

# Package ‘bootImpute’

July 10, 2020

**Type** Package

**Title** Bootstrap Inference for Multiple Imputation

**Version** 1.1.0

**Author** Jonathan Bartlett

**Maintainer** Jonathan Bartlett <j.w.bartlett@bath.ac.uk>

**Description** Bootstraps and imputes incomplete datasets. Then performs inference on estimates obtained from analysing the imputed datasets as proposed by von Hippel and Bartlett (2019) <arXiv:1210.0870v10>.

**Depends** R (>= 2.10)

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Imports** mice, smcfcs

**Suggests** testthat

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2020-07-10 11:20:02 UTC

## R topics documented:

bootImpute . . . . .	2
bootImputeAnalyse . . . . .	3
bootMice . . . . .	4
bootSmcfcs . . . . .	5
ex_linquad . . . . .	6
<b>Index</b>	<b>7</b>

---

`bootImpute`*Bootstrap then impute an incomplete dataset*

---

### Description

Bootstraps an incomplete dataset and then imputes each bootstrap a number of times. The resulting list of bootstrapped then imputed datasets can be analysed with [bootImputeAnalyse](#).

### Usage

```
bootImpute(  
  obsdata,  
  impfun,  
  nBoot = 200,  
  nImp = 2,  
  nCores = 1,  
  seed = NULL,  
  ...  
)
```

### Arguments

<code>obsdata</code>	The data frame to be imputed.
<code>impfun</code>	A function which when passed an incomplete dataset will return a single imputed data frame.
<code>nBoot</code>	The number of bootstrap samples to take. It is recommended that you use a minimum of 200. If you specify <code>nCores&gt;1</code> , <code>nBoot</code> must be a multiple of the specified <code>nCores</code> value.
<code>nImp</code>	The number of times to impute each bootstrap sample. Two is recommended.
<code>nCores</code>	The number of CPU cores to use. If specified greater than one, <code>bootImpute</code> will impute using the number of cores specified.
<code>seed</code>	Random number seed.
<code>...</code>	Other parameters that are to be passed through to <code>impfun</code> .

### Details

The `impfun` must be a function which when passed an incomplete datasets and possibly additional arguments, returns a single imputed data frame. Depending on what imputation function returns by default, you may need to write a small wrapper function that calls the imputation procedure once and returns the imputed dataset. See the Example for an illustration with the `mice` package.

To improve computation times, `bootImpute` now supports multiple cores through the `nCores` argument which uses the `parallel` package.

### Value

A list of imputed datasets.

## Examples

```
#this example shows how you can use bootImpute to impute using the mice
#package. If you do want to impute using MICE you can instead use the
#bootMice function, which essentially contains the code below
library(mice)

#write a wrapper function to call mice with one imputation and return
#the imputed dataset
impOnce <- function(inputData) {
  oneImp <- mice::mice(inputData, m=1)
  mice::complete(oneImp)
}

#bootstrap twice and impute each twice
#in practice you should bootstrap many more times, e.g. at least 200
imps <- bootImpute(ex_linquad, impOnce, nBoot=2, nImp=2, seed=564764)
```

---

bootImputeAnalyse      *Analyse bootstrapped and imputed estimates*

---

## Description

Applies the user specified analysis function to each imputed dataset contained in `imps`, then calculates estimates, confidence intervals and p-values for each parameter, as proposed by von Hippel and Bartlett (2019).

## Usage

```
bootImputeAnalyse(imps, analysisfun, nCores = 1, quiet = FALSE, ...)
```

## Arguments

<code>imps</code>	The list of imputed datasets returned by <code>bootImpute</code>
<code>analysisfun</code>	A function which when applied to a single dataset returns the estimate of the parameter(s) of interest. The dataset to be analysed is passed to <code>analysisfun</code> as its first argument.
<code>nCores</code>	The number of CPU cores to use. If specified greater than one, <code>bootImputeAnalyse</code> will impute using the number of cores specified. The number of bootstrap samples in <code>imps</code> should be divisible by <code>nCores</code> .
<code>quiet</code>	Specify whether to print a table of estimates, confidence intervals and p-values.
<code>...</code>	Other parameters that are to be passed through to <code>analysisfun</code> .

## Details

Multiple cores can be used by using the `nCores` argument, which may be useful for reducing computation times.

**Value**

A vector containing the point estimate(s), variance estimates, and degrees of freedom.

**References**

von Hippel PT, Bartlett JW. Maximum likelihood multiple imputation: faster, more efficient imputation without posterior draws. arXiv, 2019, 1210.0870v10 <https://arxiv.org/pdf/1210.0870v10.pdf>

**Examples**

```
library(mice)

set.seed(564764)

#bootstrap twice and impute each twice
#in practice you should bootstrap many more times, e.g. at least 200
imps <- bootMice(ex_linquad, nBoot=2, nImp=2)

#analyse estimates
#write a wrapper to analyse an imputed dataset
analyseImp <- function(inputData) {
  coef(lm(y~z+x+xsq,data=inputData))
}
ests <- bootImputeAnalyse(imps, analyseImp)
```

---

bootMice

*Bootstrap then impute using mice*


---

**Description**

Bootstraps an incomplete dataset and then imputes each bootstrap a number of times using the mice package. The resulting list of bootstrapped then imputed datasets can be analysed with [bootImputeAnalyse](#). To run this function requires the mice package to be installed.

**Usage**

```
bootMice(obsdata, nBoot = 200, nImp = 2, nCores = 1, seed = NULL, ...)
```

**Arguments**

obsdata	The data frame to be imputed.
nBoot	The number of bootstrap samples to take. It is recommended that you use a minimum of 200. If you specify nCores>1, nBoot must be a multiple of the specified nCores value.
nImp	The number of times to impute each bootstrap sample. Two is recommended.
nCores	The number of CPU cores to use. If specified greater than one, bootImpute will impute using the number of cores specified.

seed	Random number seed.
...	Other arguments that are to be passed to mice.

**Value**

A list of imputed datasets.

**Examples**

```
library(mice)

head(ex_linquad)

#bootstrap 10 times and impute each twice
imps <- bootMice(ex_linquad, nBoot=10, nImp=2, seed=564764)
```

---

bootSmcfcs	<i>Bootstrap then impute using smcfcs</i>
------------	---

---

**Description**

Bootstraps an incomplete dataset and then imputes each bootstrap a number of times using the smcfcs package. The resulting list of bootstrapped then imputed datasets can be analysed with [bootImputeAnalyse](#). To run this function requires the smcfcs package to be installed.

**Usage**

```
bootSmcfcs(obsdata, nBoot = 200, nImp = 2, nCores = 1, seed = NULL, ...)
```

**Arguments**

obsdata	The data frame to be imputed.
nBoot	The number of bootstrap samples to take. It is recommended that you use a minimum of 200. If you specify nCores>1, nBoot must be a multiple of the specified nCores value.
nImp	The number of times to impute each bootstrap sample. Two is recommended.
nCores	The number of CPU cores to use. If specified greater than one, bootImpute will impute using the number of cores specified.
seed	Random number seed.
...	Other arguments that are to be passed to smcfcs.

**Value**

A list of imputed datasets.

**Examples**

```
library(smcfc)

head(ex_linquad)
#bootstrap twice and impute each twice
#in practice you should bootstrap many more times, e.g. at least 200
imps <- bootSmcfc(ex_linquad, nBoot=2, nImp=2,
                  smtype="lm", smformula="y~z+x+xsq",
                  method=c("", "", "norm", "x^2", ""), seed=564764)
```

---

ex_linquad	<i>Simulated example data with continuous outcome and quadratic covariate effects</i>
------------	---

---

**Description**

A dataset containing simulated data where the outcome depends quadratically on a partially observed covariate.

**Usage**

```
ex_linquad
```

**Format**

A data frame with 1000 rows and 5 variables:

**y** Continuous outcome

**z** Fully observed covariate, with linear effect on outcome

**x** Partially observed normally distributed covariate, with quadratic effect on outcome

**xsq** The square of x, which thus has missing values also

**v** An auxiliary variable (i.e. not contained in the substantive model)

# Index

## \* datasets

ex\_linquad, 6

bootImpute, 2, 3

bootImputeAnalyse, 2, 3, 4, 5

bootMice, 4

bootSmcfcs, 5

ex\_linquad, 6