## Package ‘agsemisc’

August 29, 2016

### Version
1.3-1

### Date
2014-07-31

### Title
Miscellaneous plotting and utility functions

### Maintainer
Lutz Prechelt &lt;prechelt@inf.fu-berlin.de&gt;

### Depends
R (>= 2.12.0), lattice

### Imports
grid, MASS

### Description
High-featured panel functions for bwplot and xyplot,
some plot management helpers, various convenience functions

### License
GPL-2

### URL
https://github.com/prechelt/agsemisc

### Author
Lutz Prechelt [aut, cre]

### NeedsCompilation
no

### Repository
CRAN

### Date/Publication
2014-07-31 15:38:08

### R topics documented:

<table>
<thead>
<tr>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>misc helpers</td>
<td>2</td>
</tr>
<tr>
<td>panel.bwstrip</td>
<td>3</td>
</tr>
<tr>
<td>panel.xy</td>
<td>6</td>
</tr>
<tr>
<td>plotf</td>
<td>7</td>
</tr>
<tr>
<td>plothelpers</td>
<td>9</td>
</tr>
<tr>
<td>statshelpers</td>
<td>10</td>
</tr>
</tbody>
</table>

### Index
11
Description

eqc(x, number=4, overlap=1/sqrt(length(x))) an abbreviation for equal.count with useful defaults. Often useful for Lattice plots.
grep.b(pattern, x) performs a grep, but returns the result as a logical (boolean, thus the .b) vector along x rather than as an integer vector indices. Often useful for Lattice subset calls.
grep.s(pattern, x) performs grep(pattern, x, value=TRUE) and thus returns the matched strings rather than their indices.
ordervg(x, by) orders the observations from vector x using the equal-weighted ranking criteria given in matrix by (which has length(x) rows and n columns), a by vector is coerced. The ranks computed from each matrix column are summed row-wise, the resulting vector is sorted according to these rank sums, and returned. This function can be used to pair elements from two vectors (by ranking them both by equivalent criteria).
tablenA abbreviation for table with an additional argument exclude = NULL so that NA and NaN values are tabulated as well.
or.else(x, alternative = 0) returns x where available and alternative where is.na(x). alternative can be a scalar or vector and should have the same mode as x.
printf(x, digits) prints name and value of x. If the optional argument digits is used, x will be coerced to numeric and formatted with digits digits precision. Useful for quick-and-dirty debugging output.
tracebck() is like traceback except that it returns (rather than prints) only the first line of each frame. This is useful because the argument lists of calls often contain large data objects which clutter the output of traceback and make it very difficult to read.

Details

Type the name of a function to see its source code for details.

Author(s)

Lutz Prechelt <prechelt@inf.fu-berlin.de>

See Also

equal.count, grep, order, ifelse, table, traceback.

Examples

## Not run: plot(rnorm(8)-runif(8),xlim=c(10,NA))
## Not run: traceback()
## Not run: tracebck()
```
data(iris)
xyplot(Sepal.Width ~ Sepal.Length | eqc(Sepal.Length),
       subset=grepNb("v",Species), data=iris)

x = c(4,9,NA,4,4,27,NA,27)
table(or.else(x,88))
table(x)
tableNA(x)

printn(levels(iris$Species))
```

### panel.bwstrip

**Flexible panel.bwplot replacement**

#### Description

Box plots with added stripplots, densityplots, mean/stderr marker etc.

#### Usage

```r
panel.bwstrip(x, y, groups, subscripts, pch, col,
               box.ratio = 5, varwidth = FALSE, whiskerpos = 0.1, logbase,
               type="mean, mad, strip, N, grid",
               densityplot=expression(density(x, cut=2)), strip.limit=100,
               seplines = NULL, N.label = "N=\% \n",
               extend = TRUE,
               levels fos = NULL, ...)
```

#### Arguments

- **x**: the data, as in `panel.bwplot`
- **y**: the number of the box, as in `panel.bwplot`
- **groups**: a factor indicating a partitioning of the data x. Relevant for the pch and col arguments.
- **subscripts**: Argument internally used by Lattice to realize the groups functionality.
- **pch**: A integer vector, character vector or list that indicates the value to be used as pch (plot character) for the stripplot for each level of groups. You will need a list if you want to mix plot symbols (indicated by integers) with plot characters (indicated by characters). There is a simpler method to specify any of these cases, namely a single string with entries separated by commas. One-digit numeric entries will evaluate to a plot character, two-digit numeric entries will evaluate to an integer plot symbol. For instance, "1,w,13,2,02" will evaluate into `list("1", "w", 13, "2", 2)`. Defaults to all `trellis.par.get("superpose.symbol")$pch`. Note that with groups=gg you can often use pch=levels(gg), because Lattice will ignore all but the first character of a string.
col
Analogous to pch, but indicating color. Integers indicate color numbers, strings indicate color names, both can be mixed in the comma-separated string format. Defaults to all trellis.par.get("superpose.symbol")$col.

box.ratio
Like in panel.bwplot, but the different default value withstands the call default of 1 that is imposed by Lattice (as of R 2.2).

varwidth
vary box thickness according to number of data values, as in panel.bwplot

whiskerpos
With large values in the range 1..Inf, whiskerpos is equivalent to the coef argument to panel.bwplot (or really boxplot.stats), i.e. each whisker is at the farthest data value that is at most whiskerpos times the interquartile range (or box width) away from the box. Only the values 1.5, 2 or 3 are common. In contrast, with small values w in the range 0..0.25, the whiskers will indicate the fixed quantiles w and 1-w. This is easier to explain to non-statisticians and often more appropriate for larger samples. So whiskerpos==0 will produce whiskers at the min and max of the data, the default will indicate the 10-percentile and the 90-percentile, and whiskerpos==0.25 makes the whiskers disappear.

logbase
An argument logbase=b indicates that the high-level plot is using a log scale axis to base b and hence the data is logarithmic rather than real and needs to be converted back before computing mean, interpolating quantiles etc.

type
Declares which boxplot elements to include in the plot. Is either a comma-separated string of element names (as shown in the default) or a vector of such names. The elements have the following meaning: mean plots something like –M– indicating the mean and its standard error. mad will indicate the stderror of the median (median absolute deviation, as computed by mad divided by sqrt(n)) as a line left and right of the median dot. strip will produce a stripplot of the individual data points, scattered vertically to make similar values more visible. density will add a densityplot and a support line (extending along the range of the data). The plot can be customized via the densityplot argument. N will indicate the number of datapoints according to the N.label argument. grid will draw dotted vertical lines aligned with x-axis labels. Elements not mentioned will be left out of the plot. The box and median dot are always included (this is a boxplot, after all), the whiskers can be suppressed by whiskerpos=0.25.

densityplot
Relevant if "density" is mentioned in the type argument. Must be an expression describing a call to density that concerns the data vector x (an uppercase x!), which will be the data for the current boxplot for each evaluation of the expression. If a densityplot appears, the boxplot will not be color-filled.

strip.limit
If T, will reduce the stripplot to only the outliers, i.e., the values beyond the whiskers. If an integer, will suppress the stripplot entirely if there are more than this many values in the current boxplot.

seplines
A vector of vertical positions where horizontal lines will be drawn to separate the boxplots into groups. Position 1.5, 2.5 etc. is above the lowest, second-lowest boxplot etc.

N.label
A string such as " number of values underlying the boxplot and the resulting string is printed at the right of the plot iff N is mentioned in the type argument. Use trailing blanks and newlines to adjust positioning.

extend
If TRUE, will print to console some statistics for each sample: the quantiles (0, 0.25, 0.5, 0.75, 1), mean, quartile ratio (or at least interquartile range) and
the number of data points. If extend is a function, it will be called with four arguments: data vector x, current y, groups, subscripts.

levels.fos weird stuff, but means the same as in \texttt{panel.bwplot}

... all other arguments will be ignored.

Details

A lattice panel function to be used with bwplot. Can draw a boxplot plus stripplot plus densityplot, indicate groups, mark the mean and its stderr, report N, and more. Graphical parameters are controlled by the \texttt{trellis.par.set} parameters box.rectangle, box.umbrella, superpose.symbol (which is non-standard), and reference.line. The \texttt{horizontal=F} option available in \texttt{panel.bwplot} is not supported here.

Author(s)

Lutz Prechelt <prechelt@inf.fu-berlin.de>

See Also

\texttt{a.resetplotparams, plotf, panel.xy}.

Examples

# set grid.prompt(TRUE) to see each plot separately (click graphics window)
data(iris)
a.resetplotparams()
print(bwplot(\texttt{Species~Sepal.Length}, data=iris, panel=\texttt{panel.bwstrip}))
# A plot including a density plot:
print(bwplot(\texttt{Species~Sepal.Length}, data=iris, panel=\texttt{panel.bwstrip},
\texttt{type}="mean,strip,density"))
# A customized plot:
print(bwplot(\texttt{~Sepal.Length}, data=iris, panel=\texttt{panel.bwstrip},
groups=\texttt{Species}, pch=\texttt{levels(iris$Species)}, \texttt{strip.limit}=200,
\texttt{type}="mean,strip,density",
densityplot=\texttt{expression(density(X, cut=1)))})
# A conventional-style plot:
print(bwplot(\texttt{Species~Sepal.Length}, data=iris, panel=\texttt{panel.bwstrip},
\texttt{type}="mean,strip.grid", \texttt{strip.limit}=TRUE, \texttt{whiskerpos}=1.5,
densityplot=\texttt{expression(density(X, cut=1)))})
# A plot showing some other features:
print(bwplot(\texttt{cut(Sepal.Width,4)~Sepal.Length}, data=iris, panel=\texttt{panel.bwstrip},
groups=\texttt{Species}, \texttt{varwidth}=TRUE, box.ratio=20,
\texttt{which}="strip,N", \texttt{strip.limit}=50, \texttt{pch}="1,2,3"))
**Description**

`panel.xy` is used like `panel.xyplot` but offers more features. In particular, it can add an lqs resistant regression line besides a normal one, can plot a prediction interval around the standard regression line, can add text indicating correlation and number of data points, and automatically uses different colors and line styles.

**Usage**

```r
panel.xy(x, y, type = "p,r,r.pred,r=,N=,grid",
         r.min = 0.5, level = 0.8, slope=0.0, intercept=0.0,
         unicolor = FALSE, ...)
```

**Arguments**

- `x` Vector of x coordinates for the plot.
- `y` Vector of x coordinates for the plot. Pairs (x,y) containing NAs will be filtered out.
- `type` A character vector (or single comma-separated string) of options to use for plotting. Entries "p", "l", "b", "o", "h", "s", "S" are handled just as `plot` or `panel.xyplot` would (namely as points, lines, both, both overplotted, histogram lines, stair steps, stair steps with vertical preference). "g" adds a coordinate grid via `panel.grid`. "r" adds a lm regression line. See `r.min` and `unicolor` for details. The line is suppressed if there are less than 8 points. "r.pred" adds a prediction interval around the "r" line. Implies "r". See `level` for details. "r.conf" adds a confidence interval around the "r" line. This concerns the regression line itself rather than the predictions it makes and is not often of much interest. Implies "r". `level` applies just like it does for "r.pred". "lqs" adds an lqs resistant regression line. Roughly speaking, this produces approximately the best regression line that is possible if it is allowed to throw the most difficult 50% of the points away. The line is suppressed if there are less than 8 points. "smooth", "loess", "lowess" are all synonymous and add a locally weighted regression via `panel.loess`, whose configuration parameters will be passed to the call if given. The line is suppressed if there are less than 8 points. "abline" adds an arbitrary line (described by parameters `slope=` and `intercept=`) via `panel.abline`. `col`, `lty` and `lwd` are taken from `trellis.par.get("ab.line")` (non-standard extension as introduced by `a.resetplotparams`), "rug" adds a rug plot via `panel.rug`. "cor" prints the correlation coefficient computed by `cor` in the lower right corner of the plot. "N" prints the number of non-NA data points in the lower right corner of the plot. "v" (verbose) prints various statistics regarding the plot to the console.
- `r.min` Even if type "r" is requested, the regression line will appear only if `abs(cor(x,y) >= r.min)`.
- `level` The confidence level of the intervals plotted by type "r.conf" and type "r.pred".
The parameters of the fixed line inserted by means of panel.abline if type contains "abline".

The colors used for the lines drawn by types "r", "r.pred", "r.conf", "loess" are normally taken from trellis.par.get("regression.line") (as introduced by a.resetplotparams). However, if you use xyplot with panel=panel.superpose, panel.groups=party.line, you must specify unicolor=TRUE. The regression lines will then be plotted in the style given by trellis.par.get("superpose.line") according to the respective group.

Further parameters as needed for instance by panel.loess and lqs.

The types "p", "l", "b", "o", "h", "s", "S" are in fact processed by panel.xyplot. Specifying col, lty, lwd will not work, as there are calls that include both ... and some of these.

Author(s)

Lutz Prechelt, prechelt@inf.fu-berlin.de

See Also

a.resetplotparams, trellis.par.get, panel.xyplot, panel.superpose, panel.loess, lm, lqs.

Examples

# set grid.prompt(TRUE) to see each plot separately (click graphics window)
data(iris)
a.resetplotparams()
print(xyplot(Sepal.Width ~ Sepal.Length|Species, data=iris,
panel=panel.xy, type="p,grid,v,lqs,r.pred,loess")
print(xyplot(Sepal.Width ~ Sepal.Length, data=iris, groups=Species,
panel=panel.superpose, panel.groups=panel.xy,
  type=c("p","grid","v","loess"), unicolor=TRUE))

plotf

Plot to multiple devices

Description

Sends the same plot to one or several devices in a uniform manner. Helpful for preparing a script that can be configured to produce graphics in several different formats, e.g. small bitmaps for web pages, large bitmaps for slides, and pdf, eps or wmf for inclusion in documents.
Usage

plotf(theplot, file="plot%03d", type=c("active"),
      size=c(6,4,10,96), prepare=a.resetplotparams, ...)

Arguments

theplot An object representing the plot. This can either be an expression such as
expression(plot(x), points(pp)) or a Lattice object such as xyplot(y~x). The object
will be evaluated (if an expression) or print-ed (if a Lattice object) once on each device to be used. Note that this repeated processing may
be inefficient if the plotting process itself (rather than just the resulting plot) is
extremely complex. In such (rare) cases, you may want to consider manually us-
ing dev.print instead. plotf uses repeated processing rather than dev.print
to ensure that the plotting function works as intended even if it discriminates
between different devices.

file The basename of the file to be produced. A suitable suffix of .eps, .ps, .pdf, .png, .jpg,
or .wmf, respectively, will be appended for the actual files produced. The file-
name is ignored for the "active" device.

type A character vector in which each entry represents one device to be used. Al-
lowed values are "active", "current", "eps", "ps", "pdf", "png", "jpg", "jpeg", "wmf"
and they represent the use of dev.cur, dev.cur, dev.copy2eps, postscript, pdf, png, jpeg, jpeg
and win.metafile, respectively. "eps" is ignored unless "active" (or "current")
is also present.

size A numeric vector of 1 to 4 components to which missing values will be added
from the default value. Components 1 and 2 are width and height (in inches)
of the plot to be produced. They will be supplied to the devices' width and
height arguments, respectively. Component 3 is the base character size and
will be supplied to the pointsize argument. For the bitmap devices, it will be
corrected for resolution (which is indicated (in dpi) by component 4), because
those devices always assume a resolution of 72dpi. Component 4 will also be
supplied as the res argument to the bitmap devices.

prepare A function without parameters that will be called each time after a new device
has been opened and before the plot is produced.

... All further arguments will be handed over to each(!) of the device opening
functions

Details

The function uses on.exit to close a device correctly even if the plot function fails. png is almost
always preferable to jpeg. Note that expressions can contain multiple calls by separating them with
commas.

Value

invisible()
plothelpers

Author(s)
Lutz Prechelt <prechelt@inf.fu-berlin.de>

See Also
a.resetplotparams, dev.cur, dev.copy2eps, postscript, pdf, png, jpeg.

Examples

x = runif(TP)*4
y = x + runif(TP)
plotf(expression(plot(x, y, type="p"))) # plots to active device only
plotf(xyplot(y~x), "myplot", type=c("active", "png", "pdf"), size=c(3,3))

data(iris)
a.resetplotparams()
print(bwplot(Species~Sepal.Length, data=iris, panel=panel.bwstrip, prepanel=prepanel.0))
plotfit(lm(Sepal.Length~Sepal.Width, data=iris))

plothelpers

Plot helper functions

Description

a.resetplotparams() sets colors and plotting parameters suitable for document production.
plotfit(fit, ...) plots an lm object using mfrow=c(2,3)
prepanel.0 Lattice prepanel function that forces that zero be included on both axes. Does nothing for non-numeric axes (such as the factor axis in bwplot).

Details

Type the name of a function to see its source code for details.

Author(s)
Lutz Prechelt <prechelt@inf.fu-berlin.de>

See Also
trellis.par.set, plot.lm, Lattice, prepanel.lmline.

Examples

data(iris)
a.resetplotparams()
print(bwplot(Species~Sepal.Length, data=iris, panel=panel.bwstrip, prepanel=prepanel.0))
plotfit(lm(Sepal.Length~Sepal.Width, data=iris))
Simple stats and tabulation helper functions

Description

- `iqr(x)` interquartile range of numeric vector
- `qr(x)` ratio of 3rd to 1st quartile of numeric vector
- `proportion.test(x1, x2, y1, y2, totals=FALSE)` compares \(x_1/x_2\) to \(y_1/y_2\) using fisher.test and prints the result. totals=TRUE means the supplied \(x_2\) is in fact \((x_1+x_2)\); ditto for \(y_2\).
- `findcorrelations(df, vars1=names(df), vars2=vars1, min.cor=0.5)` computes correlation (of values and of ranks) for each pair of variables from \((vars1, vars2)\), sorts them by size and returns the large ones (along with descriptive names) as a vector. Ignores NAs.
- `printextremes(df, vars, largest=5, showalso=NULL` given variable names \(a,b,c\) from dataframe \(df\), prints the tuples \(a,b,c\) with the 5 largest values of \(a\). Ditto for \(b\) and for \(c\). largest can be a vector (along \(vars\)) and negative values print smallest instead of largest. Factors are moved from \(vars\) to \(showalso\).

Details

Type the name of a function to see its source code for details.

Author(s)

Lutz Prechelt <prechelt@inf.fu-berlin.de>

See Also

cor, rank, quantile, summary.

Examples

```r
set.seed(17)
base = rnorm(100)
a = floor(base*10)
b = floor(a+runif(100, -10, 11))
c=floor(base)
d=ordered(floor(b/8)) # allows for rank correlation only
df=data.frame(a=a, b=b, c=c, d=d)
a.findcorrelations(df, min.cor=0.85)
a.printextremes(iris, vars=c("Species", "Sepal.Length", "Petal.Width"),
largest=c(3, -4, -5), showalso=c("Petal.Length"))
```
Index

*Topic **aplot**
  panel.bwstrip, 3
  panel.xy, 6
*Topic **color**
  plothelpers, 9
*Topic **correlation**
  statshelpers, 10
*Topic **device**
  plotf, 7
*Topic **dplot**
  plothelpers, 9
*Topic **manip**
  misc helpers, 2
*Topic **quantile**
  statshelpers, 10
*Topic **quartile**
  statshelpers, 10
*Topic **rank**
  statshelpers, 10
*Topic **summary**
  statshelpers, 10

a.findcorrelations (statshelpers), 10
a.iqr (statshelpers), 10
a.printextremes (statshelpers), 10
a.proportion.test (statshelpers), 10
a.qr (statshelpers), 10
a.resetplotparams, 5, 7, 9
a.resetplotparams (plothelpers), 9

boxplot.stats, 4
cor, 6, 10
density, 4
dev.copy2eps, 9
dev.cur, 9
eqc (misc helpers), 2
equal.count, 2
grep, 2
grep.b (misc helpers), 2
grep.s (misc helpers), 2
ifelse, 2
jpeg, 9
Lattice, 9
lm, 7, 9
lqs, 7
mad, 4
misc helpers, 2
or.else (misc helpers), 2
order, 2
orderavg (misc helpers), 2
panel.bwplot, 3–5
panel.bwstrip, 3
panel.lminterval (panel.xy), 6
panel.loess, 7
panel.superpose, 7
panel.xy, 5, 6
panel.xyplot, 6, 7
pdf, 9
plot, 6
plot.lm, 9
plotf, 5, 7
plotfit (plothelpers), 9
plothelpers, 9
png, 9
postscript, 9
prepanel.0 (plothelpers), 9
prepanel.lmline, 9
printn (misc helpers), 2
quantile, 10
rank, 10
statshelpers, 10
summary, 10

table, 2
tableNA (misc helpers), 2
traceback, 2
tracebck (misc helpers), 2
trellis.par.get, 7
trellis.par.set, 9