

Package ‘ubms’

February 2, 2021

Version 1.0.2

Date 2021-02-01

Title Bayesian Models for Data from Unmarked Animals using 'Stan'

Depends R (>= 3.4.0), unmarked

Imports ggplot2 (>= 2.0.0), gridExtra, lme4, loo, Matrix, methods,
Rcpp (>= 0.12.0), rstan (>= 2.18.1), rstantools (>= 2.0.0),
stats

Suggests covr, devtools, knitr, pkgdown, raster, rmarkdown, roxygen2,
testthat

VignetteBuilder knitr

Description Fit Bayesian hierarchical models of animal abundance and occurrence via the 'rstan' package, the R interface to the 'Stan' C++ library. Supported models include single-season occupancy, dynamic occupancy, and N-mixture abundance models. Covariates on model parameters are specified using a formula-based interface similar to package 'unmarked', while also allowing for estimation of random slope and intercept terms. References: Carpenter et al. (2017) <doi:10.18637/jss.v076.i01>; Fiske and Chandler (2011) <doi:10.18637/jss.v043.i10>.

License GPL (>= 3)

URL <https://kenkellner.com/ubms/>

BugReports <https://github.com/kenkellner/ubms/issues>

Encoding UTF-8

RoxygenNote 7.1.1

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppArmadillo (>= 0.9.300.2.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

SystemRequirements GNU make

Collate 'RcppExports.R' 'submodel.R' 'response.R' 'inputs.R' 'fit.R' 'posterior_predict.R' 'posterior_linpred.R' 'fitted.R' 'gof.R'

'occu.R' 'colect.R' 'missing.R' 'distamp.R' 'fitlist.R'
 'occuRN.R' 'mb_chisq.R' 'multinomPois.R' 'occuTTD.R' 'pcount.R'
 'plot_marginal.R' 'predict.R' 'ranef.R' 'residuals.R'
 'stanmodels.R' 'ubms-package.R' 'ubmsFit-methods.R'
 'ubmsFitList-methods.R' 'umf.R' 'utils.R'

NeedsCompilation yes

Author Ken Kellner [cre, aut]

Maintainer Ken Kellner <contact@kenkellner.com>

Repository CRAN

Date/Publication 2021-02-02 00:10:13 UTC

R topics documented:

coef,ubmsFit-method	3
extract,ubmsFit-method	3
fitList,ubmsFit-method	4
fitted,ubmsFit-method	4
getY,ubmsFit-method	5
gof	5
loo,ubmsFit-method	6
modSel,ubmsFitList-method	7
names,ubmsFit-method	7
names,ubmsFitList-method	8
nsamples,ubmsFit-method	8
plot,ubmsFit,ANY-method	9
plot_marginal,ubmsFit-method	9
plot_residuals,ubmsFit-method	10
posterior_linpred,ubmsFit-method	11
posterior_predict,ubmsFit-method	12
predict,ubmsFit-method	13
projected	14
ranef,ubmsFit-method	15
residuals,ubmsFit-method	15
stan_colect	16
stan_distsamp	17
stan_multinomPois	19
stan_occu	20
stan_occuRN	21
stan_occuTTD	22
stan_pcount	23
summary,ubmsFit-method	25
traceplot,ubmsFit-method	25
turnover	26
ubms	26
ubmsFitList-extractors	27
waic,ubmsFit-method	27

coef,ubmsFit-method 3

[,ubmsFit,character,missing,missing-method 28
[,ubmsSubmodelList,character,missing,missing-method 28

Index 29

coef,ubmsFit-method *Extract Coefficient Values From a ubmsFit Model*

Description

Extract Coefficient Values From a ubmsFit Model

Usage

```
## S4 method for signature 'ubmsFit'  
coef(object, ...)
```

Arguments

object	A ubmsFit model
...	Currently ignored

Value

A vector of coefficient values for all submodels.

extract,ubmsFit-method
 Extract Samples From a ubmsFit Model

Description

Extract samples from a ubmsFit model

Usage

```
## S4 method for signature 'ubmsFit'  
extract(object, pars, permuted = TRUE, inc_warmup = FALSE, include = TRUE)
```

Arguments

object	A ubmsFit object
pars	An optional character vector providing parameter names of interest. If not specified, all parameters are used
permuted	Logical. If TRUE, draws are permuted and merged; if FALSE, the original order is kept
inc_warmup	Logical. If TRUE, warmup iterations are included; if FALSE they are discarded.
include	Logical. If TRUE provided parameter names in pars are kept; if FALSE they are excluded.

Value

If permuted=TRUE, a list; if permuted=FALSE, an array.

fitList,ubmsFit-method

Create a List of ubmsFit Models

Description

Create a list of ubmsFit models

Usage

```
## S4 method for signature 'ubmsFit'
fitList(...)
```

Arguments

... ubmsFit model objects, preferably named

Value

An object of class ubmsFitList containing the list of models

fitted,ubmsFit-method *Extract Fitted Values*

Description

Extract fitted values for a given submodel from a ubmsFit object. Fitted values are calculated separately for each submodel using the posterior predictive distribution of the latent state z , following Wright et al. (2019).

Usage

```
## S4 method for signature 'ubmsFit'
fitted(object, submodel, draws = NULL, ...)
```

Arguments

object	A fitted model of class ubmsFit
submodel	Submodel to get fitted values for, options are "state" or "det"
draws	An integer indicating the number of draws to return. The default and maximum number of draws is the size of the posterior sample.
...	Currently ignored

Value

A matrix of fitted values with dimension draws by observations. Note that calculation of fitted values for the detection submodel is conditional on $z > 0$, so fitted values for an observation in a posterior draw where $z = 0$ are assigned value NA (Wright et al. 2019).

References

Wright, W. J., Irvine, K. M., & Higgs, M. D. (2019). Identifying occupancy model inadequacies: can residuals separately assess detection and presence? *Ecology*, 100(6), e02703.

getY,ubmsFit-method *Extract y, the Response Variable, From a ubmsFit Model*

Description

Extract y, the Response Variable, From a ubmsFit Model

Usage

```
## S4 method for signature 'ubmsFit'
getY(object)
```

Arguments

object A ubmsFit model

Value

A matrix containing the response variable y.

gof *Check model goodness-of-fit*

Description

Goodness-of-fit tests for ubmsFit models using posterior predictive checks

Usage

```
gof(object, draws = NULL, ...)

## S4 method for signature 'ubmsFitOccu'
gof(object, draws = NULL, quiet = FALSE, ...)

## S4 method for signature 'ubmsFitAbun'
gof(object, draws = NULL, quiet = FALSE, ...)
```

Arguments

object	A fitted model of class ubmsFit
draws	Number of draws from the posterior to use in the check
...	Currently ignored
quiet	If TRUE, suppress progress bar

Value

An object of class ubmsGOF containing statistics calculated from the posterior predictive distribution.

Functions

- `gof,ubmsFitOccu-method`: Applies the MacKenzie-Bailey chi-square goodness of fit test for occupancy models (MacKenzie and Bailey 2004).
- `gof,ubmsFitAbun-method`: A goodness-of-fit test for N-mixture type models based on Pearson's chi-square.

References

MacKenzie, D. I., & Bailey, L. L. (2004). Assessing the fit of site-occupancy models. *Journal of Agricultural, Biological, and Environmental Statistics*, 9(3), 300-318.

loo,ubmsFit-method *Leave-one-out Cross Validation*

Description

Leave-one-out Cross Validation

Usage

```
## S4 method for signature 'ubmsFit'
loo(x, ..., cores = getOption("mc.cores", 1))
```

Arguments

x	A ubmsFit model
...	Currently ignored
cores	Number of cores to use for calculation

Value

A named list of class loo with estimates of the expected log predictive density and other parameters used for model comparison. See `?loo::loo` for more information.

modSel,ubmsFitList-method

Model Selection For a List of ubmsFit Models

Description

Construct a model selection table from a ubmsFitList

Usage

```
## S4 method for signature 'ubmsFitList'  
modSel(object, ...)
```

Arguments

object	An object of class ubmsFitList
...	Currently ignored

Value

A data.frame of model fit information with one row per model in the input ubmsFitList. Models are ranked in descending order by expected log pointwise predictive density (elpd).

See Also

[loo](#), [loo_compare](#)

names,ubmsFit-method *Get Parameter Names From a ubmsFit Model*

Description

Get Parameter Names From a ubmsFit Model

Usage

```
## S4 method for signature 'ubmsFit'  
names(x)
```

Arguments

x	A ubmsFit model
---	-----------------

Value

A character vector of parameter names.

names,ubmsFitList-method

Get Names of Models in a ubmsFitList

Description

Get Names of Models in a ubmsFitList

Usage

```
## S4 method for signature 'ubmsFitList'
names(x)
```

Arguments

x A ubmsFitList object

Value

A character vector of model names.

nsamples,ubmsFit-method

Get number of Posterior Samples Stored in a ubmsFit Model

Description

Get number of Posterior Samples Stored in a ubmsFit Model

Usage

```
## S4 method for signature 'ubmsFit'
nsamples(object, ...)
```

Arguments

object A ubmsFit model
 ... Currently ignored

Value

An integer representing the number of posterior samples

 plot,ubmsFit,ANY-method

Plot Residuals For All Submodels in a ubmsFit Model

Description

Plot Residuals For All Submodels in a ubmsFit Model

Usage

```
## S4 method for signature 'ubmsFit,ANY'
plot(x, y, ...)
```

Arguments

x	A ubmsFit model
y	Currently ignored
...	Currently ignored

Value

A plot object of class gtable with one panel per submodel.

plot_marginal,ubmsFit-method

Plot Marginal Effects of Covariates

Description

Generates marginal fixed effects plots of one or more covariates from a ubmsFit submodel. For each plot, the focal covariate is allowed to vary across its range (or possible discrete values, for a factor), while the other covariates are held at their means or reference levels. Random effects are ignored.

Usage

```
## S4 method for signature 'ubmsFit'
plot_marginal(object, submodel, covariate = NULL, level = 0.95, ...)
```

Arguments

object	A fitted model of class ubmsFit
submodel	Submodel to get plots for, for example "det"
covariate	Plot a specific covariate; provide the name as a string
level	Probability mass to include in the uncertainty interval
...	Currently ignored

Value

A ggplot, potentially with multiple panels, one per covariate

```
plot_residuals,ubmsFit-method
```

Plot Model Residuals

Description

Plot residuals for a submodel from a `ubmsFit` object, for multiple posterior draws. By default, residuals are plotted against fitted values. When the submodel has a binomial response (e.g., detection models), regular residual plots are not typically informative. Instead, the residuals and fitted values are divided into bins based on fitted value and the averages are plotted. For a count response (e.g., Poisson), Pearson residuals are calculated. To plot residuals against values of a particular covariate instead of the fitted values, supply the name of the covariate (as a string) to the `covariate` argument.

Usage

```
## S4 method for signature 'ubmsFit'
plot_residuals(
  object,
  submodel,
  covariate = NULL,
  draws = 9,
  nbins = NULL,
  ...
)
```

Arguments

<code>object</code>	A fitted model of class <code>ubmsFit</code>
<code>submodel</code>	Submodel to plot residuals for, for example "det"
<code>covariate</code>	If specified, plot residuals against values of a covariate. Covariate name should be provided as a string. If <code>NULL</code> , residuals are plotted against predicted values.
<code>draws</code>	An integer indicating the number of posterior draws to use. Separate plots are generated for each draw, so this number should be relatively small. The default and maximum number of draws is the size of the posterior sample.
<code>nbins</code>	For submodels with a binomial response, manually set the number of bins to use
<code>...</code>	Currently ignored

Value

A ggplot of residuals vs. fitted values or covariate values, with one panel per posterior draw. For binned residual plots, the shaded area represents plus/minus two standard deviations around the mean residual. If the model is true, we would expect about 95 fall within this area.

See Also[residuals](#)

`posterior_linpred,ubmsFit-method`*Posterior Distribution of the Linear Predictor*

Description

Extract posterior draws of the linear predictor for a `ubmsFit` submodel, possibly transformed by the inverse-link function.

Usage

```
## S4 method for signature 'ubmsFit'
posterior_linpred(
  object,
  transform = FALSE,
  submodel,
  newdata = NULL,
  draws = NULL,
  re.form = NULL,
  ...
)
```

Arguments

<code>object</code>	A fitted model of class <code>ubmsFit</code>
<code>transform</code>	Should the linear predictor be transformed using the inverse link function?
<code>submodel</code>	The name of the submodel, as a character string, for which to calculate the linear predictor
<code>newdata</code>	Optional data frame of newdata to use when calculating the linear predictor. If not provided, the model matrix is used.
<code>draws</code>	An integer indicating the number of draws to return. The default and maximum number of draws is the size of the posterior sample.
<code>re.form</code>	If <code>NULL</code> , any estimated group-level parameters ("random effects") are included. If <code>NA</code> , they are ignored
<code>...</code>	Currently ignored

Value

A matrix of simulations from the posterior predictive distribution of the linear predictor. The dimensions are draws by number of linear predictor values (e.g., number of sites or number of observations).

 posterior_predict,ubmsFit-method

Draw from the posterior predictive distribution

Description

Draw from the posterior predictive distribution after fitting a model. You can draw from the posterior of the observed outcome y or the latent unobserved state z .

Usage

```
## S4 method for signature 'ubmsFit'
posterior_predict(
  object,
  param = c("y", "z"),
  draws = NULL,
  re.form = NULL,
  ...
)
```

Arguments

object	A fitted model of class ubmsFit
param	Either "y" for the observed outcome or "z" for the unobserved latent state
draws	An integer indicating the number of draws to return. The default and maximum number of draws is the size of the posterior sample.
re.form	If NULL, any estimated group-level parameters ("random effects") are included. If NA, they are ignored
...	Currently ignored

Value

A matrix of simulations from the posterior predictive distribution. If `param = "z"`, the dimensions are draws by number of sites (or sites x primary periods in site-major order for dynamic models). If `param = "y"`, the dimensions are draws by sites x observations (or sites x primary periods x observations for dynamic models).

 predict,ubmsFit-method

Predict parameter values from a fitted model

Description

This method generates predicted parameter values for the original dataset or a new dataset using the posterior distribution. Standard deviation and a customizable uncertainty interval are also calculated.

Usage

```
## S4 method for signature 'ubmsFit'
predict(
  object,
  submodel,
  newdata = NULL,
  transform = TRUE,
  re.form = NULL,
  level = 0.95,
  ...
)
```

Arguments

object	A fitted model of class ubmsFit
submodel	Submodel to predict from, for example "det"
newdata	Optional data frame or RasterStack of covariates to generate predictions from. If not provided (the default), predictions are generated from the original data
transform	If TRUE, back-transform the predictions to their original scale
re.form	If NULL, any estimated group-level parameters ("random effects") are included. If NA, they are ignored
level	Probability mass to include in the uncertainty interval
...	Currently ignored

Value

If newdata was a data frame: A data frame with one row per prediction and four columns: 1) Predicted point estimates (posterior means), 2) Standard deviation of the posterior, 3-4) Lower and upper bounds of the specified uncertainty interval

For parameters with more than one dimension, the rows are in site-major order, or site-year-observation for dynamic models.

If newdata was a RasterStack, returns a RasterStack with four layers corresponding to the four columns above with the same projection as the original RasterStack.

See Also

posterior_linpred, posterior_interval

projected

Projected Occupancy Trajectories

Description

Generate posterior draws of occupancy probability for all sites and primary periods, i.e. the projected trajectory (Weir et al. 2009).

Usage

```
projected(object, ...)
```

```
## S4 method for signature 'ubmsFitColect'
projected(object, draws = NULL, re.form = NULL, ...)
```

Arguments

object	A fitted dynamic occupancy model of class inheriting <code>ubmsFit</code>
...	Currently ignored
draws	Number of draws from the posterior to use in the check
re.form	If NULL, any estimated group-level parameters ("random effects") are included. If NA, they are ignored

Value

A matrix of occupancy values from the posterior predictive distribution. The dimensions are draws by number of sites x primary periods in site-major order.

References

Weir LA, Fiske IJ, Royle J. 2009. Trends in Anuran Occupancy from Northeastern States of the North American Amphibian Monitoring Program. *Herpetological Conservation and Biology*. 4: 389-402.

See Also

[stan_colect](#)

ranef,ubmsFit-method *Extract Random Effects*

Description

Extract random effects from a ubmsFit model. Note that this function works like ranef for merMod objects from lme4, not like ranef for unmarkedFit objects. To get functionality similar to that of unmarkedFit, use posterior_predict.

Usage

```
## S4 method for signature 'ubmsFit'
ranef(object, submodel, summary = FALSE, ...)
```

Arguments

object	A fitted model of class ubmsFit
submodel	The name of the submodel, as a character string, for which to generate the random effects
summary	If TRUE, calculate mean, SD, and 95 for each random effect term
...	Currently ignored

Value

If summary=FALSE, a list of random effect values; if TRUE, a data frame with columns for random effect mean, SD, and 95

See Also

[ranef](#), [posterior_predict](#)

residuals,ubmsFit-method
Extract Model Residuals

Description

Extract residuals for a given submodel from a ubmsFit object. Residuals are calculated separately for each submodel using the posterior predictive distribution of the latent state z , following Wright et al. (2019).

Usage

```
## S4 method for signature 'ubmsFit'
residuals(object, submodel, draws = NULL, ...)
```

Arguments

object	A fitted model of class <code>ubmsFit</code>
submodel	Submodel to get residuals for, for example "det"
draws	An integer indicating the number of draws to return. The default and maximum number of draws is the size of the posterior sample.
...	Currently ignored

Value

A matrix of residual values with dimension draws by observations. Note that calculation of residuals for the detection submodel is conditional on $z > 0$, so residuals for an observation in a posterior draw where $z = 0$ are assigned value NA (Wright et al. 2019).

References

Wright, W. J., Irvine, K. M., & Higgs, M. D. (2019). Identifying occupancy model inadequacies: can residuals separately assess detection and presence? *Ecology*, 100(6), e02703.

 stan_colext

Fit the MacKenzie et al. (2003) Dynamic Occupancy Model

Description

This function fits the dynamic occupancy model of MacKenzie et al. (2003).

Usage

```
stan_colext(
  psiformula = ~1,
  gammaformula = ~1,
  epsilonformula = ~1,
  pformula = ~1,
  data,
  ...
)
```

Arguments

psiformula	Right-hand sided formula for the initial probability of occupancy at each site
gammaformula	Right-hand sided formula for colonization probability
epsilonformula	Right-hand sided formula for extinction probability
pformula	Right-hand sided formula for detection probability
data	A <code>unmarkedMultFrame</code> object
...	Arguments passed to the <code>stan</code> call, such as number of chains <code>chains</code> or iterations <code>iter</code>

Value

ubmsFitColect object describing the model fit.

References

MacKenzie DI, Nicholas JD, Hines JE, Knutson MG, Franklin AB. 2003. Ecology 84: 2200-2207.

See Also

[colect](#), [unmarkedMultFrame](#)

Examples

```
data(frogs)
umf <- formatMult(masspcru)
umf@y[umf@y > 1] <- 1 #convert counts to presence/absence
umf <- umf[1:100,] #Use only 100 sites

fit_frog <- stan_colect(~1, ~1, ~1, ~1, umf, chains=3, iter=300)
```

 stan_distsamp

Fit the Royle et al. (2004) Distance Sampling Model

Description

This function fits the hierarchical distance sampling model of Royle et al. (2004) to line or point transect data recorded in discrete distance intervals.

Usage

```
stan_distsamp(
  formula,
  data,
  keyfun = c("halfnorm", "exp", "hazard"),
  output = c("density", "abund"),
  unitsOut = c("ha", "kmsq"),
  ...
)
```

Arguments

formula	Double right-hand side formula describing covariates of detection and occupancy in that order
data	A unmarkedFrameDS object

keyfun	One of the following detection functions: "halfnorm" for half-normal, "exp" for negative exponential, or "hazard" for hazard-rate (see warning below)
output	Model either density "density" or abundance "abund"
unitsOut	Units of density. Either "ha" or "kmsq" for hectares and square kilometers, respectively
...	Arguments passed to the <code>stan</code> call, such as number of chains <code>chains</code> or iterations <code>iter</code>

Value

`ubmsFitDistsamp` object describing the model fit.

Warning

Use of the hazard-rate key function ("hazard") typically requires a large sample size in order to get good parameter estimates. If you have a relatively small number of points/transects (<100), you should be cautious with the resulting models. Check your results against estimates from `unmarked`, which doesn't require as much data to get good estimates of the hazard-rate shape and scale parameters.

Note

Values of 'dist.breaks' in the 'unmarkedFrameDS' should be as small as possible (<10) to facilitate convergence. Consider converting 'unitsIn' from meters to kilometers, for example. See example below.

References

Royle, J. A., Dawson, D. K., & Bates, S. (2004). Modeling abundance effects in distance sampling. *Ecology* 85: 1591-1597.

See Also

[distsamp](#), [unmarkedFrameDS](#)

Examples

```
data(issj)
#Note use of km instead of m for distance breaks
jayUMF <- unmarkedFrameDS(y=as.matrix(issj[,1:3]),
                          siteCovs=issj[,c("elevation", "forest")],
                          dist.breaks=c(0,0.1,0.2,0.3),
                          unitsIn="km", survey="point")

fm_jay <- stan_distsamp(~1~scale(elevation), jayUMF, chains=3, iter=300)
```

stan_multinomPois *Fit the Multinomial-Poisson Mixture Model*

Description

This function fits the multinomial-Poisson mixture model, useful for data collected via survey methods such as removal or double observer sampling.

Usage

```
stan_multinomPois(formula, data, ...)
```

Arguments

formula	Double right-hand side formula describing covariates of detection and abundance in that order
data	A unmarkedFrameMPois object
...	Arguments passed to the stan call, such as number of chains chains or iterations iter

Value

ubmsFitMultinomPois object describing the model fit.

See Also

[multinomPois](#), [unmarkedFrameMPois](#)

Examples

```
data(ovendata)
ovenFrame <- unmarkedFrameMPois(ovendata.list$data,
                                siteCovs=ovendata.list$covariates,
                                type="removal")

oven_fit <- stan_multinomPois(~1~scale(ufc), ovenFrame, chains=3, iter=300)
```

`stan_occu`*Fit the MacKenzie et al. (2002) Occupancy Model*

Description

This function fits the single season occupancy model of MacKenzie et al. (2002).

Usage

```
stan_occu(formula, data, ...)
```

Arguments

<code>formula</code>	Double right-hand side formula describing covariates of detection and occupancy in that order
<code>data</code>	A <code>unmarkedFrameOccu</code> object
<code>...</code>	Arguments passed to the <code>stan</code> call, such as number of chains <code>chains</code> or iterations <code>iter</code>

Value

`ubmsFitOccu` object describing the model fit.

References

MacKenzie DI, Nichols JD, Lachman GB, Droege S, Royle JA, Langtimm CA. 2002. Estimating site occupancy rates when detection probabilities are less than one. *Ecology* 83: 2248-2255.

See Also

[occu](#), [unmarkedFrameOccu](#)

Examples

```
data(frogs)
pferUMF <- unmarkedFrameOccu(pfer.bin)

#Add some covariates
siteCovs(pferUMF) <- data.frame(cov1=rnorm(numSites(pferUMF)))

#Fit model
(fm <- stan_occu(~1~cov1, pferUMF, chains=3, iter=300))
```

stan_occuRN *Fit the Occupancy Model of Royle and Nichols (2003)*

Description

Fit the occupancy model of Royle and Nichols (2003), which relates probability of detection of the species to the number of individuals available for detection at each site.

Usage

```
stan_occuRN(formula, data, K = 20, ...)
```

Arguments

formula	Double right-hand side formula describing covariates of detection and abundance in that order
data	A unmarkedFrameOccu object
K	Integer upper index of integration for N-mixture. This should be set high enough so that it does not affect the parameter estimates. Note that computation time will increase with K.
...	Arguments passed to the stan call, such as number of chains chains or iterations iter

Value

ubmsFitOccuRN object describing the model fit.

References

Royle JA, Nichols JD. 2003. Estimating abundance from repeated presence-absence data or point counts. *Ecology* 84: 777-790.

See Also

[occuRN](#), [unmarkedFrameOccu](#)

Examples

```
data(birds)
woodthrushUMF <- unmarkedFrameOccu(woodthrush.bin)
#Add a site covariate
siteCovs(woodthrushUMF) <- data.frame(cov1=rnorm(numSites(woodthrushUMF)))

(fm_wood <- stan_occuRN(~1~cov1, woodthrushUMF, chains=3, iter=300))
```

 stan_occuTTD

Fit Time-to-detection Occupancy Models

Description

Fit time-to-detection occupancy models of Garrard et al. (2008, 2013). Time-to-detection can be modeled with either an exponential or Weibull distribution.

Usage

```
stan_occuTTD(
  psiformula = ~1,
  gammaformula = ~1,
  epsilonformula = ~1,
  detformula = ~1,
  data,
  ttdDist = c("exp", "weibull"),
  linkPsi = c("logit"),
  ...
)
```

Arguments

psiformula	Right-hand sided formula for the initial probability of occupancy at each site.
gammaformula	Right-hand sided formula for colonization probability. Currently ignored as dynamic models are not yet supported.
epsilonformula	Right-hand sided formula for extinction probability. Currently ignored as dynamic models are not yet supported.
detformula	Right-hand sided formula for mean time-to-detection.
data	unmarkedFrameOccuTTD object that supplies the data (see unmarkedFrameOccuTTD).
ttdDist	Distribution to use for time-to-detection; either "exp" for the exponential, or "weibull" for the Weibull, which adds an additional shape parameter k .
linkPsi	Link function for the occupancy model. Only option is "logit" for now, in the future "cloglog" will be supported for the complimentary log-log link.
...	Arguments passed to the stan call, such as number of chains chains or iterations iter

Value

ubmsFitOccuTTD object describing the model fit.

References

- Garrard, G.E., Bekessy, S.A., McCarthy, M.A. and Wintle, B.A. 2008. When have we looked hard enough? A novel method for setting minimum survey effort protocols for flora surveys. *Austral Ecology* 33: 986-998.
- Garrard, G.E., McCarthy, M.A., Williams, N.S., Bekessy, S.A. and Wintle, B.A. 2013. A general model of detectability using species traits. *Methods in Ecology and Evolution* 4: 45-52.
- Kery, Marc, and J. Andrew Royle. 2016. *Applied Hierarchical Modeling in Ecology*, Volume 1. Academic Press.

See Also

[occuTTD](#), [unmarkedFrameOccuTTD](#)

Examples

```
#Simulate data
N <- 500; J <- 1
scovs <- data.frame(elev=c(scale(runif(N, 0,100))),
                   forest=runif(N,0,1),
                   wind=runif(N,0,1))

beta_psi <- c(-0.69, 0.71, -0.5)
psi <- plogis(cbind(1, scovs$elev, scovs$forest) %*% beta_psi)
z <- rbinom(N, 1, psi)

Tmax <- 10 #Same survey length for all observations
beta_lam <- c(-2, -0.2, 0.7)
rate <- exp(cbind(1, scovs$elev, scovs$wind) %*% beta_lam)
ttd <- rexp(N, rate)
ttd[z==0] <- Tmax #Censor at unoccupied sites
ttd[ttd>Tmax] <- Tmax #Censor when ttd was greater than survey length

#Build unmarkedFrame
umf <- unmarkedFrameOccuTTD(y=ttd, surveyLength=Tmax, siteCovs=scovs)

#Fit model
(fit <- stan_occuTTD(psiformula=~elev+forest, detformula=~elev+wind,
                   data=umf, chains=3, iter=300))
```

stan_pcount

Fit the N-mixture model of Royle (2004)

Description

This function fits the single season N-mixture model of Royle et al. (2004).

Usage

```
stan_pcount(formula, data, K = NULL, mixture = "P", ...)
```

Arguments

formula	Double right-hand side formula describing covariates of detection and abundance in that order
data	A unmarkedFramePCount object
K	Integer upper index of integration for N-mixture. This should be set high enough so that it does not affect the parameter estimates. Note that computation time will increase with K.
mixture	Character specifying mixture: "P" is only option currently.
...	Arguments passed to the stan call, such as number of chains chains or iterations iter

Value

ubmsFitPcount object describing the model fit.

References

Royle JA. 2004. N-mixture models for estimating populaiton size from spatially replicated counts. *Biometrics* 60: 105-108.

See Also

[pcount](#), [unmarkedFramePCount](#)

Examples

```
data(mallard)
mallardUMF <- unmarkedFramePCount(mallard.y, siteCovs=mallard.site)

(fm_mallard <- stan_pcount(~1~elev+forest, mallardUMF, K=30,
                           chains=3, iter=300))
```

 summary,ubmsFit-method

Extract Summary Statistics from a ubmsFit Model

Description

Extract Summary Statistics from a ubmsFit Model

Usage

```
## S4 method for signature 'ubmsFit'
summary(object, submodel, ...)
```

Arguments

object	A ubmsFit model
submodel	Name of submodel to summarize
...	Currently ignored

Value

An object of class `data.frame` containing summary statistics for posterior distributions of parameters from the chosen submodel.

 traceplot,ubmsFit-method

Markov Chain Traceplots

Description

Draws traceplots for chains from a ubmsFit object

Usage

```
## S4 method for signature 'ubmsFit'
traceplot(object, ...)
```

Arguments

object	A ubmsFit object
...	Arguments passed to <code>rstan::traceplot</code>

Value

A ggplot object.

turnover	<i>Turnover Probability</i>
----------	-----------------------------

Description

Generate posterior draws of turnover probability from dynamic occupancy models. Turnover is calculated for each site and each primary period after the first.

Usage

```
turnover(object, ...)

## S4 method for signature 'ubmsFitColect'
turnover(object, draws, re.form = NULL, ...)
```

Arguments

object	A fitted dynamic occupancy model of class inheriting <code>ubmsFit</code>
...	Currently ignored
draws	Number of draws from the posterior to use in the check
re.form	If NULL, any estimated group-level parameters ("random effects") are included. If NA, they are ignored

Value

A matrix of turnover values from the posterior predictive distribution. The dimensions are draws by number of sites x (primary periods - 1) in site-major order.

See Also

[stan_colect](#)

ubms	<i>ubms</i>
------	-------------

Description

Unmarked Bayesian Models using Stan

Author(s)

Ken Kellner

 ubmsFitList-extractors

Extractors for ubmsFitList objects Extract parts of ubmsFitList objects.

Description

Extractors for ubmsFitList objects Extract parts of ubmsFitList objects.

Usage

```
## S4 method for signature 'ubmsFitList'
x$name

## S4 method for signature 'ubmsFitList,numeric,missing'
x[[i]]

## S4 method for signature 'ubmsFitList,numeric,missing,missing'
x[i]
```

Arguments

x A list of ubmsFit models of class ubmsFitList
 name, i The names or indices of ubmsFit models in the ubmsFitList

Value

A ubmsFit object or list of such objects.

 waic,ubmsFit-method *Widely Applicable Information Criterion (WAIC)*

Description

Widely Applicable Information Criterion (WAIC)

Usage

```
## S4 method for signature 'ubmsFit'
waic(x, ...)
```

Arguments

x A ubmsFit model
 ... Currently ignored

Value

An object of class `waic` containing an estimate of WAIC and other parameters useful for model comparison. See `?loo::waic` for more information.

[,ubmsFit,character,missing,missing-method

Extract a Submodel from a ubmsFit Model

Description

Extract a Submodel from a `ubmsFit` Model

Usage

```
## S4 method for signature 'ubmsFit,character,missing,missing'
x[i]
```

Arguments

<code>x</code>	A <code>ubmsFit</code> model
<code>i</code>	The name of a submodel to extract

Value

An object of class `ubmsSubmodel`.

[,ubmsSubmodellList,character,missing,missing-method

Extract a ubmsSubmodel From a ubmsSubmodellList Object

Description

Extract a `ubmsSubmodel` From a `ubmsSubmodellList` Object

Usage

```
## S4 method for signature 'ubmsSubmodellList,character,missing,missing'
x[i]
```

Arguments

<code>x</code>	Object of class <code>ubmsSubmodellList</code>
<code>i</code>	The name of a submodel

Value

An object of class `ubmsSubmodel`.

Index

[,ubmsFit,character,missing,missing-method, 28
[,ubmsFitList,numeric,missing,missing-method (ubmsFitList-extractors), 27
[,ubmsSubmodelList,character,missing,missing-method, 28
[[,ubmsFitList,numeric,missing-method (ubmsFitList-extractors), 27
\$,ubmsFitList-method (ubmsFitList-extractors), 27

coef,ubmsFit-method, 3
colext, 17

distsamp, 18

extract,ubmsFit-method, 3

fitList (fitList,ubmsFit-method), 4
fitList,ubmsFit-method, 4
fitted,ubmsFit-method, 4

getY,ubmsFit-method, 5
gof, 5
gof,ubmsFitAbun-method (gof), 5
gof,ubmsFitOccu-method (gof), 5

loo, 7
loo,ubmsFit-method, 6
loo_compare, 7

modSel,ubmsFitList-method, 7
multinomPois, 19

names,ubmsFit-method, 7
names,ubmsFitList-method, 8
nsamples,ubmsFit-method, 8

occu, 20
occuRN, 21
occuTTD, 23

pcount, 24
plot,ubmsFit,ANY-method, 9
plot,ubmsGOF,ANY-method (gof), 5
plot_marginal (plot_marginal,ubmsFit-method), 9
plot_marginal,ubmsFit-method, 9
plot_residuals (plot_residuals,ubmsFit-method), 10
plot_residuals,ubmsFit-method, 10
posterior_linpred (posterior_linpred,ubmsFit-method), 11
posterior_linpred,ubmsFit-method, 11
posterior_predict, 15
posterior_predict (posterior_predict,ubmsFit-method), 12
posterior_predict,ubmsFit-method, 12
predict (predict,ubmsFit-method), 13
predict,ubmsFit-method, 13
projected, 14
projected,ubmsFitColext-method (projected), 14

ranef, 15
ranef (ranef,ubmsFit-method), 15
ranef,ubmsFit-method, 15
residuals, 11
residuals,ubmsFit-method, 15

stan, 16, 18–22, 24
stan_colext, 14, 16, 26
stan_distsamp, 17
stan_multinomPois, 19
stan_occu, 20
stan_occuRN, 21
stan_occuTTD, 22
stan_pcount, 23

summary, ubmsFit-method, [25](#)

traceplot, ubmsFit-method, [25](#)

turnover, [26](#)

turnover, ubmsFitColect-method
(turnover), [26](#)

ubms, [26](#)

ubmsFitList-extractors, [27](#)

unmarkedFrameDS, [17](#), [18](#)

unmarkedFrameMPois, [19](#)

unmarkedFrameOccu, [20](#), [21](#)

unmarkedFrameOccuTTD, [22](#), [23](#)

unmarkedFramePCount, [24](#)

unmarkedMultFrame, [16](#), [17](#)

waic, ubmsFit-method, [27](#)