NAG Library Routine Document

G04EAF

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

G04EAF computes orthogonal polynomial or dummy variables for a factor or classification variable.

2 Specification

```
SUBROUTINE G04EAF (TYP, N, LEVELS, IFACT, X, LDX, V, REP, IFAIL)

INTEGER              N, LEVELS, IFACT(N), LDX, IFAIL
REAL (KIND=nag_wp) X(LDX,*), V(*), REP(LEVELS)
CHARACTER(1)         TYP
```

3 Description

In the analysis of an experimental design using a general linear model the factors or classification variables that specify the design have to be coded as dummy variables. G04EAF computes dummy variables that can then be used in the fitting of the general linear model using G02DAF.

If the factor of length n has k levels then the simplest representation is to define k dummy variables, X_j such that $X_j = 1$ if the factor is at level j and 0 otherwise for $j = 1, 2, \ldots, k$. However, there is usually a mean included in the model and the sum of the dummy variables will be aliased with the mean. To avoid the extra redundant argument $k - 1$ dummy variables can be defined as the contrasts between one level of the factor, the reference level, and the remaining levels. If the reference level is the first level then the dummy variables can be defined as $X_j = 1$ if the factor is at level j and 0 otherwise, for $j = 2, 3, \ldots, k$. Alternatively, the last level can be used as the reference level.

A second way of defining the $k - 1$ dummy variables is to use a Helmert matrix in which levels $2, 3, \ldots, k$ are compared with the average effect of the previous levels. For example if $k = 4$ then the contrasts would be:

$$\begin{array}{cccc} 1 & -1 & -1 & -1 \\ 2 & 1 & -1 & -1 \\ 3 & 0 & 2 & -1 \\ 4 & 0 & 0 & 3 \end{array}$$

Thus variable j, for $j = 1, 2, \ldots, k - 1$ is given by

$X_j = -1$ if factor is at level less than $j + 1$

$X_j = \sum_{i=1}^{j} r_i / r_{j+1}$ if factor is at level $j + 1$

$X_j = 0$ if factor is at level greater than $j + 1$

where r_j is the number of replicates of level j.

If the factor can be considered as a set of values from an underlying continuous variable then the factor can be represented by a set of $k - 1$ orthogonal polynomials representing the linear, quadratic etc. effects of the underlying variable. The orthogonal polynomial is computed using Forsythe's algorithm (Forsythe (1957), see also Cooper (1968)). The values of the underlying continuous variable represented by the factor levels have to be supplied to the routine.

The orthogonal polynomials are standardized so that the sum of squares for each dummy variable is one. For the other methods integer (± 1) representations are retained except that in the Helmert representation the code of level $j + 1$ in dummy variable j will be a fraction.

4 References

Cooper B E (1968) Algorithm AS 10. The use of orthogonal polynomials *Appl. Statist.* **17** 283–287

Forsythe G E (1957) Generation and use of orthogonal polynomials for data fitting with a digital computer *J. Soc. Indust. Appl. Math.* **5** 74–88

5 Arguments

1: TYP – CHARACTER(1) *Input*

On entry: the type of dummy variable to be computed.

If TYP = 'P', an orthogonal Polynomial representation is computed.

If TYP = 'H', a Helmert matrix representation is computed.

If TYP = 'F', the contrasts relative to the First level are computed.

If TYP = 'L', the contrasts relative to the Last level are computed.

If TYP = 'C', a Complete set of dummy variables is computed.

Constraint: TYP = 'P', 'H', 'F', 'L' or 'C'.

2: N – INTEGER *Input*

On entry: n, the number of observations for which the dummy variables are to be computed.

Constraint: N $\geq$ LEVELS.

3: LEVELS – INTEGER *Input*

On entry: k, the number of levels of the factor.

Constraint: LEVELS ≥ 2.

4: IFACT(N) – INTEGER array *Input*

On entry: the n values of the factor.

Constraint: $1 \leq$ IFACT$(i) \leq$ LEVELS, for $i = 1, 2, \ldots, n$.

5: X(LDX, $*$) – REAL (KIND=nag_wp) array *Output*

Note: the second dimension of the array X must be at least LEVELS $-$ 1 if TYP = 'P', 'H', 'F' or 'L' and at least LEVELS if TYP = 'C'.

On exit: the n by k^* matrix of dummy variables, where $k^* = k - 1$ if TYP = 'P', 'H', 'F' or 'L' and $k^* = k$ if TYP = 'C'.

6: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G04EAF is called.

Constraint: LDX $\geq$ N.

7: V($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array V must be at least LEVELS if TYP = 'P', and at least 1 otherwise.

On entry: if TYP = 'P', the k distinct values of the underlying variable for which the orthogonal polynomial is to be computed.

If TYP $\neq$ 'P', V is not referenced.

Constraint: if TYP = 'P', the k values of V must be distinct.

8: REP(LEVELS) – REAL (KIND=nag_wp) array *Output*

On exit: the number of replications for each level of the factor, r_i, for $i = 1, 2, \ldots, k$.

9: 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, LEVELS < 2,
or N $<$ LEVELS,
or LDX $<$ N,
or TYP $\neq$ 'P', 'H', 'F', 'L' or 'C'.

IFAIL $= 2$

On entry, a value of IFACT is not in the range $1 \leq \text{IFACT}(i) \leq \text{LEVELS}$, for $i = 1, 2, \ldots, n$,
or TYP = 'P' and not all values of V are distinct,
or not all levels are represented in IFACT.

IFAIL $= 3$

An orthogonal polynomial has all values zero. This will be due to some values of V being very close together. Note this can only occur if TYP = 'P'.

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

The computations are stable.

8 Parallelism and Performance

G04EAF is threaded by NAG for parallel execution in multithreaded implementations of the NAG Library.

G04EAF makes calls to BLAS and/or LAPACK routines, which may be threaded within the vendor library used by this implementation. Consult the documentation for the vendor library for further information.

Please consult the X06 Chapter Introduction for information on how to control and interrogate the OpenMP environment used within this routine. Please also consult the Users' Note for your implementation for any additional implementation-specific information.

9 Further Comments

Other routines for fitting polynomials can be found in Chapter E02.

10 Example

Data are read in from an experiment with four treatments and three observations per treatment with the treatment coded as a factor. G04EAF is used to compute the required dummy variables and the model is then fitted by G02DAF.

10.1 Program Text

```
    Program g04eafe

!     G04EAF Example Program Text

!     Mark 26 Release. NAG Copyright 2016.

!     .. Use Statements ..
      Use nag_library, Only: g02daf, g04eaf, nag_wp
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Real (Kind=nag_wp)               :: rss, tol
      Integer                          :: i, idf, ifail, ip, irank, j, ldq,    &
                                          ldx, levels, lv, lwt, m, n, tdx
      Logical                          :: svd
      Character (1)                    :: mean, typ, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: b(:), cov(:), h(:), p(:), q(:,:),    &
                                          rep(:), res(:), se(:), v(:), wk(:),  &
                                          wt(:), x(:,:), y(:)
      Integer, Allocatable             :: ifact(:), isx(:)
!     .. Executable Statements ..
      Write (nout,*) 'G04EAF Example Program Results'
      Write (nout,*)

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

!     Read in problem information
      Read (nin,*) n, levels, typ, weight, mean

      If (typ=='P' .Or. typ=='p') Then
        lv = levels
      Else
        lv = 1
      End If
      If (typ=='C' .Or. typ=='c') Then
        tdx = levels
      Else
```

```
        tdx = levels - 1
      End If
      If (weight=='w' .Or. weight=='W') Then
        lwt = n
      Else
        lwt = 1
      End If
      ldx = n
      Allocate (x(ldx,tdx),ifact(n),v(lv),rep(levels),y(n),wt(lwt))

!     Read in data
      If (weight=='W' .Or. weight=='w') Then
        Read (nin,*)(ifact(i),y(i),wt(i),i=1,n)
      Else
        Read (nin,*)(ifact(i),y(i),i=1,n)
      End If
      If (typ=='P' .Or. typ=='p') Then
        Read (nin,*) v(1:levels)
      End If

!     Calculate dummy variables
      ifail = 0
      Call g04eaf(typ,n,levels,ifact,x,ldx,v,rep,ifail)

      If (typ=='C' .Or. typ=='c') Then
        m = levels
      Else
        m = levels - 1
      End If
      ip = m
      If (mean=='M' .Or. mean=='m') Then
        ip = ip + 1
      End If

      ldq = n
      Allocate (isx(m),b(ip),se(ip),cov(ip*(ip+1)/2),res(n),h(n),q(ldq,ip+1),p &
        (2*ip+ip*ip),wk(5*(ip-1)+ip*ip))

!     Use all the variables in the regression
      isx(1:m) = 1

!     Use the suggested value for tolerance
      tol = 0.00001E0_nag_wp

!     Fit linear regression model
      ifail = 0
      Call g02daf(mean,weight,n,x,ldx,m,isx,ip,y,wt,rss,idf,b,se,cov,res,h,q,  &
        ldq,svd,irank,p,tol,wk,ifail)

!     Display the results of the regression
      If (svd) Then
        Write (nout,99999) 'Model not of full rank, rank = ', irank
        Write (nout,*)
      End If
      Write (nout,99998) 'Residual sum of squares = ', rss
      Write (nout,99999) 'Degrees of freedom = ', idf
      Write (nout,*)
      Write (nout,*) 'Variable   Parameter estimate   Standard error'
      Write (nout,*)
      Write (nout,99997)(j,b(j),se(j),j=1,ip)

99999 Format (1X,A,I4)
99998 Format (1X,A,E12.4)
99997 Format (1X,I6,2E20.4)
    End Program g04eafe
```

10.2 Program Data

```
G04EAF Example Program Data
 12 4 'C' 'U' 'M'
1 33.63
4 39.62
2 38.18
3 41.46
4 38.02
2 35.83
4 35.99
1 36.58
3 42.92
1 37.80
3 40.43
2 37.89
```

10.3 Program Results

```
 G04EAF Example Program Results

 Model not of full rank, rank =    4

 Residual sum of squares =   0.2223E+02
 Degrees of freedom =    8

 Variable   Parameter estimate   Standard error

      1          0.3056E+02          0.3849E+00
      2          0.5447E+01          0.8390E+00
      3          0.6743E+01          0.8390E+00
      4          0.1105E+02          0.8390E+00
      5          0.7320E+01          0.8390E+00
```