Package 'gamfluence'

November 17, 2026

Type Package

Title Deletion Influence Diagnostics for 'mgcv' Additive Models

Version 0.1.1

Description For penalised additive (mixed) models fitted with gam() from 'mgcv', each site (cluster) is deleted in turn and the model is refitted twice, reusing the full-data model matrix and penalties: once with the estimated smoothing parameters (and negative binomial theta) held fixed, and once with them re-estimated by restricted maximum likelihood (REML). Comparing the changes in the other sites' fitted values shows how much of a site's influence appears only when the smoothing parameters re-tune. Includes a plot method and a dataset of coral trout counts on the Great Barrier Reef.

License GPL-3

URL https://fayesong-0223.github.io/gamfluence/, https://github.com/FayeSong-0223/gamfluence

BugReports https://github.com/FayeSong-0223/gamfluence/issues

Encoding UTF-8

Language en-GB

LazyData true

Depends R (>= 4.1.0)

Imports graphics, mgcv (>= 1.9), stats, utils

Suggests knitr, rmarkdown, testthat (>= 3.1.7)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.1

NeedsCompilation no

Author Ziqi Song [aut, cre]

Maintainer Ziqi Song <ziqisong030@gmail.com>

Repository CRAN

Date/Publication 2026-11-17 12:10:07 UTC

Contents

coral_trout 2
gam_influence 3
gam_influence-methods 5
plot.gam_influence 6

Index **8**

`coral_trout` *Coral trout counts on inshore reefs of the Great Barrier Reef*

Description

Counts of coral trout (*Plectropomus leopardus*) at 71 inshore reef sites in the Palm and Whitsunday island groups, surveyed six or seven times between 2007 and 2018 (467 site-surveys), with each site's management zone, wave exposure, and habitat and environmental covariates. It is the analysis dataset of the coral trout study cited below and is used in the vignette.

Usage

```
coral_trout
```

Format

A data frame with 467 rows and 12 columns:

SITE Site, a factor with 71 levels.

REGION Island group: `Palm` or `Whitsunday`.

YEAR Survey year.

NTR Management zone: `Fished`, or a no-take reserve zoned in 1987 (`NTR 1987`) or 2004 (`NTR 2004`). Constant within a site.

EXPOSURE Wave exposure: `Sheltered`, `Semi-Exposed` or `Exposed`. Constant within a site.

depth Survey depth, the `Corrected depth` column of the source. Constant within a site.

rugosity Reef structural complexity.

LHC Live hard coral cover (percent).

kd490 Water clarity: diffuse light attenuation at 490 nm (higher is more turbid).

maxDHW Thermal stress: maximum degree heating weeks.

Cyclone Cyclone exposure, as given in the source.

count Number of *P. leopardus* counted in the survey. The source gives densities; they are an exact multiple of the counts with a year-specific constant, so the counts were recovered from them.

Source

Derived from Australian Institute of Marine Science (AIMS). (2022). Spatio-temporal dynamics of coral reef fish assemblages on inshore reefs of the Great Barrier Reef. https://apps.aims.gov.au/metadata/view/814a0be3-ed85-4a43-87b7-59ece6eb6a05, accessed 02-Sep-2026. Licensed under Creative Commons Attribution 3.0 Australia (CC BY 3.0 AU). Based on Australian Institute of Marine Science data. The data were restricted to *P. leopardus* in the Palm and Whitsunday island groups from 2007 to 2018, and densities were converted back to counts. See `LICENSE.note` for the licence of this dataset.

References

Song, Z. (2026). *What explains variation in Plectropomus leopardus density* (Version v1.0). Zenodo. doi:10.5281/zenodo.22933039

Examples

```
str(coral_trout)
table(coral_trout$REGION, coral_trout$NTR)
```

---

gam_influence *Compare fixed and re-estimated deletion effects across sites*

Description

For each site (cluster), the model is refitted without that site twice on the frozen full-data representation: once with the estimated smoothing parameters (and theta) held at their full-fit values, and once with them re-estimated. The changes in the fitted values of the remaining sites are compared.

Usage

```
gam_influence(
  model,
  cluster,
  data = NULL,
  which = NULL,
  progress = interactive()
)
```

Arguments

`model`	A fitted `mgcv::gam()` object within the scope described in the README; anything the scope guard recognises as outside it is an error.
`cluster`	The sites: a column name (in the model frame or in `data`), or a vector with one value per row of `data` or per row used by the model. Rows dropped by `na.action` or `subset` are matched by row name, and the match is checked against the data values.

`data`	The data frame used to fit `model`. Only needed when `cluster` is not a variable of the model; if `NULL`, the data named in the model call is used, with a message.
`which`	Optional subset of site ids to evaluate.
`progress`	Show a text progress bar.

Value

An object of class `gam_influence`, a list with

- `sites`: one row per site: `id`, `n_obs`, `fixed`, `added`, `total`, `status`, `note`;
- `hyper`: one row per site and hyperparameter: `id`, `parameter`, `type` (`"sd"` for random-effect terms, reported as the standard deviation; `"lambda"` for other smooths; `"theta"`), `full`, `deleted` (re-estimated without the site) and `ratio`. A very large lambda ratio means the smooth became close to a straight line; the exact value is then not meaningful, because the REML score is flat there;
- `info`: family, number of rows, and the version numbers used.

What is reported

For each site, three weighted RMS changes in the linear predictor of the *other* sites' rows (the retained rows):

- `fixed`: refit without the site, hyperparameters fixed at their full-fit values. This is what a deletion diagnostic with fixed smoothing parameters sees.
- `total`: refit without the site, hyperparameters re-estimated.
- `added`: the difference between the two refits, i.e. what re-estimating the hyperparameters adds.

For every retained observation, fixed + added = total exactly (they are changes in the same fitted value). The three columns are RMS summaries of those changes, and RMS values do not add: `total` is never larger than `fixed` + `added`, and equals it only when one change is a non-negative multiple of the other (for example when `added` is zero). A smaller `total` therefore only says that the two changes are not perfectly aligned; on its own it does not mean that re-estimation offsets the fixed change. Offsetting shows up as `total^2 < fixed^2 + added^2`.

Magnitudes use the full fit's IRLS (Fisher) weights, so rows with more information count more, and are on the link scale. On a log link a change of d in the linear predictor multiplies the predicted mean by exp(d), a relative change of exp(d) - 1. When the individual log changes are small, exp(d) - 1 is close to d, so an RMS of 0.02 approximates a weighted RMS relative change of about 2% in the other sites' predicted means. The approximation fails for large individual changes (d = 0.2 is +22%, d = -0.2 is -18%), and a small RMS can still hide a few large changes. There is no null reference, so compare sites with each other rather than with a fixed cut-off.

The hyperparameters are the smoothing parameters mgcv estimated and, for `nb()` with theta estimated, theta. Smoothing parameters or theta fixed by the user stay fixed in both refits. All results are conditional on the frozen representation: the full fit's model matrix, identifiability constraints and penalties are reused, not rebuilt from the reduced data.

Status

`status` is `"ok"` or lists what happened: `"rank-deficient"` (the site was the only one informing some coefficients; they are dropped from its refits, which leaves the reported changes valid), `"lambda held"` (a smooth had no data left, so its smoothing parameter was held; see `note`), `"two optima"` (see below), `"not converged"` or `"error"`. Sites whose refits failed have `NA` magnitudes and a warning is issued.

The REML criterion can have more than one optimum, typically an interior one and a flat plateau where a smoothing parameter is very large (for a thin plate smooth, a nearly straight line). The re-estimated refit is therefore searched for twice, from mgcv's default starting values and from the full-fit values, and the fit with the lower REML score is used. When the two searches end at fits that differ by more than 10% of the larger total change, the status says `"two optima"` and `note` gives the other fit's `total` and how much worse its REML score was. A small difference in REML score means the data barely prefer one fit over the other, so treat that site's `added` and `total` with care. Two searches still do not guarantee the best optimum overall.

Examples

```
library(mgcv)
set.seed(1)
d <- data.frame(site = factor(rep(1:12, each = 6)), x = runif(72))
d$y <- sin(2 * pi * d$x) + rnorm(12, 0, 0.4)[d$site] + rnorm(72, 0, 0.5)
m <- gam(y ~ s(x) + s(site, bs = "re"), data = d, method = "REML")
infl <- gam_influence(m, cluster = "site", progress = FALSE)
infl
top_units(infl, by = "added", n = 3)
```

---

gam_influence-methods *Methods for gam_influence objects*

Description

Methods for gam_influence objects

Usage

```
## S3 method for class 'gam_influence'
print(x, n = 5, ...)

## S3 method for class 'gam_influence'
summary(object, ...)

## S3 method for class 'summary.gam_influence'
print(x, ...)

## S3 method for class 'gam_influence'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

top_units(x, by = "total", n = 10)
```

Arguments

x, object	A gam_influence object.
n	Number of sites to show or return.
...	Unused.
row.names, optional	Accepted for compatibility; ignored.
by	Column of the site table to rank by (largest first): "total", "fixed" or "added".

Value

- print(): the gam_influence object x, unchanged and invisibly. It is called for its side effect of printing the sites with the largest changes and the largest hyperparameter changes.
- summary(): an object of class summary.gam_influence, a list with quantiles (a matrix: minimum, median, 90th percentile and maximum of fixed, added and total, and the number of missing values), hyper (a data frame: for each hyperparameter, its full-data value and the range of its ratios when a site is removed; NULL if the model has no estimated hyperparameters), n_sites (the number of sites), status (a table of site statuses) and info (as in gam_influence()).
- print() for a summary.gam_influence object: x, unchanged and invisibly. It is called for its side effect of printing the summary.
- as.data.frame(): the site table x$sites, a data frame with one row per site and columns id, n_obs, fixed, added, total, status and note (see gam_influence()).
- top_units(): a data frame with the same columns as x$sites, holding the n sites with the largest by, largest first.

Examples

```
library(mgcv)
set.seed(1)
d <- data.frame(site = factor(rep(1:12, each = 6)), x = runif(72))
d$y <- sin(2 * pi * d$x) + rnorm(12, 0, 0.4)[d$site] + rnorm(72, 0, 0.5)
m <- gam(y ~ s(x) + s(site, bs = "re"), data = d, method = "REML")
infl <- gam_influence(m, cluster = "site", progress = FALSE)
print(infl, n = 3)
summary(infl)
top_units(infl, by = "added", n = 3)
head(as.data.frame(infl))
```

---

plot.gam_influence *Plot fixed against added deletion changes*

Description

One point per site. The x-axis is fixed, the change in the other sites' fitted values when the site is removed with the hyperparameters held; the y-axis is added, the extra change when they are re-estimated. Both are weighted RMS changes on the link scale (see gam_influence()), so the axes share one scale. Sites above the dashed diagonal change the fit more through the smoothing parameters (and theta) than through the held-parameter refit.

Usage

```
## S3 method for class 'gam_influence'
plot(
  x,
  n = 5,
  by = "total",
  main = "Site deletion: fixed vs added change",
  ...
)
```

Arguments

x	A gam_influence object.
n	Number of sites to label.
by	Which change ranks the sites to label: "total", "added" or "fixed".
main	Plot title.
...	Unused.

Details

The n sites with the largest by are labelled. Sites whose re-estimated refit had two REML optima (status "two optima") are drawn as orange triangles; read their added with care. Sites whose refits failed have no changes to plot and are left out, with a message.

Value

Called for its side effect of drawing the plot. Invisibly, a data frame with one row per plotted site (id, n_obs, fixed, added, total, status) and a logical column labelled that marks the labelled sites.

Examples

```
library(mgcv)
set.seed(1)
d <- data.frame(site = factor(rep(1:12, each = 6)), x = runif(72))
d$y <- sin(2 * pi * d$x) + rnorm(12, 0, 0.4)[d$site] + rnorm(72, 0, 0.5)
m <- gam(y ~ s(x) + s(site, bs = "re"), data = d, method = "REML")
infl <- gam_influence(m, cluster = "site", progress = FALSE)
plot(infl, n = 3)
```

Index

∗ **datasets**
coral_trout, 2

as.data.frame.gam_influence
(gam_influence-methods), 5

coral_trout, 2

gam_influence, 3
gam_influence(), *6*
gam_influence-methods, 5

mgcv::gam(), *3*

plot.gam_influence, 6
print.gam_influence
(gam_influence-methods), 5
print.summary.gam_influence
(gam_influence-methods), 5

summary.gam_influence
(gam_influence-methods), 5

top_units (gam_influence-methods), 5