Package: ubms (via r-universe)

October 9, 2026

Version 1.2.9

Date 2026-04-22

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

Depends R (>= 3.4.0), unmarked

Imports ggplot2 (>= 2.0.0), gridExtra, loo, Matrix (>= 1.5-0), methods, pbapply, Rcpp (>= 0.12.0), reformulas, rlang, RSpectra, rstan (>= 2.26.0), rstantools (>= 2.0.0), stats

Suggests knitr, raster, rmarkdown, terra, 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://ecoverseR.github.io/ubms/

BugReports https://github.com/ecoverseR/ubms/issues

Encoding UTF-8

RoxygenNote 7.3.2

Biarch true

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppArmadillo (>= 0.9.300.2.0), RcppEigen (>= 0.3.3.3.0), rstan (>= 2.26.0), StanHeaders (>= 2.26.0), RcppParallel (>= 5.0.2)

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' 'colext.R' 'missing.R' 'distsamp.R' 'fitlist.R' 'kfold.R' 'occuTTD.R' 'multinomPois.R' 'occuRN.R' 'pcount.R' 'loglik.R' 'mb_chisq.R' 'plot_effects.R' 'plot_posteriors.R' 'predict.R' 'priors.R' 'ranef.R' 'residuals.R' 'spatial.R' 'stanmodels.R' 'test-helpers.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>

Config/pak/sysreqs make

Repository https://cran.r-universe.dev

Date/Publication 2026-04-22 19:45:02 UTC

RemoteUrl https://github.com/cran/ubms

RemoteRef HEAD

RemoteSha b019fd349a70d45407936d226e8ee26bcbafc70a

Contents

[,ubmsFit,character,missing,missing-method . . . 3
[,ubmsSubmodelList,character,missing,missing-method . . . 4
coef,ubmsFit-method . . . 4
extract,ubmsFit-method . . . 5
extract_log_lik . . . 5
fitList,ubmsFit-method . . . 6
fitted,ubmsFit-method . . . 6
get_elapsed_time,ubmsFit-method . . . 7
get_stancode,ubmsFit-method . . . 8
getY,ubmsFit-method . . . 8
gof . . . 9
kfold,ubmsFit-method . . . 10
loo,ubmsFit-method . . . 10
modSel,ubmsFitList-method . . . 11
names,ubmsFit-method . . . 12
names,ubmsFitList-method . . . 12
nsamples,ubmsFit-method . . . 13
plot,ubmsFit,ANY-method . . . 13
plot_effects,ubmsFit-method . . . 14
plot_posteriors,ubmsFit-method . . . 15
plot_residuals,ubmsFit-method . . . 16
plot_spatial . . . 17
posterior_linpred,ubmsFit-method . . . 17
posterior_predict,ubmsFit-method . . . 18
predict,ubmsFit-method . . . 19

priors . . . 20
projected . . . 21
ranef,ubmsFit-method . . . 22
residuals,ubmsFit-method . . . 23
RSR . . . 24
stan_colext . . . 25
stan_distsamp . . . 27
stan_multinomPois . . . 29
stan_occu . . . 30
stan_occuRN . . . 31
stan_occuTTD . . . 33
stan_pcount . . . 35
summary,ubmsFit-method . . . 37
traceplot,ubmsFit-method . . . 37
turnover . . . 38
ubms . . . 38
ubmsFitList-extractors . . . 39
waic,ubmsFit-method . . . 39

Index **41**

`[,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

`x`	A `ubmsFit` model
`i`	The name of a submodel to extract

Value

An object of class `ubmsSubmodel`.

[,ubmsSubmodelList,character,missing,missing-method *Extract a ubmsSubmodel From a ubmsSubmodelList Object*

Description

Extract a ubmsSubmodel From a ubmsSubmodelList Object

Usage

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

Arguments

x	Object of class ubmsSubmodelList
i	The name of a submodel

Value

An object of class ubmsSubmodel.

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.

extract_log_lik *Extract Pointwise Log-likelihood From Model*

Description

Extracts the pointwise log-likelihood matrix or array from a model. This is useful as an input for functions in the loo package. If called on a ubmsFit object, the log-likelihood matrix or array is calculated using the posterior distributions of the parameters; the log-likelihood values are not actually saved inside the model object. If called on a stanfit object, loo::extract_log_lik is used. In this case, the log-likelihood must be saved as one of the output parameters in Stan.

Usage

```
extract_log_lik(object, parameter_name = "log_lik", merge_chains = TRUE)
```

Arguments

object	A ubmsFit or stanfit object
parameter_name	The name of the log-likelihood parameter in the Stan output; ignored when object is a ubmsFit
merge_chains	If TRUE (the default), all Markov chains are merged together (i.e., stacked) and a matrix is returned. If 'FALSE' they are kept separate and an array is returned.

Value

A matrix (samples x sites) or array (samples x chains x sites)

---

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, or a list of such models

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.

---

get_elapsed_time,ubmsFit-method
Get Model Runtime

Description

Get warmup and sampling time from a ubmsFit object

Usage

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

Arguments

object	A ubmsFit object
...	Arguments passed to rstan::get_elapsed_time

Value

A matrix with one row per chain and two columns, containing the warmup time and sampling time, respectively, in seconds

get_stancode,ubmsFit-method *Get Stan Code From Model*

Description

Get the Stan code used to run a model as a character string

Usage

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

Arguments

object	A ubmsFit object
...	Arguments passed to rstan::get_stancode

Details

Pass the result of get_stancode to cat to get the code in a more readable format. Note that the output in most cases is Stan code that can be used to fit several types of models, and not all Stan code will be used in all models.

Value

A character string with the model code

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): Applies the MacKenzie-Bailey chi-square goodness of fit test for occupancy models (MacKenzie and Bailey 2004).
- gof(ubmsFitAbun): 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.

kfold,ubmsFit-method *K-fold Cross-validation of a ubmsFit Model*

Description

Randomly partition data into K subsets of equal size (by site). Re-fit the model K times, each time leaving out one of the subsets. Calculate the log-likelihood for each of the sites that was left out. This function is an alternative to loo (leave-one-out cross validation).

Usage

```
## S4 method for signature 'ubmsFit'
kfold(x, K = 10, folds = NULL, quiet = FALSE, ...)
```

Arguments

x	A ubmsFit model
K	Number of folds into which the data will be partitioned
folds	An optional vector with length equal to the number of sites in the data and containing integers from 1 to K, to manually assign sites to folds. You should use this if you plan to compare multiple models, since the folds for each model should be identical. You can use loo::kfold_split_random to generate this vector
quiet	If TRUE, suppress progress bar
...	Currently ignored

Value

An object of class elpd_generic that is compatible with loo::loo_compare

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_effects,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 medians or reference levels. Random effects are ignored.

Usage

```
## S4 method for signature 'ubmsFit'
plot_effects(
  object,
  submodel,
  covariate = NULL,
  level = 0.95,
  draws = 1000,
  smooth = NULL,
  ...
)

## S4 method for signature 'ubmsFit'
plot_marginal(
  object,
  submodel,
  covariate = NULL,
  level = 0.95,
  draws = 1000,
  smooth = NULL,
  ...
)
```

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
`draws`	Number of draws from the posterior to use when generating the plots. If fewer than `draws` are available, they are all used
`smooth`	Smoother span (f) value used for LOWESS smoothing of the upper and lower credible interval bounds for a continuous covariate. No smoothing is done if

NULL. A reasonable value to try is 0.05. The larger the value, the smoother the curve. As with all smoothing, use with caution

... Currently ignored

Value

A ggplot if a single covariate is plotted, or an object of class grob if there are multiple covariates/panels

---

plot_posteriors,ubmsFit-method

Plot Posterior Distributions

Description

Plot posterior distributions for selected parameters. Posteriors can be represented as density plots or histograms.

Usage

```
## S4 method for signature 'ubmsFit'
plot_posteriors(object, pars = NULL, density = FALSE, ...)
```

Arguments

object	A fitted model of class ubmsFit
pars	A character vector of parameter names to include in the plot Look at names(object@stanfit) for the complete list of possible parameter names. If NULL, posteriors are shown for all parameters in the model summary output
density	If TRUE, show posteriors as density plots (one per chain). If FALSE, show posteriors as histograms of samples from all chains combined
...	Arguments passed to ggplot2::stat_density for density plots, or ggplot2::geom_histogram for histograms. For example, you can supply argument bins to control the number of histogram bins

Value

A ggplot

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

object	A fitted model of class ubmsFit
submodel	Submodel to plot residuals for, for example "det"
covariate	If specified, plot residuals against values of a covariate. Covariate name should be provided as a string. If NULL, residuals are plotted against predicted values.
draws	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.
nbins	For submodels with a binomial response, manually set the number of bins to use
...	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

plot_spatial *Plot A Map of the State Parameter Based on a Spatial ubms Model*

Description

Plot A Map of the State Parameter Based on a Spatial ubms Model

Usage

```
plot_spatial(object, param = c("state", "eta"), sites = TRUE, cell_size = NULL)
```

Arguments

object	A ubmsFit model with a spatial random effect
param	The parameter to plot, either "state" for, e.g., mean occupancy or abundance, or "eta" for the random effect itself
sites	If TRUE, also plot the locations of sites that were sampled on the map and if had a detection of the species
cell_size	The size of each side of the (square) cells drawn in the map, in the same units as the coordinates. If NULL, ubms will try to pick a reasonable cell size for you

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

object	A fitted model of class ubmsFit
transform	Should the linear predictor be transformed using the inverse link function?
submodel	The name of the submodel, as a character string, for which to calculate the linear predictor
newdata	Optional data frame of newdata to use when calculating the linear predictor. If not provided, the model matrix is used.
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 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, SpatRaster, 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 SpatRaster/RasterStack, returns a SpatRaster/RasterStack with four layers corresponding to the four columns above with the same projection as the original SpatRaster/RasterStack.

See Also

posterior_linpred, posterior_interval

---

priors *Prior distributions*

Description

Specify prior distributions and associated parameters for use in ubms models.

Usage

```
normal(location = 0, scale = 2.5, autoscale = TRUE)

uniform(lower = -5, upper = 5)

student_t(df = 1, location = 0, scale = 2.5, autoscale = TRUE)

logistic(location = 0, scale = 1)

cauchy(location = 0, scale = 2.5, autoscale = TRUE)

gamma(shape = 1, rate = 1)

laplace(location = 0, scale = 2.5, autoscale = TRUE)
```

Arguments

location	The mean of the distribution. If setting the priors for regression coefficients, this can be a single value, or multiple values, one per coefficient
scale	The standard deviation of the distribution. If setting the priors for regression coefficients, this can be a single value, or multiple values, one per coefficient
autoscale	If TRUE, ubms will automatically adjust priors for each regression coefficient relative to its corresponding covariate x. Specifically, the prior for a given coefficient will be divided by sd(x). This helps account for covariates with very different magnitudes in the same model. If your data are already standardized

	(e.g. with use of `scale()`), this will have minimal effect as sd(x) will be approximately 1. Standardizing your covariates is highly recommended.
`lower`	The lower bound for the uniform distribution
`upper`	The upper bound for the uniform distribution
`df`	The number of degrees of freedom for the Student-t distribution
`shape`	The gamma distribution shape parameter
`rate`	The gamma distribution rate parameter (1/scale)

Value

A `list` containing prior settings used internally by `ubms`.

Examples

```
normal()
```

`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 'ubmsFitColext'
projected(object, draws = NULL, re.form = NULL, ...)
```

Arguments

`object`	A fitted dynamic occupancy model of class inheriting `ubmsFit`
`...`	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_colext

---

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, add_mean = TRUE, ...)
```

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
add_mean	If TRUE (the default) add the overall intercept or slope mean and return the complete random intercept or slope.
...	Currently ignored

Details

Note: by default this function adds the overall intercept or slope to the (mean-0) random effect to get the complete random intercept or slope. In this way the output is more like the output of lme4::coef and not lme4::ranef. You can turn this off and return just the mean-0 random effect by setting argument add_mean = FALSE.

If you run ranef on a submodel with a spatial random effect, the function will return estimates of parameter eta.

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 `ubmsFit`
`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.

RSR *Get Information for a Restricted Spatial Regression Model*

Description

A call to RSR in the formula for a state process adds an autocorrelated spatial random effect to the model in the form of a Restricted Spatial Regression (RSR). For examples of RSRs applied to ecological data, see Johnson et al. (2013) and Broms et al. (2014). The function can also be used outside a formula to visualize the spatial relationships between sites in your data and choose an appropriate distance threshold below which two sites will be considered neighbors, and thus potentially correlated, for the RSR model. For more details, see the example vignette: `vignette("spatial-models", package = "ubms")`

Usage

```
RSR(x, y = NULL, threshold, moran_cut = NULL, plot_site = NULL)
```

Arguments

x	A vector of coordinates (should be projected)
y	An (optional) second vector of coordinates
threshold	The distance cutoff below which two sites will be considered neighbors. Should be the same units as the coordinates.
moran_cut	The number of eigenvectors to use in the RSR. The possible range of values is between 1 and the number of sites. Smaller numbers will result in faster runtime and smoother map output, and vice-versa. If not provided (the default), the number of eigenvectors will be set as 10% of the number of sites which is usually appropriate.
plot_site	If a site number (as an integer) is supplied, the function returns a plot showing that site and its neighbors under the current settings. Useful for deciding what to set your threshold at.

Value

Either a list of spatial matrices used for the RSR (only useful internally to ubms), or if `plot_site` is an integer, a `ggplot` object.

References

Broms KM, Johnson DS, Altwegg R, Conquest LL. 2014. Spatial occupancy models applied to atlas data show Southern Ground Hornbills strongly depend on protected areas. Ecological Applications 24: 363-374.

Johnson DS, Conn PB, Hooten MB, Ray JC, Pond BA. 2013. Spatial occupancy models for large data sets. Ecology 94: 801-808.

Examples

```
# Generate some coordinates
x <- runif(100,0,10)
y <- runif(100,0,10)
# Show neighbors of site 10 when threshold is 3 units
RSR(x, y, threshold=3, plot_site=10)
```

`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,
  prior_intercept_psi = logistic(0, 1),
  prior_coef_psi = logistic(0, 1),
  prior_intercept_gamma = logistic(0, 1),
  prior_coef_gamma = logistic(0, 1),
  prior_intercept_eps = logistic(0, 1),
  prior_coef_eps = logistic(0, 1),
  prior_intercept_det = logistic(0, 1),
  prior_coef_det = logistic(0, 1),
  prior_sigma = gamma(1, 1),
  log_lik = TRUE,
  ...
)
```

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 `unmarkedMultFrame` object
`prior_intercept_psi`	Prior distribution for the intercept of the psi (initial occupancy probability) model; see `?priors` for options

`prior_coef_psi` Prior distribution for the regression coefficients of the psi model

`prior_intercept_gamma`
Prior distribution on intercept for colonization probability

`prior_coef_gamma`
Prior distribution on regression coefficients for colonization probability

`prior_intercept_eps`
Prior distribution on intercept for extinction probability

`prior_coef_eps` Prior distribution on regression coefficients for extinction probability

`prior_intercept_det`
Prior distribution for the intercept of the detection probability model

`prior_coef_det` Prior distribution for the regression coefficients of the detection model

`prior_sigma` Prior distribution on random effect standard deviations

`log_lik` If `TRUE`, Stan will save pointwise log-likelihood values in the output. This can greatly increase the size of the model. If `FALSE`, the values are calculated post-hoc from the posteriors

`...` Arguments passed to the `stan` call, such as number of chains `chains` or iterations `iter`

Value

`ubmsFitColext` object describing the model fit.

References

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

See Also

`colext`, `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_colext(~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 discerete distance intervals.

Usage

```
stan_distsamp(
  formula,
  data,
  keyfun = c("halfnorm", "exp", "hazard"),
  output = c("density", "abund"),
  unitsOut = c("ha", "kmsq"),
  prior_intercept_state = normal(0, 5),
  prior_coef_state = normal(0, 2.5),
  prior_intercept_det = normal(0, 5),
  prior_coef_det = normal(0, 2.5),
  prior_intercept_scale = normal(0, 2.5),
  prior_sigma = gamma(1, 1),
  ...
)
```

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
prior_intercept_state	Prior distribution for the intercept of the state (abundance) model; see ?priors for options
prior_coef_state	Prior distribution for the regression coefficients of the state model
prior_intercept_det	Prior distribution for the intercept of the detection probability model
prior_coef_det	Prior distribution for the regression coefficients of the detection model

`prior_intercept_scale`
: Prior distribution for the intercept of the scale parameter (i.e., log(scale)) for Hazard-rate models

`prior_sigma`
: Prior distribution on random effect standard deviations

`...`
: Arguments passed to the `stan` call, such as number of chains `chains` or iterations `iter`

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,
  prior_intercept_state = normal(0, 5),
  prior_coef_state = normal(0, 2.5),
  prior_intercept_det = logistic(0, 1),
  prior_coef_det = logistic(0, 1),
  prior_sigma = gamma(1, 1),
  log_lik = TRUE,
  ...
)
```

Arguments

formula	Double right-hand side formula describing covariates of detection and abundance in that order
data	A unmarkedFrameMPois object
prior_intercept_state	Prior distribution for the intercept of the state (abundance) model; see ?priors for options
prior_coef_state	Prior distribution for the regression coefficients of the state model
prior_intercept_det	Prior distribution for the intercept of the detection probability model
prior_coef_det	Prior distribution for the regression coefficients of the detection model
prior_sigma	Prior distribution on random effect standard deviations
log_lik	If TRUE, Stan will save pointwise log-likelihood values in the output. This can greatly increase the size of the model. If FALSE, the values are calculated post-hoc from the posteriors
...	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,
  prior_intercept_state = logistic(0, 1),
  prior_coef_state = logistic(0, 1),
  prior_intercept_det = logistic(0, 1),
  prior_coef_det = logistic(0, 1),
  prior_sigma = gamma(1, 1),
  log_lik = TRUE,
  ...
)
```

Arguments

formula	Double right-hand side formula describing covariates of detection and occupancy in that order
data	A unmarkedFrameOccu object
prior_intercept_state	Prior distribution for the intercept of the state (occupancy probability) model; see ?priors for options
prior_coef_state	Prior distribution for the regression coefficients of the state model
prior_intercept_det	Prior distribution for the intercept of the detection probability model

prior_coef_det	Prior distribution for the regression coefficients of the detection model
prior_sigma	Prior distribution on random effect standard deviations
log_lik	If TRUE, Stan will save pointwise log-likelihood values in the output. This can greatly increase the size of the model. If FALSE, the values are calculated post-hoc from the posteriors
...	Arguments passed to the stan call, such as number of chains chains or iterations iter

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,
  prior_intercept_state = normal(0, 5),
  prior_coef_state = normal(0, 2.5),
  prior_intercept_det = logistic(0, 1),
  prior_coef_det = logistic(0, 1),
  prior_sigma = gamma(1, 1),
  log_lik = TRUE,
  ...
)
```

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.
prior_intercept_state	Prior distribution for the intercept of the state (abundance) model; see ?priors for options
prior_coef_state	Prior distribution for the regression coefficients of the state model
prior_intercept_det	Prior distribution for the intercept of the detection probability model
prior_coef_det	Prior distribution for the regression coefficients of the detection model
prior_sigma	Prior distribution on random effect standard deviations
log_lik	If TRUE, Stan will save pointwise log-likelihood values in the output. This can greatly increase the size of the model. If FALSE, the values are calculated post-hoc from the posteriors
...	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"),
  prior_intercept_state = logistic(0, 1),
  prior_coef_state = logistic(0, 1),
  prior_intercept_det = normal(0, 5),
  prior_coef_det = normal(0, 2.5),
  prior_intercept_shape = normal(0, 2.5),
  prior_sigma = gamma(1, 1),
  log_lik = TRUE,
  ...
)
```

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.
prior_intercept_state	Prior distribution for the intercept of the state (occupancy probability) model; see ?priors for options
prior_coef_state	Prior distribution for the regression coefficients of the state model
prior_intercept_det	Prior distribution for the intercept of the time-to-detection model
prior_coef_det	Prior distribution for the regression coefficients of the time-to-detection model
prior_intercept_shape	Prior distribution for the intercept of the shape parameter (i.e., log(shape)) for Weibull TTD models
prior_sigma	Prior distribution on random effect standard deviations
log_lik	If TRUE, Stan will save pointwise log-likelihood values in the output. This can greatly increase the size of the model. If FALSE, the values are calculated post-hoc from the posteriors
...	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",
  prior_intercept_state = normal(0, 5),
  prior_coef_state = normal(0, 2.5),
  prior_intercept_det = logistic(0, 1),
  prior_coef_det = logistic(0, 1),
  prior_sigma = gamma(1, 1),
  log_lik = TRUE,
  ...
)
```

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.
prior_intercept_state	Prior distribution for the intercept of the state (abundance) model; see ?priors for options
prior_coef_state	Prior distribution for the regression coefficients of the state model
prior_intercept_det	Prior distribution for the intercept of the detection probability model
prior_coef_det	Prior distribution for the regression coefficients of the detection model
prior_sigma	Prior distribution on random effect standard deviations
log_lik	If TRUE, Stan will save pointwise log-likelihood values in the output. This can greatly increase the size of the model. If FALSE, the values are calculated post-hoc from the posteriors
...	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, probs = c(0.025, 0.25, 0.5, 0.75, 0.975), ...)
```

Arguments

object	A ubmsFit model
submodel	Name of submodel to summarize
probs	Numeric vector of quantiles of interest
...	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 rstan::traceplot

Value

A ggplot object.

turnover *Turnover Probability*

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 'ubmsFitColext'
turnover(object, draws, re.form = NULL, ...)
```

Arguments

object	A fitted dynamic occupancy model of class inheriting ubmsFit
...	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_colext

ubms *ubms*

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.

Index

[,ubmsFit,character,missing,missing-method, 3
[,ubmsFitList,numeric,missing,missing-method (ubmsFitList-extractors), 39
[,ubmsSubmodelList,character,missing,missing-method, 4
[[,ubmsFitList,numeric,missing-method (ubmsFitList-extractors), 39
$,ubmsFitList-method (ubmsFitList-extractors), 39

cauchy (priors), 20
coef,ubmsFit-method, 4
colext, *26*

distsamp, *28*

extract,ubmsFit-method, 5
extract_log_lik, 5
extract_log_lik,ubmsFit-method (extract_log_lik), 5
extract_log_lik,ubmsFitDistsamp-method (extract_log_lik), 5

fitList (fitList,ubmsFit-method), 6
fitList,list-method (fitList,ubmsFit-method), 6
fitList,ubmsFit-method, 6
fitted,ubmsFit-method, 6

gamma (priors), 20
get_elapsed_time,ubmsFit-method, 7
get_stancode,ubmsFit-method, 8
getY,ubmsFit-method, 8
gof, 9
gof,ubmsFitAbun-method (gof), 9
gof,ubmsFitOccu-method (gof), 9

kfold,ubmsFit-method, 10

laplace (priors), 20
logistic (priors), 20
loo, *11*
loo,ubmsFit-method, 10
loo_compare, *11*

modSel,ubmsFitList-method, 11
multinomPois, *30*

names,ubmsFit-method, 12
names,ubmsFitList-method, 12
normal (priors), 20
nsamples,ubmsFit-method, 13

occu, *31*
occuRN, *33*
occuTTD, *34*

pcount, *36*
plot,ubmsFit,ANY-method, 13
plot,ubmsGOF,ANY-method (gof), 9
plot_effects (plot_effects,ubmsFit-method), 14
plot_effects,ubmsFit-method, 14
plot_marginal (plot_effects,ubmsFit-method), 14
plot_marginal,ubmsFit-method (plot_effects,ubmsFit-method), 14
plot_posteriors (plot_posteriors,ubmsFit-method), 15
plot_posteriors,ubmsFit-method, 15
plot_residuals (plot_residuals,ubmsFit-method), 16
plot_residuals,ubmsFit-method, 16
plot_spatial, 17

posterior_linpred
(posterior_linpred,ubmsFit-method), 17
posterior_linpred,ubmsFit-method, 17
posterior_predict, *23*
posterior_predict
(posterior_predict,ubmsFit-method), 18
posterior_predict,ubmsFit-method, 18
predict (predict,ubmsFit-method), 19
predict,ubmsFit-method, 19
priors, 20
projected, 21
projected,ubmsFitColext-method
(projected), 21

ranef, *23*
ranef (ranef,ubmsFit-method), 22
ranef,ubmsFit-method, 22
residuals, *16*
residuals,ubmsFit-method, 23
RSR, 24

stan, *26*, *28, 29*, *31, 32*, *34*, *36*
stan_colext, *22*, 25, *38*
stan_distsamp, 27
stan_multinomPois, 29
stan_occu, 30
stan_occuRN, 31
stan_occuTTD, 33
stan_pcount, 35
student_t (priors), 20
summary,ubmsFit-method, 37

traceplot,ubmsFit-method, 37
turnover, 38
turnover,ubmsFitColext-method
(turnover), 38

ubms, 38
ubms-package (ubms), 38
ubmsFitList-extractors, 39
uniform (priors), 20
unmarkedFrameDS, *27, 28*
unmarkedFrameMPois, *29, 30*
unmarkedFrameOccu, *30–33*
unmarkedFrameOccuTTD, *34*
unmarkedFramePCount, *36*
unmarkedMultFrame, *25, 26*

waic,ubmsFit-method, 39