Package 'RBtest'

March 3, 2020

Type Package

Title Regression-Based Approach for Testing the Type of Missing Data

Version 1.1

Author Serguei Rouzinov and André Berchtold

Maintainer Serguei Rouzinov <rouzinovs@gmail.com>

Description The regression-based (RB) approach is a method to test the missing data mechanism. This package contains two functions that test the type of missing data (Missing Completely At Random vs Missing At Random) on the basis of the RB approach. The first function applies the RB approach independently on each variable with missing data, using the completely observed variables only. The second function tests the missing data mechanism globally (on all variables with missing data) with the use of all available information. The algorithm is adapted both to continuous and categorical data.

License GPL-3

Imports nnet, mice, psych

Encoding UTF-8

LazyData true

RoxygenNote 7.0.2

NeedsCompilation no

Repository CRAN

Date/Publication 2020-03-03 15:00:03 UTC

R topics documented:

RBtest *Test of missing data mechanism using complete data*

Description

This function tests the missing completely at random (MCAR) vs missing at random (MAR) by using the complete variables only.

Usage

```
RBtest(data)
```

Arguments

data	Dataset with at least one complete variable. The variables could be either continuous, categorical or a mix of both.

Value

A list of the following elements:

- abs.nbrMD The absolute number of missing data per variable.
- rel.nbrMD The percentage of missing data per variable.
- type Vector of the same length than the number of variables of the dataset, where '0' is for variables with MCAR data, '1' is for variables with MAR data and '-1' is for complete variables.

Author(s)

Serguei Rouzinov <rouzinovs@gmail.com> and

André Berchtold <Andre.Berchtold@unil.ch>

Maintainer: Serguei Rouzinov <rouzinovs@gmail.com>

Examples

```
set.seed(60)
n<-100 # sample size
r<-5 # number of variables
mis<-0.2 # frequency of missing data
mydata<-matrix(NA, nrow=n, ncol=r) # mydata is a matrix of r variables
# following a U(0,1) distribution
for (i in c(1:r)){
mydata[,i]<-runif(n,0,1)
}
bin.var<-sample(LETTERS[1:2],n,replace=TRUE, prob=c(0.3,0.7)) # binary variable [A,B].
# The probability of being in one of the categories is 0.3.
cat.var<-sample(LETTERS[1:3],n,replace=TRUE, prob=c(0.5,0.3,0.2)) # categorical variable [A,B,C].
```

```
num.var<-runif(n,0,1) # Additional continuous variable following a U(0,1) distribution
mydata<-cbind.data.frame(mydata,bin.var,cat.var,num.var,stringsAsFactors = TRUE)
# dataframe with r+3 variables
colnames(mydata)=c("v1","v2","X1","X2","X3","X4","X5", "X6") # names of columns
# MCAR on X1 and X4 by using v1 and v2. MAR on X3 and X5 by using X2 and X6.
mydata$X1[which(mydata$v1<=sort(mydata$v1)[mis*n])]<-NA # X1: (mis*n)% of MCAR data.
# All data above the (100-mis)th percentile in v1 are selected
# and the corresponding observations in X1 are replaced with missing data.
mydata$X3[which(mydata$X2<=sort(mydata$X2)[mis*n])]<-NA # X3: (mis*n)% of MAR data.
# All data above the (100-mis)th percentile in X2 are selected
# and the corresponding observations in X3 are replaced with missing data.
mydata$X4[which(mydata$v2<=sort(mydata$v2)[mis*n])]<-NA # X4: (mis*n)% of MCAR data.
# All data above the (100-mis)th percentile in v2 are selected
# and the corresponding observations in X4 are replaced with missing data.
mydata$X5[which(mydata$X6<=sort(mydata$X6)[mis*n])]<-NA # X5: (mis*n)% of MAR data.
# All data above the (100-mis)th percentile in X6 are selected
# and the corresponding observations in X5 are replaced with missing data.
mydata$v1=NULL
mydata$v2=NULL
RBtest(mydata)
```

`RBtest.iter` *Test of missing data mechanism using all available information*

Description

This function tests MCAR vs MAR by using both the complete and incomplete variables.

Usage

```
RBtest.iter(data,K)
```

Arguments

`data`	Dataset with at least one complete variable.
`K`	Maximum number of iterations.

Value

A list of the following elements:

- `abs.nbrMD` The absolute quantity of missing data per variable.
- `rel.nbrMD` The percentage of missing data per variable.
- `K` The maximum admitted number of iterations.
- `iter` The final number of iterations.
- `type.final` Vector of the same length than the number of variables of the dataset, where '0' is for variables with MCAR data, '1' is for variables with MAR data and '-1' is for complete variables.

- TYPE.k Dataframe containing the type of missing data after each iteration. Each row is a vector having the same length than the number of variables of the dataset, where '0' is for variables with MCAR data, '1' is for variables with MAR data and '-1' is for complete variables.

Examples

```
set.seed(60)
n<-100 # sample size
r<-5 # number of variables
mis<-0.2 # frequency of missing data
mydata<-matrix(NA, nrow=n, ncol=r) # mydata is a matrix of r variables
# following a U(0,1) distribution
for (i in c(1:r)){
mydata[,i]<-runif(n,0,1)
}
bin.var<-sample(LETTERS[1:2],n,replace=TRUE, prob=c(0.3,0.7)) # binary variable [A,B].
# The probability of being in one of the categories is 0.3.
cat.var<-sample(LETTERS[1:3],n,replace=TRUE, prob=c(0.5,0.3,0.2)) # categorical variable [A,B,C].
# The vector of probabilities of occurence A, B and C is (0.5,0.3,0.7).
num.var<-runif(n,0,1) # Additional continuous variable following a U(0,1) distribution
mydata<-cbind.data.frame(mydata,bin.var,cat.var,num.var,stringsAsFactors = TRUE)
# dataframe with r+3 variables
colnames(mydata)=c("v1","v2","X1","X2","X3","X4","X5", "X6") # names of columns
# MCAR on X1 and X4 by using v1 and v2. MAR on X3 and X5 by using X2 and X6.
mydata$X1[which(mydata$v1<=sort(mydata$v1)[mis*n])]<-NA # X1: (mis*n)% of MCAR data.
# All data above the (100-mis)th percentile in v1 are selected
# and the corresponding observations in X1 are replaced with missing data.
mydata$X3[which(mydata$X2<=sort(mydata$X2)[mis*n])]<-NA # X3: (mis*n)% of MAR data.
# All data above the (100-mis)th percentile in X2 are selected
# and the corresponding observations in X3 are replaced with missing data.
mydata$X4[which(mydata$v2<=sort(mydata$v2)[mis*n])]<-NA # X4: (mis*n)% of MCAR data.
# All data above the (100-mis)th percentile in v2 are selected
# and the corresponding observations in X4 are replaced with missing data.
mydata$X5[which(mydata$X6<=sort(mydata$X6)[mis*n])]<-NA # X5: (mis*n)% of MAR data.
# All data above the (100-mis)th percentile in X6 are selected
# and the corresponding observations in X5 are replaced with missing data.
mydata$v1=NULL
mydata$v2=NULL

RBtest.iter(mydata,5)
```

Index