Package ‘bootImpute’

April 1, 2021

Type Package
Title Bootstrap Inference for Multiple Imputation
Version 1.2.0
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 2021-04-01 15:30:03 UTC

R topics documented:

  bootImpute .................................................. 2
  bootImputeAnalyse ........................................... 3
  bootMice ...................................................... 4
  bootSmcfcs ................................................... 5
  ex_linquad ..................................................... 6

Index 7
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

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

Arguments

- `obsdata`: The data frame to be imputed.
- `impfun`: A function which when passed an incomplete dataset will return a list of imputed data frames.
- `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 parameters that are to be passed through to `impfun`, which will often include the argument that tells `impfun` to generate as many imputations as specified by the value passed to `nImp`.

Details

The `impfun` must be a function which when passed an incomplete datasets and possibly additional arguments, returns a list of (e.g. 2) imputed datasets. The number of imputed datasets that `impfun` returns should match the value you specify for the argument `nImp`. Depending on what your imputation function returns by default, you may need to write a small wrapper function that calls the imputation procedure and returns the list of `nImp` datasets. 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.
bootImputeAnalyse

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 generating M imputations
    impM <- function(inputData, M) {
        miceImps <- mice::mice(inputData, m=M)
       imps <- vector("list", M)
        for (i in 1:M) {
            imps[[i]] <- mice::complete(miceImps, i)
        }
        imps
    }

    #bootstrap twice and impute each twice
    #in practice you should bootstrap many more times, e.g. at least 200
    #note you have to tell bootImpute how many imputations per bootstrap in
    #nImp=2 and also pass through whatever your imp function argument is called
    #for specifying number of imputations, which here is M=2.
    imps <- bootImpute(ex_linquad, impM, nBoot=2, nImp=2, M=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

imps

The list of imputed datasets returned by bootImpute

analysisfun

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 analysisfun as its first argument.

nCores

The number of CPU cores to use. If specified greater than one, bootImputeAnalyse will impute using the number of cores specified. The number of bootstrap samples in imps should be divisible by nCores.
quiet Specify whether to print a table of estimates, confidence intervals and p-values.

Other parameters that are to be passed through to analysisfun.

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


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

Description

Bootstrap then impute using mice

Bootsraps 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, ...)
**bootSmcfcs**  

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>obsdata</td>
<td>The data frame to be imputed.</td>
</tr>
<tr>
<td>nBoot</td>
<td>The number of bootstrap samples to take. It is recommended that you use a</td>
</tr>
<tr>
<td></td>
<td>minimum of 200. If you specify nCores&gt;1, nBoot must be a multiple of the</td>
</tr>
<tr>
<td></td>
<td>specified nCores value.</td>
</tr>
<tr>
<td>nImp</td>
<td>The number of times to impute each bootstrap sample. Two is recommended.</td>
</tr>
<tr>
<td>nCores</td>
<td>The number of CPU cores to use. If specified greater than one, bootImpute</td>
</tr>
<tr>
<td></td>
<td>will impute using the number of cores specified.</td>
</tr>
<tr>
<td>seed</td>
<td>Random number seed.</td>
</tr>
<tr>
<td>...</td>
<td>Other arguments that are to be passed to mice.</td>
</tr>
</tbody>
</table>

**Value**

A list of imputed datasets.

**Examples**

```r
library(mice)
head(ex_linquad)

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

**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, ...)`
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(smcfcs)

head(ex_linquad)

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

---

ex_linquad  Simulated example data with continuous outcome and quadratic covariate effects

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