```
#### R script ########################################
# Last modfied: 2023-07-13
# File name: test_TSEXTEND.R
# Keywords:
#   test_TSEXTEND: test whether bimets::EXTEND() function works
#     as expected or not,
#   Currenly, EXTEND() function does not show changes even I changed EXTMODE options,
#
#
# Written by: Hiroshi Murao
#
# Purpose: perfome the forecast,
#      using a SEM model based on an IS-LM model.
#
# Data source:
# (1) Website of Japan Cabinet Office:
#       https://www.esri.cao.go.jp/jp/sna/data/data_list/kakuhou/files/files_kakuhou.html
# (2) Website of the Bank of Japan:
#       https://www.stat-search.boj.or.jp/
#
# Data information:
#   Data type: Japan quarterly data
#   Time periods of original data set: 1994:01 - 2022:03
#   Time periods of data set for use: 2003:02 - 2022:03
#
# List of variables:
# (1) date : date information such as "1994Q1".
# (2) year :
# (3) quarter : one of (1, 2, 3, 4)
# (4) GDP : Gross Domestic Product, unit: 10 billion yen
# (5) C_prv : Consuption of private sector
# (6) I_prv_jutaku
# (7) I_prv_factory
# (8) I_prv = I_prv_jutaku + I_prv_factory : Investment of private sector
# (9) C_gov: Consuption of public sector
# (10) I_gov : Investment of public sector
# (11) Exports
# (12) Imports
# (13) GDI : Gross Domestic Income
# (14) GNI : Gross National Income
# (15) I_fixed = I_prv + I_gov = I_gross - I_invento
# (16) P_GDP : Deflator of GDP
# (17) P_C_prv : Deflator for C_prv
# (18) P_exports
# (19) P_imports
# (20) P_GNI : Deflator of GNI
# (21) M2 : Money stock M2, unit: 10 billion yen
# (22) M3
# (23) M_base: Monetary Base, unit: 10 billion yen
# (24) R_discount: Interest rate, discount rate, unit: percent (%),
# (25) R_call : Interest rate, call rate
# (26) R_loan_sogo : Interest rate, loan rate, "sogo" type
# (27) R_loan_short
# (28) R_loan_long
# (29) NDI : National Disporsable Income, unit: 10 billion yen
# (30) CPI: Consumer Price Index
# (31) IIP_sAjst: Index of Industrical Production, seasonal adjusted
#
############################################################
#### first things to do
rm(list = ls())
setwd("C:/web_pages/kyoto/study_room/R_bimets/R_prg")

library(systemfit) # Estimating Systems of Simultaneous Equations
library(sem) # Structural Equation Models
library(bimets) # Time Series and Econometric Modeling
search()

#### get data
in_data <- read.csv("./data/data_ISLM.csv",
           na.strings = c("", ".", "NA", "#N/A"),
           stringsAsFactors = FALSE,
           colClasses = c("character", rep("numeric", 28)))
```

```
head(in_data)
sapply(in_data, class)

#### make seasonal dummies
num_inObs <- nrow(in_data)
print(num_inObs)

dum_Qs <- matrix(0, nrow = num_inObs, ncol = 4)

#### start t-loop
for (t in (1:num_inObs)){
  if (in_data[t, "quarter"] == 1) {
    dum_Qs[t, 1] <- 1
  } else if (in_data[t, "quarter"] == 2) {
    dum_Qs[t, 2] <- 1
  } else if (in_data[t, "quarter"] == 3) {
    dum_Qs[t, 3] <- 1
  } else dum_Qs[t, 4] <- 1
}
#### end of t-loop

#### check dum_Qs
head(dum_Qs)
colnames(dum_Qs) <- c("dum_Q1", "dum_Q2", "dum_Q3", "dum_Q4")
dum_Qs <- dum_Qs[, -4]
colnames(dum_Qs)
head(dum_Qs)

#### marge dum_Qs to in_data
in_data <- cbind(as.matrix(in_data), as.matrix(dum_Qs))
in_data <- as.data.frame(in_data)
head(in_data)

## choose the time periods without missing values
num_del <- 37
# num_del <- 57
data_noNA <- in_data[-(1:num_del), ]
head(data_noNA)
# make sure that data_noNA starts from date = 2003Q2.

#### get data for use
data_use <- data_noNA
# data_use <- as.data.frame(subset(data_noNA, data_noNA$year >= 2004))
head(data_use)

#### get variables for possible use
# use as.nemeric() function if "colClasses" is not used.
# unit: trillion yen
# X*4 => growth rates would be annual rates
C_prv <- as.numeric(data_use$C_prv)/1000
I_prv <- as.numeric(data_use$I_prv)/1000
GDP <- as.numeric(data_use$GDP)/1000
IIP <- as.numeric(data_use$IIP_sAjst)/1000
M2 <- as.numeric(data_use$M2)/1000
M3 <- as.numeric(data_use$M3)/1000
M_base <- as.numeric(data_use$M_base)/1000
P_GDP <- as.numeric(data_use$P_GDP)
CPI <- as.numeric(data_use$CPI)
NDI <- as.numeric(data_use$NDI)/1000

# intereste rates are already annual rates
R_discount <- as.numeric(data_use$R_discount)
R_call <- as.numeric(data_use$R_call)
R_loan_sogo <- as.numeric(data_use$R_sogo)

dum_Qs <- data_use[, c("dum_Q1", "dum_Q2", "dum_Q3")]
head(dum_Qs)

# variables for time information
year_org <- as.numeric(data_use$year)
quarter_org <- as.numeric(data_use$quarter)

#### convert date_info to date_var
```

```
date_info <- data_use$date
# date_info contains the value of form "1994Q2".
head(date_info)

## The following function is found from Internet.
date_var <- as.Date(
  sapply(strsplit(date_info, 'Q'),
  function(x) paste(1, seq(1,10,3)[as.integer(x[2])], x[1], sep = '-')),
  format = '%d-%m-%Y')
## end of the function

#### check the output of as.Date() function
head(date_var)
# date with the format of "year-month-day", for example, "1994-04-01".

#### choose variables for the model ##################################################
year <- year_org[-1]
quarter <- quarter_org[-1]
date <- date_var[-1]

Y <- GDP[-1]
Y_lag <- na.omit(lag(GDP, lag = 1))
Y_diff <- na.omit(diff(GDP, lag = 1))
CP <- C_prv[-1]
CP_lag <- na.omit(lag(C_prv, lag = 1))
IP <- I_prv[-1]
IP_lag <- na.omit(lag(I_prv, lag = 1))
R <- R_call[-1]
R_lag <- na.omit(lag(R_call, lag = 1))

YD <- NDI[-1]
TAX <- Y - YD
P <- P_GDP[-1]
M <- M2[-1]
MrP <- (M/P)*100
G <- Y - CP - IP

head(Y)
head(Y_lag)

Y_lag <- Y_lag[-length(Y_lag)]
CP_lag <- CP_lag[-length(CP_lag)]
IP_lag <- IP_lag[-length(IP_lag)]
R_lag <- R_lag[-length(R_lag)]

print(c(length(date), length(year), length(quarter),
      length(CP), length(IP), length(Y), length(YD),
      length(TAX), length(R), length(MrP), length(G),
      length(Y_lag), length(Y_diff)))

#### check the contents of some variables
plot(R, type = "l", lwd = 2)
plot(TAX, type = "l", lwd = 2)
plot(G, type = "l", lwd = 2)

#### set data_obs
data_obs <- data.frame(date = date, year = year, quarter = quarter,
            Y = Y, Y_lag = Y_lag, Y_diff = Y_diff,
            CP = CP, CP_lag = CP_lag,
            IP = IP, IP_lag = IP_lag,
            R = R, R_lag = R_lag,
            MrP = MrP, G = G, TAX = TAX, YD = YD)

head(data_obs) # starting 2003Q3
tail(data_obs) # ending 2022Q3

#### convert data_obs to bimets-type data ###################################
data_bimets <- lapply(as.list(xts(data_obs[, -1],
               order.by = as.Date(data_obs[, 1],"%m/%d/%Y"))),
               as.bimets)

#### make model_text, based on an IS-LM model ###################################
model_text <- "
```

```
MODEL

COMMENT> Consumption Equation
BEHAVIORAL> CP
EQ> CP = a10 + a11*R + a12*(Y - TAX) + a13*TSLAG(CP, 1)
COEFF> a10 a11 a12 a13

COMMENT> Investment Equation
BEHAVIORAL> IP
EQ> IP = a20 + a21*R + a22*TSLAG(IP, 1)
COEFF> a20 a21 a22

COMMENT> Interest Equation
BEHAVIORAL> R
EQ> R = a30 + a31*Y + a32*MrP
COEFF> a30 a31 a32

COMMENT> Maket Clearing Condition for Goods
IDENTITY> Y
EQ> Y = CP + IP + G

END
"

#### load model_text into "bimodel", which is a "bimets model".
#### bimodel is a "model-box" in the bimets enveroment.
bimodel <- LOAD_MODEL(modelText = model_text)

# head(data_bimets)
# tail(data_bimets)

#### load data_bimets into bimodel
bimodel <- LOAD_MODEL_DATA(bimodel, data_bimets)

#### estimate the model and store the estimation result into bimodel ############################
# using bimets::ESTIMATE() function
IV_names <- c("1", "MrP", "G", "TAX", "TSLAG(CP,1)", "TSLAG(IP,1)")
obs_range <- c(2003, 3, 2022, 3)
est_range <- c(2003, 4, 2022, 3)
# Due to the use of TSLAG(x, 1), we need to use "est_range".
# It is important to use the same object name
# for the output of ESIMATE() and the first input of ESIMATE().

bimodel <- ESTIMATE(bimodel,
           TSRANGE = est_range,
           estTech = "IV", IV = IV_names)
           # estTech = "OLS")

summary(bimodel)
# summary.BIMETS.MODEL(bimodel) # This line does not work.

#### make data_tail for forecast
# head(data_obs)
# tail(data_obs)
# using a last part of data_obs
data_tail <- subset(data_obs, year >= 2018)
# Corona pandemic started in 2020
head(data_tail)
tail(data_tail)

#### do "in-sample" forecast and store the result into bimodel ###########################
# using SIMULATE() function in "bimets" package
sim_range <- c(2018, 1, 2022, 3)
bimodel <- SIMULATE(bimodel,
           TSRANGE = sim_range,
           simType = "FORECAST",
           simCovergence = 0.01,
           simIterLimit = 100,
           quietly = TRUE)

#### check the content of out_sim
names(bimodel)
names(bimodel$simulation)
```

```
TABIT(bimodel$simulation$Y)

#### preparations for plotting
Y_sim_ts <- fromBIMETStoTS(bimodel$simulation$Y)
print(Y_sim_ts)
Y_sim <- as.vector(Y_sim_ts)
head(Y_sim)

#### get time horizon
num_H <- length(Y_sim)
x_data <- seq(1, num_H, by = 1)
x_max <- max(x_data)
x_min <- min(x_data)
print(c(x_min, x_max))

#### graph for Y,
Y_graph <- data.frame(Y = data_tail$Y, Y_sim = Y_sim)
head(Y_graph)

Y_min <- min(Y_graph)
Y_max <- max(Y_graph)
print(c(Y_min, Y_max))

plot.new()
matplot(x_data, Y_graph,
    type = "l", lwd = 2,
    lty = c("solid", "dotted", "dashed", "dotdash"),
    xlim = c(x_min, x_max + 4),
    ylim = c(120, 150),
    xlab = "time (quarter)",
    ylab = "GDP (trillion yen)",
    main = "Use of bimets::SIMULATE(), Y, (2018Q1-2022Q3)")

### additional information to the graph
# abline(h = 0, col = "gray")
text(x_max, Y_graph[(x_data == x_max), 1], labels = "Y_obs", pos = 4)
text(x_max, Y_graph[(x_data == x_max), 2], labels = "Y_frcst", pos = 4)
# pos = 1: below,  pos = 2: left,  pos = 3: up,  pos = 4: right.

#### save the graph into a file
# dev.copy(jpeg, file = "./outputs/forecast_bimets_in_sample.jpg")
# dev.off()

#### preparing for "out-sample" forecast, #############################################
# The last observed time period: "2022Q3"
# Extend time series, from "2022Q4" to "2024Q4".
# make the extention for exogenous variables in "modelData".

bimodel$modelData <- within(bimodel$modelData,{
  G = TSEXTEND(G, UPTO = c(2024,4), EXTMODE = "QUADRATIC")
  # G <- TSEXTEND(G, UPTO = c(2024, 4), EXTMODE = "CONSTANT")
  # G = TSEXTEND(G, UPTO = c(2024, 4), EXTMODE = "MYRATE", FACTOR = 30)
  MrP = TSEXTEND(MrP, UPTO = c(2024, 4), EXTMODE = "QUADRATIC")
  TAX = TSEXTEND(TAX, UPTO = c(2024, 4), EXTMODE = "QUADRATIC")
})
# end of TSEXTEND()

# check the content of extended data
TABIT(bimodel$modelData$G)

TABIT(bimodel$modelData$G,
      bimodel$modelData$MrP,
      bimodel$modelData$TAX)


#### Comments
# (1)
# (2)

#### end ##############################################
```