Package 'Durga'

January 20, 2025

Type Package

Title Effect Size Estimation and Visualisation

Version 2.0

Date 2023-12-18

Description An easy-to-use yet powerful system for plotting grouped data effect sizes. Various types of effect size can be estimated, then plotted together with a representation of the original data. Select from many possible data representations (box plots, violin plots, raw data points etc.), and combine as desired. 'Durga' plots are implemented in base R, so are compatible with base R methods for combining plots, such as 'layout()'. See Khan & McLean (2023) <doi:10.1101/2023.02.06.526960>.

License MIT + file LICENSE

URL https://github.com/KhanKawsar/EstimationPlot

Encoding UTF-8

LazyData true

Imports vipor, boot, RColorBrewer, methods

Suggests covr, knitr, rmarkdown, tibble, data.table, testthat (>= 3.0.0)

Config/testthat/edition 3

RoxygenNote 7.2.3

VignetteBuilder knitr

BuildVignettes true

Depends R (>= 2.10)

NeedsCompilation no

Author Kawsar Khan [aut] (<https://orcid.org/0000-0003-1795-1315>), Jim McLean [aut, cre] (<https://orcid.org/0000-0001-6229-7063>)

Maintainer Jim McLean <jim_mclean@optusnet.com.au>

Repository CRAN

Date/Publication 2023-12-18 01:10:02 UTC

Contents

`damselfly` *Damselfly data*

Description

Measurements of the body size and mass for adult and juvenile males of *Xanthagrion erythroneurum* damselflies. In this species, juvenile males are coloured yellow and change to red upon sexual maturity.

Usage

```
damselfly
```

Format

A data frame with 77 observations and 3 variables.

`length` Measured body length (mm) of damselflies

`mass` Measured body mass (mg) of damselflies

`maturity` Male age groups; `adult` or `juvenile` used for body size and weight measurements

Source

Khan, M. K., & Herberstein, M. E. (2021). Male-male interactions select for conspicuous male coloration in damselflies. Animal Behaviour, 176, 157-166.

See Also

petunia, insulin.wide, insulin

DurgaBrackets *Annotate a* DurgaPlot *with confidence brackets*

Description

Brackets are added to a DurgaPlot that already exists. That means you must ensure there is sufficient space for the brackets above the plot. To do this, either specify ylim to DurgaPlot, or create a large top margin (par(mar = c(...))) and the turn off the plot frame (DurgaPlot(..., frame.plot = FALSE)). In either case, experiment with the values until the result is visually pleasing. The annotation can be drawn into the margin as it will not be cropped.

Usage

```
DurgaBrackets(
  plot.stats,
  contrasts,
  labels = "level CI",
  br.lwd = NULL,
  br.col = NULL,
  br.lty = 1,
  lb.col = NULL,
  lb.font = NULL,
  lb.cex = 1,
  snap.to = 1,
  shorten = 1.5,
  tip.length = 2,
  data.gap = 2.5,
  vertical.gap = 1.3,
  text.pad = 1.5,
  round.fn = function(x) signif(x, 2),
  ...
)
```

Arguments

plot.stats	Object returned by the call to DurgaPlot
contrasts	Set of contrasts (i.e. group comparisons) to be displayed as brackets. Defaults to contrasts passed to DurgaDiff. Can be specified as a character string ("group 1 - group 2") or a list of DurgaDiff objects. The bracket label always displays the effect size for right-hand-group - left-hand-group, regardless of the order that groups are specified in contrasts, i.e. contrasts = "G1 - G2" will appear the same as contrasts = "G2 - G1".
labels	Text to display above each bracket. May be NULL, otherwise one of: "diff" (displayed text is "<difference in means>"), "CI" ("[<lower>, <upper>]"), "level CI" ("<level>% CI [<lower>, <upper>]") or "diff CI" ("<difference in means> [<lower>, <upper>]"); a vector of texts to display for each element

of diffs, or a function called with one argument; a DurgaGroupDiff object, which should return the label to be displayed.

br.col, br.lwd, br.lty
: Graphical parameters (colour, line weight and style) that control the bracket appearance - passed to segments. May be a single value or a vector with one value per bracket. Refer to Details for default values.

lb.col, lb.cex, lb.font
: Graphical parameters (colour, scale and font) that control the label appearance - passed to text. May be a single value or a vector with one value per bracket. Refer to Details for default values.

snap.to
: Snaps the base of the lowest brackets onto horizontal grid lines separated by snap.to mm. Used to improve aesthetics of vertical alignment.

shorten
: Amount (mm) to shrink brackets at each end

tip.length
: Length of bracket tips (mm). May be a vector with length 2; length of tip at groups 1 and 2 respectively

data.gap
: Vertical distance (mm) between top-most data point and bottom of bracket

vertical.gap
: Vertical distance (mm) between overlapping brackets

text.pad
: Gap (mm) between bracket and text

round.fn
: By default, numbers displayed as text are printed to 2 significant figures. To change this behaviour, set round.fn to a function with one argument that converts its argument to the value to be displayed.

...
: Additional arguments passed to text

Details

Default values for br.lwd, br.col, lb.col and lb.font depend on the confidence intervals (CI) being plotted. If the CI covers 0, brackets and text are grey. If the CI does not cover 0, text is dark grey and bold, and brackets are dark grey with a line width of 2.

Value

No return value. DurgaBrackets is called for its side effect of adding confidence brackets to the current plot.

See Also

DurgaPlot

Examples

```
d <- DurgaDiff(petunia, 1, 2)
# Don't draw frame because brackets will appear in the upper margin
p <- DurgaPlot(d, ef.size = FALSE, frame.plot = FALSE)
# Add the brackets to the plot
DurgaBrackets(p, lb.cex = 0.8)

# Only draw brackets that do not include zero
```

```
p <- DurgaPlot(d, ef.size = FALSE, frame.plot = FALSE)
diffs <- Filter(function(pwes) (pwes$bca[4] > 0 || pwes$bca[5] < 0), p$es$group.differences)
DurgaBrackets(p, contrasts = diffs)
```

DurgaDiff *Estimate group mean differences*

Description

Estimates differences between groups in preparation for plotting by DurgaPlot.

Usage

```
DurgaDiff(x, ...)

## Default S3 method:
DurgaDiff(
  x,
  data.col,
  group.col,
  id.col,
  groups,
  contrasts = "*",
  effect.type = "mean",
  R = 1000,
  boot.params = list(),
  ci.conf = 0.95,
  boot.ci.params = list(),
  na.rm = FALSE,
  ...
)
```

Arguments

x	A data frame (or similar) containing values to be compared, or a formula (see DurgaDiff.formula).
...	Ignored
data.col	Name (character) or index (numeric) of the column within x containing the measurement data.
group.col	Name or index of the column within x containing the values to group by. May be a vector of column names/indices, in which case values from each column are concatenated to define groups.
id.col	Specify for paired data/repeated measures/with-subject comparisons only. Name or index of ID column for repeated measures/paired data. Observations for the same individual must have the same ID. For non-paired data, do not specify an id.col, (or use id.col = NA).

groups	Vector of group names. Defaults to all groups in x in *natural* order. If groups is a named vector, the names are used as group labels for plotting or printing. If data.col and group.col are not specified, x is assumed be to in *wide format*, and groups must be a list of column names identifying the group/treatment data (see example).
contrasts	Specify the pairs of groups to be compared. By default, all pairwise differences are generated. May be a single string, a vector of strings, or a matrix. Specify NULL to avoid calculating any contrasts. See Details for more information.
effect.type	Type of group difference to be estimated. Values cannot be abbreviated. See Details for further information.
R	The number of bootstrap replicates. R should be larger than your sample size, so the default value of 1000 may need to be increased for large sample sizes. If R <= nrow(x), an error such as "Error in bca.ci... estimated adjustment 'a' is NA" will be thrown. Additionally, warnings such as "In norm.inter(t, adj.alpha) : extreme order statistics used as endpoints" may be avoided by increasing R. Specify R = NA if you do not wish to calculate any CIs, either for group means for for effect sizes. This may be useful if Durga is only being used for plotting large data sets.
boot.params	Optional list of additional names parameters to pass to the boot function.
ci.conf	Numeric confidence level of the required confidence interval, e.g. ci.conf = 0.95 specifies that 95\ be calculated. Applies to both CI of effect sizes and CI of group means.
boot.ci.params	Optional list of additional names parameters to pass to the boot.ci function.
na.rm	a logical evaluating to TRUE or FALSE indicating whether NA values should be stripped before the computation proceeds. If TRUE for "paired" data (i.e. id.col is specified), all rows (observations) for IDs with missing data are stripped.

Details

Data format:

x may be a formula; see DurgaDiff.formula.

If x is a data.frame (or similar) it may be in either *long* or *wide* format. In long format, one column (data.col) contains the measurement or value to be compared, and another column (group.col) contains the group identity. Repeated measures/paired data/within-subject comparisons in long format require a subject identity column (id.col).

Wide format contains different measurements in different columns of the same row, and is well-suited for repeated measures/paired/within-subject comparison data. To pass wide format data, do not specify the arguments data.col or group.col. Instead, you must explicitly specify the groups to be compared in the groups argument. Each group must be the name of a column in x. For paired data, you may specify id.col, although it is not required, as wide format data is assumed to be paired. The id.col can be a column that already exists and uniquely identifies each specimen, or it can be the name of a column to be created, in which case the specimen ID will be a generated integer sequence. Unpaired data can be in wide format, but it is necessary to inform Durga by passing id.col = NULL. Wide format data will be internally converted to long format, then processing continues as for long format input.

Contrasts:

The pairs of groups to be compared are defined by the parameter `contrasts`. An asterisk (`"*"`, the default) creates contrasts for all possible pairs of groups. A single string has a format such as `"group1 - group2, group3 - group4"`. A single string such as `".-control"` compares all groups against the `"control"` group, i.e. the `"."` expands to all groups except the named group. A vector of strings looks like `c("group1 - group2", "group3 - group4")`. If a matrix is specified, it must have a column for each contrast, with the first group in row 1 and the second in row 2.

Effect types:

The `effect.type` parameter determines the effect size measure to be calculated. Our terminology generally follows Lakens (2013), with *d* meaning a biased estimate and *g* meaning a bias-corrected estimate. Some writers reverse this usage or use alternative terminology. Cumming (2012) recommends always using a bias-corrected estimate (although bias correction is unnecessary for large sample sizes). Durga applies Hedges' exact method for bias correction.

The effect type we call $Cohen's\ d$ for unpaired data is called $Cohen's\ d_s^*$ by Delacre et al. (2021). For paired data, our $Cohen's\ d$ is identical to $Cohen's\ d$ for unpaired data (Delacre et al. 2021); it is called d_{av} by Cumming (2012; equation 11.10). For further details, refer to Khan and McLean (2023).

The set of possible values for the `effect.type` argument, and their meanings, is described below.

Unpaired effect types:

Code	Label	Effect type
`mean`	$Mean\ difference$	Unstandardised difference of group means
`cohens d`	$Cohen's\ d$	Difference in means standardised by non-pooled average SD (Delacre et al. 2021
`hedges g`	$Hedges'\ g$	Bias-corrected $Cohen's\ d$ (Delacre et al. 2021)
`cohens d_s`	$Cohen's\ d_s$	Difference in means standardised by the pooled standard deviation (Lakens 2013,
`hedges g_s`	$Hedges'\ g_s$	Bias-corrected $Cohen's\ d_s$ (Lakens 2013, equation 4)
`glass delta_pre`	$Glass's\ \Delta_{pre}$	Difference in means standardised by the standard deviation of the pre-measureme
`glass delta_post`	$Glass's\ \Delta_{post}$	Difference in means standardised by the standard deviation of the post-measurem

Paired effect types:

Code	Label	Effect type
`mean`	$Mean\ difference$	Unstandardised mean of group differences
`cohens d`	$Cohen's\ d$	Similar to $Cohen's\ d_{av}$ except that the normaliser is non-pooled average SD rather thar
`hedges g`	$Hedges'\ g$	Bias-corrected $Cohen's\ d$
`cohens d_z`	$Cohen's\ d_z$	Mean of differences, standardised by the standard deviation of the differences, (Lakens
`hedges g_z`	$Hedges'\ g_z$	Bias-corrected $Cohen's\ d_z$
`cohens d_av`	$Cohen's\ d_{av}$	Difference in means standardised by the average standard deviation of the groups (Lake
`hedges g_av`	$Hedges'\ g_{av}$	Bias-corrected $Cohen's\ d_{av}$

As a simple rule of thumb, if you want a standardised effect type and you don't know which one to use, use `"hedges g"` for either paired or unpaired data, as it is recommended by Delacre et al., (2021) for unpaired data and cumming (2012) for paired data.

Additional effect types can be applied by passing a function for `effect.type`. The function must accept two parameters and return a single numeric value, the effect size. Each parameter is a vector of values from one of the two groups to be compared (group 2 and group 1).

Confidence intervals:

Confidence intervals for the estimate are determined using bootstrap resampling, using the adjusted bootstrap percentile (BCa) method (see `boot` and `boot.ci`). Additional arguments can be passed to the `boot` (`boot.ci`) by passing a named list of values as the argument `boot.params` (`boot.ci.params`).

Value

A `DurgaDiff` object, which is a list containing:

`group.statistics`
: Matrix with a row for each group, columns are: `mean`, `median`, `sd` (standard deviation), `se` (standard error of the mean), `CI.lower` and `CI.upper` (lower and upper bootstrapped confidence intervals of the mean, confidence level as set by the `ci.conf` parameter) and `n` (group sample size). If there are fewer than 3 distinct values in the group, or if `R` is `NA`, the confidence interval will not be calculated and `CI.lower` and `CI.upper` will be `NA`.

`group.differences`
: List of `DurgaGroupDiff` objects, which are `boot` objects with added confidence interval information. See `boot` and `boot.ci`. This element will be missing if `contrasts` is empty or `NULL`

`groups`
: Vector of group names

`group.names`
: Labels used to identify groups

`effect.type`
: Value of `effect.type` parameter

`effect.name`
: Name of the effect type; may include formatting such as subscripts

`effect.name.print`
: Text-only version of `effect.name` for printing; subscripts are indicated by `"_"`

`data.col`
: Value of `data.col` parameter; may be an index or a name

`data.col.name`
: Name of the `data.col` column

`group.col`
: Value of `group.col` parameter; may be an index or a name

`group.col.name`
: Name of the `group.col` column

`id.col`
: Value of `id.col` parameter. May be `NULL`

`paired.data`
: `TRUE` if paired differences were estimated

`data`
: The input data frame (`x`), or the reshaped (long format) data frame if the input data set was in wide format

`call`
: How this function was called

A `DurgaGroupDiff` object is a `boot` object (as returned by `boot`) with added `bootci` components (as returned by `boot.ci`) and components identifying the groups used to estimate the difference. Particularly relevant members are:

`t0`
: The observed value of the statistic

`bca[4]`	The lower endpoint of the confidence interval
`bca[5]`	The upper endpoint of the confidence interval
`groups`	The difference is estimated on `groups[1]` - `groups[2]`

References

- Cumming, G. (2012). Understanding the new statistics : effect sizes, confidence intervals, and meta-analysis (1st ed.). New York: Routledge.
- Delacre, M., Lakens, D., Ley, C., Liu, L., & Leys, C. (2021). Why Hedges' g* based on the non-pooled standard deviation should be reported with Welch's t-test. doi:10.31234/osf.io/tu6mp
- Khan, M. K., & McLean, D. J. (2023). Durga: An R package for effect size estimation and visualisation. bioRxiv, 2023.2002.2006.526960. doi:10.1101/2023.02.06.526960
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. Frontiers in Psychology, 4. doi:10.3389/fpsyg.2013.00863

See Also

`DurgaDiff.formula`, `boot`, `boot.ci`, `DurgaPlot`

Examples

```
d <- DurgaDiff(insulin, "sugar", "treatment", "id")
print(d)

# Change group order and displayed group labels, reverse the
# direction of one of the contrasts from the default
d <- DurgaDiff(petunia, 1, 2,
               groups = c("Self-fertilised" = "self_fertilised",
                          "Intercrossed" = "inter_cross",
                          "Westerham-crossed" = "westerham_cross"),
               contrasts = c("Westerham-crossed - Self-fertilised",
                             "Westerham-crossed - Intercrossed",
                             "Intercrossed - Self-fertilised"))

# Wide format data
d <- DurgaDiff(insulin.wide, groups = c("sugar.before", "sugar.after"))
```

`DurgaDiff.formula` *Formula interface for estimating group mean differences*

Description

Estimates differences between groups in preparation for plotting by `DurgaPlot`. The formula interface allows the value and group columns to be specified in a formula, which means, for example, that transformation functions can be applied to columns.

Usage

```
## S3 method for class 'formula'
DurgaDiff(x, data = NULL, id.col, ...)
```

Arguments

x	a formula, such as y ~ grp, where y is a numeric vector of data values or measurements to be split into groups according to the grouping variable grp, which is typically a categorical value. Multiple group columns can be separated by +, in which case Durga treats each unique combination of group variables as a distinct group.
data	a data.frame (or list) from which the variables in formula should be taken.
id.col	Specify for paired data/repeated measures/with-subject comparisons only. Name or index of ID column for repeated measures/paired data. Observations for the same individual must have the same ID. For non-paired data, do not specify an id.col, (or use id.col = NA).
...	Arguments passed on to DurgaDiff.default groups Vector of group names. Defaults to all groups in x in *natural* order. If groups is a named vector, the names are used as group labels for plotting or printing. If data.col and group.col are not specified, x is assumed be to in *wide format*, and groups must be a list of column names identifying the group/treatment data (see example). contrasts Specify the pairs of groups to be compared. By default, all pairwise differences are generated. May be a single string, a vector of strings, or a matrix. Specify NULL to avoid calculating any contrasts. See Details for more information. effect.type Type of group difference to be estimated. Values cannot be abbreviated. See Details for further information. R The number of bootstrap replicates. R should be larger than your sample size, so the default value of 1000 may need to be increased for large sample sizes. If R <= nrow(x), an error such as "Error in bca.ci... estimated adjustment 'a' is NA" will be thrown. Additionally, warnings such as "In norm.inter(t, adj.alpha) : extreme order statistics used as endpoints" may be avoided by increasing R. Specify R = NA if you do not wish to calculate any CIs, either for group means for for effect sizes. This may be useful if Durga is only being used for plotting large data sets. boot.params Optional list of additional names parameters to pass to the boot function. ci.conf Numeric confidence level of the required confidence interval, e.g. ci.conf = 0.95 specifies that 95\ be calculated. Applies to both CI of effect sizes and CI of group means. boot.ci.params Optional list of additional names parameters to pass to the boot.ci function. na.rm a logical evaluating to TRUE or FALSE indicating whether NA values should be stripped before the computation proceeds. If TRUE for "paired" data (i.e. id.col is specified), all rows (observations) for IDs with missing data are stripped.

Details

Applies the formula, x, and a data set, data, to construct a data frame that is then passed, with all remaining arguments, to the function DurgaDiff.default.

Value

A DurgaDiff object, which is a list containing:

group.statistics
: Matrix with a row for each group, columns are: mean, median, sd (standard deviation), se (standard error of the mean), CI.lower and CI.upper (lower and upper bootstrapped confidence intervals of the mean, confidence level as set by the ci.conf parameter) and n (group sample size). If there are fewer than 3 distinct values in the group, or if R is NA, the confidence interval will not be calculated and CI.lower and CI.upper will be NA.

group.differences
: List of DurgaGroupDiff objects, which are boot objects with added confidence interval information. See boot and boot.ci. This element will be missing if contrasts is empty or NULL

groups
: Vector of group names

group.names
: Labels used to identify groups

effect.type
: Value of effect.type parameter

effect.name
: Name of the effect type; may include formatting such as subscripts

effect.name.print
: Text-only version of effect.name for printing; subscripts are indicated by "_"

data.col
: Value of data.col parameter; may be an index or a name

data.col.name
: Name of the data.col column

group.col
: Value of group.col parameter; may be an index or a name

group.col.name
: Name of the group.col column

id.col
: Value of id.col parameter. May be NULL

paired.data
: TRUE if paired differences were estimated

data
: The input data frame (x), or the reshaped (long format) data frame if the input data set was in wide format

call
: How this function was called

A DurgaGroupDiff object is a boot object (as returned by boot) with added bootci components (as returned by boot.ci) and components identifying the groups used to estimate the difference. Particularly relevant members are:

t0
: The observed value of the statistic

bca[4]
: The lower endpoint of the confidence interval

bca[5]
: The upper endpoint of the confidence interval

groups
: The difference is estimated on groups[1] - groups[2]

References

- Cumming, G. (2012). Understanding the new statistics : effect sizes, confidence intervals, and meta-analysis (1st ed.). New York: Routledge.
- Delacre, M., Lakens, D., Ley, C., Liu, L., & Leys, C. (2021). Why Hedges' g* based on the non-pooled standard deviation should be reported with Welch's t-test. doi:10.31234/osf.io/tu6mp
- Khan, M. K., & McLean, D. J. (2023). Durga: An R package for effect size estimation and visualisation. bioRxiv, 2023.2002.2006.526960. doi:10.1101/2023.02.06.526960
- Lakens, D. (2013). Calculating and reporting effect sizes to facilitate cumulative science: a practical primer for t-tests and ANOVAs. Frontiers in Psychology, 4. doi:10.3389/fpsyg.2013.00863

See Also

DurgaDiff.default, boot, boot.ci, DurgaPlot

Examples

```
d <- DurgaDiff(log(sugar) ~ treatment, insulin, id.col = "id")
print(d)
```

DurgaPlot *Group and effect size plotting in base R.*

Description

Plot grouped data and effect size in base R, with control over a large range of possible display formats and options. To plot your data, first calculate group differences by calling DurgaDiff, then pass the result to DurgaPlot. Because there are so many parameters to this function, they are prefixed according to the component they affect. Hence, for example, all parameters that affect box plots are prefixed with box.

Usage

```
DurgaPlot(
  es,
  contrasts,
  group.dx = 0,
  group.colour = "Set2",
  points = TRUE,
  points.method = c("quasirandom", "pseudorandom", "smiley", "maxout", "frowney",
    "minout", "tukey", "tukeyDense", "jitter", "overplot"),
  points.spread = ifelse(points.method == "jitter", 0.1, 0.3),
  points.dx = group.dx,
  points.adjust = 1,
  points.params = list(),
```

```
violin = isFALSE(box) && isFALSE(bar),
violin.shape = c("left-half", "right-half", "full"),
violin.fill = TRUE,
violin.params = list(),
violin.adj = 1.5,
violin.width = 0.35,
violin.trunc = TRUE,
violin.dx = group.dx,
box = FALSE,
box.fill = TRUE,
box.outline = TRUE,
box.notch = FALSE,
box.params = list(boxwex = 0.8, staplewex = 0.5, outwex = 0.5),
box.dx = group.dx,
bar = FALSE,
bar.fill = TRUE,
bar.width = 0.8,
bar.dx = group.dx,
ef.size = TRUE,
ef.size.position = c("right", "below"),
ef.size.violin = TRUE,
ef.size.violin.fill = TRUE,
ef.size.violin.shape = c("right-half", "left-half", "full"),
ef.size.violin.trunc = TRUE,
ef.size.pch = 17,
ef.size.lty = 1,
ef.size.lwd = 2,
ef.size.ticks = NULL,
ef.size.label = es$effect.name,
ef.size.dx = 0,
ef.size.adj.margin = TRUE,
ef.size.top.pad = 2.5,
ef.size.height = 0.35,
ef.size.mean.line.dx = group.dx,
ef.size.line.col = "grey50",
ef.size.line.lty = ifelse(ef.size.position == "below", 3, 1),
ef.size.line.lwd = 1,
ef.size.params = list(),
paired = es$paired.data,
paired.lty = 1,
paired.lwd = 1,
central.tendency = isFALSE(box) && isFALSE(bar),
central.tendency.type = c("mean", "median"),
central.tendency.symbol = c("point", "segment"),
central.tendency.width = violin.width,
central.tendency.params = list(),
central.tendency.dx = group.dx,
error.bars = !isFALSE(central.tendency) || !isFALSE(bar),
```

```
  error.bars.type = c("CI", "SD", "SE"),
  error.bars.lty = 1,
  error.bars.lwd = 3,
  error.bars.cross.width = 0,
  x.axis = TRUE,
  x.axis.dx = group.dx,
  xlab = "",
  left.ylab = es$data.col.name,
  left.las = graphics::par("las"),
  add = FALSE,
  xlim,
  ylim,
  ...
)
```

Arguments

es	Data returned from a call to DurgaDiff
contrasts	Set of contrasts (i.e. group comparisons) to be plotted. Defaults to contrasts passed to DurgaDiff, otherwise ". - group1" (where group1 is the first group). See Details for more information.
group.dx	Used to shift group centres horizontally. E.g., group.dx = c(0.1, -0.1) will group into pairs. Individual components can be shifted independently using the appropriate *.dx parameters.
group.colour	Colours to use for each group. Either an RColorBrewer palette name or a vector of colours.
points	If not FALSE, points are plotted. If TRUE, points are displayed with a default colour (which is the group colour with 40% transparency). You may specify a vector of colours; if length 1, all points are drawn with the specified colour. If length is less than the number of data points, points in each group are drawn with the appropriate colour (extra colours are ignored). Otherwise, points should be a vector of colours with a value for each data point.
points.method	Method used to avoid overplotting points. Use "overplot" to overplot points and "jitter" to add random noise to each x-value. See offsetX for remaining methods.
points.spread	Numeric value used to adjust the points scatter method points horizontally (ignored if points.method = "overplot").
points.dx	Horizontal shift to be applied to points in each group.
points.adjust	Adjust the bandwidth used to calculate kernel density when drawing points. Smaller values mean a tighter fit.
points.params	List of named parameters to pass on to points, e.g. DurgaPlot(es, points = "black", points.params = list(pch = 21, bg = as.numeric(factor(data$Sex)) + 1)).
violin	If not FALSE, violin plots are drawn. Violins are simply probability density plots, with density on the x-axis and value on the y-axis. If TRUE, violins are drawn in default colours. Otherwise specifies the colour of the violin borders.

`violin.shape` Desired violin shape - left-half only (`"left"`), right-half only (`"right"`), or a full violin (`"full"`).

`violin.fill` Colour used to fill violins. Specify `FALSE` or `NA` to leave violins unfilled.

`violin.params` Additional graphical parameters applied to drawing violins. May include `density`, `angle`, `lty`, `lwd`, `lend` etc. Values are passed on to `polygon`; see its help page for details.

`violin.adj` Value used to control violin plot smoothness by adjusting the kernel density bandwidth. Higher values produce a smoother plot. Passed unchanged as the `adjust` argument to the `density` function.

`violin.width` Width of maximum violin horizontal extents, as a proportion of the distance between groups.

`violin.trunc` Numeric value that specifies what vertical proportion of the violin is truncated.

`violin.dx` Horizontal shift to be applied to each violin.

`box` If not `FALSE`, draw a box-and-whisker plot of the grouped values. Value may be a colour, in which case the box borders are plotted with the colour(s). See `boxplot`.

`box.fill` Colour used to fill the bodies of the box-and-whisker plot. If FALSE or NA, bodies are not filled.

`box.outline` If FALSE, don't draw outliers with the box plot.

`box.notch` If TRUE, draws notches in the sides of the boxes. See `boxplot.stats` for the calculations used.

`box.params` List with additional graphical parameters to control the box plot appearance. Many aspects of the plots can be controlled, e.g. `box.params = list(outpch = NA)` prevents outlier points from being drawn. See `bxp` graphical parameters for a complete list.

`box.dx` Horizontal shift to be applied to each box.

`bar` If not `FALSE`, draw a bar plot of the group means or medians, according to `central.tendency`. May be `TRUE` or a colour.

`bar.fill` Colour used to fill bars.

`bar.width` Width of bars.

`bar.dx` Horizontal shift to be applied to each bar.

`ef.size` If not `FALSE`, effect sizes are plotted. May be `TRUE` or a colour.

`ef.size.position`
Effect sizes are plotted to the right of the main plot if there is only one effect size to plot and `ef.size.position != "below"`. If the effect size is drawn to the right, you will need to increase the size of the right margin before plotting (see `par(mar = ...)`).

`ef.size.violin` If not `FALSE`, boostrapped effect size estimates are shown as a violin plot. May be a colour that is used for the violin border and fill (unless `ef.size.violin.fill` is specified).

`ef.size.violin.fill`
Colour used to fill effect size violins. Default is a transparent version of `ef.size.violin`.

`ef.size.violin.shape`
: Shape of the effect size violin. One of `"right-half"`, `"left-half"` or `"full"`.

`ef.size.violin.trunc`
: If `TRUE`, effect size violin is truncated vertically so that it just covers the estimated effect size.

`ef.size.pch`
: Symbol to represent mean effect size.

`ef.size.lty`
: Line style for the effect size error bar.

`ef.size.lwd`
: Line weight for the effect size error bar.

`ef.size.ticks`
: Optional locations and labels for ticks on the effect size y-axis. E.g. to interpret effect size using Cohen's default values, specify `ef.size.ticks = c("Large negative effect" = -0.8, "Medium negative effect" = -0.5, "Small negative effect" = -0.2, "No effect" = 0, "Small positive effect" = 0.2, "Medium positive effect" = 0.5, "Large positive effect" = 0.8)`

`ef.size.label`
: Label to display on y-axis for effect size.

`ef.size.dx`
: Horizontal shift to be applied to each contrast/effect size. Unlike other `.dx` parameters, `ef.size.dx` is indexed by contrast rather than group. When effect size is below the plot, the `group.dx` for the group above the effect size is also added.

`ef.size.adj.margin`
: If TRUE (the default), the right margin (if ES is right) or bottom margin (if ES is below) is automatically increased to make room to display the effect size or axis annotations. The margins are restored before control returns from `DurgaPlot`.

`ef.size.top.pad`
: Gap (in units of default character height scaled by `cex`) between the bottom of the main plot region and the top of the effect size plot region. Only applies when effect size is positioned below.

`ef.size.height`
: Height of the effect size plot region as a proportion of the main plot region. Only applies when effect size is positioned below.

`ef.size.mean.line.dx`
: Horizontal shift to be applied to the start (i.e. left end) of the group mean horizontal lines when effect size is on the right.

`ef.size.line.col`
: Colour of horizontal effect-size lines that depict group means if effect size is on the right, otherwise colour of line at y = 0.

`ef.size.line.lty`
: Line style of horizontal effect-size lines.

`ef.size.line.lwd`
: Line width of horizontal effect-size lines.

`ef.size.params`
: List of graphical parameters to apply when drawing effect sizes. These parameters are passed to `par` before drawing the effect size. E.g. `ef.size.params = list(mgp = c(3.5, 1, 0))` will shift the effect size y-axis label to the left or right (for `ef.size.position` `"below"` or `"right"` respectively). `ef.size.params = list(las = 1)` will rotate the effect size axis labels without rotating the main axis labels.

`paired` If not `FALSE` and the data a paired, lines are drawn joining the individual data points. May be `TRUE` or a colour. Defaults to `TRUE` if the data are paired (i.e. the `id.col` argument was specified in the call to `DurgaDiff`).

`paired.lty` Line style for pair lines.

`paired.lwd` Line width for pair lines.

`central.tendency`
If not `FALSE`, a visual indicator of central tendency is drawn. May be `TRUE` or a colour, in which case it is used for mean/median and error bars.

`central.tendency.type`
Should the indicated measure of central tendency be `"mean"` or `"median"`?

`central.tendency.symbol`
Should central tendency be shown as a point or a horizontal line segment?

`central.tendency.width`
Width of the central tendency line segment.

`central.tendency.params`
Additional arguments to be passed to `points` (if `central.tendency.symbol == "point"`) or `segments` (if `central.tendency.symbol == "segment"`).

`central.tendency.dx`
Horizontal shift to apply to central tendency indicator and error bars.

`error.bars` Should error bars be displayed? May be the colour to be used for error bars.

`error.bars.type`
Should error bars depict 95% confidence intervals of the mean (`"CI"`), standard deviation (`"SD"`) or standard error (`"SE"`)?

`error.bars.lty` Line style for error bars.

`error.bars.lwd` Line width for error bars.

`error.bars.cross.width`
Length (in inches) of the horizontal crossbars at the ends of the error bars. If 0, no crossbar is drawn.

`x.axis` if TRUE, display the x-axis ticks and labels. The effect of `x.axis = FALSE` is similar to setting `xaxt = "n"` in a base R plot.

`x.axis.dx` Horizontal shifts to be applied to each x-axis tick and label.

`xlab` X axis label.

`left.ylab` Left-hand y-axis label.

`left.las` Orientation of axis labels on left-hand y-axis label (0 = parallel to axis, 1 = horizontal).

`add` If TRUE, the effect size plot is added to the current plot. If FALSE, a new plot is created.

`xlim, ylim` If specified, overrides the default plot extents.

`...` Additional arguments are passed on to the `plot` function.

Details

Group data may be visualised in multiple ways: `points`, `violin`, `box` and `bar`. Each visualisation type is controlled by a set of parameters with the same prefix. To display a type, for example box plots, specify `box = TRUE`. Rather than `box = TRUE`, you may specify a colour (e.g. `box \ "blue"`), which is used as the border/outline for the boxes. You may also specify a vector of colours, one for each group. For `points`, you may specify a colour for each individual point. When colours are not specified, they default to the group colours (`group.colour`).

Group data annotations are controlled with parameters `central.tendency` and `error.bars`. `central.tendency` visually represents the mean or median (`central.tendency.type`) of each group, while `error.bars` are vertical bars showing the 95% CI of the mean, standard deviation or standard error of the groups (`error.bars.type`).

An effect size (for our purposes) is the difference in means between two groups. Effect size display is controlled by `ef.size`. The set of effect sizes (aka "contrasts") to be plotted is controlled by the `contrasts` parameter. If a single effect size is displayed, it may be positioned to the right of - or below - the main plot (`ef.size.position`). If more than one effect size is displayed, it must be below the main plot. If below, an effect size is drawn underneath its primary group. See `DurgaBrackets` for a way to display multiple effect sizes that would overlap if displayed as normal effect sizes.

Custom labels for individual effects can be specified as part of the `contrasts` parameter. If `contrasts` is a named vector, the names are used as contrast labels, e.g. `contrasts = c("Adult change" = "adult - control", "Juvenile change" = "juvenile - control")`. A more flexible (although more advanced) method is to assign the `label.plot` member of a `DurgaDiff` object within `x`, see Examples for usage.

The `contrasts` parameter may be a single string, a vector of strings, or a matrix. A single string has a format such as `"group1 - group2,group3 - group4"`. A single asterisk, `"*"` creates contrasts for all possible pairs of groups. A single string such as `".- control"` compares all groups against the `"control"` group, i.e. the `"."` expands to all groups except the named group. A vector of strings looks like `c("group1 - group2", "group3 - group4")`. If a matrix is specified, it must have a column for each contrast, with the first group in row 1 and the second in row 2. See also the `contrasts` parameter to `DurgaDiff`. It is an error to attempt to plot a contrast that was not estimated by `DurgaDiff`.

Value

A list (returned invisibly) with 4 elements:

`es`	Value of the `es` parameter.
`extents`	Matrix with the x-axis locations and y-axis extents of each displayed group.
`plot.differences`	A list of the displayed differences, as `DurgaGroupDiff` objects. Will be an empty list if no effect sizes are shown.
`palette`	Vector of colours used by default for each group.

References

Gardner, M. J., & Altman, D. G. (1986). Confidence intervals rather than P values: estimation rather than hypothesis testing. Br Med J (Clin Res Ed), 292(6522), 746-750. doi:10.1136/bmj.292.6522.746

Cumming, G. (2012). Understanding the new statistics : effect sizes, confidence intervals, and meta-analysis (1st edition ed.). New York: Routledge.

See Also

DurgaDiff, DurgaBrackets, DurgaTransparent, offsetX, boxplot, bxp

Examples

```
d <- DurgaDiff(petunia, "height", "group")
# Default plot
DurgaPlot(d)

# Boxplot with a single effect size plotted on the right
DurgaPlot(d, contrasts = "westerham_cross - self_fertilised",
          box = TRUE, points = "black", points.params = list(cex = 0.8))

# Use confidence brackets to show all group differences
p <- DurgaPlot(d, ef.size = FALSE, group.colour = "Set1",
          points = "black", points.method = "jitter",
          points.params = list(pch = 21), points.dx = 0.15,
          violin.dx = -0.05, violin = "black", violin.adj = 0.5,
          ylim = c(12, 75))
DurgaBrackets(p)

# Adjust group names, contrasts
d <- DurgaDiff(petunia, 1, 2,
               groups = c("self-fertilised" = "self_fertilised",
                          "intercrossed" = "inter_cross",
                          "Westerham-crossed" = "westerham_cross"),
               contrasts = c("Westerham-crossed - self-fertilised",
                             "Westerham-crossed - intercrossed",
                             "intercrossed - self-fertilised"))
# Shift the 2nd effect size horizontally (Westerham-crossed - intercrossed)
# so it doesn't overlap another
DurgaPlot(d, ef.size.dx = c(0, -2, 0))

# Custom difference labels with italics
d <- DurgaDiff(petunia, 1, 2)
d$group.differences[[3]]$label.plot <- expression(italic("sp. 2")~"-"~italic("sp. 1"))
d$group.differences[[2]]$label.plot <- expression(italic("sp. 3")~"-"~italic("sp. 1"))
DurgaPlot(d)
```

DurgaTransparent *Returns a transparent version of the specified colour(s).*

Description

Returns a transparent version of the specified colour(s).

Usage

```
DurgaTransparent(colour, transparency, relative = FALSE)
```

Arguments

colour	The R colour (or colours) to be made transparent. May be specified in any way recognised by col2rgb: a colour name, a hexadecimal string such as "#ffbc48" or a positive integer i meaning meaning palette()[i].
transparency	Transparency, from 0, meaning fully opaque, through to 1, which is completely transparent (i.e. invisible).
relative	Determines what happens if colour is already transparent. If relative is FALSE (the default), then the transparency value of colour is ignored and transparency defines the transparency of the returned colour. If TRUE, the existing transparency value is multiplied by transparency.

Value

A colour or colours that are transparent versions of colour.

See Also

col2rgb, rgb

Examples

```
transparentPink <- DurgaTransparent("red", 0.8)
transparentPink
```

---

insulin *Insulin data*

Description

Selected results of experiments performed by Banting *et al*., (1922), testing whether insulin reduces blood sugar. Insulin was administered in rabbits and blood sugar was measured within three hours. Data collated from Table 1 and Table 2 of Banting *et al*., (1922). This data set is in *long format*. insulin.wide is an equivalent data set in *wide format*.

Usage

```
insulin
```

Format

A data frame with 104 observations and 5 variables.

`sugar` Measured blood sugar level

`treatment` Blood sugar measurements treatment group; `before` or `after` administering insulin

`id` Identifier of individual being measured (not in original data set)

`experimenter_time` Initial of researchers who performed the experiment and at what date

`time` Time of blood sugar measurement; minutes after administration of insulin

Source

Banting, F. G., Best, C. H., Collip, J. B., Macleod, J. J., & Noble, E. C. (1922). The effect of pancreatic extract (insulin) on normal rabbits. American Journal of Physiology-Legacy Content, 62(1), 162-176.

See Also

insulin.wide, petunia, damselfly

`insulin.wide` *"Wide format" Insulin data*

Description

This data set contains the same information as insulin, however it is in *wide format* rather than *long format*. Refer to insulin for further details.

Usage

```
insulin.wide
```

Format

A data frame with 52 observations and 5 variables.

`sugar.before` Blood sugar measurement before administering insulin

`sugar.after` Blood sugar measurement after administering insulin

`time` Time of blood sugar measurement; minutes after administration of insulin

`experimenter` Initials of researcher who performed the experiment

`date` Date of experiment (month day)

Source

Banting, F. G., Best, C. H., Collip, J. B., Macleod, J. J., & Noble, E. C. (1922). The effect of pancreatic extract (insulin) on normal rabbits. American Journal of Physiology-Legacy Content, 62(1), 162-176.

See Also

insulin, petunia, damselfly

`petunia` *Petunia data*

Description

Charles Darwin's experimental results on petunia plants to determine the difference of plant length between self fertilised and cross-fertilised, either with the same stock (inter-cross) or with a fresh stock (westerham-cross)

Usage

```
petunia
```

Format

A data frame with 64 observations and 3 variables. Plants (self fertilised, inter-cross and westerham-cross) were measured to the tops of their stems when coming into flower. Data collated from Darwin (1877)

`height` Measured height of plants

`group` Plant fertilisation groups; `westerham_cross`, `inter_cross` or `self_fertilised`

`pot_no` Identifier of pot number in which each plant was grown

Source

Darwin, C. (1877). The effects of cross and self fertilisation in the vegetable kingdom. John Murray, Albemarle Street, London.

See Also

insulin, insulin.wide, damselfly

Index