

[Description](#)  
[Options](#)  
[References](#)[Quick start](#)  
[Remarks and examples](#)  
[Also see](#)[Menu](#)  
[Stored results](#)[Syntax](#)  
[Methods and formulas](#)

## Description

`mgarch dcc` estimates the parameters of dynamic conditional correlation (DCC) multivariate generalized autoregressive conditionally heteroskedastic (MGARCH) models in which the conditional variances are modeled as univariate generalized autoregressive conditionally heteroskedastic (GARCH) models and the conditional covariances are modeled as nonlinear functions of the conditional variances. The conditional quasicorrelation parameters that weight the nonlinear combinations of the conditional variances follow the GARCH-like process specified in [Engle \(2002\)](#).

The DCC MGARCH model is about as flexible as the closely related varying conditional correlation MGARCH model (see [\[TS\] mgarch vcc](#)), more flexible than the conditional correlation MGARCH model (see [\[TS\] mgarch ccc](#)), and more parsimonious than the diagonal vech MGARCH model (see [\[TS\] mgarch dvech](#)).

## Quick start

Fit dynamic conditional correlation multivariate GARCH with first- and second-order ARCH components for dependent variables `y1` and `y2` using `tsset` data

```
mgarch dcc (y1 y2), arch(1 2)
```

Add regressors `x1` and `x2` and first-order GARCH component

```
mgarch dcc (y1 y2 = x1 x2), arch(1 2) garch(1)
```

Add `z1` to the model for the conditional heteroskedasticity

```
mgarch dcc (y1 y2 = x1 x2), arch(1 2) garch(1) het(z1)
```

## Menu

Statistics > Multivariate time series > Multivariate GARCH

## Syntax

```
mgarch dcc eq [eq ... eq] [if] [in] [, options]
```

where each *eq* has the form

```
(depvars = [indepvars] [, eqoptions])
```

<i>options</i>	Description
Model	
<u>arch</u> ( <i>numlist</i> )	ARCH terms for all equations
<u>garch</u> ( <i>numlist</i> )	GARCH terms for all equations
<u>het</u> ( <i>varlist</i> )	include <i>varlist</i> in the specification of the conditional variance for all equations
<u>distribution</u> ( <i>dist</i> [#])	use <i>dist</i> distribution for errors [may be <u>gaussian</u> (synonym <u>normal</u> ) or <u>t</u> ; default is <u>gaussian</u> ]
<u>constraints</u> ( <i>numlist</i> )	apply linear constraints
SE/Robust	
<u>vce</u> ( <i>vcetype</i> )	<i>vcetype</i> may be <u>oim</u> or <u>robust</u>
Reporting	
<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>from</u> ( <i>matname</i> )	initial values for the coefficients; seldom used
<u>coeflegend</u>	display legend instead of statistics

<i>eqoptions</i>	Description
<u>noconstant</u>	suppress constant term in the mean equation
<u>arch</u> ( <i>numlist</i> )	ARCH terms
<u>garch</u> ( <i>numlist</i> )	GARCH terms
<u>het</u> ( <i>varlist</i> )	include <i>varlist</i> in the specification of the conditional variance

You must tsset your data before using `mgarch dcc`; see [TS] tsset.

*indepvars* and *varlist* may contain factor variables; see [U] 11.4.3 **Factor variables**.

*depvars*, *indepvars*, and *varlist* may contain time-series operators; see [U] 11.4.4 **Time-series varlists**.

`by`, `collect`, `fp`, `rolling`, and `statsby` are allowed; see [U] 11.1.10 **Prefix commands**.

`coeflegend` does not appear in the dialog box.

See [U] 20 **Estimation and postestimation commands** for more capabilities of estimation commands.

## Options

### Model

`arch(numlist)` specifies the ARCH terms for all equations in the model. By default, no ARCH terms are specified.

`garch(numlist)` specifies the GARCH terms for all equations in the model. By default, no GARCH terms are specified.

`het(varlist)` specifies that *varlist* be included in the specification of the conditional variance for all equations. This varlist enters the variance specification collectively as multiplicative heteroskedasticity.

`distribution(dist [#])` specifies the assumed distribution for the errors. *dist* may be gaussian, normal, or t.

gaussian and normal are synonyms; each causes mgarch dcc to assume that the errors come from a multivariate normal distribution. *#* may not be specified with either of them.

t causes mgarch dcc to assume that the errors follow a multivariate Student *t* distribution, and the degree-of-freedom parameter is estimated along with the other parameters of the model. If `distribution(t #)` is specified, then mgarch dcc uses a multivariate Student *t* distribution with *#* degrees of freedom. *#* must be greater than 2.

`constraints(numlist)` specifies linear constraints to apply to the parameter estimates.

### SE/Robust

`vce(vcetype)` specifies the estimator for the variance–covariance matrix of the estimator.

`vce(oim)`, the default, specifies to use the observed information matrix (OIM) estimator.

`vce(robust)` specifies to use the Huber/White/sandwich estimator.

### Reporting

`level(#)`, `nocnsreport`; see [\[R\] Estimation options](#).

`display_options`: `nocl`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `novlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

### Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(matname)`; see [\[R\] Maximize](#) for all options except `from()`, and see below for information on `from()`. These options are seldom used.

`from(matname)` specifies initial values for the coefficients. `from(b0)` causes mgarch dcc to begin the optimization algorithm with the values in `b0`. `b0` must be a row vector, and the number of columns must equal the number of parameters in the model.

The following option is available with mgarch dcc but is not shown in the dialog box:

`coeflegend`; see [\[R\] Estimation options](#).

## Eqoptions

`noconstant` suppresses the constant term in the mean equation.

`arch(numlist)` specifies the ARCH terms in the equation. By default, no ARCH terms are specified. This option may not be specified with model-level `arch()`.

`garch(numlist)` specifies the GARCH terms in the equation. By default, no GARCH terms are specified. This option may not be specified with model-level `garch()`.

`het(varlist)` specifies that *varlist* be included in the specification of the conditional variance. This *varlist* enters the variance specification collectively as multiplicative heteroskedasticity. This option may not be specified with model-level `het()`.

## Remarks and examples

We assume that you have already read [TS] [mgarch](#), which provides an introduction to MGARCH models and the methods implemented in `mgarch dcc`.

MGARCH models are dynamic multivariate regression models in which the conditional variances and covariances of the errors follow an autoregressive-moving-average structure. The DCC MGARCH model uses a nonlinear combination of univariate GARCH models with time-varying cross-equation weights to model the conditional covariance matrix of the errors.

As discussed in [TS] [mgarch](#), MGARCH models differ in the parsimony and flexibility of their specifications for a time-varying conditional covariance matrix of the disturbances, denoted by  $\mathbf{H}_t$ . In the conditional correlation family of MGARCH models, the diagonal elements of  $\mathbf{H}_t$  are modeled as univariate GARCH models, whereas the off-diagonal elements are modeled as nonlinear functions of the diagonal terms. In the DCC MGARCH model,

$$h_{ij,t} = \rho_{ij,t} \sqrt{h_{ii,t} h_{jj,t}}$$

where the diagonal elements  $h_{ii,t}$  and  $h_{jj,t}$  follow univariate GARCH processes and  $\rho_{ij,t}$  follows the dynamic process specified in [Engle \(2002\)](#) and discussed below.

Because the  $\rho_{ij,t}$  varies with time, this model is known as the DCC GARCH model.

## □ Technical note

The DCC GARCH model proposed by Engle (2002) can be written as

$$\begin{aligned}
 \mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \boldsymbol{\epsilon}_t \\
 \boldsymbol{\epsilon}_t &= \mathbf{H}_t^{1/2} \boldsymbol{\nu}_t \\
 \mathbf{H}_t &= \mathbf{D}_t^{1/2} \mathbf{R}_t \mathbf{D}_t^{1/2} \\
 \mathbf{R}_t &= \text{diag}(\mathbf{Q}_t)^{-1/2} \mathbf{Q}_t \text{diag}(\mathbf{Q}_t)^{-1/2} \\
 \mathbf{Q}_t &= (1 - \lambda_1 - \lambda_2) \mathbf{R} + \lambda_1 \tilde{\boldsymbol{\epsilon}}_{t-1} \tilde{\boldsymbol{\epsilon}}'_{t-1} + \lambda_2 \mathbf{Q}_{t-1}
 \end{aligned} \tag{1}$$

where

$\mathbf{y}_t$  is an  $m \times 1$  vector of dependent variables;

$\mathbf{C}$  is an  $m \times k$  matrix of parameters;

$\mathbf{x}_t$  is a  $k \times 1$  vector of independent variables, which may contain lags of  $\mathbf{y}_t$ ;

$\mathbf{H}_t^{1/2}$  is the Cholesky factor of the time-varying conditional covariance matrix  $\mathbf{H}_t$ ;

$\boldsymbol{\nu}_t$  is an  $m \times 1$  vector of normal, independent, and identically distributed innovations;

$\mathbf{D}_t$  is a diagonal matrix of conditional variances,

$$\mathbf{D}_t = \begin{pmatrix} \sigma_{1,t}^2 & 0 & \cdots & 0 \\ 0 & \sigma_{2,t}^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{m,t}^2 \end{pmatrix}$$

in which each  $\sigma_{i,t}^2$  evolves according to a univariate GARCH model of the form

$$\sigma_{i,t}^2 = s_i + \sum_{j=1}^{p_i} \alpha_j \epsilon_{i,t-j}^2 + \sum_{j=1}^{q_i} \beta_j \sigma_{i,t-j}^2$$

by default, or

$$\sigma_{i,t}^2 = \exp(\gamma_t \mathbf{z}_{i,t}) + \sum_{j=1}^{p_i} \alpha_j \epsilon_{i,t-j}^2 + \sum_{j=1}^{q_i} \beta_j \sigma_{i,t-j}^2$$

when the `het()` option is specified, where  $\gamma_t$  is a  $1 \times p$  vector of parameters,  $\mathbf{z}_i$  is a  $p \times 1$  vector of independent variables including a constant term, the  $\alpha_j$ 's are ARCH parameters, and the  $\beta_j$ 's are GARCH parameters;

$\mathbf{R}_t$  is a matrix of conditional quasicorrelations,

$$\mathbf{R}_t = \begin{pmatrix} 1 & \rho_{12,t} & \cdots & \rho_{1m,t} \\ \rho_{12,t} & 1 & \cdots & \rho_{2m,t} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{1m,t} & \rho_{2m,t} & \cdots & 1 \end{pmatrix}$$

$\tilde{\boldsymbol{\epsilon}}_t$  is an  $m \times 1$  vector of standardized residuals,  $\mathbf{D}_t^{-1/2} \boldsymbol{\epsilon}_t$ ; and

$\lambda_1$  and  $\lambda_2$  are parameters that govern the dynamics of conditional quasicorrelations.  $\lambda_1$  and  $\lambda_2$  are nonnegative and satisfy  $0 \leq \lambda_1 + \lambda_2 < 1$ .

When  $\mathbf{Q}_t$  is stationary, the  $\mathbf{R}$  matrix in (1) is a weighted average of the unconditional covariance matrix of the standardized residuals  $\tilde{\epsilon}_t$ , denoted by  $\overline{\mathbf{R}}$ , and the unconditional mean of  $\mathbf{Q}_t$ , denoted by  $\overline{\mathbf{Q}}$ . Because  $\overline{\mathbf{R}} \neq \overline{\mathbf{Q}}$ , as shown by [Aielli \(2013\)](#),  $\mathbf{R}$  is neither the unconditional correlation matrix nor the unconditional mean of  $\mathbf{Q}_t$ . For this reason, the parameters in  $\mathbf{R}$  are known as quasicorrelations; see [Aielli \(2013\)](#) and [Engle \(2009\)](#) for discussions.



## Some examples

### ► Example 1: Model with common covariates

We have daily data on the stock returns of three car manufacturers—Toyota, Nissan, and Honda, from January 2, 2003, to December 31, 2010—in the variables `toyota`, `nissan` and `honda`. We model the conditional means of the returns as a first-order vector autoregressive process and the conditional covariances as a DCC MGARCH process in which the variance of each disturbance term follows a GARCH(1,1) process.

```
. use https://www.stata-press.com/data/r19/stocks
(Data from Yahoo! Finance)

. mgarch dcc (toyota nissan honda = L.toyota L.nissan L.honda, noconstant),
> arch(1) garch(1)

Calculating starting values....

Optimizing log likelihood
(setting technique to bhhh)
Iteration 0: Log likelihood = 16902.435
Iteration 1: Log likelihood = 17005.448
Iteration 2: Log likelihood = 17157.958
Iteration 3: Log likelihood = 17267.363
Iteration 4: Log likelihood = 17318.29
Iteration 5: Log likelihood = 17353.029
Iteration 6: Log likelihood = 17369.115
Iteration 7: Log likelihood = 17388.035
Iteration 8: Log likelihood = 17401.254
Iteration 9: Log likelihood = 17435.556
(switching technique to nr)
Iteration 10: Log likelihood = 17451.739
Iteration 11: Log likelihood = 17471.654 (not concave)
Iteration 12: Log likelihood = 17479.145
Iteration 13: Log likelihood = 17482.202
Iteration 14: Log likelihood = 17484.821
Iteration 15: Log likelihood = 17484.949
Iteration 16: Log likelihood = 17484.95

Refining estimates
Iteration 0: Log likelihood = 17484.95
Iteration 1: Log likelihood = 17484.95

Dynamic conditional correlation MGARCH model
Sample: 2 thru 2015
Distribution: Gaussian
Log likelihood = 17484.95
Number of obs = 2,014
Wald chi2(9) = 19.54
Prob > chi2 = 0.0210
```

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
toyota						
toyota						
L1.	-.0510867	.0339825	-1.50	0.133	-.1176911	.0155176
nissan						
L1.	.029783	.0247455	1.20	0.229	-.0187173	.0782833
honda						
L1.	-.0162824	.0300323	-0.54	0.588	-.0751447	.0425799
ARCH_toyota						
arch						
L1.	.0608224	.0086687	7.02	0.000	.0438321	.0778127
garch						
L1.	.9222202	.0111055	83.04	0.000	.9004539	.9439866
_cons	4.47e-06	1.15e-06	3.90	0.000	2.22e-06	6.72e-06

nissan						
toyota						
L1.	-.0056721	.0389348	-0.15	0.884	-.0819829	.0706386
nissan						
L1.	-.0287096	.0309379	-0.93	0.353	-.0893468	.0319275
honda						
L1.	.0154979	.0358802	0.43	0.666	-.054826	.0858217
ARCH_nissan						
arch						
L1.	.0844245	.0128192	6.59	0.000	.0592992	.1095497
garch						
L1.	.89942	.0151125	59.51	0.000	.8698	.92904
_cons	7.21e-06	1.93e-06	3.74	0.000	3.43e-06	.000011
honda						
toyota						
L1.	-.0272416	.0361819	-0.75	0.452	-.0981567	.0436736
nissan						
L1.	.0617492	.0271378	2.28	0.023	.00856	.1149383
honda						
L1.	-.0635071	.0332918	-1.91	0.056	-.1287579	.0017437
ARCH_honda						
arch						
L1.	.0490136	.0073696	6.65	0.000	.0345694	.0634577
garch						
L1.	.9331123	.0103687	89.99	0.000	.9127901	.9534345
_cons	5.35e-06	1.35e-06	3.95	0.000	2.69e-06	8.00e-06
corr(toyota, nissan)	.6689539	.0168019	39.81	0.000	.6360227	.7018851
corr(toyota, honda)	.7259623	.0140155	51.80	0.000	.6984925	.7534322
corr(nissan, honda)	.6335652	.018041	35.12	0.000	.5982055	.6689248
/Adjustment						
lambda1	.0315279	.0088383	3.57	0.000	.0142052	.0488506
lambda2	.8704109	.0613335	14.19	0.000	.7501994	.9906223

The iteration log has three parts: the dots from the search for initial values, the iteration log from optimizing the log likelihood, and the iteration log from the refining step. A detailed discussion of the optimization methods is in [Methods and formulas](#).

The header describes the estimation sample and reports a Wald test against the null hypothesis that all the coefficients on the independent variables in the mean equations are zero. Here the null hypothesis is rejected at the 5% level.



The output table first presents results for the mean or variance parameters used to model each dependent variable. Subsequently, the output table presents results for the conditional quasicorrelations. For example, the conditional quasicorrelation between the standardized residuals for Toyota and Nissan is estimated to be 0.67. Finally, the output table presents results for the adjustment parameters  $\lambda_1$  and  $\lambda_2$ . In the example at hand, the estimates for both  $\lambda_1$  and  $\lambda_2$  are statistically significant.

The DCC MGARCH model reduces to the CCC MGARCH model when  $\lambda_1 = \lambda_2 = 0$ . The output below shows that a Wald test rejects the null hypothesis that  $\lambda_1 = \lambda_2 = 0$  at all conventional levels.

```
. test _b[/Adjustment:lambda1] = _b[/Adjustment:lambda2] = 0
( 1)  [/Adjustment]lambda1 - [/Adjustment]lambda2 = 0
( 2)  [/Adjustment]lambda1 = 0
      chi2( 2) = 1102.29
      Prob > chi2 =    0.0000
```

These results indicate that the assumption of time-invariant conditional correlations maintained in the CCC MGARCH model is too restrictive for these data.

◀

## ► Example 2: Model with covariates that differ by equation

We improve the [previous example](#) by removing the insignificant parameters from the model. To remove these parameters, we specify the honda equation separately from the toyota and nissan equations:

```
. mgarch dcc (toyota nissan = , noconstant) (honda = L.nissan, noconstant),
> arch(1) garch(1)

Calculating starting values....

Optimizing log likelihood
(setting technique to bhhh)
Iteration 0:  Log likelihood = 16884.502
Iteration 1:  Log likelihood = 16970.755
Iteration 2:  Log likelihood = 17140.318
Iteration 3:  Log likelihood = 17237.807
Iteration 4:  Log likelihood = 17306.12
Iteration 5:  Log likelihood = 17342.533
Iteration 6:  Log likelihood = 17363.511
Iteration 7:  Log likelihood = 17392.501
Iteration 8:  Log likelihood = 17407.242
Iteration 9:  Log likelihood = 17448.702
(switching technique to nr)
Iteration 10: Log likelihood = 17472.199
Iteration 11: Log likelihood = 17475.84
Iteration 12: Log likelihood = 17476.345
Iteration 13: Log likelihood = 17476.35
Iteration 14: Log likelihood = 17476.35

Refining estimates
Iteration 0:  Log likelihood = 17476.35
Iteration 1:  Log likelihood = 17476.35
```

Dynamic conditional correlation MGARCH model

Sample: 2 thru 2015

Distribution: Gaussian

Log likelihood = 17476.35

Number of obs = 2,014

Wald chi2(1) = 2.21

Prob > chi2 = 0.1374

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ARCH_toyota						
arch						
L1.	.0608188	.0086675	7.02	0.000	.0438308	.0778067
garch						
L1.	.9219957	.0111066	83.01	0.000	.9002271	.9437643
_cons	4.49e-06	1.14e-06	3.95	0.000	2.27e-06	6.72e-06
ARCH_nissan						
arch						
L1.	.0876162	.01302	6.73	0.000	.0620974	.1131349
garch						
L1.	.8950964	.0152908	58.54	0.000	.865127	.9250657
_cons	7.69e-06	1.99e-06	3.86	0.000	3.79e-06	.0000116
honda						
nissan						
L1.	.019978	.0134488	1.49	0.137	-.0063811	.0463371
ARCH_honda						
arch						
L1.	.0488799	.0073767	6.63	0.000	.0344218	.063338
garch						
L1.	.9330047	.0103944	89.76	0.000	.912632	.9533774
_cons	5.42e-06	1.36e-06	3.98	0.000	2.75e-06	8.08e-06
corr(toyota,						
nissan)	.6668433	.0163209	40.86	0.000	.6348548	.6988317
corr(toyota,						
honda)	.7258101	.0137072	52.95	0.000	.6989446	.7526757
corr(nissan,						
honda)	.6313515	.0175454	35.98	0.000	.5969631	.6657399
/Adjustment						
lambda1	.0324493	.0074013	4.38	0.000	.0179429	.0469556
lambda2	.8574681	.0476273	18.00	0.000	.7641202	.950816

It turns out that the coefficient on L1.nissan in the honda equation is now statistically insignificant. We could further improve the model by removing L1.nissan from the model.

There is no mean equation for Toyota or Nissan. In [TS] mgarch dcc postestimation, we discuss prediction from models without covariates.

### ► Example 3: Model with constraints

Here we fit a bivariate DCC MGARCH model for the Toyota and Nissan shares. We believe that the shares of these car manufacturers follow the same process, so we impose the constraints that the ARCH coefficients are the same for the two companies and that the GARCH coefficients are also the same.

```
. constraint 1 _b[ARCH_toyota:L.arch] = _b[ARCH_nissan:L.arch]
. constraint 2 _b[ARCH_toyota:L.garch] = _b[ARCH_nissan:L.garch]
. mgarch dcc (toyota nissan = , noconstant), arch(1) garch(1) constraints(1 2)
```

Calculating starting values....

Optimizing log likelihood

(setting technique to bhhh)

```
Iteration 0: Log likelihood = 10307.609
Iteration 1: Log likelihood = 10656.153
Iteration 2: Log likelihood = 10862.137
Iteration 3: Log likelihood = 10987.457
Iteration 4: Log likelihood = 11062.347
Iteration 5: Log likelihood = 11135.207
Iteration 6: Log likelihood = 11245.619
Iteration 7: Log likelihood = 11253.56
Iteration 8: Log likelihood = 11294
Iteration 9: Log likelihood = 11296.364
```

(switching technique to nr)

```
Iteration 10: Log likelihood = 11296.76
Iteration 11: Log likelihood = 11297.087
Iteration 12: Log likelihood = 11297.091
Iteration 13: Log likelihood = 11297.091
```

Refining estimates

```
Iteration 0: Log likelihood = 11297.091
Iteration 1: Log likelihood = 11297.091
```

Dynamic conditional correlation MGARCH model

Sample: 1 thru 2015

Distribution: Gaussian

Log likelihood = 11297.09

Number of obs = 2,015

Wald chi2(.) = .

Prob > chi2 = .

- ( 1) [ARCH\_toyota]L.arch - [ARCH\_nissan]L.arch = 0  
 ( 2) [ARCH\_toyota]L.garch - [ARCH\_nissan]L.garch = 0

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
ARCH_toyota						
arch						
L1.	.080889	.0103229	7.84	0.000	.0606565	.1011216
garch						
L1.	.9060711	.011911	76.07	0.000	.882726	.9294163
_cons	4.21e-06	1.10e-06	3.83	0.000	2.05e-06	6.36e-06
ARCH_nissan						
arch						
L1.	.080889	.0103229	7.84	0.000	.0606565	.1011216
garch						
L1.	.9060711	.011911	76.07	0.000	.882726	.9294163
_cons	5.92e-06	1.47e-06	4.03	0.000	3.04e-06	8.80e-06
corr(toyota, nissan)	.6646283	.0187794	35.39	0.000	.6278214	.7014351
/Adjustment						
lambda1	.0446559	.0123017	3.63	0.000	.020545	.0687668
lambda2	.8686055	.0510884	17.00	0.000	.768474	.9687369

We could test our constraints by fitting the unconstrained model and performing a likelihood-ratio test. The results indicate that the restricted model is preferable.



## ► Example 4: Model with a GARCH term

In this example, we have data on fictional stock returns for the Acme and Anvil corporations, and we believe that the movement of the two stocks is governed by different processes. We specify one ARCH and one GARCH term for the conditional variance equation for Acme and two ARCH terms for the conditional variance equation for Anvil. In addition, we include the lagged value of the stock return for Apex, the main subsidiary of Anvil corporation, in the variance equation of Anvil. For Acme, we have data on the changes in an index of futures prices of products related to those produced by Acme in `afrelated`. For Anvil, we have data on the changes in an index of futures prices of inputs used by Anvil in `afinputs`.

```
. use https://www.stata-press.com/data/r19/acmeh
. mgarch dcc (acme = afrelated, noconstant arch(1) garch(1))
> (anvil = afinputs, arch(1/2) het(L.apex))
```

Calculating starting values....

Optimizing log likelihood

(setting technique to bhhh)

Iteration 0: Log likelihood = -13260.522

(iteration log omitted)

Iteration 9: Log likelihood = -12362.876

(switching technique to nr)

Iteration 10: Log likelihood = -12362.876

Refining estimates

Iteration 0: Log likelihood = -12362.876

Iteration 1: Log likelihood = -12362.876

Dynamic conditional correlation MGARCH model

Sample: 2 thru 2500

Number of obs = 2,499

Distribution: Gaussian

Wald chi2(2) = 2596.18

Log likelihood = -12362.88

Prob > chi2 = 0.0000

	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
acme						
afrelated	.950805	.0557082	17.07	0.000	.841619	1.059991
ARCH_acme						
arch						
L1.	.1063295	.015716	6.77	0.000	.0755266	.1371324
garch						
L1.	.7556294	.0391568	19.30	0.000	.6788836	.8323752
_cons	2.197566	.458343	4.79	0.000	1.29923	3.095901
anvil						
afinputs	-1.015657	.0209959	-48.37	0.000	-1.056808	-.9745054
_cons	.0808653	.019445	4.16	0.000	.0427538	.1189767
ARCH_anvil						
arch						
L1.	.5261675	.0281586	18.69	0.000	.4709777	.5813572
L2.	.2866454	.0196504	14.59	0.000	.2481314	.3251595
apex						
L1.	1.953173	.0594862	32.83	0.000	1.836582	2.069763
_cons	-.0062964	.0710842	-0.09	0.929	-.1456188	.1330261
corr(acme,						
anvil)	-.5600358	.0326358	-17.16	0.000	-.6240008	-.4960708
/Adjustment						
lambda1	.1904321	.0154449	12.33	0.000	.1601607	.2207035
lambda2	.7147267	.0226204	31.60	0.000	.6703916	.7590618

The results indicate that increases in the futures prices for related products lead to higher returns on the Acme stock, and increased input prices lead to lower returns on the Anvil stock. In the conditional variance equation for Anvil, the coefficient on `L1.apex` is positive and significant, which indicates that an increase in the return on the Apex stock leads to more variability in the return on the Anvil stock.



## Stored results

`mgarch dcc` stores the following in `e()`:

### Scalars

<code>e(N)</code>	number of observations
<code>e(k)</code>	number of parameters
<code>e(k_extra)</code>	number of extra estimates added to <code>_b</code>
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_dv)</code>	number of dependent variables
<code>e(df_m)</code>	model degrees of freedom
<code>e(ll)</code>	log likelihood
<code>e(chi2)</code>	$\chi^2$
<code>e(p)</code>	<i>p</i> -value for model test
<code>e(estdf)</code>	1 if distribution parameter was estimated, 0 otherwise
<code>e(usr)</code>	user-provided distribution parameter
<code>e(tmin)</code>	minimum time in sample
<code>e(tmax)</code>	maximum time in sample
<code>e(N_gaps)</code>	number of gaps
<code>e(rank)</code>	rank of <code>e(V)</code>
<code>e(ic)</code>	number of iterations
<code>e(rc)</code>	return code
<code>e(converged)</code>	1 if converged, 0 otherwise

### Macros

<code>e(cmd)</code>	<code>mgarch</code>
<code>e(model)</code>	<code>dcc</code>
<code>e(cmdline)</code>	command as typed
<code>e(depvar)</code>	names of dependent variables
<code>e(covariates)</code>	list of covariates
<code>e(dv_eqs)</code>	dependent variables with mean equations
<code>e(indeps)</code>	independent variables in each equation
<code>e(tvar)</code>	time variable
<code>e(hetvars)</code>	variables included in the conditional variance equations
<code>e(title)</code>	title in estimation output
<code>e(chi2type)</code>	Wald; type of model $\chi^2$ test
<code>e(vce)</code>	<i>vce</i> type specified in <code>vce()</code>
<code>e(vcetype)</code>	title used to label Std. err.
<code>e(tmins)</code>	formatted minimum time
<code>e(tmaxs)</code>	formatted maximum time
<code>e(dist)</code>	distribution for error term: gaussian or t
<code>e(arch)</code>	specified ARCH terms
<code>e(garch)</code>	specified GARCH terms
<code>e(technique)</code>	maximization technique
<code>e(properties)</code>	b V
<code>e(estat_cmd)</code>	program used to implement <code>estat</code>
<code>e(predict)</code>	program used to implement <code>predict</code>
<code>e(marginsok)</code>	predictions allowed by margins
<code>e(marginsnotok)</code>	predictions disallowed by margins
<code>e(marginsdefault)</code>	default <code>predict()</code> specification for margins
<code>e(asbalanced)</code>	factor variables <code>fvset</code> as <code>asbalanced</code>
<code>e(asobserved)</code>	factor variables <code>fvset</code> as <code>asobserved</code>

Matrices	
e(b)	coefficient vector
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(hessian)	Hessian matrix
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
e(pinfo)	parameter information, used by predict
Functions	
e(sample)	marks estimation sample

In addition to the above, the following is stored in `r()`:

Matrices	
r(table)	matrix containing the coefficients with their standard errors, test statistics, $p$ -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

## Methods and formulas

`mgarch dcc` estimates the parameters of the DCC MGARCH model by maximum likelihood. The log-likelihood function based on the multivariate normal distribution for observation  $t$  is

$$l_t = -0.5m \log(2\pi) - 0.5 \log \{ \det(\mathbf{R}_t) \} - \log \{ \det(\mathbf{D}_t^{1/2}) \} - 0.5 \tilde{\boldsymbol{\epsilon}}_t' \mathbf{R}_t^{-1} \tilde{\boldsymbol{\epsilon}}_t$$

where  $\tilde{\boldsymbol{\epsilon}}_t = \mathbf{D}_t^{-1/2} \boldsymbol{\epsilon}_t$  is an  $m \times 1$  vector of standardized residuals,  $\boldsymbol{\epsilon}_t = \mathbf{y}_t - \mathbf{C}\mathbf{x}_t$ . The log-likelihood function is  $\sum_{t=1}^T l_t$ .

If we assume that  $\boldsymbol{\nu}_t$  follow a multivariate  $t$  distribution with degrees of freedom (df) greater than 2, then the log-likelihood function for observation  $t$  is

$$l_t = \log \Gamma \left( \frac{\text{df} + m}{2} \right) - \log \Gamma \left( \frac{\text{df}}{2} \right) - \frac{m}{2} \log \{ (\text{df} - 2)\pi \} \\ - 0.5 \log \{ \det(\mathbf{R}_t) \} - \log \{ \det(\mathbf{D}_t^{1/2}) \} - \frac{\text{df} + m}{2} \log \left( 1 + \frac{\tilde{\boldsymbol{\epsilon}}_t' \mathbf{R}_t^{-1} \tilde{\boldsymbol{\epsilon}}_t}{\text{df} - 2} \right)$$

The starting values for the parameters in the mean equations and the initial residuals  $\hat{\boldsymbol{\epsilon}}_t$  are obtained by least-squares regression. The starting values for the parameters in the variance equations are obtained by a procedure proposed by [Gouriéroux and Monfort \(1997, sec. 6.2.2\)](#). The starting values for the quasicorrelation parameters are calculated from the standardized residuals  $\tilde{\boldsymbol{\epsilon}}_t$ . Given the starting values for the mean and variance equations, the starting values for the parameters  $\lambda_1$  and  $\lambda_2$  are obtained from a grid search performed on the log likelihood.

The initial optimization step is performed in the unconstrained space. Once the maximum is found, we impose the constraints  $\lambda_1 \geq 0$ ,  $\lambda_2 \geq 0$ , and  $0 \leq \lambda_1 + \lambda_2 < 1$ , and maximize the log likelihood in the constrained space. This step is reported in the iteration log as the refining step.

GARCH estimators require initial values that can be plugged in for  $\epsilon_{t-i}\epsilon'_{t-i}$  and  $\mathbf{H}_{t-j}$  when  $t-i < 1$  and  $t-j < 1$ . `mgarch dcc` substitutes an estimator of the unconditional covariance of the disturbances

$$\widehat{\Sigma} = T^{-1} \sum_{t=1}^T \widehat{\epsilon}_t \widehat{\epsilon}'_t \quad (2)$$

for  $\epsilon_{t-i}\epsilon'_{t-i}$  when  $t-i < 1$  and for  $\mathbf{H}_{t-j}$  when  $t-j < 1$ , where  $\widehat{\epsilon}_t$  is the vector of residuals calculated using the estimated parameters.

`mgarch dcc` uses numerical derivatives in maximizing the log-likelihood function.

## References

- Aielli, G. P. 2013. Dynamic conditional correlations: On properties and estimation. *Journal of Business and Economic Statistics* 31: 282–299. <https://doi.org/10.1080/07350015.2013.771027>.
- Engle, R. F. 2002. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business and Economic Statistics* 20: 339–350. <https://doi.org/10.1198/073500102288618487>.
- . 2009. *Anticipating Correlations: A New Paradigm for Risk Management*. Princeton, NJ: Princeton University Press.
- Gouriéroux, C. S., and A. Monfort. 1997. *Time Series and Dynamic Models*. Trans. ed. G. M. Gallo. Cambridge: Cambridge University Press.

## Also see

- [TS] [mgarch dcc postestimation](#) — Postestimation tools for `mgarch dcc`
- [TS] [arch](#) — Autoregressive conditional heteroskedasticity (ARCH) family of estimators
- [TS] [mgarch](#) — Multivariate GARCH models
- [TS] [tsset](#) — Declare data to be time-series data
- [TS] [var](#) — Vector autoregressive models
- [U] [20 Estimation and postestimation commands](#)

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2025 StataCorp LLC, College Station, TX, USA. All rights reserved.

For suggested citations, see the FAQ on [citing Stata documentation](#).

