NAG Library Routine Document

G03CCF

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

G03CCF computes factor score coefficients from the result of fitting a factor analysis model by maximum likelihood as performed by G03CAF.

2 Specification

```
SUBROUTINE G03CCF (METHOD, ROTATE, NVAR, NFAC, FL, LDFL, PSI, E, R, LDR,    &
                   FS, LDFS, WK, IFAIL)

INTEGER             NVAR, NFAC, LDFL, LDR, LDFS, IFAIL
REAL (KIND=nag_wp) FL(LDFL,NFAC), PSI(NVAR), E(NVAR), R(LDR,*),            &
                   FS(LDFS,NFAC), WK(NVAR)
CHARACTER(1)        METHOD, ROTATE
```

3 Description

A factor analysis model aims to account for the covariances among p variables, observed on n individuals, in terms of a smaller number, k, of unobserved variables or factors. The values of the factors for an individual are known as factor scores. G03CAF fits the factor analysis model by maximum likelihood and returns the estimated factor loading matrix, Λ, and the diagonal matrix of variances of the unique components, Ψ. To obtain estimates of the factors, a p by k matrix of factor score coefficients, Φ, is formed. The estimated vector of factor scores, $\hat{f}$, is then given by:

$$\hat{f} = x^{\mathrm{T}}\Phi,$$

where x is the vector of observed variables for an individual.

There are two commonly used methods of obtaining factor score coefficients.

The regression method:

$$\Phi = \Psi^{-1}\Lambda\left(I + \Lambda^{\mathrm{T}}\Psi^{-1}\Lambda\right)^{-1},$$

and Bartlett's method:

$$\Phi = \Psi^{-1}\Lambda\left(\Lambda^{\mathrm{T}}\Psi^{-1}\Lambda\right)^{-1}.$$

See Lawley and Maxwell (1971) for details of both methods. In the regression method as given above, it is assumed that the factors are not correlated and have unit variance; this is true for models fitted by G03CAF. Further, for models fitted by G03CAF,

$$\Lambda^{\mathrm{T}}\Psi^{-1}\Lambda = \Theta - I,$$

where Θ is the diagonal matrix of eigenvalues of the matrix S^*, as described in G03CAF.

The factors may be orthogonally rotated using an orthogonal rotation matrix, R, as computed by G03BAF. The factor scores for the rotated matrix are then given by ΛR.

4 References

Lawley D N and Maxwell A E (1971) *Factor Analysis as a Statistical Method* (2nd Edition) Butterworths

5 Parameters

1: METHOD – CHARACTER(1) *Input*

On entry: indicates which method is to be used to compute the factor score coefficients.

METHOD = 'R'
The regression method is used.

METHOD = 'B'
Bartlett's method is used.

Constraint: METHOD = 'B' or 'R'.

2: ROTATE – CHARACTER(1) *Input*

On entry: indicates whether a rotation is to be applied.

ROTATE = 'R'
A rotation will be applied to the coefficients and the rotation matrix, R, must be given in R.

ROTATE = 'U'
No rotation is applied.

Constraint: ROTATE = 'R' or 'U'.

3: NVAR – INTEGER *Input*

On entry: p, the number of observed variables in the factor analysis.

Constraint: NVAR $\geq$ NFAC.

4: NFAC – INTEGER *Input*

On entry: k, the number of factors in the factor analysis.

Constraint: NFAC ≥ 1.

5: FL(LDFL, NFAC) – REAL (KIND=nag_wp) array *Input*

On entry: Λ, the matrix of unrotated factor loadings as returned by G03CAF.

6: LDFL – INTEGER *Input*

On entry: the first dimension of the array FL as declared in the (sub)program from which G03CCF is called.

Constraint: LDFL $\geq$ NVAR.

7: PSI(NVAR) – REAL (KIND=nag_wp) array *Input*

On entry: the diagonal elements of Ψ, as returned by G03CAF.

Constraint: PSI$(i) > 0.0$, for $i = 1, 2, \ldots, p$.

8: E(NVAR) – REAL (KIND=nag_wp) array *Input*

On entry: the eigenvalues of the matrix S^*, as returned by G03CAF.

Constraint: E$(i) > 1.0$, for $i = 1, 2, \ldots, p$.

9: R(LDR, $*$) – REAL (KIND=nag_wp) array *Input*

Note: the second dimension of the array R must be at least 1 if ROTATE = 'U' and at least NFAC if ROTATE = 'R'.

On entry: if ROTATE = 'R', R must contain the orthogonal rotation matrix, R, as returned by G03BAF.

If ROTATE = 'U', R need not be set.

10: LDR – INTEGER *Input*

On entry: the first dimension of the array R as declared in the (sub)program from which G03CCF is called.

Constraints:

if ROTATE = 'R', LDR $\geq$ NFAC;
otherwise 1.

11: FS(LDFS, NFAC) – REAL (KIND=nag_wp) array *Output*

On exit: the matrix of factor score coefficients, Φ. FS(i,j) contains the factor score coefficient for the jth factor and the ith observed variable, for $i = 1, 2, \ldots, p$ and $j = 1, 2, \ldots, k$.

12: LDFS – INTEGER *Input*

On entry: the first dimension of the array FS as declared in the (sub)program from which G03CCF is called.

Constraint: LDFS $\geq$ NVAR.

13: WK(NVAR) – REAL (KIND=nag_wp) array *Workspace*

14: IFAIL – INTEGER *Input/Output*

On entry: IFAIL must be set to 0, -1 or 1. If you are unfamiliar with this parameter you should refer to Section 3.3 in the Essential Introduction 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 parameter, 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, NFAC $<$ 1,
or NVAR $<$ NFAC,
or LDFL $<$ NVAR,
or LDFS $<$ NVAR,
or ROTATE = 'R' and LDR $<$ NFAC,
or METHOD $\neq$ 'R' or 'B',
or ROTATE $\neq$ 'R' or 'U'.

IFAIL = 2

On entry, a value of PSI $\leq$ 0.0,
or a value of E $\leq$ 1.0.

IFAIL $= -99$

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

See Section 3.8 in the Essential Introduction for further information.

IFAIL $= -399$

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

See Section 3.7 in the Essential Introduction for further information.

IFAIL $= -999$

Dynamic memory allocation failed.

See Section 3.6 in the Essential Introduction for further information.

7 Accuracy

Accuracy will depend on the accuracy requested when computing the estimated factor loadings using G03CAF.

8 Parallelism and Performance

G03CCF is not threaded by NAG in any implementation.

G03CCF 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

To compute the factor scores using the factor score coefficients the values for the observed variables first need to be standardized by subtracting the sample means and, if the factor analysis is based upon a correlation matrix, dividing by the sample standard deviations. This may be performed using G03ZAF. The standardized variables are then post-multiplied by the factor score coefficients. This may be performed using routines from Chapter F06, for example F06YAF (DGEMM).

If principal component analysis is required the routine G03AAF computes the principal component scores directly. Hence, the factor score coefficients are not needed.

10 Example

This example is taken from Lawley and Maxwell (1971). The correlation matrix for 220 observations on six school subjects is input and a factor analysis model with two factors fitted using G03CAF. The factor score coefficients are computed using the regression method.

10.1 Program Text

```
    Program g03ccfe

!     G03CCF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g03caf, g03ccf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
```

```
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, ldfl, ldfs, ldr, ldx,      &
                                          liwk, lres, lwk, lwt, m, n, nfac,    &
                                          nvar, tdr
      Character (80)                   :: fmt
      Character (1)                    :: matrix, method, rotate, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: com(:), e(:), fl(:,:), fs(:,:),      &
                                          psi(:), r(:,:), res(:), wk(:),       &
                                          wt(:), x(:,:)
      Real (Kind=nag_wp)               :: stat(4)
      Integer                          :: iop(5)
      Integer, Allocatable             :: isx(:), iwk(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count, max
!     .. Executable Statements ..
      Write (nout,*) 'G03CCF Example Program Results'
      Write (nout,*)
      Flush (nout)

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

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

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

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

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

!     Calculate NVAR
      nvar = count(isx(1:m)==1)

!     Do not apply a rotation
      rotate = 'U'
      tdr = 1
      ldr = 1

      lres = nvar*(nvar-1)/2
      liwk = 4*nvar + 2
      lwk = 5*nvar*nvar + 33*nvar - 4/2
      If (matrix/='C' .And. matrix/='c') Then
        lwk = max(lwk,n*nvar+7*nvar+nvar*(nvar-1)/2)
      End If
      lwk = max(lwk,nvar)
      ldfs = nvar
      ldfl = nvar
      Allocate (e(nvar),com(nvar),psi(nvar),res(lres),fl(ldfl,nfac),wk(lwk), &
        iwk(liwk),fs(ldfs,nfac),r(ldr,tdr))
```

```
!     Read in options
      Read (nin,*) iop(1:5)

!     Fit factor analysis model
      ifail = -1
      Call g03caf(matrix,weight,n,m,x,ldx,nvar,isx,nfac,wt,e,stat,com,psi,res, &
        fl,ldfl,iop,iwk,wk,lwk,ifail)
      If (ifail/=0) Then
        If (ifail<=4) Then
          Go To 100
        End If
      End If

!     Display results
      Write (nout,*) ' Loadings, Communalities and PSI'
      Write (nout,*)
      Write (fmt,99999) '(', nfac + 2, '(1X,F8.3))'
      Write (nout,fmt)(fl(i,1:nfac),com(i),psi(i),i=1,nvar)

!     Read in details of how to compute factor scores
      Read (nin,*) method

!     Compute factor scores
      ifail = 0
      Call g03ccf(method,rotate,nvar,nfac,fl,ldfl,psi,e,r,ldr,fs,ldfs,wk, &
        ifail)

!     Display factor score coefficients
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',nvar,nfac,fs,ldfs,'Factor score coefficients', &
        ifail)

100   Continue

99999 Format (A,I0,A)
    End Program g03ccfe
```

10.2 Program Data

```
G03CCF Example Program Data
'C' 'U' 220 6  2
1.000 0.439 0.410 0.288 0.329 0.248
0.439 1.000 0.351 0.354 0.320 0.329
0.410 0.351 1.000 0.164 0.190 0.181
0.288 0.354 0.164 1.000 0.595 0.470
0.329 0.320 0.190 0.595 1.000 0.464
0.248 0.329 0.181 0.470 0.464 1.000
  1    1    1   1    1   1
1 -1 500 3 5
'R'
```

10.3 Program Results

```
 G03CCF Example Program Results

  Loadings, Communalities and PSI

    0.553   -0.429    0.490    0.510
    0.568   -0.288    0.406    0.594
    0.392   -0.450    0.356    0.644
    0.740    0.273    0.623    0.377
    0.724    0.211    0.569    0.431
    0.595    0.132    0.372    0.628

 Factor score coefficients
          1       2
 1   0.1932 -0.3920
```

```
 2    0.1703 -0.2265
 3    0.1085 -0.3262
 4    0.3495  0.3374
 5    0.2989  0.2286
 6    0.1688  0.0978
```