Package ‘BCA’

February 19, 2015

Type Package

Title Business and Customer Analytics

Version 0.9-3

Date 2014-09-01

Author Dan Putler <putler@yahoo.com>

Maintainer Dan Putler <putler@yahoo.com>

Depends R (>= 3.0.0)

Imports RcmdrMisc (>= 1.0-1), Rcmdr (>= 2.1-0), car (>= 2.0-21),
        rpart, flexclust, clv

Suggests rgl

LazyLoad yes

Description Underlying support functions for RcmdrPlugin.BCA and a
        companion to the book Customer and Business Analytics: Applied
        Data Mining for Business Decision Making Using R by Daniel S.
        Putler and Robert E. Krider

License GPL (>= 2)

URL http://www.customeranalyticsbook.com

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-04 06:54:56

R topics documented:

Athletic ................................................... 2
bootCVD .................................................. 3
bpCent .................................................... 4
bpCent3d ............................................... 6
CCS ....................................................... 7
create.samples ......................................... 9
Eggs ..................................................... 10
The Athletic data set has 168 observations and 7 variables. The data come from a survey of stakeholders different (students, alumni, faculty, and athletic department employees) of a large US state university. The variables in the data set are conjoint analysis based relative importance weights for seven potential indicators of intercollegiate athletic program success.

```r
data(Athletic)
```

This data set contains the following variables:

- **Win** The importance of winning (won/loss record percentage) in determining the respondent’s judgment of an intercollegiate athletic program’s success.
- **Grad** The importance of student athlete graduation rates in determining the respondent’s judgment of an intercollegiate athletic program’s success.
- **Violat** The importance of NCAA rule violations in determining the respondent’s judgment of an intercollegiate athletic program’s success.
- **Attnd** The importance of home game attendance in determining the respondent’s judgment of an intercollegiate athletic program’s success.
- **Fem** The importance of gender equity (based on the ratio of female to male student athletes) in determining the respondent’s judgment of an intercollegiate athletic program’s success.
- **Teams** The importance of the number of different sports teams in determining the respondent’s judgment of an intercollegiate athletic program’s success.
- **Finan** The importance of the financial success of the program in determining the respondent’s judgment of an intercollegiate athletic program’s success.
Source

Wolfe, Richard A. and Daniel S. Putler (2002), "How Tight are the Ties that Bind Stakeholder Groups?, Organization Science, 13(January-February), 64-82.

Cluster Solution Diagnostics Using Bootstrap Replicates

Description

Provides a plot of both the Rand index and the Calinski-Harabas index for different numbers of clusters for a common underlying dataset using either the K-Means, K-Medians, or Neural Gas clustering algorithms based on a set of bootstrap replicates of the data.

Usage

\[
\text{bootCVD}(x, k, \text{nboot}=100, \text{nrep}=1, \text{method}=c("kmn", "kmd", "neuralgas"), \text{col1}, \text{col2}, \text{dsname})
\]

\[
\text{bootCH}(\text{xdat}, \text{k_vals}, \text{clstr1}, \text{clstr2}, \text{cntrs1}, \text{cntrs2}, \text{method}=c("kmn", "kmd", "neuralgas"))
\]

\[
\text{bootPlot}(\text{fc}, \text{ch}, \text{col1}="\text{blue}" , \text{col2}="\text{green}"
\]

Arguments

- \(x\) A numeric matrix of the data to be clustered.
- \(k\) An integer vector giving the set of clustering solutions to be examined.
- \(\text{nboot}\) The number of bootstrap replicates to use for the assessment.
- \(\text{nrep}\) The number of each set of initial cluster seeds on which to base a solution.
- \(\text{method}\) The clustering method, one of "kmn" (K-Means), "kmd" (K-Medians), and "neuralgas" (neural gas).
- \(\text{col1}\) The color to use for the plot of the Rand index values.
- \(\text{col2}\) The color to use for the plot of the Calinski-Harabas index values.
- \(\text{dsname}\) The name of the dataset being used (used only for output purposes).
- \(\text{xdat}\) A numeric matrix of the data to be clustered.
- \(\text{k_vals}\) An integer vector giving the set of clustering solutions to be examined.
- \(\text{clstr1}\) The cluster assignments from a bootFlexclust object for one side of the Rand index paired comparisons.
- \(\text{clstr2}\) The cluster assignments from a bootFlexclust object for the other side of the Rand index paired comparisons.
- \(\text{cntrs1}\) The cluster centers from a bootFlexclust object for one side of the bootFlexclust Rand index paired comparisons.
- \(\text{cntrs2}\) The cluster centers from a bootFlexclust object for the other side of the bootFlexclust Rand index paired comparisons.
- \(\text{fc}\) A bootFlexclust object.
- \(\text{ch}\) A matrix of Calinski-Harabas index values from bootCH.
Details

The Rand index provides a measure of cluster stability, with relatively higher values indicating relatively more stable clusters, and the Calinski-Harabas index gives a ratio of cluster separation to cluster homogeneity, with higher values of the index being comparatively more preferred. The use of bootstrap replicates addresses both potential randomness in both the sample data and the clustering algorithms.

Value

The functions `bootcvd` and `bootplot` return invisibly. Their benefit is the side effect plot produced and the printed summary of the index values. The function `bootCH` a matrix of Calinski-Harabas index values, the rows are the replicates, and each column corresponds to a particular number of clusters for a solution.

Author(s)

Dan Putler

References


See Also

`bootFlexclust`

---

**bpCent**

*Two Dimensional Biplot of a Clustering Solution*

Description

Plot a biplot of a clustering solution on the current graphics device.

Usage

```r
bpCent(pc, clsAsgn, data.pts = TRUE, centroids = TRUE, choices = 1:2, scale = 1, pc.biplot=FALSE, var.axes = TRUE, col, cex = rep(par("cex"), 2), xlabs = NULL, ylabs = NULL, expand=1, xlim = NULL, ylim = NULL, arrow.len = 0.1, main = NULL, sub = NULL, xlab = NULL, ylab = NULL, ...)```

Arguments

- **pc**: The `prcomp` object of the data used in clustering.
- **clsAsgn**: A vector containing the cluster assignment for each record in the clustering data.
- **data.pts**: If TRUE the point for each record is plotted.
- **centroids**: If TRUE the centroid for each cluster is plotted.
- **choices**: length 2 vector specifying the components to plot.
- **scale**: The variables scaled by \( \lambda^\text{scale} \) and the observations are scaled by \( \lambda^\text{(1-scale)} \), where \( \lambda \) are the eigen values of the principal components solution. scale should be between 0 and 1.
- **pc.biplot**: If true, then \( \lambda = 1 \) and the observations are are scaled up the \( \sqrt{n} \) and the variables scaled down by \( \sqrt{n} \). In this case the inner product between variables approximate covariances, and the distances between observations approximate Mahalanobis distance. Gabriel refers to this as a "principal component biplot".
- **var.axes**: If TRUE the second set of points have arrows representing them as (unscaled) axes.
- **col**: A vector of length 2 giving the colours for the first and second set of points respectively (and the corresponding axes). If a single colour is specified it will be used for both sets. If missing the default colour is looked for in the palette: if there it and the next colour as used, otherwise the first two colours of the palette are used.
- **cex**: The character expansion factor used for labelling the points. The labels can be of different sizes for the two sets by supplying a vector of length two.
- **xlabs**: A vector of character strings to label the first set of points: the default is to use the row dimname of \( x \), or \( 1:n \) is the dimname is NULL.
- **ylabs**: A vector of character strings to label the second set of points: the default is to use the row dimname of \( y \), or \( 1:n \) is the dimname is NULL.
- **expand**: An expansion factor to apply when plotting the second set of points relative to the first. This can be used to tweak the scaling of the two sets to a physically comparable scale.
- **arrow.len**: The length of the arrow heads on the axes plotted in var.axes is true. The arrow head can be suppressed by arrow.len = 0.
- **xlim, ylim**: Limits for the x and y axes in the units of the first set of variables.
- **main, sub, xlab, ylab, ...**: graphical parameters.

Details

This function is a reworking of `biplot` in order to allow the cluster centroids of a clustering solution to be displayed as well as the individual data points. If both `data.pts` and centroids are set to FALSE then only the variable directional vectors are displayed.

Graphical parameters can also be given to `biplot`.
Side Effects

A plot is produced on the current graphics device.

See Also

biplot
xlabs A vector of character strings to label the first set of points: the default is to use the row dimname of x, or 1:n is the dimname is NULL.

ylabs A vector of character strings to label the second set of points: the default is to use the row dimname of y, or 1:n is the dimname is NULL.

xlim, ylim, zlim Limits for the x, y, z axes in the units of the first set of variables.

xlab, ylab, dim.lab, fov Graphical parameters for rgl.

Details
This function is a reworking of biplot in order to allow the cluster centroids of a clustering solution to be displayed as well as the individual data points, and extend it to three dimensions. The use of arrow heads is discouraged since their creation is very resource intensive. If both dataNpts and centroids are set to FALSE then only the variable directional vectors are displayed.

Side Effects
A plot is produced on the current graphics device.

See Also
biplot

---

CCS 

Charitable Giving Dataset

Description
The CCS data set has 1600 observations and 20 variables, and is from the data is from the British Columbia and Yukon region of a nationwide Canadian charitable society. The data examine whether a donor joined the Monthly Giver Program, in which he or she opts to make a pre-arranged monthly donation to the charity by credit card. The actual monthly givers have been highly over-sampled. In the charity’s database they represent one percent of donors, but they comprise 50 percent of this sample. This level of oversampling is common in data mining applications.

Usage

data(CCS)

Format
This data set contains the following variables:

MonthGive A factor indicating whether the donor joined the monthly giver program in the last two annual campaigns with levels:
No
Yes
Region A factor indicating where in British Columbia or the Yukon the donor resides with levels:
   R1 (Vancouver Island)
   R2 (Greater Vancouver)
   R3 (The Fraser Valley)
   R4 (The North Coast of BC)
   R5 (The Central and Southern Interior of BC)
   R6 (The Northern Interior of BC and the Yukon)

YearsGive The number of years the individual has given to the Canadian Cancer Society.
AveDonAmt The average dollar amount of past donations by the donor.
LastDonAmt The dollar amount of the last donation by the donor.
DonPerYear The rate of donating to the Canadian Cancer Society measured as the number of donations per year by the donor.
NewDonor A factor that indicates whether the individual is a first time donor with levels.
   No
   Yes

Age20t29 The percentage of people between age 20 and 29 residing in the Enumeration Area in which the donor resides.
Age20t39 The percentage of people between age 20 and 39 residing in the Enumeration Area in which the donor resides.
Age60pls The percentage of people 60 years of age or older residing in the Enumeration Area in which the donor resides.
Age70pls The percentage of people 70 years of age or older residing in the Enumeration Area in which the donor resides.
Age80pls The percentage of people 80 years of age or older residing in the Enumeration Area in which the donor resides.
AdultAge The average age of adult residents in the Enumeration Area in which the donor resides.
SomeUnivP The percentage of adults in the postal code in which the donor resides that have an educational attainment of some college or university attendance.
FinUnivP The percentage of adults in the Enumeration Area in which the donor resides that have a university degree.
hh1t2mem The percentage of households in the Enumeration Area in which the donor resides that have only one or two members present.
hh1mem The percentage of households in the Enumeration Area in which the donor resides that have only a single member present.
AveIncEA The average pre-tax income of households in the Enumeration Area in which the donor resides.
DwelValEA The average dwelling value in the Enumeration Area in which the donor resides.
EngPrmLang The percentage of households in the Enumeration Area in which the donor resides that have have English as their primary language.

Source
An anonymous charity.
Create a Sample Membership Character Variable

**Description**

Provides a character vector with possible values of "Estimation", "Validation" and "Holdout" that can then be used to assign observations of a data frame to estimation, validation, or (optionally) holdout samples using the subset option of a variety of functions.

**Usage**

```r
create.samples(x, est=0.34, val=0.33, rand.seed=1)
```

**Arguments**

- **x** A data frame.
- **est** The percentage of the total sample to allocate to the estimation sample. The value of `est` should range from zero to one.
- **val** The percentage of the total sample to allocate to the validation sample. The value of `val` should range from zero to one.
- **rand.seed** A parameter passed to `set.seed` for to specify the seed of the random number generator.

**Details**

The values of `est` and `val` should sum to a value between zero and one. If greater than one, an error is returned. If less than one, the remaining percentage of the sample is allocated to the holdout sample.

**Value**

A character vector with possible values of "Estimation", "Validation" and (optionally) "Holdout". The length of the vector equals the number of rows in the original data frame.

**Author(s)**

Dan Putler

**See Also**

`set.seed`
Examples

```r
data(CCS)
# Create a new set of samples with 40 percent in each of the estimation and
# validation samples, and 20 percent in the holdout sample.
CCS$Sample <- create.samples(CCS, est=0.4, val=0.4)
```

---

### Retail Sales of Eggs

**Description**

The Eggs data set has 105 observations and 10 variables. The data contains information on weekly sales of eggs in Southern California over a two year period.

**Usage**

```r
data(Eggs)
```

**Format**

This data set contains the following variables:

- **Week** The observation week (1 to 105). This variable can be used as a time trend.
- **Month** A factor that gives the name of the month in which the observation occurred.
- **First.Week** A factor indicating whether the observation fell on the first week of the month with levels:
  - No
  - Yes
- **Easter** A factor that indicates whether the observation fell the week prior to the week containing Easter Sunday, the week containing Easter Sunday, the week following the week containing Easter Sunday, or a non-Easter week with levels:
  - Non Easter
  - Pre Easter
  - Easter
  - Post Easter
- **Cases** Retail sales of eggs in cases.
- **Egg.Pr** Average retail egg price in cents per dozen.
- **Beef.Pr** Average retail price of 7-bone beef roast in cents per pound.
- **Pork.Pr** Average retail price of strip bacon in cents per pound.
- **Chicken.Pr** Average retail price of whole frying chicken in cents per pound.
- **Cereal.Pr** Average retail price of Cheerios breakfast cereal in cents per pound.

**Source**

Putler (1992)
Description

The jack.jill data set has 557 observations and 8 variables. The data contains information on children’s apparel spending and household level demographic and socioeconomic information for a sample of households residing in Alberta and British Columbia.

Usage

data(jack.jill)

Format

This data set contains the following variables:

SPENDING  Dollars spent on children’s apparel over a one-year period.

CHILDREN  Pre-tax income given as a factor with levels:
            1
            2
            3
            4+

INCOME    Pre-tax income given as a factor with levels:
            $0k–$20k
            $20k–$30k
            $30k–$40k
            $40k–$50k
            $50k–$60k
            $60k–$75k
            $75k–$100k
            $100k+

EMPLOYMENT The employment status of the female head of household with levels:
            No female head
            Unemployed
            Part-time
            Full-time

AGE      Age of the female head of household given as a factor with levels:
            No female head
            29 and under
            30 to 39
            40 to 49
            50 to 59
            60 and over
EDUCATION  The educational attainment of the female head of household given as a factor with levels:
No female head
Not stated
Elementary or less
Some or completed secondary
Some post-secondary
Post-secondary diploma
University degree

OCCUPATION  The occupation group of the female head of household with levels:
No female head
Blue collar
Pink collar
White collar
Other
Non-working or retired

BIRTHCOUNTRY  The birth country of the female head of household with levels:
No female head
Canada
US, N&W Europe
S&E Europe
Asia and Oceania
Other (Caribbean, Middle East, and Africa)

Source
Statistics Canada

---

**lift.chart**  
*Lift Charts to Compare Binary Predictive Models*

**Description**

Provides either a total cumulative response or incremental response rate lift chart for the purposes of comparing the predictive capability of different binary predictive models.

**Usage**

```r
lift.chart(modellist, data, targLevel, trueResp, type = "cumulative", sub = "")
```

**Arguments**

- `modellist`  
  A character vector containing the names of the different models to be compared. The selected models must have the same y variable that must be a binary factor, and have been estimated using the same data set.
**data**
The dataframe that constitutes the comparison sample. If this dataframe is not the same as the dataframe used to estimated models, the dataframe must contain all the variables used in the models to be compared.

**targLevel**
The label for the level of the binary factor of interest. For example, in a database marketing application, this level could be "Yes" for a variable that takes on the values "Yes" and "No" to indicate if a customer responded favorably to a promotion offer.

**trueResp**
The true rate of the target level for the master database the estimation and comparison dataframes were originally drawn from.

**type**
A character string that must either have the value of "cummulative" (to produce a total cumulative response chart) or "incremental" (to produce an incremental response rate chart).

**sub**
A sub-title for the plot, typically to identify the sample used.

**Details**
Lift charts are a commonly used tool in business data mining applications. They are used to assess how well a model is able to predict a desirable (from an organization’s point-of-view) response on the part of a customer compared to alternative estimated models and a benchmark model of approaching customers randomly. The total cumulative response chart shows the percentage of the total response the organization would receive from only contacting a given percentage (grouped by deciles) of its entire customer base. This chart is best for selecting between alternative models, and in predicting the revenues the organization will receive by contacting a given percentage of their customers that the model predicts are most likely to favorably respond. The incremental response rate chart provides the response rate among each of ten decile groups of the organization’s customers, with the decile groups ordered by their estimated likelihood of a favorable response.

**Value**
The function returns invisibly. Its benefit is the side effect plot produced.

**Author(s)**
Dan Putler

**Examples**
```
library(rpart)
layout(matrix(c(1,2), 2, 1))
data(CCS)
CCS$Sample <- create.samples(CCS, est=0.4, val=0.4)
CCSEst <- CCS[CCS$Sample == "Estimation",]
CCS.glm <- glm(MonthGive ~ DonPerYear + LastDonAmt + Region + YearsGive,
               family=binomial(logit), data=CCSEst)
library(rpart)
CCS.rpart <- rpart(MonthGive ~ DonPerYear + LastDonAmt + Region + YearsGive,
                    data=CCSEst, cp=0.0074)
CCSVal <- CCS[CCS$Sample == "Validation",]
lift.chart(c("CCS.glm", "CCS.rpart"), data=CCSVal, targLevel="Yes",)
```
Relabel Factor Levels

Description

Relabel the levels of factors to provide more descriptive names and reduce the number of factor levels.

Usage

relabel.factor(x, new.labels, old.labels=levels(x))

Arguments

x A factor.
new.labels The new factor level labels.
old.labels The old factor level labels.

Details

The number of new factor labels/levels must be less than the number of labels/levels than the original factor.

Value

A factor whose length is equal to the old factor.

Author(s)

Dan Putler
Description

A minor modification of the car package’s scatterplot function that makes enhanced scatterplots, with boxplots in the margins, a loess smooth, smoothed conditional spread, outlier identification, and a regression line; sp is an abbreviation for scatterplot.

Usage

```r
scatterplotBCA(x, ...) 
```

```
## S3 method for class 'formula'
scatterplotBCA(x, data, subset, xlab, ylab, legend.title, legend.coords, labels, ...)

## Default S3 method:
scatterplotBCA(x, y, smooth = TRUE, spread = !by.groups, 
               span = 0.5, loess.threshold = 2, reg.line = lm, 
               boxplots = if (by.groups) "" else "xy", xlab = deparse(substitute(x)), 
               ylab = deparse(substitute(y)), las = par("las"), lwd = 2, 
               lwd.smooth = lwd, lwd.spread = lwd, lty = 1, lty.smooth = lty, 
               lty.spread = 2, labels, id.method = "mahal", 
               id.n = if(id.method[1] == "identify") length(x) else 0, id.cex = 1, 
               id.col = palette()[1], log = "", jitter = list(), xlim = NULL, 
               ylim = NULL, cex = par("cex"), cex.axis = par("cex.axis"), 
               cex.lab = par("cex.lab"), cex.main = par("cex.main"), 
               cex.sub = par("cex.sub"), groups, by.groups = !missing(groups), 
               legend.title = deparse(substitute(groups)), legend.coords, 
               ellipse = FALSE, levels = c(0.5, 0.95), robust = TRUE, col = if 
               (n.groups == 1) palette()[c(2, 1, 3)] else rep(palette(), 
               length = n.groups), pch = 1:n.groups, legend.plot = !missing(groups), 
               reset.par = TRUE, grid = TRUE, ...) 
```

Arguments

- `x` vector of horizontal coordinates, or a “model” formula, of the form `y ~ x` or (to plot by groups) `y ~ x | z`, where `z` evaluates to a factor or other variable dividing the data into groups. If `x` is a factor, then parallel boxplots are produced using the `Boxplot` function.
- `y` vector of vertical coordinates.
- `data` data frame within which to evaluate the formula.
- `subset` expression defining a subset of observations.
smooth if TRUE (the default) a loess nonparametric regression line is drawn on the plot.

spread if TRUE (the default when there are no groups), a smoother is applied to the root-mean-square positive and negative residuals from the loess line to display conditional spread and asymmetry.

span span for the loess smoother.

loess.threshold suppress the loess smoother if there are fewer than loess.threshold unique values (default, 5) of y.

reg.line function to draw a regression line on the plot or FALSE not to plot a regression line.

boxplots if "x" a boxplot for x is drawn below the plot; if "y" a boxplot for y is drawn to the left of the plot; if "xy" both boxplots are drawn; set to "" or FALSE to suppress both boxplots.

xlab label for horizontal axis.

ylab label for vertical axis.

las if P, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see par).

lwd width of linear-regression lines (default 1).

lwd.smooth width for smooth regression lines (default is the same as lwd).

lwd.spread width for lines showing spread (default is the same as lwd).

lty type of linear-regression lines (default 1, solid line).

lty.smooth type of smooth regression lines (default is the same as lty).

lty.spread width for lines showing spread (default is 2, broken line).

id.method,id.n,id.cex,id.col Arguments for the labelling of points. The default is id.n=0 for labeling no points. See showLabels for details of these arguments. If the plot uses different colors for groups, then the id.col argument is ignored and label colors are determined by the col argument.

labels a vector of point labels; if absent, the function tries to determine reasonable labels, and, failing that, will use observation numbers.

log same as the log argument to plot, to produce log axes.

jitter a list with elements x or y or both, specifying jitter factors for the horizontal and vertical coordinates of the points in the scatterplot. The jitter function is used to randomly perturb the points; specifying a factor of 1 produces the default jitter. Fitted lines are unaffected by the jitter.

xlim the x limits (min, max) of the plot; if NULL, determined from the data.

ylim the y limits (min, max) of the plot; if NULL, determined from the data.

groups a factor or other variable dividing the data into groups; groups are plotted with different colors and plotting characters.

by.groups if TRUE, regression lines are fit by groups.

legend.title title for legend box; defaults to the name of the groups variable.
legend.coords  coordinates for placing legend; an be a list with components x and y to specify
the coordinates of the upper-left-hand corner of the legend; or a quoted keyword,
such as "topleft", recognized by legend.

ellipse        if TRUE data-concentration ellipses are plotted.
levels         level or levels at which concentration ellipses are plotted; the default is c(.5, .95).
robust         if TRUE (the default) use the cov.trob function in the MASS package to calculate
the center and covariance matrix for the data ellipses.

col             colors for lines and points; the default is taken from the color palette, with
palette()[2] for nonparametric regression lines and palette()[1] for linear
regression line and points if there are no groups, and successive colors for
the groups if there are groups.
pch             plotting characters for points; default is the plotting characters in order (see par).
cex, cex.axis,  cex.lab, cex.main, cex.sub
set sizes of various graphical elements; (see par).
legend.plot    if TRUE then a legend for the groups is plotted in the upper margin.
reset.par      if TRUE then plotting parameters are reset to their previous values when scatterplot
exits; if FALSE then the mar and mfcol parameters are altered for the current plotting
device. Set to FALSE if you want to add graphical elements (such as lines)
to the plot.
...            other arguments passed down and to plot.
grid           If TRUE, the default, a light-gray background grid is put on the graph

Value

If points are identified, their labels are returned; otherwise NULL is returned invisibly.

Author(s)

John Fox with modifications made by Dan Putler

See Also

scatterplot

---

Description

A minor modification of the car package’s scatterplotMatrix function that makes enhanced scatter-
plot matrices with univariate displays down the diagonal; spmBCA is an abbreviation for scatterplotMatrixBCA.
This function just sets up a call to pairs with custom panel functions.
Usage

scatterplotMatrixBCA(x, ...)

## S3 method for class 'formula'
scatterplotMatrixBCA(x, data=NULL, subset, labels, ...)

## Default S3 method:
scatterplotMatrixBCA(x, var.labels = colnames(x), diagonal = c("density", "boxplot", "histogram", "oned", "qqplot", "none"),
                      adjust = 1, nclass, plot.points = TRUE, smooth = TRUE,
                      spread = smooth & & !by.groups, span = 0.5,
                      loess.threshold = 2, reg.line = lm, transform = FALSE,
                      family = c("bcPower", "yjPower"), ellipse = FALSE,
                      levels = c(0.5, 0.95), robust = TRUE, groups = NULL,
                      by.groups = FALSE, labels, id.method = "mahal", id.n =
                      0, id.cex = 1, id.col = palette()[1], col = if
                      (n.groups == 1) palette()[[c(2, 1, 3)]] else rep(palette(),
                      length = n.groups), pch = 1:n.groups, lwd = 2,
                      lwd.smooth = lwd, lwd.spread = lwd, lty = 1,
                      lty.smooth = lty, lty.spread = 2, cex = par("cex"),
                      cex.axis = par("cex.axis"), cex.labels = NULL,
                      cex.main = par("cex.main"), legend.plot =
                      length(levels(groups)) > 1, row1atop = TRUE, ...)

spmBCA(x, ...)

Arguments

x

a data matrix, numeric data frame, or a one-sided “model” formula, of the form
~ x1 + x2 + ... + xk or ~ x1 + x2 + ... + xk | z where z evaluates
to a factor or other variable to divide the data into groups.

data

for scatterplotMatrix. formula, a data frame within which to evaluate the
formula.

subset

expression defining a subset of observations.

labels, id.method, id.n, id.cex, id.col
Arguments for the labelling of points. The default is id.n=0 for labeling no
points. See showLabels for details of these arguments. If the plot uses different
colors for groups, then the id.col argument is ignored and label colors are
determined by the col argument.

var.labels

variable labels (for the diagonal of the plot).

diagonal

contents of the diagonal panels of the plot.

adjust

relative bandwidth for density estimate, passed to density function.

nclass

number of bins for histogram, passed to hist function.

plot.points

if TRUE the points are plotted in each off-diagonal panel.
smooth  if TRUE a loess smooth is plotted in each off-diagonal panel.
spread if TRUE (the default when not smoothing by groups), a smoother is applied to the root-mean-square positive and negative residuals from the loess line to display conditional spread and asymmetry.
span span for loess smoother.
loess.threshold suppress the loess smoother if there are fewer than loess.threshold unique values (default, 5) of the variable on the vertical axis.
reg.line if not FALSE a line is plotted using the function given by this argument; e.g., using rlm in package MASS plots a robust-regression line.
transform if TRUE, multivariate normalizing power transformations are computed with powerTransform, rounding the estimated powers to 'nice' values for plotting; if a vector of powers, one for each variable, these are applied prior to plotting. If there are groups and by.groups is TRUE, then the transformations are estimated conditional on the groups factor.
family family of transformations to estimate: "bcPower" for the Box-Cox family or "yjPower" for the Yeo-Johnson family (see powerTransform).
ellipse if TRUE data-concentration ellipses are plotted in the off-diagonal panels.
levels levels or levels at which concentration ellipses are plotted; the default is c(0.5, 0.9).
robust if TRUE use the cov.trob function in the MASS package to calculate the center and covariance matrix for the data ellipses.
groups a factor or other variable dividing the data into groups; groups are plotted with different colors and plotting characters.
by.groups if TRUE, regression lines are fit by groups.
pch plotting characters for points; default is the plotting characters in order (see par).
col colors for lines and points; the default is taken from the color palette, with palette()[2] for nonparametric regression lines and palette()[1] for linear regression lines and points if there are no groups, and successive colors for the groups if there are groups.
lwd width of linear-regression lines (default 1).
lwd.smooth width for smooth regression lines (default is the same as lwd).
lwd.spread width for lines showing spread (default is the same as lwd).
lty type of linear-regression lines (default 1, solid line).
lty.smooth type of smooth regression lines (default is the same as lty).
lty.spread width for lines showing spread (default is 2, broken line).
cex, cex.axis, cex.labels, cex.main set sizes of various graphical elements (see par).
legend.plot if TRUE then a legend for the groups is plotted in the first diagonal cell.
rowlattop If TRUE (the default) the first row is at the top, as in a matrix, as opposed to at the bottom, as in graph (argument suggested by Richard Heiberger).
... arguments to pass down.
**Value**

NULL. This function is used for its side effect: producing a plot.

**Author(s)**

John Fox with minor modifications by Dan Putler

**See Also**

scatterplotMatrix

---

**score**

*Score a Database based on a Predictive Model*

**Description**

Provides either an integer vector that contains the "desirability" rank of a case in a data set, the fitted probability of a desired response, or the fitted probability adjusted for the true response rate based on the fitted values of a predictive model.

**Usage**

```r
rankScore(model, data, targLevel)
rawProbScore(model, data, targLevel)
adjProbScore(model, data, targLevel, trueResp)
```

**Arguments**

- `model`: A character string containing the name of the model to use to score the database.
- `data`: A data frame of the database to be scored. All the predictor variables of the model need to be among the variables of the data frame.
- `targLevel`: The "desired" level of the y variable factor as a character string.
- `trueResp`: The true "desired" response rate for the overall population of interest.

**Details**

Only binomial glm, binary rpart, and binary nnet models can be used as the basis of scoring a database.

**Value**

An integer vector that indicates the rank order desirability (a value of 1 means most desirable) of the corresponding case of the database being scored or a probability measure bounded between zero and one.

**Author(s)**

Dan Putler
Description

Provides a wrapper to several function calls in the clv package needed to construct the SD index value for a clustering solution. The number of clusters that has the lowest value of the SD index represents the "best" solution under the criteria used to construct the SD index.

Usage

sdNclv(x, clus, alpha)

Arguments

x A numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a dataframe with all numeric columns) used to construct the clustering solution.
clus The cluster to which each row of x was assigned.
alpha A weight to be placed on the average scattering of the clustering solution.

Details

The SD index corresponds to the weighted sum of the average "scattering" of points within clusters and the inverse of the total separation between clusters. The average scattering measure is based on the average sum of the squared differences between a cluster's centroid all the points in a cluster, while total separation is measured by the sum of the squared distance between cluster centroids. A solution with a low average scattering and a low value of the inverse total separation is considered to be better than a solution with higher levels of these two measures.

Value

A scalar SD index value for the clustering solution.

Author(s)

Dan Putler

References


See Also

clv.SD
Description

Provides a plot of SD cluster validation index values for different numbers of k-means clusters for a common underlying dataset. The number of clusters that has the lowest value of the SD index represents the "best" solution under the criteria used to construct the SD index.

Usage

SDIndex(x, minClust, maxClust, iter.max=10, num.seeds=10)

Arguments

x A numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a dataframe with all numeric columns).

minClust The minimum number of clusters to be considered for a solution.

maxClust The maximum number of clusters to be considered for a solution.

iter.max The maximum number of iterations allowed for a solution.

num.seeds The number of different starting random seeds to use for a solution with a given number of clusters.

Details

The SD index corresponds to the weighted sum of the average "scattering" of points within clusters and the inverse of the total separation between clusters. The average scattering measure is based on the average sum of the squared differences between a clusters centroid all the points in a cluster, while total separation is measured by the sum of the squared distance between cluster centroids. A solution with a low average scattering and a low value of the inverse total separation is considered to be better than a solution with higher levels of these two measures.

Value

The function returns invisibly. Its benefit is the side effect plot produced.

Author(s)

Dan Putler
variable.summary

References

See Also
KMeans, SD.clv

Examples

```r
data(iris)
iris.data <- iris[,1:4]
SDIndex(iris.data, minClust=2, maxClust=6, iter.max=10, num.seeds=10)
```

---

### variable.summary

*Basic summary information of the variables of a data frame*

#### Description
The function returns a data frame where, the row names correspond to the variable names, and a set of columns with summary information for each variable. Its purpose is to allow the user to quickly scan the data frame for potentially problematic variables.

#### Usage

```r
variable.summary(dframe)
```

#### Arguments

- **dframe**: A data frame.

#### Value
The returned data frame contains the variables Class (numeric, integer, factor, or character), missing values, Levels (the levels of a factor variable, or NA for non-factor variables), Min.Level.Size (the number of cases for the smallest level of a factor, or NA for a non-factor), Mean (the mean of non-missing cases for a numeric or integer variable, or NA for factor and character variables), and SD (the standard deviation of non-missing cases for a numeric or integer variable, or NA for factor and character variables).

#### Author(s)
Dan Putler

#### Examples

```r
data(iris)
variable.summary(iris)
```

---

```r
data(iris)
iris.data <- iris[,1:4]
SDIndex(iris.data, minClust=2, maxClust=6, iter.max=10, num.seeds=10)
```
Index

*Topic **cluster**
  bootCVD, 3
  SD.clv, 21
  SIndex, 22
*Topic **datasets**
  Athletic, 2
  CCS, 7
  Eggs, 10
  jack.jill, 11
*Topic **hplot**
  bpCent, 4
  bpCent3d, 6
  scatterplotBCA, 15
  scatterplotMatrixBCA, 17
*Topic **misc**
  create.samples, 9
  lift.chart, 12
  relabel.factor, 14
  score, 20
  variable.summary, 23
*Topic **multivariate**
  bpCent, 4
  bpCent3d, 6

adjProbScore (score), 20
Athletic, 2

biplot, 6, 7
bootCH (bootCVD), 3
bootCVD, 3
bootFlexclust, 4
bootPlot (bootCVD), 3
Boxplot, 15
bpCent, 4
bpCent3d, 6

CCS, 7
clv.SD, 21
create.samples, 9

Eggs, 10

jack.jill, 11
jitter, 16
KMeans, 23
legend, 17
lift.chart, 12
palette, 5, 6
par, 16, 17, 19
plot, 16
powerTransform, 19
rankScore (score), 20
rawProbScore (score), 20
relabel.factor, 14
scatterplot, 17
scatterplotBCA, 15
scatterplotMatrix, 20
scatterplotMatrixBCA, 17
score, 20
SD.clv, 21, 23
SIndex, 22
set.seed, 9
showLabels, 16, 18
spBCA (scatterplotBCA), 15
spmBCA (scatterplotMatrixBCA), 17

variable.summary, 23