

NAG Toolbox for MATLAB

g13fe

1 Purpose

g13fe estimates the parameters of a univariate regression-GJR GARCH(p, q) process (see Glosten *et al.* 1993).

2 Syntax

```
[theta, se, sc, covr, hp, et, ht, lgf, ifail] = g13fe(dist, yt, x, ip,
iq, nreg, mn, theta, hp, copts, maxit, tol, 'num', num, 'npar', npar)
```

3 Description

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

$$y_t = b_o + x_t^T b + \epsilon_t \quad (1)$$

$$h_t = \alpha_0 + \sum_{i=1}^q (\alpha_i + \gamma S_{t-i}) \epsilon_{t-i}^2 + \sum_{i=1}^p \beta_i h_{t-i}, \quad t = 1, \dots, T \quad (2)$$

where $S_t = 1$, if $\epsilon_t < 0$, $S_t = 0$, if $\epsilon_t \geq 0$, and $\epsilon_t | \psi_{t-1} = N(0, h_t)$ or $\epsilon_t | \psi_{t-1} = S_t(df, h_t)$. 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 is the conditional variance, and ψ_t the set of all information up to time t .

g13fe provides an estimate for $\hat{\theta}$, the parameter vector $\theta = (b_o, b^T, \omega^T)$ where $b^T = (b_1, \dots, b_k)$, $\omega^T = (\alpha_0, \alpha_1, \dots, \alpha_q, \beta_1, \dots, \beta_p, \gamma)$ when **dist** = 'N' and $\omega^T = (\alpha_0, \alpha_1, \dots, \alpha_q, \beta_1, \dots, \beta_p, \gamma, df)$ when **dist** = 'T'.

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

No Regression and No Mean

$$y_t = \epsilon_t,$$

$$\mathbf{mn} = 0,$$

$$\mathbf{nreg} = 0 \text{ and}$$

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

No Regression

$$y_t = b_o + \epsilon_t,$$

$$\mathbf{mn} = 1,$$

$$\mathbf{nreg} = 0 \text{ and}$$

θ is a $(p + q + 3)$ vector when **dist** = 'N', and a $(p + q + 4)$ vector when **dist** = 'T'.

Note: if the $y_t = \mu + \epsilon_t$, where μ is known (not to be estimated by g13fe) 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

$$y_t = x_t^T b + \epsilon_t,$$

$$\mathbf{mn} = 0,$$

nreg = k and

θ is a $(p + q + k + 2)$ vector when **dist** = 'N', and a $(p + q + k + 3)$ 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** – string

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)** – double array

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

3: **x(ldx,*)** – double array

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

The second dimension of the array must be at least $\max(1, \mathbf{nreg} + \mathbf{mn})$

Row t of **x** must contain the time dependent exogenous vector x_t , where $x_t^T = (x_t^1, \dots, x_t^k)$, for $t = 1, \dots, T$.

4: **ip** – int32 scalar

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

Constraint: **ip** ≥ 0 (see also **npar**).

5: **iq** – int32 scalar

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

Constraint: **iq** ≥ 1 (see also **npar**).

6: **nreg** – int32 scalar

k , the number of regression coefficients.

Constraint: **nreg** ≥ 0 (see also **npar**).

7: **mn – int32 scalar**

If **mn** = 1, the mean term b_0 will be included in the model.

Constraint: **mn** = 0 or 1.

8: **theta(np) – double array**

The initial parameter estimates for the vector θ .

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

The next **ip** elements must contain the coefficients β_j , for $j = 1, \dots, p$.

The next element must contain the asymmetry parameter γ .

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

If **mn** = 1, the next element must contain the mean term b_o .

If **copts(2)** = **false**, the remaining **nreg** elements are taken as initial estimates of the linear regression coefficients b_i , for $i = 1, \dots, k$.

9: **hp – double scalar**

If **copts(2)** = **false**, **hp** is the value to be used for the pre-observed conditional variance, otherwise **hp** is not referenced.

10: **copts(2) – logical array**

The options to be used by g13fe.

copts(1) = **true**

Stationary conditions are enforced, otherwise they are not.

copts(2) = **true**

The function provides initial parameter estimates of the regression terms, otherwise these are to be provided by you.

11: **maxit – int32 scalar**

The maximum number of iterations to be used by the optimization function when estimating the GARCH(p, q) parameters. If **maxit** is set to 0 then the standard errors, score vector and variance-covariance are calculated for the input value of θ in **theta**; however the value of θ is not updated.

Constraint: **maxit** ≥ 0 .

12: **tol – double scalar**

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

5.2 Optional Input Parameters

1: **num – int32 scalar**

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

T , the number of terms in the sequence.

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

2: **npar – int32 scalar**

Default: The dimension of the arrays **theta**, **se**, **sc**, **covar**. (An error is raised if these dimensions are not equal.)

the number of parameters to be included in the model. **npar** = 2 + **iq** + **ip** + **mn** + **nreg** when **dist** = 'N' and **npar** = 3 + **iq** + **ip** + **mn** + **nreg** when **dist** = 'T'.

Constraint: **npar** < 20.

5.3 Input Parameters Omitted from the MATLAB Interface

ldx, ldcovr, work, lwork

5.4 Output Parameters1: **theta(npar) – double array**

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

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

The next **ip** elements are the moving average coefficients β_j , for $j = 1, \dots, p$.

The next element contains the estimate for the asymmetry parameter γ .

If **dist** = 'T', the next element contains an estimate for df , 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 $i = 1, \dots, k$.

2: **se(npar) – double 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 $i = 1, \dots, q$.

The next **ip** elements are the standard errors for β_j , for $j = 1, \dots, p$.

The next element contains the standard error for γ .

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 $j = 1, \dots, k$.

3: **sc(npar) – double array**

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

The first element contains the score for α_o , the next **iq** elements contain the scores for α_i , for $i = 1, \dots, q$.

The next **ip** elements are the score for β_j , for $j = 1, \dots, p$. The next element contains the score for γ .

If **dist** = 'T', the next element contains the score 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 $j = 1, \dots, k$.

4: **covr(ldcovr,npar) – double array**

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

5: **hp – double scalar**

If **copts(2) = true**, **hp** is the estimated value of the pre-observed of the conditional variance.

6: **et(num) – double array**

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

7: **ht(num) – double array**

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

8: **lgf – double scalar**

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

9: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

6 Error Indicators and Warnings

Note: g13fe may return useful information for one or more of the following detected errors or warnings.

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** ≠ 'N',
 or **dist** ≠ 'T',
 or **maxit** < 0,
 or **num** < max(**ip**, **iq**).

ifail = 2

On entry, **lwork** < (**nreg** + 3) × **num** + **npar** + 403.

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.

7 Accuracy

Not applicable.

8 Further Comments

None.

9 Example

```

iseed = zeros(4, 1, 'int32');
iseed(1) = int32(111);
igen = int32(0);
rwsav = zeros(9, 1);
rvec = zeros(40, 1);
dist = 'N';
num = int32(2000);
param = [0.4; 0.1; 0.7];
bx = [1.5; 2.5; 3];
theta = [0.2; 0.05; 0.35; 0.05; 2; 0.75; 1.25];
gamma = 0.1;
df = 5.1;
ip = int32(1);
iq = int32(1);
nreg = int32(2);
mn = int32(1);
hp = 0;
mean = 4;
copts = [true; true];
maxit = int32(100);
tol = 1e-05;
x = zeros(2000, 3);
for i = 1:num
    fac1 = double(i)*0.01;
    x(i,2) = 0.01 + 0.7*sin(fac1);
    x(i,1) = 0.5 + fac1*0.1;
    x(i,3) = 1.0;
end
[igen, iseed] = g05kb(igen, iseed);
fcall = true;
[ht, yt, rvec, iseed, rwsav, ifail] = ...
    g05hm(dist, int32(200), ip, iq, param, gamma, df, fcall, rvec, igen,
iseed, rwsav);
fcall = false;
[ht, yt, rvec, iseed, rwsav, ifail] = ...
    g05hm(dist, num, ip, iq, param, gamma, df, fcall, rvec, igen, iseed,
rwsav);

for i = 1:num
    xterm = 0;
    for k = 1:nreg
        xterm = xterm + x(i,k)*bx(k);
    end
    if (mn == 1)
        yt(i) = mean + xterm + yt(i);
    else
        yt(i) = xterm + yt(i);
    end
end

```

```

    end
end

[thetaOut, se, sc, covar, hpOut, et, ht, lgf, ifail] = ...
    g13fe(dist, yt, x, ip, iq, nreg, mn, theta, hp, copts, maxit, tol)

thetaOut =
    0.3707
    0.1036
    0.7079
    0.1188
    4.0998
    1.4251
    2.2604
se =
    0.0780
    0.0257
    0.0414
    0.0370
    0.0950
    0.0592
    0.0683
sc =
    0.2470
    0.6943
    0.5456
    0.0522
    0.1537
    0.1608
    -0.1966
covar =
    0.0061    0.0004   -0.0028    0.0007         0         0         0
    0.0004    0.0007   -0.0005   -0.0004         0         0         0
   -0.0028   -0.0005    0.0017   -0.0005         0         0         0
    0.0007   -0.0004   -0.0005    0.0014         0         0         0
         0         0         0         0    0.0090   -0.0053   -0.0012
         0         0         0         0   -0.0053    0.0035    0.0007
         0         0         0         0   -0.0012    0.0007    0.0047
hpOut =
    2.9154
et =
    array elided
ht =
    array elided
lgf =
   -1.9696e+03
ifail =
    0

```