```
#*********************************************
# Randall R. Rojas
# Email: rrojas@econ.ucla.edu
# Date: 05/10/2015
# Comment(s): R code for fitting VAR models
# Data File(s): housecomp.dat
#*********************************************
# Variable Definitions
# U.S. monthly seasonally adjusred housing starts and completion
# from 1968.01-1996.06
# housecomp.dat[,2] = housing starts
# housecomp.dat[,3] = hoising completion
#*********************************************

# Set your 'working directory' to the folder where all the data and respective codes are located.
#setwd("/Users/DrDrR4/Documents/Courses/2015/Spring/Econ144/R_Codes")

# Clear all variables and prior sessions
rm(list=ls(all=TRUE))

# Load Libraries
library(lattice)
library(foreign)
library(MASS)
library(car)
require(stats)
require(stats4)
library(KernSmooth)
library(fastICA)
library(cluster)
library(leaps)
library(mgcv)
library(rpart)
library(pan)
library(mgcv)
library(DAAG)
library(TTR)
library(tis)
require("datasets")
require(graphics)
library(forecast)
#install.packages("astsa")
#require(astsa)
library(xtable)
# New libraries added:
library(stats)
library(TSA)
library(timeSeries)
library(fUnitRoots)
library(fBasics)
library(tseries)
library(timsac)
library(TTR)
library(fpp)
library(strucchange)
#library(MSBVAR)
library(vars)
library(lmtest)
library(dlnm)

# Look at the data
data=read.table("housecomp.dat")
starts<-ts(data[,2],start=1968.1,freq=12)
comps<-ts(data[,3],start=1968.1,freq=12)
quartz()

plot(starts)
nberShade()
lines(starts,ylab="Housing Starts and Completions")
```

```
lines(comps,col="blue")
legend("topright",legend=c("Starts","Completions"),text.col=c("black","blue"),bty="n")


# Look at the ACF, PACF, and CCF (cros-correlation function)
quartz()
tsdisplay(starts,main="Housing Starts")
quartz()
tsdisplay(comps,main="Housing Completions")
quartz()
ccf(starts,comps,ylab="Cross-Correlation Function", main = "Starts and Completions CCF")
# Completions are maximally correlated with starts lagged by 6-12 months.


# Fit a VAR(p) model to the data
# Note, we need to combine the variables into 1 data frame first:
y=cbind(starts, comps)
#y_ts=ts.union(starts, comps) # You can also use this function
y_tot=data.frame(y)

# To fit a VAR(p) model, simply call 'VAR' and set p=value
y_model=VAR(y_tot,p=4)
summary(y_model)
# We interpret the coefficients in the usual way,but now have a
# system of equations. For example, for VAR(1) we have:
# y1 = c11 y(1,t-1) + c12 y(2,t-1)
# y2 = c21 y(1,t-1) + c22 y(2,t-1)
# The ourtput from summary are cij, cov, and corr.

# Plot the fit and orginal data
quartz()
plot(y_model)
#pdf("varplot.pdf", width=8, height=8)
#plot(y_model)
#dev.off()

# Look at ACF and PACf
quartz()
par(mfrow=c(2,1))
acf(residuals(y_model)[,1])
pacf(residuals(y_model)[,1])

quartz()
par(mfrow=c(2,1))
acf(residuals(y_model)[,2])
pacf(residuals(y_model)[,2])

# or even better
quartz()
tsdisplay(residuals(y_model)[,2],main ="Comps = starts(t-k) + comps(t-k)")
# Impulse Response Function
irf(y_model)
#pdf("irf.pdf", width=8, height=8)
quartz()
plot(irf(y_model, n.ahead=36))
#dev.off()

#Forecast
#holdout_matrix = hold out data
#var.predict = predict(object=y_model, n.ahead=52, dumvar=holdout_matrix);
var.predict = predict(object=y_model, n.ahead=52)
quartz()
plot(var.predict)

dev.print(device=postscript,"forecast.eps",width=7,height=7, horizontal=FALSE)
dev.off()

#Granger Test
#Does LA granger-cause Riverside?
```

```
grangertest(comps ~ starts, order = 8)

#Variance Decomposition (Forecast Error Variance Decomposition)
quartz()
plot(fevd(y_model, n.ahead = 5))

#CUSUM Plot
quartz()
plot(stability(y_model, type = "Rec-CUSUM"), plot.type="single")
```