

NAG Toolbox

nag_tsa_uni_garch_exp_forecast (g13fh)

1 Purpose

nag_tsa_uni_garch_exp_forecast (g13fh) forecasts the conditional variances, $h_t, t = T + 1, \dots, T + \xi$ from an exponential GARCH(p, q) sequence, where ξ is the forecast horizon and T is the current time (see Engle and Ng (1993)).

2 Syntax

```
[fht, ifail] = nag_tsa_uni_garch_exp_forecast(nt, ip, iq, theta, ht, et, 'num',
num)
[fht, ifail] = g13fh(nt, ip, iq, theta, ht, et, 'num', num)
```

3 Description

Assume the GARCH(p, q) process represented by:

$$\ln(h_t) = \alpha_0 + \sum_{i=1}^q \alpha_i z_{t-i} + \sum_{j=1}^q \phi_j (|z_{t-j}| - E[|z_{t-j}|]) + \sum_{j=1}^p \beta_j \ln(h_{t-j}), \quad t = 1, 2, \dots, T.$$

where $\epsilon_t | \psi_{t-1} = N(0, h_t)$ or $\epsilon_t | \psi_{t-1} = S_t(df, h_t)$, and $z_t = \frac{\epsilon_t}{\sqrt{h_t}}$, $E[|z_{t-i}|]$ denotes the expected value of $|z_{t-i}|$, has been modelled by nag_tsa_uni_garch_exp_estim (g13fg), and the estimated conditional variances and residuals are contained in the arrays **ht** and **et** respectively.

nag_tsa_uni_garch_exp_forecast (g13fh) will then use the last $\max(p, q)$ elements of the arrays **ht** and **et** to estimate the conditional variance forecasts, $h_t | \psi_T$, where $t = T + 1, \dots, T + \xi$ and ξ is the forecast horizon.

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: **nt** – INTEGER

ξ , the forecast horizon.

Constraint: **nt** > 0.

2: **ip** – INTEGER

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

Constraints:

$$\begin{aligned} \max(\mathbf{ip}, \mathbf{iq}) &\leq 20; \\ \mathbf{ip} &\geq 0. \end{aligned}$$

3: **iq** – INTEGER

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

Constraints:

$$\begin{aligned} \max(\mathbf{ip}, \mathbf{iq}) &\leq 20; \\ \mathbf{iq} &\geq 1. \end{aligned}$$

4: **theta(2 × iq + ip + 1)** – REAL (KIND=nag_wp) array

The initial parameter estimates for the vector θ . The first element must contain the coefficient α_0 and the next **iq** elements must contain the autoregressive coefficients α_i , for $i = 1, 2, \dots, q$. The next **iq** elements must contain the coefficients ϕ_i , for $i = 1, 2, \dots, q$. The next **ip** elements must contain the moving average coefficients β_j , for $j = 1, 2, \dots, p$.

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

The sequence of past conditional variances for the GARCH(p, q) process, h_t , for $t = 1, 2, \dots, T$.

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

The sequence of past residuals for the GARCH(p, q) process, ϵ_t , for $t = 1, 2, \dots, T$.

5.2 Optional Input Parameters

1: **num** – INTEGER

Default: the dimension of the arrays **ht**, **et**. (An error is raised if these dimensions are not equal.)

The number of terms in the arrays **ht** and **et** from the modelled sequence.

Constraint: $\max(\mathbf{ip}, \mathbf{iq}) \leq \mathbf{num}$.

5.3 Output Parameters

1: **fht(nt)** – REAL (KIND=nag_wp) array

The forecast values of the conditional variance, h_t , for $t = T + 1, \dots, T + \xi$.

2: **ifail** – INTEGER

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

6 Error Indicators and Warnings

Errors or warnings detected by the function:

ifail = 1

On entry, **num** < $\max(\mathbf{ip}, \mathbf{iq})$,
or **iq** < 1,
or **ip** < 0,
or $\max(\mathbf{ip}, \mathbf{iq}) > 20$,
or **nt** ≤ 0.

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

See Section 10 in nag_tsa_uni_garch_exp_estim (g13fg).

9.1 Program Text

```
function g13fh_example

fprintf('g13fh 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 = zeros(num, nreg+mn);
x(:, 1:nreg) = [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, 'nreg', nreg);

% 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 = 1: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 = 1:l+iq-1
    fprintf('Psi%d %16.2f%16.2f\n', i-1+1, theta(i), se(i));
end
l = l+iq;

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

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

```

9.2 Program Results

g13fh example results

	Parameter estimates	Standard 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	
