Financial Econometrics

R Commands used in Lecture 3

Prof. Hamed Ghoddusi

2019

Commands: Loading the Data

```
> gnp=scan(file="dgnp82.txt")
```

- **It only works for a single variable in the data file without heading.**

```
> read.table ("m-dec12910-6114.txt",header=T)
```

- **Loading the data with or without heading and multiple variables.**

Commands: Plotting Time Series

```
ts.plot(gnp) ## time-series plot
> rtn <- read.table("m-dec12910-6114.txt",header=T)
> head(rtn)
> ts.plot(rtn$dec1)
> source("lagplot.R") ## compile the R script lagplot.R
> lagplot(gnp)
> lagplot(gnp,lag=2)
```

Commands: Computing ACF

> acf(gnp)

> acf(gnp,lag=20) <== **specify the number of ACF to compute**

> acf(rtn$dec9)

> Box.test(gnp,lag=10,type="Ljung") <== **Compute Ljung-Box Q(m) statistics**

Simulate From an ARIMA MODEL (arima.sim)

- **Usage**

> y <-arima.sim(model, n, rand.gen = rnorm, innov = rand.gen(n, …), n.start = NA, start.innov = rand.gen(n.start, …), …)

- **Arguments**

- **Model:** A list with component **ar** and/or **ma** giving the **AR** and **MA** coefficients respectively. Optionally a component order can be used. An empty list gives an ARIMA(0, 0, 0) model, that is white noise.
- **n**: length of output series, before un-differencing. A strictly positive integer.
- The rest of the arguments are optional.

Commands: Simulations of AR(1) Models

> y1 <- arima.sim(list(order=c(1,0,0), ar=.5), n=1000) <== **simulate 1000 data points**

Simulate White Noise

- par(mfcol=c(2,3))
- for (i in 1:6){
- y1 <- arima.sim(list(order = c(0, 0, 0)),n=1000, sd = sqrt(1+20*(i-1)))
- plot(y1)
- }

Commands: Simulations of AR(2) Models

> y1 <- arima.sim(model=list(ar=c(1.3,-.4)),1000) <== **simulate 1000 data points**
> acf(y1) <== compute ACF (Exponential decay)
> y2 <- arima.sim(model=list(ar=c(.8,-.7)),1000) <== **simulate 1000 data points**
> acf(y2) <== ACF shows dampening sine and cosine pattern.
> m1 <- ar(y1,method="mle",order.max=10) <== **compute AIC criterion**
> m1$order
> pacf(y1) <== compute PACF
> t.test(y1) <=== testing mean = 0.
> Box.test(y1,lag=10,type='Ljung') <=== **Perform Ljung-Box Q-test**.

Commands: Finding the Roots

> p1 <- c(1,-1.3,.4)

> r1 <- polyroot(p1)

> r1 <== **See two real roots**

> Mod(r1) <== **Compute absolute values of the roots. They are greater than 1.**

> p2 <- c(1,-.8,.7)

> r2 <- polyroot(p2)

> r2 <== **See two complex roots > Mod(r2)**

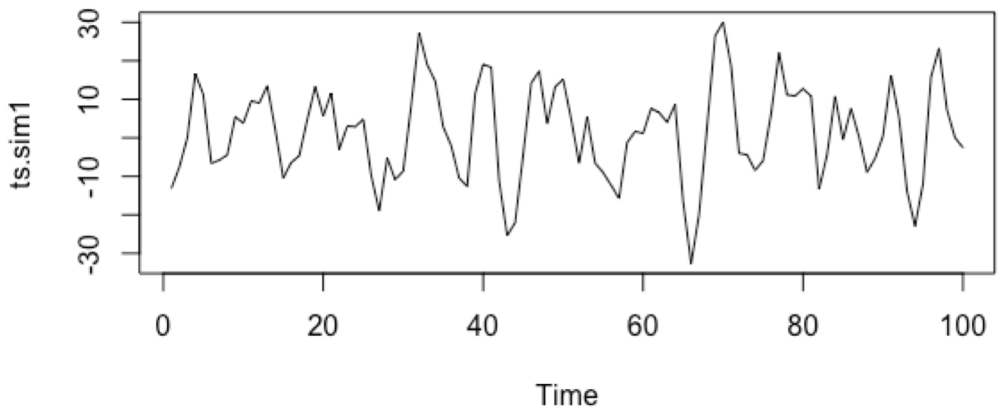
AR(2) Models with different Standard Deviations

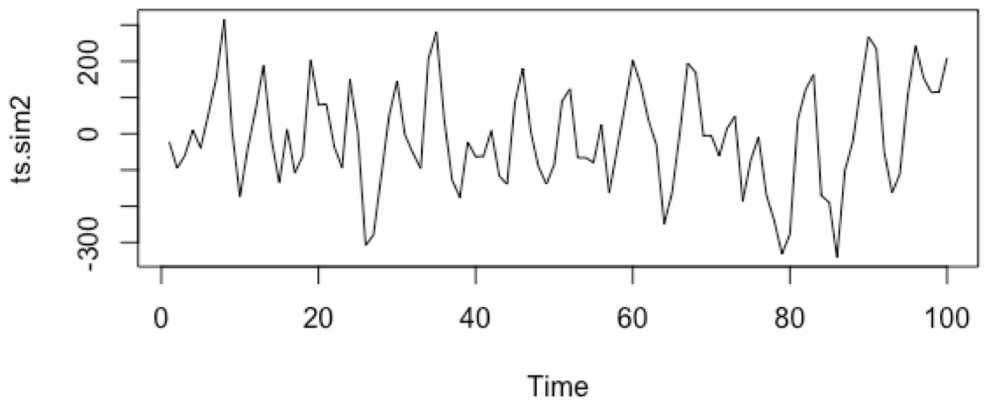
> ts.sim1 <- arima.sim(n = 100, list(ar = c(0.8897, -0.4858)), sd = sqrt(100)) <== **simulate 100 data points**

> ts.sim2 <- arima.sim(n = 100, list(ar = c(0.8897, -0.4858)), sd = sqrt(10000)) <== **simulate 100 data points**

> ts.plot(ts.sim1)

> ts.plot(ts.sim2)





Commands: Computing Partial ACF

> pacf(gnp) <== **compute partial ACF of the GNP growth rate**

Commands: Fitting an AR model to the Data

```
> m1 <- ar(gnp,method='mle')
> names(m1)
> m1$order
```

Commands: Estimation of an ARIMA Model (Fitting)

```
> m2 <- arima(gnp,order=c(3,0,0))
> m2 <== see parameter estimates
> names(m2) <== See the output
> tsdiag(m2) # Model checking
> tsdiag(m2,gof=20) ## increasing the number of residual ACFs used in checking.
> p1 <- c(1,-m2$coef[1:3]) ## Obtain the corresponding polynomial (characteristic equation of the model)
> p1
> roots <- polyroot(p1)
> roots
> Mod(roots)
> k <- 2*pi/acos(1.590253/1.913308)
> k
```

Commands: Estimation of an ARIMA Model (Prediction)

```
> predict(m2,8) # Prediction
> m2p = predict(m2,8)
> names(m2p)
> lcl = m2p$pred-1.96*m2p$se <=== calculate lower 95% interval
> ucl = m2p$pred+1.96*m2p$se <=== calculate upper 95% interval
> cl <- cbind(lcl,ucl)
> print(cl)
```

Commands: Fitting and Prediction of an MA(1) Model

```
> m3 <- arima(rtn$dec9,order=c(0,0,1)) # fit an MA(1) model
> m3
>tsdiag(m3)
> tsdiag(m3,gof=20) ### Chekcing the first 20 lags of residual serial correlations
> predict(m3,5) ### Use model "m3" to produce 1-step to 5-step ahead forecasts
```

Commands: Specifying an ARIMA Model (Alternative Method)

> require(forecast) ### **another R package**

> auto.arima(gnp) ## **Specify an ARIMA model**. (Not recommended, especially for seasonal time series)