NAG Toolbox

nag_tsa_uni_garch_exp_estim (g13fg)

1 Purpose

nag_tsa_uni_garch_exp_estim (g13fg) estimates the parameters of a univariate regression-exponential GARCH(p,q) process (see Engle and Ng (1993)).

2 Syntax

```
[theta, se, sc, covr, hp, et, ht, lgf, ifail] = nag_tsa_uni_garch_exp_estim
(dist, yt, x, ip, iq, nreg, mn, theta, hp, copts, maxit, tol, 'num', num,
'npar', npar)
```

```
[theta, se, sc, covr, hp, et, ht, lgf, ifail] = g13fg(dist, yt, x, ip, iq, nreg,
mn, theta, hp, copts, maxit, tol, 'num', num, 'npar', npar)
```

Note: the interface to this routine has changed since earlier releases of the toolbox:

At Mark 25: **nreg** was made optional.

3 Description

A univariate regression-exponential GARCH(p,q) process, with q coefficients α_i, for $i = 1,2,\dots,q$, q coefficients ϕ_i, for $i = 1,2,\dots,q$, p coefficients, β_i, for $i = 1,2,\dots,p$, and k linear regression coefficients b_i, for $i = 1,2,\dots,k$, can be represented by:

$$y_t = b_o + x_t^{\mathrm{T}} b + \epsilon_t$$
$$ln(h_t) = \alpha_0 + \sum_{i=1}^{q} \alpha_i z_{t-i} + \sum_{i=1}^{q} \phi_i(|z_{t-i}| - E[|z_{t-i}|]) + \sum_{i=1}^{p} \beta_i ln(h_{t-i}), \quad t = 1,2,\dots,T \tag{1}$$

where $z_t = \frac{\epsilon_t}{\sqrt{h_t}}$, $E[|z_{t-i}|]$ denotes the expected value of $|z_{t-i}|$ and $\epsilon_t \mid \psi_{t-1} = N(0,h_t)$ or $\epsilon_t \mid \psi_{t-1} = S_t(df,h_t)$. Here S_t is a standardized Student's t-distribution with df degrees of freedom and variance h_t, T is the number of terms in the sequence, y_t denotes the endogenous variables, x_t the exogenous variables, b_o the regression mean, b the regression coefficients, ϵ_t the residuals, h_t the conditional variance, df the number of degrees of freedom of the Student's t-distribution, and ψ_t the set of all information up to time t.

nag_tsa_uni_garch_exp_estim (g13fg) provides an estimate $\hat{\theta}$, for the vector $\theta = (b_o, b^{\mathrm{T}}, \omega^{\mathrm{T}})$ where $b^{\mathrm{T}} = (b_1,\dots,b_k)$, $\omega^{\mathrm{T}} = (\alpha_0,\alpha_1,\dots,\alpha_q,\phi_1,\dots,\phi_q,\beta_1,\dots,\beta_p,\gamma)$ when **dist** = 'N', and $\omega^{\mathrm{T}} = (\alpha_0,\alpha_1,\dots,\alpha_q,\phi_1,\dots,\phi_q,\beta_1,\dots,\beta_p,\gamma,df)$ when **dist** = 'T'.

mn, **nreg** can be used to simplify the GARCH(p,q) expression in (1) as follows:

No Regression and No Mean

$y_t = \epsilon_t$,

mn $= 0$,

nreg $= 0$ and

θ is a $(2 \times q + p + 1)$ vector when **dist** = 'N', and a $(2 \times q + p + 2)$ vector, when **dist** = 'T'.

No Regression

$y_t = b_o + \epsilon_t$,

mn $= 1$,

nreg $= 0$ and

θ is a $(2 \times q + p + 2)$ vector when **dist** = 'N' and a $(2 \times q + p + 3)$ vector, when **dist** = 'T'.

Note: if the $y_t = \mu + \epsilon_t$, where μ is known (not to be estimated by nag_tsa_uni_garch_exp_estim (g13fg)) then (1) can be written as $y_t^\mu = \epsilon_t$, where $y_t^\mu = y_t - \mu$. This corresponds to the case **No Regression and No Mean**, with y_t replaced by $y_t - \mu$.

No Mean

$y_t = x_t^{\mathrm{T}} b + \epsilon_t$,

mn $= 0$,

nreg $= k$ and

θ is a $(2 \times q + p + 1 + k)$ vector when **dist** = 'N' and a $(2 \times q + p + 2 + k)$ vector, when **dist** = 'T'.

4 References

Bollerslev T (1986) Generalised autoregressive conditional heteroskedasticity *Journal of Econometrics* **31** 307–327

Engle R (1982) Autoregressive conditional heteroskedasticity with estimates of the variance of United Kingdom inflation *Econometrica* **50** 987–1008

Engle R and Ng V (1993) Measuring and testing the impact of news on volatility *Journal of Finance* **48** 1749–1777

Glosten L, Jagannathan R and Runkle D (1993) Relationship between the expected value and the volatility of nominal excess return on stocks *Journal of Finance* **48** 1779–1801

Hamilton J (1994) *Time Series Analysis* Princeton University Press

5 Parameters

5.1 Compulsory Input Parameters

1: **dist** – CHARACTER(1)

The type of distribution to use for e_t.

dist = 'N'
A Normal distribution is used.

dist = 'T'
A Student's t-distribution is used.

Constraint: **dist** = 'N' or 'T'.

2: **yt**(**num**) – REAL (KIND=nag_wp) array

The sequence of observations, y_t, for $t = 1, 2, \ldots, T$.

3: **x**(ldx, :) – REAL (KIND=nag_wp) array

The first dimension of the array **x** must be at least **num**.

The second dimension of the array **x** must be at least **nreg**.

Row t of **x** must contain the time dependent exogenous vector x_t, where $x_t^{\mathrm{T}} = \left(x_t^1, \ldots, x_t^k\right)$, for $t = 1, 2, \ldots, T$.

4: **ip** – INTEGER

The number of coefficients, β_i, for $i = 1, 2, \ldots, p$.

Constraint: $\mathbf{ip} \geq 0$ (see also **npar**).

5: **iq** – INTEGER

The number of coefficients, α_i, for $i = 1, 2, \ldots, q$.

Constraint: $\mathbf{iq} \geq 1$ (see also **npar**).

6: **nreg** – INTEGER

k, the number of regression coefficients.

Constraint: $\mathbf{nreg} \geq 0$ (see also **npar**).

7: **mn** – INTEGER

If $\mathbf{mn} = 1$, the mean term b_0 will be included in the model.

Constraint: $\mathbf{mn} = 0$ or 1.

8: **theta**(**npar**) – REAL (KIND=nag_wp) array

The initial parameter estimates for the vector θ.

The first element must contain the coefficient α_o and the next **iq** elements must contain the autoregressive coefficients α_i, for $i = 1, 2, \ldots, q$.

The next **iq** elements contain the coefficients ϕ_i, for $i = 1, 2, \ldots, q$.

The next **ip** elements must contain the moving average coefficients β_i, for $i = 1, 2, \ldots, p$.

If **dist** = 'T', the next element must contain an estimate for df, the number of degrees of freedom of the Student's t-distribution.

If $\mathbf{mn} = 1$, the next element must contain the mean term b_o.

If $\mathbf{copts} = \mathit{false}$, the remaining **nreg** elements are taken as initial estimates of the linear regression coefficients b_i, for $i = 1, 2, \ldots, k$.

9: **hp** – REAL (KIND=nag_wp)

If $\mathbf{copts} = \mathit{false}$ then **hp** is the value to be used for the pre-observed conditional variance, otherwise **hp** is not referenced.

10: **copts** – LOGICAL

If $\mathbf{copts} = \mathit{true}$, the function provides initial parameter estimates of the regression terms, otherwise these are provided by you.

11: **maxit** – INTEGER

The maximum number of iterations to be used by the optimization function when estimating the GARCH(p, q) parameters.

Constraint: $\mathbf{maxit} > 0$.

12: **tol** – REAL (KIND=nag_wp)

The tolerance to be used by the optimization function when estimating the GARCH(p, q) parameters.

5.2 Optional Input Parameters

1: **num** – INTEGER

Default: the dimension of the array **yt** and the first dimension of the array **x**. (An error is raised if these dimensions are not equal.)

T, the number of terms in the sequence.

Constraints:

num $\geq \max($**ip**, **iq**$)$;
num $\geq$ **nreg** + **mn**.

2: **npar** – INTEGER

Default: the dimension of the array **theta**.

The number of parameters to be included in the model. **npar** $= 1 + 2 \times$ **iq** + **ip** + **mn** + **nreg** when **dist** = 'N' and **npar** $= 2 + 2 \times$ **iq** + **ip** + **mn** + **nreg** when **dist** = 'T'.

Constraint: **npar** < 20.

5.3 Output Parameters

1: **theta**(**npar**) – REAL (KIND=nag_wp) array

The estimated values $\hat{\theta}$ for the vector θ.

The first element contains the coefficient α_o and the next **iq** elements contain the coefficients α_i, for $i = 1, 2, \ldots, q$.

The next **iq** elements contain the coefficients ϕ_i, for $i = 1, 2, \ldots, q$.

The next **ip** elements are the moving average coefficients β_i, for $i = 1, 2, \ldots, p$.

If **dist** = 'T', the next element contains an estimate for df then the number of degrees of freedom of the Student's t-distribution.

If **mn** = 1, the next element contains an estimate for the mean term b_o.

The final **nreg** elements are the estimated linear regression coefficients b_i, for $i = 1, 2, \ldots, k$.

2: **se**(**npar**) – REAL (KIND=nag_wp) array

The standard errors for $\hat{\theta}$.

The first element contains the standard error for α_o and the next **iq** elements contain the standard errors for α_i, for $i = 1, 2, \ldots, q$. The next **iq** elements contain the standard errors for ϕ_i, for $i = 1, 2, \ldots, q$. The next **ip** elements are the standard errors for β_j, for $j = 1, 2, \ldots, p$.

If **dist** = 'T', the next element contains the standard error for df, the number of degrees of freedom of the Student's t-distribution.

If **mn** = 1, the next element contains the standard error for b_o.

The final **nreg** elements are the standard errors for b_j, for $j = 1, 2, \ldots, k$.

3: **sc**(**npar**) – REAL (KIND=nag_wp) array

The scores for $\hat{\theta}$.

The first element contains the scores for α_o, the next **iq** elements contain the scores for α_i, for $i = 1, 2, \ldots, q$, the next **iq** elements contain the scores for ϕ_i, for $i = 1, 2, \ldots, q$, the next **ip** elements are the scores for β_j, for $j = 1, 2, \ldots, p$.

If **dist** = 'T', the next element contains the scores for df, the number of degrees of freedom of the Student's t-distribution.

If **mn** $= 1$, the next element contains the score for b_o.

The final **nreg** elements are the scores for b_j, for $j = 1, 2, \ldots, k$.

4: **covr**($ldcovr$, **npar**) – REAL (KIND=nag_wp) array

The covariance matrix of the parameter estimates $\hat{\theta}$, that is the inverse of the Fisher Information Matrix.

5: **hp** – REAL (KIND=nag_wp)

If **copts** $= true$ then **hp** is the estimated value of the pre-observed conditional variance.

6: **et**(**num**) – REAL (KIND=nag_wp) array

The estimated residuals, ϵ_t, for $t = 1, 2, \ldots, T$.

7: **ht**(**num**) – REAL (KIND=nag_wp) array

The estimated conditional variances, h_t, for $t = 1, 2, \ldots, T$.

8: **lgf** – REAL (KIND=nag_wp)

The value of the log-likelihood function at $\hat{\theta}$.

9: **ifail** – INTEGER

ifail $= 0$ unless the function detects an error (see Section 5).

6 Error Indicators and Warnings

Note: nag_tsa_uni_garch_exp_estim (g13fg) may return useful information for one or more of the following detected errors or warnings.

Errors or warnings detected by the function:

ifail $= 1$

On entry, **nreg** < 0,
or **mn** > 1,
or **mn** < 0,
or **iq** < 1,
or **ip** < 0,
or **npar** ≥ 20,
or **npar** has an invalid value,
or $ldcovr <$ **npar**,
or $ldx <$ **num**,
or **dist** $\neq$ 'N',
or **dist** $\neq$ 'T',
or **maxit** ≤ 0,
or **num** $< \max($**ip**, **iq**$)$,
or **num** $<$ **nreg** $+$ **mn**.

ifail $= 2$

On entry, $lwork < ($**nreg** $+ 3) \times$ **num** $+ 3$.

ifail $= 3$

The matrix X is not full rank.

ifail $= 4$

The information matrix is not positive definite.

ifail $= 5$

The maximum number of iterations has been reached.

ifail $= 6$

The log-likelihood cannot be optimized any further.

ifail $= 7$

No feasible model parameters could be found.

ifail $= -99$

An unexpected error has been triggered by this routine. Please contact NAG.

ifail $= -399$

Your licence key may have expired or may not have been installed correctly.

ifail $= -999$

Dynamic memory allocation failed.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

This example fits a GARCH$(1,2)$ model with Student's t-distributed residuals to some simulated data.

The process parameter estimates, $\hat{\theta}$, are obtained using nag_tsa_uni_garch_exp_estim (g13fg), and a four step ahead volatility estimate is computed using nag_tsa_uni_garch_exp_forecast (g13fh).

The data was simulated using nag_rand_times_garch_exp (g05pg).

9.1 Program Text

```
function g13fg_example

fprintf('g13fg example results\n\n');

num = 100;
mn  = nag_int(1);
nreg = nag_int(2);

% The series
yt = [7.53; 6.64; 7.39; 7.15; 6.42; 6.32; 6.98; 7.09; 6.63; 6.93;
      7.01; 5.30; 7.86; 6.73; 7.39; 5.61; 7.02; 6.04; 7.46; 4.33;
      6.02; 6.37; 3.93; 7.24; 8.58; 5.70; 9.13; 7.99; 7.79; 6.13;
      8.78; 6.52; 6.79; 7.77; 7.31; 7.58; 8.78; 7.39; 8.00; 7.07;
      7.65; 9.15; 8.32; 7.32; 7.58; 9.78; 8.17; 9.26; 7.79; 7.03;
      7.45; 7.09; 8.06; 7.06; 9.91; 7.01; 8.32; 6.41; 8.59; 8.55;
      7.77; 8.04; 9.54; 8.28; 7.97; 8.42; 8.30; 7.98; 7.60; 8.77;
      7.54; 7.40; 9.26; 7.30; 9.33; 9.54; 8.08; 6.93; 4.27; 2.65;
      5.03; 0.91;12.63;10.87; 9.26; 8.30; 6.85; 7.48; 9.67; 9.54;
      7.33; 8.84; 7.75; 8.12; 7.29; 8.58; 7.80; 3.07; 9.33;16.91];
```

```
% The exogenous variables
x = [ 2.40, 0.12; 2.40, 0.12; 2.40, 0.13; 2.40, 0.14;
      2.40, 0.14; 2.40, 0.15; 2.40, 0.16; 2.40, 0.16;
      2.40, 0.17; 2.41, 0.18; 2.41, 0.19; 2.41, 0.19;
      2.41, 0.20; 2.41, 0.21; 2.41, 0.21; 2.41, 0.22;
      2.41, 0.23; 2.41, 0.23; 2.41, 0.24; 2.42, 0.25;
      2.42, 0.25; 2.42, 0.26; 2.42, 0.26; 2.42, 0.27;
      2.42, 0.28; 2.42, 0.28; 2.42, 0.29; 2.42, 0.30;
      2.42, 0.30; 2.43, 0.31; 2.43, 0.32; 2.43, 0.32;
      2.43, 0.33; 2.43, 0.33; 2.43, 0.34; 2.43, 0.35;
      2.43, 0.35; 2.43, 0.36; 2.43, 0.37; 2.44, 0.37;
      2.44, 0.38; 2.44, 0.38; 2.44, 0.39; 2.44, 0.39;
      2.44, 0.40; 2.44, 0.41; 2.44, 0.41; 2.44, 0.42;
      2.44, 0.42; 2.45, 0.43; 2.45, 0.43; 2.45, 0.44;
      2.45, 0.45; 2.45, 0.45; 2.45, 0.46; 2.45, 0.46;
      2.45, 0.47; 2.45, 0.47; 2.45, 0.48; 2.46, 0.48;
      2.46, 0.49; 2.46, 0.49; 2.46, 0.50; 2.46, 0.50;
      2.46, 0.51; 2.46, 0.51; 2.46, 0.52; 2.46, 0.52;
      2.46, 0.53; 2.47, 0.53; 2.47, 0.54; 2.47, 0.54;
      2.47, 0.54; 2.47, 0.55; 2.47, 0.55; 2.47, 0.56;
      2.47, 0.56; 2.47, 0.57; 2.47, 0.57; 2.48, 0.57;
      2.48, 0.58; 2.48, 0.58; 2.48, 0.59; 2.48, 0.59;
      2.48, 0.59; 2.48, 0.60; 2.48, 0.60; 2.48, 0.61;
      2.48, 0.61; 2.49, 0.61; 2.49, 0.62; 2.49, 0.62;
      2.49, 0.62; 2.49, 0.63; 2.49, 0.63; 2.49, 0.63;
      2.49, 0.64; 2.49, 0.64; 2.49, 0.64; 2.50, 0.64];

% Details of the model to fit
dist = 't';
ip = nag_int(1);
iq = nag_int(2);

% Control parameters
copts = true;
maxit = nag_int(200);
tol = 0.00001;

% Initial parameter estimates
theta = [0.05; -0.15; -0.05; 0.05; 0.15; 0.35; 3.25; 1.5; 0; 0];

% Forecast horizon
nt = nag_int(4);

% Fit the GARCH model
[theta, se, sc, covar, hp, et, ht, lgf, ifail] = ...
  g13fg( ...
         dist, yt, x, ip, iq, mn, theta, 0, ...
         copts, maxit, tol, 'ifail', nag_int(-1));

% Calculate the volatility forecast
[fht, ifail] = g13fh( ...
                      nt, ip, iq, theta, ht, et);

% Output the results
fprintf('\n              Parameter        Standard\n');
fprintf('               estimates        errors\n');

% Output the coefficient alpha_0
fprintf('Alpha0 %16.2f%16.2f\n', theta(1), se(1));
l = 2;

% Output the coefficients alpha_i
for i = l:l+iq-1
  fprintf('Alpha%d %16.2f%16.2f\n', i-1, theta(i), se(i));
end
l = l+iq;

% Output the coefficients psi_i
for i = l:l+iq-1
  fprintf('  Psi%d %16.2f%16.2f\n', i-l+1, theta(i), se(i));
```

```
end
l = l+iq;

% Output the coefficients beta_j
fprintf('\n');
for i = l:l+ip-1
  fprintf(' Beta%d %16.2f%16.2f\n', i-l+1, theta(i), se(i));
end
l = l+ip;

% Output the estimated degrees of freedom, df
if (dist == 't')
  fprintf('\n    DF %16.2f%16.2f\n', theta(l), se(l));
  l = l + 1;
end

% Output the estimated mean term, b_0
if (mn == 1)
  fprintf('\n    B0 %16.2f%16.2f\n', theta(l), se(l));
  l = l + 1;
end

% Output the estimated linear regression coefficients, b_i
for i = l:l+nreg-1
  fprintf('    B%d %16.2f%16.2f\n', i-l+1, theta(i), se(i));
end

% Display the volatility forecast
fprintf('\nVolatility forecast = %12.4f\n', fht(nt));
```

9.2 Program Results

```
     g13fg example results

               Parameter        Standard
               estimates         errors
Alpha0            0.17            0.19
Alpha1           -0.65            0.23
Alpha2           -0.44            0.24
  Psi1           -0.06            0.22
  Psi2            0.35            0.25

 Beta1            0.42            0.17

    DF            5.59            3.75

    B0          128.75           42.09
    B1          -51.74           17.78
    B2           13.01            3.40

Volatility forecast =       1.3404
```