

# Package ‘clusterSim’

October 23, 2009

**Title** Searching for optimal clustering procedure for a data set

**Version** 0.36-6

**Date** 2009-10-23

**Author** Marek Walesiak <marek.walesiak@ue.wroc.pl> Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

**Maintainer** Andrzej Dudek <andrzej.dudek@ue.wroc.pl>

**Depends** ade4, cluster, R2HTML, e1071, rgl, MASS

**Description** GDM Distance, Sokal-Michener Distance, Bray-Curtis Distance, Calinski-Harabasz Index, G2 Index, G3 Index, Silhouette Index, Krzanowski-Lai Index, Hartigan Index, Gap Index, DB Index, Data Normalization, HINoV method, Replication analysis for cluster validation, Clustering with several algorithms, distances, normalizations and icq indices, Symbolic interval distances, Plot functions, Random cluster generation

**License** GPL (>= 2)

**URL** <http://keii.ue.wroc.pl/clusterSim>

**Encoding** UTF-8

**Repository** CRAN

**Date/Publication** 2009-10-23 11:00:36

## R topics documented:

cluster.Description	2
cluster.Gen	4
cluster.Sim	7
data.Normalization	10
data_binary	12
data_interval	12
data_mixed	13
data_nominal	13
data_ordinal	14

data_ratio . . . . .	14
data_symbolic . . . . .	15
dist.BC . . . . .	15
dist.GDM . . . . .	16
dist.SM . . . . .	17
dist.Symbolic . . . . .	18
HINoV.Mod . . . . .	19
HINoV.Symbolic . . . . .	21
index.DB . . . . .	23
index.G1 . . . . .	25
index.G2 . . . . .	27
index.G3 . . . . .	29
index.Gap . . . . .	31
index.H . . . . .	33
index.KL . . . . .	36
index.S . . . . .	38
initial.Centers . . . . .	40
plotCategorical . . . . .	41
plotCategorical3d . . . . .	43
plotInterval . . . . .	44
replication.Mod . . . . .	45
<b>Index</b>	<b>48</b>

---

cluster.Description

*Descriptive statistics calculated separately for each cluster and variable*

---

## Description

Descriptive statistics calculated separately for each cluster and variable: arithmetic mean and standard deviation, median and median absolute deviation, mode

## Usage

```
cluster.Description(x, cl, sdType="sample")
```

## Arguments

x	matrix or dataset
cl	a vector of integers indicating the cluster to which each object is allocated
sdType	type of standard deviation: for "sample" (n-1) or for "population" (n)

**Value**

Three-dimensional array:

First dimension contains cluster number

Second dimension contains original coordinate (variable) number from matrix or data set

Third dimension contains number from 1 to 5:

1 - arithmetic mean

2 - standard deviation

3 - median

4 - median absolute deviation (mad)

5 - mode (value of the variable which has the largest observed frequency. If several are tied, the N.A. value is returned. This formula is applicable for nominal and ordinal data only).

For example:

```
desc<-cluster.Description(x,cl)
```

```
desc[2,4,2] - standard deviation of fourth coordinate of second cluster
```

```
desc[3,1,5] - mode of first coordinate (variable) of third cluster
```

```
desc[1,,] - all statistics for all dimensions (variables) of first cluster
```

```
desc[,3] - medians of all dimensions (variables) for each cluster
```

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland

<http://keii.ue.wroc.pl/clusterSim>

**See Also**

[cluster.Sim](#), [mean](#), [sd](#), [median](#), [mad](#)

**Examples**

```
library(clusterSim)
data(data_ratio)
cl <- pam(data_ratio,5)
desc <- cluster.Description(data_ratio,cl$cluster)
print(desc)
```

cluster.Gen

*Random cluster generation with known structure of clusters***Description**

Random cluster generation with known structure of clusters (optionally with noisy variables and outliers)

**Usage**

```
cluster.Gen(numObjects=50, means=NULL, cov=NULL, fixedCov=TRUE,
            model=1, dataType="m", numCategories=NULL,
            numNoisyVar=0, numOutliers=0, rangeOutliers=
            c(1,10), inputType="csv2", inputHeader=TRUE,
            inputRowNames=TRUE, outputCsv="", outputCsv2="",
            outputColNames=TRUE, outputRowNames=TRUE)
```

**Arguments**

numObjects	number of objects in each cluster - positive integer value or vector with the same size as <i>nrow(means)</i> , e.g. <code>numObjects=c(50,20)</code>
means	matrix of cluster means (e.g. <code>means=matrix(c(0,8,0,8),2,2)</code> ). If <code>means = NULL</code> matrix should be read from <i>means_&lt;modelNumber&gt;.csv file</i>
cov	covariance matrix (the same for each cluster, e.g. <code>cov=matrix(c(1,0,0,1),2,2)</code> ). If <code>cov=NULL</code> matrix should be read from <i>cov_&lt;modelNumber&gt;.csv file</i> . Note: you cannot use this argument for generation of clusters with different covariance matrices. Those kind of generation should be done by setting <code>fixedCov</code> to <code>FALSE</code> and using appropriate model
model	model number, <code>model=1</code> - no cluster structure. Observations are simulated from uniform distribution over the unit hypercube in number of dimensions (variables) given in <code>numNoisyVar</code> argument; <code>model=2</code> - means and covariances are taken from arguments <code>means</code> and <code>cov</code> (see Example 1); <code>model=3,4,...,20</code> - see file <a href="#">\$R_HOME/library/clusterSim/pdf/clusterGen_details.pdf</a> ; <code>model=21,22,...</code> - if <code>fixedCov=TRUE</code> means should be read from <i>means_&lt;modelNumber&gt;.csv</i> and covariance matrix for all clusters should be read from <i>cov_&lt;modelNumber&gt;.csv</i> and if <code>fixedCov=FALSE</code> means should be read from <i>means_&lt;modelNumber&gt;.csv</i> and covariance matrices should be read separately for each cluster from <i>cov_&lt;modelNumber&gt;_&lt;clusterNumber&gt;.csv</i>
fixedCov	if <code>fixedCov=TRUE</code> covariance matrix for all clusters is the same and if <code>fixedCov=FALSE</code> each cluster is generated from different covariance matrix - see <code>model</code>

dataType	"m" - metric (ratio, interval), "o" - ordinal, "s" - symbolic interval
numCategories	number of categories (for ordinal data only). Positive integer value or vector with the same size as <i>ncol(means)</i> plus number of noisy variables.
numNoisyVar	number of noisy variables. For <i>model=1</i> it means number of variables
numOutliers	number of outliers (for metric and symbolic interval data only). If a positive integer - number of outliers, if value from $<0,1>$ - percentage of outliers in whole data set
rangeOutliers	range for outliers (for metric and symbolic interval data only). The default range is $[1, 10]$ . The outliers are generated independently for each variable for the whole data set from uniform distribution. The generated values are randomly added to maximum of <i>j</i> -th variable or subtracted from minimum of <i>j</i> -th variable
inputType	"csv" - a dot as decimal point or "csv2" - a comma as decimal point in <i>means_&lt;modelNumber&gt;.csv</i> and <i>cov_&lt;modelNumber&gt;.csv</i> files
inputHeader	<i>inputHeader=TRUE</i> indicates that input files ( <i>means_&lt;modelNumber&gt;.csv</i> ; <i>cov_&lt;modelNumber...&gt;.csv</i> ) contain header row
inputRowNames	<i>inputRowNames=TRUE</i> indicates that input files ( <i>means_&lt;modelNumber&gt;.csv</i> ; <i>cov_&lt;modelNumber...&gt;.csv</i> ) contain first column with row names or with number of objects (positive integer values)
outputCsv	optional, name of csv file with generated data (first column contains id, second - number of cluster and others - data)
outputCsv2	optional, name of csv (a comma as decimal point and a semicolon as field separator) file with generated data (first column contains id, second - number of cluster and others - data)
outputColNames	<i>outputColNames=TRUE</i> indicates that output file (given by <i>outputCsv</i> and <i>outputCsv2</i> parameters) contains first row with column names
outputRowNames	<i>outputRowNames=TRUE</i> indicates that output file (given by <i>outputCsv</i> and <i>outputCsv2</i> parameters) contains a vector of row names

## Details

See file `$R_HOME/library/clusterSim/pdf/clusterGen_details.pdf` for further details

## Value

clusters	cluster number for each object, for <i>model=1</i> each object belongs to its own cluster thus this variable contains objects numbers
data	generated data: for metric and ordinal data - matrix with objects in rows and variables in columns; for symbolic interval data three-dimensional structure: first dimension represents object number, second - variable number and third dimension contains lower- and upper-bounds of intervals

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

- Billard, L., Diday, E. (2006): *Symbolic data analysis. Conceptual statistics and data mining*, Wiley, Chichester.
- Qiu, W., Joe, H. (2006), *Generation of random clusters with specified degree of separation*, "Journal of Classification", vol. 23, 315-334.
- Steinley, D., Henson, R. (2005), *OCCLUS: an analytic method for generating clusters with known overlap*, "Journal of Classification", vol. 22, 221-250.
- Walesiak, M., Dudek, A. (2008), *Identification of noisy variables for nonmetric and symbolic data in cluster analysis*, In: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker (Eds.), *Data analysis, machine learning and applications*, Springer-Verlag, Berlin, Heidelberg, 85-92.

**Examples**

```
# Example 1
library(clusterSim)
means <- matrix(c(0,7,0,7),2,2)
cov <- matrix(c(1,0,0,1),2,2)
grnd <- cluster.Gen(numObjects=60,means=means,cov=cov,model=2,
numOutliers=8)
colornames <- c("red","blue","green")
grnd$clusters[grnd$clusters==0]<-length(colornames)
plot(grnd$data,col=colornames[grnd$clusters],ask=TRUE)

# Example 2
library(clusterSim)
grnd <- cluster.Gen(50,model=4,dataType="m",numNoisyVar=2)
data <- as.matrix(grnd$data)
colornames <- c("red","blue","green")
plot(grnd$data,col=colornames[grnd$clusters],ask=TRUE)

# Example 3
library(clusterSim)
grnd<-cluster.Gen(50,model=4,dataType="o",numCategories=7, numNoisyVar=2)
plotCategorical(grnd$data,,grnd$clusters,ask=TRUE)

# Example 4 (1 nonnoisy variable and 2 noisy variables, 3 clusters)
library(clusterSim)
grnd <- cluster.Gen(c(40,60,20), model=2, means=c(2,14,25),
cov=c(1.5,1.5,1.5),numNoisyVar=2)
colornames <- c("red","blue","green")
plot(grnd$data,col=colornames[grnd$clusters],ask=TRUE)

# Example 5
```

```

library(clusterSim)
grnd <- cluster.Gen(c(20,35,20,25),model=14,dataType="m",numNoisyVar=1,
fixedCov=FALSE, numOutliers=0.1, outputCsv2="data14.csv")
data <- as.matrix(grnd$data)
colornames <- c("red", "blue", "green", "brown", "black")
grnd$clusters[grnd$clusters==0]<-length(colornames)
plot(grnd$data,col=colornames[grnd$clusters],ask=TRUE)

# Example 6 (this example needs files means_24.csv)
# and cov_24.csv to be placed in working directory
# library(clusterSim)
# grnd<-cluster.Gen(c(50,80,20),model=24,dataType="m",numNoisyVar=1,
# numOutliers=10, rangeOutliers=c(1,5))
# print(grnd)
# data <- as.data.frame(grnd$data)
# colornames<-c("red", "blue", "green", "brown")
# grnd$clusters[grnd$clusters==0]<-length(colornames)
# plot(data,col=colornames[grnd$clusters],ask=TRUE)

# Example 7 (this example needs files means_25.csv and cov_25_1.csv)
# cov_25_2.csv, cov_25_3.csv, cov_25_4.csv, cov_25_5.csv
# to be placed in working directory
# library(clusterSim)
# grnd<-cluster.Gen(c(40,30,20,35,45),model=25,numNoisyVar=3,fixedCov=F)
# data <- as.data.frame(grnd$data)
# colornames<-c("red", "blue", "green", "magenta", "brown")
# plot(data,col=colornames[grnd$clusters],ask=TRUE)

```

---

cluster.Sim

*Determination of optimal clustering procedure for a data set*


---

## Description

Determination of optimal clustering procedure for a data set by varying all combinations of normalization formulas, distance measures, and clustering methods

## Usage

```

cluster.Sim(x,p,minClusterNo,maxClusterNo,icq="S",outputHtml="",
outputCsv="",outputCsv2="",normalizations=NULL,
distances=NULL,methods=NULL)

```

## Arguments

x	matrix or dataset
p	path of simulation: 1 - ratio data, 2 - interval or mixed (ratio & interval) data, 3 - ordinal data, 4 - nominal data, 5 - binary data, 6 - ratio data without normalization, 7 - interval or mixed (ratio & interval) data without normalization, 8 - ratio data with k-means, 9 - interval or mixed (ratio & interval) data with k-means

<code>minClusterNo</code>	minimal number of clusters, between 2 and no. of objects - 1 (for G3: no. of objects - 2)
<code>maxClusterNo</code>	maximal number of clusters, between 2 and no. of objects - 1 (for G3: no. of objects - 2; for KL: no. of objects - 3), greater or equal <code>minClusterNo</code>
<code>icq</code>	Internal cluster quality index, "S" - Silhouette,"G1" - Calinski & Harabasz index, "G2" - Baker & Hubert index ,"G3" - Hubert & Levine index, "KL" - Krzanowski & Lai index
<code>outputHtml</code>	optional, name of html file with results
<code>outputCsv</code>	optional, name of csv file with results
<code>outputCsv2</code>	optional, name of csv (comma as decimal point sign) file with results
<code>normalizations</code>	optional, vector of normalization formulas that should be used in procedure
<code>distances</code>	optional, vector of distance measures that should be used in procedure
<code>methods</code>	optional, vector of classification methods that should be used in procedure

## Details

Parameter `normalizations` for each path may be the subset of the following values

path 1: "n6" to "n11" (if measurement scale of variables is ratio and transformed measurement scale of variables is ratio) or "n1" to "n5" (if measurement scale of variables is ratio and transformed measurement scale of variables is interval)

path 2: "n1" to "n5"

path 3 to 7: "n0"

path 8: "n1" to "n11"

path 9: "n1" to "n5"

Parameter `distances` for each path may be the subset of the following values

path 1: "d1" to "d7" (if measurement scale of variables is ratio and transformed measurement scale of variables is ratio) or "d1" to "d5" (if measurement scale of variables is ratio and transformed measurement scale of variables is interval)

path 2: "d1" to "d5"

path 3: "d8"

path 4: "d9"

path 5: "b1" to "b10"

path 6: "d1" to "d7"

path 7: "d1" to "d5"

path 8 and 9: N.A.

Parameter `methods` for each path may be the subset of the following values

path 1 to 7: "m1" to "m8"

path 8: "m9"

path 9: "m9"

See file `$R_HOME/library/clusterSim/pdf/clusterSim_details.pdf` for further details

**Value**

result	optimal value of icq for all classifications
normalization	normalization used to obtain optimal value of icq
distance	distance measure used to obtain optimal value of icq
method	clustering method used to obtain optimal value of icq
classes	number of clusters for optimal value of icq
optClustering	clustering vector for optimal value of icq
optClusteringDescription	description of clusters for optimal value of icq (see <a href="#">cluster.Description</a> )
time	time of all calculations for path

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

- Everitt, B.S., Landau, E., Leese, M. (2001), *Cluster analysis*, Arnold, London.
- Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław.
- Gordon, A.D. (1999), *Classification*, Chapman & Hall/CRC, London.
- Milligan, G.W., Cooper, M.C. (1985), *An examination of procedures of determining the number of cluster in a data set*, "Psychometrika", vol. 50, no. 2, 159-179.
- Milligan, G.W., Cooper, M.C. (1988), *A study of standardization of variables in cluster analysis*, "Journal of Classification", vol. 5, 181-204.
- Walesiak, M., Dudek, A. (2006), *Symulacyjna optymalizacja wyboru procedury klasyfikacyjnej dla danego typu danych - oprogramowanie komputerowe i wyniki badan*, Prace Naukowe AE we Wrocławiu, 1126, 120-129.
- Walesiak, M., Dudek, A. (2007), *Symulacyjna optymalizacja wyboru procedury klasyfikacyjnej dla danego typu danych - charakterystyka problemu*, Zeszyty Naukowe Uniwersytetu Szczecińskiego nr 450, 635-646.

**See Also**

[data.Normalization](#), [dist.GDM](#), [dist.BC](#), [dist.SM](#), [index.G1](#), [index.G2](#),  
[index.G3](#), [index.S](#), [index.KL](#), [hclust](#), [dist](#),

**Examples**

```

library(clusterSim)
data(data_ratio)
cluster.Sim(data_ratio, 1, 2, 10, "G1",
outputCsv="results1")
data(data_interval)
cluster.Sim(data_interval, 2, 2, 15, "G1",
outputHtml="results2")
data(data_ordinal)
cluster.Sim(data_ordinal, 3, 2, 12, "G2",
outputCsv2="results3")
data(data_nominal)
cluster.Sim(data_nominal, p=4, 2, 5, icq="G3",
outputHtml="results4", methods=c("m2", "m3", "m5"))
data(data_binary)
cluster.Sim(data_binary, p=5, 2, 6, icq="S",
outputHtml="results5", distances=c("b1", "b3", "b6"))
data(data_ratio)
cluster.Sim(data_ratio, 1, 2, 10, "G1",
outputCsv="results6", normalizations=c("n1", "n3"),
distances=c("d2", "d5"), methods=c("m5", "m3", "m1"))

```

---

data.Normalization *Types of variable normalization formulas*

---

**Description**

Types of variable normalization formulas

**Usage**

```
data.Normalization (x, type="n0")
```

**Arguments**

x                    vector, matrix or dataset

type                type of normalization:

n0 - without normalization

n1 - standardization  $((x-\text{mean})/\text{sd})$

n2 - Weber standardization  $((x-\text{Me})/\text{MAD})$

n3 - unitization  $((x-\text{mean})/\text{range})$

n4 - unitization with zero minimum  $((x-\text{min})/\text{range})$

n5 - normalization in range  $<-1,1>$   $((x-\text{mean})/\text{max}(\text{abs}(x-\text{mean})))$

n6 - quotient transformation  $(x/\text{sd})$

n7 - quotient transformation  $(x/\text{range})$

- n8 - quotient transformation (x/max)
- n9 - quotient transformation (x/mean)
- n10 - quotient transformation (x/sum)
- n11 - quotient transformation (x/sqrt(SSQ))

### Details

See file `$R_HOME/library/clusterSim/pdf/dataNormalization_details.pdf` for further details

### Value

Normalized data

### Author(s)

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

### References

Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław, 35-38.

Jajuga, K., Walesiak, M. (2000), *Standardisation of data set under different measurement scales*, In: R. Decker, W. Gaul (Eds.), *Classification and information processing at the turn of the millennium*, Springer-Verlag, Berlin, Heidelberg, 105-112.

Milligan, G.W., Cooper, M.C. (1988), *A study of standardization of variables in cluster analysis*, "Journal of Classification", vol. 5, 181-204.

### See Also

[cluster.Sim](#)

### Examples

```
library(clusterSim)
data(data_ratio)
z1 <- data.Normalization(data_ratio, "n1")
z2 <- data.Normalization(data_ratio, type="n2")
z8 <- data.Normalization(data_ratio, type="n8")
```

---

`data_binary`*Binary data*

---

**Description**

Binary variables for eight people

**Format**

data.frame: 8 objects, 10 variables

**Source**

Kaufman, L., Rousseeuw, P.J. (1990), *Finding groups in data: an introduction to cluster analysis*, Wiley, New York, p. 24.

**Examples**

```
library(clusterSim)
data(data_binary)
cluster.Sim(data_binary, p=5, 2, 6, icq="S",
outputHtml="results5", distances=c("b1", "b3", "b6"))
```

---

`data_interval`*Interval data*

---

**Description**

Artificially generated interval data

**Format**

data.frame: 75 objects, 5 variables, 5-class structure

**Source**

Artificially generated data

**Examples**

```
library(clusterSim)
data(data_interval)
cluster.Sim(data_interval, 2, 2, 15, "G1",
outputHtml="results2")
```

---

data_mixed	<i>Mixed data</i>
------------	-------------------

---

**Description**

Artificial mixed data

**Format**

data.frame: 25 objects, 4 variables (1, 3 - interval variables, 2 - ordinal variable, 4, nominal variable)

**Source**

Artificial data

**Examples**

```
library(clusterSim)
data(data_mixed)
r3 <- HINoV.Mod(data_mixed, type=c("m","n","m","n"), s=2, 3, distance="d1",
  method="complete", Index="cRAND")
print(r3$stopri)
plot(r3$stopri[,2], xlab="Variable number", ylab="topri", xaxt="n")
axis(1, at=c(1:max(r3$stopri[,1])), labels=r3$stopri[,1])
```

---

data_nominal	<i>Nominal data</i>
--------------	---------------------

---

**Description**

Artificial nominal data

**Format**

data.frame: 26 objects, 12 variables

**Source**

Artificial data

**Examples**

```
library(clusterSim)
data(data_nominal)
cluster.Sim(data_nominal, p=4, 2, 5, icq="G3",
  outputHtml="results4", methods=c("m2","m3","m5"))
```

---

data\_ordinal      *Ordinal data*

---

**Description**

Artificial ordinal data

**Format**

data.frame: 26 objects, 12 variables

**Source**

Artificial data

**Examples**

```
library(clusterSim)
data(data_ordinal)
cluster.Sim(data_ordinal, 3, 3, 12, "S",
outputCsv2="results3")
```

---

data\_ratio      *Ratio data*

---

**Description**

Artificially generated ratio data

**Format**

data.frame: 75 objects, 5 variables, 5-class structure

**Source**

Artificially generated data

**Examples**

```
library(clusterSim)
data(data_ratio)
c <- pam(data_ratio, 10)
index.G1(data_ratio, c$clustering)
```

---

data_symbolic	<i>Symbolic interval data</i>
---------------	-------------------------------

---

**Description**

Artificially generated symbolic interval data

**Format**

3-dimensional array: 125 objects, 6 variables, third dimension represents beginning and end of interval, 5-class structure

**Source**

Artificially generated data

**Examples**

```
library(clusterSim)
data(data_symbolic)
r<- HINoV.Symbolic(data_symbolic, u=5)
print(r$stopri)
plot(r$stopri[,2], xlab="Variable number", ylab="topri",
      xaxt="n", type="b")
axis(1, at=c(1:max(r$stopri[,1])), labels=r$stopri[,1])
```

---

dist.BC	<i>Calculates Bray-Curtis distance measure for ratio data</i>
---------	---

---

**Description**

Calculates Bray-Curtis distance measure for ratio data

**Usage**

```
dist.BC (x)
```

**Arguments**

x                    matrix or dataset

**Details**

See file `$R_HOME/library/clusterSim/pdf/distBC_details.pdf` for further details

**Value**

object with calculated distance

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Cormack, R.M. (1971), *A review of classification (with discussion)*, "Journal of the Royal Statistical Society", ser. A, part 3, 321-367.

Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław, p. 41.

**See Also**

[dist.GDM](#), [dist.SM](#), [dist](#)

**Examples**

```
library(clusterSim)
sampleData <- cbind(c(2, 3, 5), c(4, 5, 6), c(5, 3, 4))
d <- dist.BC(sampleData)
```

---

dist.GDM

*Calculates Generalized Distance Measure*

---

**Description**

Calculates Generalized Distance Measure for variables measured on metric scale (ratio & interval) or ordinal scale.

**Usage**

```
dist.GDM (x, method="GDM1")
GDM (x, method="GDM1")
GDM1 (x)
GDM2 (x)
```

**Arguments**

x                   matrix or data set  
 method             GDM1 or GDM2  
 "GDM1" - metric scale (ratio & interval)  
 "GDM2" - ordinal scale

**Details**

See file `$R_HOME/library/clusterSim/pdf/distGDM_details.pdf` for further details

**Value**

object with calculated distance

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Jajuga, K., Walesiak, M., Bak, A. (2003), *On the general distance measure*, In: M. Schwaiger, O. Opitz (Eds.), *Exploratory data analysis in empirical research*, Springer-Verlag, Berlin, Heidelberg, 104-109.

Walesiak, M. (2006), *Uogólniona miara odległości w statystycznej analizie wielowymiarowej [The Generalized Distance Measure in multivariate statistical analysis]*, Wydawnictwo AE, Wrocław.

**See Also**

[dist.BC](#), [dist.SM](#), [dist](#)

**Examples**

```
library(clusterSim)
data(data_ratio)
d1 <- GDM(data_ratio, method="GDM1")
data(data_ordinal)
d2 <- GDM(data_ordinal, method="GDM2")
d3 <- GDM1(data_ratio)
d4 <- GDM2(data_ordinal)
```

---

`dist.SM`

*Calculates Sokal-Michener distance measure for nominal variables*

---

**Description**

Calculates Sokal-Michener distance measure for nominal variables

**Usage**

```
dist.SM(x)
```

**Arguments**

x                    matrix or data set

**Details**

See file `$R_HOME/library/clusterSim/pdf/distSM_details.pdf` for further details

**Value**

object with calculated distance

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław, p. 43.

Kaufman, L., Rousseeuw, P.J. (1990), *Finding groups in data: an introduction to cluster analysis*, Wiley, New York, p. 28.

**See Also**

[dist.GDM](#), [dist.BC](#), [dist](#)

**Examples**

```
library(clusterSim)
data(data_nominal)
d <- dist.SM(data_nominal)
```

---

dist.Symbolic

*Calculates distance between symbolic interval-valued data*

---

**Description**

Calculates distance between symbolic interval-valued data for four distance types

**Usage**

```
dist.Symbolic(data, type="U_2", gamma=0.5, power=2)
```

**Arguments**

data	symbolic data
type	type of distance used for symbolic interval-valued data U_2 - Ichino and Yaguchi distance M - distance between points given by means of intervals (for interval-values variables), H - Hausdorff distance, S - sum of distances between all corresponding vertices of hyperrectangles given by symbolic objects with interval-valued variables)
gamma	parameter for calculating Ichino and Yaguchi distance
power	parameter (q) for calculating Ichino and Yaguchi distance

**Details**

See file `$R_HOME\library\clusterSim\pdf\distSymbolic_details.pdf` for further details

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Billard, L., Diday, E. (2006), *Symbolic data analysis. Conceptual statistics and data mining*, Wiley, Chichester.

**See Also**

[dist.SM](#)

**Examples**

```
library(clusterSim)
dataSymbolic<-cluster.Gen(numObjects=10,model=5,dataType="s")$data
print(dist.Symbolic(dataSymbolic))
```

---

HINoV.Mod

*Modification of Carmone, Kara & Maxwell Heuristic Identification of Noisy Variables (HINoV) method*

---

**Description**

Modification of Heuristic Identification of Noisy Variables (HINoV) method

**Usage**

```
HINoV.Mod (x, type="metric", s = 2, u, distance=NULL,
          method = "kmeans", Index ="cRAND")
```

**Arguments**

x	data matrix
type	"metric" (default) - all variables are metric (ratio, interval), "nonmetric" - all variables are nonmetric (ordinal, nominal) or vector containing for each variable value "m"(metric) or "n"(nonmetric) for mixed variables (metric and nonmetric), e.g. type=c("m", "n", "n", "m")
s	for metric data only: 1 - ratio data, 2 - interval or mixed (ratio & interval) data
u	number of clusters (for metric data only)
distance	NULL for kmeans method (based on data matrix) and nonmetric data for ratio data: "d1" - Manhattan, "d2" - Euclidean, "d3" - Chebychev (max), "d4" - squared Euclidean, "d5" - GDM1, "d6" - Canberra, "d7" - Bray-Curtis for interval or mixed (ratio & interval) data: "d1", "d2", "d3", "d4", "d5"
method	NULL for nonmetric data clustering method: "kmeans" (default) , "single", "ward", "complete", "average", "mcquitty", "median", "centroid", "pam"
Index	"cRAND" - corrected Rand index (default); "RAND" - Rand index

**Details**

See file \$R\_HOME\library\clusterSim\pdf\HINoVMod\_details.pdf for further details

**Value**

parim	$m \times m$ symmetric matrix ( $m$ - number of variables). Matrix contains pairwise corrected Rand (Rand) indices for partitions formed by the $j$ -th variable with partitions formed by the $l$ -th variable
topri	sum of rows of parim
stopri	ranked values of topri in decreasing order

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

- Carmone, F.J., Kara, A., Maxwell, S. (1999), *HINoV: a new method to improve market segment definition by identifying noisy variables*, "Journal of Marketing Research", November, vol. 36, 501-509.
- Hubert, L.J., Arabie, P. (1985), *Comparing partitions*, "Journal of Classification", no. 1, 193-218.

Rand, W.M. (1971), *Objective criteria for the evaluation of clustering methods*, "Journal of the American Statistical Association", no. 336, 846-850.

Walesiak, M. (2005), *Variable selection for cluster analysis - approaches, problems, methods*, Plenary Session of the Committee on Statistics and Econometrics of the Polish Academy of Sciences, 15 March, Wroclaw.

### See Also

[hclust](#), [kmeans](#), [dist](#), [dist.GDM](#), [dist.BC](#), [dist.SM](#), [cluster.Sim](#)

### Examples

```
# for metric data
library(clusterSim)
data(data_ratio)
r1<- HINoV.Mod(data_ratio, type="metric", s=1, 4, method="kmeans",
  Index="cRAND")
print(r1$stopri)
plot(r1$stopri[,2], xlab="Variable number", ylab="topri",
  xaxt="n", type="b")
axis(1, at=c(1:max(r1$stopri[,1])), labels=r1$stopri[,1])

# for nonmetric data
library(clusterSim)
data(data_nominal)
r2<- HINoV.Mod (data_nominal, type="nonmetric", Index = "cRAND")
print(r2$stopri)
plot(r2$stopri[,2], xlab="Variable number", ylab="topri",
  xaxt="n", type="b")
axis(1, at=c(1:max(r2$stopri[,1])), labels=r2$stopri[,1])

# for mixed data
library(clusterSim)
data(data_mixed)
r3<- HINoV.Mod(data_mixed, type=c("m", "n", "m", "n"), s=2, 3, distance="d1",
  method="complete", Index="cRAND")
print(r3$stopri)
plot(r3$stopri[,2], xlab="Variable number", ylab="topri",
  xaxt="n", type="b")
axis(1, at=c(1:max(r3$stopri[,1])), labels=r3$stopri[,1])
```

---

HINoV.Symbolic

*Modification of Carmone, Kara & Maxwell Heuristic Identification of Noisy Variables (HINoV) method for symbolic interval data*

---

### Description

Modification of Heuristic Identification of Noisy Variables (HINoV) method for symbolic interval data

**Usage**

```
HINoV.Symbolic(x, u=NULL, distance="H", method = "pam",
              Index = "cRAND")
```

**Arguments**

x	symbolic interval data: a 3-dimensional table, first dimension represents object number, second dimension - variable number, and third dimension contains lower- and upper-bounds of intervals
u	number of clusters
distance	"M" - minimal distance between all vertices of hyper-cubes defined by symbolic interval variables; "H" - Hausdorff distance; "S" - sum of squares of distance between all vertices of hyper-cubes defined by symbolic interval variables
method	clustering method: "single", "ward", "complete", "average", "mcquitty", "median", "centroid", "pam" (default)
Index	"cRAND" - corrected Rand index (default); "RAND" - Rand index

**Details**

See file `$R_HOME\library\clusterSim\pdf\HINoVSymbolic_details.pdf` for further details

**Value**

parim	$m \times m$ symmetric matrix ( $m$ - number of variables). Matrix contains pairwise corrected Rand (Rand) indices for partitions formed by the $j$ -th variable with partitions formed by the $l$ -th variable
topri	sum of rows of parim
stopri	ranked values of topri in decreasing order

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Carmone, F.J., Kara, A., Maxwell, S. (1999), *HINoV: a new method to improve market segment definition by identifying noisy variables*, "Journal of Marketing Research", November, vol. 36, 501-509.

Hubert, L.J., Arabie, P. (1985), *Comparing partitions*, "Journal of Classification", no. 1, 193-218.

Rand, W.M. (1971), *Objective criteria for the evaluation of clustering methods*, "Journal of the American Statistical Association", no. 336, 846-850.

Walesiak, M., Dudek, A. (2008), *Identification of noisy variables for nonmetric and symbolic data in cluster analysis*, In: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, R. Decker (Eds.), *Data analysis, machine learning and applications*, Springer-Verlag, Berlin, Heidelberg, 85-92.

**See Also**

[hclust](#), [kmeans](#), [cluster.Sim](#)

**Examples**

```
library(clusterSim)
data(data_symbolic)
r<- HINoV.Symbolic(data_symbolic, u=5)
print(r$stopri)
plot(r$stopri[,2], xlab="Variable number", ylab="topri",
     xaxt="n", type="b")
axis(1,at=c(1:max(r$stopri[,1])),labels=r$stopri[,1])

#symbolic data from .csv file
#library(clusterSim)
#dsym<-as.matrix(read.csv2(file="csv/symbolic.csv"))
#dim(dsym)<-c(dim(dsym)[1],dim(dsym)[2]%/%2,2)
#r<- HINoV.Symbolic(dsym, u=5)
#print(r$stopri)
#plot(r$stopri[,2], xlab="Variable number", ylab="topri",
#     #xaxt="n", type="b")
#axis(1,at=c(1:max(r$stopri[,1])),labels=r$stopri[,1])
```

---

 index.DB

*Calculates Davies-Bouldin's index*


---

**Description**

Calculates Davies-Bouldin's cluster separation measure

**Usage**

```
index.DB(x, cl, d=NULL, centrotypes="centroids", p=2, q=2)
```

**Arguments**

x	data
cl	vector of integers indicating the cluster to which each object is allocated
d	optional distance matrix, used for calculations if centrotypes="medoids"
centrotypes	"centroids" or "medoids"
p	the power of the Minkowski distance between centroids or medoids of clusters: p=1 - Manhattan distance; p=2 - Euclidean distance
q	the power of dispersion measure of a cluster: q=1 - the average distance of objects in the r-th cluster to the centroid or medoid of the r-th cluster; q=2 - the standard deviation of the distance of objects in the r-th cluster to the centroid or medoid of the r-th cluster

**Details**

See file `$R_HOME\library\clusterSim\pdf\indexDB_details.pdf` for further details

**Value**

DB	Davies-Bouldin's index
r	vector of maximal R values for each cluster
R	R matrix $(S_r + S_s)/d_{rs}$
d	matrix of distances between centroids or medoids of clusters
S	vector of dispersion measures for each cluster
centers	coordinates of centroids or medoids for all clusters

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Davies, D.L., Bouldin, D.W. (1979), *A cluster separation measure*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 1, no. 2, 224-227.

**See Also**

[index.G1](#), [index.G2](#), [index.G3](#), [index.S](#), [index.H](#), [index.Gap](#), [index.KL](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
cl1 <- pam(data_ratio, 4)
d<-dist(data_ratio)
print(index.DB(data_ratio, cl1$clustering,d, centrotypes="medoids"))

# Example 2
library(clusterSim)
data(data_ratio)
cl2 <- pam(data_ratio, 5)
print(index.DB(data_ratio, cl2$clustering, centrotypes="centroids"))

# Example 3
library(clusterSim)
data(data_ratio)
md <- dist(data_ratio, method="euclidean")
# nc - number_of_clusters
min_nc=2
max_nc=8
```

```

res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
hc <- hclust(md, method="complete")
cl2 <- cutree(hc, k=nc)
res[nc-min_nc+1, 2] <- DB <- index.DB(data_ratio, cl2, centrotypes="centroids")$DB
clusters <- rbind(clusters, cl2)
}
print(paste("min DB for", (min_nc:max_nc)[which.min(res[,2])], "clusters=", min(res[,2])))
print("clustering for min DB")
print(clusters[which.min(res[,2]),])
write.table(res, file="DB_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="DB", xaxt="n")
axis(1, c(min_nc:max_nc))

# Example 4
library(clusterSim)
data(data_ordinal)
md <- dist.GDM(data_ordinal, method="GDM2")
# nc - number_of_clusters
min_nc=2
max_nc=6
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
hc <- hclust(md, method="complete")
cl2 <- cutree(hc, k=nc)
res[nc-min_nc+1, 2] <- DB <- index.DB(data_ordinal, cl2, d=md, centrotypes="medoids")$DB
clusters <- rbind(clusters, cl2)
}
print(paste("min DB for", (min_nc:max_nc)[which.min(res[,2])], "clusters=", min(res[,2])))
print("clustering for min DB")
print(clusters[which.min(res[,2]),])
write.table(res, file="DB_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="DB", xaxt="n")
axis(1, c(min_nc:max_nc))

```

---

index.G1

*Calculates Calinski-Harabasz pseudo F-statistic*


---

### Description

Calculates Calinski-Harabasz pseudo F-statistic

### Usage

```
index.G1 (x, cl, d=NULL, centrotypes="centroids")
```

**Arguments**

x	data
c1	A vector of integers indicating the cluster to which each object is allocated
d	optional distance matrix, used for calculations if centrotypes="medoids"
centrotypes	"centroids" or "medoids"

**Details**

See file `$R_HOME/library/clusterSim/pdf/indexG1_details.pdf` for further details

**Value**

Calinski-Harabasz pseudo F-statistic

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

- Calinski, R.B., Harabasz, J. (1974), *A dendrite method for cluster analysis*, "Communications in Statistics", vol. 3, 1-27.
- Everitt, B.S., Landau, E., Leese, M. (2001), *Cluster analysis*, Arnold, London, p. 103.
- Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław, p. 338.
- Gordon, A.D. (1999), *Classification*, Chapman & Hall/CRC, London, p. 62.
- Milligan, G.W., Cooper, M.C. (1985), *An examination of procedures of determining the number of cluster in a data set*, "Psychometrika", vol. 50, no. 2, 159-179.

**See Also**

[index.G2](#), [index.G3](#), [index.S](#), [index.H](#), [index.KL](#), [index.Gap](#), [index.DB](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
c<- pam(data_ratio,10)
index.G1(data_ratio,c$clustering)

# Example 2
library(clusterSim)
data(data_ratio)
```

```

md <- dist(data_ratio, method="euclidean")
# nc - number_of_clusters
min_nc=2
max_nc=20
res <- array(0,c(max_nc-min_nc+1,2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  cl2 <- pam(md, nc, diss=TRUE)
  res[nc-min_nc+1,2] <- G1 <- index.G1(data_ratio,cl2$cluster,centrotypes="centroids")
  clusters <- rbind(clusters, cl2$cluster)
}
print(paste("max G1 for", (min_nc:max_nc)[which.max(res[,2])], "clusters=", max(res[,2])))
print("clustering for max G1")
print(clusters[which.max(res[,2]),])
write.table(res, file="G1_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="G1", xaxt="n")
axis(1, c(min_nc:max_nc))

# Example 3
library(clusterSim)
data(data_ratio)
md <- dist(data_ratio, method="euclidean")
# nc - number_of_clusters
min_nc=2
max_nc=20
res <- array(0,c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  cl2 <- pam(md, nc, diss=TRUE)
  res[nc-min_nc+1,2] <- G1 <- index.G1(data_ratio,cl2$cluster,d=md,centrotypes="medoids")
  clusters <- rbind(clusters, cl2$cluster)
}
print(paste("max G1 for", (min_nc:max_nc)[which.max(res[,2])], "clusters=", max(res[,2])))
print("clustering for max G1")
print(clusters[which.max(res[,2]),])
write.table(res, file="G1_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="G1", xaxt="n")
axis(1, c(min_nc:max_nc))

```

---

index.G2

*Calculates G2 internal cluster quality index*


---

### Description

Calculates G2 internal cluster quality index - Baker & Hubert adaptation of Goodman & Kruskal's Gamma statistic

**Usage**

```
index.G2(d, cl)
```

**Arguments**

d	'dist' object
cl	A vector of integers indicating the cluster to which each object is allocated

**Details**

See file `$R_HOME\library\clusterSim\pdf\indexG2_details.pdf` for further details

**Value**

calculated G2 index

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

- Everitt, B.S., Landau, E., Leese, M. (2001), *Cluster analysis*, Arnold, London, p. 104.
- Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław, p. 339.
- Gordon, A.D. (1999), *Classification*, Chapman & Hall/CRC, London, p. 62.
- Hubert, L. (1974), *Approximate evaluation technique for the single-link and complete-link hierarchical clustering procedures*, "Journal of the American Statistical Association", vol. 69, no. 347, 698-704.
- Milligan, G.W., Cooper, M.C. (1985), *An examination of procedures of determining the number of cluster in a data set*, "Psychometrika", vol. 50, no. 2, 159-179.

**See Also**

[index.G1](#), [index.G3](#), [index.S](#), [index.H](#), [index.KL](#), [index.Gap](#), [index.DB](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
d <- dist.GDM(data_ratio)
c <- pam(d, 5, diss = TRUE)
icq <- index.G2(d, c$clustering)
print(icq)
```

```

# Example 2
library(clusterSim)
data(data_ordinal)
d <- dist.GDM(data_ordinal, method="GDM2")
# nc - number_of_clusters
min_nc=2
max_nc=6
res <- array(0,c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  cl2 <- pam(d, nc, diss=TRUE)
  res[nc-min_nc+1,2] <- G2 <- index.G2(d,cl2$cluster)
  clusters <- rbind(clusters,cl2$cluster)
}
print(paste("max G2 for", (min_nc:max_nc)[which.max(res[,2])], "clusters=", max(res[,2])))
print("clustering for max G2")
print(clusters[which.max(res[,2]),])
write.table(res, file="G2_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="G2", xaxt="n")
axis(1, c(min_nc:max_nc))

```

---

index.G3

*Calculates G3 internal cluster quality index*


---

### Description

Calculates G3 Hubert & Levine internal cluster quality index

### Usage

```
index.G3(d, cl)
```

### Arguments

d	'dist' object
cl	A vector of integers indicating the cluster to which each object is allocated

### Details

See file `$R_HOME/library/clusterSim/pdf/indexG3_details.pdf` for further details

### Value

calculated G3 index

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wrocław, p. 339.

Gordon, A.D. (1999), *Classification*, Chapman & Hall/CRC, London, p. 62.

Milligan, G.W., Cooper, M.C. (1985), *An examination of procedures of determining the number of cluster in a data set*, "Psychometrika", vol. 50, no. 2, 159-179.

**See Also**

[index.G1](#), [index.G2](#), [index.S](#), [index.H](#), [index.KL](#), [index.Gap](#), [index.DB](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
d <- dist.GDM(data_ratio)
c <- pam(d, 5, diss = TRUE)
icq <- index.G3(d, c$clustering)
print(icq)

# Example 2
library(clusterSim)
data(data_ordinal)
d <- dist.GDM(data_ordinal, method="GDM2")
# nc - number_of_clusters
min_nc=2
max_nc=6
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  hc <- hclust(d, method="complete")
  cl2 <- cutree(hc, k=nc)
  res[nc-min_nc+1,2] <- G3 <- index.G3(d, cl2)
  clusters <- rbind(clusters, cl2)
}
print(paste("min G3 for", (min_nc:max_nc)[which.min(res[,2])], "clusters=", min(res[,2])))
print("clustering for min G3")
print(clusters[which.min(res[,2]),])
write.table(res, file="G3_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="G3", xaxt="n")
```

```
axis(1, c(min_nc:max_nc))
```

---

index.Gap *Calculates Tibshirani, Walther and Hastie gap index*

---

### Description

Calculates Tibshirani, Walther and Hastie gap index

### Usage

```
index.Gap(x, clall, reference.distribution="unif", B=10,
          method="pam", d=NULL, centrotypes="centroids")
```

### Arguments

x	data
clall	Two vectors of integers indicating the cluster to which each object is allocated in partition of n objects into u, and u+1 clusters
reference.distribution	"unif" - generate each reference variable uniformly over the range of the observed values for that variable or "pc" - generate the reference variables from a uniform distribution over a box aligned with the principal components of the data. In detail, if $X = \{x_{ij}\}$ is our n x m data matrix, assume that the columns have mean 0 and compute the singular value decomposition $X = UDV^T$ . We transform via $X' = XV$ and then draw uniform features $Z'$ over the ranges of the columns of $X'$ , as in method a) above. Finally we back-transform via $Z = Z'V^T$ to give reference data Z
B	the number of simulations used to compute the gap statistic
method	the cluster analysis method to be used. This should be one of: "ward", "single", "complete", "average", "mcquitty", "median", "centroid", "pam", "k-means", "diana"
d	optional distance matrix, used for calculations if centrotypes="medoids"
centrotypes	"centroids" or "medoids"

### Details

See file `$R_HOME/library/clusterSim/pdf/indexGap_details.pdf` for further details

Thanks to dr Michael P. Fay from National Institute of Allergy and Infectious Diseases for finding "one column error".

### Value

Gap	Tibshirani, Walther and Hastie gap index for u clusters
diffu	necessary value for choosing correct number of clusters via gap statistic $\text{Gap}(u) - [\text{Gap}(u+1) - s(u+1)]$

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Tibshirani, R., Walther, G., Hastie, T. (2001), *Estimating the number of clusters in a data set via the gap statistic*, "Journal of the Royal Statistical Society", ser. B, vol. 63, part 2, 411-423.

**See Also**

[index.G1](#), [index.G2](#), [index.G3](#), [index.S](#), [index.H](#), [index.KL](#), [index.DB](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
cl1<-pam(data_ratio,4)
cl2<-pam(data_ratio,5)
clall<-cbind(cl1$clustering,cl2$clustering)
g<-index.Gap(data_ratio, clall, reference.distribution="unif", B=10,
  method="pam")
print(g)

# Example 2
library(clusterSim)
means <- matrix(c(0,2,4,0,3,6), 3, 2)
cov <- matrix(c(1,-0.9,-0.9,1), 2, 2)
x <- cluster.Gen(numObjects=40, means=means, cov=cov, model=2)
x <- x$data
md <- dist(x, method="euclidean")^2
# nc - number_of_clusters
min_nc=1
max_nc=15
min <- 0
clopt <- NULL
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
found <- FALSE
for (nc in min_nc:max_nc){
  cl1 <- pam(md, nc, diss=TRUE)
  cl2 <- pam(md, nc+1, diss=TRUE)
  clall <- cbind(cl1$clustering, cl2$clustering)
  gap <- index.Gap(x,clall,B=20,method="pam",centrotypes="centroids")
  res[nc-min_nc+1, 2] <- diffu <- gap$diffu
  if ((res[nc-min_nc+1, 2] >=0) && (!found)){
    nc1 <- nc
    min <- diffu
    clopt <- cl1$cluster
  }
}
```

```

        found <- TRUE
      }
    }
    if (found){
      print(paste("Minimal nc where diffu>=0 is",nc1,"for diffu=",round(min,4)),quote=FALSE)
    }else{
      print("I have not found clustering with diffu>=0", quote=FALSE)
    }
    plot(res,type="p",pch=0,xlab="Number of clusters",ylab="diffu",xaxt="n")
    abline(h=0, untf=FALSE)
    axis(1, c(min_nc:max_nc))

# Example 3
library(clusterSim)
means <- matrix(c(0,2,4,0,3,6), 3, 2)
cov <- matrix(c(1,-0.9,-0.9,1), 2, 2)
x <- cluster.Gen(numObjects=40, means=means, cov=cov, model=2)
x <- x$data
md <- dist(x, method="euclidean")^2
# nc - number_of_clusters
min_nc=1
max_nc=15
min <- 0
clopt <- NULL
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
found <- FALSE
for (nc in min_nc:max_nc){
  cl1 <- pam(md, nc, diss=TRUE)
  cl2 <- pam(md, nc+1, diss=TRUE)
  clall <- cbind(cl1$clustering, cl2$clustering)
  gap <- index.Gap(x,clall,B=20,method="pam",d=md,centrotypes="medoids")
  res[nc-min_nc+1, 2] <- diffu <- gap$diffu
  if ((res[nc-min_nc+1, 2] >=0) && (!found)){
    nc1 <- nc
    min <- diffu
    clopt <- cl1$cluster
    found <- TRUE
  }
}
}
if (found){
  print(paste("Minimal nc where diffu>=0 is",nc1,"for diffu=",round(min,4)),quote=FALSE)
}else{
  print("I have not found clustering with diffu>=0",quote=FALSE)
}
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="diffu", yaxt="n")
abline(h=0, untf=FALSE)
axis(1, c(min_nc:max_nc))

```

**Description**

Calculates Hartigan index

**Usage**

```
index.H (x, clall, d=NULL, centrotypes="centroids")
```

**Arguments**

x	data
clall	Two vectors of integers indicating the cluster to which each object is allocated in partition of n objects into u and u+1 clusters
d	optional distance matrix, used for calculations if centrotypes="medoids"
centrotypes	"centroids" or "medoids"

**Details**

See file \$R\_HOME\library\clusterSim\pdf\indexH\_details.pdf for further details

**Value**

Hartigan index

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Hartigan, J. (1975), *Clustering algorithms*, Wiley, New York.

Milligan, G.W., Cooper, M.C. (1985), *An examination of procedures of determining the number of cluster in a data set*, "Psychometrika", vol. 50, no. 2, 159-179.

Tibshirani, R., Walther, G., Hastie, T. (2001), *Estimating the number of clusters in a data set via the gap statistic*, "Journal of the Royal Statistical Society", ser. B, vol. 63, part 2, 411-423.

**See Also**

[index.G1](#), [index.G2](#), [index.G3](#), [index.S](#), [index.KL](#), [index.Gap](#), [index.DB](#)

**Examples**

```

# Example 1
library(clusterSim)
data(data_ratio)
cl1<-pam(data_ratio,4)
cl2<-pam(data_ratio,5)
clall<-cbind(cl1$clustering,cl2$clustering)
index.H(data_ratio,clall)

# Example 2
library(clusterSim)
data(data_ratio)
md <- dist(data_ratio, method="euclidean")
# nc - number_of_clusters
min_nc=1
max_nc=20
min <- 0
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
found <- FALSE
clusters <- NULL
for (nc in min_nc:max_nc)
{
  print(nc)
  hc <- hclust(md, method="complete")
  cl1 <- cutree(hc, k=nc)
  cl2 <- cutree(hc, k=nc+1)
  clall <- cbind(cl1,cl2)
  res[nc-min_nc+1,2] <- H <- index.H(data_ratio,clall,centrotyes="centroids")
  if ((res[nc-min_nc+1, 2]<10) && (!found)){
    ncl <- nc
    min <- H
    clopt <- cl1
    found <- TRUE
  }
}
if (found)
{
  print(paste("minimal nc for H<=10 equals",ncl,"for H=",min))
  print("clustering for minimal nc where H<=10")
  print(clopt)
}else
{
  print("Clustering not found with H<=10")
}
write.table(res,file="H_res.csv",sep=";",dec=",",row.names=TRUE,col.names=FALSE)
plot(res,type="p",pch=0,xlab="Number of clusters",ylab="H",xaxt="n")
abline(h=10, untf=FALSE)
axis(1, c(min_nc:max_nc))

# Example 3
library(clusterSim)

```

```

data(data_ratio)
md <- dist(data_ratio, method="manhattan")
# nc - number_of_clusters
min_nc=1
max_nc=20
min <- 0
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
found <- FALSE
clusters <- NULL
for (nc in min_nc:max_nc)
{
  print(nc)
  hc <- hclust(md, method="complete")
  cl1 <- cutree(hc, k=nc)
  cl2 <- cutree(hc, k=nc+1)
  clall <- cbind(cl1,cl2)
  res[nc-min_nc+1,2] <- H <- index.H(data_ratio,clall,d=md,centrotypes="medoids")
  if ((res[nc-min_nc+1, 2]<10) && (!found)){
    ncl <- nc
    min <- H
    clopt <- cl1
    found <- TRUE
  }
}
if (found)
{
  print(paste("minimal nc for H<=10 equals",ncl,"for H=",min))
  print("clustering for minimal nc where H<=10")
  print(clopt)
}else
{
  print("Clustering not found with H<=10")
}
write.table(res,file="H_res.csv",sep=";",dec=".",row.names=TRUE,col.names=FALSE)
plot(res,type="p",pch=0,xlab="Number of clusters",ylab="H",xaxt="n")
abline(h=10, untf=FALSE)
axis(1, c(min_nc:max_nc))

```

---

index.KL

*Calculates Krzanowski-Lai index*


---

### Description

Calculates Krzanowski-Lai index

### Usage

```
index.KL (x,clall,d=NULL,centrotypes="centroids")
```

**Arguments**

x	data
clall	Three vectors of integers indicating the cluster to which each object is allocated in partition of n objects into u-1, u, and u+1 clusters
d	optional distance matrix, used for calculations if centrotypes="medoids"
centrotypes	"centroids" or "medoids"

**Details**

See file `$R_HOME/library/clusterSim/pdf/indexKL_details.pdf` for further details

**Value**

Krzanowski-Lai index

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

- Krzanowski, W.J., Lai, Y.T. (1988), *A criterion for determining the number of groups in a data set using sum of squares clustering*, "Biometrics", 44, 23-34.
- Milligan, G.W., Cooper, M.C. (1985), *An examination of procedures of determining the number of cluster in a data set*, "Psychometrika", vol. 50, no. 2, 159-179.
- Tibshirani, R., Walther, G., Hastie, T. (2001), *Estimating the number of clusters in a data set via the gap statistic*, "Journal of the Royal Statistical Society", ser. B, vol. 63, part 2, 411-423.

**See Also**

[index.G1](#), [index.G2](#), [index.G3](#), [index.S](#), [index.H](#), [index.Gap](#), [index.DB](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
cl1<-pam(data_ratio,4)
cl2<-pam(data_ratio,5)
cl3<-pam(data_ratio,6)
clall<-cbind(cl1$clustering,cl2$clustering,cl3$clustering)
index.KL(data_ratio,clall)

# Example 2
library(clusterSim)
data(data_ratio)
```

```

md <- dist(data_ratio, method="manhattan")
# nc - number_of_clusters
min_nc=2
max_nc=15
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  cl1 <- pam(md, nc-1, diss=TRUE)
  cl2 <- pam(md, nc, diss=TRUE)
  cl3 <- pam(md, nc+1, diss=TRUE)
  clall<- cbind(cl1$clustering, cl2$clustering, cl3$clustering)
  res[nc-min_nc+1,2] <- KL <- index.KL(data_ratio,clall,centrotypes="centroids")
  clusters <- rbind(clusters, cl2$cluster)
}
print(paste("max KL for", (min_nc:max_nc)[which.max(res[,2])], "clusters=", max(res[,2])))
print("clustering for max KL")
print(clusters[which.max(res[,2]),])
write.table(res, file="KL_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="KL", xaxt="n")
axis(1, c(min_nc:max_nc))

# Example 3
library(clusterSim)
data(data_ratio)
md <- dist(data_ratio, method="manhattan")
# nc - number_of_clusters
min_nc=2
max_nc=15
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  cl1 <- pam(md, nc-1, diss=TRUE)
  cl2 <- pam(md, nc, diss=TRUE)
  cl3 <- pam(md, nc+1, diss=TRUE)
  clall<- cbind(cl1$clustering, cl2$clustering, cl3$clustering)
  res[nc-min_nc+1,2] <- KL <- index.KL(data_ratio,clall,d=md,centrotypes="medoids")
  clusters <- rbind(clusters, cl2$cluster)
}
print(paste("max KL for", (min_nc:max_nc)[which.max(res[,2])], "clusters=", max(res[,2])))
print("clustering for max KL")
print(clusters[which.max(res[,2]),])
write.table(res, file="KL_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="KL", xaxt="n")
axis(1, c(min_nc:max_nc))

```

**Description**

Calculates Rousseeuw's Silhouette internal cluster quality index

**Usage**

```
index.S(d, cl)
```

**Arguments**

d                   'dist' object  
cl                   A vector of integers indicating the cluster to which each object is allocated

**Details**

See file `$R_HOME/library/clusterSim/pdf/indexS_details.pdf` for further details

**Value**

calculated Silhouette index

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
Department of Econometrics and Computer Science, University of Economics, Wroclaw, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Gatnar, E., Walesiak, M. (Eds.) (2004), *Metody statystycznej analizy wielowymiarowej w badaniach marketingowych [Multivariate statistical analysis methods in marketing research]*, Wydawnictwo AE, Wroclaw, 342-343, erratum.

Kaufman, L., Rousseeuw, P.J. (1990), *Finding groups in data: an introduction to cluster analysis*, Wiley, New York, 83-88.

**See Also**

[index.G1](#), [index.G2](#), [index.G3](#), [index.KL](#), [index.H](#), [index.Gap](#), [index.DB](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ratio)
d <- dist.GDM(data_ratio)
c <- pam(d, 5, diss = TRUE)
icq <- index.S(d, c$clustering)
print(icq)

# Example 2
library(clusterSim)
```

```

data(data_ratio)
md <- dist(data_ratio, method="manhattan")
# nc - number_of_clusters
min_nc=2
max_nc=20
res <- array(0, c(max_nc-min_nc+1, 2))
res[,1] <- min_nc:max_nc
clusters <- NULL
for (nc in min_nc:max_nc)
{
  cl2 <- pam(md, nc, diss=TRUE)
  res[nc-min_nc+1, 2] <- S <- index.S(md,cl2$cluster)
  clusters <- rbind(clusters, cl2$cluster)
}
print(paste("max S for", (min_nc:max_nc)[which.max(res[,2])], "clusters=", max(res[,2])))
print("clustering for max S")
print(clusters[which.max(res[,2]),])
write.table(res, file="S_res.csv", sep=";", dec=".", row.names=TRUE, col.names=FALSE)
plot(res, type="p", pch=0, xlab="Number of clusters", ylab="S", xaxt="n")
axis(1, c(min_nc:max_nc))

```

---

initial.Centers      *Calculation of initial clusters centers for k-means like algorithms*

---

### Description

Function calculates initial clusters centers for k-means like algorithms with the following algorithm (similar to SPSS QuickCluster function)

(a) if the distance between  $x_k$  and its closest cluster center is greater than the distance between the two closest centers ( $M_m$  and  $M_n$ ), then  $x_k$  replaces either  $M_m$  or  $M_n$ , whichever is closer to  $x_k$ .

(b) If  $x_k$  does not replace a cluster initial center in (a), a second test is made: If that distance  $d_q$  greater than the distance between  $M_q$  and its closest  $M_i$ , then  $x_k$  replaces  $M_q$ .

where:

$M_i$  - initial center of  $i$ -th cluster

$x_k$  - vector of  $k$ -th observation

$d(\dots, \dots)$  - Euclidean distance

$d_{mn} = \min_{ij} d(M_i, M_j)$

$d_q = \min_i d(x_k, M_i)$

### Usage

```
initial.Centers(x, k)
```

### Arguments

x	matrix or dataset
k	number of initial cluster centers

**Value**

Numbers of objects chosen as initial cluster centers

**Author(s)**

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

**References**

Hartigan, J. (1975), *Clustering algorithms*, Wiley, New York.

**See Also**

[cluster.Sim](#)

**Examples**

```
#Example 1 (numbers of objects chosen as initial cluster centers)
library(clusterSim)
data(data_ratio)
ic <- initial.Centers(data_ratio, 10)
print(ic)
```

```
#Example 2 (application with kmeans algorithm)
library(clusterSim)
data(data_ratio)
kmeans(data_ratio, data_ratio[initial.Centers(data_ratio, 10),])
```

---

plotCategorical      *Plot categorical data on a scatterplot matrix*

---

**Description**

Plot categorical data on a scatterplot matrix (optionally with clusters)

**Usage**

```
plotCategorical(x, pairsofVar=NULL, cl=NULL, clColors=NULL, ...)
```

**Arguments**

<code>x</code>	data matrix (rows correspond to observations and columns correspond to variables)
<code>pairsofVar</code>	pairs of variables - all variables ( <code>pairsofVar=NULL</code> ) or selected variables, e.g. <code>pairsofVar=c(1,3,4)</code>
<code>cl</code>	cluster membership vector
<code>clColors</code>	The colors of clusters. The colors are given arbitrary ( <code>clColors=TRUE</code> ) or by hand, e.g. <code>clColors=c("red", "blue", "green")</code> . The number of colors equals the number of clusters
<code>...</code>	Arguments to be passed to methods, such as graphical parameters (see <a href="#">par</a> ).

**Author(s)**

Marek Walesiak ([marek.walesiak@ue.wroc.pl](mailto:marek.walesiak@ue.wroc.pl)), Andrzej Dudek ([andrzej.dudek@ue.wroc.pl](mailto:andrzej.dudek@ue.wroc.pl))

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland

<http://keii.ue.wroc.pl/clusterSim>

**See Also**

[plotCategorical3d](#), [colors](#), [pairs](#)

**Examples**

```
# Example 1
library(clusterSim)
data(data_ordinal)
plotCategorical(data_ordinal, pairsofVar=c(1,3,4,9), cl=NULL,
clColors = NULL)

# Example 2
library(clusterSim)
grnd <- cluster.Gen(50,model=5,dataType="o",numCategories=5)
plotCategorical(grnd$data, pairsofVar=NULL, cl=grnd$clusters,
clColors=TRUE)

# Example 3
library(clusterSim)
grnd<-cluster.Gen(50,model=4,dataType="o",numCategories=7, numNoisyVar=2)
plotCategorical(grnd$data, pairsofVar=NULL, cl=grnd$clusters,
clColors = c("red","blue","green"))
```

---

plotCategorical3d *Plot categorical data with three-dimensional plots*

---

### Description

Plot categorical data with three-dimensional plots (optionally with clusters)

### Usage

```
plotCategorical3d(x, tripleofVar=c(1,2,3), cl=NULL, clColors=NULL, ...)
```

### Arguments

x	data matrix (rows correspond to observations and columns correspond to variables)
tripleofVar	triple of variables - vector of the number of variables, e.g. tripleofVar = c(1, 3, 4)
cl	cluster membership vector
clColors	The colors of clusters. The colors are given arbitrary (clColors=TRUE) or by hand, e.g. clColors=c("red", "blue", "green"). The number of colors equals the number of clusters
...	Arguments to be passed to methods, such as graphical parameters (see <a href="#">par</a> ).

### Author(s)

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

### See Also

[plotCategorical](#), [colors](#)

### Examples

```
# These examples do not run on Mac_OS-X. We're working to fix them
# They run quite well on Windows and Linux in meantime

# Example 1
#library(clusterSim)
#data(data_ordinal)
#plotCategorical3d(data_ordinal, tripleofVar=c(1,3,9), cl=NULL,
#clColors=NULL)

# Example 2
#library(clusterSim)
#grnd <- cluster.Gen(50,model=5,dataType="o",numCategories=5)
```

```
#plotCategorical3d(grnd$data, tripleofVar=c(1,2,3), cl=grnd$clusters,
#clColors=TRUE)

# Example 3
#library(clusterSim)
#grnd <- cluster.Gen(50, model=4, dataType="o", numCategories=7, numNoisyVar=2)
#plotCategorical3d(grnd$data, tripleofVar=c(1,2,4), cl=grnd$clusters,
#clColors=c("red","blue","green"))
```

---

plotInterval      *Plot symbolic interval-valued data on a scatterplot matrix*

---

### Description

Plot symbolic interval-valued data on a scatterplot matrix (optionally with clusters)

### Usage

```
plotInterval(x, pairsofsVar=NULL, cl=NULL, clColors=NULL, ...)
```

### Arguments

x	symbolic interval-valued data
pairsofsVar	pairs of symbolic interval variables - all variables (pairsofsVar=NULL) or selected variables, e.g. pairsofsVar=c(1, 3, 4)
cl	cluster membership vector
clColors	The colors of clusters. The colors are given arbitrary (clColors=TRUE) or by hand, e.g. clColors=c("red", "blue", "green"). The number of colors equals the number of clusters
...	Arguments to be passed to methods, such as graphical parameters (see <a href="#">par</a> ).

### Author(s)

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)

Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

### See Also

[plotCategorical](#), [plotCategorical3d](#), [colors](#), [pairs](#)

**Examples**

```

# Example 1
library(clusterSim)
data(data_symbolic)
plotInterval(data_symbolic, pairsofsVar=c(1,3,4,6), cl=NULL,
             clColors=NULL)

# Example 2
library(clusterSim)
grnd <- cluster.Gen(60, model=5, dataType="s", numNoisyVar=1,
                  numOutliers=10, rangeOutliers=c(1,5))
grnd$clusters[grnd$clusters==0] <- max(grnd$clusters)+1
# To colour outliers
plotInterval(grnd$data, pairsofsVar=NULL, cl=grnd$clusters,
             clColors=TRUE)

# Example 3
library(clusterSim)
grnd <- cluster.Gen(50, model=4, dataType="s", numNoisyVar=2,
                  numOutliers=10, rangeOutliers=c(1,4))
grnd$clusters[grnd$clusters==0] <- max(grnd$clusters)+1
# To colour outliers
plotInterval(grnd$data, pairsofsVar=NULL, cl=grnd$clusters,
             clColors=c("red", "blue", "green", "yellow"))

```

---

replication.Mod      *Modification of replication analysis for cluster validation*

---

**Description**

Modification of replication analysis for cluster validation

**Usage**

```

replication.Mod(x, v="m", u=2, centrotypes="centroids",
               normalization=NULL, distance=NULL, method="kmeans",
               S=10, fixedAsample=NULL)

```

**Arguments**

x	data matrix
v	type of data: metric ("r" - ratio, "i" - interval, "m" - mixed), nonmetric ("o" - ordinal, "n" - multi-state nominal, "b" - binary)
u	number of clusters given arbitrary
centrotypes	"centroids" or "medoids"

normalization	<p>optional, normalization formulas for metric data:          for ratio data: "n0" - without normalization, "n6" - (x/sd), "n7" - (x/range), "n8" - (x/max), "n9" - (x/mean), "n10" - (x/sum), "n11" - <math>x/\sqrt{SSQ}</math>          for interval or mixed data: "n0" - without normalization, "n1" - (x-mean)/sd, "n2" - (x-Me)/MAD, "n3" - (x-mean)/range, "n4" - (x-min)/range, "n5" - (x-mean)/max[abs(x-mean)]</p>
distance	<p>distance measures          NULL for "kmeans" method (based on data matrix),          for ratio data: "d1" - Manhattan, "d2" - Euclidean, "d3" - Chebychev (max), "d4" - squared Euclidean, "d5" - GDM1, "d6" - Canberra, "d7" - Bray-Curtis          for interval or mixed (ratio &amp; interval) data: "d1", "d2", "d3", "d4", "d5"          for ordinal data: "d8" - GDM2          for multi-state nominal: "d9" - Sokal &amp; Michener          for binary data: "b1" = Jaccard; "b2" = Sokal &amp; Michener; "b3" = Sokal &amp; Sneath (1); "b4" = Rogers &amp; Tanimoto; "b5" = Czekanowski; "b6" = Gower &amp; Legendre (1); "b7" = Ochiai; "b8" = Sokal &amp; Sneath (2); "b9" = Phi of Pearson; "b10" = Gower &amp; Legendre (2)</p>
method	clustering method: "kmeans" (default), "single", "complete", "average", "mcquitty", "median", "centroid", "ward", "pam", "diana"
S	the number of simulations used to compute mean corrected Rand index
fixedAsample	if NULL A sample is generated randomly, otherwise this parameter contains object numbers arbitrarily assigned to A sample

## Details

See file `$R_HOME/library/clusterSim/pdf/replication.Mod_details.pdf` for further details

## Value

A	3-dimensional array containing data matrices for A sample of objects in each simulation (first dimension represents simulation number, second - object number, third - variable number)
B	3-dimensional array containing data matrices for B sample of objects in each simulation (first dimension represents simulation number, second - object number, third - variable number)
centroid	3-dimensional array containing centroids of u clusters for A sample of objects in each simulation (first dimension represents simulation number, second - cluster number, third - variable number)
medoid	3-dimensional array containing matrices of observations on u representative objects (medoids) for A sample of objects in each simulation (first dimension represents simulation number, second - cluster number, third - variable number)
clusteringA	2-dimensional array containing cluster numbers for A sample of objects in each simulation (first dimension represents simulation number, second - object number)

clusteringB	2-dimensional array containing cluster numbers for B sample of objects in each simulation (first dimension represents simulation number, second - object number)
clusteringBB	2-dimensional array containing cluster numbers for B sample of objects in each simulation according to 4 step of replication analysis procedure (first dimension represents simulation number, second - object number)
cRand	value of mean corrected Rand index for S simulations

### Author(s)

Marek Walesiak (marek.walesiak@ue.wroc.pl), Andrzej Dudek (andrzej.dudek@ue.wroc.pl)  
 Department of Econometrics and Computer Science, University of Economics, Wrocław, Poland  
<http://keii.ue.wroc.pl/clusterSim>

### References

- Breckenridge, J.N. (2000), *Validating cluster analysis: consistent replication and symmetry*, "Multivariate Behavioral Research", 35 (2), 261-285.
- Gordon, A.D. (1999), *Classification*, Chapman and Hall/CRC, London.
- Hubert, L., Arabie, P. (1985), *Comparing partitions*, "Journal of Classification", no. 1, 193-218.
- Milligan, G.W. (1996), *Clustering validation: results and implications for applied analyses*, In P. Arabie, L.J. Hubert, G. de Soete (Eds.), *Clustering and classification*, World Scientific, Singapore, 341-375.
- Walesiak, M. (2008), *Ocena stabilnosci wyników klasyfikacji z wykorzystaniem analizy replikacji*, In: J. Pocięcha (Ed.), *Modelowanie i prognozowanie zjawisk społeczno-gospodarczych*, Wydawnictwo AE, Krakow, 67-72.

### See Also

[cluster.Sim](#), [hclust](#), [kmeans](#), [dist](#), [dist.BC](#), [dist.SM](#), [dist.GDM](#),  
[data.Normalization](#)

### Examples

```
library(clusterSim)
data(data_ratio)
w <- replication.Mod(data_ratio, u=5, S=10)
print(w)
```

```
library(clusterSim)
data(data_binary)
replication.Mod(data_binary, "b", u=2, "medoids", NULL, "b1",
"pam", fixedAsample=c(1,3,6,7))
```

# Index

## \*Topic **cluster**

- cluster.Description, 2
- cluster.Gen, 3
- cluster.Sim, 6
- data.Normalization, 9
- dist.BC, 14
- dist.GDM, 15
- dist.SM, 17
- dist.Symbolic, 18
- HINoV.Mod, 19
- HINoV.Symbolic, 21
- index.DB, 22
- index.G1, 25
- index.G2, 27
- index.G3, 29
- index.Gap, 30
- index.H, 33
- index.KL, 36
- index.S, 38
- initial.Centers, 40
- replication.Mod, 44

## \*Topic **datasets**

- data\_binary, 11
- data\_interval, 11
- data\_mixed, 12
- data\_nominal, 12
- data\_ordinal, 13
- data\_ratio, 13
- data\_symbolic, 14

## \*Topic **data**

- cluster.Gen, 3
- cluster.Sim, 6

## \*Topic **hplot**

- plotCategorical, 41
- plotCategorical3d, 42
- plotInterval, 43

## \*Topic **multivariate**

- cluster.Gen, 3
- cluster.Sim, 6

- replication.Mod, 44

## \*Topic **optimize**

- cluster.Sim, 6

- cluster.Description, 2, 8
- cluster.Gen, 3
- cluster.Sim, 3, 6, 11, 20, 22, 41, 46
- colors, 42–44

- data.Normalization, 9, 9, 46

- data\_binary, 11
- data\_interval, 11
- data\_mixed, 12
- data\_nominal, 12
- data\_ordinal, 13
- data\_ratio, 13
- data\_symbolic, 14
- dist, 9, 15–17, 20, 46
- dist.BC, 9, 14, 16, 17, 20, 46
- dist.GDM, 9, 15, 15, 17, 20, 46
- dist.SM, 9, 15, 16, 17, 18, 20, 46
- dist.Symbolic, 18

- GDM (*dist.GDM*), 15
- GDM1 (*dist.GDM*), 15
- GDM2 (*dist.GDM*), 15

- hclust, 9, 20, 22, 46

- HINoV.Mod, 19

- HINoV.Symbolic, 21

- index.DB, 22, 26, 28, 29, 31, 34, 37, 39
- index.G1, 9, 23, 25, 28, 29, 31, 34, 37, 39
- index.G2, 9, 23, 26, 27, 29, 31, 34, 37, 39
- index.G3, 9, 23, 26, 28, 29, 31, 34, 37, 39
- index.Gap, 23, 26, 28, 29, 30, 34, 37, 39
- index.H, 23, 26, 28, 29, 31, 33, 37, 39
- index.KL, 9, 23, 26, 28, 29, 31, 34, 36, 39
- index.S, 9, 23, 26, 28, 29, 31, 34, 37, 38
- initial.Centers, 40

kmeans, [20](#), [22](#), [46](#)

mad, [3](#)

mean, [3](#)

median, [3](#)

pairs, [42](#), [44](#)

par, [41](#), [42](#), [44](#)

plotCategorical, [41](#), [43](#), [44](#)

plotCategorical3d, [42](#), [42](#), [44](#)

plotInterval, [43](#)

replication.Mod, [44](#)

sd, [3](#)