Package 'bayesboot'

June 29, 2018

Type Package

Title An Implementation of Rubin's (1981) Bayesian Bootstrap

Version 0.2.2

Date 2018-06-28

Description Functions for performing the Bayesian bootstrap as introduced by Rubin (1981) <doi:10.1214/aos/1176345338> and for summarizing the result. The implementation can handle both summary statistics that works on a weighted version of the data and summary statistics that works on a resampled data set.

License MIT + file LICENSE

LazyData TRUE

URL https://github.com/rasmusab/bayesboot

BugReports https://github.com/rasmusab/bayesboot/issues

RoxygenNote 6.0.1

Imports plyr (>= 1.8.3), HDInterval(>= 0.1.1)

Depends R (>= 3.2.0)

Suggests testthat, foreach, doParallel, boot

Encoding UTF-8

NeedsCompilation no

Author Rasmus Bååth [aut, cre]

Maintainer Rasmus Bååth <rasmus.baath@gmail.com>

Repository CRAN

Date/Publication 2018-06-29 09:26:27 UTC

R topics documented:

`as.bayesboot` *Coerce to a* `bayesboot` *object*

Description

This converts an object into a data frame and adds the class `bayesboot`. Doing this is only useful in the case you would want to use the `plot` and `summary` methods for `bayesboot` objects.

Usage

```
as.bayesboot(object)
```

Arguments

`object`	Any object that can be converted to a data frame.

Value

A `data.frame` with subclass `bayesboot`.

`bayesboot` *The Bayesian bootstrap*

Description

Performs a Bayesian bootstrap and returns a `data.frame` with a sample of size `R` representing the posterior distribution of the (possibly multivariate) summary `statistic`.

Usage

```
bayesboot(data, statistic, R = 4000, R2 = 4000, use.weights = FALSE,
  .progress = "none", .parallel = FALSE, ...)
```

Arguments

data	Either a vector or a list, or a matrix or a data.frame with one datapoint per row. The format of data should be compatible with the first argument of statistic
statistic	A function implementing the summary statistic of interest where the first argument should take the data. If use.weights = TRUE then the second argument should take a vector of weights.
R	The size of the posterior sample from the Bayesian bootstrap.
R2	When use.weights = FALSE this is the size of the resample of the data used to approximate the weighted statistic.
use.weights	When TRUE the data will be reweighted, like in the original Bayesian bootstrap. When FALSE (the default) the reweighting will be approximated by resampling the data.
.progress	The type of progress bar ("none", "text", "tk", and "win"). See the .progress argument to adply in the plyr package.
.parallel	If TRUE enables parallel processing. See the .parallel argument to adply in the plyr package.
...	Other arguments passed on to statistic

Details

The summary statistic is a function of the data that represents a feature of interest, where a typical statistic is the mean. In bayesboot it is most efficient to define the statistic as a function taking the data as the first argument and a vector of weights as the second argument. An example of such a function is weighted.mean. Indicate that you are using a statistic defined in this way by setting use.weights = TRUE.

It is also possible to define the statistic as a function only taking data (and no weights) by having use.weights = FALSE (the default). This will, for each of the R Bayesian bootstrap draws, give a resampled version of the data of size R2 to statistic. This will be much slower than using use.weights = TRUE but will work with a larger range of statistics (the median, for example)

For more information regarding this implementation of the Bayesian bootstrap see the blog post Easy Bayesian Bootstrap in R. For more information about the model behind the Bayesian bootstrap see the blog post The Non-parametric Bootstrap as a Bayesian Model and, of course, the original Bayesian bootstrap paper by Rubin (1981).

Value

A data.frame with R rows, each row being a draw from the posterior distribution of the Bayesian bootstrap. The number of columns is decided by the length of the output from statistic. If statistic does not return a vector or data frame with named values then the columns will be given the names V1, V2, V3, etc. While the output is a data.frame it has subclass bayesboot which enables specialized summary and plot functions for the result of a bayesboot call.

Note

- While R and R2 are set to 4000 by default, that should not be taken to indicate that a sample of size 4000 is sufficient nor recommended.

- When using `use.weights = FALSE` it is important to use a summary statistic that does not depend on the sample size. That is, doubling the size of a dataset by cloning data should result in the same statistic as when using the original dataset. An example of a statistic that depends on the sample size is the sample standard deviation (that is, `sd`), and when using `bayesboot` it would make more sense to use the population standard deviation (as in the example below).

References

Miller, R. G. (1974) The jackknife - a review. *Biometrika*, **61(1)**, 1–15.

Rubin, D. B. (1981). The Bayesian bootstrap. *The annals of statistics*, **9(1)**, 130–134.

Examples

```
### A Bayesian bootstrap analysis of a mean ###

# Heights of the last ten American presidents in cm (Kennedy to Obama).
heights <- c(183, 192, 182, 183, 177, 185, 188, 188, 182, 185);
b1 <- bayesboot(heights, mean)
# But it's more efficient to use the a weighted statistic.
b2 <- bayesboot(heights, weighted.mean, use.weights = TRUE)

# The result of bayesboot can be plotted and summarized
plot(b2)
summary(b2)

# It can also be easily post processed.
# Here the probability that the mean is > 182 cm.
mean( b2[,1] > 182)

### A Bayesian bootstrap analysis of a SD ###

# When use.weights = FALSE it is important that the summary statistics
# does not change as a function of sample size. This is the case with
# the sample standard deviation, so here we have to implement a
# function calculating the population standard deviation.
pop.sd <- function(x) {
  n <- length(x)
  sd(x) * sqrt( (n - 1) / n)
}

b3 <- bayesboot(heights, pop.sd)
summary(b3)

### A Bayesian bootstrap analysis of a correlation coefficient ###

# Data comparing two methods of measuring blood flow.
# From Table 1 in Miller (1974) and used in an example
# by Rubin (1981, p. 132).
blood.flow <- data.frame(
  dye = c(1.15, 1.7, 1.42, 1.38, 2.80, 4.7, 4.8, 1.41, 3.9),
  efp = c(1.38, 1.72, 1.59, 1.47, 1.66, 3.45, 3.87, 1.31, 3.75))
```

```
# Using the weighted correlation (corr) from the boot package.
library(boot)
b4 <- bayesboot(blood.flow, corr, R = 1000, use.weights = TRUE)
hist(b4[,1])

### A Bayesian bootstrap analysis of lm coefficients ###

# A custom function that returns the coefficients of
# a weighted linear regression on the blood.flow data
lm.coefs <- function(d, w) {
  coef( lm(efp ~ dye, data = d, weights = w) )
}

b5 <- bayesboot(blood.flow, lm.coefs, R = 1000, use.weights = TRUE)

# Plotting the marginal posteriors
plot(b5)

# Plotting a scatter of regression lines from the posterior
plot(blood.flow)
for(i in sample(nrow(b5), size = 20)) {
  abline(coef = b5[i, ], col = "grey")
}
```

`plot.bayesboot` *Plot the result of* `bayesboot`

Description

Produces histograms showing the marginal posterior distributions from a `bayesboot` call. Uses the `plotPost` function to produce the individual histograms.

Usage

```
## S3 method for class 'bayesboot'
plot(x, cred.mass = 0.95, plots.per.page = 3,
  cex = 1.2, cex.lab = 1.3, ...)
```

Arguments

`x`	The bayesboot object to plot.
`cred.mass`	the probability mass to include in credible intervals, or NULL to suppress plotting of credible intervals.
`plots.per.page`	The maximum numbers of plots per page.
`cex, cex.lab, ...`	Further parameters passed on to `plotPost`.

plotPost *Graphic display of a posterior probability distribution*

Description

Plot the posterior probability distribution for a single parameter from a vector of samples, typically from an MCMC process, with appropriate summary statistics.

Usage

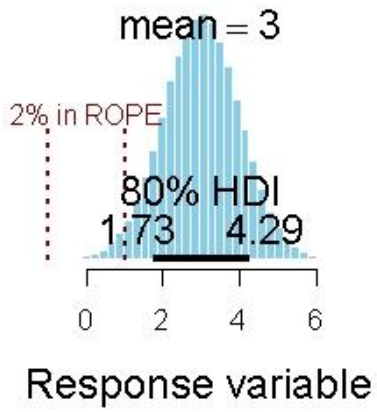
```
plotPost(paramSampleVec, credMass = 0.95, compVal = NULL, ROPE = NULL,
  HDItextPlace = 0.7, showMode = FALSE, showCurve = FALSE, ...)
```

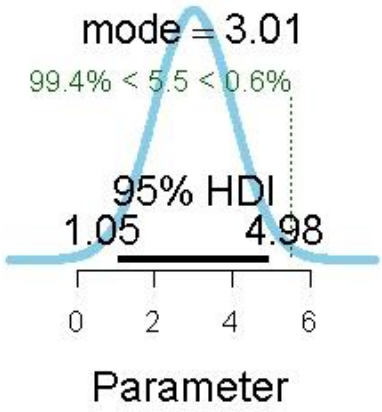
Arguments

paramSampleVec	A vector of samples drawn from the target distribution.
credMass	the probability mass to include in credible intervals, or NULL to suppress plotting of credible intervals.
compVal	a value for comparison with those plotted.
ROPE	a two element vector, such as c(-1, 1), specifying the limits of the Region Of Practical Equivalence.
HDItextPlace	a value in [0,1] that controls the horizontal position of the labels at the ends of the HDI bar.
showMode	logical: if TRUE, the mode is displayed instead of the mean.
showCurve	logical: if TRUE, the posterior density will be represented by a kernel density function instead of a histogram.
...	graphical parameters and the breaks parameter for the histogram.

Details

The data are plotted either as a histogram (above) or, if showCurve = TRUE, as a fitted kernel density curve (below). Either the mean or the mode of the distribution is displayed, depending on the parameter showMode. The Highest Density Interval (HDI) is shown as a horizontal bar, with labels for the ends of the interval.





If values for a ROPE are supplied, these are shown as dark red vertical dashed lines, together with the percentage of probability mass within the ROPE. If a comparison value (`compVal`) is supplied, this is shown as a vertical green dotted line, together with the probability mass below and above this value.

Value

Returns an object of class `histogram` invisibly. Used for its plotting side-effect.

Note

The origin of this function is the BEST package which is based on Kruschke(2015, 2013).

Author(s)

John Kruschke, modified by Mike Meredith

References

Kruschke, J. K. (2015) *Doing Bayesian data analysis, second edition: A tutorial with R, JAGS, and Stan.* Waltham, MA: Academic Press / Elsevier.

Kruschke, J. K. (2013) Bayesian estimation supersedes the t test. *Journal of Experimental Psychology: General*, **142(2)**, 573.

See Also

For details of the HDI calculation, see hdi.

Examples

```
# Generate some data
tst <- rnorm(1e5, 3, 1)
plotPost(tst)
plotPost(tst, col='wheat', border='magenta')
plotPost(tst, credMass=0.8, ROPE=c(-1,1), xlab="Response variable")
plotPost(tst, showMode=TRUE, showCurve=TRUE, compVal=5.5)

# For integers:
tst <- rpois(1e5, 12)
plotPost(tst)

# A severely bimodal distribution:
tst2 <- c(rnorm(1e5), rnorm(5e4, 7))
plotPost(tst2)                 # A valid 95% CrI, but not HDI
plotPost(tst2, showCurve=TRUE)  # Correct 95% HDI
```

---

print.bayesboot *Print the first number of draws from the Bayesian bootstrap*

Description

Print the first number of draws from the Bayesian bootstrap

Usage

```
## S3 method for class 'bayesboot'
print(x, n = 10, ...)
```

Arguments

x	The bayesboot object to print.
n	The number of draws to print.
...	Not used.

`rudirichlet` *Produce random draws from a uniform Dirichlet distribution*

Description

`rudirichlet` produces `n` draws from a `d`-dimensional uniform Dirichlet distribution. Here "uniform" implies that any combination of values on the support of the distribution is equally likely, that is, the α parameters to the Dirichlet distribution are all set to 1.0.

Usage

```
rudirichlet(n, d)
```

Arguments

`n`	the number of draws.
`d`	the dimension of the Dirichlet distribution.

Details

In the context of the Bayesian bootstrap `rudirichlet` is used to produces the bootstrap weights. Therefore, `rudirichlet` can be used if you directly want to generate Bayesian bootstrap weights.

Value

An `n` by `d` matrix.

Examples

```
set.seed(123)
rudirichlet(2, 3)
# Should produce the following matrix:
#       [,1]   [,2]   [,3]
# [1,] 0.30681 0.2097 0.4834
# [2,] 0.07811 0.1390 0.7829

# The above could be seen as a sample of two Bayesian bootstrap weights for a
# dataset of size three.
```

`summary.bayesboot` *Summarize the result of* `bayesboot`

Description

Summarizes the result of a call to `bayesboot` by calculating means, SDs, highest density intervals and quantiles of the posterior marginals.

Usage

```
## S3 method for class 'bayesboot'
summary(object, cred.mass = 0.95, ...)
```

Arguments

`object`	The bayesboot object to summarize.
`cred.mass`	The probability mass to include in the highest density intervals.
`...`	Not used.

Value

A data frame with three columns: (1) **statistic** the name of the statistic, (2) **measure** the name of the summarizing measure, and (3) **value** the value of the summarizing measure.

See Also

`hdi` in the HDInterval package for directly calculating highest density intervals from `bayesboot` objects.

Index