has the lowest validation MSE for GDP. The TVAR model with 3 lags has the lowest time series validation MSE for unemployment. Lastly, the TVAR model with 1 lag has the lowest time series validation MSE for Target.Rate.