NAG Library Routine Document

G03AAF

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

G03AAF performs a principal component analysis on a data matrix; both the principal component loadings and the principal component scores are returned.

2 Specification

```
SUBROUTINE G03AAF (MATRIX, STD, WEIGHT, N, M, X, LDX, ISX, S, WT, NVAR,    &
                   E, LDE, P, LDP, V, LDV, WK, IFAIL)
INTEGER            N, M, LDX, ISX(M), NVAR, LDE, LDP, LDV, IFAIL
REAL (KIND=nag_wp) X(LDX,M), S(M), WT(*), E(LDE,6), P(LDP,NVAR),          &
                   V(LDV,NVAR), WK(1)
CHARACTER(1)       MATRIX, STD, WEIGHT
```

3 Description

Let X be an n by p data matrix of n observations on p variables $x_1, x_2, \ldots, x_p$ and let the p by p variance-covariance matrix of $x_1, x_2, \ldots, x_p$ be S. A vector a_1 of length p is found such that:

$$a_1^{\mathrm{T}} S a_1 \quad \text{is maximized subject to} \quad a_1^{\mathrm{T}} a_1 = 1.$$

The variable $z_1 = \sum_{i=1}^{p} a_{1i} x_i$ is known as the first principal component and gives the linear combination of the variables that gives the maximum variation. A second principal component, $z_2 = \sum_{i=1}^{p} a_{2i} x_i$, is found such that:

$$a_2^{\mathrm{T}} S a_2 \quad \text{is maximized subject to } a_2^{\mathrm{T}} a_2 = 1 \text{and } a_2^{\mathrm{T}} a_1 = 0.$$

This gives the linear combination of variables that is orthogonal to the first principal component that gives the maximum variation. Further principal components are derived in a similar way.

The vectors $a_1, a_2, \ldots, a_p$, are the eigenvectors of the matrix S and associated with each eigenvector is the eigenvalue, λ_i^2. The value of $\lambda_i^2 / \sum \lambda_i^2$ gives the proportion of variation explained by the ith principal component. Alternatively, the a_i's can be considered as the right singular vectors in a singular value decomposition with singular values λ_i of the data matrix centred about its mean and scaled by $1/\sqrt{(n-1)}$, X_s. This latter approach is used in G03AAF, with

$$X_s = V \Lambda P'$$

where Λ is a diagonal matrix with elements λ_i, P is the p by p matrix with columns a_i and V is an n by p matrix with $V'V = I$, which gives the principal component scores.

Principal component analysis is often used to reduce the dimension of a dataset, replacing a large number of correlated variables with a smaller number of orthogonal variables that still contain most of the information in the original dataset.

The choice of the number of dimensions required is usually based on the amount of variation accounted for by the leading principal components. If k principal components are selected, then a test of the equality of the remaining $p-k$ eigenvalues is

$$(n - (2p + 5)/6)\left\{-\sum_{i=k+1}^{p}\log\left(\lambda_i^2\right) + (p-k)\log\left(\sum_{i=k+1}^{p}\lambda_i^2/(p-k)\right)\right\}$$

which has, asymptotically, a χ^2-distribution with $\frac{1}{2}(p-k-1)(p-k+2)$ degrees of freedom.

Equality of the remaining eigenvalues indicates that if any more principal components are to be considered then they all should be considered.

Instead of the variance-covariance matrix the correlation matrix, the sums of squares and cross-products matrix or a standardized sums of squares and cross-products matrix may be used. In the last case S is replaced by $\sigma^{-\frac{1}{2}}S\sigma^{-\frac{1}{2}}$ for a diagonal matrix σ with positive elements. If the correlation matrix is used, the χ^2 approximation for the statistic given above is not valid.

The principal component scores, F, are the values of the principal component variables for the observations. These can be standardized so that the variance of these scores for each principal component is 1.0 or equal to the corresponding eigenvalue.

Weights can be used with the analysis, in which case the matrix X is first centred about the weighted means then each row is scaled by an amount $\sqrt{w_i}$, where w_i is the weight for the ith observation.

4 References

Chatfield C and Collins A J (1980) *Introduction to Multivariate Analysis* Chapman and Hall

Cooley W C and Lohnes P R (1971) *Multivariate Data Analysis* Wiley

Hammarling S (1985) The singular value decomposition in multivariate statistics *SIGNUM Newsl.* **20(3)** 2–25

Kendall M G and Stuart A (1969) *The Advanced Theory of Statistics (Volume 1)* (3rd Edition) Griffin

Morrison D F (1967) *Multivariate Statistical Methods* McGraw–Hill

5 Parameters

1: MATRIX – CHARACTER(1) *Input*

On entry: indicates for which type of matrix the principal component analysis is to be carried out.

MATRIX = 'C'
It is for the correlation matrix.

MATRIX = 'S'
It is for a standardized matrix, with standardizations given by S.

MATRIX = 'U'
It is for the sums of squares and cross-products matrix.

MATRIX = 'V'
It is for the variance-covariance matrix.

Constraint: MATRIX = 'C', 'S', 'U' or 'V'.

2: STD – CHARACTER(1) *Input*

On entry: indicates if the principal component scores are to be standardized.

STD = 'S'
The principal component scores are standardized so that $F'F = I$, i.e., $F = X_sP\Lambda^{-1} = V$.

STD = 'U'
The principal component scores are unstandardized, i.e., $F = X_sP = V\Lambda$.

STD = 'Z'
The principal component scores are standardized so that they have unit variance.

STD = 'E'
The principal component scores are standardized so that they have variance equal to the corresponding eigenvalue.

Constraint: STD = 'E', 'S', 'U' or 'Z'.

3: WEIGHT – CHARACTER(1) *Input*

On entry: indicates if weights are to be used.

WEIGHT = 'U'
No weights are used.

WEIGHT = 'W'
Weights are used and must be supplied in WT.

Constraint: WEIGHT = 'U' or 'W'.

4: N – INTEGER *Input*

On entry: n, the number of observations.

Constraint: $\mathrm{N} \geq 2$.

5: M – INTEGER *Input*

On entry: m, the number of variables in the data matrix.

Constraint: $\mathrm{M} \geq 1$.

6: X(LDX, M) – REAL (KIND=nag_wp) array *Input*

On entry: $\mathrm{X}(i,j)$ must contain the ith observation for the jth variable, for $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, m$.

7: LDX – INTEGER *Input*

On entry: the first dimension of the array X as declared in the (sub)program from which G03AAF is called.

Constraint: $\mathrm{LDX} \geq \mathrm{N}$.

8: ISX(M) – INTEGER array *Input*

On entry: $\mathrm{ISX}(j)$ indicates whether or not the jth variable is to be included in the analysis.

If $\mathrm{ISX}(j) > 0$, the variable contained in the jth column of X is included in the principal component analysis, for $j = 1, 2, \ldots, m$.

Constraint: $\mathrm{ISX}(j) > 0$ for NVAR values of j.

9: S(M) – REAL (KIND=nag_wp) array *Input/Output*

On entry: the standardizations to be used, if any.

If MATRIX = 'S', the first m elements of S must contain the standardization coefficients, the diagonal elements of σ.

Constraint: if $\mathrm{ISX}(j) > 0$, $\mathrm{S}(j) > 0.0$, for $j = 1, 2, \ldots, m$.

On exit: if MATRIX = 'S', S is unchanged on exit.

If MATRIX = 'C', S contains the variances of the selected variables. $\mathrm{S}(j)$ contains the variance of the variable in the jth column of X if $\mathrm{ISX}(j) > 0$.

If MATRIX = 'U' or 'V', S is not referenced.

10: WT($*$) – REAL (KIND=nag_wp) array *Input*

Note: the dimension of the array WT must be at least N if WEIGHT = 'W', and at least 1 otherwise.

On entry: if WEIGHT = 'W', the first n elements of WT must contain the weights to be used in the principal component analysis.

If WT(i) = 0.0, the ith observation is not included in the analysis. The effective number of observations is the sum of the weights.

If WEIGHT = 'U', WT is not referenced and the effective number of observations is n.

Constraints:

WT(i) ≥ 0.0, for $i = 1, 2, \ldots, n$;
the sum of weights $\geq$ NVAR + 1.

11: NVAR – INTEGER *Input*

On entry: p, the number of variables in the principal component analysis.

Constraint: $1 \leq \text{NVAR} \leq \min(\text{N} - 1, \text{M})$.

12: E(LDE, 6) – REAL (KIND=nag_wp) array *Output*

On exit: the statistics of the principal component analysis.

E(i, 1)
The eigenvalues associated with the ith principal component, λ_i^2, for $i = 1, 2, \ldots, p$.

E(i, 2)
The proportion of variation explained by the ith principal component, for $i = 1, 2, \ldots, p$.

E(i, 3)
The cumulative proportion of variation explained by the first ith principal components, for $i = 1, 2, \ldots, p$.

E(i, 4)
The χ^2 statistics, for $i = 1, 2, \ldots, p$.

E(i, 5)
The degrees of freedom for the χ^2 statistics, for $i = 1, 2, \ldots, p$.

If MATRIX $\neq$ 'C', E(i, 6) contains significance level for the χ^2 statistic, for $i = 1, 2, \ldots, p$.

If MATRIX = 'C', E(i, 6) is returned as zero.

13: LDE – INTEGER *Input*

On entry: the first dimension of the array E as declared in the (sub)program from which G03AAF is called.

Constraint: LDE $\geq$ NVAR.

14: P(LDP, NVAR) – REAL (KIND=nag_wp) array *Output*

On exit: the first NVAR columns of P contain the principal component loadings, a_i. The jth column of P contains the NVAR coefficients for the jth principal component.

15: LDP – INTEGER *Input*

On entry: the first dimension of the array P as declared in the (sub)program from which G03AAF is called.

Constraint: LDP $\geq$ NVAR.

16: V(LDV, NVAR) – REAL (KIND=nag_wp) array *Output*

On exit: the first NVAR columns of V contain the principal component scores. The jth column of V contains the N scores for the jth principal component.

If WEIGHT = 'W', any rows for which WT(i) is zero will be set to zero.

17: LDV – INTEGER *Input*

On entry: the first dimension of the array V as declared in the (sub)program from which G03AAF is called.

Constraint: LDV $\geq$ N.

18: WK(1) – REAL (KIND=nag_wp) array *Input*

This parameter is no longer accessed by G03AAF. Workspace is provided internally by dynamic allocation instead.

19: 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, M < 1,
or N < 2,
or NVAR < 1,
or NVAR > M,
or NVAR $\geq$ N,
or LDX < N,
or LDV < N,
or LDP < NVAR,
or LDE < NVAR,
or MATRIX $\neq$ 'C', 'S', 'U' or 'V',
or STD $\neq$ 'S', 'U', 'Z' or 'E',
or WEIGHT $\neq$ 'U' or 'W'.

IFAIL = 2

On entry, WEIGHT = 'W' and a value of WT < 0.0.

IFAIL = 3

On entry, there are not NVAR values of ISX > 0,
or WEIGHT = 'W' and the effective number of observations is less than NVAR + 1.

IFAIL $= 4$

On entry, $\mathrm{S}(j) \leq 0.0$ for some $j = 1, 2, \ldots, m$, when MATRIX = 'S' and $\mathrm{ISX}(j) > 0$.

IFAIL $= 5$

The singular value decomposition has failed to converge. This is an unlikely error exit.

IFAIL $= 6$

All eigenvalues/singular values are zero. This will be caused by all the variables being constant.

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

As G03AAF uses a singular value decomposition of the data matrix, it will be less affected by ill-conditioned problems than traditional methods using the eigenvalue decomposition of the variance-covariance matrix.

8 Parallelism and Performance

G03AAF is not threaded by NAG in any implementation.

G03AAF 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

None.

10 Example

A dataset is taken from Cooley and Lohnes (1971), it consists of ten observations on three variables. The unweighted principal components based on the variance-covariance matrix are computed and the principal component scores requested. The principal component scores are standardized so that they have variance equal to the corresponding eigenvalue.

10.1 Program Text

```
    Program g03aafe

!     G03AAF Example Program Text

!     Mark 25 Release. NAG Copyright 2014.

!     .. Use Statements ..
      Use nag_library, Only: g03aaf, nag_wp, x04caf
!     .. Implicit None Statement ..
      Implicit None
!     .. Parameters ..
      Integer, Parameter               :: nin = 5, nout = 6
!     .. Local Scalars ..
      Integer                          :: i, ifail, lde, ldp, ldv, ldx, lwt,   &
                                          m, n, nvar
      Character (1)                    :: matrix, std, weight
!     .. Local Arrays ..
      Real (Kind=nag_wp), Allocatable  :: e(:,:), p(:,:), s(:), v(:,:), wk(:), &
                                          wt(:), x(:,:)
      Integer, Allocatable             :: isx(:)
!     .. Intrinsic Procedures ..
      Intrinsic                        :: count
!     .. Executable Statements ..
      Write (nout,*) 'G03AAF Example Program Results'
      Write (nout,*)

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

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

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

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

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

!     Read in standardizations
      If (matrix=='S' .Or. matrix=='s') Then
        Read (nin,*) s(1:m)
      End If

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

      lde = nvar
      ldp = nvar
      ldv = n
      Allocate (e(lde,6),p(ldp,nvar),v(ldv,nvar),wk(1))

!     Perform PCA
      ifail = 0
      Call g03aaf(matrix,std,weight,n,m,x,ldx,isx,s,wt,nvar,e,lde,p,ldp,v,ldv, &
        wk,ifail)

!     Display results
```

```
      Write (nout,*) &
        'Eigenvalues  Percentage  Cumulative     Chisq      DF     Sig'
      Write (nout,*) '              variation   variation'
      Write (nout,*)
      Write (nout,99999)(e(i,1:6),i=1,nvar)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',nvar,nvar,p,ldp,'Principal component loadings' &
        ,ifail)
      Write (nout,*)
      Flush (nout)
      ifail = 0
      Call x04caf('General',' ',n,nvar,v,ldv,'Principal component scores', &
        ifail)

99999 Format (1X,F11.4,2F12.4,F10.4,F8.1,F8.4)
    End Program g03aafe
```

10.2 Program Data

```
G03AAF Example Program Data
 'V' 'E' 'U' 10 3
7.0 4.0 3.0
4.0 1.0 8.0
6.0 3.0 5.0
8.0 6.0 1.0
8.0 5.0 7.0
7.0 2.0 9.0
5.0 3.0 3.0
9.0 5.0 8.0
7.0 4.0 5.0
8.0 2.0 2.0
 1   1   1
```

10.3 Program Results

```
 G03AAF Example Program Results

 Eigenvalues  Percentage  Cumulative     Chisq      DF     Sig
               variation   variation

     8.2739      0.6515      0.6515    8.6127     5.0  0.1255
     3.6761      0.2895      0.9410    4.1183     2.0  0.1276
     0.7499      0.0590      1.0000    0.0000     0.0  0.0000

 Principal component loadings
          1       2       3
 1  -0.1376  0.6990  0.7017
 2  -0.2505  0.6609 -0.7075
 3   0.9583  0.2731 -0.0842

 Principal component scores
              1          2          3
  1     -2.1514    -0.1731    -0.1068
  2      3.8042    -2.8875    -0.5104
  3      0.1532    -0.9869    -0.2694
  4     -4.7065     1.3015    -0.6517
  5      1.2938     2.2791    -0.4492
  6      4.0993     0.1436     0.8031
  7     -1.6258    -2.2321    -0.8028
  8      2.1145     3.2512     0.1684
  9     -0.2348     0.3730    -0.2751
 10     -2.7464    -1.0689     2.0940
```

Example Program
Principal Component Analysis