Package: pastaPlot (via r-universe)

November 4, 2024

Type Package

Title Spaghetti-Plot Fixed and Random Effects of Linear Mixed Models

Version 0.1.0

Description Plot both fixed and random effects of linear mixed models, multilevel models in a single spaghetti plot. The package allows to visualize the effect of a predictor on a criterion between different levels of a grouping variable. Additionally, confidence intervals can be displayed for fixed effects. Calculation of predicted values of random effects allows only models with one random intercept and/or one random slope to be plotted. Confidence intervals and predicted values of fixed effects are computed using the 'ggpredict' function from the 'ggeffects' package. Lüdecke, D. (2018) <doi:10.21105/joss.00638>.

Encoding UTF-8

LazyData true

Imports ggeffects, ggplot2, glmmTMB, lme4

RoxygenNote 7.3.1

Depends R (>= 2.10)

License MIT + file LICENSE

Repository https://janpalnau.r-universe.dev

RemoteUrl https://github.com/janpalnau/pastaplot

RemoteRef HEAD

RemoteSha 65582f119a03e09c1577950c095df7fedc7b6919

Contents

cookPasta *Spaghetti-plot fixed and random effects of linear mixed models*

Description

cookPasta() creates dataframes from fixed and random effects of 'lme4' or 'glmmTMB' models (e.g., for plotting)

Usage

```
cookPasta(
  model = NULL,
  predictor = NULL,
  nested.in = NULL,
  group = NULL,
  ci.int = FALSE,
  ci.lvl = ci.lvl
)
```

Arguments

model	lme4 or glmmTMB model object
predictor	(Character) Name of predictor (e.g., "time" or "math_score"), as it is present in the model
nested.in	(Character) Name of the variable your time points or subjects are nested in (e.g.,"school" or "id")
group	(Optional, character) The name of your grouping variable (e.g., "condition" or "gender")
ci.int	(Optional, boolean) Enable confidence (prediction) intervals, disabled by default
ci.lvl	(Optional, numeric) Set level of confidence (prediction) intervals (default: 0.95). Requires ci.int to be set to TRUE

Value

Returns a list of two dataframes, in which the first element is the fixed effects dataframe and the second element the random effects dataframe

ecovia_data *ECOVIA data*

Description

A subset of data from the ECOVIA app

Usage

```
ecovia_data
```

Format

A data frame with 1743 rows and 4 columns:

id Subjects, in which time points are nested in

time Day of intervention

condition Condition (control vs. intervention)

CO2 Daily dietary carbon emissions

Source

<https://osf.io/qd7vw/?view_only=cc22e0d1de8844e1850d8ef1442fbecb>

jsp_data *JSP data*

Description

A subset of JSP secondary school exam data

Usage

```
jsp_data
```

Format

A data frame with 728 rows and 7 columns:

math_score_y1 Score on math test year 1

math_score_y3 Score on math test year 3

gender Gender of subjects (pupils)

social_class Social class of subjects (pupils)

school School, in which subjects (pupils) are nested in

normal_score_y1 Normal test score year 1

normal_score_y3 Normal test score year 3

Source

<https://www.bristol.ac.uk/cmm/team/hg/msm-3rd-ed/datasets.html>

`pastaPlot` *Spaghetti-plot fixed and random effects of linear mixed models*

Description

`pastaPlot()` plots slopes for both fixed and random effects of linear mixed models from 'lme4' or 'glmmTMB' packages as a single spaghetti plot, optionally between conditions including confidence bands for fixed effects.

Usage

```
pastaPlot(
  model = NULL,
  predictor = NULL,
  nested.in = NULL,
  group = NULL,
  legend.title = "Legend",
  group.labels = NULL,
  xlab = NULL,
  ylab = NULL,
  font.family = NULL,
  colors = NULL,
  ci.lvl = 0.95,
  ci.int = FALSE,
  ci.linetype = 0,
  lwd.fix = 1,
  lwd.ran = 0.5,
  xlab.inc = 0,
  xlab.int = NULL,
  ylim = NULL,
  opacity.ci = 0.25,
  opacity.ran = 0.3,
  colors.ci = NULL
)
```

Arguments

`model`	lme4 or glmmTMB model object
`predictor`	(Character) Name of predictor (e.g., "time" or "math_score"), as it is present in the model
`nested.in`	(Character) Name of the variable your time points or subjects are nested in (e.g.,"school" or "id")

group	(Optional, character) The name of your grouping variable (e.g., "condition" or "gender")
legend.title	(Optional, character) Name of legend in plot (e.g., "Condition", or "Gender")
group.labels	(Optional, vector of characters) Names of group labels to be displayed in the plot (e.g., c("Control", "Intervention"))
xlab	(Optional, character) Label of x-axis (predictor) (e.g., "Time (days)")
ylab	(Optional, character) Label of y-axis (dependant variable) (e.g., "GAF")
font.family	(Optional, character) Name of the font family (e.g. "serif")
colors	(Optional, vector of characters) Set color of slopes. Length of vector should correspond to number of values in group variable (e.g., c("#5e9aff", "blue")). If no group variable is specified, pass a single color.
ci.lvl	(Optional, numeric) Set confidence interval (default: 0.95)
ci.int	(Optional, logical) Enable confidence (prediction) intervals, disabled by default
ci.linetype	(Optional, numeric) Set linetype of confidence bands outline (default: 0)
lwd.fix	(Optional, numeric) Line width of fixed effects (default: 1)
lwd.ran	(Optional, numeric) Line width of random effects (default: 0.5)
xlab.inc	(Optional, numeric) Increment the displayed values of your predictor (e.g., xlab_int = 1 changes range of x from 0-29 to 1-30), set to 0 by default
xlab.int	(Optional, numeric) Interval between displayed predictor values on x-axis (e.g., "1"), disabled by default
ylim	(Optional, numeric vector) Limited range of values on y-axis (e.g. c(1,5.5))
opacity.ci	(Optional, numeric) Set opacity of confidence bands in the range of 0 to 1 (default = 0.1)
opacity.ran	(Optional, numeric) Set opacity of random slopes in the range of 0 to 1 (default = 0.4)
colors.ci	(Optional, vector of characters) Set color of confidence bands. Length of vector should correspond to number of values in group variable (e.g., c("#5e9aff", "blue")). If no group variable is specified, pass a single color.

Value

Returns a ggplot2 plot object to further be modified

Examples

```
lme4_model <- lme4::lmer(CO2 ~ 1 + time*condition + (1 + time | id),
data=ecovia_data, REML = FALSE, control = lme4::lmerControl(optimizer = "bobyqa"))
pastaPlot(lme4_model, "time", "id", group = "condition", legend.title = "Condition",
group.labels = c("Control", "Intervention"), ci.int = TRUE, xlab = "Time (days)",
ylab = "CO2")

glmmTMB_model <- glmmTMB::glmmTMB(math_score_y3 ~ 1 + math_score_y1*gender +
(1 + math_score_y1 | school), data=jsp_data, REML = FALSE)
pastaPlot(glmmTMB_model, "math_score_y1", "school", group = "gender",
legend.title = "Gender", group.labels = c("Male", "Female"), ci.int = FALSE,
xlab = "Math score (year 1)", ylab = "Math score (year 3)")
```

Index