

Package ‘baggr’

August 1, 2019

Type Package

Title Bayesian Aggregate Treatment Effects

Version 0.1.0

Maintainer Witold Wiecek <witold.wiecek@gmail.com>

Description Running and comparing meta-analyses of data with hierarchical Bayesian models in Stan, including convenience functions for formatting data, plotting and pooling measures specific to meta-analysis.

License GPL (>= 3)

Encoding UTF-8

LazyData true

ByteCompile true

Depends R (>= 3.5.0), Rcpp (>= 0.12.17), methods

Imports rstan (>= 2.18.1), rstantools (>= 1.5.0), bayesplot, crayon, ggplot2, gridExtra, utils, stats

LinkingTo StanHeaders (>= 2.18.1), rstan (>= 2.18.1), BH (>= 1.66.0-1), Rcpp (>= 0.12.17), RcppEigen (>= 0.3.3.4.0)

SystemRequirements GNU make

NeedsCompilation yes

RoxygenNote 6.1.1

Suggests testthat, knitr

VignetteBuilder knitr

URL <https://github.com/wwiecek/baggr>

BugReports <https://github.com/wwiecek/baggr/issues>

Language en-GB

Author Witold Wiecek [cre, aut],
Rachael Meager [aut],
Trustees of Columbia University [cph] (tools/make_cc.R)

Repository CRAN

Date/Publication 2019-08-01 11:30:02 UTC

R topics documented:

baggr	2
baggr_compare	4
baggr_package	5
baggr_plot	5
convert_inputs	6
group_effects	7
loocv	8
microcredit	9
microcredit_simplified	9
plot.baggr	10
pooling	11
prepare_ma	11
print.baggr	13
schools	13
show_model	14
treatment_effect	14
Index	15

baggr

Bayesian aggregate treatment effects model

Description

Bayesian inference on parameters of an average treatment effects model that's appropriate to the supplied individual- or group-level data, using Hamiltonian Monte Carlo in Stan. (For overall package help file see [baggr_package](#))

Usage

```
baggr(data, model = NULL, prior = NULL, pooling = "partial",
      joint_prior = TRUE, test_data = NULL, quantiles = seq(0.05, 0.95,
      0.1), outcome = "outcome", group = "group",
      treatment = "treatment", warn = TRUE, ...)
```

Arguments

data	data frame with summary or individual level data to meta-analyse
model	if NULL, detected automatically from input data otherwise choose from "rubin", "mutau", "individual", "quantiles"
prior	list of prior arguments passed directly to each model (see Details)
pooling	choose from "none", "partial" (default) and "full"
joint_prior	If TRUE, mu and tau will have joint distribution. If FALSE, they have independent priors. Ignored if no control (mu) data exists.

test_data	data for cross-validation; NULL for no validation, otherwise a data frame with the same columns as data argument
quantiles	if model = "quantiles", a vector indicating which quantiles of data to use (with values between 0 and 1)
outcome	character; column name in (individual-level) data with outcome variable values
group	character; column name in data with grouping factor; it's necessary for individual-level data, for summarised data it will be used as labels for groups when displaying results
treatment	character; column name in (individual-level) data with treatment factor;
warn	print an additional warning if Rhat exceeds 1.05
...	extra options passed to Stan function, e.g. control = list(adapt_delta = 0.99), number of iterations etc.

Details

Running baggr requires 1/ data preparation, 2/ choice of model, 3/ choice of priors. All three are discussed in depth in the package vignette (`vignette("baggr")`).

Data. For aggregate data models you need a data frame with columns tau and se or tau, mu, se.tau, se.mu. An additional column can be used to provide labels for each group (by default column group is used if available, but this can be customised – see the example below). For individual level data three columns are needed: outcome, treatment, group. These are identified by using the outcome, treatment and group arguments.

When working with individual-level data, many data preparation steps (summarising, standardisation etc.) can be done through a helper function `prepare_ma`. Using it will also automatically format data inputs to be work with `baggr()`.

Models. Available models are:

- for the means: "rubin" model for average treatment effect, "mutau" version which takes into account means of control groups, "full" model which reduces to "mu and tau" (if no covariates are used)
- "quantiles" model is also available (see Meager, 2019 in references)

If no model is specified, the function tries to infer the appropriate model automatically.

Priors. It is optional to specify priors yourself, as the package will try propose an appropriate prior for the input data if prior=NULL. To set the priors yourself, please refer to the list in the vignette("baggr")

Value

baggr class structure: a list including Stan model fit alongside input data, pooling metrics, various model properties. If test data is used, mean value of $-2 \cdot \text{lpd}$ is reported as mean_lpd

Author(s)

Witold Wiecek, Rachael Meager

Examples

```
df_pooled <- data.frame("tau" = c(1, -1, .5, -.5, .7, -.7, 1.3, -1.3),
  "se" = rep(1, 8),
  "state" = datasets::state.name[1:8])
baggr(df_pooled) #baggr automatically detects the input data
# correct labels, different pooling & passing some options to Stan
baggr(df_pooled, group = "state", pooling = "full", iter = 500)
```

baggr_compare

Compare a(ny) number of baggr models side by side

Description

Compare a(ny) number of baggr models side by side

Usage

```
baggr_compare(..., style = "areas", arrange = "single")
```

Arguments

...	Either any number of objects of class baggr (you can name your objects, see the example below) or the same arguments you'd pass to baggr() function, but with pooling = ... omitted. In the latter case 3 models will be run, with pooling set to none, partial and full.
style	What kind of plot to display (if arrange = "grid"), passed to style argument in baggr_plot
arrange	If single, generate a comparison single plot; if grid, display multiple plots side-by-side.

Value

ggplot graphic is rendered

Author(s)

Witold Wiecek

Examples

```
# Most basic comparison between no, partial and full pooling
# (This will run the models)
baggr_compare(schools)

# Compare existing models:
bg1 <- baggr(schools, pooling = "partial")
```

```
bg2 <- baggr(schools, pooling = "full")
baggr_compare("Partial pooling model" = bg1, "Full pooling" = bg2,
              arrange = "grid")

# You can also compare different subsets of input data
bg1_small <- baggr(schools[1:6,], pooling = "partial")
baggr_compare("8 schools model" = bg1, "First 6 schools" = bg1_small)
```

baggr_package

baggr - a package for Bayesian meta-analysis

Description

This is *baggr* (pronounced as *bagger* or *badger*), a Bayesian meta-analysis package for R using [Stan](#). *Baggr* is intended to be user-friendly and transparent so that it's easier to understand the models you are building and criticise them. The current version is a stable prototype of a tool that's in active development so we are counting on your feedback.

Details

Baggr provides a suite of Bayesian aggregation models for both summary statistics and full data sets to synthesise evidence collected from different groups, contexts or time periods.

Getting help

This is only a simple package help file. For documentation of the main function for conducting analyses see [baggr](#). For description of models, data types and priors available in the package, try the built-in vignette (`vignette("baggr")`).

Author(s)

Witold Wiecek, Rachael Meager

baggr_plot

Plotting method in baggr package

Description

Extracts study effects from the *baggr* model and sends them to one of *bayesplot* package plotting functions.

Usage

```
baggr_plot(bg, mean = FALSE, style = "intervals", prob = 0.5,
           prob_outer = 0.95, vline = TRUE, order = TRUE, ...)
```

Arguments

bg	object of class baggr
mean	logical; plot mean treatment effect alongside individual study effects?
style	one of areas, intervals
prob	Probability mass for the inner interval in visualisation
prob_outer	Probability mass for the outer interval in visualisation
vline	logical; show vertical line through 0 in the plot?
order	logical; sort groups by magnitude of treatment effect?
...	extra arguments to pass to the bayesplot functions

Value

ggplot2 object

Author(s)

Witold Wiecek, Rachael Meager

See Also

[bayesplot::MCMC-intervals](#)

Examples

```
fit <- baggr(schools, pooling = "none")
plot(fit)
plot(fit, style = "areas", order = FALSE)
```

convert_inputs	<i>Convert inputs for baggr models Converts data to Stan inputs, checks integrity of data and suggests default model if needed.</i>
----------------	---

Description

Convert inputs for baggr models

Converts data to Stan inputs, checks integrity of data and suggests default model if needed.

Usage

```
convert_inputs(data, model, quantiles, group = "group",
  outcome = "outcome", treatment = "treatment", test_data = NULL)
```

Arguments

data	‘data.frame‘ with desired modelling input
model	valid model name used by baggr; see baggr for allowed models if model = NULL, this function will try to find appropriate model automatically
quantiles	vector of quantiles to use (only applicable if model = "quantiles")
group	name of the column with grouping variable
outcome	name of column with outcome variable
treatment	name of column with treatment variable
test_data	same format as data argument, gets left aside for testing purposes (see baggr)

Details

Typically this function is only called within [baggr](#) and you do not need to use it yourself. It can be useful to understand inputs or to run models which you modified yourself.

Value

R structure that’s appropriate for use by [baggr](#) Stan models; group_label, model and n_groups are included as attributes and are necessary for [baggr](#) to work correctly

Author(s)

Witold Wiecek, Rachael Meager

group_effects	<i>Extract baggr study effects</i>
---------------	------------------------------------

Description

Internal function used as a helper for plotting and printing of results.

Usage

```
group_effects(bg, summary = FALSE, interval = 0.95)
```

Arguments

bg	baggr object
summary	logical; summarise result?
interval	uncertainty interval width (numeric between 0 and 1), used if summarising

Value

a matrix with MCMC samples or summaries

loocv	<i>Leave one out cross-validation for baggr models</i>
-------	--

Description

Creates k baggr models by leaving out one group at the time and calculating log predictive density lpd_k for that group (see Gelman *et al*). Returned value is a $-2 \sum lpd_k$. Inputs to this function are same as to [baggr](#), with additional option to return individual models.

Usage

```
loocv(data, return_models = FALSE, ...)
```

Arguments

data	Input data frame - same as for baggr function.
return_models	logical; if FALSE, summary statistics will be returned and the models discarded; if TRUE, a list of models will be returned alongside summaries
...	Additional arguments passed to baggr .

Details

For running more computation-intensive models, consider setting the `mc.cores` option before running `loocv`, e.g. `options(mc.cores = 4)` (by default `baggr` runs 4 MCMC chains in parallel). As a default, `rstan` runs "silently" (`refresh=0`). To see sampling progress, please set e.g. `loocv(data, refresh = 500)`.

For more information on cross-validation see [this overview article](#)

Value

log predictive density value, an object of class `baggr_cv`; full model, prior values and lpd of each model are also returned these can be examined by using `attributes()` function

Author(s)

Witold Wiecek

Examples

```
# even simple examples may take a while
cv <- loocv(schools, pooling = "partial")
print(cv)      # returns the lpd value
attributes(cv) # more information is included in the object
```

microcredit	<i>7 studies on effect of microcredit supply</i>
-------------	--

Description

This dataframe contains the data used in Meager (2019) to estimate hierarchical models on the data from 7 randomized controlled trials of expanding access to microcredit.

Usage

```
microcredit
```

Format

A data frame with 40267 rows, 7 study identifiers and 7 outcomes

Details

The columns include the group indicator which gives the name of the lead author on each of the respective studies, the value of the 6 outcome variables of most interest (consumer durables spending, business expenditures, business profit, business revenues, temptation goods spending and consumption spending) all of which are standardised to USD PPP in 2009 dollars per two weeks (these are flow variables), and finally a treatment assignment status indicator.

The dataset has not otherwise been cleaned and therefore includes NAs and other issues common to real-world datasets.

For more information on how and why these variables were chosen and standardised, see Meager (2019) or consult the associated code repository which includes the standardisation scripts: [link](#)

References

Meager, Rachael (2019) Understanding the average impact of microcredit expansions: A Bayesian hierarchical analysis of seven randomized experiments. *American Economic Journal: Applied Economics*, 11(1), 57-91.

microcredit_simplified	<i>Simplified version of the microcredit dataset.</i>
------------------------	---

Description

This dataframe contains the data used in Meager (2019) to estimate hierarchical models on the data from 7 randomized controlled trials of expanding access to microcredit.

Usage

```
microcredit_simplified
```

Format

A data frame with 14224 rows, 7 study identifiers and 1 outcome

Details

The columns include the group indicator which gives the name of the lead author on each of the respective studies, the value of the household consumer durables spending standardised to USD PPP in 2009 dollars per two weeks (these are flow variables), and finally a treatment assignment status indicator.

The dataset has not otherwise been cleaned and therefore includes NAs and other issues common to real data.

For more information on how and why these variables were chosen and standardised, see Meager (2019) or consult the associated code repository: [link](#)

This dataset includes only complete cases and only the consumer durables outcome variable.

References

Meager, Rachael (2019) Understanding the average impact of microcredit expansions: A Bayesian hierarchical analysis of seven randomized experiments. *American Economic Journal: Applied Economics*, 11(1), 57-91.

plot.baggr

Generic plot for baggr package

Description

Using generic plot() on baggr output invokes `baggr_plot` visual. See therein for customisation options. Note that plot output is ggplot2 object.'

Usage

```
## S3 method for class 'baggr'  
plot(x, ...)
```

Arguments

x object of class baggr
... optional arguments, see baggr_plot

Value

ggplot2 object from baggr_plot

Author(s)

Witold Wiecek

pooling	<i>Pooling metrics for baggr</i>
---------	----------------------------------

Description

Compute pooling metrics given a baggr meta-analysis model

Usage

```
pooling(bg, metric = "gelman-hill", summary = TRUE)
```

Arguments

bg	output of a baggr() function
metric	for now we use "gelman-hill", other options will be added in the future
summary	logical; if FALSE a whole vector of pooling statistics is returned, otherwise only the means and intervals

Details

Pooling statistic describes the extent to which group-level estimates of treatment effect are shrunk toward average treatment effect in the meta-analysis model.

Value

Matrix with mean and intervals for chosen pooling metric, each row corresponding to one meta-analysis group.

Author(s)

Witold Wiecek, Rachael Meager

prepare_ma	<i>Convert from individual to summary data in meta-analyses</i>
------------	---

Description

Allows one-way conversion from full to summary data. Input must be pre-formatted appropriately.

Usage

```
prepare_ma(data, log = FALSE, cfb = FALSE, summarise = TRUE,
  treatment = "treatment", baseline = NULL, group = "group",
  outcome = "outcome")
```

Arguments

data	data.frame of individual-level observations with columns for outcome (numeric), treatment (values 0 and 1) and group (numeric, character or factor); column names can be user-defined (see below)
log	logical; log-transform the outcome variable?
cfb	logical; calculate change from baseline? If yes, the outcome variable is taken as a difference between values in outcome and baseline columns
summarise	logical; convert to aggregate level data?
treatment	name of column with treatment variable
baseline	name of column with baseline variable
group	name of the column with grouping variable
outcome	name of column with outcome variable

Details

The conversions done by this function are not typically needed and may happen automatically when data is fed to [baggr](#). However, this function can be used to explicitly convert from full to reduced (summarised) data without analysing it in any model. It can be useful for examining your data.

If multiple operations are performed, they are taken in this order:

1. conversion to log scale,
2. calculating change from baseline,
3. summarising data.

Value

data.frame with columns mu, se.mu, tau and se.tau

Author(s)

Witold Wiecek

See Also

[convert_inputs](#) for how data is converted into Stan inputs;

print.baggr	<i>S3 print method for objects of class baggr (model fits)</i>
-------------	--

Description

This print method for a very concise summary of main model features. More info is included in the summary of the model and its attributes.

Usage

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

Arguments

x	object of class baggr
...	currently unused by this package: further arguments passed to or from other methods (print requirement)

schools	<i>8 schools example</i>
---------	--------------------------

Description

A classic example of aggregate level continuous data in Bayesian hierarchical modelling. This dataframe contains a column of estimated treatment effects of an SAT prep program implemented in 8 different schools in the US, and a column of estimated standard errors.

Usage

```
schools
```

Format

An object of class `data.frame` with 8 rows and 3 columns.

Details

See Gelman et al (1995), Chapter 5, for context and applied example.

References

Gelman, Andrew, John B. Carlin, Hal S. Stern, and Donald B. Rubin. Bayesian Data Analysis. Taylor & Francis, 1995.

show_model	<i>Show Stan code for baggr models or objects</i>
------------	---

Description

Show Stan code for baggr models or objects

Usage

```
show_model(model)
```

Arguments

model either a baggr object (fitted model) or one of "rubin", "mutau", "individual"

Value

Nothing is returned in R. Stan code will be opened externally (e.g. via notepad).

treatment_effect	<i>Average treatment effect in a baggr model</i>
------------------	--

Description

Average treatment effect in a baggr model

Usage

```
treatment_effect(bg)
```

Arguments

bg a [baggr](#) model

Value

A list with 2 vectors (corresponding to MCMC samples) tau (mean effect) and sigma_tau (SD)

Index

*Topic **datasets**

- microcredit, [9](#)
- microcredit_simplified, [9](#)
- schools, [13](#)

- baggr, [2](#), [5](#), [7](#), [8](#), [12](#), [14](#)
- baggr_compare, [4](#)
- baggr_package, [2](#), [5](#)
- baggr_package-package (baggr_package), [5](#)
- baggr_plot, [4](#), [5](#), [10](#)
- bayesplot::MCMC-intervals, [6](#)

- convert_inputs, [6](#), [12](#)

- group_effects, [7](#)

- loocv, [8](#)

- microcredit, [9](#)
- microcredit_simplified, [9](#)

- plot.baggr, [10](#)
- pooling, [11](#)
- prepare_ma, [3](#), [11](#)
- print.baggr, [13](#)

- schools, [13](#)
- show_model, [14](#)

- treatment_effect, [14](#)