PL – Liste des fichiers de **mvpart** contenant de la documentation à vérifier

- gdist, p. 2–3 / Addition de PL surlignée en jaune
- mvpart, p. 4–6 / Rien à signaler (RAS)
- plotcp, p. 7–8 / Correction typographique p. 7
- printcp, p. 9 / RAS
- rpart.pca, p. 10–11 / Une correction dans le titre de la page
- rpconvert, p. 12 / RAS
- scaler, p. 13-14 / Deux corrections dans Details, p. 13
- snip.rpart, p. 15 / RAS
- solder, p. 16–17 / RAS
- spider, p. 18–19 / RAS
- xdiss, p. 20-21 / RAS

Supprimé: F

a mis en forme : Police :Gras

a mis en forme : Droite : -0,41 cm

Supprimé: C

Supprimé: /

gdist {mvpart} R Documentation

R Documentation

Dissimilarity Measures

Description

The function computes useful dissimilarity indices which are known to have a good rank-order relation with gradient separation and are thus efficient in community ordination with multidimensional scaling.

Usage

```
gdist(x, method = "bray", keepdiag = FALSE, full = FALSE, sq = FALSE)
```

Arguments

`x` Data matrix.

`method` Dissimilarity index. Choices are : "manhattan", "euclidean", "canberra", "bray", "kulczynski", "gower", "maximum", "binary", "chisq", "chord", "beta0", "beta1", "beta2"

`keepdiag` Compute amd keep diagonals.

`full` Return the square dissimilarity matrix.

`sq` Square the dissimilarities – useful for distance-based partitioning.

Details

Infamous ”double zeros” are removed in Canberra dissimilarity.

Euclidean and Manhattan dissimilarities are not good in gradient separation without proper standardization but are still included for comparison and special needs.

Some of indices become identical or rank-order similar after some standardization.

Value

Should be interchangeable with `dist` and returns a distance object of the same type.

Note

The function is an alternative to `dist` adding some ecologically meaningful indices. Both methods should produce similar types of objects which can be interchanged in any method accepting either. Manhattan and Euclidean dissimilarities should be identical in both methods, and Canberra dissimilarity may be similar.

Author(s)

Jari Oksanen – modified Glenn De'ath (Dec 03)

References

Faith, D.P, Minchin, P.R. and Belbin, L. (1987) Compositional dissimilarity as a robust measure of ecological distance. *Vegetatio*69, 57-68.

```
function (x, method = "bray", keepdiag = FALSE, full = FALSE,
  sq = FALSE)
{
  METHODS <- c("manhattan", "euclidean", "canberra", "bray",
    "kulczynski", "gower", "maximum", "binary", "chisq",
    "chord", "beta0", "beta1", "beta2")
  method <- pmatch(method, METHODS)
  if (is.na(method))
    stop("invalid distance method")
  N <- nrow(x <- as.matrix(x))
  if (method == 6L)
    x <- scaler(x, col = c("min0", "max1"))
  if (method == 9L) {
    rr <- apply(x, 1L, sum)
    cc <- apply(x, 2L, sum)
    x <- diag(1/sqrt(rr)) %*% x %*% diag(1/sqrt(cc))
    method <- 2L
  }
  else if (method == 10L) {   # Chord distance, normed to maximum=1
    mns <- sqrt(apply(x^2, 1L, sum))
    x <- x/(mns * sqrt(2))
    method <- 2L
  }
  else if (method > 10L)
    method <- method - 2L
  d <- .C("gdistance", x = as.double(x), nr = N, nc = ncol(x),
    d = double((N * (N - 1L))/2L), keepdiag = as.integer(FALSE),
    method = as.integer(method), PACKAGE = "mvpart")$d
  attr(d, "Size") <- N
  class(d) <- "dist"
  if (full)
    d <- distfull(d)
  if (sq)
    d <- d^2
  d
}
```

mvpart {mvpart}

Recursive Partitioning and Regression Trees

Description

Wrapper function for fitting and plotting `rpart` models.

Usage

```
mvpart(
  form,
  data,
  minauto = TRUE,
  size,
  xv = c("1se", "min", "pick", "none"),
  xval = 10,
  xvmult = 0,
  xvse = 1,
  snip = FALSE,
  plot.add = TRUE,
  text.add = TRUE,
  digits = 3,
  margin = 0,
  uniform = FALSE,
  which = 4,
  pretty = TRUE,
  use.n = TRUE,
  all.leaves = FALSE,
  bars = TRUE,
  legend,
  bord = FALSE,
  xadj = 1,
  yadj = 1,
  prn = FALSE,
  branch = 1,
  rsq = FALSE,
  big.pts = FALSE,
  pca = FALSE,
  interact.pca = FALSE,
  wgt.ave.pca = FALSE,
  keep.y = TRUE,
  ...
)
```

Arguments

`form` As for `rpart` function.

`data` Optional data frame in which to interpret the variables named in the formula

`minauto` If `TRUE` uses smart minsplit and minbucket based on N cases.

`size` The size of tree to be generated.

`xv` Selection of tree by cross-validation: `"1se"` - gives best tree within one SE of the overall best, `"min"` - the best tree, `"pick"` - pick the tree size interactively, `"none"` - no cross-validation.

`xval` Number of cross-validations or vector defining cross-validation groups.

`xvmult` Number of multiple cross-validations.

`xvse` Multiplier for the number of SEs used for `xv = "1se"`.

`snip` Interactively prune the tree.

`plot.add` Plot the tree and (optionally) add text.

`text.add` Add output of `text.rpart` to tree.

`digits` Number of digits on labels.

`margin` Margin around plot, 0.1 gives an extra 10 percent space around the plot.

`uniform` Uniform lengths to the branches of the tree.

`which` Which split labels and where to plot them, 1=centered, 2 = left, 3 = right and 4 = both.

`pretty` Pretty labels or full labels.

`use.n` Add number of cases at each node.

`all.leaves` Annotate all nodes.

`bars` If `TRUE` adds barplots to nodes.

`legend` If `TRUE` adds legend for mrt and classification trees.

`bord` Border (box) around the barplots.

`xadj` Adjust the horizontal size of the individual barplots (default = 1).

`yadj` Adjust the vertical size of the individual barplots (default = 1).

`prn` If `TRUE` prints tree details.

`branch` Controls spread of branches: 1=vertical lines, 0=maximum slope.

`rsq` If `TRUE` gives "rsq" plot.

`big.pts` Plot colored points at leaves – useful to link to PCA plot.

`pca` If `TRUE` plots PCA of group means and add species and site information.

`interact.pca` If `TRUE` runs interactive PCA. See `rpart.pca`.

`wgt.ave.pca` If `TRUE` plot weighted averages across sites for species.

`keep.y` If `TRUE` y values are returned.

`...` Other arguments passed to `rpart`.

Value

An `rpart-class` object, which contain a superset of regression or classification trees.

See Also

rpart, rpart.pca

Examples

Run examples

```
## Load the spider data set:
data(spider)

## Splitting the table into the responses (species, Y)
## and the descriptors (X):
Y <- data.matrix(spider[,1L:12L])
X <- spider[,13L:18L]
## Note: the multivariate response needs to procesed using function
## data.matrix before being used as a multivariate response by function
## rpart.

## Multivariate regression tree on the spider species response matrix and all
## the descriptors in the spider data set.

## Using the default settings:
mvpart(Y ~ herbs + reft + moss + sand + twigs + water, data=X)

## Pick-the optimal tree size using cross-validation:
mvpart(Y ~ herbs + reft + moss + sand + twigs + water, data=X, xv="p")

## Pick-the optimal tree size and generate a PCA plot (interactive):
spider1 <- mvpart(Y ~ herbs + reft + moss + sand + twigs + water, data=X,
                  xv="1se", pca=TRUE)

## interactive PCA plot of saved multivariate tree
rpart.pca(spider1, interact=TRUE, wgt.ave=TRUE)
```

[Package *mvpart* version 2.0-1 Index]

plotcp {mvpart} R Documentation

Plot a Complexity Parameter Table for an Rpart Fit

Description

Gives a visual representation of the cross-validation results in an rpart-class object.

Usage

```
plotcp(
  x,
  xvse = 1,
  minline = TRUE,
  lty = 3,
  col = 1,
  upper = c("size", "splits", "none"),
  tab,
  resub.err = TRUE,
  adj.df = FALSE,
  ...
)
```

Arguments

`x`	An object of class `rpart`.
`xvse`	Multiplier for xvse * SE above the minimum of the curve.
`minline`	Whether a horizontal line is drawn 1SE above the minimum of the curve.
`lty`	Type of lines used in plot, default = 3 (a dotted line).
`col`	Color of lines, default = 1 (black).
`upper`	What is plotted on the top axis: the size of the tree (the number of leaves), the number of splits or nothing.
`tab`	Used for multiple cross-validation.
`resub.err`	Uses the re-substitution error for the calculation of SEs.
`adj.df`	Adjusts the degrees of freedom of the re-substitution error estimate in the calculation of SEs.
`...`	Additional graphical parameters

Details

The set of possible cost-complexity prunings of a tree from a nested set. For the geometric means of the intervals of values of `cp`for which a pruning is optimal, a cross-validation has (usually) been done in the initial construction by `rpart`. Member `cptable`in the `rpart-class` object contains the mean and standard deviation of the errors in the cross-validated prediction against each of the geometric means, and these are plotted by this function. A

Commenté [PL1]: Insérer une espace avant « in » good choice of cp for pruning is often the leftmost value for which the mean lies below the horizontal line.

Value

NULL (invisibly).

See Also

rpart, printcp, rpart-class, and mvpart (for an example).

printcp {mvpart} R Documentation

Displays CP table for Fitted Rpart Object

Description

Displays the `cp` table for fitted `rpart` object.

Usage

```
printcp(x, digits = getOption("digits") - 2L)
```

Arguments

`x`	An `rpart-class` object.
`digits`	The number of digits of numbers to print.

Details

Prints a table of optimal pruning based on a complexity parameter.

Value

The `rpart-class` object's cp table.

See Also

`summary.rpart`, `rpart.object`

Examples

Run examples

```
## Load the car data set:
data("car.test.frame")

## Estimating a regression tree with a single descriptor:
rt.auto1 <- rpart(Mileage ~ Weight, data=car.test.frame)

## Print the model:
rt.auto1

## Print the cp table:
printcp(rt.auto1)
```

rpart.pca {mvpart} R Documentation

Principal Components Plot of a Multivariate Rpart Object

Supprimé: e

Description

Plots a PCA of the rpart object on the current graphics device.

Usage

```
rpart.pca(
  tree,
  pts = TRUE,
  plt.allx = TRUE,
  speclabs = TRUE,
  specvecs = TRUE,
  wgt.ave = FALSE,
  add.tree = TRUE,
  cv1 = 1L,
  cv2 = 2L,
  chulls = TRUE,
  interact = FALSE,
  ...
)
```

Arguments

`tree` A fitted object of class `rpart` containing a multivariate regression tree.

`pts` If `TRUE`, large points representing the leaf means are plotted.

`plt.allx` If `TRUE`, small points representing individual cases are plotted.

`speclabs` If `TRUE` the labels of the response variables are plotted.

`specvecs` If `TRUE` the vectors of the response variables are plotted provided `wgt.ave` is `FALSE`.

`wgt.ave` If `TRUE` use weighted averages of responses not vectors.

`add.tree` If `TRUE` add the tree structure to the plot.

`cv1` Defines the principal component to plot horizontally – but see `interact`.

`cv2` Defines the principal component to plot vertically – but see `interact`.

`chulls` If `TRUE` adds convex hulls to thr tree groups.

`interact` If `TRUE` the plot can be viewed in dimensions by left-clicking to top-left, bottom-right or bottom-left (reset).

`...` Arguments to be passed to or from other functions or methods.

Details

This function plots a PCA biplot of the group means (leaves) of multivariate regresssion objects of class `rpart`. The responses and group means and indivdual cases can be shown on the plot. If responses are positive (e.g., species-environment data) weighted averages of responses can be plotted.

The PCA biplot plot is produced on the current graphics device.

Value

`NULL` (well, this is not really true).

See Also

[rpart]

Examples

[Run examples]

```
## Load the spider data set:
data(spider)

## Splitting the table into the responses (species, Y)
## and the descriptors (X):
Y <- data.matrix(spider[,1L:12L])
X <- spider[,13L:18L]
## Note: the multivariate response needs to processed using function
data.matrix
## before being used as a multivariate response by function rpart.

## Multivariate regression tree on the spider species response matrix and all
## the descriptors in the spider data set.

## Using the default settings:
spider1 <- mvpart(Y ~ herbs + reft + moss + sand + twigs + water, data=X)

## Non-interactive PCA plot:
rpart.pca(spider1)

## Interactive PCA plot:
rpart.pca(spider1, wgt.ave=TRUE, interact=TRUE)
```

rpconvert {mvpart} R Documentation

Update an Rpart Object

Description

Objects of class rpart changed (slightly) in their internal format in order to accommodate the changes for user-written split functions. This routine updates an old object to the new format.

Usage

```
rpconvert(x)
```

Arguments

`x` An `rpart-class` object.

Value

An updated rpart object

See Also

`rpart`

scaler {mvpart} R Documentation

Row and Column Scaling of a Data Matrix

Description

The function provides some popular (and effective) standardization methods for community ecologists.

Usage

```
scaler(
  x,
  col = c("mean1", "max1", "min0", "ssq1", "range01", "zsc", "pa",
"rank")[2L],
  row = c("mean1", "max1", "min0", "ssq1", "range01", "zsc", "pa", "rank")[1L]
)
```

Arguments

`x`	Data matrix.
`col`	Character vector of column standardization.
`row`	Character vector of row standardizations.

Details

The function offers the following data matrix standardizations:

- `mean1`: scale to mean of 1.
- `max1`: scale to maximum of 1.
- `ssq1`: scale to sums of squares equal 1.
- `range01`: scale range to 0-1.
- `zsc`: standardize to z-scores (mean=0, sd=1).
- `pa`: scale to presence/absence scale (0/1).
- `rank`: scale to rank order (1=lowest).

Supprimé: au

Standardization is performed first on columns then on rows. "pa" applies to the whole matrix and can be specified using row or col.

Value

Returns the standardized matrix.

Note

Common transformations can be made with standard R functions.

Author(s)

Jari Oksanen – modified Glenn De'ath (Dec 03) snip.rpart {mvpart} R Documentation

Snip Subtrees of an Rpart Object

Description

Creates a "snipped" rpart-class object, containing the nodes that remain after selected sub trees have been snipped off. The user can snip nodes using the toss argument, or interactively by clicking the mouse button on specified nodes within the graphics window.

Usage

```
snip.rpart(x, toss)
```

Arguments

`x` An rpart-class object.

`toss` An integer vector containing indices (node numbers) of all sub trees to be snipped off. If missing, user selects branches to snip off as described below.

Details

A dendrogram of `rpart` is expected to be visible on the graphics device, and a graphics input device (e.g., a mouse) is required. Clicking (the selection button) on a node displays the node number, sample size, response yvalue, and Error (dev). Clicking a second time on the same node snips that sub tree off and visually erases the sub tree. This process may be repeated an number of times. Warnings result from selecting the root or leaf nodes. Clicking the exit button will stop the snipping process and return the resulting rpart-class object.

Warning: visually erasing the plot is done by over-plotting with the background colour. This will do nothing if the background is transparent (often true for screen devices).

See the documentation for the specific graphics device for details on graphical input techniques.

Value

A rpart-class object containing the nodes that remain after specified or selected sub trees have been snipped off.

See Also

plot.rpart

solder {mvpart} R Documentation

Soldering of Components on Printed-Circuit Boards

Description

The `solder` data frame has 720 rows and six columns, representing a balanced subset of a designed experiment varying five factors on the soldering of components on printed-circuit boards.

Usage

```
data(solder)
```

Format

This data frame contains the following columns:

`Opening`

a factor with levels `L M S` indicating the amount of clearance around the mounting pad.

`Solder`

a factor with levels `Thick Thin` giving the thickness of the solder used.

`Mask`

a factor with levels `A1.5 A3 B3 B6` indicating the type and thickness of mask used.

`PadType`

a factor with levels `D4 D6 D7 L4 L6 L7 L8 L9 W4 W9` giving the size and geometry of the mounting pad.

`Panel`

`1:3` indicating the panel on a board being tested.

`skips`

a numeric vector giving the number of visible solder skips.

Source

Chambers and Hastie (1992).

References

John M. Chambers and Trevor J. Hastie eds. (1992) Statistical Models in S, Wadsworth and Brooks/Cole, Pacific Grove, CA 1992.

Examples

Run examples

```
## Load the data set:
data("solder")

## Number of rows and columns in the data table:
dim(solder)

## First six rows of the data table:
head(solder)

## Summary of the data table:
summary(solder)

## A few plots of the data within the data frame:
plot(skips ~ Opening, data=solder, las=1L)
plot(skips ~ Solder, data=solder, las=1L)
plot(skips ~ Mask, data=solder, las=1L)
plot(skips ~ PadType, data=solder, las=1L)
plot(skips ~ as.factor(Panel), data=solder, las=1L, xlab="Panel number")
```

spider {mvpart} R Documentation

Spider Data

Description

Data set on abundances of spiders and environmental predictors. All variables are rated on a 0-9 scale.

Usage

```
data(spider)
```

Format

A data frame with 28 observations with 12 species and six environmental predictors.

Details

Provided with the original package. The first 12 columns appears to represent spider species since their names look like abbreviated binomial names (for instance, the first name, arct.lute, might be Arctosa lutetiana, whereas the second name, pard.lugu, might be Pardosa lugubris). The last six columns appear to be abiotic (two of them: water, sand) and biotic (four of them: moss, reft, twigs, herbs) environmental descriptors.

Source

Van der Aart and Smeeck-Enserink 1975.

References

Van der Aart, P. J. and N. Smeeck-Enserink. 1975. Correlations between distributions of hunting spiders (Lycosidae, Ctenidae) and environmental characteristics in a dune area. Netherlands Journal of Zoology 25: 1-45.

These data were analysed using multivariate trees in De'ath, G. 2002. Multivariate Regression Trees: A New Technique for Modelling Species-Environment Relationships. Ecology 83(4): 1103-1117

Examples

Run examples

```
## Load the data set:
data("spider")
```

```
## Number of rows and columns in the data table:
dim(spider)

## First six rows of the data table:
head(spider)

## Splitting the table into the responses (species, Y)
## and the descriptors (X):
Y <- spider[,1L:12L]
X <- spider[,13L:18L]

## Summary of the species data table:
summary(Y)

## Summary of the descriptors data table:
summary(X)

## Showing the species data using a principal component analysis:
prY <- princomp(Y)

## Data table of the species loading for the first two principal components:
data.frame(
  AX1 = prY$loadings[,1L],
  AX2 = prY$loadings[,2L]
) -> dat

par(mar=c(4.25,4.25,1.25,1.25))
plot(AX2 ~ AX1, data=dat, asp=1, xlim=c(-0.4,0.8), ylim=c(-0.4,0.8))
abline(v=0, lty=3L)
abline(h=0, lty=3L)
text(x=dat$AX1, y=dat$AX2 + 0.03, labels=colnames(Y), cex=0.75)

## Descriptor: Water
par(mfrow=c(3L,2L), mar=c(4.25,4.25,1.25,1.25))
plot(arct.lute ~ water, data = spider, ylim=c(0,9), pch=21L, bg="red")
plot(pard.lugu ~ water, data = spider, ylim=c(0,9), pch=21L, bg="orange")
plot(zora.spin ~ water, data = spider, ylim=c(0,9), pch=21L, bg="yellow")
plot(aulo.albi ~ water, data = spider, ylim=c(0,9), pch=21L, bg="green")
plot(troc.terr ~ water, data = spider, ylim=c(0,9), pch=21L, bg="blue")
plot(alop.cune ~ water, data = spider, ylim=c(0,9), pch=21L, bg="purple")

## Descriptor: Sand
par(mfrow=c(3L,2L), mar=c(4.25,4.25,1.25,1.25))
plot(arct.lute ~ sand, data = spider, ylim=c(0,9), pch=21L, bg="red")
plot(pard.lugu ~ sand, data = spider, ylim=c(0,9), pch=21L, bg="orange")
plot(zora.spin ~ sand, data = spider, ylim=c(0,9), pch=21L, bg="yellow")
plot(aulo.albi ~ sand, data = spider, ylim=c(0,9), pch=21L, bg="green")
plot(troc.terr ~ sand, data = spider, ylim=c(0,9), pch=21L, bg="blue")
plot(alop.cune ~ sand, data = spider, ylim=c(0,9), pch=21L, bg="purple")
```

xdiss {mvpart} R Documentation

Extended Dissimilarity Measures

Description

The function computes extended dissimilarity indices which are for long gradients have better good rank-order relation with gradient separation and are thus efficient in community ordination with multidimensional scaling.

Usage

```
xdiss(
  data,
  dcrit = 1,
  dauto = TRUE,
  dinf = 0.5,
  method = "man",
  use.min = TRUE,
  eps = 1e-04,
  replace.neg = TRUE,
  big = 10000,
  sumry = TRUE,
  full = FALSE,
  sq = FALSE
)
```

Arguments

`data`	Data matrix.
`dcrit`	Dissimilarities < `dcrit` are considered to have no species in common and are recalculated.
`dauto`	Automatically select tuning parameters – recommended.
`dinf`	Internal parameter – leave as is usually.
`method`	Dissimilarity index.
`use.min`	Minimum dissimilarity of pairs of distances used – recommended.
`eps`	Internal parameter – leave as is usually.
`replace.neg`	Internal parameter – leave as is usually.
`big`	Internal parameter – leave as is usually.
`sumry`	Whether to print summary of extended dissimilarities.
`full`	Return the square dissimilarity matrix.
`sq`	Square the dissimilarities – useful for distance-based partitioning.

Details

The function knows the same dissimilarity indices as `gdist`.

Value

Returns an object of class distance with attributes "Size" and "ok". "ok" is TRUE if rows are not disconnected (De'ath 1999).

Author(s)

Glenn De'ath

References

De'ath, G. (1999) Extended dissimilarity: a method of robust estimation of ecological distances from high beta diversity data. Plant Ecology 144(2): 191-199.

Faith, D.P, Minchin, P.R. and Belbin, L. (1987) Compositional dissimilarity as a robust measure of ecological distance. Vegetatio 69: 57-68.

Examples

Run examples

```
 ## Load the spider data set:
data(spider)

## Getting the table of the first six species:
Y <- data.matrix(spider[1L:6L,1L:12L])

## Calculate the dissimilarity measures:
xdiss(Y, method="manhattan")  ## Default
xdiss(Y, method="euclidean")
xdiss(Y, method="canberra")
xdiss(Y, method="bray")
xdiss(Y, method="kulczynski")
xdiss(Y, method="gower")
xdiss(Y, method="maximum")
xdiss(Y, method="binary")
xdiss(Y, method="chisq")
xdiss(Y, method="chord")
```