Chapter 11: Garch models

For the moment the GARCH library is rather limited[1], since it allows only the estimation of a GARCH(p,q) model. The function itself is rather general, since p and q are not restricted and since it allows a GARCH estimation to be applied to the residuals of any linear model.

Formally, the garch function estimates the following model:

```
 y(t) = X(t)*b + e(t), e(t) ~> N(0,h(t))
h(t) = a0 +
        ar(1)*h(t-1) + ... + ar(p)*h(t-p) +
        ma(1)*e(t-1)^2  + ... + ma(q)*e(t-q)^2
```

The estimation of such a model is made trough the following instruction:

```
garch(y,x1,...,xn,'b=[b1;....,bn]','a0=p','ar=[ar1;...;arp]','ma=[ma1;...;maq]',options)
```

Here y is the endogenous variable: it can take any type allowed for an endogenous variable in any regression, that is: a real vector, a ts or -better- one of these 2 types between quotes, so that the name of the variable is kept for printings. x1,...,xn are the name of the exogenous variables: they can take any type allowed for an exogenous variable in any regression, that is: a real vector matrix, a ts or -better- one of these 2 types between quotes, so that the name of the variable is kept for printings. All types can be mixed and you can even enter a list of various elements of such types.

The function needs initial values. Beyond feeding the maximisation program with starting values, these starting values have another purpose: determine the p and q orders of the GARCH. All parameters must be given initial values and the names b, a0, ar and ma are compulsory names. The vectors [b1;....,bn], [ar1;...;arp] and [ma1;...;maq] and the value p can be explicit values such as [0.1;0.2;0.1] or vectors existing in the environment (such as beta): see garch_d for an example that mixes these 2 options.

The options are: the traditional 'noprint', 'optmeth=maxlik' or 'optmeth=optim' and any option to maxlik, if this is the optimisation program used. I however recommend to use 'optim', since robustness is greatly needed to estimate a GARCH model.

The estimation imposes the constraint that all parameters a0, ar and ma are positive: this a sufficient (although not necessary, see Hamilton (1994)) condition to obtain a positive variance h(t). This is done through a transformation of the original parameters, programmed in function garch_trans.

The function uses exact derivatives of the likelihood to feed the maximisation program: this is recommended in such a context. This is done by 2 functions: garch_grad for the true parameters of the garch model and garch_grad2 for the parameters before their transformation. (Minus) the log-likelihood of the function is calculated by function garch_like.

As an example, here is the result of the estimation of a garch(1,1) model, taken from Greene(2000):

[1] But some new features (E-Garch, GARCH-M,…) should be added soon….

```
-->load('SCI/macros/grocer/db/garchd.dat');
```
// load the data base

```
-->bounds('1949q1','1983q4');
```
// set the bounds

```
r1    =    ols('gnpdef','cte','lagts(gnpdef)','lagts(2,gnpdef)','lagts(3,gnpdef)',
'lagts(4,gnpdef)','noprint'); beta = r1('beta'); s = r1('ser');
```
// do ols to have intial values

```
r2   =   garch('gnpdef','cte','lagts(gnpdef)','lagts(2,gnpdef)','lagts(3,gnpdef)',
'lagts(4,gnpdef)','a0=s','ar=0','ma=0','b=beta','meth=maxlik');
```
// perform garch estimates, using maxlik optimisation method, using initial values for b and a0, ar
// and ma being set to 0

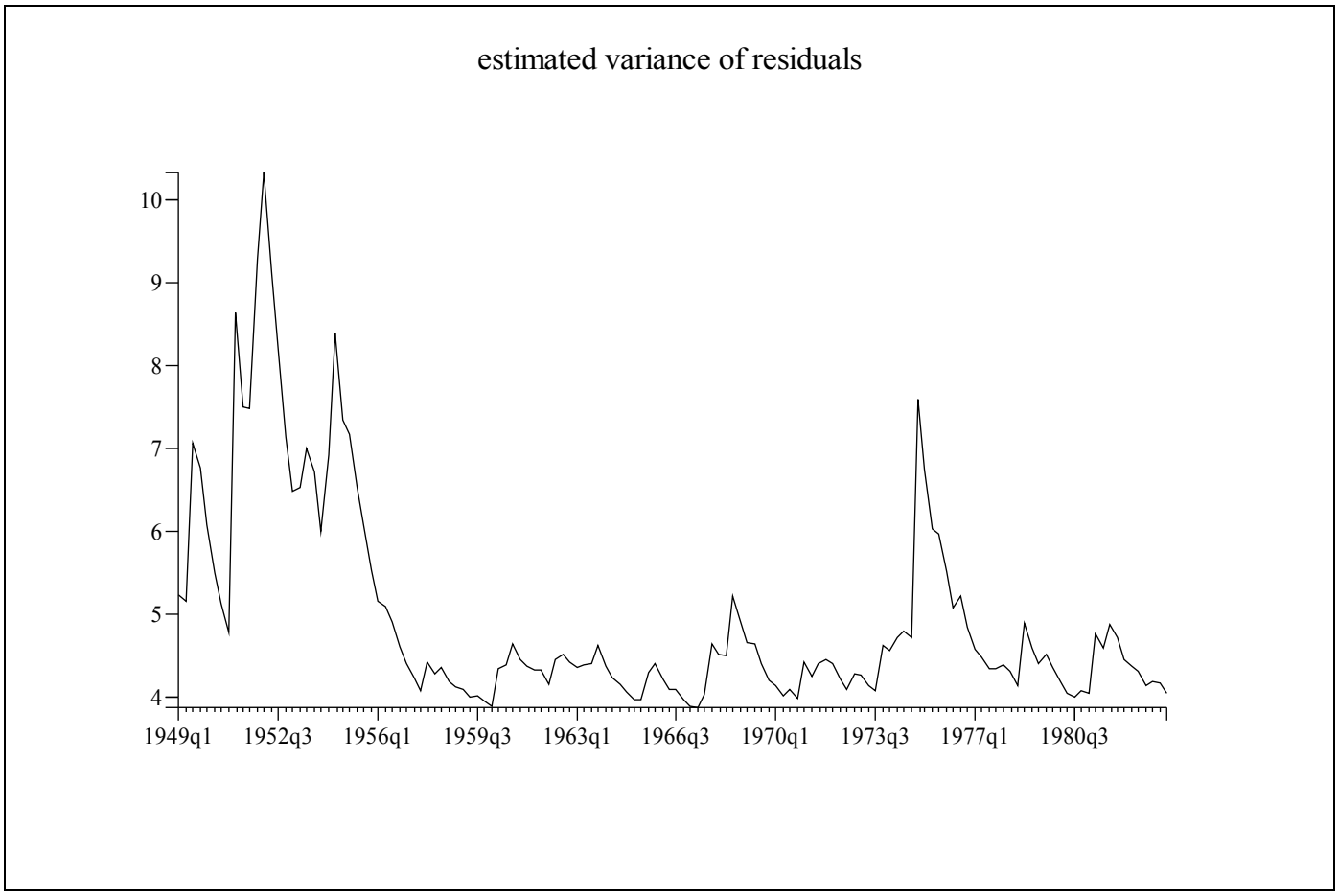
```
CONVERGENCE CRITERIA MET: Change in Objective Function



garch estimation results for dependent variable: gnpdef
estimation period: 1949q1-1983q4
number of observations: 140
number of variables: 8
R² =  .5163777      ajusted R² = .5020481
Overall F test: F(7,132) = 36.035864       p-value = 0
standard error of the regression: 2.3524133
sum of squared residuals: 747.06953
DW(0) =1.9978162
Belsley, Kuh, Welsch Condition index: 19
log likelihood: -310.64706


variable        coeff     t-statistic p value
cte              .8981733 2.4678099    .0135943
lagts(gnpdef)    .4048429 4.4103101    .0000103
lagts(2,gnpdef)  .2946811 3.0328436    .0024226
lagts(3,gnpdef)  .0182799  .1843161    .8537655
lagts(4,gnpdef)  .0776021  .8591777    .3902425
sig0             .8178598  .5217132    .60187
sig_ar(1)        .7702418 1.9299701    .0536105
sig_ma(1)        .0578560  .8388696    .4015425


                        *
                     *     *
```



// The program provides the result of the estimation and a graph of the time-varying estimated
// variance of residuals

The functions and their specifications.

garch_____garch estimation

CALLING SEQUENCE
rgarch=garch(namey,varargin)

PARAMETERS

INPUT:
* grocer_namey = a time series, a real (nx1) vector or a string equal to the name of a time series or a (nx1) real vector between quotes
* varargin = arguments which can be:
  . a time series
  . a real (nxp) vector
  . a string equal to the name of a time series or a (nxp) real vector between quotes
  . the string 'noprint' if the user doesn't want to print the results of the regression
  . 'b = xx' where xx is a (k x 1) vector of B starting values

. 'a0 = xx' where xx is a0 starting values to feed the maximisation programm
. 'ar = xx' where xx is (p x 1) vector of ar starting values to feed the maximisation program
. 'ma = xx' where xx is (q x 1) vector of ma starting values to feed the maximisation program
. 'meth=maxlik' or 'meth=optim' (default)
. any option to maxlik (see maxlik for a list)

OUTPUT:
a result tlist with:
. rgarch('meth') = 'garch'
. rgarch('optmeth') = 'optim' or 'maxlik' (method used for the optimisation)
. rgarch('y') = y data vector
. rgarch('x') = x data matrix
. rgarch('nobs') = # observations
. rgarch('nvar') = # variables
. rgarch('beta') = bhat
. rgarch('yhat') = yhat
. rgarch('resid') = residuals
. rgarch('vcovar') = estimated variance-covariance matrix of beta
. rgarch('sige') = estimated variance of the residuals
. rgarch('sigu') = sum of squared residuals
. rgarch('ser') = standard error of the regression
. rgarch('tstat') = t-stats
. rgarch('pvalue') = pvalue of the betas
. rgarch('dw') = Durbin-Watson Statistic
. rgarch('condindex') = multicolinearity cond index
. rgarch('prescte') = boolean indicating the presence or absence of a constant in the regression
. rgarch('b') = estimated b
. rgarch('a0') = estimated a0
. rgarch('ar') = estimated ar
. rgarch('ma') = estimated ma
. rgarch('sigt') = estimated h(t)
. rgarch('like') = log-likelihood
. rgarch('rbar') = rbar-squared
. rgarch('f') = F-stat for the nullity of coefficients other than the constant
. rgarch('pvaluef') = its significance level
. rgarch('prests') = boolean indicating the presence or absence of a time series in the regression
. rgarch('namey') = name of the y variable
. rgarch('namex') = name of the x variables
. rgarch('bounds') = if there is a timeseries in the regression, the bounds of the regression

DESCRIPTION

Estimates a regression with garch(p,q) errors:

$$y(t) = X(t)*b + e(t),\ e(t) = N(0,h(t))$$
$$h(t) = a0 + ar(1)*h(t-1) + \ldots + ar(p)*h(t-p) + ma(1)*e(t-1)^2 + \ldots + ma(q)*e(t-q)^2$$

Example:

r2 = garch('gnpdef','cte','lagts(gnpdef)','lagts(2,gnpdef)','lagts(3,gnpdef)','lagts(4,gnpdef)','a0=s', 'ar=0','ma=0','b=beta','meth=maxlik');

Example taken from function garch_d(). Estimates a garch(1,1) model, taken from Greene's book. Starting values for a0, ar and ma are s (in garch_d, s is the estimated variance of the corresponding ols model), 0 and 0. Starting value for b is also taken from ols estimation. Optimisation method is maxlik.

garch_grad2_____Generates garch gradient

CALLING SEQUENCE
[g,dht,gradt]=garch_grad2(parm,nar,nma,y,x)

PARAMETERS

INPUT:
* parm= vector of parameters (beta, a0, ar and ma in that order)
* nar = # of ar parameters
* nma = # of ma parameters)
* y = (n x 1) vector of the endogenous variable
* x = (n x k) vector of the exogenous variables

OUTPUT:
* g = (k+nar+nma+1 x 1) -gradient at param
* dht = (nobs x nar+nma+1) derivative of sigt w.r.t a0, ar, ma
* scores = (k+nar+nma+1 x 1) sub-gradient at each date

DESCRIPTION

Generates garch likelihood gradient. The maximised parameters are the parameters before the transformation that insures the positivity of the variance, whereas garch_gard applies to the true garch parameters, that is the ones obtained after this transformation.

Example:
g=garch_grad2(p,nar,nma,grocer_y,grocer_x)

Example taken from function garch().

garch_grad_____Generates garch gradient

CALLING SEQUENCE
[g,dht,gradt]=garch_grad(parm,nar,nma,y,x)

PARAMETERS

INPUT:
* parm= vector of parameters (beta, a0, ar and ma in that order)

* nar = # of ar parameters
* nma = # of ma parameters)
* y = (n x 1) vector of the endogenous variable
* x = (n x k) vector of the exogenous variables

OUTPUT:
* g = (k+nar+nma+1 x 1) -gradient at param
* dht = (nobs x nar+nma+1) derivative of sigt w.r.t a0, ar, ma
* scores = (k+nar+nma+1 x 1) sub-gradient at each date

DESCRIPTION

Generates garch likelihood gradient with respect to the true garch parameters. This in not the function used for maximisation, because maximised parameters are transformed in order to insure the positivity of the variance.

Example:
[g,dht,scores]=garch_grad(parmd,nar,nma,grocer_y,grocer_x)

Example taken from function garch().

garch_like_____log likelihood for garch model

CALLING SEQUENCE
llik=garch_like(parm,nar,nma,y,x)

PARAMETERS

INPUT:
* parm= vector of parameters (beta, a0, ar and ma in that order)
* nar = # of ar parameters
* nma = # of ma parameters)
* y = (n x 1) vector of the endogenous variable
* x = (n x k) vector of the exogenous variables

OUTPUT:
-log likelihood function value

DESCRIPTION

Generates log likelihood for garch model. Parameters are transformed in order to ensure the positivity of the variance.

Example:
f=garch_like(p,nar,nma,grocer_y,grocer_x)

Example taken from function garch().

garch_sigt_____Generates garch sigmas

CALLING SEQUENCE
sigt=garch_sigt(y,x,param,nar,nma)

PARAMETERS

INPUT:
* y = (n x 1) vector of the endogenous variable
* x = (n x k) vector of the exogenous variables
* parm= vector of parameters (beta, a0, ar and ma in that order)
* nar = # of ar parameters
* nma = # of ma parameters)

OUTPUT:
sigt=(nx1) vector of sigma(t) estimates

DESCRIPTION

Generates garch model sigmas over time given maximum likelihood estimates

Example:
sigt = garch_sigt(grocer_y,grocer_x,parm1,nar,nma);

Example taken from function garch().

garch_trans_____Transforms garch parameters

CALLING SEQUENCE
param=garch_trans(param,k)

PARAMETERS

INPUT:
* param = [b ; a0 ; ar ; ma]
* k = size(b,1)

OUTPUT:
param = vector of transformed parameters

DESCRIPTION

Function to transform the a0,ar(p),ma(q) parameters of a garch(p,q) in order that the variance is always positive.

Example:
parmd = garch_trans(parm1,k);

Example taken from function garch().