

# Package 'gbm.auto'

October 1, 2021

**Title** Automated Boosted Regression Tree Modelling and Mapping Suite

**Version** 1.5.0

**Description** Automates delta log-normal boosted regression tree abundance prediction. Loops through parameters provided (LR (learning rate), TC (tree complexity), BF (bag fraction)), chooses best, simplifies, & generates line, dot & bar plots, & outputs these & predictions & a report, makes predicted abundance maps, and Unrepresentativeness surfaces.  
Package core built around 'gbm' (gradient boosting machine) functions in 'dismo' (Hijmans, Phillips, Leathwick & Jane Elith, 2020 & ongoing), itself built around 'gbm' (Greenwell, Boehmke, Cunningham & Metcalfe, 2020 & ongoing, originally by Ridgeway). Indebted to Elith/Leathwick/Hastie 2008 'Working Guide' <doi:10.1111/j.1365-2656.2008.01390.x>; workflow follows Appendix S3. See <<http://www.simondedman.com/>> for published guides and papers using this package.

**Depends** R (>= 3.5.0)

**License** MIT + file LICENSE

**Encoding** UTF-8

**Language** en-GB

**LazyData** true

**Imports** gbm (>= 2.1.1), dismo (>= 1.0-15), beepr (>= 1.2), mapplots (>= 1.5), maptools (>= 0.9-1), rgdal (>= 1.1-10), rgeos (>= 0.3-19), raster (>= 2.5-8), sf (>= 0.9-7), shapefiles (>= 0.7), stats (>= 3.3.1)

**RoxygenNote** 7.1.2

**NeedsCompilation** no

**Author** Simon Dedman [aut, cre],  
Hans Gerritsen [aut]

**Maintainer** Simon Dedman <simondedman@gmail.com>

**Repository** CRAN

**Date/Publication** 2021-10-01 04:20:02 UTC

## R topics documented:

Adult_Females	2
AllPreds_E	3
AllScaledData	4
breaks.grid	4
calibration	5
gbm.auto	6
gbm.basemap	10
gbm.bfcheck	12
gbm.cons	13
gbm.loop	15
gbm.map	18
gbm.rsb	20
gbm.subset	21
gbm.valuemap	22
grids	24
Juveniles	26
roc	28
samples	28

## Index 30

---

Adult_Females	<i>Data: Numbers of 4 adult female rays caught in 2137 Irish Sea trawls, 1994 to 2014</i>
---------------	---

---

### 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>

---

AllPreds_E	<i>Data: Predicted abundances of 4 ray species generated using gbm.auto</i>
------------	---

---

**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	<i>Data: Scaled abundance data for 2 subsets of 4 rays in the Irish Sea, by gbm.cons</i>
---------------	--

---

### 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	<i>Defines breakpoints for draw.grid and legend.grid; mapplots fork</i>
-------------	---

---

### 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

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

**Arguments**

grid	An array produced by <code>make.grid</code> or a list produced by <code>make.multigrd</code> 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 <code>grid</code> . If <code>quantile = 1</code> then the maximum value of the breaks will be the same as the maximum value in <code>grid</code> .
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**

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

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

---

calibration

*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](http://www.GitHub.com/SimonDedman/gbm.auto) for issues, feedback, and development suggestions. See [SimonDedman.com](http://SimonDedman.com) for links to walkthrough paper, and papers and thesis published using this package.

**Usage**

```
gbm.auto(  
  grids = NULL,  
  samples,  
  expvar,  
  resvar,  
  tc = c(2),  
  lr = c(0.01, 0.005),  
  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 = 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 periods. Spaces may be converted to periods in directory names, underscores won't. Can be a subset of a large dataset.
expvar	List 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), "Rock-fish". No default. Column name is ideally species name.
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.
n. trees	From gbm.step, number of initial trees to fit. Can be single or list but not vector i.e. list(fam1,fam2).
ZI	Are data zero-inflated? TRUE FALSE "CHECK". Choose one. TRUE: delta BRT, log-normalised Gaus, reverse log-norm and bias corrected. FALSE: do Gaussian only, no log-normalisation. CHECK: Tests data for you. Default is CHECK.
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?

gridslat	Column number for latitude in 'grids'.
gridslon	Column number for longitude in 'grids'.
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 1*white: 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 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	Set coast shapefile, else bounds calculated by gbm.map which then calls gbm.basemap to download and auto-generate the base map. Read in existing files by installing the shapefiles package then DesiredMapName <- read.shapefile("ShapeFileName") omitting the .shp extension.
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 zero in breaks.grid in gbm.map, legend in legend.grid in gbm.map, and 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.



## Details

Errors and their origins:

0. install ERROR: dependencies 'rgdal', 'rgeos' are not available for package 'gbm.auto' For Linux/\*buntu systems, in terminal, type `sudo apt install libgeos-dev sudo apt install libproj-dev sudo apt install libgdal-dev`
1. Error in FUN(X[[i]], ...) : 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 <- samples1[-which(is.na(samples[,resvar_column_number]))],`
2. At `bf=0.5`, if `nrows <= 42` `gbm.step` will crash > Use `gbm.bfcheck` to determine optimal viable `bf` size
3. Maps/plots don't work/output > If on a Mac, try changing `pngtype` to "quartz"
4. Error in `while (delta.deviance > tolerance.test AMPERSAND n.fitted < max.trees)`: missing value where TRUE/FALSE needed > If running a zero-inflated delta model (`bernoulli/bin & gaussian/kaus`), 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
5. 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)
6. 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)
7. 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
8. 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
9. 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
10. Error in `plot.new()` : figure margins too large: > In RStudio, adjust plot frame (usually bottom right) to increase its size Still fails? Set `multiplot=FALSE`
11. 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()`
12. RStudio crashed: set `alerts=F` and pause cloud sync programs if outputting to a synced folder
13. Error in `grDevices::dev.copy(device = function (filename = "Rplot (or similar) > Your resvar column name contains an illegal character e.g. / & ' _ Fix with colnames(samples)[n] <- "Better-Name"`

**Value**

Line, dot and bar plots, a report of all variables used, statistics for tests, variable interactions, predictors used and dropped, etc. If selected generates predicted abundance maps, and Unrepresentativeness surface

**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/1WHYpftP3roozVKwi_R_IpW7t1ZIhZA7r
# /view?usp=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. Error in writeSpatialShape(cropshp, savename) x is a NULL object, not a compatible Spatial\*DataFrame. Ensure that your lats and longs are the the right way around

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

4. 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.

**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
```

---

gbm.bfcheck

*Calculates minimum Bag Fraction size for gbm.auto*

---

### Description

Provides minimum bag fractions for gbm.auto, preventing failure due to bf & samples rows limit.  
Simon Dedman, 2016, simondedman@gmail.com, GitHub.com/SimonDedman/gbm.auto

### Usage

```
gbm.bfcheck(samples, resvar, ZI = "CHECK", grv = NULL)
```

### Arguments

samples	Samples dataset, same as gbm.auto.
resvar	Response variable column in samples.
ZI	Are samples zero-inflated? TRUE/FALSE/"CHECK".
grv	Dummy param for package testing for CRAN, ignore.

### 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")
```

**Description**

Runs gbm.auto for multiple subsets of the same overall dataset and scales the combined results, leading to maps which highlight areas of high conservation importance for multiple species in the same study area e.g. using juvenile and adult female subsets to locate candidate nursery grounds and spawning areas respectively.

**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)),
  linesfiles = 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,
  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**

```
# Not run: downloads and saves external data.
data(grid)
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),
    4:10, 4:10, 4:10, 4:10),
  tcs = list(c(2, 14), c(2, 14), 13, c(2, 14), c(2, 6), 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.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**

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

```

resvar,
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.
expvar	See gbm.auto help.
resvar	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.
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, <simondedman@gmail.com>

## Examples

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

---

 gbm.map

---

*Maps of predicted abundance from Boosted Regression Tree modelling*


---

## Description

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

## Usage

```
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,
  ...
)
```

**Arguments**

x	Vector of longitudes, from <code>make.grid</code> in <code>mapplots</code> ; x. Order by this (descending) SECOND.
y	Vector of latitudes, from <code>make.grid</code> in <code>mapplots</code> ; <code>grids[,gridslat]</code> . Order by this (descending) first.
z	Vector of abundances generated by <code>gbm.predict.grids</code> , from <code>make.grid</code> in <code>mapplots</code> ; <code>grids[,predabund]</code> .
byx	Longitudinal width of grid cell, from <code>make.grid</code> in <code>mapplots</code> . Autogenerated if left blank.
byy	Latitudinal height of grid cell, from <code>make.grid</code> in <code>mapplots</code> . Autogenerated if left blank.
grdfun	<code>make.grid</code> operand for $\geq 2$ 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 <code>basemap</code> in <code>mapplots</code> ; <code>names(samples[i])</code> . Defaults to "Response Variable".
heatcolours	Vector for abundance colour scale, defaults to the <code>heatcol</code> from <code>legend.grid</code> and <code>draw.grid</code> in <code>mapplots</code> which is <code>c("white", "yellow", "orange", "red", "brown4")</code> .
colournumber	Number of colours to spread <code>heatcol</code> over, default:8.
shape	Basemap shape to draw, from <code>draw.shape</code> in <code>mapplots</code> . Defaults to NULL which calls <code>gbm.basemap</code> to generate it for you.
landcol	Colour for 'null' area of map (for marine plots, this is land), from <code>draw.shape</code> in <code>mapplots</code> . Default "grey80" (light grey).
mapback	Basemap background colour, defaults to lightblue (ocean for marine plots).
legendloc	Location on map of legend box, from <code>legend.grid</code> in <code>mapplots</code> , default bottom-right.
legendtitle	The metric of abundance, e.g. CPUE for fisheries, from <code>legend.grid</code> in <code>mapplots</code> . Default "CPUE".
lejback	Background colour of legend, from <code>legend.grid</code> in <code>mapplots</code> . Default "white".
zero	Force include 0-only bin in <code>breaks.grid</code> and thus legend? Default TRUE.
quantile	Set max quantile of data to include in bins, from <code>breaks.grid</code> in <code>mapplots</code> ; lower to e.g. 0.975 cutoff outliers; default 1.
byxout	Export <code>byx</code> 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 <code>legend.grid</code> 's ... which passes to <code>legend</code> .

**Details**

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. 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()
```

---

gbm.rsb

*Representativeness Surface Builder*

---

**Description**

Loops through explanatory variables comparing their histogram in 'samples' to their histogram in 'grids' to see how well the explanatory variable range in samples represents the range being predicted to in grids. Assigns a representativeness score per variable per site in grids, and takes the average score per site if there's more than 1 expvar. Saves this to a CSV; it's plotted by `gbm.map` if called in `gbm.auto`. This shows you which areas have the most and least representative coverage by samples, therefore where you can have the most/least confidence in the predictions from `gbm.predict.grids`. Can be called directly, and choosing a subset of expvars allows one to see their individual / collective representativeness.

**Usage**

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

**Arguments**

<code>samples</code>	Data frame with response and explanatory variables.
<code>grids</code>	Data frame of (more/different) explanatory variables and no response variable, to be predicted to by <code>gbm.predict.grids</code> .

**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)

```

---

gbm.subset	<i>Subset gbm.auto input datasets to 2 groups using the partial deviance plots</i>
------------	--

---

### 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

```
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**

```
# 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**

```
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,
    BnW = TRUE,
    shape = NULL,
    pngtype = c("cairo-png", "quartz", "Xlib"),
    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**

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\*0.02 rectangles, all Irish Sea, 2006-14

**MI\_Av\_LPUE** Average scallop CPUE, Marine Institute Scallop VMS, 0.03\*0.02 rectangles, all Irish Sea, 2006-14

**MI\_Sum\_Liv** Sum of live weight. Average scallop CPUE, Marine Institute Scallop VMS, 0.03\*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\*0.02 rectangles, all Irish Sea, 2006-14

**MI\_Av\_LPUE** Average scallop CPUE, Marine Institute Scallop VMS, 0.03\*0.02 rectangles, all Irish Sea, 2006-14

**MI\_Sum\_Liv** Sum of live weight. Average scallop CPUE, Marine Institute Scallop VMS, 0.03\*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

**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>

---

roc	<i>roc</i>
-----	------------

---

### Description

Internal use only. Adapted from Ferrier, Pearce and Watson's code, by J.Elith, see: Hanley, J.A. & McNeil, B.J. (1982) The meaning and use of the area under a Receiver Operating Characteristic (ROC) curve. *Radiology*, 143, 29-36. Also Pearce, J. & Ferrier, S. (2000) Evaluating the predictive performance of habitat models developed using logistic regression. *Ecological Modelling*, 133, 225-245. This is the non-parametric calculation for area under the ROC curve, using the fact that a MannWhitney U statistic is closely related to the area. In *dismo*, this is used in the *gbm* routines, but not elsewhere (see *evaluate*).

### 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	<i>Data: Numbers of 4 ray species caught in 2137 Irish Sea trawls, 1994 to 2014</i>
---------	---

---

### 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, [24](#)
- Juveniles, [26](#)
- samples, [28](#)

- Adult\_Females, [2](#)
- AllPreds\_E, [3](#)
- AllScaledData, [4](#)

- breaks.grid, [4](#)

- calibration, [5](#)

- gbm.auto, [6](#)
- gbm.basemap, [10](#)
- gbm.bfcheck, [12](#)
- gbm.cons, [13](#)
- gbm.loop, [15](#)
- gbm.map, [18](#)
- gbm.rsb, [20](#)
- gbm.subset, [21](#)
- gbm.valuemap, [22](#)
- grids, [24](#)

- Juveniles, [26](#)

- roc, [28](#)

- samples, [28](#)