Financial Econometrics

R Commands Used in Lecture 4

Prof. Hamed Ghoddusi

2019

Commands: Loading and Viewing the Data

```
> require(quantmod)
> getSymbols("UNRATE",src="FRED") <=== loading the data
[1] "UNRATE"
> dim(UNRATE)  <=== returning the lengths of the row.names
[1] 831   1
> head(UNRATE)  <=== observing the top rows (by date)
           UNRATE
1948-01-01    3.4
1948-02-01    3.8
1948-03-01    4.0
1948-04-01    3.9
1948-05-01    3.5
1948-06-01    3.6
> rate  <-  as.numeric(UNRATE[,1])
> ts.plot(rate)
```

Commands: AR Order Selection Using AIC

```
> m1 <- ar(rate, order.max=15) ## AR order selection using AIC
> m1$order
[1] 13
> m2 <- arima(rate, order=c(13,0,0))
> m2
Call: arima(x = rate, order = c(13, 0, 0))

Coefficients:
         ar1     ar2      ar3      ar4     ar5      ar6      ar7     ar8
      1.0045  0.1911  -0.0594  -0.0457  0.0399  -0.1287  -0.0432  0.0431
s.e.  0.0346  0.0488   0.0492   0.0491  0.0493   0.0493   0.0495  0.0493
         ar9     ar10    ar11     ar12    ar13  intercept
     -0.0040  -0.0824  0.1145  -0.1642  0.1211     5.6337
s.e.  0.0493   0.0492  0.0494   0.0490  0.0348     0.4787

sigma^2 estimated as 0.0366:  log likelihood = 192.75,  aic = -355.49
> tsdiag(m2,gof=36)
```

Commands: Model Refinement

```
> c1 <- c(NA,NA,0,0,0,NA,0,0,0,NA,NA,NA,NA,NA)
> m3 <- arima(rate, order=c(13,0,0), fixed=c1)
Warning message: In arima(rate, order = c(13, 0, 0), fixed = c1) :
some AR parameters were fixed: setting transform.pars = FALSE
> m3
Call: arima(x = rate, order = c(13, 0, 0), fixed = c1)

Coefficients:
         ar1     ar2  ar3  ar4  ar5      ar6  ar7  ar8  ar9     ar10    ar11
      0.9992  0.1406    0    0    0  -0.1528    0    0    0  -0.0718  0.1169
s.e.  0.0340  0.0404    0    0    0   0.0264    0    0    0   0.0410  0.0487
         ar12    ar13  intercept
      -0.1744  0.1287     5.6368
s.e.   0.0488  0.0345     0.4738

sigma^2 estimated as 0.0368:  log likelihood = 190.42,  aic = -362.84
```

- **The AIC reduces to -362.84 indicating that it is ok to set those parameters to zero (as given in c1).**

Commands: Using ARIMA Models

> require(forecast)

> auto.arima(rate) <=== **specifying the ARIMA model automatically**

Series: rate

ARIMA(2,1,2)

```
Coefficients:
          ar1      ar2      ma1     ma2
      1.6537  -0.7759  -1.6319  0.8487
s.e.  0.0414   0.0459   0.0408  0.0474

sigma^2 estimated as 0.03831:  log likelihood=177.6
AIC=-345.19   AICc=-345.12   BIC=-321.59
```

Commands: Using ARIMA Models

> m4 <- arima(rate,order=c(2,1,2))

> m4

```
Call: arima(x = rate, order = c(2, 1, 2))

Coefficients:
          ar1      ar2      ma1     ma2
      1.6537  -0.7759  -1.6319  0.8487
s.e.  0.0414   0.0459   0.0408  0.0474

sigma^2 estimated as 0.03813:  log likelihood = 177.6,  aic = -345.19
```

> tsdiag (m4,gof=36) <=== **The model checking result shows some serial correlations remain in the residuals.**

Model Comparison

Two types of comparison:

A). **In-sample fit**

- Selecting one of the criteria, such as AIC.
- Selecting the model with smaller AIC as the preferred model.
- For unemployment rate, model "m3" has AIC = -362.84. On the other hand, model "m4" has AIC = -345.19. So model "m3" is selected.

B). **Out-of-sample comparison**

- Dividing the data into two subsamples: training sample and forecasting sample
- This is done by selecting a forecast origin. For example, we use **t = 770** as the forecast origin.

Back Testing (Out of Sample Comparison)

- Using a rolling of estimation-prediction to compute **1-step** ahead forecasts for a given model, starting with forecast chosen forecast origin.

- We estimate the model using the first 770 data points, then forecast 771 to compute forecast error. We then re-estimate the model using the first 771 data points and predict t=772 to compute forecast error. This procedure is repeated until the end of the sample.
- In unempolyment example, we estimate the model using 830 data points, then predict t=831. The forecast errors can be used to compute **mean squared forecast error (MSE)** and **mean absolute forecast error (MAE)**.

- If MSE is used, one selects the model with the smallest MSE as the preferred model.
- The **"backtest"** can be done using the R script "backtest.R".

Commands: Back Testing

```
> source("backtest.R")
> backtest(m3,rate,770,fixed=c1)
[1] "RMSE of out-of-sample forecasts"
[1] 0.1455117
[1] "Mean absolute error of out-of-sample forecasts"
[1] 0.118224
There were 50 or more warnings (use warnings() to see the first 50)
> backtest(m4,rate,770)
[1] "RMSE of out-of-sample forecasts"
[1] 0.1482988
[1] "Mean absolute error of out-of-sample forecasts"
[1] 0.1180253
```

- From the output, model "m3" is preferred based on MSE. But "m4" is preferred based on MAE.
- A drawback of "backtest" comparison is that the result may depend on the forecast origin.
- One should use sufficient number of data points in the forecasting subsample to obtain more reliable comparison.

Commands: Exponential Smoothing

```
require(quantmod)
> getSymbols("^IXIC") <== loading daily values of NASDAQ index
> chartSeries(IXIC)
> rt = as.numeric(IXIC[,6])
> lrt = log(rt)
> acf(lrt)
> drt <- diff(lrt)
> acf(drt) > ms = arima(lrt,order=c(0,1,1))
> ms <== shows the estimate ma1, which is the discounting rate
```

Commands: Seasonal Model

```
> da=read.table("q-earn-jnj.txt") <== loading the data
> jnj=da[,1]
> ts.plot(jnj)
> ljnj=log(jnj)
> ts.plot(ljnj)
> acf(ljnj)
> djnj=diff(ljnj)
> acf(djnj,lag=20)
```

```
> dd <- diff(djnj,4) <== seasonal difference
> acf(dd)
> m5 =
arima(ljnj,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=4))
> m5
> tsdiag(m5,gof=12)
> predict(m5,8)
```

Commands: Seasonal Model

```
> da=read.table("m-houst-5917.txt",header=T) <== loading the data
> hs=da[,2]
> ts.plot(hs)
> acf(hs)
> dhs=diff(hs)
> acf(dhs,lag=36)
> ddhs <- diff(dhs,12)
> acf(ddhs)
> m6=arima(hs,order=c(0,1,1),seasonal=list(order=c(0,1,1),period=12))
> m6
> tsdiag(m6,gof=36)
```

Commands: Regression Models with Time Series Errors

```
> da=read.table("m-tb3n6.txt", header=T) <== loading the data
> tb3=da[,3]
> tb6=da[,4]
> m7=lm(tb6~tb3)
> summary(m7)
> names(m7)
> ts.plot(m7$residuals)
> acf(m7$residuals)
> m8=ar(m7$residuals, mehtod="mle")
> m8$order
> m9=arima(tb6,order=c(3,0,0), xreg=tb3)
> m9
> tsdiag(m9)
```

Commands: Handling Outliers

```
> names(m2a)
 [1] "coef"      "sigma2"    "var.coef"  "mask"      "loglik"    "aic"
 [7] "arma"      "residuals" "call"      "series"    "code"      "n.cond"
[13] "nobs"      "model"
> which.min(m2a$residuals)
[1] 23
> i23 <- rep(0,819)
> i23[23]=1
> m2b <- arima(rate,order=c(11,0,0), xreg=i23)
> m2b
Call: arima(x = rate, order = c(11, 0, 0), xreg = i23)

Coefficients:
         ar1     ar2      ar3      ar4      ar5      ar6      ar7     ar8
      1.0453  0.1298  -0.0485  -0.0219   0.0156  -0.1083  -0.0665  0.0610
s.e.  0.0354  0.0524   0.0514   0.0508   0.0513   0.0509   0.0507  0.0508
         ar9    ar10    ar11  intercept      i23
     -0.0304  -0.0935  0.103     5.6981  -0.7710
s.e.  0.0508   0.0506  0.035     0.4469   0.1362

sigma^2 estimated as 0.03596:  log likelihood = 197.16,  aic = -366.32
```

Commands: Handling Outliers

```
> c2 <- c(NA,NA,0,0,0,NA,0,0,0,NA,NA,NA,NA)
> m2c <- arima (rate, order=c(11,0,0), xreg=i23, fixed=c2)
Warning message: In arima(rate, order = c(11, 0, 0), xreg = i23, fixed = c2) :
some AR parameters were fixed: setting transform.pars = FALSE
 > m2c
Call: arima(x = rate, order = c(11, 0, 0), xreg = i23, fixed = c2)

Coefficients:
         ar1     ar2  ar3  ar4  ar5      ar6  ar7  ar8  ar9     ar10    ar11
      1.0458  0.0866    0    0    0  -0.1497    0    0    0  -0.1005  0.1032
s.e.  0.0352  0.0422    0    0    0   0.0254    0    0    0   0.0415  0.0347
      intercept      i23
         5.6964  -0.8014
s.e.     0.4404   0.1325

sigma^2 estimated as 0.03615:  log likelihood = 195.06,  aic = -374.11
> tsdiag(m2c,gof=24)
```

Commands: Handling Outliers

```
> which.max(m2c$residuals)
[1] 22
> i22 <- rep(0,819)
> i22[22]=1
> x <- cbind(i22, i23)
> c3 <- c(c2, NA)
> m2d <- arima(rate,order=c(11,0,0),xreg=x,fixed=c3)
Warning message: In arima(rate, order = c(11, 0, 0), xreg = x, fixed =
c3)
: some AR parameters were fixed: setting transform.pars = FALSE
> m2d
Call: arima(x = rate, order = c(11, 0, 0), xreg = x, fixed = c3)
> tsdiag(m2d, gof=24)
```

```
Coefficients:
          ar1     ar2  ar3  ar4  ar5      ar6  ar7  ar8  ar9     ar10    ar11
      1.0747  0.0496    0    0    0  -0.1375    0    0    0  -0.0938  0.0927
s.e.  0.0349  0.0420    0    0    0   0.0250    0    0    0   0.0419  0.0350
      intercept     i22      i23
         5.6993  1.1573  -0.2809
s.e.     0.4328  0.1411   0.1400

sigma^2 estimated as 0.03339:  log likelihood = 227.52,  aic = -437.04
```

Commands: ADF Test (Package: tseries)

- Performs the Augmented Dickey-Fuller test for the null hypothesis of a unit root of a univarate time series x (**equivalently, x is a non-stationary time series**).

- **Usage**

> require(tseries)

> adf.test(x, nlag = NULL, output = TRUE)

- **Arguments**

x a numeric vector or univariate time series.

nlag the lag order with default to calculate the test statistic.

Output a logical value indicating to print the test results in R console. The default is TRUE.

- **Example**

> x <- arima.sim(list(order = c(1,0,0),ar = 0.2),n = 100) <== **ADF test for AR(1) process**

> adf.test(x)

Commands: ADF Test (Package: urca)

- **Usage**

```
> require(urca)
> ur.df(y, type = c("none", "drift", "trend"), lags = 1, selectlags = c("Fixed", "AIC", "BIC"))
```

- **Arguments**

y Vector to be tested for a unit root

type Test type, either "none", "drift" or "trend".

lags Number of lags for endogenous variable to be included.

Selectlags Lag selection can be achieved according to the Akaike **"AIC"** or the Bayes **"BIC"** information criteria. The maximum number of lags considered is set by lags. The default is to use a "fixed" lag length set by lags.

- **Example**

```
> x <- ur.df(y=lc, lags=3, type='trend')
> summary(x)
```

ADF Test (Package: urca)

- If type is set to "none" neither an intercept nor a trend is included in the test regression.
- If it is set to "drift" an intercept is added and if it is set to "trend" both an intercept and a trend is added.
- The critical values are taken from Hamilton (1994) and Dickey and Fuller(1981).