

Package ‘gratia’

March 13, 2019

Version 0.2-8

Date 2019-03-12

Title Graceful 'ggplot'-Based Graphics and Other Functions for GAMs Fitted Using 'mgcv'

Maintainer Gavin L. Simpson <ucfagls@gmail.com>

Depends R (>= 3.5.0)

Imports mgcv, ggplot2, tibble, dplyr, tidyr, cowplot, grid, mvtnorm, stats, tools, grDevices

Suggests testthat, vdiffr, MASS, scam, datasets

Description Graceful 'ggplot'-based graphics and utility functions for working with generalized additive models (GAMs) fitted using the 'mgcv' package. Provides a reimplementa- tion of the plot() method for GAMs that 'mgcv' provides, as well as 'tidyverse' compatible repre- sentations of estimated smooths.

License MIT + file LICENSE

LazyData true

URL <https://gavinsimpson.github.io/gratia>

BugReports <https://github.com/gavinsimpson/gratia/issues>

RoxygenNote 6.1.1

Encoding UTF-8

NeedsCompilation no

Author Gavin L. Simpson [aut, cre] (<<https://orcid.org/0000-0002-9084-8413>>)

Repository CRAN

Date/Publication 2019-03-12 23:00:02 UTC

R topics documented:

appraise	3
bird_move	4
check_user_select_smooths	4

coef.scam	5
confint.fderiv	6
confint.gam	7
datagen	9
data_class	10
data_slice	10
derivatives	11
draw	13
draw.derivatives	13
draw.evaluated_smooth	14
draw.gam	16
evaluate_smooth	18
factor_var_names	19
fderiv	19
fix_offset	21
get_by_smooth	22
get_smooth	22
get_smooths_by_id	23
is_by_smooth	23
is_factor_var	24
is_mgcv_smooth	24
is_offset	25
n_smooths	25
observed_fitted_plot	26
parametric_terms	27
qq_plot	27
rep_first_factor_value	29
residuals_hist_plot	29
residuals_linpred_plot	30
seq_min_max	31
seq_min_max_eps	31
shift_values	32
simulate.gam	32
smallAges	34
smooths	34
smooth_dim	35
which_smooths	35
zooplankton	36

appraise	<i>Model diagnostic plots</i>
----------	-------------------------------

Description

Model diagnostic plots

Usage

```
appraise(model, method = c("direct", "simulate", "normal"),
         n_uniform = 10, n_simulate = 50, type = c("deviance", "pearson",
         "response"), n_bins = c("sturges", "scott", "fd"), ncol = 2,
         level = 0.9, alpha = 0.2, ...)
```

Arguments

model	a fitted model. Currently only class "gam".
method	character; method used to generate theoretical quantiles.
n_uniform	numeric; number of times to randomize uniform quantiles in the direct computation method (method = "direct") for QQ plots.
n_simulate	numeric; number of data sets to simulate from the estimated model when using the simulation method (method = "simulate") for QQ plots.
type	character; type of residuals to use. Only "deviance", "response", and "pearson" residuals are allowed.
n_bins	character or numeric; either the number of bins or a string indicating how to calculate the number of bins.
ncol	numeric; number of columns to draw plots in. See <code>cowplot::plot_grid()</code> .
level	numeric; the coverage level for QQ plot reference intervals. Must be strictly $0 < \text{level} < 1$. Only used with method = "simulate".
alpha	numeric; the level of alpha transparency for the QQ plot reference interval when method = "simulate".
...	arguments passed to <code>cowplot::plot_grid()</code> , except for align and axis, which are set internally.

See Also

The plots are produced by functions `gratia::qq_plot()`, `gratia::residuals_linpred_plot()`, `gratia::residuals_hist_plot()`, and `gratia::observed_fitted_plot()`.

Examples

```
library(mgcv)

## simulate some data...
dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
mod <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat)
## run some basic model checks, including checking
## smoothing basis dimensions...
appraise(mod)
```

bird_move	<i>Simulated bird migration data</i>
-----------	--------------------------------------

Description

Data generated from a hypothetical study of bird movement along a migration corridor, sampled throughout the year. This dataset consists of simulated sample records of numbers of observed locations of 100 tagged individuals each from six species of bird, at ten locations along a latitudinal gradient, with one observation taken every four weeks. Counts were simulated randomly for each species in each location and week by creating a species-specific migration curve that gave the probability of finding an individual of a given species in a given location, then simulated the distribution of individuals across sites using a multinomial distribution, and subsampling that using a binomial distribution to simulation observation error (i.e. not every bird present at a location would be detected). The data set (bird_move) consists of the variables count, latitude, week and species.

Format

A data frame

Source

Pedersen EJ, Miller DL, Simpson GL, Ross N. 2018. Hierarchical generalized additive models: an introduction with mgcv. *PeerJ Preprints* 6:e27320v1 doi: [10.7287/peerj.preprints.27320v1](https://doi.org/10.7287/peerj.preprints.27320v1).

check_user_select_smooths	<i>Select smooths based on user's choices</i>
---------------------------	---

Description

Given a vector indexing the smooths of a GAM, returns a logical vector selecting the requested smooths.

Usage

```
check_user_select_smooths(smooths, select = NULL,
  partial_match = FALSE)
```

Arguments

smooths	character; a vector of smooth labels.
select	numeric, logical, or character vector of selected smooths.
partial_match	logical; in the case of character select, should select match partially against smooths? If partial_match = TRUE, select must only be a single string, a character vector of length 1.

Value

A logical vector the same length as `length(smooths)` indicating which smooths have been selected.

Author(s)

Gavin L. Simpson

coef.scam

Extract coefficients from a fitted scam model.

Description

Extract coefficients from a fitted scam model.

Usage

```
## S3 method for class 'scam'
coef(object, parametrized = TRUE, ...)
```

Arguments

object	a model object fitted by scam()
parametrized	logical; extract parametrized coefficients, which respect the linear inequality constraints of the model.
...	other arguments.

confint.fderiv	<i>Point-wise and simultaneous confidence intervals for derivatives of smooths</i>
----------------	--

Description

Calculates point-wise confidence or simultaneous intervals for the first derivatives of smooth terms in a fitted GAM.

Usage

```
## S3 method for class 'fderiv'
confint(object, parm, level = 0.95,
        type = c("confidence", "simultaneous"), nsim = 10000, ...)
```

Arguments

object	an object of class "fderiv" containing the estimated derivatives.
parm	which parameters (smooth terms) are to be given intervals as a vector of terms. If missing, all parameters are considered.
level	numeric, $0 < \text{level} < 1$; the confidence level of the point-wise or simultaneous interval. The default is 0.95 for a 95 interval.
type	character; the type of interval to compute. One of "confidence" for point-wise intervals, or "simultaneous" for simultaneous intervals.
nsim	integer; the number of simulations used in computing the simultaneous intervals.
...	additional arguments for methods

Value

a data frame with components:

1. term; factor indicating to which term each row relates,
2. lower; lower limit of the confidence or simultaneous interval,
3. est; estimated derivative
4. upper; upper limit of the confidence or simultaneous interval.

Author(s)

Gavin L. Simpson

Examples

```

suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
mod <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

## first derivatives of all smooths...
fd <- fderiv(mod)

## point-wise interval
ci <- confint(fd, type = "confidence")
head(ci)

## simultaneous interval for smooth term of x1
set.seed(42)
x1.sint <- confint(fd, parm = "x1", type = "simultaneous", nsim = 1000)
head(x1.sint)

```

confint.gam

Point-wise and simultaneous confidence intervals for smooths

Description

Calculates point-wise confidence or simultaneous intervals for the smooth terms of a fitted GAM.

Usage

```

## S3 method for class 'gam'
confint(object, parm, level = 0.95, newdata = NULL,
        n = 200, type = c("confidence", "simultaneous"), nsim = 10000,
        shift = FALSE, transform = FALSE, unconditional = FALSE, ...)

## S3 method for class 'gamm'
confint(object, ...)

```

Arguments

object	an object of class "gam" or "gamm".
parm	which parameters (smooth terms) are to be given intervals as a vector of terms. If missing, all parameters are considered, although this is not currently implemented.
level	numeric, $0 < \text{level} < 1$; the confidence level of the point-wise or simultaneous interval. The default is 0.95 for a 95% interval.
newdata	data frame; containing new values of the covariates used in the model fit. The selected smooth(s) will be evaluated at the supplied values.
n	numeric; the number of points to evaluate smooths at.

type	character; the type of interval to compute. One of "confidence" for point-wise intervals, or "simultaneous" for simultaneous intervals.
nsim	integer; the number of simulations used in computing the simultaneous intervals.
shift	logical; should the constant term be add to the smooth?
transform	logical; should the smooth be evaluated on a transformed scale? For generalised models, this involves applying the inverse of the link function used to fit the model. Alternatively, the name of, or an actual, function can be supplied to transform the smooth and it's confidence interval.
unconditional	logical; if TRUE (and freq == FALSE) then the Bayesian smoothing parameter uncertainty corrected covariance matrix is returned, if available.
...	additional arguments for methods

Value

a data frame with components:

1. term; factor indicating to which term each row relates,
2. x; the vector of values at which the smooth was evaluated,
3. lower; lower limit of the confidence or simultaneous interval,
4. est; estimated value of the smooth
5. upper; upper limit of the confidence or simultaneous interval,
6. crit; critical value for the $100 * \text{level}$

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))
set.seed(2)
dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
mod <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")
##'
## point-wise interval
ci <- confint(mod, parm = "x1", type = "confidence")
head(ci)
##'
## simultaneous interval for smooth term of x1
set.seed(42)
si <- confint(mod, parm = "x1", type = "simultaneous", nsim = 100)
head(si)
```

datagen	<i>Generate data over the range of variables used in smooths</i>
---------	--

Description

For each smooth in a GAM, generate new data over the range of the variables involved in a smooth.

Usage

```
datagen(x, ...)  
  
## S3 method for class 'mgcv.smooth'  
datagen(x, n = 100, data, ...)  
  
## S3 method for class 'fs.interaction'  
datagen(x, n = 100, data, ...)  
  
## S3 method for class 'gam'  
datagen(x, n = 200, ...)  
  
## S3 method for class 'gamm'  
datagen(x, ...)
```

Arguments

x	an object for which new data is required. Currently objects of classes "gam", and "gamm" are supported, as are smooths from mgcv inheriting from class "mgcv.smooth".
...	arguments passed to methods
n	numeric; the number of data values to generate per term in each smooth.
data	data frame; for "mgcv.smooth" objects, the data used to fit the GAM need to be supplied.

Value

A data frame of new values spread over the range of the observed values.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))  
  
## 1d example  
set.seed(2)
```

```

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")
df <- datagen(m1)
head(df)

## 2d example
dat <- gamSim(2, n = 400, dist = "normal", scale = 2)
m2 <- gam(y ~ s(x, z), data = dat$data, method = "REML")
df <- datagen(m2)
head(df)
## alternative showing using the mgcv.smooth method for a single smooth
df2 <- datagen(m2[["smooth"]][[1L]], data = dat$data)
head(df2)

```

data_class	<i>Vectorized version of data.class</i>
------------	---

Description

Vectorized version of data.class

Usage

```
data_class(df)
```

Arguments

df a data frame or tibble.##'

Value

A named character vector of data classes.

See Also

The underlying functionality is provided by [data.class\(\)](#).

data_slice	<i>Prepare a data slice through covariates</i>
------------	--

Description

Prepare a data slice through covariates

Usage

```
data_slice(object, ...)

## Default S3 method:
data_slice(object, ...)

## S3 method for class 'gam'
data_slice(object, var1, var2, var3 = NULL, var4 = NULL,
  data = NULL, n = 50, offset = NULL, ...)
```

Arguments

object	an R model object.
...	arguments passed to other methods.
var1	character;
var2	character;
var3	character; ignored currently.
var4	character; ignored currently.
data	a 1-row data frame or tibble containing values for variables in the fitted model that are not varying in the slice.
n	numeric; the number of values to create for each of var1 and var2 in the slice.
offset	numeric; value to use for an offset term in the model.

 derivatives

Derivatives of estimated smooths via finite differences

Description

Derivatives of estimated smooths via finite differences

Usage

```
derivatives(object, ...)

## Default S3 method:
derivatives(object, ...)

## S3 method for class 'gamm'
derivatives(object, ...)

## S3 method for class 'gam'
derivatives(object, term, newdata, order = 1L,
  type = c("forward", "backward", "central"), n = 200, eps = 1e-07,
  interval = c("confidence", "simultaneous"), n_sim = 10000,
  level = 0.95, unconditional = FALSE, frequentist = FALSE,
  offset = NULL, ...)
```

Arguments

object	an R object to compute derivatives for.
...	arguments passed to other methods.
term	character; vector of one or more smooth terms for which derivatives are required. If missing, derivatives for all smooth terms will be returned. Can be a partial match to a smooth term.
newdata	a data frame containing the values of the model covariates at which to evaluate the first derivatives of the smooths.
order	numeric; the order of derivative.
type	character; the type of finite difference used. One of "forward", "backward", or "central".
n	numeric; the number of points to evaluate the derivative at.
eps	numeric; the finite difference.
interval	character; the type of interval to compute. One of "confidence" for point-wise intervals, or "simultaneous" for simultaneous intervals.
n_sim	integer; the number of simulations used in computing the simultaneous intervals.
level	numeric; $0 < \text{level} < 1$; the confidence level of the point-wise or simultaneous interval. The default is 0.95 for a 95 interval.
unconditional	logical; use smoothness selection-corrected Bayesian covariance matrix?
frequentist	logical; use the frequentist covariance matrix?
offset	numeric; a value to use for any offset term

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2, verbose = FALSE)
mod <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

## first derivative of all smooths using central finite differenc
derivatives(mod, type = "central")
```

draw	<i>Generic plotting via ggplot2</i>
------	-------------------------------------

Description

Generic plotting via ggplot2

Usage

```
draw(object, ...)
```

Arguments

object	and R object to plot.
...	arguments passed to other methods.

Details

Generic function for plotting of R objects that uses the ggplot2 package.

Value

A `ggplot2::ggplot()` object.

Author(s)

Gavin L. Simpson

draw.derivatives	<i>Plot derivatives of smooths</i>
------------------	------------------------------------

Description

Plot derivatives of smooths

Usage

```
## S3 method for class 'derivatives'  
draw(object, select = NULL, scales = c("free",  
    "fixed"), alpha = 0.2, align = "hv", axis = "lrtb", ...)
```

Arguments

object	a fitted GAM, the result of a call to <code>mgcv::gam()</code> .
select	character, logical, or numeric; which smooths to plot. If NULL, the default, then all model smooths are drawn. Numeric select indexes the smooths in the order they are specified in the formula and stored in object. Character select matches the labels for smooths as shown for example in the output from <code>summary(object)</code> . Logical select operates as per numeric select in the order that smooths are stored.
scales	character; should all univariate smooths be plotted with the same y-axis scale? The default, <code>scales = "fixed"</code> , ensures this is done. If <code>scales = "free"</code> each univariate smooth has its own y-axis scale. Currently does not affect the y-axis scale of plots of the parametric terms.
alpha	numeric; alpha transparency for confidence or simultaneous interval.
align	character; see argument <code>align</code> in <code>cowplot::plot_grid()</code> . Defaults to "hv" so that plots are nicely aligned.
axis	character; see argument <code>axis</code> in <code>cowplot::plot_grid()</code> . Defaults to "lrbt" so that plots are nicely aligned.
...	arguments passed to <code>cowplot::plot_grid()</code> . Any arguments to <code>plot_grid()</code> may be supplied, except for: <code>plotlist</code> and <code>align</code> .

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2, verbose = FALSE)
mod <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

## first derivative of all smooths
df <- derivatives(mod)
draw(df)
```

draw.evaluated_smooth *Plot estimated smooths*

Description

Plots estimated univariate and bivariate smooths using `ggplot2`.

Usage

```
## S3 method for class 'evaluated_1d_smooth'
draw(object, rug = NULL, xlab, ylab,
      title = NULL, subtitle = NULL, caption = NULL, ...)

## S3 method for class 'evaluated_2d_smooth'
```

```

draw(object, show = c("estimate", "se"),
      contour = TRUE, xlab, ylab, title = NULL, subtitle = NULL,
      caption = NULL, ...)

## S3 method for class 'evaluated_re_smooth'
draw(object, qq_line = TRUE, xlab, ylab,
      title = NULL, subtitle = NULL, caption = NULL, ...)

## S3 method for class 'evaluated_fs_smooth'
draw(object, rug = NULL, xlab, ylab,
      title = NULL, subtitle = NULL, caption = NULL,
      colour_scale = scale_colour_discrete, ...)

## S3 method for class 'evaluated_parametric_term'
draw(object, xlab, ylab,
      title = NULL, subtitle = NULL, caption = NULL, rug = TRUE,
      position = "identity", ...)

```

Arguments

object	an object, the result of a call to evaluate_smooth() .
rug	For evaluate_smooth() , a numeric vector of values for the location of data on the x axis. The default of NULL results in no rug plot being drawn. For evaluate_parametric_terms() , a logical to indicate if a rug plot should be drawn.
xlab	character or expression; the label for the x axis. If not supplied, a suitable label will be generated from object.
ylab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated from object.
title	character or expression; the title for the plot. See ggplot2::labs() .
subtitle	character or expression; the subtitle for the plot. See ggplot2::labs() .
caption	character or expression; the plot caption. See ggplot2::labs() .
...	arguments passed to other methods.
show	character; plot the estimated smooth ("estimate") or its standard error ("se").
contour	logical; should contours be draw on the plot using ggplot2::geom_contour() .
qq_line	logical; draw a reference line through the lower and upper theoretical quartiles.
colour_scale	function; an appropriate discrete colour scale from ggplot2 .
position	Position adjustment, either as a string, or the result of a call to a position adjustment function.

Value

A [ggplot2::ggplot\(\)](#) object.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

sm <- evaluate_smooth(m1, "s(x2)")
draw(sm)

dat <- gamSim(2, n = 1000, dist = "normal", scale = 1)
m2 <- gam(y ~ s(x, z, k = 40), data = dat$data, method = "REML")

sm <- evaluate_smooth(m2, "s(x,z)", n = 100)
draw(sm)
```

draw.gam

*Plot estimated smooths from a fitted GAM***Description**

Plots estimated smooths from a fitted GAM model in a similar way to `mgcv::plot.gam()` but instead of using base graphics, `ggplot2::ggplot()` is used instead.

Usage

```
## S3 method for class 'gam'
draw(object, parametric = NULL, select = NULL,
      scales = c("free", "fixed"), align = "hv", axis = "lrtb",
      n = 100, unconditional = FALSE, overall_uncertainty = TRUE,
      dist = 0.1, rug = TRUE, partial_match = FALSE, ...)
```

Arguments

object	a fitted GAM, the result of a call to <code>mgcv::gam()</code> .
parametric	logical; plot parametric terms also? Default is TRUE, only if select is NULL. If select is used, parametric is set to FALSE unless the user specifically sets parametric = TRUE.
select	character, logical, or numeric; which smooths to plot. If NULL, the default, then all model smooths are drawn. Numeric select indexes the smooths in the order they are specified in the formula and stored in object. Character select matches the labels for smooths as shown for example in the output from <code>summary(object)</code> . Logical select operates as per numeric select in the order that smooths are stored.

scales	character; should all univariate smooths be plotted with the same y-axis scale? The default, <code>scales = "fixed"</code> , ensures this is done. If <code>scales = "free"</code> each univariate smooth has its own y-axis scale. Currently does not affect the y-axis scale of plots of the parametric terms.
align	character; see argument <code>align</code> in <code>cowplot::plot_grid()</code> . Defaults to <code>"hv"</code> so that plots are nicely aligned.
axis	character; see argument <code>axis</code> in <code>cowplot::plot_grid()</code> . Defaults to <code>"lrtpb"</code> so that plots are nicely aligned.
n	numeric; the number of points over the range of the covariate at which to evaluate the smooth.
unconditional	logical; should confidence intervals include the uncertainty due to smoothness selection? If TRUE, the corrected Bayesian covariance matrix will be used.
overall_uncertainty	logical; should the uncertainty in the model constant term be included in the standard error of the evaluate values of the smooth?
dist	numeric; if greater than 0, this is used to determine when a location is too far from data to be plotted when plotting 2-D smooths. The data are scaled into the unit square before deciding what to exclude, and <code>dist</code> is a distance within the unit square. See <code>mgcv::exclude.too.far()</code> for further details.
rug	logical; draw a rug plot at the bottom of each plot?
partial_match	logical; should smooths be selected by partial matches with <code>select</code> ? If TRUE, <code>select</code> can only be a single string to match against.
...	arguments passed to <code>cowplot::plot_grid()</code> . Any arguments to <code>plot_grid()</code> may be supplied, except for: <code>plotlist</code> and <code>align</code> .

Value

The object returned is created by `cowplot::plot_grid()`.

Note

Internally, plots of each smooth are created using `ggplot2::ggplot()` and composed into a single plot using `cowplot::plot_grid()`. As a result, it is not possible to use `+` to add to the plots in the way one might typically work with `ggplot()` plots.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

draw(m1)
```

evaluate_smooth	<i>Evaluate a smooth</i>
-----------------	--------------------------

Description

Evaluate a smooth at a grid of evenly spaced value over the range of the covariate associated with the smooth. Alternatively, a set of points at which the smooth should be evaluated can be supplied.

Usage

```
evaluate_smooth(object, ...)

## S3 method for class 'gam'
evaluate_smooth(object, smooth, n = 100, newdata = NULL,
  unconditional = FALSE, overall_uncertainty = TRUE, dist = 0.1, ...)

## S3 method for class 'gamm'
evaluate_smooth(object, ...)

evaluate_parametric_term(object, ...)

## S3 method for class 'gam'
evaluate_parametric_term(object, term,
  unconditional = FALSE, ...)
```

Arguments

object	an object of class "gam" or "gamm".
...	arguments passed to other methods.
smooth	character; a single smooth to evaluate.
n	numeric; the number of points over the range of the covariate at which to evaluate the smooth.
newdata	a vector or data frame of points at which to evaluate the smooth.
unconditional	logical; should confidence intervals include the uncertainty due to smoothness selection? If TRUE, the corrected Bayesian covariance matrix will be used.
overall_uncertainty	logical; should the uncertainty in the model constant term be included in the standard error of the evaluate values of the smooth?
dist	numeric; if greater than 0, this is used to determine when a location is too far from data to be plotted when plotting 2-D smooths. The data are scaled into the unit square before deciding what to exclude, and dist is a distance within the unit square. See <code>mgcv::exclude.too.far()</code> for further details.
term	character; which parametric term whose effects are evaluated

Value

A data frame, which is of class "evaluated_1d_smooth" or evaluated_2d_smooth, which inherit from classes "evaluated_smooth" and "data.frame".

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

evaluate_smooth(m1, "s(x1)")

## 2d example
set.seed(2)
dat <- gamSim(2, n = 1000, dist = "normal", scale = 1)
m2 <- gam(y ~ s(x, z, k = 30), data = dat$data, method = "REML")

evaluate_smooth(m2, "s(x,z)", n = 100)
```

factor_var_names	<i>Names of any factor variables in model data</i>
------------------	--

Description

Names of any factor variables in model data

Usage

```
factor_var_names(df)
```

Arguments

df a data frame or tibble

fderiv	<i>First derivatives of fitted GAM functions</i>
--------	--

Description

The first derivative of the smooth functions of a GAM model calculated using finite differences.

Usage

```
fderiv(model, ...)

## S3 method for class 'gam'
fderiv(model, newdata, term, n = 200, eps = 1e-07,
        unconditional = FALSE, offset = NULL, ...)

## S3 method for class 'gamm'
fderiv(model, ...)
```

Arguments

model	A fitted GAM. Currently only models fitted by <code>mgcv::gam()</code> and <code>mgcv::gamm()</code> are supported.
...	Arguments that are passed to other methods.
newdata	a data frame containing the values of the model covariates at which to evaluate the first derivatives of the smooths.
term	character; vector of one or more terms for which derivatives are required. If missing, derivatives for all smooth terms will be returned.
n	integer; if <code>newdata</code> is missing the original data can be reconstructed from <code>model</code> and then <code>n</code> controls the number of values over the range of each covariate with which to populate <code>newdata</code> .
eps	numeric; the value of the finite difference used to approximate the first derivative.
unconditional	logical; if TRUE, the smoothing parameter uncertainty corrected covariance matrix is used, <i>if available</i> , otherwise the uncorrected Bayesian posterior covariance matrix is used.
offset	numeric; value of offset to use in generating predictions.

Value

An object of class "fderiv" is returned.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
mod <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

## first derivatives of all smooths...
fd <- fderiv(mod)
```

```
## ...and a selected smooth
fd2 <- fderiv(mod, term = "x1")

## Models with factors
set.seed(2)
dat <- gamSim(4, n = 400, dist = "normal", scale = 2)
mod <- gam(y ~ s(x0) + s(x1) + fac, data = dat, method = "REML")

## first derivatives of all smooths...
fd <- fderiv(mod)

## ...and a selected smooth
fd2 <- fderiv(mod, term = "x1")
```

fix_offset

Fix the names of a data frame containing an offset variable.

Description

Identifies which variable, if any, is the model offset, and fixed the name such that `offset(foo(var))` is converted to `var`, and possibly sets the values of that variable to `offset_val`.

Usage

```
fix_offset(model, newdata, offset_val = NULL)
```

Arguments

<code>model</code>	a fitted GAM.
<code>newdata</code>	data frame; new values at which to predict at.
<code>offset_val</code>	numeric, optional; if provided, then the offset variable in <code>newdata</code> is set to this constant value before returning <code>newdata</code>

Value

The original `newdata` is returned with fixed names and possibly modified offset variable.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))

df <- gamSim(1, n = 400, dist = "normal")
m <- gam(y ~ s(x0) + s(x1) + offset(x2), data = df, method = "REML")
names(model.frame(m))
names(fix_offset(m, model.frame(m), offset_val = 1L))
```

get_by_smooth	<i>Extract an factor-by smooth by name</i>
---------------	--

Description

Extract an factor-by smooth by name

Usage

```
get_by_smooth(object, term, level)
```

Arguments

object	a fitted GAM model object.
term	character; the name of a smooth term to extract.
level	character; which level of the factor to extract the smooth for..

Value

A single smooth object, or a list of smooths if several match the named term.

get_smooth	<i>Extract an mgcv smooth by name</i>
------------	---------------------------------------

Description

Extract an mgcv smooth by name

Usage

```
get_smooth(object, term)
```

Arguments

object	a fitted GAM model object.
term	character; the name of a smooth term to extract

Value

A single smooth object, or a list of smooths if several match the named term.

get_smooths_by_id	<i>Extract an mgcv smooth given its position in the model object</i>
-------------------	--

Description

Extract an mgcv smooth given its position in the model object

Usage

```
get_smooths_by_id(object, id)
```

Arguments

object	a fitted GAM model object.
id	numeric; the position of the smooth in the model object.

is_by_smooth	<i>Tests for by variable smooths</i>
--------------	--------------------------------------

Description

Functions to check if a smooth is a by-variable one and to test of the type of by-variable smooth is a factor-smooth or a continuous-smooth interaction.

Usage

```
is_by_smooth(smooth)
is_factor_by_smooth(smooth)
is_continuous_by_smooth(smooth)
by_variable(smooth)
by_level(smooth)
```

Arguments

smooth	an object of class "mgcv.smooth"
--------	----------------------------------

Value

A logical vector.

Author(s)

Gavin L. Simpson

is_factor_var	<i>Are variables in a data frame factors?</i>
---------------	---

Description

Are variables in a data frame factors?

Usage

```
is_factor_var(df)
```

Arguments

df a data frame or tibble

is_mgcv_smooth	<i>Check if objects are smooths or are a particular type of smooth</i>
----------------	--

Description

Check if objects are smooths or are a particular type of smooth

Usage

```
is_mgcv_smooth(smooth)
```

```
is_mrf_smooth(smooth)
```

Arguments

smooth an R object, typically a list

is_offset	<i>Is a model term an offset?</i>
-----------	-----------------------------------

Description

Given a character vector of model terms, checks to see which, if any, is the model offset.

Usage

```
is_offset(terms)
```

Arguments

terms character vector of model terms.

Value

A logical vector of the same length as terms.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))
df <- gamSim(1, n = 400, dist = "normal")
m <- gam(y ~ s(x0) + s(x1) + offset(x0), data = df, method = "REML")
nm <- names(model.frame(m))
nm
is_offset(nm)
```

n_smooths	<i>How many smooths in a fitted model</i>
-----------	---

Description

How many smooths in a fitted model

Usage

```
n_smooths(object)

## Default S3 method:
n_smooths(object)

## S3 method for class 'gam'
n_smooths(object)

## S3 method for class 'gamm'
n_smooths(object)

## S3 method for class 'bam'
n_smooths(object)
```

Arguments

`object` a fitted GAM or related model. Typically the result of a call to `mgcv::gam()`, `mgcv::bam()`, or `mgcv::gamm()`.

`observed_fitted_plot` *Plot of fitted against observed response values*

Description

Plot of fitted against observed response values

Usage

```
observed_fitted_plot(model, ylab = NULL, xlab = NULL, title = NULL,
  subtitle = NULL, caption = NULL)
```

Arguments

`model` a fitted model. Currently only class "gam".

`ylab` character or expression; the label for the y axis. If not supplied, a suitable label will be generated.

`xlab` character or expression; the label for the y axis. If not supplied, a suitable label will be generated.

`title` character or expression; the title for the plot. See `ggplot2::labs()`.

`subtitle` character or expression; the subtitle for the plot. See `ggplot2::labs()`.

`caption` character or expression; the plot caption. See `ggplot2::labs()`.

parametric_terms	<i>Names of any parametrix terms in a GAM</i>
------------------	---

Description

Names of any parametrix terms in a GAM

Usage

```
parametric_terms(model, ...)
```

```
## Default S3 method:  
parametric_terms(model, ...)
```

```
## S3 method for class 'gam'  
parametric_terms(model, ...)
```

Arguments

model	a fitted model.
...	arguments passed to other methods.

qq_plot	<i>Quantile-quantile plot of model residuals</i>
---------	--

Description

Quantile-quantile plot of model residuals

Usage

```
qq_plot(model, ...)
```

```
## S3 method for class 'gam'  
qq_plot(model, method = c("direct", "simulate", "normal"),  
  type = c("deviance", "response", "pearson"), n_uniform = 10,  
  n_simulate = 50, level = 0.9, ylab = NULL, xlab = NULL,  
  title = NULL, subtitle = NULL, caption = NULL, alpha = 0.2, ...)
```

Arguments

model	a fitted model. Currently only class "gam".
...	arguments passed to other methods.
method	character; method used to generate theoretical quantiles.
type	character; type of residuals to use. Only "deviance", "response", and "pearson" residuals are allowed.
n_uniform	numeric; number of times to randomize uniform quantiles in the direct computation method (method = "direct").
n_simulate	numeric; number of data sets to simulate from the estimated model when using the simulation method (method = "simulate").
level	numeric; the coverage level for reference intervals. Must be strictly $0 < \text{level} < 1$. Only used with method = "simulate".
ylab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated.
xlab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated.
title	character or expression; the title for the plot. See <code>ggplot2::labs()</code> .
subtitle	character or expression; the subtitle for the plot. See <code>ggplot2::labs()</code> .
caption	character or expression; the plot caption. See <code>ggplot2::labs()</code> .
alpha	numeric; the level of alpha transparency for the reference interval when method = "simulate".

Examples

```

suppressPackageStartupMessages(library("mgcv"))
## simulate binomial data...
set.seed(0)
n.samp <- 200
dat <- gamSim(1, n = n.samp, dist = "binary", scale = .33)
p <- binomial()$linkinv(dat$f) # binomial p
n <- sample(c(1, 3), n.samp, replace = TRUE) # binomial n
dat <- transform(dat, y = rbinom(n, n, p), n = n)
m <- gam( y / n ~ s(x0) + s(x1) + s(x2) + s(x3),
         family = binomial, data = dat, weights = n,
         method = "REML")

## Q-Q plot; default using direct randomization of uniform quantiles
qq_plot(m)

## Alternatively use simulate new data from the model, which
## allows construction of reference intervals for the Q-Q plot
qq_plot(m, method = "simulate")

## ... or use the usual normality assumption
qq_plot(m, method = "normal")

```

 rep_first_factor_value

Repeat the first level of a factor n times

Description

Function to repeat the first level of a factor n times and return this vector as a factor with the original levels intact

Usage

```
rep_first_factor_value(f, n)
```

Arguments

f	a factor
n	numeric; the number of times to repeat the first level of f

Value

A factor of length n with the levels of f, but whose elements are all the first level of f.

 residuals_hist_plot *Histogram of model residuals*

Description

Histogram of model residuals

Usage

```
residuals_hist_plot(model, type = c("deviance", "pearson", "response"),
  n_bins = c("sturges", "scott", "fd"), ylab = NULL, xlab = NULL,
  title = NULL, subtitle = NULL, caption = NULL)
```

Arguments

model	a fitted model. Currently only class "gam".
type	character; type of residuals to use. Only "deviance", "response", and "pearson" residuals are allowed.
n_bins	character or numeric; either the number of bins or a string indicating how to calculate the number of bins.
ylab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated.

xlab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated.
title	character or expression; the title for the plot. See ggplot2::labs() .
subtitle	character or expression; the subtitle for the plot. See ggplot2::labs() .
caption	character or expression; the plot caption. See ggplot2::labs() .

residuals_linpred_plot

Plot of residuals versus linear predictor values

Description

Plot of residuals versus linear predictor values

Usage

```
residuals_linpred_plot(model, type = c("deviance", "pearson",
  "response"), ylab = NULL, xlab = NULL, title = NULL,
  subtitle = NULL, caption = NULL)
```

Arguments

model	a fitted model. Currently only class "gam".
type	character; type of residuals to use. Only "deviance", "response", and "pearson" residuals are allowed.
ylab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated.
xlab	character or expression; the label for the y axis. If not supplied, a suitable label will be generated.
title	character or expression; the title for the plot. See ggplot2::labs() .
subtitle	character or expression; the subtitle for the plot. See ggplot2::labs() .
caption	character or expression; the plot caption. See ggplot2::labs() .

seq_min_max	<i>Create a sequence of evenly-spaced values</i>
-------------	--

Description

Creates a sequence of n evenly-spaced values over the range $\min(x) - \max(x)$.

Usage

```
seq_min_max(x, n)
```

Arguments

x	numeric; vector over which evenly-spaced values are returned
n	numeric; the number of evenly-spaced values to return

Value

A numeric vector of length n .

Examples

```
x <- rnorm(10)
n <- 10L
seq_min_max(x, n = n)
```

seq_min_max_eps	<i>Create a sequence of evenly-spaced values adjusted to accommodate a small adjustment</i>
-----------------	---

Description

Creates a sequence of n evenly-spaced values over the range $\min(x) - \max(x)$, where the minimum and maximum are adjusted such that they are always contained within the range of x when x may be shifted forwards or backwards by an amount related to eps . This is particularly useful in computing derivatives via finite differences where without this adjustment we may be predicting for values outside the range of the data and hence the constraints of the penalty.

Usage

```
seq_min_max_eps(x, n, order, type = c("forward", "backward", "central"),
  eps)
```

Arguments

x	numeric; vector over which evenly-spaced values are returned
n	numeric; the number of evenly-spaced values to return
order	integer; the order of derivative. Either 1 or 2 for first or second order derivatives
type	character; the type of finite difference used. One of "forward", "backward", or "central"
eps	numeric; the finite difference

Value

A numeric vector of length n.

shift_values	<i>Shift numeric values in a data frame by an amount eps</i>
--------------	--

Description

Shift numeric values in a data frame by an amount eps

Usage

```
shift_values(df, h, i, FUN = "+")
```

Arguments

df	a data frame or tibble.
h	numeric; the amount to shift values in df by.
i	logical; a vector indexing columns of df that should not be included in the shift.
FUN	function; a function to apply the shift. Typically + or -.

simulate.gam	<i>Simulate from the posterior distribution of a GAM</i>
--------------	--

Description

Simulations from the posterior distribution of a fitted GAM model involve making random draws from a multivariate normal with mean vector equal to the estimated model coefficients and covariance matrix equal to the covariance matrix of the coefficients.

Usage

```
## S3 method for class 'gam'
simulate(object, nsim = 1, seed = NULL, newdata = NULL,
         freq = FALSE, unconditional = FALSE, weights = NULL, ...)

## S3 method for class 'gamm'
simulate(object, nsim = 1, seed = NULL,
         newdata = NULL, freq = FALSE, unconditional = FALSE,
         weights = NULL, ...)

## S3 method for class 'scam'
simulate(object, nsim = 1, seed = NULL,
         newdata = NULL, freq = FALSE, weights = NULL, ...)
```

Arguments

object	a fitted GAM, typically the result of a call to <code>mgcv::gam</code> or <code>mgcv::gamm()</code> .
nsim	numeric; the number of posterior simulations to return.
seed	numeric; a random seed for the simulations.
newdata	data frame; new observations at which the posterior draws from the model should be evaluated. If not supplied, the data used to fit the model will be used for newdata, if available in object.
freq	logical; TRUE to return the frequentist covariance matrix of the parameter estimators, FALSE to return the Bayesian posterior covariance matrix of the parameters.
unconditional	logical; if TRUE (and <code>freq == FALSE</code>) then the Bayesian smoothing parameter uncertainty corrected covariance matrix is returned, if available.
weights	numeric; a vector of prior weights. If newdata is null then defaults to <code>object[["prior.weights"]]</code> , otherwise a vector of ones.
...	arguments passed to methods

Value

(Currently) A matrix with `nsim` columns.

Author(s)

Gavin L. Simpson

Examples

```
suppressPackageStartupMessages(library("mgcv"))

dat <- gamSim(1, n = 400, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

sims <- simulate(m1, nsim = 5, seed = 42)
head(sims)
```

smallAges

Lead-210 age-depth measurements for Small Water

Description

A dataset containing lead-210 based age depth measurements for the SMALL1 core from Small Water.

Format

A data frame with 12 rows and 7 variables.

Details

The variables are as follows:

- Depth
- Drymass
- Date
- Age
- Error
- SedAccRate
- SedPerCentChange

Source

Simpson, G.L. (Unpublished data).

smooths

Names of smooths in a GAM

Description

Names of smooths in a GAM

Usage

```
smooths(object)
```

Arguments

object a fitted GAM or related model. Typically the result of a call to `mgcv::gam()`, `mgcv::bam()`, or `mgcv::gamm()`.

smooth_dim	<i>Dimension of a smooth</i>
------------	------------------------------

Description

Extracts the dimension of an estimated smooth.

Usage

```
smooth_dim(object)

## S3 method for class 'gam'
smooth_dim(object)

## S3 method for class 'gamm'
smooth_dim(object)

## S3 method for class 'mgcv.smooth'
smooth_dim(object)
```

Arguments

object an R object. See Details for list of supported objects.

Details

This is a generic function with methods for objects of class "gam", "gamm", and "mgcv.smooth".

Value

A numeric vector of dimensions for each smooth.

Author(s)

Gavin L. Simpson

which_smooths	<i>Identify a smooth term by it's label</i>
---------------	---

Description

Identify a smooth term by it's label

Usage

```

which_smooths(object, ...)

## Default S3 method:
which_smooths(object, ...)

## S3 method for class 'gam'
which_smooths(object, terms, ...)

## S3 method for class 'bam'
which_smooths(object, terms, ...)

## S3 method for class 'gamm'
which_smooths(object, terms, ...)

```

Arguments

object	a fitted GAM.
...	arguments passed to other methods.
terms	character; one or more (partial) term labels with which to identify required smooths.

zooplankton

Madison lakes zooplankton data

Description

The Madison lake zooplankton data are from a long-term study in seasonal dynamics of zooplankton, collected by the Richard Lathrop. The data were collected from a chain of lakes in Wisconsin (Mendota, Monona, Kegonsa, and Waubesa) approximately bi-weekly from 1976 to 1994. They consist of samples of the zooplankton communities, taken from the deepest point of each lake via vertical tow. The data are provided by the Wisconsin Department of Natural Resources and their collection and processing are fully described in Lathrop (2000).

Format

A data frame

Details

Each record consists of counts of a given zooplankton taxon taken from a subsample from a single vertical net tow, which was then scaled to account for the relative volume of subsample versus the whole net sample and the area of the net tow and rounded to the nearest 1000 to give estimated population density per m² for each taxon at each point in time in each sampled lake.

Source

Pedersen EJ, Miller DL, Simpson GL, Ross N. 2018. Hierarchical generalized additive models: an introduction with mgcv. *PeerJ Preprints* **6**:e27320v1 doi: [10.7287/peerj.preprints.27320v1](https://doi.org/10.7287/peerj.preprints.27320v1).

References

Lathrop RC. (2000). Madison Wisconsin Lakes Zooplankton 1976–1994. Environmental Data Initiative.

Index

*Topic **data**

- bird_move, 4
 - smallAges, 34
 - zooplankton, 36
- appraise, 3
- bird_move, 4
- by_level(is_by_smooth), 23
- by_variable(is_by_smooth), 23
- check_user_select_smooths, 4
- coef.scam, 5
- confint.fderiv, 6
- confint.gam, 7
- confint.gamm(confint.gam), 7
- cowplot::plot_grid(), 3, 17
- data.class(), 10
- data_class, 10
- data_slice, 10
- datagen, 9
- derivatives, 11
- draw, 13
- draw.derivatives, 13
- draw.evaluated_1d_smooth
(draw.evaluated_smooth), 14
- draw.evaluated_2d_smooth
(draw.evaluated_smooth), 14
- draw.evaluated_fs_smooth
(draw.evaluated_smooth), 14
- draw.evaluated_parametric_term
(draw.evaluated_smooth), 14
- draw.evaluated_re_smooth
(draw.evaluated_smooth), 14
- draw.evaluated_smooth, 14
- draw.gam, 16
- evaluate_parametric_term
(evaluate_smooth), 18
- evaluate_smooth, 18
- evaluate_smooth(), 15
- factor_var_names, 19
- fderiv, 19
- fix_offset, 21
- geom_rug(draw.evaluated_smooth), 14
- get_by_smooth, 22
- get_smooth, 22
- get_smooths_by_id, 23
- ggplot2::geom_contour(), 15
- ggplot2::ggplot(), 13, 15–17
- ggplot2::labs(), 15, 26, 28, 30
- gratia::observed_fitted_plot(), 3
- gratia::qq_plot(), 3
- gratia::residuals_hist_plot(), 3
- gratia::residuals_linpred_plot(), 3
- is_by_smooth, 23
- is_continuous_by_smooth(is_by_smooth), 23
- is_factor_by_smooth(is_by_smooth), 23
- is_factor_var, 24
- is_mgcv_smooth, 24
- is_mrf_smooth(is_mgcv_smooth), 24
- is_offset, 25
- mgcv::bam(), 26, 34
- mgcv::exclude_too_far(), 17, 18
- mgcv::gam, 33
- mgcv::gam(), 14, 16, 20, 26, 34
- mgcv::gamm(), 20, 26, 33, 34
- n_smooths, 25
- observed_fitted_plot, 26
- parametric_terms, 27
- qq_plot, 27
- rep_first_factor_value, 29

residuals_hist_plot, [29](#)
residuals_linpred_plot, [30](#)

seq_min_max, [31](#)
seq_min_max_eps, [31](#)
shift_values, [32](#)
simulate.gam, [32](#)
simulate.gamm(simulate.gam), [32](#)
simulate.scam(simulate.gam), [32](#)
smallAges, [34](#)
smooth_dim, [35](#)
smooths, [34](#)

which_smooths, [35](#)

zooplankton, [36](#)