Package: mccca (via r-universe)

October 23, 2026

Type Package

Title Visualizing Class Specific Heterogeneous Tendencies in Categorical Data

Version 2.4

Date 2026-07-17

Author Mariko Takagishi [aut, cre]

Maintainer Mariko Takagishi <m.takagishi0728@gmail.com>

Description Provides functions for performing multiple-class cluster correspondence analysis(MCCCA). The main functions are create.MCCCAdata() to create a list to be applied to MCCCA, MCCCA() to apply MCCCA, and plot.mccca() for visualizing MCCCA result. Methods used in the package are described in Mariko Takagishi and Michel van de Velden (2022)<doi:10.1080/10618600.2022.2035737>.

License GPL (>= 2)

Depends R (>= 4.1.0)

Imports magic, stringr, ggplot2, gridExtra, wordcloud, RColorBrewer, stats, utils, colorspace, clusterSim, MASS, grDevices, Rcpp (>= 1.0.11)

NeedsCompilation yes

LinkingTo Rcpp, RcppArmadillo

Encoding UTF-8

LazyData true

Config/roxygen2/version 8.0.0

RoxygenNote 7.3.3

Config/pak/sysreqs libicu-dev

Repository https://mtakagishi0.r-universe.dev

Date/Publication 2026-07-21 07:20:02 UTC

RemoteUrl https://github.com/cran/mccca

RemoteRef HEAD

RemoteSha 66b4cb263a5a2969d7a0489fa14a8ec722048bf4

Contents

mccca-package 2
accident 3
create.MCCCAdata 3
decideK 5
generate.ext 8
generate.onedata 9
MCCCA 10
mealDrink 13
plot.calcIndex 13
plot.mccca 14
rcpp_hello_world 17
summary.calcIndex 17
summary.mccca 18
summary.mcccadata 19

Index **21**

`mccca-package` *Visualizing Class Specific Heterogeneous Tendencies in Categorical Data*

Description

The mccca package provides functions for performing multiple-class cluster correspondence analysis (MCCCA).

Details

The main functions are `create.MCCCAdata()`, `MCCCA()`, and `plot.mccca()`. The package also provides functions for selecting the number of clusters, generating simulated data, and summarizing analysis results.

Author(s)

Mariko Takagishi

Maintainer: Mariko Takagishi <m.takagishi0728 at gmail.com>

References

Takagishi, M., & Velden, M. V. D. (2022). Visualizing class specific heterogeneous tendencies in categorical data. Journal of Computational and Graphical Statistics, 31(3), 790-801.

accident — *Accident data*

Description

A dataset obtained from the U.K. Department for Transports road safety statistics.

Usage

```
accident
```

Format

A list of two datasets (4 active variables and 2 external variables, respectively), with 3024 individuals.

Active variable dataset (active):

Light (Light conditions) 4 categories...L0(Daylight), L1 (Darkness:street lights present and lit), L2 (Darkness:street lights present but unlit), L3(Darkness:no street lighting)

Weather (Weather conditions) 8 categories...Fine,Rain,Snow,Fine_w,Rain_w,Snow_w,Fog,Other.

Road surface (Road surface conditions) 5 categories...Dry,Wet,Snow,Frost,Flood.

Speed (Speed limit) 2 categories...S30 (Speed limit is up to 30km/h), S70 (Speed limit is up to 70km/h) External variable dataset (external):

Casualty class 2 categories..Driver (Casualty is one driver), Ped (Casualty is one pedestrian).

Area 2 categories...Urban (Occurring in urban area), Rural (Occurring in rural area).

Source

https://www.gov.uk/government/collections/road-accidents-and-safety-statistics

create.MCCCAdata — *Create a list (class : mcccadata) which is later applied to MCCCA.*

Description

Creates a `mcccadata` object which is later applied to `MCCCA`.

Usage

```
create.MCCCAdata(dat.act,dat.ext)
```

Arguments

`dat.act` An (N x J) dataframe (matrix) of active categorical variables (N:the number of observations, J:the number of active variables). If `rownames(dat.act)` is `NULL`, `c(obj1,..,objN)` are used as `rownames(dat.act)`.

`dat.ext` An (N x H) dataframe (matrix) of external variables (H : the number of external variables).

Value

Returns a list with the following elements.

`C` The number of classes created from in this function.

`dat.act` A copy of `dat.act`.

`act.list` A list of C data frames, each of which is obtained by dividing `dat.act` into C classes.

`N.vec` An integer vector of length C giving the number of observations belonging to each of the C classes.

`q.vec` An integer vector of length J giving the number of categories in each of the J active variables.

`classlabel` A character vector of length C giving the class label for each of the C classes.

`class.label.n.vec`
A character vector vector of length N indicating the class label for each observation belongs to. The order is the same as the order of the rows of `dat.act` (i.e., `names(class.label.n.vec)=rownames(dat.act)`).

`class.index.n.vec`
An integer (with values from 1:C) vector of length N, giving the same information as `class.label.n.vec` but represented by class indices.

`obs.index.n.list`
A list of C integer (with values from 1:N) vectors, with the observation indices corresponding to each of the C classes.

`classlab.mat` A (Cx(H+1)) table showing the correspondence between each class and the external variables. The rows indicate the classes, while the first H columns indicate the external category of each external variable that constitutes the class, and the last column indicates the class label.

References

Takagishi, M. and van de Velden, M. (2022), Visualizing class specific heterogeneous tendencies in categorical data. *Journal of Computational and Graphical Statistics*, 31(3), pp.790-801.

See Also

`summary.mcccadata`, `MCCCA`

Examples

```
# Load the example data
data(mealDrink)

# Prepare active and external variables
active<-c(1,2)
external<-c(3,4)
dat.act=mealDrink[,active]
dat.ext=mealDrink[,external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Show a summary of the created classes
summary(mcccadata)

# Specify the numbers of class-specific clusters
K.vec <- c(2, 1, 2, 2)

# Apply MCCCA
res <- MCCCA(mcccadata, K.vec = K.vec)

# Display the MCCCA biplot
shortlabels <- c("AF", "AM", "JF", "JM")
plot(res, classlabel = shortlabels)

# Show a numerical summary
summary(res)
```

---

decideK *Calculate chosen cluster index using specified K.*

Description

Calculates a cluster validity index to determine the numbers of class-specific clusters.

Usage

```
decideK(
  mccca.data,
  subthr = 20,
  cindex = "KL",
  cindexnm = NULL,
  Krange = NULL,
  Krange.list = NULL,
  return.clus = FALSE,
  fulldim = FALSE,
  nstart = 1000
)
```

Arguments

`mccca.data`	A list created in `create.MCCCAdata`.
`subthr`	An integer. The cluster validity index is not calculated for classes with fewer observation than this value. The default is 20.
`cindex`	A character string specifying the cluster validity index used to determine the numbers of class-specific clusters. One of `"KL"`, `"CH"`, `"S"`, or `"DB"`. See details. The default is `"KL"`.
`cindexnm`	An integer specifying the cluster validity index. If specified, this argument is used instead of `cindex`. See details.
`Krange`	An integer vector of length 2, giving the range of candidate number of clusters which are common to all classes.
`Krange.list`	A list of length C vectors, specifying the ranges of candidate numbers of clusters for each C classes. This is used when you want to use different `Kcand` for each class. The default is `NULL`.
`return.clus`	A logical value indicating whether the clustering results for all candidate numbers of clusters, are returned. The default is `FALSE`.
`fulldim`	A logical value indicating whether k-means is applied to all MCA dimensions. If `FALSE` (default), only the first two dimensions are used.
`nstart`	An integer specifying the number of random initial configurations used in k-means. Larger values generally improve the stability of the solution but increase the computation time. The default is 1000.

Details

Four cluster validity indices are available:

- `"KL"`: Krzanowski–Lai index (Krzanowski and Lai 1988),
- `"CH"`: Calinski–Harabasz index (Cali\'nski and Harabasz 1974),
- `"S"`: Silhouette index (Kaufman and Rousseeuw 2009),
- `"DB"`: Davies–Bouldin index (Davies and Bouldin 1979).

Alternatively, the cluster validity index can be specified by `cindexnm`, where `1 = "KL"`, `2 = "CH"`, `3 = "S"`, and `4 = "DB"`.

Value

Returns a list with the following elements.

`cls.best`	An integer vector of length `C`, giving the best number of clusters for each class according to the specified cluster validity index.
`ind.list`	A list of C vectors, giving the calculated values of the specified cluster validity index for each specified number of clusters.
`cindexnm`	The numeric code of the cluster validity index used.
`index`	The name of the cluster validity index used.
`Cls.list`	A list of C matrices, containing the clustering results for all candidate numbers of clusters in each class. This is returned only when `return.clus=TRUE`; otherwise, `NULL`.

References

Cali\'nski, T. and Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics*, 3(1), pp.1-27.

Davies, D. L. and Bouldin, D. W. (1979). A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2), pp.224-227.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

Krzanowski, W. J. and Lai, Y. T. (1988). A criterion for determining the number of groups in a data set using sum-of-squares clustering. *Biometrics*, 44(1), pp.23-34.

Takagishi, M. and van de Velden, M. (2022), Visualizing class specific heterogeneous tendencies in categorical data. *Journal of Computational and Graphical Statistics*, 31(3), pp.790-801

See Also

summary.calcIndex, plot.calcIndex

Examples

```
# Load the example data
data(accident)

# Prepare active and external variables
active<-c(1:4)
external<-c(5:6)
dat.act<-accident[,active]
dat.ext<-accident[,external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Specify the candidate numbers of clusters (2 to 8)
Krange=c(2,4)

# Determine the numbers of class-specific clusters using the default KL index
# (using a small nstart for a faster example)
KLres=decideK(mcccadata,Krange=Krange,nstart=10)


# Show the calculated cluster validity indices
summary(KLres)


# Plot the cluster validity indices
plot(KLres)

# Use the numbers of clusters selected by decideK() to fit MCCCA
K.vec <- KLres$cls.best

# Apply MCCCA (using a small nstart for a faster example)
res <- MCCCA(mcccadata, K.vec = K.vec, nstart=5)
```

`generate.ext` *generates an artificial (NxH) external variable matrix.*

Description

Generates an artificial (NxH) external variable matrix.

Usage

```
generate.ext(N,extcate.vec=extcate.vec,unbala.cate=FALSE)
```

Arguments

`N`	The number of observation.
`extcate.vec`	A vector of length H, each element indicates the number of category for each H external variables.
`unbala.cate`	logical value. If TRUE, the proportion of categories in the external variable is unbalanced. The default is FALSE.

Value

An (NxH) external variable matrix.

See Also

`generate.onedata`

Examples

```
# setting
N <- 100 ; J <- 5 ; Ktrue <- 2 ; q.vec <- rep(5,J) ; noise.prop <- 0.2
extcate.vec=c(2,3) #the number of categories for each external variable

# generate categorical variable data
catedata.list <- generate.onedata(N=N,J=J,Ktrue=Ktrue,q.vec=q.vec,noise.prop = noise.prop)
data.act=catedata.list$data.mat

# generate external variable data
data.ext=generate.ext(N,extcate.vec=extcate.vec)

# create mccca.list to be applied to MCCCA function
mccca.data=create.MCCCAdata(data.act,data.ext)
```

`generate.onedata` *Generate (NxJ) categorical data matrix.*

Description

Generate (NxJ) categorical data matrix.

Usage

```
generate.onedata(N=100,J=5,Ktrue=3,q.vec=rep(3,5),noise.prop=0.3)
```

Arguments

`N`	The number of observations. Default is 100.
`J`	The number of active variables. Default is 5.
`Ktrue`	The number of true clusters. Default is 3.
`q.vec`	A vector of length J giving the number of categories for each active variable. Default is rep(3,5).
`noise.prop`	A numeric value between 0 and 1 indicating the proportion of noise variables among J variables. Default is 0.3.

Value

Returns a list with the following elements.

`data.mat`	A (NxJ) data frame of categorical data.
`clstr0.vec`	A vector of integers (from 1:Ktrue) length N giving the cluster to which each observation is allocated.

See Also

`generate.onedata`

Examples

```
# setting
N <- 100 ; J <- 5 ; Ktrue <- 2 ; q.vec <- rep(5,J) ; noise.prop <- 0.2
extcate.vec=c(2,3) #the number of categories for each external variable

# generate categorical variable data
catedata.list <- generate.onedata(N=N,J=J,Ktrue=Ktrue,q.vec=q.vec,noise.prop = noise.prop)
data.act=catedata.list$data.mat

# generate external variable data
data.ext=generate.ext(N,extcate.vec=extcate.vec)

# create mccca.list to be applied to MCCCA function
mccca.data=create.MCCCAdata(data.act,data.ext)
```

MCCCA *apply MCCCA for dataset.*

Description

Applies MCCCA to a mcccadata object created by create.MCCCAdata.

Usage

```
MCCCA(
  mcccadata,
  K.vec = NULL,
  nstart = 300,
  maxit = 50,
  ndim = 2,
  tol = 1e-06,
  verbose = FALSE,
  update.kmeans = FALSE
)
```

Arguments

mcccadata	A mcccadata object, a list created by create.MCCCAdata.
K.vec	An integer vector of length C (C:the number of classes). Each element corresponds to the number of clusters in each class, which needs to be specified for estimation.
nstart	An integer indicating the number of random initial values. The default is 30.
maxit	An integer indicating the maximum number of iterations.
ndim	An integer indicating the dimension of quantification. The default is 2.
tol	A numeric value indicating the absolute convergence tolerance. The default is 1e-6.
verbose	A logical value. If TRUE, tracing information on the progress of the optimization is produced.
update.kmeans	A logical value, indicating whether a cluster indicator matrix is updated by k-means or referring to the distance (the way described in the reference paper). The default is FALSE (not using k-means).

Details

Bg,Gg and Fng are re-scaled versions of B,G and Fn respectively, where the average squared deviations from the origin for the row and column points is the same (See section 2.3 in Takagishi, M. and van de Velden, M. (2022) for details).

The first column of both inertia and inertia.G contains the Benzécri-adjusted inertias.

In `inertia`, the second column contains the relative adjusted inertias obtained by dividing each adjusted inertia by the sum of all adjusted inertias, and the third column contains their cumulative values.

In `inertia.G`, the second column contains the relative adjusted inertias calculated using Greenacre's normalization, and the third column contains their cumulative values.

See Benzécri (1979) and Greenacre (1993) for details of these inertia corrections.

Value

Returns a list with the following elements.

- `G` — A (K x `ndim`) quantification matrix for all clusters (K: the total number of clusters, i.e.,K=`sum(K.vec)`).
- `Gg` — Re-scaled version of `G`. See details.
- `B` — A (Q x `ndim`) quantification matrix for all categories (Q: the total number of categories), and `q.vec` is given by `create.MCCCAdata`).
- `Bg` — Re-scaled version of `B`.
- `Fn` — A (N x `ndim`) quantification matrix for all observations.
- `Fng` — Re-scaled version of `Fn`.
- `cluster.class.index` — An integer (from 1:C) vector of length K showing the class indices to which each cluster corresponds.
- `cluster.index` — An integer (from 1:K) vector of length K showing the cluster indices to which each cluster corresponds.
- `cluster.label.n.vec` — A character vector of length N, where each element represents the estimated cluster labels for each observation that corresponds to the rows of `dat.act` (given in `mcccadata` object).
- `cluster.label.n.list` — A list of C vectors, giving the estimated cluster labels for each observation in each class.
- `cluster.index.n.vec` — An integer (from 1:K) vector of length N, showing the same information as `cluster.label.n.vec` but organized by cluster indices.
- `cluster.index.n.list` — A list of C vectors, showing the same information as `cluster.label.n.list` but given by cluster indices.
- `catename.vec` — A character vector of length Q with the category names of each categorical variable.
- `catename.vari.vec` — A character vector of length Q with `catename.vec` plus the name of categorical variable (by default, this is used as the column name of `B` and `Bg`).
- `cate.removed` — If there is a category that is not chosen by any observation, `cate.removed` gives which category was removed (given by the index of column in dummy matrix). Otherwise, return `NULL`.

q.vec	A copy of q.vec in the mcccadata list.
K.vec	A copy of K.vec in the argument of MCCCA().
classlabel	A copy of classlabel in the mcccadata list.
inertia	A table containing the adjusted inertias and their relative and cumulative values. See details.
inertia.G	A table containing the adjusted inertias and their relative and cumulative values based on Greenacre's normalization. See details.

References

Takagishi, M. and van de Velden, M. (2022), Visualizing class specific heterogeneous tendencies in categorical data. *Journal of Computational and Graphical Statistics*, 31(3), pp.790-801.

Benzécri, J.-P. (1979). Sur le calcul des taux d'inertie dans l'analyse d'un questionnaire. *Cahiers de l'Analyse des Données*, 4, pp.377-378.

Greenacre, M. J. (1993). *Correspondence Analysis in Practice*. Academic Press.

See Also

create.MCCCAdata, summary.mccca, plot.mccca

Examples

```
# Load the example data
data(mealDrink)

# Prepare active and external variables
active<-c(1,2)
external<-c(3,4)
dat.act=mealDrink[,active]
dat.ext=mealDrink[,external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Specify the numbers of class-specific clusters
K.vec <- c(2, 1, 2, 2)

# Apply MCCCA
res <- MCCCA(mcccadata, K.vec = K.vec)

# Display the MCCCA biplot
shortlabels <- c("AF", "AM", "JF", "JM")
plot(res, classlabel = shortlabels)

# Show a numerical summary
summary(res)
```

mealDrink — *Meal and Drink preference data*

Description

Artificial datasets containing active variables (meal and drink) and external variables (nationality and gender)

Usage

```
mealDrink
```

Format

A list of two datasets (2 active variables and 2 external variables, respectively), with 390 individuals.

Active variable dataset (active):

Meal meal preference

Drink drink preference External variable dataset (external):

Nationality nationality

Gender gender

Source

Takagishi, M. and van de Velden, M. (2022), Visualizing class specific heterogeneous tendencies in categorical data. *Journal of Computational and Graphical Statistics*, 31(3), pp.790-801

plot.calcIndex — *Plot a calcIndex Object*

Description

Displays the cluster-number selection index for each class.

By default, up to four plots are displayed on each page in a 2-by-2 arrangement. If more than four plots are available, press Enter to display the next page.

Usage

```
## S3 method for class 'calcIndex'
plot(x, onebyone = FALSE, ...)
```

Arguments

x	A calcIndex object, a list created by decideK.
onebyone	Logical. If TRUE, plots are displayed one at a time. Press Enter to display the next plot. If FALSE, up to four plots are displayed on each page in a 2-by-2 arrangement. The default is FALSE.
...	Additional arguments. Currently not used.

See Also

decideK, summary.calcIndex

Examples

```
# Load the example data
data(accident)

# Prepare active and external variables
active <- 1:4
external <- 5:6
dat.act <- accident[, active]
dat.ext <- accident[, external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Specify the candidate numbers of clusters
Krange <- c(2, 4)

# Determine the numbers of class-specific clusters
# using a small nstart for a faster example
KLres <- decideK(
  mcccadata,
  Krange = Krange,
  nstart = 10
)

# Plot the cluster validity indices
plot(KLres)
```

---

plot.mccca *plot* mccca *object.*

Description

Shows a biplot using quantifications estimated by MCCCA.

Usage

```
## S3 method for class 'mccca'
plot(
  x,
  main = "MCCCA result",
  xlim = NULL,
  ylim = NULL,
  plot.ind = FALSE,
  classlabel = NULL,
  classlabel.legend = NULL,
  catelabel = NULL,
  include.variname = TRUE,
  variname = NULL,
  show.onlyminmax = FALSE,
  break.size = NULL,
  scale.gamma = TRUE,
  scatter.level = 2,
  txtsize = 3.8,
  alp.trans = 0.4,
  show.inertia = FALSE,
  ...
)
```

Arguments

`x`	A `mccca` object, a list created by `MCCCA`.
`main`	A character string giving the title of biplot. The default is "MCCCA result".
`xlim`	A numeric vector of length 2 giving the plotting range for the x (horizontal) axis. If `NULL`, the range is automatically determined.
`ylim`	A numeric vector of length 2 giving the the plotting range for the y (vertical) axis (the same role as `xlim`).
`plot.ind`	If `TRUE`, coordinates of observations are also plotted. The default is `FALSE`.
`classlabel`	A character vector of length C (C:the number of classes) giving labels for each class to be displayed on the biplot. If `NULL`, `classlabel` saved in `mcccadata` is used.
`classlabel.legend`	A character vector of length C giving labels for each classes to be shown on the legend. If `NULL`, `classlabel` is used for the legend.
`catelabel`	A character vector of length Q (Q: the total number of categories) giving labels for each category to be displayed on the biplot. If `NULL`, `rownames(B)` is used.
`include.variname`	If `TRUE`, the variable name is included in the category label on the biplot (e.g., the category point named "male" for the variable named "v1" is displayed as "v1:male" on the biplot). The default is `TRUE`.
`variname`	A character vector of length J (J:the number of active variables) giving labels for each variable to be displayed on the biplot. If `NULL`, the variable names included in the row names of `B` or `Bg` are used.

`show.onlyminmax`
: If `TRUE`, only the bubbles corresponding to the maximum and minimum cluster sizes are displayed on the legend. See details. The default is `FALSE`.

`break.size`
: A numeric vector that adjusts the size of bubble displayed on the legend. See details.

`scale.gamma`
: If `TRUE`, quantifications are scaled such that the average squared deviation from the origin of the row and column points is the same. See details. The default is `TRUE`.

`scatter.level`
: A numeric value that adjusts the scatter of points on the biplot. Larger values spread the labels farther apart to reduce overlap. The default is 2.

`txtsize`
: A numeric value that adjusts the text size of labels on the biplot. The default is 3.8.

`alp.trans`
: A numeric value from 0 to 1 which adjusts the transparency of the bubble point. The default is 0.4.

`show.inertia`
: If `TRUE`, the axis labels show the percentages of adjusted inertia obtained by applying MCA to the original data. The default is `FALSE`.

`...`
: Additional arguments passed to `print`.

Details

By default, the breaks in the cluster-size legend are determined automatically. Use `break.size` to specify the cluster sizes displayed in the legend. Alternatively, if `show.onlyminmax = TRUE`, only the minimum and maximum cluster sizes are displayed.

The class-specific cluster points and category points may have different spreads in the original coordinate system. By default, the coordinates are rescaled so that the average squared distances from the origin are equal for the two sets of points. To use the original coordinates, set `scale.gamma = FALSE`.

References

Takagishi, M. and van de Velden, M. (2022), Visualizing class specific heterogeneous tendencies in categorical data. *Journal of Computational and Graphical Statistics*, 31(3), pp.790-801.

See Also

`MCCCA`, `summary.mccca`

Examples

```
# Load the example data
data(mealDrink)

# Prepare active and external variables
active<-c(1,2)
external<-c(3,4)
dat.act=mealDrink[,active]
dat.ext=mealDrink[,external]
```

```
# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Specify the numbers of class-specific clusters
K.vec <- c(2, 1, 2, 2)

# Apply MCCCA
res <- MCCCA(mcccadata, K.vec = K.vec)

# Display the MCCCA biplot
shortlabels <- c("AF", "AM", "JF", "JM")
plot(res, classlabel = shortlabels)

# Show a numerical summary
summary(res)
```

---

rcpp_hello_world *Simple function using Rcpp*

Description

Simple function using Rcpp

Usage

```
rcpp_hello_world()
```

Examples

```
## Not run:
rcpp_hello_world()

## End(Not run)
```

---

summary.calcIndex *Summarize Cluster-Number Selection Results*

Description

Produces a numerical summary of a `calcIndex` object created by `decideK`.

Usage

```
## S3 method for class 'calcIndex'
summary(object, ...)
```

Arguments

object	A calcIndex object created by decideK.
...	Additional arguments passed to the summary method.

See Also

decideK, plot.calcIndex

Examples

```
# Load the example data
data(accident)

# Prepare active and external variables
active<-c(1:4)
external<-c(5:6)
dat.act<-accident[,active]
dat.ext<-accident[,external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Specify the candidate numbers of clusters (2 to 8)
Krange=c(2,4)

# Determine the numbers of class-specific clusters using the default KL index
# (using a small nstart for a faster example)
KLres=decideK(mcccadata,Krange=Krange,nstart=10)

# Show the calculated cluster validity indices
summary(KLres)
```

---

summary.mccca *Summarize MCCCA Results*

Description

Produces a numerical summary of an mccca object created by MCCCA.

Usage

```
## S3 method for class 'mccca'
summary(object, ...)
```

Arguments

object	An mccca object created by MCCCA.
...	Additional arguments passed to the summary method.

See Also

MCCCA, plot.mccca

Examples

```
# Load the example data
data(mealDrink)

# Prepare active and external variables
active<-c(1,2)
external<-c(3,4)
dat.act=mealDrink[,active]
dat.ext=mealDrink[,external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Specify the numbers of class-specific clusters
K.vec <- c(2, 1, 2, 2)

# Apply MCCCA
res <- MCCCA(mcccadata, K.vec = K.vec)

# Display the MCCCA biplot
shortlabels <- c("AF", "AM", "JF", "JM")
plot(res, classlabel = shortlabels)

# Show a numerical summary
summary(res)
```

---

summary.mcccadata *Summarize an mcccadata Object*

Description

Produces a numerical summary of an mcccadata object created by create.MCCCAdata.

Usage

```
## S3 method for class 'mcccadata'
summary(object, ...)
```

Arguments

object	An mcccadata object created by create.MCCCAdata.
...	Additional arguments passed to the summary method.

See Also

create.MCCCAdata

Examples

```
# Load the example data
data(mealDrink)

# Prepare active and external variables
active<-c(1,2)
external<-c(3,4)
dat.act=mealDrink[,active]
dat.ext=mealDrink[,external]

# Create a mcccadata object
mcccadata <- create.MCCCAdata(dat.act, dat.ext)

# Show a summary of the created classes
summary(mcccadata)
```

Index

∗ **datasets**
    accident, 3
    mealDrink, 13

accident, 3

create.MCCCAdata, 3, *6*, *10*, *12*, *19*, *20*

decideK, 5, *14*, *17*, *18*

generate.ext, 8
generate.onedata, *8*, 9, *9*

MCCCA, *3*, *4*, 10, *14–16*, *18*, *19*
mccca *(*mccca-package*)*, 2
mccca-package, 2
mealDrink, 13

plot.calcIndex, *7*, 13, *18*
plot.mccca, *12*, 14, *19*
print, *16*

rcpp_hello_world, 17

summary.calcIndex, *7*, *14*, 17
summary.mccca, *12*, *16*, 18
summary.mcccadata, *4*, 19