Financial Econometrics
Review Session Notes 7

Nina Boyarchenko
February 26, 2010

Contents

1 Simulating GARCH models 2
2 Estimating Vector Autoregressions 4
1 Simulating GARCH models

In the session, we will be working with the daily return series for the S&P 500. Begin by loading the sp500daily.xls file into EViews. Estimate a T-GARCH(1,1,1) model for the data. To do this, use the following code:

```
equation tgarch11.arch(1,1,thrsh=1) return
```

To save the conditional variances, use:

```
tgarch11.makegarch tgarch11var 'save the conditional GARCH variance
series tgarch11vol=@sqrt(tgarch11var) 'compute the corresponding vol
series tgarch11err=return/tgarch11vol 'compute the standardized residuals
'graph errhist.distplot tgarch11err
```

The histogram of the standardized residuals is presented in Fig. 1. Consider now simulating the model for \( k \) periods into the future, using normally distributed errors. To do this, we will follow the procedure described in the homework. Let the last observation in your sample be at time \( T \). The first draw \( (r_{T+1}) \) will be obtained by randomly drawing a value from a Normal(0,1) distribution and multiplying it by \( \sqrt{h_{T+1}} \) (which is the last conditional variance estimated in your model). You now have a simulated draw of \( r_{T+1} \) that can be used to update and get \( h_{T+2} \). Again, take another random draw from a
Normal(0,1) and multiply it by \( \sqrt{h_{T+2}} \) to get a simulated value for \( r_{T+2} \). Continue this out through \( r_{T+30} \) and sum them up to get the cumulative return over the 30-day period. This is one realization (possible outcome) of the 30-day return obtained by simulating the model. We repeat this 2000 times and plot the histogram in Fig. 2. The code for this is below.

```matlab
scalar omega=c(1)
scalar alpha=c(2)
scalar beta=c(3)
scalar gamma=c(4)

scalar T=7329 'last date in the sample
scalar nperiods=30
scalar nsim=2000
matrix(nperiods+1, nsim) ht_norm_for=tgarch11var(T) 'pre-create a matrix for ht
matrix(nperiods, nsim) rt_norm_for 'pre-create a matrix for the simulations of rt
matrix eps=@mnrnd(nperiods, nsim) 'pre-simulate the matrix of random errors
vector(nsim) ret_cum_norm=0 'pre-create the vector of cumulative returns

for !j=1 to nsim
  for !i=1 to nperiods
    rt_norm_for(!i, !j)=eps(!i, !j)*ht_norm_for(!i, !j)
    ht_norm_for(!i+1, !j)=omega+alpha*rt_norm_for(!i, !j)^2+beta*ht_norm_for(!i, !j)+gamma*rt_norm_for(!i, !j)*(rt_norm_for(!i, !j)-@abs(rt_norm_for(!i, !j)))/2
    ret_cum_norm(!j)=ret_cum_norm(!j)+rt_norm_for(!i, !j)
  next
next
```

Once we have the distribution of 30-day returns, we can use the quantile function to compute the cut-off levels for different probabilities. For example, to compute the 1% VaR, we would use

```matlab
scalar VaR_norm=@quantile(ret_cum_norm,0.01)
```

Consider now repeating the above exercise but using the bootstrapped errors. The first draw (\( r_{T+1} \)) will be obtained by randomly selecting a single value of the \( z_t \)s obtained from your fitted model above and multiplying it by \( \sqrt{h_{T+1}} \). You now have a simulated draw of \( r_{T+1} \) that can be used to update and get \( h_{T+2} \). Again, take another random draw from the \( z_t \)s and multiply it by \( \sqrt{h_{T+2}} \) to get a simulated value for \( r_{T+2} \). Continue this out through \( r_{T+30} \) and sum them up to get the cumulative return over the 30-day period. This is one realization (possible outcome) of the 30-day return obtained by simulating the model. We repeat this 2000 times and plot the histogram in Fig. 3. The code for this is below.
2 Estimating Vector Autoregressions

For this part of the session, we will be working the T-bills data, contained in Tbill.xls. For simplicity, rename the series \texttt{yield3m}, \texttt{yield6m}, \texttt{yield1y}, \texttt{yield5y}. To estimate a VAR, use:

```plaintext
matrix u = @round((T-1)*@mrnd(nperiods, nsim)) + 1
matrix(nperiods+1, nsim) ht_boot_for = tgarch11var(T) 'pre-create a matrix for ht
matrix(nperiods, nsim) rt_boot_for 'pre-create a matrix for the simulations of rt
vector(nsim) ret_cum_boot = 0 'pre-create the vector of cumulative returns

for !j=1 to nsim
for !i=1 to nperiods
rt_boot_for(!i, !j) = tgarch11err(u(!i, !j))*ht_boot_for(!i, !j)
ht_norm_for(!i+1, !j) = omega + alpha*rt_boot_for(!i, !j)^2 + beta*ht_boot_for(!i, !j)
+ gamma*rt_boot_for(!i, !j)*(rt_boot_for(!i, !j) - @abs(rt_boot_for(!i, !j)))/2
ret_cum_boot(!j) = ret_cum_boot(!j) + rt_boot_for(!i, !j)
next
next
```
Figure 3: 30-day cumulated return, Empirical errors

This creates a var object in your EViews workfile. Most of the diagnostics concerning the model can be done directly from the object. For example, to examine the impulse response functions for the time series, we select the Impulse tab and select the appropriate options for the output. For the graph in Fig. 4, I selected the Combined Graphs option. Notice that, by default, EViews calculates the non-cumulated impulse response function. To get the cumulated impulse response function, select Accumulated Responses (Fig. 5).
Figure 4: Impulse response functions
Figure 5: Cumulated impulse response functions

Accumulated Response of YIELD3M to Cholesky
One S.D. Innovations

Accumulated Response of YIELD6M to Cholesky
One S.D. Innovations

Accumulated Response of YIELD1Y to Cholesky
One S.D. Innovations

Accumulated Response of YIELD5Y to Cholesky
One S.D. Innovations