Package 'grafify'

May 8, 2026

Type Package

Title Easy Graphs for Data Visualisation and Linear Models for ANOVA

Date 2025-08-23

Version 5.1.0

Description Easily explore data by plotting graphs with a few lines of code. Use these ggplot() wrappers to quickly draw graphs of scatter/dots with box-whiskers, violins or SD error bars, data distributions, before-after graphs, factorial ANOVA and more. Customise graphs in many ways, for example, by choosing from colour blind-friendly palettes (12 discreet, 3 continuous and 2 divergent palettes). Use the simple code for ANOVA as ordinary (lm()) or mixed-effects linear models (lmer()), including randomised-block or repeated-measures designs, and fit non-linear outcomes as a generalised additive model (gam) using mgcv(). Obtain estimated marginal means and perform post-hoc comparisons on fitted models (via emmeans()). Also includes small datasets for practising code and teaching basics before users move on to more complex designs. See vignettes for details on usage <https://grafify.shenoylab.com/>. Citation: <doi:10.5281/zenodo.5136508>.

License GPL (>= 2)

Imports car, dplyr, emmeans, Hmisc, lme4, lmerTest, magrittr, mgcv, patchwork, purrr, stats, tidyr

Depends R (>= 4.0), ggplot2 (>= 3.4.0)

Encoding UTF-8

LazyData true

Language en-GB

RoxygenNote 7.3.2

Suggests knitr, Matrix (>= 1.6-5), rlang, rmarkdown, pbkrtest, testthat (>= 3.0.0)

URL https://github.com/ashenoy-cmbi/grafify

Config/testthat/edition 3

NeedsCompilation no

Author Avinash R Shenoy [cre, aut] (ORCID: <https://orcid.org/0000-0001-6228-9303>)

Maintainer Avinash R Shenoy <a.shenoy@imperial.ac.uk>

Repository CRAN

Date/Publication 2025-08-25 12:10:02 UTC

Contents

data_1w_death . . . 3
data_2w_Festing . . . 4
data_2w_Tdeath . . . 4
data_cholesterol . . . 5
data_doubling_time . . . 6
data_t_pdiff . . . 6
data_t_pratio . . . 7
data_zooplankton . . . 7
ga_anova . . . 8
ga_model . . . 9
get_graf_colours . . . 10
graf_colours . . . 11
graf_col_palette . . . 12
graf_col_palette_default . . . 12
graf_palettes . . . 13
make_1way_data . . . 14
make_1way_rb_data . . . 15
make_2way_data . . . 16
make_2way_rb_data . . . 18
mixed_anova . . . 19
mixed_anova_slopes . . . 21
mixed_model . . . 23
mixed_model_slopes . . . 25
plot_3d_point_sd . . . 27
plot_3d_scatterbar . . . 30
plot_3d_scatterbox . . . 33
plot_3d_scatterviolin . . . 36
plot_4d_point_sd . . . 39
plot_4d_scatterbar . . . 42
plot_4d_scatterbox . . . 45
plot_4d_scatterviolin . . . 48
plot_befafter_box . . . 51
plot_befafter_colours . . . 54
plot_befafter_shapes . . . 57
plot_density . . . 60
plot_dotbar_sd . . . 62
plot_dotbox . . . 65
plot_dotviolin . . . 67
plot_gam_predict . . . 70
plot_grafify_palette . . . 71
plot_histogram . . . 72

plot_lm_predict . . . 74
plot_logscale . . . 76
plot_point_sd . . . 77
plot_qqline . . . 80
plot_qqmodel . . . 82
plot_qq_gam . . . 83
plot_scatterbar_sd . . . 84
plot_scatterbox . . . 86
plot_scatterviolin . . . 89
plot_xy_CatGroup . . . 92
plot_xy_Group . . . 95
plot_xy_NumGroup . . . 98
posthoc_Levelwise . . . 101
posthoc_Pairwise . . . 102
posthoc_Trends_Levelwise . . . 103
posthoc_Trends_Pairwise . . . 105
posthoc_Trends_vsRef . . . 106
posthoc_vsRef . . . 108
scale_colour_grafify . . . 109
scale_fill_grafify . . . 111
simple_anova . . . 113
simple_model . . . 115
table_summary . . . 116
table_x_reorder . . . 117
theme_grafify . . . 118

Index **120**

`data_1w_death` *In vitro experiments measuring percentage cell death in three genotypes of cells.*

Description

These data are from in vitro measurements of death of host cells (measured as percentage of total cells) after infection with three different strains of a pathogenic bacterium, from five independent experiments. The three strains are three levels within the fixed factor Genotype. The five independent experiments are levels within the random variable Experiment. These data can be analysed using linear mixed effects modelling. These data are from Goddard *et al*, Cell Rep, 2019 (doi:10.1016/j.celrep.2019.03.100).

Usage

```
data_1w_death
```

Format

data.frame: 15 obs. of 3 variables.

Experiment Experiment - a random factor with 5 levels "Exp_1","Exp_2"...

Genotype Genotypes - a fixed factor with 3 levels: "WT","KO_1","KO_2".

Death Numerical dependent variable indicating percentage cell death.

`data_2w_Festing` *Data from two-way ANOVA with randomised block design of treatments of strains of mice.*

Description

Data from Festing, ILAR Journal (2014) 55, 472–476 (doi:10.1093/ilar/ilu045). These data are suitable for two-way linear mixed effects modelling. The activity of GST (numerical dependent variable) was measured in 4 strains of mice (levels with the fixed factor Strain) either treated or controls (levels within the fixed factor Treatment). Once mouse each was used in two randomised blocks, which is the random factor (Block).

Usage

```
data_2w_Festing
```

Format

data.frame: 16 obs. of 4 variables:

Block A random factor with 2 levels "A" and "B".

Treatment A fixed factor with 2 levels: "Control" & "Treated"

Strain A fixed factor with 4 levels: "129Ola", "A/J", "NIH" & "BALB/C"

GST Numerical dependent variable indicating GST activity measurement

---

`data_2w_Tdeath` *In vitro measurement of percentage cell death - two-way ANOVA design with repeated measures, and randomised blocks.*

Description

These are measurements of death of infected host cells (as percentage of total cells) upon infection with two strains of bacteria, measured at two time points, in 6 independent experiments. These data repeated-measures data suitable for two-way linear mixed effects modelling with experiment and subjects as random factors.

Usage

```
data_2w_Tdeath
```

Format

data.frame: 24 obs. of 6 variables:

Experiment A random factor with 6 levels "e1", "e2"...

Time A fixed factor with 2 levels: "t100" & "t300".

Time2 A numeric column that allows plotting data on a quantitative "Time" axis. The "Time" column has "factor" type values that should be used for the ANOVA..

Genotype A fixed factor with 2 levels that we want to compare "WT" & "KO".

Subject A random factor with 12 levels: "s1", "s2"... These are cell culture wells that were measured at two time points, and indicate "subjects" that underwent repeated-measures within each of 6 experiments. Subject IDs for WT and KO are unique and clearly indicate different wells.

PI Numerical dependent variable indicating propidium iodide dye uptake as a measure of cell death. These are percentage of dead cells out of total cells plated.

---

data_cholesterol *Hierarchical data from 25 subjects either treated or not at 5 hospitals - two-way ANOVA design with repeated measures.*

Description

An example dataset on measurements of blood cholesterol levels measured in 5 subjects measured before and after receiving a Drug. Five patients each were recruited at 5 hospitals (a-e), so that there are 25 different subjects (1-25) measured twice. Data are from Micro/Immuno Stats

Usage

```
data_cholesterol
```

Format

tibble: 30 obs. of 3 variables:

Hospital Factor with 5 levels (a-e), representing different hospitals where subjects were recruited.

Subject A factor with 25 levels denoting individuals on whom measurements were made twice.

Treatment A factor with 2 levels indicating when measurements were made, i.e. before and after drug.

Cholesterol Numerical dependent variable indicating measured doubling time in min.

`data_doubling_time` *Doubling time of E.coli measured by 10 students three independent times.*

Description

An example dataset showing measurements of *E. coli* doubling times (in min) measured by 10 different students in 3 independent experiments each. Note that Experiments are just called Exp1-Exp3 even though Exp1 of any of the students are not connected in anyway - this will confuse R! Data are from Micro/Immuno Stats

Usage

```
data_doubling_time
```

Format

tibble: 30 obs. of 3 variables:

Student Factor with 10 levels, representing different students.

Experiment A factor with 3 levels representing independent experiments.

Doubling_time Numerical dependent variable indicating measured doubling time in min.

`data_t_pdiff` *Matched data from two groups where difference between them is consistent.*

Description

An example dataset for paired difference Student's *t* test. These are bodyweight (Mass) in grams of same mice left untreated or treated, which are two groups to be compared. The data are in a longtable format, and the two groups are levels within the factor "Condition". The Subject column lists ID of matched mice that were measured without and with treatment. These data are from Sanchez-Garrido *et al*, Sci Signal, 2018 (doi:10.1126/scisignal.aat6903).

Usage

```
data_t_pdiff
```

Format

data.frame: 20 obs. of 3 variables:

Subject Factor with 10 levels, denoted by capital letters, representing individuals or subjects.

Condition A fixed factor with 2 levels: "Untreated" & "Treated".

Mass Numerical dependent variable indicating body mass of mice

data_t_pratio *Matched data from two groups where ratio between them is consistent.*

Description

An example dataset for paired ratio Student's *t* test. These are Cytokine measurements by ELISA (in ng/ml) from 33 independent in vitro experiments performed on two Genotypes that we want to compare. The data are in a longtable format, and the two groups are levels within the factor "Genotype". The Experiment column lists ID of matched experiments.

Usage

```
data_t_pratio
```

Format

data.frame: 66 obs. of 3 variables:

Genotype Factor with 2 levels, representing genotypes to be compared ("WT" & "KO").

Experiment A random factor with 33 levels representing independent experiments, denoted as "Exp_1", "Exp_2"...

Cytokine Numerical dependent variable indicating cytokine measured by ELISA.

data_zooplankton *Time-series data on zooplankton in lake Menon.*

Description

A subset of data from (Lathro RC, 2000) (doi:10.6073/pasta/ec3d0186753985147d4f283252388e05) provided by the Wisconsin Department of Natural Resources

Usage

```
data_zooplankton
```

Format

tibble: 1127 obs. of 8 variables:

day Numeric integer variable.

year Numeric integer variable of years during which data were collected.

lake This data is for lake Menon; data for other others not included in this subset.

taxon Names of zooplankton taxa as factor of 8 levels.

density Numeric values of density of measurements.

density_adj Numeric values of adjusted density .

min_density Numeric values of minimum densities.

desnsity_scaled Numeric value of scaled density.

---

ga_anova *ANOVA table from a generalised additive model (*gam*)*

Description

One of two functions for fitting generalised additive models (gam) with the mgcv package. It will use the gam() function in mgcv for ANOVA designs with **up to two categorical fixed factors** (with two or more levels; Fixed_Factor), and **exactly one factor is a continuous variable** (e.g. time), which is called Smooth_Factor.

1. ga_model
2. ga_anova

Usage

```
ga_anova(
  data,
  Y_value,
  Fixed_Factor,
  Smooth_Factor,
  Random_Factor = NULL,
  Nodes = NULL,
  ...
)
```

Arguments

data	a data frame where categorical independent variables are converted to factors using as.factor() first. The function will throw errors without this.
Y_value	name of column containing quantitative (dependent) variable, provided within "quotes".
Fixed_Factor	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). Convert to factors first with as.factor.
Smooth_Factor	the continuous variable to fit smoothly with a basis function, provided within "quotes" (only 1 Smooth_Factor allowed).
Random_Factor	name(s) of random factors to be provided in "quotes" (only 1 Random_Factor allowed). Convert to factor with as.factor first.
Nodes	number of nodes (the parameter k in gam).
...	any additional variables to pass on to gam or anova

Details

A smooth function is fitted with factor-wise smooth basis function (by =). A default value for number of nodes (the argument k in gam) may work, but a specific number can be provided using the Nodes argument. The model is fit using the REML method. When two categorical fixed factors are provided, an interaction term is included for main effects and smooth basis functions.

If a Random_Factor is also provided, it is fitted using bs = "re" smooth.

Value

ANOVA table of class "anova" and "data.frame".

Examples

```
#with zooplankton data
ga_anova(data = data_zooplankton,
Y_value = "log(density_adj)",
Fixed_Factor = "taxon",
Smooth_Factor = "day")
```

---

ga_model *Fit a generalised additive model (*gam*)*

Description

One of two functions for fitting generalised additive models (gam) with the mgcv package. It will use the gam() function in mgcv for ANOVA designs with **up to two categorical fixed factors** (with two or more levels; Fixed_Factor), and **exactly one factor is a continuous variable** (e.g. time), which is called Smooth_Factor.

1. ga_model
2. ga_anova

A smooth function is fitted with factor-wise smooth basis function (by =). A default value for number of nodes (the argument k in gam) may work, but a specific number can be provided using the Nodes argument. The model is fit using the REML method. When two categorical fixed factors are provided, an interaction term is included for main effects and smooth basis functions.

Usage

```
ga_model(
  data,
  Y_value,
  Fixed_Factor,
  Smooth_Factor,
  Random_Factor = NULL,
```

```
  Nodes = "NULL",
  ...
)
```

Arguments

data	a data frame where categorical independent variables are converted to factors using as.factor() first. The function will throw errors without this.
Y_value	name of column containing quantitative (dependent) variable, provided within "quotes".
Fixed_Factor	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). Convert to factors first with as.factor.
Smooth_Factor	the continuous variable to fit smoothly with a basis function, provided within "quotes" (only 1 Smooth_Factor allowed).
Random_Factor	name(s) of random factors to be provided in "quotes" (only 1 Random_Factor allowed). Convert to factor with as.factor first.
Nodes	number of nodes (the parameter k in gam).
...	any additional variables to pass on to gam or anova

Details

If a Random_Factor is also provided, it is fitted using bs = "re" smooth.

Value

This function gives a generalised additive model object of class "gam", "lm" and "glm".

Examples

```
#fit a model with zooplankton data
z1 <- ga_model(data = data_zooplankton,
Y_value = "log(density_adj)",
Fixed_Factor = "taxon",
Smooth_Factor = "day")
```

---

get_graf_colours *Get graf internal*

Description

Function to make grafify colour scheme. Thank you Dr Simon.

Usage

```
get_graf_colours(...)
```

Arguments

...	internal

Details

To visualise grafify colours use `plot_grafify_palette`.

Value

This function returns names and hexcodes of colours in grafify as a character vector.

`graf_colours` *List of hexcodes of colours in grafify palettes*

Description

To visualise these colours use `plot_grafify_palette`. `okabe_ito`, `bright`, `contrast`, `dark`, `light`, `muted`, `pale`, `vibrant`, `yello_conti` from Paul Tol's colours (also see `khroma` package). Zesty, Pastel, Elegant from this link. Colour hexcodes for `fishy`, `kelly`, `r4`, `safe`, `OrBl_div`, `PrGn_div`, `blue_conti`, `grey_conti` taken from cols4all:c4a_gui package. All schemes are colour blind-friendly.

Usage

```
graf_colours
```

Format

An object of class `character` of length 154.

Value

This is a character vector with names and hexcodes of colours used by palette functions. It is used by `get_graf_colours` to generate palettes.

graf_col_palette *Call* grafify *palettes for scale & fill functions*

Description

graf_col_palette and graf_col_palette_default functions generate colours for grafify scale functions. graf_col_palette picks sequential colours when the number of discrete colours needed is less than that in the palette. This is the default for grafify with ColoSeq = TRUE. If the number of colours required is more than that in the discrete palette, it fills intervening colours using the colorRampPalette[grDevices] function.

Usage

```
graf_col_palette(palette = "okabe_ito", reverse = FALSE, ...)
```

Arguments

palette	internal
reverse	internal
...	additional parameters

Details

graf_col_palette_default picks the most distant colours within the palette, rather than in the sequence they are in the palette, when the number of colours required is less than that in the palette.

Colour order can be reversed in both functions.

When only one colour discreet is required, and you want to reverse the colour palette, ColSeq should be set to FALSE.

Value

This generates required number of sequential colours from the chosen grafify palette when called by scale functions of ggplot2.

graf_col_palette_default *Call* grafify *palettes for scale & fill functions*

Description

graf_col_palette and graf_col_palette_default functions generate colours for grafify scale functions. graf_col_palette picks sequential colours when the number of discrete colours needed is less than that in the palette. This is the default for grafify with ColoSeq = TRUE. If the number of colours required is more than that in the discrete palette, it fills intervening colours using the colorRampPalette[grDevices] function.

Usage

```
graf_col_palette_default(palette = "okabe_ito", reverse = FALSE, ...)
```

Arguments

palette	internal
reverse	internal
...	additional parameters

Details

graf_col_palette_default picks the most distant colours within the palette, rather than in the sequence they are in the palette, when the number of colours required is less than that in the palette.

Colour order can be reversed in both functions.

When only one colour discreet is required, and you want to reverse the colour palette, ColSeq should be set to FALSE.

Value

This generates required number of distant colours from the chosen grafify palette when called by scale functions of ggplot2.

---

graf_palettes *List of palettes available in grafify package*

Description

To visualise these colours use plot_grafify_palette.

Usage

```
graf_palettes
```

Format

An object of class list of length 18.

Value

This function returns a list of palettes in grafify with names and hexcodes of colours in those palettes. Names of palettes available are as follows:

Categorical/discreet palettes:

- okabe_ito
- bright
- contrast

- dark
- kelly
- light
- muted
- pale
- r4
- safe
- vibrant

Sequential quantitative palettes:

- grey_conti
- blue_conti
- yellow_conti

Divergent quantitative palettes:

- OrBl_div
- PrGn_div

`make_1way_data` *Make one-way or two-way independent group or randomised block design data.*

Description

The `make_1way_data`, `make_1way_rb_data`, `make_2way_data` and `make_2way_rb_data` functions generate independent or randomised block (rb) design data of one-way or two-way designs.

Usage

```
make_1way_data(Group_means, Num_obs, Residual_SD)
```

Arguments

`Group_means`	a vector with means of each level of the first fixed factor (FixFac_X1) measured within Group 1.
`Num_obs`	a single numeric value indicating the number of independent measurements, i.e. levels within the random factor Experiment.
`Residual_SD`	a single numeric value indicating residual SD in the model.

Details

Random variates from the normal distribution based on user provided mean and SD provided are generated. For independent designs, the `Residual_SD` argument is used to set expected residual SD from the linear model. Exp_SD is used to set experiment-to-experiment SD, that will be assigned to the random factor for rb designs.

Num_exp sets the number of independent measurements per group.

For one-way designs, the user provides Group_means as a vector. Number of levels are recognised based on number of means. For two-way designs, two vectors are to be provided by the user containing means of levels of a second factor. Number of means in both vectors should be the same. These functions can only handle balanced designs, i.e. same number of observations in all groups.

The output is a data frame with one or two columns denoting the fixed factor with levels that match the number of means entered. For rb data, the column for RandFac denotes levels of the blocking factor. The quantitative response variables are in the numeric Values column.

Value

This function produces a `data.frame` object containing simulated data.

Examples

```
#Basic usage with three levels within Factor_X,
#20 observations in each group, with residual SD 15

one_independent_tab <- make_1way_data(c(350, 250, 100), 15, 20)

str(one_independent_tab)
head(one_independent_tab)
```

`make_1way_rb_data` *Make one-way or two-way independent group or randomised block design data.*

Description

The `make_1way_data`, `make_1way_rb_data`, `make_2way_data` and `make_2way_rb_data` functions generate independent or randomised block (rb) design data of one-way or two-way designs.

Usage

```
make_1way_rb_data(Group_means, Num_exp, Exp_SD, Residual_SD)
```

Arguments

Group_means	a vector with means of each level of the first fixed factor (FixFac_X1) measured within Group 1.
Num_exp	a single numeric value. indicating the number of independent measurements, i.e. levels within the random factor RandFac.
Exp_SD	a single numeric value indicating the standard deviation (SD) between experiments, i.e. within RandFac.
Residual_SD	a single numeric value indicating residual SD in the model.

Details

Random variates from the normal distribution based on user provided mean and SD provided are generated. For independent designs, the Residual_SD argument is used to set expected residual SD from the linear model. Exp_SD is used to set experiment-to-experiment SD, that will be assigned to the random factor for rb designs.

Num_exp sets the number of independent measurements per group.

For one-way designs, the user provides Group_means as a vector. Number of levels are recognised based on number of means. For two-way designs, two vectors are to be provided by the user containing means of levels of a second factor. Number of means in both vectors should be the same. These functions can only handle balanced designs, i.e. same number of observations in all groups.

The output is a data frame with one or two columns denoting the fixed factor with levels that match the number of means entered. For rb data, the column for RandFac denotes levels of the blocking factor. The quantitative response variables are in the numeric Values column.

Value

This function produces a data.frame object containing simulated data.

Examples

```
#Basic usage with two levels within FactorX2,
#20 experiments with inter-experiment SD 20, and residual SD 15

two_rb_tab <- make_2way_rb_data(c(100, 20), c(200, 300), 20, 20, 15)

str(two_rb_tab)
head(two_rb_tab)
```

---

make_2way_data *Make one-way or two-way independent group or randomised block design data.*

Description

The make_1way_data, make_1way_rb_data, make_2way_data and make_2way_rb_data functions generate independent or randomised block (rb) design data of one-way or two-way designs.

Usage

```
make_2way_data(Group_1_means, Group_2_means, Num_obs, Residual_SD)
```

Arguments

Group_1_means	a vector with means of each level of the first fixed factor (FixFac_X1) measured within Group 1.
Group_2_means	only for make_2way_data and make_2way_rb_data: a vector with mean(s) of each level of FactorX2 measured within Group 2.
Num_obs	a single numeric value indicating the number of independent measurements, i.e. levels within the random factor Experiment.
Residual_SD	a single numeric value indicating residual SD in the model.

Details

Random variates from the normal distribution based on user provided mean and SD provided are generated. For independent designs, the Residual_SD argument is used to set expected residual SD from the linear model. Exp_SD is used to set experiment-to-experiment SD, that will be assigned to the random factor for rb designs.

Num_obs sets the number of independent measurements per group.

For one-way designs, the user provides Group_means as a vector. Number of levels are recognised based on number of means. For two-way designs, two vectors are to be provided by the user containing means of levels of a second factor. Number of means in both vectors should be the same. These functions can only handle balanced designs, i.e. same number of observations in all groups.

The output is a data frame with one or two columns denoting the fixed factor with levels that match the number of means entered. For rb data, the column for RandFac denotes levels of the blocking factor. The quantitative response variables are in the numeric Values column.

Value

This function produces a data.frame object containing simulated data.

Examples

```
#Basic usage with two levels within FactorX2, 20 observations in each group, with residual SD 15

two_independent_tab <- make_2way_data(c(100, 20), c(200, 300), 20, 15)

#Four levels with 5 observations and residual SD 5
two_independent_tab <- make_2way_data(c(100, 20, 1500, 20), c(150, 5, 1450, 25), 5, 5)
```

make_2way_rb_data *Make one-way or two-way independent group or randomised block design data.*

Description

The make_1way_data, make_1way_rb_data, make_2way_data and make_2way_rb_data functions generate independent or randomised block (rb) design data of one-way or two-way designs.

Usage

```
make_2way_rb_data(Group1_means, Group2_means, Num_exp, Exp_SD, Residual_SD)
```

Arguments

Group1_means	a vector with means of each level of the first fixed factor (FixFac_X1) measured within Group 1.
Group2_means	only for make_2way_data and make_2way_rb_data: a vector with mean(s) of each level of FactorX2 measured within Group 2.
Num_exp	a single numeric value indicating the number of independent measurements, i.e. levels within the random factor RandFac.
Exp_SD	a single numeric value indicating the standard deviation (SD) between experiment, i.e. within RandFac.
Residual_SD	a single numeric value indicating residual SD in the model.

Details

Random variates from the normal distribution based on user provided mean and SD provided are generated. For independent designs, the Residual_SD argument is used to set expected residual SD from the linear model. Exp_SD is used to set experiment-to-experiment SD, that will be assigned to the random factor (RandFac) for rb designs.

Num_exp sets the number of independent measurements per group.

For one-way designs, the user provides Group_means as a vector. Number of levels are recognised based on number of means. For two-way designs, two vectors are to be provided by the user containing means of levels of a second factor. Number of means in both vectors should be the same. These functions can only handle balanced designs, i.e. same number of observations in all groups.

The output is a data frame with one or two columns denoting the fixed factor with levels that match the number of means entered. For rb data, the column for RandFac denotes levels of the blocking factor. The quantitative response variables are in the numeric Values column.

Value

This function produces a data.frame object containing simulated data.

Examples

```
#Basic usage with two levels within FactorX2,
#20 experiments with inter-experiment SD 20, and residual SD 15

two_rb_tab <- make_2way_rb_data(c(100, 20), c(200, 300), 20, 20, 15)

str(two_rb_tab)
head(two_rb_tab)
```

`mixed_anova` *ANOVA table from linear mixed effects analysis.*

Description

One of four related functions for mixed effects analyses (based on `lmer` and `as_lmerModLmerTest`) to get a linear model for downstream steps, or an ANOVA table.

1. `mixed_model`
2. `mixed_anova`
3. `mixed_model_slopes`
4. `mixed_anova_slopes`.

Usage

```
mixed_anova(
  data,
  Y_value,
  Fixed_Factor,
  Random_Factor,
  Df_method = "Kenward-Roger",
  SS_method = "II",
  AvgRF = TRUE,
  Formula = NULL,
  ...
)
```

Arguments

`data`	a data table object, e.g. data.frame or tibble.
`Y_value`	name of column containing quantitative (dependent) variable, provided within "quotes". The following transformations are permitted: "log(Y_value)", "log(Y_value + c)" where c a positive number, "logit(Y_value)" or "logit(Y_value/100)" which may be useful when `Y_value` are percentages (note quotes outside the log or logit calls); "sqrt(Y_value)" or "(Y_value)^2" should also work. During posthoc-comparisons, log and logit transformations will be back-transformed to the original scale. Other transformations, e.g., "sqrt(Y_value)" will not be back-transformed. Check out the `regrid` and `ref_grid` for details if you need back-transformation to the response scale.

Fixed_Factor	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). If a numeric variable is used, transformations similar to Y_value are permitted.
Random_Factor	name(s) of random factors to allow random intercepts; to be provided within quotes (e.g., "R") or as a vector when more than one (e.g., c("R1", "R2")).
Df_method	method for calculating degrees of freedom. Default is Kenward-Roger, can be changed to "Satterthwaite".
SS_method	type of sum of square, default is type II, can be changed to "I", "III", "1" or "2", or others.
AvgRF	this is a new argument since v5.0.0. The default AvgRF = TRUE will use the mean of Y_value (the response variable) grouped by levels of the Fixed_Factor and Random_Factor (using table_summary). This ensures that replicates within Random_Factor (or any other unused variable) are averaged (e.g., technical replicates nested within experimental blocks) before fitting a linear model and the denominator Df values are sensible. The name of the data frame in the model object will have (AvgRF) appended to it to indicate the averaging within levels of the Random_Factor. Using AvgRF = FALSE will lead to behaviour like versions <5.0.0.
Formula	directly provide an a formula (within quotes) as you would if you were using lmer. If Y_value, Fixed_Factor and Random_Factor are provided, they will be ignored. This is basically a wrapper, which may be useful if fitting more complex random factor structures.
...	any additional arguments to pass on to lmer if required.

Details

These functions require a data table, one dependent variable (Y_value), one or more independent variables (Fixed_Factor), and at least one random factor (Random_Factor). These should match names of variables in the long-format data table exactly. Since v5.0.0, if AvgRF = TRUE, the response variable is averaged over levels of the fixed and random factors (to collapse replicate observations) and reduce the number of denominator degrees of freedom. If you do not want to do this, set AvgRF = FALSE.

Outputs of mixed_model and mixed_model_slopes can be used for post-hoc comparisons with posthoc_Pairwise, posthoc_Levelwise, posthoc_vsRef, posthoc_Trends_Pairwise, posthoc_Trends_Levelwise and posthoc_Trends_vsRefor with emmeans.

More than one fixed factors can be provided as a vector (e.g. c("A", "B")). A full model with interaction term is fitted. This means when Y_value = Y, Fixed_factor = c("A", "B"), Random_factor = "R" are entered as arguments, these are passed on as Y ~ A*B + (1|R) (which is equivalent to Y ~ A + B + A:B + (1|R)).

In mixed_model_slopes and mixed_anova_slopes, the following kind of formula is used: Y ~ A*B + (S|R) (which is equivalent to Y ~ A + B + A:B + (S|R)). In this experimental implementation, random slopes and intercepts are fitted ((Slopes_Factor|Random_Factor)). Only one term each is allowed for Slopes_Factor and Random_Factor.

Value

ANOVA table of class "anova" and "data.frame".

Examples

```
#Usage with one fixed (Student) and random factor (Experiment)
mixed_anova(data = data_doubling_time,
Y_value = "Doubling_time",
Fixed_Factor = "Student",
Random_Factor = "Experiment")

#with formula
mixed_anova(data = data_doubling_time,
Formula = "Doubling_time ~ Student +(1|Experiment)")
```

`mixed_anova_slopes` *ANOVA table from linear mixed effects analysis.*

Description

One of four related functions for mixed effects analyses (based on `lmer` and `as_lmerModLmerTest`) to get a linear model for downstream steps, or an ANOVA table.

1. `mixed_model`
2. `mixed_anova`
3. `mixed_model_slopes`
4. `mixed_anova_slopes`.

Usage

```
mixed_anova_slopes(
  data,
  Y_value,
  Fixed_Factor,
  Slopes_Factor,
  Random_Factor,
  Df_method = "Kenward-Roger",
  SS_method = "II",
  AvgRF = TRUE,
  ...
)
```

Arguments

`data`	a data table object, e.g. data.frame or tibble.
`Y_value`	name of column containing quantitative (dependent) variable, provided within "quotes". The following transformations are permitted: "log(Y_value)", "log(Y_value + c)" where c a positive number, "logit(Y_value)" or "logit(Y_value/100)" which may be useful when `Y_value` are percentages (note quotes outside the log or

	logit calls); "sqrt(Y_value)" or "(Y_value)^2" should also work. During posthoc-comparisons, log and logit transformations will be back-transformed to the original scale. Other transformations, e.g., "sqrt(Y_value)" will not be back-transformed. Check out the `regrid` and `ref_grid` for details if you need back-transformation to the response scale.
`Fixed_Factor`	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). If a numeric variable is used, transformations similar to `Y_value` are permitted.
`Slopes_Factor`	name of factor to allow varying slopes on. Only one variable is allowed.
`Random_Factor`	name(s) of random factors to allow random intercepts; to be provided within quotes (e.g., "R") or as a vector when more than one (e.g., c("R1", "R2")). Only one variable is allowed.
`Df_method`	method for calculating degrees of freedom. Default is Kenward-Roger, can be changed to "Satterthwaite".
`SS_method`	type of sum of square, default is type II, can be changed to "I", "III", "1" or "2", or others.
`AvgRF`	this is a new argument since v5.0.0. The default `AvgRF = TRUE` will use the mean of `Y_value` (the response variable) grouped by levels of the `Fixed_Factor` and `Random_Factor` (using `table_summary`). This ensures that replicates within `Random_Factor` (or any other unused variable) are averaged (e.g., technical replicates nested within experimental blocks) before fitting a linear model and the denominator Df values are sensible. The name of the data frame in the model object will have `(AvgRF)` appended to it to indicate the averaging within levels of the `Random_Factor`. Using `AvgRF = FALSE` will lead to behaviour like versions <5.0.0.
`...`	any additional arguments to pass on to `lmer` if required.

Details

These functions require a data table, one dependent variable (Y_value), one or more independent variables (Fixed_Factor), and at least one random factor (Random_Factor). These should match names of variables in the long-format data table exactly. Since v5.0.0, if `AvgRF = TRUE`, the response variable is averaged over levels of the fixed and random factors (to collapse replicate observations) and reduce the number of denominator degrees of freedom. If you do not want to do this, set `AvgRF = FALSE`. If you do not want to do this, set `AvgRF = FALSE`.

For more advanced models with slopes and intercept, use `mixed_model` or `mixed_anova` using the `Formula` argument.

Outputs of `mixed_model` and `mixed_model_slopes` can be used for post-hoc comparisons with `posthoc_Pairwise`, `posthoc_Levelwise`, `posthoc_vsRef`, `posthoc_Trends_Pairwise`, `posthoc_Trends_Levelwise` and `posthoc_Trends_vsRef`or with `emmeans`.

More than one fixed factors can be provided as a vector (e.g. c("A", "B")). A full model with interaction term is fitted. This means when `Y_value = Y, Fixed_factor = c("A", "B"), Random_factor = "R"` are entered as arguments, these are passed on as `Y ~ A*B + (1|R)` (which is equivalent to `Y ~ A + B + A:B + (1|R)`).

In `mixed_model_slopes` and `mixed_anova_slopes`, the following kind of formula is used: `Y ~ A*B + (S|R)` (which is equivalent to `Y ~ A + B + A:B + (S|R)`). In this experimental implementation,

random slopes and intercepts are fitted (`(Slopes_Factor|Random_Factor)`). Only one term each is allowed for `Slopes_Factor` and `Random_Factor`.

Value

ANOVA table of class "anova" and "data.frame".

Examples

```
mixed_anova_slopes(data = data_2w_Tdeath,
Y_value = "PI",
Fixed_Factor = c("Genotype", "Time"),
Slopes_Factor = "Time",
Random_Factor = "Experiment")
```

`mixed_model` *Model from a linear mixed effects model*

Description

One of four related functions for mixed effects analyses (based on `lmer` and `as_lmerModLmerTest`) to get a linear model for downstream steps, or an ANOVA table.

1. `mixed_model`
2. `mixed_anova`
3. `mixed_model_slopes`
4. `mixed_anova_slopes`.

Usage

```
mixed_model(
  data,
  Y_value,
  Fixed_Factor,
  Random_Factor,
  AvgRF = TRUE,
  Formula = NULL,
  ...
)
```

Arguments

`data`	a data table object, e.g. data.frame or tibble.

Y_value	name of column containing quantitative (dependent) variable, provided within "quotes". The following transformations are permitted: "log(Y_value)", "log(Y_value + c)" where c a positive number, "logit(Y_value)" or "logit(Y_value/100)" which may be useful when Y_value are percentages (note quotes outside the log or logit calls); "sqrt(Y_value)" or "(Y_value)^2" should also work. During posthoc-comparisons, log and logit transformations will be back-transformed to the original scale. Other transformations, e.g., "sqrt(Y_value)" will not be back-transformed. Check out the regrid and ref_grid for details if you need back-transformation to the response scale.
Fixed_Factor	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). If a numeric variable is used, transformations similar to Y_value are permitted.
Random_Factor	name(s) of random factors to allow random intercepts; to be provided within quotes (e.g., "R") or as a vector when more than one (e.g., c("R1", "R2")).
AvgRF	this is a new argument since v5.0.0. The default AvgRF = TRUE will use the mean of Y_value (the response variable) grouped by levels of the Fixed_Factor and Random_Factor (using table_summary). This ensures that replicates within Random_Factor (or any other unused variable) are averaged (e.g., technical replicates nested within experimental blocks) before fitting a linear model and the denominator Df values are sensible. The name of the data frame in the model object will have (AvgRF) appended to it to indicate the averaging within levels of the Random_Factor. Using AvgRF = FALSE will lead to behaviour like versions <5.0.0.
Formula	directly provide an a formula (within quotes) as you would if you were using lmer. If Y_value, Fixed_Factor and Random_Factor are provided, they will be ignored. This is basically a wrapper, which may be useful if fitting more complex random factor structures.
...	any additional arguments to pass on to lmer if required.

Details

These functions require a data table, one dependent variable (Y_value), one or more independent variables (Fixed_Factor), and at least one random factor (Random_Factor). These should match names of variables in the long-format data table exactly. Since v5.0.0, if AvgRF = TRUE, the response variable is averaged over levels of the fixed and random factors (to collapse replicate observations) and reduce the number of denominator degrees of freedom. If you do not want to do this, set AvgRF = FALSE.

Outputs of mixed_model and mixed_model_slopes can be used for post-hoc comparisons with posthoc_Pairwise, posthoc_Levelwise, posthoc_vsRef, posthoc_Trends_Pairwise, posthoc_Trends_Levelwise and posthoc_Trends_vsRefor with emmeans.

More than one fixed factors can be provided as a vector (e.g. c("A", "B")). A full model with interaction term is fitted. This means when Y_value = Y, Fixed_factor = c("A", "B"), Random_factor = "R" are entered as arguments, these are passed on as Y ~ A*B + (1|R) (which is equivalent to Y ~ A + B + A:B + (1|R)).

In mixed_model_slopes and mixed_anova_slopes, the following kind of formula is used: Y ~ A*B + (S|R) (which is equivalent to Y ~ A + B + A:B + (S|R)). In this experimental implementation,

random slopes and intercepts are fitted ((Slopes_Factor|Random_Factor)). Only one term each is allowed for Slopes_Factor and Random_Factor.

Value

This function returns an S4 object of class "lmerModLmerTest".

Examples

```
#one fixed factor and random factor
mixed_model(data = data_doubling_time,
Y_value = "Doubling_time",
Fixed_Factor = "Student",
Random_Factor = "Experiment")

#with formula
mixed_anova(data = data_doubling_time,
Formula = "Doubling_time ~ Student +(1|Experiment)")

#' #save model
model <- mixed_model(data = data_doubling_time,
Y_value =  "Doubling_time",
Fixed_Factor = "Student",
Random_Factor = "Experiment")

#get model summary
summary(model)
```

---

mixed_model_slopes *Model from a linear mixed effects model with varying slopes*

Description

One of four related functions for mixed effects analyses (based on lmer and as_lmerModLmerTest) to get a linear model for downstream steps, or an ANOVA table.

1. mixed_model
2. mixed_anova
3. mixed_model_slopes
4. mixed_anova_slopes.

Usage

```
mixed_model_slopes(
  data,
  Y_value,
  Fixed_Factor,
  Slopes_Factor,
```

```
  Random_Factor,
  AvgRF = TRUE,
  ...
)
```

Arguments

`data`	a data table object, e.g. data.frame or tibble.
`Y_value`	name of column containing quantitative (dependent) variable, provided within "quotes". The following transformations are permitted: "log(Y_value)", "log(Y_value + c)" where c a positive number, "logit(Y_value)" or "logit(Y_value/100)" which may be useful when `Y_value` are percentages (note quotes outside the log or logit calls); "sqrt(Y_value)" or "(Y_value)^2" should also work. During posthoc-comparisons, log and logit transformations will be back-transformed to the original scale. Other transformations, e.g., "sqrt(Y_value)" will not be back-transformed. Check out the `regrid` and `ref_grid` for details if you need back-transformation to the response scale.
`Fixed_Factor`	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). If a numeric variable is used, transformations similar to `Y_value` are permitted.
`Slopes_Factor`	name of factor to allow varying slopes on. One one variable is allowed.
`Random_Factor`	name(s) of random factors to allow random intercepts; to be provided within quotes (e.g., "R") or as a vector when more than one (e.g., c("R1", "R2")). Only one variable is allowed.
`AvgRF`	this is a new argument since v5.0.0. The default `AvgRF = TRUE` will use the mean of `Y_value` (the response variable) grouped by levels of the `Fixed_Factor` and `Random_Factor` (using `table_summary`). This ensures that replicates within `Random_Factor` (or any other unused variable) are averaged (e.g., technical replicates nested within experimental blocks) before fitting a linear model and the denominator Df values are sensible. The name of the data frame in the model object will have `(AvgRF)` appended to it to indicate the averaging within levels of the `Random_Factor`. Using `AvgRF = FALSE` will lead to behaviour like versions <5.0.0.
`...`	any additional arguments to pass on to `lmer` if required.

Details

These functions require a data table, one dependent variable (Y_value), one or more independent variables (Fixed_Factor), and at least one random factor (Random_Factor). These should match names of variables in the long-format data table exactly. Since v5.0.0, if `AvgRF = TRUE`, the response variable is averaged over levels of the fixed and random factors (to collapse replicate observations) and reduce the number of denominator degrees of freedom. If you do not want to do this, set `AvgRF = FALSE`.

For more advanced models with slopes and intercept, use `mixed_model` or `mixed_anova` using the `Formula` argument.

Outputs of mixed_model and mixed_model_slopes can be used for post-hoc comparisons with posthoc_Pairwise, posthoc_Levelwise, posthoc_vsRef, posthoc_Trends_Pairwise, posthoc_Trends_Levelwise and posthoc_Trends_vsRefor with emmeans.

More than one fixed factors can be provided as a vector (e.g. c("A", "B")). A full model with interaction term is fitted. This means when Y_value = Y, Fixed_factor = c("A", "B"), Random_factor = "R" are entered as arguments, these are passed on as Y ~ A*B + (1|R) (which is equivalent to Y ~ A + B + A:B + (1|R)).

In mixed_model_slopes and mixed_anova_slopes, the following kind of formula is used: Y ~ A*B + (S|R) (which is equivalent to Y ~ A + B + A:B + (S|R)). In this experimental implementation, random slopes and intercepts are fitted ((Slopes_Factor|Random_Factor)). Only one term each is allowed for Slopes_Factor and Random_Factor.

Value

This function returns an S4 object of class "lmerModLmerTest".

Examples

```
#two fixed factors as a vector,
#exactly one slope factor and random factor
mod <- mixed_model_slopes(data = data_2w_Tdeath,
Y_value = "PI",
Fixed_Factor = c("Genotype", "Time"),
Slopes_Factor = "Time",
Random_Factor = "Experiment")
#get summary
summary(mod)
```

---

plot_3d_point_sd *Plot of mean & error bars for 1-way ANOVAs with matched shapes mapped to blocking factor.*

Description

One of 4 related functions for plotting 1-way ANOVA designs with a blocking factor.

1. plot_3d_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_3d_scatterbar (bar & SD, SEM or CI95 error bars)
3. plot_3d_scatterbox (box & whiskers)
4. plot_3d_scatterviolin (box & whiskers, violin)

Usage

```
plot_3d_point_sd(
  data,
  xcol,
  ycol,
  shapes,
  facet,
  ErrorType = "SD",
  symsize = 3.5,
  s_alpha = 1,
  symshape = 22,
  all_alpha = 0.3,
  all_size = 2.5,
  all_shape = 0,
  all_jitter = 0,
  ewid = 0.2,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  ethick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to be plotted on X axis.
ycol	name of the column (without quotes) with quantitative variable to plot on the Y axis.
shapes	name of the column (without quotes) with the blocking factor or another categorical variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).

symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set s_alpha = 0 to not show scatter plot.
symshape	The mean is shown with symbol of the shape number 21 (default, filled circle). Pick a number between 0-25 to pick a different type of symbol from ggplot2.
all_alpha	fractional opacity of all data points (default = 0.3). Set to non-zero value if you would like all data points plotted in addition to the mean.
all_size	size of symbols of all data points, if shown (default = 2.5).
all_shape	all data points are shown with symbols of the shape number 0 (default, open square). Pick a number between 0-25 to pick a different type of symbol from ggplot2.
all_jitter	reduce overlap of all data points, if shown, by setting a value between 0-1 (default = 0).
ewid	width of error bars, default set to 0.2.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
ethick	thickness of error bar lines; default fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify palettes or base R to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass.

Details

The blocking factor (or any other categorical variable) can be mapped to the shapes argument (up to 25 levels allowed). Variables passed to xcol and shapes are internally converted to factors even if they are numeric or other type of variables.

In plot_3d_point_sd and plot_3d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_3d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#3d version for 1-way data with blocking
#use plot_point_sd when no a blocking factor is not used
plot_3d_point_sd(data = data_1w_death,
xcol = Genotype, ycol = Death,
shapes = Experiment)
```

---

plot_3d_scatterbar *Plot a bar graph for 1-way ANOVAs with matched shapes mapped to blocking factor.*

Description

One of 4 related functions for plotting 1-way ANOVA designs with a blocking factor.

1. plot_3d_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_3d_scatterbar (bar & SD, SEM or CI95 error bars)
3. plot_3d_scatterbox (box & whiskers)
4. plot_3d_scatterviolin (box & whiskers, violin)

Usage

```
plot_3d_scatterbar(
  data,
  xcol,
  ycol,
  shapes,
  facet,
  ErrorType = "SD",
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  jitter = 0.1,
  ewid = 0.2,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to be plotted on X axis.
ycol	name of the column (without quotes) with quantitative variable to plot on the Y axis.
shapes	name of the column (without quotes) with the second categorical factor, for example from a two-way ANOVA design.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set s_alpha = 0 to not show scatter plot.

b_alpha	fractional opacity of boxes. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
ewid	width of error bars, default set to 0.2.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of lines of boxes; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify palettes or base R to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass.

Details

The blocking factor (or any other categorical variable) can be mapped to the shapes argument (up to 25 levels allowed). Variables passed to xcol and shapes are internally converted to factors even if they are numeric or other type of variables.

In plot_3d_point_sd and plot_3d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_3d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from

within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#3d version for 1-way data with blocking
#use plot_scatterbar_sd without blocking factor
plot_3d_scatterbar(data = data_1w_death,
xcol = Genotype, ycol = Death,
shapes = Experiment)
```

---

plot_3d_scatterbox *Plot a scatter and box plot for 1-way ANOVAs with matched shapes mapped to blocking factor.*

Description

One of 4 related functions for plotting 1-way ANOVA designs with a blocking factor.

1. plot_3d_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_3d_scatterbar (bar & SD, SEM or CI95 error bars)
3. plot_3d_scatterbox (box & whiskers)
4. plot_3d_scatterviolin (box & whiskers, violin)

Usage

```
plot_3d_scatterbox(
  data,
  xcol,
  ycol,
  shapes,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.5,
  jitter = 0.1,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
```

```
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to be plotted on X axis. If your table has numeric X, enter `xcol = factor(name of colum)`.
ycol	name of the column (without quotes) with quantitative variable to plot on the Y axis.
shapes	name of the column (without quotes) with the second categorical factor in a two-way ANOVA design.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set `s_alpha = 0` to not show scatter plot.
b_alpha	fractional opacity of boxes. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
bwid	width of boxes; default 0.5.
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is `waiver()`, or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be `fixed` (default), `free`, `free_y` or `free_x` (for Y and X axis one at a time, respectively).
fontsize	parameter of `base_size` of fonts in theme_classic, default set to size 20.

symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of lines of boxes; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_boxplot.

Details

The blocking factor (or any other categorical variable) can be mapped to the shapes argument (up to 25 levels allowed). Variables passed to xcol and shapes are internally converted to factors even if they are numeric or other type of variables.

In plot_3d_point_sd and plot_3d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_3d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#3d version for 1-way data with blocking
plot_3d_scatterbox(data = data_1w_death,
xcol = Genotype, ycol = Death,
shapes = Experiment)
#use plot_scatterbox without a blocking factor
```

plot_3d_scatterviolin *Plot a scatter with violin & box plot for 1-way ANOVAs with matched shapes mapped to blocking factor.*

Description

One of 4 related functions for plotting 1-way ANOVA designs with a blocking factor.

1. plot_3d_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_3d_scatterbar (bar & SD, SEM or CI95 error bars)
3. plot_3d_scatterbox (box & whiskers)
4. plot_3d_scatterviolin (box & whiskers, violin)

Usage

```
plot_3d_scatterviolin(
  data,
  xcol,
  ycol,
  shapes,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 0,
  v_alpha = 1,
  bwid = 0.3,
  vadjust = 1,
  jitter = 0.1,
  TextXAngle = 0,
  scale = "width",
  trim = TRUE,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
  vthick,
  bvthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
 ColSeq = TRUE,
 ColRev = FALSE,
 SingleColour = "NULL",
```

```
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to be plotted on X axis. If your table has numeric X, enter xcol = factor(name of colum).
ycol	name of the column (without quotes) with quantitative variable to plot on the Y axis.
shapes	name of the column (without quotes) with the second categorical factor in a two-way ANOVA design.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set s_alpha = 0 to not show scatter plot.
b_alpha	fractional opacity of boxes. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
v_alpha	fractional opacity of violins, default set to 1.
bwid	width of boxes (default 0.3).
vadjust	number to adjust the smooth/wigglyness of violin plot (default is 1).
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
scale	set to "area" by default, can be changed to "count" or "width".
trim	set whether tips of violin plot should be trimmed at high/low data. Default trim = T, can be changed to F.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxplots; default = fontsize/22.

vthick	thickness (in 'pt' units) of violins; default = fontsize/22.
bvthick	thickness (in 'pt' units) of both violins and boxplots; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_boxplot or geom_violin.

Details

The blocking factor (or any other categorical variable) can be mapped to the shapes argument (up to 25 levels allowed). Variables passed to xcol and shapes are internally converted to factors even if they are numeric or other type of variables.

In plot_3d_point_sd and plot_3d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_3d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#3d version for 1-way data with blocking
plot_3d_scatterviolin(data = data_1w_death,
xcol = Genotype, ycol = Death,
shapes = Experiment)
#use plot_scatterviolin without a blocking factor
```

plot_4d_point_sd *Plot mean & error bars for 2-way ANOVAs with or without a blocking factor.*

Description

There are 4 related functions for 2-way ANOVA type plots. In addition to a categorical variable along the X-axis, a grouping factor is passed to either `points`, `bars` or `boxes` argument in these functions. A blocking factor (or any other categorical variable) can be optionally passed to the `shapes` argument.

1. `plot_4d_point_sd` (mean & SD, SEM or CI95 error bars)
2. `plot_4d_scatterbar` (bar & SD, SEM or CI95 error bars)
3. `plot_4d_scatterbox` (box & whiskers)
4. `plot_4d_scatterviolin` (box & whiskers, violin)

Usage

```
plot_4d_point_sd(
  data,
  xcol,
  ycol,
  points,
  shapes,
  facet,
  ErrorType = "SD",
  symsize = 3.5,
  s_alpha = 1,
  symshape = 22,
  all_alpha = 0.3,
  all_size = 2.5,
  all_shape = 0,
  all_jitter = 0,
  ewid = 0.2,
  group_wid = 0.8,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  ethick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
```

```
  ColSeq = TRUE,
  ColRev = FALSE,
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the variable to plot on X axis (will be converted to a factor/categorical variable).
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
points	name of the column with grouping within the factor plotted on X-axis (will be converted to a factor/categorical variable).
shapes	name of the column (without quotes) that contains matched observations (e.g. subject IDs, experiment number) or another variable to pass on to symbol shapes (will be converted to a factor/categorical variable). If not provided, the shapes for all groups is the same, and can be changed with `all_shapes`, `all_alpha`, `all_size` etc.
facet	add another variable (without quotes) from the data table to create faceted graphs using `facet_wrap`.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
symsize	size of symbols, default set to 3.5.
s_alpha	fractional opacity of symbols, default set to 1 (i.e. fully opaque).
symshape	The mean is shown with symbol of the shape number 21 (default, filled circle). Pick a number between 0-25 to pick a different type of symbol from ggplot2.
all_alpha	fractional opacity of all data points (default = 0.3).
all_size	size of symbols of all data points, if shown (default = 2.5).
all_shape	all data points are shown with symbols of the shape number 0 (default, open square). Pick a number between 0-25 to pick a different type of symbol from ggplot2. This argument only has an effect if `shapes` argument is used.
all_jitter	reduce overlap of all data points, if shown, by setting a value between 0-1 (default = 0).
ewid	width of error bars, default set to 0.2.
group_wid	space between the factors along X-axis, i.e., dodge width. Default `group_wid = 0.8` (range 0-1), which can be set to 0 if you'd like the two plotted as `position = position_identity()`.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for `scale_y_continuous` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.

LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
ethick	thickness of error bar lines; default fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	any additional arguments to pass to stat_summary or geom_point.

Details

These can be especially useful when the fourth variable shapes is a random factor or blocking factor (up to 25 levels are allowed; there will be an error with more levels). The shapes argument can be left blank to plot ordinary 2-way ANOVAs without blocking.

In plot_4d_point_sd and plot_4d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_4d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#4d version for 2-way data with blocking
plot_4d_point_sd(data = data_2w_Tdeath,
xcol = Genotype,
ycol = PI,
points = Time,
shapes = Experiment)
```

```
#4d version without blocking factor
#`shapes` can be left blank
plot_4d_point_sd(data = data_2w_Festing,
xcol = Strain,
ycol = GST,
points = Treatment)
```

`plot_4d_scatterbar` *Plot scatter plot with bar & error bars for 2-way ANOVAs with or without a blocking factor.*

Description

There are 4 related functions for 2-way ANOVA type plots. In addition to a categorical variable along the X-axis, a grouping factor is passed to either `points`, `bars` or `boxes` argument in these functions. A blocking factor (or any other categorical variable) can be optionally passed to the `shapes` argument.

1. `plot_4d_point_sd` (mean & SD, SEM or CI95 error bars)
2. `plot_4d_scatterbar` (bar & SD, SEM or CI95 error bars)
3. `plot_4d_scatterbox` (box & whiskers)
4. `plot_4d_scatterviolin` (box & whiskers, violin)

Usage

```
plot_4d_scatterbar(
  data,
  xcol,
  ycol,
  bars,
  shapes,
  facet,
  ErrorType = "SD",
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.7,
  jitter = 0.1,
  ewid = 0.2,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
```

```
  fontsize = 20,
  group_wid = 0.8,
  symthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColRev = FALSE,
  ColSeq = TRUE,
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to plot on X axis. If column is numeric, enter as `factor(col)`.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
bars	name of the column containing grouping within the factor plotted on X axis. Can be categorical or numeric X. If your table has numeric X and you want to plot as factor, enter `xcol = factor(name of colum)`.
shapes	name of the column (without quotes) that contains matched observations (e.g. subject IDs, experiment number) or another variable to pass on to symbol shapes. If not provided, the shapes for all groups is the same.
facet	add another variable (without quotes) from the data table to create faceted graphs using `facet_wrap`.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set `s_alpha = 0` to not show scatter plot.
b_alpha	fractional opacity of boxes. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
bwid	width of boxes; default 0.7.
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
ewid	width of error bars, default set to 0.2.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for `scale_y_continuous` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.

LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
group_wid	space between the factors along X-axis, i.e., dodge width. Default group_wid = 0.8 (range 0-1), which can be set to 0 if you'd like the two plotted as position = position_identity().
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of bar and error bar lines; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
...	any additional arguments to pass to stat_summary or geom_point.

Details

These can be especially useful when the fourth variable shapes is a random factor or blocking factor (up to 25 levels are allowed; there will be an error with more levels). The shapes argument can be left blank to plot ordinary 2-way ANOVAs without blocking.

In plot_4d_point_sd and plot_4d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_4d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#4d version for 2-way data with blocking
plot_4d_scatterbar(data = data_2w_Tdeath,
```

```
xcol = Genotype,
ycol = PI,
bars = Time,
shapes = Experiment)

#4d version without blocking factor
#`shapes` can be left blank
plot_4d_scatterbar(data = data_2w_Tdeath,
xcol = Genotype,
ycol = PI,
bars = Time)
```

`plot_4d_scatterbox` *Plot scatter, box & whiskers for 2-way ANOVAs with or without a blocking factor.*

Description

There are 4 related functions for 2-way ANOVA type plots. In addition to a categorical variable along the X-axis, a grouping factor is passed to either `points`, `bars` or `boxes` argument in these functions. A blocking factor (or any other categorical variable) can be optionally passed to the `shapes` argument.

1. `plot_4d_point_sd` (mean & SD, SEM or CI95 error bars)
2. `plot_4d_scatterbar` (bar & SD, SEM or CI95 error bars)
3. `plot_4d_scatterbox` (box & whiskers)
4. `plot_4d_scatterviolin` (box & whiskers, violin)

Usage

```
plot_4d_scatterbox(
  data,
  xcol,
  ycol,
  boxes,
  shapes,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.7,
  jitter = 0.1,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
```

```
  facet_scales = "fixed",
  fontsize = 20,
  group_wid = 0.8,
  symthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to plot on X axis. If column is numeric, enter as `factor(col)`.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
boxes	name of the column containing grouping within the factor plotted on X axis. Can be categorical or numeric X. If your table has numeric X and you want to plot as factor, enter `xcol = factor(name of colum)`.
shapes	name of the column (without quotes) that contains matched observations (e.g. subject IDs, experiment number) or another variable to pass on to symbol shapes. If not provided, the shapes for all groups is the same.
facet	add another variable (without quotes) from the data table to create faceted graphs using `facet_wrap`.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set `s_alpha = 0` to not show scatter plot.
b_alpha	fractional opacity of boxes. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
bwid	width of boxes; default 0.7.
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for `scale_y_continuous` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.
LogYLabels	argument for `scale_y_continuous` for Y axis labels on log scales, default is `waiver()`, or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.

facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
group_wid	space between the factors along X-axis, i.e., dodge width. Default group_wid = 0.8 (range 0-1), which can be set to 0 if you'd like the two plotted as position = position_identity().
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxplot lines; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	any additional arguments to pass to geom_boxplot.

Details

These can be especially useful when the fourth variable shapes is a random factor or blocking factor (up to 25 levels are allowed; there will be an error with more levels). The shapes argument can be left blank to plot ordinary 2-way ANOVAs without blocking.

In plot_4d_point_sd and plot_4d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_4d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#4d version for 2-way data with blocking
plot_4d_scatterbox(data = data_2w_Tdeath,
xcol = Genotype,
ycol = PI,
boxes = Time,
shapes = Experiment)
```

```
#4d version without blocking factor
#`shapes` can be left blank
plot_4d_scatterbox(data = data_2w_Tdeath,
xcol = Genotype,
ycol = PI,
boxes = Time)
```

`plot_4d_scatterviolin` *Plot scatter, box & violin for 2-way ANOVAs with or without a blocking factor.*

Description

There are 4 related functions for 2-way ANOVA type plots. In addition to a categorical variable along the X-axis, a grouping factor is passed to either `points`, `bars` or `boxes` argument in these functions. A blocking factor (or any other categorical variable) can be optionally passed to the `shapes` argument.

1. `plot_4d_point_sd` (mean & SD, SEM or CI95 error bars)
2. `plot_4d_scatterbar` (bar & SD, SEM or CI95 error bars)
3. `plot_4d_scatterbox` (box & whiskers)
4. `plot_4d_scatterviolin` (box & whiskers, violin)

Usage

```
plot_4d_scatterviolin(
  data,
  xcol,
  ycol,
  boxes,
  shapes,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  v_alpha = 1,
  b_alpha = 0,
  bwid = 0.3,
  vadjust = 1,
  jitter = 0.1,
  TextXAngle = 0,
  scale = "width",
  trim = TRUE,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
```

```
  fontsize = 20,
  group_wid = 0.8,
  symthick,
  bthick,
  vthick,
  bvthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  ...
)
```

Arguments

data	a data table, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with the categorical factor to plot on X axis. If column is numeric, enter as `factor(col)`.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
boxes	name of the column containing grouping within the factor plotted on X axis. Can be categorical or numeric X. If your table has numeric X and you want to plot as factor, enter `xcol = factor(name of colum)`.
shapes	name of the column (without quotes) that contains matched observations (e.g. subject IDs, experiment number) or another variable to pass on to symbol shapes. If not provided, the shapes for all groups is the same.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e. 80% opacity). Set `s_alpha = 0` to not show scatter plot.
v_alpha	fractional opacity of violins, default set to 1.
b_alpha	fractional opacity of boxes. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
bwid	width of boxes; default 0.3.
vadjust	number to adjust the smooth/wigglyness of violin plot (default set to 1).
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
scale	set to "area" by default, can be changed to "count" or "width".
trim	set whether tips of violin plot should be trimmed at high/low data. Default `trim = T`, can be changed to F.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).

LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
group_wid	space between the factors along X-axis, i.e., dodge width. Default group_wid = 0.8 (range 0-1), which can be set to 0 if you'd like the two plotted as position = position_identity().
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxplots; default = fontsize/22.
vthick	thickness (in 'pt' units) of violins; default = fontsize/22.
bvthick	thickness (in 'pt' units) of both violins and boxplots; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	any additional arguments to pass to geom_boxplot or geom_violin.

Details

These can be especially useful when the fourth variable shapes is a random factor or blocking factor (up to 25 levels are allowed; there will be an error with more levels). The shapes argument can be left blank to plot ordinary 2-way ANOVAs without blocking.

In plot_4d_point_sd and plot_4d_scatterbar, the default error bar is SD (can be changed to SEM or CI95). In plot_4d_point_sd, a large coloured symbol is plotted at the mean, all other data are shown as smaller symbols. Boxplot uses geom_boxplot to depict median (thicker line), box (interquartile range (IQR)) and the whiskers (1.5*IQR).

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#4d version for 2-way data with blocking
plot_4d_scatterviolin(data = data_2w_Tdeath,
xcol = Genotype,
ycol = PI,
boxes = Time,
shapes = Experiment)

#4d version without blocking factor
#`shapes` can be left blank
plot_4d_scatterviolin(data = data_2w_Tdeath,
xcol = Genotype,
ycol = PI,
boxes = Time)
```

---

plot_befafter_box *Before-after style graph with a boxplot*

Description

One of 3 related functions to plot matching data joined by lines. The variable containing information for matching (e.g. matched subjects or experiments etc.) is passed to the `match` argument.

1. `plot_befafter_colours` or `plot_befafter_colors`,
2. `plot_befafter_shapes`
3. `plot_befafter_box`

Usage

```
plot_befafter_box(
  data,
  xcol,
  ycol,
  match,
  facet,
  PlotShapes = FALSE,
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.4,
  jitter = 0.1,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
```

```
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
  lthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) containing the categorical variable to be plotted on the X axis.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
match	name of the column (without quotes) with the grouping variable to pass on to geom_line.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
PlotShapes	logical TRUE or FALSE (default = FALSE) if the shape of the symbol is to be mapped to the match variable. Note that only 25 shapes allowed.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e., 80% opacity).
b_alpha	fractional opacity of boxes, default set to 1.
bwid	width of boxplots; default 0.4.
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).

fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxes; default = (fontsize)/22.
lthick	thickness (in 'pt' units) of lines; default = (fontsize/1.2)/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_line, geom_point, or facet_wrap.

Details

In plot_befafter_colours/plot_befafter_colors and plot_befafter_shapes setting Boxplot = TRUE will also plot a box and whiskers plot.

Note that only 25 shapes are available, and there will be errors with plot_befafter_shapes when there are fewer than 25 matched observations; instead use plot_befafter_colours instead.

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

To plot a graph with a single colour along the X axis variable, use the SingleColour argument.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_befafter_box(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
match = Subject)

#with PlotShapes = TRUE
plot_befafter_box(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
match = Subject, PlotShapes = TRUE)
```

plot_befafter_colours *Plot a before-after plot with lines joining colour-matched symbols.*

Description

One of 3 related functions to plot matching data joined by lines. The variable containing information for matching (e.g. matched subjects or experiments etc.) is passed to the `match` argument.

1. `plot_befafter_colours` or `plot_befafter_colors`,
2. `plot_befafter_shapes`
3. `plot_befafter_box`

Usage

```
plot_befafter_colours(
  data,
  xcol,
  ycol,
  match,
  facet,
  Boxplot = FALSE,
  symsize = 3,
  s_alpha = 1,
  jitter = 0.1,
  bwid = 0.4,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
  lthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)

plot_befafter_colors(
  data,
  xcol,
```

```
  ycol,
  match,
  facet,
  Boxplot = FALSE,
  symsize = 3,
  s_alpha = 1,
  jitter = 0.1,
  bwid = 0.4,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
  lthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) containing the categorical variable to be plotted on the X axis.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
match	name of the column (without quotes) with the matching variable to pass on to geom_line.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
Boxplot	logical TRUE/FALSE, whether to show box and whisker plot or not (default is FALSE)
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 1 (i.e. maximum opacity & zero transparency).
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
bwid	width of boxplots; default 0.4.

TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxes; default = (fontsize)/22.
lthick	thickness (in 'pt' units) of lines; default = (fontsize/1.2)/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_line or geom_point.

Details

In plot_befafter_colours/plot_befafter_colors and plot_befafter_shapes setting Boxplot = TRUE will also plot a box and whiskers plot.

Note that only 25 shapes are available, and there will be errors with plot_befafter_shapes when there are fewer than 25 matched observations; instead use plot_befafter_colours instead.

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

To plot a graph with a single colour along the X axis variable, use the SingleColour argument.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#plot without legends if necessary
plot_befafter_colours(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
match = Subject, s_alpha = .9, ColSeq = FALSE)+
guides(fill = "none",
colour = "none") #remove guides
#plot with boxplot
plot_befafter_colours(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
match = Subject, s_alpha = .9, ColSeq = FALSE,
Boxplot = TRUE)+
guides(fill = "none",
colour = "none") #remove guides
```

---

plot_befafter_shapes *Plot a before-after plot with lines joining shape-matched symbols.*

Description

One of 3 related functions to plot matching data joined by lines. The variable containing information for matching (e.g. matched subjects or experiments etc.) is passed to the `match` argument.

1. `plot_befafter_colours` or `plot_befafter_colors`,
2. `plot_befafter_shapes`
3. `plot_befafter_box`

Usage

```
plot_befafter_shapes(
  data,
  xcol,
  ycol,
  match,
  facet,
  Boxplot = FALSE,
  symsize = 3,
  s_alpha = 1,
  bwid = 0.4,
  jitter = 0.1,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
```

```
  fontsize = 20,
  symthick,
  bthick,
  lthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) containing the categorical variable to be plotted on the X axis.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis.
match	name of the column (without quotes) with the matching variable to pass on to geom_line.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
Boxplot	logical TRUE/FALSE, whether to show box and whisker plot or not (default is FALSE)
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 1.
bwid	width of boxplots; default 0.4.
jitter	extent of jitter (scatter) of symbols, default is 0.1. Increase to reduce symbol overlap, set to 0 for aligned symbols.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.

bthick	thickness (in 'pt' units) of boxes; default = (fontsize)/22.
lthick	thickness (in 'pt' units) of lines; default = (fontsize/1.2)/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_line or geom_point.

Details

In plot_befafter_colours/plot_befafter_colors and plot_befafter_shapes setting Boxplot = TRUE will also plot a box and whiskers plot.

Note that only 25 shapes are available, and there will be errors with plot_befafter_shapes when there are fewer than 25 matched observations; instead use plot_befafter_colours instead.

Colours can be changed using ColPal, ColRev or ColSeq arguments. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

To plot a graph with a single colour along the X axis variable, use the SingleColour argument.

The resulting ggplot2 graph can take additional geometries or other layers.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#plot without legends if necessary
plot_befafter_colors(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
match = Subject, s_alpha = .9, ColSeq = FALSE)+
guides(fill = "none",
colour = "none") #remove guides
#plot with boxplot
plot_befafter_colors(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
match = Subject, s_alpha = .9, ColSeq = FALSE,
Boxplot = TRUE)+
guides(fill = "none",
colour = "none") #remove guides
```

plot_density *Plot density distribution of data.*

Description

This function takes a data table, ycol of quantitative variable and a categorical grouping variable (group), if available, and plots a density graph using geom_density). Alternatives are plot_histogram, or plot_qqline.

Usage

```
plot_density(
  data,
  ycol,
  group,
  facet,
  PlotType = c("Density", "Counts", "Normalised counts"),
  c_alpha = 0.2,
  TextXAngle = 0,
  facet_scales = "fixed",
  fontsize = 20,
  linethick,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = NULL,
  ...
)
```

Arguments

data	a data table e.g. data.frame or tibble.
ycol	name of the column (without quotes) with the quantitative variable whose density distribution is to be plotted.
group	name of the column containing a categorical grouping variable (optional Since v5.0.0).
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
PlotType	the default (Density) plot will be the probability density curve, which can be changed to Counts or Normalised counts.

c_alpha	fractional opacity of filled colours under the curve, default set to 0.2 (i.e. 20% opacity).
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
linethick	thickness of symbol border, default set to fontsize/22.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify palettes or base R to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_density.

Details

Note that the function requires the quantitative Y variable first, and groups them based on an X variable. The group variable is mapped to the fill and colour aesthetics in geom_density. Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_density(data = data_t_pratio,
ycol = log(Cytokine), group = Genotype)

#with faceting
plot_density(data = data_cholesterol,
ycol = Cholesterol, group = Treatment,
facet = Treatment, fontsize = 12)

#Counts
plot_density(data = data_cholesterol,
ycol = Cholesterol, group = Treatment,
PlotType = "Counts",
facet = Treatment, fontsize = 12)
```

`plot_dotbar_sd` *Plot a dotplot on a bar graph with SD error bars with two variables.*

Description

There are three types of `plot_dot_` functions that plot data as "dots" using the `geom_dotplot` geometry. They all take a data table, a categorical X variable and a numeric Y variable.

1. plot_dotbar_sd (bar & SD, SEM or CI95 error bars)
2. plot_dotbox (box & whiskers)
3. plot_dotviolin (box & whiskers, violin)

Usage

```
plot_dotbar_sd(
  data,
  xcol,
  ycol,
  facet,
  ErrorType = "SD",
  dotsize = 1.5,
  d_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.5,
  ewid = 0.2,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
```

```
  fontsize = 20,
  dotthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) to plot on X axis. This should be a categorical variable.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis. This should be a quantitative variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
dotsize	size of dots relative to binwidth used by geom_dotplot. Default set to 1.5, increase/decrease as needed.
d_alpha	fractional opacity of dots, default set to 0.8 (i.e., 80% opacity).
b_alpha	fractional opacity of bars, default set to 1.
bwid	width of bars; default 0.5.
ewid	width of error bars, default set to 0.2.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
dotthick	thickness of dot border (stroke parameter of geom_dotplot), default set to fontsize/22.

bthick	thickness (in 'pt' units) of bar and error bar lines; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_dotplot.

Details

Related plot_scatter_ variants show data symbols using the geom_point geometry. These are plot_scatterbar_sd (or SEM or CI95 error bars), plot_scatterbox and plot_scatterviolin. Over plotting in plot_scatter variants can be reduced with the jitter argument.

The X variable is mapped to the fill aesthetic of dots, symbols, bars, boxes and violins.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

If you prefer a single colour for the graph, use the SingleColour argument.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_dotbar_sd(data = data_cholesterol,
xcol = Treatment,
ycol = Cholesterol)

plot_dotbar_sd(data = data_1w_death,
xcol = Genotype, ycol = Death,
ColPal = "pale", ColSeq = FALSE, ColRev = TRUE)

#single colour along X
plot_dotbar_sd(data = data_1w_death,
xcol = Genotype, ycol = Death,
SingleColour = "light_orange")
```

plot_dotbox *Plot a dotplot on a boxplot with two variables.*

Description

There are three types of plot_dot_ functions that plot data as "dots" using the geom_dotplot geometry. They all take a data table, a categorical X variable and a numeric Y variable.

1. plot_dotbar_sd (bar & SD, SEM or CI95 error bars)
2. plot_dotbox (box & whiskers)
3. plot_dotviolin (box & whiskers, violin)

Usage

```
plot_dotbox(
  data,
  xcol,
  ycol,
  facet,
  dotsize = 1.5,
  d_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.5,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  dotthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) to plot on X axis. This should be a categorical variable.

ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis. This should be a quantitative variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
dotsize	size of dots relative to binwidth used by geom_dotplot. Default set to 1.5, increase/decrease as needed.
d_alpha	fractional opacity of dots, default set to 0.8 (i.e., 80% opacity).
b_alpha	fractional opacity of boxes, default set to 1.
bwid	width of boxplots; default 0.5.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
dotthick	thickness of dot border (stroke parameter of geom_dotplot), default set to fontsize/22.
bthick	thickness (in 'pt' units) of boxplot lines; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_boxplot or geom_dotplot.

Details

Related plot_scatter_ variants show data symbols using the geom_point geometry. These are plot_scatterbar_sd (or SEM or CI95 error bars), plot_scatterbox and plot_scatterviolin. Over plotting in plot_scatter variants can be reduced with the jitter argument.

The X variable is mapped to the fill aesthetic of dots, symbols, bars, boxes and violins.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

If you prefer a single colour for the graph, use the SingleColour argument.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_dotbox(data = data_1w_death,
xcol = Genotype, ycol = Death)

plot_dotbox(data = data_1w_death,
xcol = Genotype, ycol = Death,
ColPal = "vibrant", b_alpha = 0.5)
plot_dotbox(data = data_1w_death,
xcol = Genotype, ycol = Death,
SingleColour = "safe_bluegreen", b_alpha = 0.5)
```

---

plot_dotviolin *Plot a dotplot on a violin plot with two variables.*

Description

There are three types of plot_dot_ functions that plot data as "dots" using the geom_dotplot geometry. They all take a data table, a categorical X variable and a numeric Y variable.

1. plot_dotbar_sd (bar & SD, SEM or CI95 error bars)
2. plot_dotbox (box & whiskers)
3. plot_dotviolin (box & whiskers, violin)

Usage

```
plot_dotviolin(
  data,
  xcol,
  ycol,
  facet,
  dotsize = 1.5,
  d_alpha = 0.8,
  b_alpha = 0,
  v_alpha = 1,
```

```
  bwid = 0.3,
  vadjust = 1,
  trim = TRUE,
  scale = "width",
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  dotthick,
  bthick,
  vthick,
  bvthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) to plot on X axis. This should be a categorical variable.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis. This should be a quantitative variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
dotsize	size of dots relative to binwidth used by geom_dotplot. Default set to 1.5, increase/decrease as needed.
d_alpha	fractional opacity of dots, default set to 0.8 (i.e., 80% opacity).
b_alpha	fractional opacity of boxplots. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
v_alpha	fractional opacity of violins, default set to 1.
bwid	width of boxplots; default 0.3.
vadjust	number to adjust the smooth/wigglyness of violin plot (default set to 1).
trim	set whether tips of violin plot should be trimmed at high/low data. Default trim = TRUE, can be changed to FALSE.
scale	set to "area" by default, can be changed to "count" or "width".
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.

LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
dotthick	thickness of dot border (stroke parameter of geom_dotplot), default set to fontsize/22.
bthick	thickness (in 'pt' units) of boxplots; default = fontsize/22.
vthick	thickness (in 'pt' units) of violins; default = fontsize/22.
bvthick	thickness (in 'pt' units) of both violins and boxplots; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_boxplot, geom_dotplot or geom_violin.

Details

Related plot_scatter_ variants show data symbols using the geom_point geometry. These are plot_scatterbar_sd (or SEM or CI95 error bars), plot_scatterbox and plot_scatterviolin. Over plotting in plot_scatter variants can be reduced with the jitter argument.

The X variable is mapped to the fill aesthetic of dots, symbols, bars, boxes and violins.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

If you prefer a single colour for the graph, use the SingleColour argument.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#plot with trim = FALSE
plot_dotviolin(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
dotsize = 2, trim = FALSE)

plot_dotviolin(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
trim = FALSE, b_alpha = 0.5,
ColPal = "pale", ColSeq = FALSE)

#single colour along X
plot_dotviolin(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
trim = FALSE, b_alpha = 0.5,
SingleColour = "pale_cyan")
```

---

plot_gam_predict *Plot prediction of* gam *model*

Description

Plot prediction of gam model

Usage

```
plot_gam_predict(
  Model,
  xcol,
  ycol,
  ByFactor,
  symsize = 1,
  s_alpha = 0.1,
  smooth_alpha = 0.7,
  linethick,
  fontsize = 20,
  ...
)
```

Arguments

Model	a generalised additive model (gam) fitted with ga_model or mgcv
xcol	the smooth in the gam (should match variable in the model exactly)
ycol	the dependent variable in gam (should match variable in the model exactly)
ByFactor	the by factor used in gam (should match variable in the model exactly)
symsize	size of symbols (default = 1)

s_alpha	opacity of symbols (default = 0.1)
smooth_alpha	opacity of the predicted CI interval (default = 0.7)
linethick	thickness of symbol lines (default = fontsize/22)
fontsize	base font size for graph
...	additional arguments to pass to plot_xy_CatGroup.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#fit zooplankton data
z1 <- ga_model(data = data_zooplankton,
Y_value = "log(density_adj)",
Fixed_Factor = "taxon",
Smooth_Factor = "day")

#plot fitted data
plot_gam_predict(Model = z1,
xcol = day,
ycol = `log(density_adj)`,
ByFactor = taxon)
```

---

plot_grafify_palette *See grafify colour palettes*

Description

This simple function allows quick visualisation of colours in grafify palettes and their hex codes. It uses plot_scatterbar_sd and some arguments are similar and can be adjusted.

Usage

```
plot_grafify_palette(
 palette = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant", "OrBl_div", "PrGn_div",
    "blue_conti", "grey_conti", "yellow_conti"),
  fontsize = 14,
  ...
)
```

Arguments

palette	name of grafify palettes: "okabe_ito", "vibrant, "bright", "pale", "muted", "dark", "light", "contrast" or "all_grafify".
fontsize	font size.
...	any additional parameters to pass to plot_scatterbar_sd

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_grafify_palette("pale")
plot_grafify_palette("contrast")
```

---

plot_histogram *Plot data distribution as histograms.*

Description

This function takes a data table, a quantitative variable (ycol) and a grouping variable (group), if available, and plots a histogram graph using geom_histogram). Alternatives are plot_density, or plot_qqline.

Usage

```
plot_histogram(
  data,
  ycol,
  group,
  facet,
  PlotType = c("Counts", "Normalised counts"),
  BinSize = 30,
  c_alpha = 0.8,
  TextXAngle = 0,
  facet_scales = "fixed",
  fontsize = 20,
  linethick,
  alpha,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
```

```
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = NULL,
  ...
)
```

Arguments

data	a data table e.g. data.frame or tibble.
ycol	name of the column (without quotes) with the quantitative variable whose histogram distribution is to be plotted.
group	name of the column containing a categorical grouping variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
PlotType	the default (Counts) plot will plot counts in the bins, which can be changed to Normalised counts.
BinSize	number of distinct bins to use on X-axis, default set to 30.
c_alpha	fractional opacity of colour filled within histograms, default set to 0.8 (i.e. 80% opacity).
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
linethick	thickness of symbol border, default set to fontsize/22.
alpha	deprecated old argument for c_alpha; retained for backward compatibility.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify palettes or base R to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_histogram.

Details

Note that the function requires the quantitative Y variable first, and groups them based on a categorical variable passed on via the `group` argument. The grouping variable is mapped to the `fill` aesthetics in `geom_histogram`.

ColPal & ColRev options are applied to both `fill` and `colour` scales. Colours available can be seen quickly with `plot_grafify_palette`. Colours can be changed using `ColPal`, `ColRev` or `ColSeq` arguments. `ColPal` can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". `ColRev` (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. `ColSeq` decides whether colours are picked by respecting the order in the palette or the most distant ones using `colorRampPalette`.

Value

This function returns a `ggplot2` object of class "gg" and "ggplot".

Examples

```
#Basic usage
plot_histogram(data = data_t_pratio,
ycol = Cytokine, group = Genotype,
BinSize = 10)
#with log transformation
plot_histogram(data = data_t_pratio,
ycol = log(Cytokine), group = Genotype,
BinSize = 10)
#Normalised counts
plot_histogram(data = data_t_pratio,
ycol = log(Cytokine), group = Genotype,
PlotType = "Normalised counts",
BinSize = 10)
```

---

plot_lm_predict *Plot data and predictions from linear model*

Description

This function takes a linear model, and up to three variables and plots observe data (circles) and model predictions (squares). If the X-variable is categorical, a box and whiskers plot is overlaid. A variable (`ByFactor`) can be used for faceting.

Usage

```
plot_lm_predict(
  Model,
  xcol,
  ycol,
  ByFactor,
  obs_size = 2,
```

```
  obs_alpha = 0.3,
  pred_size = 2,
  pred_alpha = 0.8,
  linethick,
  base_size = 15,
  ...
)
```

Arguments

Model	a linear model saved with simple_model, mixed_model or ga_model.
xcol	variable along the X axis (should match one of the dependent variables in model exactly).
ycol	independent variable along the Y axis (should match independent variable in model exactly).
ByFactor	optional faceting variable (should match one of the variables in model exactly).
obs_size	size of symbols for observed data (default = 2).
obs_alpha	opacity of symbols for observed data (default = 0.3).
pred_size	size of symbols for predicted data (default = 2).
pred_alpha	opacity of symbols for predicted data (default = 0.8).
linethick	thickness of border lines for boxes and symbols (default is base_size/20).
base_size	base fontsize for theme_grafify
...	any other parameters to be passed to theme_grafify

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#fit a model
deathm1 <- mixed_model(data_2w_Tdeath,
"PI", c("Genotype", "Time"),
"Experiment")
#plot model
plot_lm_predict(deathm1,
Genotype, PI, Time)
#fit zooplankton data
z1 <- ga_model(data = data_zooplankton,
Y_value = "log(density_adj)",
Fixed_Factor = "taxon",
Smooth_Factor = "day")

#plot fitted data
plot_lm_predict(Model = z1,
xcol = day,
ycol = `log(density_adj)`,
ByFactor = taxon)
```

plot_logscale *Add log transformations to graphs*

Description

This function allows "log10" or "log2" transformation of X or Y axes. With "log10" transformation, log10 ticks are also added on the outside.

Usage

```
plot_logscale(
  Plot,
  LogYTrans = "log10",
  LogXTrans,
  LogYBreaks = waiver(),
  LogXBreaks = waiver(),
  LogYLimits = NULL,
  LogXLimits = NULL,
  LogYLabels = waiver(),
  LogXLabels = waiver(),
  fontsize = 22,
  ...
)
```

Arguments

Plot	a ggplot2 object.
LogYTrans	transform Y axis into "log10" (default) or "log2"
LogXTrans	transform X axis into "log10" or "log2"
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogXBreaks	argument for scale_x_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
LogXLimits	a vector of length two specifying the range (minimum and maximum) of the X axis.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogXLabels	argument for scale_x_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
fontsize	this parameter sets the linewidth of the log10 tickmarks (8*fontsize/22 for long ticks and 4*fontsize/22 for middle ticks). It is set to 20 as default to be consistent with rest of grafify. It will need to be changed to 12, which is the default fontsize for graphs produced natively with ggplot2.
...	any other arguments to pass to scale_y_continuous or scale_x_continuous

Details

Arguments allow for axes limits, breaks and labels to passed on.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#save a ggplot object
P <- ggplot(data_t_pratio,
aes(Genotype,Cytokine))+
geom_jitter(shape = 21,
size = 5, width = .2,
aes(fill = Genotype),
alpha = .7)
#transform Y axis
plot_logscale(Plot = P)

#or in one go
plot_logscale(ggplot(data_t_pratio,
aes(Genotype,Cytokine))+
geom_jitter(shape = 21,
size = 5, width = .2,
aes(fill = Genotype),
alpha = .7))
```

---

plot_point_sd *Plot a point as mean with SD error bars using two variables.*

Description

There are 4 related functions that use geom_point to plot a categorical variable along the X axis.

1. plot_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_scatterbar_sd (bar & SD, SEM or CI95 error bars)
3. plot_scatterbox (box & whiskers)
4. plot_scatterviolin (box & whiskers, violin)

Usage

```
plot_point_sd(
  data,
  xcol,
  ycol,
  facet,
  ErrorType = "SD",
```

```
  symsize = 3.5,
  s_alpha = 1,
  symshape = 22,
  all_alpha = 0.3,
  all_size = 2.5,
  all_shape = 1,
  all_jitter = 0,
  ewid = 0.2,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  ethick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with a X variable (will be forced to be a factor/categorical variable).
ycol	name of the column (without quotes) with quantitative Y variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
symsize	size of point symbols, default set to 3.5.
s_alpha	fractional opacity of symbols, default set to 1 (i.e. maximum opacity & zero transparency).
symshape	The mean is shown with symbol of the shape number 22 (default, filled square). Pick a number between 21-25 to pick a different type of symbol from ggplot2.
all_alpha	fractional opacity of all data points (default = 0.3). Set to non-zero value if you would like all data points plotted in addition to the mean.
all_size	size of symbols of all data points, if shown (default = 2.5).
all_shape	all data points are shown with symbols of the shape number 1 (default, transparent circle). Pick a number between 0-25 to pick a different type of symbol from ggplot2.

all_jitter	reduce overlap of all data points, if shown, by setting a value between 0-1 (default = 0).
ewid	width of error bars, default set to 0.2.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	thickness of symbol border, default set to fontsize/22.
ethick	thickness of error bar lines; default fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to stat_summary.

Details

These functions take a data table, categorical X and numeric Y variables, and plot various geometries. The X variable is mapped to the fill aesthetic of symbols.

In plot_point_sd and plot_scatterbar_sd, default error bars are SD, which can be changed to SEM or CI95.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

If there are many groups along the X axis and you prefer a single colour for the graph,use the SingleColour argument.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#Basic usage
plot_point_sd(data = data_doubling_time,
xcol = Student, ycol = Doubling_time)
```

---

plot_qqline *Plot quantile-quantile (QQ) graphs from data.*

Description

This function takes a data table, a quantitative variable (ycol), and a categorical grouping variable (group), if available, and plots a QQ graph using stat_qq and stat_qq_line. Alternatives are plot_histogram, or plot_qqline.

Usage

```
plot_qqline(
  data,
  ycol,
  group,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  TextXAngle = 0,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  linethick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  ...
)
```

Arguments

data	a data table e.g. data.frame or tibble.
ycol	name of the column (without quotes) with the quantitative variable whose distribution is to be plotted.
group	name of the column containing a categorical grouping variable.

facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 1 (i.e. maximum opacity & zero transparency).
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	thickness of symbol border, default set to fontsize/22.
linethick	thickness of lines, default set to fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	any additional arguments to pass to geom_qq or geom_qq_line.

Details

Note that the function requires the quantitative Y variable first, and a grouping variable as group if required. The graph plots sample quantiles on Y axis & theoretical quantiles on X axis. The X variable is mapped to the fill aesthetic instat_qq and colour aesthetic for the stat_qq_line.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. When only one level is present within group, symbols will receive "ok_orange" colour. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_qqline(data = data_cholesterol,
ycol = Cholesterol, group = Treatment)
```

plot_qqmodel *Plot quantile-quantile (QQ) graphs from residuals of linear models.*

Description

This function takes a linear model (simple or mixed effects) and plots a QQ graph after running rstudent from rstudent to generate a table of Studentised model residuals on an ordinary (simple_model), mixed model (mixed_model or mixed_model_slopes. The graph plots studentised residuals from the model (sample) on Y axis & Theoretical quantiles on X axis.

Usage

```
plot_qqmodel(
  Model,
  symsize = 3,
  s_alpha = 0.8,
  fontsize = 20,
  symthick,
  linethick,
  SingleColour = "#E69F00"
)
```

Arguments

Model	name of a saved model generated by simple_model or mixed_model.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e., 80% opacity).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	thickness of symbol border, default set to fontsize/22.
linethick	thickness of line, default set to fontsize/22.
SingleColour	colour of symbols (default = #E69F00, which is ok_orange)

Details

For generalised additive models fit with mgcv, scaled Pearson residuals are plotted.

The function uses geom_qq and geom_qq_line geometries. Symbols receive "ok_orange" colour by default.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#Basic usage
m1 <- simple_model(data = data_2w_Festing,
Y_value = "GST",
Fixed_Factor = c("Treatment", "Strain"))
plot_qqmodel(m1)
```

---

plot_qq_gam *Plot model diagnostics for generalised additive models*

Description

This is a clone of the appraise function in the gratia package (rewritten to avoid depending on gratia package for these plots). This function will plot 4 diagnostic plots when given a generalised additive model fitted with ga_model or mgcv. It creates graphs that use grafify colours and theme_grafify().

Usage

```
plot_qq_gam(
  Model,
  symsize = 2,
  s_colour = "#E69F00",
  s_alpha = 0.6,
  line_col = "black",
  base_size = 12,
  linethick,
  n_bins = c("sturges", "scott", "fd")
)
```

Arguments

Model	a model of class gam fitted with ga_model or the mgcv package.
symsize	size of symbols (default = 2)
s_colour	colour of symbols (default = ok_orange)
s_alpha	opacity of symbols (default = 0.8)
line_col	colour of lines (default = black)
base_size	font size for theme (default = 12)
linethick	thickness in 'pt' units of lines and symbol orders (default = base_size/22)
n_bins	one of either "sturges", "scott", "fd"

Value

This function returns an object of classes "ggplot" and "gg".

This function returns a ggplot2 object of class "gg" and "ggplot".

plot_scatterbar_sd *Plot scatter dots on a bar graph with SD error bars with two variables.*

Description

There are 4 related functions that use `geom_point` to plot a categorical variable along the X axis.

1. plot_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_scatterbar_sd (bar & SD, SEM or CI95 error bars)
3. plot_scatterbox (box & whiskers)
4. plot_scatterviolin (box & whiskers, violin)

Usage

```
plot_scatterbar_sd(
  data,
  xcol,
  ycol,
  facet,
  ErrorType = "SD",
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.5,
  ewid = 0.3,
  jitter = 0.1,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.

xcol	name of the column (without quotes) to plot on X axis. This should be a categorical variable.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis. This should be a quantitative variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
symsize	size of point symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e, 80% opacity).
b_alpha	fractional opacity of boxes, default set to .5 (i.e. 50% transparent).
bwid	width of bars, default set to 0.5.
ewid	width of error bars, default set to 0.3.
jitter	extent of jitter (scatter) of symbols, default is 0.1.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all facet facet graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of both bar and error bar lines; default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify palettes or base R to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to facet_wrap.

Details

These functions take a data table, categorical X and numeric Y variables, and plot various geometries. The X variable is mapped to the `fill` aesthetic of symbols.

In plot_point_sd and plot_scatterbar_sd, default error bars are SD, which can be changed to SEM or CI95.

Colours can be changed using `ColPal`, `ColRev` or `ColSeq` arguments. Colours available can be seen quickly with `plot_grafify_palette`. `ColPal` can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". `ColRev` (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. `ColSeq` decides whether colours are picked by respecting the order in the palette or the most distant ones using `colorRampPalette`.

If you prefer a single colour for the graph, use the `SingleColour` argument.

Value

This function returns a `ggplot2` object of class "gg" and "ggplot".

Examples

```
#with jitter
plot_scatterbar_sd(data = data_cholesterol,
xcol = Treatment, ycol = Cholesterol)

plot_scatterbar_sd(data = data_doubling_time,
xcol = Student, ycol = Doubling_time,
SingleColour = "ok_grey")
```

---

plot_scatterbox *Plot a scatter plot on a boxplot with two variables.*

Description

There are 4 related functions that use `geom_point` to plot a categorical variable along the X axis.

1. plot_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_scatterbar_sd (bar & SD, SEM or CI95 error bars)
3. plot_scatterbox (box & whiskers)
4. plot_scatterviolin (box & whiskers, violin)

Usage

```
plot_scatterbox(
  data,
  xcol,
  ycol,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 1,
  bwid = 0.5,
  jitter = 0.1,
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
  Ylabels = waiver(),
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) to plot on X axis. This should be a categorical variable.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis. This should be a quantitative variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
symsize	size of symbols, default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e, 80% opacity).
b_alpha	fractional opacity of boxes, default set to 1.
bwid	width of boxplots; default 0.5.
jitter	extent of jitter (scatter) of symbols, default is 0.1.
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.

LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
facet_scales	whether or not to fix scales on X & Y axes for all graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxplot lines; default = fontsize/22.
Ylabels	deprecated, use LogYLabels instead.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_boxplot.

Details

These functions take a data table, categorical X and numeric Y variables, and plot various geometries. The X variable is mapped to the fill aesthetic of symbols.

In plot_point_sd and plot_scatterbar_sd, default error bars are SD, which can be changed to SEM or CI95.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

If you prefer a single colour for the graph, use the SingleColour argument.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
plot_scatterbox(data = data_cholesterol,
xcol = Treatment, ycol = Cholesterol)

plot_scatterbox(data = data_doubling_time,
xcol = Student, ycol = Doubling_time,
SingleColour = "ok_grey")
```

`plot_scatterviolin` *Plot a scatter plot on a violin plot with two variables.*

Description

There are 4 related functions that use `geom_point` to plot a categorical variable along the X axis.

1. plot_point_sd (mean & SD, SEM or CI95 error bars)
2. plot_scatterbar_sd (bar & SD, SEM or CI95 error bars)
3. plot_scatterbox (box & whiskers)
4. plot_scatterviolin (box & whiskers, violin)

Usage

```
plot_scatterviolin(
  data,
  xcol,
  ycol,
  facet,
  symsize = 3,
  s_alpha = 0.8,
  b_alpha = 0,
  v_alpha = 1,
  bwid = 0.3,
  vadjust = 1,
  jitter = 0.1,
  trim = TRUE,
  scale = "width",
  TextXAngle = 0,
  LogYTrans,
  LogYBreaks = waiver(),
  LogYLabels = waiver(),
  LogYLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  symthick,
  bthick,
```

```
  vthick,
  bvthick,
  Ylabels = waiver(),
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  SingleColour = "NULL",
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) to plot on X axis. This should be a categorical variable.
ycol	name of the column (without quotes) with the quantitative variable to plot on the Y axis. This should be a quantitative variable.
facet	add another variable (without quotes) from the data table to create faceted graphs using facet_wrap.
symsize	size of dots relative to binwidth used by geom_point. Default set to 3.
s_alpha	fractional opacity of symbols, default set to to 0.8 (i.e, 80% opacity). Set s_alpha = 0 to not show scatter plot.
b_alpha	fractional opacity of boxplots. Default is set to 0, which results in white boxes inside violins. Change to any value >0 up to 1 for different levels of transparency.
v_alpha	fractional opacity of violins, default set to 1.
bwid	width of boxplots; default 0.3.
vadjust	number to adjust the smooth/wigglyness of violin plot (default set to 1).
jitter	extent of jitter (scatter) of symbols, default is 0 (i.e. aligned symbols). To reduce symbol overlap, try 0.1-0.3 or higher.
trim	set whether tips of violin plot should be trimmed at high/low data. Default trim = T, can be changed to F.
scale	set to "area" by default, can be changed to "count" or "width".
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogYBreaks	argument for scale_y_continuous for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for scale_y_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.

facet_scales	whether or not to fix scales on X & Y axes for all graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	thickness (in 'pt' units) of boxplots; default = fontsize/22.
vthick	thickness (in 'pt' units) of violins; default = fontsize/22.
bvthick	thickness (in 'pt' units) of both violins and boxplots; default = fontsize/22.
Ylabels	deprecated, use LogYLabels instead.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
SingleColour	a colour hexcode (starting with #, e.g., "#E69F00"), a number between 1-154, or names of colours from grafify or base R palettes to fill along X-axis aesthetic. Accepts any colour other than "black"; use grey_lin11, which is almost black.
...	any additional arguments to pass to geom_boxplot, geom_point or geom_violin.

Details

These functions take a data table, categorical X and numeric Y variables, and plot various geometries. The X variable is mapped to the fill aesthetic of symbols.

In plot_point_sd and plot_scatterbar_sd, default error bars are SD, which can be changed to SEM or CI95.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

If you prefer a single colour for the graph, use the SingleColour argument.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#plot without jitter
plot_scatterviolin(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
symsize = 2, trim = FALSE)

#no symbols
```

```
plot_scatterviolin(data = data_t_pdiff,
xcol = Condition, ycol = Mass,
s_alpha = 0,
symsize = 2, trim = FALSE)
```

`plot_xy_CatGroup` *Plot points on a quantitative X - Y plot & a categorical grouping variable.*

Description

This function takes a data table, quantitative X and Y variables along with a categorical grouping variable, and a and plots a graph with using `geom_point`. The categorical `CatGroup` variable is mapped to the `fill` aesthetic of symbols.

Usage

```
plot_xy_CatGroup(
  data,
  xcol,
  ycol,
  CatGroup,
  facet,
  Boxplot = FALSE,
  Mean = FALSE,
  ErrorType = "SD",
  symsize = 3,
  s_alpha = 0.8,
  TextXAngle = 0,
  LogYTrans,
  LogXTrans,
  LogYBreaks = waiver(),
  LogXBreaks = waiver(),
  LogYLabels = waiver(),
  LogXLabels = waiver(),
  LogYLimits = NULL,
  LogXLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  bwid = 0.3,
  b_alpha = 0.3,
  l_alpha = 0.8,
  e_alpha = 0.8,
  all_size = 2,
  all_alpha = 0.5,
  symthick,
```

```
  bthick,
  ethick,
  ewid = 0.2,
 ColPal = c("okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly",
    "light", "muted", "pale", "r4", "safe", "vibrant"),
  ColSeq = TRUE,
  ColRev = FALSE,
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with quantitative X variable.
ycol	name of the column (without quotes) with quantitative Y variable.
CatGroup	a categorical variable as grouping factor for colour of data points, should be a categorical variable for default colours to work. Will be converted to `factor` if your column is numeric
facet	add another variable (without quotes) from the data table to create faceted graphs using `facet_wrap`.
Boxplot	logical TRUE/FALSE to plot box and whiskers plot (default = FALSE).
Mean	logical TRUE/FALSE to plot mean and SD/SEM/CI95 error bars (default = FALSE).
ErrorType	select the type of error bars to display. Default is "SD" (standard deviation). Other options are "SEM" (standard error of the mean) and "CI95" (95% confidence interval based on t distributions).
symsize	size of symbols used by `geom_point`. Default set to 3.
s_alpha	fractional opacity of symbols, default set to to 0.8 (i.e, 80% opacity).
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogXTrans	transform X axis into "log10" or "log2"
LogYBreaks	argument for `ggplot2[scale_y_continuous]` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.
LogXBreaks	argument for `ggplot2[scale_x_continuous]` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.
LogYLabels	argument for `ggplot2[scale_y_continuous]` for Y axis labels on log scales, default is `waiver()`, or provide a vector of desired labels.
LogXLabels	argument for `ggplot2[scale_x_continuous]` for Y axis labels on log scales, default is `waiver()`, or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
LogXLimits	a vector of length two specifying the range (minimum and maximum) of the X axis.

facet_scales	whether or not to fix scales on X & Y axes for all graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
bwid	width of boxplot (default = 0.3).
b_alpha	fractional opacity of boxes, (default = 0.3).
l_alpha	fractional opacity of lines joining boxes, (default = 0.8).
e_alpha	fractional opacity of error bars, (default = 0.8).
all_size	size of symbols of all data points, if shown (default = 2.5).
all_alpha	fractional opacity of all data points (default = 0.3). Set to non-zero value if you would like all data points plotted in addition to the mean.
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	size (in 'pt' units) of outline of boxes, whisker and joining lines (stroke), default = fontsize/22.
ethick	thickness of error bar lines; default fontsize/22.
ewid	width of error bars, default set to 0.2.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	any additional arguments to pass on.

Details

A box and whisker plot with lines joining the medians can be plotted with Boxplot = TRUE. If only box plot is needed without the line, set the opacity of the line to 0 (i.e., l_alpha = 0).

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

This plot is related to plot_xy_NumGroup which requires a numeric grouping factor. When summary statistics (mean/median) are required, use plot_3d_scatterbar, plot_3d_scatterbox or plot_4d_scatterbox.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#The grouping factor cyl  is automatically converted to categorical variable
plot_xy_CatGroup(data = mtcars,
xcol = mpg, ycol = disp, CatGroup = cyl,
ColPal = "vibrant", ColSeq = FALSE)

#with boxplot
plot_xy_CatGroup(data = mpg,
xcol = cyl, ycol = cty,
CatGroup = fl, Boxplot = TRUE)

#add another variable
#with boxplot
plot_xy_CatGroup(data = mpg,
xcol = cyl, ycol = cty,
CatGroup = fl, facet = drv,
Boxplot = TRUE)
```

---

plot_xy_Group *Plot points on a quantitative X - Y plot & a grouping variable.*

Description

This function takes a data table, quantitative X and Y variables along with a grouping variable that is either categorical or numeric. The colour of data symbols is mapped to the grouping variable. This function is related to plot_xy_CatGroup and plot_xy_NumGroup, and could eventually replace them in future updates.

Central value and scatter can be shown by choosing the ErrorType argument. Mean and error bars (SD, SEM or CI95) or box and whisker plot options are available.

Usage

```
plot_xy_Group(
  data,
  xcol,
  ycol,
  Group,
  facet,
  ErrorType = c("none", "SD", "SEM", "CI95", "Boxplot"),
  SmoothType = c("none", "Loess", "Linear"),
  symsize = 3,
  s_alpha,
  TextXAngle = 0,
  mean_size,
  m_alpha = 1,
  LogYTrans,
  LogXTrans,
```

```
  LogYBreaks = waiver(),
  LogXBreaks = waiver(),
  LogYLabels = waiver(),
  LogXLabels = waiver(),
  LogYLimits = NULL,
  LogXLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  bwid = 0.3,
  b_alpha = 0.3,
  l_alpha = 0.8,
  sm_alpha = 0.3,
  symthick,
  bthick,
  ewid = 0.1,
  e_alpha = 1,
  ColPal = NULL,
  ColSeq = TRUE,
  ColRev = FALSE,
  ...
)
```

Arguments

`data`	A data frame containing the variables to be plotted.
`xcol`	A column name in `data` for the x-axis (typically a factor or grouping variable).
`ycol`	A column name in `data` for the y-axis (numeric).
`Group`	A grouping variable used for colour/fill aesthetics. Whether this variable is numeric or not will determine colour palette choice for the `ColPal` argument.
`facet`	An optional variable for faceting the plot using `facet_wrap()`.
`ErrorType`	select the way to show data centrality and dispersion. The default is "none", which can be changed to "SD" (standard deviation), "SEM" (standard error of the mean) or "CI95" (95% confidence interval based on *t* distributions); all these will be displayed with a square symbol representing the mean. Choosing `Boxplot` will show a box and whiskers plot and the median. A line joining the central values will also appear. Use `l_alpha = 0` to not show the line.
`SmoothType`	Add a smoothened average using 'Loess' or 'Linear', which will be passed on to `stat_smooth` as `method = "loess"` or `method = "lm"`, respectively.
`symsize`	Size of the raw data points. Default is `3`.
`s_alpha`	Alpha transparency for raw data points. Default is `0.8`, which will reduce to `0.2` when an `ErrorType` is set.
`TextXAngle`	Angle of x-axis text labels. Default is `0`.
`mean_size`	Size of the square symbol representing the mean. Default is `symsize + 3` to prominently show the central value.
`m_alpha`	Alpha transparency for the mean symbol. Default is `1`.

LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogXTrans	transform X axis into "log10" or "log2"
LogYBreaks	argument for ggplot2[scale_y_continuous] for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogXBreaks	argument for ggplot2[scale_x_continuous] for Y axis breaks on log scales, default is waiver(), or provide a vector of desired breaks.
LogYLabels	argument for ggplot2[scale_y_continuous] for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogXLabels	argument for ggplot2[scale_x_continuous] for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
LogXLimits	a vector of length two specifying the range (minimum and maximum) of the X axis.
facet_scales	whether or not to fix scales on X & Y axes for all graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
bwid	width of boxplot (default = 0.3).
b_alpha	fractional opacity of boxes (default = 0.3).
l_alpha	fractional opacity of lines joining boxes, (default = 0.8).
sm_alpha	fractional opacity of error range around loess or linear smooth fit (default = 0.3).
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	size (in 'pt' units) of outline of boxes, whisker and joining lines (stroke), default = fontsize/22.
ewid	width of errorbars (default = 0.1).
e_alpha	fractional opacity of error bars (default = 1).
ColPal	Character. Name of the colour palette to use from grafify. For categorical variables, one of: "okabe_ito", "all_grafify", "bright", "contrast", "dark", "fishy", "kelly", "light", "muted", "pale", "r4", "safe", "vibrant". For quantitative variables, one of: blue_conti, yellow_conti, grey_conti, PrGn_div, PrGn_div.
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours, which will be applied using scale_fill_grafify2.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	Additional arguments passed to ggplot2 geoms or scales.

Details

When ErrorType is set to a value other than none, a line joining the central value (mean or median, respectively) will also appear (set l_alpha = 0 if this is not desired). Other options for ErrorType are: SD, SEM, CI95, Boxplot.

When SD/SEM/CI95 are chosen, the mean value will appear as a larger square symbol. Its opacity and size can be adjusted with m_alpha and mean_size, respectively.

The smooth fitted line type can be shown with the SmoothType argument, which can take the following options: none, Linear or Loess. Latter two options are fitted using stat_smooth with lm or loess options, respectively.

Colours can be changed using ColPal, ColRev or ColSeq arguments. Colours available can be seen quickly with plot_grafify_palette. ColPal can be one of the following: "okabe_ito", "dark", "light", "bright", "pale", "vibrant, "muted" or "contrast". ColRev (logical TRUE/FALSE) decides whether colours are chosen from first-to-last or last-to-first from within the chosen palette. ColSeq (logical TRUE/FALSE) decides whether colours are picked by respecting the order in the palette or the most distant ones using colorRampPalette.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#The grouping factor cyl  is automatically converted to categorical variable
plot_xy_Group(data = mpg,
xcol = cyl, ycol = cty, Group = drv,
ErrorType = "SD")

#with a Linear smooth line
plot_xy_Group(data = mpg,
xcol = cyl, ycol = cty, Group = drv,
ErrorType = "SD", SmoothType = "Linear")
```

---

plot_xy_NumGroup *Plot points on a quantitative X - Y plot & a numeric grouping variable.*

Description

This function takes a data table, quantitative X and Y variables, and a numeric grouping variable, and a and plots a graph with using geom_point. The numerical NumGroup variable is mapped to the fill aesthetic of symbols, which receives the scale_fill_grafify default quantitative palette (blue_conti). Alternatives are yellow_conti, grey_conti, OrBl_div and PrGn_div. Colour order can be reversed with ColRev = TRUE (default is FALSE).

Usage

```
plot_xy_NumGroup(
  data,
  xcol,
  ycol,
  NumGroup,
  facet,
```

```
  Boxplot = FALSE,
  symsize = 3,
  s_alpha = 0.8,
  TextXAngle = 0,
  LogYTrans,
  LogXTrans,
  LogYBreaks = waiver(),
  LogXBreaks = waiver(),
  LogYLabels = waiver(),
  LogXLabels = waiver(),
  LogYLimits = NULL,
  LogXLimits = NULL,
  facet_scales = "fixed",
  fontsize = 20,
  bwid = 0.3,
  b_alpha = 0.3,
  l_alpha = 0.8,
  symthick,
  bthick,
 ColPal = c("blue_conti", "yellow_conti", "grey_conti", "PrGn_div", "OrBl_div"),
  ColRev = FALSE,
  ...
)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
xcol	name of the column (without quotes) with quantitative X variable.
ycol	name of the column (without quotes) with quantitative Y variable.
NumGroup	a numeric factor for `fill` aesthetic of data points.
facet	add another variable (without quotes) from the data table to create faceted graphs using `facet_wrap`.
Boxplot	logical TRUE/FALSE to plot box and whiskers plot (default = FALSE).
symsize	size of symbols used by `geom_point`. Default set to 3.
s_alpha	fractional opacity of symbols, default set to 0.8 (i.e, 80% opacity).
TextXAngle	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
LogYTrans	transform Y axis into "log10" or "log2" (in quotes).
LogXTrans	transform X axis into "log10" or "log2"
LogYBreaks	argument for `scale_y_continuous` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.
LogXBreaks	argument for `scale_x_continuous` for Y axis breaks on log scales, default is `waiver()`, or provide a vector of desired breaks.
LogYLabels	argument for `scale_y_continuous` for Y axis labels on log scales, default is `waiver()`, or provide a vector of desired labels.

LogXLabels	argument for scale_x_continuous for Y axis labels on log scales, default is waiver(), or provide a vector of desired labels.
LogYLimits	a vector of length two specifying the range (minimum and maximum) of the Y axis.
LogXLimits	a vector of length two specifying the range (minimum and maximum) of the X axis.
facet_scales	whether or not to fix scales on X & Y axes for all graphs. Can be fixed (default), free, free_y or free_x (for Y and X axis one at a time, respectively).
fontsize	parameter of base_size of fonts in theme_classic, default set to size 20.
bwid	width of boxplot (default = 0.3).
b_alpha	fractional opacity of boxes, (default = 0.3).
l_alpha	fractional opacity of lines joining boxes, (default = 0.8).
symthick	size (in 'pt' units) of outline of symbol lines (stroke), default = fontsize/22.
bthick	size (in 'pt' units) of outline of boxes, whisker and joining lines (stroke), default = fontsize/22.
ColPal	grafify colour palette to apply (in quotes), default "okabe_ito"; see graf_palettes for available palettes.
ColRev	whether to reverse order of colour within the selected palette, default F (FALSE); can be set to T (TRUE).
...	any additional arguments to pass on.

Details

This plot is related to plot_xy_CatGroup which requires a categorical grouping factor. When summary statistics (mean/median) are required, use plot_3d_scatterbar, plot_3d_scatterbox or plot_4d_scatterbox.

Value

This function returns a ggplot2 object of class "gg" and "ggplot".

Examples

```
#The grouping factor gear is numeric
plot_xy_NumGroup(data = mtcars,
xcol = mpg, ycol = disp, NumGroup = cyl,
s_alpha = 0.8)
#change colour palette
plot_xy_NumGroup(data = mtcars,
xcol = mpg, ycol = disp, NumGroup = cyl,
s_alpha = 0.8,
ColPal = "grey_conti")
```

posthoc_Levelwise *Level-wise post-hoc comparisons from a linear or linear mixed effects model.*

Description

This function is a wrapper based on emmeans, and needs a ordinary linear model produced by simple_model or a mixed effects model produced by mixed_model or mixed_model_slopes (or generated directly with lm, lme4 or lmerTest calls). It also needs to know the fixed factor(s), which should match those in the model and data table.

Usage

```
posthoc_Levelwise(Model, Fixed_Factor, P_Adj = "fdr", Factor, ...)
```

Arguments

Model	a model object fit using simple_model or mixed_model or related.
Fixed_Factor	one or more categorical variables, provided as a vector (e.g., c("A", "B")), whose levels you wish to compare pairwise. Names of Fixed_Factor should match Fixed_Factor used to fit the model. When more than one factor is provided e.g. Fixed_factor = c("A", "B"), this function passes this on as specs = A|B (note the vertical | between the two Fixed_Factor) to emmeans to produce comparisons between each level A with each other listed separately at each level of B.
P_Adj	method for correcting P values for multiple comparisons. Default is set to false discovery rate ("fdr"), can be changed to "none", "tukey", "bonferroni", "sidak" (in quotes). See Interaction analysis in emmeans in the manual for emmeans.
Factor	old argument name for Fixed_Factor; retained for backward compatibility.
...	additional arguments for emmeans such as lmer.df or others. See help for sophisticated models in emmeans.

Details

The function will generate level-wise comparisons (as described in Comparisons and contrasts in emmeans), i.e. comparison between of every level of one factor separately at each level of the other factor. By default, P values are corrected by the FDR method (which can be changed). If the model was fit by transforming the quantitative response variable using "log", "logit", "sqrt" etc., results will still be on the original scale, i.e. type = "response" is the default; data will be back-transformed (check results to confirm this), and for log or logit see Transformations and link functions in emmeans, ratios will be compared. The first part of the emmeans results has the estimated marginal means, SE and CI ($emmeans), which are generated from the fitted model, and **not** the original data table. The second part has the results of the comparisons ($contrasts).

Value

returns an "emm_list" object containing contrasts and emmeans through emmeans.

Examples

```
#make a linear model first
CholMod <- mixed_model(data = data_cholesterol,
Y_value =  "Cholesterol",
Fixed_Factor = c("Hospital", "Treatment"),
Random_Factor =  "Subject")

#note Fixed_Factor needs a vector or variable in quotes
#to get comparisons between different hospitals separately for each level of Treatment
posthoc_Levelwise(Model = CholMod,
Fixed_Factor = c("Hospital", "Treatment"))

#get comparisons between treatments separately at each hospital
posthoc_Levelwise(Model = CholMod,
Fixed_Factor = c("Treatment", "Hospital"))
```

`posthoc_Pairwise` *Pairwise post-hoc comparisons from a linear or linear mixed effects model.*

Description

This function is a wrapper based on `emmeans`, and needs a ordinary linear model produced by `simple_model` or a mixed effects model produced by `mixed_model` or `mixed_model_slopes` (or generated directly with `lm`, `lme4` or `lmerTest` calls). It also needs to know the fixed factor(s), which should match those in the model and data table.

Usage

```
posthoc_Pairwise(Model, Fixed_Factor, P_Adj = "fdr", Factor, ...)
```

Arguments

`Model`	a model object fit using `simple_model` or `mixed_model` or related.
`Fixed_Factor`	one or more categorical variables, provided as a vector (e.g., c("A", "B")), whose levels you wish to compare pairwise. Names of Fixed_Factor should match Fixed_Factor used to fit the model. When more than one factor is provided e.g. `Fixed_factor = c("A", "B")`, this function passes this on as `specs = A:B` (note the colon between the two Fixed_Factor) to `emmeans` to produce pairwise comparisons.
`P_Adj`	method for correcting P values for multiple comparisons. Default is set to false discovery rate ("fdr"), can be changed to "none", "tukey", "bonferroni", "sidak" (in quotes). See Interaction analysis in emmeans in the manual for `emmeans`.
`Factor`	old argument name for `Fixed_Factor`; retained for backward compatibility.
`...`	additional arguments for `emmeans` such as `lmer.df` or others. See help for sophisticated models in emmeans.

Details

The function will generate pairwise comparisons of every level of every factor (as described in Comparisons and contrasts in emmeans). Too many comparisons will be generated and only use this when necessary. By default, P values are corrected by the FDR method (which can be changed). If the model was fit by transforming the quantitative response variable using "log", "logit", "sqrt" etc., results will still be on the original scale, i.e. `type = "response"` is the default; data will be back-transformed (check results to confirm this), and for log or logit see Transformations and link functions in emmeans, ratios will be compared. The first part of the `emmeans` results has the estimated marginal means, SE and CI (`$emmeans`), which are generated from the fitted model, and **not** the original data table. The second part has the results of the comparisons (`$contrasts`).

Value

returns an "emm_list" object containing contrasts and emmeans through `emmeans`.

Examples

```
#make linear models first
DoublMod <- simple_model(data = data_doubling_time,
Y_value =  "Doubling_time", Fixed_Factor = "Student")

#mixed model
CholMod <- mixed_model(data = data_cholesterol,
Y_value = "Cholesterol",
Fixed_Factor = c("Hospital", "Treatment"),
Random_Factor = "Subject")

posthoc_Pairwise(Model = DoublMod,
Fixed_Factor = "Student")

#basic use with two Fixed_Factor provided as a vector
posthoc_Pairwise(Model = CholMod,
Fixed_Factor = c("Treatment", "Hospital"))

#same call with "tukey" adjustment
posthoc_Pairwise(Model = CholMod,
Fixed_Factor = c("Treatment", "Hospital"),
P_adj = "tukey")
```

`posthoc_Trends_Levelwise` *Use emtrends to get level-wise comparison of slopes from a linear model.*

Description

This function is a wrapper based on emmeans, and needs a ordinary linear model produced by simple_model or a mixed effects model produced by mixed_model or mixed_model_slopes (or generated directly with lm, lme4 or lmerTest calls). At least one of the factors should be a numeric covariate whose slopes you wish to find. It also needs to know the fixed factor(s), which should match those in the model and data table.

Usage

```
posthoc_Trends_Levelwise(
  Model,
  Fixed_Factor,
  Trend_Factor,
  P_Adj = "sidak",
  ...
)
```

Arguments

Model	a model object fit using simple_model or mixed_model (or lm or lmer).
Fixed_Factor	one or more categorical variables, provided as a vector (e.g., c("A", "B")), whose levels you wish to compare pairwise. Names of Fixed_Factor should match Fixed_Factor used to fit the model. When more than one factor is provided e.g. Fixed_factor = c("A", "B"), this function passes this on as specs = A:B (note the colon between the two Fixed_Factor) to emmeans to produce pairwise comparisons.
Trend_Factor	a quantitative variable that interacts with a factor and whose slope (trend) is to be compared
P_Adj	method for correcting P values for multiple comparisons. Default is "sidak", can be changed to "bonferroni". See Interaction analysis in emmeans in the manual for emmeans.
...	additional arguments for emmeans such as lmer.df or others. See help for sophisticated models in emmeans.

Details

Checkout the Interactions with covariates section in the emmeans vignette for more details. One of the independent variables should be a quantitative (e.g. time points) variable whose slope (trend) you want to find at levels of the other factor.

Value

returns an "emm_list" object containing slopes and their contrasts calculated through emtrends.

Examples

```
#create an lm model
#Time2 is numeric (time points)
```

```
m1 <- simple_model(data = data_2w_Tdeath,
Y_value = "PI", Fixed_Factor = c("Genotype", "Time2"))
posthoc_Trends_Levelwise(Model = m1,
Fixed_Factor = "Genotype",
Trend_Factor = "Time2")
```

`posthoc_Trends_Pairwise`

Use emtrends to get pairwise comparison of slopes from a linear model.

Description

This function is a wrapper based on `emmeans`, and needs a ordinary linear model produced by `simple_model` or a mixed effects model produced by `mixed_model` or `mixed_model_slopes` (or generated directly with `lm`, `lme4` or `lmerTest` calls). At least one of the factors should be a numeric covariate whose slopes you wish to find. It also needs to know the fixed factor(s), which should match those in the model and data table.

Usage

```
posthoc_Trends_Pairwise(
  Model,
  Fixed_Factor,
  Trend_Factor,
  P_Adj = "sidak",
  ...
)
```

Arguments

`Model`	a model object fit using `simple_model` or `mixed_model` (or `lm` or `lmer`).
`Fixed_Factor`	one or more categorical variables, provided as a vector (e.g., c("A", "B")), whose levels you wish to compare pairwise. Names of Fixed_Factor should match Fixed_Factor used to fit the model. When more than one factor is provided e.g. `Fixed_factor = c("A", "B")`, this function passes this on as `specs = A:B` (note the colon between the two Fixed_Factor) to `emmeans` to produce pairwise comparisons.
`Trend_Factor`	a quantitative variable that interacts with a factor and whose slope (trend) is to be compared
`P_Adj`	method for correcting P values for multiple comparisons. Default is "sidak", can be changed to "bonferroni". See Interaction analysis in emmeans in the manual for `emmeans`.
`...`	additional arguments for `emmeans` such as `lmer.df` or others. See help for sophisticated models in emmeans.

Details

Checkout the Interactions with covariates section in the emmeans vignette for more details. One of the independent variables should be a quantitative (e.g. time points) variable whose slope (trend) you want to find at levels of the other factor.

Value

returns an "emm_list" object containing slopes and their contrasts calculated through `emtrends`.

Examples

```
#create an lm model
#Time2 is numeric (time points)
m1 <- simple_model(data = data_2w_Tdeath,
Y_value = "PI", Fixed_Factor = c("Genotype", "Time2"))
posthoc_Trends_Pairwise(Model = m1,
Fixed_Factor = "Genotype",
Trend_Factor = "Time2")
```

---

posthoc_Trends_vsRef *Use emtrends to get level-wise comparison of slopes from a linear model.*

Description

This function is a wrapper based on `emmeans`, and needs a ordinary linear model produced by `simple_model` or a mixed effects model produced by `mixed_model` or `mixed_model_slopes` (or generated directly with `lm`, `lme4` or `lmerTest` calls). At least one of the factors should be a numeric covariate whose slopes you wish to find. It also needs to know the fixed factor(s), which should match those in the model and data table.

Usage

```
posthoc_Trends_vsRef(
  Model,
  Fixed_Factor,
  Trend_Factor,
  Ref_Level = 1,
  P_Adj = "sidak",
  ...
)
```

Arguments

Model	a model object fit using simple_model or mixed_model (or lm or lmer).
Fixed_Factor	one or more categorical variables, provided as a vector (e.g., c("A", "B")), whose levels you wish to compare pairwise. Names of Fixed_Factor should match Fixed_Factor used to fit the model. When more than one factor is provided e.g. Fixed_factor = c("A", "B"), this function passes this on as specs = A:B (note the colon between the two Fixed_Factor) to emmeans to produce pairwise comparisons.
Trend_Factor	a quantitative variable that interacts with a factor and whose slope (trend) is to be compared
Ref_Level	the level within that factor to be considered the reference or control to compare other levels to (to be provided as a number - by default R orders levels alphabetically); default Ref_Level = 1.
P_Adj	method for correcting P values for multiple comparisons. Default is "sidak", can be changed to "bonferroni". See Interaction analysis in emmeans in the manual for emmeans.
...	additional arguments for emmeans such as lmer.df or others. See help for sophisticated models in emmeans.

Details

Checkout the Interactions with covariates section in the emmeans vignette for more details. One of the independent variables should be a quantitative (e.g. time points) variable whose slope (trend) you want to find at levels of the other factor.

Value

returns an "emm_list" object containing slopes and their contrasts calculated through emtrends.

Examples

```
#create an lm model
#Time2 is numeric (time points)
m1 <- simple_model(data = data_2w_Tdeath,
Y_value = "PI", Fixed_Factor = c("Genotype", "Time2"))
posthoc_Trends_vsRef(Model = m1,
Fixed_Factor = "Genotype",
Trend_Factor = "Time2",
Ref_Level = 2)
```

posthoc_vsRef *Post-hoc comparisons to a control or reference group.*

Description

This function is a wrapper based on emmeans, and needs a ordinary linear model produced by simple_model or a mixed effects model produced by mixed_model or mixed_model_slopes (or generated directly with lm, lme4 or lmerTest calls). It also needs to know the fixed factor(s), which should match those in the model and data table.

Usage

```
posthoc_vsRef(Model, Fixed_Factor, Ref_Level = 1, P_Adj = "fdr", Factor, ...)
```

Arguments

Model	a model object fit using simple_model or mixed_model or related.
Fixed_Factor	Fixed_Factor one or more categorical variables, provided as a vector (e.g., c("A", "B")), whose levels you wish to compare pairwise. Names of Fixed_Factor should match Fixed_Factor used to fit the model. When more than one factor is provided e.g. Fixed_factor = c("A", "B"), this function passes this on as specs = A|B (note the vertical | between the two Fixed_Factor) to emmeans. The specification internally is set to specs = trt.vs.ctrl, Ref_Level = 1 to compare each group in A to the reference first group in A, separately at each level of B.
Ref_Level	the level within that factor to be considered the reference or control to compare other levels to (to be provided as a number - by default R orders levels alphabetically); default Ref_Level = 1.
P_Adj	method for correcting P values for multiple comparisons. Default is set to false discovery rate ("fdr"), can be changed to "none", "tukey", "bonferroni", "sidak" (in quotes). See Interaction analysis in emmeans in the manual for emmeans.
Factor	old argument name for Fixed_Factor; retained for backward compatibility.
...	additional arguments for emmeans such as lmer.df or others. See help for sophisticated models in emmeans.

Details

The function will generate treatment vs control type of comparisons (as described in Comparisons and contrasts in emmeans), i.e. comparison of each level of a factor to a reference level, which is set by default to the first level in the factor (Ref_Level = 1). By default, P values are corrected by the FDR method (which can be changed). If the model was fit by transforming the quantitative response variable using "log", "logit", "sqrt" etc., results will still be on the original scale, i.e. type = "response" is the default; data will be back-transformed (check results to confirm this), and for log or logit see Transformations and link functions in emmeans, ratios will be compared. The first part of the emmeans results has the estimated marginal means, SE and CI ($emmeans), which are generated from the fitted model, and **not** the original data table. The second part has the results of the comparisons ($contrasts).

Value

returns an "emm_list" object containing contrasts and emmeans through emmeans.

Examples

```
#make linear models first
DoublMod <- simple_model(data = data_doubling_time,
Y_value =  "Doubling_time",
Fixed_Factor =  "Student")

CholMod <- mixed_model(data = data_cholesterol,
Y_value =  "Cholesterol",
Fixed_Factor = c("Hospital", "Treatment"),
Random_Factor = "Subject")

#to compare all students with student #9
posthoc_vsRef(Model = DoublMod,
Fixed_Factor = "Student", Ref_Level = 9)

#for comparison between hospital_a to every other hospital, separately at levels of Treatment
posthoc_vsRef(Model = CholMod,
Fixed_Factor = c("Hospital", "Treatment"), Ref_Level = 1)
```

---

scale_colour_grafify scale_colour_ *and* scale_fill_ *functions*

Description

These let you apply grafify discrete or continuous palettes as fill or colour aesthetics to any ggplot2 (scale_color_ spelling is also accepted).

Usage

```
scale_colour_grafify(
  palette = "okabe_ito",
  ColSeq = TRUE,
  reverse = FALSE,
  discrete = TRUE,
  ...
)

scale_color_grafify(
  palette = "okabe_ito",
  ColSeq = TRUE,
  reverse = FALSE,
  discrete = TRUE,
  ...
)
```

Arguments

palette	Name of the grafify palettes from above, provide within quotes, e.g., palette = "vibrant". Default discrete palette is okabe_ito. For quantitative palette, set discrete = FALSE (which will apply blue_conti unless another palette is chosen).
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours.
reverse	Whether the colour order should be reversed.
discrete	not used.
...	Additional parameters for scale_fill or scale_colour.

Details

The default is palette = "okabe_ito". The discrete argument is not used at present. The following discrete and quantitative palettes can be used.

Categorical/discreet palettes:

- okabe_ito (default)
- bright
- contrast
- dark
- kelly
- light
- muted
- pale
- r4
- safe
- vibrant

By default, sequential colours from above palettes will be chosen. To choose the most distant colours set ColSeq = TRUE.

Sequential quantitative palettes:

- grey_conti
- blue_conti
- yellow_conti

Divergent quantitative palettes:

- OrBl_div
- PrGn_div

Value

ggplot scale_fill function for discrete colours.

Examples

```
#add a grafify fill scheme to ggplot
ggplot(emmeans::neuralgia, aes(x = Treatment,
                               y = Duration))+
  geom_boxplot(aes(fill = Treatment),
               alpha = .6)+
  geom_point(aes(colour = Treatment,
                 shape = Treatment),
             size = 3)+
  scale_fill_grafify(palette = "bright")+
  scale_colour_grafify(palette = "bright")+
  facet_wrap("Sex")+
  theme_classic()
#distant colours `ColSeq = FALSE`
ggplot(emmeans::neuralgia, aes(x = Treatment,
                               y = Duration))+
  geom_boxplot(aes(fill = Treatment),
               alpha = .6)+
  geom_point(aes(colour = Treatment,
                 shape = Treatment),
             size = 3)+
  scale_fill_grafify(palette = "bright",
                     ColSeq = FALSE)+
  scale_colour_grafify(palette = "bright",
                       ColSeq = FALSE)+
  facet_wrap("Sex")+
  theme_classic()
#quantitative colour scheme
ggplot(mtcars, aes(x = disp,
                   y = mpg))+
  geom_point(aes(colour = cyl),
             size = 3)+
  scale_colour_grafify(palette = "blue_conti")
```

---

scale_fill_grafify scale_colour_ *and* scale_fill_ *functions*

Description

These let you apply grafify discrete or continuous palettes as fill or colour aesthetics to any ggplot2 (scale_color_ spelling is also accepted).

Usage

```
scale_fill_grafify(
  palette = "okabe_ito",
  ColSeq = TRUE,
  reverse = FALSE,
  discrete = TRUE,
```

```
  ...
)
```

Arguments

palette	Name of the grafify palettes from above, provide within quotes, e.g., palette = "vibrant". Default discrete palette is okabe_ito. For quantitative palette, set discrete = FALSE (which will apply blue_conti unless another palette is chosen).
ColSeq	logical TRUE or FALSE. Default TRUE for sequential colours from chosen palette. Set to FALSE for distant colours.
reverse	Whether the colour order should be reversed.
discrete	not used.
...	Additional parameters for scale_fill or scale_colour.

Details

The default is palette = "okabe_ito". The discrete argument is not used at present. The following discrete and quantitative palettes can be used.

Categorical/discreet palettes:

- okabe_ito (default)
- bright
- contrast
- dark
- kelly
- light
- muted
- pale
- r4
- safe
- vibrant

By default, sequential colours from above palettes will be chosen. To choose the most distant colours set ColSeq = TRUE.

Sequential quantitative palettes:

- grey_conti
- blue_conti
- yellow_conti

Divergent quantitative palettes:

- OrBl_div
- PrGn_div

Value

ggplot scale_fill function for discrete colours.

Examples

```
#add a grafify fill scheme to ggplot
ggplot(emmeans::neuralgia, aes(x = Treatment,
                               y = Duration))+
  geom_boxplot(aes(fill = Treatment),
               alpha = .6)+
  geom_point(aes(colour = Treatment,
                 shape = Treatment),
             size = 3)+
  scale_fill_grafify(palette = "bright")+
  scale_colour_grafify(palette = "bright")+
  facet_wrap("Sex")+
  theme_classic()
#distant colours `ColSeq = FALSE`
ggplot(emmeans::neuralgia, aes(x = Treatment,
                               y = Duration))+
  geom_boxplot(aes(fill = Treatment),
               alpha = .6)+
  geom_point(aes(colour = Treatment,
                 shape = Treatment),
             size = 3)+
  scale_fill_grafify(palette = "bright",
                     ColSeq = FALSE)+
  scale_colour_grafify(palette = "bright",
                       ColSeq = FALSE)+
  facet_wrap("Sex")+
  theme_classic()
#quantitative colour schemes
ggplot(mtcars, aes(x = disp,
                   y = mpg))+
  geom_point(aes(colour = cyl),
             size = 3)+
  scale_colour_grafify(palette = "blue_conti")
```

`simple_anova` *ANOVA table from a linear model fit to data.*

Description

One of two functions for simple ANOVA tables and linear models without random effects, which use `lm` to fit a linear models.

1. `link{simple_anova}`
2. `link{simple_model}`

Usage

```
simple_anova(data, Y_value, Fixed_Factor, ...)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
Y_value	name of column containing quantitative (dependent) variable, provided within "quotes". The following transformations are permitted: "log(Y_value)", "log(Y_value + c)" where c a positive number, "logit(Y_value)" or "logit(Y_value/100)" which may be useful when Y_value are percentages (note quotes outside the log or logit calls); "sqrt(Y_value)" or "(Y_value)^2" should also work. During posthoc-comparisons, log and logit transformations will be back-transformed to the original scale. Other transformations, e.g., "sqrt(Y_value)" will not be back-transformed. Check out the regrid and ref_grid for details if you need back-transformation to the response scale.
Fixed_Factor	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). If a numeric variable(s) is used, transformations similar to Y_value are permitted.
...	any additional argument to pass on to lm if required.

Details

Update in v0.2.1: This function uses lm to fit a linear model to data, passes it on to Anova, and outputs the ANOVA table with type II sum of squares with F statistics and *P* values. (Previous versions produced type I sum of squares using anova call.)

It requires a data table, one quantitative dependent variable and one or more independent variables. If your experiment design has random factors, use the related function mixed_anova.

This function is related to link{simple_model}.

Value

ANOVA table of class "anova" and "data.frame".

Examples

```
#Basic usage
simple_anova(data = data_doubling_time,
Y_value = "Doubling_time",
Fixed_Factor = "Student")
```

simple_model *Model from a linear model fit to data.*

Description

One of two functions for simple ANOVA tables and linear models without random effects, which use lm to fit a linear models.

1. link{simple_anova}
2. link{simple_model}

Usage

```
simple_model(data, Y_value, Fixed_Factor, ...)
```

Arguments

data	a data table object, e.g. data.frame or tibble.
Y_value	name of column containing quantitative (dependent) variable, provided within "quotes". The following transformations are permitted: "log(Y_value)", "log(Y_value + c)" where c a positive number, "logit(Y_value)" or "logit(Y_value/100)" which may be useful when Y_value are percentages (note quotes outside the log or logit calls); "sqrt(Y_value)" or "(Y_value)^2" should also work. During posthoc-comparisons, log and logit transformations will be back-transformed to the original scale. Other transformations, e.g., "sqrt(Y_value)" will not be back-transformed. Check out the regrid and ref_grid for details if you need back-transformation to the response scale.
Fixed_Factor	name(s) of categorical fixed factors (independent variables) provided within quotes (e.g., "A") or as a vector if more than one (e.g., c("A", "B"). If a numeric variable(s) is used, transformations similar to Y_value are permitted.
...	any additional arguments to pass on to lm if required.

Details

Update in v0.2.1: This function uses lm to fit a linear model to data, passes it on to Anova, and outputs the ANOVA table with type II sum of squares with F statistics and *P* values.

(Previous versions produced type I sum of squares using anova call.) It requires a data table, one quantitative dependent variable and one or more independent variables.

The model output can be used to extract coefficients and other information, including post-hoc comparisons. If your experiment design has random factors, use the related function mixed_model.

This function is related to link{simple_anova}. Output of this function can be used with posthoc_Pairwise, posthoc_Levelwise and posthoc_vsRef, or with emmeans.

Value

This function returns an object of class "lm".

Examples

```
#fixed factors provided as a vector
Doubmodel <- simple_model(data = data_doubling_time,
Y_value =  "Doubling_time",
Fixed_Factor = "Student")
#get summary
summary(Doubmodel)
```

`table_summary` *Get numeric summary grouped by factors*

Description

This is a wrapper around `aggregate` function in base R to obtain mean, median, standard deviation and count for quantitative variable(s) grouped by one or more factors. More than one column containing of quantitative variables can be passed on, and summaries for each is provided with column names with a ..

Usage

```
table_summary(data, Ycol, ByGroup)
```

Arguments

`data`	name of the data table.
`Ycol`	name of one column (in quotes) or a vector of column names (e.g., c("Y1", "Y2")) containing the numerical variable to be summarised.
`ByGroup`	name of one column (in quotes) or a vector of column names (e.g., c("A", "B")) containing the grouping factors

Value

this function takes in a data.frame or tibble and returns a data.frame or tibble.

Examples

```
table_summary(Ycol = "cty",
ByGroup = c("fl", "drv"),
data = mpg)
```

table_x_reorder *Reordering groups along X-axis*

Description

This simple function takes in a data table and reorders groups (categorical variables or factors) to be plotted along the X-axis in a user-defined order.

Usage

```
table_x_reorder(data, xcol, OrderX, ...)
```

Arguments

data	a data table or tibble.
xcol	name of column in above data table (in quotes), e.g., "A", whose levels are to be reordered.
OrderX	a vector of group names within the column selected in xcol in the desired order, .e.g., c("D", "A", "C").
...	any additional arguments for factor call.

Details

It uses two base R functions: as.factor to first force the user-selected column into a factor, and factor that reorders levels based on a user-provided vector.

Value

This function returns a data frame with a selected column converted into factor with reordered levels.

Examples

```
#reorder levels within Genotype
new_data <- table_x_reorder(data_t_pratio,
xcol = "Genotype",
OrderX = c("KO", "WT"))
#compare
plot_scatterbox(data_t_pratio,
Genotype,
Cytokine)
#with
plot_scatterbox(new_data,
Genotype,
Cytokine)
#also works within the plot call
plot_scatterbox(data = table_x_reorder(data_t_pratio,
xcol = "Genotype",
```

```
  OrderX = c("KO", "WT")),
  xcol = Genotype,
  ycol = Cytokine)
```

`theme_grafify` *A modified* `theme_classic()` *for* `grafify`*-like graphs.*

Description

This is a slightly modified `theme_classic` with two key differences: no border & background for facet panel labels, and font size of text on axes is the same as that of the axes titles (prior to v3.2.0, this was 0.85 times the base font size). The size of text legend title is also same as base font.

Usage

```
theme_grafify(
  base_size = 20,
  base_family = "",
  base_line_size = base_size/22,
  base_rect_size = base_size/22,
  TextXAngle = 0,
  vjust = 0,
  hjust = 0,
  ...
)
```

Arguments

`base_size`	base font size for all text (default is 20). Other text is relative to this.
`base_family`	default font family
`base_line_size`	default line size (default is base font size/22)
`base_rect_size`	default size of rectangles (default is base font size/22)
`TextXAngle`	orientation of text on X-axis; default 0 degrees. Change to 45 or 90 to remove overlapping text.
`vjust`	vertical adjustment of X-axis text alignment (between 0 and 1). Set `hjust` and `vjust` to 1 if `TextXAngle = 45`. Try other options if using other angles.
`hjust`	horizontal adjustment of X-axis text alignment (between 0 and 1). Set `hjust` and `vjust` to 1 if `TextXAngle = 45`. Try other options if using other angles.
`...`	for any other arguments to pass to `theme`. A useful one is `aspect.ratio = 1` for square plots.

Details

Since v3.2.0, `theme_grafify` produces transparent backgrounds.

Value

this returns an output with class "theme" and "gg".

Examples

```
ggplot(mpg, aes(drv, cty, colour = fl))+
geom_jitter(width = 0.2,
size = 3, alpha = .7)+
theme_grafify()
```

Index

∗ **datasets**
data_1w_death, 3
data_2w_Festing, 4
data_2w_Tdeath, 4
data_cholesterol, 5
data_doubling_time, 6
data_t_pdiff, 6
data_t_pratio, 7
data_zooplankton, 7
graf_colours, 11
graf_palettes, 13

aggregate, *116*
Anova, *114, 115*
anova, *114, 115*
appraise, *83*
as_lmerModLmerTest, *19*, *21*, *23*, *25*

colorRampPalette, *12*, *30*, *33*, *35*, *38*, *41*, *44*, *47*, *50*, *53*, *56*, *59*, *61*, *64*, *67*, *69*, *74*, *79*, *81*, *86*, *88*, *91*, *94*, *98*

data_1w_death, 3
data_2w_Festing, 4
data_2w_Tdeath, 4
data_cholesterol, 5
data_doubling_time, 6
data_t_pdiff, 6
data_t_pratio, 7
data_zooplankton, 7

emmeans, *20*, *22*, *24*, *27*, *101–109*, *115*
emtrends, *104*, *106, 107*

facet_wrap, *28*, *31*, *34*, *37*, *40*, *43*, *46*, *49*, *52, 53*, *55*, *58*, *60*, *63*, *66*, *68*, *73*, *78*, *81*, *85*, *87*, *90*, *93*, *99*

ga_anova, *8*, 8, *9*
ga_model, *8, 9*, 9, *83*

geom_boxplot, *30*, *32*, *35*, *38*, *41*, *44*, *47*, *50*, *66*, *69*, *88*, *91*
geom_density, *60, 61*
geom_dotplot, *62–69*
geom_histogram, *72–74*
geom_line, *52, 53*, *55, 56*, *58, 59*
geom_point, *41*, *44*, *53*, *56*, *59*, *64*, *66*, *69*, *77*, *84*, *86*, *89–93*, *98, 99*
geom_qq, *81, 82*
geom_qq_line, *81, 82*
geom_violin, *38*, *50*, *69*, *91*
get_graf_colours, 10
graf_col_palette, 12
graf_col_palette_default, 12
graf_colours, 11
graf_palettes, 13, *29*, *32*, *35*, *38*, *41*, *44*, *47*, *50*, *53*, *56*, *59*, *61*, *64*, *66*, *69*, *73*, *79*, *81*, *85*, *88*, *91*, *94*, *100*

lm, *104, 105*, *107*, *113–115*
lmer, *19–26*, *104, 105*, *107*

make_1way_data, *14*, 14, *15, 16*, *18*
make_1way_rb_data, *14, 15*, 15, *16*, *18*
make_2way_data, *14–16*, 16, *18*
make_2way_rb_data, *14–16*, *18*, 18
mixed_anova, 19, *22*, *26*, *114*
mixed_anova_slopes, 21
mixed_model, *22*, 23, *26*, *82*, *101, 102*, *104–108*, *115*
mixed_model_slopes, 25, *82*, *101, 102*, *104–106*, *108*

plot_3d_point_sd, *27*, 27, *30*, *33*, *36*
plot_3d_scatterbar, *27*, *30*, 30, *33*, *36*, *94*, *100*
plot_3d_scatterbox, *27*, *30*, *33*, 33, *36*, *94*, *100*
plot_3d_scatterviolin, *27*, *30*, *33*, *36*, 36
plot_4d_point_sd, *39*, 39, *42*, *45*, *48*

plot_4d_scatterbar, *39*, *42*, 42, *45*, *48*
plot_4d_scatterbox, *39*, *42*, *45*, 45, *48*, *94*, *100*
plot_4d_scatterviolin, *39*, *42*, *45*, *48*, 48
plot_befafter_box, *51*, 51, *54*, *57*
plot_befafter_colors, *51*, *54*, *57*
plot_befafter_colors (plot_befafter_colours), 54
plot_befafter_colours, *51*, *53*, *54*, 54, *56*, *57*, *59*
plot_befafter_shapes, *51*, *53*, *54*, *56*, *57*, 57, *59*
plot_density, 60, *72*
plot_dotbar_sd, *62*, 62, *65*, *67*
plot_dotbox, *62*, *65*, 65, *67*
plot_dotviolin, *62*, *65*, *67*, 67
plot_gam_predict, 70
plot_grafify_palette, *61*, *64*, *67*, *69*, 71, *74*, *79*, *81*, *86*, *88*, *91*, *94*, *98*
plot_histogram, *60*, 72, *80*
plot_lm_predict, 74
plot_logscale, 76
plot_point_sd, *77*, 77, *79*, *84*, *86*, *88*, *89*, *91*
plot_qq_gam, 83
plot_qqline, *60*, *72*, *80*, 80
plot_qqmodel, 82
plot_scatterbar_sd, *64*, *66*, *69*, *72*, *77*, *79*, *84*, 84, *86*, *88*, *89*, *91*
plot_scatterbox, *64*, *66*, *69*, *77*, *84*, *86*, 86, *89*
plot_scatterviolin, *64*, *66*, *69*, *77*, *84*, *86*, *89*, 89
plot_xy_CatGroup, *71*, 92, *95*, *100*
plot_xy_Group, 95
plot_xy_NumGroup, *94*, *95*, 98
posthoc_Levelwise, *20*, *22*, *24*, *27*, 101, *115*
posthoc_Pairwise, *20*, *22*, *24*, *27*, 102, *115*
posthoc_Trends_Levelwise, *20*, *22*, *24*, *27*, 103
posthoc_Trends_Pairwise, *20*, *22*, *24*, *27*, 105
posthoc_Trends_vsRef, *20*, *22*, *24*, *27*, 106
posthoc_vsRef, *20*, *22*, *24*, *27*, 108, *115*

ref_grid, *19*, *22*, *24*, *26*, *114*, *115*
regrid, *19*, *22*, *24*, *26*, *114*, *115*
rstudent, *82*

scale_color_grafify (scale_colour_grafify), 109
scale_colour_grafify, 109
scale_fill_grafify, 111
scale_x_continuous, *76*, *99*, *100*
scale_y_continuous, *29*, *32*, *34*, *37*, *40*, *41*, *43*, *44*, *46*, *50*, *52*, *56*, *58*, *61*, *63*, *66*, *69*, *73*, *76*, *79*, *85*, *88*, *90*, *99*
simple_anova, 113
simple_model, *82*, *101*, *102*, *104–108*, 115
stat_qq, *80*, *81*
stat_qq_line, *80*, *81*
stat_smooth, *98*
stat_summary, *41*, *44*, *79*

table_summary, *20*, *22*, *24*, *26*, 116
table_x_reorder, 117
theme_classic, *29*, *32*, *34*, *37*, *41*, *44*, *47*, *50*, *53*, *56*, *58*, *61*, *63*, *66*, *69*, *73*, *79*, *81*, *82*, *85*, *88*, *91*, *94*, *97*, *100*, *118*
theme_grafify, 118