

# Package ‘BiodiversityR’

February 14, 2012

**Type** Package

**Title** GUI for biodiversity and community ecology analysis

**Version** 1.6

**Date** 2011-09-04

**Author** Roeland Kindt

**Maintainer** Roeland Kindt (World Agroforestry Centre) <R.Kindt@CGIAR.org>

**Description** This package provides a GUI (Graphical User Interface, via the R-Commander) and some utility functions (often based on the vegan package) for statistical analysis of biodiversity and ecological communities, including species accumulation curves, diversity indices, Renyi profiles, GLMs for analysis of species abundance and presence-absence, distance matrices, Mantel tests, and cluster, constrained and unconstrained ordination analysis. A book on biodiversity and community ecology analysis is available for free download from the website.

**License** GPL-2

**URL** <http://www.r-project.org>,  
<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Depends** R (>= 2.13.1), vegan (>= 1.17-12)

**Suggests** MASS, mgcv, cluster, Rcmdr (>= 1.3-14), car, RODBC, rpart, effects, multcomp, ellipse, maptree, sp, splancs, spatial, akima, nnet

**Repository** CRAN

**Date/Publication** 2011-09-04 17:09:32

**R topics documented:**

BiodiversityR-package . . . . .	3
accumresult . . . . .	4
add.spec.scores . . . . .	6
balanced.specaccum . . . . .	7
BCI.env . . . . .	9
BiodiversityRGUI . . . . .	10
CAPdiscrim . . . . .	34
caprescale . . . . .	36
crosstabanalysis . . . . .	38
deviancepercentage . . . . .	39
dist.eval . . . . .	40
dist.zeroes . . . . .	41
distdisplayed . . . . .	42
disttransform . . . . .	44
diversityresult . . . . .	45
faramea . . . . .	47
import.from.Excel . . . . .	48
loaded.citations . . . . .	50
makecommunitydataset . . . . .	50
multiconstrained . . . . .	52
nested.anova.dbrda . . . . .	53
NMSrandom . . . . .	55
nnetrandom . . . . .	56
ordicoeno . . . . .	58
ordisymbol . . . . .	59
PCAsignificance . . . . .	61
radfitresult . . . . .	62
rankabundance . . . . .	63
removeNAcomm . . . . .	65
renyireresult . . . . .	67
residualssurface . . . . .	70
spatialsample . . . . .	71
transfgredient . . . . .	73
transfspecies . . . . .	73
warcom . . . . .	75
warenv . . . . .	80
<b>Index</b>	<b>82</b>

## Description

This package provides a GUI (Graphical User Interface, via the R-Commander; [BiodiversityRGUI](#)) and some utility functions (often based on the [vegan](#) package) for statistical analysis of biodiversity and ecological communities, including species accumulation curves, diversity indices, Renyi profiles, GLMs for analysis of species abundance and presence-absence, distance matrices, Mantel tests, and cluster, constrained and unconstrained ordination analysis. A book on biodiversity and community ecology analysis is available for free download from the website.

## Details

We warmly thank all that provided inputs that lead to improvement of the Tree Diversity Analysis manual that describes common methods for biodiversity and community ecology analysis and its accompanying software. We especially appreciate the comments received during training sessions with draft versions of this manual and the accompanying software in Kenya, Uganda and Mali. We are equally grateful to the thoughtful reviews by Dr Simoneta Negrete-Yankelevich (Instituto de Ecologia, Mexico) and Dr Robert Burn (Reading University, UK) of the draft version of this manual, and to Hillary Kipruto for help in editing of this manual. We also want to specifically thank Mikkel Grum, Jane Poole and Paulo van Breugel for helping in testing the packaged version of the software. We also want to give special thanks for all the support that was given by Jan Beniést, Tony Simons and Kris Vanhoutte in realizing the book and software.

We highly appreciate the support of the Programme for Cooperation with International Institutes (SII), Education and Development Division of the Netherlands Ministry of Foreign Affairs, and VVOB (The Flemish Association for Development Cooperation and Technical Assistance, Flanders, Belgium) for funding the development for this manual. We also thank VVOB for seconding Roeland Kindt to the World Agroforestry Centre (ICRAF). The tree diversity analysis manual was inspired by research, development and extension activities that were initiated by ICRAF on tree and landscape diversification. We want to acknowledge the various donor agencies that have funded these activities, especially VVOB, DFID, USAID and EU.

We are grateful for the developers of the R Software for providing a free and powerful statistical package that allowed development of BiodiversityR. We also want to give special thanks to Jari Oksanen for developing the [vegan](#) package and John Fox for developing the [Rcmdr](#) package, which are key packages that are used by BiodiversityR.

## Author(s)

Roeland Kindt

Maintainer: Roeland Kindt (World Agroforestry Centre [ICRAF]) <[R.Kindt@CGIAR.org](mailto:R.Kindt@CGIAR.org)>

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

We suggest to use this citation for this software as well (together with citations of all other packages that were used)

---

accumresult

*Alternative Species Accumulation Curve Results*

---

## Description

Provides alternative methods of obtaining species accumulation results than provided by functions `specaccum` and `plot.specaccum` (**vegan**).

## Usage

```
accumresult(x,y="", factor="", level, scale="", method="exact", permutations=100,
  conditioned=T, gamma="boot", ...)
accumplot(xr, addit=F, labels="", col=1, ci=2, pch=1, type="p", cex=1, xlim=c(1, xmax),
  ylim=c(1, rich), xlab="sites", ylab="species richness", ...)
accumcomp(x,y="", factor, scale="", method="exact", permutations=100,
  conditioned=T, gamma="boot", plotit=T, labelit=T, legend=T, rainbow=T,
  xlim=c(1, max), ylim=c(0, rich), type="p", xlab="sites",
  ylab="species richness", ...)
```

## Arguments

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
y	Environmental data frame.
factor	Variable of the environmental data frame that defines subsets to calculate species accumulation curves for.
level	Level of the variable to create the subset to calculate species accumulation curves.
scale	Continuous variable of the environmental data frame that defines the variable that scales the horizontal axis of the species accumulation curves.
method	Method of calculating the species accumulation curve (as in function <code>specaccum</code> ). Method "collector" adds sites in the order they happen to be in the data, "random" adds sites in random order, "exact" finds the expected (mean) species richness, "coleman" finds the expected richness following Coleman et al. 1982, and "rarefaction" finds the mean when accumulating individuals instead of sites.
permutations	Number of permutations to calculate the species accumulation curve (as in function <code>specaccum</code> ).
conditioned	Estimation of standard deviation is conditional on the empirical dataset for the exact SAC (as in function <code>specaccum</code> ).
gamma	Method for estimating the total extrapolated number of species in the survey area (as in <code>specaccum</code> ).

<code>addit</code>	Add species accumulation curve to an existing graph.
<code>xr</code>	Result from <code>specaccum</code> or <code>accumresult</code> .
<code>col</code>	Colour for drawing lines of the species accumulation curve (as in function <code>plot.specaccum</code> ).
<code>labels</code>	Labels to plot at left and right of the species accumulation curves.
<code>ci</code>	Multiplier used to get confidence intervals from standard deviation (as in function <code>plot.specaccum</code> ).
<code>pch</code>	Symbol used for drawing the species accumulation curve (as in function <code>points</code> ).
<code>type</code>	Type of plot (as in function <code>plot</code> ).
<code>cex</code>	Character expansion factor (as in function <code>plot</code> ).
<code>xlim</code>	Limits for the horizontal axis.
<code>ylim</code>	Limits for the vertical axis.
<code>xlab</code>	Label for the horizontal axis.
<code>ylab</code>	Label for the vertical axis.
<code>plotit</code>	Plot the results.
<code>labelit</code>	Label the species accumulation curves with the levels of the categorical variable.
<code>legend</code>	Add the legend (you need to click in the graph where the legend needs to be plotted).
<code>rainbow</code>	Use rainbow colouring for the different curves.
<code>...</code>	Other items passed to function <code>specaccum</code> or <code>plot.specaccum</code> .

### Details

These functions provide some alternative methods of obtaining species accumulation results, although function `specaccum` is called by these functions to calculate the actual species accumulation curve.

Functions `accumresult` and `accumcomp` allow to calculate species accumulation curves for subsets of the community and environmental data sets. Function `accumresult` calculates the species accumulation curve for the specified level of a selected environmental variable. Method `accumcomp` calculates the species accumulation curve for all levels of a selected environmental variable separately. Both methods allow to scale the horizontal axis by multiples of the average of a selected continuous variable from the environmental dataset (hint: add the abundance of each site to the environmental data frame to scale accumulation results by mean abundance).

Functions `accumcomp` and `accumplot` provide alternative methods of plotting species accumulation curve results, although function `plot.specaccum` is called by these functions. When you choose to add a legend, make sure that you click in the graph on the spot where you want to put the legend.

### Value

The functions provide alternative methods of obtaining species accumulation curve results, although results are similar as obtained by functions `specaccum` and `plot.specaccum`.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune.env)
data(dune)
dune.env$site.totals <- apply(dune,1,sum)
Accum.1 <- accumresult(dune, y=dune.env, scale='site.totals', method='exact', conditioned=TRUE)
Accum.1
accumplot(Accum.1)
accumcomp(dune, y=dune.env, factor='Management', method='exact', legend=FALSE, conditioned=TRUE)
## CLICK IN THE GRAPH TO INDICATE WHERE THE LEGEND NEEDS TO BE PLACED FOR
## OPTION WHERE LEGEND=TRUE (DEFAULT).
```

---

add.spec.scores

*Add Species Scores to Unconstrained Ordination Results*

---

## Description

Calculates scores (coordinates) to plot species for PCoA or NMS results that do not naturally provide species scores. The function can also rescale PCA results to use the choice of rescaling used in **vegan** for the `rda` function (after calculating PCA results via PCoA with the euclidean distance first).

## Usage

```
add.spec.scores(ordi, comm, method="cor.scores", multi=1, Rscale=F, scaling="1")
```

## Arguments

<code>ordi</code>	Ordination result as calculated by <code>cmdscale</code> , <code>isoMDS</code> , <code>sammon</code> , <code>postMDS</code> , <code>metaMDS</code> or <code>NMStandom</code> .
<code>comm</code>	Community data frame with sites as rows, species as columns and species abundance as cell values.
<code>method</code>	Method for calculating species scores. Method "cor.scores" calculates the scores by the correlation between site scores and species vectors (via function <code>cor</code> ), method "wa.scores" calculates the weighted average scores (via function <code>wascores</code> ) and method "pcoa.scores" calculates the scores by weighing the correlation between site scores and species vectors by variance explained by the ordination axes.
<code>multi</code>	Multiplier for the species scores.
<code>Rscale</code>	Use the same scaling method used by <b>vegan</b> for <code>rda</code> .
<code>scaling</code>	Scaling method as used by <code>rda</code> .

## Value

The function returns a new ordination result with new information on species scores. For PCoA results, the function calculates eigenvalues (not sums-of-squares as provided in results from function `cmdscale`), the percentage of explained variance per axis and the sum of all eigenvalues. PCA results (obtained by PCoA obtained by function `cmdscale` with the Euclidean distance) can be scaled as in function `rda`, or be left at the original scale.

## Author(s)

Roeland Kindt

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune)
distmatrix <- vegdist(dune, method='euc')
## Principal coordinates analysis with 19 axes to estimate total variance
Ordination.model1 <- cmdscale(distmatrix, k=19, eig=TRUE, add=FALSE)
Ordination.model1 <- add.spec.scores(Ordination.model1, dune, method='pcoa.scores', Rscale=TRUE, scaling=1, multi=1)
Ordination.model1
## Compare Ordination.model1 with:
Ordination.model2 <- rda(dune)
summary(Ordination.model2, scaling=1)
```

---

balanced.specaccum      *Balanced Species Accumulation Curves*

---

## Description

Provides species accumulation results calculated from balanced (equal subsample sizes) subsampling from each stratum. Sites can be accumulated in a randomized way, or alternatively sites belonging to the same stratum can be kept together. Results are in the same format as `specaccum` and can be plotted with `plot.specaccum` (**vegan**).

## Usage

```
balanced.specaccum(comm, permutations=100, strata=strata, grouped=TRUE,
  reps=0, scale=NULL)
```

**Arguments**

<code>comm</code>	Community data frame with sites as rows, species as columns and species abundance as cell values.
<code>permutations</code>	Number of permutations to calculate the species accumulation curve.
<code>strata</code>	Categorical variable used to specify strata.
<code>grouped</code>	Should sites from the same stratum be kept together (TRUE) or not.
<code>reps</code>	Number of subsamples to be taken from each stratum (see details).
<code>scale</code>	Quantitative variable used to scale the sampling effort (see details).

**Details**

This function provides an alternative method of obtaining species accumulation results as provided by `specaccum` and `accumresult`.

Balanced sampling is achieved by randomly selecting the same number of sites from each stratum. The number of sites selected from each stratum is determined by `reps`. Sites are selected from strata with sample sizes larger or equal than `reps`. In case that `reps` is smaller than 1 (default: 0), then the number of sites selected from each stratum is equal to the smallest sample size of all strata. Sites from the same stratum can be kept together (`grouped=TRUE`) or the order of sites can be randomized (`grouped=FALSE`).

The results can be scaled by the average accumulation of a quantitative variable (default is number of sites), as in `accumresult` (hint: add the abundance of each site to the environmental data frame to scale accumulation results by mean abundance). When sites are not selected from all strata, then the average is calculated only for the strata that provided sites.

**Value**

The functions provide alternative methods of obtaining species accumulation curve results, although results are similar as obtained by functions `specaccum` and `accumresult`.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
library(vegan)
data(dune.env)
data(dune)
Accum.1 <- balanced.specaccum(dune, strata=dune.env$Management, reps=3)
Accum.1
dune.env$site.totals <- apply(dune,1,sum)
```

```
Accum.2 <- balanced.specaccum(dune, strata=dune.env$Management, reps=3, scale=dune.env$site.totals)
Accum.2
```

---

BCI.env

*Barro Colorado Island Quadrat Descriptions*

---

## Description

Environmental characteristics and UTM coordinates of a 50 ha sample plot (consisting of 50 1-ha quadrats) from Barro Colorado Island of Panama. Dataset [BCI](#) provides the tree species composition (trees with diameter at breast height equal or larger than 10 cm) of the same plots.

## Usage

```
data(BCI.env)
```

## Format

A data frame with 50 observations on the following 6 variables.

UTM.EW a numeric vector

UTM.NS a numeric vector

Precipitation a numeric vector

Elevation a numeric vector

Age.cat a factor with levels c1 c2 c3

Geology a factor with levels pT Tb Tbo Tc Tcm Tct Tgo Tl Tlc

## Source

<http://www.sciencemag.org/cgi/content/full/295/5555/666/DC1>

## References

Pyke CR, Condit R, Aguilar S and Lao S. (2001). Floristic composition across a climatic gradient in a neotropical lowland forest. *Journal of Vegetation Science* 12: 553-566.

Condit, R, Pitman, N, Leigh, E.G., Chave, J., Terborgh, J., Foster, R.B., Nunez, P., Aguilar, S., Valencia, R., Villa, G., Muller-Landau, H.C., Losos, E. & Hubbell, S.P. (2002). Beta-diversity in tropical forest trees. *Science* 295: 666-669.

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

[http://www.worldagroforestry.org/treesandmarkets/tree\\_diversity\\_analysis.asp](http://www.worldagroforestry.org/treesandmarkets/tree_diversity_analysis.asp)

## Examples

```
data(BCI.env)
```

## Description

This function provides a GUI (Graphical User Interface) for some of the functions of **vegan**, some other packages and some new functions to run biodiversity analysis, including species accumulation curves, diversity indices, Renyi profiles, rank-abundance curves, GLMs for analysis of species abundance and presence-absence, distance matrices, Mantel tests, cluster and ordination analysis (including constrained ordination methods such as RDA, CCA, db-RDA and CAP). The function depends and builds on **Rcmdr**, performing all analyses on the community and environmental datasets that the user selects. A thorough description of the package and the biodiversity and ecological methods that it accomodates (including examples) is provided in the freely available Tree Diversity Analysis manual (Kindt and Coe, 2005).

## Usage

```
BiodiversityRGUI()
```

## Details

The function launches the R-Commander GUI with an extra menu list for common statistical methods for biodiversity and community ecology analysis.

The R-Commander is launched by changing the location of the Rcmdr "etc" folder to the "etc" folder of BiodiversityR. As the files of the "etc" folder of BiodiversityR are copied from Rcmdr 1.3-14, it is possible that newer versions of the R-Commander will not be launched properly. In such situations, it is possible that copying all files from the Rcmdr "etc" folder again and adding the BiodiversityR menu options to the Rcmdr-menus.txt is all that is needed to launch the R-Commander again.

BiodiversityR uses two data sets for analysis: the community dataset (or community matrix or species matrix) and the environmental dataset (or environmental matrix). The environmental dataset is the same dataset that is used as the "active dataset" of The R-Commander. (Note that you could sometimes use the same dataset as both the community and environmental dataset. For example, you could use the community dataset as environmental dataset as well to add information about specific species to ordination diagrams. As another example, you could use the environmental dataset as community dataset if you first calculated species richness of each site, saved this information in the environmental dataset, and then use species richness as response variable in a regression analysis.) Some options of analysis of ecological distance allow the community matrix to be a distance matrix (the community data set will be interpreted as distance matrix via `as.dist` prior to further analysis).

BiodiversityR provides the following menu options (each described below in greater detail):

- **Select community dataset** (Community matrix menu) Selects a dataset to be the community dataset.
- **Import datasets from Excel** (Community matrix menu) Imports a community and environmental dataset from an Excel workbook.

- **Import datasets from Access** (Community matrix menu) Imports a community and environmental dataset from an Access database.
- **View community data set** (Community matrix menu) Invoke the R text editor to view the data of the community data set.
- **Edit community data set** (Community matrix menu) Invoke the R text editor to edit the data of the community data set.
- **Check data sets** (Community matrix menu) Check whether the community and environmental data sets have compatible dimensions.
- **Same sites for community and environmental** (Community matrix menu) Creates a new community dataset with the same sites sequence as the environmental matrix.
- **Make community dataset** (Community matrix menu) Creates a community dataset from the environmental dataset.
- **Remove NA** (Community matrix menu) Removes the same sites with NA from the environmental and community datasets.
- **Transform community matrix** (Community matrix menu) Transforms the community matrix.
- **Select environmental data set** (Environmental matrix menu) Selects a dataset to be the environmental dataset.
- **View environmental data set** (Environmental matrix menu) Invoke the R text editor to view the data of the environmental dataset.
- **Edit environmental data set** (Environmental matrix menu) Invoke the R text editor to edit the data of the environmental dataset.
- **Summary** (Environmental matrix menu) Explores variables of the environmental dataset.
- **Box Cox transformation** (Environmental matrix menu) Creates a transformed variable from one of the variables of the environmental dataset.
- **Species accumulation curves** (Analysis of diversity menu) Estimates and plots species accumulation curves.
- **Diversity indices** (Analysis of diversity menu) Calculates and plots diversity indices.
- **Rank abundance** (Analysis of diversity menu) Calculates and plots rank-abundance curves.
- **Renyi profile** (Analysis of diversity menu) Calculates and plots Renyi diversity profiles.
- **Species abundance as response** (Analysis of species as response menu) Fits and plots regression models assuming that the response variable is count data.
- **Species presence-absence as response** (Analysis of species as response menu) Fits and plots regression models transforming and analysing the response variable as presence-absence.
- **Calculate distance matrix** (Analysis of ecological distance menu) Calculates a distance matrix.
- **Unconstrained ordination** (Analysis of ecological distance menu) Fits and plots unconstrained ordination models.
- **Constrained ordination** (Analysis of ecological distance menu) Fits and plots constrained ordination models.
- **Clustering** (Analysis of ecological distance menu) Calculates and plots results from clustering algorithms.

- **Compare distance matrices** (Analysis of ecological distance menu) Conducts some analysis such as Mantel, MRPP and ANOSIM tests on distance matrices.
- **Help about BiodiversityR** (Help menu) Opens the help file available for the BiodiversityR package (including this html file).
- **Citations for loaded packages** (Help menu) Provides a list of all the loaded packages and gives citation information.
- **Go to website for BiodiversityR** (Help menu) Links to the website for the BiodiversityR package and Tree Diversity Analysis manual.
- **Tree diversity analysis manual** (Help menu) Links to the PDF version of the Tree Diversity Analysis manual. Separate chapters can be downloaded from the website of BiodiversityR (see directly above).

### Value

None

### Select Community Dataset

This window selects the community dataset to be used in the biodiversity analyses and provides the following options:

- **Data Sets (pick one)** A drop-down list is provided with all the datasets that are available. The current community data set is indicated, or the first data set of the list is shown. New datasets can be loaded through the Data menu of the Rcmdr or through the "import from Excel" option of BiodiversityR.
- **OK** Make the selected data set the community data set.
- **Cancel** Close the window and do not select a new data set.

### Import datasets from Excel

This window enables to import community and environmental data sets from an Excel workbook with a specific format (sheets that are named "community" and "environmental" or "stacked" and "environmental"; first row containing the names of the variables) (an example is provided in the etc folder of the BiodiversityR package for the dune meadow dataset: dune.xls). The menu provides the following options:

- **Enter name for community data set** The name for the new community dataset.
- **Enter name for environmental data set** The name for the new environmental dataset.
- **Enter name for variable for sites** The name for the variable that indicates site names in the new dataset. The same variable should be available from the various sheets. Passed as argument for "sitenames" for function [import.from.Excel](#)
- **Import community dataset from stacked format** Import the community data set from the stacked format or not. The stacked format is the only possibility for community data with more than 255 species (related to the maximum number of columns in Excel).  
Option "Yes" will result in argument of "stacked" for "sheet" for function [import.from.Excel](#) to import the community data set.

Option "No" will result in argument of "community" for "sheet" for function `import.from.Excel` to import the community data set.

Either option will result in argument of "environmental" for "sheet" for function `import.from.Excel` to import the environmental data set.

- **Enter variable for species** This option is only available for stacked data. The list shows the variables that can be used for the names of species (shown as names for the columns). Passed as argument for "column" of function `import.from.Excel`.
- **Enter variable for abundance** This option is only available for stacked data. The list shows the variables that can be used for the abundance values (shown as totals for cells). Passed as argument for "value" of function `import.from.Excel`.
- **Enter factor for subset** This option is only available for stacked data. The list shows the variables that can be used for the abundance values (shown as totals for cells). Passed as argument for "factor" of function `import.from.Excel`.
- **Enter level for subset** Chooses the value for the subset variable to create the subset. Passed as argument for "level" of function `import.from.Excel`.
- **OK** Import the community and environmental datasets and make these the active datasets.
- **Cancel** Close the window and do not import new datasets.

### Import datasets from Access

This window enables to import community and environmental data sets from an Access database with a specific format (sheets that are named "community" and "environmental" or "stacked" and "environmental"; first row containing the names of the variables) (an example is provided in the etc folder of the BiodiversityR package for the dune meadow dataset: dune.xls). The menu provides the following options:

- **Enter name for community data set** The name for the new community dataset.
- **Enter name for environmental data set** The name for the new environmental dataset.
- **Enter name for variable for sites** The name for the variable that indicates site names in the new dataset. The same variable should be available from the various sheets. Passed as argument for "sitenames" for function `import.from.Access`
- **Import community dataset from stacked format** Import the community data set from the stacked format or not.
  - Option "Yes" will result in argument of "stacked" for "sheet" for function `import.from.Access` to import the community data set.
  - Option "No" will result in argument of "community" for "sheet" for function `import.from.Access` to import the community data set.
  - Either option will result in argument of "environmental" for "sheet" for function `import.from.Access` to import the environmental data set.
- **Enter variable for species** This option is only available for stacked data. The list shows the variables that can be used for the names of species (shown as names for the columns). Passed as argument for "column" of function `import.from.Access`.
- **Enter variable for abundance** This option is only available for stacked data. The list shows the variables that can be used for the abundance values (shown as totals for cells). Passed as argument for "value" of function `import.from.Access`.

- **Enter factor for subset** This option is only available for stacked data. The list shows the variables that can be used for the abundance values (shown as totals for cells). Passed as argument for "factor" of function `import.from.Access`.
- **Enter level for subset** Chooses the value for the subset variable to create the subset. Passed as argument for "level" of function `import.from.Access`.
- **OK** Import the community and environmental datasets and make these the active datasets.
- **Cancel** Close the window and do not import new datasets.

### Same sites for community and environmental datasets

This window maps the community dataset onto the rownames of the environmental dataset by function `same.sites`. Having the same sequence of sites is an assumption for analysis with BiodiversityR. It may be useful to use this function after making a community dataset from a stacked environmental dataset (especially as sites are ordered in an alphabetic way from the stacked dataset, which may create problems with X1, X10, X100 site names versus the X001, X010 and X100 formats; the function is also useful where some sites do not contain any species). The menu provides the following options:

- **save original community matrix** If this option is selected, the original data set is saved under the name of the community dataset followed by ".orig".
- **OK** Order the sites of the community dataset in exactly the same way as the sites of the environmental data set, leaving out sites that do not have matching names in the environmental data set.
- **Cancel** Close the window and do not re-order and select the sites.

### Make Community Dataset

This window selects the variables that indicates sites, species and abundance to create a new community dataset. This dataset becomes the active community dataset. The menu provides the following options:

- **Save result as** The name for the new community dataset.
- **Site variable (rows)** The list shows the variables that can be used for the names of sites (shown as names for the rows). Passed as argument for "row" of function `makecommunitydataset`.
- **Species variable (columns)** The list shows the variables that can be used for the names of species (shown as names for the columns). Passed as argument for "column" of function `makecommunitydataset`.
- **Abundance variable** The list shows the variables that can be used for the abundance values (shown as totals for cells). Passed as argument for "value" of function `makecommunitydataset`.
- **Subset options** The list shows the variables that can be used for the abundance values (shown as totals for cells). Passed as argument for "factor" of function `makecommunitydataset`.
- **Subset** Chooses the value for the subset variable to create the subset. Passed as argument for "level" of function `makecommunitydataset`.
- **OK** Create the community data set and make it the active community dataset.
- **Cancel** Close the window and do not create a new community dataset.

### Remove NA

This window removes the sites that have NA (missing values) for a selected variable of the environmental dataset. When environmental variables have missing values, this often creates problems with biodiversity analysis. The menu provides the following options:

- **Select variable** The list shows the variables that can be used to remove sites with NA. Passed as argument for var for functions `removeNAcomm` and `removeNAenv`.
- **OK** Remove the sites with NA.
- **Cancel** Close the window and do not remove the sites with NA.

### Transform community matrix

This window transforms the community matrix. The menu provides the following options:

- **Method** Method of transforming the community dataset. Passed as argument for "method" for function `disttransform`. The transformed community matrix is saved under the same name of the original dataset, and the current community dataset therefore becomes the transformed community dataset.
- **Save original community matrix** This option saves the untransformed community dataset by adding `.orig` to the name of the community dataset, as the function replaces the original dataset with the transformed community dataset.
- **OK** Calculate the new community matrix.
- **Cancel** Close the window and do not calculate a new community matrix.

### Select Environmental Dataset

This window selects the environmental dataset to be used in the biodiversity analyses. The environmental dataset is always the active dataset for non-Biodiversity Rcmdr options. By selecting the community dataset as the environmental dataset as well, you can also manipulate the community dataset with the other Rcmdr options. The menu provides the following options:

- **Data Sets (pick one)** A drop-down list is provided with all the datasets that are available. The current community data set is indicated, or the first data set of the list is shown. New datasets can be loaded through the Data menu of the Rcmdr or through the "import from Excel" option of BiodiversityR.
- **OK** Make the selected data set the environmental data set.
- **Cancel** Close the window and do not select a new data set.

### Summary

This window makes a summary of all or a selection of the variables of the environmental dataset, or plots the variables. In case that you want to make a summary of the community dataset, then you need to make the community dataset the environmental dataset at the same time. The menu provides the following options:

- **Select variable** A drop-down list is provided with all the variables of the environmental dataset. The first item of the list (all) is reserved to make a summary of all variables. datasets that are available.

- **OK** Make a summary of all variables or the selected variable by function `summary`.
- **Plot** Plots all variables against each other with function `pairs`, plots a selected continuous variable with function `plot` or plots a categorical with function `boxplot`.
- **Cancel** Close the window and do provide any summary or plot.

### Box Cox transformation

This window makes a Box-Cox transformation of a selected variable from the environmental dataset. The menu provides the following options:

- **Select variable** A drop-down list is provided with all the variables of the environmental dataset. Click on the variable to transform.
- **OK** Calculates a Box-Cox transformation of the selected variable with function `box.cox.powers`. Makes a QQ-plot (function `qq.plot`), and performs a Shapiro test (function `shapiro.test`) and Kolmogorov-Smirnov test (function `ks.test`) of the original and transformed variable.
- **Cancel** Close the window.

### Species accumulation curves

This window fits and plots species accumulation curves. The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the estimated species accumulation curve after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you plot the result by typing in the name of previous result first in this box.
- **Accumulation method** Select the method of species accumulation. Passed as argument for "method" of functions `accumresult` or `accumcomp`.
- **permutations** Number of permutations for random species accumulation. Passed as argument for "permutation" of functions `accumresult` or `accumcomp`.
- **scale of x axis** Method of scaling the horizontal axis. Passed as argument for "scale" of functions `accumresult` or `accumcomp`.
- **subset options** The list shows the variables that can be used for selecting subsets. Option "all" indicates that no subset will be calculated. In case a variable is selected, it will be passed as argument for "factor" of functions `accumresult` or `accumcomp`.
- **Subset** Subset chooses which subsets are calculated. In case that the value of "." (a period) is selected then function `accumcomp` will be used to calculate the species accumulation curve and to plot the curve (you may need to click in the graph to show where the legend needs to be placed). In case another value is chosen, then this will be the argument for "level" of function `accumresult`.
- **Plot options** Options for plotting passed to function `accumplot`.
  - Option "addplot" sets "addit=T" meaning that the species accumulation curve will be added to an existing graph.
  - Option "x limits" sets "xlim". Providing "1,10" will plot between 1 and 10.
  - Option "y limits" sets "ylim". Providing "2,20" will plot between 2 and 20.
  - Option "ci" sets "ci".

Option "symbol" sets "pch".

Option "cex" sets "cex".

Option "colour" sets "col".

- **OK** Calculate the species accumulation curve with functions [accumresult](#) or [accumcomp](#).
- **Plot** Plot the species accumulation curve with the name listed on top with function [accumplot](#). You may need to click in the graph to indicate where the legend needs to be placed.
- **Cancel** Close the window and do not calculate a new species accumulation curve.

### Diversity indices

The window calculates and fits diversity indices from the community dataset. The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the estimated diversity indices after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you plot the result by typing in the name of previous result first in this box. To obtain a meaningful graph, you need to provide similar selections as for the original result (and it may thus be easier to recalculate first and then plot immediately).
- **Diversity index** Select the diversity index. Passed as argument for "index" of functions [diversityresult](#) or [diversitycomp](#).
- **Calculation method** Select the method of calculation. Passed as argument for "method" of functions [diversityresult](#) or [diversitycomp](#).
- **subset options** The list shows the variables that can be used for selecting subsets. Option "all" indicates that no subset will be calculated. In case a variable is selected, it will be passed as argument for "factor" of functions [diversityresult](#) or [diversitycomp](#).
- **Subset** Subset chooses which subsets are calculated. In case that the value of "." (a period) is selected then function [diversitycomp](#) will be used to calculate the species accumulation curve and to plot the curve (you may need to click in the graph to show where the legend needs to be placed). In case another value is chosen, then this will be the argument for "level" of function [diversityresult](#).
- **Output options** Options for obtaining results with functions [diversityresult](#), [diversitycomp](#) or for plotting results.
  - Option "save results" results in adding a new variable with the diversity indices to the environmental dataset. This method only works for calculation method "separate per site" and function [diversityresult](#).
  - Option "sort results" results in setting option "sortit=T" for functions [diversityresult](#) or [diversitycomp](#).
  - Option "label results" results in labeling points in the resulting graph.
  - Option "add plot" results in adding points to an existing graph.
  - Option "y limits" results in setting limits for the y axis. Providing "0,10" results in limits of 0 and 10 for the vertical axis.
  - Option "symbol" sets "pch" to choose symbols as in function [points](#).
- **OK** Calculate the diversity indices with [diversityresult](#) or [diversitycomp](#).

- **Plot** Plot the diversity results with the name listed on top (should have been calculated first). This will only provide meaningful results if similar options are provided as when calculating the results.
- **Cancel** Close the window and do not calculate new diversity indices.

### Rank Abundance

The window fits and plots rank abundance curves for the community dataset. The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the estimated rank abundance curve after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you plot the result by typing in the name of previous result first in this box.
- **subset options** The list shows the variables that can be used for selecting subsets. Option "all" indicates that no subset will be calculated. In case a variable is selected, it will be passed as argument for "factor" of functions [rankabundance](#) or [rankabuncomp](#).
- **Subset** Subset chooses which subsets are calculated. In case that the value of "." (a period) is selected then function [rankabuncomp](#) will be used to calculate and plot the rank abundance curves (you may need to click in the graph to show where the legend needs to be placed). In case another value is chosen, then this will be the argument for "level" of function [rankabundance](#).
- **Plot options** The list provides options for scaling the vertical axis. The selection is passed as argument for "scale" of function [rankabunplot](#).  
Option "fit RAD" fits distribution models to the observed rank-abundance distribution with function [radfitresult](#) and plots the results.  
Option "add plot" sets `addit=T` for function [rankabunplot](#) meaning that the rank abundance curve will be added to an existing graph.  
Option "x limits" sets `xlim` for function [rankabunplot](#). Providing "1,10" will plot between 1 and 10.  
Option "y limits" sets `ylim` for function [rankabunplot](#). Providing "2,20" will plot between 2 and 20.
- **OK** Calculate the rank abundance curve with functions [rankabundance](#) or [rankabuncomp](#).
- **Plot** Plot the rank abundance curve with the name listed on top (should have been calculated first) with function [rankabunplot](#), or fit models to rank abundance distribution.
- **Cancel** Close the window and do not calculate a new rank abundance curve.

### Renyi diversity profiles

The window fits and plots Renyi diversity profiles from the community dataset. The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the diversity profiles after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you plot the result by typing in the name of previous result first in this box.

- **Calculation method** The list allows to select the method of calculating the diversity profile. Options "all" and "separate per site" are passed as argument for "method" of function [renyiresult](#). Option "accumulation" results in using function [renyiaccumresult](#). These options are not valid when [renyicomp](#) is invoked (see Subset options).
- **Scale parameters** The "scale parameters" are passed as argument for "scale" for functions [renyiresult](#), [renyiaccumresult](#) or [renyicomp](#).
- **Permutations** The "permutations" are passed as argument for "permutations" for functions [renyiaccumresult](#) or [renyicomp](#).
- **subset options** The list shows the variables that can be used for selecting subsets. Option "all" indicates that no subset will be calculated. In case a variable is selected, it will be passed as argument for "factor" of functions [renyiresult](#) or [renyicomp](#).
- **Subset** Subset chooses which subsets are calculated. In case that the value of "." (a period) is selected then function [renyicomp](#) will be used to calculate the diversity profile and to plot the curve (you may need to click in the graph to show where the legend needs to be placed). In case another value is chosen, then this will be the argument for "level" of function [renyiresult](#).
- **Plot options** Options for plotting passed to function [renyiplot](#).  
Option "evenness profile" sets "evenness=T".  
Option "evenness profile" sets addit=T meaning that the diversity profiles will be added to an existing graph.  
Option "y limits" sets ylim. Providing "2,20" will plot between 2 and 20.  
Option "symbol" sets pch.  
Option "cex" sets cex.  
Option "colour" sets col.
- **OK** Calculate the diversity profile with functions [renyiresult](#), [renyiaccumresult](#) or [renyicomp](#).
- **Plot** Plot the species accumulation curve with the name listed on top with functions [renyiplot](#) or [persp.renyiaccum](#). The calculation method will determine which plot function is used.
- **Cancel** Close the window and do not calculate a new diversity profile.

### Species abundance as response

The window fits and plots regression models for abundance data with a response variable selected from the community dataset and explanatory variables selected from the environmental dataset. (Hint: to analysis species richness patterns, save site-specific species richness (from diversity indices menu) into the environmental data set, and then make the environmental data set to be the community dataset as well). The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the fitted regression model after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you can plot the result by typing in the name of previous result first in this box.
- **Model options** Select the method of regression analysis.  
Option "linear model" fits a simple linear regression model with function [lm](#).  
Option "Poisson model" fits GLMs with Poisson variance functions and log link functions through function [glm](#).

Option "quasi-Poisson model" fits GLMs with quasi-Poisson variance functions and log link functions through function `glm`.

Option "negative binomial model" fits GLMs with negative binomial variance functions and log link functions through function `glm.nb`.

Option "gam model" fits GAMs with Poisson variance functions and log link functions through function `gam`.

Option "gam negbinom model" fits GAMs with negative binomial variance functions and log link functions through function `gam`.

Option "glmmPQL" fits GLMMs with negative binomial variance functions and log link functions through function `glmmPQL`.

Option "rpart" fits a regression tree through function `rpart`.

- **Standardize** Fit the regression to a standardised dataset with function `scale` (only continuous variables are standardised, not categorical variables).
- **Print summary** Provide a summary of the regression with functions `summary.lm`, `summary.glm` or `summary.gam`.
- **Print anova** Provide a summary of the regression with functions `anova.lm`, `anova.glm`, `anova.gam`, `drop1` or `Anova` (latter two type-II ANOVAs only invoked for multiple regression).
- **add predictions to data frame** Adds the predicted values to the environmental dataset using the model name combined with ".fit" (using the appropriate `predict` function).
- **Response variable** Type the name of the response variable, or select and double-click from the list that is provided. This variable will be displayed on the left-hand side of the formula (variable ~) and is also the response variable that is plotted in the various result plots. The variable is selected as one of the variables (species) of the community dataset, and is first added to the environmental dataset. When you select the environmental dataset to be the community dataset as well, then you can select variables of the environmental dataset as response variable.
- **Explanatory** Type the right-hand side of the model formula (~ explanatory), or select and double-click for variables and select and click for operators to construct the right-hand side of the model formula.
- **Remove site with name** The name of the site to be removed from the environmental dataset.
- **Plot options** The options provide various functions that can be used to plot regression results of the current model (shown on top of the window; should have been estimated first).
  - Option "diagnostic plots" chooses functions `plot.lm` or `gam.check` to plot diagnostic plots. For regression trees, the residuals are plotted against the residuals via `predict.rpart` and `residuals.rpart`.
  - Option "levene test" chooses function `levene.test` and plots residuals of the selected categorical variable (shown on the right).
  - Option "term plot" chooses functions `termplot` or `plot.gam` to plot a termplot of the selected categorical variable (shown on the right).
  - Option "effect plot" chooses function `effect` to plot an effect plot of the selected variable (shown on the right). (The menu option of the R-Commander of models > Graphs plots all the variables).
  - Option "qq plot" chooses function `qq.plot` to plot the residuals from the model.
  - Option "result plot (new)" chooses an appropriate `predict` function to plot a new plot of the model predictions for the selected variable (shown on the right).

Option "result plot (add)" chooses an appropriate `predict` function to add a new plot of the model predictions for the selected variable (shown on the right)

Option "result plot (interpolate)" chooses an appropriate `predict` function to add a new plot of the model predictions for the selected variable (shown on the right). This model is predicted from a new dataset that only contains 1000 interpolated values for the selected explanatory variable.

Option "cr plot" chooses function `cr.plots` to plot a component + residual plots of the selected variable (shown on the right). (The menu option of the R-Commander of models > Graphs plots all the variables).

Option "av plot" chooses function `av.plots` to plot added variable plots of the selected variable (shown on the right). (The menu option of the R-Commander of models > Graphs plots all the variables and has an option of identifying sites with the mouse.)

Option "influence plot" chooses function `influence.plot` to plot influence plots. (The menu option of the R-Commander of models > Graphs includes the option of identifying sites with the mouse.)

Option "multcomp" chooses function `glht` to plot simultaneous confidence intervals of the selected categorical variable (shown on the right).

Option "rpart" chooses functions `plot.rpart` and `text.rpart` to plot a dendrogram for the regression tree result.

- **Plot variable** Variable of the environmental dataset that is used for some plotting functions.
- **OK** Fit the selected models.
- **Plot** Plot results for the model with name that appears on top. The model options need to apply to the model (e.g. if a GLM method was used to fit the model, this option should also be selected when plotting the results).
- **Cancel** Close the window and do not estimate new regression models.

### Species presence-absence as response

The window fits and plots regression models for presence-absence data with a response variable selected from the community dataset and explanatory variables selected from the environmental dataset. The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the fitted regression model after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you can plot the result by typing in the name of previous result first in this box.
- **Model options** Select the method of regression analysis.
  - Option "crosstab" calculates a cross-tabulation of the selected response (rescaled as presence-absence) and one selected environmental variable, and estimates a Chi-square test of the contingency table with function `chisq.test`.
  - Option "binomial model" fits GLMs with binomial variance functions and logit link functions through function `glm`.
  - Option "quasi-binomial model" fits GLMs with quasi-binomial variance functions and log link functions through function `glm`.
  - Option "gam model" fits GAMs with binomial variance functions and logit link functions through function `gam`.

Option "gam quasi-binomial model" fits GAMs with quasi-binomial variance functions and logit link functions through function `gam`.

Option "rpart" fits a regression tree through function `rpart`.

Option "nnet" fits a forward-feeding artificial neural network through function `nnetrandom`.

- **Standardize** Fit the regression to a standardised dataset with function `scale` (only continuous variables are standardised, not categorical variables).
- **Print summary** Provide a summary of the regression with functions `summary.glm` or `summary.gam`, or use `summary.rpart` or `summary.nnet`
- **Print anova** Provide a summary of the regression with functions `anova.glm`, `anova.gam`, `drop1` or `Anova` (latter two type-II ANOVAs only invoked for multiple regression).
- **add predictions to data frame** Adds the predicted values to the environmental dataset using the model name combined with ".fit" (using the appropriate `predict` function).
- **Response variable** Type the name of the response variable, or select and double-click from the list that is provided. This variable will be displayed on the left-hand side of the formula (variable >0 ~) and is also the response variable that is plotted in the various result plots. The variable is selected as one of the variables (species) of the community dataset, it will be transformed to presence-absence and is first added to the environmental dataset. When you select the environmental dataset to be the community dataset as well, then you can select variables of the environmental dataset as response variable.
- **Explanatory** Type the right-hand side of the model formula (~ explanatory), or select and double-click for variables and select and click for operators to construct the right-hand side of the model formula.
- **Remove site with name** The name of the site to be removed from the environmental dataset.
- **Plot options** The options provide various functions that can be used to plot regression results of the current model (shown on top of the window; should have been estimated first).
  - Option "tabular" chooses function `plot` to plot presence-absence of the response variable against the selected categorical variable (shown on the right).
  - Option "diagnostic plots" chooses functions `plot.lm` or `gam.check` to plot diagnostic plots. For regression trees and artificial neural networks, the predicted values are plotted against the original presence-absence information.
  - Option "levene test" chooses function `levene.test` and plots residuals of the selected categorical variable (shown on the right).
  - Option "term plot" chooses functions `termplot` or `plot.gam` to plot a termplot of the selected categorical variable (shown on the right).
  - Option "effect plot" chooses function `effect` to plot an effect plot of the selected variable (shown on the right). (The menu option of the R-Commander of models > Graphs plots all the variables).
  - Option "qq plot" chooses function `qq.plot` to plot the residuals from the model.
  - Option "result plot (new)" chooses an appropriate `predict` function to plot a new plot of the model predictions for the selected variable (shown on the right).
  - Option "result plot (add)" chooses an appropriate `predict` function to add a new plot of the model predictions for the selected variable (shown on the right)
  - Option "result plot (interpolate)" chooses an appropriate `predict` function to add a new plot of the model predictions for the selected variable (shown on the right). This model is predicted

from a new dataset that only contains 1000 interpolated values for the selected explanatory variable.

Option "cr plot" chooses function `cr.plots` to plot a component + residual plots of the selected variable (shown on the right). (The menu option of the R-Commander of models > Graphs plots all the variables.)

Option "av plot" chooses function `av.plots` to plot added variable plots of the selected variable (shown on the right). (The menu option of the R-Commander of models > Graphs plots all the variables and has an option of identifying sites with the mouse.)

Option "influence plot" chooses function `influence.plot` to plot influence plots. (The menu option of the R-Commander of models > Graphs has an option of identifying sites with the mouse.)

Option "multcomp" chooses function `glht` to plot simultaneous confidence intervals of the selected categorical variable (shown on the right).

Option "rpart" chooses functions `plot.rpart` and `text.rpart` to plot a dendrogram for the regression tree result.

- **Plot variable** Variable of the environmental dataset that is used for some plotting functions.
- **OK** Fit the selected models.
- **Plot** Plot results for the model with name that appears on top. The model options need to apply to the model (e.g. if a GLM method was used to fit the model, this option should also be selected when plotting the results).
- **Cancel** Close the window and do not estimate new regression models.

### Calculate distance matrix

This window calculates a distance matrix from the community dataset and provides the following options:

- **Save result as** The name for the new distance matrix that will be calculated after "OK" was clicked.
- **Distance** Ecological distance measure. Passed as argument for "method" for function `vegdist`.
- **Make community dataset** Make the data frame derived from the new distance matrix the active community data set. This distance matrix can be used directly in the other menus for analysis of ecological distance after selecting the "as.dist" options of these windows.
- **OK** Calculate the distance matrix.
- **Cancel** Close the window and do not calculate a new distance matrix.

### Unconstrained ordination

The window fits and plots unconstrained ordination models. The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the unconstrained ordination model after "OK" was clicked, or the name of the object that will be plotted when Plot is clicked. In case that you saved a result earlier, then you can plot the result by typing in the name of previous result first in this box.

- **Ordination method** Select the method of ordination analysis.
  - Option "PCA" fits a Principal Components Analysis model with function `rda`.
  - Option "PCA (prcomp)" fits a Principal Components Analysis model with function `prcomp`.
  - Option "PCoA" fits a Principal Coordinates Analysis model with function `cmdscale` using the distance measure selected on the right-hand side (except if the community matrix is interpreted as distance matrix).
  - Option "PCoA (Caillez)" fits a Principal Coordinates Analysis model with function `cmdscale` using the distance measure selected on the right-hand side (except if the community matrix is interpreted as distance matrix) and setting `add=T`.
  - Option "CA" fits a Correspondence Analysis (Reciprocal Averaging) model with function `cca`.
  - Option "DCA" fits a Detrended Correspondence Analysis model with function `decorana`.
  - Option "metaMDS" fits a Non-metric Multidimensional Scaling model with function `metaMDS` using the distance measure selected on the right-hand side (except if the community matrix is interpreted as distance matrix).
  - Option "NMS (standard)" fits a Non-metric Multidimensional Scaling model with function `NMStandom` using the distance measure selected on the right-hand side (except if the community matrix is interpreted as distance matrix).
- **Distance** Select the distance measure for the PCoA and NMS methods (other methods have fixed intrinsic distance measures [Euclidean or chi] that can not be changed).
  - For the methods that provide ordinations based on a distance matrix (PCoA and NMSstandard): passed as argument for "method" for function `vegdist` that calculates the distance matrix first.
  - Passed as argument for "distance" for function `metaMDS`.
- **PCoA or NMS axes** Select the number of axes to feature in PCoA and NMS results. Passed as argument for "k" for functions `cmdscale`, `metaMDS` or `NMStandom`.
- **NMS permutation** Select the number of permutations for the NMS results. The solution with the lowest stress after all permutations of random starting positions will be provided. Passed as argument for "trymax" for function `metaMDS` or argument for "perm" for function `NMStandom`.
- **PCoA or NMS species** Fit species scores to PCoA and NMS results with function `add.spec.scores`. This function adds some other information for PCoA.
- **Model summary** Provide a summary of the ordination with functions `summary.cca`, `summary.decorana` or otherwise list the model object.
- **Scaling** Provide the scaling method. Passed as argument for "scaling" for functions `summary.cca`, `summary.decorana` or `add.spec.scores`.
- **as.dist(Community)** Treat the community dataset as a distance matrix. The community dataset will be used as a distance matrix (via `as.dist`) for unconstrained ordination methods that use a distance matrix as input (`cmdscale` and `NMStandom` for ordination results and via `ordicluster`, `lines.spantree`, `ordicluster2`, `ordinearest` or `distdisplayed` for plotting options).
- **Plot method** The options provide various functions that can be used to plot ordination results, or to add information to ordination diagrams.
  - Option "plot" chooses function `plot.cca` to plot results from `rda`, `cca`, `metaMDS` or `decorana` and function `plot` to plot the other ordination results (obtained by function `scores`).
  - Option "ordiplot" chooses function `ordiplot` to plot ordination results.

Option "ordiplot empty" chooses function `ordiplot` to plot ordination results, but sites and species will be invisible.

Option "identify sites" chooses function `identify.ordiplot` to add names of sites to site symbols (circles) created by function `ordiplot`. You can choose where the name is added by left-clicking in the quadrant next to the symbol where you want to symbol to be plotted. You can stop identifying sites by right-clicking.

Option "identify species" chooses function `identify.ordiplot` to add names of species to species symbols (crosses) created by function `ordiplot`. You can choose where the name is added by left-clicking in the quadrant next to the symbol where you want to symbol to be plotted. You can stop identifying species by right-clicking.

Option "text sites" chooses function `text.ordiplot` to add names of all sites to ordination diagrams created by function `ordiplot`.

Option "text species" chooses function `text.ordiplot` to add names of all species to ordination diagrams created by function `ordiplot`.

Option "points sites" chooses function `points.ordiplot` to add symbols for all sites to ordination diagrams created by function `ordiplot`.

Option "points species" chooses function `points.ordiplot` to add symbols for all species to ordination diagrams created by function `ordiplot`.

Option "origin axes" adds a horizontal and vertical line through the origin of the ordination graph (the origin is the location with coordinates [0,0]).

Option "envfit" chooses function `envfit` to add information for the variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordihull" chooses function `ordihull` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordiarrows" chooses function `ordiarrows` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordisegments" chooses function `ordisegments` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordispider" chooses function `ordispider` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordiellipse" chooses function `ordiellipse` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordisurf" chooses function `ordisurf` to add information for the continuous variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordicluster" chooses function `ordicluster` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix.) to ordination diagrams created by function `ordiplot`.

Option "ordispantree" chooses function `lines.spantree` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "ordibubble" chooses function `ordibubble` to add information for the continuous variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordisymbol" chooses function `ordisymbol` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`. Make sure that you click in the graph to show where the legend should be placed!

Option "ordivector" chooses function `ordivector` to add information on the selected species of the community dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`. You should first make the community dataset the environmental dataset to get the list of species on the right-hand side.

Option "ordivector interpretation" chooses function `ordivector` to add information on the selected species of the community dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`. You should first make the community dataset the environmental dataset to get the list of species on the right-hand side. The function will drop down perpendicular lines from each site to the line connecting the origin and the species position.

Option "ordicluster2" chooses function `ordicluster2` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "ordinearest" chooses function `ordinearest` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "ordiequilibriumcircle" chooses function `ordiequilibriumcircle` to plot an equilibrium circle to ordination diagrams created by function `ordiplot` from the Principal Components Analysis fitted by `rda`.

Option "distance displayed" compares the distances between each pair of sites in a distance matrix (with distance measure selected in window above) with distances in ordination diagrams created by function `ordiplot` by means of function `distdisplayed`.

Option "screeplot.cca" provides a screeplot for PCA results obtained by function `rda` by means of function `screeplot.cca`.

Option "stress" provides a stress plot (Shepard diagram) for NMS results obtained by function `metaMDS` by means of function `stressplot`.

Option "coenocline" fits coenoclines for all species to the first ordination axis of ordination diagrams created by function `ordiplot` by means of function `ordicoeno`.

- **Plot variable** Variable of the environmental dataset that is used for some plotting functions. For Plot method "ordivector", make the community dataset the environmental dataset first. Some other plot methods may also work with the community dataset as the environmental dataset as well (e.g. "ordibubble", "ordisurf"). Some methods run into problems when the variable has missing observations: in this case, you may need to repeat the ordination analysis after removing sites with missing observations for the variable with the "remove NA" option of the Community dataset menu list.
- **axes** The position of the axes of the ordination result to be plotted in the ordination diagram ("1,2" selects the first two axes of the ordination result). Passed as argument for "choices" for functions `plot.cca`, `scores` or `ordiplot`.
- **add scores to dataframe** Adds the scores of the sites from the `ordiplot` graph to the environmental dataset using the model name combined with ".ax1" and ".ax2".

- **cex** The size of the characters in the resulting plot when "Plot" is clicked.
- **colour** The colour of the resulting plot when "Plot" is clicked.
- **OK** Fit the selected models.
- **Plot** Plot results for the model with name that appears on top. The model options need to apply to the model (e.g. if `rda` was used to fit the model, this option should also be selected when plotting the results).
- **Cancel** Close the window and do not fit or plot ordination models.

### Constrained ordination

The window fits and plots constrained ordination models to the community dataset, using variables of the environmental dataset to constrain the ordination model (direct gradient analysis, canonical ordination analysis). The menu provides the following options:

- **Save result as** The name for the new object that will save the results from the unconstrained ordination model after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you can plot the result by typing in the name of previous result first in this box.
- **Ordination method** Select the method of ordination analysis.
  - Option "RDA" fits a Redundancy Analysis model with function `rda`.
  - Option "CCA" fits a Canonical Correspondence Analysis (Reciprocal Averaging) model with function `cca`.
  - Option "capscale" fits a scaled Constrained Analysis of Principal Coordinates (distance-based Redundancy Analysis) with function `capscale` using the distance measure selected on the right-hand side (except if the community matrix is interpreted as distance matrix).
  - Option "CAPdiscrim" fits a Constrained Analysis of Principal Coordinates (based on discriminant analysis) with function `CAPdiscrim` using the distance measure selected on the right-hand side (except if the community matrix is interpreted as distance matrix) and the categorical variable selected as explanatory variable.
  - Option "prc" fits principal response curves with function `prc`. To implement the example provided in the documentation for the `prc` function, you need to include the additional steps of defining `pyrifos.env <- data.frame(dose,week)` and making this data set the environmental data set.
  - Option "multiconstrained (RDA)" provides the first row of all ANOVA results (`anova.cca`) for all possible pairwise combinations of the levels of the first explanatory variable (assumed to be a categorical variable) through function `multiconstrained` with `method="rda"`. (When you change contrast to a particular contrast indicator, you obtain an ordination result that can be analyzed further. For several plotting options, you need to change the community and environmental datasets to "newcommunity" and "newenvdata").
  - Option "multiconstrained (CCA)" provides the first row of all ANOVA results (`anova.cca`) for all possible pairwise combinations of the levels of the first explanatory variable (assumed to be a categorical variable) through function `multiconstrained` with `method="cca"`. (When you change contrast to a particular contrast indicator, you obtain an ordination result that can be analyzed further. For several plotting options, you need to change the community and environmental datasets to "newcommunity" and "newenvdata").
  - Option "multiconstrained (capscale)" provides the first row of all ANOVA results (`anova.cca`) for all possible pairwise combinations of the levels of the first explanatory variable (assumed

to be a categorical variable) through function `multiconstrained` with `method="capscale"`. (When you change contrast to a particular contrast indicator, you obtain an ordination result that can be analyzed further. For several plotting options, you need to change the community and environmental datasets to "newcommunity" and "newenvdata").

- **Distance** Select the distance measure for the CAP methods (other methods have fixed intrinsic distance measures [Euclidean or chi] that can not be changed). Passed as argument for "dist" for function `capscale` or `CAPdiscrim`. This argument is ignored by the actual functions if the community dataset is interpreted to be a distance matrix already.
- **Model summary** Provide a summary of the ordination with functions `summary.cca` or `summary.prc`, or otherwise list the model object (`CAPdiscrim`).
- **as.dist(Community)** Treat the community dataset as a distance matrix. The community dataset will be used as a distance matrix (via `as.dist`) for constrained ordination methods that can use a distance matrix as input (`capscale` or `CAPdiscrim` for ordination results and via `ordicluster`, `lines.spantree`, `ordicluster2`, `ordinearest` or `distdisplayed` for plotting options).
- **Scaling** Provide the scaling method. This option is not available for function `CAPdiscrim`. Passed as argument for "scaling" for function `summary.cca` or `summary.prc`.
- **permutations** Select the number of permutations for testing the significance of the constrained ordination by Monte-Carlo randomization tests. The default of "0" means that no permutation test will be done. Passed as argument for "permutations" for functions `permutest.cca`, `CAPdiscrim` or `envfit` (one of the plotting options) or as argument for "step" for function `anova.cca` (which is also called by `multiconstrained`).
- **Explanatory** Type the right-hand side of the model formula (~ explanatory), or select and double-click for variables and select and click for operators to construct the right-hand side of the model formula. It is possible to include conditional variables for partial ordination analysis, except for function `CAPdiscrim` and `prc`. For function `prc`, the explanatory variables should be separated by a comma and indicate the "treatment" and "time" factors.
- **Plot method** The options provide various functions that can be used to plot ordination results, or to add information to ordination diagrams.
 

Option "plot" chooses function `plot.cca` to plot results from `rda`, `cca` or `capscale`, function `plot.prc` to plot results from `prc` and function `plot` to plot the other ordination results (obtained by function `scores`).

Option "ordiplot" chooses function `ordiplot` to plot ordination results.

Option "ordiplot empty" chooses function `ordiplot` to plot ordination results, but sites and species will be invisible.

Option "identify sites" chooses function `identify.ordiplot` to add names of sites to site symbols (circles) created by function `ordiplot`. You can choose where the name is added by left-clicking in the quadrant next to the symbol where you want to symbol to be plotted. You can stop identifying sites by right-clicking.

Option "identify species" chooses function `identify.ordiplot` to add names of species to species symbols (crosses) created by function `ordiplot`. You can choose where the name is added by left-clicking in the quadrant next to the symbol where you want to symbol to be plotted. You can stop identifying species by right-clicking.

Option "identify centroids" chooses function `identify.ordiplot` to add names of centroids to centroid symbols (X) created by function `ordiplot`. You can choose where the name is

added by left-clicking in the quadrant next to the symbol where you want to symbol to be plotted. You can stop identifying species by right-clicking.

Option "text sites" chooses function `text.ordiplot` to add names of all sites to ordination diagrams created by function `ordiplot`.

Option "text species" chooses function `text.ordiplot` to add names of all species to ordination diagrams created by function `ordiplot`.

Option "text centroids" chooses function `text.ordiplot` to add names of all centroids to ordination diagrams created by function `ordiplot`.

Option "points sites" chooses function `points.ordiplot` to add symbols for all sites to ordination diagrams created by function `ordiplot`.

Option "points species" chooses function `points.ordiplot` to add symbols for all species to ordination diagrams created by function `ordiplot`.

Option "points centroids" chooses function `points.ordiplot` to add symbols for all centroids to ordination diagrams created by function `ordiplot`.

Option "origin axes" adds a horizontal and vertical line through the origin of the ordination graph (the origin is the location with coordinates [0,0]).

Option "envfit" chooses function `envfit` to add information for the variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordihull" chooses function `ordihull` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordiarrows" chooses function `ordiarrows` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordisegments" chooses function `ordisegments` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordispider" chooses function `ordispider` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordiellipse" chooses function `ordiellipse` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordisurf" chooses function `ordisurf` to add information for the continuous variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordicluster" chooses function `ordicluster` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "ordispantree" chooses function `lines.spantree` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "ordibubble" chooses function `ordibubble` to add information for the continuous variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`.

Option "ordisymbol" chooses function `ordisymbol` to add information for the categorical variable of the environmental dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`. Make sure that you click in the graph to show where the legend should be placed!

Option "ordivector" chooses function `ordivector` to add information on the selected species of the community dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`. You should first make the community dataset the environmental dataset to get the list of species on the right-hand side.

Option "ordivector interpretation" chooses function `ordivector` to add information on the selected species of the community dataset selected on the right-hand side to ordination diagrams created by function `ordiplot`. You should first make the community dataset the environmental dataset to get the list of species on the right-hand side. The function will drop down perpendicular lines from each site to the line connecting the origin and the species position.

Option "ordicluster2" chooses function `ordicluster2` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "ordinearest" chooses function `ordinearest` to add information (with distance measure selected in window above - except if the community matrix is interpreted as distance matrix) to ordination diagrams created by function `ordiplot`.

Option "distance displayed" compares the distances between each pair of sites in a distance matrix (with distance measure selected in window above) with distances in ordination diagrams created by function `ordiplot` by means of function `distdisplayed`.

Option "coenocline" fits coenoclines for all species to the first ordination axis of ordination diagrams created by function `ordiplot` by means of function `ordicoeno`.

- **Plot variable** Variable of the environmental dataset that is used for some plotting functions. For Plot method "ordivector", make the community dataset the environmental dataset first. Some other plot methods may also work with the community dataset as the environmental dataset as well (e.g. "ordibubble", "ordisurf"). Some methods run into problems when the variable has missing observations: in this case, you may need to repeat the ordination analysis after removing sites with missing observations for the variable with the "remove NA" option of the Community dataset menu list.
- **axes** The position of the axes of the ordination result to be plotted in the ordination diagram ("1,2" selects the first two axes of the ordination result). Passed as argument for "choices" for functions `plot.cca`, `scores` or `ordiplot`.
- **add scores to dataframe** Adds the scores of the sites from the `ordiplot` graph to the environmental dataset using the model name combined with ".ax1" and ".ax2".
- **cex** The size of the characters in the resulting plot when "Plot" is clicked.
- **colour** The colour of the resulting plot when "Plot" is clicked.
- **OK** Fit the selected models.
- **Plot** Plot results for the model with name that appears on top. The model options need to apply to the model (e.g. if `rda` was used to fit the model, this option should also be selected when plotting the results).
- **Cancel** Close the window and do not fit or plot ordination models.

## Clustering

This window performs various methods of cluster analysis based on the information of the community dataset. The menu provides the following options:

- **Save cluster as** The name for the new object that will save the results from the cluster analysis after "OK" was clicked, or the name of the object that will be plotted when "Plot" is clicked. In case that you saved a result earlier, then you can plot the result by typing in the name of previous result first in this box.
- **Cluster method** Select the method of ordination analysis.
  - Option "hclust" results in a cluster analysis fitted by function [hclust](#). The distance for the distance matrix derived from the community dataset is selected on the right-hand side.
  - Option "agnes" results in a cluster analysis fitted by function [agnes](#). The distance for the distance matrix derived from the community dataset is selected on the right-hand side.
  - Option "diana" results in a cluster analysis fitted by function [diana](#). The distance for the distance matrix derived from the community dataset is selected on the right-hand side.
  - Option "kmeans" results in a cluster analysis fitted by function [kmeans](#). This method is based on the Euclidean distance.
  - Option "cascadeKM" results in a cluster analysis fitted by function [cascadeKM](#). This method is based on the Euclidean distance as it is based on K-means clustering.
  - Option "pam" results in a cluster analysis fitted by function [pam](#). The distance for the distance matrix derived from the community dataset is selected on the right-hand side.
  - Option "clara" results in a cluster analysis fitted by function [clara](#). The distance for the distance matrix derived from the community dataset is selected on the right-hand side.
  - Option "fanny" results in a cluster analysis fitted by function [fanny](#). The distance for the distance matrix derived from the community dataset is selected on the right-hand side.
- **Distance** Ecological distance measure used for the distance matrix. Passed as argument for "method" for function [vegdist](#).
- **as.dist(Community)** Treat the community dataset as a distance matrix. The community dataset will be used as a distance matrix (via [as.dist](#)). This option is not available for [kmeans](#)).
- **cluster summary** Provide the results of the cluster analysis with [summary.agnes](#), [summary.diana](#), [summary.pam](#), [summary.clara](#) or [summary.fanny](#) or provide results of [hclust](#) or [kmeans](#)
- **cophenetic correlation** Calculate the correlation of the distances in the distance matrix with the distances in the dendrogram (estimated with function [cophenetic](#)) by the Mantel test ([mantel](#)). It only works for hierarchical clustering methods ([hclust](#), [agnes](#) and [diana](#)).
- **clusters** Determine a fixed number of clusters.
  - This number selects the number of clusters to be calculated by the non-hierarchical cluster methods as it is passed as argument for "centers" for function [kmeans](#) and argument for "k" for functions [pam](#), [clara](#) and [fanny](#).
  - This number selects the number of groups for the partition with the largest number of groups of the cascade as it is passed as argument for "sup.gr" for function [cascadeKM](#) (the argument for "inf.gr" is set to "2").
  - This number selects the number of clusters to be reported for cluster membership for hierarchical clustering methods ([hclust](#), [agnes](#) and [diana](#)) as determined by function [cutree](#): passed as argument for "k" for this function.

This number selects the number of rectangles to be plotted on a dendrogram with plotting option of "rectangles": passed as argument for "k" for function `rect.hclust`.

This number selects the number of clusters to be plotted with plotting option of "pruned dendrogram": passed as argument for "k" for function `clip.clust`.

- **Save cluster membership** Save the identity of the cluster to which each site belongs into the environmental data set. For hierarchical clustering methods (`hclust`, `agnes` and `diana`) as determined by function `cutree`, with parameter "k" obtained from the box above.
- **Cluster options** Choose the options that are available for some of the hierarchical clustering methods.
 

Options "average", "single", "complete", "ward", "median" and "centroid" can be passed meaningfully as argument for "method" for `hclust`.

Options "average", "single", "complete", "ward" and "weighted" can be passed meaningfully as argument for "method" for `agnes`.
- **Plot options** Choose the options that are available for plotting hierarchical clustering results (except for "cascadeKM").
 

Option "dendrogram1" selects function `plot.hclust`, `plot.agnes` or `plot.diana` to plot clustering results.

Option "dendrogram2" selects function `plot.hclust`, `plot.agnes` or `plot.diana` to plot clustering results with argument `hang` set to -1. This option will result in each branch of the dendrogram to reach "ground level".

Option "rectangles" selects function `rect.hclust` to plot rectangles around the number of cluster determined by option "clusters" selected above.

Option "pruned dendrogram" selects function `clip.clust` to prune the cluster to the number of cluster selected by option "clusters" selected above. This option may only work with cluster results obtained by `plot.hclust`.

Option "kgs" selects function `kgs` as one method of selecting the optimal number of clusters and plots its results.

Option "cophenetic" uses function `cophenetic` to the distance in the dendrogram against the distance of the distance matrix (calculated earlier for the clustering algorithm). A reference line ( $y=x$ ) is added to the graph.

Option "cascadeKM" selects function `plot.cascadeKM` to plot results obtained by function `cascadeKM`.
- **cex** The size of the characters in the resulting plot when "Plot" is clicked.
- **colour** The colour of the resulting plot when "Plot" is clicked.
- **OK** Fit the selected models.
- **Plot** Plot results for the cluster with name that appears on top. Plotting will only be meaningful for hierarchical methods (`hclust`, `agnes` and `diana`).
- **Cancel** Close the window and do not analyse or plot clusters..

### Compare distance matrices

This window calculates a distance matrix from the community dataset. This distance matrix can be analysed by a Mantel, MRPP or ANOSIM test based on information from the environmental dataset. You can compare two different community datasets if you make one the community dataset and the other one the environmental dataset. The menu provides the following options:

- **Type of test** Selects the type of test to be used.
  - Option "mantel" results in a Mantel test estimated by function `mantel`. The distance for distance matrix derived from the community dataset is selected below, the distance to be derived from the environmental dataset is selected on the right-hand side.
  - Option "anosim" results in a ANOSIM test estimated by function `anosim` as summarized by `summary.anosim`. The distance measure for the distance matrix derived from the community dataset is selected below, the categorical variable of the environmental dataset is selected at the right-hand side.
  - Option "mrpp" results in a MRPP test estimated by function `mrpp`. The distance measure for the distance matrix derived from the community dataset is selected below, the categorical variable of the environmental dataset is selected at the right-hand side.
  - Option "rankindex" results in a series of Mantel tests with a series of distance measures selected by function `rankindex` for the community dataset and the Euclidean distance for the environmental dataset (except for datasets that contain factors where `daisy` is used).
- **Environmental distance** The environmental distance is only used for the test option of "mantel" (test option of "rank index" makes its own choice in between "daisy" or "euclidean" distance). The distance determines the type of distance matrix that is obtained from the environmental data set.
  - Option "daisy" results in function `daisy` to be used for providing the distance matrix. This is the only realistic method for environmental datasets that contain categorical variables.
  - The other options are passed as arguments for "method" for function `vegdist`.
- **Community distance** Ecological distance measure used for the distance matrix obtained from the community data set.
  - Passed as argument for "method" for function `vegdist`.
  - For the "rankindex" type of test, a series of distance measures are tested automatically.
- **Environmental variable** Selection of the environmental variable(s). Some methods run into problems when the variable has missing observations: in this case, you may need to repeat the ordination analysis after removing sites with missing observations for the variable with the "remove NA" option of the Community dataset menu list.
  - For test option "mantel", when "all" is selected, then the distance matrix is calculated for all variables of the environmental dataset. For environmental datasets with some categorical variables, only environmental distance "daisy" will result in actual distance matrices.
  - For test option "mantel", when a variable is selected, then the distance matrix is only calculated for that variable. In case that the variable is categorical, then the `daisy` distance is used automatically.
  - For test option "anosim", the selected environmental variable is passed as argument for "grouping" for function `anosim`.
  - For test option "mrpp", the selected environmental variable is passed as argument for "grouping" for function `mrpp`.
  - For test option "rankindex", when "all" is selected, then the environmental dataset is passed as argument for "grad" for function `rankindex`.
  - For test option "rankindex", the selected variable is passed as argument for "grad" for function `rankindex`.
- **as.dist(Community)** Treat the community dataset as a distance matrix. The community dataset will be used as a distance matrix (via `as.dist`).

- **Plot results** Plots the distances of the community dataset against the distance of the environmental dataset for test options "mantel", "anosim" and "mrpp". For categorical variables (the only possibility for "anosim" and "mrpp"), environmental distance equals "0" if sites belong to the same group and "1" if they belong to a different group except if they are ordered categorical variables (depending on the results of the [daisy](#) distance; for ordered factors, it is recommended to create a new factor that is unordered and use this variable for the analysis; see [factor](#)).
- **permutations** Number of permutations. Passed as argument for "permutations" for functions [mantel](#), [anosim](#) and [mrpp](#).
- **correlation** Correlation method. Passed as argument for "method" for function [mantel](#).
- **OK** Estimate the selected tests.
- **Cancel** Close the window and do estimate a new test.

### Author(s)

Roeland Kindt (with some help from Jari Oksanen)

### References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

---

CAPdiscrim

*Canonical Analysis of Principal Coordinates based on Discriminant Analysis*

---

### Description

This function provides a method for CAP as described by the authors of the ordination method. The CAP method implemented in **vegan** through [capscale](#) conforms more to distance-based Redundancy Analysis (Legendre & Anderson, 1999) than to the original description for CAP (Anderson & Willis, 2003).

### Usage

```
CAPdiscrim(formula,data,dist="bray",axes=4,m=0,permutations=0)
```

### Arguments

formula	Formula with a community data frame (with sites as rows, species as columns and species abundance as cell values) or distance matrix on the left-hand side and a categorical variable on the right-hand side (only the first explanatory variable will be used).
data	Environmental data set.

dist	Method for calculating ecological distance with function <code>vegdist</code> : partial match to "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "morisita", "horn" or "mountford". This argument is ignored in case that the left-hand side of the formula already is a distance matrix.
axes	Number of PCoA axes ( <code>cmdscale</code> ) to provide in the result.
m	Number of PCoA axes to be investigated by discriminant analysis ( <code>lda</code> ). If m=0 then the number of axes that provides the best distinction between the groups is calculated (following the method of Anderson and Willis).
permutations	The number of permutations for significance testing.

### Details

This function provides a method of Constrained Analysis of Principal Coordinates (CAP) that conforms to the description of the method by the developers of the method, Anderson and Willis. The method investigates the results of a Principal Coordinates Analysis (function `cmdscale`) with linear discriminant analysis (`lda`). Anderson and Willis advocate to use the number of principal coordinate axes that result in the best prediction of group identities of the sites.

For permutations > 0, the analysis is repeated by randomising the observations of the environmental data set. The significance is estimated by dividing the number of times the randomisation generated a larger percentage of correct predictions.

### Value

The function returns an object with information on CAP based on discriminant analysis. The object contains following elements:

PCoA	the positions of the sites as fitted by PCoA
m	the number of axes analysed by discriminant analysis
tot	the total variance (sum of all eigenvalues of PCoA)
varm	the variance of the m axes that were investigated
group	the original group of the sites
CV	the predicted group for the sites by discriminant analysis
percent	the percentage of correct predictions
x	the positions of the sites provided by the discriminant analysis
F	the squares of the singular values of the discriminant analysis
manova	the results for MANOVA with the same grouping variable
signi	the significance of the percentage of correct predictions
manova	a summary of the observed randomised prediction percentages

The object can be plotted with `ordiplot`, and species scores can be added by `add.spec.scores`.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Legendre, P. & Anderson, M.J. (1999). Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. *Ecological Monographs* 69: 1-24.

Anderson, M.J. & Willis, T.J. (2003). Canonical analysis of principal coordinates: a useful method of constrained ordination for ecology. *Ecology* 84: 511-525.

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
library(MASS)
data(dune)
data(dune.env)
Ordination.model1 <- CAPdiscrim(dune~Management, data=dune.env,
  dist="bray", axes=2, m=0)
Ordination.model1
plot1 <- ordiplot(Ordination.model1)
ordisymbol(plot1, dune.env, "Management", legend=FALSE)
## CLICK IN THE GRAPH TO INDICATE THE POSITION FOR THE LEGEND
## IN CASE THAT THE OPTION WAS LEGEND=TRUE.
```

---

caprescale

*Rescaling of Capscale Results to Reflect Total Sums of Squares Of Distance Matrix*

---

## Description

This is a simple function that rescales the ordination coordinates obtained from the distance-based redundancy analysis method implemented in **vegan** through [capscale](#). The rescaling of the ordination coordinates results in the distances between fitted site scores in ordination results (scaling=1 obtained via [ordiplot](#) to be equal to the distances between sites on the axes corresponding to positive eigenvalues obtained from principal coordinates analysis ([cmdscales](#)).

## Usage

```
caprescale(x, verbose=FALSE)
```

## Arguments

x	Ordination result obtained with <a href="#">capscale</a> .
verbose	Give some information on the pairwise distances among sites (TRUE) or not.

## Details

The first step of distance-based redundancy analysis involves principal coordinates analysis whereby the distances among sites from a distance matrix are approximated by distances among sites in a multidimensional configuration (ordination). In case that the principal coordinates analysis does not result in negative eigenvalues, then the distances from the distance matrix are the same as the distances among the sites in the ordination. In case that the principal coordinates analysis results in negative eigenvalues, then the distances among the sites on all ordination axes are related to the sum of positive eigenvalues, a sum which is larger than the sum of squared distances of the distance matrix.

The distance-based redundancy analysis method implemented in **vegan** through `capscale` uses a specific rescaling method for ordination results. Function `capscale` modifies the results of `capscale` so that an ordination with `scaling=1` (a distance biplot) obtained via `ordiplot` preserves the distances reflected in the principal coordinates analysis implemented as the first step of the analysis. See Legendre and Legendre (1998) about the relationship between fitted site scores and eigenvalues.

## Value

The function modifies and returns an object obtained via `capscale`.

## Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

- Legendre, P. & Legendre, L. (1998). Numerical Ecology. Amsterdam: Elsevier. 853 pp.
- Legendre, P. & Anderson, M.J. (1999). Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. Ecological Monographs 69: 1-24.

## Examples

```
library(vegan)
library(MASS)
data(dune)
data(dune.env)
Distmatrix.1 <- vegdist(dune,method='bray')
Ordination.model1 <- cmdscale(Distmatrix.1, k=19, eig=TRUE, add=FALSE)
# Sum of all eigenvalues
sum(Ordination.model1$eig)
# [1] 4.395807541512926
sum(Ordination.model1$eig[1:14])
# [1] 4.593946896588808
Distmatrix.2 <- as.matrix(vegdist(Ordination.model1$points[,1:14],method='euc'))
totalsumsquares1 <- sum(Distmatrix.2^2)/(2*20)
# Sum of distances among sites in principal coordinates analysis on axes
# corresponding to positive eigenvalues
totalsumsquares1
# [1] 4.593946896588808
Ordination.model2 <- capscale(dune ~ Management,dune.env,dist='bray', add=FALSE)
```

```
# Total sums of positive eigenvalues of the distance-based redundancy analysis
Ordination.model2$CA$tot.chi+Ordination.model2$CCA$tot.chi
# [1] 4.593946896588808
Ordination.model3 <- caprescale(Ordination.model2, verbose=TRUE)
Distmatrix.3 <- as.matrix(vegdist(summary(Ordination.model3,axes=17,scaling=1)$constraints,method='euc'))
totalsumsquares2 <- sum((Distmatrix.3)^2)/(2*20)/19
totalsumsquares2
# [1] 4.593946896588808
```

---

crosstabanalysis

*Presence-absence Analysis by Cross Tabulation*


---

## Description

This function makes a cross-tabulation of two variables after transforming the first variable to presence-absence and then returns results of `chisq.test`.

## Usage

```
crosstabanalysis(x,variable,factor)
```

## Arguments

x	Data set that contains the variables "variable" and "factor".
variable	Variable to be transformed in presence-absence in the resulting cross-tabulation.
factor	Variable to be used for the cross-tabulation together with the transformed variable.

## Value

The function returns the results of `chisq.test` on a crosstabulation of two variables, after transforming the first variable to presence-absence first.

## Author(s)

Roeland Kindt

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune.env)
crosstabanalysis(dune.env,"Manure","Management")
```

---

deviancepercentage	<i>Calculate Percentage and Significance of Deviance Explained by a GLM</i>
--------------------	---

---

### Description

This function calculates the percentage of deviance explained by a GLM model and calculates the significance of the model.

### Usage

```
deviancepercentage(x,data,test="F",digits=2)
```

### Arguments

x	Result of GLM as calculated by <code>glm</code> or <code>glm.nb</code> .
data	Data set to be used for the null model (preferably the same data set used by the 'full' model).
test	Test statistic to be used for the comparison between the null model and the 'full' model as estimated by <code>anova.glm</code> or <code>anova.negbin</code> : partial match of one of "Chisq", "F" or "Cp".
digits	Number of digits in the calculation of the percentage.

### Details

The function calculates the percentage of explained deviance and the significance of the 'full' model by contrasting it with the null model.

For the null model, the data is subjected to `na.omit`. You should check whether the same data are used for the null and 'full' models.

### Value

The function calculates the percentage of explained deviance and the significance of the 'full' model by contrasting it with the null model by ANOVA. The results of the ANOVA are also provided.

### Author(s)

Roeland Kindt

### References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune)
data(dune.env)
dune.env$Agrsto<- dune$Agrsto
Count.model1 <- glm(Agrsto ~ Management + A1, family=quasipoisson(link=log),
  data=dune.env, na.action=na.omit)
summary(Count.model1)
deviancepercentage(Count.model1,dune.env)
```

---

dist.eval

*Distance Matrix Evaluation*

---

## Description

Provides one test of a distance matrix, and continuous with `distconnected` (**vegan**).

## Usage

```
dist.eval(x,dist)
```

## Arguments

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
dist	Method for calculating ecological distance with function <code>vegdist</code> : partial match to "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "morisita", "horn" or "mountford".

## Details

This functions provides two tests of a distance matrix.

The first test checks whether any pair of sites that share some species have a larger distance than any other pair of sites that do not share any species. In case that cases are found, then a warning message is given.

The second test is the one implemented by the `distconnected` function (**vegan**). The `distconnected` test is only calculated for distances that calculate a value of 1 if sites share no species (i.e. not manhattan or euclidean), using the threshold of 1 as an indication that the sites do not share any species. Interpretation of analysis of distance matrices that provided warnings should be cautious.

## Value

The function tests whether distance matrices have some desirable properties and provide warnings if this is not the case.

## Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune)
dist.eval(dune,"euclidean")
dist.eval(dune,"bray")
```

---

dist.zeroes

*Distance Matrix Transformation*

---

## Description

Sample units without any species result in "NaN" values in the distance matrix for some of the methods of `vegdist` (**vegan**). The function replaces "NA" by "0" if both sample units do not contain any species and "NA" by "1" if only one sample unit does not have any species.

## Usage

```
dist.zeroes(comm,dist)
```

## Arguments

comm	Community data frame with sites as rows, species as columns and species abundance as cell values.
dist	Distance matrix as calculated with function <code>vegdist</code> .

## Details

This functions changes a distance matrix by replacing "NaN" values by "0" if both sample units do not contain any species and by "1" if only one sample unit does not contain any species.

Please note that there is a valid reason (deliberate removal of zero abundance values from calculations) that the original distance matrix contains "NaN", so you may not wish to do this transformation and remove sample units with zero abundances from further analysis.

## Value

The function provides a new distance matrix where "NaN" values have been replaced by "0" or "1".

## Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
matrix <- array(0,dim=c(5,3))
matrix[4,] <- c(1,2,3)
matrix[5,] <- c(1,0,0)
dist1 <- vegdist(matrix,method="kulc")
dist1
dist2 <- dist.zeroes(matrix,dist1)
dist2
```

---

distdisplayed	<i>Compare Distance Displayed in Ordination Diagram with Distances of Distance Matrix</i>
---------------	---

---

## Description

This function compares the distance among sites as displayed in an ordination diagram (generated by `ordiplot`) with the actual distances among sites as available from a distance matrix (as generated by `vegdist`).

## Usage

```
distdisplayed(x, ordiplot, distx = "bray", plotit = T, addit = F,
  method = "spearman", permutations = 100, abline = F, gam = T, ...)
```

## Arguments

x	Community data frame (with sites as rows, species as columns and species abundance as cell values) or distance matrix.
ordiplot	Ordination diagram generated by <code>ordiplot</code> or distance matrix.
distx	Ecological distance used to calculate the distance matrix (theoretically the same distance as displayed in the ordination diagram); passed to <code>vegdist</code> and partial match to "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "morisita", "horn", "mountford", "raup", "binomial" or "chao". This argument is ignored in case that "x" is already a distance matrix.
plotit	Should a plot comparing the distance in ordination diagram (or the distance matrix) with the distance from the distance matrix be generated (or not).
addit	Should the GAM regression result be added to an existing plot (or not).

method	Method for calculating the correlation between the ordination distance and the complete distance; from function <code>mantel</code> passed to function <code>cor</code> : "pearson", "spearman" or "kendall".
permutations	Number of permutations to assess the significance of the Mantel test; passed to <code>mantel</code> .
abline	Should a reference line ( $y=x$ ) be added to the graph (or not).
gam	Evaluate the correspondence between the original distance and the distance from the ordination diagram with GAMs estimated by <code>gam</code> .
...	Other arguments passed to <code>mantel</code> .

### Details

This function compares the Euclidean distances (between sites) displayed in an ordination diagram with the distances of a distance matrix. Alternatively, the distances of one distance matrix are compared against the distances of another distance matrix.

These distances are compared by a Mantel test (`mantel`) and (optionally) a GAM regression (`gam`). Optionally, a graph is provided comparing the distances and adding GAM results. .

### Value

The function returns the results of a Mantel test and (optionally) the results of a GAM analysis.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

### References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

### Examples

```
library(vegan)
library(mgcv)
data(dune)
distmatrix <- vegdist(dune,method="kulc")
ordination.model1 <- cmdscale(distmatrix,k=2)
ordiplot1 <- ordiplot(ordination.model1)
distdisplayed(dune,ordiplot=ordiplot1,distx="kulc",plotit=TRUE,
              method="spearman",permutations=100,gam=TRUE)
```

disttransform

*Community Matrix Transformation*

---

**Description**

Transforms a community matrix. Some transformation methods are described by distances for the original community matrix that result in the same distance matrix as calculated with the euclidean distance from the transformed community matrix. In many cases (methods of hellinger, profiles, chi.square and pa), the method makes use of function `decostand` (**vegan**).

**Usage**

```
disttransform(x,method="hellinger")
```

**Arguments**

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
method	Distance measure for the original community matrix that the euclidean distance will calculate for the transformed community matrix: partial match to "hellinger", "chord", "profiles", "chi.square", "log", "square" or "pa".

**Details**

This functions transforms a community matrix.

Some transformation methods ("hellinger", "chord", "profiles" and "chi.square") have the behaviour that the euclidean distance from the transformed matrix will equal a distance of choice for the original matrix. For example, using method "hellinger" and calculating the euclidean distance will result in the same distance matrix as by calculating the Hellinger distance from the original community matrix.

**Value**

The function returns a transformed community matrix.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Legendre, P. & Gallagher, E.D. (2001). Ecologically meaningful transformations for ordination of species data. *Oecologia* 129: 271-280.

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune)
Community.1 <- disttransform(dune, method='hellinger')
Distmatrix.1 <- vegdist(Community.1,method='euclidean')
Distmatrix.1
```

---

 diversityresult

*Alternative Diversity Results*


---

## Description

Provides alternative methods of obtaining results on diversity statistics than provided directly by functions `diversity`, `fisher.alpha`, `specpool` and `specnumber` (all from **vegan**), although these same functions are called. Some other statistics are also calculated such as the reciprocal Berger-Parker diversity index and abundance (not a diversity statistic). The statistics can be calculated for the entire community, for each site separately, the mean of the sites can be calculated or a jackknife estimate can be calculated for the community.

## Usage

```
diversityresult(x,y="", factor, level, index="Shannon",method="all",sortit=F,
  digits=8)
diversitycomp(x,y="", factor1, factor2="", index="Shannon",
  method="all",sortit=F,...)
```

## Arguments

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
y	Environmental data frame.
factor	Variable of the environmental data frame that defines subsets to calculate diversity statistics for.
level	Level of the variable to create the subset to calculate diversity statistics.
index	Type of diversity statistic with "richness" to calculate species richness, "abundance" to calculate abundance, "Shannon" to calculate the Shannon diversity index, "Simpson" to calculate 1-Simpson concentration index, "inverseSimpson" to calculate the reciprocal Simpson diversity index, "Logalpha" to calculate the log series alpha diversity index, "Berger" to calculate the reciprocal Berger-Parker diversity index, "Jevenness" to calculate one Shannon evenness index, "Eevenness" to calculate another Shannon evenness index, "jack1" to calculate the first-order jackknife gamma diversity estimator, "jack2" to calculate the second-order jackknife gamma diversity estimator, "chao" to calculate the Chao gamma diversity estimator and "boot" to calculate the bootstrap gamma diversity estimator.

method	Method of calculating the diversity statistics: "all" calculates the diversity of the entire community (all sites pooled), "s" calculates the diversity of each site separately, "mean" calculates the average diversity of the sites, "sd" calculates the standard deviation of the diversity of the sites, whereas "Jackknife" calculates the jackknifed diversity for the entire data frame. Method "s" is not allowed for <code>diversitycomp</code> .
sortit	Sort the sites by increasing values of the diversity statistic.
digits	Number of digits in the results.
factor1	Variable of the environmental data frame that defines subsets to calculate diversity statistics for.
factor2	Optional second variable of the environmental data frame that defines subsets to calculate diversity statistics for in a crosstabulation with the other variable of the environmental data frame.
...	Other arguments passed to function <code>diversityresult</code> .

### Details

These functions provide some alternative methods of obtaining results with diversity statistics, although functions `diversity`, `fisher.alpha`, `specpool`, `estimateR` and `specnumber` (all from **vegan**) are called to calculate the various statistics.

The reciprocal Berger-Parker diversity index is the reciprocal of the proportional abundance of the most dominant species.

J-evenness is calculated as:  $H / \ln(S)$  where  $H$  is the Shannon diversity index and  $S$  the species richness.

E-evenness is calculated as:  $\exp(H) / S$  where  $H$  is the Shannon diversity index and  $S$  the species richness.

The method of calculating the diversity statistics include following options: "all" calculates the diversity of the entire community (all sites pooled together), "s" calculates the diversity of each site separately, "mean" calculates the average diversity of the sites, whereas "Jackknife" calculates the jackknifed diversity for the entire data frame. Methods "s" and "mean" are not available for function `diversitycomp`. Gamma diversity estimators assume that the method is "all".

Functions `diversityresult` and `diversitycomp` allow to calculate diversity statistics for subsets of the community and environmental data sets. Function `diversityresult` calculates the diversity statistics for the specified level of a selected environmental variable. Function `diversitycomp` calculates the diversity statistics for all levels of a selected environmental variable separately. When a second environmental variable is provided, function `diversitycomp` calculates diversity statistics as a crosstabulation of both variables.

### Value

The functions provide alternative methods of obtaining diversity results. For function `diversitycomp`, the number of sites is provided as "n".

### Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune.env)
data(dune)
Diversity.1 <- diversityresult(dune, y=dune.env, factor='Management',
  level='NM', index='Shannon', method='s', sortit=TRUE, digits=3)
Diversity.1
diversitycomp(dune, y=dune.env, factor1='Management', factor2="Moisture",
  index='Shannon', method='all', sortit=TRUE, digits=3)
```

---

faramaea

*Faramaea occidentalis* abundance in Panama

---

## Description

This dataset describes the abundance (number of trees with diameter at breast height equal or larger than 10 cm) of the tree species *Faramaea occidentalis* as observed in a 1-ha quadrat survey from the Barro Colorado Island of Panama. For each quadrat, some environmental characteristics are also provided.

## Usage

```
data(faramaea)
```

## Format

A data frame with 45 observations on the following 8 variables.

UTM.EW a numeric vector

UTM.NS a numeric vector

Precipitation a numeric vector

Elevation a numeric vector

Age a numeric vector

Age.cat a factor with levels c1 c2 c3

Geology a factor with levels pT Tb Tbo Tc Tcm Tgo Tl

Faramaea.occidentalis a numeric vector

**Details**

Although the original survey documented tree species composition of all 1-ha subplots of larger (over 1 ha) sample plot, only the first (and sometimes the last) quadrats of the larger plots were included. This selection was made to avoid that larger sample plots dominated the analysis. This selection of sites is therefore different from the selection of the 50 1-ha quadrats of the largest sample plot of the same survey ([BCI](#) and [BCI.env](#))

This dataset is the main dataset used for the examples provided in chapters 6 and 7 of the Tree Diversity Analysis manual (Kindt & Coe, 2005).

**Source**

<http://www.sciencemag.org/cgi/content/full/295/5555/666/DC1>

**References**

Pyke CR, Condit R, Aguilar S and Lao S. (2001). Floristic composition across a climatic gradient in a neotropical lowland forest. *Journal of Vegetation Science* 12: 553-566.

Condit, R, Pitman, N, Leigh, E.G., Chave, J., Terborgh, J., Foster, R.B., Nunez, P., Aguilar, S., Valencia, R., Villa, G., Muller-Landau, H.C., Losos, E. & Hubbell, S.P. (2002). Beta-diversity in tropical forest trees. *Science* 295: 666-669.

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
data(faramea)
```

---

```
import.from.Excel      Import Community or Environmental Data from Excel or Access
```

---

**Description**

This function provides a method for importing community or environmental data from Excel (through functions [odbcConnectExcel](#) or [odbcConnectAccess](#)). In case of stacked data, a community dataset is formed by function [makecommunitydataset](#).

**Usage**

```
import.from.Excel(file = file.choose(), sheet = "community", sitenames = "sites",
  column = "species", value = "abundance", factor = "", level = "", cepnames = FALSE)
import.from.Access(file = file.choose(), table = "community", sitenames = "sites",
  column = "species", value = "abundance", factor = "", level = "", cepnames = FALSE)
```

**Arguments**

file	Location of the Excel file.
sheet	Name of the sheet of the Excel file to import: one of "community", "environmental" or "stacked".
table	Name of the table of the Access file to import: one of "community", "environmental" or "stacked".
sitenames	Name of categorical variable that provides the names for the sites.
column	Name of the categorical variable for the columns of the crosstabulation (typically indicating species); passed to <a href="#">makecommunitydataset</a> .
value	Name of numerical variable for the cells of the crosstabulation (typically indicating abundance). The cells provide the sum of all values in the data frame; passed to <a href="#">makecommunitydataset</a> .
factor	Name of the variable to calculate a subset of the data frame; passed to <a href="#">makecommunitydataset</a> .
level	Value of the subset of the factor variable to calculate a subset of the data frame; passed to <a href="#">makecommunitydataset</a> .
cepnames	Should the names of columns be abbreviated via <a href="#">make.cepnames</a> (TRUE) or not (FALSE).

**Details**

This function provides a method of method of importing community or environmental datasets through functions [odbcConnectExcel](#) and [sqlFetch](#).

For stacked datasets, a community data set is created with function [makecommunitydataset](#). For community data with more species than the limited number of columns in Excel, this may be the only option of importing a community dataset.

An additional advantage of the function is that the community and environmental data can be stored in the same file.

You may want to check compatibility of the community and environmental datasets with functions [check.datasets](#) and modify the community dataset through [same.sites](#).

**Value**

The function returns a dataset that can be used as community or environmental data for analysis in BiodiversityR.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
## Example is not run as requires RODBC
## Check for file "dune.xls" in the etc folder of BiodiversityR
##
#### Default 'ordiplot3d'
library(RODBC)
## Not run:
dunenew <- import.from.Excel(sheet="community", sitenames="sites")
dune.envnew <- import.from.Excel(sheet="environmental", sitenames="sites")

## End(Not run)
```

---

loaded.citations

*Give Citation Information for all Loaded Packages*


---

**Description**

This function provides citation information for all loaded packages.

**Usage**

```
loaded.citations()
```

**Details**

The function checks for the loaded packages via [.packages](#). Citation information is provided for the base package and for all the non-standard packages via [citation](#).

**Value**

The function provides a list of all loaded packages and the relevant citation information.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

---

makecommunitydataset

*Make a Community Dataset from a Stacked Dataset*


---

**Description**

Makes a community data set from a stacked dataset (with separate variables for the site identities, the species identities and the abundance).

**Usage**

```
makecommunitydataset(x, row, column, value, factor="", level="", drop=F)
```

**Arguments**

x	Data frame.
row	Name of the categorical variable for the rows of the crosstabulation (typically indicating sites)
column	Name of the categorical variable for the columns of the crosstabulation (typically indicating species)
value	Name of numerical variable for the cells of the crosstabulation (typically indicating abundance). The cells provide the sum of all values in the data frame.
factor	Name of the variable to calculate a subset of the data frame.
level	Value of the subset of the factor variable to calculate a subset of the data frame.
drop	Drop rows without species (species with total abundance of zero are always dropped)

**Details**

This function calculates a cross-tabulation from a data frame, summing up all the values of the numerical variable identified as variable for the cell values. If factor="", then no subset is calculated from the data frame in the first step.

**Value**

The function provides a community dataset from another data frame.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
library(vegan)
data(dune.env)
makecommunitydataset(dune.env, row='Manure', column='Use', value='A1')
makecommunitydataset(dune.env, row='Manure', column='Use', value='A1',
  factor='Management', level='NM')
```

---

multiconstrained	<i>Pairwise Comparisons for All Levels of a Categorical Variable by RDA, CCA or Capscale</i>
------------------	--

---

### Description

This function implements pairwise comparisons for categorical variable through [capscale](#), [cca](#) or [rda](#) followed by [anova.cca](#). The function simply repeats constrained ordination analysis by selecting subsets of data that correspond to two factor levels.

### Usage

```
multiconstrained(method="capscale", formula, data, distance = "bray"
, comm = NULL, add = FALSE, multicomp="", contrast=0, ...)
```

### Arguments

method	Method for constrained ordination analysis; one of "rda", "cca" or "capscale".
formula	Model formula as in <a href="#">capscale</a> , <a href="#">cca</a> or <a href="#">rda</a> . The LHS can be a community data matrix or a distance matrix for <a href="#">capscale</a> .
data	Data frame containing the variables on the right hand side of the model formula as in <a href="#">capscale</a> , <a href="#">cca</a> or <a href="#">rda</a> .
distance	Dissimilarity (or distance) index in <a href="#">vegdist</a> used if the LHS of the formula is a data frame instead of dissimilarity matrix; used only with function <a href="#">vegdist</a> and partial match to "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "morisita", "horn" or "mountford". This argument is only used for <a href="#">capscale</a> in case that the LHS of the formula is a community matrix.
comm	Community data frame which will be used for finding species scores when the LHS of the formula was a dissimilarity matrix as only allowed for <a href="#">capscale</a> . This is not used if the LHS is a data frame.
add	Logical indicating if an additive constant should be computed, and added to the non-diagonal dissimilarities such that all eigenvalues are non-negative in underlying Principal Co-ordinates Analysis; only applicable in <a href="#">capscale</a> .
multicomp	Categorical variable used to construct the contrasts from. In case that this variable is missing, then the first explanatory variable of the formula will be used.
contrast	Return the ordination results for the particular contrast indicated by this number (e.g. with 5 levels, one can choose in between contrast 1-10). In case=0, then the first row of the <a href="#">anova.cca</a> results for all contrasts is provided.
...	Other parameters passed to <a href="#">anova.cca</a> .

## Details

This function provides a simple expansion of `capscale`, `cca` and `rda` by conducting the analysis for subsets of the community and environmental datasets that only contain two levels of a categorical variable.

When the choice is made to return results from all contrasts (`contrast=0`), then the first row of the `anova.cca` tables for each contrast are provided. It is therefore possible to compare differences in results by modifying the "by" argument of this function (i.e. obtain the total of explained variance, the variance explained on the first axis or the variance explained by the variable alone).

When the choice is made to return results from a particular contrast (`contrast>0`), then the ordination result is returned and two new datasets ("newcommunity" and "newenvdata") are created that only contain data for the two selected contrasts.

## Value

The function returns an ANOVA table that contains the first rows of the ANOVA tables obtained for all possible combinations of levels of the first variable. Alternatively, it returns an ordination result for the selected contrast and creates two new datasets ("newcommunity" and "newenvdata")

## Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Legendre, P. & Anderson, M.J. (1999). Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. *Ecological Monographs* 69: 1-24.

Anderson, M.J. & Willis, T.J. (2003). Canonical analysis of principal coordinates: a useful method of constrained ordination for ecology. *Ecology* 84: 511-525.

## Examples

```
library(vegan)
library(MASS)
data(dune)
data(dune.env)
multiconstrained(method="capscale", dune~Management, data=dune.env,
  distance="bray", add=TRUE)
multiconstrained(method="capscale", dune~Management+Condition(A1),
  data=dune.env, distance="bray", add=TRUE, contrast=3)
```

## Description

The functions provide nested analysis of variance for a two-level hierarchical model. The functions are implemented by estimating the correct F-ratio for the main and nested factors (assuming the nested factor is random) and using the recommended permutation procedures to test the significance of these F-ratios. F-ratios are estimated from variance estimates that are provided by distance-based redundancy analysis ([capscale](#)) or non-parametric multivariate analysis of variance ([adonis](#)).

## Usage

```
nested.anova.dbrda(formula, data, method="euc", add=FALSE,
  permutations=100, warnings=FALSE)
nested.npmanova(formula, data, method="euc", permutations=100, warnings=FALSE)
```

## Arguments

formula	Formula with a community data frame (with sites as rows, species as columns and species abundance as cell values) or (for <code>nested.anova.dbrda</code> only) distance matrix on the left-hand side and two categorical variables on the right-hand side (with the second variable assumed to be nested within the first).
data	Environmental data set.
method	Method for calculating ecological distance with function <a href="#">vegdist</a> : partial match to "manhattan", "euclidean", "canberra", "bray", "kulczynski", "jaccard", "gower", "morisita", "horn" or "mountford". This argument is ignored in case that the left-hand side of the formula already is a distance matrix.
add	Should a constant be added to the off-diagonal elements of the distance-matrix (TRUE) or not.
permutations	The number of permutations for significance testing.
warnings	Should warnings be suppressed (TRUE) or not.

## Details

The functions provide two alternative procedures for multivariate analysis of variance on the basis of any distance measure. Function `nested.anova.dbrda` proceeds via [capscale](#), whereas `nested.npmanova` proceeds via [adonis](#). Both methods are complementary to each other as `nested.npmanova` always provides correct F-ratios and estimations of significance, whereas `nested.anova.dbrda` does not provide correct F-ratios and estimations of significance when negative eigenvalues are encountered or constants are added to the distance matrix, but always provides an ordination diagram.

The F-ratio for the main factor is estimated as the mean square of the main factor divided by the mean square of the nested factor. The significance of the F-ratio of the main factor is tested by permuting entire blocks belonging to levels of the nested factor. The significance of the F-ratio of the nested factor is tested by permuting sample units within strata defined by levels of the main factor.

## Value

The functions provide an ANOVA table.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Legendre, P. & Anderson, M. J. (1999). Distance-based redundancy analysis: testing multispecies responses in multifactorial ecological experiments. *Ecological Monographs* 69, 1-24.

Anderson, M.J. (2001). A new method for non-parametric multivariate analysis of variance. *Austral Ecology*, 26: 32-46.

McArdle, B.H. and M.J. Anderson. (2001). Fitting multivariate models to community data: A comment on distance-based redundancy analysis. *Ecology*, 82: 290-297.

**Examples**

```
library(vegan)
data(warcom)
data(warenv)
# use larger number of permutations for real studies
nested.npmanova(warcom~rift.valley+popshort, data=warenv, method="jac",
  permutations=10)
nested.anova.dbrda(warcom~rift.valley+popshort, data=warenv, method="jac",
  permutations=10)
```

---

NMSrandom

*Calculate the NMS Result with the Smallest Stress from Various Random Starts*

---

**Description**

This function provides a simplified version of the method of calculating NMS results implemented by the function `metaMDS` ([vegan](#)).

**Usage**

```
NMSrandom(x, perm=100, k=2, stressresult=F, method="isoMDS")
```

**Arguments**

x	Distance matrix.
perm	Number of permutations to select the configuration with the lowest stress.
k	Number of dimensions for the non metric scaling result; passed to <a href="#">isoMDS</a> or <a href="#">sammon</a> .
stressresult	Provide the calculated stress for each permutation.
method	Method for calculating the NMS: <a href="#">isoMDS</a> or <a href="#">sammon</a> .

### Details

This function is an easier method of calculating the best NMS configuration after various random starts than implemented in the `metaMDS` function (**vegan**). The function uses a distance matrix (as calculated for example by function `vegdist` from a community data set) and calculates random starting positions by function `initMDS` (**vegan**) analogous to `metaMDS`.

### Value

The function returns the NMS ordination result with the lowest stress (calculated by `isoMDS` or `sammon`), or the stress of each NMS ordination.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

### References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

### Examples

```
library(vegan)
library(MASS)
data(dune)
distmatrix <- vegdist(dune)
Ordination.model1 <- NMSrandom(distmatrix, perm=100, k=2)
Ordination.model1 <- add.spec.scores(Ordination.model1, dune,
  method='wa.scores')
Ordination.model1
```

---

nnetrandom

*Calculate the NNET Result with the Smallest Value from Various Random Starts*

---

### Description

This function provides the best solution from various calls to the `nnet` feed-forward artificial neural networks function (**nnet**).

### Usage

```
nnetrandom(formula, data, tries=10, leave.one.out=F, ...)
```

**Arguments**

formula	Formula as passed to <code>nnet</code> .
data	Data as passed to <code>nnet</code> .
tries	Number of calls to <code>nnet</code> to obtain the best solution.
leave.one.out	Calculate leave-one-out predictions.
...	Other arguments passed to <code>nnet</code> .

**Details**

This function makes various calls to `nnet`. If desired by the user, leave-one-out statistics are provided that report the prediction if one particular sample unit was not used for iterating the networks.

**Value**

The function returns the same components as `nnet`, but adds the following components:

range	Summary of the observed "values".
tries	Number of different attempts to iterate an ANN.
CV	Predicted class when not using the respective sample unit for iterating ANN.
successful	Test whether leave-one-out statistics provided the same class as the original class.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**Examples**

```
data(faramea)
faramea <- na.omit(faramea)
faramea$presence <- as.numeric(faramea$Faramea.occidentalis > 0)
attach(faramea)
library(nnet)
result <- nnetrandom(presence ~ Elevation, data=faramea, size=2,
  skip=FALSE, entropy=TRUE, trace=FALSE, maxit=1000, tries=100,
  leave.one.out=FALSE)
summary(result)
result$fitted.values
result$value
result2 <- nnetrandom(presence ~ Elevation, data=faramea, size=2,
  skip=FALSE, entropy=TRUE, trace=FALSE, maxit=1000, tries=50,
  leave.one.out=TRUE)
result2$range
result2$CV
result2$successful
```

---

ordicoeno

*Coenoclines for an Ordination Axis*

---

### Description

A graph is produced that summarizes (through GAM as implemented by `gam`) how the abundance of all species of the community data set change along an ordination axis (based on the position of sites along the axis and the information from the community data set).

### Usage

```
ordicoeno(x, ordiplot, axis=1, ...)
```

### Arguments

<code>x</code>	Community data frame with sites as rows, species as columns and species abundance as cell values.
<code>ordiplot</code>	Ordination plot created by <code>ordiplot</code> .
<code>axis</code>	Axis of the ordination graph (1: horizontal, 2: vertical).
<code>...</code>	Other arguments passed to functions <code>plot</code> and <code>points</code> .

### Details

This functions investigates the relationship between the species vectors and the position of sites on an ordination axis. A GAM (`gam`) investigates the relationship by using the species abundances of each species as response variable, and the site position as the explanatory variable. The graph shows how the abundance of each species changes over the gradient of the ordination axis.

### Value

The function plots coenoclines and provides the expected degrees of freedom (complexity of the relationship) estimated for each species by GAM.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

### References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
library(vegan)
library(mgcv)
data(dune)
Ordination.model1 <- rda(dune)
plot1 <- ordiplot(Ordination.model1, choices=c(1,2), scaling=1)
ordicoeno(dune, plot1)
```

ordisymbol

*Add Other Graphical Items to Ordination Diagrams***Description**

Functions to add some other graphical itmes to ordination diagrams than provided within **vegan** by [ordihull](#), [ordispider](#), [ordiarrows](#), [ordisegments](#), [ordigrid](#), [ordiellipse](#), [ordicluste](#)r and [lines.spantree](#).

**Usage**

```
ordisymbol(ordiplot, y, factor, col=1, rainbow=T, legend=T,...)
ordibubble(ordiplot,var,...)
ordicluste
```

r2(ordiplot, cluster, mingroups=1, maxgroups=nrow(ordiplot\$sites),...)
ordinearest(ordiplot, dist,...)
ordivector(ordiplot, spec, lty=2,...)
**Arguments**

ordiplot	An ordination graph created by <a href="#">ordiplot</a> ( <b>vegan</b> ).
y	Environmental data frame.
factor	Variable of the environmental data frame that defines subsets to be given different symbols.
var	Continous variable of the environmental dataset or species from the community dataset.
col	Colour (as <a href="#">points</a> ).
rainbow	Use rainbow colours.
legend	Add the legend (you need to click in the graph where the legend needs to be plotted).
cluster	Cluster object.
mingroups	Minimum of clusters to be plotted.
maxgroups	Maximum of clusters to be plotted..
dist	Distance matrix.
spec	Species name from the community dataset.
lty	Line type as specified for <a href="#">par</a> .
...	Other arguments passed to functions <a href="#">points</a> , <a href="#">symbols</a> , <a href="#">ordihull</a> or <a href="#">arrows</a> .

## Details

Function `ordisymbol` plots different levels of the specified variable in different symbols and different colours (if `rainbow` option was selected). When you opt to add the legend, then you need to click in the ordination graph to indicate the position of the legend.

Function `ordibubble` draws bubble diagrams indicating the value of the specified continuous variable. Circles indicate positive values, squares indicate negative values.

Function `ordicluster2` provides an alternative method of overlaying information from hierarchical clustering on an ordination diagram than provided by function `ordicluster`. The method draws convex hulls around sites that are grouped into the same cluster. You can select the minimum and maximum number of clusters that are plotted (i.e. the range of clustering steps to be shown).

Function `ordinearest` draws a vector from each site to the site that is nearest to it as determined from a distance matrix. When you combine the method with `lines.spantree` using the same distance measure, then you can evaluate in part how the minimum spanning tree was constructed.

Function `ordivector` draws a vector for the specified species on the ordination diagram and draws perpendicular lines from each site to a line that connects the origin and the head of species vector. This method helps in the biplot interpretation of a species vector as described by Jongman, ter Braak and van Tongeren (1995).

## Value

These functions add graphical items to an existing ordination diagram.

## Author(s)

Roeland Kindt (World Agroforestry Centre) and Jari Oksanen (`ordinearest`)

## References

Jongman, R.H.G, ter Braak, C.J.F & van Tongeren, O.F.R. (1987). *Data Analysis in Community and Landscape Ecology*. Pudog, Wageningen.

Kindt, R. & Coe, R. (2005). *Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies*.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune)
data(dune.env)
Ordination.model1 <- rda(dune)
plot1 <- ordiplot(Ordination.model1, choices=c(1,2), scaling=2)
ordisymbol(plot1,dune.env,"Management",legend=FALSE)
## Make sure you click in the graph to show where the legend should be placed
## In case that you choose option of legend=TRUE
plot2 <- ordiplot(Ordination.model1, choices=c(1,2), scaling=1)
distmatrix <- vegdist(dune, method='bray')
cluster <- hclust(distmatrix, method='single')
ordicluster2(plot2, cluster)
```

```
ordinearest(plot2,distmatrix,col=2)
ordivector(plot2,"Agrsto",lty=2)
```

---

PCAsignificance	<i>PCA Significance</i>
-----------------	-------------------------

---

### Description

Calculates the number of significant axes from a Principal Components Analysis based on the broken-stick criterion, or adds an equilibrium circle to an ordination diagram.

### Usage

```
PCAsignificance(pca,axes=8)
ordiequilibriumcircle(pca,ordiplot,...)
```

### Arguments

pca	Principal Components Analysis result as calculated by <a href="#">rda</a> ( <b>vegan</b> ).
axes	Number of axes to calculate results for.
ordiplot	Ordination plot created by <a href="#">ordiplot</a> ( <b>vegan</b> )
...	Other arguments passed to function <a href="#">arrows</a> .

### Details

These functions provide two methods of providing some information on significance for a Principal Components Analysis (PCA).

Function `PCAsignificance` uses the broken-stick distribution to evaluate how many PCA axes are significant. This criterion is one of the most reliable to check how many axes are significant. PCA axes with larger percentages of (accumulated) variance than the broken-stick variances are significant (Legendre and Legendre, 1998).

Function `ordiequilibriumcircle` draws an equilibrium circle to a PCA ordination diagram. Only species vectors with heads outside of the equilibrium circle significantly contribute to the ordination diagram (Legendre and Legendre, 1998). Vectors are drawn for these species. The function considers the scaling methods used by [rda](#) for `scaling=1`. The method should only be used for `scaling=1` and PCA calculated by function [rda](#).

### Value

Function `PCAsignificance` returns a matrix with the variances that are explained by the PCA axes and by the broken-stick criterion.

Function `ordiequilibriumcircle` plots an equilibrium circle and returns a list with the radius and the scaling constant used by [rda](#).

### Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Legendre, P. & Legendre, L. (1998). Numerical Ecology. 2nd English Edition. Elsevier.

Kindt, R. & Coe, R. (2005). Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
data(dune)
Ordination.model1 <- rda(dune)
PCAsignificance(Ordination.model1)
plot1 <- ordiplot(Ordination.model1, choices=c(1,2), scaling=1)
ordiequilibriumcircle(Ordination.model1,plot1)
```

---

radfitresult

*Alternative Rank Abundance Fitting Results*

---

## Description

Provides alternative methods of obtaining rank abundance curves than provided by functions `radfit`, `fisherfit` and `prestonfit` (**vegan**), although these same functions are called.

## Usage

```
radfitresult(x,y="", factor="", level, plotit=T)
```

## Arguments

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
y	Environmental data frame.
factor	Variable of the environmental data frame that defines subsets to calculate fitted rank-abundance curves for.
level	Level of the variable to create the subset to calculate fitted rank-abundance curves.
plotit	Plot the results obtained by <code>plot.radfit</code> .

## Details

These functions provide some alternative methods of obtaining fitted rank-abundance curves, although functions `radfit`, `fisherfit` and `prestonfit` (**vegan**) are called to calculate the actual results.

**Value**

The function returns the results from three methods of fitting rank-abundance curves:

```
radfit          results of radfit.
fisherfit      results of fisherfit.
prestonfit     results of prestonfit.
```

Optionally, a plot is provided of the `radfit` results by `plot.radfit`.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
library(vegan)
data(BCI)
BCIall <- t(as.matrix(colSums(BCI)))
radfitresult(BCIall)
```

---

rankabundance	<i>Rank Abundance Curves</i>
---------------	------------------------------

---

**Description**

Provides methods of calculating rank-abundance curves.

**Usage**

```
rankabundance(x,y="", factor="", level, digits=1, t=qt(0.975, df=n-1))
rankabunplot(xr, addit=F, labels="", scale="abundance", scaledx=F, type="o",
             xlim=c(min(xpos), max(xpos)), ylim=c(0, max(x[, scale])), specnames=c(1:5), ...)
rankabuncomp(x,y="", factor, scale="abundance", scaledx=F, type="o", rainbow=T,
             legend=T, xlim=c(1, max1), ylim=c(0, max2), ...)
```

**Arguments**

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
y	Environmental data frame.
factor	Variable of the environmental data frame that defines subsets to calculate rank abundance curves for.
level	Level of the variable to create the subset to calculate rank abundance curves.
digits	Number of digits in the results.
t	t-value to calculate confidence interval limits for the species proportion for cluster sampling (following Hayek and Buzas 1997).
xr	Result from rankabundance.
addit	Add rank abundance curve to an existing graph.
labels	Labels to plot at left of the rank abundance curves.
scale	Method of scaling the vertical axis. Method "abundance" uses abundance, "proportion" uses proportional abundance (species abundance / total abundance), "logabun" calculates the logarithm of abundance using base 10 and "accumfreq" accumulates the proportional abundance.
scaledx	Scale the horizontal axis to 100 percent of total number of species.
type	Type of plot (as in function <a href="#">plot</a> )
xlim	Limits for the horizontal axis.
ylim	Limits for the vertical axis.
specnames	Vector positions of species names to add to the rank-abundance curve.
rainbow	Use rainbow colouring for the different curves.
legend	Add the legend (you need to click in the graph where the legend needs to be plotted).
...	Other arguments to be passed to functions <a href="#">plot</a> or <a href="#">points</a> .

**Details**

These functions provide methods of calculating and plotting rank-abundance curves.

The vertical axis can be scaled by various methods. Method "abundance" uses abundance, "proportion" uses proportional abundance (species abundance / total abundance), "logabun" calculates the logarithm of abundance using base 10 and "accumfreq" accumulates the proportional abundance.

The horizontal axis can be scaled by the total number of species, or by 100 percent of all species by option "scaledx".

The method of calculating the confidence interval for species proportion is described in Hayek and Buzas (1997).

Functions rankabundance and rankabuncomp allow to calculate rank abundance curves for subsets of the community and environmental data sets. Function rankabundance calculates the rank abundance curve for the specified level of a selected environmental variable. Method rankabuncomp calculates the rank abundance curve for all levels of a selected environmental variable separately.

**Value**

The functions provide information on rankabundance curves. Function rankabundance provides information on abundance, proportional abundance, logarithmic abundance and accumulated proportional abundance. The function also provides confidence interval limits for the proportion of each species (plower, pupper) and the proportion of species ranks (in percentage).

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Hayek, L.-A. C. & Buzas, M.A. (1997). Surveying Natural Populations. Columbia University Press.

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
library(vegan)
data(dune.env)
data(dune)
RankAbun.1 <- rankabundance(dune)
RankAbun.1
rankabunplot(RankAbun.1,scale='abundance', addit=FALSE, specnames=c(1,2,3))
rankabuncomp(dune, y=dune.env, factor='Management',
             scale='proportion', legend=FALSE)
## CLICK IN THE GRAPH TO INDICATE WHERE THE LEGEND NEEDS TO BE PLACED
## IF YOU OPT FOR LEGEND=TRUE.
```

---

removeNAcomm

*Synchronize Community and Environmental Datasets*

---

**Description**

These functions may assist to ensure that the sites of the community dataset are the same sites as those from the environmental dataset, something that is assumed to be the case for the **BiodiversityR** and **vegan** packages.

**Usage**

```
same.sites(x,y)
check.datasets(x,y)
check.ordiscores(x,ord,check.species=T)
removeNAcomm(x,y,variable)
removeNAenv(x,variable)
removezerospecies(x)
subsetcomm(x,y,factor,level,returncomm=T)
```

### Arguments

x	Data frame assumed to be the community dataset with variables corresponding to species.
y	Data frame assumed to be the environmental dataset with variables corresponding to descriptors of sites.
ord	Ordination result.
check.species	Should the species scores be checked (TRUE) or not.
variable	Name of the variable from the environmental dataset with NA values that indicate those sites that should be removed.
factor	Variable of the environmental data frame that defines subsets to calculate diversity statistics for.
level	Level of the variable to create the subset to calculate diversity statistics.
returncomm	For the selected sites, return the community dataset (TRUE) or the environmental dataset.

### Details

Function `same.sites` provides a new data frame that has the same row names as the row names of the environmental data set and the same (species) variables as the original community data set. Sites from the original community data set that have no corresponding sites in the environmental data set are not included in the new community data set. (Hint: this function can be especially useful when some sites do not contain any species and where a community dataset was generated by the `makecommunitydataset` function.)

Function `check.datasets` checks whether the community and environmental data sets have the same number of rows, and (if this was the case) whether the rownames of both data sets are the same. The function also returns the dimensions of both data sets.

Function `check.ordiscores` checks whether the community data set and the ordination result have the same number of rows (sites) and columns (species, optional for `check.species==TRUE`), and (if this was the case) whether the row and column names of both data sets are the same. Site and species scores for the ordination result are obtained via function `scores` (**vegan**).

Functions `removeNAcomm` and `removeNAenv` provide a new data frame that does not contain NA for the specified variable. The specified variable is part of the environmental data set. These functions are particularly useful when using community and environmental datasets, as new community and environmental datasets can be calculated that contain information from the same sample plots (sites). An additional result of `removeNAenv` is that factor levels of any categorical variable that do not occur any longer in the new data set are removed from the levels of the categorical variable.

Function `removezerospecies` removes species from a community dataset that have total abundance that is smaller or equal to zero.

Function `subsetcomm` makes a subset of sites that contain a specified level of a categorical variable from the environmental data set. The same functionality of selecting subsets of the community or environmental data sets are implemented in various functions of **BiodiversityR** (for example `diversityresult`, `renyireresult` and `accumresult`) and have the advantage that it is not necessary to create a new data set. If a community dataset is returned, species that did not contain any individuals were removed from the data set. If an environmental dataset is returned, factor levels that did not occur were removed from the data set.

**Value**

The functions return a data frame or results of tests on the correspondence between community and environmental data sets.

**Author(s)**

Roeland Kindt (World Agroforestry Centre)

**References**

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

**Examples**

```
library(vegan)
data(dune.env)
data(dune)
dune.env2 <- dune.env
dune.env2[1:4,"Moisture"] <- NA
dune2 <- removeNAcomm(dune,dune.env2,"Moisture")
dune.env2 <- removeNAenv(dune.env2,"Moisture")
dune3 <- same.sites(dune,dune.env2)
check.datasets(dune,dune.env2)
check.datasets(dune2,dune.env2)
check.datasets(dune3,dune.env2)
dune4 <- subsetcomm(dune,dune.env,"Management","NM",returncomm=TRUE)
dune.env4 <- subsetcomm(dune,dune.env,"Management","NM",returncomm=FALSE)
dune5 <- same.sites(dune,dune.env4)
check.datasets(dune4,dune5)
```

---

renyiresult

*Alternative Renyi Diversity Results*


---

**Description**

Provides some alternative methods of obtaining results on Renyi diversity profile values than provided by [renyi](#) ([vegan](#)).

**Usage**

```
renyiresult(x,y="",factor,level,method="all",
  scales=c(0,0.25,0.5,1,2,4,8,Inf),evenness=F,...)
renyiplot(xr,addit=F,pch=1,ylim=c(0,m),labelit=T,legend=T,col=1,cex=1,
  rainbow=T,evenness=F,...)
renyiaccumresult(x,y="",factor,level,
  scales=c(0,0.25,0.5,1,2,4,8,Inf),permutations=100,...)
```

```
renyicomp(x,y,factor,sites=Inf,
          scales=c(0,0.25,0.5,1,2,4,8,Inf),permutations=100,plotit=T,...)
```

### Arguments

x	Community data frame with sites as rows, species as columns and species abundance as cell values.
y	Environmental data frame.
factor	Variable of the environmental data frame that defines subsets to calculate diversity profiles for.
level	Level of the variable to create the subset to calculate diversity profiles.
method	Method of calculating the diversity profiles: "all" calculates the diversity of the entire community (all sites pooled together), "s" calculates the diversity of each site separately.
scales	Scale parameter values as in function <a href="#">renyi</a> ( <b>vegan</b> ).
evenness	Calculate or plot the evenness profile.
xr	Result from <a href="#">renyi</a> or <code>renyiresult</code> .
addit	Add diversity profile to an existing graph.
pch	Symbol used for drawing the diversity profiles (as in function <a href="#">points</a> ).
ylim	Limits of the vertical axis.
labelit	Provide site labels (site names) at beginning and end of the diversity profiles.
legend	Add the legend (you need to click in the graph where the legend needs to be plotted).
col	Colour for the diversity profile (as in function <a href="#">points</a> ).
cex	Character expansion factor (as in function <a href="#">points</a> ).
rainbow	Use rainbow colours for the diversity profiles.
sites	Maximum number of sites to provide profile values.
permutations	Number of permutations for the Monte-Carlo simulations for accumulated renyi diversity profiles (estimated by <a href="#">renyiaccum</a> ).
plotit	Plot the results (you need to click in the graph where the legend should be plotted).
...	Other arguments to be passed to functions <a href="#">renyi</a> or <a href="#">plot</a> .

### Details

These functions provide some alternative methods of obtaining results with diversity profiles, although function [renyi](#) is always used to calculate the diversity profiles.

The method of calculating the diversity profiles: "all" calculates the diversity profile of the entire community (all sites pooled together), whereas "s" calculates the diversity profile of each site separately. The evenness profile is calculated by subtracting the profile value at scale 0 from all the profile values.

Functions `renyiresult`, `renyiaccumresult` and `renyicomp` allow to calculate diversity profiles for subsets of the community and environmental data sets. functions `renyiresult` and `renyiaccumresult`

calculate the diversity profiles for the specified level of a selected environmental variable. Method `renyicomp` calculates the diversity profile for all levels of a selected environmental variable separately.

Functions `renyicomp` and `renyiaccumresult` calculate accumulation curves for the Renyi diversity profile by randomised pooling of sites and calculating diversity profiles for the pooled sites as implemented in `renyiaccum`. The method is similar to the random method of species accumulation (`specaccum`). If the number of "sites" is not changed from the default, it is replaced by the sample size of the level with the fewest number of sites.

### Value

The functions provide alternative methods of obtaining Renyi diversity profiles.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

### References

Kindt R., Degrande A., Turyomurugendo L., Mbosso C., Van Damme P., Simons A.J. (2001). Comparing species richness and evenness contributions to on-farm tree diversity for data sets with varying sample sizes from Kenya, Uganda, Cameroon and Nigeria with randomised diversity profiles. Paper presented at IUFRO conference on forest biometry, modeling and information science, 26-29 June, University of Greenwich, UK

Kindt R. (2002). Methodology for tree species diversification planning for African agroecosystems. Thesis submitted in fulfilment of the requirement of the degree of doctor (PhD) in applied biological sciences. Faculty of agricultural and applied biological sciences, Ghent University, Ghent (Belgium), 332+xi pp.

Kindt R., Van Damme P. & Simons A.J. (2006). Tree diversity in western Kenya: using diversity profiles to characterise richness and evenness. *Biodiversity and Conservation* 15: 1253-1270.

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

### Examples

```
library(vegan)
data(dune.env)
data(dune)
Renyi.1 <- renyiresult(dune, y=dune.env, factor='Management', level='NM',
  method='s')
Renyi.1
renyiplot(Renyi.1, evenness=FALSE, addit=FALSE, pch=1,col='1', cex=1,
  legend=FALSE)
## CLICK IN THE GRAPH TO INDICATE WHERE THE LEGEND NEEDS TO BE PLACED
## IN CASE THAT YOU OPT FOR LEGEND=TRUE
```

---

residualssurface      *Show and Interpolate Two Dimensional Distribution of Residuals*

---

### Description

This function interpolates the spatial structure of residuals of a GLM through [gam](#) or [surf.ls](#) and optionally provides a graph.

### Usage

```
residualssurface(model, data, x, y, gam = F, npol = 2, plotit = T, filled = F, bubble = F)
```

### Arguments

model	Result of GLM as calculated by <a href="#">glm</a> or <a href="#">glm.nb</a> .
data	Data set that contains the spatial coordinates of the sample units used for the original model (specified as "x" and "y").
x	Horizontal position of the sample units.
y	Vertical position of the sample units.
gam	Interpolate the spatial structure by <a href="#">gam</a> (if "TRUE") or by <a href="#">surf.ls</a> (if "FALSE").
npol	Degree of polynomial surface as passed to <a href="#">surf.ls</a> .
plotit	Plot the interpolated surface (through <a href="#">interp</a> and the residuals).
filled	Fill the contours by <a href="#">filled.contour</a> .
bubble	Provide a bubble graph of the residuals: circles indicate positive residuals, whereas squares indicate negative residuals.

### Details

The function reports the results of a GAM or least-squares trend surface analysis of the spatial distribution of residuals of a model (through [residuals](#)).

Optionally, a graph is produced that can contain the trend surface, filled contours and bubble graphs in addition to the spatial location of the sample units.

### Value

The function reports the results of a GAM or least-squares trend surface analysis of the spatial distribution of residuals. Optionally, a graph is provided.

### Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(vegan)
library(mgcv)
library(akima)
data(faramea)
Count.model1 <- lm(Faramea.occidentalis ~ Precipitation,
  data=faramea, na.action=na.exclude)
surface.1 <- residualssurface(Count.model1, na.omit(faramea),
  'UTM.EW', 'UTM.NS', gam=TRUE, plotit=TRUE, bubble=TRUE)
```

---

 spatialsample

*Spatial Sampling within a Polygon*


---

## Description

Spatial sampling within a polygon provides several methods of selecting rectangular sample plots within a polygon. Using a GIS package may be preferred for actual survey design.

## Usage

```
spatialsample(x,method="random",n=5,xwidth=0.5,ywidth=0.5,xleft=0,
  ylower=0,xdist=0,ydist=0,plotit=T,plothull=F)
```

## Arguments

x	2-column matrix with the coordinates of the vertices of the polygon. The first column contains the horizontal (x) position, the second column contains the vertical (y) position.
method	Method of sampling, any of "random", "grid" or "random grid".
n	Number of sample plots to be selected, or number of horizontal and vertical grid positions.
xwidth	Horizontal width of the sample plots.
ywidth	Vertical width of the sample plots.
xleft	Horizontal starting position of the grid.
ylower	Vertical starting position of the grid.
xdist	Horizontal distance between grid locations.
ydist	Vertical distance between grid locations.
plotit	Plot the sample plots on the current graph.
plothull	Plot a convex hull around the sample plots.

## Details

Spatial sampling within a polygon provides several methods of selecting the position of sample plots.

Method "random" selects random positions of the sample plots using simple random sampling.

Method "grid" selects sample plots from a grid defined by "xleft", "ylower", "xdist" and "ydist". In case `xdist=0` or `ydist=0`, then the number of grid positions are defined by "n". In case "xleft" or "ylower" are below the minimum position of any vertex of the polygon, then a random starting position is selected for the grid.

Method "random grid" selects sample plots at random from the sampling grid using the same methods of defining the grid as for method "grid".

## Value

The function returns a list of centres of rectangular sample plots.

## Author(s)

Roeland Kindt (World Agroforestry Centre)

## References

Kindt, R. & Coe, R. (2005) Tree diversity analysis: A manual and software for common statistical methods for ecological and biodiversity studies.

<http://www.worldagroforestry.org/resources/databases/tree-diversity-analysis>

## Examples

```
library(splancs)
area <- array(c(10,10,15,35,40,35,5,35,35,30,30,10), dim=c(6,2))
landuse1 <- array(c(10,10,15,15,30,35,35,30), dim=c(4,2))
landuse2 <- array(c(10,10,15,15,35,30,10,30,30,35,30,15), dim=c(6,2))
landuse3 <- array(c(10,10,30,35,40,35,5,10,15,30,30,10), dim=c(6,2))
plot(area[,1], area[,2], type="n", xlab="horizontal position",
      ylab="vertical position", lwd=2, bty="l")
polygon(landuse1)
polygon(landuse2)
polygon(landuse3)
spatialsample(area, method="random", n=20, xwidth=1, ywidth=1, plotit=TRUE,
              plothull=FALSE)
spatialsample(area, method="grid", xwidth=1, ywidth=1, plotit=TRUE, xleft=12,
              ylower=7, xdist=4, ydist=4)
spatialsample(area, method="random grid", n=20, xwidth=1, ywidth=1,
              plotit=TRUE, xleft=12, ylower=7, xdist=4, ydist=4)
```

---

transfgradient	<i>Gradient for Hypothetical Example of Turnover of Species Composition</i>
----------------	---

---

### Description

This dataset documents the site sequence of 19 sites on a gradient determined from unimodal species distributions. The dataset is accompanied by [transfspecies](#) that documents the species composition of the sites. This is a hypothetical example that allows to investigate how well ecological distance measures or ordination methods recover the expected best sequence of sites.

### Usage

```
data(transfgradient)
```

### Format

A data frame with 19 observations on the following variable.

`gradient` a numeric vector

### Source

Legendre, P. & Gallagher, E.D. (2001) Ecologically meaningful transformations for ordination of species data. *Oecologia* 129: 271-280.

### References

Figure 3a.

### Examples

```
data(transfspecies)
data(transfgradient)
plot(transfspecies[,1]~transfgradient[,1],xlab="gradient",
      ylab="species abundance",type="n",ylim=c(0.5,8.5))
for (i in 1:9) {points(transfgradient[,1],transfspecies[,i],type="o",pch=i)}
```

---

transfspecies	<i>Hypothetical Example of Turnover of Species Composition</i>
---------------	--

---

### Description

This dataset documents the species composition of 19 sites that follow a specific sequence of sites as determined from unimodal species distributions. The dataset is accompanied by [transfgradient](#) that documents the gradient in species turnover. This is a hypothetical example that allows to investigate how well ecological distance measures or ordination methods recover the expected best sequence of sites.

**Usage**

```
data(transfspecies)
```

**Format**

A data frame with 19 observations on the following 9 variables.

species1 a numeric vector

species2 a numeric vector

species3 a numeric vector

species4 a numeric vector

species5 a numeric vector

species6 a numeric vector

species7 a numeric vector

species8 a numeric vector

species9 a numeric vector

**Details**

The example in the Tree Diversity Analysis manual only looks at the ecological distance from the first site. Hence, only the first 10 sites that share some species with this site should be selected.

This dataset enables investigations of how well ecological distance measures and ordination diagrams reconstruct the gradient (sequence of sites). The gradient expresses how the sites would be arranged based on their species composition.

**Source**

Legendre, P. & Gallagher, E.D. (2001) Ecologically meaningful transformations for ordination of species data. *Oecologia* 129: 271-280.

**References**

Figure 3a.

**Examples**

```
data(transfspecies)
data(transfgradient)
plot(transfspecies[,1]~transfgradient[,1],xlab="gradient",
      ylab="species abundance",type="n",ylim=c(0.5,8.5))
for (i in 1:9) {points(transfgradient[,1],transfspecies[,i],type="o",pch=i)}
```

---

`warcom`*Warburgia ugandensis* AFLP Scores

---

**Description**

This data set contains scores for 185 loci for 100 individuals of the *Warburgia ugandensis* tree species (a medicinal tree species native to Eastern Africa). Since the data set is a subset of a larger data set that originated from a study of several *Warburgia* species, some of the loci did not produce bands for *W. ugandensis* (i.e. some loci only contain zeroes). This data set is accompanied by `warenv` that describes population and regional structure of the 100 individuals.

**Usage**

```
data(warcom)
```

**Format**

A data frame with 100 observations on the following 185 variables.

```
locus001 a numeric vector  
locus002 a numeric vector  
locus003 a numeric vector  
locus004 a numeric vector  
locus005 a numeric vector  
locus006 a numeric vector  
locus007 a numeric vector  
locus008 a numeric vector  
locus009 a numeric vector  
locus010 a numeric vector  
locus011 a numeric vector  
locus012 a numeric vector  
locus013 a numeric vector  
locus014 a numeric vector  
locus015 a numeric vector  
locus016 a numeric vector  
locus017 a numeric vector  
locus018 a numeric vector  
locus019 a numeric vector  
locus020 a numeric vector  
locus021 a numeric vector  
locus022 a numeric vector
```

locus023 a numeric vector  
locus024 a numeric vector  
locus025 a numeric vector  
locus026 a numeric vector  
locus027 a numeric vector  
locus028 a numeric vector  
locus029 a numeric vector  
locus030 a numeric vector  
locus031 a numeric vector  
locus032 a numeric vector  
locus033 a numeric vector  
locus034 a numeric vector  
locus035 a numeric vector  
locus036 a numeric vector  
locus037 a numeric vector  
locus038 a numeric vector  
locus039 a numeric vector  
locus040 a numeric vector  
locus041 a numeric vector  
locus042 a numeric vector  
locus043 a numeric vector  
locus044 a numeric vector  
locus045 a numeric vector  
locus046 a numeric vector  
locus047 a numeric vector  
locus048 a numeric vector  
locus049 a numeric vector  
locus050 a numeric vector  
locus051 a numeric vector  
locus052 a numeric vector  
locus053 a numeric vector  
locus054 a numeric vector  
locus055 a numeric vector  
locus056 a numeric vector  
locus057 a numeric vector  
locus058 a numeric vector  
locus059 a numeric vector

locus060 a numeric vector  
locus061 a numeric vector  
locus062 a numeric vector  
locus063 a numeric vector  
locus064 a numeric vector  
locus065 a numeric vector  
locus066 a numeric vector  
locus067 a numeric vector  
locus068 a numeric vector  
locus069 a numeric vector  
locus070 a numeric vector  
locus071 a numeric vector  
locus072 a numeric vector  
locus073 a numeric vector  
locus074 a numeric vector  
locus075 a numeric vector  
locus076 a numeric vector  
locus077 a numeric vector  
locus078 a numeric vector  
locus079 a numeric vector  
locus080 a numeric vector  
locus081 a numeric vector  
locus082 a numeric vector  
locus083 a numeric vector  
locus084 a numeric vector  
locus085 a numeric vector  
locus086 a numeric vector  
locus087 a numeric vector  
locus088 a numeric vector  
locus089 a numeric vector  
locus090 a numeric vector  
locus091 a numeric vector  
locus092 a numeric vector  
locus093 a numeric vector  
locus094 a numeric vector  
locus095 a numeric vector  
locus096 a numeric vector

locus097 a numeric vector  
locus098 a numeric vector  
locus099 a numeric vector  
locus100 a numeric vector  
locus101 a numeric vector  
locus102 a numeric vector  
locus103 a numeric vector  
locus104 a numeric vector  
locus105 a numeric vector  
locus106 a numeric vector  
locus107 a numeric vector  
locus108 a numeric vector  
locus109 a numeric vector  
locus110 a numeric vector  
locus111 a numeric vector  
locus112 a numeric vector  
locus113 a numeric vector  
locus114 a numeric vector  
locus115 a numeric vector  
locus116 a numeric vector  
locus117 a numeric vector  
locus118 a numeric vector  
locus119 a numeric vector  
locus120 a numeric vector  
locus121 a numeric vector  
locus122 a numeric vector  
locus123 a numeric vector  
locus124 a numeric vector  
locus125 a numeric vector  
locus126 a numeric vector  
locus127 a numeric vector  
locus128 a numeric vector  
locus129 a numeric vector  
locus130 a numeric vector  
locus131 a numeric vector  
locus132 a numeric vector  
locus133 a numeric vector

locus134 a numeric vector  
locus135 a numeric vector  
locus136 a numeric vector  
locus137 a numeric vector  
locus138 a numeric vector  
locus139 a numeric vector  
locus140 a numeric vector  
locus141 a numeric vector  
locus142 a numeric vector  
locus143 a numeric vector  
locus144 a numeric vector  
locus145 a numeric vector  
locus146 a numeric vector  
locus147 a numeric vector  
locus148 a numeric vector  
locus149 a numeric vector  
locus150 a numeric vector  
locus151 a numeric vector  
locus152 a numeric vector  
locus153 a numeric vector  
locus154 a numeric vector  
locus155 a numeric vector  
locus156 a numeric vector  
locus157 a numeric vector  
locus158 a numeric vector  
locus159 a numeric vector  
locus160 a numeric vector  
locus161 a numeric vector  
locus162 a numeric vector  
locus163 a numeric vector  
locus164 a numeric vector  
locus165 a numeric vector  
locus166 a numeric vector  
locus167 a numeric vector  
locus168 a numeric vector  
locus169 a numeric vector  
locus170 a numeric vector

locus171 a numeric vector  
locus172 a numeric vector  
locus173 a numeric vector  
locus174 a numeric vector  
locus175 a numeric vector  
locus176 a numeric vector  
locus177 a numeric vector  
locus178 a numeric vector  
locus179 a numeric vector  
locus180 a numeric vector  
locus181 a numeric vector  
locus182 a numeric vector  
locus183 a numeric vector  
locus184 a numeric vector  
locus185 a numeric vector

### Source

Muchugi, A.N. (2007) Population genetics and taxonomy of important medicinal tree species of the genus *Warburgia*. PhD Thesis. Kenyatta University, Kenya.

### Examples

```
data(warcom)
```

---

warenv

*Warburgia ugandensis* Population Structure

---

### Description

This data set contains population and regional locations for 100 individuals of the *Warburgia ugandensis* tree species (a medicinal tree species native to Eastern Africa). This data set is associated with `warcom` that contains scores for 185 AFLP loci.

### Usage

```
data(warenv)
```

### Format

A data frame with 100 observations on the following 4 variables.

`population` a factor with levels Kibale Kitale Laikipia Lushoto Mara

`popshort` a factor with levels KKIT KLAI KMAR TLUS UKIB

`country` a factor with levels Kenya Tanzania Uganda

`rift.valley` a factor with levels east west

**Source**

Muchugi, A.N. (2007) Population genetics and taxonomy of important medicinal tree species of the genus Warburgia. PhD Thesis. Kenyatta University, Kenya.

**Examples**

```
data(warenv)
```

# Index

## \*Topic **datasets**

- BCI.env, [9](#)
- faramea, [47](#)
- transfgradient, [73](#)
- transfspecies, [73](#)
- warcom, [75](#)
- warenv, [80](#)

## \*Topic **multivariate**

- accumresult, [4](#)
- add.spec.scores, [6](#)
- balanced.specaccum, [7](#)
- BiodiversityRGUI, [10](#)
- CAPdiscrim, [34](#)
- caprescale, [36](#)
- crosstabanalysis, [38](#)
- deviancepercentage, [39](#)
- dist.eval, [40](#)
- dist.zeroes, [41](#)
- distdisplayed, [42](#)
- disttransform, [44](#)
- diversityresult, [45](#)
- import.from.Excel, [48](#)
- loaded.citations, [50](#)
- makecommunitydataset, [50](#)
- multiconstrained, [52](#)
- nested.anova.dbrda, [53](#)
- NMSrandom, [55](#)
- nnetrandom, [56](#)
- ordicoeno, [58](#)
- ordisymbol, [59](#)
- PCAsignificance, [61](#)
- radfitresult, [62](#)
- rankabundance, [63](#)
- removeNAcomm, [65](#)
- renyiresult, [67](#)
- residualssurface, [70](#)
- spatialsample, [71](#)

## \*Topic **package**

- BiodiversityR-package, [3](#)

- .packages, [50](#)

- accumcomp, [16](#), [17](#)
- accumcomp (accumresult), [4](#)
- accumplot, [16](#), [17](#)
- accumplot (accumresult), [4](#)
- accumresult, [4](#), [8](#), [16](#), [17](#), [66](#)
- add.spec.scores, [6](#), [24](#), [35](#)
- adonis, [54](#)
- agnes, [31](#), [32](#)
- anosim, [33](#), [34](#)
- Anova, [20](#), [22](#)
- anova.cca, [27](#), [28](#), [52](#), [53](#)
- anova.gam, [20](#), [22](#)
- anova.glm, [20](#), [22](#), [39](#)
- anova.lm, [20](#)
- anova.negbin, [39](#)
- arrows, [59](#), [61](#)
- as.dist, [10](#), [24](#), [28](#), [31](#), [33](#)
- av.plots, [21](#), [23](#)

- balanced.specaccum, [7](#)
- BCI, [9](#), [48](#)
- BCI.env, [9](#), [48](#)
- BiodiversityR (BiodiversityR-package), [3](#)
- BiodiversityR-package, [3](#)
- BiodiversityRGUI, [3](#), [10](#)
- box.cox.powers, [16](#)
- boxplot, [16](#)

- CAPdiscrim, [27](#), [28](#), [34](#)
- caprescale, [36](#)
- capscale, [27](#), [28](#), [34](#), [36](#), [37](#), [52–54](#)
- cascadeKM, [31](#), [32](#)
- cca, [24](#), [27](#), [28](#), [52](#), [53](#)
- check.datasets, [49](#)
- check.datasets (removeNAcomm), [65](#)
- check.ordiscores (removeNAcomm), [65](#)
- chisq.test, [21](#), [38](#)
- citation, [50](#)

- clara, 31
- clip.clust, 32
- cmdscale, 6, 7, 24, 35, 36
- cophenetic, 31, 32
- cor, 6, 43
- cr.plots, 21, 23
- crosstabanalysis, 38
- cutree, 31, 32
  
- daisy, 33, 34
- decorana, 24
- decostand, 44
- deviancepercentage, 39
- diana, 31, 32
- dist.eval, 40
- dist.zeros, 41
- distconnected, 40
- distdisplayed, 24, 26, 28, 30, 42
- disttransform, 15, 44
- diversity, 45, 46
- diversitycomp, 17
- diversitycomp (diversityresult), 45
- diversityresult, 17, 45, 66
- drop1, 20, 22
  
- effect, 20, 22
- envfit, 25, 28, 29
- estimateR, 46
  
- factor, 34
- fanny, 31
- famea, 47
- filled.contour, 70
- fisher.alpha, 45, 46
- fisherfit, 62, 63
  
- gam, 20–22, 43, 58, 70
- gam.check, 20, 22
- glht, 21, 23
- glm, 19–21, 39, 70
- glm.nb, 20, 39, 70
- glmmPQL, 20
  
- hclust, 31, 32
  
- identify.ordiplot, 25, 28
- import.from.Access, 13, 14
- import.from.Access (import.from.Excel), 48
- import.from.Excel, 12, 13, 48
  
- influence.plot, 21, 23
- initMDS, 56
- interp, 70
- isoMDS, 6, 55, 56
  
- kgs, 32
- kmeans, 31
- ks.test, 16
  
- lda, 35
- levene.test, 20, 22
- lines.spantree, 24, 25, 28, 29, 59, 60
- lm, 19
- loaded.citations, 50
  
- make.cepnames, 49
- makecommunitydataset, 14, 48, 49, 50, 66
- mantel, 31, 33, 34, 43
- metaMDS, 6, 24, 26, 55, 56
- mrpp, 33, 34
- multiconstrained, 27, 28, 52
  
- na.omit, 39
- nested.anova.dbrda, 53
- nested.npmanova (nested.anova.dbrda), 53
- NMStest, 6, 24, 55
- nnet, 56, 57
- nnetrandom, 22, 56
  
- odbcConnectAccess, 48
- odbcConnectExcel, 48, 49
- ordiarrows, 25, 29, 59
- ordibubble, 26, 29
- ordibubble (ordisymbol), 59
- ordicluster, 24, 25, 28, 29, 59, 60
- ordicluster2, 24, 26, 28, 30
- ordicluster2 (ordisymbol), 59
- ordicoeno, 26, 30, 58
- ordiellipse, 25, 29, 59
- ordiequilibriumcircle, 26
- ordiequilibriumcircle (PCAsignificance), 61
- ordigrd, 59
- ordihull, 25, 29, 59
- ordinearest, 24, 26, 28, 30
- ordinearest (ordisymbol), 59
- ordiplot, 24–26, 28–30, 35–37, 42, 58, 59, 61
- ordisegments, 25, 29, 59
- ordispider, 25, 29, 59

- ordisurf, 25, 29
- ordisymbol, 26, 30, 59
- ordivector, 26, 30
- ordivector (ordisymbol), 59
- pairs, 16
- pam, 31
- par, 59
- PCAsignificance, 61
- permutest.cca, 28
- persp.renyiaccum, 19
- plot, 5, 16, 22, 24, 28, 58, 64, 68
- plot.agnes, 32
- plot.cascadeKM, 32
- plot.cca, 24, 26, 28, 30
- plot.diana, 32
- plot.gam, 20, 22
- plot.hclust, 32
- plot.lm, 20, 22
- plot.prc, 28
- plot.radfit, 62, 63
- plot.rpart, 21, 23
- plot.specaccum, 4, 5, 7
- points, 5, 17, 58, 59, 64, 68
- points.ordiplot, 25, 29
- postMDS, 6
- prc, 27, 28
- prcomp, 24
- predict, 20–22
- predict.rpart, 20
- prestonfit, 62, 63
- qq.plot, 16, 20, 22
- radfit, 62, 63
- radfitresult, 18, 62
- rankabuncomp, 18
- rankabuncomp (rankabundance), 63
- rankabundance, 18, 63
- rankabunplot, 18
- rankabunplot (rankabundance), 63
- rankindex, 33
- rda, 6, 7, 24, 26–28, 30, 52, 53, 61
- rect.hclust, 32
- removeNAcomm, 15, 65
- removeNAenv, 15
- removeNAenv (removeNAcomm), 65
- removezerospecies (removeNAcomm), 65
- renyi, 67, 68
- renyiaccum, 68, 69
- renyiaccumresult, 19
- renyiaccumresult (renyiresult), 67
- renyicomp, 19
- renyicomp (renyiresult), 67
- renyiplot, 19
- renyiplot (renyiresult), 67
- renyiresult, 19, 66, 67
- residuals, 70
- residuals.rpart, 20
- residualssurface, 70
- rpart, 20, 22
- same.sites, 14, 49
- same.sites (removeNAcomm), 65
- sammon, 6, 55, 56
- scale, 20, 22
- scores, 24, 26, 28, 30, 66
- screeplot.cca, 26
- shapiro.test, 16
- spatialsample, 71
- specaccum, 4, 5, 7, 8, 69
- specnumber, 45, 46
- specpool, 45, 46
- sqlFetch, 49
- stressplot, 26
- subsetcomm (removeNAcomm), 65
- summary, 16
- summary.agnes, 31
- summary.anosim, 33
- summary.cca, 24, 28
- summary.clara, 31
- summary.decorana, 24
- summary.diana, 31
- summary.fanny, 31
- summary.gam, 20, 22
- summary.glm, 20, 22
- summary.lm, 20
- summary.nnet, 22
- summary.pam, 31
- summary.prc, 28
- summary.rpart, 22
- surf.ls, 70
- symbols, 59
- termpart, 20, 22
- text.ordiplot, 25, 29
- text.rpart, 21, 23
- transfgradient, 73, 73

transspecies, [73](#), [73](#)

vegdist, [23](#), [24](#), [31](#), [33](#), [35](#), [40–42](#), [52](#), [54](#), [56](#)

warcom, [75](#)

warenv, [80](#)

wascores, [6](#)