Principal Component Analysis: nagdmc_pca

Purpose

nagdmc_pca computes a principal component analysis.

Declaration

```
#include <nagdmc.h>
void nagdmc_pca(long rec1, long nvar, long nrec, long dblk, double data[],
                void (*dfun)(long, long, double [], char *, int *), char *comm,
                long chunksize, long iwts, int pcatype, double xbar[],
                double s[], double loadings[], double results[], int *info);
```

Parameters

1: **rec1** – `long` *Input*

On entry: the index in the data of the first data record used in the analysis.

Constraint: **rec1** ≥ 0.

2: **nvar** – `long` *Input*

On entry: the number of variables in the data.

Constraint: **nvar** > 1.

3: **nrec** – `long` *Input*

On entry: the number of consecutive records, beginning at **rec1**, used in the analysis.

Constraint: **nrec** > 1.

4: **dblk** – `long` *Input*

On entry: the total number of records in the data block.

Constraint: **dblk** $\geq$ **rec1** + **nrec**.

5: **data**[**dblk** $*$ **nvar**] – `double` *Input*

On entry: the data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar**-1) are stored in **data**[$i*$**nvar**$+j$], for $i = 0, 1, \ldots,$ **dblk** $- 1$. When the data function is used, **data** is not referenced.

6: **dfun** – function supplied by user *External Procedure*

On entry: the pointer to a data function supplied by the user.

Constraint: if **dfun** is a valid pointer, **data** must be 0.

The specification of **dfun** is:

```
void dfun(long irec, long chunksize, double x[], char *comm, int *ierr)
```

1: **irec** – `long` *Input*

On entry: the index in the data of the first record returned.

2: **chunksize** – `long` *Input*

On entry: the number of consecutive records returned.

3: **x**[**chunksize**$*$**nvar**] – `double` *Output*

On exit: data values for the jth variable (for $j = 0, 1, \ldots,$ **nvar** $- 1$) must be returned in **x**[$i *$ **nvar** $+ j$], for $i = 0, 1, \ldots,$ **chunksize** $- 1$.

4: **comm** – `char *` *Input*

On entry: a communication parameter allowing additional information to be passed to **dfun**. This parameter is passed 'as is' through the calling function.

5: **ierr** – `int *` *Output*

On exit: if the value pointed to by **ierr** on return is greater than 100, the NAG DMC function will terminate immediately and **info** will point to this value.

7: **comm** – `char *` *Input*

On entry: a communication parameter allowing additional information to be passed to **dfun**. This parameter is passed 'as is' through the calling function.

8: **chunksize** – `long` *Input*

On entry: if the data function is used, the function inputs no more than **chunksize** data records at a time; otherwise **chunksize** is not referenced.

Constraint: if **dfun** $\neq 0$, **chunksize** ≥ 1.

9: **iwts** – `long` *Input*

On entry: if **iwts** $= -1$, no weights are used; otherwise **iwts** is the index in **data** in which the weights are stored.

Constraints: $-1 \leq$ **iwts** $<$ **nvar**; **iwts** $\neq$ **yvar**; and if **nxvar** > 0, **iwts** $\neq$ **xvar**$[i]$, for $i = 0, 1, \ldots,$ **nxvar** $- 1$.

10: **pcatype** – `int` *Input*

On entry: indicates the equivalent matrix for which the principal components are derived:

pcatype $= 0$ sum of squares and cross-products;
pcatype $= 1$ variance-covariance matrix;
pcatype $= 2$ correlation matrix;
pcatype $= 3$ user-supplied standardisation.

Constraint: **pcatype** $\in \{0, 1, 2, 3\}$.

11: **xbar**[**nvar**] – `double` *Input*

On entry: if variable means are available, they should be supplied in **xbar**; otherwise **xbar** should be set to 0 and **nagdmc_pca** will compute the means internally using an additional pass through the data. Note that values corresponding to the column of weights, if any, will be ignored.

12: **s**[**nvar**] – `double` *Input*

On entry: the vector of standard deviations or scaling factors. If **pcatype** $= 2$ and standard deviations are available, they should be supplied in **s**; otherwise **nagdmc_pca** will compute the means and standard deviations internally using an additional pass through the data. If **pcatype** $= 3$, **s** must contain the user-supplied standardisations. If **pcatype** is zero or one, **s** is not referenced and can be set to 0.

Constraints: if **s** is not 0 and is referenced, **s**$[i] > 0.0$, for $i = 0, 1, \ldots,$ **nvar** $- 1$.

13: **loadings**[**nvar**$*$**nvar**] – `double` *Output*

On exit: **loadings**$[i *$ **nvar** $+ j]$ is the jth loading from the principal component analysis for the ith variable, for $i = 0, 1, \ldots,$ **nvar** $- 1$; for $j = 0, 1, \ldots,$ **nvar** $- 1$.

14: **results**[6$*$**nvar**] – `double` *Output*

On exit: **results**$[i * 6 + j]$ is element j of the variance decomposition results for the ith variable, where elements have the meaning:

Element 0: the eigenvalue.
Element 1: the proportion of variation explained by the component.
Element 2: the cumulative proportion of variation explained by the components.
Element 3: the χ^2 statistic.
Element 4: the degrees of freedom.
Element 5: the significance.

15: **info** – `int *` *Output*

On exit: **info** gives information on the success of the function call:

0: the function successfully completed its task.
i; $i = i = 1, 2, 3, 4, 6, 8, 9, 10, 12$: the specification of the ith formal parameter was incorrect.

51: all eigenvalues are zero; this indicates that a null matrix has been entered.
52: the sum of weights is less than 1.
98: the singular value decomposition used in the calculation failed to converge.
99: the function failed to allocate enough memory.
> 100: an error occurred in a function specified by the user.

Notation

nrec the number of records, n.
nvar the number of variables in the analysis, p.
data the data matrix, X.

Description

Let X be an n by p data matrix of n data records 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

$$a_1^T S a_1 \text{ is maximised subject to } a_1^T a_1 = 1.$$

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

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

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 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 NAG DMC.

$$X_s = V \Lambda P',$$

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 data set, replacing a large number of correlated variables with a smaller number of orthogonal variables that still contain most of the information in the original data set.

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

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

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 standardised sums of squares and cross-products matrix may be used. In the last case S is replaced by $\sigma^{-1/2} S \sigma^{-1/2}$ 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.

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 data record.

References and Further Reading

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

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

Krzanowski W J (1990) *Principles of Multivariate Analysis* Oxford University Press.

See Also

nagdmc_pca_score	computes PCA scores for a given number of component directions.
pca_ex.c	the example calling program.