Package ‘gbm.auto’

September 1, 2023

Title Automated Boosted Regression Tree Modelling and Mapping Suite

Version 2023.08.31


License MIT + file LICENSE

Depends R (>= 3.5.0)

Imports beepr (>= 1.2), dismo (>= 1.3-14), dplyr (>= 1.0.9), gbm (>= 2.1.1), ggmap (>= 3.0.2), ggplot2 (>= 3.4.2), ggspatial (>= 1.1.9), lifecycle, lubridate (>= 1.9.2), mapplots (>= 1.5), Metrics (>= 0.1.4), readr (>= 2.1.4), sf (>= 0.9-7), stars (>= 0.6-3), starsExtra (>= 0.2.7), stats (>= 3.3.1), stringi (>= 1.6.1), tidyselect (>= 1.2.0), viridis (>= 0.6.4)

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Simon Dedman [aut, cre] (<https://orcid.org/0000-0002-9108-972X>)

Maintainer Simon Dedman <simondedman@gmail.com>

Repository CRAN

Date/Publication 2023-09-01 17:00:03 UTC
Adult_Females

Data: Numbers of 4 adult female rays caught in 2137 Irish Sea trawls, 1994 to 2014

Description

2137 capture events of adult female cuckoo, thornback, spotted and blonde rays in the Irish Sea from 1994 to 2014 by the ICES IBTS, including explanatory variables: Length Per Unit Effort in that area by the commercial fishery, depth, temperature, distance to shore, and current speed at the bottom.

Usage

data(Adult_Females)
Format

A data frame with 2137 rows and 13 variables:

- **Longitude**: Decimal longitudes in the Irish Sea
- **Latitude**: Decimal latitudes in the Irish Sea
- **Haul_Index**: ICES IBTS area, survey, station, and year
- **F_LPUE**: Commercial fishery LPUE in Kg/Hr
- **Depth**: Metres, decimal
- **Temperature**: Degrees, decimal
- **Salinity**: PPM
- **Distance_to_Shore**: Metres, decimal
- **Current_Speed**: Metres per second at the seabed
- **Cuckoo**: Numbers of cuckoo rays caught, standardised to 1 hour
- **Thornback**: Numbers of thornback rays caught, standardised to 1 hour
- **Blonde**: Numbers of blonde rays caught, standardised to 1 hour
- **Spotted**: Numbers of spotted rays caught, standardised to 1 hour

Author(s)

Simon Dedman, <simondedman@gmail.com>

Source

[http://datras.ices.dk](http://datras.ices.dk)

---

**AllPreds_E**

Data: Predicted abundances of 4 ray species generated using gbm.auto

**Description**

Predicted abundances of 4 ray species generated using gbm.auto, and Irish commercial beam trawler effort 2012.

**Usage**

data(AllPreds_E)
Format

A data frame with 378570 rows and 7 variables:

- **Latitude**  Decimal latitudes in the Irish Sea
- **Longitude**  Decimal longitudes in the Irish Sea
- **Cuckoo**  Predicted abundances of cuckoo rays in the Irish Sea, generated using gbm.auto
- **Thornback**  Predicted abundances of thornback rays in the Irish Sea, generated using gbm.auto
- **Blonde**  Predicted abundances of blonde rays in the Irish Sea, generated using gbm.auto
- **Spotted**  Predicted abundances of spotted rays in the Irish Sea, generated using gbm.auto
- **Effort**  Irish commercial beam trawler effort 2012

Author(s)

Simon Dedman, <simondedman@gmail.com>

---

**AllScaledData**

Data: Scaled abundance data for 2 subsets of 4 rays in the Irish Sea, by gbm.cons

Description

A dataset containing the output of the gbm.cons example run, conservation priority areas within the Irish Sea for juvenile and adult female cuckoo, blonde, thornback and spotted rays.

Usage

data(AllScaledData)

Format

A data frame with 378570 rows and 3 variables:

- **Longitude**  Decimal longitudes in the Irish Sea
- **Latitude**  Decimal latitudes in the Irish Sea
- **allscaled**  Relative abundance. Each juvenile and adult female cuckoo, blonde, thornback and spotted ray scaled to 1 and added together

Author(s)

Simon Dedman, <simondedman@gmail.com>
**breaks.grid**

*Defines breakpoints for draw.grid and legend.grid; mapplots fork*

**Description**

Defines breakpoints from values in `grd` with options to exclude outliers, set number of bins, and include a dedicated zero column. Forked by SD 05/01/2019 to add 'lo', else bins always begin at 0, killing plotting when all data are in a tight range at high values e.g. 600:610

**Usage**

```r
breaks.grid(grd, quantile = 0.975, ncol = 12, zero = TRUE)
```

**Arguments**

- `grd`: An array produced by `make.grid` or a list produced by `make.multigrid` or a vector of positive values.
- `quantile`: The maximum value of the breaks will be determined by the quantile given here. This can be used to deal with outlying values in `grd`. If `quantile = 1` then the maximum value of the breaks will be the same as the maximum value in `grd`.
- `ncol`: Number of colours to be used, always one more than the number of breakpoints. Defaults to 12.
- `zero`: Logical, should zero be included as a separate category? Defaults to TRUE.

**Value**

A vector of breakpoints for `draw.grid` in `mapplots`

**Author(s)**

Simon Dedman, <simondedman@gmail.com>

Hans Gerritsen

**Examples**

```r
breaks.grid(100, ncol=6)
b breaks.grid(100, ncol=5, zero=FALSE)

# create breaks on the log scale
exp(breaks.grid(log(10000), ncol=4, zero=FALSE))
```
calibration

Description

Internal use only. Jane Elith/John Leathwick 17th March 2005. Calculates calibration statistics for either binomial or count data but the family argument must be specified for the latter a conditional test for the latter will catch most failures to specify the family.

Usage

    calibration(obs, preds, family = c("binomial", "bernoulli", "poisson"))

Arguments

    obs          Observed data.
    preds        Predicted data.
    family       Statistical distribution family. Choose one.

Value

    roc & calibration stats internally within gbm runs e.g. in gbm.auto.

Author(s)

    Simon Dedman, <simondedman@gmail.com>

gbm.auto

Automated Boosted Regression Tree modelling and mapping suite

Description

Automates delta log normal boosted regression trees abundance prediction. Loops through all permutations of parameters provided (learning rate, tree complexity, bag fraction), chooses the best, then simplifies it. Generates line, dot and bar plots, and outputs these and the predictions and a report of all variables used, statistics for tests, variable interactions, predictors used and dropped, etc. If selected, generates predicted abundance maps, and Unrepresentativeness surfaces. See www.GitHub.com/SimonDedman/gbm.auto for issues, feedback, and development suggestions. See SimonDedman.com for links to walkthrough paper, and papers and thesis published using this package.
Usage

gbm.auto(
    grids = NULL,
    samples,
    expvar,
    resvar,
    randomvar = FALSE,
    tc = c(2),
    lr = c(0.01, 0.005),
    bf = 0.5,
    offset = NULL,
    n.trees = 50,
    ZI = "CHECK",
    fam1 = c("bernoulli", "binomial", "poisson", "laplace", "gaussian"),
    fam2 = c("gaussian", "bernoulli", "binomial", "poisson", "laplace"),
    simp = TRUE,
    gridslat = 2,
    gridson = 1,
    samplesGridsAreaScaleFactor = 1,
    multiplot = TRUE,
    cols = grey.colors(1, 1, 1),
    linesfiles = TRUE,
    smooth = FALSE,
    savedir = tempdir(),
    savegbm = TRUE,
    loadgbm = NULL,
    varint = TRUE,
    map = TRUE,
    shape = NULL,
    RSB = TRUE,
    BnW = TRUE,
    alerts = TRUE,
    pngtype = c("cairo-png", "quartz", "Xlib"),
    gaus = TRUE,
    MLEvaluate = TRUE,
    brv = NULL,
    grv = NULL,
    Bin_Preds = NULL,
    Gaus_Preds = NULL,
    ...
)

Arguments

grids  Explanatory data to predict to. Import with (e.g.) read.csv and specify object name. Defaults to NULL (won't predict to grids).

samples Explanatory and response variables to predict from. Keep col names short (~17 characters max), no odd characters, spaces, starting numerals or terminal peri-
gbm.auto

ods. Spaces may be converted to periods in directory names, underscores won’t. Can be a subset of a large dataset.

expvar Vector of names or column numbers of explanatory variables in 'samples': c(1,3,6) or c("Temp","Sal"). No default.

resvar Name or column number(s) of response variable in samples: 12, c(1,4), "Rockfish". No default. Column name is ideally species name.

randomvar Add a random variable (uniform distribution, 0-1) to the expvars, to see whether other expvars perform better or worse than random.

tc Permutations of tree complexity allowed, can be vector with the largest sized number no larger than the number of explanatory variables e.g. c(2,7), or a list of 2 single numbers or vectors, the first to be passed to the binary BRT, the second to the Gaussian, e.g. tc = list(c(2,6), 2) or list(6, c(2,6)).

lr Permutations of learning rate allowed. Can be a vector or a list of 2 single numbers or vectors, the first to be passed to the binary BRT, the second to the Gaussian, e.g. lr = list(c(0.01,0.02),0.0001) or list(0.01,c(0.001, 0.0005)).

bf Permutations of bag fraction allowed, can be single number, vector or list, per tc and lr. Defaults to 0.5.

offset Column number or quoted name in samples, containing offset values relating to

n.trees From gbm.step, number of initial trees to fit. Can be single or list but not vector i.e. list(fam1,fam2).


fam1 Probability distribution family for 1st part of delta process, defaults to "bernoulli". Choose one.

fam2 Probability distribution family for 2nd part of delta process, defaults to "gaussian". Choose one.

simp Try simplifying best BRTs?

gridsrlt Column number for latitude in 'grids'.

gridslon Column number for longitude in 'grids'.
samplesGridsAreaScaleFactor Scale up or down factor so values in the predict-to pixels of 'grids' match the spatial scale sampled by rows in 'samples'. Default 1 means no change.

multiplot Create matrix plot of all line files? Default true. turn off if big n of exp vars causes an error due to margin size problems.

cols Barplot colour vector. Assignment in order of explanatory variables. Default 1white: white bars black borders. '1' repeats.

linesfiles Save individual line plots’ data as csv’s? Default TRUE.

smooth Apply a smoother to the line plots? Default FALSE.

savedir Save outputs to a temporary directory (default) else change to current directory e.g. "/home/me/folder". Do not use getwd() here.
savegbm
Save gbm objects and make available in environment after running? Open with load("Bin_Best_Model") Default TRUE.

loadgbm
Relative or (very much preferably) absolute location of folder containing Bin_Best_Model and Gaus_Best_Model. If set will skip BRT calculations and do predicted maps and csvs. Avoids re-running BRT models again (the slow bit), can run normally once with savegbm=T then multiple times with new grids & loadgbm to predict to multiple grids e.g. different seasons, areas, etc. Default NULL, character vector, ",/" for working directory.

varint
Calculate variable interactions? Default:TRUE, FALSE for error: "contrasts can be applied only to factors with 2 or more levels".

map
Save abundance map png files?

shape
Enter the full path to downloaded map e.g. coastline shapefile, possibly from gbm.basemap, typically Crop_Map.shp, including the .shp. Can also name an existing object in the environment, read in with sf::st_read. Default NULL, in which case bounds calculated by gbm.map which then calls gbm.basemap to download and auto-generate the base map.

RSB
Run Unrepresentativeness surface builder? Default TRUE.

BnW
Repeat maps in black and white e.g. for print journals. Default TRUE.

alerts
Play sounds to mark progress steps. Default TRUE but running multiple small BRTs in a row (e.g. gbm.loop) can cause RStudio to crash.

pngtype
Filetype for png files, alternatively try "quartz" on Mac. Choose one.

gaus
Do family2 (typically Gaussian) runs as well as family1 (typically Bin)? Default TRUE.

MLEvaluate
do machine learning evaluation metrics & plots? Default TRUE.

brv
Dummy param for package testing for CRAN, ignore.

grv
Dummy param for package testing for CRAN, ignore.

Bin_Preds
Dummy param for package testing for CRAN, ignore.

Gaus_Preds
Dummy param for package testing for CRAN, ignore.

... Optional arguments for gbm.step (dismo package) arguments n.trees and max.trees, both of which can be added in list(1,2) format to pass to fam1 and 2; for gbm.mapsf colourscale, heatcolours, colournumber, and others.

Details

Errors and their origins:


2. Error in FUN(X[[j]], ...) : only defined on a data frame with all numeric variables. Check your variable types are correct, e.g. numerics haven’t been imported as factors because there’s an errant first row of text information before the data. Remove NA rows from the response variable if present: convert blank cells to NA on import with read.csv(x, na.strings = "") then samples2 <- samples[-which(is.na(samples[,resvar_column_number])),]
3. At BF=0.5, if nrows <= 42, gbm.step will crash. Use gbm.bfcheck to determine optimal viable BF size.

4. Maps/plots don’t work/output. If on a Mac, try changing pngtype to "quartz".

5. Error in while (delta.deviance > tolerance.test & n.fitted < max.trees): missing value where TRUE/FALSE needed. If running a zero-inflated delta model (bernoulli/bin & gaussian/gaus), Data are expected to contain zeroes (lots of them in zero- inflated cases), have you already filtered them out, i.e. are only testing the positive cases? Or do you only have positive cases? If so only run (e.g.) Gaussian: set ZI to FALSE.

6. Error in round(gbm.object$cv.statistics$deviance.mean, 4) : non-numeric argument to mathematical function. LR or BF probably too low in earlier BRT (normally Gaus run with highest TC).

7. Error in if (n.trees > x$n.trees) : argument is of length zero. LR or BF probably too low in earlier BRT (normally Gaus run with highest TC).

8. Error in gbm.fit(x, y, offset = offset, distribution = distribution, w = w): The dataset size is too small or subsampling rate is too large: nTrain*bag.fraction <= n.minobsinnode. LR or BF probably too low in earlier BRT (normally Gaus run with highest TC). It may be that you don’t have enough positive samples to run BRT modelling. Run gbm.bfcheck to check recommended minimum BF size.

9. Warning message: In cor(y_i, u_i) : the standard deviation is zero. LR or BF probably too low in earlier BRT (normally Gaus run with highest TC). It may be that you don’t have enough positive samples to run BRT modelling. Run gbm.bfcheck to check recommended minimum BF size. Similarly: glm.fit: fitted probabilities numerically 0 or 1 occurred, and glm.fit: algorithm did not converge. Similarly: Error in if (get(paste0("Gaus_BRT", ",.tc", j, ",.lr", k, ",bf", l)))$self.statistics$correlation[[1]]: argument is of length zero. See also: Error 15.

10. Anomalous values can obfuscate clarity in line plots e.g. salinity range 32:35ppm but dataset has errant 0 value: plot axis will be 0:35, and 99.99% of the data will be in the tiny bit at the right. Clean your data beforehand.

11. Error in plot.new() : figure margins too large: In RStudio, adjust plot pane (usually bottom right) to increase its size. Still fails? Set multiplot=FALSE.

12. Error in dev.print(file = paste0("./", names(samples[i]), "/pred_dev_bin.jpeg"): can only print from a screen device. An earlier failed run (e.g. LR/BF too low) left a plotting device open. Close it with: 'dev.off()'.

13. RStudio crashed: set alerts=F and pause cloud sync programs if outputting to a synced folder.

14. Error in grDevices::dev.copy(device = function (filename = "Rplot%03d.jpeg", could not open file './resvar/pred_dev_bin.jpeg' (or similar). Your resvar column name contains an illegal character e.g. /&'_. Fix with colnames(samples)$n <- "BetterName".

15. Error in gbm.fit: Poisson requires the response to be a positive integer. If running Poisson distributions, ensure the response variables are positive integers, but if they are, try a smaller LR.

16. If lineplots of factorial variables include empty columns be sure to remove unused levels with samples %<>% droplevels() before the gbm.auto run.

17. Error in seq.default(from = min(x$var.levels[[i.var[i]]]), to = max(x$var.levels[[i.var[i]]]));’from’ must be a finite number. If you logged any expvars with log() and they has zeroes in them, those zeroes became imaginary numbers. Use log1p() instead.
gbm.basemap

18. Error in loadNamespace...’dismo’ 1.3-9 is being loaded, but >= 1.3.10 is required: first do
remotes::install_github("rspatial/dismo") then library(dismo).

ALSO: check this section in the other functions run by gbm.auto e.g. gbm.map, gbm.basemap. Use
traceback() to find the source of errors.

I strongly recommend that you download papers 1 to 5 (or just the doctoral thesis) on http://
www.simonedman.com, with emphasis on P4 (the guide) and P1 (statistical background). Elith et
al 2008 (http://refhub.elsevier.com/S0304-3800(15)00207-0/sbref0085) is also strongly
recommended. Just because you CAN try every conceivable combination of tc, lr, bf, all, at once
doesn’t mean you should. Try a range of lr in shrinking orders of magnitude from 0.1 to 0.000001,
find the best, THEN try tc c(2, n.expvars), find the best THEN bf c(0.5, 0.75, 0.9) and then in
between if either outperform 0.5.

Value

Line, dot and bar plots, a report of all variables used, statistics for tests, variable interactions, pre-
dictors used and dropped, etc. If selected, generates predicted abundance maps, and Unrepresenta-

Author(s)

Simon Dedman, <simondedman@gmail.com>

Examples

# Not run. Note: grids file was heavily cropped for CRAN upload so output map
# predictions only cover patchy chunks of the Irish Sea, not the whole area.
# Full versions of these files:
# https://drive.google.com/file/d/1WHyP7tP3roozVKwi_R_IpW7tZ1hZA7r
# /?view=sharing
library(gbm.auto)
data(grids)
data(samples)
# Set your working directory
gbm.auto(grids = grids, samples = samples, expvar = c(4:8, 10), resvar = 11,
tc = c(2,7), lr = c(0.005, 0.001), ZI = TRUE, savegbm = FALSE)

---

**gbm.basemap**

Creates Basemaps for Gbm.auto mapping from your data range

**Description**

Downloads unzips crops & saves NOAAs global coastline shapefiles to user-set box. Use for 'shape'
in gbm.map. If downloading in RStudio uncheck "Use secure download method for HTTP" in Tools
> Global Options > Packages. Simon Dedman, 2015/6 simondedman@gmail.com GitHub.com/SimonDedman/gbm.auto
Usage

gbm.basemap(
    bounds = NULL,
    grids = NULL,
    gridslat = NULL,
    gridslon = NULL,
    getzip = TRUE,
    zipvers = "2.3.7",
    savedir = tempdir(),
    savename = "Crop_Map",
    res = "CALC",
    extrabounds = FALSE
)

Arguments

bounds Region to crop to: c(xmin,xmax,ymin,ymax).
grids If bounds unspecified, name your grids database here.
gridslat If bounds unspecified, specify which column in grids is latitude.
gridslon If bounds unspecified, specify which column in grids is longitude.
getzip Download & unpack GSHHS data to WD? "TRUE" else absolute/relative reference to GSHHS_shp folder, including that folder.
zipvers GSHHS version, in case it updates. Please email developer (SD) if this is incorrect.
savedir Save outputs to a temporary directory (default) else change to current directory e.g. "/home/me/folder". Do not use getwd() here.
savename Shapefile save-name, no shp extension, default is "Crop_Map"
res Resolution, 1:5 (low:high) OR c,l,i,h,f (coarse, low, intermediate, high, full) or "CALC" to calculate based on bounds. Choose one.
extrabounds Grow bounds 16pct each direction to expand rectangular datasets basemaps over the entire square area created by basemap in mapplots.

Details

errors and their origins:

1. Error in setwd(getzip) : cannot change working directory If you’ve specified the location of the local GSHHS_shp folder, ensure you’re in the correct directory relative to it. This error means it looked for the folder and couldn’t find it.

2. If rgdal install fails in Linux try: sudo apt-get install libgdal-dev && sudo apt-get install libproj-dev"

3. Error in as.environment(pos):no item called "package:shapefiles" on the search list: strange error occurring despite shapefiles being coded like all other packages. Correct output produced regardless.
4. subscript out of bounds: can’t crop world map to your bounds. Check lat/lon are the right way around: check gridslat and gridslon point to the correct columns for lat and lon in grids, and those columns named (something like) lat and lon, ARE ACTUALLY the latitudes and longitudes, and not the wrong way around. 
5. If your download is timing out use options(timeout = 240).
6. Error in attachNamespace("shapefiles"): namespace is already attached. Use: unloadNamespace("shapefiles"). 

Value 
basemap coastline file for gbm.map in gbm.auto. "cropshp" SpatialPolygonsDataFrame in in local environment & user-named files in "CroppedMap" folder. Load later with maptools function: MyMap <- readShapePoly("./CroppedMap/Crop_Map")

Author(s) 
Simon Dedman, <simondedman@gmail.com>

Examples 

# Not run: downloads and saves external data. 
data(samples) 
mybounds <- c(range(samples[,3]),range(samples[,2])) 
gbm.basemap(bounds = mybounds, getzip = "./GSHHS_shp/", 
savename = "My_Crop_Map", res = "f") 
# In this example GSHHS folder already downloaded to the working directory 
# hence I pointed getzip at that rather than having it download the zip again
Value

Prints minimum Bag Fraction size for gbm.auto.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Examples

data(samples)
gbm.bfcheck(samples = samples, resvar = "Cuckoo")

gbm.cons

Usage

gbm.cons(
mygrids,
subsets,
alerts = TRUE,
map = TRUE,
BnW = TRUE,
resvars,
gbmautos = TRUE,
savedir = tempdir(),
expvars,
tcs = NULL,
lrs = rep(list(c(0.01, 0.005)), length(resvars)),
bfs = rep(0.5, length(resvars)),
ZIs = rep("CHECK", length(resvars)),
colss = rep(list(grey.colors(1, 1, 1)), length(resvars)),
linesfiless = rep(FALSE, length(resvars)),
savegbms = rep(TRUE, length(resvars)),
varints = rep(TRUE, length(resvars)),
maps = rep(TRUE, length(resvars)),
RSBs = rep(TRUE, length(resvars)),
BnWs = rep(TRUE, length(resvars)),
zeroes = rep(TRUE, length(resvars)),
shape = NULL,

gbm.cons

```r
pngtype = c("cairo-png", "quartz", "Xlib"),
gridslat = 2,
gridslon = 1,
grids = NULL
)
```

**Arguments**

- `mygrids` Gridded lat+long+data object to predict to.
- `subsets` Subset name(s): character; single or vector, corresponding to matching-named dataset objects e.g. read in by `read.csv()`.
- `alerts` Play sounds to mark progress steps.
- `map` Produce maps.
- `BnW` Also produce B&W maps?
- `resvars` Vector of resvars cols from dataset objects for gbm.autos, length(subsets)*species, no default.
- `gbmautos` Do gbm.auto runs for species? Default TRUE, set FALSE if already run and output files in expected directories.
- `savedir` Save outputs to a temporary directory (default) else change to current directory e.g. "/home/me/folder". Do not use `getwd()` here.
- `expvars` List object of expvar vectors for gbm.autos, length = no. of subsets * no. of species. No default.
- `tcs` Gbm.auto parameters, auto-calculated below if not provided by user.
- `lrs` Gbm.auto parameter, uses defaults if not provided by user.
- `bfs` Gbm.auto parameter, uses defaults if not provided by user.
- `ZIs` Gbm.auto parameter, autocalculated below if not provided by user. Choose one entry.
- `colss` Gbm.auto parameter, uses defaults if not provided by user.
- `linesfiless` Gbm.auto parameter, uses defaults if not provided by user.
- `savegbms` Gbm.auto parameter, uses defaults if not provided by user.
- `varints` Gbm.auto parameter, uses defaults if not provided by user.
- `maps` Gbm.auto parameter, uses defaults if not provided by user.
- `RSBs` Gbm.auto parameter, uses defaults if not provided by user.
- `BnWs` Gbm.auto parameter, uses defaults if not provided by user.
- `zeroes` For breaks.grid, include zero-only category in colour breakpoints and subsequent legend. Defaults to TRUE.
- `shape` Coastline file for gbm.map.
- `pngtype` File-type for png files, alternatively try "quartz" on Mac. Choose one.
- `gridslat` Per Gbm.auto defaults to 2.
- `gridslon` Per Gbm.auto defaults to 1.
- `grids` Dummy param for package testing for CRAN, ignore.
Value

Maps via gbm.map & saved data as csv file.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Examples

```r
# Not run: downloads and saves external data.
data(grids)
gbm.cons(mygrids = grids, subsets = c("Juveniles","Adult_Females"),
         resvars = c(44:47,11:14),
         expvars = list(c(4:11,15,17,21,25,29,37),
                         c(4:11,15,18,22,26,30,38),
                         c(4:11,15,19,23,27,31),
                         c(4:11,15,20,24,28,32,39),
                         c(4:10, 4:10, 4:10, 4:10),
                         tcs = list(c(2,14), c(2,14), 13, c(2,14), c(2,6), 6, c(2,6)),
                         lrs = list(c(0.01,0.005), c(0.01,0.005), 0.005, c(0.01,0.005),
                                   0.005, 0.005, 0.001, 0.005),
                         ZIs = rep(TRUE, 8),
                         savegbms = rep(FALSE, 8),
                         varints = rep(FALSE, 8),
                         RSBs = rep(FALSE, 8),
                         BnWs = rep(FALSE, 8),
                         zeroes = rep(FALSE,8))
```

---

**gbm.factorplot**

*Creates ggplots of marginal effect for factorial variables from plot.gbm in gbm.auto.*

### Description

Creates an additional plot to those created by gbm.plot within gbm.auto. Can also take Bin/Gaus_Best_line.csv or similar csvs directly. Allows changing of x axis levels and all ggplot and ggsave params.

### Usage

```r
gbm.factorplot(
  x,
  factorplotlevels = NULL,
  ggplot2guideaxisangle = 0,
  ggplot2labsx = "",
  ggplot2labsy = "Marginal Effect",
```
gbm.factorplot

```r
... 
```

**Arguments**

- **x**
  
  Input data.frame or tibble or csv (full file address including .csv) to read, must be a categorical variable.

- **factorplotlevels**
  
  Character vector of the variable’s levels to reorder the x axis by, all must match those in the first column of the csv exactly. Default NULL orders from high to low Y value.

- **ggplot2guideaxisangle**
  
  Default 0. Set at e.g. 90 to rotate.

- **ggplot2labsx**
  
  Default: "".

- **ggplot2labsy**
  
  Default: "Marginal Effect".

- **ggplot2axistext**
  
  Default: 1.5.

- **ggplot2axistitle**
  
  Default: 2.

- **ggplot2legendtext**
  
  Default: 1.

- **ggplot2legendtitle**
  
  Default: 1.5.

- **ggplot2legendtitlealign**
  
  Default: 0.

- **ggplot2plotbackgroundfill**
  
  Default: "white".

- **ggplot2plotbackgroundcolour**
  
  Default: "grey50".

- **ggplot2striptextx**
  
  Default: 2.

- **ggplot2panelbordercolour**
  
  Default: "black".

- **ggplot2panelborderfill**
  
  Default: NA.

- **ggplot2panelborderlinewidth**
  
  Default: 1.

- **ggplot2legendspacingx**
  
  Default: grid::unit(0, "cm").

- **ggplot2legendbackground**
  
  Default: ggplot2::element_blank().

- **ggplot2panelbackgroundfill**
  
  Default: "white".

- **ggplot2panelbackgroundcolour**
  
  Default: "grey50".

- **ggplot2panelgridcolour**
  
  Default: "grey90".

- **ggplot2legendkey**
  
  Default: ggplot2::element_blank().

- **ggsavefilename**
  
  Default: paste0(lubridate::today(), "_Categorical-variable.png").

- **ggsaveplot**
  
  Default: ggplot2::last_plot().

- **ggsavedevice**
  
  Default: "png".

- **ggsavepath**
  
  Default: "".

- **ggsavescale**
  
  Default: 2.

- **ggsavewidth**
  
  Default: 10.

- **ggsaveheight**
  
  Default: 4.

- **ggsaveunits**
  
  Default: "in".

- **ggsavedpi**
  
  Default: 300.

- **ggsavelimitsize**
  
  Default: TRUE.

...
gbm.factorplot

ggplot2legendtitle
    Default: 1.5.

ggplot2legendtitlealign
    Default: 0, # otherwise effect type title centre aligned for some reason.

ggplot2plotbackgroundfill
    Default: "white", white background.

ggplot2plotbackgroundcolour
    Default: "grey50", background lines.

ggplot2striptextx
    Default: 2.

ggplot2panelbordercolour
    Default: "black".

ggplot2panelborderfill
    Default: NA.

ggplot2panelborderlinewidth
    Default: 1.

ggplot2legendspacingx
    Default: unit(0, "cm"), # compress spacing between legend items, this is min.

ggplot2legendbackground
    Default: ggplot2::element_blank().

ggplot2panelbackgroundfill
    Default: "white".

ggplot2panelbackgroundcolour
    Default: "grey50".

ggplot2panelgridcolour
    Default: "grey90".

ggplot2legendkey
    Default: ggplot2::element_blank().

ggsavefilename
    Default: paste0(saveloc, lubridate::today(), "_SankeyAlluvial_EMT.SoEv-EfTyp_Col-EfSz.png").

ggsaveplot
    Default: last_plot().

ggsavedevice
    Default: "png".

ggsavepath
    Default: ".".

ggsavescale
    Default: 2.

ggsavewidth
    Default: 10.

ggsaveheight
    Default: 4.

ggsaveunits
    Default: "in".

ggsavedpi
    Default: 300.

ggsavelimitsize
    Default: TRUE.

...  Allow params to be called from higher function esp gbm.auto.

Details

‘r lifecycle::badge("experimental")
**gbm.lmplots**

**Value**

Factorial ggplot saved with users preferred location and name.

**Author(s)**

Simon Dedman, <simondedman@gmail.com>

---

**Plot linear models for all expvar against the resvar**

**Description**

Loops the lmplot function, shows linear model plots for all expvar against the resvar. Good practice to do this before running gbm.auto so you have a sense of the basic relationship of the variables.

**Usage**

```r
gbm.lmplots(
  samples = NULL,
  expvar = NULL,
  resvar = NULL,
  expvarnames = NULL,
  resvarname = NULL,
  savedir = NULL,
  plotname = NULL,
  pngtype = c("cairo-png", "quartz", "Xlib"),
  r2line = TRUE,
  pointtext = FALSE,
  pointlabs = resvar,
  pointcol = "black",
  ...
)
```

**Arguments**

- **samples**: Explanatory and response variables to predict from. Keep col names short (~17 characters max), no odd characters, spaces, starting numerals or terminal periods. Spaces may be converted to periods in directory names, underscores won’t. Can be a subset of a large dataset.
- **expvar**: Vector of names or column numbers of explanatory variables in `samples`: c(1,3,6) or c("Temp","Sal"). No default.
- **resvar**: Name or column number(s) of response variable in samples: 12, c(1,4), "Rockfish". No default. Column name is ideally species name.
- **expvarnames**: Vector of names same length as expvar, if you want nicer names.
- **resvarname**: Single character object, if you want a nicer resvar name.
gbm.loop

- **savedir**: Save location, end with "/".
- **plotname**: Character vector of plot names else expvarnames else expvar will be used.
- **pngtype**: Filetype for png files, alternatively try "quartz" on Mac.
- **r2line**: Plot rsquared trendline, default TRUE.
- **pointtext**: Label each point? Default FALSE.
- **pointlabs**: Point labels, defaults to resvar value.
- **pointcol**: Points colour, default "black".
- **...**: Allows controlling of text label params e.g. adj cex &.

Details

Errors and their origins:

Value

Invisibly saves png plots into savedir.

Author(s)

Simon Dedman, <simondedman@gmail.com>

---

gbm.loop Calculate Coefficient Of Variation surfaces for gbm.auto predictions

Description

Bagging introduces stochasticity which can result in sizeable variance in output predictions by gbm.auto for small datasets. This function runs a user-specified number of loops through the same gbm.auto parameter combinations and calculates the Coefficient Of Variation in the predicted abundance scores for each site aka cell. This can be mapped, to spatially demonstrate the output variance range.

Usage

```r
gbm.loop(
  loops = 10,
  savedir = tempdir(),
  savecsv = TRUE,
  calcpreds = TRUE,
  varmap = TRUE,
  measure = "CPUE",
  cleanup = FALSE,
  grids = NULL,
  samples,
  expvar,
)```
gbm.loop

resvar,
randomvar = FALSE,
tc = c(2),
lr = c(0.01),
bf = 0.5,
n.trees = 50,
ZI = "CHECK",
fam1 = c("bernoulli", "binomial", "poisson", "laplace", "gaussian"),
fam2 = c("gaussian", "bernoulli", "binomial", "poisson", "laplace"),
simp = TRUE,
gridslat = 2,
gridslon = 1,
multiplot = FALSE,
cols = grey.colors(1, 1, 1),
linesfiles = TRUE,
smooth = FALSE,
savegbm = FALSE,
loadgbm = NULL,
varint = FALSE,
map = TRUE,
shape = NULL,
RSB = FALSE,
BnW = FALSE,
alerts = FALSE,
pngtype = c("cairo-png", "quartz", "Xlib"),
gaus = TRUE,
MLEvaluate = TRUE,
runautos = TRUE,
Min.Inf = NULL,
...)

Arguments

loops The number of loops required. integer.
savedir Save outputs to a temporary directory (default) else change to current directory
e.g. "/home/me/folder". Do not use getwd() here.
savecsv Save coefficients of variation in simple & extended format.
calcpreds Calculate coefficients of variation of predicted abundance?
varmap Create a map of the coefficients of variation outputs?
measure Map legend, coefficients of variation of what? Default CPUE.
cleanup Remove gbm.auto-generated directory each loop? Default FALSE.
grids See gbm.auto help for all subsequent params.
samples See gbm.auto help.
exptar See gbm.auto help.
resvar See gbm.auto help.
randomvar  See gbm.auto help.
tc         See gbm.auto help.
lr         See gbm.auto help.
bf         See gbm.auto help.
n.trees    See gbm.auto help. Choose one.
ZI          See gbm.auto help. Choose one.
fam1       See gbm.auto help. Choose one.
fam2       See gbm.auto help. Choose one.
simp       See gbm.auto help.
gridslat   See gbm.auto help.
gridslon   See gbm.auto help.
multiplot  See gbm.auto help. Default False
cols       See gbm.auto help.
linesfiles See gbm.auto help; TRUE or linesfiles calculations fail.
smooth     See gbm.auto help.
savegbm    See gbm.auto help.
loadgbm    See gbm.auto help.
varint     See gbm.auto help.
map        See gbm.auto help.
shape      See gbm.auto help.
RSB        See gbm.auto help.
BnW        See gbm.auto help.
alerts     See gbm.auto help; default FALSE as frequent use can crash RStudio.
pngtype    See gbm.auto help. Choose one.
gaus       See gbm.auto help.
MLEvaluate See gbm.auto help.
runautos   Run gbm.autos, default TRUE, turn off to only collate numbered-folder results.
Min.Inf    Dummy param for package testing for CRAN, ignore.
...        Additional params for gbm.auto sub-functions including gbm.step.

**Value**

Returns a data frame of lat, long, 1 predicted abundance per loop, and a final variance score per cell.

**Author(s)**

Simon Dedman, <simonedman@gmail.com>
**gbm.map**

Maps of predicted abundance from Boosted Regression Tree modelling

**Description**

Generates maps from the outputs of gbm.step then Gbm.predict.grids, handled automatically within gbm.auto but can be run alone, and generates representativeness surfaces from the output of gbm.rsb.

**Usage**

```r
gbm.map(
  x,
  y,
  z,
  byx = NULL,
  byy = NULL,
  grdfun = mean,
  mapmain = "Predicted CPUE (numbers per hour): ",
  species = "Response Variable",
  heatcolours = c("white", "yellow", "orange", "red", "brown4"),
  colournumber = 8,
  shape = NULL,
  landcol = "grey80",
  mapback = "lightblue",
  legendloc = "bottomright",
  legendtitle = "CPUE",
  lejback = "white",
  zero = TRUE,
  quantile = 1,
  byxout = FALSE,
  breaks = NULL,
  byxport = NULL,
  ...
)
```

**Examples**

```r
# Not run: downloads and saves external data.
library("gbm.auto")
data(grids) # load grids
data(samples) # load samples
gbmloophexample <- gbm.loop(loops = 2, samples = samples,
grids = grids, expvar = c(4:10), resvar = 11, simp = F)
```
Arguments

- **x**  
  Vector of longitudes, from make.grid in mapplots; x. Order by this (descending) SECOND.

- **y**  
  Vector of latitudes, from make.grid in mapplots; grids[,gridslat]. Order by this (descending) first.

- **z**  
  Vector of abundances generated by gbm.predict.grids, from make.grid in mapplots; grids[,predabund].

- **byx**  
  Longitudinal width of grid cell, from make.grid in mapplots. Autogenerated if left blank.

- **byy**  
  Latitudinal height of grid cell, from make.grid in mapplots. Autogenerated if left blank.

- **grdfun**  
  make.grid operand for >=2 values per cell. Default: mean, other options: sum prod min max sd se var.

- **mapmain**  
  Plot title, has species value appended. Default "Predicted CPUE (numbers per hour): ".

- **species**  
  Response variable name, from basemap in mapplots; names(samples[i]). Defaults to "Response Variable".

- **heatcolours**  
  Vector for abundance colour scale, defaults to the heatcol from legend.grid and draw.grid in mapplots which is c("white", "yellow", "orange", "red", "brown4").

- **colournumber**  
  Number of colours to spread heatcol over, default:8.

- **shape**  
  Basemap shape to draw, from draw.shape in mapplots. Defaults to NULL which calls gbm.basemap to generate it for you. First read in a shp file e.g. myshape <- shapefiles::read.shapefile("Crop_Map") WITHOUT the .shp extension, then use shape = myshape.

- **landcol**  
  Colour for 'null' area of map (for marine plots, this is land), from draw.shape in mapplots. Default "grey80" (light grey).

- **mapback**  
  Basemap background colour, defaults to lightblue (ocean for marine plots).

- **legendloc**  
  Location on map of legend box, from legend.grid in mapplots, default bottom-right.

- **legendtitle**  
  The metric of abundance, e.g. CPUE for fisheries, from legend.grid in mapplots. Default "CPUE".

- **lejback**  
  Background colour of legend, from legend.grid in mapplots. Default "white".

- **zero**  
  Force include 0-only bin in breaks.grid and thus legend? Default TRUE.

- **quantile**  
  Set max quantile of data to include in bins, from breaks.grid in mapplots; lower to e.g. 0.975 cutoff outliers; default 1.

- **byxout**  
  Export byx to use elsewhere? Default:FALSE.

- **breaks**  
  Vector of breakpoints for colour scales; default blank, generated automatically.

- **byxport**  
  Dummy param for package testing for CRAN, ignore.

- **...**  
  Additional arguments for legend.grid’s ... which passes to legend.
Details

[Superseded] Superseded by gbm.mapsf on 2023-08-07, but still works.

Errors and their origins:

Error in seq.default(xlim[1], xlim[2], by = byx):wrong sign in 'by' argument Check that your lat & long columns are the right way around. Ensure grids data are gridded, i.e. they are in a regular pattern of same/similar lines of lat/lon, even if they’re missing sections.

Suggested parameter values: z = rsbdf[,“Unrepresentativeness”]

mapmain = “Unrepresentativeness: ”
legendtitle = “UnRep 0-1”

Value

Species abundance maps using data provided by gbm.auto, and Representativeness Surface Builder maps using data provided by gbm.rsb, to be run in a png/par/gbm.map/dev.off sequence.

Author(s)

Simon Dedman, <simondedman@gmail.com>
Hans Gerritsen

Examples

# Not run: downloads and saves external data.
# Suggested code for outputting to png:
data(grids)
# set working directory somewhere suitable
png(filename = “gbmmap.png”, width = 7680, height = 7680, units = “px”,
pointsize = 192, bg = “white”, res = NA, family = “”, type = “cairo-png”)
par(mar = c(3.2,3,1.3,0), las = 1, mgp = c(2.1,0.5,0), xpd = FALSE)
gbm.map(x = grids[,“Longitude”], y = grids[,“Latitude”], z = grids[,“Effort”]
, species = “Effort”)
dev.off()
Usage

gbm.mapsf(
    predabund = NULL,
    predabundlon = 2,
    predabundlat = 1,
    predabundpreds = 3,
    myLocation = NULL,
    trim = TRUE,
    trimfivepct = FALSE,
    scale100 = FALSE,
    gmapsAPI = NULL,
    mapsource = "google",
    googlemap = TRUE,
    maptype = "satellite",
    darkenproportion = 0,
    mapzoom = NULL,
    shape = NULL,
    expandfactor = 0,
    colourscale = "viridis",
    colorscale = NULL,
    heatcolours = c("white", "yellow", "orange", "red", "brown4"),
    colournumber = 8,
    colourscalelimits = NULL,
    colourscalebreaks = NULL,
    colourscalelabels = NULL,
    colourscaleexpand = NULL,
    studyspecies = "MySpecies",
    plottitle = paste0("Predicted abundance of ", studyspecies),
    plotsubtitle = "CPUE",
    legendtitle = "CPUE",
    plotcaption = paste0("gbm.auto::gbm.mapsf, ", lubridate::today()),
    axisxlabel = "Longitude",
    axisylabel = "Latitude",
    legendposition = c(0.05, 0.15),
    fontsize = 12,
    fontfamily = "Times New Roman",
    filesavename = paste0(lubridate::today(), "_*", studyspecies, "_*", legendtitle, ".png"),
    savedir = tempdir(),
    receiverlats = NULL,
    receiverlons = NULL,
    receivernames = NULL,
    receiverrange = NULL,
    recpointscol = "black",
    recpointsfill = "white",
    recpointsalpha = 0.5,
    recpointssize = 1,
    recpointsshape = 21,
    recbufcol = "grey75",
)
Arguments

predabund  Predicted abundance data frame produced by gbm.auto (Abundance_Preds_only.csv), with Latitude, Longitude, and Predicted Abundance columns. Default NULL.

You need to read the csv in R if not already present as an object in the environment.

predabundlon  Longitude column number. Default 2.

predabundlat  Latitude column number. Default 1.

predabundpreds  Predicted abundance column number, default 3.

myLocation  Location for extents, format c(xmin, ymin, xmax, ymax). Default NULL, extents autocreated from data.

trim  Remove NA & <=0 values and crop to remaining date extents? Default TRUE.

trimfivepct  Replace anything < 5% of the max value (i.e. < 95% UD contour in home range analysis) with NA since it won’t be drawn (for movegroup dBBMMs). Default FALSE.

scale100  Scale Predicted Abundance to 100? Default FALSE.

gmapsAPI  Enter your Google maps API here, quoted character string. Default NULL.

mapsource  Source for ggmap::get_map; uses Stamen as fallback if no Google Maps API present. Options: "google", "stamen", "gbm.basemap". Default "google". Using "gbm.basemap" requires one to have run that function already, and enter its location using the shape parameter below.

googlemap  If pulling basemap from Google maps, this sets expansion factors since Google Maps tiling zoom setup doesn’t align to myLocation extents. Default TRUE.


darkenproportion  Amount to darken the google/stamen basemap. 0-1. Default 0.


shape  If mapsource is "gbm.basemap", enter the full path to gbm.basemaps downloaded map, typically Crop_Map.shp, including the .shp. Default NULL. Can also name an existing object in the environment, read in with sf::st_read.
expandfactor  Extents expansion factor for basemap. default 0.
colourscale  Scale fill colour scheme to use, default "viridis". other option is "gradient".
colorscale  Scale fill colour scheme to use, default NULL, populating this will overwrite colourscale.
heatcolours  Vector of colours if gradient selected for colourscale, defaults to heatmap theme.
colournumber  Number of colours to spread heatcolours over, if gradient selected for colourscale. Default 8.
colourscalelimits  Colour scale limits, default NULL, vector of 2, e.g. c(0, 0).
colourscalebreaks  Colour scale breaks, default NULL.
colourscalelabels  Colour scale labels, default NULL, must match number of breaks.
colourscaleexpand  Colour scale expand, default NULL, vector of 2, e.g. c(0, 0).
studyspecies  Name of your study species, appears in plot title and savename. Default "MySpecies".
plottitle  Title of the resultant plot, default paste0("Predicted abundance of ", studyspecies).
plotsubtitle  Plot subtitle, default "CPUE". Can add the n of your individuals.
legendtitle  Legend title, default "CPUE".
plotcaption  Plot caption, default "gbm.auto::gbm.mapsf" + today’s date.
axisxlabel  Default "Longitude".
axisylabel  Default "Latitude".
legendposition  Vector of 2, format c(1,2), Proportional distance of (middle?) of legend box from L to R, percent distance from Bottom to Top. Values 0 to 1. Default c(0.05, 0.15).
fontsize  Font size, default 12.
fontfamily  = Font family, default "Times New Roman".
filesavename  File savename, default today’s date + studyspecies + legendtitle.
savedir  Save outputs to a temporary directory (default) else change to current directory e.g. "~/home/me/folder". Do not use getwd() here. No terminal slash. E.g. paste0(mov groupsavedir, "Plot/").
receiverlats  Vector of latitudes for receivers to be plotted.
receiverlons  Vector of longitudes for receivers to be plotted. Same length as receiverlats.
receivernames  Vector of names for receivers to be plotted. Same length as receiverlats.
receiverrange  Single (will be recycled), or vector (same length as receiverlats) of detection ranges in metres for receivers to be plotted. If you have a max and a (e.g.) 90 percent detection range, probably use max.
recpointscol  Colour of receiver centrepoint outlines. Default "black".
recpointsfill  Colour of receiver centrepoint fills. Default "white".
recpointsalpha  Alpha value of receiver centrepoint fills, 0 (invisible) to 1 (fully visible). Default 0.5.
recpointssize  Size of receiver points. Default 1.
recpointsshape  Shape of receiver points, default 21, circle with outline and fill.
recbufcol  Colour of the receiver buffer circle outlines. Default "grey75"
recbuffill  Colour of the receiver buffer circle fills. Default "grey".
recbufalpha  Alpha value of receiver buffer fills, 0 (invisible) to 1 (fully visible). Default 0.5.
reclabcol  Receiver label text colour. Default "black".
reclabfill  Receiver label fill colour, NA for no fill. Default NA.
reclabnudgeX  Receiver label offset nudge in X dimension. Default 0.
reclabpad  Receiver label padding in lines. Default 0.
reclabrad  Receiver label radius in lines. Default 0.15.
reclabbor  Receiver label border in mm. Default 0.

Details

Error in seq.default(xlim[1], xlim[2], by = byx): wrong sign in 'by' argument Check that your lat & long columns are the right way around. Ensure grids (predabund) data are gridded, i.e. they are in a regular pattern of same/similar lines of lat/lon, even if they’re missing sections.

Suggested parameter values: z = rsbdf[,"Unrepresentativeness"]
mapmain = "Unrepresentativeness: "
legendtitle = "UnRep 0-1"

How to get Google map basemaps:
(from https://www.youtube.com/watch?v=O5cUoVpVUjU):

1. Sign up with dev console: a. You must enter credit card details, but won’t be charged if your daily API requests stay under the limit. b. Follow the link: https://console.cloud.google.com/projectselector2/apis/dashboard c. Sign up for Google cloud account (it may auto populate your current gmail), click agree and continue. d. Click the navigation email in the top left corner and click on Billing. e. Create a billing account – they will NOT auto charge after trial ends. f. Enter information, click on ’start my free trial’. They may offer a free credit for trying out their service. More pricing details: https://mapsplatform.google.com/pricing/. g. Click “Select a Project” then “New project” in the top right corner. h. Enter Project Name, leave Location as is, click “Create”. i. You should now see your project name at the top, where the drop-down menu is.
3. Create Credentials for API Key: a. Return to ’APIs & Services’ page. b. Click on Credentials. c. At the top click ’Create Credentials > API Key’. d. API key should pop up with option to copy it. e. You can restrict the key if you want by following steps 4 & 5 here: https://www.youtube.com/watch?v=O5cUoVpVUjU&t=232s
Species abundance maps using data provided by gbm.auto, and Representativeness Surface Builder maps using data provided by gbm.rsb, to be run in a png/par/gbm.map/dev.off sequence.

Author(s)
Simon Dedman, <simondedman@gmail.com>

Examples

# Not run

gbm.rsb(samples, grids, expvarnames, gridslat, gridslon)

Arguments

samples Data frame with response and explanatory variables.
grids Data frame of (more/different) explanatory variables and no response variable, to be predicted to by gbm.predict.grids.
expvarnames Vector of column names of explanatory variables being tested. Can be length 1. Names must match in samples and grids.
gridslat Column number for latitude in 'grids'.
gridslon Column number for longitude in 'grids'.

Value
Gridded data table of representativeness values which is then mapped with gbm.map and also saved as a csv
Author(s)
Simon Dedman, <simondedman@gmail.com>

Examples

data(samples)
data(grids)
rsbdf_bin <- gbm.rsb(samples, grids, expvarnames = names(samples[c(4:8, 10)]),
gridslat = 2, gridslon = 1)

Description

Usage

gbm.step.sd(
data,
gbm.x,
gbm.y,
offset = NULL,
fold.vector = NULL,
tree.complexity = 1,
learning.rate = 0.01,
bag.fraction = 0.75,
site.weights = rep(1, nrow(data)),
var.monotone = rep(0, length(gbm.x)),
n.folds = 10,
prev.stratify = TRUE,
family = "bernoulli",
n.trees = 50,
step.size = n.trees,
max.trees = 10000,
tolerance.method = "auto",
tolerance = 0.001,
plot.main = TRUE,
plot.folds = FALSE,
verbose = TRUE,
silent = FALSE,
keep.fold.models = FALSE,
keep.fold.vector = FALSE,
keep.fold.fit = FALSE,
...
)

Arguments

data The input dataframe.
gbm.x The predictors.
gbm.y The response.
offset Allows an offset to be specified.
fold.vector Allows a fold vector to be read in for CV with offsets.
tree.complexity Sets the complexity of individual trees.
learning.rate Sets the weight applied to individual trees.
bag.fraction Sets the proportion of observations used in selecting variables.
site.weights Allows varying weighting for sites.
var.monotone Restricts responses to individual predictors to monotone.
n.folds Number of folds.
prev.stratify Prevalence stratify the folds - only for p/a data.
family Family - bernoulli (=binomial), poisson, laplace or gaussian.
n.trees Number of initial trees to fit.
step.size Numbers of trees to add at each cycle.
max.trees Max number of trees to fit before stopping.
tolerance.method Method to use in deciding to stop - "fixed" or "auto".
tolerance Tolerance value to use - if method == fixed is absolute, if auto is multiplier * total mean deviance.
plot.main Plot hold-out deviance curve.
plot.folds Plot the individual folds as well.
verbose Control amount of screen reporting.
silent To allow running with no output for simplifying model).
keep.fold.models Keep the fold models from cross validation.
keep.fold.vector Allows the vector defining fold membership to be kept.
keep.fold.fit Allows the predicted values for observations from CV to be kept.
... Allows for any additional plotting parameters.
gbm.subset

Details

Divides the data into 10 subsets, with stratification by prevalence if required for pa data then fits a gbm model of increasing complexity along the sequence from n.trees to n.trees + (n.steps * step.size) calculating the residual deviance at each step along the way after each fold processed, calculates the average holdout residual deviance and its standard error then identifies the optimal number of trees as that at which the holdout deviance is minimised and fits a model with this number of trees, returning it as a gbm model along with additional information from the cv selection process.

requires gbm library from Cran requires roc and calibration scripts of J Elith requires calc.deviance script of J Elith/J Leathwick

Value

GBM models using gbm as the engine.

---

gbm.subset  
Subsets gbm.auto input datasets to 2 groups using the partial deviance plots

Description

Set your working directory to the output folder of a gbm.auto/gbm.loop run. This function returns the variable value corresponding to the 0 value on the lineplots, which should be the optimal place to split the dataset into 2 subsets, low and high, IF the relationship doesn’t cross 0 more than once. Function is similarly useful to quickly get the 0-point value in these cases, i.e. where values below are detrimental, values above beneficial (check plots though)

Usage

```r
gbm.subset(x, fams = c("Bin", "Gaus"), loop = FALSE)
```

Arguments

- `x` Vector of variable names.
- `fams` Vector of statistical data distribution family names to be modelled by gbm.
- `loop` Is the folder a gbm.loop output?

Details

loop varnames are BinLineLoop_VAR.csv & GausLineLoop_VAR.csv normal varnames are Bin_Best_line_VAR.csv & Gaus_Best_line_VAR.csv

Just use average between the last negative & first positive point unless any points fall on zero

Value

a list of breakpoint values which datasets can be subsetted using.
**Author(s)**

Simon Dedman, <simondedman@gmail.com>

**Examples**

```r
# Not run: requires completed gbm.auto run.
# having run gbm.auto (with linesfiles=TRUE), set working directory there
data(samples)
gbm.subset(x = names(samples[c(4:8, 10)]), fams = c("Bin", "Gaus"))
```

---

**gbm.valuemap**  
*Decision Support Tool that generates (Marine) Protected Area options using species predicted abundance maps*

**Description**

Scales response variable data, maps a user-defined explanatory variable to be avoided, e.g. fishing effort, combines them into a map showing areas to preferentially close. Bpa, the precautionary biomass required to protect the spawning stock, is used to calculate MPA size. MPA is then grown to add subsequent species starting from the most conservationally at-risk species, resulting in one MPA map per species, and a multicolour MPA map of all. All maps list the percentage of the avoid-variables total that is overlapped by the MPA in the map legend.

**Usage**

```r
gbm.valuemap(
  dbase,
  loncolno = 1,
  latcolno = 2,
  goodcols,
  badcols,
  conservecol = NULL,
  plotthis = c("good", "bad", "both", "close"),
  maploops = c("Combo", "Biomass", "Effort", "Conservation"),
  savedir = tempdir(),
  savethis = TRUE,
  HRMSY = 0.15,
  goodweight = NULL,
  badweight = NULL,
  m = 1,
  alerts = TRUE,
  BrnW = TRUE,
  shape = NULL,
  pngtype = c("cairo-png", "quartz", "Xlib"),
```
gbm.valuemap

byxport = NULL,
...
)

Arguments

dbase Data.frame to load. Expects Lon, Lat & data columns: predicted abundances, fishing effort etc. E.g.: Abundance_Preds_All.csv from gbm.auto.
loncolno Column number in dbase which has longitudes.
latcolno Column number in dbase which has latitudes.
goodcols Which column numbers are abundances (where higher = better)? List them in order of highest conservation importance first e.g. c(3,1,2,4).
badcols Which col no.s are 'negative' e.g. fishing (where higher = worse)?
conservecol Conservation column, from gbm.cons.
plotthis Vector of variable types to plot. Delete any, or all w/ NULL.
maploops Vector of sort loops to run.
savedir Save outputs to a temporary directory (default) else change to current directory e.g. "~/home/me/folder". Do not use getwd() here.
savethis Export all data as csv?
HRMSY Maximum percent of each goodcols stock which can be removed yearly, as decimal (0.15 = 15 pct). Must protect remainder: 1-HRMSY. Single number or vector. If vector, same order as goodcols. Required.
goodweight Single/vector weighting multiple(s) for goodcols array.
badweight Ditto for badcols array.
m Multiplication factor for Bpa units. 1000 to convert tonnes to kilos, 0.001 kilos to tonnes. Assumedly the same for all goodcols.
alerts Play sounds to mark progress steps.
BnW Also produce greyscale images for print publications.
shape Set coastline shapefile, else uses British Isles. Generate your own with gbm.basemap.
pngtype File-type for png files, alternatively try "quartz" on Mac. Choose one.
byxport Dummy param for package testing for CRAN, ignore.
...
Optional terms for gbm.map.

Value

Species abundance, abundance vs avoid variable, and MPA maps per species and sort type, in b&w if set. CSVs of all maps if set.

Author(s)

Simon Dedman, <simondedman@gmail.com>
**grids**

*Data: Explanatory variables for rays in the Irish Sea*

**Description**
A dataset containing explanatory variables for environment, fishery and predators of rays including juveniles in the Irish Sea.

**Usage**
```r
data(grids)
```

**Format**
A data frame with 378570 rows and 43 variables:

- **Longitude**  Decimal longitudes in the Irish Sea
- **Latitude**  Decimal latitudes in the Irish Sea
- **Depth**  Metres, decimal
- **Temperature**  Degrees, decimal
- **Salinity**  PPM
- **Current_Speed**  Metres per second at the seabed
- **Distance_to_Shore**  Metres, decimal
- **F_LPUE**  Commercial fishery LPUE in Kg/Hr
- **Scallop**  Average KwH Scallop effort from logbooks, Marine Institute and MMO combined
- **MI_Av_E_Hr**  Average effort hours, Marine Institute Scallop VMS, 0.03 x 0.02 rectangles, all Irish Sea, 2006-14
- **MI_Av_LPUE**  Average scallop CPUE, Marine Institute Scallop VMS, 0.03 x 0.02 rectangles, all Irish Sea, 2006-14
- **MI_Sum_Liv**  Sum of live weight. Average scallop CPUE, Marine Institute Scallop VMS, 0.03 x 0.02 rectangles, all Irish Sea, 2006-14
- **Whelk**  MMO Whelk LPUE 2009-12, pivot, polygons to points
- **MmoAvScKwh**  MMO Scallop Effort 2009-12, pivot, polygons to points. ICES rectangles
- **HubDist**  map calc, distance of grid point to nearest datras point representing it (for preds)
- **Cod_C**  ICES IBTS CPUE of cod caught between 1994 - 2014 large enough to predate upon <= year 1 cuckoo rays
- **Cod_T**  As Cod_C for yr1 thornback rays
- **Cod_B**  As Cod_C for yr1 blonde rays
- **Cod_S**  As Cod_C for yr1 spotted rays
- **Haddock_C**  As Cod_C, haddock predating upon cuckoo rays
- **Haddock_T**  As Cod_C, haddock predating upon thornback rays
Haddock_B  As Cod_C, haddock predating upon blonde rays
Haddock_S  As Cod_C, haddock predating upon spotted rays
Plaice_C  As Cod_C, plaice predating upon cuckoo rays
Plaice_T  As Cod_C, plaice predating upon thornback rays
Plaice_B  As Cod_C, plaice predating upon blonde rays
Plaice_S  As Cod_C, plaice predating upon spotted rays
Whiting_C  As Cod_C, whiting predating upon cuckoo rays
Whiting_T  As Cod_C, whiting predating upon thornback rays
Whiting_B  As Cod_C, whiting predating upon blonde rays
Whiting_S  As Cod_C, whiting predating upon spotted rays
ComSkt_C  As Cod_C, common skate predating upon cuckoo rays
ComSkt_T  As Cod_C, common skate predating upon thornback rays
ComSkt_B  As Cod_C, common skate predating upon blonde rays
ComSkt_S  As Cod_C, common skate predating upon spotted rays
Blonde_C  As Cod_C, blonde ray predating upon cuckoo rays
Blonde_T  As Cod_C, blonde ray predating upon thornback rays
Blonde_S  As Cod_C, blonde ray predating upon spotted rays
C_Preds  All predator CPUEs combined for cuckoo rays
T_Preds  All predator CPUEs combined for thornback rays
B_Preds  All predator CPUEs combined for blonde rays
S_Preds  All predator CPUEs combined for spotted rays
Effort  Irish commercial beam trawler effort 2012

Author(s)
Simon Dedman, <simondedman@gmail.com>

Source
http://oar.marine.ie/handle/10793/958
Juveniles

Data: Explanatory and response variables for 4 juvenile rays in the Irish Sea

Description
A dataset containing explanatory variables for environment, fishery and predators of juvenile rays in the Irish Sea, and the response variables, abundance CPUEs of cuckoo, thornback, blonde and spotted rays.

Usage
data(Juveniles)

Format
A data frame with 2136 rows and 46 variables:

**Survey_StNo_HaulNo_Year** Index column of combined Survey number, station number, haul number, and year
**Latitude** Decimal latitudes in the Irish Sea
**Longitude** Decimal longitudes in the Irish Sea
**Depth** Metres, decimal
**Temperature** Degrees, decimal
**Salinity** PPM
**Current_Speed** Metres per second at the seabed
**Distance_to_Shore** Metres, decimal
**F_LPUE** Commercial fishery LPUE in Kg/Hr
**Scallop** Average KwH Scallop effort from logbooks, Marine Institute and MMO combined
**MI_Av_E_Hr** Average effort hours, Marine Institute Scallop VMS, 0.03 x 0.02 rectangles, all Irish Sea, 2006-14
**MI_Av_LPUE** Average scallop CPUE, Marine Institute Scallop VMS, 0.03 x 0.02 rectangles, all Irish Sea, 2006-14
**MI_Sum_Liv** Sum of live weight. Average scallop CPUE, Marine Institute Scallop VMS, 0.03 x 0.02 rectangles, all Irish Sea, 2006-14
**Whelk** MMO Whelk LPUE 2009-12, pivot, polygons to points
**MmoAvScKwh** MMO Scallop Effort 2009-12, pivot, polygons to points. ICES rectangles
**Cod_C** ICES IBTS CPUE of cod caught between 1994 - 2014 large enough to predate upon <= year 1 cuckoo rays
**Cod_T** As Cod_C for yr1 thornback rays
**Cod_B** As Cod_C for yr1 blonde rays
**Cod_S** As Cod_C for yr1 spotted rays
Juveniles

- **Haddock_C** As Cod_C, haddock predating upon cuckoo rays
- **Haddock_T** As Cod_C, haddock predating upon thornback rays
- **Haddock_B** As Cod_C, haddock predating upon blonde rays
- **Haddock_S** As Cod_C, haddock predating upon spotted rays
- **Plaice_C** As Cod_C, plaice predating upon cuckoo rays
- **Plaice_T** As Cod_C, plaice predating upon thornback rays
- **Plaice_B** As Cod_C, plaice predating upon blonde rays
- **Plaice_S** As Cod_C, plaice predating upon spotted rays
- **Whiting_C** As Cod_C, whiting predating upon cuckoo rays
- **Whiting_T** As Cod_C, whiting predating upon thornback rays
- **Whiting_B** As Cod_C, whiting predating upon blonde rays
- **Whiting_S** As Cod_C, whiting predating upon spotted rays
- **ComSkt_C** As Cod_C, common skate predating upon cuckoo rays
- **ComSkt_T** As Cod_C, common skate predating upon thornback rays
- **ComSkt_B** As Cod_C, common skate predating upon blonde rays
- **ComSkt_S** As Cod_C, common skate predating upon spotted rays
- **Blonde_C** As Cod_C, blonde ray predating upon cuckoo rays
- **Blonde_T** As Cod_C, blonde ray predating upon thornback rays
- **Blonde_S** As Cod_C, blonde ray predating upon spotted rays
- **C_Preds** All predator CPUEs combined for cuckoo rays
- **T_Preds** All predator CPUEs combined for thornback rays
- **B_Preds** All predator CPUEs combined for blonde rays
- **S_Preds** All predator CPUEs combined for spotted rays
- **Cuckoo** Numbers of juvenile cuckoo rays caught, standardised to 1 hour
- **Thornback** Numbers of juvenile thornback rays caught, standardised to 1 hour
- **Blonde** Numbers of juvenile blonde rays caught, standardised to 1 hour
- **Spotted** Numbers of juvenile spotted rays caught, standardised to 1 hour

**Author(s)**

Simon Dedman, <simondedman@gmail.com>
lmplot  

Plot linear model for two variables with R2 & P printed and saved

Description
Simple function to plot and name a linear model

Usage

lmplot(
  x,
  y,
  xname = "X variable",
  yname = "Y variable",
  pngtype = c("cairo-png", "quartz", "Xlib"),
  xlab = xname,
  ylab = yname,
  plotname = xname,
  r2line = TRUE,
  pointtext = FALSE,
  pointlabs = x,
  pointcol = "black",
  savedir = "",
  ...
)

Arguments

x                   Explanatory variable data.
y                   Response variable data.
xname               Variable name for plot header.
yname               Variable name for plot header.
pngtype             Filetype for png files, alternatively try "quartz" on Mac.
xlab                X axis label, parsed from xname unless specified.
ylab                Y axis label, parsed from yname unless specified.
plotname            Filename for png, parsed from xname unless specified.
r2line              Plot rsquared trendline, default TRUE.
pointtext           Label each point? Default FALSE.
pointlabs           Point labels, defaults to resvar value.
pointcol            Points colour, default "black".
savedir             Save location, end with "/".
...                  Allows controlling of text label params e.g. adj cex &.
Details

Errors and their origins:

Value

Invisibly saves png plot into savedir.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Usage

roc(obsdat, preddat)

Arguments

obsdat       Observed data.
preddat      Predicted data.

Value

roc & calibration stats internally within gbm runs e.g. in gbm.auto.

Author(s)

Simon Dedman, <simondedman@gmail.com>

Examples

roc(obsdat = rbinom(100, size = 1, prob = 0.5), preddat = runif(100))
samples

Data: Numbers of 4 ray species caught in 2137 Irish Sea trawls, 1994 to 2014

Description

2244 capture events of cuckoo, thornback, spotted and blonde rays in the Irish Sea from 1994 to 2014 by the ICES IBTS, including explanatory variables: Length Per Unit Effort in that area by the commercial fishery, fishing effort by same, depth, temperature, distance to shore, and current speed at the bottom.

Usage

data(samples)

Format

A data frame with 2244 rows and 14 variables:

Survey_StNo_HaulNo_Year Index column of combined Survey number, station number, haul number, and year
Latitude Decimal latitudes in the Irish Sea
Longitude Decimal longitudes in the Irish Sea
Depth Metres, decimal
Temperature Degrees, decimal
Salinity PPM
Current_Speed Metres per second at the seabed
Distance_to_Shore Metres, decimal
F_LPUE Commercial fishery LPUE in Kg/Hr
Effort Irish commercial beam trawler effort 2012
Cuckoo Numbers of juvenile cuckoo rays caught, standardised to 1 hour
Thornback Numbers of juvenile thornback rays caught, standardised to 1 hour
Blonde Numbers of juvenile blonde rays caught, standardised to 1 hour
Spotted Numbers of juvenile spotted rays caught, standardised to 1 hour

Author(s)

Simon Dedman, <simondedman@gmail.com>

Source

http://oar.marine.ie/handle/10793/958
Index

* datasets
  Adult_Females, 2
  AllPreds_E, 3
  AllScaledData, 4
  grids, 36
  Juveniles, 38
  samples, 42

Adult_Females, 2
AllPreds_E, 3
AllScaledData, 4

breaks.grid, 5

 calibration, 6

gbm.auto, 6
gbm.basemap, 11
gbm.bfcheck, 13
gbm.cons, 14
gbm.factorplot, 16
gbm.lmplots, 19
gbm.loop, 20
gbm.map, 23
gbm.mapsf, 25
gbm.rsb, 30
gbm.step.sd, 31
gbm.subset, 33
gbm.valuemap, 34
grids, 36

Juveniles, 38

lmplot, 40

n, 10

roc, 41

samples, 42