

# Package ‘tidymv’

April 21, 2021

**Type** Package

**Title** Tidy Model Visualisation for Generalised Additive Models

**Version** 3.2.1

**Date** 2021-04-21

**Description** Provides functions for visualising generalised additive models and getting predicted values using tidy tools from the 'tidyverse' packages.

**URL** <https://github.com/stefanocoretta/tidymv>

**BugReports** <https://github.com/stefanocoretta/tidymv/issues>

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Depends** R (>= 2.10)

**Imports** dplyr, ggplot2, magrittr, mgcv, rlang, tibble, tidyr

**Suggests** knitr, rmarkdown

**VignetteBuilder** knitr

**Language** en\_GB

**License** GPL (>= 3)

**NeedsCompilation** no

**Author** Stefano Coretta [aut, cre],  
Jacolien van Rij [cph] (Imported functions from itsadug),  
Martijn Wieling [cph] (Imported functions from itsadug)

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

**Repository** CRAN

**Date/Publication** 2021-04-21 09:40:02 UTC

## 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
<b>Index</b>	<b>12</b>

---

create_start_event	<i>Create a start event column.</i>
--------------------	-------------------------------------

---

### Description

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

### Usage

```
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

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

---

geom_smooth_ci	<i>Smooths and confidence intervals.</i>
----------------	--

---

## 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

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

## Arguments

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

## Examples

```
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().

**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**

<code>model</code>	A <a href="#">gam</a> or <a href="#">bam</a> model object.
<code>series</code>	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.
<code>difference</code>	A named list with the levels to compute the difference of.
<code>conditions</code>	A named list specifying the levels to plot from the model terms not among series or difference. Notice the difference with <a href="#">plot_smooths</a> , which uses quos.
<code>exclude_random</code>	Whether to exclude random smooths (the default is TRUE).
<code>series_length</code>	An integer indicating how many values along the time series to use for predicting the outcome term.
<code>time_series</code>	Deprecated, use <code>series</code> instead.

**Value**

A tibble.

**Examples**

```
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	<i>Plot difference smooth from a GAM.</i>
-----------------	---

---

## Description

It plots the difference smooth from a [gam](#) or [bam](#). Significant differences are marked with red areas.

## Usage

```
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 <a href="#">plot_smooths</a> , 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

```
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.



<code>exclude_random</code>	Whether to exclude random smooths (the default is TRUE).
<code>exclude_terms</code>	Terms to be excluded from the prediction. Term names should be given as they appear in the model summary (for example, "s(x0, x1)").
<code>series_length</code>	An integer indicating how many values along the time series to use for predicting the outcome term.
<code>split</code>	Columns to separate as a named list.
<code>sep</code>	Separator between columns (default is "\.", which is the default with ). If character, it is interpreted as a regular expression.
<code>transform</code>	Function used to transform the fitted values (useful for getting plots on the response scale).
<code>ci_z</code>	The z-value for calculating the CIs (the default is 1.96 for 95 percent CI).
<code>time_series</code>	Deprecated, use <code>series</code> instead.

**Value**

A `ggplot` object.

**Examples**

```
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)

# alternative model specification
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)
```

---

pois\_df

*Dataset with a Poisson outcome variable*

---

**Description**

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	<i>Get all predictions from a GAM model.</i>
-------------	--

---

**Description**

It returns a tibble with the predictions from all the terms in a [gam](#) or [bam](#) model.

**Usage**

```
predict_gam(model, exclude_terms = NULL, length_out = 50, values = NULL)
```

**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).

**Value**

A tibble with predictions from a [gam](#) or [bam](#) model.

**Examples**

```
## 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

create\_start\_event, 2

gam, 4, 5, 7, 8, 10

geom\_smooth\_ci, 3

get\_gam\_predictions, 4

get\_smooths\_difference, 5

ggplot, 7, 9

inter\_df, 6

plot\_difference, 7

plot\_smooths, 5, 7, 8

pois\_df, 9

predict\_gam, 3, 10