Package ‘adegraphics’

December 18, 2018

Type Package

Title An S4 Lattice-Based Package for the Representation of Multivariate Data

Version 1.0-15

Date 2018-12-18

Author Stéphane Dray <stephane.dray@univ-lyon1.fr> and Aurélie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>, with contributions from Jean Thioulouse. Based on earlier work by Alice Julien-Laferrière.

Maintainer Aurélie Siberchicot <aurelie.siberchicot@univ-lyon1.fr>

Description Graphical functionalities for the representation of multivariate data. It is a complete re-implementation of the functions available in the 'ade4' package.

Depends R (>= 3.0.2)

License GPL (>= 2)

Imports ade4 (>= 1.7-13), graphics, grid, KernSmooth, lattice, latticeExtra, methods, RColorBrewer, sp (>= 1.1-1), stats

Suggests Guerry, knitr, maptools, pixmap, rmarkdown, spdep, splancs


BugReports https://github.com/sdray/adegraphics/issues
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>prepare-methods</td>
</tr>
<tr>
<td>s.arrow</td>
</tr>
<tr>
<td>s.class</td>
</tr>
<tr>
<td>s.corcircle</td>
</tr>
<tr>
<td>s.density</td>
</tr>
<tr>
<td>s.distri</td>
</tr>
<tr>
<td>s.image</td>
</tr>
<tr>
<td>s.label</td>
</tr>
<tr>
<td>s.logo</td>
</tr>
<tr>
<td>s.match</td>
</tr>
<tr>
<td>s.Spatial</td>
</tr>
<tr>
<td>s.traject</td>
</tr>
<tr>
<td>s.value</td>
</tr>
<tr>
<td>S1.boxplot-class</td>
</tr>
<tr>
<td>S1.class-class</td>
</tr>
<tr>
<td>S1.distri-class</td>
</tr>
<tr>
<td>S1.label-class</td>
</tr>
<tr>
<td>S1.match-class</td>
</tr>
<tr>
<td>s1d.barchart</td>
</tr>
<tr>
<td>s1d.boxplot</td>
</tr>
<tr>
<td>s1d.class</td>
</tr>
<tr>
<td>s1d.curve</td>
</tr>
<tr>
<td>s1d.curves</td>
</tr>
<tr>
<td>s1d.density</td>
</tr>
<tr>
<td>s1d.distri</td>
</tr>
<tr>
<td>s1d.dotplot</td>
</tr>
<tr>
<td>s1d.gauss</td>
</tr>
<tr>
<td>s1d.hist</td>
</tr>
<tr>
<td>s1d.interval</td>
</tr>
<tr>
<td>s1d.label</td>
</tr>
<tr>
<td>s1d.match</td>
</tr>
<tr>
<td>S2.arrow-class</td>
</tr>
<tr>
<td>S2.class-class</td>
</tr>
<tr>
<td>S2.corcircle-class</td>
</tr>
<tr>
<td>S2.density-class</td>
</tr>
<tr>
<td>S2.distri-class</td>
</tr>
<tr>
<td>S2.image-class</td>
</tr>
<tr>
<td>S2.label-class</td>
</tr>
<tr>
<td>S2.logo-class</td>
</tr>
<tr>
<td>S2.match-class</td>
</tr>
<tr>
<td>S2.traject-class</td>
</tr>
<tr>
<td>S2.value-class</td>
</tr>
<tr>
<td>setlimits1D</td>
</tr>
<tr>
<td>sortparamADEg</td>
</tr>
<tr>
<td>superpose</td>
</tr>
<tr>
<td>T.cont-class</td>
</tr>
<tr>
<td>T.image-class</td>
</tr>
<tr>
<td>T.value-class</td>
</tr>
</tbody>
</table>
Description

This package was created to replace graphics functionalities of the ade4 package and to offer customizable representations of data and result analysis.

Graphics are objects of S4 class, which can be displayed but also stored for latter modifications. Those modifications can be graphical changes, but also superposition or juxtaposition of various graphical objects (creating an other type of object). Each object will contain graphical parameters and instructions for the display (calls, positions, etc.) and the data set used. Sometimes data is heavy, due to its size for example. Two storing systems exist:

- full storage: data is assigned to an object's slot.
- names and position: data names (as a string, obtained using deparse(substitute)) and their frame position (using sys.nframe()) are stored. Then the full data can be retrieve with those two informations (and only if the data objects are still in the environment)

This new system is based on the lattice package and grid graphics.

Details

Package: adegraphics
Type: Package
Version: 1.0-12
Date: 2018-08-31
License: GPL (>=2)
Depends: ade4 (>= 1.7-13), graphics, grid, KernSmooth, lattice, latticeExtra, methods, RColorBrewer, sp (>= 1.1-1), stats

A lot of classes were implemented. Two superclass structures the architecture in class. Simple and complex graphics are distinguished in the former version:
**add.ADEg**

- ADEg class provides simple graphics using one kind of data (most of the time, only a data frame) and one representation method (points, labels, arrows...)
- ADEgS class provides complex graphics making juxtaposition, superposition and/or insertion of several simple graphics.

5 subclasses inherits from the superclass abstract ADEg:

- ADEg.S1: one-dimensional plot
- ADEg.S2: bi-dimensional plot
- ADEg.C1: one-dimensional data plotted in 2-D
- ADEg.T: table plot
- ADEg.Tr: triangle plot

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

**References**


**See Also**

lattice ADEg ADEgS

**Examples**

```r
showClass("ADEg")
showClass("ADEgS")
```

---

**Description**

Adds an ADEg to the current ADEg or ADEgS plot.

**Usage**

`add.ADEg(object)`

**Arguments**

- **object** an ADEg object
Details

This function uses the last plotted ADEg or ADEgS object. It calls superpose.

Value

an ADEgS object

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

superpose ADEg ADEgS

Examples

df1 <- cbind(rnorm(24), rnorm(24))
df2 <- cbind(rnorm(24), rnorm(24))
g1 <- s.label(df1, ppoints.col = "blue")
g2 <- s.label(df2, ppoints.col = "red", plot = FALSE)
add.ADEg(g2)

data(jv73, package = "ade4")
pca1 <- ade4::dudi.pca(jv73$morpho, scannf = FALSE)
g5 <- s.label(pca1$li, plabels.optim = TRUE)
g6 <- s.class(pca1$li, jv73$fac.riv, starSize = 0, ellipseSize = 0, chullSize = 1,
        ppolygons.alpha = 0.4, col = rainbow(12), ppoints.cex = 0, plot = FALSE)
add.ADEg(g6)

addhist

Adds histograms and density lines against a bi-dimensional graphics.

Description

Adds the two marginal histograms and density lines of each axis against an ADEg.S2 object.

Usage

addhist(object, bandwidth, gridsize = 60, kernel = "normal", cbreaks = 2,
        storeData = TRUE, plot = TRUE, pos = -1, ...)
Arguments

object  
an ADEg.S2 object

bandwidth  
used for the calculations of the density lines (see the bkde function of the KernSmooth package).

gridsize  
used for the calculations of the density lines (see the bkde function of the KernSmooth package).

kernel  
used for the calculations of the density lines (see the bkde function of the KernSmooth package).

cbreaks  
number of cells for the histograms per interval of the grid of the bi-dimensional graphics.

plot  
a logical indicating if the graphics is displayed

storeData  
a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored

pos  
an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

Additional graphical parameters (see adegpar and trellis.par.get)

Details

Density is calculated using the function bkde of the KernSmooth package.

Value

An ADEgS object, a list of four graphical objects, one ADEg.S2 and three trellis (from lattice). Their names are:

object  
the ADEg.S2 object

densX  
top histogram, a trellis object

densY  
right histogram, a trellis object

link  
corner graphics linking the two histograms, a trellis object

Note

Into the dots arguments, the usual parameters for the s.label can be given with the object key.

Trellis parameters are used for the three remaining graphics. plot.polygon handles the histogram aspect, add.line the graduations lines and plot.line the density lines.

Finally, for the link graphic, labels aspect can be changed using a plot.labels list, as for an S2.label object.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
See Also

ADEg.S2 ADEgS

Examples

data(rpjdl, package = "ade4")
coa1 <- ade4::dudi.coa(rpjdl$fau, scannf = FALSE, nf = 4)
labli <- s.label(coa1$li)
g1 <- addhist(labli)
g2 <- addhist(labli, plabels.cex = 0, cbreaks = 3)
lablo <- s.label(coa1$co)
g3 <- addhist(labco, plabels.cex = 0, cbreaks = 3)
update(g3, pbackground.col = "grey85")

addline

Adds lines on graphics.

Description

Adds a trellis object containing one or several lines on one or several graphical objects.

Usage

addline(object, a = NULL, b = NULL, h = NULL, v = NULL, plot = TRUE, ...)

Arguments

object an object of class ADEg or ADEgS

a, b coefficients of the line to be added, passed to the panel.abline function of the lattice package

h, v numeric vectors giving locations respectively of horizontal and vertical lines to be added to the plot, in native coordinates, passed to the panel.abline function of the lattice package

plot a logical indicating if the graphics is displayed

... Other arguments. Additional graphical parameters (see the plines list in adegpar and trellis.par.get). If object is an ADEgS, the argument which identify which ADEg is/are used for superposition.

Value

An object of class ADEgS.

Author(s)

Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
addpoint

See Also

ADEg ADEgS panel.abline

Examples

# example extracted from the pedagogic file, here: http://pbil.univ-lyon1.fr/R/pdf/tdr65.pdf
data(monde84, package = "ade4")
dfx <- cbind.data.frame(lpib = log(monde84$pib), croipop = monde84$croipop)
dfy <- cbind.data.frame(lmorta = log(monde84$morta), lanal = log(monde84$anal + 1),
rscol = sqrt(100 - monde84$scol))
dfx0 <- ade4::scalewt(dfx)
dfy0 <- ade4::scalewt(dfy)
can1 <- cancor(dfx0, dfy0)
varcanoX <- dfx0
varcanoY <- dfy0
g1 <- s.label(cbind(varcanoY, varcanoX), labels = row.names(monde84), plabel.cex = 0.8, plot = FALSE)
addline(g1, 0, 1, plines.col = "red", plines.lwd = 0.5, plines.lty = 2)

Description

Adds a trellis object containing one or several points on one or several graphical objects.

Usage

addpoint(object, xcoord, ycoord, plot = TRUE, ...)

Arguments

object an object of class ADEg or ADEgS
xcoord an integer (or a vector) indicating where label is(are) plotted on the x-axis,
passed to the panel.points function of the lattice package
ycoord an integer (or a vector) indicating where label is(are) plotted on the y-axis,
passed to the panel.points function of the lattice package
plot a logical indicating if the graphics is displayed
... Other arguments. Additional graphical parameters (see the ppoints list in adegpar
and trellis.par.get). If object is an ADEgS, the argument which identify
which ADEg is/are used for superposition.

Value

An object of class "ADEgS".

Author(s)

Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
addsegment

Adds segments on graphics.

**Description**

Adds a trellis object containing one or several segments on one or several graphical objects.

**Usage**

```r
addsegment(object, x0 = NULL, y0 = NULL, x1, y1, plot = TRUE, ...)```

**Arguments**

- `object` an object of class ADEg or ADEgS
- `x0, y0` coordinates of points FROM which to draw, passed to the `panel.segments` function of the lattice package. See Details.
- `x1, y1` coordinates of points TO which to draw, passed to the `panel.segments` function of the lattice package. See Details.
- `plot` a logical indicating if the graphics is displayed
- `...` Other arguments. Additional graphical parameters (see the `plines` list in `adegpar` and `trellis.par.get`). If object is an ADEgS, the argument which identify which ADEg is/are used for superposition.

**Details**

`x0, y0, x1` and `y1` can be vectors. A line segment is drawn, for each i, between the point `(x0[i], y0[i])` and the point `(x1[i], y1[i])`. The coordinate vectors will be recycled to the length of the longest.

**Value**

An object of class ADEgS.

**Author(s)**

Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
addtext

Adds labels on graphics.

Description

Adds a trellis object containing one or several labels on one or several graphical objects.

Usage

addtext(object, xcoord, ycoord, label, plot = TRUE, ...)

Arguments

object an object of class ADEG or ADEg
dxcoord an integer (or a vector) indicating where label is(are) plotted on the x-axis, passed to the adeg.panel.label

ycoord an integer (or a vector) indicating where label is(are) plotted on the y-axis, passed to the adeg.panel.label

See Also

ADEg ADEgS panel.segments

data(deug, package = "ade4")
g11 <- sld.density(deug$tab[, 1], plot = FALSE)
g12 <- addsegment(g11, x0 = deug$cent[1], x1 = deug$cent[1], y0 = 0, y1 = 1,
    plines = list(col = "grey30", lwd = 3))
g13 <- addsegment(g11,
    x0 = deug$cent + seq(0, 1, length.out = length(deug$cent)),
    x1 = deug$cent + seq(0, 1, length.out = length(deug$cent)),
    y0 = 0, y1 = 1,
    plines = list(col = 1:length(deug$cent), lty = 1:length(deug$cent)))

# example extracted from the pedagogic file, here: http://pbil.univ-lyon1.fr/R/pdf/tdr65.pdf
data(monde84, package = "ade4")
dfx <- cbind.data.frame(lpib = log(monde84$pib), croipop = monde84$croipop)
dfy <- cbind.data.frame(lmort = log(monde84$morta), lanal = log(monde84$anal + 1),
    rscol = sqrt(100 - monde84$scol))
dfx0 <- ade4::scalewt(dfx)
dfy0 <- ade4::scalewt(dfy)
can1 <- cancort(dfx0, dfy0)
varcanoX <- dfx0
varcanoY <- dfy0
g21 <- s.label(cbind(varcanoY, varcanoX), labels = row.names(monde84), plabel.cex = 0.8,
    plot = FALSE)
g22 <- addsegment(g21, -1.25, -1.25, 1.25, 1.25, plines.col = "purple", plines.lwd = 1.5, plines.lty = 2)
ADEg-class

label

a character string (or a vector) containing the label(s) displayed on object

plot

a logical indicating if the graphics is displayed

... Other arguments. Additional graphical parameters (see the plabels list in adegpar and trellis.par.get). If object is an ADEgS, the argument which identify which ADEg is/are used for superposition.

Value

An object of class ADEgS.

Author(s)

Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEgS adeg.panel.label

Examples

data(dunedata, package = "ade4")
afc1 <- ade4::dudi.coa(dunedata$veg, scannf = FALSE)
g1 <- table.value(dunedata$veg, symbol = "circle", ppoints.cex = 0.5, plot = FALSE)
addtext(g1, 1, 20, "A", plabels.srt = 45, plabels.box.draw = FALSE, plabels.col = "red")

xy <- cbind.data.frame(x = runif(200, -1, 1), y = runif(200, -1, 1))

posi <- factor(xy$x > 0) + factor(xy$y > 0)
g2 <- s.class(xy, fac = posi, facets = posi, pellipses.col = 1:4, plabels.cex = 0,
plegend.drawKey = FALSE, psymb.cex = 0, plot = FALSE)
addtext(g2, c(0.5, 0.5, -0.5, -0.5), c(0.5, -0.5), levels(posi), plabels.cex = 2, plabels.col = 1:4)

Description

An object of ADEg class is a simple graphic. This object can be blended in with another one (superposition, insertion and/or juxtaposition) to form a more complex graphics (an ADEgS object).

The ADEg class is a virtual class, i.e. a class which is not possible to create objects but which have heirs. This class has five son classes: ADEg.S1, ADEg.S2, ADEg.C1, ADEg.T and ADEg.Tr.

Objects from the Class

None object of this class can be instantiated.
ADEg-class

Slots

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list of two elements to create the trellis object:
  • graphictype: the lattice function to use
  • arguments: its parameters to obtain the trellis object

g.args a list containing some parameters linked with the created object of ADEg class:
  • xlim, ylim
  • main, sub
  • xlab, ylab
  • samelimits
  • scales: a list of scales informations (ticks, marks and labels for the x-axis or the y-axis) in the form of the lattice argument scales in the xyplot function

stats a list of internal preliminary calculations

s.misc a list of some other internal parameters

Call an object of class call

Methods

panelbase signature(object = "ADEg"): draws grid and text and produces graphical output from the graphical object

getcall signature(object = "ADEg"): returns the Call slot

getlatticecall signature(object = "ADEg"): returns the lattice.call slot

getstats signature(object = "ADEg"): returns the stats slot

getparameters signature(object = "ADEg", number): if number is 1, returns the trellis.par slot, if it is 2, returns the adeg.par slot and if it is 0, returns the both slots

add.ADEg signature(object = "ADEg"): superposes an ADEg on the current one plotted
+ signature(e1 = "ADEg", e2 = "ADEg"): superposes e2 on e1

superpose signature(g1 = "ADEgOrtrellis", g2 = "ADEgOrtrellis", which = "ANY", plot = "ANY"): creates a new ADEgS object performing a superposition of g2 on g1.

printSuperpose signature(g1 = "ADEgOrtrellis", refg = "ADEgOrtrellis"): internal method, not for users.

cbindADEg signature(g1 = "ADEgOrADEgS", g2 = "ADEgORADEgS"): creates a new "ADEgS" object combining g1 on g2.

rbindADEg signature(g1 = "ADEgORADEgS", g2 = "ADEgORADEgS"): creates a new "ADEgS" object combining g1 on g2 by rows.

insert signature(graphics = "ADEgOrtrellis", oldgraphics = "missing", posi, ratio, inset, plot, which): creates a new ADEgS object performing an insertion of graphics into the current device.

insert signature(graphics = "ADEgOrtrellis", oldgraphics = "ADEg", posi, ratio, inset, plot): creates a new ADEgS object performing an insertion of graphics into oldgraphics.
show signature(x = "ADEg"): prints the ADEg object
plot signature(x = "ADEg"): prints the ADEg object
print signature(x = "ADEg"): displays the ADEg object in the current device or in a new one
update signature(object = "ADEg"): modifies graphical parameters after the ADEg creation, updates the current display and returns the modified ADEg

Note

For any ADEg creation, various graphical parameters can be passed into the dots (...) arguments.

- the parameters listed in adegpar() can be changed, even if some of them do no modify the graphic representation chosen.
- the lattice parameters listed in trellis.par.get() can also be changed.
- limits, main and sub title, and axes labels can be changed using the keys xlim, ylim, main, sub, xlab and ylab.
- a neighbouring graph (object of class nb or listw) and a spatial one (object of class sp) can be display in the background using the keys nbobject, Sp and sp.layout.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEgS adegpar superpose insert

Examples

showClass("ADEg")

---

ADEg.C1-class

Class ADEg.C1

Description

An object of ADEg.C1 class represents unidimensional data into two dimensions.

The ADEg.C1 class is a virtual class, i.e. a class which is not possible to create objects but which have heirs. This class inherits from ADEg class and has three son classes : C1.barchart, C1.curve, C1.density, C1.dotplot, C1.gauss, C1.hist, C1.interval

Objects from the Class

None object of this class can be instantiated.
ADEg.C1-class

Slots

data a list containing data or data's name.
  • score: the displayed values in the form of a numeric vector, a name or a matching call.
  • frame: a positive or null integer. It is the number of the frame containing the data
    (used with sys.frame(..., env = data$frame)). Only if the data are not stored
    (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If
    FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings
  arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list of two elements to create the trellis object:
  • graphictype: xyplot
  • arguments: its parameters to obtain the trellis object

g.args a list containing some method parameters linked with the created object of ADEg.C1 class
stats a list of internal preliminary calculations
s.misc a list of some others internal parameters:
  • hori.update: a logical indicating if the sense of direction of the graphics is updating
  • backgrid: a list of two elements for grid lines. backgrid$x defines the coordinates of
    the lines (horizontal or vertical depending on the graphics orientation) and backgrid$d
    the grid mesh

Call an object of class call

Extends

Class ADEg, directly.

Methods

prepare signature(object = "ADEg.C1"): performs the calculations before display the object
  (e.g. limits, grid and axis calculations)

setlatticecall signature(object = "ADEg.C1"): prepares the lattice.call slot

panelbase signature(object = "ADEg.C1"): defines the graphical background (e.g. grid, rugs
  and box)

gettrellis signature(object = "ADEg.C1"): converts the graphic into a trellis object of
  lattice class

Note

The ADEg.S1 class and ADEg.C1 class are both used to represent an unidimensional information
  (e.g. a score). The difference between these two classes is mainly ideological: an ADEg.S1 object
  is a representation into one dimension (e.g. one line) while an ADEg.C1 object is a representation
  into two dimensions (e.g. curves).
Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

adegpar C1.barchart C1.curve C1.density C1.dotplot C1.gauss C1.hist C1.interval ADEg

Examples

showClass("ADEg.C1")


deg.panel.hist                  Panel function for adding histograms.

Description

Panel function for displaying histograms into a trellis graphic (lattice package) and level lines.

Usage

adeg.panel.hist(histValues, horizontal = TRUE, densi, drawLines, params = list(), identifier = "histogramADEg")

Arguments

  histValues an object of class histogram. See hist.
  horizontal a logical indicating if the plot is horizontal
  densi a list returns by the bkde containing the coordinates of the binned kernel density
  estimate of the probability density of the data
  drawLines a vector containing the level values
  params graphical parameters : plot_polygon, add_line and plot_line (lattice)
  identifier A character string that is prepended to the name of the grob that is created.

Value

Displays the histogram and level lines.

Note

For more information about the use of panel functions, please see the lattice package developed by Deepayan Sarkar.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
See Also

bkde and hist

Examples

```r
if(require(KernSmooth, quietly = TRUE) & require(lattice, quietly = TRUE)) {
  z <- round(rnorm(100, 30, 5))
  h <- hist(z, plot = FALSE)
  d <- bkde(z, kernel = "normal", gridsize = 60)
  l <- c(10, 20, 30, 40)
  xyplot(1:50 ~ 1:50, histValues = h, densi = d, drawLines = l,
         panel = function(drawLines, histValues, densi){
           adeg.panel.hist(histValues = histValues, drawLines = drawLines, densi = densi))
  }
}
```

adeg.panel.join

Panel function for joining lines.

Description

Panel function for drawing lines as part of a circle centred in (0, 0) into a trellis graphic (lattice package).

Usage

```r
adeg.panel.join(drawLines, params = list())
```

Arguments

- `drawLines`: a vector containing the level values used as radius of the circle
- `params`: graphical parameters: `plabels` and `add.line` (lattice)

Value

Displays level lines and their values.

Note

For more information about the use of panel functions, please see the lattice package developed by Deepayan Sarkar.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
Examples

```r
if(require(lattice, quietly = TRUE)) {
  xyplot(0:20 ~ 0:20, drawLines = c(5, 10, 15), params = list(plabels.cex = 2),
         panel = function(drawLines, params){
           adeg.panel.join(drawLines = drawLines, params = params))
    }
```

Description

Panel function for drawing labels into a trellis graphic (lattice package) with or without boxes around labels.

Usage

```r
adeg.panel.label(x, y, labels, plabels, pos = NULL)
```

Arguments

- `x`: a numeric vector, x-coordinates for the labels
- `y`: a numeric vector, y-coordinates for the labels
- `labels`: a vector of character string, the labels
- `plabels`: a list of parameters as an extract of adegpar("plabels"), used for labels’ drawing. Each value can be a vector and will be recycled if necessary:
  - `alpha`, `cex`, `col`: drawing parameters for the text
  - `srt`: orientation of the labels, horizontal, vertical or an angle indication (in degrees). Boxes are not rotated. If the orientation is not near to horizontal/vertical (0/90), it is best not to draw the boxes
  - `optim`: logical. If TRUE, uses an algorithm trying to avoid labels’ overlapping and outside limits
  - `boxes`: concerns the label’s boxes. a list:
    - `draw`: logical. If TRUE, labels are framed
    - `alpha`, `border`, `col`, `lwd`, `lty`: rule transparency, border lines and background color
- `pos`: a position specifier for the text, used in panel.text. Values of 1, 2, 3 and 4 respectively indicate positions below, to the left of, above and to the right of the specified coordinates.

Value

Draws the labels.
Note

For more information about the use of panel functions, please see the *lattice* package developed by Deepayan Sarkar.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References

The algorithm used for labels positions optimization is inspired by the `pointLabel` function of the `maptools` package developed by Tom Short.

See Also

`pointLabel`

Examples

```r
if(require(lattice, quietly = TRUE)) {
  param <- adegpar("labels")[[1]]
  xyplot(1:10 ~ 1:10, panel = function(x, y, ...){
    adeg.panel.label(x, y, LETTERS[1:10], labels = param))
}

if(require(lattice, quietly = TRUE)) {
  param$boxes$draw <- FALSE
  param$col <- "blue"
  xyplot(1:10 ~ 1:10, panel = function(x, y, ...){
    adeg.panel.label(x, y, LETTERS[1:10], labels = param))
}
```

Description

Panel function for representing a graph into a trellis graphic (*lattice* package). Two types of graph objects can be used: `nb` or `listw` object (*spdep* package) or simple edges informations. Directions associated with the edges are not displayed.
Usage

adeg.panel.nb(nbobject, coords, col.edge = "black", lwd = 1, lty = 1, pch = 20, cex = 1, col.node = "black", alpha = 1)

adeg.panel.edges(edges, coords, col.edge = "black", lwd = 1, lty = 1, pch = 20, cex = 1, col.node = "black", alpha = 1)

Arguments

- **nbobject**: a object of class `nb` or `listw`
- **edges**: a two columns matrix, representing the edges between the nodes. For a row i, x[i, 1] and x[i, 2] are linked, x[i, 1] and x[i, 2] being vertices number.
- **coords**: a two columns matrix containing vertices’ coordinates
- **col.edge**: edges’ color(s)
- **lwd**: line width (edges). Can be a vector
- **lty**: line type (edges). Can be a vector
- **pch**: vertices’ representation type (symbols). Can be a vector
- **cex**: symbols’ size(s) (vertices). Can be a vector
- **col.node**: vertices’ color(s). Can be a vector
- **alpha**: symbols’ transparency

Value

Displays the neighboring graph.

Note

For more information about the use of panel functions, please see the `lattice` package developed by Deepayan Sarkar.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References

Package `spdep`. Author: Roger Bivand

See Also

`plot.nb`
Examples

```r
if(require(lattice, quietly = TRUE) & require(spdep, quietly = TRUE)) {
  data(elec88, package = "ade4")
  coords <- elec88$xy
  xyplot(coords[, 2] ~ coords[, 1],
         panel = function(...) (adeg.panel.nb(elec88$nb, coords, col.edge = c("blue", "red"))))
}

if(require(lattice, quietly = TRUE)) {
  edges <- matrix(c(1, 2, 3, 2, 4, 1, 3, 4), byrow = TRUE, ncol = 2)
  coords <- matrix(c(0, 1, 1, 0, 0, -1, -1, 0), byrow = TRUE, ncol = 2)
  xyplot(coords[, 2] ~ coords[, 1],
         panel = function(...) (adeg.panel.edges(edges, coords, lty = 1:4, cex = 5)))
}
```

adeg.panel.Spatial  Panel function for adding spatial objects.

Description

Panel function adapted from the Sp package for displaying all kind of spatial objects handled by Sp (for classes inherited from the superclass Spatial) into a trellis graphic (lattice package).

Usage

```r
adeg.panel.Spatial(SpObject, sp.layout = NULL, col = 1, border = 1, lwd = 1, lty = 1, alpha = 0.8, cex = 1, pch = 20, n = length(col), spIndex = 1, ...)
```

Arguments

- **SpObject**
- **sp.layout**
  - a list of layout items. See `spplot` for more information
- **col**
  - background color (fill) of Spobject
- **border**
  - border color
- **lwd**
  - line width (border)
- **lty**
  - line type (border)
- **alpha**
  - background transparency of Spobject
- **cex**
  - point size
- **pch**
  - point type
if SpObject contains data, the _desired_ number of intervals splitting the data (using pretty).

spIndex if the SpObject contains a data frame, its values are represented with a color code. Only the spIndex data frame is represented

... for coherence with panel functions

Value

Draws the Spatial object and layout.

Note

If SpObject contains several maps, only the first one is selected. Also for objects containing more data (for classes data.frame with a slot data), this information is also shown. To do so, various colors can be used (according to the col arguments).

For more information about the use of panel functions, please see the lattice package developed by Deepayan Sarkar.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References

Package Sp. Author: Edzer Pebesma, Roger Bivand, Barry Rowlingson and Virgilo Gomez-Rubio.

See Also

spplot sp.lines sp.polygons sp.grid

Examples

if(require(lattice, quietly = TRUE) & require(sp, quietly = TRUE)) {
  data(elec88, package = "ade4")

  xy <- elec88$xy
  arrow <- list("SpatialPolygonsRescale", offset = c(150000,1700000),
                 layout.north.arrow(), scale = 100000)

  xyplot(xy[, 2] - xy[, 1], aspect = "iso", panel = function(...){
    adeg.panel.Spatial(SpObject = elec88$Spatial, sp.layout = list(arrow),
                      col = colorRampPalette(c("yellow", "blue"))(5), border =
                      "transparent"))
  }
}
adeq.panel.values  Panel function drawing a third variable into a two-dimensional scatterplot

Description
Panel function for drawing coordinates with variable representation. The values can be represented through symbols with proportional size or various colors.

Usage
adeq.panel.values(x, y, z, method, symbol, ppoints, breaks, centerpar = NULL, center = 0)

Arguments
- **x**: a numeric vector, x-coordinates for the symbols
- **y**: a numeric vector, y-coordinates for the symbols
- **z**: a numeric vector, the third variable with one value per coordinates (x, y)
- **method**: a character string equal to `color` or `size`. If `color`, a palette of color is used for the symbols (one color per interval defined by `breaks`). If `size`, symbols’ area is proportional to the value. Area is 0 for values equals to `center`. Two colors are used, one for values smaller than center and the other for values larger than center.
- **symbol**: a character string equal to `square` or `circle`.
- **ppoints**: a list of parameters as an extract of `adegpar("ppoints")`, used for points’ drawing.
  - `alpha`: transparency of points
  - `cex`: size of points
  - `col`: border color of points
  - `pch`: symbol to use
  - `fill`: filling color
- **breaks**: a vector, the breaks used for splitting z if method is `color`
- **centerpar**: a list to represent center value using elements in the `adegpar("ppoints")` list or NULL value. If the method is `size`, z-values equals to center have a size of zero. If `centerpar` is not NULL, those z-values are shown as points with the `centerpar` drawing parameters.
- **center**: a center value for method `size`

Value
Draws the points.
Note

For more information about the use of panel functions, please see the lattice package developed by Deepayan Sarkar.

For the symbols size, the method is size uses perceptual scaling (Tanimura et al. 2006).

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References


Examples

```r
if(require(lattice, quietly = TRUE)) {
  param <- adegpar("ppoints")[[1]]
  param$col <- adegpar("palette")[[1L]]$quanti(2)
  z <- rnorm(10)
  xyplot(1:10 ~ 1:10, panel = function(x, y, ...){
    adeg.panel.values(x, y, z, method = "size", symbol = "square", ppoints =
    param, breaks = pretty(z, 4)))
}

if(require(lattice, quietly = TRUE)) {
  param$col <- adegpar()$palette$quali((length(pretty(z, 2)) - 1))
  xyplot(1:10 ~ 1:10, panel = function(x, y, ...){
    adeg.panel.values(x, y, z, method = "color", symbol = "circle",
    ppoints = param, breaks = pretty(z, 2)))
}
```

ADEg.S1-class

Class ADEg.S1

Description

An object of ADEg.S1 class represents unidimensional data into one dimension.

The ADEg.S1 class is a virtual class, i.e. a class which is not possible to create objects but which have heirs. This class inherits from ADEg class and has five son classes: S1.boxplot, S1.class, S1.distri, S1.label and S1.match.

Objects from the Class

None object of this class can be instantiated.
ADEg.S1-class

Slots

data  a list containing data or data's name.

  • score: the displayed values in the form of a numeric vector, a name or a matching call.
  • at: the index value.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to `par.settings` arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by `adegpar()` function.

lattice.call a list of two elements to create the trellis object:

  • graphictype: `xyplot`
  • arguments: its parameters to obtain the trellis object

g.args a list containing some method parameters linked with the created object of ADEg.S1 class.

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters:

  • hori.update: a logical indicating if the sense of direction of the graphics is updating
  • backgrid: a list of two elements for grid lines. `backgrid$x` defines the coordinates of the lines (horizontal or vertical depending on the graphics orientation) and `backgrid$d` the grid mesh
  • rug: an index value indicating where the rugs are drawn

Call an object of class `call`

Extends

Class `ADEg`, directly.

Methods

prepare signature(object = "ADEg.S1"): performs the calculations before display the object (e.g. limits, grid and axis calculations)

setlatticecall signature(object = "ADEg.S1"): prepares the lattice.call slot

panelbase signature(object = "ADEg.S1"): defines the graphical background (e.g. grid, rugs and box)

gettrellis signature(object = "ADEg.S1"): converts the graphic into a trellis object of lattice class

zoom signature(object = "ADEg.S1", zoom = "numeric", center = "missing"): performs a zoom in (if zoom < 1) or out (if zoom > 1) centered, only in one-dimension

zoom signature(object = "ADEg.S1", zoom = "numeric", center = "numeric"): performs a zoom in (if zoom < 1) or out (if zoom > 1) around the center passed in parameter, only in one-dimension
Note

Various graphical parameters are used for display an ADEg.S1 object. The list `p1d` in `adegpar()` is thought specific for ADEg.S1 objects.

The ADEg.S1 class and ADEg.C1 class are both used to represent an unidimensional information (e.g. a score). The difference between these two classes is mainly ideological: an ADEg.S1 object is a representation into one dimension (e.g. one line) while an ADEg.C1 object is a representation into two dimensions (e.g. curves).

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot<auirelie.sibcherchicot@univ-lyon1.fr> and Stephane Dray

See Also

`adegpar`, `zoom`, `S1.boxplot`, `S1.class`, `S1.distri`, `S1.label`, `S1.match`, `ADEg` 

Examples

```r
showClass("ADEg.S1")
adegpar("p1d")
```

---

**ADEg.S2-class**

**Class** ADEg.S2

**Description**

An object of ADEg.S2 class represents bi-dimensional data.

The ADEg.S2 class is a virtual class, i.e. a class which is not possible to create objects but which have heirs. This class inherits from ADEg class and has eleven son classes: S2.arrow, S2.class, S2.corcircle, S2.density, S2.distri, S2.image, S2.label, S2.logo, S2.match, S2.trajec and S2.value.

**Objects from the Class**

None object of this class can be instantiated.

**Slots**

- `data`: a list containing data or data’s name.
  - `dfxy`: the displayed values in the form of a data frame, a name or a matching call.
  - `xax`: an integer or a vector indicating the columns of `dfxy` kept for the x-axes.
  - `yax`: an integer or a vector indicating the columns of `dfxy` kept for the y-axes.
  - `frame`: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(...)`, `env = data$frame`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
• storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list of two elements to create the trellis object:
  • graphictype: xyplot
  • arguments: its parameters to obtain the trellis object

g.args a list containing some method parameters linked with the created object of ADEg.S2 class:
  • fullcircle: only for S2.corcircle objects
  • method: only for S2.value objects
  • symbol: only for S2.value objects
  • center: only for S2.value objects

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters:
  • xfullcircle.update and yfullcircle.update: a logical indicating if the circle size is updating (only for S2.corcircle objects)
  • plegend.update: a logical indicating if the legend parameters are updating
  • breaks.update: a logical indicating if the legend breaks are updating
  • backgrid: a list of elements for grid lines

Call an object of class call

Extends

Class ADEg, directly.

Methods

prepare signature(object = "ADEg.S2"): performs the calculations before display the object (e.g. limits, grid and axis calculations)

setlatticecall signature(object = "ADEg.S2"): prepares the lattice.call slot

panelbase signature(object = "ADEg.S2"): defines the graphical background (e.g. grid and box)

gettrellis signature(object = "ADEg.S2"): converts the graphic into a trellis object of lattice class

zoom signature(object = "ADEg.S2", zoom = "numeric", center = "missing"): performs a zoom in (if zoom < 1) or out (if zoom > 1) centered

zoom signature(object = "ADEg.S2", zoom = "numeric", center = "numeric"): performs a zoom in (if zoom < 1) or out (if zoom > 1) around the center passed in parameter (center should be a two-length vector)

addhist signature(object = "ADEg.S2"): adds histograms and density lines against a bi-dimensional graphics
ADEg.T-class

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

addhist zoom adegpar S2.arrow S2.class S2.corcircle S2.density S2.distri S2.image S2.label S2.logo S2.match S2.trajec S2.value ADEg

Examples

showClass("ADEg.S2")

ADEg.T-class

Description

An object of ADEg.T class represents table data.

The ADEg.T class is a virtual class, i.e. a class which is not possible to create objects but which have heirs. This class inherits from ADEg class and has two son classes : T.image and T.value.

Objects from the Class

None object of this class can be instantiated.

Slots

data: a list containing data or data’s name.

• dftab: the displayed values which can be table, dist or matrix in the form of a data frame, a name or a matching call
• coordsx: an integer or a vector indicating the columns of dftab kept
• coordsy: an integer or a vector indicating the rows of dftab kept
• labelsx: the columns' labels
• labelsy: the rows' labels
• "frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
• storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.
adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list of two elements to create the trellis object:
• graphictype: xyplot
• arguments: its parameters to obtain the trellis object

g.args a list containing some method parameters linked with the created object of ADEg.T class:
• method: only for T.value objects
• symbol: only for T.value objects
• center: only for T.value objects

stats a list of internal preliminary calculations
s.misc a list of some others internal parameters:
• breaks.update: a logical indicating if the legend breaks is updating
• axes$dx and axes$dy: intervals for the cell size

Call an object of class call

Extends

Class ADEg, directly.

Methods

prepare signature(object = "ADEg.T"): performs the calculations before display the object (e.g. limits, grid and axis calculations)

setlatticecall signature(object = "ADEg.T"): prepares the lattice.call slot

panelbase signature(object = "ADEg.T"): defines the graphical background (e.g. axes, labels, ticks, box and grid)

gettrellis signature(object = "ADEg.T"): converts the graphic into a trellis object of lattice class

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

adegpar T.image T.value ADEg

Examples

showClass("ADEg.T")
ADEg.Tr-class

Class "ADEg.Tr"

Description

An object of ADEg.Tr class represents triangular coordinates in 2D.

The ADEg.Tr class is a virtual class, i.e. a class which is not possible to create objects but which have heirs. This class inherits from ADEg class and has three son classes: Tr.class, Tr.label, T.match and T.trajec.

Objects from the Class

None object of this class can be instantiated.

Slots

data: a list containing data or data’s name.
  • dfxyz: the displayed values in the form of a data frame with three columns, a name or a matching call.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list of two elements to create the trellis object:
  • graphictype: xyplot
  • arguments: its parameters to obtain the trellis object

g.args a list containing some method parameters linked with the created object of ADEg.Tr class:
  • max3d and min3d: triangular limits
  • adjust: a logical to adjust the device with the limits

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters:
  • adjust.update: a logical indicating if the adjust slot is updating
  • cornerp: coordinates of the triangle extremities.
  • lgrid: a list containing the three coordinates of the grid segments extremities(pts1, pts2, pts3) and the value of the division (posgrid)

Call an object of class call
adeqpar

Extends

Class ADEg, directly.

Methods

- `prepare` signature(object = "ADEg.Tr"): performs the calculations before display the object (e.g. limits, grid and axis calculations)
- `setlatticecall` signature(object = "ADEg.Tr"): prepares the lattice.call slot
- `panelbase` signature(object = "ADEg.Tr"): defines the graphical background (e.g. triangle and grid)
- `gettrellis` signature(object = "ADEg.Tr"): converts the graphic into a trellis object of lattice class

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

adeqpar Tr.class Tr.label Tr.match Tr.traject ADEg

Examples

showClass("ADEg.Tr")

Description

adeqpar can be used to set or query graphical parameters used in ADEg object display.

It is inspired by the `par` function of graphics package.

Usage

adeqpar(...)

Arguments

... If it is empty, the return value is a named list containing all the current settings.
If it is a string of characters, the corresponding sub-list of parameters is return as information.
If it is a list containing keys and values, the corresponding changes in current settings are made.
Details

The graphical parameters control apparency of the graphic. Calls can be made using either a list of list (e.g. plabels = list(col = "red")) or a list grouping both keys with "." (e.g. plabels.col = "red").

Parameters are re-used if needed in all ADEg object. If set globally, meaning using adegpar, all created objects afterwards will be affected.

Value

Several parameters are used to create complete plot and accessible through adegpar.

p1d: parameters for one-dimension graphic, object of class inherited from "ADEg.S1" or "ADEg.C1"

- **horizontal**: a logical indicating if the plot is horizontal
- **reverse**: a logical indicating if the bottom of the plot is at the bottom (for horizontal as TRUE) or at the left of the device (for horizontal as FALSE). If FALSE, the graphical display bottom is at the top (for horizontal as TRUE) or at the right of the device (for horizontal as FALSE).
- **rug**: a list dedicated to tick marks
  - **draw**: a logical indicating if the rugs are drawn
  - **tck**: size of the rug (ticks) in proportion from the reference line and the origin of the device (0.5 by default)
  - **margin**: where to draw the reference line (0.07 by default)
  - **line**: a logical indicating if the reference line is drawn using porigin arguments

parrows: arrows' parameters. see panel.arrows for more information

- **angle**: angle from the shaft of the arrow to the edge of the arrow head
- **ends**: kind of arrows to be drawn. Can be first, last or both
- **length**: length of the edges of the arrow head

paxes: axis' parameters. Mostly inspired by xyplot function of lattice package

- **aspectratio**: a character string to control physical aspect ratio of the graphic (drawing panel more specifically). iso for isometric scales, fill for drawing as big as possible or xy for banking rule
- **draw**: a logical indicating if axis (tick marks and labels) are drawn around the graphic
- **x**: a list used for the creation of x-axis in the trellis object. See xyplot for more information
  - **draw**: a logical indicating if x-axis (tick marks and labels) are drawn around the graphic
- **y**: the same list as for x with draw parameters

pbackground: background’s parameters

- **col**: background color
- **box**: a logical indicating if a box is drawn surrounding the plot

pellipses: ellipses’ drawing parameters

- **alpha**: a value between 0 and 1 controlling ellipses’ background transparency
- **axes**: a list dedicated to ellipses’ axis
– draw: a logical indicating whether ellipses’ axis are drawn
– col: ellipses’ axis color
– lty: line type of ellipses’ axis
– lwd: line width of ellipses’ axis
• border: ellipses’s border color
• lty: line type of ellipses’ border
• lwd: line width of ellipses’ border
• col: ellipses’ background color

pgrid: grid’s drawing parameters
• draw: a logical indicating if grid is drawn in the background
• col: grid’s line color
• lty: line type of grid line
• lwd: line width of grid line
• nint: an integer indicating the number of grid intervals expected
• text: a list dedicated to grid legend text
  – cex: text size of grid legend
  – col: text color of grid legend
  – pos: a character string (topright, topleft, bottomleft, bottomright) or a vector of length 2 indicating text position of grid legend. If it is a vector, the default unit is npc (normalized parent coordinates).

plabels: labels’ drawing parameters
• alpha: a value between 0 and 1 controlling label transparency
• cex: labels’ text size
• col: labels’ text color
• srt: labels’ text orientation. It can be horizontal, vertical or an angle indication in degrees
• optim: a logical indicating if an algorithm is used to avoid labels’ overlapping or outside limits
• boxes: label’s boxes parameters
  – draw: a logical indicating if labels are framed
  – alpha: a value between 0 and 1 controlling labels’ boxes transparency
  – border: boxes’ border color
  – col: boxes’ background color
  – lty: line type of boxes’ border
  – lwd: line width of boxes’ border

plegend: legend’s drawing parameters (used for object of class inherited from T.value and S2.value)
• drawKey: a logical indicating if the legend should be drawn. Legend can be provided by the key argument or is automatically generated for *.class and *.value functions
• drawColorKey: a logical indicating if the color legend should be drawn (only for *.image functions)
• size: size of the legend

plines: lines’ drawing parameters
• col: lines color
• lty: lines type
• lwd: lines width

pnb: drawing parameters for neighbourhood graph
• edge: edge’s drawing parameters
  – col: edge color
  – lty: line type of edge
  – lwd: line width of edge
• node: node’s drawing parameters
  – pch: node’s symbol type
  – cex: node’s symbol size
  – col: node’s symbol color
  – alpha: a value between 0 and 1 controlling node’s symbol transparency

porigin: drawing parameters for origin’s lines. See panel.lines for more information
• draw: a logical indicating if vertical and horizontal lines are drawn to indicate origin
• include: a logical indicating if origin is included in the drawing limits
• origin: a two-length vector indicating origin coordinates
• alpha: a value between 0 and 1 controlling origin’s lines transparency
• col: color of origin’s lines
• lty: origin’s line type
• lwd: origin’s line width

ppalette: a function taking one integer in argument indicating the number of expecting colors (for example using colorRampPalette)
• quanti: adegpar()$ppalette$quanti(n) returns n colors shaded grey to white
• quali: adegpar()$ppalette$quali(n, name) returns n differentiated colors. name argument is passed to the brewer.pal function of the RColorBrewer package and must be Accent, Dark2, Paired, Pastel1, Pastel2, Set1 (the default value), Set2 or Set3. When n is equal to 2, values for ’white’ and ’black’ colors are returned and can be not quite visible on the display.

ppoints: points’ drawing parameters
• alpha: a value between 0 and 1 controlling points transparency
• cex: points size
• col: points color
• pch: points type
• fill: points’ background color (only for filled points type)

ppolygons: polygons’ drawing parameters (used for example to draw convex hull for S2.class or Gaussian curves for C1.gauss objects). See lpolygon for more information.
• border: polygon’s border color
• col: polygon’s background color
• lty: line type of polygon border
• lwd: line width of polygon border
• alpha: a value between 0 and 1 controlling polygons’ background transparency

pSp: drawing parameters for spatial object
• col: spatial object’s background color
• border: spatial object’s border color
• lty: line type of spatial object border
• lwd: line width of spatial object border
• alpha: a value between 0 and 1 controlling spatial object transparency

psub: subtitle’s drawing parameters
• cex: text size of subtitle
• col: text color of subtitle
• position: a character string (topright, topleft, bottomleft, bottomright) or a vector of length 2 indicating text position of subtitle. If it is a vector, the default unit is npc (normalized parent coordinates).
• text: the character string to display

ptable: for table graphic, object of class inherited from adeG.T
• x: x-axis parameters
  – srt: text rotation
  – pos: position of the axis. It can be top or bottom. Otherwise axis and labels’ axis are not drawn
  – tck: ticks size
  – adj: justification of labels
• y: same as x list, but for y-axis
  – str, tck, adj
  – pos: position of the axis. It can be left or right. Otherwise axis and labels’ axis are not drawn
• margin: margin surrounding the drawing panel. The numbers indicate the bottom, left, top and right margins. Results are obtained passing margin to padding argument in lattice. Please see layout.heights and layout.widths parameters in lattice package for more information

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg par

Examples

oldparamadeg <- adegpar()

X <- data.frame(x = runif(50, -1, 2), y = runif(50, -1, 2))
s.label(X)
ADegS

Creation of ADegS objects

Description

Creates and displays an "ADegS" object, a set of ADeg, trellis and/or ADegS objects, managed by superposition, insertion and/or juxtaposition.

Usage

ADegS(adeglist, positions, layout, add = NULL, plot = TRUE)

Arguments

- **adeglist**: a list of several trellis, ADeg and/or ADegS objects.
- **positions**: a matrix with four columns and as many rows as the number of graphical objects in ADeglist slot. For each simple graphic, i.e. in each row, the coordinates of the top-right and the bottom-left hand corners are in npc unit (normalized parent coordinates).
- **layout**: a layout indication in two possible forms:
  - a list containing arguments of the layout function
  - a two-length vector containing rows' and columns' number of layout
- **add**: a square matrix with as many rows and columns as the number of graphical objects in the ADeglist slot. The value at the i-th row and j-th column is equal to 1 whether the j-th graphical object in ADeglist slot is superpose to i-th graphical one. Otherwise, this value is equal to 0.
- **plot**: a logical. If the graphics should be displayed

Value

an ADegS object. If plot = TRUE, the created object is displayed.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
ADEgS-class

See Also

ADEgS

Examples

```r
xy <- matrix(rnorm(20), ncol = 2)
g1 <- s.label(xy)
g2 <- s.class(xy, fac = as.factor(rep(LETTERS[1:2], length.out = 10)), ppoints.cex = 0, col = c("blue", "red"))
g3 <- ADEgS(list(g1, g2), rbind(c(0, 0.5, 1), c(0.5, 0, 1, 1)))
g4 <- ADEgS(list(g1, g2), layout = c(2, 1))
g5 <- ADEgS(list(g1, g2))
g6 <- ADEgS(list(g1, g2), add = matrix(c(1, 1, 0, 0), byrow = TRUE, ncol = 2))

data(olympic, package = "ade4")
dudi1 <- ade4::dudi.pca(olympic$tab, scan = FALSE)
g7 <- s.arrow(dudi1$li)
g8 <- s.corcircle(dudi1$co, lab = names(olympic$tab))
g9 <- ADEgS(list(g7, g8), rbind(c(0, 0.5, 1), c(0.5, 0, 1, 1)))
g9[1]
g9[1, drop = FALSE]
length(g9)
```

ADEgS-class

Class "ADEgS"

Description

An object of ADEgS class is a complex graphic. This class allows the superposition, the insertion and/or the juxtaposition of several ADEg, trellis and/or ADEgS objects.

The ADEgS class have neither father class nor son classes.

Objects from the Class

ADEgS objects can be created by calls of the form new("ADEgS", ...).

The regular usage in this package is to use the ADEgS, add.ADEg, superpose, insert or + functions.

Slots

- **ADEglist**: a list of several trellis, ADEg and/or ADEgS objects.
- **positions**: a matrix with four columns and as many rows as the number of graphical objects in the ADEglist slot. For each simple graphic, i.e. in each row, the coordinates of the top-right and the bottom-left hand corners are in npc unit (normalized parent coordinates).
- **add**: a square matrix with as many rows and columns as the number of graphical objects in the ADEglist slot. The value at the i-th row and j-th column is equal to 1 whether the j-th graphical object in the ADEglist slot is superpose to i-th graphical one. Otherwise, this value is equal to 0.
- **Call**: an object of class call

```r
ADEgS-class

Class "ADEgS"

Description

An object of ADEgS class is a complex graphic. This class allows the superposition, the insertion and/or the juxtaposition of several ADEg, trellis and/or ADEgS objects.

The ADEgS class have neither father class nor son classes.

Objects from the Class

ADEgS objects can be created by calls of the form new("ADEgS", ...).

The regular usage in this package is to use the ADEgS, add.ADEg, superpose, insert or + functions.

Slots

- **ADEglist**: a list of several trellis, ADEg and/or ADEgS objects.
- **positions**: a matrix with four columns and as many rows as the number of graphical objects in the ADEglist slot. For each simple graphic, i.e. in each row, the coordinates of the top-right and the bottom-left hand corners are in npc unit (normalized parent coordinates).
- **add**: a square matrix with as many rows and columns as the number of graphical objects in the ADEglist slot. The value at the i-th row and j-th column is equal to 1 whether the j-th graphical object in the ADEglist slot is superpose to i-th graphical one. Otherwise, this value is equal to 0.
- **Call**: an object of class call
```
Methods

\[ \text{signature}(x = \text{"ADEgS"}, i = \text{"numeric"}, j = \text{"missing"}, \text{drop} = \text{"logical"}): \text{extracts the } i\text{-th sub-graphics in the } x@\text{ADeglist. } i \text{ can be a vector. If } i \text{ is a single number and if the extracted graphic in an ADEg object, the sub-selection is in the form of ADEg if drop is TRUE and in the form of ADEgS otherwise.} \]

\[ \text{signature}(x = \text{"ADEgS"}, i = \text{"numeric"}, j = \text{"missing"}, \text{drop} = \text{"missing"}): \text{the same than the previous method. drop is FALSE by default} \]

\[ \text{signature}(x = \text{"ADEgS"}, i = \text{"character"}, j = \text{"missing"}): \text{extracts one sub-graphic, the } i\text{-th one, in the } x@\text{ADeglist} \]

\[ \text{signature}(x = \text{"ADEgS"}, i = \text{"numeric"}, j = \text{"missing"}, \text{value} = \text{"ADEg"}): \text{replaces one sub graphic, the } i\text{-th one, by an ADEg object in the } x@\text{ADeglist} \]

\[ \text{signature}(x = \text{"ADEgS"}, i = \text{"numeric"}, j = \text{"missing"}, \text{value} = \text{"ADEgS"}): \text{replaces one sub graphic, the } i\text{-th one, by an ADEgS object in the } x@\text{ADeglist} \]

\$ \text{signature}(x = \text{"ADEgS"}): \text{extracts one sub-graphic by its name in the } x@\text{ADeglist} \]

\text{getpositions signature(object} = \text{"ADEgS"): \text{returns the positions matrix of the object, i.e. object@positions} \]

\text{getgraphics signature(object} = \text{"ADEgS"): \text{returns the list of graphics of the object, i.e. object@ADeglist} \]

\text{getcall signature(object} = \text{"ADEgS"): \text{returns the call of the object, i.e. object@Call} \]

\text{names signature(object} = \text{"ADEgS"): \text{returns the graphics’ names of the object, i.e. the names of object@ADeglist} \]

\text{names<- signature(object} = \text{"ADEgS"): \text{replaces the graphics’ names of the object, i.e. the names of object@ADeglist} \]

\text{length signature(x} = \text{"ADEgS"): \text{returns the number of graphics into x, i.e. the length of x@ADeglist} \]

\text{plot signature(x} = \text{"ADEgS"): same as print} \]

\text{print signature(x} = \text{"ADEgS"): \text{displays the graphical elements into one device using positions and superposition management (x@add matrix)} \]

\text{show signature(object} = \text{"ADEgS"): same as print} \]

\text{superpose signature(g1} = \text{"ADEgS"}, g2 = \text{"ADEg"}, which = \text{"numeric"}, plot = \text{"logical"): \text{creates a new "ADEgS" object performing a superposition of g2 on the which-th ADEg object of g1. This object is printed if plot is TRUE.} \]

\text{superpose signature(g1} = \text{"ADEgS"}, g2 = \text{"ADEg"}, which = \text{"numeric"}, plot = \text{"ANY"): \text{creates a new "ADEgS" object performing a superposition of g2 on the which-th ADEg object of g1. This object is printed only if plot is TRUE.} \]

\text{superpose signature(g1} = \text{"ADEgS"}, g2 = \text{"ADEg"}, which = \text{"missing"}, plot = \text{"ANY"): \text{creates a new "ADEgS" object performing a superposition of g2 on the last ADEg object of g1.} \]

\text{This object is printed only if plot is TRUE.} \]

\text{superpose signature(g1} = \text{"ADEgS"}, g2 = \text{"ADEgS"}, which = \text{"missing"}, plot = \text{"ANY"): \text{creates a new "ADEgS" object performing a superposition between two ADEgS having the same length and the same positions slot. It is used when g1 and g2 are both created with a partition of individual groups, variables or analysis’ axis.} \]
C1.barchart-class

+ signature(e1 = "ADEg", e2 = "ADEgS"): creates a new "ADEgS" object performing a superposition of e1 on e2.
+ signature(e1 = "ADEgS", e2 = "ADEg"): creates a new "ADEgS" object performing a superposition of e2 on e1.

\texttt{cbindADEg} signature(g1 = "ADEgORADEgS", g2 = "ADEgORADEgS"): creates a new "ADEgS" object combining g1 on g2 by columns.

\texttt{rbindADEg} signature(g1 = "ADEgORADEgS", g2 = "ADEgORADEgS"): creates a new "ADEgS" object combining g1 on g2 by rows.

\texttt{update} signature(object = "ADEgS"): modifies the graphical parameters of each sub-graphics listed in object@ADEglist and/or the object’s names (with the key word names) and/or the object@positions slot (with the key word positions), after creation of the object. The current display is updated and a modified object is returned.

\texttt{insert} signature(graphics = "ADEgS", oldgraphics = "missing", posi, ratio, inset, plot, which, dispatch): creates a new "ADEgS" object performing an insertion of graphics into the current device.

\texttt{insert} signature(graphics = "ADEgS", oldgraphics = "ADEg", posi, ratio, inset, plot): creates a new "ADEgS" object performing an insertion of graphics into oldgraphics.

\texttt{insert} signature(graphics = "ADEgORtrellis", oldgraphics = "ADEgS", posi, ratio, inset, plot, which): creates a new "ADEgS" object performing an insertion of graphics into oldgraphics.

\texttt{insert} signature(graphics = "ADEgS", oldgraphics = "ADEgS", posi, ratio, inset, plot, which, dispatch): creates a new "ADEgS" object performing an insertion of graphics into oldgraphics.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEgS superpose insert

Examples

\begin{verbatim}
showClass("ADEgS")
\end{verbatim}

Description

A class for the creation and display of a numeric score using barcharts.

Objects from the Class

\texttt{C1.barchart} objects can be created by calls of the form \texttt{new("C1.barchart", ...)}.

The regular usage in this package is to use the \texttt{s1d.barchart} function.
Slots

data a list containing data or data’s name.
  • score: the displayed values in the form of a vector, a factor, a name or a matching call.
  • labels: the labels’ names drawn on the top of bars.
  • at: the index value.
  • frame: a positive or null integer. It is the number of the frame containing the data
    (used with sys.frame(..., env = data$frame)). Only if the data are not stored
    (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If
    FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings
arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of ADEg.C1 class.
stats a list of internal preliminary calculations
s.misc a list of some others internal parameters
Call an object of class call

Extends

Class ADEg.C1, directly.
Class ADEg, by class ADEg.C1, distance 2.
Class ADEgORTrellis, by class ADEg.C1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.C1, distance 3.

Methods

The methods of the father classes "ADEg.C1" and "ADEg" can be used by inheritance. The specific
methods for C1.barchart are:

prepare signature(object = "C1.barchart"): calls the parent method (prepare for ADEg.C1)
and modifies some graphical parameters used by default.

panel signature(object = "C1.barchart"): draws bar charts and labels.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane
Dray

See Also

ADEg ADEg.C1 s1d.barchart

Examples

showClass("C1.barchart")
Description

A class for the creation and display of a numeric score linked by curves. The C1.curve allows to deal with multiple scores.

Objects from the Class

C1.curve objects can be created by calls of the form new("C1.curve", ...). The regular usage in this package is to use the s1d.curve function.

C1.curves objects can be created by calls of the form new("C1.curves", ...). The regular usage in this package is to use the s1d.curves function. Class C1.curves extends C1.curve directly.

Slots

data a list containing data or data's name.

• score: the displayed values in the form of a vector, a factor, a name or a matching call.
• at: the index value.
• frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
• storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of ADEg.C1 class.
stats a list of internal preliminary calculations
s.misc a list of some others internal parameters

Call an object of class call

Extends

Class ADEg.C1, directly.
Class ADEg, by class ADEg.C1, distance 2.
Class ADEgORTrellis, by class ADEg.C1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.C1, distance 3.
Methods

The methods of the father classes "ADEg.C1" and "ADEg" can be used by inheritance. The specific methods for C1.curve and C1.curves are:

- **prepare** signature(object = "C1.curve"): calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default.
- **panel** signature(object = "C1.curve"): draws points and curves.
- **panel** signature(object = "C1.curves"): draws points and curves.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.C1 s1d.curve s1d.curves

Examples

```r
showClass("C1.curve")
showClass("C1.curves")
```

Description

A class for the creation and display of a numeric score using density curves.

Objects from the Class

C1.density objects can be created by calls of the form new("C1.density", ...). The regular usage in this package is to use the s1d.density function.

Slots

data  a list containing data or data's name.
- **score**: the displayed values in the form of a numeric vector, a name or a matching call.
- **fac**: a factor for score to split in the form of a vector, a factor, a name or a matching call.
- **frame**: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
- **storeData**: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.
trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of AdeG.C1 class.

The specific slots for C1.density objects are:

- kernel, bandwidth and gridsize: passed in parameters in bkde function of the KernSmooth package.
- fill: a logical to yield the polygons density curves filled.
- col: a logical, a color or a colors vector to color labels, rugs, lines and polygons.

stats a list of internal preliminary calculations. The specific slot for C1.density objects is:

- densit: the values of density curve calculated for each factor in fac computes with the bkde function of the KernSmooth package.

s.misc a list of some others internal parameters. The specific slot for C1.density objects is:

- rug: an index value indicating where the rugs are drawn

Call an object of class call

Extends

Class AdeG.C1, directly.
Class AdeG, by class AdeG.C1, distance 2.
Class AdeGORTrellis, by class AdeG.C1, distance 3.
Class AdeGORADEGsORTrellis, by class AdeG.C1, distance 3.

Methods

The methods of the father classes "AdeG.C1" and "AdeG" can be used by inheritance. The specific methods for C1.density are:

prepare signature(object = "C1.density"): calls the parent method (prepare for AdeG.C1), modifies some graphical parameters used by default and calculates the density curves according to the numeric score and the values’ categories.

panel signature(object = "C1.density"): draws density curves.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg AdeG.C1 sld.density

Examples

showClass("C1.density")
Description

A class for the creation and display of a numeric score using dots.

Objects from the Class

C1.dotplot objects can be created by calls of the form new("C1.dotplot", ...).

The regular usage in this package is to use the s1d.dotplot function.

Slots

data  a list containing data or data’s name.

  • score: the displayed values in the form of a vector, a factor, a name or a matching call.
  • at: the index value.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par  a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par  a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call  a list to create the trellis object.

g.args  a list containing some method parameters linked with the created object of ADEg.C1 class.

stats  a list of internal preliminary calculations

s.misc  a list of some others internal parameters

Call  an object of class call

Extends

Class ADEg.C1, directly.
Class ADEg, by class ADEg.C1, distance 2.
Class ADEgORTrellis, by class ADEg.C1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.C1, distance 3.

Methods

The methods of the father classes "ADEg.C1" and "ADEg" can be used by inheritance. The specific methods for C1.dotplot are:

prepare signature(object = "C1.dotplot"): calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default.

panel signature(object = "C1.dotplot"): draws segments and dots.
Description

A class for the creation and display of a numeric score using gauss’ curves.

Objects from the Class

C1.gauss objects can be created by calls of the form new("C1.gauss", ...).

The regular usage in this package is to use the s1d.gauss function.

Slots

data  a list containing data or data's name.

  • score: the displayed values in the form of a numeric vector, a name or a matching call.
  • fac: a factor for score splitting in the form of a vector, a factor, a name or a matching call.
  • wt: a vector of weights for score.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of ADEg.C1 class. The specific slots for C1.gauss objects are:

  • fill: a logical to yield the gauss curves transparent.
  • col: a logical, a color or a colors vector to color labels, rugs, lines and polygons.
• steps: a value for the number of segments used to draw Gauss curves.

stats  a list of internal preliminary calculations. The specific slots for C1.gauss objects are:

• means: the weighted mean calculated for each fac value.
• var: the weighted variance calculated for each fac value.
• gausscurves: the density gauss curve calculated for each fac value.

s.misc  a list of some others internal parameters. The specific slot for C1.gauss objects is:

• rug: an index value indicating where the rugs are drawn

Call  an object of class call

Extends

Class ADEg.C1, directly.
Class ADEg, by class ADEg.C1, distance 2.
Class ADEgORtrellis, by class ADEg.C1, distance 3.
Class ADEgORADEgSORtrellis, by class ADEg.C1, distance 3.

Methods

The methods of the father classes "ADEg.C1" and "ADEg" can be used by inheritance. The specific methods for C1.gauss are:

prepare  signature(object = "C1.gauss"): calls the parent method (prepare for ADEg.C1), modifies some graphical parameters used by default and calculates the Gauss curves according to the numeric score and the values’ categories (using weighted mean and standard deviation).

panel  signature(object = "C1.gauss"): draws Gauss curves and level names of each curve.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.C1 s1d.gauss

Examples

showClass("C1.gauss")
Description

A class for the creation and display of a numeric score using a histogram.

Objects from the Class

C1.hist objects can be created by calls of the form new("C1.hist", ...).

The regular usage in this package is to use the s1d.hist function.

Slots

data  a list containing data or data’s name.

  • score: the displayed values in the form of a vector, a factor, a name or a matching call.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par  a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par  a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call  a list to create the trellis object.

g.args  a list containing some method parameters linked with the created object of ADEg.C1 class.

  The specific slots for C1.hist objects are:
  • breaks: a vector of values to split score. If NULL, pretty(score, nclass) is used.
  • nclass: an integer for the number of desired intervals, ignored if breaks is not missing.
  • type: a value among count, density, percent to indicate the unit of the cell height.
  • right: a logical indicating if the histogram cells are right-closed (left open) intervals.

stats  a list of internal preliminary calculations. The specific slots for C1.hist objects are:

  • heights: the cell height.
  • breaks: the cell boundaries.

s.misc  a list of some others internal parameters

Call  an object of class call

Extends

Class ADEg.C1, directly.
Class ADEg, by class ADEg.C1, distance 2.
Class ADEgORTrellis, by class ADEg.C1, distance 3.
Class ADEgORADEgSORtrellis, by class ADEg.C1, distance 3.
Methods

The methods of the father classes "ADEg.C1" and "ADEg" can be used by inheritance. The specific methods for C1.hist are:

**prepare** signature(object = "C1.hist"): calls the parent method (prepare for ADEg.C1), modifies some graphical parameters used by default and calculates the boundaries and the height of cells.

**panel** signature(object = "C1.hist"): draws rectangles.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.C1 s1d.hist

Examples

showClass("C1.hist")

---

C1.interval-class  Class C1.interval

Description

A class for the creation and display of an interval between two numeric scores.

Objects from the Class

C1.interval objects can be created by calls of the form new("C1.interval", ...).

The regular usage in this package is to use the s1d.interval function.

Slots

data  a list containing data or data’s name.
  • score: the displayed values in the form of a vector, a factor, a name or a matching call.
  • at: the index value.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par  a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.
adeq.par  a list of graphical parameters, corresponding to the ones given by adegpars() function.
lattice.call a list to create the trellis object.
g.args  a list containing some method parameters linked with the created object of ADEg.C1 class.

The specific slot for C1.density objects is:

- method: a value, bars or area, to represent either segments or areas between scores.

stats  a list of internal preliminary calculations
s.misc  a list of some others internal parameters

Call  an object of class call

**Extends**

Class ADEg.C1, directly.
Class ADEg, by class ADEg.C1, distance 2.
Class ADEgORTrellis, by class ADEg.C1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.C1, distance 3.

**Methods**

The methods of the father classes "ADEg.C1" and "ADEg" can be used by inheritance. The specific methods for C1.interval are:

**prepare** signature(object = "C1.interval"): calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default.

**panel** signature(object = "C1.interval"): draws segments or polygons.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

**See Also**

ADEg ADEg.C1 s1d.interval

**Examples**

showClass("C1.interval")
**cbindADEg**

*Combine ADEg objects by columns or rows*

**Description**

Take a sequence of ADEg, ADEgS or trellis arguments and combine by columns or rows, respectively.

**Usage**

```r
cbindADEg(g1, g2, ..., plot = FALSE)
rbindADEg(g1, g2, ..., plot = FALSE)
```

**Arguments**

- `g1`: an object of class ADEg, ADEgS or trellis
- `g2`: an object of class ADEg, ADEgS or trellis
- `...`: other objects of class ADEg, ADEgS or trellis
- `plot`: a logical indicating if the graphics is displayed

**Value**

an ADEgS object

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot (<aurelie.siberchicot@univ-lyon1.fr>) and Stephane Dray

**See Also**

ADEg ADEgS ADEgS

**Examples**

```r
data(jv73, package = "ade4")
pca1 <- ade4::dudi.pca(jv73$morpho, scanf = FALSE)
g1 <- s.label(pca1$li, plabels.optim = TRUE, plot = FALSE)
g2 <- s.class(pca1$li, jv73$fac.riv, starSize = 0, ellipseSize = 0, chullSize = 1,
              ppolygons.alpha = 0.4, col = rainbow(12), ppoints.cex = 0, plot = FALSE)
g3 <- s.corcircle(pca1$co, pbackground.box = FALSE, plot = FALSE)
g4 <- rbindADEg(cbindADEg(g1, g2), cbindADEg(superpose(g1, g2), g3), plot = TRUE)
```
changelatticetheme  

Change the lattice theme used for adegraphics

Description

This function allows to modify the default theme existing for adegraphics objects. The created theme also affects previously created objects.

Usage

changelatticetheme(...)

Arguments

... lattice parameters, the same used in `trellis.par.set` and provided by `trellis.par.get`. If empty, reset the theme to the adegraphics one.

Note

The adegraphics theme removes all margins, sets a transparent background and grey regions.

A further development will be the creation of various themes for adegraphics.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

trellis.par.get  trellis.par.set  show.settings

Examples

```r
if(require(lattice, quietly = TRUE)) {
  show.settings()
  changelatticetheme(list(superpose.symbol = list(pch = c(21, 22, 35), cex = 1)))
  show.settings()
  show.settings()[1]
}
```
**getcall-methods** *Method for ADEg and ADEgS objects*

**Description**

getcall returns the call used to create the object.

**Methods**

signature(object = "ADEg") returns the slot Call of the object ADEg

signature(object = "ADEgS") returns the slot Call of the object ADEgS

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

---

**insert** *Insert a graphic into an existing one*

**Description**

This function inserts a first graphic into a previously created and/or a displayed one.

**Usage**

insert(graphics, oldgraphics, posi = c("bottomleft", "bottomright", "topleft", "topright"), ratio = 0.2, inset = 0.0, plot = TRUE, which, dispatch = FALSE)

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>graphics</td>
<td>an object of class ADEg, ADEgS or trellis</td>
</tr>
<tr>
<td>oldgraphics</td>
<td>an object of class ADEg, ADEgS or missing. If oldgraphics is missing, graphics is added on the current device.</td>
</tr>
<tr>
<td>posi</td>
<td>a character value or a two-length numeric vector (in normalized parent coordinates npc from 0 to 1) indicating the position of oldgraphics added into graphics</td>
</tr>
<tr>
<td>ratio</td>
<td>a numeric value from 0 to 1 indicating the size of oldgraphics regarding the plot region</td>
</tr>
<tr>
<td>inset</td>
<td>the inset from which the graph is drawn regarding the plot region. It can be a two-length vector giving the inset in x and y. If atomic, same inset is used in x and y.</td>
</tr>
<tr>
<td>plot</td>
<td>a logical indicating if the graphics is displayed</td>
</tr>
</tbody>
</table>
which a numeric value or a vector of values only used if oldgraphics is an ADEgS object, indicating the which-th sub-graphic of oldgraphics where graphics is added.

dispatch a logical only used if both graphics and oldgraphics are ADEgS objects with same length, indicating if graphics is added one by one into oldgraphics. It is used when both graphics and oldgraphics are created with facets option.

Value
An object of class "ADEgS".

Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also
ADEg ADEgS

Examples

data(deug, package = "ade4")
dd1 <- ade4::dudi.pca(deug$tab, scannf = FALSE, nf = 4)
g1 <- s.label(dd1$li, labels = rownames(dd1$li), plabels = list(cex = 0.75), plot = FALSE)
g2 <- s1d.barchart(score = dd1$eig, plot = FALSE,
ppolygons = list(col = c(rep("black", 2), rep("grey", 2), rep("white", 5))),
pld = list(horizontal = FALSE), psub = list(position = "topright", text = "Eigenvalues"),
pgrid = list(draw = FALSE), pbackground = list(box = TRUE), xlim = c(0.5, 9.5))
g1

mat <- g3@positions
mat[2, 1] <- c(0.8, 0, 1, 0.2)
update(g3, positions = mat, plot = FALSE)
print(g3) ## square == NULL
print(g3, square = TRUE)
print(g3, square = FALSE)

ng4 <- insert(g2, g1, posi = "topleft")
data(jv73, package = "ade4")
pca1 <- ade4::dudi.pca(jv73$morpho, scannf = FALSE)
g5 <- s.value(jv73$xy, pca1$li[, 1:2], porigin.include = FALSE, plot = FALSE)
g6 <- s.corcircle(pca1$co, pbackground.box = FALSE, plot = FALSE)
g7 <- insert(g6, g5, posi = c(0.3, 0.4, 0.5, 0.6))
Transform a layout matrix into a position one

### Description
This function transforms layout's informations into a position matrix useful for ADEgS and for lattice graphics.

### Usage
```r
layout2position(mat, widths = rep(1, NCOL(mat)), heights = rep(1, NROW(mat)), ng, square = FALSE)
```

### Arguments
- **mat**: a matrix indicating the location of figures to display (each value must be 0 or a positive integer) or a two-length vector indicating the number of rows and columns in the corresponding layout.
- **widths**: a vector of relative values for the columns' widths on the device. Their sum must be equal to the number of columns.
- **heights**: a vector of relative values for the rows' heights on the device. Their sum must be equal to the number of rows.
- **ng**: a value for the number of positions needed (i.e. the number of graphics to plot).
- **square**: a logical indicating if the graphics is an isometric plot.

### Value
A four-columns matrix indicating the coordinates (in normalized parent coordinates npc) of the top-right and bottom-left hand corners of each displayed figure on the device.

### Note
This function is strongly inspired by the layout function in graphics package.

### Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

### See Also
- `layout`

### Examples
```r
layout2position(mat = rbind(c(0, 0, 1), c(2, 2, 1)))
layout2position(mat = cbind(c(0, 0, 1), c(2, 2, 1)), widths = c(0.5, 1.5))
```
Description

The method `panel` displays all specific graphical components.

Methods

signature(object = "C1.barchart") draws bar charts and labels
signature(object = "C1.curve") draws points and curves
signature(object = "C1.curves") draws multiple points and curves
signature(object = "C1.density") draws density curves
signature(object = "C1.dotplot") draws segments and dots
signature(object = "C1.gauss") draws Gauss curves and level names of each curve
signature(object = "C1.hist") draws rectangles
signature(object = "C1.interval") draws segments or polygons
signature(object = "S1.boxplot") draws box-and-whiskers diagrams, mean points and labels
signature(object = "S1.class") draws labels and lines matching with score values
signature(object = "S1.distri") draws mean points and segments with matching labels
signature(object = "S1.label") draws labels and its links with score points
signature(object = "S1.match") draws score points and matching segments and labels
signature(object = "S2.arrow") draws points, arrows and labels
signature(object = "S2.class") draws ellipses, convex hulls, stars, labels and points
signature(object = "S2.corcircle") draws arrows, labels and axes
signature(object = "S2.density") draws densities and external points
signature(object = "S2.distri") draws ellipses, stars, labels and points
signature(object = "S2.image") draws raster image
signature(object = "S2.label") draws points and labels
signature(object = "S2.logo") displays the logos
signature(object = "S2.match") draws arrows and labels
signature(object = "S2.traject") draws points, arrows and labels
signature(object = "S2.value") draws symbols
signature(object = "T.cont") draws mean points and regression lines
signature(object = "T.image") draws raster image
signature(object = "T.value") draws symbols
signature(object = "Tr.class") draws arrows, labels and points
signature(object = "Tr.label") draws lines, labels and points
signature(object = "Tr.match") draws arrows, labels and points
signature(object = "Tr.traject") draws arrows, labels and points
Methods to display the outputs of an analysis performed with ade4

Description

S3 methods to display the outputs of an analysis performed with ade4

Usage

```r
## S3 method for class 'foucart'
kplot(object, xax = 1, yax = 2, which.tab = 1:length(object$blo), pos = -1,
       storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'mcoa'
kplot(object, xax = 1, yax = 2, which.tab = 1:nrow(object$cov2),
       option = c("points", "axis", "columns"), pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'mfa'
kplot(object, xax = 1, yax = 2, which.tab = 1:length(object$blo), traject = FALSE,
       permute = FALSE, pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'mbpcaiv'
kplot(object, xax = 1, yax = 2, which.tab = 1:length(object$blo), pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'pta'
kplot(object, xax = 1, yax = 2, which.tab = 1:nrow(object$RV), which.graph = 1:4,
       pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'sepan'
kplot(object, xax = 1, yax = 2, which.tab = 1:length(object$blo), permute = FALSE,
       traject = FALSE, posieig = "bottomleft", pos = -1, storeData = TRUE, plot = TRUE, ...)
kplotsepan.coa(object, xax = 1, yax = 2, which.tab = 1:length(object$blo),
               permute = FALSE, posieig = "bottomleft", pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'statis'
kplot(object, xax = 1, yax = 2, which.tab = 1:length(object$tab.names), traject = FALSE,
       arrow = TRUE, class = NULL, pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'acm'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'betcoi'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'betdpcoa'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'betwitdpcoa'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)
## S3 method for class 'betrlq'
```
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'between'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'coinertia'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'discrimin'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'dpcoa'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'fca'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'foucart'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'krandboot'
plot(x, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'krandxval'
plot(x, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'mcoa'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'mfa'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'multiblock'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'multispati'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'niche'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'pcaiv'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'pta'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'procuste'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'randboot'
plot(x, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'randxval'
plot(x, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'rlq'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'sepan'
plot(x, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'statis'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)  # S3 method for class 'witcoi'
plot(x, xax = 1, yax = 2, pos = -1, storeData = TRUE, plot = TRUE, ...)
## Arguments

- `object, x` objects used to select a method
- `xax` an integer (or a vector) indicating which column(s) of object or x is(are) plotted on the x-axis
- `yax` an integer (or a vector) indicating which column(s) of object or x is(are) plotted on the y-axis
- `which.tab` a numeric vector (used in `kplot.*`) containing the numbers of the tables used for the analysis
- `option` a string of characters (only used in `kplot.mfa`) indicating the drawing option: `points` plot of the projected scattergram onto the co-inertia axes, `axis` projection of the axes and the position of the points
tions of inertia axes onto the co-inertia axes, columns projections of variables onto the synthetic variables planes.

which.graph an integer between 1 and 4 (only used in kplot.pta) indicating the drawing option. For each table of which.tab, are drawn: 1 the projections of the principal axes, 2 the projections of the rows, 3 the projections of the columns, 4 the projections of the principal components onto the planes of the compromise.

permute a logical value (used in kplot.sepan, kplotsepan.coa and scatter.dudi). If FALSE, the rows are plotted by points or density surface and the columns by arrows. If TRUE, it is the opposite.

trajec a logical value (used in kplot.sepan and kplot.statis) indicating whether the trajectories between rows should be drawn in a natural order

posieig a character value or a two-length numeric vector (in normalized parent coordinates npc from 0 to 1) or none value indicating the position of the eigenvalues bar plot (used in kplot.sepan, kplotsepan.coa and scatter.*).

arrow a logical value (only used in kplot.statis) indicating whether the column factorial diagrams should be plotted

class if not NULL, a factor of length equal to the number of the total columns of the K-tables (only used in kplot.statis)

prop a logical value (only used in scatter.dudi) indicating if the size of the arrows' labels is proportional to the analysis score.

density.plot a logical value (only used in scatter.dudi) indicating if the points are displayed as density surface (using s.density).

method an integer between 1 and 3 (only used in scatter.coa) indicating the drawing option. Are drawn: 1 rows and columns with the coordinates of lambda variance, 2 rows variance 1 and columns by averaging, 3 columns variance 1 and rows by averaging.

which.var the numbers of the kept columns for the analysis, otherwise all columns (used in score.*)

type a string of characters (only used in score.acm and score.mix) indicating if points (points) or boxplot (boxplot) are used to represent levels of factors

col.kept one color value to color the kept axes in the barchart (used in screeplot.dudi)

col one color value to color the axes in the barchart (used in screeplot.dudi)

plot a logical indicating if the graphics is displayed

storeData a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored

pos an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

... additional graphical parameters (see adegpar and trellis.par.get)

Value

Returns an ADEg or an ADEgS object. The result is displayed if plot is TRUE.
Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References

See ade4 website: <URL: http://pbil.univ-lyon1.fr/ADE-4/>

Examples

```r
cat("To run the example on 'topic'\n")
cat("Type in your R console: example(topic, package = 'ade4') \n")
```

plot.inertia

Display the decomposition of inertia which measure the contributions of rows/columns in multivariate methods

Description

S3 method to display the decomposition of inertia (inertia object) which measure the contributions of rows/columns in multivariate methods (dudi objects from ade4)

Usage

```r
## S3 method for class 'inertia'
plot(x, xax = 1, yax = 2, threshold = 0.1,
    contrib = c("abs", "rel"), type = c("label", "cross", "ellipse", "both"),
    ellipseSize = 1.5, posieig = "none", plot = TRUE,
    storeData = TRUE, pos = -1, ...)

## S3 method for class 'inertia'
score(x, xax = 1, threshold = 0.1, contrib = c("abs", "rel"),
    posieig = "none", pos = -1, storeData = TRUE, plot = TRUE, ...)
```

Arguments

- `x`: an object of the dudi class; it must be the output of a correspondence analysis (coa object).
- `xax`: an integer indicating which column of `x` is plotted on the x-axis
- `yax`: an integer indicating which column of `x` is plotted on the y-axis. If `yax` is equal to `xax`, a one-dimensional graph is displayed.
- `threshold`: a numeric value containing the contribution threshold (between 0 and 1) at which points should be drawn on the graphic. Low contribution points will be represented by a grey point and without label. When the contributions are displayed on a single axis, a dotted line describes the contribution threshold.
contrib a character value indicating which contributions are plotted: abs for absolute 
contributions (rows/columns involved in the factor axis/map construction) and 
rel for relative contribution (quality of rows/columns representation on the fac-
tor axis/map).

type a character value indicating which type represents contribution. Labels size 
(label), crosses size(cross) or ellipses size (ellipse) can be proportional to 
the contributions. If type is both, crosses and ellipses both have sizes propor-
tional to the contributions.

ellipseSize a positive number for ellipse size when type is ellipse

posieig a character value or a two-length numeric vector (in normalized parent coordi-
nates npc from 0 to 1) or none value indicating the position of the eigenvalues 
bar plot.

plot a logical indicating if the graphics is displayed

storeData a logical indicating if the data should be stored in the returned object. If FALSE, 
only the names of the data arguments are stored

pos an integer indicating the position of the environment where the data are stored, 
relative to the environment where the function is called. Useful only if storeData 
is FALSE

... additional graphical parameters (see adegpar and trellis.par.get)

Value

Returns an ADEgS object. The result is displayed if plot is TRUE.

Author(s)

Clément Claustre, Anne-Béatrice Dufour, Aurélie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> 
and Stéphane Dray

Examples

# First example

data(bf88, package = "ade4")
coa1 <- ade4::dudi.coa(bf88$S1, scannf = FALSE, nf = 2)

#### row=F / col=F

res11 <- ade4:::inertia(coa1, row = TRUE, col = FALSE, nf = 2)
g11 <- plot(res11, threshold = 0.06)
g12 <- plot(res11, threshold = 0.06, plabels.boxes.draw = TRUE, plines.lwd = 0, 
   light_row.ppoints.cex = 0, posieig = "bottomleft")
g13 <- score(res11, threshold = 0.06) 
   names(g13)
g14 <- score(res11, xax = 2, threshold = 0.06)

#### row=T / col=F

res12 <- ade4:::inertia(coa1, row = FALSE, col = TRUE, nf = 2)
res12$col.abs
id <- which(res12$col.abs[, 1]/100 >= 0.1 | res12$col.abs[, 2]/100 >= 0.1)
rownames(res12$col.abs[idx, ])

### Description

This function represents a simplified barchart adapted to display eigen values. The bar color depends on whether the axis is displayed, kept or not.

### Usage

```r
plotEig(eigvalue, nf, xax = 1, yax = 2, col.plot = "black", col.kept = "grey", col = "white", facets = NULL, plot = TRUE, storeData = FALSE, pos = -1,...)
```

### Arguments

- `eigvalue` a numeric vector of eigenvalues
- `nf` the number of retained factors, NULL if not provided
- `xax` an integer indicating which factor is plotted on the x-axis
- `yax` an integer indicating which factor is plotted on the y-axis
**plotEig**

- `col.plot` a color value to fill the bar corresponding to the displayed factors
- `col.kept` a color value to fill the bar corresponding to the kept by not displayed factors
- `col` a color value to fill the bar corresponding to the other factors
- `facets` a factor splitting the rows of `dfxy` so that subsets of the data are represented on different sub-graphics
- `plot` a logical indicating if the graphics is displayed
- `storeData` a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
- `pos` an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...` additional graphical parameters (see `adegpar` and `trellis.par.get`)

**Details**

Graphical parameters for bars are available in `ppolygons` of `adegpar`. Some appropriated graphical parameters in `p1d` are also available.

**Value**

An object of class ADEg (subclass `C1.barchart`). The result is displayed if `plot` is TRUE.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

**See Also**

`C1.barchart` ADEg.C1

**Examples**

```r
data(microsatt, package = "ade4")
w <- ade4::dudi.coa(data.frame(t(microsatt$tab)), scann = FALSE, nf = 3)
g1 <- s.label(w$co, plot = FALSE)
g2 <- plotEig(w$eig, w$nf, psub = list(text = "Eigenvalues"), pbackground = list(box = TRUE), plot = FALSE)
G <- insert(g2, g1, posi = "bottomright", ratio = 0.25)
```
prepare-methods

Methods prepare for ADEg objects

Description

The method `prepare` performs the first calculus needed for the display.

Methods

- `signature(object = "ADEg.C1")` performs the calculations before display the object (e.g. limits, grid and axis calculations)
- `signature(object = "C1.barchart")` calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default
- `signature(object = "C1.curve")` calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default
- `signature(object = "C1.density")` calls the parent method (prepare for ADEg.C1), modifies some graphical parameters used by default and calculates the density curves according to the numeric score and the values’ categories
- `signature(object = "C1.dotplot")` calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default
- `signature(object = "C1.gauss")` calls the parent method (prepare for ADEg.C1), modifies some graphical parameters used by default and calculates the Gauss curves according to the numeric score and the values’ categories (using weighted mean and standard deviation)
- `signature(object = "C1.hist")` calls the parent method (prepare for ADEg.C1), modifies some graphical parameters used by default and calculates the boundaries and the height of cells
- `signature(object = "C1.interval")` calls the parent method (prepare for ADEg.C1) and modifies some graphical parameters used by default
- `signature(object = "ADEg.S1")` performs the calculations before display the object (e.g. limits, grid and axis calculations)
- `signature(object = "S1.boxplot")` calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default
- `signature(object = "S1.class")` calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default
- `signature(object = "S1.distri")` calls the parent method (prepare for ADEg.S1), modifies some graphical parameters used by default and calculates weighted mean and standard deviation
- `signature(object = "S1.label")` calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default
- `signature(object = "S1.match")` calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default
- `signature(object = "ADEg.S2")` performs the calculations before display the object (e.g. limits, grid and axis calculations)
signature(object = "S2.arrow") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates limits

signature(object = "S2.class") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates ellipses, convex hulls and centroids

signature(object = "S2.corrcircle") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and prepares the drawn grid

signature(object = "S2.density") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates densities

signature(object = "S2.distri") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates ellipses and centroids

signature(object = "S2.image") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates grid expansion and limits

signature(object = "S2.label") calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default

signature(object = "S2.logo") calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default

signature(object = "S2.match") calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default

signature(object = "S2.trajec") calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default

signature(object = "S2.value") calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates limits

signature(object = "ADEg.T") performs the calculations before display the object (e.g. limits, grid and axis calculations)

signature(object = "T.image") calls the parent method (prepare for ADEg.T) and modifies some graphical parameters used by default and calculates limits and grid

signature(object = "T.value") calls the parent method (prepare for ADEg.T) and modifies some graphical parameters used by default and calculates limits and grid

signature(object = "ADEg.Tr") performs the calculations before display the object (e.g. limits, grid and axis calculations)

signature(object = "Tr.class") calls the parent method (prepare for ADEg.Tr), modifies some graphical parameters used by default and calculated ellipses, convex hulls and centroids

signature(object = "Tr.label") calls the parent method (prepare for ADEg.Tr) and modifies some graphical parameters used by default

signature(object = "Tr.match") calls the parent method (prepare for ADEg.Tr), modifies some graphical parameters used by default and defines the mean point and the axis

signature(object = "Tr.trajec") calls the parent method (prepare for ADEg.Tr) and modifies some graphical parameters used by default

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
s.arrow 2-D scatter plot with arrows

Description
This function represents a two dimensional scatter plot with arrows linking points to the origin.

Usage
```
s.arrow(dfx, xax = 1, yax = 2, labels = row.names(as.data.frame(dfx)), facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)```

Arguments
- `dfx`: a data frame used to produce the plot
- `xax`: an integer (or a vector) indicating which column(s) of `dfx` is(are) plotted on the x-axis
- `yax`: an integer (or a vector) indicating which column(s) of `dfx` is(are) plotted on the y-axis
- `labels`: a character vector containing labels for arrows
- `facets`: a factor splitting the rows of `dfx` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add`: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Details
An other origin for arrows can be specified using an `adegpar` parameters: `porigin`. Graphical parameters for points and arrows are available in `parrows` and `ppoints` of `adegpar`.

Value
An object of class ADEg (subclass S2.arrow) or ADEgS (if `add` is TRUE and/or if `facets` or vectors for `xax/yax` are used).

The result is displayed if `plot` is TRUE.
s.class

2-D scatter plot with a partition in classes (levels of a factor)

Description

This function represents a two dimensional scatter plot grouping points to the same class. Classes are represented by ellipses, stars and/or convex hulls.

Usage

\[
s.class(dfxy, fac, xax = 1, yax = 2, wt = rep(1, NROW(fac)), labels = levels(fac), ellipseSize = 1.5, starSize = 1, chullSize = NULL, col = NULL, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
\]

Arguments

dfxy a data frame used to produce the plot
fac a factor (or a matrix of factors) splitting the rows of dfxy
xax an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on the x-axis
yax an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on the y-axis
wt a vector of weights for fac
labels a character vector containing the class’ labels

Examples

data(doubs, package = "ade4")
dudi1 <- ade4::dudi.pca(doubs$env, scale = TRUE, scannf = FALSE, nf = 3)
dudi2 <- ade4::dudi.pca(doubs$fish, scale = TRUE, scannf = FALSE, nf = 2)
coin1 <- ade4::coinertia(dudi1, dudi2, scannf = FALSE, nf = 2)
g11 <- s.arrow(coin1$1, plabels.cex = 0.87, plot = FALSE)
g12 <- s.arrow(coin1$c1, plabels.cex = 1, plabels.col = "red", plot = FALSE)
g1 <- superpose(g12, g11, plot = TRUE)

xy <- cbind(rnorm(50), rnorm(50))
g2 <- s.arrow(xy, plabels.cex = 0.9, plines = list(lwd = 1.5), parrows.angle = 20)
update(g2, plines = list(col = rainbow(5)))

See Also

S2.arrow ADEg.S2
ellipseSize  a positive number for ellipse size
starSize    a number between 0 and 1 for the size of the stars segments joining the stars’
center (centroids) and the matching points
chullSize   NULL or a vector of numbers between 0 and 1 for the fraction of points included
            in the convex hull
col         a color or a colors vector to color points, ellipses, labels, lines and polygons
facets      a factor splitting the rows of dfxy so that subsets of the data are represented on
            different sub-graphics
plot        a logical indicating if the graphics is displayed
storeData   a logical indicating if the data should be stored in the returned object. If FALSE,
            only the names of the data arguments are stored
add         a logical. If TRUE, the graphic is superposed to the graphics already plotted in
            the current device
pos         an integer indicating the position of the environment where the data are stored,
            relative to the environment where the function is called. Useful only if storeData
            is FALSE
            ... additional graphical parameters (see adegpar and trellis.par.get)

Details

Graphical parameters for ellipses, stars and convex hulls are available in pellipses, plines and
ppolygons of adegpar.

Value

An object of class ADEg (subclass S2.class) or ADEgS (if add is TRUE and/or if facets or multidimen-
sional fac or vectors for xax/yax are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane
Dray

See Also

S2.class ADEg.S2

Examples

xy <- cbind.data.frame(x = runif(200, -1, 1), y = runif(200, -1, 1))
posi <- factor(xy$x > 0) : factor(xy$y > 0)
coul <- c("black", "red", "green", "blue")
s.class(xy, fac = posi, col = coul, psub.text = "example s.class", pellipses.col = coul)
s.class(xy, fac = posi, ppoints.cex = 1.5, ellipseSize = 0, starSize = 0,
        ppolygons = list(border = 4:1, col = 1:4, lty = 1:4, lwd = 2, alpha = 0.4),
s.corcircle

chullSize = c(1L, 0.5)

s.class(xy, fac = posi, facets = posi, ppoints.cex = 1.5, ellipseSize = 0, starSize = 0,
ppolygons = list(border = 4:1, col = 1:4, lty = 1:4, lwd = 2, alpha = 0.4),
chullSize = c(1L, 0.5))

## Not run:
s.class(xy, fac = posi, col = coul, psub.text = "example s.class", pellipses.col = coul,
plabels.cex = 0, key = list(space = "left"))

data(banque, package = "ade4")
dudi1 <- ade4::dudi.acm(banque, scannf = FALSE)
col <- rainbow(length(levels(banque[, 20])))
g1 <- s.label(dudi1$li, psub = list(text = "Factorial map from ACM", cex = 1.5,
position = "topleft"), plot = FALSE)
g2 <- s.class(dudi1$li, banque[, 20], psub = list(text = names(banque)[20], cex = 1.5,
position = "bottomright"), ellipseSize = 0, starSize = 0.5, pgrid.text.cex = 0, plot = FALSE)
g3 <- s.class(dudi1$li, banque[, 20], starSize = 0, ellipseSize = 2, pgrid.text.cex = 0,
plabels.cex = 1.5, plot = FALSE)
g4 <- s.class(dudi1$li, banque[, 20], psub = list(text = names(banque)[20],
position = "topright"), pgrid.text.cex = 0, col = col, pellipses.lwd = 1.5, plot = FALSE)
G1 <- ADEgS(c(g1, g2, g3, g4), layout = c(2, 2))
G2 <- s.class(dudi1$li, banque, psub = list(position = "topleft"), pgrid.text.cex = 0,
starSize = 0, ppoints.cex = 0)

## End(Not run)

---

**s.corcircle**  
*Correlation circle*

**Description**

This function produces a correlation circle.

**Usage**

\[
\text{s.corcircle}(dfxy, xax = 1, yax = 2, labels = row.names(as.data.frame(dfxy)),
fullcircle = TRUE, facets = NULL, plot = TRUE, storeData = TRUE,
add = FALSE, pos = -1, ...)\]

**Arguments**

- **dfxy**: a data frame used to produce the plot
- **labels**: a vector containing the points' labels
- **xax**: an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on
  the x-axis
- **yax**: an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on
  the y-axis
fullcircle  a logical to include the complete circle (limits are then c(-1, 1))
facets  a factor splitting the rows of dfxy so that subsets of the data are represented on different sub-graphics
plot  a logical indicating if the graphics is displayed
storeData  a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
add  a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
pos  an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
...
additional graphical parameters (see adepar and trellis.par.get)

Value

An object of class ADEg (subclass S2.corcircle) or ADEgS (if add is TRUE and/or if facets or vectors for xax/yax are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

S2.corcircle ADEg.S2

Examples

data (olympic, package = "ade4")
dudi1 <- ade4::dudi.pca(olympic$tab, scannf = FALSE)
g1 <- s.corcircle(dudi1$co)
g2 <- s.corcircle(dudi1$co, fullcircle = FALSE, pback.col = "grey")

---

s.density  2-D scatter plot with kernel density estimation

Description

This function represents a two dimensional scatter plot of points distribution. Densities’ representation is based on the levelplot graphic in lattice (density’s surface, filled with colors and/or contour lines).
s.density

Usage

s.density(dfxy, xax = 1, yax = 2, bandwidth = NULL, gridsize = c(450L, 450L),
          nrpoints = 300, threshold = 0.1, col = NULL, contour = FALSE, region = !contour,
          nclass = 8, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

dfxy a data frame used to produce the plot
xax an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on
      the x-axis
yax an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on
      the y-axis
bandwidth bandwidth for density calculations which is passed in parameters in the bkde2d
      function of the KernSmooth package
gridsize grid dimension
nrpoints number of points on the density image
threshold a value between 0 and 1 to draw densities greater than this threshold. No density
          is visible whether it is equal to 1
col a color or a colors vector to color densities
contour a logical to draw contour lines
region a logical to fill grid regions with col
nclass number of class for density
facets a factor splitting the rows of dfxy so that subsets of the data are represented on
        different sub-graphics
plot a logical indicating if the graphics is displayed
storeData a logical indicating if the data should be stored in the returned object. If FALSE,
           only the names of the data arguments are stored
add a logical. If TRUE, the graphic is superposed to the graphics already plotted in
     the current device
pos an integer indicating the position of the environment where the data are stored,
     relative to the environment where the function is called. Useful only if storeData
     is FALSE

... additional graphical parameters (see adegpar and trellis.par.get)

Details

Density calculation is made using the kde2d function of the KernSmooth package.

Value

An object of class ADEg (subclass S2.density) or ADEgS (if add is TRUE and/or if facets or vectors
for xax/yax are used).

The result is displayed if plot is TRUE.
Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also
S2.density ADEg.S2

Examples
xx2 <- c(rnorm(50000, 1, 1), rnorm(50000, -1, 1))
yy2 <- c(rnorm(50000, -1, 0.5), rnorm(50000, 1, 0.5))
s.density(cbind(xx2, yy2), paxes.draw = TRUE, gridsize = c(200, 200), region = TRUE,
contour = TRUE, plabels.cex = 0, threshold = 0.05, nclass = 3,
col = colorRampPalette(c("lightgrey", "black"))(100))

s.distri 2-D scatter plot with means/standard deviations computed using an external table of weights

Description
This function represents a two dimensional scatter plot of a frequency distribution. Class are defined by ellipses and/or stars.

Usage
s.distri(dfxy, dfdistri, xax = 1, yax = 2, starSize = 1,
ellipseSize = 1.5, col = NULL, facets = NULL, plot = TRUE,
storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments
dfxy a data frame used to produce the plot
dfdistri a data frame containing the mass distribution in columns
xax an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on the x-axis
yax an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on the y-axis
starSize NULL or number between 0 and 1 for the size of the stars segments joining the stars’ center (centroids) and the matching points
ellipseSize NULL or number between 0 and 1 for ellipse size
col a color or a colors vector to color points, ellipses, labels, lines and polygons
facets a factor splitting the rows of dfxy so that subsets of the data are represented on different sub-graphics
s.image

plot  a logical indicating if the graphics is displayed
storeData  a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
add  a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
pos  an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
...  additional graphical parameters (see adegpar and trellis.par.get)

Value

An object of class ADEg (subclass S2.distri) or ADEgS (if add is TRUE and/or if facets or vectors for xax/yax are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

S2.distri ADEg.S2

Examples

data(rpjdl, package = "ade4")
xy <- ade4::dudi.coa(rpjdl$fau, scan = FALSE)$li
j <- c(1, 5, 8, 20, 21, 23, 26, 33, 36, 44, 47, 49)
dfdistri <- rpjdl$fau[, j]
coli <- colorRampPalette(c("blue", "red", "orange"))(49)[j]
s.distri(xy, dfdistri, ellipseSize = 1, starSize = 0, porigin.include = FALSE, pellipses = list(col = coli, alpha = 0.3), plabels.cex = 0)

2-D scatter plot with loess estimation of an additional numeric score (levelplot)

Description

This function represents a two dimensional scatter plot with a continuous convex colored surface and/or contour lines representing a third variable.
Usage

`s.image(dfxy, z, xax = 1, yax = 2, span = 0.5, gridsize = c(80L, 80L),
  contour = TRUE, region = TRUE, outsideLimits = NULL, breaks = NULL,
  nclass = 8, col = NULL, facets = NULL,
  plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)`

Arguments

- `dfxy`: a data frame used to produce the plot
- `z`: a vector (or a matrix) of values on the `dfxy` rows
- `xax`: an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the x-axis
- `yax`: an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the y-axis
- `span`: a value to control the degree of smoothing
- `gridsize`: a 1 or 2-length vector indicating the cell numbers (horizontally and vertically) of the grid for the colored surface
- `contour`: a logical to draw contour lines
- `region`: a logical to fill inter-contour regions
- `breaks`: a vector of values to split `z`. If `NULL`, `pretty(z, nclass)` is used.
- `nclass`: an integer for the number of desired intervals, ignored if `breaks` is not missing.
- `outsideLimits`: specific limits for the surface as a set of polygons. It must be an `SpatialPolygons` object. Hole are authorized.
- `col`: a color or a colors vector used for the colored cells
- `facets`: a factor splitting the rows of `dfxy` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data should be stored in the returned object. If `FALSE`, only the names of the data arguments are stored
- `add`: a logical. If `TRUE`, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is `FALSE`
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Value

An object of class `ADEg` (subclass `S2.image`) or `ADEgS` (if `add` is `TRUE` and/or if `facets` or multidimensional `z` or vectors for `xax/yax` are used).

The result is displayed if `plot` is `TRUE`.
Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also
S2.image ADEg.S2

Examples

```r
df1 <- data.frame(expand.grid(-3:3, -3:3))
names(df1) <- c("x", "y")
z1 <- (1 / sqrt(2)) * exp(-((df1$x ^ 2 + df1$y ^ 2) / 2))
g1 <- s.image(df1, z1)

# add a continuous color bar as legend
# update(g1, plegend.drawColorKey = TRUE)

# Not run:
if(require(splancs, quietly = TRUE) & require(sp, quietly = TRUE)) {
  Sr1 <- Polygon(cbind(c(0, 1, 2, 1, 2, 0, -2, -1, -2, -1, 0),
      c(2.5, 1.5, 2, 0, -2, -1, -2, 0, 2, 1.5, 2.5))
  Sr2 <- Polygon(cbind(c(-0.5, 0.5, 0.5, -0.5, -0.5), c(0, 0, 1, 1, 0)), hole = TRUE)
  Srs2 <- Polygons(list(Sr1, Sr2), ID = "star and hole")
  SPP <- SpatialPolygons(list(Srs2))
  df2 <- cbind(c(rnorm(2000, 1, 0.25), rnorm(3000, -1, 1.5)), c(rnorm(2000, 1, 0.5),
      rnorm(3000, -1, 3)))
  z2 <- c(rnorm(2000, 12, 1), rnorm(3000, 1, 2))
  g5 <- s.image(df2, z2, outsidelimits = SPP, grid = 200, xlim = c(-2.5, 2.5),
      ylim = c(-2, 3), ppalette.quanti = colorRampPalette(c(grey(0.1), grey(0.9))))
}

## End(Not run)
```

\( s.label \)

2-D scatter plot with labels

Description
This function represents a two dimensional scatter plot associating labels with points.
Usage

`s.label(dfxy, labels = rownames(dfxy), xax = 1, yax = 2, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)`

Arguments

- `dfxy`: a data frame used to produce the plot
- `labels`: a vector of character strings for the points’ labels
- `xax`: an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the x-axis
- `yax`: an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the y-axis
- `facets`: a factor splitting the rows of `dfxy` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add`: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Value

An object of class `ADEg` (subclass `S2.label`) or `ADEg$` (if `add` is TRUE and/or if facets or vectors for `xax/yax` are used).
The result is displayed if `plot` is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Stephane Dray

See Also

`S2.label` `ADEg.S2`

Examples

```r
x0 <- runif(50, -2, 2)
y0 <- runif(50, -2, 2)
z <- x0 ^ 2 + y0 ^ 2
g1 <- s.label(data.frame(x0, y0), label = as.character(z < 1), paxes.draw = TRUE, axis.text = list(col = "grey"))```
data(mafragh, package = "ade4")
g2 <- s.label(mafragh$xy, nb = mafragh$nb, paxes.draw = FALSE)

data(irishdata, package = "ade4")
g3 <- s.label(irishdata$xy.utm, Sp = irishdata$Spatial.contour)
## update irishdata$xy.utm call to irishdata$xy

## Not run: data(atlas, package = "ade4")
g4 <- s.label(atlas$xy, lab = atlas$names.district, Sp = atlas$Spatial.contour)
g5 <- s.label(atlas$xy, lab = atlas$names.district, Sp = atlas$Spatial)

## End(Not run)

s.logo 2-D scatter plot with logos (bitmap objects)

Description
This function represents a two dimensional scatter plot associating logos with points.

Usage
s.logo(dfxy, logos, xax = 1, yax = 2, facets = NULL,
plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments
dfxy  a data frame used to produce the plot
logos   a list containing the picture to use for each point
xax  an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on

the x-axis

yax  an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on

the y-axis

facets a factor splitting the rows of dfxy so that subsets of the data are represented on

different sub-graphics

plot  a logical indicating if the graphics is displayed
storeData a logical indicating if the data should be stored in the returned object. If FALSE,

only the names of the data arguments are stored
add a logical. If TRUE, the graphic is superposed to the graphics already plotted in

the current device

pos  an integer indicating the position of the environment where the data are stored,

relative to the environment where the function is called. Useful only if storeData

is FALSE

... additional graphical parameters (see adegpar and trellis.par.get)
Value

An object of class ADEg (subclass S2.logo) or ADEgS (if add is TRUE and/or if facets or vectors for xax/yax are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

S2.logo ADEg S2

Examples

data(ggtortoises, package = "ade4")
g1 <- s.match(ggtortoises$pop, 
ggtortoises$ico[as.character(ggtortoises$pop$carap)], 
pori.incl = FALSE, ppoints.cex = 0.5)
g1 <- s.match(ggtortoises$pop, add = TRUE, plabels.boxes.alpha = 0)

g2 <- s.match(ggtortoises$misc, pgrid.draw = FALSE, 
porigin.include = FALSE, paxes.draw = FALSE, 
Sp = ggtortoises$Spatial, pback.col = "lightblue", pSp.col = "white")
g2 <- s.match(ggtortoises$pop, ggtortoises$ico[as.character(ggtortoises$pop$carap)], 
ppoints.cex = 0.5, add = TRUE)

data(capitales, package = "ade4")
g3 <- s.match(capitales$xy[sort(rownames(capitales$xy))], capitales$logo, 
Sp = capitales$Spatial, pback.col = "lightblue", pSp.col = "white", 
pgrid.draw = FALSE)

Description

This function represents a two dimensional scatter plot linking paired coordinates.

Usage

s.match(dfxy1, dfxy2, xax = 1, yax = 2, labels = 
row.names(as.data.frame(dfxy1)), arrows = TRUE, 
facets = NULL, plot = TRUE, storeData = TRUE, 
add = FALSE, pos = -1, ...)
**Arguments**

- `dfxy1` a data frame, the first system of coordinates, used to produce the plot
- `dfxy2` a data frame, the second system of coordinates, with as many rows as `dfxy1`, used to produce the plot.
- `labels` a vector of character strings containing the matches’ labels
- `xax` an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the x-axis
- `yax` an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the y-axis
- `arrows` a logical to draw arrows
- `facets` a factor splitting the rows of `dfxy` so that subsets of the data are represented on different sub-graphics
- `plot` a logical indicating if the graphics is displayed
- `storeData` a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add` a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos` an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...` additional graphical parameters (see `adegpar` and `trellis.par.get`)

**Value**

An object of class ADEg (subclass S2.match) or ADEgS (if `add` is TRUE and/or if `facets` or `xax/yax` are used).

The result is displayed if `plot` is TRUE.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

**See Also**

`S2.match ADEg.S2`

**Examples**

```r
X <- data.frame(x = runif(50, -1, 2), y = runif(50, -1, 2))
Y <- X + rnorm(100, sd = 0.3)
g1 <- s.match(X, Y, arr = TRUE, ppoints.cex = 2, ppoints.col = c("blue", "green"))
data(doubs, package = "ade4")
dudi1 <- ade4::dudi.pca(doubs$env, scale = TRUE, scannf = FALSE, nf = 3)
dudi2 <- ade4::dudi.pca(doubs$fish, scale = FALSE, scannf = FALSE, nf = 2)
```
s.Spatial

Mapping of a Spatial* object

Description

This function represents a background map linked with data or not.

Usage

s.Spatial(spobj, col = TRUE, nclass = 5, scale = TRUE, plot = TRUE, storeData = TRUE, pos = -1, ...)

Arguments

spobj: an object deriving from class Spatial (package sp)
col: a logical or a color to fill the background color of spobj
nclass: if spobj contains data, the desired number of intervals splitting the data (using pretty)
scale: a logical indicating if numeric variables should be scaled
plot: a logical indicating if the graphics is displayed
storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
pos: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
...
additional graphical parameters (see adegpar and trellis.par.get)

Value

An object of class ADEg (subclass S2.label) or ADEgS (if spobj contains more than one column). The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

S2.label spplot sp.lines sp.polygons sp.grid
Examples

data(elec88, package = "ade4")
## mapping without data
obj <- s.Spatial(elec88$Spatial)
## Not run:
if(require(sp, quietly = TRUE)) {
## mapping with data
obj <- SpatialPolygonsDataFrame(Sr = elec88$Spatial, data = elec88$tab)

}
## End(Not run)

s.traject 2-D scatter plot with trajectories

Description

This function represents a two dimensional scatter plot with trajectories.

Usage

s.traject(dfxy, fac = gl(1, nrow(dfxy)), order, labels = levels(fac),
xax = 1, yax = 2, col = NULL, facets = NULL, plot = TRUE,
storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

dfxy  a data frame used to produce the plot
fac  a factor (or a matrix of factors) splitting the rows of dfxy
order  a vector containing the drawing order of the trajectories. A vector of length equal to factor.
lables  a vector of character strings containing the trajectories’ labels
xax  an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on the x-axis
yax  an integer (or a vector) indicating which column(s) of dfxy is(are) plotted on the y-axis
col  a color or a colors vector to color points, labels and lines
facets  a factor splitting the rows of dfxy so that subsets of the data are represented on different sub-graphics
plot  a logical indicating if the graphics is displayed
storeData  a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
add a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device.

pos an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE.

... additional graphical parameters (see adegpar and trellis.par.get)

Details

The fac factor is used to display several trajectories: each level of fac is a specific trajectory.

Value

An object of class ADEg (subclass S2.trajec) or ADEgS (if add is TRUE and/or if facets or multidimensional fac or vectors for xax/yax are used). The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

S2.trajec ADEg.S2

Examples

```r
rw <- function(a) {
  x <- 0
  for(i in 1:49) x <- c(x, x[length(x)] + runif(1, -1, 1))
  x
}
x1 <- unlist(lapply(1:5, rw), use.names = FALSE)
y1 <- unlist(lapply(1:5, rw), use.names = FALSE)
z1 <- gl(5, 50)
g1 <- s.trajec(data.frame(x1, y1), z1, ppoints.pch = 19:23, plines.col = rainbow(5))
x2 <- unlist(lapply(1:2, rw), use.names = FALSE)
y2 <- unlist(lapply(1:2, rw), use.names = FALSE)
z2 <- gl(2, 50)
g2 <- s.trajec(data.frame(x2, y2), z2, ppoints.pch = 21:20, plines.col = 1:2)
```
s.value

| 2-D scatter plot with proportional symbols (bubble plot) |

**Description**

This function represents a two dimensional scatter plot with a third value represented by symbols.

**Usage**

```r
s.value(dfxy, z, breaks = NULL, xax = 1, yax = 2, method = c("size", "color"), symbol = c("square", "circle", "diamond", "uptriangle", "downtriangle"), col = NULL, nclass = 4, center = 0, centerpar = NULL, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
```

**Arguments**

- `dfxy`: a data frame used to produce the plot
- `z`: a vector (or a matrix) with as many values as rows in `dfxy`
- `breaks`: a vector containing the breaks used for splitting `z` value. If `NULL`, `pretty(z, n)` is used.
- `xax`: an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the x-axis
- `yax`: an integer (or a vector) indicating which column(s) of `dfxy` is(are) plotted on the y-axis
- `method`: color or size value for represent `z`. If color, a palette of color is used for the symbols (one color per interval). If size, symbols of proportional area are used. Area is 0 for values equals to center (default 0). Two colors are used, for values less than center and larger than center.
- `symbol`: value for symbol type
- `col`: a color or a colors vector to color symbols. If `method` is size, a 2-length vector of color is expected. If `method` is color, it must have as many colors as the number of class.
- `nclass`: an integer for the number of desired intervals, ignored if `breaks` is not missing.
- `center`: a center value for method size
- `centerpar`: a logical or a list to represent center value using elements in the `adegpar("ppoints")` list
- `facets`: a factor splitting the rows of `dfxy` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data should be stored in the returned object. If `FALSE`, only the names of the data arguments are stored
- `add`: a logical. If `TRUE`, the graphic is superposed to the graphics already plotted in the current device
pos an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

... additional graphical parameters (see adepar and trellis.par.get)

Value

An object of class ADEg (subclass S2.value) or ADEgS (if add is TRUE and/or if facets or multidimensional z or vectors for xax/yax are used). The result is displayed if plot is TRUE.

Note

For the symbol size, if the method is size, we use perceptual scaling (Tanimura et al. 2006).

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References


See Also

S2.value ADEg.S2

Examples

data(rpjdl, package = "ade4")
fau.coa <- ade4::dudi.coa(rpjdl$fau, scan = FALSE, nf = 3)
g1 <- s.value(fau.coa$li, fau.coa$li[,3])
update(g1, key = list(space = "right", columns = 1))
g2 <- s.value(fau.coa$li, fau.coa$li[,3], method = "color", plegend.size = 0.8)
g3 <- s.value(fau.coa$li, fau.coa$li[,3], plegend.size = 0.8, symbol = "square",
method = "color", col = colorRampPalette(c("yellow", "blue"))(6))
g4 <- s.value(fau.coa$li, fau.coa$li[,3], plot = FALSE)
g5 <- s.value(fau.coa$li, fau.coa$li[,3], center = 0, method = "size",
symbol = "circle", col = c("yellow", "red"), plot = FALSE)
g6 <- ADEgS(c(g4, g5), positions = layout2position(matrix(c(1, 2), 1, 2)),
add = matrix(0, ncol = 2, nrow = 2))

data(irishdata, package = "ade4")
irq0 <- data.frame(scale(irishdata$tab, scale = TRUE))
g7 <- s.value(irishdata$xxy.utm, irq0, Sp = irishdata$Spatial.contour, paxes.draw = FALSE,
pgrid.draw = FALSE, pSp.alpha = 0.4)
Description

A class for the representation of the link between a variable and a qualitative variable using box-and-whisker plots.

Objects from the Class

S1.boxplot objects can be created by calls of the form new("S1.boxplot", ...).

The regular usage in this package is to use the s1d.boxplot function.

Slots

data a list containing data or data’s name.

• score: the displayed values in the form of a numeric vector, a name or a matching call.
• fac: a factor for score splitting in the form of a vector, a factor, a name or a matching call.
• at: the index value.
• frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
• storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of ADEg.S1 class.

The specific slot for S1.boxplot objects is:

• col: a NULL value, a color or a colors vector to color points, labels, lines and polygons.

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters

Call an object of class call

Extends

Class ADEg.S1, directly.
Class ADEg, by class ADEg, S1, distance 2.
Class ADEgORTrellis, by class ADEg, S1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg, S1, distance 3.
Methods

The methods of the father classes "ADEg.S1" and "ADEg" can be used by inheritance. The specific methods for S1.boxplot are:

**prepare** signature(object = "S1.boxplot"): calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default.

**panel** signature(object = "S1.boxplot"): draws box-and-wiskers diagrams, mean points and labels.

**setlatticecall** signature(object = "S1.boxplot"): prepares the lattice.call slot

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S1 s1d.boxplot

Examples

```
showClass("S1.boxplot")
```

---

## S1.class-class

### Description

A class for the creation and display of a numeric score aggregated in class by an associated factor.

### Objects from the Class

S1.class objects can be created by calls of the form `new("S1.class", ...)`.

The regular usage in this package is to use the `s1d.class` function.

### Slots

- **data**: a list containing data or data's name.
  - **score**: the displayed values in the form of a numeric vector, a name or a matching call.
  - **fac**: a factor for score splitting in the form of a vector, a factor, a name or a matching call.
  - **wt**: a vector of weights for score
  - **labels**: the labels' names drawn for each class.
  - **at**: the index value.
S1.class-class

- frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
- storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of ADEg.S1 class.

The specific slots for S1.class objects are:

- col: a NULL value, a color or a colors vector to color points, labels and lines.
- poslabel: the label position of each class, it can be regular or value.

stats a list of internal preliminary calculations. The specific slot for S1.class objects is:

- means: the weighted mean calculated for each fac value.

s.misc a list of some others internal parameters. The specific slot for S1.class objects is:

- rug: an index value indicating where the rugs are drawn.

Call an object of class call

Extends

Class ADEg.S1, directly.
Class ADEg, by class ADEg.S1, distance 2.
Class ADEgORTrellis, by class ADEg.S1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.S1, distance 3.

Methods

The methods of the father classes "ADEg.S1" and "ADEg" can be used by inheritance. The specific methods for S1.class are:

prepare signature(object = "S1.class"): calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default.

panel signature(object = "S1.class"): draws labels and lines matching with score values.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S1 s1d.class

Examples

showClass("S1.class")
**S1.distri-class**

**Description**

A class for the representation of a set of distributions on a numeric score.

**Objects from the Class**

`S1.distri` objects can be created by calls of the form `new("S1.distri", ...)`. The regular usage in this package is to use the `s1d.distri` function.

**Slots**

- `data`: a list containing data or data's name.
- `score`: the displayed values in the form of a numeric vector, a name or a matching call.
- `dfdistri`: the mass distribution in which each column is a class.
- `labels`: the labels' names drawn for each distribution.
- `at`: the index value.
- `frame`: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
- `storeData`: a logical indicating if the data should be stored in the returned object. If `FALSE`, only the names of the data arguments are stored.
- `trellis.par`: a list of parameters for `lattice` call. It will be passed directly to `par.settings` arguments of the `lattice` function.
- `adeg.par`: a list of graphical parameters, corresponding to the ones given by `adegpar()` function.
- `lattice.call`: a list to create the trellis object.
- `g.args`: a list containing some method parameters linked with the created object of `ADEg.S1` class.

The specific slots for `S1.distri` objects are:

- `sdSize`: the size of the standard deviation segments.
- `yrank`: a logical to draw the distributions sorted by means ascending order.

**stats**

A list of internal preliminary calculations. The specific slots for `S1.distri` objects are:

- `means`: the weighted mean calculated for each distribution.
- `sds`: the weighted variance calculated for each distribution.

**s.misc**

A list of some others internal parameters

**Call**

An object of class `call`

**Extends**

Class `ADEg.S1`, directly.
Class `ADEg`, by class `ADEg.S1`, distance 2.
Class `ADEgORtrellis`, by class `ADEg.S1`, distance 3.
Class `ADEgORADEgORtrellis`, by class `ADEg.S1`, distance 3.
Methods
The methods of the father classes "ADEg.S1" and "ADEg" can be used by inheritance. The specific methods for S1.distri are:

**prepare** signature(object = "S1.distri"): calls the parent method (prepare for ADEg.S1), modifies some graphical parameters used by default and calculates weighted mean and standard deviation.

**panel** signature(object = "S1.distri"): draws mean points and segments with matching labels.

Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also
ADEg ADEg.S1 s1d.distri

Examples
```
showClass("S1.distri")
```

Description
A class for the creation and display of a numeric score with labels.

Objects from the Class
S1.label objects can be created by calls of the form new("S1.label", ...). The regular usage in this package is to use the s1d.label function.

Slots
data a list containing data or data’s name.

- score: the displayed values in the form of a numeric vector, a name or a matching call.
- labels: the labels’ names drawn for each score value.
- at: the index value.
- frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
- storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.
trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of adeg.S1 class.

The specific slot for S1.class objects is:

• poslabel: the label position of each score value, it can be "regular" or "value".

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters. The specific slot for S1.label objects is:

• rug: an index value indicating where the rugs are drawn.

Call an object of class call

Extends

Class ADEg.S1, directly.
Class ADEg, by class ADEg.S1, distance 2.
Class ADEgORTrellis, by class ADEg.S1, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.S1, distance 3.

Methods

The methods of the father classes "ADEg.S1" and "ADEg" can be used by inheritance. The specific methods for S1.label are:

prepare signature(object = "S1.label"): calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default.

panel signature(object = "S1.label"): draws labels and its links with score points.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S1 s1d.label

Examples

showClass("S1.label")
Description

A class for the creation and display of paired scores.

Objects from the Class

S1.match objects can be created by calls of the form `new("S1.match", ...)`. The regular usage in this package is to use the `s1d.match` function.

Slots

data  a list containing data or data's name.
  • score: the displayed values in the form of a numeric vector, a name or a matching call.
  • labels: the labels' names drawn for each score.
  • at: the index value.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(...) env = data$frame`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par  a list of parameters for `lattice` call. It will be passed directly to `par.settings` arguments of the `lattice` function.

adeg.par  a list of graphical parameters, corresponding to the ones given by `adegpar()` function.

lattice.call  a list to create the `trellis` object.

g.args  a list containing some method parameters linked with the created object of `ADEg.S1` class.

stats  a list of internal preliminary calculations

s.misc  a list of some others internal parameters. The specific slot for `S1.match` objects is:
  • rug: an index value indicating where the rugs are drawn.

Call  an object of class call

Extends

Class `ADEg.S1`, directly.
Class `ADEg`, by class `ADEg.S1`, distance 2.
Class `ADEgORtrellis`, by class `ADEg.S1`, distance 3.
Class `ADEgORADeGOSORtrellis`, by class `ADEg.S1`, distance 3.
Methods

The methods of the father classes "ADEg.S1" and "ADEg" can be used by inheritance. The specific methods for S1.match are:

**prepare** signature(object = "S1.match"): calls the parent method (prepare for ADEg.S1) and modifies some graphical parameters used by default.

**panel** signature(object = "S1.match"): draws score points and matching segments and labels.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S1 s1d.match

Examples

showClass("S1.match")

---

### s1d.barchart

**1-D plot of a numeric score by bars**

Description

This function represents a score using a chart with rectangular bars for which length is proportional to this score.

Usage

s1d.barchart(score, labels = NULL, at = 1:NROW(score), facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

- **score**: a numeric vector (or a data frame) used to produce the plot
- **labels**: the labels' names drawn on the top of bars
- **at**: a numeric vector used as an index
- **facets**: a factor splitting score so that subsets of the data are represented on different sub-graphics
- **plot**: a logical indicating if the graphics is displayed
- **storeData**: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- **add**: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
s1d.boxplot

pos

an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

... additional graphical parameters (see adegpar and trellis.par.get)

Details

Graphical parameters for bars are available in ppolygons of adegpar. Some appropriated graphical parameters in p1d are also available.

Value

An object of class ADEg (subclass C1.barchart) or ADEgS (if add is TRUE and/or if facets or data frame for score are used).

The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

C1.barchart ADEg.C1

Examples

data(rpjdl, package = "ade4")
rpjdl.coa <- ade4::dudi.coa(rpjdl$fau, scannf = FALSE, nf = 4)
s1d.barchart(rpjdl.coa$eig, p1d.horizontal = FALSE, ppolygons.col = "grey")

s1d.boxplot

1-D box plot of a numeric score partitioned in classes (levels of a factor)

Description

This function represents the link between a variable and a set of qualitative variables using box-and-whisker plots.

Usage

s1d.boxplot(score, fac = gl(1, NROW(score)), at = 1:nlevels(fac), col = NULL,
facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)


Arguments

- `score`: a numeric vector (or a data frame) used to produce the plot
- `fac`: a factor (or a matrix of factors) to split `score`
- `at`: a numeric vector used as an index
- `col`: a color or a colors vector for points, labels, lines and polygons according to their factor level. Colors are recycled whether there are not one color by factor level.
- `facets`: a factor splitting `score` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add`: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...`: additional graphical parameters (see `adepar` and `trellis.par.get`)

Details

Graphical parameters for rugs are available in `plines` of `adepar` and the ones for boxes in `ppolygons`. Some appropriated graphical parameters in `p1d` are also available.

Value

An object of class `ADEg` (subclass `S1.boxplot`) or `ADEgS` (if `add` is TRUE and/or if `facets` or data frame for `score` or data frame for `fac` are used).

The result is displayed if `plot` is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

`S1.boxplot` `ADEg.S1`

Examples

data(banque, package = "ade4")
bande.acm <- ade4::dudi.acm(banque, scan = FALSE, nf = 4)
s1d.boxplot(bande.acm$li[, 1], banque[, 2], psub.text = names(banque)[2], psub.position = "topleft", col = c("red", "blue", "green", "purple", "orange"))
s1d.boxplot(bande.acm$li[, 1], banque[, 1:6], psub.position = "topleft")
1-D plot of a numeric score partitioned in classes (levels of a factor)

Description
This function represents the link between scores values and their matching labeled classes.

Usage
```r
sld.class(score, fac, wt = rep(1, NROW(fac)), labels = levels(fac), at = 0.5,
poslabel = c("regular", "value"), col = NULL, facets = NULL, plot = TRUE,
storeData = TRUE, add = FALSE, pos = -1, ...)
```

Arguments
- **score**: a numeric vector (or a data frame) used to produce the plot
- **fac**: a factor (or a matrix of factors) to split score
- **wt**: a vector of weights for score
- **labels**: the labels' names drawn for each class
- **at**: a numeric vector used as an index
- **poslabel**: the label position of each class (each level of fac), it can be regular or value. If regular, labels are evenly spaced. If value, labels are placed on the weighted mean of their class.
- **col**: a color or a colors vector for points, labels and lines according to their factor level. Colors are recycled whether there are not one color by factor level.
- **facets**: a factor splitting score so that subsets of the data are represented on different sub-graphics
- **plot**: a logical indicating if the graphics is displayed
- **storeData**: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- **add**: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- **pos**: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
- **...**: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Details
The weighted means of class are available in the object slot stats using `object@stats$means`. Graphical parameters for rugs are available in `p1d` of `adegpar`. Some appropriated graphical parameters in `p1d` are also available.
sld.curve

1-D plot of a numeric score linked by curves

Description

This function represents a score using points linked by curves.

Usage

sld.curve(score, at = 1:NROW(score), facets = NULL, plot = TRUE,
storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>score</td>
<td>a numeric vector (or a data frame) used to produce the plot</td>
</tr>
<tr>
<td>at</td>
<td>a numeric vector used as an index</td>
</tr>
<tr>
<td>facets</td>
<td>a factor splitting score so that subsets of the data are represented on different sub-graphics</td>
</tr>
<tr>
<td>plot</td>
<td>a logical indicating if the graphics is displayed</td>
</tr>
</tbody>
</table>
storeData  a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored.

add  a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device.

pos  an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE.

...  additional graphical parameters (see adegpar and trellis.par.get).

Details

Graphical parameters for lines and points are available in p.lines and in p.points of adegpar. Some appropriated graphical parameters in p1d are also available.

Value

An object of class ADEg (subclass C1.curve) or ADEgS (if add is TRUE and/or if facets or data frame for score are used). The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray.

See Also

c1.curve ADEg.C1

Examples

data(rpjdl, package = "ade4")
rpjdl.coa <- ade4::dudi.coa(rpjdl$fau, scannf = FALSE, nf = 4)
s1d.curve(rpjdl.coa$eig)

set.seed(40)
score1 <- rnorm(10)
s1d.curve(score1)

s1d.curves  1-D plot of multiple scores linked by curves

Description

This function represents multiple scores using points linked by curves.
s1d.curves

Usage

s1d.curves(score, at = 1:NCOL(score), facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1,...)

Arguments

score a numeric matrix (or a data frame) used to produce the plot
at a numeric vector used as an index
facets a factor splitting score so that subsets of the data are represented on different sub-graphics
plot a logical indicating if the graphics is displayed
storeData a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
add a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
pos an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
... additional graphical parameters (see adegpar and trellis.par.get)

Details

Graphical parameters for lines and points are available in p1lines and in ppoints of adegpar. Some appropriated graphical parameters in p1d are also available.

Value

An object of class ADEg (subclass C1.curves) or ADEg$ (if add is TRUE and/or if facets are used). The result is displayed if plot is TRUE.

Author(s)

Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

C1.curves ADEg.C1

Examples

scores <- matrix(1:50, nrow = 10)
s1d.curves(scores)
s1d.density

1-D plot of a numeric score by density curves

Description

This function represents a score with a density curve for each level of a factor.

Usage

s1d.density(score, fac = gl(1, NROW(score)), kernel = c("normal", "box", "epanech", "biweight", "triweight"), bandwidth = NULL, gridsize = 450, col = NULL, fill = TRUE, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

- `score`: a numeric vector (or a data frame) used to produce the plot
- `fac`: a factor (or a matrix of factors) to split score
- `kernel`: the smoothing kernel used, see `bkde`
- `bandwidth`: the kernel bandwidth smoothing parameter
- `gridsize`: the number of equally spaced points at which to estimate the density
- `col`: a logical, a color or a colors vector for labels, rugs, lines and polygons according to their factor level. Colors are recycled whether there are not one color by factor level.
- `fill`: a logical to yield the polygons density curves filled
- `facets`: a factor splitting score so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add`: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Details

Kernel, bandwidth and gridsize are passed as parameters to `bkde` function of the KernSmooth package.

Graphical parameters for rugs are available in `plines` of `adegpar` and the ones for density curves filled in `ppolygons`. Some appropriated graphical parameters in `p1d` are also available.
An object of class ADEg (subclass C1.density) or ADEgS (if add is TRUE and/or if facets or data frame for score or data frame for fac are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

C1.density ADEg.C1

Examples

```r
score <- c(rnorm(1000, mean = -0.5, sd = 0.5), rnorm(1000, mean = 1))
fac <- rep(c("A", "B"), each = 1000)
s1d.density(score, fac, col = c(2, 4), p1d.reverse = TRUE)
```

s1d.distri  1-D plot of a numeric score by means/standard deviations computed using an external table of weights

Description

This function represents a set of distributions on a numeric score using a mean-standard deviation display.

Usage

```r
s1d.distri(score, dfdistri, labels = colnames(dfdistri), at = 1:NCOL(dfdistri),
yrank = TRUE, sdSize = 1, facets = NULL, plot = TRUE,
storeData = TRUE, add = FALSE, pos = -1, ...)
```

Arguments

- `score`: a numeric vector (or a data frame) used to produce the plot
- `dfdistri`: a data frame containing the mass distribution in which each column is a class
- `yrank`: a logical to draw the distributions sorted by means ascending order
- `labels`: the labels’ names drawn for each distribution
- `at`: a numeric vector used as an index
- `sdSize`: a numeric for the size of the standard deviation segments
- `facets`: a factor splitting score so that subsets of the data are represented on different sub-graphics
**plot** a logical indicating if the graphics is displayed

**storeData** a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored

**add** a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device

**pos** an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if **storeData** is FALSE

... additional graphical parameters (see adegpar and trellis.par.get)

**Details**

Graphical parameters for rugs are available in plines of adegpar. Some appropriated graphical parameters in p1d are also available. The weighted means and standard deviations of class are available in the object slot stats using object@stats$means and object@stats$sds.

**Value**

An object of class ADEg (subclass S1.distri) or ADEgS (if add is TRUE and/or if facets or data frame for score are used).

The result is displayed if **plot** is TRUE.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot \(<aurelie.siberchicot@univ-lyon1.fr>\) and Stephane Dray

**See Also**

S1.distri ADEg.S1

**Examples**

```r
w <- seq(-1, 1, le = 200)
distri <- data.frame(lapply(1:50,
  function(x) sample(200:1) * ((w >= (- x / 50)) & (w <= x / 50))))
names(distri) <- paste("w", 1:50, sep = "")
g11 <- s1d.distri(w, distri, yrank = TRUE, sds = 1.5, plot = FALSE)
g12 <- s1d.distri(w, distri, yrank = FALSE, sds = 1.5, plot = FALSE)
G1 <- ADEgS(c(g11, g12), layout = c(1, 2))

data(rpjdl, package = "ade4")
coa1 <- ade4::dudi.coa(rpjdl$fau, scannf = FALSE)
G2 <- s1d.distri(coa1$li[,1], rpjdl$fau, labels = rpjdl$frlab,
  plabels = list(cex = 0.8, boxes = list(draw = FALSE)))

## Not run:
g31 <- s1d.distri(coa1$li[,1], rpjdl$fau, plabels = list(cex = 0.8, boxes = list(draw = FALSE)),
  plot = FALSE)
snc1 <- ade4::dudi.nsc(rpjdl$fau, scannf = FALSE)
```
s1d.dotplot  

1-D plot of a numeric score by dots

Description

This function represents a score using dots.

Usage

s1d.dotplot(score, at = 1:NROW(score), facets = NULL, plot = TRUE, 
storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

score  
a numeric vector (or a data frame) used to produce the plot

at  
a numeric vector used as an index

facets  
a factor splitting score so that subsets of the data are represented on different sub-graphics

plot  
a logical indicating if the graphics is displayed

storeData  
a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored

add  
a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device

pos  
an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

...  
additional graphical parameters (see adegpar and trellis.par.get)

Details

Graphical parameters for segments and dots are available in plines and in ppoints of adegpar. Some appropriated graphical parameters in p1d are also available.

Value

An object of class ADEg (subclass C1.dotplot) or ADEgS (if add is TRUE and/or if facets or data frame for score are used).
The result is displayed if plot is TRUE.
**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

**See Also**

`C1.dotplot` `ADEg.C1`

**Examples**

```r
data(rpjdl, package = "ade4")
rpjdl.coa <- ade4::dudi.coa(rpjdl$fau, scannf = FALSE, nf = 4)
s1d.dotplot(rpjdl.coa$eig)

set.seed(40)

score1 <- rnorm(10)
s1d.dotplot(score1)
```

---

**s1d.gauss**

**1-D plot of a numeric score by Gaussian curves**

**Description**

This function represents a score with a Gauss curve for each level of a factor.

**Usage**

```r
s1d.gauss(score, fac = gl(1, NROW(score)), wt = rep(1, NROW(score)), steps = 200, col = NULL, fill = TRUE, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
```

**Arguments**

- `score`: a numeric vector (or a data frame) used to produce the plot
- `fac`: a factor (or a matrix of factors) to split score
- `wt`: a vector of weights for score
- `steps`: a value for the number of segments used to draw the Gauss curves
- `col`: a logical, a color or a colors vector for labels, rugs, lines and polygons according to their factor level. Colors are recycled whether there are not one color by factor level.
- `fill`: a logical to yield the polygons Gauss curves filled
- `facets`: a factor splitting `score` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
storeData  
a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored

add  
a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device

pos  
an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

...  
additional graphical parameters (see adepar and trellis.par.get)

Details

Graphical parameters for rugs are available in plines of adepar and the ones for Gauss curves filled in ppolygons. Some appropriated graphical parameters in pld are also available.

Value

An object of class ADEg (subclass C1.gauss) or ADEgS (if add is TRUE and/or if facets or data frame for score or data frame for fac are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

C1.gauss ADEg.C1

Examples

data(meauL package = "ade4")
envpca <- ade4::dudi.pca(meau$env, scannf = FALSE)
dffac <- cbind.data.frame(meau$design$season, meau$design$site)
g1 <- s1d.gauss(envpca$li[, 1], fac = dffac, fill = TRUE, col = 1:6)
update(g1, steps = 10)
g2 <- s1d.gauss(envpca$li[, 1], dffac[, 2], ppoly.col = 1:4, paxes.draw = TRUE, ylim = c(0, 2),
    fill = TRUE, p1d.hori = FALSE)

s1d.hist  
1-D plot of a numeric score by bars

Description

This function represents a score using a chart with rectangular bars.
Usage

`s1d.hist(score, breaks = NULL, nclass = round(log2(length(score)) + 1),
    type = c("count", "density", "percent"), right = TRUE, facets = NULL,
    plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)`

Arguments

- `score`: a numeric vector (or a data frame) used to produce the plot
- `breaks`: a vector of values to split `score`. If `NULL`, `pretty(score, nclass)` is used.
- `nclass`: an integer for the number of desired intervals, ignored if `breaks` is not missing.
- `type`: a value among `count`, `density`, `percent` to indicate the unit of the cell height.
- `right`: a logical indicating if the histogram cells are right-closed (left open) intervals.
- `facets`: a factor splitting `score` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If `false`, only the names of the data arguments are stored
- `add`: a logical. If `true`, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is `false`
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Details

Graphical parameters for polygons are available in `ppolygons` of `adegpar`. Some appropriated graphical parameters in `p1d` are also available.

Value

An object of class `ADEg` (subclass `C1.hist`) or `ADEgS` (if `add` is `TRUE` and/or if `facets` or data frame for `score` are used).
The result is displayed if `plot` is `TRUE`.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

`C1.hist` ADEg.C1 hist
s1d.interval

1-D plot of the interval between two numeric scores

Description
This function represents the interval between two scores using either segments or filled areas.

Usage
s1d.interval(score1, score2, at = 1:NROW(score1), method = c("bars", "area"), facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments
score1 a numeric vector (or a data frame) used to produce the plot
score2 a numeric vector with as many values as values (or rows) in score1
at a numeric vector used as an index
method a value, bars or area, to represent either segments or areas between scores.
facets a factor splitting score so that subsets of the data are represented on different sub-graphics
plot a logical indicating if the graphics is displayed
storeData a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
add a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
pos an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
...
additional graphical parameters (see adegpar and trellis.par.get)

Details
Graphical parameters for polygons, lines and segment boundaries are available in respectively ppolygons, plines and parrows of adegpar. Some appropriated graphical parameters in p1d are also available.

Value
An object of class ADEg (subclass C1.interval) or ADEgS (if add is TRUE and/or if facets or data frame for score are used).
The result is displayed if plot is TRUE.
Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

C1.interval ADEg.C1

Examples

```r
set.seed(40)
s1 <- rnorm(10)
s2 <- rnorm(10)
s1d.interval(s1, s2, method = "bars")
s1d.interval(s1, s2, method = "area")
```

s1d.label 1-D plot of a numeric score with labels

Description

This function represents a numeric labeled score

Usage

```r
s1d.label(score, labels = 1:NROW(score), at = 0.5, poslabel = c("regular", "value"), facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
```

Arguments

- **score**: a numeric vector (or a data frame) used to produce the plot
- **labels**: the labels' names drawn for each score value
- **at**: a numeric vector used as an index
- **poslabel**: the label position of each class (each level of fac), it can be regular or value. If regular, labels are evenly spaced. If value, labels are placed on the weighted mean of their class.
- **facets**: a factor splitting score so that subsets of the data are represented on different sub-graphics
- **plot**: a logical indicating if the graphics is displayed
- **storeData**: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- **add**: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
pos

an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

... additional graphical parameters (see adepar and trellis.par.get)

Details

Graphical parameters for rugs are available in plines of adepar. Some appropriated graphical parameters in p1d are also available.

Value

An object of class ADEg (subclass S1.label) or ADEgS (if add is TRUE and/or if facets or data frame for score are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

S1.label ADEg S1

Examples

data(meau, package = "ade4")
envpca <- ade4::dudi.pca(meau$env, scannf = FALSE)
g1 <- s1d.match(envpca$li[, 1], row.names(envpca$li), plot = FALSE)
g2 <- s1d.match(envpca$co[, 1], row.names(envpca$co), p1d.reverse = TRUE, plot = FALSE)
G <- ADEgS(c(g1, g2), layout = c(2, 1))

s1d.match

1-D plot of the matching between two numeric scores

Description

This function represents paired scores with evenly spaced labels.

Usage

s1d.match(score1, score2, labels = 1:NROW(score1), at = 0.5,
 facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE,
 pos = -1, ...)
**Arguments**

- `score1`: a numeric vector (or a data frame) used to produce the plot
- `score2`: a numeric vector used to produce the plot with as many values as values (or rows) in `score1`
- `labels`: the labels' names drawn for each `score1` value
- `at`: a numeric vector used as an index
- `facets`: a factor splitting `score1` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add`: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

**Details**

Graphical parameters for rugs are available in `plines` of `adegpar`. Some appropriated graphical parameters in `p1d` are also available.

**Value**

An object of class `ADEg` (subclass `S1.match`) or `ADEgS` (if `add` is TRUE and/or if `facets` or data frame for `score` or data frame for `fac` are used).

The result is displayed if `plot` is TRUE.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

**See Also**

`S1.match` `ADEg.S1`

**Examples**

```r
s1d.match(-5:5, 2 * (-5:5))
```
S2.arrow-class

Description

A class for creating and drawing bi-dimensional plot with arrows from the origin to the coordinates and labeled.

Objects from the Class

S2.arrow objects can be created by calls of the form new("S2.arrow", ...).

The regular usage in this package is to use the s.arrow function.

Slots

data a list containing data or data's name.
  • dfxy: the displayed values in the form of a data frame, a name or a matching call.
  • xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
  • yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
  • labels: a vector containing the arrows' labels.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of ADEg.S2 class.

The specific slot for S2.arrow objects is:
  • Sp: a spatial object stem from Sp package.

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters:
  • lim.update: a logical indicating if the limits are updating

Call an object of class call

Extends

Class ADEg.S2, directly.
Class ADEg, by class ADEg.S2, distance 2.
Class ADEgORtrellis, by class ADEg.S2, distance 3.
Class ADEgORADEgSORtrellis, by class ADEg.S2, distance 3.
Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.arrow are:

- **prepare** signature(object = "S2.arrow"): calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates limits.
- **panel** signature(object = "S2.arrow"): draws points, arrows and labels.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S2 s.arrow

Examples

showClass("S2.arrow")

Description

A class for group representation in bi-dimensional plot.

Objects from the Class

S2.class objects can be created by calls of the form `new("S2.class", ...)`. The regular usage in this package is to use the s.class function.

Slots

data: a list containing data or data’s name.

- dfxy: the displayed values in the form of a data frame, a name or a matching call.
- fac: a factor (or a matrix of factors) splitting the rows of dfxy.
- xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
- yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
- wt: a vector of weights for fac.
- labels: a vector containing the class' labels.
- frame: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(...)`, `env = data$frame`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
• storeData: a logical indicating if the data should be stored in the returned object. If false, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of ADEg.S2 class.

The specific slots for S2.class objects are:
• ellipseSize: a positive number for ellipse size.
• starSize: a number between 0 and 1 for star size.
• chullSize: NULL or a vector of numbers between 0 and 1 for the convex hulls.
• col: a logical or a vector of colors that apply to points, ellipses, labels, lines and polygons.

stats a list of internal preliminary calculations. The specific slots for S2.class objects are:
• means: a matrix containing the weighted mean calculated for each fac value.
• covvar: a list containing the weighted variance-covariance matrices calculated for each fac value.

s.misc a list of some others internal parameters:
• ellipses: ellipses’ coordinates.
• chullcoord: convex hulls’ coordinates.

Call an object of class call

Extends

Class ADEg.S2, directly.
Class ADEg, by class ADEg.S2, distance 2.
Class ADEgORTrellis, by class ADEg.S2, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.S2, distance 3.

Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.class are:

prepare signature(object = "S2.class"): calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates ellipses, convex hulls and centroids.

panel signature(object = "S2.class"): draws ellipses, convex hulls, stars, labels and points.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S2 s.class
Examples

showClass("S2.class")

S2.corcircle-class     Class S2.corcircle

Description

A class for creating and drawing a correlation circle.

Objects from the Class

S2.corcircle objects can be created by calls of the form new("S2.corcircle", ...).
The regular usage in this package is to use the s.corcircle function.

Slots

data  a list containing data or data's name.
• dfxy: the displayed values in the form of a data frame, a name or a matching call.
• xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
• yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
• labels: a vector containing the points' labels.
• frame: a positive or null integer. It is the number of the frame containing the data
  (used with sys.frame(..., env = data$frame)). Only if the data are not stored
  (i.e. data$storeData = FALSE).
• storeData: a logical indicating if the data should be stored in the returned object. If
  FALSE, only the names of the data arguments are stored.

trellis.par  a list of parameters for lattice call. It will be passed directly to par.settings
arguments of the lattice function.

adeg.par  a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call  a list to create the trellis object.

g.args  a list containing some method parameters linked with the created object of ADEg.S2 class.
The specific slot for S2.corcircle objects is:
• fullcircle: a logical to include the complete circle (limits are then c(-1, 1)).

stats  a list of internal preliminary calculations

s.misc  a list of some others internal parameters:
  • backgrid: a list of elements for grid lines

Call  an object of class call

Extends

Class ADEg.S2, directly.
Class ADEg, by class ADEg.S2, distance 2.
Class ADEgORTrellis, by class ADEg.S2, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.S2, distance 3.
Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.corcircle are:

**prepare** signature(object = "S2.corcircle"): calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and prepares the drawn grid.

**panel** signature(object = "S2.corcircle"): draws arrows, labels and axes.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S2 s.corcircle

Examples

```r
showClass("S2.corcircle")
```
adeq.par  a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call  a list to create the trellis object.
g.args  a list containing some method parameters linked with the created object of Adeg.S2 class.

The specific slots for S2.density objects are:

- **bandwidth**: bandwidth for density calculations which is passed in parameters in the bkde2D function of the KernSmooth package.
- **gridsize**: grid dimension.
- **threshold**: a value between 0 and 1 to draw densities greater than this threshold. No density is visible whether it is equal to 1.
- **col**: a NULL value, a color or a colors vector to color densities.
- **nrpoints**: number of points on the density image.
- **contour**: a logical to draw contour lines.
- **region**: a logical to fill grid regions with col.
- **nclass**: number of class for density.

stats  a list of internal preliminary calculations. The specific slot for S2.density objects is:

- **densit**: a list containing the results of the bkde2D function.

s.misc  a list of some others internal parameters

Call  an object of class call

Extends

Class Adeg.S2, directly.
Class Adeg, by class Adeg.S2, distance 2.
Class AdegORTrellis, by class Adeg.S2, distance 3.
Class AdegORAdegSORTrellis, by class Adeg.S2, distance 3.

Methods

The methods of the father classes "Adeg.S2" and "Adeg" can be used by inheritance. The specific methods for S2.density are:

- **prepare** signature(object = "S2.density"): calls the parent method (prepare for Adeg.S2), modifies some graphical parameters used by default and calculates densities.
- **panel** signature(object = "S2.density"): draws densities and external points.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

Adeg Adeg.S2 s.density

Examples

showClass("S2.density")
S2.distri-class  

Class S2.distri

Description

A class for distributions on a numeric score using a mean-standard deviation display.

Objects from the Class

S2.distri objects can be created by calls of the form new("S2.distri", ...).
The regular usage in this package is to use the s.distri function.

Slots

data a list containing data or data’s name.
  - dfxy: the displayed values in the form of a data frame, a name or a matching call.
  - dfdistri: the mass distribution in which each column is a class.
  - xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
  - yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
  - frame: a positive or null integer. It is the number of the frame containing the data
    (used with sys.frame(...) env = data$frame). Only if the data are not stored
    (i.e. data$storeData = FALSE).
  - storeData: a logical indicating if the data should be stored in the returned object. If
    FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings
arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of adeG.s2 class.
The specific slots for S2.distri objects are:
  - ellipseSize: NULL or number between 0 and 1 for ellipse size.
  - starSize: NULL or number between 0 and 1 for star size.
  - col: a NULL value, a color or a colors vector to color ellipses, labels, lines and polygons.

stats a list of internal preliminary calculations. The specific slots for S2.distri objects are:
  - means: a matrix containing the weighted mean calculated for each class indfdistri.
  - covvar: a list containing the weighted variance-covariance matrices calculated for each
    class indfdistri.

s.misc a list of some others internal parameters:
  - ellipses: ellipses’ coordinates.

Call an object of class call
**S2.image-class**

**Extends**

- Class `ADEg.S2`, directly.
- Class `ADEg`, by class `ADEg.S2`, distance 2.
- Class `ADEgORtrellis`, by class `ADEg.S2`, distance 3.
- Class `ADEgORADEgSORtrellis`, by class `ADEg.S2`, distance 3.

**Methods**

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for `S2.distri` are:

- **prepare** signature(object = "S2.distri"): calls the parent method (prepare for `ADEg.S2`), modifies some graphical parameters used by default and calculates ellipses and centroids.
- **panel** signature(object = "S2.distri"): draws ellipses, stars, labels and points.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot `<aurelie.siberchicot@univ-lyon1.fr>` and Stephane Dray

**See Also**

- `ADEg ADEg.S2 s.distri`

**Examples**

```r
showClass("S2.distri")
```

**Description**

A class for the creation of a bi-dimensional plot with a third value represented as a continuous colored surface.

**Objects from the Class**

`S2.image` objects can be created by calls of the form `new("S2.image", ...)`. The regular usage in this package is to use the `s.image` function.
Slots

data  a list containing data or data's name.
  • dfxy: the displayed values in the form of a data frame, a name or a matching call.
  • z: a vector (or a matrix) of values on the dfxy rows.
  • xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
  • yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
  • frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of ADEg.S2 class.

The specific slots for S2.image objects are:
  • gridsize: a 1 or 2-length vector indicating the cell numbers (horizontally and vertically) of the grid for the colored surface.
  • outsidelimits: specific limits for the surface as a set of polygons. It must be an SpatialPolygons object. Hole are authorized.
  • span: a value to control the degree of smoothing.
  • contour: a logical to draw contour lines.
  • region: a logical to fill inter-contour regions.
  • col: a NULL value, a color or a colors vector used for the colored cells.

stats a list of internal preliminary calculations. The specific slot for S2.image objects is:
  • value: a prediction value yielded by a local polynomial regression fitting.

s.misc a list of some others internal parameters:
  • newgrid: the grid expansion calculated within the prepare method.

Call an object of class call

Extends

Class ADEg.S2, directly.
Class ADEg, by class ADEg.S2, distance 2.
Class ADEgORTrellis, by class ADEg.S2, distance 3.
Class ADEgORADEgSORTrellis, by class ADEg.S2, distance 3.

Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.image are:

prepare signature(object = "S2.image"): calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates grid expansion and limits.

panel signature(object = "S2.image"): draws raster image.
Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also
ADEg ADEg.S2 s.image

Examples
showClass("S2.image")

S2.label-class  Class S2.label

Description
A class for creating and drawing bi-dimensional plot with point label.

Objects from the Class
S2.label objects can be created by calls of the form new("S2.label", ...).
The regular usage in this package is to use the s.label function.

Slots
data a list containing data or data’s name.
  - dfxy: the displayed values in the form of a data frame, a name or a matching call.
  - labels: a vector of character strings for the points’ labels
  - xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
  - yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
  - frame: a positive or null integer. It is the number of the frame containing the data
    (used with sys.frame(..., env = data$frame)). Only if the data are not stored
    (i.e. data$storeData = FALSE).
  - storeData: a logical indicating if the data should be stored in the returned object. If
    FALSE, only the names of the data arguments are stored.
trellis.par a list of parameters for lattice call. It will be passed directly to par.settings
  arguments of the lattice function.
adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of ADEg.S2 class.
stats a list of internal preliminary calculations
s.misc a list of some others internal parameters
Call an object of class call
Extends

Class **ADEg.S2**, directly.
Class **ADEg**, by class ADEg.S2, distance 2.
Class **ADEgORtrellis**, by class ADEg.S2, distance 3.
Class **ADEgORADEgSORtrellis**, by class ADEg.S2, distance 3.

Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.label are:

- **prepare** signature(object = "S2.label"): calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default.
- **panel** signature(object = "S2.label"): draws points and labels.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

**ADEg ADEg.S2 s.label**

Examples

```r
showClass("S2.label")
```

---

**S2.logo-class**

*Class S2.logo*

Description

A class for the creation of a bi-dimensional plot with pictures for points representation.

Objects from the Class

S2.logo objects can be created by calls of the form `new("S2.logo", ...`).

The regular usage in this package is to use the s.logo function.
Slots

data a list containing data or data’s name.
  • dfxy: the displayed values in the form of a data frame, a name or a matching call.
  • logos: a list containing the picture to use for each point.
  • xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
  • yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
  • frame: a positive or null integer. It is the number of the frame containing the data
    (used with sys.frame(..., env = data$frame)). Only if the data are not stored
    (i.e. data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If
    FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings
  arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of ADEg.S2 class.

stats a list of internal preliminary calculations
s.misc a list of some others internal parameters

Call an object of class call

Extends

• Class ADEg.S2, directly.
• Class ADEg, by class ADEg.S2, distance 2.
• Class ADEgORTrellis, by class ADEg.S2, distance 3.
• Class ADEgORADEgSORTrellis, by class ADEg.S2, distance 3.

Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific
methods for S2.class are:

• prepare signature(object = "S2.class"): calls the parent method (prepare for ADEg.S2)
  and modifies some graphical parameters used by default.

• panel signature(object = "S2.class"): displays the logos.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S2 s.logo
Examples

showClass("S2_logo")

---

**S2.match-class**  
**Class** S2.match

**Description**

A class for the creation and display of paired coordinates in a bi-dimensional plot.

**Objects from the Class**

S2.match objects can be created by calls of the form `new("S2.match", ...)`. The regular usage in this package is to use the `s.match` function.

**Slots**

- `data`: a list containing data or data's name.
  - `dfxy`: the displayed values in the form of a data frame, a name or a matching call.
  - `xax`: an integer or a vector indicating the columns of `dfxy` kept for the x-axes.
  - `yax`: an integer or a vector indicating the columns of `dfxy` kept for the y-axes.
  - `labels`: a vector of character strings containing the matches' labels.
  - `frame`: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
  - `storeData`: a logical indicating if the data should be stored in the returned object. If `FALSE`, only the names of the data arguments are stored.
- `trellis.par`: a list of parameters for `lattice` call. It will be passed directly to `par.settings` arguments of the `lattice` function.
- `adeg.par`: a list of graphical parameters, corresponding to the ones given by `adegpar()` function.
- `lattice.call`: a list to create the trellis object.
- `g.args`: a list containing some method parameters linked with the created object of ADEg.S2 class.
  - The specific slot for S2.match objects is:
    - `arrows`: a logical to draw arrows.
- `stats`: a list of internal preliminary calculations
- `s.misc`: a list of some others internal parameters

**Call** an object of class `call`

**Extends**

- Class ADEg.S2, directly.
- Class ADEg, by class ADEg.S2, distance 2.
- Class ADEgORTrellis, by class ADEg.S2, distance 3.
- Class ADEgORADEgSORTrellis, by class ADEg.S2, distance 3.
Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.match are:

- **prepare** signature(object = "S2.match"): calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default.
- **panel** signature(object = "S2.match"): draws arrows and labels.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S2 s.match

Examples

```
showClass("S2.match")
```

---

S2.traject-class

Class S2.traject

Description

A class for the creation of a bi-dimensional plot with trajectories linking the points.

Objects from the Class

S2.traject objects can be created by calls of the form `new("S2.traject", ...)`. The regular usage in this package is to use the `s.traject` function.

Slots

data  a list containing data or data's name.

- dfxy: the displayed values in the form of a data frame, a name or a matching call.
- fac: a factor (or a matrix of factors) splitting the rows of dfxy.
- labels: a vector of character strings containing the trajectories' labels.
- xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
- yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
- frame: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(...)`, env = data$frame). Only if the data are not stored (i.e. data$storeData = FALSE).
- storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.
trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeq.par a list of graphical parameters, corresponding to the ones given by adeqpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of ADEg.S2 class.

The specific slots for S2.traject objects are:

- order: a vector containing the drawing order of the trajectories. A vector of length equal to factor.
- col: a NULL value, a color or a colors vector to color points, labels and lines.

stats a list of internal preliminary calculations
s.misc a list of some others internal parameters

Call an object of class call

Extends

Class ADEg.S2, directly.
Class ADEg by class ADEg.S2, distance 2.
Class ADEgORTrellis, by class ADEg.S2, distance 3.
Class ADEgOREADEgSORTrellis, by class ADEg.S2, distance 3.

Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.traject are:

- prepare signature(object = "S2.traject"): calls the parent method (prepare for ADEg.S2) and modifies some graphical parameters used by default.
- panel signature(object = "S2.traject"): draws points, arrows and labels.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

ADEg ADEg.S2 s.traject

Examples

showClass("S2.traject")
Description

A class for the creation and display of bi-dimensional plot with a third value represented (as a variable) by symbols.

Objects from the Class

S2.value objects can be created by calls of the form new("S2.value", ...).

The regular usage in this package is to use the s.value function.

Slots

data: a list containing data or data’s name.

• dfxy: the displayed values in the form of a data frame, a name or a matching call.
• z: a vector (or a matrix) with as many values as rows in dfxy.
• xax: an integer or a vector indicating the columns of dfxy kept for the x-axes.
• yax: an integer or a vector indicating the columns of dfxy kept for the y-axes.
• frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$storeData = FALSE).
• storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call a list to create the trellis object.

g.args a list containing some method parameters linked with the created object of ADEg.S2 class.

The specific slots for S2.value objects are:

• method: the method of representation for z (color shading or proportional size).
• symbol: the type of symbol (square or circle).
• center: a center value for method size.
• centerpar: a logical or a list to represent center value using elements in the adegpar("ppoints") list.
• breaks: a vector containing the breaks used for splitting z value. If NULL, pretty(z, n) is used.
• nclass: an integer for the number of desired intervals, ignored if breaks is not missing.
• col: a NULL value, a color or a colors vector to color symbols.

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters:
• plegend.update: a logical indicating if the legend parameters are updating
• breaks.update: a logical indicating if the legend breaks are updating
• lim.update: a logical indicating if the limits are updating

Call an object of class call

Extends

Class ADEg.S2, directly.
Class ADEg, by class ADEg.S2, distance 2.
Class ADEgortrellis, by class ADEg.S2, distance 3.
Class ADEgorADEgSORtrellis, by class ADEg.S2, distance 3.

Methods

The methods of the father classes "ADEg.S2" and "ADEg" can be used by inheritance. The specific methods for S2.value are:

prepare signature(object = "S2.value"): calls the parent method (prepare for ADEg.S2), modifies some graphical parameters used by default and calculates limits.

panel signature(object = "S2.value"): draws symbols.

Note

For the symbol size, if the method is size, we use perceptual scaling (Tanimura et al. 2006).

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot<aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References


See Also

ADEg ADEg.S2 s.value

Examples

showClass("S2.value")
setlimits1D

Computes limits for 1D and 2D displays.

Description

Computes limits for 1D and 2D displays adding 10% of margins around the extreme values.

Usage

setlimits1d(mini, maxi, origin, includeOr)
setlimits2d(minX, maxX, minY, maxY, origin = c(0, 0), aspect.ratio = "iso", includeOr)

Arguments

mini the smallest value of a unidimensional dataset
maxi the largest value of a unidimensional dataset
minX the smallest value of the first dimension of a bidimensional dataset
maxX the largest value of the first dimension of a bidimensional dataset
minY the smallest value of the second dimension of a bidimensional dataset
maxY the largest value of the second dimension of a bidimensional dataset
origin a value (in setlimits1d) or a two-length vector (in setlimits2d) indicating origin coordinate(s)
aspect.ratio a character string to control physical aspect ratio of the graphic. iso for isometric scales, fill for drawing as big as possible or xy for banking rule
includeOr a boolean value indicating whether the origin is included in the graphics window

Value

setlimits1D returns a two-length vector containing the limits of the graphics window on one axis.
setlimits2D returns a two-length list where the first element, named xlim, contains a two-length vector containing the limits of the graphics window on the first axis and the second, named ylim, contains the limits on the second axis.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

Examples

setlimits1D(mini = -2, maxi = 2, origin = 0, includeOr = TRUE)
setlimits2D(minX = -2, maxX = 2, minY = -3, maxY = 4, origin = c(0, 0), includeOr = TRUE)
sortparamADEg

Sort a sequence of graphical parameters

Description

Sort a sequence of graphical parameters in several lists.

Usage

sortparamADEg(...)
sortparamADEgS(..., graphnames, nbsubgraphs = rep(1, length(graphnames)))

Arguments

... a sequence of graphical parameters
graphnames a sequence containing the name of each simple graph of the ADEgS
nbsubgraphs a sequence containing the number of sub-graphs in each graph named in graphnames

Value

sortparamADEg return a list of four lists named adepar, trellis, g.args and rest. sortparamADEgS return a list of as many lists as the length of graphnames, i.e., as the number of sub-graphs of the ADEgS. The names of the lists are graphnames and each sub-list is the result of the sortparamADEg function apply on each sub-graph.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

Examples

l1 <- sortparamADEg(xlab = "x-axis label", ylab = "y-axis label", plabels.cex = 1.5,
  porigin.include = FALSE)
length(l1)
names(l1)

l2 <- sortparamADEgS(xlab = "x-axis label", eig.main = "Eigenvalues", row.ppoints.col = "red",
  porigin.include = FALSE, graphnames = c("row", "col", "eig"))
names(l2)
names(l2$row)

l3 <- sortparamADEgS(xlab = "x-axis label", eig.main = "Eigenvalues", row.ppoints.col = "pink",
  porigin.include = FALSE, graphnames = c("row", "col", "eig"), nbsubgraphs = c(1, 2, 1))
names(l3)
length(l3$row)
length(l3$col)
**superpose**

Superpose two graphics

**Description**

This function superposes two graphics and extends the graphical constraints of a first graphic to a second one.

**Usage**

```r
superpose(g1, g2, which, plot = FALSE)
```

```r
# S4 method for signature 'ADEg'
e1 + e2
```

**Arguments**

- `g1`: an object of class ADEg, ADEgS or trellis
- `g2`: an object of class ADEg, ADEgS or trellis superposed on `g1`
- `e1`: an object of class ADEg or ADEgS
- `e2`: an object of class ADEg or ADEgS superposed on `e1`
- `which`: if `g1` is an ADEgS, which ADEg is used as the base of superposition (`g2` is superposed on `g1[[which]]`)
- `plot`: a logical indicating if the graphics is displayed

**Details**

The created ADEgS object is a layout of two graphical objects. Each of the two objects superposed still have its graphical parameters in the created layout. However, the ADEgS displayed favour the graphical parameters of the object below: displayed limits, grid, legend and axes are those of `g1` (respectively `e1`) and `g2` (respectively `e2`) has transparent background and labels’ boxes.

The superpose method is defined for:

- signature(`g1 = "ADEgS", g2 = "ADEg", which = "numeric", plot = "logical")
- signature(`g1 = "ADEgS", g2 = "ADEg", which = "numeric", plot = "ANY")
- signature(`g1 = "ADEgS", g2 = "ADEg", which = "missing", plot = "ANY"`): If `which` is missing, the last ADEg of `g1@ADEglist` is used as the base of superposition. In that case, `which = length(g1)`
- signature(`g1 = "ADEgORtrellis", g2 = "ADEgORtrellis", which = "ANY", plot = "ANY"`): If `g1` is an ADEg object, no which is needed.
- signature(`g1 = "ADEgS", g2 = "ADEgS", which = "missing", plot = "ANY"`)

The + method is defined for:

- signature(`e1 = "ADEg", e2 = "ADEg"`): superpose `e2` on `e1`
- signature(`e1 = "ADEg", e2 = "ADEgS"`): superpose `e2` to `e1`
- signature(`e1 = "ADEgS", e2 = "ADEg"`): calls the + method with signature (`e1 = "ADEg", e2 = "ADEgS"`).
Value

An object of class "ADEgS".

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

add.ADEg ADEgS ADEg

Examples

cha <- LETTERS[1:20]
xy <- cbind.data.frame(runif(length(cha)), runif(length(cha)))
g1 <- s.label(xy, labels = cha, ppoints.alpha = 0, pbackground.col = "grey85")
g2 <- s.label(xy, labels = cha, plabels.cex = 0, paxes.draw = TRUE, ppoints.pch = 4,
ppoints.col = "red")
g3 <- superpose(g1, g2, plot = TRUE)
g4 <- superpose(g2, g1, plot = TRUE)

data(jv73, package = "ade4")
pca1 <- ade4::dudi.pca(jv73$morpho, scanf = FALSE)
g5 <- s.label(pca1$li, plabels.optim = TRUE)
g6 <- s.class(pca1$li, jv73$fac.riv, starSize = 0, ellipseSize = 0, chullSize = 1,
ppolygons.alpha = 0.4, col = rainbow(12), ppoints.cex = 0)
g5 + g6

## Not run: g7 <- s.label(pca1$li, plabels.optim = TRUE, facets = jv73$fac.riv, plot = FALSE)
g8 <- s.class(pca1$li, jv73$fac.riv, facets = jv73$fac.riv, starSize = 0, chullSize = 1,
ellipseSize = 0, ppolygons.alpha = 0.4, col = rainbow(12), ppoints.cex = 0, plot = FALSE)
g9 <- superpose(g7, g8, plot = TRUE)

## End(Not run)
Slots

data: a list containing data or data’s name.
  • dftab: a contingency table object in the form of a data frame, a name or a matching call
  • coordsx: an integer or a vector indicating the columns of dftab kept
  • coordsy: an integer or a vector indicating the rows of dftab kept
  • labelsx: the columns’ labels
  • labelsy: the rows’ labels
  • "frame: a positive or null integer. It is the number of the frame containing the data
    (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e.
    data$storeData = FALSE).
  • storeData: a logical indicating if the data should be stored in the returned object. If
    FALSE, only the names of the data arguments are stored.

trellis.par a list of parameters for lattice call. It will be passed directly to par.settings
arguments of the lattice function.

adeg.par a list of graphical parameters, corresponding to the ones given by adegpar() function.
lattice.call a list to create the trellis object.
g.args a list containing some method parameters linked with the created object of T.value class.

The specific slots for T.cont objects are:
  • meanX: a logical to represent columns’ means by points.
  • meanY: a logical to represent rows’ means by points.
  • ablineX: a logical to represent columns’ regression lines.
  • ablineY: a logical to represent columns’ regression lines.

stats a list of internal preliminary calculations
s.misc a list of some others internal parameters

call an object of class call

Extends

Class T.value, directly.
Class ADEg.T, by class T.value, distance 2.
Class ADEg, by class T.value, distance 3.
Class ADEgORTrellis, by class T.value, distance 4.
Class ADEgORADEgSORtrellis, by class T.value, distance 4.

Methods

The methods of the father classes "T.value", "ADEg.T" and "ADEg" can be used by inheritance.
The specific methods for T.cont are:

  panel signature(object = "T.cont"): draws mean points and regression lines.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
See Also

ADEg ADEg.T T.value table.value

Examples

showClass("T.cont")

---

T.image-class  

Class T.image

Description

A class for the representation of a matrix or table object in which values have different colors.

Objects from the Class

T.image objects can be created by calls of the form new("T.image", ...).

The regular usage in this package is to use the table.image function.

Slots

data  
a list containing data or data's name.

  * dftab: the displayed values which can be table, dist or matrix in the form of a data frame, a name or a matching call
  * coordsx: an integer or a vector indicating the columns of dftab kept
  * coordsy: an integer or a vector indicating the rows of dftab kept
  * labelsx: columns labels
  * labelsy: rows labels
  * frame: a positive or null integer. It is the number of the frame containing the data (used with sys.frame(..., env = data$frame)). Only if the data are not stored (i.e. data$datastoreData = FALSE).
  * storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

trellis.par  
a list of parameters for lattice call. It will be passed directly to par.settings arguments of the lattice function.

adeg.par  
a list of graphical parameters, corresponding to the ones given by adegpar() function.

lattice.call  
a list to create the trellis object.

g.args  
a list containing some method parameters linked with the created object of ADEg.T class.

  The specific slots for T.image objects are:

  * breaks: a vector of values to split dftab. If NULL, pretty(dftab, nclass) is used.
  * nclass: an integer for the number of desired intervals, ignored if breaks is not missing.
  * col: a NULL value, a color or a colors vector used for the cells.

stats  
a list of internal preliminary calculations

s.misc  
a list of some others internal parameters:

  * breaks.update: a logical indicating if the legend breaks is updating.

Call  
an object of class call
Extends

- Class \texttt{A\textit{De}g.\textit{T}}, directly.
- Class \texttt{A\textit{De}g}, by class \texttt{A\textit{De}g.\textit{T}}, distance 2.
- Class \texttt{A\textit{De}gOR\textit{trellis}}, by class \texttt{A\textit{De}g.\textit{T}}, distance 3.
- Class \texttt{A\textit{De}gORA\textit{De}gSOR\textit{trellis}}, by class \texttt{A\textit{De}g.\textit{T}}, distance 3.

Methods

The methods of the father classes "\texttt{A\textit{De}g.\textit{T}}" and "\texttt{A\textit{De}g}" can be used by inheritance. The specific methods for \texttt{T.image} are:

- **\texttt{prepare}** signature(object = "\texttt{T.image}"): calls the parent method (prepare for \texttt{A\textit{De}g.\textit{T}}) and modifies some graphical parameters used by default and calculates limits and grid.
- **\texttt{panel}** signature(object = "\texttt{T.image}"): draws raster image.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

\texttt{A\textit{De}g \texttt{A\textit{De}g.\textit{T table.image}}}

Examples

```r
showClass("\texttt{T.image}")
```

---

T.value-class

\begin{tabular}{ll}
\texttt{T.value-class} & \textit{Class T.value} \\
\end{tabular}

Description

A class for the representation of a matrix, a data frame, or a distance matrix using symbols, varying in size or color.

Objects from the Class

\texttt{T.value} objects can be created by calls of the form \texttt{\texttt{new(\texttt{"T.value", ...)}}. The regular usage in this package is to use the \texttt{table.value} function.
Slots

data: a list containing data or data’s name.
  • dftab: the displayed values which can be table, dist or matrix in the form of a data frame, a name or a matching call
  • coordsx: an integer or a vector indicating the columns of dftab kept
  • coordsy: an integer or a vector indicating the rows of dftab kept
  • labsx: the columns’ labels
  • labsy: the rows’ labels
  • frame: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
  • storeData: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

`trellis.par` a list of parameters for `lattice` call. It will be passed directly to `par.settings` arguments of the `lattice` function.

`adeg.par` a list of graphical parameters, corresponding to the ones given by `adegpar()` function.

`lattice.call` a list to create the `trellis` object.

`g.args` a list containing some method parameters linked with the created object of `ADEg.T` class.

The specific slots for `T.value` objects are:
  • breaks: a vector of values to split dftab. If NULL, `pretty(dftab, nclass)` is used.
  • nclass: an integer for the number of desired intervals, ignored if breaks is not missing.
  • col: a NULL value, a color or a colors vector to color symbols.
  • method: the method of representation for dftab (color shading or proportional size).
  • symbol: the type of symbol (square or circle).
  • center: a center value for method size.
  • centerpar: a logical or a list to represent center value using elements in the `adegpar("ppoints")` list.

`stats` a list of internal preliminary calculations

`s.misc` a list of some others internal parameters:
  • breaks.update: a logical indicating if the legend breaks is updating.

Call an object of class `call`
prepare signature(object = "T.value"): calls the parent method (prepare for ADEg.T) and modifies some graphical parameters used by default and calculates limits and grid.

panel signature(object = "T.value"): draws symbols.

Note
For the symbol size, if the method is size, we use perceptual scaling (Tanimura et al. 2006).

Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References

See Also
ADEg ADEg.T T.cont table.value

Examples

showClass("T.value")

---

table.image

Heat map-like representation with colored cells

Description
This function represents a two dimensional table plot in which cells are colored according with their value.

Usage

table.image(dftab, coordsx = 1:ncol(as.matrix(dftab)), coordsy = nrow(as.matrix(dftab)):1, labelsx, labelsy, nclass = 3, breaks = NULL, col = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
Arguments

dftab a data frame, matrix, contingency table or distance matrix used to produce the plot
coordsx an integer or a vector indicating the columns of dftab kept
coordsy an integer or a vector indicating the rows of dftab kept
labelsx columns labels
labelsy rows labels
breaks a vector of values to split dftab. If NULL, pretty(dftab, nclass) is used.
nclass an integer for the number of desired intervals, ignored if breaks is not missing.
col a color or a colors vector used for the cells
plot a logical indicating if the graphics is displayed
storeData a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored
add a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
pos an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE
... additional graphical parameters (see adegpar and trellis.par.get)

Value

An object of class ADEg (subclass T.image) or ADEgS (if add is TRUE). The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

T.image ADEg.T

Examples

tab <- as.table(matrix(rnorm(900), ncol = 30))
g1 <- table.image(tab)

# add a continuous color bar as legend
# update(g1, legend.drawColorKey = TRUE)

g2 <- table.image(tab, n = 100, coordsx = c(30, 1:29), legend.drawKey = FALSE)
data(rpjdl, package = "ade4")
X <- data.frame(t(rpjdl$fau))
Y <- data.frame(t(rpjd$mil))
coa1 <- ade4::dudi.coa(X, scannf = FALSE)
g3 <- table.image(Y, coordsx = rank(coa1$co[, 1]), coordsy = 1:8, nclas = 5,
   labelsx = "", plegend.drawKey = FALSE)

**table.value** | Heat map-like representation with proportional symbols

**Description**

This function represents a two dimensional table plot with proportional or colored squares or circles for each value.

**Usage**

```r
table.value(dftab, coordsx = 1:ncol(as.matrix(dftab)), coordsy =
nrow(as.matrix(dftab)):1, labelsx, labelsy, breaks = NULL, method =
c("size", "color"), symbol = c("square", "circle", "diamond",
"uptriangle", "downtriangle"), col = NULL, nclass = 3, center = 0,
centerpar = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
```

**Arguments**

dftab | a data frame, matrix, contingency table or distance matrix used to produce the plot  
coordsx | an integer or a vector indicating the columns of dftab kept  
coordsy | an integer or a vector indicating the rows of dftab kept  
labelsx | columns labels  
labelsy | rows labels  
breaks | a vector of values to split dftab. If NULL, pretty(dftab, nclass) is used.  
method | color or size value for represent z. If color, a palette of color is used for the symbols (one color per interval). If size, symbols of proportional area are used. Area is 0 for values equals to center (default 0). Two colors are used, for values less than center and larger than center.  
symbol | value for symbol type  
col | a color or a colors vector to color symbols. If method is size, a 2-length vector of color is expected. If method is color, it must have as many colors as the number of class  
nclass | an integer for the number of desired intervals, ignored if breaks is not missing.  
center | a center value for method size  
centerpar | a logical or a list to represent center value using elements in the adegpar("ppoints") list  
plot | a logical indicating if the graphics is displayed
storeData | a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored

add | a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device

pos | an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

... | additional graphical parameters (see adegpar and trellis.par.get)

Value

An object of class ADEg (subclass T.cont if dftab is an table object, otherwise subclass T.value) or ADEgS (if add is TRUE).

The result is displayed if plot is TRUE.

Note

For the symbol size, if the method is size, we use perceptual scaling (Tanimura et al. 2006).

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

References


See Also

T.value T.cont ADEg.T

Examples

```r
## data.frame
data(olympic, package = "ade4")
w <- olympic$tab
w <- data.frame(scale(w))
wpca <- ade4::dudi.pca(w, scann = FALSE)
g1 <- table.value(w, ppoints.cex = 0.5, axis.line = list(col = "darkblue"), axis.text = list(col = "darkgrey"))

# update the legend position
update(g1, key = list(space = "left"))
update(g1, key = list(columns = 1))

g2 <- table.value(w, coordsy = rank(wpca$li[, 1]), ppoints.cex = 0.5, axis.line = list(col = "darkblue"), axis.text = list(col = "darkgrey"))
g3 <- table.value(w, coordsy = wpca$li[, 1], coordsx = wpca$co[, 1], ppoints.cex = 0.5,
```
### Description

A class for group representation in triangular plot.

### Objects from the Class

`Tr.class` objects can be created by calls of the form `new("Tr.class", ...)`. The regular usage in this package is to use the `triangle.class` function.

### Slots

- **data**: a list containing data or data’s name.
  - `dfxyz`: the displayed values in the form of a data frame with three columns, a name or a matching call.
  - `fac`: a factor partitionning the rows of `dfxyz`.
  - `wt`: a vector of weights for `fac`.
  - `labels`: a vector containing the class’ labels.
  - `frame`: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
  - `storeData`: a logical indicating if the data should be stored in the returned object. If `FALSE`, only the names of the data arguments are stored.

- **trellis.par**: a list of parameters for `lattice` call. It will be passed directly to `par.settings` arguments of the `lattice` function.

- **adeg.par**: a list of graphical parameters, corresponding to the ones given by `adegpar()` function.

- **lattice.call**: a list to create the `trellis` object.
The specific slots for `Tr.class` objects are:

- `ellipsesize`: a positive number for ellipse size.
- `starSize`: a number between 0 and 1 for star size.
- `chullSize`: NULL or a vector of numbers between 0 and 1 for the convex hulls.
- `col`: a NULL value, a color or a colors vector to color points, ellipses, labels, lines and polygons.
- `max3d` and `min3d`: vectors of three values for triangular maximal and minimal limits.
- `adjust`: a logical to adjust the device with the limits of the smaller equilateral triangle containing the values.

The specific slots for `S2.class` objects are:

- `means`: a matrix containing the weighted mean calculated for each `fac` value.
- `mean2d`: a matrix containing the weighted mean calculated for each `fac` value on two-dimension.
- `covvar`: a list containing the weighted variance-covariance matrices calculated for each `fac` value.
- `covvar2d`: a list containing the weighted variance-covariance matrices calculated for each `fac` value on two-dimension.

The specific slots for `S.misc` objects are:

- `ellipses`: ellipses' coordinates.
- `chullcoord`: convex hulls' coordinates.

Call an object of class `call`

**Extends**

- Class `AdeG.Tr`, directly.
- Class `AdeG`, by class `AdeG.Tr`, distance 2.
- Class `AdeGORTrellis`, by class `AdeG.Tr`, distance 3.
- Class `AdeGORADEGSORTrellis`, by class `AdeG.Tr`, distance 3.

**Methods**

The methods of the father classes "AdeG.Tr" and "AdeG" can be used by inheritance. The specific methods for `Tr.class` are:

- `prepare` signature(object = "Tr.class"): calls the parent method (prepare for `AdeG.Tr`), modifies some graphical parameters used by default and calculates ellipses, convex hulls and centroids.
- `panel` signature(object = "Tr.class"): draws arrows, labels and points.

**Author(s)**

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray
See Also

ADEg ADEg.Tr triangle.class

Examples

showClass("Tr.class")

---

**Tr.label-class**

**Class** Tr.label

---

Description

A class for creating and drawing triangular plot with point label.

**Objects from the Class**

Tr.label objects can be created by calls of the form `new("Tr.label", ...)`.

The regular usage in this package is to use the triangle.label function.

**Slots**

data a list containing data or data’s name.

- `dfxyz`: the displayed values in the form of a three columns data frame, a name or a matching call.
- `labels`: a character vector containing labels for points.
- `frame`: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
- `storeData`: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

**trellis.par** a list of parameters for lattice call. It will be passed directly to `par.settings` arguments of the lattice function.

**adeg.par** a list of graphical parameters, corresponding to the ones given by `adegpar()` function.

**lattice.call** a list to create the trellis object.

**g.args** a list containing some method parameters linked with the created object of ADEg.Tr class.

The specific slots for Tr.class objects are:

- `addmean`: a logical to plot the mean.
- `addaxes`: a logical to draw the principal axes.
- `meanpar`: a list to represent mean points using `pch`, `cex` and `col`.
- `axespar`: a list to represent axes lines using `col`, `lwd` and `lty`.
- `max3d` and `min3d`: vectors of three values for triangular maximal and minimal limits.
- `adjust`: a logical to adjust the device with the limits of the smaller equilateral triangle containing the values.
stats a list of internal preliminary calculations
s.miss a list of some others internal parameters:
  • cornerp: coordinates of the triangle extremities.

Call an object of class call

Extends

Class adeG Tr, directly.
Class adeG, by class adeG Tr, distance 2.
Class adeG0RTrellis, by class adeG Tr, distance 3.
Class adeG0RADEgSORTrellis, by class adeG Tr, distance 3.

Methods

The methods of the father classes "adeG Tr" and "adeG" can be used by inheritance. The specific methods for Tr.label are:

prepare signature(object = "Tr.label"): calls the parent method (prepare for adeG Tr), modifies some graphical parameters used by default and defines the mean point and the axes.

panel signature(object = "Tr.label"): draws lines, labels and points.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

adeG adeG Tr triangle.label

Examples

showClass("Tr.label")

---

Tr.match-class Class Tr.match

Description

A class for the creation and display of paired coordinates in a triangular plot.

Objects from the Class

Tr.match objects can be created by calls of the form new("Tr.match", ...).
The regular usage in this package is to use the triangle.match function.
Slots

data a list containing data or data's name.

• dfxyz: the displayed values in the form of a three columns data frame, a name or a
  matching call.
• labels: a vector of character strings containing the matches' labels.
• frame: a positive or null integer. It is the number of the frame containing the data
  (used with \texttt{sys.frame(...) env = data$frame}). Only if the data are not stored
  (i.e. \texttt{data$storeData = FALSE}).
• storeData: a logical indicating if the data should be stored in the returned object. If
  \texttt{FALSE}, only the names of the data arguments are stored.

trellis.par a list of parameters for \texttt{lattice} call. It will be passed directly to \texttt{par.settings}
arguments of the \texttt{lattice} function.

adeg.par a list of graphical parameters, corresponding to the ones given by \texttt{adegpar()} function.

lattice.call an object of class \texttt{lattice}.

\texttt{g.args} a list containing some method parameters linked with the created object of \texttt{ADEg.Tr} class.

The specific slots for \texttt{tr.match} objects are:

• max3d and min3d: vectors of three values for triangular maximal and minimal limits.
• adjust: a logical to adjust the device with the limits of the smaller equilateral triangle
  containing the values

stats a list of internal preliminary calculations

s.misc a list of some others internal parameters

Call an object of class call

Extends

Class \texttt{ADEg.Tr}, directly.
Class \texttt{ADEg}, by class \texttt{ADEg.Tr}, distance 2.
Class \texttt{ADEgORTrellis}, by class \texttt{ADEg.Tr}, distance 3.
Class \texttt{ADEgORADEgSORTrellis}, by class \texttt{ADEg.Tr}, distance 3.

Methods

The methods of the father classes "ADEg.Tr" and "ADEg" can be used by inheritance. The specific
methods for \texttt{Tr.match} are:

\texttt{prepare} signature(object = "Tr.match"): calls the parent method (\texttt{prepare} for \texttt{ADEg.Tr})
and modifies some graphical parameters used by default.

\texttt{panel} signature(object = "Tr.match"): draws arrows, labels and points.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane
Dray
See Also

`ADEg ADEg.Tr triangle.match`

Examples

```r
showClass("Tr.match")
```

Description

A class for the creation and display of triangular plot with trajectories linking the points.

Objects from the Class

`Tr.traj` objects can be created by calls of the form `new("Tr.traj", ...)`. The regular usage in this package is to use the `triangle.traj` function.

Slots

data a list containing data or data’s name.

- `dfxyz`: the displayed values in the form of a three columns data frame, a name or a matching call.
- `fac`: a factor (or a matrix of factors) splitting the rows of `dfxyz`.
- `labels`: a vector of character strings containing the trajectories’ labels.
- `frame`: a positive or null integer. It is the number of the frame containing the data (used with `sys.frame(..., env = data$frame)`). Only if the data are not stored (i.e. `data$storeData = FALSE`).
- `storeData`: a logical indicating if the data should be stored in the returned object. If FALSE, only the names of the data arguments are stored.

`trellis.par` a list of parameters for `lattice` call. It will be passed directly to `par.settings` arguments of the lattice function.

`adeg.par` a list of graphical parameters, corresponding to the ones given by `adegpar()` function.

`lattice.call` a list to create the `trellis` object.

g.args a list containing some method parameters linked with the created object of `ADEg.Tr` class.

The specific slots for `Tr.traj` objects are:

- `max3d` and `min3d`: vectors of three values for triangular maximal and minimal limits.
- `adjust`: a logical to adjust the device with the limits of the smaller equilateral triangle containing the values.
- `order`: a vector containing the drawing order of the trajectories. A vector of length equal to `factor`.
- `col`: a `NULL` value, a color or a colors vector to color points, labels and lines.

`stats` a list of internal preliminary calculations

`s.misc` a list of some others internal parameters

Call an object of class `call`
extends

Class `ADEg.Tr`, directly.
Class `ADEg`, by class `ADEg.Tr`, distance 2.
Class `ADEgORTrellis`, by class `ADEg.Tr`, distance 3.
Class `ADEgORADEgSORTrellis`, by class `ADEg.Tr`, distance 3.

Methods

The methods of the father classes "ADEg.Tr" and "ADEg" can be used by inheritance. The specific methods for `Tr.traject` are:

`prepare` signature(`object = "Tr.traject"`): calls the parent method (prepare for `ADEg.Tr`) and modifies some graphical parameters used by default.

`panel` signature(`object = "Tr.traject"`): draws arrows, labels and points.

Authors

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

`ADEg` `ADEg.Tr` `triangle.traject`

Examples

```r
showClass("Tr.traject")
```

Description

This function represents a three dimensional scatter plot with a partition in classes (levels of a factor).

Usage

```r
triangle.class(dxyz, fac, wt = rep(1, NROW(fac)), labels = levels(fac),
   col = NULL, ellipseSize = 1, starSize = 1, chullSize = NULL, adjust = TRUE,
   min3d = NULL, max3d = NULL, showposition = TRUE, facets = NULL, plot = TRUE,
   storeData = TRUE, add = FALSE, pos = -1, ...)
```
Arguments

- `dfxyz`: a three columns data frame used to produce the plot
- `fac`: a factor (or a matrix of factors) splitting the rows of `dfxyz`
- `wt`: a vector of weights for `fac`
- `labels`: a character vector containing the class’ labels
- `col`: a logical, a color or a colors vector to color points, ellipses, labels, lines and polygons
- `ellipseSize`: a positive number for ellipse size
- `starSize`: a number between 0 and 1 for the size of the stars segments joining the stars’ center (centroids) and the matching points
- `chullSize`: NULL or a vector of numbers between 0 and 1 for the fraction of points included in the convex hull
- `adjust`: a logical to adjust the device with the limits of the smaller equilateral triangle containing the values
- `min3d`: a vector of three values for triangular minimal limits
- `max3d`: a vector of three values for triangular maximal limits
- `showposition`: a logical indicating whether the used triangle should be shown in the complete one
- `facets`: a factor splitting the rows of `dfxyz` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored
- `add`: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is FALSE
- ... additional graphical parameters (see `adegpar` and `trellis.par.get`)

Value

An object of class ADEg (subclass `Tr.class`) or ADEgS (if `showposition` is TRUE, if `add` is TRUE and/or if `facets` are used).

The result is displayed if `plot` is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

`Tr.class` ADEg.Tr
Examples

data(euro123, package = "ade4")
fac1 <- euro123$plan$an
df1 <- rbind.data.frame(euro123$in78, euro123$in86, euro123$in97)
triangle.class(df1, fac = fac1, showposition = TRUE, col = c(1, 2, 3))
triangle.class(df1, fac = fac1, showposition = FALSE, plabels.cex = 0, col = c(1, 2, 3),
key = list(space = "left"))

data(dfxyz, labels = rownames(dfxyz), adjust = TRUE, min3d = NULL,
max3d = NULL, addaxes = FALSE, addmean = FALSE, meanpar = NULL, axespar = NULL,
showposition = TRUE, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE,
pos = -1, ...)

triangle.label Ternary plot with labels

Description

This function represents a three dimensional scatter plot with labels.

Usage

triangle.label(dfxyz, labels = rownames(dfxyz), adjust = TRUE, min3d = NULL,
max3d = NULL, addaxes = FALSE, addmean = FALSE, meanpar = NULL, axespar = NULL,
showposition = TRUE, facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE,
pos = -1, ...)
pos

an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE

... additional graphical parameters (see adepar and trellis.par.get)

Value

An object of class ADEg (subclass Tr.label) or ADEgS (if showposition is TRUE, if add is TRUE and/or if facets are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

Tr.label ADEg.Tr

Examples

data(euro123, package = "ade4")
df <- rbind.data.frame(euro123$in78, euro123$in86, euro123$in97)
row.names(df) <- paste(row.names(euro123$in78), rep(c(1, 2, 3), rep(12, 3)), sep = "")
g1 <- triangle.match(df, label = row.names(df), showposition = TRUE, plot = FALSE)
g2 <- triangle.match(euro123$in78, labels = row.names(euro123$in78), plabels.cex = 0, ppoints.cex = 2, addmean = TRUE, show = FALSE, plot = FALSE)
g3 <- triangle.match(euro123$in86, labels = row.names(euro123$in86), plabels.cex = 0.8, plot = FALSE)
g4 <- triangle.match(rbind.data.frame(euro123$in78, euro123$in86), labels = row.names(euro123$in78), plabels.cex = 0.8, addaxes = TRUE, psub.te = "Principal axis", psub.cex = 1.5, psub.pos = "topright", plot = FALSE)
G <- ADEgS(c(g1, g2, g3, g4), layout = c(2, 2))

---

triangle.match  Ternary plot of the matching between two sets of coordinates

Description

This function represents a three dimensional scatter plot of paired coordinates.

Usage

triangle.match(dfxyz1, dfxyz2, labels = row.names(as.data.frame(dfxyz1)),
min3d = NULL, max3d = NULL, adjust = TRUE, showposition = TRUE, facets = NULL,
plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)

Arguments

- **dfxyz1**: a three columns data frame, the first system of coordinates, used to produce the plot.
- **dfxyz2**: a three columns data frame, the second system of coordinates, with as many rows as dfxyz1, used to produce the plot.
- **labels**: a vector of character strings containing the matches’ labels.
- **adjust**: a logical to adjust the device with the limits of the smaller equilateral triangle containing the values.
- **min3d**: a vector of three values for triangular minimal limits.
- **max3d**: a vector of three values for triangular maximal limits.
- **showposition**: a logical indicating whether the used triangle should be shown in the complete one.
- **facets**: a factor splitting the rows of dfxyz so that subsets of the data are represented on different sub-graphics.
- **plot**: a logical indicating if the graphics is displayed.
- **storeData**: a logical indicating if the data are stored in the returned object. If FALSE, only the names of the data arguments are stored.
- **add**: a logical. If TRUE, the graphic is superposed to the graphics already plotted in the current device.
- **pos**: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if storeData is FALSE.
- **NNN**: additional graphical parameters (see adegpar and trellis.par.get).

Value

An object of class ADEg (subclass Tr.match) or ADEgS (if showposition is TRUE, if add is TRUE and/or if facets are used).

The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

- Tr.match ADEg.Tr

Examples

```r
data(eurol23, package = "ade4")
triangle.match(eurol23$in78, eurol23$in86, plabels.cex = 0.8)
```
Triangle trajectory

Description
This function represents a three-dimensional scatter plot with trajectories.

Usage
```r
triangle.traject(dfxyz, fac = gl(1, nrow(dfxyz)), order, labels = levels(fac),
                 col = NULL, adjust = TRUE, min3d = NULL, max3d = NULL, showposition = TRUE,
                 facets = NULL, plot = TRUE, storeData = TRUE, add = FALSE, pos = -1, ...)
```

Arguments
- `dfxyz`: a three columns data frame, the first system of coordinates, used to produce the plot
- `fac`: a factor (or a matrix of factors) splitting the rows of `dfxyz`
- `order`: a vector containing the drawing order of the trajectories. A vector of length equal to factor.
- `labels`: a vector of character strings containing the trajectories’ labels
- `col`: a color or a colors vector to color points, labels, and lines
- `adjust`: a logical to adjust the device with the limits of the smaller equilateral triangle containing the values
- `min3d`: a vector of three values for triangular minimal limits
- `max3d`: a vector of three values for triangular maximal limits
- `showposition`: a logical indicating whether the used triangle should be shown in the complete one
- `facets`: a factor splitting the rows of `dfxyz` so that subsets of the data are represented on different sub-graphics
- `plot`: a logical indicating if the graphics is displayed
- `storeData`: a logical indicating if the data are stored in the returned object. If `FALSE`, only the names of the data arguments are stored
- `add`: a logical. If `TRUE`, the graphic is superposed to the graphics already plotted in the current device
- `pos`: an integer indicating the position of the environment where the data are stored, relative to the environment where the function is called. Useful only if `storeData` is `FALSE`
- `...`: additional graphical parameters (see `adegpar` and `trellis.par.get`)

Details
The `fac` factor is used to display several trajectories: each level of `fac` is a specific trajectory.
Value

An object of class ADEg (subclass Tr.traject) or ADEgS (if showposition is TRUE, if add is TRUE and/or if facets are used).
The result is displayed if plot is TRUE.

Author(s)

Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also

Tr.traject ADEg.Tr

Examples

```r
ex1 <- matrix(c(51.88, 32.55, 15.57, 44.94, 34.59, 20.47, 25.95, 39.15, 34.9,
                 37.87, 43.19, 18.94, 34.2, 43.32, 22.48, 16.13, 42.18, 41.69,
                 7.76, 70.93, 21.31, 6.22, 65.96, 27.82, 6.44, 57.06, 36.5,
                 37.24, 32.45, 30.31, 16.09, 31.22, 52.69, 6.54, 24.68, 68.78), ncol = 3, byr = TRUE)
ex1 <- as.data.frame(ex1)
names(ex1) <- c("agr", "ouv", "ter")
com <- as.factor(rep(c("Gig", "Lun", "Gan", "Mat"), c(3, 3, 3, 3)))
rec <- as.factor(rep(c("68", "75", "82"), 4))
row.names(ex1) <- paste(com, rec, sep = "")
tri1 <- triangle.traject(ex1, fac = com, showposition = FALSE, pgrid.draw = FALSE,
                          col = TRUE, axis.text = list(cex = 0))
```

Description

This function performs a zoom on a ADEg.S1 or ADEg.S2 displayed object.

Usage

`zoom(object, zoom, center)`

Arguments

- `object`: a ADEg.S1 or ADEg.S2 object
- `zoom`: a numeric value to zoom in (if zoom > 1) or out (if zoom < 1)
- `center`: a numeric value (if object is a ADEg.S1 object) or a two-length vector (if object is a ADEg.S2 object) as a reference point to zoom (in or out). If it is missing, the displayed center point is used.
Value
Updated display after zoom.

Author(s)
Alice Julien-Laferriere, Aurelie Siberchicot <aurelie.siberchicot@univ-lyon1.fr> and Stephane Dray

See Also
ADEg.S2 ADEg.S1

Examples
```r
data(olympic, package = "ade4")
dudi1 <- ade4::dudi.pca(olympic$tab, scan = FALSE)
g <- s.corcircle(dudi1$co, lab = names(olympic$tab), fullcircle = TRUE, psub.text = "data:olympic")
zoom(g, 0.5)
zoom(g, 2, center = c(-0.4, 0.8))
```
Index

*Topic `aplot`
  - `add.ADEg`, 5
  - `addhist`, 6
  - `addline`, 8
  - `addpoint`, 9
  - `addsegment`, 10
  - `addtext`, 11
  - `adeg.panel.hist`, 16
  - `adeg.panel.join`, 17
  - `adeg.panel.label`, 18
  - `adeg.panel.nb`, 19
  - `adeg.panel.Spatial`, 21
  - `adeg.panel.values`, 23
  - `insert`, 52
  - `plotEig`, 62
  - `s.arrow`, 66
  - `s.class`, 67
  - `s.corcircle`, 69
  - `s.density`, 70
  - `s.distri`, 72
  - `s.image`, 73
  - `s.label`, 75
  - `s.logo`, 77
  - `s.match`, 78
  - `s.traject`, 81
  - `s.value`, 83
  - `s1d.barchart`, 92
  - `s1d.boxplot`, 93
  - `s1d.class`, 95
  - `s1d.curve`, 96
  - `s1d.curves`, 97
  - `s1d.density`, 99
  - `s1d.distri`, 100
  - `s1d.dotplot`, 102
  - `s1d.gauss`, 103
  - `s1d.hist`, 104
  - `s1d.interval`, 106
  - `s1d.label`, 107
  - `s1d.match`, 108
  - `setlimits1D`, 127
  - `sortparamADEg`, 128
  - `table.image`, 135
  - `table.value`, 137
  - `triangle.class`, 145
  - `triangle.label`, 147
  - `triangle.match`, 148
  - `triangle.trajec`, 150

*Topic `classes`
  - `ADEg-class`, 12
  - `ADEg.C1-class`, 14
  - `ADEg.S1-class`, 24
  - `ADEg.S2-class`, 26
  - `ADEg.T-class`, 28
  - `ADEg.Tr-class`, 30
  - `ADEgS-class`, 37
  - `C1.barchart-class`, 39
  - `C1.curve-class`, 41
  - `C1.density-class`, 42
  - `C1.dotplot-class`, 44
  - `C1.gauss-class`, 45
  - `C1.hist-class`, 47
  - `C1.interval-class`, 48
  - `S1.boxplot-class`, 85
  - `S1.class-class`, 86
  - `S1.distri-class`, 88
  - `S1.label-class`, 89
  - `S1.match-class`, 91
  - `S2.arrow-class`, 110
  - `S2.class-class`, 111
  - `S2.corcircle-class`, 113
  - `S2.density-class`, 114
  - `S2.distri-class`, 116
  - `S2.image-class`, 117
  - `S2.label-class`, 119
  - `S2.logo-class`, 120
  - `S2.match-class`, 122
  - `S2.trajec-class`, 123
  - `S2.value-class`, 125
INDEX

addsegment, ADEg-method (addsegment), 10
addsegment, ADEgS-method (addsegment), 10
addsegment-methods (addsegment), 10
addtext, 11
addtext, ADEg-method (addtext), 11
addtext, ADEgS-method (addtext), 11
addtext-methods (addtext), 11
ADEg (ADEg-class), 12
ADEg-class, 12
ADEg.C1, 40–49, 63, 93, 97, 98, 100, 103–105, 107
ADEg.C1 (ADEg.C1-class), 14
ADEg.C1-class, 14
adeg.panel.edges (adeg.panel.nb), 19
adeg.panel.hist, 16
adeg.panel.join, 17
adeg.panel.label, 18
adeg.panel.nb, 19
adeg.panel.Spacial, 21
adeg.panel.values, 23
ADEg.S1, 85–92, 94, 96, 101, 108, 109, 152
ADEg.S1 (ADEg.S1-class), 24
ADEg.S1-class, 24
ADEg.S2, 8, 67, 68, 70, 72, 73, 75, 76, 78, 79, 82, 84, 110–115, 117–124, 126, 152
ADEg.S2 (ADEg.S2-class), 26
ADEg.S2-class, 26
ADEg.T, 131–136, 138
ADEg.T (ADEg.T-class), 28
ADEg.T-class, 28
ADEg.Tr, 140–146, 148, 149, 151
ADEg.Tr (ADEg.Tr-class), 30
ADEg.Tr-class, 30
ADEgORADEgSORTrellis, 40, 41, 43, 44, 46, 47, 49, 85, 87, 88, 90, 91, 110, 112, 113, 115, 117, 120–122, 124, 126, 131, 133, 134, 140, 142, 143, 145
ADEgORTrellis, 40, 41, 43, 44, 46, 47, 49, 85, 87, 88, 90, 91, 110, 112, 113, 115, 117, 118, 120–122, 124, 126, 131, 133, 134, 140, 142, 143, 145
adegraphics (adegraphics-package), 4
adegraphics-package, 4
ADEgS, 5, 6, 8–12, 14, 36, 37, 39, 50, 53, 130
ADEgS-class, 37
biplot (plot), 56
bkde, 16, 17, 99

C1.barchart, 16, 63, 93
C1.barchart (C1.barchart-class), 39
C1.barchart-class, 39
C1.curve, 16, 97
C1.curve (C1.curve-class), 41
C1.curve-class, 41
C1.curves, 41, 98
C1.curves (C1.curve-class), 41
C1.curves-class (C1.curve-class), 41
C1.density, 16, 100
C1.density (C1.density-class), 42
C1.density-class, 42
C1.dotplot, 16, 103
C1.dotplot (C1.dotplot-class), 44
C1.dotplot-class, 44
C1.gauss, 16, 104
C1.gauss (C1.gauss-class), 45
C1.gauss-class, 45
C1.hist, 16, 105
C1.hist (C1.hist-class), 47
C1.hist-class, 47
C1.interval, 16, 107
C1.interval (C1.interval-class), 48
C1.interval-class, 48
cbindAEG, 50
cbindADEg, ADEgORADEgSORTrellis, ADEgORADEgSORTrellis-methods
(cbindADEg), 50
cbindADEg-methods (cbindADEg), 50
changelatticetheme, 51
getcall (getcall-methods), 52
getcall, ADEg-method (ADEg-class), 12
getcall, ADEgS-method (ADEgS-class), 37
getcall-methods, 52
getgraphics (ADEgS-class), 37
getgraphics, ADEgS-method (ADEgS-class), 37
getlatticecall (ADEg-class), 12
getlatticecall, ADEg-method (ADEg-class), 12
getparameters (ADEg-class), 12
getparameters, ADEg-method (ADEg-class), 12
getpositions (ADEgS-class), 37
getpositions, ADEgS-method (ADEgS-class), 37
getstats (ADEg-class), 12
getstats, ADEg-method (ADEg-class), 12
getstats-methods (ADEg-class), 12
gettrellis (ADEg-class), 12
gettrellis, ADEg-method (ADEg-class), 12
gettrellis, ADEg.C1-method (ADEg.C1-class), 14
gettrellis, ADEg.S1-method (ADEg.S1-class), 24
gettrellis, ADEg.S2-method (ADEg.S2-class), 26
gettrellis, ADEg.T-method (ADEg.T-class), 28
gettrellis, ADEg.TR-method (ADEg.TR-class), 30
gettrellis-methods (ADEg-class), 12
hist, 16, 17, 105
insert, 14, 39, 52
insert, ADEgOrtrellis, ADEg-method (insert), 52
insert, ADEgOrtrellis, ADEgS-method (insert), 52
insert, ADEgOrtrellis, missing-method (insert), 52
insert, ADEgS, ADEg-method (insert), 52
insert, ADEgS, ADEgS-method (insert), 52
insert, ADEgS, missing-method (insert), 52
insert-methods (insert), 52

kplot (plot), 56
kplotsepan.coa (plot), 56

lattice, 5
layout, 54
layout2position, 54
length, ADEgS-method (ADEgS-class), 37
names, ADEgS-method (ADEgS-class), 37
names<-, ADEgS, character-method (ADEgS-class), 37

panel (panel-methods), 55
panel, C1.barchart-method (C1.barchart-class), 39
panel, C1.curve-method (C1.curve-class), 41
panel, C1.curves-method (C1.curve-class), 41
panel, C1.density-method (C1.density-class), 42
panel, C1.dotplot-method (C1.dotplot-class), 44
panel, C1.gauss-method (C1.gauss-class), 45
panel, C1.hist-method (C1.hist-class), 47
panel, C1.interval-method (C1.interval-class), 48
panel, S1.boxplot-method (S1.boxplot-class), 85
panel, S1.class-method (S1.class-class), 86
panel, S1.distri-method (S1.distri-class), 88
panel, S1.label-method (S1.label-class), 89
panel, S1.match-method (S1.match-class), 91
panel, S2.arrow-method (S2.arrow-class), 110
panel, S2.class-method (S2.class-class), 111
panel, S2.corcircle-method (S2.corcircle-class), 113
panel, S2.density-method (S2.density-class), 114
panel, S2.distri-method (S2.distri-class), 116
panel, S2.image-method (S2.image-class), 117
panel, S2.label-method (S2.label-class), 119
panel, S2.logo-method (S2.logo-class), 120
panel, S2.match-method (S2.match-class), 122
panel, S2.trajec-method (S2.trajec-class), 123
panel, S2.value-method (S2.value-class), 125
panel, T. cont-method (T. cont-class), 130
panel, T. image-method (T. image-class), 132
panel, T. value-method (T. value-class), 133
panel, Tr. class-method (Tr. class-class), 139
panel, Tr. label-method (Tr. label-class), 141
panel, Tr. match-method (Tr. match-class), 142
panel, Tr. traject-method
(Tr. traject-class), 144
panel-methods, 55
panel.abline, 9
panel.segments, 10
panelbase (ADEg-class), 12
panelbase, ADEg-method (ADEg-class), 12
panelbase, ADEg.C1-method
(ADEg.C1-class), 14
panelbase, ADEg.S1-method
(ADEg.S1-class), 24
panelbase, ADEg.S2-method
(ADEg.S2-class), 26
panelbase, ADEg.T-method (ADEg.T-class), 28
panelbase, ADEg.Tr-method
(ADEg.Tr-class), 30
panelbase-methods (ADEg-class), 12
par, 35
plot, 56
plot, ADEg, ANY-method (ADEg-class), 12
plot, ADEg-method (ADEg-class), 12
plot, ADEgS, ANY-method (ADEgS-class), 37
plot, ADEgS-method (ADEgS-class), 37
plot.inertia, 60
plot.nb, 20
plotEig, 62
pointLabel, 19
prepare (prepare-methods), 64
prepare, ADEg.C1-method (ADEg.C1-class), 14
prepare, ADEg.S1-method (ADEg.S1-class), 24
prepare, ADEg.S2-method (ADEg.S2-class), 26
prepare, ADEg.T-method (ADEg.T-class), 28
prepare, ADEg.Tr-method (ADEg.Tr-class), 30
prepare, C1.barchart-method
(C1.barchart-class), 39
prepare, C1.curve-method
(C1.curve-class), 41
prepare, C1.density-method
(C1.density-class), 42
prepare, C1.dotplot-method
(C1.dotplot-class), 44
prepare, C1.gauss-method
(C1.gauss-class), 45
prepare, C1.hist-method (C1.hist-class), 47
prepare, C1.interval-method
(C1.interval-class), 48
prepare, S1.boxplot-method
(S1.boxplot-class), 85
prepare, S1.class-method
(S1.class-class), 86
prepare, S1.distri-method
(S1.distri-class), 88
prepare, S1.label-method
(S1.label-class), 89
prepare, S1.match-method
(S1.match-class), 91
prepare, S2.arrow-method
(S2.arrow-class), 110
prepare, S2.class-method
(S2.class-class), 111
prepare, S2.corcircle-method
(S2.corcircle-class), 113
prepare, S2.density-method
(S2.density-class), 114
prepare, S2.distri-method
(S2.distri-class), 116
prepare, S2.image-method
(S2.image-class), 117
prepare, S2.label-method
(S2.label-class), 119
prepare, S2.logo-method (S2.logo-class), 120
prepare, S2.match-method
(S2.match-class), 122
prepare, S2.traject-method
(S2.traject-class), 123
prepare, S2.value-method
(S2.value-class), 125
prepare, T.image-method (T.image-class), 132
prepare, T.value-method (T.value-class), 133
prepare, Tr.class-method (Tr.class-class), 139
prepare, Tr.label-method (Tr.label-class), 141
prepare, Tr.match-method (Tr.match-class), 142
prepare, Tr.trajec-method (Tr.trajec-class), 144
prepare-methods, 64
print, ADEg-method (ADEg-class), 12
print, ADEgS-method (ADEgS-class), 37
printSuperpose, ADEgORtrellis, ADEgORtrellis-method
(ADEg-class), 12
rbindADEg (cbindADEg), 50
rbindADEg, ADEgORADEgORtrellis, ADEgORADEgORtrellis-method
(cbindADEg), 50
rbindADEg-methods (cbindADEg), 50
s.arrow, 66, 111
s.class, 67, 112
s.corcircle, 69, 114
s.distri, 72, 117
s.image, 73, 119
s.label, 75, 120
s.logo, 77, 121
s.match, 78, 123
s.Spatial, 80
s.trajec, 81, 124
s.value, 83, 126
S1.boxplot, 26, 94
S1.boxplot (S1.boxplot-class), 85
S1.boxplot-class, 85
S1.class, 26, 96
S1.class (S1.class-class), 86
S1.class-class, 86
S1.distri, 26, 101
S1.distri (S1.distri-class), 88
S1.distri-class, 88
S1.label, 26, 108
S1.label (S1.label-class), 89
S1.label-class, 89
S1.match, 26, 109
S1.match (S1.match-class), 91
S1.match-class, 91
s1d.barchart, 40, 92
s1d.boxplot, 86, 93
s1d.class, 87, 95
s1d.curve, 42, 96
s1d.curves, 42, 97
s1d.density, 43, 99
s1d.distri, 89, 100
s1d.dotplot, 45, 102
s1d.gauss, 46, 103
s1d.hist, 48, 104
s1d.interval, 49, 106
s1d.label, 90, 107
s1d.match, 92, 108
S2.arrow, 28, 67
S2.arrow (S2.arrow-class), 110
S2.arrow-class, 110
S2.class, 28, 68
S2.class (S2.class-class), 111
S2.class-class, 111
S2.corcircle, 28, 70
S2.corcircle (S2.corcircle-class), 113
S2.corcircle-class, 113
S2.density, 28, 72
S2.density (S2.density-class), 114
S2.density-class, 114
S2.distri, 28, 73
S2.distri (S2.distri-class), 116
S2.distri-class, 116
S2.image, 28, 75
S2.image (S2.image-class), 117
S2.image-class, 117
S2.label, 28, 76, 80
S2.label (S2.label-class), 119
S2.label-class, 119
S2.logo, 28, 78
S2.logo (S2.logo-class), 120
S2.logo-class, 120
S2.match, 28, 79
S2.match (S2.match-class), 122
S2.match-class, 122
S2.trajec, 28, 82
S2.trajec (S2.trajec-class), 123
S2.trajec-class, 123
S2.value, 28, 84
S2.value (S2.value-class), 125
S2.value-class, 125
scatter (plot), 56
score (plot), 56
score.inertia (plot.inertia), 60
scatterplot (plot), 56
setlatticecallADEg.C1-method
    (ADEg.C1-class), 14
setlatticecallADEg.S1-method
    (ADEg.S1-class), 24
setlatticecallADEg.S2-method
    (ADEg.S2-class), 26
setlatticecallADEg.T-method
    (ADEg.T-class), 28
setlatticecallS1.boxplot-method
    (S1.boxplot-class), 85
setlimits1D, 127
setlimits2D (setlimits1D), 127
show, ADEg-method (ADEg-class), 12
show, ADEgS-method (ADEgS-class), 37
show.settings, 51
sortparamADEg, 128
sortparamADEgS (sortparamADEg), 128
sp.grid, 22, 80
sp.lines, 22, 80
sp.polygons, 22, 80
spplot, 22, 80
superpose, 6, 14, 39, 129
superpose, ADEgORtrellis, ADEgORtrellis, ANY, ANY-method
    (superpose), 129
superpose, ADEgS, ADEgORtrellis, ANY-method
    (superpose), 129
superpose, ADEgS, ADEgORtrellis, missing, ANY-method
    (superpose), 129
superpose, ADEgS, ADEgORtrellis, numeric, ANY-method
    (superpose), 129
superpose, ADEgS, ADEgORtrellis, numeric, logical-method
    (superpose), 129
superpose, ADEgS, ADEgS, missing, ANY-method
    (superpose), 129
superpose-methods (superpose), 129
T.cont, 135, 138
T.cont (T.cont-class), 130
T.cont-class, 130
T.image, 29, 136
T.image (T.image-class), 132
T.image-class, 132
T.value, 29, 131, 132, 138
T.value (T.value-class), 133
T.value-class, 133
table.image, 133, 135
table.value, 132, 135, 137
Tr.class, 31, 146
Tr.class (Tr.class-class), 139
Tr.class-class, 139
Tr.label, 31, 148
Tr.label (Tr.label-class), 141
Tr.label-class, 141
Tr.match, 31, 149
Tr.match (Tr.match-class), 142
Tr.match-class, 142
Tr.traject, 31, 151
Tr.traject (Tr.traject-class), 144
Tr.traject-class, 144
trellis.par.set, 51
triangle.class, 141, 145
triangle.label, 142, 147
triangle.match, 144, 148
triangle.traject, 145, 150
update, ADEg (ADEg-class), 12
update, ADEg-method (ADEg-class), 12
update, ADEgS (ADEgS-class), 37
update, ADEgS-method (ADEgS-class), 37
update, ADEgS-method (ADEgS-class), 37
update, ADEgS-method (ADEgS-class), 37
zoom, 26, 28, 151
zoom, ADEg.S1, numeric, missing-method
    (zoom), 151
zoom, ADEg.S1, numeric, numeric-method
    (zoom), 151
zoom, ADEg.S2, numeric, missing-method
    (zoom), 151
zoom, ADEg.S2, numeric, numeric-method
    (zoom), 151
zoom-methods (zoom), 151