

Package: bycatch (via r-universe)

October 9, 2024

Type Package

Title Fit Bayesian Models to Bycatch Data

Version 1.0.7

Maintainer Eric Ward <eric.ward@noaa.gov>

Description Fits models to bycatch data, and can expand to fleet-wide estimates, like in situations when fisheries are observed with less than 100% observer coverage.

License GPL-3

Encoding UTF-8

Depends R (>= 4.3.0)

Imports dplyr (>= 1.0.2), ggplot2, loo (>= 2.4.1), methods, Rcpp (>= 0.12.0), RcppParallel (>= 5.0.1), rlang (>= 0.4.1), rstan (>= 2.18.1), rstantools (>= 2.4.0)

LinkingTo BH (>= 1.66.0), Rcpp (>= 0.12.0), RcppEigen (>= 0.3.3.3.0), RcppParallel (>= 5.0.1), rstan (>= 2.18.1), StanHeaders (>= 2.18.0)

Suggests testthat (>= 3.0.0), parallel, knitr, rmarkdown, devtools, MASS

URL <https://ericward-noaa.github.io/bycatch/>

BugReports <https://github.com/ericward-noaa/bycatch/issues>

RoxygenNote 7.3.1

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

SystemRequirements GNU make

Biarch true

Config/testthat/edition 3

Repository <https://noaa-fisheries-integrated-toolbox.r-universe.dev>

RemoteUrl <https://github.com/ericward-noaa/bycatch>

RemoteRef HEAD

RemoteSha c1e2c26af7b3e6502f667e702b5a1f631af43044

Contents

bycatch-package	2
fit_bycatch	2
get_expanded	5
get_fitted	5
get_total	6
plot_expanded	7
plot_fitted	8

Index	10
--------------	-----------

bycatch-package	<i>The 'bycatch' package.</i>
-----------------	-------------------------------

Description

A DESCRIPTION OF THE PACKAGE

References

Stan Development Team (2023). RStan: the R interface to Stan. R package version 2.21.8.
<https://mc-stan.org>

fit_bycatch	<i>fit_bycatch is the primary function for fitting bycatch models to time series of takes and effort</i>
-------------	--

Description

fit_bycatch is the primary function for fitting bycatch models to time series of takes and effort

Usage

```
fit_bycatch(
  formula,
  data,
  time = "year",
  effort = "effort",
  expansion_rate = NULL,
  family = c("poisson", "nbinom2", "poisson-hurdle", "nbinom2-hurdle", "lognormal",
    "gamma", "lognormal-hurdle", "gamma-hurdle", "normal", "normal-hurdle"),
  time_varying = FALSE,
  iter = 1000,
  chains = 3,
  control = list(adapt_delta = 0.9, max_treedepth = 20),
  ...
)
```

Arguments

formula	The model formula.
data	A data frame.
time	Named column of the 'data' data frame with the label for the time (e.g. year) variable
effort	Named column of the 'effort' variable in the data frame with the label for the fishing effort to be used in estimation of mean bycatch rate. This represents total observed effort
expansion_rate	The expansion rate to be used in generating distributions for unobserved sets. If NULL, defaults to 100% coverage (= 100)
family	Family for response distribution can be discrete ("poisson", "nbinom2", "poisson-hurdle", "nbinom2-hurdle"), or continuous ("normal", "gamma", "lognormal", "normal-hurdle", "gamma-hurdle", "lognormal-hurdle"). The default distribution is "poisson". The hurdle variants estimate the probability of zeros (theta) separately from the other models and use truncated distribution to model positive counts. All use a log link function.
time_varying	boolean TRUE/FALSE, whether to include time varying component (this is a random walk, analogous to making this a Dynamic linear model)
iter	the number of mcmc iterations, defaults to 1000
chains	the number of mcmc chains, defaults to 3
control	List to pass to <code>rstan::sampling()</code> . For example, increase adapt_delta if there are warnings about divergent transitions: <code>control = list(adapt_delta = 0.99)</code> . By default, bycatch sets <code>adapt_delta = 0.9</code> .
...	Any other arguments to pass to <code>rstan::sampling()</code> .

Value

list of the data used to fit the model, the matrix of covariates, the expanded bycatch generated via the fit and simulations, and the fitted stan model

Examples

```
d <- data.frame(
  "Year" = 2002:2014,
  "Takes" = c(0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0),
  "expansionRate" = c(24, 22, 14, 32, 28, 25, 30, 7, 26, 21, 22, 23, 27),
  "Sets" = c(391, 340, 330, 660, 470, 500, 330, 287, 756, 673, 532, 351, 486)
)
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "poisson", time_varying = FALSE
)
loo::loo(fit$fitted_model)$estimates

fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year", effort = "Sets",
  family = "poisson", time_varying = FALSE
```

```

)
plot_fitted(fit,
  xlab = "Year", ylab = "Fleet-level bycatch",
  include_points = TRUE
)

# fit a negative binomial model, with more chains and control arguments
fit_nb <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "nbinom2",
  time_varying = FALSE, iter = 2000, chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)

# fit a time varying model
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "poisson", time_varying = TRUE
)

# include data for expansion to unobserved sets
fit_nb <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "nbinom2",
  expansion_rate = "expansionRate",
  time_varying = FALSE, iter = 2000, chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)

# fit a model with a lognormal distribution
d$Takes <- rnorm(nrow(d), 5, 0.1)
fit_ln <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "lognormal",
  expansion_rate = "expansionRate",
  time_varying = FALSE, iter = 2000, chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)

# add zeros and fit a delta-gamma distribution
d$Takes <- rnorm(nrow(d), 5, 0.1)
d$Takes[c(1, 5, 10)] <- 0
fit_ln <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "gamma-hurdle",
  expansion_rate = "expansionRate",
  time_varying = FALSE, iter = 2000, chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)

```

get_expanded	<i>get_expanded is a helper function to return a matrix of posterior predictive values for unobserved bycatch</i>
--------------	---

Description

get_expanded is a helper function to return a matrix of posterior predictive values for unobserved bycatch

Usage

```
get_expanded(fitted_model)
```

Arguments

fitted_model Data and fitted model returned from estimation

Value

matrix (MCMC draws x time steps) of posterior predictive values for unobserved bycatch

Examples

```
d <- data.frame(
  "Year" = 2002:2014,
  "Takes" = c(0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0),
  "expansionRate" = c(24, 22, 14, 32, 28, 25, 30, 7, 26, 21, 22, 23, 27),
  "Sets" = c(391, 340, 330, 660, 470, 500, 330, 287, 756, 673, 532, 351, 486)
)
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets",
  family = "poisson",
  expansion_rate = "expansionRate",
  time_varying = FALSE
)
expanded <- get_expanded(fit)
```

get_fitted	<i>get_fitted returns df of observed bycatch estimates (lambda of Poisson), accounting for effort but not accounting for observer coverage</i>
------------	--

Description

get_fitted returns df of observed bycatch estimates (lambda of Poisson), accounting for effort but not accounting for observer coverage

Usage

```
get_fitted(fitted_model, alpha = 0.05)
```

Arguments

fitted_model Data and fitted model returned from `fit_bycatch()`. If a hurdle model, then only then the plot returns the total bycatch rate (including zero and non-zero components).

alpha The alpha level for the credible interval, defaults to 0.05

Value

plot called from `ggplot`

Examples

```
d <- data.frame(
  "Year" = 2002:2014,
  "Takes" = c(0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0),
  "expansionRate" = c(24, 22, 14, 32, 28, 25, 30, 7, 26, 21, 22, 23, 27),
  "Sets" = c(391, 340, 330, 660, 470, 500, 330, 287, 756, 673, 532, 351, 486)
)
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year", effort = "Sets",
  family = "poisson", time_varying = FALSE
)
get_fitted(fit)
```

<code>get_total</code>	<i>get_total is a helper function to return a matrix of total estimated bycatch</i>
------------------------	---

Description

`get_total` is a helper function to return a matrix of total estimated bycatch

Usage

```
get_total(fitted_model)
```

Arguments

fitted_model Data and fitted model returned from estimation

Value

matrix (MCMC draws x time steps) of posterior predictive values for unobserved bycatch

Examples

```
d <- data.frame(
  "Year" = 2002:2014,
  "Takes" = c(0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0),
  "expansionRate" = c(24, 22, 14, 32, 28, 25, 30, 7, 26, 21, 22, 23, 27),
  "Sets" = c(391, 340, 330, 660, 470, 500, 330, 287, 756, 673, 532, 351, 486)
)
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets",
  family = "poisson",
  expansion_rate = "expansionRate",
  time_varying = FALSE
)
expanded <- get_total(fit)
```

plot_expanded	<i>plot_expanded</i> makes plots of the expanded bycatch estimates, accounting for observer coverage and effort
---------------	---

Description

plot_expanded makes plots of the expanded bycatch estimates, accounting for observer coverage and effort

Usage

```
plot_expanded(
  fitted_model,
  xlab = "Time",
  ylab = "Events",
  show_total = TRUE,
  include_points = FALSE
)
```

Arguments

fitted_model	Data and fitted model returned from estimation
xlab	X-axis label for plot
ylab	Y-axis label for plot
show_total	Whether to show the total predicted bycatch (by default, this is TRUE) or just the expanded unobserved events (=FALSE)
include_points	whether or not to include raw bycatch events on plots, defaults to FALSE

Value

plot called from ggplot

Examples

```
d <- data.frame(
  "Year" = 2002:2014,
  "Takes" = c(0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0),
  "expansionRate" = c(24, 22, 14, 32, 28, 25, 30, 7, 26, 21, 22, 23, 27),
  "Sets" = c(391, 340, 330, 660, 470, 500, 330, 287, 756, 673, 532, 351, 486)
)
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets",
  family = "poisson",
  expansion_rate = "expansionRate",
  time_varying = FALSE
)
plot_expanded(
  fitted_model = fit,
  xlab = "Year",
  ylab = "Fleet-level bycatch",
  include_points = TRUE
)
```

plot_fitted

plot_fitted makes plots bycatch estimates (lambda of Poisson), accounting for effort but not accounting for observer coverage

Description

plot_fitted makes plots bycatch estimates (lambda of Poisson), accounting for effort but not accounting for observer coverage

Usage

```
plot_fitted(
  fitted_model,
  xlab = "Time",
  ylab = "Events",
  include_points = FALSE,
  alpha = 0.05
)
```

Arguments

fitted_model	Data and fitted model returned from fit_bycatch(). If a hurdle model, then only then the plot returns the total bycatch rate (including zero and non-zero components).
xlab	X-axis label for plot
ylab	Y-axis label for plot

include_points whether or not to include raw bycatch events on plots, defaults to FALSE
 alpha The alpha level for the credible interval, defaults to 0.05

Value

plot called from ggplot

Examples

```
d <- data.frame(
  "Year" = 2002:2014,
  "Takes" = c(0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 0, 0, 0),
  "expansionRate" = c(24, 22, 14, 32, 28, 25, 30, 7, 26, 21, 22, 23, 27),
  "Sets" = c(391, 340, 330, 660, 470, 500, 330, 287, 756, 673, 532, 351, 486)
)
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year", effort = "Sets",
  family = "poisson", time_varying = FALSE
)
plot_fitted(fit,
  xlab = "Year", ylab = "Fleet-level bycatch",
  include_points = TRUE
)

# fit a negative binomial model, with more chains and control arguments
fit_nb <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "nbinom2",
  time_varying = FALSE, iter = 2000, chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)

# fit a time varying model
fit <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "poisson", time_varying = TRUE
)

# include data for expansion to unobserved sets
fit_nb <- fit_bycatch(Takes ~ 1,
  data = d, time = "Year",
  effort = "Sets", family = "nbinom2",
  expansion_rate = "expansionRate",
  time_varying = FALSE, iter = 2000, chains = 4,
  control = list(adapt_delta = 0.99, max_treedepth = 20)
)
```

Index

`bycatch (bycatch-package)`, [2](#)
`bycatch-package`, [2](#)

`fit_bycatch`, [2](#)

`get_expanded`, [5](#)
`get_fitted`, [5](#)
`get_total`, [6](#)

`plot_expanded`, [7](#)
`plot_fitted`, [8](#)

`rstan::sampling()`, [3](#)