```
#### R script ##########################################
# Last modfied: 2023-07-27
# File name: sim_use_TSEXTEND_shock_G.R
# Keywords:
#   sim: simulation analysis based on a simultaneous-equation model (SEM),
#        specifically based on an IS-LM model,
#   use_TSEXTEND: use of TSEXTEND() function,
#   Y: response = Y,
#   shock_G: shock = G,
#       increase by 10% annually from the base projection,
#       10% up for G is about one standard deviation of G,
#
# Written by: Hiroshi Murao
#
# Purpose: perfome the dynamic simulation for policy analysis,
#      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 data_bimets, which is a data set to bimets.
data_bimets <- lapply(as.list(xts(data_obs[, -1],
               order.by = as.Date(data_obs[, 1],"%Y-%m-%d"))),
               as.bimets)
```

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

COMMENT> Consumption Equation
COMMENT> The use of TSLAS(CP, 1) does not work.
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) + a23*TSDELTA(TSLAG(IP, 1), 1)
COEFF> a20 a21 a22 a23

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 <- 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 esimation result into bimodel
# using bimets::ESTIMATE() function
obs_range <- c(2003, 3, 2022, 3)
est_range <- c(2004, 1, 2022, 3)
# Due to the use of TSDELTA(TSLAG(IP, 1), 1), we need to use "est_range".
IV_names <- c("1", "MrP", "G", "TAX",
              "TSLAG(CP,1)", "TSLAG(IP,1)", "TSDELTA(TSLAG(IP, 1), 1)")

# It is important to use the same object name for the output of ESIMATE() and the first input of
bimodel <- ESTIMATE(bimodel,
           TSRANGE = est_range,
           estTech = "IV", IV = IV_names)

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

#### (1) Base projection ##############################################################
# Base projection: (G, MrP, TAX) grow 2% annually.
sim_range <- c(2018, 1, 2024, 4)

#### make hypothetical data for the base projection
bimodel$modelData <- within(bimodel$modelData,{
G = TSEXTEND(G, UPTO = c(2024, 4), EXTMODE = "MYRATE", FACTOR = 1.005)
MrP = TSEXTEND(MrP, UPTO = c(2024, 4), EXTMODE = "MYRATE", FACTOR = 1.005)
TAX = TSEXTEND(TAX, UPTO = c(2024, 4), EXTMODE = "MYRATE", FACTOR = 1.005)
})
# FACTOR = 1.01: 1% growth rate per quarter, 2% annually seems all right for the base projection.
# end of TSEXTEND()

# check the content of extended data
TABIT(bimodel$modelData$G, TSRANGE = sim_range)

#### do the base projection
# using bimets::SIMULATE() function
bimodel <- SIMULATE(bimodel,
           TSRANGE = sim_range,
```

```
          simType = "DYNAMIC",
          simAlgo = "NEWTON",
          simCovergence = 0.01,
          simIterLimit = 100,
          quietly = TRUE)

#### check the content of base projection
names(bimodel)
names(bimodel$simulation)
TABIT(bimodel$simulation$Y)

#### preparations for plotting
# simulation value of base projection
Y_base_ts <- fromBIMETStoTS(bimodel$simulation$Y)
print(Y_base_ts)
Y_base <- as.vector(Y_base_ts)
head(Y_base)

# num_both = num_part1 + num_part2 for "out-sample" simulatio
num_both <- length(Y_base)
print(num_both)

##### keep the content of bimodel in bimodel_V1
bimodel_V1 <- bimodel
# check "bimodel_V1$modelData$G" before "upG" projection
TABIT(bimodel_V1$modelData$G, TSRANGE = sim_range)

#### (2) "upG" projection ###############################################################
#### important notes
# (1) TSEXTEND() function do not change the previous values made with the function.
# (2) TSEXTEND() function changes the previous values if we run the whole R script again.
# (3) Due to this probrem, we follow "strange" process as follows.

#### load model_text into bimodel_V2
bimodel_V2 <- LOAD_MODEL(modelText = model_text)

#### load data_bimets into bimodel_V2
bimodel_V2 <- LOAD_MODEL_DATA(bimodel_V2, data_bimets)
# names(bimodel_V2)

#### estimate the model, again
bimodel_V2 <- ESTIMATE(bimodel_V2,
              TSRANGE = est_range,
              estTech = "IV", IV = IV_names)

#### get some basic information for "upG" projection
# G is inceared by one standard deviation
mean_G <- mean(G)
var_G <- var(G)
std_G <- sqrt(var_G)
print(mean_G)
print(std_G)
shock_size = std_G/mean_G
print(shock_size)
# [1] 0.08908354 = 9% up => about 10% up

#### make hypothetical data for "upG" projection
# sim_range <- c(2018, 1, 2024, 4)
bimodel_V2$modelData <- within(bimodel_V2$modelData,{
  G = TSEXTEND(G, UPTO = c(2024,4), EXTMODE = 'MYRATE', FACTOR = 1.03)
  MrP = TSEXTEND(MrP, UPTO = c(2024, 4), EXTMODE = "MYRATE", FACTOR = 1.005)
  TAX = TSEXTEND(TAX, UPTO = c(2024, 4), EXTMODE = "MYRATE", FACTOR = 1.005)
})
# Base projection: G, FACTOR = 1.005 => 0.005*4 = 0.02 = 2% for annual growth rate
# "upG" projection, 10% up from the base projection,
# FACTOR = 1.03  means 0.03*4 = 0.12 = 12% for annual growth rate
# 12% - 2% = 10% up, which is about one standard devation of G.

#### check the data for "upG" projection
TABIT(bimodel_V2$modelData$G, TSRANGE = sim_range)

#### do "upG" projection
# sim_range <- c(2018, 1, 2024, 4)
```

```
bimodel_V2 <- SIMULATE(bimodel_V2,
              TSRANGE = sim_range,
              simType = "DYNAMIC",
              simAlgo = "NEWTON",
              simCovergence = 0.01,
              simIterLimit = 100,
              quietly = TRUE)

# get the simulation value of "upG",
Y_upG_ts <- fromBIMETStoTS(bimodel_V2$simulation$Y)
print(Y_upG_ts)
Y_upG <- as.vector(Y_upG_ts)
head(Y_upG)

# Y_upG <- as.vector(fromBIMETStoTS(bimodel_V2$simulation$Y))

#### make data_tail for graphic output
# 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)
Y_obs_tail <- data_tail$Y
head(Y_obs_tail)

num_part1 <- length(Y_obs_tail)
print(num_part1)
# num_both = num_part1 + num_part2 for "out-sample" simulatio
num_part2 <- num_both - num_part1
print(c(num_both, num_part1, num_part2))

Y_obs_both <- matrix(NA, nrow = num_both, 1)
Y_obs_both[1:num_part1] <- Y_obs_tail
print(Y_obs_both)

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

#### get Y_graph
Y_graph <- data.frame(Y = Y_obs_both, Y_base = Y_base, Y_upG = Y_upG)
colnames(Y_graph) <- c("Y", "Y_base", "Y_upG")
head(Y_graph)

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

#### plot Y_graph
plot.new()
matplot(x_data, Y_graph,
    type = "l", lwd = 2,
    lty = c("solid", "dotted", "dashed",  "dotdash"),
    xlim = c(x_min, x_max + 5),
    ylim = c(120, 150),
    xlab = "time (quarter)",
    ylab = "GDP (trillion yen)",
    main = "y = Y, x = G, Use of TSEXTEND()",
    sub = "in-sample = (2018Q1 - 2022Q3), out-sample = (2022Q4 - 2024Q4)"
)

### 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_base", pos = 4)
text(x_max, Y_graph[(x_data == x_max), 3], labels = "Y_upG", 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/sim_use_TSEXTEND_shock_G.jpg")
# dev.off()

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

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