Package ‘tidygam’

May 9, 2023

Type Package

Title Tidy Prediction and Plotting of Generalised Additive Models

Version 0.2.0

Date 2023-05-08

Description Provides functions that compute predictions from Generalised Additive Models (GAMs) fitted with 'mgcv' and return them as a tibble. These can be plotted with a generic plot()-method that uses 'ggplot2' or plotted as any other data frame. The main function is predict_gam().

License MIT + file LICENSE

URL https://github.com/stefanocoretta/tidygam, https://stefanocoretta.github.io/tidygam/

BugReports https://github.com/stefanocoretta/tidygam/issues

Encoding UTF-8

LazyData true

Imports cli, dplyr, ggplot2, glue, insight, magrittr, mgcv, rlang, stringr, tibble, tidyr, tidyselect

Suggests knitr, rmarkdown

VignetteBuilder knitr

Language en_GB

RoxygenNote 7.2.0

Depends R (>= 2.10)

NeedsCompilation no

Author Stefano Coretta [aut, cre]

Maintainer Stefano Coretta <stefano.coretta@gmail.com>

Repository CRAN

Date/Publication 2023-05-09 08:50:02 UTC

R topics documented:

gest — *Number of gestures by infants at 10, 11 and 12 months*

Description

This data table contains counts of three type of gestures performed by 60 infants from Bengali, Chinese and British backgrounds.

Usage

```
gest
```

Format

A tibble with 540 observations and 5 variables:

dyad Unique parent/infant dyad ID.

background Cultural background of dyad.

months Time point in infant months.

gesture Type of gesture.

count Number of gestures.

Source

doi:10.1111/cdev.13406

`get_difference` *Get difference between two smooths*

Description

Get difference between two smooths

Usage

```
get_difference(
  model,
  series,
  compare,
  values = NULL,
  exclude_terms = NULL,
  length_out = 25,
  ci_z = 1.96
)
```

Arguments

`model` A `gam` or `bam` model object.

`series` A string specifying the variable that corresponds to the series to be plotted on the $x$-axis. If a string is given, the other numeric variables in the model are set to their mean value, unless specific values are given in `values`. If a character vector of two strings is given, the two variables will be taken as the elements of a tensor product smooth. This allows the user to plot 2D raster plots.

`compare` A named list of factor levels to compare.

`values` User supplied values for specific variables as a named list.

`exclude_terms` Terms to be excluded from the prediction. Term names should be given as they appear in the model summary (for example, `"s(x0,x1)"`).

`length_out` An integer indicating how many values to use along the numeric variables for predicting the response (the default is `10`).

`ci_z` The z-value for calculating the CIs (the default is `1.96` for 95 percent CI).

Value

A tibble with the difference smooth.

Examples

```
library(mgcv)
set.seed(10)
data <- gamSim(4)
model <- gam(y ~ s(x2, by = fac) + s(x0), data = data)

get_difference(model, "x2", list(fac = c("1", "2")))
```

plot.tidygam *Plot methods for tidygam objects*

Description

Plotting methods for `tidygam` objects.

Usage

```
## S3 method for class 'tidygam'
plot(x, series = NULL, comparison = NULL, raster_interp = FALSE, ...)
```

Arguments

`x` A `tidygam` object (see `predict_gam()`).

`series` A string specifying the variable that corresponds to the series to be plotted on the $x$-axis. If a string is given, the other numeric variables in the model are set to their mean value, unless specific values are given in `values`. If a character vector of two strings is given, the two variables will be taken as the elements of a tensor product smooth. This allows the user to plot 2D raster plots.

`comparison` Name of a categorical predictor to compare as a string.

`raster_interp` Whether to linearly interpolate when plotting a tensor product smooth/interaction. It makes sense only when `series` has two variables. The default is `FALSE`.

`...` Arguments passed to `plot()`.

Value

A `ggplot` object.

Examples

```
library(mgcv)
set.seed(10)
sim_data <- gamSim(4)

model_1 <- gam(y ~ s(x2, by = fac) + s(x0), data = sim_data)

preds_1 <- predict_gam(model_1, length_out = 50, exclude_terms = "s(x0)")
plot(preds_1, "x2")

preds_2 <- predict_gam(model_1, length_out = 100, values = list(x0 = 0))
plot(preds_2, "x2", "fac")
library(ggplot2)
plot(preds_2, "x2", "fac") +
  scale_fill_brewer(type = "qual") +
  scale_color_brewer(type = "qual")
```

```
# Plotting tensor product smooths/interactions
model_2 <- gam(y ~ te(x0, x2, by = fac), data = sim_data)
preds_3 <- predict_gam(model_2)
preds_3 %>% plot(series = c("x0", "x2"), comparison = "fac")
```

plot.tidygam.diff *Plot methods for tidygam.diff objects*

Description

Plotting methods for tidygam.diff objects.

Usage

```
## S3 method for class 'tidygam.diff'
plot(x, ..., sig = TRUE, sig_col = "red", sig_alpha = 0.25)
```

Arguments

x A tidygam.diff object (see get_difference()).

... Arguments passed to plot().

sig Shade the interval(s) where the difference smooth does not include 0 (default is TRUE).

sig_col Colour for the shading (default is "red").

sig_alpha Alpha level for the shading (default is 0.25)

Value

A ggplot object.

Examples

```
library(mgcv)
set.seed(10)
data <- gamSim(4)
model <- gam(y ~ s(x2, by = fac) + s(x0), data = data)

model_diff <- get_difference(model, "x2", list(fac = c("1", "2")))
plot(model_diff)
```

`predict_gam` *Get predictions from a GAM model*

Description

Return predictions from a GAM model generated with mgcv. The output can be plotted with `plot()`.

Usage

```
predict_gam(
  model,
  length_out = 10,
  values = NULL,
  series = NULL,
  exclude_terms = NULL,
  ci_z = 1.96,
  tran_fun = NULL,
  separate = NULL,
  sep_by = "\\."
)
```

Arguments

`model` A `gam` or `bam` model object.

`length_out` An integer indicating how many values to use along the numeric variables for predicting the response (the default is `10`).

`values` User supplied values for specific variables as a named list.

`series` A string specifying the variable that corresponds to the series to be plotted on the $x$-axis. If a string is given, the other numeric variables in the model are set to their mean value, unless specific values are given in `values`. If a character vector of two strings is given, the two variables will be taken as the elements of a tensor product smooth. This allows the user to plot 2D raster plots.

`exclude_terms` Terms to be excluded from the prediction. Term names should be given as they appear in the model summary (for example, `"s(x0,x1)"`).

`ci_z` The z-value for calculating the CIs (the default is `1.96` for 95 percent CI).

`tran_fun` Function to use for transforming the predicted values and CIs.

`separate` Names list of factor interaction variables to be separated.

`sep_by` Character to separate by (the default is `\\.`).

Value

A tibble with predictions.

Examples

```
library(mgcv)
set.seed(10)

sim_data_1 <- gamSim(1, n = 200, scale = 2)
model <- gam(y ~ x0 + s(I(x1^2)) + s(x2) + offset(x3), data = sim_data_1)
predict_gam(model)
predict_gam(model, values = list(x0 = mean(sim_data_1$x0)))
predict_gam(model, series = "x2")
predict_gam(model, exclude_terms = "s(I(x1^2))")

# By-variables
sim_data_2 <- gamSim(4)
model_2 <- gam(y ~ s(x2, by = fac) + s(x0), data = sim_data_2)
predict_gam(model_2)

# Poisson data
sim_data_3 <- sim_data_2
sim_data_3$y <- round(sim_data_2$y) + 20
model_3 <- gam(y ~ s(x2, by = fac), data = sim_data_3, family = poisson)
predict_gam(model_3, length_out = 50)
predict_gam(model_3, length_out = 50, tran_fun = exp)

# Bivariate smooths
model_4 <- gam(y ~ te(x1, x2), data = sim_data_1)
predict_gam(model_4)
```

`struct` *ERP to structural violation in music and language*

Description

This data table contains ERP amplitude data from 39 subjects listening to speech and music.

Usage

```
struct
```

Format

A tibble with 17160 observations and 6 variables:

t Time from stimulus onset in milliseconds.

electrode Electrode number.

voltage Electrode voltage at time t.

stimulus.condition Language vs music.

grammar.condition Structural type (grammatical vs ungrammatical).

Source

doi:10.31234/osf.io/e9w3v

Index