

# Package ‘drclust’

April 22, 2024

**Type** Package

**Title** Simultaneous Clustering and (or) Dimensionality Reduction

**Version** 0.1

**Date** 2024-04-19

**Description** Methods for simultaneous clustering and dimensionality reduction such as: Double k-means, Reduced k-means, Factorial k-means, Clustering with Disjoint PCA but also methods for exclusively dimensionality reduction: Disjoint PCA, Disjoint FA. The statistical methods implemented refer to the following articles: de Soete G., Carroll J. (1994) "K-means clustering in a low-dimensional Euclidean space" <[doi:10.1007/978-3-642-51175-2\\_24](https://doi.org/10.1007/978-3-642-51175-2_24)> ; Vichi M. (2001) "Double k-means Clustering for Simultaneous Classification of Objects and Variables" <[doi:10.1007/978-3-642-59471-7\\_6](https://doi.org/10.1007/978-3-642-59471-7_6)> ; Vichi M., Kiers H.A.L. (2001) "Factorial k-means analysis for two-way data" <[doi:10.1016/S0167-9473\(00\)00064-5](https://doi.org/10.1016/S0167-9473(00)00064-5)> ; Vichi M., Saporta G. (2009) "Clustering and disjoint principal component analysis" <[doi:10.1016/j.csda.2008.05.028](https://doi.org/10.1016/j.csda.2008.05.028)> ; Vichi M. (2017) "Disjoint factor analysis with cross-loadings" <[doi:10.1007/s11634-016-0263-9](https://doi.org/10.1007/s11634-016-0263-9)>.

**License** GPL (>= 3)

**Encoding** UTF-8

**Imports** Rcpp, RcppArmadillo, fpc, cluster, factoextra, pheatmap

**LinkingTo** Rcpp, RcppArmadillo

**RoxygenNote** 7.3.1

**NeedsCompilation** yes

**Author** Ionel Prunila [aut, cre],  
Maurizio Vichi [aut]

**Maintainer** Ionel Prunila <[ionel.prunila@uniroma1.it](mailto:ionel.prunila@uniroma1.it)>

**Repository** CRAN

**Date/Publication** 2024-04-22 18:22:47 UTC

## R topics documented:

apseudoF . . . . .	2
centree . . . . .	3

cluster . . . . .	4
CronbachAlpha . . . . .	5
disfa . . . . .	6
dispca . . . . .	7
doublekm . . . . .	9
dpcakm . . . . .	11
dpseudoF . . . . .	12
factkm . . . . .	13
heatm . . . . .	15
kaiserCrit . . . . .	16
mrans . . . . .	17
redkm . . . . .	18
silhouette . . . . .	19

## Index 21

---

apseudoF	<i>pseudoF (pF or Calinski-Harabsz) index for choosing k in partitioning models</i>
----------	---

---

### Description

Calculates and plots the CH index for  $k = 2, \dots, \text{maxK}$ . The function provides an interval wide ( $2\text{tol} * \text{pF}$ ) so that the choice of K is less conservative. Instead of just choosing the maximum pF, if it exists, picks the value such that its upper bound is larger than max pF.

### Usage

```
apseudoF(data, maxK, tol, model, Q)
```

### Arguments

data	Units x variables numeric data matrix.
maxK	Maximum number of clusters for the units to be tested.
tol	Approximation value. It is half of the length of the interval put for each pF. $0 \leq \text{tol} < 1$ . Its default value is 0.05.
model	Partitioning Models to run for each value of k. (1 = doublekm; 2 = redkm; 3 = factkm; 4 = dpcakm)
Q	Number of principal components w.r.t. variables selected for the maxK -1 partitions to be tested.

### Value

bestK	best value of K (scalar).
-------	---------------------------

### Author(s)

Ionel Prunila, Maurizio Vichi

**References**

Calinski T., Harabasz J. (1974) "A dendrite method for cluster analysis" <doi:10.1080/03610927408827101>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

apF <- apseudoF(iris, maxK=10, tol = 0.05, model = 3, Q = 2)
```

---

centree

*Ward-dendrogram of centroids of partitioning models*

---

**Description**

Plots the Ward-dendrogram of the centroids of a partitioning model. The plot is useful as a diagnosis tool for the choice of the number of clusters.

**Usage**

```
centree(drclust_out)
```

**Arguments**

drclust\_out      Output of either doublekm, redkm, factkm or dpcakm.

**Value**

centroids-dkm    Centroids x centroids distance matrix.

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Ward J. H. (1963) "Hierarchical Grouping to Optimize an Objective Function" <doi:10.1080/01621459.1963.10500845>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

dc_out <- dpcakm(iris, 20, 3)
d <- centree(dc_out)
```

---

cluster	<i>classification variable</i>
---------	--------------------------------

---

### Description

Recodes the binary and row-stochastic membership matrix U into the classification variable (similar to the "cluster" output returned by kmeans()).

### Usage

```
cluster(U)
```

### Arguments

U                    Binary and row-stochastic matrix.

### Value

cl                    vector of length n indicating, for each element, the index of the cluster to which it has been assigned.

### Author(s)

Ionel Prunila, Maurizio Vichi

### Examples

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# standardizing the data
iris <- scale(iris)

# double k-means with 3 unit-clusters and 2 components for the variables
p1 <- redkm(iris, K = 3, Q = 2)
cl <- cluster(p1$U)
```

---

`CronbachAlpha`*Cronbach Alpha*

---

**Description**

Computes the Cronbach Alpha index on a units x variables data matrix. It measures the internal reliability, i.e., the propensity of J variables of a data matrix (n units x J variables) to be concordantly correlated with a single factor (composite indicator).

**Usage**

```
CronbachAlpha(X)
```

**Arguments**

X                      Units x variables numeric data matrix.

**Value**

as                      Cronbach's Alpha

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Cronbach L. J. (1951) "Coefficient alpha and the internal structure of tests" <doi:10.1007/BF02310555>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# standardizing the data
iris <- scale(iris)

# compute Cronbach's Alpha
as <- CronbachAlpha(iris)
```

---

disfa *Disjoint Factor Analysis*

---

### Description

Performs disjoint factor analysis, i.e., a Factor Analysis with a simple structure. In fact, each factor is defined by a disjoint subset of variables, resulting thus, in a simplified, easier to interpret loading matrix  $A$  and factors. Estimation is carried out via Maximum Likelihood.

### Usage

```
disfa(X, Q, Rndstart, verbose, maxiter, tol, constr, prep, print)
```

### Arguments

<code>X</code>	Units x variables numeric data matrix.
<code>Q</code>	Number of factors.
<code>Rndstart</code>	Number of runs to be performed (Defaults is 20).
<code>verbose</code>	Outputs basic summary statistics for each run (1 = enabled; 0 = disabled, default option).
<code>maxiter</code>	Maximum number of iterations allowed (if convergence is not yet reached. Default is 100).
<code>tol</code>	Tolerance threshold (maximum difference between the values of the objective function of two consecutive iterations such that convergence is assumed. Default is 1e-6).
<code>constr</code>	is a vector of length $J = nr.$ of variables, pre-specifying to which cluster some of the variables must be assigned. Each component of the vector can assume integer values from 1 to $Q$ (See example for more details), or 0 if no constraint on the variable is imposed (i.e., it will be assigned based on the plain algorithm).
<code>prep</code>	Pre-processing of the data. 1 performs the z-score transform (default choice); 2 performs the min-max transform; 0 leaves the data un-pre-processed.
<code>print</code>	Prints summary statistics of the performed method (1 = enabled; 0 = disabled, default option).

### Value

returns a list of estimates and some descriptive quantities of the final results.

<code>V</code>	Variables x factors membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which cluster each variable has been assigned.
<code>A</code>	Variables x components loading matrix.
<code>Psi</code>	Specific variance of each observed variable, not accounted for by the common factors (matrix).

discrepancy	Value of the objective function, to be minimized. Difference between the observed and estimated covariance matrices (scalar).
RMSEA	Adjusted Root Mean Squared Error (scalar).
AIC	Aikake Information Criterion (scalar).
BIC	Bayesian Information Criterion (scalar).
GFI	Goodness of Fit Index (scalar).

### Author(s)

Ionel Prunila, Maurizio Vichi

### References

Vichi M. (2017) "Disjoint factor analysis with cross-loadings" <doi:10.1007/s11634-016-0263-9>

### Examples

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# No constraint on variables
out <- disfa(iris, Q = 2)

# Constraint: the first two variables must contribute to the same factor.
outc <- disfa(iris, Q = 2, constr = c(1,1,0,0))
```

---

dispca

*Disjoint Principal Components Analysis*

---

### Description

Performs disjoint PCA, that is, a simplified version of PCA. Computes each one of the Q principal components from a different subset of the J variables (resulting thus, in a simplified, easier to interpret loading matrix A).

### Usage

```
dispca(X, Q, Rndstart, verbose, maxiter, tol, prep, print, constr)
```

**Arguments**

<code>X</code>	Units x variables numeric data matrix.
<code>Q</code>	Number of factors.
<code>Rndstart</code>	Number of runs to be performed (Defaults is 20).
<code>verbose</code>	Outputs basic summary statistics for each run (1 = enabled; 0 = disabled, default option).
<code>maxiter</code>	Maximum number of iterations allowed (if convergence is not yet reached. Default is 100).
<code>tol</code>	Tolerance threshold (maximum difference between the values of the objective function of two consecutive iterations such that convergence is assumed). Default is 1e-6.
<code>prep</code>	Pre-processing of the data. 1 performs the z-score transform (default choice); 2 performs the min-max transform; 0 leaves the data un-pre-processed.
<code>print</code>	Prints summary statistics of the results (1 = enabled; 0 = disabled, default option).
<code>constr</code>	is a vector of length $J = \text{nr. of variables}$ , pre-specifying to which cluster some of the variables must be assigned. Each component of the vector can assume integer values from $1 \text{ o } Q$ (See example for more details), or 0 if no constraint on the variable is imposed (i.e., it will be assigned based on the plain algorithm).

**Value**

returns a list of estimates and some descriptive quantities of the final results.

<code>V</code>	Variables x factors membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which cluster it has been assigned.
<code>A</code>	Variables x components loading matrix.
<code>betweenss</code>	Amount of deviance captured by the model (scalar).
<code>totss</code>	total amount of deviance (scalar).
<code>size</code>	Number of variables assigned to each column-cluster (vector).
<code>loop</code>	The index of the (best) run from which the results have been chosen.
<code>it</code>	the number of iterations performed during the (best) run.

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Vichi M., Saporta G. (2009) "Clustering and disjoint principal component analysis" <doi:10.1016/j.csda.2008.05.028>



**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# No constraint on variables
out <- dispca(iris, Q = 2)

# Constraint: the first two variables must contribute to the same factor.
outc <- dispca(iris, Q = 2, constr = c(1,1,0,0))
```

---

doublekm

*Double k-means Clustering*


---

**Description**

Performs simultaneous *k*-means partitioning on units and variables (rows and columns of the data matrix).

**Usage**

```
doublekm(Xs, K, Q, Rndstart, verbose, maxiter, tol, prep, print)
```

**Arguments**

Xs	Units x variables numeric data matrix.
K	Number of clusters for the units.
Q	Number of clusters for the variables.
Rndstart	Number of runs to be performed (Defaults is 20).
verbose	Outputs basic summary statistics for each run (1 = enabled; 0 = disabled, default option).
maxiter	Maximum number of iterations allowed (if convergence is not yet reached. Default is 100).
tol	Tolerance threshold. It is the maximum difference between the values of the objective function of two consecutive iterations such that convergence is assumed (default is 1e-6).
prep	Pre-processing of the data. 1 performs the z-score transform (default choice); 2 performs the min-max transform; 0 leaves the data un-pre-processed.
print	Prints summary statistics of the results (1 = enabled; 0 = disabled, default option).

**Value**

returns a list of estimates and some descriptive quantities of the final results.

U	Units x clusters membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which unit-cluster each unit has been assigned.
V	Variables x clusters membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which variable-cluster each variable has been assigned.
centers	K x Q matrix of centers containing the row means expressed in terms of column means.
totss	The total sum of squares (scalar).
withinss	Vector of within-row-cluster sum of squares, one component per cluster.
columnwise_withinss	Vector of within-column-cluster sum of squares, one component per cluster.
betweenss	Amount of deviance captured by the model (scalar).
K-size	Number of units assigned to each row-cluster (vector).
Q-size	Number of variables assigned to each column-cluster (vector).
pseudof	Calinski-Harabasz index of the resulting (row-) partition (scalar).
loop	The index of the (best) run from which the results have been chosen.
it	the number of iterations performed during the (best) run.

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Vichi M. (2001) "Double k-means Clustering for Simultaneous Classification of Objects and Variables" <doi:10.1007/978-3-642-59471-7\_6>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# double k-means with 3 unit-clusters and 2 variable-clusters
out <- doublekm(iris, K = 3, Q = 2)
```

dpcakm

*Clustering with Disjoint Principal Components Analysis***Description**

Performs simultaneously k-means partitioning on units and disjoint PCA on the variables, computing each principal component from a different subset of variables. The result is a simplified, easier to interpret loading matrix A, the principal components and the clustering. The reduced subspace is identified by the centroids.

**Usage**

```
dpcakm(X, K, Q, Rndstart, verbose, maxiter, tol, constr, print, prep)
```

**Arguments**

X	Units x variables numeric data matrix.
K	Number of clusters for the units.
Q	Number of principal components.
Rndstart	Number of runs to be performed (Defaults is 20).
verbose	Outputs basic summary statistics for each run (1 = enabled; 0 = disabled, default option).
maxiter	Maximum number of iterations allowed (if convergence is not yet reached. Default is 100).
tol	Tolerance threshold (maximum difference between the values of the objective function of two consecutive iterations such that convergence is assumed. Default is 1e-6).
constr	is a vector of length $J = nr.$ of variables, pre-specifying to which cluster some of the variables must be assigned. Each component of the vector can assume integer values from $1 \text{ o } Q = nr.$ of variable-cluster / principal components (See examples for more details), or 0 if no constraint on the variable is imposed (i.e., it will be assigned based on the plain algorithm).
print	Prints summary statistics of the results (1 = enabled; 0 = disabled, default option).
prep	Pre-processing of the data. 1 performs the z-score transform (default choice); 2 performs the min-max transform; 0 leaves the data un-pre-processed.

**Value**

returns a list of estimates and some descriptive quantities of the final results.

V	Variables x factors membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which cluster each variable has been assigned.
U	Units x clusters membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which cluster each unit has been assigned.

A	Variables x components loading matrix.
centers	K x Q matrix of centers containing the row means expressed in the reduced space of Q principal components.
totss	The total sum of squares (scalar).
withinss	Vector of within-cluster sum of squares, one component per cluster.
betweenss	Amount of deviance captured by the model (scalar).
K-size	Number of units assigned to each row-cluster (vector).
Q-size	Number of variables assigned to each column-cluster (vector).
pseudoF	Calinski-Harabasz index of the resulting partition (scalar).
loop	The index of the (best) run from which the results have been chosen.
it	the number of iterations performed during the (best) run.

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Vichi M., Saporta G. (2009) "Clustering and disjoint principal component analysis" <doi:10.1016/j.csda.2008.05.028>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# No constraint on variables
out <- dpcakm(iris, K = 3, Q = 2, Rndstart = 5)

# Constraint: the first two variables must contribute to the same factor.
outc <- dpcakm(iris, K = 3, Q = 2, Rndstart = 5, constr = c(1,1,0,0))
```

---

dpseudoF

*double pseudoF (Calinski-Harabasz) index*

---

**Description**

A pseudoF version for double partitioning, for the choice of the number of clusters of the units and variables (rows and columns of the data matrix). It is a diagnostic tool for inspecting simultaneously the optimal number of unit-clusters and variable-clusters.

**Usage**

```
dpseudoF(data, maxK, maxQ)
```

**Arguments**

data	Units x variables numeric data matrix.
maxK	Maximum number of clusters for the units to be tested.
maxQ	Maximum number of clusters for the variables to be tested.

**Value**

dpseudoF	matrix containing the pF value for each pair of K and Q within the specified range
----------	--

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

- R. Rocci, M. Vichi (2008) "Two-mode multi-partitioning" <doi:10.1016/j.csda.2007.06.025>  
T. Calinski & J. Harabasz (1974). A dendrite method for cluster analysis. Communications in Statistics, 3:1, 1-27

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

dpseudoF <- dpseudoF(iris, maxK=10, maxQ = 3)
```

---

factkm

*Factorial k-means*

---

**Description**

Performs simultaneously k-means partitioning on units and principal component analysis on the variables. Identifies the best partition in a Least-Squares sense in the best reduced space of the data. Both the data and the centroids are used to identify the best Least-Squares reduced subspace, where also their distances is measured.

**Usage**

```
factkm(X, K, Q, Rndstart, verbose, maxiter, tol, rot, prep, print)
```

**Arguments**

X	Units x variables numeric data matrix.
K	Number of clusters for the units.
Q	Number of principal components w.r.t. variables.
Rndstart	Number of runs to be performed (Defaults is 20).
verbose	Outputs basic summary statistics for each run (1 = enabled; 0 = disabled, default option).
maxiter	Maximum number of iterations allowed (if convergence is not yet reached. Default is 100).
tol	Tolerance threshold (maximum difference in the values of the objective function of two consecutive iterations such that convergence is assumed. Default is 1e-6).
rot	performs varimax rotation of axes obtained via PCA. (=1 enabled; =0 disabled, default option)
prep	Pre-processing of the data. 1 performs the z-score transform (default choice); 2 performs the min-max transform; 0 leaves the data un-pre-processed.
print	Prints summary statistics of the results (1 = enabled; 0 = disabled, default option).

**Value**

returns a list of estimates and some descriptive quantities of the final results.

U	Units x clusters membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which cluster each unit has been assigned.
A	Variables x components loading matrix (orthonormal).
centers	K x Q matrix of centers containing the row means expressed in the reduced space of Q principal components.
totss	The total sum of squares.
withinss	Vector of within-cluster sum of squares, one component per cluster.
betweenss	amount of deviance captured by the model.
size	Number of units assigned to each cluster.
pseudoF	Calinski-Harabasz index of the resulting partition.
loop	The index of the (best) run from which the results have been chosen.
it	the number of iterations performed during the (best) run.

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

- Vichi M., Kiers H.A.L. (2001) "Factorial k-means analysis for two-way data" <doi:10.1016/S0167-9473(00)00064-5>
- Kaiser H.F. (1958) "The varimax criterion for analytic rotation in factor analysis" <doi:10.1007/BF02289233>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# factorial k-means with 3 unit-clusters and 2 components for the variables
out <- factkm(iris, K = 3, Q = 2, Rndstart = 15, verbose = 0, maxiter = 100, tol = 1e-7, rot = 1)
```

---

heatm	<i>Heatmap of a partition in a reduced subspace</i>
-------	---

---

**Description**

Plots the heatmap of a partition on a reduced subspace obtained via either: doublekm, redkm, factkm or dpcakm.

**Usage**

```
heatm(data, drclust_out)
```

**Arguments**

data	Units x variables data matrix.
drclust_out	Out of either doublekm, redkm, factkm or dpcakm.

**Value**

No return value, called for side effects

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Kolde R. (2019) "pheatmap: Pretty Heatmaps" <<https://cran.r-project.org/web/packages/pheatmap/index.html>>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# standardizing the data
iris <- scale(iris)

# applying a clustering algorithm
```

```
drclust_out <- dpcakm(iris, 20, 3)

# obtain a heatmap based on the output of the clustering algorithm and the data
h <- heatm(iris, drclust_out)
```

---

kaiserCrit	<i>Selecting the number of principal components to be extracted from a dataset</i>
------------	--

---

### Description

Selects the optimal number of principal components to be extracted from a dataset based on Kaiser's criterion

### Usage

```
kaiserCrit(data)
```

### Arguments

data	Units x variables data matrix.
------	--------------------------------

### Value

bestQ	Number of components to be extracted (scalar).
-------	--

### Author(s)

Ionel Prunila, Maurizio Vichi

### References

Kaiser H. F. (1960) "The Application of Electronic Computers to Factor Analysis" <doi:10.1177/001316446002000>

### Examples

```
# Iris data
# Loading the numeric variables of iris data
iris <- scale(as.matrix(iris[,-5]))

# Apply the Kaiser rule
h <- kaiserCrit(iris)
```



---

mrand	<i>Adjusted Rand Index</i>
-------	----------------------------

---

**Description**

Performs the Adjusted Rand Index on a confusion matrix (row-by-column product of two partition-matrices). ARI is a measure of the similarity between two data clusterings.

**Usage**

```
mrand(N)
```

**Arguments**

N                    Confusion matrix.

**Value**

mri                    Adjusted Rand Index of a confusion matrix (scalar).

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Rand W. M. (1971) "Objective criteria for the evaluation of clustering methods" <doi:10.2307/2284239>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# standardizing the data
iris <- scale(iris)

# double k-means with 3 unit-clusters and 2 components for the variables
p1 <- redkm(iris, K = 3, Q = 2, Rndstart = 10)
p2 <- doublekm(iris, K=3, Q=2, Rndstart = 10)
mri <- mrand(t(p1$U)%*%p2$U)
```

---

redkm *k-means on a reduced subspace*

---

### Description

Performs simultaneously k-means partitioning on units and principal component analysis on the variables.

### Usage

```
redkm(X, K, Q, Rndstart, verbose, maxiter, tol, rot, prep, print)
```

### Arguments

X	Units x variables numeric data matrix.
K	Number of clusters for the units.
Q	Number of principal components w.r.t. variables.
Rndstart	Number of runs to be performed (Defaults is 20).
verbose	Outputs basic summary statistics for each run (1 = enabled; 0 = disabled, default option).
maxiter	Maximum number of iterations allowed (if convergence is not yet reached. Default is 100).
tol	Tolerance threshold (maximum difference between the values of the objective function of two consecutive iterations such that convergence is assumed. Default is 1e-6).
rot	performs varimax rotation of axes obtained via PCA. (=1 enabled; =0 disabled, default option)
prep	Pre-processing of the data. 1 performs the z-score transform (default choice); 2 performs the min-max transform; 0 leaves the data un-pre-processed.
print	Tolerancestats summary statistics of the performed method (1 = enabled; 0 = disabled, default option).

### Value

returns a list of estimates and some descriptive quantities of the final results.

U	Units x clusters membership matrix (binary and row-stochastic). Each row is a dummy variable indicating to which cluster each unit has been assigned.
A	Variables x components loading matrix (orthonormal).
centers	K x Q matrix of centers containing the row means expressed in the reduced space of Q principal components.
totss	The total sum of squares (scalar).
withinss	Vector of within-cluster sum of squares, one component per cluster.

betweenss	Amount of deviance captured by the model (scalar).
size	Number of units assigned to each cluster (vector).
pseudoF	Calinski-Harabasz index of the resulting partition (scalar).
loop	The index of the (best) run from which the results have been chosen.
it	the number of iterations performed during the (best) run.

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

- de Soete G., Carroll J. (1994) "K-means clustering in a low-dimensional Euclidean space" <doi:10.1007/978-3-642-51175-2\_24>
- Kaiser H.F. (1958) "The varimax criterion for analytic rotation in factor analysis" <doi:10.1007/BF02289233>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

# reduced k-means with 3 unit-clusters and 2 components for the variables
out <- redkm(iris, K = 3, Q = 2, Rndstart = 15, verbose = 0, maxiter = 100, tol = 1e-7, rot = 1)
```

---

silhouette	<i>Silhouette</i>
------------	-------------------

---

**Description**

Computes and plots the silhouette of a partition

**Usage**

```
silhouette(data, drclust_out)
```

**Arguments**

data	Units x variables data matrix.
drclust_out	Out of either doublekm, redkm, factkm or dpcakm.

**Value**

cl.silhouette	Silhouette index for the given partition, for each object (matrix).
fe.silhouette	Factoextra silhouette graphical object

**Author(s)**

Ionel Prunila, Maurizio Vichi

**References**

Rousseeuw P. J. (1987) "Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis" <doi:10.1016/0377-0427(87)90125-7>

Maechler M. et al. (2023) "cluster: Cluster Analysis Basics and Extensions" <<https://CRAN.R-project.org/package=cluster>>

Kassambara A. (2022) "factoextra: Extract and Visualize the Results of Multivariate Data Analyses" <<https://cran.r-project.org/web/packages/factoextra/index.html>>

**Examples**

```
# Iris data
# Loading the numeric variables of iris data
iris <- as.matrix(iris[,-5])

#standardizing the data
iris <- scale(iris)

#applying a clustering algorithm
drclust_out <- dpcakm(iris, 20, 3)

#silhouette based on the data and the output of the clustering algorithm
d <- silhouette(iris, drclust_out)
```

# Index

apseudoF, 2  
centree, 3  
cluster, 4  
CronbachAlpha, 5  
  
disfa, 6  
dispca, 7  
doublekm, 9  
dpcakm, 11  
dpseudoF, 12  
  
factkm, 13  
  
heatm, 15  
  
kaiserCrit, 16  
  
mrand, 17  
  
redkm, 18  
  
silhouette, 19