Chapter 6

More on Multivariate GARCH

We'll now discuss adjustments to the basic multivariate GARCH models and other types of analysis which can be done with results from the **GARCH** instruction.

We'll look at two new data sets. For Sections 6.1 and 6.2 we will use (a reconstruction of) the data set from Hafner and Herwartz (2006) (from now on HH). The full data set (called HHDATA.XLS) has 3270 daily observations on ten exchange rates vs the dollar, running from 31 December 1979 to 1 April 1994. The data as used in the paper are expressed in local currency/USD, while the data on the file are USD/local currency, so we have to change the sign when computing the returns.[1] There's a separate date column, with the date coded numerically as a six digit number *yymmdd*. Although the data set includes all five days a week and so could be handled as **CALENDAR(D)** with RATS, we'll treat it as irregular, and show how to locate an entry based upon a coded date field like that. The data are read with

```
open data hhdata.xls
data(format=xls,org=columns) 1 3720 usxjpn usxfra usxsui $
 usxnld usxuk usxbel usxger usxswe usxcan usxita date
```

Our focus will be on bivariate models on returns for two of the currencies: the British pound and the Deutsche mark. The authors choose to use separate univariate autoregressions for the mean models—if you estimate a one lag VAR, the "other" lags have t-stats less than 1, so leaving them out isn't unreasonable. The mean model can be set up with

```
set demret = -100.0*log(usxger/usxger{1})
set gbpret = -100.0*log(usxuk/usxuk{1})
*
equation demeqn demret
# constant demret{1}
equation gbpeqn gbpret
# constant gbpret{1}
group uniar1 demeqn gbpeqn
```

The **GROUP** instruction combines the separate equations into a single model for input into **GARCH**. This is the only way to create a mean model with different

[1] Though this has no effect on any variance calculations.

Table 6.1: BEKK Estimates for Hafner-Herwartz Data

MV-GARCH, BEKK - Estimation by BFGS
Convergence in 54 Iterations. Final criterion was 0.0000075 <= 0.0000100
Usable Observations 3718
Log Likelihood -5259.4997

	Variable	Coeff	Std Error	T-Stat	Signif
1.	Constant	0.0086	0.0089	0.9677	0.3332
2.	DEMRET{1}	0.0035	0.0123	0.2853	0.7754
3.	Constant	-0.0030	0.0085	-0.3546	0.7229
4.	GBPRET{1}	0.0184	0.0125	1.4662	0.1426
5.	C(1,1)	0.0963	0.0122	7.9131	0.0000
6.	C(2,1)	0.0708	0.0138	5.1138	0.0000
7.	C(2,2)	-0.0456	0.0039	-11.7169	0.0000
8.	A(1,1)	0.2891	0.0235	12.2816	0.0000
9.	A(1,2)	-0.0072	0.0225	-0.3205	0.7486
10.	A(2,1)	-0.0518	0.0239	-2.1674	0.0302
11.	A(2,2)	0.2459	0.0209	11.7592	0.0000
12.	B(1,1)	0.9565	0.0071	134.6023	0.0000
13.	B(1,2)	0.0046	0.0065	0.7170	0.4734
14.	B(2,1)	0.0100	0.0075	1.3378	0.1810
15.	B(2,2)	0.9636	0.0055	175.5495	0.0000
16.	Shape	4.3861	0.2451	17.8967	0.0000

regressors; a full VAR *can* be done using the REGRESSORS option, as that puts the same set of regressors into each equation, though we would recommend the more convenient **SYSTEM** definition for the model.

They choose a BEKK model with t errors:

```
garch(model=uniar1,mv=bekk,rvectors=rd,hmatrices=hh,distrib=t,$
   pmethod=simplex,piters=20,iters=500)
```

which produces Table 6.1. If we compare this with the BEKK estimates in Table 5.4, off-diagonal elements for A and B are, in this data set, close to zero. A further restricted version of the BEKK called a diagonal BEKK or DBEKK appears to be appropriate. A DBEKK is also a restricted version of the DVECH where the coefficients on A and B in the covariance recursions are the geometric means of the coefficients on their corresponding variance recursions. The option MV=DBEKK was added with RATS version 8.2 to estimate this model. Since it won't make much difference, we'll use the unresticted BEKK in the examples in Sections 6.1 and 6.2.

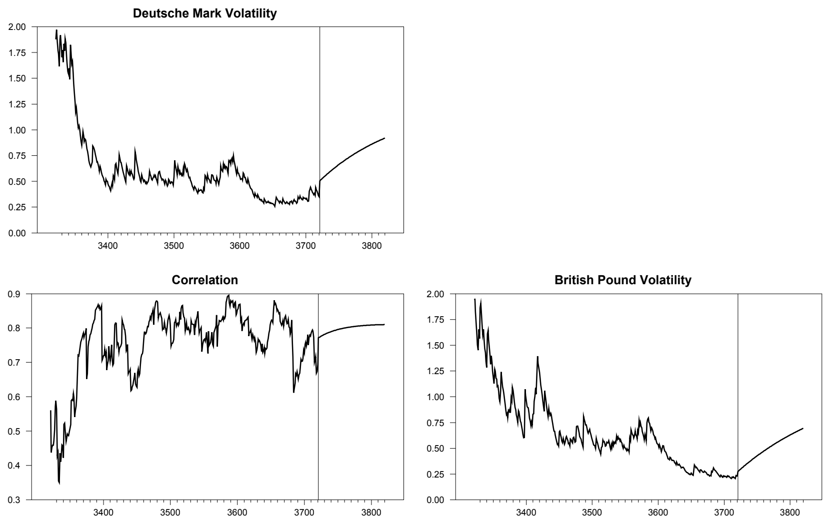

Figure 6.1: Forecasts from a BEKK GARCH Model

6.1 Forecasting

For any model that can be cast into VECH form, the out-of-sample forecasts can be done using the recursion (5.2). As with univariate forecasts, this will require input of the estimates for the residuals and covariance matrices. The calculations can be done most easily using the procedure **@MVGARCHFore**. In Example **6.1**, we'll use the BEKK model estimated above. To forecast out-of-sample for 100 steps, we use

```
@MVGARCHFore(mv=bekk,steps=100) hh rd
```

The HH and RD are input to the procedure, and HH is also used for the output. The following graphs the last 400 observed values and the 100 forecast steps for the two volatilities and the correlations, producing Figure 6.1:

```
spgraph(vfields=2,hfields=2)
compute gstart=%regend()-399,gend=%regend()+100
set h11fore gstart gend = hh(t)(1,1)
set h22fore gstart gend = hh(t)(2,2)
set h12fore gstart gend = hh(t)(1,2)/sqrt(h11fore*h22fore)
graph(row=1,col=1,grid=(t==%regend()+1),min=0.0,$
   header="Deutsche Mark Volatility")
# h11fore
graph(row=2,col=2,grid=(t==%regend()+1),min=0.0,$
   header="British Pound Volatility")
# h22fore
graph(row=2,col=1,grid=(t==%regend()+1),header="Correlation")
# h12fore
spgraph(done)
```

As you can see, there's a quick move on the D-Mark, which results from a fairly large residual in its last data point. If you graphed just the forecasts without the context of the actual data, the forecasts would appear to be explosive. However, the model has eigenvalues just below 1, and is simply (very slowly) converging back from the historically low volatility at the end of the sample towards the average.

6.2 Volatility Impulse Response Functions

Hafner and Herwartz (2006) introduced the concept of the *volatility impulse response function* (VIRF) for multivariate GARCH models. The recursion (5.1) is very similar to the one governing a one-lag VAR in the mean, and we generally describe the dynamics of a VAR in terms of its impulse responses or some information derived from them.

The calculation of the VIRF is not difficult for models which can be put into the VECH form, once we define what an "impulse" means in this context. For the IRF for a VAR, a shock is any non-zero $\mathbf{u}_t$. Because of linearity, we can do responses to a standardized set of shocks (for instance, unit shocks to each variable in turn) and add or rescale them to get responses to any other set of shocks. IRF's are usually presented as an $n \times n$ set of graphs, one for each combination of shock and target variable. For a GARCH model, that no longer will work, since the $\mathbf{u}_t$ enters as a "square" (actually as the outer product). Now a unit shock may be completely out-of-scale, and we can't simply aggregate them anyway, since the inputs get squared before being used. Instead, VIRF's need to be calculated as the responses to a complete vector of shocks.

HH describe several ways to create interesting sets of shocks to input to the recursion. To make sense, these must somehow be "typical" of the data. We pick either $\mathbf{u}_t$ and transform to $\mathbf{u}_t\mathbf{u}_t'$ or pick $\mathbf{u}_t\mathbf{u}_t'$ directly. The difference in the volatility forecasts (5.2) with the input shocks (compared to $\mathbf{u}_t = 0$) is then:

$$\begin{aligned}
vech(\mathbf{v}_{t+1}) &= \mathbf{A}vech(\mathbf{u}_t\mathbf{u}_t') \\
vech(\mathbf{v}_{t+k}) &= (\mathbf{A}+\mathbf{B})vech(\mathbf{v}_{t+k-1})
\end{aligned}$$

This defines what the authors call the *conditional volatility profile*. This is a function just of the model coefficients and the shock, and not the data. The VIRF itself is the same basic formula with a slightly different input. In the standard IRF, we're computing the revision in the forecast due to observing the given shock. For the analogous idea in the volatility equation, we need to do the calculations as:

$$\begin{aligned}
vech(\mathbf{V}_{t+1}) &= \mathbf{A}vech(\mathbf{u}_t{\mathbf{u}_t}' - \mathbf{H}_t) \\
vech(\mathbf{V}_{t+k}) &= (\mathbf{A}+\mathbf{B})vech(\mathbf{V}_{t+k-1})
\end{aligned} \tag{6.1}$$

where $\mathbf{H}_t$ is the GARCH covariance matrix for time t. The VIRF depends upon the data now through $\mathbf{H}_t$—the "shock" to the variance is the amount by which the $\mathbf{u}_t\mathbf{u}_t'$ exceeds its expected value. Of course, it's possible for that to be negative, even on the diagonals, while the input to the conditional volatility profile has to have positive diagonals.

HH computed VIRF for two historical episodes, and we'll show how to handle those. Note, however, that the data set is a reproduction, so the results don't quite match. We are also scaling the data by 100 relative to what they do, which eliminates the need to rescale the responses themselves.

The model in Example **6.2** is the same as the one used in Section 6.1—a BEKK with t errors. To compute the VIRF, we need to convert the BEKK estimates to VECH. This also looks at the eigenvalues of the persistence matrix:

```
@MVGARCHtoVECH(mv=bekk)
eigen(cvalues=cv) %%vech_a+%%vech_b
disp "Eigenvalues from BEKK-t" *.### cv
```

The eigenvalues are just barely inside the stable region:

```
Eigenvalues from BEKK-t (0.994,-0.000) (0.993,0.004) (0.993,-0.004)
```

The shocks used in the two experiments are from "Black Wednesday" (September 16, 1992), when the Italian lira and the UK pound dropped out of the European Exchange Rate Mechanism (ERM), and August 2, 1993, when the EC finance ministers changed the variability bands on currencies in the ERM. The simplest way to locate those entries given the DATE field on the file is with:

```
sstats(max) / %if(date==920916,t,0)>>xblackwed $
              %if(date==930802,t,0)>>xecpolicy
compute blackwed=fix(xblackwed),ecpolicy=fix(xecpolicy)
```

Now, if we had a daily **CALENDAR** date scheme, we could simply use

```
compute blackwed=92:9:16
compute ecpolicy=93:8:2
```

for the entries, but we didn't set that up with this data set, so we're showing how to locate an entry given the coded date field. XBLACKWED and XECPOLICY are both REAL variables (which is what **SSTATS** returns), so the **COMPUTE** instruction is used to convert them to INTEGER entry numbers.

Example **6.2** wraps the calculations in an **SPGRAPH** which does two columns, one for each shock, with three fields in each, one for the each variance and one for the covariance. In practice, that would be the *last* thing you did—you would want to get the calculations right first before being concerned with the appearance of the output. So we'll skip that for later and jump right to the calculation.

The description of the calculation for the VIRF in the paper is more complicated than it needs to be—they transform the observed residuals to a standardized vector using (in our notation) $\mathbf{H}_t$, but the standardization gets reversed in putting in back into the recursion. Instead, you can just compute the final matrix in the first row of (6.1) using

```
compute eps0=rd(blackwed)
compute sigma0=hh(blackwed)
compute shock=%vec(%outerxx(eps0)-sigma0)
```

Because the VECH recursion is in *vech* form, we need to convert the shock from a matrix to a vector. %VEC does the VECH stacking as long as its argument is identifiable as a symmetric array. That's the case here because SIGMA0, as an element of HH, is SYMMETRIC and %OUTERXX creates a SYMMETRIC by construction. If you want to be careful, you can write %VEC([SYMMETRIC]...) to force the desired interpretation.

The calculation will generate 3 (that is, $n(n+1)/2$) output series over the requested number of steps (400 in this example, which has already been saved into the variable NSTEP). This creates a target set of SERIES for this:

```
dec vect[series] sept92virf(3)
do i=1,3
   set sept92virf(i) 1 nstep = 0.0
end do i
```

The actual calculation is quite simple. HVEC is an $n(n+1)/2$ vector which, at each point, has the previous period's *vech*'ed response. This gets overwritten by the recalculated value. At each step, this gets split up into the 3 SEPT92VIRF series using %PT.

```
do step=1,nstep
   if step==1
      compute hvec=%%vech_a*shock
   else
      compute hvec=(%%vech_a+%%vech_b)*hvec
   compute %pt(sept92virf,step,hvec)
end do step
```

The first column of graphs is done with the following, which has a few options added to improve the appearance:

```
do i=1,3
   graph(column=1,row=i,picture="*.###",vticks=5)
   # sept92virf(i) 1 nstep
end do i
```

Note that because of the way the *vech* operator works, the second graph is the covariance between the two currencies.

The calculation of the VIRF for the EC policy shock is similar:

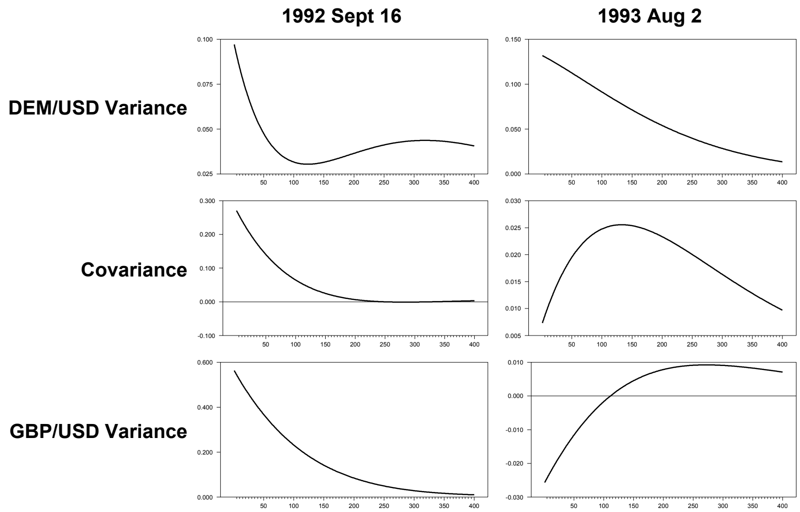

Figure 6.2: Volatility Impulse Response Functions

```
compute eps0=rd(ecpolicy)
compute sigma0=hh(ecpolicy)
compute shock=%vec(%outerxx(eps0)-sigma0)
```

Given that, the rest of the calculation is the same as above, with results put into a different VECTOR[SERIES]:

```
dec vect[series] aug93virf(3)
do i=1,3
   set aug93virf(i) 1 nstep = 0.0
end do i
*
do step=1,nstep
   if step==1
      compute hvec=%%vech_a*shock
   else
      compute hvec=(%%vech_a+%%vech_b)*hvec
   compute %pt(aug93virf,step,hvec)
end do step
do i=1,3
   graph(column=2,row=i,picture="*.###",vticks=5)
   # aug93virf(i) 1 nstep
end do i
```

The **SPGRAPH** that encloses the graphs puts labels on the rows and columns on the margins of the graph array. The result is Figure 6.2.

```
spgraph(vfields=3,hfields=2,footer="Figure 1",$
  xlabels=||"1992 Sept 16","1993 Aug 2"||,$
  ylabels=||"DEM/USD Variance","Covariance","GBP/USD Variance"||)
```

Ordinarily, with standard impulse response functions, we recommend graphing all responses *of* a variable over a common range, as that makes it easier to compare the importance of the different shocks in explaining a variable. HH recommend scaling the responses by the historical variances at the time of the shock in question.[2] In this case, it would make almost no difference, since the variances (in our scaling of the data) are very near 1 in both situations.[3] And that would not change the fact that the first incident disproportionately hit the pound, while the second had almost no effect on it. It probably still makes sense to use a common scale across a row, but be aware of the fact that, unlike the IRF for a VAR, the size of the responses isn't just a function of the model, but depends upon your choice of historical shocks to apply.

6.3 Asymmetry

Asymmetry is more complicated with multivariate models than univariate. With the univariate model, there are many equivalent ways to parameterize asymmetry. For instance,

$$\begin{aligned} au_{t-1}^2 + du_{t-1}^2 I(u_{t-1} < 0) &= (a+d)u_{t-1}^2 - du_{t-1}^2 I(u_{t-1} > 0) \\ &= au_{t-1}^2 I(u_{t-1} > 0) + (a+d)\, u_{t-1}^2 I(u_{t-1} < 0) \end{aligned}$$

so it doesn't matter whether the asymmetry term uses the positive or negative sign. The first formulation is used most often because the assumption is that negative shocks increase variance more than positive ones, so that would give $d > 0$. But if that assumption is incorrect—if it's positive shocks which increase the variance—there is nothing preventing d from being negative.

In a multivariate setting, it's no longer true that the sign convention is innocuous. The problem comes from the interaction terms between the shocks. The most commonly used adjustment for asymmetry in the standard multivariate GARCH is to define $\mathbf{v}_t$ as

$$\mathbf{v}_t = \mathbf{u}_t \circ I(\mathbf{u}_t < 0)$$

that is $v_{it} = u_{it}$ if $u_{it} < 0$ and $v_{it} = 0$ otherwise, done component by component. The recursion for $\mathbf{H}$ is

$$\mathbf{H}_t = \mathbf{C} + \mathbf{A} \circ \left(\mathbf{u}_{t-1}\mathbf{u}_{t-1}'\right) + \mathbf{B} \circ \mathbf{H}_{t-1} + \mathbf{D} \circ \left(\mathbf{v}_{t-1}\mathbf{v}_{t-1}'\right) \tag{6.2}$$

The diagonal terms are identical to those in a corresponding asymmetric univariate GARCH. However, for off-diagonal element ij, the asymmetry term is non-zero only if *both* $u_{i,t-1}$ and $u_{j,t-1}$ are negative. The recursion differentiates between those data points with both negative (where $\mathbf{D}$ comes into play) and those where at least one is positive, which only get the first three terms. If

[2] Presumably dividing the covariance by the square root of the product of the variances, though that's not explicit in the paper.

[3] With their scaling, these would all be smaller by a factor of 10000.

we change the sign convention in the definition of v, the data points which get the added term are those where *both* u are positive. Both of these leave out those data points with one positive and one negative. Unless you add an additional term or terms to deal with those, the likelihood function will be different depending upon which sign convention you choose for the asymmetry.

For univariate models with the basic GARCH recursion (not EGARCH), forecasting can be done relatively easily with the asymmetry, because

$$E\left(u_{t-1}^2 I(u_{t-1} > 0)\right) = .5Eu_{t-1}^2$$

which is true for any symmetric density for u. And that's true on the *diagonals* for (6.2), but *not* the off-diagonals. The expected value of $v_{it}v_{jt}$ for $i \neq j$ is a complicated function of the covariance matrix. For a Normal distribution, this is:

$$Ev_i v_j = h_{ij}F(0,0,\rho) + \frac{\sqrt{1-\rho^2}}{2\pi}\sqrt{h_{ii}h_{jj}} \tag{6.3}$$

where $F(x,y,r)$ is the standard bivariate normal CDF with correlation r.[4] (6.3) is a special case of the general formula in Muthen (1990) with ρ as the correlation between u_i and u_j. (6.3) only gives $.5h_{ij}$ if $\rho = 1$. We've seen several papers that did calculations with asymmetric models assuming that the .5 factor could apply to the full matrix; this is incorrect. Not only is it not a simple function, but it also depends upon the distribution: (6.3) is specific to the Normal. For the same reason, there is no simple extension of the VIRF of Section 6.2 to an asymmetric model—even if you want to incorporate the calculation in (6.3), it can't be computed without H itself at every stage, so there is no way to compute it as an impulse response.

As with the univariate model, you add asymmetry to the GARCH model by adding the `ASYMMETRIC` option to **GARCH**. What parameters that adds to the model will depend upon the choice for the `MV` option.

The data file used in Example **6.3** is daily return data on the US SP500, the Japanese Nikkei and the Hong Kong Hang Seng, daily (holidays included) from 1 January 1986 to 10 April 2000. The original data are just raw daily returns, so we scale up by 100:

[4] The standard bivariate normal has variance 1 for each component and correlation r. The CDF is the probability of the quadrant southwest of (x,y). This can't be calculated using the univariate CDF unless $r = 0$; instead, you can use the RATS function `%BICDF`.

```
open data mhcdata.xls
cal(d) 1986:1:1
data(format=xls,org=columns) 1 3724 rnk rhs rsp
*
* We scale up the original data by 100.0
*
set rsp = rsp*100.0
set rnk = rnk*100.0
set rhs = rhs*100.0
```

This is taken from the working paper “Empirical Modelling of Multivariate Asymmetric Financial Volatility” by Chan and McAleer. Asymmetry is a common feature in GARCH models of stock market return data, presumably due to the leverage effect. As with the example above, the authors chose univariate AR1 models for the mean:

```
equation speq rsp
# constant rsp{1}
equation nkeq rnk
# constant rnk{1}
equation hseq rhs
# constant rhs{1}
*
group armeans speq nkeq hseq
```

The simplest form of asymmetry is in the basic CC model—this just adds the one univariate asymmetry parameter to each variance equation. As we will be looking at several non-nested models, we'll compute a set of information criteria on each model:

```
garch(model=armeans,mv=cc,asymm)
@regcrits(title="CC Model with Asymmetry")
```

For the simple CC model, the asymmetry coefficients are labeled as `D(1)`, `D(2)`, `D(3)` for the three equations. The output is Table 6.2. As you can see, the asymmetry terms are *highly* significant, and are larger (even when multiplied by .5) than the corresponding A coefficients.

The final model in the working paper was the CC model with asymmetric VARMA GARCH variances. The added asymmetry parameters are one per variable, just like the basic CC. The VARMA GARCH is an extension of the spillover model (page 98) which includes B coefficients on all the lagged variances, not just own variances.[5] It is estimated with `MV=CC` and `VARIANCES=VARMA`.

[5] Unfortunately, the phrase VARMA-GARCH is used to mean both a GARCH with the VARMA recursion for the variances, and a standard GARCH model with a VARMA model for the mean.

Table 6.2: CC Model with Asymmetry

MV-GARCH, CC - Estimation by BFGS
Convergence in 51 Iterations. Final criterion was 0.0000053 <= 0.0000100
Daily(5) Data From 1986:01:03 To 2000:04:10
Usable Observations 3722
Log Likelihood -16751.3755

	Variable	Coeff	Std Error	T-Stat	Signif
1.	Constant	0.0565	0.0136	4.1561	0.0000
2.	RSP{1}	0.0301	0.0179	1.6866	0.0917
3.	Constant	0.0502	0.0172	2.9117	0.0036
4.	RNK{1}	-0.0031	0.0178	-0.1761	0.8602
5.	Constant	0.0902	0.0210	4.3053	0.0000
6.	RHS{1}	0.1006	0.0187	5.3759	0.0000
7.	C(1)	0.0198	0.0042	4.7367	0.0000
8.	C(2)	0.0297	0.0051	5.7849	0.0000
9.	C(3)	0.0892	0.0107	8.3576	0.0000
10.	A(1)	0.0276	0.0090	3.0629	0.0022
11.	A(2)	0.0377	0.0079	4.7964	0.0000
12.	A(3)	0.0528	0.0094	5.6350	0.0000
13.	B(1)	0.9087	0.0122	74.7463	0.0000
14.	B(2)	0.8702	0.0095	91.4406	0.0000
15.	B(3)	0.8369	0.0103	80.8687	0.0000
16.	D(1)	0.0847	0.0138	6.1593	0.0000
17.	D(2)	0.1750	0.0165	10.6008	0.0000
18.	D(3)	0.1738	0.0207	8.3871	0.0000
19.	R(2,1)	0.2693	0.0136	19.7330	0.0000
20.	R(3,1)	0.3148	0.0144	21.8621	0.0000
21.	R(3,2)	0.2540	0.0156	16.3017	0.0000

```
garch(model=armeans,mv=cc,variances=varma,asymm,$
  pmethod=simplex,piters=10,method=bfgs)
@regcrits(title="CC-VARMA Model with Asymmetry")
```

This adds an extra 12 parameters compared to the basic CC model. We'll omit the output—it's does somewhat worse than the asymmetric CC on the BIC and somewhat better on less stringent criteria like the AIC.

With asymmetry, the BEKK model becomes

$$\mathbf{H}_t = \mathbf{C}\mathbf{C}' + \mathbf{A}'\mathbf{u}_{t-1}\mathbf{u}'_{t-1}\mathbf{A} + \mathbf{B}'\mathbf{H}_{t-1}\mathbf{B} + \mathbf{D}'\mathbf{v}_{t-1}\mathbf{v}'_{t-1}\mathbf{D}$$

which adds the $n \times n$ matrix $\mathbf{D}$. More than in the other models, this is sensitive to the choice of sign for the asymmetric effect since that final term is forceably positive semi-definite—the variance increment has to be at least as high for the data points covered by the final term as it is for those that aren't. The default is for $\mathbf{v}$ to be constructed using the negative branch. RATS 8.2 adds the `SIGNS` option, which allows you to use a different sign convention.. `SIGNS` provides a n vector with the desired signs for the asymmetry for each variable. The default is all -1, but you can change that to all 1's for the positive branch, or can mix-and-match if you have a combination of variables for which that's appropriate. For instance, SIGNS=||-1,1|| in a bivariate system would have negative asymmetry on the first variable and positive on the second.

```
garch(model=armeans,mv=bekk,asymm,$
  pmethod=simplex,piters=10,method=bfgs)
@regcrits(title="BEKK Model with Asymmetry")
```

Asymmetry in the standard GARCH uses the formula (6.2). D is symmetric, so it adds $n(n+1)/2$ new parameters. This ends up being the (slightly) preferred specification of the four, at least by the BIC and HQ. BEKK is very slightly favored by AIC. The output from **GARCH** is in Table 6.3.

```
garch(model=armeans,asymm,rvectors=rd,hmatrices=hh,$
  pmethod=simplex,piters=10,method=bfgs)
@regcrits(title="Standard GARCH with Asymmetry")
```

We did standard multivariate diagnostics for this model:

```
dec vect[series] rstd(%nvar)
do time=%regstart(),%regend()
   eigen(scale) hh(time) * eigfac
   compute %pt(rstd,time,%solve(eigfac,rd(time)))
end do time
*
@mvqstat(lags=10)
# rstd
@mvarchtest
# rstd
```

producing

```
Multivariate Q(10)=     110.19303
Significance Level as Chi-Squared(90)=      0.07295

Test for Multivariate ARCH
Statistic Degrees Signif
   850.40      36 0.00000
```

The test for residual ARCH is clearly much worse than we would like. One thing to note, however, is that none of the estimates used t errors. We can do a multivariate Jarque-Bera test on the standardized residuals using the **@MVJB** procedure with

```
@mvjb(factor=%identity(3)) rstd
```

```
Var    JB    P-Value
  1 1861.392   0.000
  2 5726.954   0.000
  3 1265.721   0.000
All 8854.067   0.000
```

which shows that each component and the joint test are highly significant. If we wanted to spend more time with this model, we would go back and re-

Table 6.3: Standard GARCH with Asymmetry

MV-GARCH - Estimation by BFGS
Convergence in 111 Iterations. Final criterion was 0.0000059 <= 0.0000100
Daily(5) Data From 1986:01:03 To 2000:04:10
Usable Observations 3722
Log Likelihood -16686.6651

	Variable	Coeff	Std Error	T-Stat	Signif
1.	Constant	0.0597	0.0135	4.4183	0.0000
2.	RSP{1}	0.0316	0.0174	1.8130	0.0698
3.	Constant	0.0575	0.0165	3.4867	0.0005
4.	RNK{1}	-0.0022	0.0165	-0.1347	0.8928
5.	Constant	0.1006	0.0187	5.3884	0.0000
6.	RHS{1}	0.0948	0.0159	5.9826	0.0000
7.	C(1,1)	0.0140	0.0031	4.4976	0.0000
8.	C(2,1)	0.0033	0.0017	1.9017	0.0572
9.	C(2,2)	0.0267	0.0046	5.7624	0.0000
10.	C(3,1)	0.0093	0.0022	4.2025	0.0000
11.	C(3,2)	0.0086	0.0030	2.8602	0.0042
12.	C(3,3)	0.0785	0.0085	9.2891	0.0000
13.	A(1,1)	0.0272	0.0066	4.1069	0.0000
14.	A(2,1)	0.0047	0.0070	0.6664	0.5051
15.	A(2,2)	0.0330	0.0075	4.3865	0.0000
16.	A(3,1)	0.0116	0.0063	1.8307	0.0671
17.	A(3,2)	0.0142	0.0078	1.8051	0.0711
18.	A(3,3)	0.0517	0.0075	6.8830	0.0000
19.	B(1,1)	0.9231	0.0087	106.1192	0.0000
20.	B(2,1)	0.9144	0.0091	99.9792	0.0000
21.	B(2,2)	0.8761	0.0092	95.4095	0.0000
22.	B(3,1)	0.9269	0.0095	97.6067	0.0000
23.	B(3,2)	0.9033	0.0098	92.4051	0.0000
24.	B(3,3)	0.8484	0.0088	96.8754	0.0000
25.	D(1,1)	0.0709	0.0109	6.4983	0.0000
26.	D(2,1)	0.0917	0.0117	7.8123	0.0000
27.	D(2,2)	0.1760	0.0152	11.6115	0.0000
28.	D(3,1)	0.0566	0.0122	4.6524	0.0000
29.	D(3,2)	0.0823	0.0145	5.6790	0.0000
30.	D(3,3)	0.1604	0.0168	9.5343	0.0000

estimate with t errors. It's also the case that the ARCH test fails largely because of the period around the October 1987 crash in the US which create huge outliers in the standardized residuals.[6]

6.4 GARCH-X

The XREGRESSORS option can be added to multivariate GARCH models as well as univariate. Typically, these are dummies of some form. XREGRESSORS adjusts the $\mathbf{C}$ term in the GARCH recursion. For any form of CC, this adds a coefficient for each variance equation for each of the "X" variables. For a standard GARCH, it adds a coefficient for each regressor to each of the components. In either case, it's relatively straightforward.

The only model type for which it's not clear how to handle such regressors is the BEKK because of the desire to enforce positive-definiteness. If we restrict our attention to dummy variables, it's a standard result that it's irrelevant to the fit whether a dummy is 0-1 or 1-0 for an event and its complement. For instance, if we're working with scalars:

$$\alpha + \beta d_t = (\alpha + \beta) - \beta(1 - d_t) = \alpha(1 - d_t) + (\alpha + \beta)\, d_t$$

If there are no sign restrictions, all three of these are equivalent. The problem with the BEKK is that there *are* sign restrictions, since each term is required to be positive semi-definite. If we apply the dummy to a factor matrix and try to alter the interpretation of the dummy, we get

$$\mathbf{CC}' + \mathbf{EE}' d_t = \mathbf{CC}' + \mathbf{EE}' + \left(-\mathbf{EE}'\right)(1 - d_t)$$

If the left side is OK, the right side isn't a permitted parameterization.

An alternative (which is what the **GARCH** instruction does) is to replace $\mathbf{CC}'$ with

$$\left(\mathbf{C} + \mathbf{E}d_t\right)\left(\mathbf{C} + \mathbf{E}d_t\right)'$$

where $\mathbf{E}$ is (like $\mathbf{C}$) a lower triangular matrix. This enforces positive-definiteness, but doesn't require that the dummy add a positive semi-definite increment to the variance. Because the adjustments are made before "squaring", the model isn't sensitive to the choice of representation for the dummy.

As an illustration of XREGRESSORS, we will add a "Monday" dummy to the variance model for the three stock market returns. With a daily **CALENDAR**, it's easy to create that with

```
set monday = %weekday(t)==1
```

%WEEKDAY maps the entry number to 1 to 7 for Monday to Sunday. We'll add this to the standard and BEKK models, while also adding DISTRIB=T to deal with the fat tails:

[6] You can add range parameters like `1988:1 *` to the **@mvarchtest** to restrict the sample.

```
garch(model=armeans,asymm,xreg,distrib=t,$
  pmethod=simplex,piters=10,method=bfgs)
# monday
garch(model=armeans,mv=bekk,asymm,xreg,distrib=t,$
   pmethod=simplex,piters=10,method=bfgs)
# monday
```

The Monday dummies are generally insignificant (and have what would generally be seen as the wrong sign). The one place they come in significant is with the covariance between the SP500 and HS.

Example 6.1 Multivariate GARCH: Forecasting

This is the example from Section 6.1.

```
open data hhdata.xls
data(format=xls,org=columns) 1 3720 usxjpn usxfra usxsui $
 usxnld usxuk usxbel usxger usxswe usxcan usxita date
*
* This is rescaled to 100.0 compared with data used in the paper. The
* paper also uses local currency/USD, while the data set has USD/local
* currency, so we change the sign on the return.
*
set demret = -100.0*log(usxger/usxger{1})
set gbpret = -100.0*log(usxuk/usxuk{1})
*
* The mean model is a univariate AR on each variable separately.
*
equation demeqn demret
# constant demret{1}
equation gbpeqn gbpret
# constant gbpret{1}
group uniar1 demeqn gbpeqn
*
* Estimate BEKK model with student t errors
*
garch(model=uniar1,mv=bekk,rvectors=rd,hmatrices=hh,distrib=t,$
  pmethod=simplex,piters=20,iters=500)
*
@MVGARCHFore(mv=bekk,steps=100) hh rd
*
spgraph(vfields=2,hfields=2)
compute gstart=%regend()-399,gend=%regend()+100
set h11fore gstart gend = hh(t)(1,1)
set h22fore gstart gend = hh(t)(2,2)
set h12fore gstart gend = hh(t)(1,2)/sqrt(h11fore*h22fore)
graph(row=1,col=1,grid=(t==%regend()+1),min=0.0,$
   header="Deutsche Mark Volatility")
# h11fore
graph(row=2,col=2,grid=(t==%regend()+1),min=0.0,$
   header="British Pound Volatility")
# h22fore
graph(row=2,col=1,grid=(t==%regend()+1),header="Correlation")
# h12fore
spgraph(done)
```

Example 6.2 Volatility Impulse Responses

This is the example from Section 6.2.

```
open data hhdata.xls
data(format=xls,org=columns) 1 3720 usxjpn usxfra usxsui $
 usxnld usxuk usxbel usxger usxswe usxcan usxita date
*
* This is rescaled to 100.0 compared with data used in the paper. The
* paper also uses local currency/USD, while the data set has USD/local
* currency, so we change the sign on the return.
*
set demret = -100.0*log(usxger/usxger{1})
set gbpret = -100.0*log(usxuk/usxuk{1})
*
* The mean model is a univariate AR on each variable separately.
*
equation demeqn demret
# constant demret{1}
equation gbpeqn gbpret
# constant gbpret{1}
group uniar1 demeqn gbpeqn
*
garch(model=uniar1,mv=bekk,rvectors=rd,hmatrices=hh,distrib=t,$
  pmethod=simplex,piters=20,iters=500)
*
* Transform the BEKK model to its equivalent VECH representation
*
@MVGARCHtoVECH(mv=bekk)
eigen(cvalues=cv) %%vech_a+%%vech_b
disp "Eigenvalues from BEKK-t" *.### cv
*
* VIRF with historical incidents
*
sstats(max) / %if(date==920916,t,0)>>xblackwed $
              %if(date==930802,t,0)>>xecpolicy
compute blackwed=fix(xblackwed),ecpolicy=fix(xecpolicy)
*
compute nstep=400
spgraph(vfields=3,hfields=2,footer="Figure 1",$
  xlabels=||"1992 Sept 16","1993 Aug 2"||,$
  ylabels=||"DEM/USD Variance","Covariance","GBP/USD Variance"||)
*
* Black Wednesday shocks. These are computed using a baseline of the
* estimated volatility state, so they are excess over the predicted
* covariance.
*
compute eps0=rd(blackwed)
compute sigma0=hh(blackwed)
compute shock=%vec(%outerxx(eps0)-sigma0)
*
* This generates responses for the 3 elements of the covariance matrix,
* which will be (in order) the (1,1),(1,2) and (2,2).
```

```
*
dec vect[series] sept92virf(3)
do i=1,3
   set sept92virf(i) 1 nstep = 0.0
end do i
*
* Use the VECH representation to compute the VIRF to the original shock.
*
do step=1,nstep
   if step==1
      compute hvec=%%vech_a*shock
   else
      compute hvec=(%%vech_a+%%vech_b)*hvec
   compute %pt(sept92virf,step,hvec)
end do step
do i=1,3
   graph(column=1,row=i,picture="*.###",vticks=5)
   # sept92virf(i) 1 nstep
end do i
*
* EC Policy Change shock.
*
compute eps0=rd(ecpolicy)
compute sigma0=hh(ecpolicy)
compute shock=%vec(%outerxx(eps0)-sigma0)
dec vect[series] aug93virf(3)
do i=1,3
   set aug93virf(i) 1 nstep = 0.0
end do i
*
do step=1,nstep
   if step==1
      compute hvec=%%vech_a*shock
   else
      compute hvec=(%%vech_a+%%vech_b)*hvec
   compute %pt(aug93virf,step,hvec)
end do step
do i=1,3
   graph(column=2,row=i,picture="*.###",vticks=5)
   # aug93virf(i) 1 nstep
end do i
spgraph(done)
```

Example 6.3 Multivariate GARCH with Asymmetry or Variance Dummies

This is the example from Section 6.3.

```
open data mhcdata.xls
cal(d) 1986:1:1
data(format=xls,org=columns) 1 3724 rnk rhs rsp
*
* We scale up the original data by 100.0
*
set rsp = rsp*100.0
set rnk = rnk*100.0
set rhs = rhs*100.0
*
* Define univariate AR's for each dependent variable
*
equation speq rsp
# constant rsp{1}
equation nkeq rnk
# constant rnk{1}
equation hseq rhs
# constant rhs{1}
*
group armeans speq nkeq hseq
*
* CC models
*
garch(model=armeans,mv=cc,asymm)
@regcrits(title="CC Model with Asymmetry")
garch(model=armeans,mv=cc,variances=varma,asymm,$
  pmethod=simplex,piters=10,method=bfgs)
@regcrits(title="CC-VARMA Model with Asymmetry")
*
* BEKK model
*
garch(model=armeans,mv=bekk,asymm,$
  pmethod=simplex,piters=10,method=bfgs)
@regcrits(title="BEKK Model with Asymmetry")
*
* Standard GARCH
*
garch(model=armeans,asymm,rvectors=rd,hmatrices=hh,$
  pmethod=simplex,piters=10,method=bfgs)
@regcrits(title="Standard GARCH with Asymmetry")
*
* Multivariate diagnostics
*
dec vect[series] rstd(%nvar)
do time=%regstart(),%regend()
   eigen(scale) hh(time) * eigfac
   compute %pt(rstd,time,%solve(eigfac,rd(time)))
end do time
```

```
*
@mvqstat(lags=10)
# rstd
@mvarchtest
# rstd
@mvjb(factor=%identity(3)) rstd
*
* Standard GARCH with shift dummy
*
set monday = %weekday(t)==1
garch(model=armeans,asymm,xreg,distrib=t,$
  pmethod=simplex,piters=10,method=bfgs)
# monday
*
* BEKK GARCH with shift dummy
*
garch(model=armeans,mv=bekk,asymm,xreg,distrib=t,$
   pmethod=simplex,piters=10,method=bfgs)
# monday
```