# NAG Toolbox for MATLAB

# g03ef

## 1 Purpose

g03ef performs $K$-means cluster analysis.

## 2 Syntax

```
[cmeans, inc, nic, css, csw, ifail] = g03ef(weight, n, x, isx, k, cmeans, wt, 'm', m, 'nvar', nvar, 'maxit', maxit)
```

## 3 Description

Given $n$ objects with $p$ variables measured on each object, $x_{ij}$ for $i = 1, 2, \ldots, n$; $j = 1, 2, \ldots, p$, g03ef allocates each object to one of $K$ groups or clusters to minimize the within-cluster sum of squares:

$$\sum_{k=1}^{K}\sum_{i\in S_k}\sum_{j=1}^{p}\left(x_{ij}-\bar{x}_{kj}\right)^2,$$

where $S_k$ is the set of objects in the $k$th cluster and $\bar{x}_{kj}$ is the mean for the variable $j$ over cluster $k$. This is often known as $K$-means clustering.

In addition to the data matrix, a $K$ by $p$ matrix giving the initial cluster centres for the $K$ clusters is required. The objects are then initially allocated to the cluster with the nearest cluster mean. Given the initial allocation, the procedure is to iteratively search for the $K$-partition with locally optimal within-cluster sum of squares by moving points from one cluster to another.

Optionally, weights for each object, $w_i$, can be used so that the clustering is based on within-cluster weighted sums of squares:

$$\sum_{k=1}^{K}\sum_{i\in S_k}\sum_{j=1}^{p}w_i\left(x_{ij}-\tilde{x}_{kj}\right)^2,$$

where $\tilde{x}_{kj}$ is the weighted mean for variable $j$ over cluster $k$.

The function is based on the algorithm of Hartigan and Wong 1979.

## 4 References

Everitt B S 1974 *Cluster Analysis* Heinemann

Hartigan J A and Wong M A 1979 Algorithm AS136: A K-means clustering algorithm *Appl. Statist.* **28** 100–108

Kendall M G and Stuart A 1976 *The Advanced Theory of Statistics (Volume 3)* (3rd Edition) Griffin

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

## 5 Parameters

### 5.1 Compulsory Input Parameters

1: **weight – string**

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'.

2: **n – int32 scalar**

$n$, the number of objects.

*Constraint*: $\mathbf{n} > 1$.

3: **x(ldx,m) – double array**

**ldx**, the first dimension of the array, must be at least **n**.

$\mathbf{x}(i,j)$ must contain the value of the $j$th variable for the $i$th object, for $i = 1,2,\ldots,n$ and $j = 1,2,\ldots,\mathbf{m}$.

4: **isx(m) – int32 array**

$\mathbf{isx}(j)$ indicates whether or not the $j$th variable is to be included in the analysis. If $\mathbf{isx}(j) > 0$, the variable contained in the $j$th column of **x** is included, for $j = 1,2,\ldots,\mathbf{m}$.

*Constraint*: $\mathbf{isx}(j) > 0$ for **nvar** values of $j$.

5: **k – int32 scalar**

$K$, the number of clusters.

*Constraint*: $\mathbf{k} \geq 2$.

6: **cmeans(ldc,nvar) – double array**

**ldc**, the first dimension of the array, must be at least **k**.

$\mathbf{cmeans}(i,j)$ must contain the value of the $j$th variable for the $i$th initial cluster centre, for $i = 1,2,\ldots,K$ and $j = 1,2,\ldots,p$.

7: **wt(*) – double array**

**Note**: the dimension of the array **wt** must be at least **n** if **weight** = 'W', and at least 1 otherwise.

If **weight** = 'W', the first $n$ elements of **wt** must contain the weights to be used.

If $\mathbf{wt}(i) = 0.0$, the $i$th observation is not included in the analysis. The effective number of observation is the sum of the weights.

If **weight** = 'U', **wt** is not referenced and the effective number of observations is $n$.

*Constraint*: if **weight** = 'W', $\mathbf{wt}(i) \geq 0.0$ and $\mathbf{wt}(i) > 0.0$ for at least two values of $i$, for $i = 1,2,\ldots,n$.

## 5.2 Optional Input Parameters

1: **m – int32 scalar**

*Default*: The dimension of the arrays **isx**, **x**. (An error is raised if these dimensions are not equal.)

the total number of variables in array **x**.

*Constraint*: $\mathbf{m} \geq \mathbf{nvar}$.

2: **nvar – int32 scalar**

*Default*: The dimension of the array **cmeans**.

$p$, the number of variables included in the sums of squares calculations.

*Constraint*: $1 \leq \mathbf{nvar} \leq \mathbf{m}$.

3: **maxit – int32 scalar**

The maximum number of iterations allowed in the analysis.

*Constraint*: $\mathbf{maxit} > 0$.

*Suggested value*: $\mathbf{maxit} = 10$.

*Default*: 10

### 5.3 Input Parameters Omitted from the MATLAB Interface

ldx, ldc, iwk, wk

### 5.4 Output Parameters

1: **cmeans(ldc,nvar) – double array**

**cmeans**$(i,j)$ contains the value of the $j$th variable for the $i$th computed cluster centre, for $i = 1,2,\ldots,K$ and $j = 1,2,\ldots,p$.

2: **inc(n) – int32 array**

**inc**$(i)$ contains the cluster to which the $i$th object has been allocated, for $i = 1,2,\ldots,n$.

3: **nic(k) – int32 array**

**nic**$(i)$ contains the number of objects in the $i$th cluster, for $i = 1,2,\ldots,K$.

4: **css(k) – double array**

**css**$(i)$ contains the within-cluster (weighted) sum of squares of the $i$th cluster, for $i = 1,2,\ldots,K$.

5: **csw(k) – double array**

**csw**$(i)$ contains the within-cluster sum of weights of the $i$th cluster, for $i = 1,2,\ldots,K$. If **weight** = 'U', the sum of weights is the number of objects in the cluster.

6: **ifail – int32 scalar**

0 unless the function detects an error (see Section 6).

## 6 Error Indicators and Warnings

Errors or warnings detected by the function:

**ifail** = 1

On entry, **weight** $\neq$ 'W' or 'U',
or **n** $< 2$,
or **nvar** $< 1$,
or **m** $<$ **nvar**,
or **k** $< 2$,
or **ldx** $<$ **n**,
or **ldc** $<$ **k**,
or **maxit** $\leq 0$.

**ifail** $= 2$

On entry, **weight** $=$ 'W' and a value of $\mathbf{wt}(i) < 0.0$ for some $i$,
or **weight** $=$ 'W' and $\mathbf{wt}(i) = 0.0$ for all or all but one values of $i$.

**ifail** $= 3$

On entry, the number of positive values in **isx** does not equal **nvar**.

**ifail** $= 4$

On entry, at least one cluster is empty after the initial assignment. Try a different set of initial cluster centres in **cmeans** and also consider decreasing the value of **k**. The empty clusters may be found by examining the values in **nic**.

**ifail** $= 5$

Convergence has not been achieved within the maximum number of iterations given by **maxit**. Try increasing **maxit** and, if possible, use the returned values in **cmeans** as the initial cluster centres.

## 7 Accuracy

g03ef produces clusters that are locally optimal; the within-cluster sum of squares may not be decreased by transferring a point from one cluster to another, but different partitions may have the same or smaller within-cluster sum of squares.

## 8 Further Comments

The time per iteration is approximately proportional to $npK$.

## 9 Example

```
weight = 'u';
n = int32(20);
x = [77.3, 13, 9.699999999999999, 1.5, 6.4;
     82.5, 10, 7.5, 1.5, 6.5;
     66.90000000000001, 20.6, 12.5, 2.3, 7;
     47.2, 33.8, 19, 2.8, 5.8;
     65.3, 20.5, 14.2, 1.9, 6.9;
     83.3, 10, 6.7, 2.2, 7;
     81.59999999999999, 12.7, 5.7, 2.9, 6.7;
     47.8, 36.5, 15.7, 2.3, 7.2;
     48.6, 37.1, 14.3, 2.1, 7.2;
     61.6, 25.5, 12.9, 1.9, 7.3;
     58.6, 26.5, 14.9, 2.4, 6.7;
     69.3, 22.3, 8.4, 4, 7;
     61.8, 30.8, 7.4, 2.7, 6.4;
     67.7, 25.3, 7, 4.8, 7.3;
     57.2, 31.2, 11.6, 2.4, 6.5;
     67.2, 22.7, 10.1, 3.3, 6.2;
     59.2, 31.2, 9.6, 2.4, 6;
     80.2, 13.2, 6.6, 2, 5.8;
     82.2, 11.1, 6.7, 2.2, 7.2;
     69.7, 20.7, 9.6, 3.1, 5.9];
isx = [int32(1);
     int32(1);
     int32(1);
     int32(1);
     int32(1)];
k = int32(3);
cmeans = [82.5, 10, 7.5, 1.5, 6.5;
     47.8, 36.5, 15.7, 2.3, 7.2;
     67.2, 22.7, 10.1, 3.3, 6.2];
```

```
wt = [0];
[cmeansOut, inc, nic, css, csw, ifail] = g03ef(weight, n, x, isx, k,
cmeans, wt)
```

```
cmeansOut =
   81.1833   11.6667    7.1500    2.0500    6.6000
   47.8667   35.8000   16.3333    2.4000    6.7333
   64.0455   25.2091   10.7455    2.8364    6.6545
inc =
           1
           1
           3
           2
           3
           1
           1
           2
           2
           3
           3
           3
           3
           3
           3
           3
           1
           1
           3
nic =
           6
           3
          11
css =
   46.5717
   20.3800
  468.8964
csw =
     6
     3
    11
ifail =
           0
```