Package ‘ClusterBootstrap’

July 21, 2025

Title Analyze Clustered Data with Generalized Linear Models using the Cluster Bootstrap

Date 2021-02-16

Version 1.1.2

Description Provides functionality for the analysis of clustered data using the cluster bootstrap.

Depends R (>= 3.0), stats, utils, graphics, parallel, magrittr, dplyr

License GPL-3 | file LICENSE

URL `https://github.com/mathijsdeen/ClusterBootstrap`

BugReports `https://github.com/mathijsdeen/ClusterBootstrap/issues`

Maintainer Mathijs Deen <`m.l.deen@fsw.leidenuniv.nl`>

LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Author Mathijs Deen [aut, cre],
Mark de Rooij [aut]

Repository CRAN

Date/Publication 2021-02-16 12:00:02 UTC

Contents

`clusbootglm` *Fit generalized linear models with the cluster bootstrap*

Description

Fit a generalized linear model with the cluster bootstrap for analysis of clustered data.

Usage

```
clusbootglm(
  model,
  data,
  clusterid,
  family = gaussian,
  B = 5000,
  confint.level = 0.95,
  n.cores = 1
)
```

Arguments

`model`	generalized linear model to be fitted with the cluster bootstrap. This should either be a formula (or be able to be interpreted as one) or a `glm` / `lm` object. From the `(g)lm` objects, the formula will be used.
`data`	dataframe that contains the data.
`clusterid`	variable in data that identifies the clusters.
`family`	error distribution to be used in the model, e.g. `gaussian` or `binomial`.
`B`	number of bootstrap samples.
`confint.level`	level of confidence interval.
`n.cores`	number of CPU cores to be used.

Details

Some useful methods for the obtained `clusbootglm` class object are `summary.clusbootglm`, `coef.clusbootglm`, and `clusbootsample`.

Value

`clusbootglm` produces an object of class `"clusbootglm"`, containing the following relevant components:

`coefficients`	A matrix of `B` rows, containing the parameter estimates for all bootstrap samples.
`bootstrap.matrix`	n*B matrix, of which each column represents a bootstrap sample; each value in a column represents a unit of `subjectid`.
`lm.coefs`	Parameter estimates from a single (generalized) linear model.
`boot.coefs`	Mean values of the paramater estimates, derived from the bootstrap coefficients.
`boot.sds`	Standard deviations of cluster bootstrap parameter estimates.
`ci.level`	User defined confidence interval level.
`percentile.interval`	Confidence interval based on percentiles, given the user defined confidence interval level.
`parametric.interval`	Confidence interval based on `lm.coefs` and column standard deviations of `coefficients`, given the user defined confidence interval level.
`BCa.interval`	Confidence interval based on percentiles with bias correction and acceleration, given the user defined confidence interval level.
`samples.with.NA.coef`	Cluster bootstrap sample numbers with at least one coefficient being `NA`.
`failed.bootstrap.samples`	For each of the coefficients, the number of failed bootstrap samples are given.

Author(s)

Mathijs Deen, Mark de Rooij

Examples

```
## Not run:
data(opposites)
clusbootglm(SCORE~Time*COG,data=opposites,clusterid=Subject)
## End(Not run)
```

`clusbootsample` *Return data for specified bootstrap sample*

Description

Returns the full data frame for a specified bootstrap sample in a `clusbootglm` object.

Usage

```
clusbootsample(object, samplenr)
```

Arguments

`object`	object of class `clusbootglm`, created with the `clusbootglm` function.
`samplenr`	sample number for which the data frame should be returned.

Author(s)

Mark de Rooij, Mathijs Deen

Examples

```
## Not run:
data(opposites)
cbglm.1 <- clusbootglm(SCORE~Time*COG,data=opposites,clusterid=Subject)
clusbootsample(cbglm.1, samplenr=1)
## End(Not run)
```

`coef.clusbootglm` *Obtain coefficients from cluster bootstrap object*

Description

Returns the coefficients of an object of class `clusbootglm`.

Usage

```
## S3 method for class 'clusbootglm'
coef(object, estimate.type = "bootstrap", ...)
```

Arguments

`object`	object of class `clusbootglm`.
`estimate.type`	type of coefficient (`bootstrap` or `GLM`).
`...`	other arguments.

Author(s)

Mathijs Deen

Examples

```
## Not run:
data(opposites)
cbglm.1 <- clusbootglm(SCORE~Time*COG,data=opposites,clusterid=Subject)
coef(cbglm.1, estimate.type="bootstrap")
## End(Not run)
```

`confint.clusbootglm` *Confidence intervals for cluster bootstrap model parameters*

Description

Computes confidence intervals for one or more parameters in a fitted GLM with the cluster bootstrap.

Usage

```
## S3 method for class 'clusbootglm'
confint(object, parm = "all", level = 0.95, interval.type = "BCa", ...)
```

Arguments

`object`	object of class `clusbootglm`.
`parm`	a specification of which parameters are to be given confidence intervals, either a vector of numbers or a vector of names. Defaults to all parameters.
`level`	the required confidence level
`interval.type`	type of confidence level. Options are `BCa`, `percentile`, and `parametric`.
`...`	other arguments.

Author(s)

Mathijs Deen

Examples

```
## Not run:
data(opposites)
cbglm.1 <- clusbootglm(SCORE~Time*COG,data=opposites,clusterid=Subject)
confint(cbglm.1,parm=c("Time","COG"), level=.90, interval.type="percentile")
## End(Not run)
```

`emm` *Calculate estimated marginal means for a cluster bootstrap GLM*

Description

Returns the estimated marginal means of an `clusbootglm` object. This function works with a maximum of one between-subjects and one within-subjects variable.

Usage

```
emm(object, confint.level = 0.95)
```

Arguments

`object`	object of class `clusbootglm`.
`confint.level`	level of the confidence interval.

Value

`emmeans` returns an object of class `clusbootemm`, containing the following components:

`grid`	Grid with estimated marginal means for each combination of levels of the variables.
`bootstrapsample.emm`	p*B matrix, with p being the number of estimates and B being the number of bootstrap samples.

Author(s)

Mathijs Deen

Examples

```
## Not run:
medication <- medication[medication$time %% 1 == 0,]
medication$time_f <- as.factor(medication$time)
set.seed(1)
model.1 <- clusbootglm(pos~time_f*treat, clusterid = id, data = medication)
emm.1 <- emm(object = model.1)
summary(object = emm.1)
## End(Not run)
```

`medication` *Medication data*

Description

The `medication` dataframe consists of 1242 observations within 73 individuals that were part of a placebo controlled clinical trial, as reported in Tomarken, Shelton, Elkins, and Anderson (1997).

The data were retrieved from the accompanied website of Singer & Willett (2003), at https://stats.idre.ucla.edu/other/examples

Usage

```
medication
```

Format

the following variables are available:

- `id`: subject indicator
- `treat`: either placebo (0) or antidepressant (1)
- `time`: number of days since trial start.
- `pos`: positive affect. Higher scores indicate a more positive mood.

References

- Singer, J.D., & Willett, J.B. (2003). *Applied longitudinal data analysis. Modeling change and event occurence.* NY: Oxford University Press, Inc.
- Tomarken, A.J., Shelton, R.C., Elkins, L., & Anderson, T (1997). *Sleep deprivation and anti-depressant medication: Unique effects on positive and negative affect.* Poster session presented at the 9th annual meeting of the American Psychological Society, Washington, DC.

`opposites` *Opposites naming data*

Description

The `opposites` dataframe consists of 144 observations within 36 individuals that completed an inventory that assesses their performance on a timed cognitive task called "opposites naming".

The dataset does not contain the empirical data within 35 individuals from the experiment by Willett (1988), but a simulation based on the multilevel model from Singer & Willett (2003) within 36 individuals.

Usage

```
opposites
```

Format

the following variables are available:

- `Subject`: subject indicator
- `Time`: a time variable, ranging 0-3
- `COG`: cognitive skill, measured once (at time=0)
- `SCORE`: score on opposites naming task

References

- Willett, J.B. (1988). Questions and answers in the measurement of change. In: E. Rothkopf (Ed.), *Review of research in education (1988-89)* (pp. 345-422). Washington, DC: American Educational Research Association.
- Singer, J.D., & Willett, J.B. (2003). *Applied longitudinal data analysis. Modeling change and event occurence.* NY: Oxford University Press, Inc.

`plot.clusbootemm` *Plot estimated marginal means for a cluster bootstrap GLM*

Description

Plots the estimated marginal means of an `clusbootglm` object. Works with one within-subjects and/or one between-subjects variable.

Usage

```
## S3 method for class 'clusbootemm'
plot(
  x,
  within,
  between,
  pch,
  lty,
  ylab = "Estimated marginal mean",
  xlab = "Within subject",
  ...
)
```

Arguments

`x`	object of class `clusbootemm`.
`within`	within-subjects variable. Should be numeric or numerically labeled factor.
`between`	between-subjects variable.
`pch`	point character. Length must be equal to the number of between-subjects levels.
`lty`	linetype. Length must be equal to the number of between-subjects levels.
`ylab`	label for y-axis.
`xlab`	label for x-axis.
`...`	other arguments to be passed to the `plot` function (see `par`).

Author(s)

Mathijs Deen

Examples

```
## Not run:
medication <- medication[medication$time %% 1 == 0,]
medication$time_f <- as.factor(medication$time)
set.seed(1)
model.1 <- clusbootglm(pos~time_f*treat, clusterid=id, data=medication)
emm.1 <- emm(object = model.1)
plot(x = emm.1, within = time_f, between = treat, pch = c(15,17), lty = c(1,2))
## End(Not run)
```

`plot.clusbootptest` *Plot results of a permutation test*

Description

Plot results of a permutation test performed with ptest

Usage

```
## S3 method for class 'clusbootptest'
plot(x, pcol = "red", pty = 1, mfrow = c(1, 1), ...)
```

Arguments

`x`	object of class `clusbootptest`
`pcol`	color of vertical line indicating the observed Welch t test statistic
`pty`	type of vertical line indicating the observed Welch t test statistic
`mfrow`	vector of length 2 indicating the numbers of rows and columns in which the histograms will be drawn on the device.
`...`	other arguments to be passed into the `hist` function.

Author(s)

Mathijs Deen, Mark de Rooij

Examples

```
## Not run:
medication <- medication[medication$time %% 1 == 0,]
set.seed(1)
permtest.1 <- ptest(data = meds, outcome = pos, within = time, between = treat,
                    at.within = c(0,2,4,6), at.between = c(0,1), pn = 2000)
plot(permtest.1, pcol = "red", pty=2, mfrow = c(2,2), breaks="FD")
## End(Not run)
```

`ptest` *Permutation test for group differences at within-subject levels*

Description

Perform permutation tests for differences between two groups at given within-subject levels in a long-formatted dataframe

Usage

```
ptest(
  data,
  outcome,
  within,
  between,
  at.within,
  at.between,
  pn = 1000,
  progress.bar = TRUE
)
```

Arguments

`data`	dataframe that contains the data in long format.
`outcome`	outcome variable (i.e., the variable for which the difference should be tested).
`within`	within-subject variable.
`between`	between-subjects variable.
`at.within`	determine for which within-subject levels (e.g., which timepoint) the difference should be tested.
`at.between`	determine the groups in the difference test (should always be of length 2).
`pn`	the number of permutations that should be performed.
`progress.bar`	indicates whether a progress bar will be shown.

Details

In every permutation cycle, the outcome variable gets permutated and the Welch t test statistic is calculated.

Value

`ptest` produces an object of class `"clusbootptest"`, containing the following relevant components:

`perm.statistics`	A matrix of `length(at.within)` rows and `pn` columns, containing the Welch t-test statics for all permutations within the `at.within` level in the columns. The first column contains the t statistic for the observed data.
`pvalues`	Data frame containing the p values for every `at.within` level.

Author(s)

Mathijs Deen, Mark de Rooij

See Also

A useful method for the obtained `clusbootptest` class object is `plot.clusbootptest`.

Examples

```
## Not run:
medication <- medication[medication$time %% 1 == 0,]
set.seed(1)
permtest.1 <- ptest(data = meds, outcome = pos, within = time, between = treat,
                    at.within = c(0,2,4,6), at.between = c(0,1), pn = 2000)
permtest.1$pvalues
## End(Not run)
```

`summary.clusbootemm` *Summarize estimated marginal means for cluster bootstrap GLM into a grid*

Description

Returns the summary of the EMM for a `clusbootglm` class object.

Usage

```
## S3 method for class 'clusbootemm'
summary(object, ...)
```

Arguments

`object`	object of class `clusbootemm`.
`...`	other arguments.

Author(s)

Mathijs Deen

Examples

```
## Not run:
medication <- medication[medication$time %% 1 == 0,]
medication$time_f <- as.factor(medication$time)
set.seed(1)
model.1 <- clusbootglm(pos~time_f*treat, clusterid=id, data=medication)
emm.1 <- emm(object = model.1)
summary(object = emm.1)
## End(Not run)
```

`summary.clusbootglm` *Summarize output of cluster bootstrap GLM*

Description

Returns the summary of an object of class `clusbootglm`.

Usage

```
## S3 method for class 'clusbootglm'
summary(object, estimate.type = "bootstrap", interval.type = "BCa", ...)
```

Arguments

`object`	object of class `clusbootglm`.
`estimate.type`	specify which type of estimate should be returned, either bootstrap means (default) or GLM estimates from model fitted on original data.
`interval.type`	which confidence interval should be used. Options are `parametric`, `percentile`, and `BCa` intervals.
`...`	other arguments.

Author(s)

Mathijs Deen

Examples

```
## Not run:
data(opposites)
cbglm.1 <- clusbootglm(SCORE~Time*COG,data=opposites,clusterid=Subject)
summary(cbglm.1, interval.type="percentile")
## End(Not run)
```

Index