Financial Econometrics

R Commands Used in Lecture 6

Prof. Hamed Ghoddusi

2019

Command: Volatility Models

```
require(fGarch)
> da=read.table("m-intc7303.txt",header=T)
> intc=log(da$rtn+1) ### Log returns
> acf(intc)
> Box.test(intc,lag=10,type="Ljung")
> t.test(intc) ## Test mean = 0
> at <- intc - mean(intc)
> Box.test(at^2,lag=10,type="Ljung") ## ARCH test
> m1=garchFit(~garch(1,1),data=intc) # lots of output
> m1=garchFit(~garch(1,1),data=intc,trace=F) # no output printed.
> summary(m1) # Obtain results and model checking statistics
                    ## Should understand the meaning of model checking
```

Command: Volatility Models

> sresi=residuals(m1,standardize=T) **% Obtain standardized residuals - epsilon(t)-hat**

> sigma.t=volatility(m1) **# obtain the fitted volatility sigma_t.**

> plot(m1)

> predict(m1,6) **% predictions of 1-step to 6-step ahead.**

Command: Innovations

- **use standardized Student-t innovations**

```
> m2=garchFit(~garch(1,1),data=intc,trace=F,cond.dist="std")
> summary(m2)
```

- **use skewed Student-t innovations**

```
> m3=garchFit(~garch(1,1),data=intc,trace=F,cond.dist="sstd")
> summary(m3) ### Should understand how to check the skewness.
```

Command: Understanding Skew Standardized Student-t Distribution

```
x <- seq(-5,5,0.05) ## create a sequence of real number
d1 <- dsstd(x,nu=5,xi=1) ### symmetric density plot(x,d1,type="l")
d2 <- dsstd(x,nu=5,xi=1.5) ## skew to right plot(x,d2,type="l")
d3 <- dsstd(x,nu=5,xi=0.5) ## skew to left plot(x,d3,type="l")
r <- rsstd(3000,nu=5,xi=1.5) basicStats(r) ### Skewness is positive
r1 <- rsstd(3000,nu=5,xi=0.5) basicStats(r1) ### Skewness is negative
```

Command: ARMA and GARCH

 # Use ARMA(0,1)+GARCH(1,1) model with normal innovations (in case needs an ARMA model for mean eq.)

```
> m4=garchFit(~arma(0,1)+garch(1,0),data=intc,trace=F)
> summary(m4)
```

Command: IGARCH Model

```
> source("Igarch.R")
> m5=Igarch(intc) ## mu = 0, omega = 0
> names(m5) ### to see that the volatility is stored.
> m6=Igarch(intc,include.mean=T) ### mu is not zero.
> m6a=Igarch(intc,volcnt=T) ## Also estimate omega.
### You use the last fitted volatility (last observation) to compute the forecasts.
```

Command: GARCH-M Model

```
> sp=scan(file='sp500.txt')
> source("garchM.R")
> m7=garchM(sp)
> names(m7) ## To see that the residuals and volatility are stored.
```

Command: EGARCH Model

```
> da=read.table("m-ibmsp6709.txt",header=T)
> ibm=log(da$ibm+1)
> sp=log(da$sp+1)
> source("Tgarch11.R")
> m1=Tgarch11(ibm)
> names(m1)
> resi=m1$residuals/m1$volatility ## For model checking
> Box.test(resi,lag=12,type="Ljung")
> Box.test(resi^2,lag=12,type="Ljung")
> source("Egarch.R")
> m2=Egarch(ibm)
> names(m2)
> plot(m2$volatility,type='l')
```

Command: Fitting GARCH Model to Data

- **The following two models are identical.**

```
> m3=garchFit(~garch(1,1),data=ibm,trace=F,leverage=T)
> summary(m3)


> m4=garchFit(~aparch(1,1),data=ibm,trace=F,delta=2,include.delta=F)
> summary(m4) ### You may use a longer series of IBM lontly returns.
```

Command: rugarch Package

> require(rugarch)

- **The following four specifications are helpful. You can copy them when needed.**
- **You can also modify them if needed once you are familiar with the package.**
- **Specify an standard GARCH(1,1) model**

```
>spec1=ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)))
```

- **Specify an IGARCH(1,1) model**

```
>spec2=ugarchspec(variance.model=list(model="iGARCH"),mean.model=list(armaOrder=c(0,0)))
### Specify an eGARCH(1,1) model
```

- **Specify an eGARCH(1,1) model**

```
>spec3=ugarchspec(variance.model=list(model="eGARCH"),mean.model=list(armaOrder=c(0,0)))
```

- **Specify a GJR-GARCH(1,1) model**

```
>spec4=ugarchspec(variance.model=list(model="gjrGARCH"),mean.model=list(armaOrder=c(0,0)))
```

Command: rugarch Package

- **Estimation**

```
> mm=ugarchfit(spec=spec1,data=ibm)
> mm ### see output
> plot(mm)
```

- **Prediction, 1-step to 5-step ahead**

```
> p1 <- ugarchforecast(mm,n.ahead=5)
> p1
> sigma(p1) ### volatility prediction
fitted(p1) ### mean prediction
```

- **Using standardized Student-t distribution**

```
spec5=ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1)),
mean.model=list(armaOrder=c(0,0)),distribution.model="std")
> m8 <- ugarchfit(data=ibm,spec=spec5)
> m8
```

Command: GARCH with External Regressors

```
> require(quantmod)
> getSymbols("^VIX",from="2007-01-03",to="2016-04-15")
> vix <- as.numeric(VIX[,6])/100 > getSymbols("AAPL",from="2007-01-03",to="2016-04-15")
> rtn <- diff(log(as.numeric(AAPL[,6]))) ### use VIX index the day before
> x1 <- as.matrix(vix[-length(vix)])
> spec6 <- ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1),external.regressors=x1),
mean.model=list(armaOrder=c(0,0)))
> m9 <- ugarchfit(data=rtn,spec=spec6)
> m9 ### Use the vix of the same day (## Not useful for prediction)
> x2 <- as.matrix(vix[-1])
> spec7 <- ugarchspec(variance.model=list(model="sGARCH",garchOrder=c(1,1),external.regressors=x2),
mean.model=list(armaOrder=c(0,0)))
> m10 <- ugarchfit(data=rtn,spec=spec7)
> m10
```

Command: Stochastic Volatility Models

```
### This is easier to use the package "stochvol" below.
> source("svfit.R")
### The script requires "fGarch" and "mvtnorm" packages
> m5=svfit(ibm,200,500)
> names(m5)
> require(stochvol)
> sp <- scan(file="sp500.txt")
> sp <- sp-mean(sp) ### The program assumes zero mean.
> sv1 <- svsample(sp)
> names(sv1)
> apply(sv1$para,2,mean) ## posterior mean of parameters
> apply(sv1$para,2,var) ## posterior variance of parameters
> ht <- apply(sv1$latent,2,median)
> v1 <- exp(ht/2) ### volatility
> ts.plot(v1)
```