Package 'eiPack'

June 3, 2006

Version 0.0-3

Date 2006-05-04

Title eiPack: Ecological Inference and Higher-Dimension Data Management

Author Olivia Lau <olau@fas.harvard.edu>, Ryan T. Moore <rtmoore@fas.harvard.edu>, Michael Kellermann <kellerm@fas.harvard.edu>

Maintainer Olivia Lau <olau@fas.harvard.edu>

Depends R (>= 2.0.0), MASS, coda, msm

Description Provides methods for analyzing RxC ecological contingency tables using the extreme case analysis, ecological regression, and Multinomial-Dirichlet ecological inference models. Also provides tools for manipulating higher-dimension data objects.

License GPL version 2 or newer

URL http://www.people.fas.harvard.edu/~olau/software

R topics documented:

---

bounds   *Deterministic bounds for units satisfying row thresholds*

Description

Calculates the deterministic bounds on the proportion of row members within a specified column.

Usage

```
bounds(formula, data, rows, column, excluded = NULL,
    threshold = 0.9, total = NULL)
```

Arguments

formula	a formula of the form `cbind(col1, col2, ...)` ~ `cbind(row1, row2, ...)`. Columns and rows must have the same totals, and should include all individuals in the ecological unit.
data	a data frame
rows	a character vector specifying the rows of interest
column	a character string specifying the column marginal of interest
excluded	an optional character string or vector specifying the columns that are part of the total set of individuals, but not part of the quantity of interest for which the bounds are calculated. For example, if the quantity of interest is Democratic share of the two-party vote, non-voters would likely be an excluded column.
threshold	the minimum proportion of the unit that row members must comprise for the bounds to be calculated for that unit. If `threshold = 0`, bounds will be calculated for all units.
total	if row and/or column data are given as proportions, this vector of total individuals by unit is required

Value

bounds	a list of deterministic bounds for all units in which row proportions meet the threshold
intersection	if the intersection of the deterministic bounding intervals is non-empty, the intersection is returned. Otherwise, 'NA' is returned.

Author(s)

Ryan T. Moore <⟨rtmoore@fas.harvard.edu⟩>

See Also

`plot.bounds`

---

cover.plot   *Unit-level coverage plots for beta parameters from MD EI model*

Description

Generates a plot of central credible intervals for the unit-level beta parameters from the Multinomial-Dirichlet ecological inference model (see `ei.MD.bayes`).

Usage

```
cover.plot(object, row, column, x = NULL, CI = 0.95,
           medians = TRUE, col = NULL, ylim = c(0,1),
           ylab, lty = par("lty"), lwd = par("lwd"), ...)
```

Arguments

object	output from `ei.MD.bayes`
row	a character string specifying the row marginal of interest
column	a character string specifying the column marginal of interest
x	an optional covarate to index the units along the x-axis
CI	(defaults to 0.95) a fraction between 0 and 1, specifying the coverage of the central credible interval to be plotted for each unit
medians	(defaults to `TRUE`) specifyies whether to plot the median. If `medians = FALSE`, the medians are not plotted.
col	an optional vector of colors to be passed to `plot` and `segments`. If `col` is of length two, then the first color is used for `plot` and the second for `segments`.
ylim	(defaults to `c(0,1)`) an optional range for the y-axis.
ylab	(defaults to `"Proportion of row in column"`) an optional label for the y-axis.
lty	an optional line type passed to `segments`.
lwd	an optional line width argument passed to `segments`.
...	additional arguments passed to `plot`.

Value

A plot with vertical intervals indicating the central credible intervals for each ecological unit.

Author(s)

Olivia Lau <⟨olau@fas.harvard.edu⟩>

See Also

`plot`, `segments`, `par`

---

density.plot	*Density plots for population level parameters*

Description

Generates a density plot for population level quantities of interest output by `lambda.MD`, `lambda.reg`, and `lambda.reg.bayes`. For the Bayesian methods, `density.plot` plots the kernel density for the draws. For the frequentist `lambda.reg` method, `density.plot` plots the canonical Normal density conditional on the mean and standard error output by `lambda.reg`.

Usage

```
density.plot(x, by = "column", col,
             xlim, ylim,
             main = "", sub = NULL, xlab, ylab,
             lty = par("lty"), lwd = par("lwd"), ...)
```

Arguments

x	output from `lambda.MD`, `lambda.reg`, or `lambda.reg.bayes`.
by	character string (defaulting to `"column"`) specifying whether to panel the density plot by `"row"` or `"column"` marginal.
col	an optional vector of colors, with length corresponding to the number of marginals selected in `by`. Defaults to `rainbow`.
xlim,ylim	optional limits for the x-axis and y-axis, passed to `plot`.
main,sub	optional title and subtitle, passed to `plot`.
xlab,ylab	optional labels for the x- and y-axes, passed to `plot`.
lty,lwd	optional arguments for line type and line width, passed to `lines` and `plot`.
...	additional arguments passed to `par`.

Value

A plot with density lines for the selected margin (row or column).

Author(s)

Olivia Lau <⟨olau@fas.harvard.edu⟩>

See Also

`plot`, `segments`, `par`

`ei.MD.bayes`	*Multinomial Dirichlet model for Ecological Inference in RxC tables*

Description

Implements a version of the model suggested in Rosen et al. (2001)

Usage

```
ei.MD.bayes(formula, covariate = NULL, total = NULL, data,
     lambda1 = 4, lambda2 = 2, tune.list = NULL,
     start.list = NULL, sample = 1000, thin = 1,
     burnin = 1000, verbose = 0, ret.beta = 'r',
     ret.mcmc = TRUE, usrfun = NULL, ...)
```

Arguments

`formula`	An R formula object of the form 'cbind(column.names) cbind(row.names)' with rows as the predictor and columns as the response.
`covariate`	An R formula object of the form ' covariate'. The default is `covariate = NULL`, which fits the model without a covariate.
`total`	A vector containing the number of individuals in each unit. Required when row or column marginals are proportions. The default is `total = NULL`, which assumes that the data is in the form of counts.
`data`	A data frame containing the variables specified in formula
`lambda1`	The shape parameter for the gamma prior. The default is 4.
`lambda2`	The rate parameter for the gamma prior. The default is 2.
`tune.list`	A list containing tuning parameters for each block of parameters. See 'details' for more information. Typically, this will be a list generated by `tuneMD`. The default is NULL, in which case fixed tuning parameters are used.
`start.list`	A list containing starting values for each block of parameters. See 'details' for more information. The default is '`start.list = NULL`', which generates appropriate random starting values.
`sample`	Desired number of draws to be returned from chain. The default is 1000.
`thin`	Thinning interval for posterior draws. The default is 1, but in most problems a much larger thinning interval will be necessary.
`burnin`	Number of initial iterations to be discarded. The default is 1000, but in most problems, a longer burnin period will be necessary.
`verbose`	A switch which determines whether or not the progress of the sampler is printed to the screen. If 'verbose' is greater than 0 the iteration number is printed to the screen every 'verbose'th iteration. The default is 0.
`ret.beta`	A character indicating how the posterior draws of beta should be handled: 'r'eturn as an R object, 's'ave as .txt.gz files, 'd'iscard. Default is to return as an R object.
`ret.mcmc`	A logical indicating how the samples from the posterior should be returned. If TRUE, samples are returned as coda 'mcmc' objects. If FALSE, samples are returned as arrays. The default is TRUE.

usrfun   Accepts a user defined function to obtain quantities of interest while drawing from the MCMC chain. The default is NULL, in which case no function is necessary.

...   Additional arguments passed to function.

Details

The function will accept user-specified starting values as an argument. If the model includes a covariate, the starting values must be a list with the following elements, in this order:

start.dr   a vector of length 'rows' of starting values for Dr. Starting values for Dr must be greater than zero.

start.betas   an r by c by precincts array of starting values for Beta. Each row of every precinct must sum to 1.

start.gamma   an r by c matrix of starting values for Gamma. Values in the right-most column must be zero.

start.delta   an r by c matrix of starting values for Delta. Values in the right-most column must be zero.

If there is no covariate, the starting values must be a list with the following elements:

start.alphas   an r by c matrix of starting values for Alpha. Starting values for Alpha must be greater than zero.

start.betas   an r by c by precincts array of starting values for Beta. Each row of every precinct must sum to 1.

The function will accept user-specified tuning parameters as an argument. If the model includes a covariate, the tuning parameters must be a list with the following elements, in this order:

tune.dr   a vector of length rows of tuning parameters for Dr

tune.beta   an r by (c - 1) by precincts array of tuning parameters for Beta

tune.gamma   an r by (c - 1) matrix of tuning parameters for Gamma

tune.delta   an r by (c - 1) matrix of tuning parameters for Delta

If there is no covariate, the tuning parameters are a list with the following elements:

tune.alpha   an r by c matrix of tuning parameters for Alpha

tune.beta   an r by (c - 1) by precincts array of tuning parameters for Beta

Value

A list containing

draws   A list containing samples from the posterior distribution of the parameters. If a covariate is included in the model, the list contains:

         Dr   Posterior draws for Dr parameters. If 'ret.mcmc = TRUE', a rows by sample matrix saved as an mcmc object. Otherwise, a rows by sample matrix

      Beta   Posterior draws for beta parameters. Only returned if 'ret.beta = TRUE'. If 'ret.mcmc = TRUE', a (rows x columns x units) by sample matrix saved as an mcmc object. Otherwise, a rows by columns by units by sample array

  Gamma   Posterior draws for gamma parameters. If 'ret.mcmc = TRUE', a (rows x (columns - 1)) by sample matrix saved as an mcmc object. Otherwise, a rows by (columns - 1) by sample array

Delta	Posterior draws for delta parameters. If 'ret.mcmc = TRUE', a (rows x (columns - 1)) by sample matrix saved as an mcmc object. Otherwise, a rows by (columns - 1) by sample array
Cell.count	Total number of individuals in each interior cell for each saved iteration
	If the model is fit without a covariate, the list includes:
Alpha	Posterior draws for alpha parameters. If 'ret.beta = TRUE', a (rows x columns) by sample matrix saved as an mcmc object. Otherwise, a rows by columns by sample array
Beta	Same as above.
Cell.count	Same as above.
acc.ratios	A list containing acceptance ratios for the parameters. If the model includes a covariate, the list includes:
dr.acc	Acceptance ratios for Dr
beta.acc	Acceptance ratios for Beta
gamma.acc	Acceptance ratios for Gamma and Delta
	If the model is fit without a covariate , the list includes:
alpha.acc	Acceptance ratios for Alpha
beta.acc	Acceptance ratios for Beta
usrfun	Output from user-defined function, if included
call	Call to function

Author(s)

Michael Kellermann <⟨kellerm@fas.harvard.edu⟩> and Olivia Lau <⟨olau@fas.harvard.edu⟩>

References

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. http://www-fis.iarc.fr/coda/.

Ori Rosen, Wenxin Jiang, Gary King, and Martin A. Tanner. 2001. "Bayesian and Frequentist Inference for Ecological Inference: The $R \times C$ Case." *Statistica Neerlandica* 55: 134-156.

See Also

lambda.MD, cover.plot, density.plot, tuneMD, mergeMD

---

ei.reg.bayes	*Ecological regression using Bayesian Normal regression*

Description

Estimate an ecological regression using Bayesian normal regression.

Usage

```
ei.reg.bayes(formula, data, contest = NULL, sample = 1000,
    weights = NULL)
```

Arguments

`formula`	An R formula object
`data`	data frame containing the variables specified in formula
`contest`	a character string identifying the contest
`sample`	number of draws from the posterior
`weights`	a vector of weights

Value

A rows x columns x sample array containing posterior draws for each cell fraction.

Author(s)

Olivia Lau <⟨olau@fas.harvard.edu⟩>, Ryan T. Moore <⟨rtmoore@fas.harvard.edu⟩>

`ei.reg`   *Ecological regression*

Description

Estimate an ecological regression using least squares.

Usage

```
ei.reg(formula, data, contest = NULL, ...)
```

Arguments

`formula`	An R formula object.
`data`	data frame containing the variables specified in formula
`contest`	a character string identifying the contests for which you would like to run the model
`...`	Additional arguments passed to `lm`.

Value

A rows x columns table containing point estimates for each cell fraction.

Author(s)

Olivia Lau <⟨olau@fas.harvard.edu⟩>, Ryan T. Moore <⟨rtmoore@fas.harvard.edu⟩>

---

lambda.MD   *Calculate shares using data from MD model*

Description

Calculates the share of row members in the selected columns.

Usage

```
lambda.MD(object, columns, ret.mcmc = TRUE)
```

Arguments

object	An R object of class 'eiMD', the output from `ei.MD.bayes`
columns	a vector of column names to be included in calculating the shares
ret.mcmc	A logical indicating how the samples from the posterior should be returned. If `TRUE` (default), samples are returned as coda 'mcmc' objects. If `FALSE`, samples are returned as arrays.

Value

Returns either a ((rows * included columns) x samples) matrix as an MCMC object or a (rows x included columns x samples) array.

Author(s)

Michael Kellermann <⟨kellerm@fas.harvard.edu⟩> and Olivia Lau <⟨olau@fas.harvard.edu⟩>

---

lambda.reg.bayes   *Calculate shares using data from Bayesian regression model*

Description

Calculates the share of row members in a particular column

Usage

```
lambda.reg.bayes(object, columns, ret.mcmc = TRUE)
```

Arguments

object	An R object of class 'eiRegBayes', the output from ei.reg.bayes
columns	a vector of column names to be included in calculating the shares
ret.mcmc	If TRUE, posterior shares are returned as a coda mcmc object in a sample by (groups*length(columns)) matrix. If FALSE, posterior shares are returned in a 3-way array of dimension groups by length(columns) by samples.

Value

Returns either a matrix mcmc object or a 3-way array. See argument `ret.mcmc` above.

Author(s)

Ryan T. Moore <⟨rtmoore@fas.harvard.edu⟩>

---

lambda.reg	*Calculate shares using data from regression model*

Description

Calculates the share of row members in a particular column

Usage

```
lambda.reg(object, columns)
```

Arguments

`object`	An R object of class 'eiReg', the output from ei.reg
`columns`	a vector of column names to be included in calculating the shares

Details

Standard errors are calculated using the delta method as implemented in the library `msm`. The arguments passed to `deltamethod` in `msm` include

g A list of tranformations of the form $\sim x1 / (x1 + x2 + x3)$, $\sim x2 / (x1 + x2 + x3)$, etc.

mean The estimated proportions of the row members in the specified columns, as a proportion of the total number of row members.

cov A diagonal matrix with the estimated variance of each element of X on the diagonal. Estimate come from `object$cov.matrices`, the estimated covariance matrix from the regression of the relevant column.

Value

Returns a rows by included columns matrix. Standard errors are calculated using the delta method as implemented in the library `msm`.

Author(s)

Ryan T. Moore <⟨rtmoore@fas.harvard.edu⟩>

| mergeMD | *Combine output from multiple eiMD objects* |

Description

Allows users to combine output from several MCMC chains

Usage

```
mergeMD(objectlist, discard = 0)
```

Arguments

objectlist A list containing the names of multiple eiMD objects generated from the same model.

discard The number of draws to discard from the beginning of each chain. Default is to retain all draws.

Value

Returns an eiMD object of the same format as the input.

Author(s)

Michael Kellermann <⟨kellerm@fas.harvard.edu⟩> and Olivia Lau <⟨olau@fas.harvard.edu⟩>

References

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. 2002. *Output Analysis and Diagnostics for MCMC (CODA)*. http://www-fis.iarc.fr/coda/.

Ori Rosen, Wenxin Jiang, Gary King, and Martin A. Tanner. 2001. "Bayesian and Frequentist Inference for Ecological Inference: The $R \times C$ Case." *Statistica Neerlandica* 55: 134-156.

See Also

ei.MD.bayes

| plot.bounds | *Plot of deterministic bounds for units satisfying row thresholds* |

Description

Plots the deterministic bounds on the proportion of row members within a specified column.

Usage

```
plot.bounds(x, row, column, labels = TRUE, order = NULL,
    intersection = TRUE, xlab, ylab, col = par("fg"),
    lty = par("lty"), lwd = par("lwd"), ...)
```

Arguments

x	output from [bounds](#)
row	a character string specifying the row of interest
column	a character string specifying the column of interest
labels	a logical toggle specifying whether precinct labels should be printed above interval bounds
order	an optional vector of values between 0 and 1 specifying the order (left-to-right) in which interval bounds are plotted
intersection	a logical toggle specifying whether the intersection of all plotted bounds (if it exists) should be plotted
xlab, ylab, ...	
	additional arguments passed to `plot`
col, lty, lwd	
	additional arguments passed to `segments`

Value

A plot with vertical intervals indicating the deterministic bounds on the quantity of interest, and (optionally) a single horizontal interval indicating the intersection of these unit bounds.

Author(s)

Ryan T. Moore <⟨rtmoore@fas.harvard.edu⟩>

See Also

`bounds`

---

read.betas	*Function to read in eiMD parameter chains saved to disk*

Description

In [ei.MD.bayes](#), users have the option to save parameter chains for the unit-level betas to disk rather than returning them to the workspace. This function reconstructs the parameter chains by reading them back into R and producing either an array or an `mcmc` object.

Usage

```
read.betas(rows, columns, units, dir = NULL, ret.mcmc = TRUE)
```

Arguments

rows	a character vector of the row marginals to be read back in
columns	a character vector of the column marginals to be read back in
units	a character of numeric vector with the units to be read back in
dir	an optional character string identifying the directory in which parameter chains are stored (defaults to `getwd`)
ret.mcmc	a logical value specifying whether to return the parameters as an `mcmc` object (defaults to `TRUE`)

Value

If ret.mcmc = TRUE, an mcmc object with row names corresponding to the parameter chains. If ret.mcmc = FALSE, an array with dimensions named according to the selected rows, columns, and units.

Author(s)

Olivia Lau <olau@fas.harvard.edu>

See Also

ei.MD.bayes

---

redistrict   *Redistricting Monte-Carlo data*

Description

Precinct-level observations for a hypothetical jursidiction with four proposed districts.

Usage

```
data(redistrict)
```

Format

A table containing 150 observations and 9 variables:

precinct precinct identifier

district proposed district number

avg.age average age

per.own percent homeowners

black number of black voting age persons

white number of white voting age persons

hispanic number of hispanic voting age persons

total total number of voting age persons

dem Number of votes for the Democratic candidate

rep Number of votes for the Republican candidate

no.vote Number of non voters

Source

Daniel James Greiner

---

senc *Party registration in south-east North Carolina*

Description

Registration data for White, Black, and Native American voters in eight counties of south-eastern North Carolina in 2001.

Usage

```
data(senc)
```

Format

A table containing 212 observations and 18 variables:

county county name

precinct precinct name

total number of registered voters in precinct

white number of White registered voters

black number of Black registered voters

natam number of Native American registered voters

dem number of registered Democracts

rep number of registered Republicans

other number of registered voters without major party affiliation

whdem number of White registered Democrats

whrep number of White registered Republicans

whoth number of White registered voters without major party affiliation

bldem number of Black registered Democrats

blrep number of Black registered Republicans

bloth number of Black registered voters without major party affiliation

natamdem number of Native American registered Democrats

natamrep number of Native American registered Republicans

natamoth number of Native American registered voters without major party affiliation

Source

Exerpted from North Carolina General Assembly 2001 redistricting data, http://www.ncga.state.nc.us/redistricting/Archives/Data/2001/available_data.html

---

tuneA *Tuning parameters for alpha hyperpriors in RxC EI model*

Description

Tuning parameters for hyperpriors in RxC EI model

Usage

```
data(tuneA)
```

Format

A table containing 3 rows and 3 colums.

---

tuneB *Tuning parameters for the precinct level parameters in the RxC EI model*

Description

A vector containing tuning parameters for the precinct level parameters in the RxC EI model.

Usage

```
data(tuneB)
```

Format

A vector of length 3 x 2 x 150 containing the precinct level tuning parameters for the redistricting sample data.

Examples

```
data(tuneB)
tuneB <- array(tuneB[[1]], dim = c(3, 2, 150))
```

---

tuneMD   *Generate tuning parameters for MD model*

Description

An adaptive algorithm to generate tuning parameters for the MCMC algorithm implemented in
`ei.MD.bayes`. Since we are drawing each parameter one at a time, target acceptance rates are
between 0.4 to 0.6.

Usage

```
tuneMD(formula, covariate = NULL, data, ntunes = 10,
    sample = 1000, ...)
```

Arguments

`formula`	An R formula object in the form `cbind(row1, row2, ...)` ~ `cbind(column1, column2, ...)`
`covariate`	An R formula for the optional covariate in the form `~ x`
`data`	data frame containing the variables specified in `formula` and `covariate`
`ntunes`	number of times to iterate the tuning algorithm
`sample`	number of iterations for each tuning run
`...`	additional arguments passed to `ei.MD.bayes`

Value

A list containing matricies of tuning parameters.

Author(s)

Olivia Lau <⟨olau@fas.harvard.edu⟩>

See Also

`ei.MD.bayes`

Index

17