**R topics documented:**

- `create_start_event` .................................................. 2
- `geom_smooth_ci` ...................................................... 3
- `get_gam_predictions` ................................................. 4
- `get_smooths_difference` ........................................... 5
- `inter_df` ............................................................... 6
- `plot_difference` ...................................................... 7
- `plot_smooths` .......................................................... 8
- `pois_df` ................................................................. 9
- `predict_gam` ........................................................... 10

**Index**

---

`create_start_event`  
Create a start event column.

**Description**

Create a new column which marks the beginning of each series in a tibble (for example, time series).

**Usage**

```r
create_start_event(tibble, series_col)
```

**Arguments**

- `tibble`  
  A tibble arranged according to the series.
- `series_col`  
  The name of the column that defines the group of series, as an unquoted expression.

**Value**

A tibble with an extra column that marks the beginning of the series.

**Examples**

```r
library(dplyr)
series_tbl <- tibble(
  time_series = rep(1:5, 3),
  group = rep(c("a", "b", "c"), each = 5)
) %>%
create_start_event(group)
```
Description

It provides a 'geom' for plotting GAM smooths with confidence intervals from the output of `predict_gam`. It inherits the following aesthetics from a call to `ggplot`:

- The term defining the x-axis.
- The fitted values (the `fit` column in the tibble returned by `predict_gam`).
- The standard error of the fit (the `se.fit` column in the tibble returned by `predict_gam`).

Usage

```r
geom_smooth_ci(group = NULL, ci_z = 1.96, ci_alpha = 0.1, data = NULL, ...)
```

Arguments

- `group` The optional grouping factor.
- `ci_z` The z-value for calculating the CIs (the default is 1.96 for 95 percent CI).
- `ci_alpha` Transparency value of CIs (the default is 0.1).
- `data` The data to be displayed in this layer. If NULL, it is inherited.
- `...` Arguments passed to `geom_path()`.

Examples

```r
library(mgcv)
library(ggplot2)
set.seed(10)
data <- gamSim(4)
model <- gam(y ~ fac + s(x2) + s(x2, by = fac), data = data)

# get predictions
p <- predict_gam(model)

# plot smooths and confidence intervals
ggplot(p, aes(x2, fit)) + geom_smooth_ci(fac)
```
get_gam_predictions  

Get predictions from a GAM model.

Description

It returns a tibble with the predictions from a gam or bam object.

Usage

get_gam_predictions(
  model,
  series,
  series_length = 25,
  conditions = NULL,
  exclude_random = TRUE,
  exclude_terms = NULL,
  split = NULL,
  sep = "\.",
  time_series,
  transform = NULL,
  ci_z = 1.96,
  .comparison = NULL
)

Arguments

- **model**: A gam or bam model object.
- **series**: An unquoted expression indicating the model term that defines the series on which smoothing is applied. This is the term that is displayed on the x-axis when plotting.
- **series_length**: An integer indicating how many values along the time series to use for predicting the outcome term.
- **conditions**: A list of quosures with quos specifying the levels to plot from the model terms.
- **exclude_random**: Whether to exclude random smooths (the default is TRUE).
- **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)").
- **split**: Columns to separate as a named list.
- **sep**: Separator between columns (default is ",", which is the default with ). If character, it is interpreted as a regular expression.
- **time_series**: Deprecated, use series instead.
- **transform**: Function used to transform the fitted values (useful for getting plots on the response scale).
- **ci_z**: The z-value for calculating the CIs (the default is 1.96 for 95 percent CI).
- **.comparison**: Internal parameter, passed from plot_smooths().
get_smooths_difference

Value
A tibble with predictions from a gam or bam model.

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

get_smooths_difference
Get difference of smooths from a GAM model

Description
It returns a tibble with difference of the specified levels of a smooth from a gam or bam. The sig_diff column states whether the CI includes 0.

Usage
get_smooths_difference(
  model, 
  series, 
  difference, 
  conditions = NULL, 
  exclude_random = TRUE, 
  series_length = 100, 
  time_series 
)

Arguments
model A gam or bam model object.
series An unquoted expression indicating the model term that defines the series on which smoothing is applied. This is the term that is displayed on the x-axis when plotting.
difference A named list with the levels to compute the difference of.
conditions A named list specifying the levels to plot from the model terms not among series or difference. Notice the difference with plot_smooths, which uses quos.
exclude_random Whether to exclude random smooths (the default is TRUE).
series_length An integer indicating how many values along the time series to use for predicting the outcome term.
time_series Deprecated, use series instead.
**Value**

A tibble.

**Examples**

```r
library(mgcv)
set.seed(10)
data <- gamSim(4)
model <- gam(y ~ fac + s(x2) + s(x2, by = fac) + s(x0), data = data)
get_smooths_difference(model, x2, list(fac = c("1", "2"))

# For details, see vignette
## Not run:
vignette("plot-smooths", package = "tidymv")
## End(Not run)
```

---

**inter_df**

*Dataset with two factors*

**Description**

A dataset with a normal-distributed outcome variable and two factors.

**Usage**

`inter_df`

**Format**

A tibble with 1259 observations and 4 variables.

- `x0` time series
- `y` outcome variable
- `x1` factor with three levels
- `x2` factor with two levels
plot_difference

Plot difference smooth from a GAM.

Description

It plots the difference smooth from a `gam` or `bam`. Significant differences are marked with red areas.

Usage

```r
plot_difference(
  model, series, difference, conditions = NULL,
  exclude_random = TRUE, series_length = 100,
  ci_z = 1.96,
  time_series
)
```

Arguments

- `model`: A `gam` or `bam` model object.
- `series`: An unquoted expression indicating the model term that defines the series on which smoothing is applied. This is the term that is displayed on the x-axis when plotting.
- `difference`: A named list with the levels to compute the difference of.
- `conditions`: A named list specifying the levels to plot from the model terms not among `series` or `difference`. Notice the difference with `plot_smooths`, which uses `quos`.
- `exclude_random`: Whether to exclude random smooths (the default is `TRUE`).
- `series_length`: An integer indicating how many values along the time series to use for predicting the outcome term.
- `ci_z`: The z-value for calculating the CIs (the default is 1.96 for 95 percent CI).
- `time_series`: Deprecated, use `series` instead.

Value

A `ggplot` object.

Examples

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

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

# For details, see vignette
## Not run:
vignette("plot-smooths", package = "tidymv")
## End(Not run)

plot_smooths

Plot GAM smooths.

Description

It plots the smooths from the estimates of a `gam` or `bam` object.

Usage

`plot_smooths(
  model,
  series,
  comparison = NULL,
  facet_terms = NULL,
  conditions = NULL,
  exclude_random = TRUE,
  exclude_terms = NULL,
  series_length = 25,
  split = NULL,
  sep = "\",
  transform = NULL,
  ci_z = 1.96,
  time_series
)

Arguments

- **model**: A `gam` or `bam` model object.
- **series**: An unquoted expression indicating the model term that defines the series on which smoothing is applied. This is the term that is displayed on the x-axis when plotting.
- **comparison**: An unquoted expression indicating the model term for which the comparison will be plotted.
- **facet_terms**: An unquoted formula with the terms used for faceting.
- **conditions**: A list of quosures with `quos` specifying the levels to plot from the model terms not among `series`, `comparison`, or `facet_terms`. 

Whether to exclude random smooths (the default is `TRUE`).

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

An integer indicating how many values along the time series to use for predicting the outcome term.

Columns to separate as a named list.

Separator between columns (default is ".", which is the default with `:`). If character, it is interpreted as a regular expression.

Function used to transform the fitted values (useful for getting plots on the response scale).

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

Deprecated, use `series` instead.

A `ggplot` object.

**Examples**

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

plot_smooths(model, x2, fac)

model <- gam(y ~ s(fac, bs = "re") + s(x2) + s(x2, by = fac) + s(x0), data = data)
plot_smooths(model, x2, fac)

# For details, see vignette
## Not run:
vignette("plot-smooths", package = "tidymv")
## End(Not run)
```

A dataset with a Poisson-distributed outcome variable and a factor.

**Usage**

`pois_df`
Format

A tibble with 2500 observations and 3 variables.

- `y` outcome count variable
- `x` time series
- `fac` factor with two levels

**predict_gam**  
*Get predictions from a GAM model.*

Description

It returns a tibble with the predictions from all the terms in a `gam` or `bam` model.

Usage

```r
predict_gam(
  model,
  exclude_terms = NULL,
  length_out = 50,
  values = NULL,
  type = "link"
)
```

Arguments

- `model` A `gam` or `bam` model object.
- `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 along the numeric predictors to use for predicting the outcome term (the default is 50).
- `values` User supplied values for specific terms as a named list. If the value is NULL, the first value of the term is selected (useful when excluding terms).
- `type` Either "link" or "terms". See Details below.

Details

If you simply want to return a tibble with the predicted values of the response/outcome variable based on all terms (minus excluded smooth terms), set `type = "link"` (the default). Note that if `type = "link"`, parametric terms cannot be excluded from the prediction, due to limitations of `mgcv`. If you want to return a tibble with the predicted values of the response/outcome variable for each term in the model separately, set `type = "terms"`. This type can be helpful if you want more flexibility in plotting.
Value

A tibble with predictions from a \texttt{gam} or \texttt{bam} model.

Examples

```r
## Not run:
library(mgcv)
set.seed(10)
data <- gamSim(4)
model <- gam(y ~ fac + s(x2) + s(x2, by = fac) + s(x0), data = data)

# get predictions
p <- predict_gam(model)

# get predictions excluding x0 (the coefficient of x0 is set to 0);
# setting the value for the excluded term to NULL with the argument 'values'
# reduces computation time
p_2 <- predict_gam(model, exclude_terms = "s(x0)", values = list(x0 = NULL))

# get predictions with chosen values of x0
p_3 <- predict_gam(model, values = list(x0 = c(0.250599, 0.503313, 0.756028)))
## End(Not run)
```
Index

* datasets
  inter_df, 6
  pois_df, 9

bam, 4, 5, 7, 8, 10, 11

create_start_event, 2

gam, 4, 5, 7, 8, 10, 11
geom_smooth_ci, 3
get_gam_predictions, 4
get_smooths_difference, 5

inter_df, 6

plot_difference, 7
plot_smooths, 5, 7, 8
pois_df, 9
predict_gam, 3, 10