

Package ‘gratia’

January 24, 2021

Version 0.5.1

Date 2021-01-23

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

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

Depends R (>= 3.6.0)

Imports mgcv, ggplot2, tibble, dplyr, tidyr, rlang, cowplot,
patchwork, vctrs, grid, mvnfast, purrr, stats, tools, grDevices

Suggests gamm4, 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 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Gavin L. Simpson [aut, cre] (<<https://orcid.org/0000-0002-9084-8413>>),
Henrik Singmann [ctb] (<<https://orcid.org/0000-0002-4842-3657>>)

Repository CRAN

Date/Publication 2021-01-24 00:20:02 UTC

R topics documented:

add_constant	3
add_fitted	4

add_fitted.gam	4
add_partial_residuals	5
add_residuals	6
add_residuals.gam	7
appraise	8
basis	9
bird_move	10
check_user_select_smooths	11
coef.scam	11
confint.fderiv	12
confint.gam	13
data_sim	15
data_slice	16
derivatives	17
difference_smooths	19
draw	20
draw.derivatives	21
draw.difference_smooth	22
draw.evaluated_smooth	23
draw.gam	27
draw.mgcv_smooth	30
draw.penalty_df	31
draw.smooth_samples	32
evaluate_smooth	34
eval_smooth	36
family.gam	38
fderiv	38
fitted_samples	40
fix_offset	42
get_by_smooth	43
get_smooth	44
get_smooths_by_id	44
gss_vocab	45
is_by_smooth	45
is_factor_term	46
is_mgcv_smooth	47
is_offset	47
link	48
load_mgcv	50
n_smooths	50
observed_fitted_plot	51
parametric_terms	51
partial_residuals	52
penalty	53
posterior_samples	55
qq_plot	56
ref_sims	58
rep_first_factor_value	58

residuals_hist_plot	59
residuals_linpred_plot	59
seq_min_max	60
seq_min_max_eps	61
shift_values	62
simulate.gam	62
smallAges	63
smooths	64
smooth_coefs	65
smooth_dim	65
smooth_estimates	66
smooth_samples	67
term_variables	69
tidy_basis	70
transform_fun	71
vars_from_label	71
which_smooths	72
zooplankton	73

Index	74
--------------	-----------

add_constant	<i>Add a constant to estimated values</i>
--------------	---

Description

Add a constant to estimated values

Usage

```
add_constant(object, constant = NULL, ...)
```

```
## S3 method for class 'evaluated_smooth'
add_constant(object, constant = NULL, ...)
```

```
## S3 method for class 'evaluated_parametric_term'
add_constant(object, constant = NULL, ...)
```

Arguments

object	a object to add a constant to.
constant	the constant to add.
...	additional arguments passed to methods.

Value

Returns object but with the estimate shifted by the addition of the supplied constant.

Author(s)

Gavin L. Simpson

add_fitted	<i>Add fitted values from a model to a data frame</i>
------------	---

Description

Add fitted values from a model to a data frame

Usage

```
add_fitted(data, model, value = ".value", ...)
```

Arguments

data	a data frame containing values for the variables used to fit the model. Passed to <code>stats::predict()</code> as newdata.
model	a fitted model for which a <code>stats::predict()</code> method is available. S3 method dispatch is performed on the model argument.
value	character; the name of the variable in which model predictions will be stored.
...	additional arguments passed to methods.

Value

A data frame (tibble) formed from data and fitted values from model.

add_fitted.gam	<i>Add fitted values from a GAM to a data frame</i>
----------------	---

Description

Add fitted values from a GAM to a data frame

Usage

```
## S3 method for class 'gam'
add_fitted(data, model, value = ".value", type = "response", prefix = ".", ...)
```

Arguments

data	a data frame containing values for the variables used to fit the model. Passed to <code>stats::predict()</code> as <code>newdata</code> .
model	a fitted model for which a <code>stats::predict()</code> method is available. S3 method dispatch is performed on the <code>model</code> argument.
value	character; the name of the variable in which model predictions will be stored.
type	character; the type of predictions to return. See <code>mgcv::predict.gam()</code> for options.
prefix	character; string to prepend to names of predicted values when <code>type</code> is "terms", "iterms", "lpmatrix". These prediction types result in a matrix of values being returned. <code>prefix</code> will be prepended to each of the names of columns returned by such prediction types.
...	additional arguments passed to <code>mgcv::predict.gam()</code> .

Value

A data frame (tibble) formed from `data` and predictions from `model`.

Examples

```
load_mgcv()

df <- gamSim(eg = 1, verbose = FALSE)
df <- df[, c("y", "x0", "x1", "x2", "x3")]
m <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = df, method = 'REML')

##
add_fitted(df, m)

## with type = "terms" or "iterms"
add_fitted(df, m, type = "terms")
```

add_partial_residuals *Add partial residuals*

Description

Add partial residuals

Usage

```
add_partial_residuals(data, model, ...)

## S3 method for class 'gam'
add_partial_residuals(data, model, select = NULL, partial_match = FALSE, ...)
```

Arguments

data	a data frame containing values for the variables used to fit the model. Passed to <code>stats::residuals()</code> as <code>newdata</code> .
model	a fitted model for which a <code>stats::residuals()</code> method is available. S3 method dispatch is performed on the <code>model</code> argument.
...	arguments passed to other methods.
select	character, logical, or numeric; which smooths to plot. If <code>NULL</code> , the default, then all model smooths are drawn. Numeric <code>select</code> indexes the smooths in the order they are specified in the formula and stored in object. Character <code>select</code> matches the labels for smooths as shown for example in the output from <code>summary(object)</code> . Logical <code>select</code> operates as per numeric <code>select</code> in the order that smooths are stored.
partial_match	logical; should smooths be selected by partial matches with <code>select</code> ? If <code>TRUE</code> , <code>select</code> can only be a single string to match against.

Examples

```
load_mgcv()

df <- data_sim("eg1", seed = 1)
df <- df[, c("y", "x0", "x1", "x2", "x3")]
m <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = df, method = 'REML')

## add partial residuals
add_partial_residuals(df, m)

## add partial residuals for selected smooths
add_partial_residuals(df, m, select = "s(x0)")
```

add_residuals	<i>Add residuals from a model to a data frame</i>
---------------	---

Description

Add residuals from a model to a data frame

Usage

```
add_residuals(data, model, value = ".residual", ...)
```

Arguments

data	a data frame containing values for the variables used to fit the model. Passed to <code>stats::residuals()</code> as <code>newdata</code> .
model	a fitted model for which a <code>stats::residuals()</code> method is available. S3 method dispatch is performed on the <code>model</code> argument.

value character; the name of the variable in which model residuals will be stored.
 ... additional arguments passed to methods.

Value

A data frame (tibble) formed from data and residuals from model.

add_residuals.gam	<i>Add residuals from a GAM to a data frame</i>
-------------------	---

Description

Add residuals from a GAM to a data frame

Usage

```
## S3 method for class 'gam'
add_residuals(data, model, value = ".residual", type = "deviance", ...)
```

Arguments

data a data frame containing values for the variables used to fit the model. Passed to `stats::predict()` as newdata.

model a fitted model for which a `stats::predict()` method is available. S3 method dispatch is performed on the model argument.

value character; the name of the variable in which model predictions will be stored.

type character; the type of residuals to return. See `mgcv::residuals.gam()` for options.

... additional arguments passed to `mgcv::residuals.gam()`.

Value

A data frame (tibble) formed from data and residuals from model.

Examples

```
load_mgcv()

df <- gamSim(eg = 1, verbose = FALSE)
df <- df[, c("y", "x0", "x1", "x2", "x3")]
m <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = df, method = 'REML')

##
add_residuals(df, m)
```

appraise

*Model diagnostic plots***Description**

Model diagnostic plots

Usage

```
appraise(model, ...)

## S3 method for class 'gam'
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,
  ci_col = "black",
  ci_alpha = 0.2,
  point_col = "black",
  point_alpha = 1,
  line_col = "red",
  ...
)

## S3 method for class 'lm'
appraise(model, ...)
```

Arguments

model	a fitted model. Currently only class "gam".
...	arguments passed to <code>cowplot::plot_grid()</code> , except for align and axis, which are set internally.
method	character; method used to generate theoretical quantiles.
n_uniform	numeric; number of times to randomize uniform quantiles in the direct computation method (<code>method = "direct"</code>) for QQ plots.
n_simulate	numeric; number of data sets to simulate from the estimated model when using the simulation method (<code>method = "simulate"</code>) 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.

<code>ncol</code>	numeric; number of columns to draw plots in. See <code>cowplot::plot_grid()</code> .
<code>level</code>	numeric; the coverage level for QQ plot reference intervals. Must be strictly $0 < \text{level} < 1$. Only used with <code>method = "simulate"</code> .
<code>ci_alpha, ci_col</code>	numeric; the level of alpha transparency for the QQ plot reference interval when <code>method = "simulate"</code> , or points drawn in plots.
<code>point_col, point_alpha</code>	colour and transparency used to draw points in the plots. See <code>graphics::par()</code> section Color Specification . This is passed to the individual plotting functions, and therefore affects the points of all plots.
<code>line_col</code>	colour specification for the 1:1 line in the QQ plot and the reference line in the residuals vs linear predictor plot.

See Also

The plots are produced by functions `qq_plot()`, `residuals_linpred_plot()`, `residuals_hist_plot()`, and `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
appraise(mod, point_col = "steelblue", point_alpha = 0.4)
```

basis	<i>Basis expansions for smooths</i>
-------	-------------------------------------

Description

Creates a basis expansion from a definition of a smoother using the syntax of *mgcv*'s smooths via `mgcv::s()`, `mgcv::te()`, `mgcv::ti()`, and `mgcv::t2()`.

Usage

```
basis(smooth, data, knots = NULL, constraints = FALSE, ...)
```

Arguments

<code>smooth</code>	a smooth specification, the result of a call to one of <code>mgcv::s()</code> , <code>mgcv::te()</code> , <code>mgcv::ti()</code> , or <code>mgcv::t2()</code> .
<code>data</code>	a data frame containing the variables used in <code>smooth</code> .
<code>knots</code>	a list or data frame with named components containing knots locations. Names must match the covariates for which the basis is required. See <code>mgcv::smoothCon()</code> .

constraints logical; should identifiability constraints be applied to the smooth basis. See argument `absorb.cons` in `mgcv::smoothCon()`.

... other arguments passed to `mgcv::smoothCon()`.

Value

A tibble.

Author(s)

Gavin L. Simpson

Examples

```
load_mgcv()

df <- gamSim(4, n = 400, verbose = FALSE)

bf <- basis(s(x0), data = df)
bf <- basis(s(x2, by = fac, bs = 'bs'), data = df, constraints = TRUE)
```

bird_move

Simulated bird migration data

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

Select smooths based on user's choices

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,
  ncores = 1L,
  ...
)
```

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.
ncores	number of cores for generating random variables from a multivariate normal distribution. Passed to <code>mvnfast::rmvn()</code> . Parallelization will take place only if OpenMP is supported (but appears to work on Windows with current R).
...	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

```

load_mgcv()

dat <- gamSim(1, n = 1000, 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
x1.sint <- confint(fd, parm = "x1", type = "simultaneous", nsim = 2500)
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,
  ncores = 1,
  partial_match = FALSE,
  ...
)

```

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

## S3 method for class 'list'
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.
ncores	number of cores for generating random variables from a multivariate normal distribution. Passed to <code>mvnfast::rmvn()</code> . Parallelization will take place only if OpenMP is supported (but appears to work on Windows with current R).
partial_match	logical; should matching parm use a partial match or an exact match? Can only be used if <code>length(parm)</code> is 1.
...	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 $100 * \text{level}\%$ confidence interval.

Author(s)

Gavin L. Simpson

Examples

```
load_mgcv()

dat <- gamSim(1, n = 500, 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 = "s(x1)", type = "confidence")
ci

## simultaneous interval for smooth term of x1

si <- confint(mod, parm = "s(x1)", type = "simultaneous", nsim = 100)
si
```

data_sim

Simulate example data for fitting GAMs

Description

A tidy reimplementation of the functions implemented in `mgcv::gamSim()` that can be used to fit GAMs. An new feature is that the sampling distribution can be applied to all the example types.

Usage

```
data_sim(
  model = "eg1",
  n = 400,
  scale = 2,
  dist = c("normal", "poisson", "binary"),
  seed = NULL
)
```

Arguments

model	character; either "egX" where X is an integer 1:7, or the name of a model. See Details for possible options.
n	numeric; the number of observations to simulate.
scale	numeric; the level of noise to use.
dist	character; a sampling distribution for the response variable.
seed	numeric; the seed for the random number generator. Passed to <code>base::set.seed()</code> .

Examples

```
data_sim("eg1")
```

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

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	<i>Derivatives of estimated smooths via finite differences</i>
-------------	--

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,
  ncores = 1,
  ...
)
```

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.

<code>newdata</code>	a data frame containing the values of the model covariates at which to evaluate the first derivatives of the smooths.
<code>order</code>	numeric; the order of derivative.
<code>type</code>	character; the type of finite difference used. One of "forward", "backward", or "central".
<code>n</code>	numeric; the number of points to evaluate the derivative at.
<code>eps</code>	numeric; the finite difference.
<code>interval</code>	character; the type of interval to compute. One of "confidence" for point-wise intervals, or "simultaneous" for simultaneous intervals.
<code>n_sim</code>	integer; the number of simulations used in computing the simultaneous intervals.
<code>level</code>	numeric; $0 < \text{level} < 1$; the confidence level of the point-wise or simultaneous interval. The default is 0.95 for a 95% interval.
<code>unconditional</code>	logical; use smoothness selection-corrected Bayesian covariance matrix?
<code>frequentist</code>	logical; use the frequentist covariance matrix?
<code>offset</code>	numeric; a value to use for any offset term
<code>ncores</code>	number of cores for generating random variables from a multivariate normal distribution. Passed to <code>mvnfast::rmvn()</code> . Parallelization will take place only if OpenMP is supported (but appears to work on Windows with current R).

Value

A tibble, currently with the following variables:

- `smooth`: the smooth each row refers to,
- `var`: the name of the variable involved in the smooth,
- `data`: values of `var` at which the derivative was evaluated,
- `derivative`: the estimated derivative,
- `se`: the standard error of the estimated derivative,
- `crit`: the critical value such that $\text{derivative} \pm (\text{crit} * \text{se})$ gives the upper and lower bounds of the requested confidence or simultaneous interval (given `level`),
- `lower`: the lower bound of the confidence or simultaneous interval,
- `upper`: the upper bound of the confidence or simultaneous interval.

Author(s)

Gavin L. Simpson

Examples

```
load_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 derivatives of all smooths using central finite differences
derivatives(mod, type = "central")
```

difference_smooths *Differences of factor smooth interactions*

Description

Differences of factor smooth interactions

Usage

```
difference_smooths(model, ...)
```

```
## S3 method for class 'gam'
difference_smooths(
  model,
  smooth,
  n = 100,
  ci_level = 0.95,
  newdata = NULL,
  unconditional = FALSE,
  frequentist = FALSE,
  ...
)
```

Arguments

model	A fitted model.
...	arguments passed to other methods.
smooth	character; which smooth to compute differences for.
n	numeric; the number of points at which to evaluate the difference between pairs of smooths.
ci_level	numeric between 0 and 1; the coverage of credible interval.
newdata	data frame of locations at which to evaluate the difference between smooths.
unconditional	logical; account for smoothness selection in the model?
frequentist	logical; use the frequentist covariance matrix?

Examples

```
load_mgcv()

df <- data_sim("eg4")
m <- gam(y ~ fac + s(x2, by = fac) + s(x0), data = df, method = "REML")

difference_smooths(m, smooth = "s(x2)")
```

draw

Generic plotting via ggplot2

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 align in <code>cowplot::plot_grid()</code> . Defaults to "hv" so that plots are nicely aligned.
axis	character; see argument axis in <code>cowplot::plot_grid()</code> . Defaults to "lrtb" 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

```
load_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.difference_smooth
```

Plot differences of smooths

Description

Plot differences of smooths

Usage

```

## S3 method for class 'difference_smooth'
draw(
  object,
  select = NULL,
  rug = FALSE,
  ref_line = FALSE,
  contour = FALSE,
  contour_col = "black",
  n_contour = NULL,
  ci_alpha = 0.2,
  ci_colour = "black",
  line_col = "steelblue",
  scales = c("free", "fixed"),
  ncol = NULL,
  nrow = NULL,
  guides = "keep",
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ...
)

```

Arguments

`object` a fitted GAM, the result of a call to `mgcv::gam()`.

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.
rug	logical;
ref_line	logical;
contour	logical;
contour_col	colour specification for contour lines.
n_contour	numeric; the number of contour bins. Will result in <code>n_contour - 1</code> contour lines being drawn. See <code>ggplot2::geom_contour()</code> .
ci_alpha	numeric; alpha transparency for confidence or simultaneous interval.
ci_colour	colour specification for the confidence/credible intervals band.
line_col	colour specification for drawing lines
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.
ncol, nrow	numeric; the numbers of rows and columns over which to spread the plots
guides	character; one of "keep" (the default), "collect", or "auto". Passed to <code>patchwork::plot_layout()</code>
xlab, ylab, title, subtitle, caption	character; labels with which to annotate plots
...	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

```
load_mgcv()

df <- data_sim("eg4", seed = 42)
m <- gam(y ~ fac + s(x2, by = fac) + s(x0), data = df, method = "REML")

diffs <- difference_smooths(m, smooth = "s(x2)")
draw(diffs)
```

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,  
  ci_level = 0.95,  
  constant = NULL,  
  fun = NULL,  
  xlab,  
  ylab,  
  title = NULL,  
  subtitle = NULL,  
  caption = NULL,  
  partial_residuals = NULL,  
  response_range = NULL,  
  ...  
)  
  
## S3 method for class 'evaluated_2d_smooth'  
draw(  
  object,  
  show = c("estimate", "se"),  
  contour = TRUE,  
  contour_col = "black",  
  n_contour = NULL,  
  constant = NULL,  
  fun = NULL,  
  xlab,  
  ylab,  
  title = NULL,  
  subtitle = NULL,  
  caption = NULL,  
  response_range = NULL,  
  continuous_fill = NULL,  
  ...  
)  
  
## S3 method for class 'evaluated_re_smooth'  
draw(  
  object,  
  qq_line = TRUE,  
  constant = NULL,  
  fun = NULL,  
  xlab,  
  ylab,  
  title = NULL,  
  subtitle = NULL,  
  caption = NULL,
```



```

    response_range = NULL,
    ...
)

## S3 method for class 'evaluated_fs_smooth'
draw(
  object,
  rug = NULL,
  constant = NULL,
  fun = NULL,
  xlab,
  ylab,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  response_range = NULL,
  discrete_colour = NULL,
  ...
)

## S3 method for class 'evaluated_parametric_term'
draw(
  object,
  ci_level = 0.95,
  constant = NULL,
  fun = NULL,
  xlab,
  ylab,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  rug = TRUE,
  position = "identity",
  response_range = NULL,
  ...
)

```

Arguments

object	an object, the result of a call to <code>evaluate_smooth()</code> .
rug	For <code>evaluate_smooth()</code> , a numeric vector of values for the location of data on the x axis. The default of <code>NULL</code> results in no rug plot being drawn. For <code>evaluate_parametric_terms()</code> , a logical to indicate if a rug plot should be drawn.
ci_level	numeric between 0 and 1; the coverage of credible interval.
constant	numeric; a constant to add to the estimated values of the smooth. <code>constant</code> , if supplied, will be added to the estimated value before the confidence band is computed.

<code>fun</code>	function; a function that will be applied to the estimated values and confidence interval before plotting. Can be a function or the name of a function. Function <code>fun</code> will be applied after adding any constant, if provided.
<code>xlab</code>	character or expression; the label for the x axis. If not supplied, a suitable label will be generated from object.
<code>ylab</code>	character or expression; the label for the y axis. If not supplied, a suitable label will be generated from object.
<code>title</code>	character or expression; the title for the plot. See <code>ggplot2::labs()</code> .
<code>subtitle</code>	character or expression; the subtitle for the plot. See <code>ggplot2::labs()</code> .
<code>caption</code>	character or expression; the plot caption. See <code>ggplot2::labs()</code> .
<code>partial_residuals</code>	data frame; partial residuals and data values if partial residuals are drawn. Should have names <code>..p_resid</code> and <code>..orig_x</code> if supplied.
<code>response_range</code>	numeric; a vector of two values giving the range of response data for the guide. Used to fix plots to a common scale/range. Ignored if <code>show</code> is set to "se".
<code>...</code>	arguments passed to other methods.
<code>show</code>	character; plot the estimated smooth ("estimate") or its standard error ("se").
<code>contour</code>	logical; should contours be draw on the plot using <code>ggplot2::geom_contour()</code> .
<code>contour_col</code>	colour specification for contour lines.
<code>n_contour</code>	numeric; the number of contour bins. Will result in <code>n_contour - 1</code> contour lines being drawn. See <code>ggplot2::geom_contour()</code> .
<code>continuous_fill</code>	suitable scale used for the filled surface. If NULL, the default used is <code>scale_fill_distiller(palette = "RdBu", type = "div")</code> .
<code>qq_line</code>	logical; draw a reference line through the lower and upper theoretical quartiles.
<code>discrete_colour</code>	an appropriate discrete colour scale from <code>ggplot2</code> . The scale will need to be able to provide as many colours as there are levels in the factor variable involved in the smooth. Suitable alternatives include <code>ggplot2::scale_colour_viridis_d()</code> .
<code>position</code>	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

```

load_mgcv()

dat <- data_sim("eg1", n = 400, dist = "normal", scale = 2, seed = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

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

## supply constant to shift y axis scale
draw(sm, constant = coef(m1)[1])

dat <- data_sim("eg2", n = 1000, dist = "normal", scale = 1, seed = 2)
m2 <- gam(y ~ s(x, z, k = 40), data = dat, 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,
  residuals = FALSE,
  scales = c("free", "fixed"),
  align = "hv",
  axis = "lrtb",
  ci_level = 0.95,
  n = 100,
  unconditional = FALSE,
  overall_uncertainty = TRUE,
  constant = NULL,
  fun = NULL,
  dist = 0.1,
  rug = TRUE,
  contour = TRUE,
  contour_col = "black",
  n_contour = NULL,

```

```

    partial_match = FALSE,
    discrete_colour = NULL,
    continuous_colour = NULL,
    continuous_fill = NULL,
    ...
)

```

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.
residuals	logical; should partial residuals for a smooth be drawn? Ignored for anything but a simple univariate smooth.
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 align in <code>cowplot::plot_grid()</code> . Defaults to "hv" so that plots are nicely aligned.
axis	character; see argument axis in <code>cowplot::plot_grid()</code> . Defaults to "lrtb" so that plots are nicely aligned.
ci_level	numeric between 0 and 1; the coverage of credible interval.
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?
constant	numeric; a constant to add to the estimated values of the smooth. constant, if supplied, will be added to the estimated value before the confidence band is computed.
fun	function; a function that will be applied to the estimated values and confidence interval before plotting. Can be a function or the name of a function. Function fun will be applied after adding any constant, if provided.
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.
<code>rug</code>	logical; draw a rug plot at the bottom of each plot?
<code>contour</code>	logical; should contours be drawn on the plot using <code>ggplot2::geom_contour()</code> .
<code>contour_col</code>	colour specification for contour lines.
<code>n_contour</code>	numeric; the number of contour bins. Will result in <code>n_contour - 1</code> contour lines being drawn. See <code>ggplot2::geom_contour()</code> .
<code>partial_match</code>	logical; should smooths be selected by partial matches with <code>select</code> ? If <code>TRUE</code> , <code>select</code> can only be a single string to match against.
<code>discrete_colour</code> , <code>continuous_colour</code> , <code>continuous_fill</code>	suitable scales for the types of data.
<code>...</code>	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

```
load_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)

## can add partial residuals
draw(m1, residuals = TRUE)

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

## change the number of contours drawn and the fill scale used for
## the surface
draw(m2, n_contour = 5,
```

```
continuous_fill = ggplot2::scale_fill_distiller(palette = "Spectral",
                                                type = "div")
```

draw.mgcv_smooth *Plot basis functions*

Description

Plots basis functions using ggplot2

Usage

```
## S3 method for class 'mgcv_smooth'
draw(
  object,
  legend = FALSE,
  use_facets = TRUE,
  labeller = NULL,
  xlab,
  ylab,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ...
)
```

Arguments

object	an object, the result of a call to basis() .
legend	logical; should a legend be drawn to indicate basis functions?
use_facets	logical; for factor by smooths, use facets to show the basis functions for each level of the factor? If FALSE, a separate ggplot object will be created for each level and combined using cowplot::plot_grid() . Currently ignored.
labeller	a labeller function with which to label facets. The default is to use ggplot2::label_both() .
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. Not used by this method.

Value

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

Author(s)

Gavin L. Simpson

Examples

```
load_mgcv()

df <- gamSim(4, n = 400, verbose = FALSE)

bf <- basis(s(x0), data = df)
draw(bf)

bf <- basis(s(x2, by = fac, bs = 'bs'), data = df)
draw(bf)
```

draw.penalty_df

*Display penalty matrices of smooths using ggplot***Description**

Displays the penalty matrices of smooths as a heatmap using ggplot

Usage

```
## S3 method for class 'penalty_df'
draw(
  object,
  normalize = FALSE,
  continuous_fill = NULL,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  ncol = NULL,
  nrow = NULL,
  guides = "keep",
  ...
)
```

Arguments

object	an object, the result of a call to <code>evaluate_smooth()</code> .
normalize	logical; normalize the penalty to the range -1, 1?
continuous_fill	suitable scale used for the filled surface. If NULL, the default used is <code>scale_fill_distiller(palette = "RdBu", type = "div")</code> .

xlab	character or expression; the label for the x axis. If not supplied, no axis label will be drawn. May be a vector, one per penalty.
ylab	character or expression; the label for the y axis. If not supplied, no axis label will be drawn. May be a vector, one per penalty.
title	character or expression; the title for the plot. See <code>ggplot2::labs()</code> . May be a vector, one per penalty.
subtitle	character or expression; the subtitle for the plot. See <code>ggplot2::labs()</code> . May be a vector, one per penalty.
caption	character or expression; the plot caption. See <code>ggplot2::labs()</code> . May be a vector, one per penalty.
ncol, nrow	numeric; the numbers of rows and columns over which to spread the plots
guides	character; one of "keep" (the default), "collect", or "auto". Passed to <code>patchwork::plot_layout()</code>
...	arguments passed to other methods.

Examples

```
load_mgcv()
dat <- data_sim("eg4", n = 400, seed = 42)
m <- gam(y ~ s(x0) + s(x1, bs = 'cr') + s(x2, bs = 'bs', by = fac),
         data = dat, method = "REML")

## produce a multi-panel plot of all penalties
draw(penalty(m))

# for a specific smooth
draw(penalty(m, smooth = "s(x2):fac1"))
```

draw.smooth_samples *Plot posterior smooths*

Description

Plot posterior smooths

Usage

```
## S3 method for class 'smooth_samples'
draw(
  object,
  select = NULL,
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
  alpha = 1,
```



```

  colour = "black",
  scales = c("free", "fixed"),
  align = "hv",
  axis = "lrtb",
  rug = TRUE,
  partial_match = FALSE,
  ...
)

```

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.
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 <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; alpha transparency for confidence or simultaneous interval.
colour	The colour to use to draw the posterior smooths. Passed to <code>ggplot2::geom_line()</code> as argument colour.
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 align in <code>cowplot::plot_grid()</code> . Defaults to "hv" so that plots are nicely aligned.
axis	character; see argument axis in <code>cowplot::plot_grid()</code> . Defaults to "lrtb" so that plots are nicely aligned.
rug	logical; draw a rug plot at the bottom of each plot?
partial_match	logical; should smooths be selected by partial matches with select? If TRUE, select can only be a single string to match against.
...	arguments to be passed to <code>cowplot::plot_grid()</code> .

Author(s)

Gavin L. Simpson

Examples

```

load_mgcv()

dat1 <- gamSim(1, n = 400, dist = "normal", scale = 2, verbose = FALSE)
## a single smooth GAM
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat1, method = "REML")
## posterior smooths from m1
sm1 <- smooth_samples(m1, n = 15, seed = 23478)
## plot
draw(sm1, alpha = 0.7)

dat2 <- gamSim(4, verbose = FALSE)
## a multi-smooth GAM with a factor-by smooth
m2 <- gam(y ~ fac + s(x2, by = fac) + s(x0), data = dat2, method = "REML")
## posterior smooths from m1
sm2 <- smooth_samples(m2, n = 15, seed = 23478)
## plot, this time selecting only the factor-by smooth
draw(sm2, select = "s(x2)", partial_match = TRUE, alpha = 0.7)

```

evaluate_smooth

Evaluate a smooth

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, ...)

## S3 method for class 'list'

```

```

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

```

load_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

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)

```

`eval_smooth`*S3 methods to evaluate individual smooths*

Description

S3 methods to evaluate individual smooths

Usage

```
eval_smooth(smooth, ...)
```

```
## S3 method for class 'mgcv.smooth'
```

```
eval_smooth(  
  smooth,  
  model,  
  n = 100,  
  data = NULL,  
  unconditional = FALSE,  
  overall_uncertainty = TRUE,  
  ...  
)
```

```
## S3 method for class 'fs.interaction'
```

```
eval_smooth(  
  smooth,  
  model,  
  n = 100,  
  data = NULL,  
  unconditional = FALSE,  
  overall_uncertainty = TRUE,  
  ...  
)
```

```
## S3 method for class 'random.effect'
```

```
eval_smooth(  
  smooth,  
  model,  
  n = 100,  
  data = NULL,  
  unconditional = FALSE,  
  overall_uncertainty = TRUE,  
  ...  
)
```

```
## S3 method for class 'mrf.smooth'
```

```
eval_smooth(  
  smooth,
```

```

    model,
    n = 100,
    data = NULL,
    unconditional = FALSE,
    overall_uncertainty = TRUE,
    ...
)

## S3 method for class 't2.smooth'
eval_smooth(
  smooth,
  model,
  n = 100,
  data = NULL,
  unconditional = FALSE,
  overall_uncertainty = TRUE,
  dist = 0.1,
  ...
)

## S3 method for class 'tensor.smooth'
eval_smooth(
  smooth,
  model,
  n = 100,
  data = NULL,
  unconditional = FALSE,
  overall_uncertainty = TRUE,
  ...
)

```

Arguments

smooth	currently an object that inherits from class <code>mgcv.smooth</code> .
...	arguments assed to other methods
model	a fitted model; currently only <code>mgcv::gam()</code> and <code>mgcv::bam()</code> models are supported.
n	numeric; the number of points over the range of the covariate at which to evaluate the smooth.
data	an optional data frame of values to evaluate smooth at.
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 `mgcv::exclude.too.far()` for further details.

family.gam *Extract family objects from models*

Description

Provides a `stats::family()` method for a range of GAM objects.

Usage

```
## S3 method for class 'gam'
family(object, ...)
```

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

```
## S3 method for class 'bam'
family(object, ...)
```

```
## S3 method for class 'list'
family(object, ...)
```

Arguments

`object` a fitted model. Models fitted by `mgcv::gam()`, `mgcv::bam()`, `mgcv::gamm()`, and `gamm4::gamm4()` are currently supported.

`...` arguments passed to other methods.

fderiv *First derivatives of fitted GAM functions*

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

```

load_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")
```

fitted_samples	<i>Draw fitted values from the posterior distribution</i>
----------------	---

Description

Expectations (fitted values) of the response drawn from the posterior distribution of fitted model, created via `simulate()` (e.g. `simulate.gam()`) and returned in a tidy, long, format.

Predicted values of the response drawn from the posterior distribution of fitted model, created via `simulate()` (e.g. `simulate.gam()`) and returned in a tidy, long, format.

Usage

```
fitted_samples(model, ...)

## S3 method for class 'gam'
fitted_samples(
  model,
  n = 1,
  newdata,
  seed,
  scale = c("response", "linear_predictor"),
  freq = FALSE,
  unconditional = FALSE,
  ncores = 1L,
  ...
)

predicted_samples(model, ...)

## S3 method for class 'gam'
predicted_samples(
  model,
  n = 1,
  newdata = NULL,
```



```

    seed = NULL,
    weights = NULL,
    ...
  )

```

Arguments

model	a fitted model of the supported types
...	arguments passed to other methods. For <code>fitted_samples()</code> , these are passed on to <code>predict.gam()</code> .
n	numeric; the number of posterior samples to return.
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 <code>newdata</code> , if available in <code>model</code> .
seed	numeric; a random seed for the simulations.
scale	character;
freq	logical; TRUE to use the frequentist covariance matrix of the parameter estimators, FALSE to use 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 used, if available.
ncores	number of cores for generating random variables from a multivariate normal distribution. Passed to <code>mvnfast::rmvn()</code> . Parallelization will take place only if OpenMP is supported (but appears to work on Windows with current R).
weights	numeric; a vector of prior weights. If <code>newdata</code> is null then defaults to <code>object[["prior.weights"]]</code> , otherwise a vector of ones.

Value

A tibble (data frame) with 3 columns containing the posterior predicted values in long format. The columns are

- row (integer) the row of `newdata` that each posterior draw relates to,
- draw (integer) an index, in range `1:n`, indicating which draw each row relates to,
- response (numeric) the predicted response for the indicated row of `newdata`.

A tibble (data frame) with 3 columns containing the posterior predicted values in long format. The columns are

- row (integer) the row of `newdata` that each posterior draw relates to,
- draw (integer) an index, in range `1:n`, indicating which draw each row relates to,
- response (numeric) the predicted response for the indicated row of `newdata`.

Author(s)

Gavin L. Simpson

Gavin L. Simpson

Examples

```

load_mgcv()

dat <- gamSim(1, n = 1000, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

fitted_samples(m1, n = 5, seed = 42)

load_mgcv()

dat <- gamSim(1, n = 1000, dist = "normal", scale = 2)
m <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

predicted_samples(m, n = 5, seed = 42)

## Can pass arguments to predict.gam()
newd <- data.frame(x0 = runif(10), x1 = runif(10), x2 = runif(10),
                  x3 = runif(10))

## Exclude s(x2)
predicted_samples(m, n = 5, newd, exclude = "s(x2)", seed = 25)

## Exclude s(x1)
predicted_samples(m, n = 5, newd, exclude = "s(x1)", seed = 25)

## Select which terms --- result same as previous
predicted_samples(m, n = 5, newd, seed = 25,
                 terms = c("s(x0)", "s(x2)", "s(x3)"))

```

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

```
load_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.

gss_vocab	<i>Data from the General Social Survey (GSS) from the National Opinion Research Center of the University of Chicago</i>
-----------	---

Description

A subset of the data from the `carData::GSSvocab` dataset from the `carData` package, containing observations from 2016 only.

Format

A data frame with 1858 rows and 3 variables:

- `vocab`: numeric; the number of words out of 10 correct on a vocabulary test.
- `nativeBorn`: factor; Was the respondent born in the US? A factor with levels `no` and `yes`.
- `ageGroup`: factor; grouped age of the respondent with levels `18-29`, `30-39`, `40-49`, `50-59`, and `60+.`

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_term	<i>Is a model term a factor (categorical)?</i>
----------------	--

Description

Given the name (a term label) of a term in a model, identify if the term is a factor term or numeric. This is useful when considering interactions, where terms like `fac1:fac2` or `num1:fac1` may be requested by the user. Only for terms of the type `fac1:fac2` will this function return TRUE.

Usage

```
is_factor_term(object, term, ...)

## S3 method for class 'terms'
is_factor_term(object, term, ...)

## S3 method for class 'gam'
is_factor_term(object, term, ...)

## S3 method for class 'bam'
is_factor_term(object, term, ...)

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

## S3 method for class 'list'
is_factor_term(object, term, ...)
```

Arguments

<code>object</code>	an R object on which method dispatch is performed
<code>term</code>	character; the name of a model term, in the sense of <code>attr(terms(object), "term.labels")</code> . Currently not checked to see if the term exists in the model.
<code>...</code>	arguments passed to other methods.

Value

A logical: TRUE if and only if all variables involved in the term are factors, otherwise FALSE.

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

```
load_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)
```

`link`*Extract link and inverse link functions from models*

Description

Returns the link or its inverse from an estimated model, and provides a simple way to extract these functions from complex models with multiple links, such as location scale models.

Usage

```
link(object, ...)  
  
## S3 method for class 'family'  
link(  
  object,  
  parameter = c("location", "scale", "shape"),  
  which_eta = NULL,  
  ...  
)  
  
## S3 method for class 'gam'  
link(object, parameter = c("location", "scale", "shape"), ...)  
  
## S3 method for class 'bam'  
link(object, parameter = c("location", "scale", "shape"), ...)  
  
## S3 method for class 'gamm'  
link(object, ...)  
  
## S3 method for class 'glm'  
link(object, ...)  
  
## S3 method for class 'list'  
link(object, ...)  
  
inv_link(object, ...)  
  
## S3 method for class 'family'  
inv_link(  
  object,  
  parameter = c("location", "scale", "shape"),  
  which_eta = NULL,  
  ...  
)  
  
## S3 method for class 'gam'  
inv_link(object, parameter = c("location", "scale", "shape"), ...)
```



```
## S3 method for class 'bam'
inv_link(object, parameter = c("location", "scale", "shape"), ...)

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

## S3 method for class 'list'
inv_link(object, ...)

## S3 method for class 'glm'
inv_link(object, ...)
```

Arguments

object	a family object or a fitted model from which to extract the family object. Models fitted by <code>stats::glm()</code> , <code>mgcv::gam()</code> , <code>mgcv::bam()</code> , <code>mgcv::gamm()</code> , and <code>gamm4::gamm4()</code> are currently supported.
...	arguments passed to other methods.
parameter	character; which parameter of the distribution. Usually "location" but "scale" and "shape" may be provided for location scale models. Other options include "mu" and "sigma" as synonyms for "location" and "scale" respectively, "pi" for the zero-inflation term in <code>mgcv::ziplss()</code> , "power" for the <code>mgcv::twlss()</code> power parameter, and "xi", the shape parameter for <code>mgcv::gevlss()</code> .
which_eta	numeric; the linear predictor to extract for families <code>mgcv::mvn()</code> and <code>mgcv::multinom()</code> .

Author(s)

Gavin L. Simpson

Examples

```
load_mgcv()

link(gaussian())
link(nb())

inv_link(nb())

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

link(mod, parameter = "scale")
inv_link(mod, parameter = "scale")
```

load_mgcv	<i>Load mgcv quietly</i>
-----------	--------------------------

Description

Simple function that loads the *mgcv* package whilst suppressing the startup messages that it prints to the console.

Usage

```
load_mgcv()
```

Value

Returns a logical vectors invisibly, indicating whether the package was loaded or not.

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,
  point_col = "black",
  point_alpha = 1
)
```

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 x 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() .
point_col	colour used to draw points in the plots. See graphics::par() section Color Specification . This is passed to the individual plotting functions, and therefore affects the points of all plots.
point_alpha	numeric; alpha transparency for points in plots.

parametric_terms *Names of any parametric terms in a GAM*

Description

Names of any parametric 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.

partial_residuals	<i>Partial residuals</i>
-------------------	--------------------------

Description

Partial residuals

Usage

```

partial_residuals(object, ...)

## S3 method for class 'gam'
partial_residuals(object, select = NULL, partial_match = FALSE, ...)

```

Arguments

object	an R object, typically a model. Currently only objects of class "gam" (or that inherit from that class) are supported.
...	arguments passed to other methods.
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.
partial_match	logical; should smooths be selected by partial matches with select? If TRUE, select can only be a single string to match against.

Examples

```
## load mgcv
load_mgcv()

## example data - Gu & Wabha four term model
df <- data_sim("eg1", n = 400, seed = 42)
## fit the model
m <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = df, method = 'REML')

## extract partial residuals
partial_residuals(m)

## and for a select term
partial_residuals(m, select = "s(x2)")

## or with partial matching
partial_residuals(m, select = "x", partial_match = TRUE) # returns all
```

penalty

Extract and tidy penalty matrices

Description

Extract and tidy penalty matrices

Usage

```
penalty(object, ...)

## S3 method for class 'gam'
penalty(object, smooth = NULL, rescale = FALSE, ...)

## S3 method for class 'mgcv.smooth'
penalty(object, rescale = FALSE, ...)

## S3 method for class 'tensor.smooth'
penalty(object, margins = FALSE, ...)

## S3 method for class 't2.smooth'
penalty(object, margins = FALSE, ...)
```

Arguments

object a fitted GAM or a smooth.
... additional arguments passed to methods.

smooth	character; vector of smooths to extract penalty matrices for. If NULL, penalty matrices for all smooths in object are extracted.
rescale	logical; by default, <i>mgcv</i> will scale the penalty matrix for better performance in <code>mgcv::gamm()</code> . If <code>rescale</code> is TRUE, this scaling will be undone to put the penalty matrix back on the original scale.
margins	logical; extract the penalty matrices for the tensor product or the marginal smooths of the tensor product?

Value

A 'tibble' (data frame) of class `penalty_df` inheriting from `tbl_df`, with the following components:

- `smooth` - character; the label *mgcv* uses to refer to the smooth,
- `type` - character; the type of smooth,
- `penalty` - character; the label for the specific penalty. Some smooths have multiple penalty matrices, so the `penalty` component identifies the particular penalty matrix and uses the labelling that *mgcv* uses internally,
- `row` - character; a label of the form `fn` where `n` is an integer for the `n`th basis function, referencing the columns of the penalty matrix,
- `col` - character; a label of the form `fn` where `n` is an integer for the `n`th basis function, referencing the columns of the penalty matrix,
- `value` - double; the value of the penalty matrix for the combination of `row` and `col`,

Note

The `print()` method uses `base::zapsmall()` to turn very small numbers into 0s for display purposes only; the underlying values of the penalty matrix or matrices are not changed.

Author(s)

Gavin L. Simpson

Examples

```
load_mgcv()
dat <- data_sim("eg4", n = 400, seed = 42)
m <- gam(y ~ s(x0) + s(x1) + s(x2, by = fac),
         data = dat, method = "REML")
penalty(m)

# for a specific smooth
penalty(m, smooth = "s(x2):fac1")
```

posterior_samples *Draw samples from the posterior distribution of an estimated model*

Description

Draw samples from the posterior distribution of an estimated model

Usage

```
posterior_samples(model, ...)

## S3 method for class 'gam'
posterior_samples(
  model,
  n,
  newdata,
  seed,
  scale = c("response", "linear_predictor"),
  freq = FALSE,
  unconditional = FALSE,
  weights = NULL,
  ncores = 1L,
  ...
)
```

Arguments

model	a fitted model of the supported types
...	arguments passed to other methods. For <code>fitted_samples()</code> , these are passed on to <code>predict.gam()</code> .
n	numeric; the number of posterior samples to return.
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 model.
seed	numeric; a random seed for the simulations.
scale	character;
freq	logical; TRUE to use the frequentist covariance matrix of the parameter estimators, FALSE to use 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 used, 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.
ncores	number of cores for generating random variables from a multivariate normal distribution. Passed to <code>mvnfast::rmvn()</code> . Parallelization will take place only if OpenMP is supported (but appears to work on Windows with current R).

Value

A tibble (data frame) with 3 columns containing the posterior predicted values in long format. The columns are

- row (integer) the row of newdata that each posterior draw relates to,
- draw (integer) an index, in range 1:n, indicating which draw each row relates to,
- response (numeric) the predicted response for the indicated row of newdata.

Author(s)

Gavin L. Simpson

qq_plot

Quantile-quantile plot of model residuals

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,
  ci_col = "black",
  ci_alpha = 0.2,
  point_col = "black",
  point_alpha = 1,
  line_col = "red",
  ...
)

## S3 method for class 'glm'
qq_plot(model, ...)
```


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> .
ci_col, ci_alpha	fill colour and alpha transparency for the reference interval when method = "simulate".
point_col, point_alpha	colour and alpha transparency for points on the QQ plot.
line_col	colour used to draw the 1:1 reference line.

Examples

```
load_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", point_col = "steelblue",
```

```

        point_alpha = 0.4)

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

```

ref_sims	<i>Reference simulation data</i>
----------	----------------------------------

Description

A set of reference objects for testing `data_sim()`.

Format

A named list of simulated data sets created by `data_sim()`.

rep_first_factor_value	<i>Repeat the first level of a factor n times</i>
------------------------	---

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,
  point_col = "black",
  point_alpha = 1,
  line_col = "red"
)
```

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() .
point_col	colour used to draw points in the plots. See graphics::par() section Color Specification . This is passed to the individual plotting functions, and therefore affects the points of all plots.
point_alpha	numeric; alpha transparency for points in plots.
line_col	colour specification for 1:1 line.

seq_min_max

Create a sequence of evenly-spaced values

Description

For a continuous vector x , `seq_min_max()` creates a sequence of n evenly-spaced values over the range $\min(x)$ –

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 computing predicted values for the observation data for which simulated data are required, then generating random draws from the probability distribution used when fitting the model.

Usage

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

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

## S3 method for class 'scam'
simulate(object, nsim = 1, seed = NULL, newdata = NULL, 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.
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. <code>simulate.gam()</code> and <code>simulate.scam()</code> pass ... on to <code>predict.gam()</code> . As such you can pass additional arguments such as <code>terms</code> , <code>exclude</code> , to select which model terms are included in the predictions. This may be useful, for example, for excluding the effects of random effect terms.

Details

For `simulate.gam()` to function, the family component of the fitted model must contain, or be updateable to contain, the required random number generator. See `mgcv::fix.family.rd()`.

Value

(Currently) A matrix with `nsim` columns.

Author(s)

Gavin L. Simpson

Examples

```
load_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_coefs	<i>Indices of the parametric terms for a particular smooth</i>
--------------	--

Description

Returns a vector of indices of the parametric terms that represent the supplied smooth. Useful for extracting model coefficients and columns of their covariance matrix.

Usage

```
smooth_coefs(smooth)
```

Arguments

smooth an object that inherits from class `mgcv.smooth`

Value

A numeric vector of indices.

Author(s)

Gavin L. Simpson

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

smooth_estimates *New evaluate_smooth() alike*

Description

New evaluate_smooth() alike

Usage

```
smooth_estimates(object, ...)

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

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.
data	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 `mgcv::exclude.too.far()` for further details.

Value

A data frame (tibble), which is of class "smooth_estimates".

Examples

```
load_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 all smooths
smooth_estimates(m1)

## or selected smooths
smooth_estimates(m1, smooth = c("s(x0)", "s(x1)"))
```

<code>smooth_samples</code>	<i>Posterior draws for individual smooths</i>
-----------------------------	---

Description

Returns draws from the posterior distributions of smooth functions in a GAM. Useful, for example, for visualising the uncertainty in individual estimated functions.

Usage

```
smooth_samples(model, ...)

## S3 method for class 'gam'
smooth_samples(
  model,
  term = NULL,
  n = 1,
  newdata = NULL,
  seed = NULL,
  freq = FALSE,
  unconditional = FALSE,
  ncores = 1L,
```

```

  n_vals = 200,
  ...
)
```

Arguments

model	a fitted model of the supported types
...	arguments passed to other methods. For <code>fitted_samples()</code> , these are passed on to <code>predict.gam()</code> .
term	character; select which smooth's posterior to draw from. The default (NULL) means the posteriors of all smooths in <code>model</code> will be sampled from. If supplied, a character vector of requested terms.
n	numeric; the number of posterior samples to return.
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 <code>newdata</code> , if available in <code>model</code> .
seed	numeric; a random seed for the simulations.
freq	logical; TRUE to use the frequentist covariance matrix of the parameter estimators, FALSE to use 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 used, if available.
ncores	number of cores for generating random variables from a multivariate normal distribution. Passed to <code>mvnfast::rmvn()</code> . Parallelization will take place only if OpenMP is supported (but appears to work on Windows with current R).
n_vals	numeric; how many locations to evaluate the smooth at if <code>newdata</code> not supplied

Value

A tibble with additional classes "smooth_samples" and "posterior_samples".

For the "gam" method, the columns currently returned (not in this order) are:

- `smooth`; character vector. Indicates the smooth function for that particular draw,
- `term`; character vector. Similar to `smooth`, but will contain the full label for the smooth, to differentiate factor-by smooths for example.
- `by_variable`; character vector. If the smooth involves a by term, the by variable will be named here, `NA_character_ otherwise`.
- `row`; integer. A vector of values `seq_len(n_vals)`, repeated if `n > 1L`. Indexes the row in `newdata` for that particular draw.
- `draw`; integer. A vector of integer values indexing the particular posterior draw that each row belongs to.
- `value`; numeric. The value of smooth function for this posterior draw and covariate combination.

- `.xN`; numeric. A series of one or more columns containing data required for the smooth. `.x1` will always be present and contains the values of the covariate in the smooth. For example if smooth is $s(z)$ then `.x1` will contain the values of covariate z at which the smooth was evaluated. Further covariates for multi-dimensional thin plate splines (e.g. $s(x, z)$) or tensor product smooths (e.g. $te(x, z, a)$) will result in variables `.x1` and `.x2`, and `.x1`, `.x2`, and `.x3` respectively, with the number (1, 2, etc) representing the order in which the covariates were specified in the smooth.
- Additional columns will be present in the case of factor by smooths, which will contain the level for the factor named in `by_variable` for that particular posterior draw.

Warning

The set of variables returned and their order in the tibble is subject to change in future versions. Don't rely on position.

Author(s)

Gavin L. Simpson

Examples

```
load_mgcv()

dat <- gamSim(1, n = 1000, dist = "normal", scale = 2)
m1 <- gam(y ~ s(x0) + s(x1) + s(x2) + s(x3), data = dat, method = "REML")

smooth_samples(m1, term = "s(x0)", n = 5, seed = 42)

## A factor by example (with a spurious covariate x0)

dat <- gamSim(4)

## fit model...
m2 <- gam(y ~ fac + s(x2, by = fac) + s(x0), data = dat)
sms <- smooth_samples(m2, n = 5, seed = 42)
draw(sms)
```

term_variables

Names of variables involved in a specified model term

Description

Given the name (a term label) of a term in a model, returns the names of the variables involved in the term.

Usage

```
term_variables(object, term, ...)

## S3 method for class 'terms'
term_variables(object, term, ...)

## S3 method for class 'gam'
term_variables(object, term, ...)

## S3 method for class 'bam'
term_variables(object, term, ...)
```

Arguments

object	an R object on which method dispatch is performed
term	character; the name of a model term, in the sense of <code>attr(terms(object), "term.labels")</code> . Currently not checked to see if the term exists in the model.
...	arguments passed to other methods.

Value

A character vector of variable names.

tidy_basis	<i>A tidy basis representation of a smooth object</i>
------------	---

Description

Takes an object of class `mgcv.smooth` and returns a tidy representation of the basis.

Usage

```
tidy_basis(smooth, data)
```

Arguments

smooth	a smooth object.
data	a data frame containing the variables used in smooth.

Value

A tibble.

Author(s)

Gavin L. Simpson

transform_fun	<i>Transform estimated values and confidence intervals by applying a function</i>
---------------	---

Description

Transform estimated values and confidence intervals by applying a function

Usage

```
transform_fun(object, fun = NULL, ...)
```

```
## S3 method for class 'evaluated_smooth'
transform_fun(object, fun = NULL, ...)
```

```
## S3 method for class 'evaluated_parametric_term'
transform_fun(object, fun = NULL, ...)
```

Arguments

object	an object to apply the transform function to.
fun	the function to apply.
...	additional arguments passed to methods.

Value

Returns object but with the estimate and upper and lower values of the confidence interval transformed via the function.

Author(s)

Gavin L. Simpson

vars_from_label	<i>Returns names of variables from a smooth label</i>
-----------------	---

Description

Returns names of variables from a smooth label

Usage

```
vars_from_label(label)
```

Arguments

label character; a length 1 character vector containing the label of a smooth.

Examples

```
vars_from_label("s(x1)")
vars_from_label("t2(x1,x2,x3)")
```

which_smooths *Identify a smooth term by its label*

Description

Identify a smooth term by its 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

- * **data**
 - bird_move, 10
 - gss_vocab, 45
 - ref_sims, 58
 - smallAges, 63
 - zooplankton, 73

- add_constant, 3
- add_fitted, 4
- add_fitted.gam, 4
- add_partial_residuals, 5
- add_residuals, 6
- add_residuals.gam, 7
- appraise, 8

- base::set.seed(), 15
- base::zapsmall(), 54
- basis, 9
- basis(), 30
- bird_move, 10
- by_level(is_by_smooth), 45
- by_variable(is_by_smooth), 45

- check_user_select_smooths, 11
- coef.scam, 11
- confint.fderiv, 12
- confint.gam, 13
- confint.gamm(confint.gam), 13
- confint.list(confint.gam), 13
- cowplot::plot_grid(), 8, 9, 29, 30, 33

- data_sim, 15
- data_sim(), 58
- data_slice, 16
- derivatives, 17
- difference_smooths, 19
- draw, 20
- draw.derivatives, 21
- draw.difference_smooth, 22
- draw.evaluated_1d_smooth
 - (draw.evaluated_smooth), 23

- draw.evaluated_2d_smooth
 - (draw.evaluated_smooth), 23
- draw.evaluated_fs_smooth
 - (draw.evaluated_smooth), 23
- draw.evaluated_parametric_term
 - (draw.evaluated_smooth), 23
- draw.evaluated_re_smooth
 - (draw.evaluated_smooth), 23
- draw.evaluated_smooth, 23
- draw.gam, 27
- draw.mgcv_smooth, 30
- draw.penalty_df, 31
- draw.smooth_samples, 32

- eval_smooth, 36
- evaluate_parametric_term
 - (evaluate_smooth), 34
- evaluate_smooth, 34
- evaluate_smooth(), 25, 31

- family.bam(family.gam), 38
- family.gam, 38
- family.gamm(family.gam), 38
- family.list(family.gam), 38
- fderiv, 38
- fitted_samples, 40
- fix_offset, 42

- gamm4::gamm4(), 38, 49
- geom_rug(draw.evaluated_smooth), 23
- get_by_smooth, 43
- get_smooth, 44
- get_smooths_by_id, 44
- ggplot2::geom_contour(), 23, 26, 29
- ggplot2::geom_line(), 33
- ggplot2::ggplot(), 20, 26, 27, 29, 30
- ggplot2::label_both(), 30
- ggplot2::labs(), 26, 30, 32, 33, 51, 57, 59, 60
- ggplot2::scale_colour_viridis_d(), 26

graphics::par(), 9, 51, 60
 gss_vocab, 45

 inv_link(link), 48
 is_by_smooth, 45
 is_continuous_by_smooth(is_by_smooth), 45
 is_factor_by_smooth(is_by_smooth), 45
 is_factor_term, 46
 is_mgcv_smooth, 47
 is_mrf_smooth(is_mgcv_smooth), 47
 is_offset, 47

 link, 48
 load_mgcv, 50

 mgcv::bam(), 37, 38, 49, 50, 64
 mgcv::exclude.too.far(), 29, 35, 38, 67
 mgcv::fix.family.rd(), 63
 mgcv::gam, 63
 mgcv::gam(), 21, 22, 28, 33, 37–39, 49, 50, 64
 mgcv::gamm(), 38, 39, 49, 50, 54, 63, 64
 mgcv::gamSim(), 15
 mgcv::gevlss(), 49
 mgcv::multinom(), 49
 mgcv::mvn(), 49
 mgcv::predict.gam(), 5
 mgcv::residuals.gam(), 7
 mgcv::s(), 9
 mgcv::smoothCon(), 9, 10
 mgcv::t2(), 9
 mgcv::te(), 9
 mgcv::ti(), 9
 mgcv::twlss(), 49
 mgcv::ziplss(), 49
 mvnfast::rmvn(), 12, 14, 18, 41, 55, 68

 n_smooths, 50

 observed_fitted_plot, 51
 observed_fitted_plot(), 9

 parametric_terms, 51
 partial_residuals, 52
 patchwork::plot_layout(), 23, 32
 penalty, 53
 posterior_samples, 55
 predicted_samples(fitted_samples), 40

 qq_plot, 56

 qq_plot(), 9

 ref_sims, 58
 rep_first_factor_value, 58
 residuals_hist_plot, 59
 residuals_hist_plot(), 9
 residuals_linpred_plot, 59
 residuals_linpred_plot(), 9

 seq_min_max, 60
 seq_min_max_eps, 61
 shift_values, 62
 simulate.gam, 62
 simulate.gam(), 40
 simulate.gamm(simulate.gam), 62
 simulate.scam(simulate.gam), 62
 smallAges, 63
 smooth_coefs, 65
 smooth_dim, 65
 smooth_estimates, 66
 smooth_samples, 67
 smooths, 64
 stats::family(), 38
 stats::glm(), 49
 stats::predict(), 4, 5, 7
 stats::residuals(), 6

 term_variables, 69
 tidy_basis, 70
 transform_fun, 71

 vars_from_label, 71

 which_smooths, 72

 zooplankton, 73