Financial Econometrics

R Commands Used in Lecture 5

Prof. Hamed Ghoddusi

2019

Package: Multivariate Time Series (MTS)

MTS is General package for:

- Analyzing multivariate linear time series
- Estimating multivariate volatility models
- Handling factor models and constrained factor models
- Asymptotic principal component analysis, and principal volatility component analysis

Commands: Multivariate Time Series Analysis

```
> require(MTS) ### Package used. You need to install it.
> da=read.table("q-gdpun.txt",header=T)
> head(da)
> xt=da[,4:5]
> MTSplot(xt) #### Plot multiple time series on a page
> tdx=da[,1]+da[,2]/12
> MTSplot(xt,tdx) ## with actual calendar date
> ccm(xt)
> rt=diffM(xt) ### differencing all series
> ccm(rt) ### Cross-correlation matrices
> mq(rt,lag=10)
```

VAR Model

- Estimates a VAR by OLS per equation. The model is of the following form

$$y_t = A_1 y_{t-1} + \ldots + A_p y_{t-p} + u_t$$

Where:

- y_t is a **K×1** vector of endogenous variables.
- u_t assigns a spherical disturbance term of the same dimension.
- The coefficient matrices $A_1,\ldots,A_p$ are of dimension **K×K**.
- No seasonality or trend term can be included in the model.

Commands: VAR Model

Estimates A Vector Autoregressive Model Of Order p**.**

- Estimation of a Vector Autoregressive model (VAR) by computing OLS per equation.

Usage

- **VAR(y, p = 1, exogen = NULL)**

Arguments

y	Endogenous variable for the VAR model.
P	lag-order for the autoregressive model.
Exogen	Exogenous variable for the VAR model.

Examples

```
> library(LPTime)
> data(EyeTrack.sample)
> head( VAR(y = EyeTrack.sample, p = 2))
```

Commands: Fitting and Predicting using a VAR Model

```
data("mts-examples",package="MTS")
> head(qgdp)
> gdp=log(da[,3:5]) ## pick up the GDP series
> zt=cbind(gdp$ca,gdp$us)
> MTSplot(zt)
> gt=diffM(zt)
> VARorder(gt) ### Order specification
> m1=VAR(gt,3) ### Fit a VAR(3) model
> MTSdiag(m1) ### Model checking
> VARpred(m1,4) ### prediction. 1-step to 4-step forecasts.
```

Commands: Co-integration and Pairs Trading

```
> da=read.table("d-bhp0206.txt",header=T) ### Loading dataset 1
 > da1=read.table("d-vale0206.txt",header=T) ### Loading dataset 2
> bhp=log(da$adjclose)
> vale=log(da1$adjclose)
> m1=lm(bhp~vale)
> wt=bhp-0.717*vale
> plot(wt,type='l')
> abline(h=c(1.82)
> abline(h=c(1.85),col="red")
> abline(h=c(1.79),col="blue")
```

Commands: Multivariate Volatility Modeling

```
da=read.table("d-ibmbaml3a-0110.txt",header=T)
> rtn=log(da[,2:3]+1)
> cor(rtn)
> m2=EWMAvol(rtn) ## Use default discounting rate 0.96
> names(m2)
> Sigma.t = m2$Sigma.t
> MCHdiag(rt,Sigma.t) ### Model checking
> rho=Sigma.t[,2]/sqrt(Sigma.t[,1]*Sigma.t[,4])
> ts.plot(rho)
> mean(rho)
> m3=EWMAvol(rtn,-0.1) ## -0.1 means estimating lambda
```

```
m4=dccPre(rtn)
> names(m4)
> sresi=m4$sresi
 > m5=dccFit(sresi,ub=c(0.981,0.01899))
> names(m5)
> ts.plot(m5$rho.t[,2])
> apply(m5$rho.t,2,mean)
```

Commands: Two Dimensional Multivariate Volatility Modeling

```
da=read.table("m-mcd3dx6614.txt",header=T)
> rt = log(da[,c(3,6)]+1)
> mm = BEKK11(rt) ### This will take a while to complete.
> names(mm)
> S.t = mm$Sigma.t
> MCHdiag(rt,S.t) ### Model checking
> rho.t = S.t[,2]/sqrt(S.t[,1]*S.t[,4])
> ts.plot(rho.t)
> mean(rho.t)
```

Command: Johansen Procedure For VAR

Conducts the Johansen procedure on a given data set.

- The **"trace"** or **"eigen"** statistics are reported and the matrix of eigenvectors as well as the loading matrix.

Usage

```
> ca.jo(x, type = c("eigen", "trace"), ecdet = c("none", "const", "trend"), K = 2, spec=c("longrun", "transitory"), season = NULL, dumvar = NULL)
```

Arguments

x	Data matrix to be investigated for cointegration.
Type	The test to be conducted, either eigen or trace.
Ecdet.	Character, none for no intercept in cointegration, const for constant term in cointegration and trend for trend variable in cointegration.
K	The lag order of the series (levels) in the VAR.
Spec	Determines the specification of the VECM, see details below.
Season	If seasonal dummies should be included, the data frequency must be set accordingly, *i.e* 4 for quarterly data.
Dumvar	If dummy variables should be included, a matrix with row dimension equal to x can be provided.

Command: Finding the Relationship Between Variables

```
> da=read.table("d-bhp0206.txt",header=T)
> da1=read.table("d-vale0206.txt",header=T)
> head(da)
  Mon day year  open  high   low close volume adjclose
1   7   1 2002 11.80 11.92 11.55 11.60 156700     8.39
....
6   7   9 2002 12.25 12.65 12.25 12.60 142000     9.12
> head(da1)
  Mon day year  open  high   low close  volume adjclose
1   7   1 2002 27.60 27.60 27.10 27.16 2307600     1.89
....
6   7   9 2002 27.05 27.55 27.05 27.30 2534400     1.90
> dim(da)
[1] 946  9
>  bhp=log(da[,9])
> vale=log(da1[,9])
> plot(bhp,type='l')
> plot(vale,type='l')
> m1=lm(bhp~vale)
> summary(m1)

Call:
lm(formula = bhp ~ vale)

Residuals:
      Min        1Q    Median        3Q       Max
-0.151818 -0.028265  0.003121  0.029803  0.147105

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.822648   0.003662   497.7   <2e-16 ***
vale        0.716664   0.002354   304.4   <2e-16 ***
---
Residual standard error: 0.04421 on 944 degrees of freedom
Multiple R-squared: 0.9899,     Adjusted R-squared: 0.9899
F-statistic: 9.266e+04 on 1 and 944 DF,  p-value: < 2.2e-16
```

Command: Cointegration Tests

```
> library(urca)
> help(ca.jo) # Johansen's co-integration test
> bhp1=ts(bhp,frequency=252,start=c(2002,127))
> vale1=ts(vale,frequency=252,start=c(2002,127))
> plot(bhp1,type='l')
> plot(vale1,type='l')
> x=cbind(bhp,vale)
> m1=ar(x)
> m1$order
[1] 2
> m2=ca.jo(x, K=2)
> summary(m2)
```

Command: Johansen Procedure

Test type: maximal eigenvalue statistic (lambda max) , with linear trend

Eigenvalues (lambda):

[1] 0.0406019854 0.0000101517

Values of test statistic and critical values of test:

```
          test 10pct  5pct  1pct
r <= 1 |  0.01  6.50  8.18 11.65
r = 0  | 39.13 12.91 14.90 19.19
```

Eigenvectors, normalised to first column: **(These are the cointegration relations)**

```
            bhp.l2  vale.l2
bhp.l2    1.000000 1.000000
vale.l2 -0.717784 2.668019
```

Weights W: (This is the loading matrix)

```
             bhp.l2       vale.l2
bhp.d  -0.06272119 -2.179372e-05
vale.d  0.03303036 -3.274248e-05
```

Command: Johansen Procedure

```
> wt=bhp-0.718*vale
> acf(wt)
> pacf(wt)
> m4=arima(wt,order=c(2,0,0))
> m4
```

```
Call:
arima(x = wt, order = c(2, 0, 0))

Coefficients:
         ar1     ar2  intercept
      0.8050  0.1215      1.820
s.e.  0.0323  0.0325      0.008

sigma^2 estimated as 0.000333:  log likelihood = 2444.26,  aic = -4880.52
```

```
> tsdiag(m4)
> plot(wt,type='l')
```