

Package: moosecounter (via r-universe)

September 20, 2024

Version 0.8-0

Date 2024-02-22

Title Adaptive Moose Surveys

Author Subhash Lele [aut], Sophie Czetwertynski [aut], Peter Solymos [aut, cre] (<<https://orcid.org/0000-0001-7337-1740>>), Steffi LaZerte [aut], Government of Yukon [fnd]

Maintainer Peter Solymos <psolymos@gmail.com>

Description Adaptive Moose surveys.

Imports pscl (>= 1.5.5), partykit (>= 1.2.15), MASS (>= 7.3.54), pbapply (>= 1.4.3), shiny (>= 1.6.0), magrittr, VGAM (>= 1.1.5), Matrix (>= 1.3.4), ggplot2 (>= 3.3.5), rlang (>= 1.0.2), dplyr (>= 1.0.7), purrr (>= 0.3.4)

Suggests DT (>= 0.18), ggiraph (>= 0.7.10), kableExtra (>= 1.3.4), mapproj (>= 1.2.7), openxlsx, shinyBS (>= 0.61), shinydashboard (>= 0.7.1), shinyjs (>= 2.1.0), systemfonts (>= 0.3.0), testthat (>= 3.0.0), patchwork (>= 1.2.0)

License GPL-2

RoxygenNote 7.2.3

Encoding UTF-8

Config/testthat/edition 3

Repository <https://psolymos.r-universe.dev>

RemoteUrl <https://github.com/psolymos/moosecounter>

RemoteRef HEAD

RemoteSha b5318e9a7662f69765515d009bb4d2a3ad93eb62

Contents

comp	2
explore	4
gassaway	5
mc_options	6

run_app	7
total	7

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

comp	<i>Moose Composition Workflow</i>
------	-----------------------------------

Description

Fit composition model for Moose using a multinomial model to capture how predictors affect composition data; calculate prediction intervals based on composition model; and extract useful summaries.

Usage

```
mc_check_comp(x)

mc_fit_comp(x, vars = NULL)

mc_models_comp(model_list_comp, coefs = TRUE)

mc_predict_comp(
  total_model_id,
  comp_model_id,
  model_list_total,
  model_list_comp,
  x,
  do_avg = FALSE,
  fix_mean = FALSE,
  PI = NULL
)

subset_CPI_data(CPI, ss)

pred_density_moose_CPI(CPI, ...)
```

Arguments

x	A Moose data frame object.
vars	Column names of 'x' to be used as predictors for the composition model.
model_list_comp	Named list of total composition models.
coefs	logical, return coefficient table too.
total_model_id	Model ID or model IDs for total moose model (can be multiple model IDs from 'names(model_list_total)').
comp_model_id	Model ID or model IDs for composition model (single model ID from 'names(model_list_comp)').

model_list_total	Named list of total moose models.
do_avg	Logical, to do model averaging or not.
fix_mean	logical, use the fixed (rounded) mean as the Multinomial size instead of the bootstrap PI counts.
PI	Total Moose PI object.
CPI	Composition PI object.
ss	A subset of rows (logical or numeric vector).
...	Other arguments passed to underlying functions.

Examples

```
mc_options(B=10)
x <- read.csv(
  system.file("extdata/MayoMMU_QuerriedData.csv",
    package="moosecounter"))
## Prepare Moose data frame object
x <- mc_update_total(x)

## Total moose model list
vars <- c("ELC_Subalpine", "Fire1982_2012", "Fire8212_DEM815",
  "NALC_Needle", "NALC_Shrub", "Subalp_Shrub_250buf",
  "ELCSub_Fire8212DEM815", "SubShrub250_Fire8212DEM815")
ML <- list()
ML[["Model 0"]] <- mc_fit_total(x, dist="ZINB")
ML[["Model 1"]] <- mc_fit_total(x, vars[1:2], dist="ZINB")
ML[["Model 2"]] <- mc_fit_total(x, vars[2:3], dist="ZIP")
ML[["Model 3"]] <- mc_fit_total(x, vars[3:4], dist="ZINB")

## Composition odel list
CML <- list()
CML[["FireDEMSub']] <- mc_fit_comp(x, "Fire8212_DEM815")

## Stats from the models
mc_models_comp(CML)

## Calculate PI
CPI <- mc_predict_comp(
  total_model_id="Model 3",
  comp_model_id='FireDEMSub',
  model_list_total=ML,
  model_list_comp=CML,
  x=x,
  do_avg=FALSE)

## Predict density
pred_density_moose_CPI(CPI)
```

 explore

Exploration

Description

Functions to explore relationships between total Moose or composition as response vs. environmental predictor variables.

Usage

```
mc_plot_univariate(
  i,
  x,
  dist = "ZINB",
  base = TRUE,
  type = c("density", "map", "fit"),
  interactive = FALSE
)

mc_plot_multivariate(vars, x, alpha = NULL)

mc_plot_comp(i, x)
```

Arguments

<code>i</code>	Column name from ‘x’ to be used as a predictor.
<code>x</code>	Data frame with Moose data.
<code>dist</code>	Count distribution (‘P’, ‘NB’, ‘ZIP’, or ‘ZINB’).
<code>base</code>	Logical, draw base graphics or ggplot2.
<code>type</code>	Character, type of plot to be drawn (“density”, “map”, “fit”). Base plot can draw all 3, ggplot2 can only draw one at a time.
<code>interactive</code>	Logical, draw interactive plot (not available for base plots).
<code>vars</code>	A vector of column names from ‘x’ to be used as a predictor.
<code>alpha</code>	Alpha level defining ‘mincriterion = 1 - alpha’ for ‘partykit::ctree()’.

Details

‘mc_plot_univariate’ implements visual univariate (single predictor) exploration for the total Moose count models.

‘mc_plot_multivariate’ implements visual multivariate (multiple predictors) exploration based on regression trees (recursive partitioning in a conditional inference framework) for total Moose counts.

‘mc_plot_comp’ implements visual univariate (single predictor) exploration for the multinomial composition models.

Examples

```
## Prepare Moose data from Mayo
x <- read.csv(
  system.file("extdata/MayoMMU_QuerriedData.csv",
    package="moosecounter"))
switch_response("total")
x <- mc_update_total(x)

## Univariate exploration for total Moose
mc_plot_univariate("Subalp_Shrub_250buf", x, "ZINB")

## Multivariate exploration for total Moose
vars <- c("ELC_Subalpine", "Fire1982_2012", "Fire8212_DEM815",
  "NALC_Needle", "NALC_Shrub", "Subalp_Shrub_250buf",
  "ELCSub_Fire8212DEM815", "SubShrub250_Fire8212DEM815")
mc_plot_multivariate(vars, x)

## Univariate exploration for composition
mc_plot_comp("Fire8212_DEM815", x)
```

gassaway

*Total Estimate with Gassaway Formula***Description**

We do the stratification BEFORE the surveys are done using the fixed wing aircraft. Every cell is stratified as LOW or HIGH. Then some of the LOW and some of HIGH are surveyed to find how many moose are there.

Usage

```
mc_gassaway(y1, y2, N1, N2)

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

Arguments

y1, y2	surveyed cell-level population counts in the 2 strata
N1, N2	total number of cells that were stratified
x	object to print
...	arguments passed to the print method

Examples

```

y1 <- rpois(20, 50)
y2 <- rpois(30, 5)
N1 <- 40
N2 <- 50
mc_gassaway(y1, y2, N1, N2)

```

mc_options

Set Moose Options

Description

A function to get and set the package related options.

Usage

```
mc_options(...)
```

Arguments

... options to set (see Details)

Details

* 'method': Optimization algorithm * 'response': Response variable * 'MAXCELL': Max total abundance in cells * 'MINCELL': Min abundance for composition in cells * 'B': Number of bootstrap runs * 'alpha': Type I error rate for PI * 'wscale': Weight * 'sightability': Sightability * 'seed': Random seed * 'Ntot': set to 'MOOSE_TOTA' (internally changing to "COW_TOTA" when modeling cows) * 'srv_name': Filtering variable * 'srv_value': Filtering value * 'area_srv': Column indicating survey areas * 'Area': Area * 'xy': Long/Lat * 'composition': Composition variables

Examples

```

## original values
o <- mc_options()
str(o)
o$B

## set B to new value
mc_options(B = 20)
mc_options()$B

## restore original values
mc_options(o)
mc_options()$B

```

run_app	<i>Run Shiny App</i>
---------	----------------------

Description

Run the Shiny app to estimate total Moose and composition.

Usage

```
run_app()
```

Examples

```
if (interactive()) {
  run_app()
}
```

total	<i>Total Moose Workflow</i>
-------	-----------------------------

Description

‘switch_response’ switches between total Moose vs. cows only. This sets the column name for total Moose estimation. ‘mc_update_total’ Updates/prepares the Moose data set for downstream analyses (i.e. calculates some derived variables, sets a surveyed/unsurveyed indicator, and optionally takes a subset). ‘mc_fit_total’ fit total Moose abundance models. ‘mc_models_total’ prints out estimates from the models.

Usage

```
switch_response(type = "total")
```

```
mc_update_total(x, srv = NULL, ss = NULL)
```

```
mc_fit_total(
  x,
  vars = NULL,
  zi_vars = NULL,
  dist = "ZINB",
  weighted = FALSE,
  robust = FALSE,
  intercept = c("both", "count", "zero", "none"),
  xv = FALSE,
  ...
)
```

```

mc_models_total(ml, x, coefs = TRUE)

mc_predict_total(model_id, ml, x, do_boot = TRUE, do_avg = FALSE)

mc_get_pred(PI, ss = NULL)

pred_density_moose_PI(PI)

mc_plot_residuals(model_id, ml, x)

mc_plot_predpi(PI)

mc_plot_pidistr(PI, id = NULL, plot = TRUE, breaks = "Sturges")

mc_plot_predfit(i, PI, ss = NULL, interactive = FALSE)

```

Arguments

type	The type of the response, can be "total" or "cows" for 'switch_response'.
x	A data frame with Moose data, or a data frame from 'mc_update_total()'.
srv	Logical vector, rows of 'x' that are surveyed, falls back to global options when 'NULL'.
ss	Logical vector to subset 'x', default is to take no subset.
vars	column names of 'x' to be used as predictors for the count model.
zi_vars	optional, column names of 'x' to be used as predictors for the zero model.
dist	Count distribution ('P', 'NB', 'ZIP', 'ZINB').
weighted	Logical, to use weighting to moderate influential observations.
robust	Logical, use robust regression approach.
intercept	Which intercepts to keep. Dropped intercepts lead to regression through the origin (at the linear predictor scale).
xv	Logical, should leave-one-out error be calculated.
...	Other arguments passed to 'zeroinfl2()'.
ml	Named list of models.
coefs	logical, return coefficient table too.
model_id	model ID or model IDs (can be multiple from 'names(ml)').
do_boot	Logical, to do bootstrap or not.
do_avg	Logical, to do model averaging or not.
PI	PI object returned by 'mc_predict_total()'
id	Cell ID.
plot	Logical, to plot or just give summary.
breaks	Breaks argument passed to 'graphics::hist()'.
i	Column (variable) name or index.
interactive	Logical, draw interactive plot.

Examples

```
mc_options(B=10)

x <- read.csv(
  system.file("extdata/MayoMMU_QuerriedData.csv",
    package="moosecounter"))

#switch_response("cows") # for cows only
switch_response("total")

x <- mc_update_total(x)

mc_plot_univariate("Subalp-Shrub_250buf", x, "ZINB")

vars <- c("ELC-Subalpine", "Fire1982_2012", "Fire8212_DEM815",
  "NALC-Needle", "NALC-Shrub", "Subalp-Shrub_250buf",
  "ELCSub_Fire8212DEM815", "SubShrub250_Fire8212DEM815")

mc_plot_multivariate(vars, x)

ML <- list()
ML[["Model 0"]] <- mc_fit_total(x, dist="ZINB")
ML[["Model 1"]] <- mc_fit_total(x, vars[1:2], dist="ZINB")
ML[["Model 2"]] <- mc_fit_total(x, vars[2:3], dist="ZIP")
ML[["Model 3"]] <- mc_fit_total(x, vars[3:4], dist="ZINB")

mc_models_total(ML, x)
mc_plot_residuals("Model 3", ML, x)

PI <- mc_predict_total(
  model_id=c("Model 1", "Model 3"),
  ml=ML,
  x=x,
  do_boot=TRUE, do_avg=TRUE)

mc_get_pred(PI)
pred_density_moose_PI(PI)
mc_plot_predpi(PI)
mc_plot_pidistr(PI)
mc_plot_pidistr(PI, id=2)
```

Index

- * **dynamic**
 - run_app, 7
- * **misc**
 - run_app, 7
- * **models**
 - comp, 2
 - explore, 4
 - total, 7
- * **regression**
 - comp, 2
 - explore, 4
 - total, 7
- * **tree**
 - explore, 4

comp, 2

explore, 4

gassaway, 5

mc_check_comp (comp), 2

mc_fit_comp (comp), 2

mc_fit_total (total), 7

mc_gassaway (gassaway), 5

mc_get_pred (total), 7

mc_models_comp (comp), 2

mc_models_total (total), 7

mc_options, 6

mc_plot_comp (explore), 4

mc_plot_multivariate (explore), 4

mc_plot_pidistr (total), 7

mc_plot_predfit (total), 7

mc_plot_predpi (total), 7

mc_plot_residuals (total), 7

mc_plot_univariate (explore), 4

mc_predict_comp (comp), 2

mc_predict_total (total), 7

mc_update_total (total), 7

pred_density_moose_CPI (comp), 2

pred_density_moose_PI (total), 7

print.mc_gassaway (gassaway), 5

run_app, 7

subset_CPI_data (comp), 2

switch_response (total), 7

total, 7