Title

stata.com

mgarch ccc — Constant conditional correlation multivariate GARCH models

Description

`mgarch ccc` estimates the parameters of constant conditional correlation (CCC) 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 correlation parameters that weight the nonlinear combinations of the conditional variance are constant in the CCC MGARCH model.

The CCC MGARCH model is less flexible than the dynamic conditional correlation MGARCH model (see [TS] **mgarch dcc**) and varying conditional correlation MGARCH model (see [TS] **mgarch vcc**), which specify GARCH-like processes for the conditional correlations. The conditional correlation MGARCH models are more parsimonious than the diagonal vech MGARCH model (see [TS] **mgarch dvech**).

Quick start

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

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

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

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

Add `z1` to the model for the conditional heteroskedasticity

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

Menu

Statistics > Multivariate time series > Multivariate GARCH

Syntax

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

where each *eq* has the form

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

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

eqoptions	Description
`noconstant`	suppress constant term in the mean equation
`arch(`*numlist*`)`	ARCH terms
`garch(`*numlist*`)`	GARCH terms
`het(`*varlist*`)`	include *varlist* in the specification of the conditional variance

You must `tsset` your data before using `mgarch ccc`; 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 ccc` to assume that the errors come from a multivariate normal distribution. # cannot be specified with either of them.

`t` causes `mgarch ccc` 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 ccc` uses a multivariate Student t distribution with # degrees of freedom. # must be greater than 2.

`unconcentrated` specifies that optimization be performed on the unconcentrated log likelihood. The default is to start with the concentrated log likelihood.

`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: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `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 ccc` 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 ccc` 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

stata.com

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

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

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 CCC MGARCH model,

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

where the diagonal elements $h_{ii,t}$ and $h_{jj,t}$ follow univariate GARCH processes and ρ_{ij} is a time-invariate weight interpreted as a conditional correlation.

In the dynamic conditional correlation (DCC) and varying conditional correlation (VCC) MGARCH models discussed in [TS] **mgarch dcc** and [TS] **mgarch vcc**, the ρ_{ij} are allowed to vary over time. Although the conditional-correlation structure provides a useful tradeoff between parsimony and flexibility in the DCC MGARCH and VCC MGARCH models, the time-invariant parameterization used in the CCC MGARCH model is generally viewed as too restrictive for many applications; see Silvennoinen and Teräsvirta (2009). The baseline CCC MGARCH estimates are frequently compared with DCC MGARCH and VCC MGARCH estimates.

❑ Technical note

Formally, the CCC MGARCH model derived by Bollerslev (1990) can be written as

$$\begin{aligned}\mathbf{y}_t &= \mathbf{C}\mathbf{x}_t + \boldsymbol{\epsilon}_t\\ \boldsymbol{\epsilon}_t &= \mathbf{H}_t^{1/2}\boldsymbol{\nu}_t\\ \mathbf{H}_t &= \mathbf{D}_t^{1/2}\mathbf{R}\mathbf{D}_t^{1/2}\end{aligned}$$

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,

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

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

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

by default, or

$\sigma^2_{i,t} = \exp(\boldsymbol{\gamma}_i \mathbf{z}_{i,t}) + \sum_{j=1}^{p_i} \alpha_j \epsilon^2_{i,t-j} + \sum_{j=1}^{q_i} \beta_j \sigma^2_{i,t-j}$

when the `het()` option is specified, where $\boldsymbol{\gamma}_t$ 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 α_j's are ARCH parameters, and the β_j's are GARCH parameters; and

$\mathbf{R}$ is a matrix of time-invariant unconditional correlations of the standardized residuals $\mathbf{D}_t^{-1/2}\boldsymbol{\epsilon}_t$,

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

This model is known as the constant conditional correlation MGARCH model because $\mathbf{R}$ is time invariant.

❑

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 CCC MGARCH process in which the variance of each disturbance term follows a GARCH(1,1) process. We specify the `noconstant` option, because the returns have mean zero. The estimated constants in the variance equations are near zero in this example because of how the data are scaled.

```
. use https://www.stata-press.com/data/r18/stocks
(Data from Yahoo! Finance)
. mgarch ccc (toyota nissan honda = L.toyota L.nissan L.honda, noconstant),
> arch(1) garch(1)
Calculating starting values....
Optimizing concentrated log likelihood
(setting technique to bhhh)
Iteration 0:  Log likelihood =  16898.994
Iteration 1:  Log likelihood =  17008.914
Iteration 2:  Log likelihood =  17156.946
Iteration 3:  Log likelihood =  17249.527
Iteration 4:  Log likelihood =  17287.251
Iteration 5:  Log likelihood =    17313.5
Iteration 6:  Log likelihood =  17335.087
Iteration 7:  Log likelihood =  17356.534
Iteration 8:  Log likelihood =  17376.051
Iteration 9:  Log likelihood =  17400.035
(switching technique to nr)
Iteration 10: Log likelihood =  17423.634
Iteration 11: Log likelihood =  17440.659
Iteration 12: Log likelihood =   17446.75
Iteration 13: Log likelihood =  17447.631
Iteration 14: Log likelihood =  17447.645
Iteration 15: Log likelihood =  17447.645
Optimizing unconcentrated log likelihood
Iteration 0:  Log likelihood =  17447.645
Iteration 1:  Log likelihood =  17447.651
Iteration 2:  Log likelihood =  17447.651
Constant conditional correlation MGARCH model
Sample: 2 thru 2015                                   Number of obs =  2,014
Distribution: Gaussian                                Wald chi2(9)  =  17.46
Log likelihood = 17447.65                             Prob > chi2   = 0.0420
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
toyota						
toyota						
L1.	-.0537817	.0353211	-1.52	0.128	-.1230098	.0154463
nissan						
L1.	.026686	.024841	1.07	0.283	-.0220015	.0753734
honda						
L1.	-.0043073	.0302761	-0.14	0.887	-.0636473	.0550327
ARCH_toyota						
arch						
L1.	.0615321	.0087313	7.05	0.000	.0444191	.0786452
garch						
L1.	.9213798	.0110412	83.45	0.000	.8997395	.9430201
_cons	4.42e-06	1.12e-06	3.93	0.000	2.21e-06	6.62e-06

nissan						
toyota						
L1.	-.0232321	.0400563	-0.58	0.562	-.1017411	.0552769
nissan						
L1.	-.0299552	.0309362	-0.97	0.333	-.0905891	.0306787
honda						
L1.	.0369229	.0360532	1.02	0.306	-.0337401	.1075859
ARCH_nissan						
arch						
L1.	.0740294	.0119353	6.20	0.000	.0506366	.0974222
garch						
L1.	.9102548	.0142328	63.95	0.000	.882359	.9381506
_cons	6.36e-06	1.76e-06	3.61	0.000	2.91e-06	9.81e-06
honda						
toyota						
L1.	-.0378616	.036792	-1.03	0.303	-.1099727	.0342495
nissan						
L1.	.0551649	.0272559	2.02	0.043	.0017444	.1085855
honda						
L1.	-.0431919	.0331268	-1.30	0.192	-.1081193	.0217354
ARCH_honda						
arch						
L1.	.0433036	.0070224	6.17	0.000	.0295399	.0570674
garch						
L1.	.939117	.010131	92.70	0.000	.9192605	.9589735
_cons	5.02e-06	1.31e-06	3.83	0.000	2.45e-06	7.59e-06
corr(toyota, nissan)	.6532264	.0128035	51.02	0.000	.628132	.6783208
corr(toyota, honda)	.7185412	.0108132	66.45	0.000	.6973477	.7397346
corr(nissan, honda)	.6298972	.0135336	46.54	0.000	.6033717	.6564226

The iteration log has three parts: the dots from the search for initial values, the iteration log from optimizing the concentrated log likelihood, and the iteration log from maximizing the unconcentrated log likelihood. A detailed discussion of the optimization methods can be found 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 correlation parameters. For example, the conditional correlation between the standardized residuals for Toyota and Nissan is estimated to be 0.65.

The output above indicates that we may not need all the vector autoregressive parameters, but that each of the univariate ARCH, univariate GARCH, and conditional correlation parameters are statistically significant. That the estimated conditional correlation parameters are positive and significant indicates that the returns on these stocks rise or fall together.

That the conditional correlations are time invariant is a restrictive assumption. The DCC MGARCH model and the VCC MGARCH model nest the CCC MGARCH model. When we test the time-invariance assumption with Wald tests on the parameters of these more general models in [TS] **mgarch dcc** and [TS] **mgarch vcc**, we reject the null hypothesis that these conditional correlations are time invariant. ◁

▷ 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 ccc (toyota nissan = , noconstant) (honda = L.nissan, noconstant),
> arch(1) garch(1)

Calculating starting values....

Optimizing concentrated log likelihood

(setting technique to bhhh)
Iteration 0:  Log likelihood =   16886.88
Iteration 1:  Log likelihood =  16974.779
Iteration 2:  Log likelihood =  17147.893
Iteration 3:  Log likelihood =  17247.473
Iteration 4:  Log likelihood =  17285.549
Iteration 5:  Log likelihood =  17311.153
Iteration 6:  Log likelihood =  17333.588
Iteration 7:  Log likelihood =  17353.717
Iteration 8:  Log likelihood =  17374.895
Iteration 9:  Log likelihood =  17400.669
(switching technique to nr)
Iteration 10: Log likelihood =  17425.661
Iteration 11: Log likelihood =    17436.8
Iteration 12: Log likelihood =  17439.741
Iteration 13: Log likelihood =  17439.865
Iteration 14: Log likelihood =  17439.866

Optimizing unconcentrated log likelihood

Iteration 0:  Log likelihood =  17439.865
Iteration 1:  Log likelihood =  17439.872
Iteration 2:  Log likelihood =  17439.872
```

```
Constant conditional correlation MGARCH model
Sample: 2 thru 2015                                   Number of obs =  2,014
Distribution: Gaussian                                Wald chi2(1)  =   1.81
Log likelihood = 17439.87                             Prob > chi2   = 0.1781
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
ARCH_toyota						
arch						
L1.	.0619604	.0087942	7.05	0.000	.044724	.0791968
garch						
L1.	.9208961	.0110995	82.97	0.000	.8991414	.9426508
_cons	4.43e-06	1.13e-06	3.94	0.000	2.23e-06	6.64e-06
ARCH_nissan						
arch						
L1.	.0773095	.012328	6.27	0.000	.0531471	.1014719
garch						
L1.	.906088	.0147303	61.51	0.000	.8772171	.9349589
_cons	6.77e-06	1.85e-06	3.66	0.000	3.14e-06	.0000104
honda						
nissan						
L1.	.0186628	.0138575	1.35	0.178	-.0084975	.0458231
ARCH_honda						
arch						
L1.	.0433741	.006996	6.20	0.000	.0296622	.0570861
garch						
L1.	.9391094	.0100707	93.25	0.000	.9193712	.9588476
_cons	5.02e-06	1.31e-06	3.83	0.000	2.45e-06	7.60e-06
corr(toyota, nissan)	.652299	.0128271	50.85	0.000	.6271583	.6774396
corr(toyota, honda)	.7189531	.0108005	66.57	0.000	.6977845	.7401218
corr(nissan, honda)	.628435	.0135653	46.33	0.000	.6018475	.6550225

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.

As expected, removing the insignificant parameters from conditional mean equations had almost no effect on the estimated conditional variance parameters.

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

◁

Example 3: Model with constraints

Here we fit a bivariate CCC 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 and the GARCH coefficients are the same for the two companies.

```
. 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 ccc (toyota nissan = , noconstant), arch(1) garch(1) constraints(1 2)

Calculating starting values....

Optimizing concentrated log likelihood

(setting technique to bhhh)
Iteration 0:  Log likelihood =  10317.225
Iteration 1:  Log likelihood =  10630.464
Iteration 2:  Log likelihood =  10865.964
Iteration 3:  Log likelihood =  11063.329
  (output omitted)
Iteration 8:  Log likelihood =  11273.962
Iteration 9:  Log likelihood =  11274.409
(switching technique to nr)
Iteration 10: Log likelihood =  11274.494
Iteration 11: Log likelihood =  11274.499
Iteration 12: Log likelihood =  11274.499

Optimizing unconcentrated log likelihood

Iteration 0:  Log likelihood =  11274.499
Iteration 1:  Log likelihood =  11274.501
Iteration 2:  Log likelihood =  11274.501

Constant conditional correlation MGARCH model
Sample: 1 thru 2015                                   Number of obs = 2,015
Distribution: Gaussian                                Wald chi2(.)  =     .
Log likelihood = 11274.5                              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.	.0742677	.0095467	7.78	0.000	.0555564	.092979
garch						
L1.	.9131676	.0111563	81.85	0.000	.8913017	.9350335
_cons	3.77e-06	1.02e-06	3.71	0.000	1.78e-06	5.77e-06
ARCH_nissan						
arch						
L1.	.0742677	.0095467	7.78	0.000	.0555564	.092979
garch						
L1.	.9131676	.0111563	81.85	0.000	.8913017	.9350335
_cons	5.30e-06	1.36e-06	3.89	0.000	2.63e-06	7.97e-06
corr(toyota, nissan)	.651389	.0128482	50.70	0.000	.626207	.6765709

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/r18/acmeh
. mgarch ccc (acme = afrelated, noconstant arch(1) garch(1))
> (anvil = afinputs, arch(1/2) het(L.apex))
Calculating starting values....
Optimizing concentrated log likelihood
(setting technique to bhhh)
Iteration 0:  Log likelihood = -12996.245
Iteration 1:  Log likelihood = -12609.982
Iteration 2:  Log likelihood = -12563.103
Iteration 3:  Log likelihood =  -12554.73
Iteration 4:  Log likelihood = -12554.542
Iteration 5:  Log likelihood = -12554.534
Iteration 6:  Log likelihood = -12554.534
Iteration 7:  Log likelihood = -12554.534
Optimizing unconcentrated log likelihood
Iteration 0:  Log likelihood = -12554.534
Iteration 1:  Log likelihood = -12554.533
Constant conditional correlation MGARCH model
Sample: 2 thru 2500                                   Number of obs =   2,499
Distribution: Gaussian                                Wald chi2(2)  = 2212.30
Log likelihood = -12554.53                            Prob > chi2   =  0.0000
```

	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
acme						
afrelated	.9175148	.0651088	14.09	0.000	.7899039	1.045126
ARCH_acme						
arch						
L1.	.0798719	.0169526	4.71	0.000	.0466455	.1130983
garch						
L1.	.7336823	.0601569	12.20	0.000	.6157768	.8515877
_cons	2.880836	.760206	3.79	0.000	1.390859	4.370812
anvil						
afinputs	-1.015561	.0226437	-44.85	0.000	-1.059942	-.97118
_cons	.0703606	.0211689	3.32	0.001	.0288703	.1118508
ARCH_anvil						
arch						
L1.	.4893288	.0286012	17.11	0.000	.4332714	.5453862
L2.	.2782296	.0208172	13.37	0.000	.2374287	.3190305
apex						
L1.	1.894972	.0616293	30.75	0.000	1.774181	2.015763
_cons	.1034111	.0735512	1.41	0.160	-.0407466	.2475688
corr(acme, anvil)	-.5354047	.0143275	-37.37	0.000	-.563486	-.5073234

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. That the estimated conditional correlation between the two returns is -0.54 indicates that these

returns tend to move in opposite directions; in other words, an increase in the return for the Acme stock tends to be associated with a decrease in the return for the Anvil stock, and vice versa.

◁

Stored results

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

Scalars

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

Macros

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

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 ccc estimates the parameters of the CCC MGARCH model by maximum likelihood. The unconcentrated log-likelihood function based on the multivariate normal distribution for observation t is

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

where $\widetilde{\boldsymbol{\epsilon}}_t = \mathbf{D}_t^{-1/2}\boldsymbol{\epsilon}_t$ 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 unconcentrated log-likelihood function for observation t is

$$\begin{aligned} l_t = {} & \log\Gamma\left(\frac{\mathrm{df}+m}{2}\right) - \log\Gamma\left(\frac{\mathrm{df}}{2}\right) - \frac{m}{2}\log\{(\mathrm{df}-2)\pi\} \\ & - 0.5\log\{\det(\mathbf{R})\} - \log\left\{\det\left(\mathbf{D}_t^{1/2}\right)\right\} - \frac{\mathrm{df}+m}{2}\log\left(1+\frac{\widetilde{\boldsymbol{\epsilon}}_t\mathbf{R}^{-1}\widetilde{\boldsymbol{\epsilon}}_t'}{\mathrm{df}-2}\right) \end{aligned} \tag{2}$$

The correlation matrix $\mathbf{R}$ can be concentrated out of (1) and (2) by defining the (i,j)th element of $\mathbf{R}$ as

$$\widehat{\rho}_{ij} = \left(\sum_{t=1}^{T}\widetilde{\boldsymbol{\epsilon}}_{it}\widetilde{\boldsymbol{\epsilon}}_{jt}\right)\left(\sum_{t=1}^{T}\widetilde{\boldsymbol{\epsilon}}_{it}^2\right)^{-\frac{1}{2}}\left(\sum_{t=1}^{T}\widetilde{\boldsymbol{\epsilon}}_{jt}^2\right)^{-\frac{1}{2}}$$

mgarch ccc starts the optimization process with the concentrated log-likelihood function.

The starting values for the parameters in the mean equations and the initial residuals $\widehat{\boldsymbol{\epsilon}}_t$ 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). If the optimization is started with the unconcentrated log likelihood, then the initial values for the parameters in $\mathbf{R}$ are calculated from the standardized residuals $\widetilde{\boldsymbol{\epsilon}}_t$.

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

$$\widehat{\boldsymbol{\Sigma}} = T^{-1}\sum_{t=1}^{T}\widehat{\boldsymbol{\epsilon}}_t\widehat{\boldsymbol{\epsilon}}_t' \tag{3}$$

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

`mgarch ccc` requires a sample size that at the minimum is equal to the number of parameters in the model plus twice the number of equations.

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

References

Bollerslev, T. 1990. Modelling the coherence in short-run nominal exchange rates: A multivariate generalized ARCH model. *Review of Economics and Statistics* 72: 498–505. https://doi.org/10.2307/2109358.

Gouriéroux, C. S., and A. Monfort. 1997. *Time Series and Dynamic Models*. Trans. ed. G. M. Gallo. Cambridge: Cambridge University Press.

Silvennoinen, A., and T. Teräsvirta. 2009. Multivariate GARCH models. In *Handbook of Financial Time Series*, ed. T. G. Andersen, R. A. Davis, J.-P. Kreiß, and T. Mikosch, 201–229. New York: Springer. https://doi.org/10.1007/978-3-540-71297-8_9.

Also see

[TS] **mgarch ccc postestimation** — Postestimation tools for mgarch ccc

[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**