Package 'dsm'

September 16, 2014

Maintainer David Lawrence Miller <dave@ninepointeightone.net>

License GPL (>= 2)

Title Density surface modelling of distance sampling data

LazyLoad yes

Author David L. Miller, Eric Rexstad, Louise Burt, Mark V. Bravington,Sharon Hedley.

Description This library implements density surface modelling of line
transect data, based on the methods of Hedley et al. (2004). Some
recent developments in the literature have also be incorporated.
Outputs are point and interval estimates of population abundance and density.
Please note that this version of dsm WILL NOT work with the Windows
package DISTANCE earlier than version 6.2. For a version that works with
DISTANCE 6.1, please go to: https://github.com/DistanceDevelopment/dsm-distance-6.1
For the latest version of dsm, please use the github version at the URL listed below.
Miller, D. L., M. L. Burt, E. Rexstad and L. Thomas. 2013. Spatial
models for distance sampling data: recent developments and future
directions. Methods in Ecology and Evolution
Hedley, S.L., S.T. Buckland and D.L. Borchers. 2004. ``Spatial
distance sampling methods" pp 48-
70 in Advanced Distance Sampling,Buckland, S.T. et al. (eds). Oxford University Press.

Version 2.2.5

URL http://github.com/DistanceDevelopment/dsm

Date 2014-10-15

Depends R (>= 3.0), mgcv (>= 1.7), mrds (>= 2.1.8)

Imports nlme, ggplot2, statmod

Suggests Distance

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-16 02:14:20

R topics documented:

dsm-package . . . 2
block.info.per.su . . . 3
check.cols . . . 3
dsm . . . 4
dsm-data . . . 7
dsm.cor . . . 7
dsm.var.gam . . . 9
dsm.var.movblk . . . 10
dsm.var.prop . . . 11
generate.ds.uncertainty . . . 13
generate.mb.sample . . . 14
latlong2km . . . 15
make.soapgrid . . . 15
matrixnotposdef.handler . . . 16
mexdolphins . . . 17
offsets . . . 18
plot.dsm . . . 18
plot.dsm.var . . . 19
predict.dsm . . . 20
print.dsm . . . 21
print.dsm.var . . . 21
print.summary.dsm.var . . . 22
rqgam.check . . . 22
summary.dsm . . . 23
summary.dsm.var . . . 24
trim.var . . . 25

Index **26**

`dsm-package` *Density surface modelling*

Description

Some blurb will eventually go here.

block.info.per.su *Find the block information*

Description

Takes the transect data and works out how many blocks of a given size (in segment terms) fit into each.

Usage

```
block.info.per.su(block.size, data, name.su)
```

Arguments

block.size	number of segments per block
data	data used to build the model
name.su	names of the sampling units (ie. transects)

Value

a data.frame with the following columns

name	the sample unit name (e.g. transect label)
num.seg	number of segments in that transect
num.block	number of blocks available
start.block	block # for first block
end.block	block # for last block
num.req	number of blocks needed for the unit

check.cols *Check column names exist*

Description

Internal function to check that supplied ‘data.frames‘ have the correct columns and checks that sample labels are all unique.

Usage

```
check.cols(ddf.obj, segment.data, observation.data, strip.width, segment.area)
```

Arguments

ddf.obj	a ddf object from 'mrds'
segment.data	segment data as defined in dsm
observation.data	observation data as defined in dsm
strip.width	strip width if strip transects are being used
segment.area	area of segments

Value

nothing, but throws an error if something went wrong

Author(s)

David Lawrence Miller

---

dsm *Fit a density surface model to segment-specific estimates of abundance or density.*

Description

Fits a density surface model (DSM) to detection adjusted counts from a spatially-referenced distance sampling analysis. dsm takes observations of animals, allocates them to segments of line (or strip transects) and optionally adjusts the counts based on detectability using a supplied detection function model. A generalized additive model, generalized mixed model or generalised is then used to model these adjusted counts based on a formula involving environmental covariates.

Usage

```
dsm(formula, ddf.obj, segment.data, observation.data, engine = "gam",
  convert.units = 1, family = quasipoisson(link = "log"), group = FALSE,
  gamma = 1.4, control = list(keepData = TRUE), availability = 1,
  strip.width = NULL, segment.area = NULL, weights = NULL, ...)
```

Arguments

formula	formula for the surface. This should be a valid glm/gam/gamm formula. See "Details", below, for how to define the response.
ddf.obj	result from call to ddf or ds. If ddf.obj is NULL then strip transects are assumed.
segment.data	segment data, see dsm-data.
observation.data	observation data, see dsm-data.
engine	which fitting engine should be used for the DSM (glm/gam/gamm/bam).

convert.units	value to alter length or width for calculation of the offset, applied to 'segment.area' if used.
family	response distribution (popular choices include quasipoisson, Tweedie and negbin). Defaults to quasipossion.
group	if TRUE the abundance of groups will be calculated rather than the abundance of individuals. Setting this option to TRUE is equivalent to setting the size of each group to be 1.
control	the usual control argument for a gam; keepData must be TRUE for variance estimation to work.
availability	an availability bias used to scale the counts/estimated counts by. If we have N animals in a segment, then N/availability will be entered into the model. Uncertainty in the availability is not handled at present.
gamma	parameter to gam() set to a value of 1.4 (from advice in Wood (2006)) such that the gam() is inclined to not 'overfit' when GCV is used to select the smoothing parameter (ignored for REML, see link{gam} for further details).
strip.width	if ddf.obj, above, is NULL, then this is where the strip width is specified (i.e. for a strip transect survey). This is sometimes (and more correctly) referred to as the half-width, i.e. right truncation minus left truncation.
segment.area	if 'NULL' (default) segment areas will be calculated by multiplying the 'Effort' column in 'segment.data' by the (right minus left) truncation distance for the 'ddf.obj' or by 'strip.width'. Alternatively a vector of segment areas can be provided (which must be the same length as the number of rows in 'segment.data') or a character string giving the name of a column in 'segment.data' which contains the areas. If segment.area is specified it takes precident.
weights	weights for each observation used in model fitting. The default, weights=NULL, weights each observation by its area (see Details). Setting a scalar value (e.g. weights=1) all observations are equally weighted.
...	anything else to be passed straight to glm/gam/gamm/bam.

Details

The response (LHS of 'formula') can be one of the following:

n, count, N, abundance	count in each segment
Nhat, abundance.est	estimated abundance per segment, estimation is via a Horvitz-Thompson estimator. This sh
presence	interpret the data as presence/absence (remember to change the family argument to binom
D, density, Dhat, density.est	density per segment

The offset used in the model is dependent on the response:

count	area of segment multiplied by average probability of detection in the segment
estimated count	area of the segment
presence	zero
density	zero

In the latter two cases (density and presence estimation) observations can be weighted by segment areas via the `weights=` argument. By default (`weights=NULL`), when density or presence are estimated the weights are set to the segment areas (using `segment.area` or by calculating `2*`(strip width)`*Effort`) Alternatively `weights=1` will set the weights to all be equal. A third alternative is to pass in a vector of length equal to the number of segments, containing appropriate weights.

Value

a glm/gam/gamm object, with an additional element, `ddf` which holds the detection function object.

Large models

For large models, `engine="bam"` with `method="fREML"` may be useful. Models specified for `bam` should be as `gam`. READ bam before using this option; this option is considered EXPERIMENTAL at the moment. In particular note that the default basis choice (thin plate regression splines) will be slow and that in general fitting is less stable than when using `gam`. For negative binomial response, theta must be specified when using `bam`.

Author(s)

David L. Miller

References

Hedley, S. and S. T. Buckland. 2004. Spatial models for line transect sampling. JABES 9:181-199.

Miller, D. L., Burt, M. L., Rexstad, E. A., Thomas, L. (2013), Spatial models for distance sampling data: recent developments and future directions. Methods in Ecology and Evolution, 4: 1001-1010. doi: 10.1111/2041-210X.12105 (Open Access, available at http://onlinelibrary.wiley.com/doi/10.1111/2041-210X.12105/abstract)

Wood, S.N. 2006. Generalized Additive Models: An Introduction with R. CRC/Chapman & Hall.

Examples

```
library(Distance)
library(dsm)

# load the Gulf of Mexico dolphin data (see ?mexdolphins)
data(mexdolphins)

# fit a detection function and look at the summary
hr.model <- ds(mexdolphins$distdata, max(mexdolphins$distdata$distance),
               key = "hr", adjustment = NULL)
summary(hr.model)

# fit a simple smooth of x and y
mod1<-dsm(N~s(x,y), hr.model, mexdolphins$segdata, mexdolphins$obsdata)
summary(mod1)

# create an offset (in metres)
# each prediction cell is 444km2
```

```
off.set <- 444*1000*1000

# predict over a grid
mod1.pred <- predict(mod1, mexdolphins$preddata, off.set)

# calculate the predicted abundance over the grid
sum(mod1.pred)

# plot the smooth
plot(mod1)
```

---

dsm-data *Data format for DSM*

Description

Two `data.frames` must be provided to `dsm`. They are referred to as `observation.data` and `segment.data` (for observation and segment data, respectively).

Details

`observation.data` - the observation data frame must have the following columns:

`object`	unique object identifier
`Sample.Label`	the identifier for the segment that the observation occurred in
`size`	the size of each observed group (i.e. 1 for individuals)
`distance`	perpendicular/radial distance to observation

`segment.data` - the segment data frame must have the following columns:

`Effort`	the effort (in terms of length of the segment)
`Sample.Label`	identifier for the segment (unique!)
`???`	environmental covariates, for example `x` and `y`

---

dsm.cor *Check for autocorrelation in residuals*

Description

Once a DSM has been fitted to data, this function can be used to check for autocorrelation in the residuals.

Usage

```
dsm.cor(dsm.obj, Transect.Label = "Transect.Label",
```

```
Segment.Label = "Segment.Label", max.lag = 10,
resid.type = "scaled.pearson", fun = cor, ylim = c(0, 1),
subset = "all", ...)
```

Arguments

dsm.obj	a fitted dsm object.
Transect.Label	label for the transect (default: Transect.Label). Using different labels can be useful when transects are split over geographical features or when transects are surveyed multiple times.
Segment.Label	label for the segments (default: Segment.Label).The result of calling order() must make sense.
resid.type	the type of residuals used, see residuals.gam and residuals.gam. Defaults to "scaled.pearson" in the GAM case and "normalized" in the GAMM case (which are equivalent).
fun	the function to use, by default cor, must take two column vectors as arguments.
max.lag	maximum lag to calulate at.
ylim	user defined limits in y direction.
subset	which subset of the data should the correlation function be calculated on?
...	other options to pass to plot.

Value

a plot or a vector of fun applied at the lags.

Details

Within each Transect.Label, segments will be sorted according to their Segment.Labels. This may require some time to get right for your particular data. If one has multiple surveys where transects are revisited, for example, one may want to make Transect.Label a unique transect-survey id. Neither label need to be included in the model, they must just be present in the $data field in the model. This usually means that they have to be in the segment data passed to dsm.

The current iteration of this function will only plot correlations nicely, other things are up to you but you can get the function to return the data (by assigning the result to an object).

If there are NA values in the residuals then the correlogram will not be calculated. This usually occurs due to NA values in the covariates (so the smoother will not have fitted values there). Code like ‘any(is.na(dsm.obj$data))‘ might be helpful.

Author(s)

David L. Miller

Examples

```
library(Distance)
 library(dsm)

 data(mexdolphins)

 hr.model <- ds(mexdolphins$distdata, max(mexdolphins$distdata$distance),
                key = "hr", adjustment = NULL)
 mod1<-dsm(N~s(x,y), hr.model, mexdolphins$segdata, mexdolphins$obsdata)

 dsm.cor(mod1,resid.type="d",max.lag=9,Segment.Label="Sample.Label")
```

`dsm.var.gam` *Variance estimation via Bayesian results*

Description

Use results from the Bayesian interpretation of the GAM to obtain uncertainty estimates. See Wood (2006).

Usage

```
dsm.var.gam(dsm.obj, pred.data, off.set = NULL, seglen.varname = "Effort",
  type.pred = "response")
```

Arguments

`dsm.obj`	an object returned from running `dsm`.
`pred.data`	either: a single prediction grid or list of prediction grids. Each grid should be a `data.frame` with the same columns as the original data.
`off.set`	a a vector or list of vectors with as many elements as there are in `pred.data`. Each vector is as long as the number of rows in the corresponding element of `pred.data`. These give the area associated with each prediction point.
`seglen.varname`	name for the column which holds the segment length (default value "Effort").
`type.pred`	should the predictions be on the "response" or "link" scale? (default "response").

Details

This is based on `dsm.var.prop` by Mark Bravington and Sharon Hedley.

Value

a list with elements

`model`	the fitted model object
`pred.var`	covariances of the regions given in `pred.data`. Diagonal elements are the variances in order
`bootstrap`	logical, always `FALSE`

pred.data	as above
off.set	as above
model	the fitted model with the extra term
dsm.object	the original model, as above

Author(s)

David L. Miller

---

dsm.var.movblk *Variance estimation via parametric moving block bootstrap*

Description

Estimate the variance in abundance over an area using a moving block bootstrap. Two procedures are implemented, one incorporating detection function uncertainty, one not.

Usage

```
dsm.var.movblk(dsm.object, pred.data, n.boot, block.size, off.set,
  ds.uncertainty = FALSE, samp.unit.name = "Transect.Label",
  progress.file = NULL, bs.file = NULL, bar = TRUE)
```

Arguments

dsm.object	object returned from dsm.
pred.data	a data.frame that holds prediction points, must have the correct columns for other environmental covariates. It also MUST have a column called cell.area which gives the area for each prediction cell
n.boot	number of bootstrap resamples.
block.size	number of segments in each block.
off.set	offset to be used in the model, see offsets Note that this should NOT be log()'d.
ds.uncertainty	incorporate uncertainty in the detection function? See Details, below. Note that this feature is EXPERIMENTAL at the moment.
samp.unit.name	name sampling unit to resample (default 'Transect.Label').
progress.file	path to a file to be used (usually by Distance) to generate a progress bar (default NULL – no file written).
bs.file	path to a file to store each boostrap round. This stores all of the bootstrap results rather than just the summaries, enabling outliers to be detected and removed. (Default NULL).
bar	should a progress bar be printed to screen? (Default TRUE).

Details

Setting `ds.uncertainty=TRUE` will incorporate detection function uncertainty directly into the bootstrap. This is done by generating observations from the fitted detection function and then re-fitting a new detection function (of the same form), then calculating a new effective strip width. Rejection sampling is used to generate the observations (except in the half-normal case) so the procedure can be rather slow. Note that this is currently not supported with covariates in the detection function.

Setting `ds.uncertainty=FALSE` will incorporate detection function uncertainty using the delta method. This assumes that the detection function and the spatial model are INDEPENDENT. This is probably not reasonable.

Examples

```
## Not run:
library(Distance)
library(dsm)

# load the Gulf of Mexico dolphin data (see ?mexdolphins)
data(mexdolphins)

# fit a detection function and look at the summary
hr.model <- ds(mexdolphins$distdata, max(mexdolphins$distdata$distance),
               key = "hr", adjustment = NULL)
summary(hr.model)

# fit a simple smooth of x and y
mod1<-dsm(N~s(x,y), hr.model, mexdolphins$segdata, mexdolphins$obsdata)
summary(mod1)

# create an offset (in metres)
# each prediction cell is 444km2
off.set <- 444*1000*1000

# calculate the variance by 500 moving block bootstraps
mod1.movblk <- dsm.var.movblk(mod1, mexdolphins$preddata, n.boot = 500,
   block.size = 3, samp.unit.name = "Transect.Label", off.set = off.set,
   bar = TRUE, bs.file = "mexico-bs.csv", ds.uncertainty = TRUE)

## End(Not run)
```

`dsm.var.prop` *Variance propogation for DSM models*

Description

Rather than use a bootstrap to calculate the variance in a `dsm` model, use the clever variance propogation trick from Williams et al. (2011).

Usage

```
dsm.var.prop(dsm.obj, pred.data, off.set, seglen.varname = "Effort",
  type.pred = "response")
```

Arguments

dsm.obj	an object returned from running dsm.
pred.data	either: a single prediction grid or list of prediction grids. Each grid should be a data.frame with the same columns as the original data.
off.set	a a vector or list of vectors with as many elements as there are in pred.data. Each vector is as long as the number of rows in the corresponding element of pred.data. These give the area associated with each prediction point.
seglen.varname	name for the column which holds the segment length (default value "Effort").
type.pred	should the predictions be on the "response" or "link" scale? (default "response").

Details

The idea is to refit the spatial model but including the Hessian of the offset as an extra term. Variance estimates using this new model can then be used to calculate the variance of abundance estimates which incorporate detection function uncertainty. Further mathematical details are given in the paper in the references below.

Many prediction grids can be supplied by supplying a list of data.frames to the function.

Note that this routine is only useful if a detection function has been used in the DSM.

Based on (much more general) code from Mark Bravington and Sharon Hedley.

Value

a list with elements

model	the fitted model object
pred.var	covariances of the regions given in pred.data. Diagonal elements are the variances in order
bootstrap	logical, always FALSE
pred.data	as above
off.set	as above
model	the fitted model with the extra term
dsm.object	the original model, as above
model.check	simple check of subtracting the coefficients of the two models to see if there is a large difference
deriv	numerically calculated Hessian of the offset

.

Author(s)

Mark V. Bravington, Sharon L. Hedley. Bugs added by David L. Miller.

References

Williams, R., Hedley, S.L., Branch, T.A., Bravington, M.V., Zerbini, A.N. and Findlay, K.P. (2011). Chilean Blue Whales as a Case Study to Illustrate Methods to Estimate Abundance and Evaluate Conservation Status of Rare Species. Conservation Biology 25(3), 526-535.

Examples

```
## Not run:
 library(Distance)
 library(dsm)

 # load the Gulf of Mexico dolphin data (see ?mexdolphins)
 data(mexdolphins)

 # fit a detection function and look at the summary
 hr.model <- ds(mexdolphins$distdata, max(mexdolphins$distdata$distance),
                key = "hr", adjustment = NULL)
 summary(hr.model)

 # fit a simple smooth of x and y
 mod1 <- dsm(N~s(x,y), hr.model, mexdolphins$segdata, mexdolphins$obsdata)

 # Calculate the offset...
 off.set <- 444*1000*1000

 # Calculate the variance
 mod1.var <- dsm.var.prop(mod1, mexdolphins$pred, off.set)

 # summary over the whole area in mexdolphins$pred

 # Plot a map of the CV
 #   need to format the prediction data with split
 mod1.var.map <- dsm.var.prop(mod1,
                 split(mexdolphins$pred,1:nrow(mexdolphins$pred)), off.set)
 plot(mod1.var.map)

## End(Not run)
```

`generate.ds.uncertainty`

Generate data from a fitted detection function

Description

When `ds.uncertainty` is `TRUE`, this procedure generates data from the fitted detection function (assuming that it is correct).

Usage

```
generate.ds.uncertainty(ds.object)
```

Arguments

ds.object	a fitted detection function object (as returned by a call to ddf.ds().

Note

This function changes the random number generator seed. To avoid any potential side-effects, use something like: seed <- get(".Random.seed",envir=.GlobalEnv) ### Run some code assign(".Random.seed",seed,envir=.GlobalEnv)

Author(s)

David L. Miller

---

generate.mb.sample *Moving block bootstrap sampler*

Description

Not usually used on its own, called from within dsm.var.movblk.

Usage

```
generate.mb.sample(num.blocks.required, block.size, which.blocks, dsm.data,
  unit.info, n.units)
```

Arguments

num.blocks.required	number of blocks that we need.
block.size	number of segments per block.
which.blocks	which blocks should be sampled.
dsm.data	the $data element of the result of a call to dsm.
unit.info	result of calling block.info.per.su.
n.units	number of sampling units.

Value

vector of log-residuals

latlong2km — *Convert latitude and longitude to Northings and Eastings*

Description

Convert longitude and latitude co-ordinates to kilometres west-east and south-north from axes through (lon0,lat0) using the "spherical law of cosines".

Usage

```
latlong2km(lon, lat, lon0 = sum(range(lon))/2, lat0 = sum(range(lat))/2)
```

Arguments

lon	longitude
lat	latitude
lon0	longitude reference point (defaults to mean longitude)
lat0	latitude reference point (defaults to mean latitude)

Details

WARNING: This is an approximate procedure for converting between latitude/ longitude and Northing/Easting. Consider using projection conversions available in packages sp and rgdal for better results.

Value

list with elements km.e and km.n.

Author(s)

Simon N. Wood

make.soapgrid — *Create a knot grid for the internal part of a soap film smoother.*

Description

This routine simply creates a grid of knots (in the correct format) to be used as in the "internal" part of the soap film smoother

Usage

```
make.soapgrid(bnd, n.grid)
```

Arguments

bnd	list with elements x and y which give the locations of the boundary vertices. The first and last elements should be the same.
n.grid	either one number giving the number of points along the x and y axes that should be used to create the grid, or a vector giving the number in the x direction, then y direction.

Value

a list with elements x and y, containing the knot locations.

Author(s)

David L Miller

---

matrixnotposdef.handler

Handler to suppress the "matrix not positive definite" warning

---

Description

Internal function to suppress an annoying warnings from chol()

Usage

```
matrixnotposdef.handler(w)
```

Arguments

w	a warning

Details

See: https://stat.ethz.ch/pipermail/r-help/2012-February/302407.html See: http://romainfrancois.blog.free.fr/index.php?post/
specific-warnings

Value

not a warning if the warning was "matrix not positive definite" or "the matrix is either rank-deficient or indefinite"

Author(s)

David L. Miller

`mexdolphins` *Pan-tropical spotted dolphins*

Description

Data from a combination of several NOAA shipboard surveys conducted on pan-tropical spotted dolphins in the Gulf of Mexico. 47 observations of groups of dolphins The group size was recorded, as well as the Beaufort sea state at the time of the observation. Coordinates for each observation and bathymetry data were also available as covariates for the analysis. A complete example analysis is provided at https://github.com/dill/dsm/wiki/mexico-analysis.

Format

List of 5 $ obsdata :'data.frame': 47 obs. of 5 variables: ..$ object : int [1:47] 45 61 63 85 114 120 146 165 187 214 ... ..$ Sample.Label: Factor w/ 387 levels "19960417-1","19960417-2",..: 45 61 63 85 114 120 146 165 187 214 ... ..$ size : num [1:47] 21 150 125 75 50 45 40 220 100 35 ... ..$ distance : num [1:47] 3297 929 6051 5500 7259 ... ..$ Effort : num [1:47] 36300 17800 21000 21800 13400 20900 20600 21700 29400 22900 ... $ preddata :'data.frame': 1374 obs. of 7 variables: ..$ latitude : num [1:1374] 30.1 30.1 30.1 30.1 30.1 ... ..$ longitude: num [1:1374] -87.6 -87.4 -87.2 -87.1 -86.9 ... ..$ depth : num [1:1374] 35 30 27 22 46 14 38 58 100 156 ... ..$ x : num [1:1374] 70832 86868 102904 118940 134976 ... ..$ y : num [1:1374] 341079 341079 341079 341079 341079 ... ..$ width : num [1:1374] 32072 32072 32072 32072 32072 ... ..$ height : num [1:1374] 37065 37065 37065 37065 37065 ... $ segdata :'data.frame': 387 obs. of 8 variables: ..$ latitude : num [1:387] 29.9 29.8 29.8 29.7 29.6 ... ..$ longitude : num [1:387] -86.9 -86.8 -86.7 -86.7 -86.6 ... ..$ Effort : num [1:387] 13800 14000 14000 13900 13800 13800 14000 14000 13900 14800 ... ..$ Transect.Label: Factor w/ 45 levels "19960417","19960418",..: 1 1 1 1 1 1 1 1 1 2 ... ..$ Sample.Label : Factor w/ 387 levels "19960417-1","19960417-2",..: 1 2 3 4 5 6 7 8 9 10 ... ..$ depth : num [1:387] 135 148 152 164 180 ... ..$ x : num [1:387] 134159 143496 152050 161102 169553 ... ..$ y : num [1:387] 325561 314055 304324 293475 282984 ... $ distdata :'data.frame': 47 obs. of 10 variables: ..$ object : int [1:47] 45 61 63 85 114 120 146 165 187 214 ... ..$ size : num [1:47] 21 150 125 75 50 45 40 220 100 35 ... ..$ distance : num [1:47] 3297 929 6051 5500 7259 ... ..$ Effort : num [1:47] 36300 17800 21000 21800 13400 20900 20600 21700 29400 22900 ... ..$ detected : num [1:47] 1 1 1 1 1 1 1 1 1 1 ... ..$ beaufort : num [1:47] 4 4 2 1 3 5 1 5 2 5 ... ..$ latitude : num [1:47] 27.7 26 26 27.5 27.4 ... ..$ longitude: num [1:47] -86 -87.6 -87.9 -90.4 -95 ... ..$ x : num [1:47] 228139 69199 37046 -210016 -658878 ... ..$ y : num [1:47] 79258 -113083 -112197 54208 43337 ... $ survey.area:'data.frame': 45 obs. of 2 variables: ..$ longitude: num [1:45] -87.4 -94.9 -96.3 -96.5 -96.6 ... ..$ latitude : num [1:45] 25.8 25.8 25.8 26.2 27.2 ...

References

Halpin, P.N., A.J. Read, E. Fujioka, B.D. Best, B. Donnelly, L.J. Hazen, C. Kot, K. Urian, E. LaBrecque, A. Dimatteo, J. Cleary, C. Good, L.B. Crowder, and K.D. Hyrenbach. 2009. OBIS-SEAMAP: The world data center for marine mammal, sea bird, and sea turtle distributions. Oceanography 22(2):104-115

NOAA Southeast Fisheries Science Center. 1996. Report of a Cetacean Survey of Oceanic and Selected Continental Shelf Waters of the Northern Gulf of Mexico aboard NOAA Ship Oregon II (Cruise 220)

`offsets` *Offsets*

Description

This will be documentation on calculating offsets.

`plot.dsm` *Plot a density surface model.*

Description

See `plot.gam`.

Usage

```
## S3 method for class 'dsm'
plot(x, ...)
```

Arguments

`x`	a `dsm` object
`...`	other arguments passed to `plot.gam`.

Value

a plot!

Author(s)

David L. Miller

See Also

dsm plot.gam

`plot.dsm.var` *Create plots of abundance uncertainty*

Description

Note that the prediction data set must have `x` and `y` columns even if these were not used in the model.

Usage

```
## S3 method for class 'dsm.var'
plot(x, poly = NULL, limits = NULL, breaks = NULL,
  legend.breaks = NULL, xlab = "x", ylab = "y", observations = TRUE,
  plot = TRUE, boxplot.coef = 1.5, x.name = "x", y.name = "y",
  gg.grad = NULL, ...)
```

Arguments

`x`	a `dsm.var` object
`poly`	a `list` or `data.frame` with columns `x` and `y`, which gives the coordinates of a polygon to draw. It may also optionally have a column `group`, if there are many polygons.
`limits`	limits for the fill colours
`breaks`	breaks for the colour fill
`legend.breaks`	breaks as they should be displayed
`xlab`	label for the `x` axis
`ylab`	label for the `y` axis
`observations`	should observations be plotted?
`plot`	actually plot the map, or just return a `ggplot2` object?
`boxplot.coef`	control trimming (as in `summary.dsm.var`), only has an effect if the bootstrap file was saved.
`x.name`	name of the variable to plot as the x axis.
`y.name`	name of the variable to plot as the y axis.
`gg.grad`	optional `ggplot` gradient object.
`...`	any other arguments

Value

a plot

Details

In order to get plotting to work with dsm.var.prop and dsm.var.gam, one must first format the data correctly since these functions are designed to compute very general summaries. One summary is calculated for each element of the list pred supplied to dsm.var.prop and dsm.var.gam.

For a plot of uncertainty over a prediction grid, pred (a data.frame), say, we can create the correct format by simply using pred.new <- split(pred,1:nrow(pred)).

Author(s)

David L. Miller

See Also

dsm.var.prop, dsm.var.gam, dsm.var.movblk

---

predict.dsm *Predict from a fitted density surface model*

Description

Make predictions outside (or inside) the covered area.

Usage

```
## S3 method for class 'dsm'
predict(object, newdata = NULL, off.set = NULL,
  type = "response", ...)
```

Arguments

object	a fitted dsm object as produced by dsm().
newdata	spatially referenced covariates e.g. altitude, depth, distance to shore, etc. Note covariates in this dataframe must have names *identical* to variable names used in fitting the DSM.
off.set	area of each of the cells in the prediction grid. Ignored if their is already a column in newdata called off.set.
type	what scale should the results be on. The default is "response", see predict.gam for an explanation of other options (usually not necessary).
...	any other arguments passed to predict.gam.

Value

predicted values on the response scale (density/abundance).

Author(s)

David L. Miller

print.dsm *Print a description of a density surface model object*

Description

This method just gives a short description of the fitted model. Use the summary.dsm method for more information.

Usage

```
## S3 method for class 'dsm'
print(x, ...)
```

Arguments

x	a dsm object
...	unspecified and unused arguments for S3 consistency

Author(s)

David L. Miller

See Also

summary.ds

print.dsm.var *Print a description of a density surface model variance object*

Description

This method only provides a short summary, use the summary.dsm.var method for information.

Usage

```
## S3 method for class 'dsm.var'
print(x, ...)
```

Arguments

x	a dsm variance object
...	unspecified and unused arguments for S3 consistency

Author(s)

David L. Miller

See Also

summary.dsm.var

---

print.summary.dsm.var *Print summary of density surface model variance object*

Description

See summary.dsm.var for information.

Usage

```
## S3 method for class 'summary.dsm.var'
print(x, ...)
```

Arguments

x	a summary of dsm variance object
...	unspecified and unused arguments for S3 consistency

Author(s)

David L. Miller

See Also

summary.dsm.var

---

rqgam.check *Randomised quantile residuals check plots for GAMs/DSMs*

Description

Function operates as gam.check but using randomised quantile residuals, a la Dunn and Smyth (1996). Checks of k are not computed, these need to be done using gam.check.

Usage

```
rqgam.check(gam.obj, ...)
```

Arguments

gam.obj	a gam, glm or dsm object.
...	arguments passed on to all plotting function

Value

just plots!

Author(s)

Based on code provided by Natalie Kelly, bugs added by Dave Miller

Examples

```
library(Distance)
library(dsm)

# load the Gulf of Mexico dolphin data (see ?mexdolphins)
data(mexdolphins)

# fit a detection function and look at the summary
hr.model <- ds(mexdolphins$distdata, max(mexdolphins$distdata$distance),
               key = "hr", adjustment = NULL)

# fit a simple smooth of x and y
mod1<-dsm(N~s(x,y), hr.model, mexdolphins$segdata, mexdolphins$obsdata)
rqgam.check(mod1)
```

`summary.dsm` *Summarize a fitted density surface model*

Description

Gives a brief summary of a fitted `dsm` object.

Usage

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

Arguments

`object`	a `dsm` object
`...`	other arguments passed to `summary.gam`.

Value

a summary object

Author(s)

David L. Miller

See Also

dsm

`summary.dsm.var` *Summarize the variance of a density surface model*

Description

Gives a brief summary of a fitted `dsm` variance object.

Usage

```
## S3 method for class 'dsm.var'
summary(object, alpha = 0.05, boxplot.coef = 1.5,
  bootstrap.subregions = NULL, ...)
```

Arguments

`object`	a `dsm.var` object
`alpha`	alpha level for confidence intervals
`boxplot.coef`	the value of `coef` used to calculate the outliers see `boxplot`.
`bootstrap.subregions`	list of vectors of logicals or indices for subregions for which variances need to be calculated (only for bootstraps (see `dsm.var.prop` for how to use subregions with variance propagation).
`...`	unused arguments for S3 compatibility

Value

a summary object

Author(s)

David L. Miller

See Also

dsm.var.movblk dsm.var.prop

`trim.var` *Trimmed variance*

Description

Trim the variance estimates from the bootstrap. This is defined as the percentage defined as amount necessary to bring median and trimmed mean within 8

Usage

```
trim.var(untrimmed.bootstraps, boxplot.coef = 1.5)
```

Arguments

`untrimmed.bootstraps`	(usually the `$study.area.total` element of a returned `dsm` bootstrap object.
`boxplot.coef`	the value of `coef` used to calculate the outliers see `boxplot`.

Details

Code originally by Louise Burt.

Value

trimmed variance

Author(s)

Louise Burt

Index

∗Topic **datasets**
    mexdolphins, 17
∗Topic **utility**
    print.dsm, 21
    print.dsm.var, 21
    print.summary.dsm.var, 22

bam, *4–6*
block.info.per.su, 3, *14*
boxplot, *24, 25*

check.cols, 3
cor, *8*

ddf, *4*
ds, *4*
dsm, *4*, 4, *7*, *9, 10*, *12*, *20*
dsm-data, 7
dsm-package, 2
dsm.cor, 7
dsm.var.gam, 9, *20*
dsm.var.movblk, 10, *14*
dsm.var.prop, *9*, 11, *20*, *24*

gam, *4–6*
gam.check, *22*
gamm, *4–6*
generate.ds.uncertainty, 13
generate.mb.sample, 14
ggplot, *19*
glm, *4–6*

latlong2km, 15

make.soapgrid, 15
matrixnotposdef.handler, 16
mexdolphins, 17

negbin, *5*

offsets, *10*, 18

plot.dsm, 18
plot.dsm.var, 19
plot.gam, *18*
predict.dsm, 20
predict.gam, *20*
print.dsm, 21
print.dsm.var, 21
print.summary.dsm.var, 22

quasipoisson, *5*

residuals.gam, *8*
rqgam.check, 22

summary.ds, *21*
summary.dsm, *21*, 23
summary.dsm.var, *21, 22*, 24
summary.gam, *23*

trim.var, 25
Tweedie, *5*