NAG Library Routine Document

G02ECF

Note: before using this routine, please read the Users' Note for your implementation to check the interpretation of ***bold italicised*** terms and other implementation-dependent details.

1 Purpose

G02ECF calculates R^2 and C_p-values from the residual sums of squares for a series of linear regression models.

2 Specification

```
SUBROUTINE G02ECF (MEAN, N, SIGSQ, TSS, NMOD, NTERMS, RSS, RSQ, CP,        &
                   IFAIL)

INTEGER             N, NMOD, NTERMS(NMOD), IFAIL
REAL (KIND=nag_wp) SIGSQ, TSS, RSS(NMOD), RSQ(NMOD), CP(NMOD)
CHARACTER(1)        MEAN
```

3 Description

When selecting a linear regression model for a set of n observations a balance has to be found between the number of independent variables in the model and fit as measured by the residual sum of squares. The more variables included the smaller will be the residual sum of squares. Two statistics can help in selecting the best model.

(a) R^2 represents the proportion of variation in the dependent variable that is explained by the independent variables.

$$R^2 = \frac{\text{Regression Sum of Squares}}{\text{Total Sum of Squares}},$$

where Total Sum of Squares $=$ TSS $= \sum (y - \bar{y})^2$ (if mean is fitted, otherwise TSS $= \sum y^2$) and

Regression Sum of Squares $=$ RegSS $=$ TSS $-$ RSS, where
RSS $=$ residual sum of squares $= \sum (y - \hat{y})^2$.

The R^2-values can be examined to find a model with a high R^2-value but with small number of independent variables.

(b) C_p statistic.

$$C_p = \frac{\text{RSS}}{\hat{\sigma}^2} - (n - 2p),$$

where p is the number of arguments (including the mean) in the model and $\hat{\sigma}^2$ is an estimate of the true variance of the errors. This can often be obtained from fitting the full model.

A well fitting model will have $C_p \simeq p$. C_p is often plotted against p to see which models are closest to the $C_p = p$ line.

G02ECF may be called after G02EAF which calculates the residual sums of squares for all possible linear regression models.

4 References

Draper N R and Smith H (1985) *Applied Regression Analysis* (2nd Edition) Wiley

Weisberg S (1985) *Applied Linear Regression* Wiley

5 Arguments

1: MEAN – CHARACTER(1) *Input*

On entry: indicates if a mean term is to be included.

MEAN = 'M'
A mean term, intercept, will be included in the model.

MEAN = 'Z'
The model will pass through the origin, zero-point.

Constraint: MEAN = 'M' or 'Z'.

2: N – INTEGER *Input*

On entry: n, the number of observations used in the regression model.

Constraint: N must be greater than $2 \times p_{\max}$, where $p_{\max}$ is the largest number of independent variables fitted (including the mean if fitted).

3: SIGSQ – REAL (KIND=nag_wp) *Input*

On entry: the best estimate of true variance of the errors, $\hat{\sigma}^2$.

Constraint: SIGSQ > 0.0.

4: TSS – REAL (KIND=nag_wp) *Input*

On entry: the total sum of squares for the regression model.

Constraint: TSS > 0.0.

5: NMOD – INTEGER *Input*

On entry: the number of regression models.

Constraint: NMOD > 0.

6: NTERMS(NMOD) – INTEGER array *Input*

On entry: NTERMS(i) must contain the number of independent variables (not counting the mean) fitted to the ith model, for $i = 1, 2, \ldots,$ NMOD.

7: RSS(NMOD) – REAL (KIND=nag_wp) array *Input*

On entry: RSS(i) must contain the residual sum of squares for the ith model.

Constraint: RSS(i) $\leq$ TSS, for $i = 1, 2, \ldots,$ NMOD.

8: RSQ(NMOD) – REAL (KIND=nag_wp) array *Output*

On exit: RSQ(i) contains the R^2-value for the ith model, for $i = 1, 2, \ldots,$ NMOD.

9: CP(NMOD) – REAL (KIND=nag_wp) array *Output*

On exit: CP(i) contains the C_p-value for the ith model, for $i = 1, 2, \ldots,$ NMOD.

10: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this argument you should refer to Section 3.4 in How to Use the NAG Library and its Documentation for details.

For environments where it might be inappropriate to halt program execution when an error is detected, the value -1 or 1 is recommended. If the output of error messages is undesirable, then the value 1 is recommended. Otherwise, if you are not familiar with this argument, the

recommended value is 0. **When the value -1 or 1 is used it is essential to test the value of IFAIL on exit.**

On exit: IFAIL $= 0$ unless the routine detects an error or a warning has been flagged (see Section 6).

6 Error Indicators and Warnings

If on entry IFAIL $= 0$ or -1, explanatory error messages are output on the current error message unit (as defined by X04AAF).

Errors or warnings detected by the routine:

IFAIL $= 1$

On entry, NMOD < 1,
or SIGSQ ≤ 0.0,
or TSS ≤ 0.0.
or MEAN $\neq$ 'M' or 'Z'.

IFAIL $= 2$

On entry, the number of arguments for a model is too large for the number of observations, i.e., $2 \times p \geq n$.

IFAIL $= 3$

On entry, RSS$(i) >$ TSS, for some $i = 1, 2, \ldots,$ NMOD.

IFAIL $= 4$

A value of C_p is less than 0.0. This may occur if SIGSQ is too large or if RSS, N or IP are incorrect.

IFAIL $= -99$

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

See Section 3.9 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -399$

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

See Section 3.8 in How to Use the NAG Library and its Documentation for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.7 in How to Use the NAG Library and its Documentation for further information.

7 Accuracy

Accuracy is sufficient for all practical purposes.

8 Parallelism and Performance

G02ECF is not threaded in any implementation.

9 Further Comments

None.

10 Example

The data, from an oxygen uptake experiment, is given by Weisberg (1985). The independent and dependent variables are read and the residual sums of squares for all possible models computed using G02EAF. The values of R^2 and C_p are then computed and printed along with the names of variables in the models.

10.1 Program Text

```
    Program g02ecfe

!     G02ECF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g02eaf, g02ecf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6, vnlen = 3
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: sigsq, tss
      Integer                          :: i, ifail, k, ldmodl, ldx, lwt, m, n, &
                                          nmod
      Character (1)                    :: mean, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: cp(:), rsq(:), rss(:), wk(:), wt(:), &
                                          x(:,:), y(:)
      Integer, Allocatable             :: isx(:), mrank(:), nterms(:)
      Character (vnlen), Allocatable   :: modl(:,:), vname(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count, max, real
!     .. Executable Statements ..
      Write (nout,*) 'G02ECF Example Program Results'
      Write (nout,*)

!     Skip heading in data file
      Read (nin,*)

!     Read in the problem size
      Read (nin,*) n, m, mean, weight

      If (weight=='W' .Or. weight=='w') Then
        lwt = n
      Else
        lwt = 0
      End If
      ldx = n
      Allocate (x(ldx,m),wt(lwt),y(n),isx(m),vname(m))

!     Read in data
      If (lwt>0) Then
        Read (nin,*)(x(i,1:m),y(i),wt(i),i=1,n)
      Else
        Read (nin,*)(x(i,1:m),y(i),i=1,n)
      End If

!     Read in variable inclusion flags
      Read (nin,*) isx(1:m)

!     Read in first VNLEN characters of the variable names
      Read (nin,*) vname(1:m)

!     Calculate the number of free variables
      k = count(isx(1:m)==1)

      ldmodl = max(m,2**k)
      Allocate (modl(ldmodl,m),rss(ldmodl),nterms(ldmodl),mrank(ldmodl),wk(n*( &
```

```
        m+1)))

!     Calculate residual sums of squares
      ifail = 0
      Call g02eaf(mean,weight,n,m,x,ldx,vname,isx,y,wt,nmod,modl,ldmodl,rss,   &
        nterms,mrank,wk,ifail)

!     Extract total sums of squares
      tss = rss(1)

!     Calculate best estimate of true variance from full model
      sigsq = rss(nmod)/real(n-nterms(nmod)-1,kind=nag_wp)

      Allocate (rsq(nmod),cp(nmod))

!     Calculate R-squared and Mallows Cp
      ifail = 0
      Call g02ecf('M',n,sigsq,tss,nmod,nterms,rss,rsq,cp,ifail)

!     Display results
      Write (nout,*) 'Number of     CP      RSQ         MODEL'
      Write (nout,*) 'parameters'
      Write (nout,*)
      Do i = 1, nmod
        Write (nout,99999) nterms(i), cp(i), rsq(i), modl(i,1:nterms(i))
      End Do

99999 Format (1X,I7,F11.2,F8.4,1X,5(1X,A))
    End Program g02ecfe
```

10.2 Program Data

```
G02ECF Example Program Data
  20 6 'M' 'U'                                      :: N,M,MEAN,WEIGHT
   0. 1125.0 232.0 7160.0 85.9 8905.0  1.5563
   7.  920.0 268.0 8804.0 86.5 7388.0  0.8976
  15.  835.0 271.0 8108.0 85.2 5348.0  0.7482
  22. 1000.0 237.0 6370.0 83.8 8056.0  0.7160
  29. 1150.0 192.0 6441.0 82.1 6960.0  0.3010
  37.  990.0 202.0 5154.0 79.2 5690.0  0.3617
  44.  840.0 184.0 5896.0 81.2 6932.0  0.1139
  58.  650.0 200.0 5336.0 80.6 5400.0  0.1139
  65.  640.0 180.0 5041.0 78.4 3177.0 -0.2218
  72.  583.0 165.0 5012.0 79.3 4461.0 -0.1549
  80.  570.0 151.0 4825.0 78.7 3901.0  0.0000
  86.  570.0 171.0 4391.0 78.0 5002.0  0.0000
  93.  510.0 243.0 4320.0 72.3 4665.0 -0.0969
 100.  555.0 147.0 3709.0 74.9 4642.0 -0.2218
 107.  460.0 286.0 3969.0 74.4 4840.0 -0.3979
 122.  275.0 198.0 3558.0 72.5 4479.0 -0.1549
 129.  510.0 196.0 4361.0 57.7 4200.0 -0.2218
 151.  165.0 210.0 3301.0 71.8 3410.0 -0.3979
 171.  244.0 327.0 2964.0 72.5 3360.0 -0.5229
 220.   79.0 334.0 2777.0 71.9 2599.0 -0.0458  :: End of X,Y
  0     1    1      1     1    1               :: ISX
'DAY' 'BOD' 'TKN' 'TS ' 'TVS' 'COD'            :: VNAME
```

10.3 Program Results

```
 G02ECF Example Program Results

 Number of     CP      RSQ         MODEL
 parameters

       0      55.45  0.0000
       1      56.84  0.0082  TKN
       1      20.33  0.5054  TVS
       1      13.50  0.5983  BOD
       1       6.57  0.6926  COD
       1       6.29  0.6965  TS
```

```
 2      21.36  0.5185  TKN TVS
 2      11.33  0.6551  BOD TVS
 2       9.09  0.6856  BOD TKN
 2       7.70  0.7045  BOD COD
 2       7.33  0.7095  TKN TS
 2       7.16  0.7119  TS  TVS
 2       6.88  0.7157  BOD TS
 2       6.87  0.7158  TKN COD
 2       5.27  0.7376  TVS COD
 2       1.74  0.7857  TS  COD
 3       8.68  0.7184  BOD TKN TVS
 3       8.16  0.7255  TKN TS  TVS
 3       8.15  0.7256  BOD TS  TVS
 3       7.15  0.7392  BOD TVS COD
 3       6.51  0.7479  BOD TKN COD
 3       6.25  0.7515  BOD TKN TS
 3       5.67  0.7595  TKN TVS COD
 3       3.44  0.7898  BOD TS  COD
 3       3.42  0.7900  TS  TVS COD
 3       2.32  0.8050  TKN TS  COD
 4       7.70  0.7591  BOD TKN TS  TVS
 4       6.78  0.7716  BOD TKN TVS COD
 4       5.07  0.7948  BOD TS  TVS COD
 4       4.32  0.8050  BOD TKN TS  COD
 4       4.00  0.8094  TKN TS  TVS COD
 5       6.00  0.8094  BOD TKN TS  TVS COD
```