Package ‘MHDA’

October 9, 2024

Type Package

Title Massive Hierarchically Data Analysis

Version 1.3

Date 2024-10-09

Maintainer Yarong Yang <Yi.YA_yaya@hotmail.com>

Depends methods, stats, graphics

Description Three main functions about analyzing massive data (missing observations are allowed) considered from multiple layers of categories.

Collate MHDA.R MHDA.plot.R Steps.analysis.R AllClasses.R

License GPL-2

NeedsCompilation no

Author Yarong Yang [aut, cre],
Jacob Zhang [aut]

Repository CRAN

Date/Publication 2024-10-08 22:00:07 UTC

Contents

MHDA-package 2
MHDA 3
MHDA.plot 4
Res.mhda.1 6
Res.mhda.2 6
Steps.analysis 7

Index **9**

MHDA-package *Massive Hierarchically Data Analysis*

Description

Three main functions about analyzing massive data (missing observations are allowed) considered from multiple layers of categories.

Details

Package:	MHDA
Type:	Package
Version:	1.3
Date:	2024-10-09
License:	GPL-2

Author(s)

Yarong Yang and Jacob Zhang

Examples

```
##generating a small data for example###
Slots<-c("2021-01","2021-02")
Units<-c("Store-1","Store-2","Store-3","Store-4")
Class.I<-c("Mall_1","Mall_2","Mall_3","Mall_a","Mall_b","Mall_c")
Class.II<-c("B&H","F&B","HOM","KID","LEI&ENT","RET-SHO-ACC","SPM&SER")
Infor.1<-c("Mall_2","HOM")
Infor.2<-c("Mall_c","B&H")
Infor.3<-c("Mall_2","KID")
Infor.4<-c("Mall_c","F&B")
Store.sales<-list()
Store.sales[[1]]<-Store.sales[[2]]<-list()
names(Store.sales)<-Slots
for(i in 1:2) {
    for(j in 1:4) {
        Store.sales[[i]][[j]]<-list()
        n<-sample(1:30,1)
        for(k in 1:n) {
            t<-Store.sales[[i]][[j]][[k]]<-abs(rnorm(sample(1:50,1),0,1))
            names(Store.sales[[i]][[j]][[k]])<-sample(c(0,1),length(t),replace=TRUE)
        }
        names(Store.sales[[i]][[j]])<-paste("s",1:n,sep="")
    }
    Store.sales[[i]][[4+1]]<-c(Infor.1[1],Infor.2[1],Infor.3[1],Infor.4[1])
```

```
    Store.sales[[i]][[4+2]]<-c(Infor.1[2],Infor.2[2],Infor.3[2],Infor.4[2])
    names(Store.sales[[i]])<-c(Units,"Level.I","Level.II")
}

Res<-MHDA(Data=Store.sales,data.infor=NULL,type="Value",is.binary=TRUE,Unit=NULL,
Category.I="Mall_c",Category.II=Class.II,Slot=c("2021-01","2021-02"))
```

MHDA *Massive Hierarchically Data Analysis*

Description

This function conducts Massive Hierarchically Data Analysis.

Usage

```
MHDA(Data,data.infor,type,is.binary,Unit,Category.I,Category.II,Slot)
```

Arguments

`Data`	List. Each element of the list keeps data observations in one slot. Each unit in each slot has a series of data cells with each data cell keeping a vector of observations. When this argument is not NULL, argument data.infor is ignored.
`data.infor`	Character String. When argument Data is NULL, a .rdata file name is assigned to this argument. The content of the .rdata file is a data.frame with three columns. The first column is the name vector of units. The second column shows the levels of the units in Category I. The third column shows the levels of the units in Category II.
`type`	Character. "Value" for the value number of the observations. "Count" for counting the number of observations.
`is.binary`	Logical. TRUE for binary "positive" observations. FALSE for all the observations. Binary identification is labeled as the name of an observation.
`Unit`	Character String. ID of a unit in the first column of the data information matrix. When Unit is not NULL, Category.I and Category.II are ignored.
`Category.I`	Character Strings. Partial or full levels of the categories in the second column of the data informaion matrix. NULL to ignore this argument.
`Category.II`	Character Strings. Partial or full levels of the categories in the third column of the data information matrix. NULL to ignore this argument.
`Slot`	Character Strings. Names of the folders with each folder keeping data under a specific slot. For example, "2021-01" means that the folder "2021-01" keeps data observations in slot "2021-01". Every unit has a .rdata data file in every slot. In each of these .rdata file, there are a list of data cells. In each data cell, there is a series of observations.

Value

A list.

Examples

```
##generating a small data for example###
Slots<-c("2021-01","2021-02")
Units<-c("Store-1","Store-2","Store-3","Store-4")
Class.I<-c("Mall_1","Mall_2","Mall_3","Mall_a","Mall_b","Mall_c")
Class.II<-c("B&H","F&B","HOM","KID","LEI&ENT","RET-SHO-ACC","SPM&SER")
Infor.1<-c("Mall_2","HOM")
Infor.2<-c("Mall_c","B&H")
Infor.3<-c("Mall_2","KID")
Infor.4<-c("Mall_c","F&B")
Store.sales<-list()
Store.sales[[1]]<-Store.sales[[2]]<-list()
names(Store.sales)<-Slots
for(i in 1:2) {
    for(j in 1:4) {
        Store.sales[[i]][[j]]<-list()
        n<-sample(1:30,1)
        for(k in 1:n) {
            t<-Store.sales[[i]][[j]][[k]]<-abs(rnorm(sample(1:50,1),0,1))
            names(Store.sales[[i]][[j]][[k]])<-sample(c(0,1),length(t),replace=TRUE)
        }
        names(Store.sales[[i]][[j]])<-paste("s",1:n,sep="")
    }
    Store.sales[[i]][[4+1]]<-c(Infor.1[1],Infor.2[1],Infor.3[1],Infor.4[1])
    Store.sales[[i]][[4+2]]<-c(Infor.1[2],Infor.2[2],Infor.3[2],Infor.4[2])
    names(Store.sales[[i]])<-c(Units,"Level.I","Level.II")
}

Res<-MHDA(Data=Store.sales,data.infor=NULL,type="Value",is.binary=TRUE,
Unit="Store-1",Category.I="Mall_2",Category.II=Class.II,Slot="2021-01")
```

`MHDA.plot` *Plotting Results from Massive Hierarchically Data Analysis*

Description

This function plots results from Massive Hierarchically Data Analysis.

Usage

```
MHDA.plot(data,plot.type,Class,ID,Category.I,Category.II,Slot)
```

Arguments

`data`	List. Result object from MHDA function.
`plot.type`	Character. "line" for line plot for Unit. "pie" for pie plot for Category I and Category II.
`Class`	Character. "Unit", "Category.I", or "Category.II".
`ID`	Character. A level of the category of argument Class.
`Category.I`	Character Strings. Partial or full levels of Category I according to the object assigned to arugument "data".
`Category.II`	Character Strings. Partial or full levels of Category II according to the object assigned to argument "data".
`Slot`	Character Strings. Names of slots. Line plot only shows results for the first single slot.

Examples

```
##generating a small data for example###
Slots<-c("2021-01","2021-02")
Units<-c("Store-1","Store-2","Store-3","Store-4")
Class.I<-c("Mall_1","Mall_2","Mall_3","Mall_a","Mall_b","Mall_c")
Class.II<-c("B&H","F&B","HOM","KID","LEI&ENT","RET-SHO-ACC","SPM&SER")
Infor.1<-c("Mall_2","HOM")
Infor.2<-c("Mall_c","B&H")
Infor.3<-c("Mall_2","KID")
Infor.4<-c("Mall_c","F&B")
Store.sales<-list()
Store.sales[[1]]<-Store.sales[[2]]<-list()
names(Store.sales)<-Slots
for(i in 1:2) {
    for(j in 1:4) {
        Store.sales[[i]][[j]]<-list()
        n<-sample(1:30,1)
        for(k in 1:n) {
            t<-Store.sales[[i]][[j]][[k]]<-abs(rnorm(sample(1:50,1),0,1))
            names(Store.sales[[i]][[j]][[k]])<-sample(c(0,1),length(t),replace=TRUE)
        }
        names(Store.sales[[i]][[j]])<-paste("s",1:n,sep="")
    }
    Store.sales[[i]][[4+1]]<-c(Infor.1[1],Infor.2[1],Infor.3[1],Infor.4[1])
    Store.sales[[i]][[4+2]]<-c(Infor.1[2],Infor.2[2],Infor.3[2],Infor.4[2])
    names(Store.sales[[i]])<-c(Units,"Level.I","Level.II")
}

Res<-MHDA(Data=Store.sales,data.infor=NULL,type="Value",is.binary=TRUE,
Unit="Store-1",Category.I="Mall_c",Category.II=Class.II,Slot=c("2021-01","2021-02"))

MHDA.plot(data=Res,plot.type="line",Class="Unit",ID="Store-1",Category.I=Class.I,
Category.II=Class.II,Slot="2021-01")

Res.2<-MHDA(Data=Store.sales,data.infor=NULL,type="Count",is.binary=FALSE,Unit=NULL,
```

```
Category.I="Mall_c",Category.II=Class.II,Slot=c("2021-01","2021-02"))

MHDA.plot(data=Res.2,plot.type="pie",Class="Category.I",ID="Mall_c",
Category.I=Class.I,Category.II=Class.II,Slot="2021-02")
```

`Res.mhda.1` *Class to contain the MHDA results for a single unit.*

Description

The function MHDA returns object of class Res.mhda.1 when the Unit argument is not NULL.

Objects from the Class

new("Res.mhda.1",Obj.a.unit=new("list"),type=new("character"),is.binary=new("character"))

Slots

`Obj.a.unit`**:** A list. Each element of the list is a marix, corresponding to one slot. Each row of the matrix corresponds to one data cell. The first element of the row is sum of the observations in the data cell. The second element of the row is the number of the observations in the data cell.

`is.binary`**:** Logical. TRUE for binary "positive" observations. FALSE for all observations.

Examples

```
showClass("Res.mhda.1")
```

`Res.mhda.2` *Class to contain the MHDA results for Category.I and Category.II.*

Description

The function MHDA returns object of class Res.mhda.2 when the Unit argument is NULL.

Objects from the Class

new("Res.mhda.2",Obj.all.units=new("matrix"),Obj.category=new("list"),type=new("character"),is.binary=new("character")

Slots

`Obj.all.units`: Matrix. Each row of the matrix corresponds to one unit. Odd elements of the row are sum of the observations in each slot. Even elements of the row are the number of observations in each slot. in one slot for one unit. Each row corresponds to one unit. Each column corresponds to one slot.

`Obj.category`: List. Each element of the list is a matrix corresponding to one slot. The (i,j) element of a matrix is sum of the observations in the i-th level of Category.I and the j-th level of Category.II. When Category.I or Category.II is NULL in the arguments, it is meant that that category is ignored to be whole levels together.

`type`: Character. "Value" for the value number of the observations. "Count" for counting the number of observations.

`is.binary`: Logical. TRUE for binary "positive" observations. FALSE for all observations.

Examples

```
showClass("Res.mhda.2")
```

Steps.analysis — *Stepwise analysis of a series of numeric observations on multiple levels of rhythms*

Description

It's a function to conduct stepwise analysis on a series of numeric observations, specifically stepwisely computing the mean of observations on each rhythm. Missing observations are allowed. Neither the number of rhythms nor the number of steps on any rhythm is of limit.

Usage

```
Steps.analysis(ID, Tag, S, Rhythms, Start, plot, pick.plot)
```

Arguments

`ID`	Character String. Label for the data.
`Tag`	Character String. Label for the data.
`S`	Numeric. A series of numeric numbers. Missing observations are allowed in the series.
`Rhythms`	Integer vector. Each element of the vector is the number of steps on the corresponding rhythm. For example, Rhythms=c(7,5,3,2,4) means there are five rhythms with the number of steps on each being 7,5,3,2,4, respectively.
`Start`	Character String. Description for the initial number with format "a+b", where a and b are integers. For example, Start="2+3" means that the initial observation in S is the third step in the second rhythm.
`plot`	Logical. TRUE for stepwise plotting of the whole periodic rhythms. FALSE for not plotting.

`pick.plot` Integer vector. Each element of the vector is the index of a rhythms to be plotted isolatedly. For example, pick.plot=1 is meant to be plotting only the first rhythm and pick.plot=c(4,2) is meant to be plotting the fourth rhythm and then plotting the second rhythm.

Value

A list. The first element of the list is the series of the mean values along the whole periodic rhythms. The second element of the list shows the total mean on each rhythm.

Author(s)

Yarong Yang

Examples

```
data1<-abs(rnorm(150,0,1))
data2<-rep(NA,150)
data<-sample(c(data1,data2),300,replace=FALSE)
T<-paste("Rhythm",1:5,"=",c(7,5,3,2,4),sep="")
tag<-NULL
for(i in 1:length(T)) tag<-paste(tag,T[i])
Res<-Steps.analysis(ID="300 Abs Normal with Missing",Tag=tag,S=data,Rhythms=c(7,5,3,2,4),
Start="3+1",plot=TRUE,pick.plot=NULL)
Res<-Steps.analysis(ID="300 Abs Normal with Missing",Tag=tag,S=data,Rhythms=c(7,5,3,2,4),
Start="3+1",plot=FALSE,pick.plot=c(3,5,2))
```

Index

∗ **classes**
    Res.mhda.1, 6
    Res.mhda.2, 6
∗ **package**
    MHDA-package, 2

MHDA, 3
MHDA-package, 2
MHDA.plot, 4

Res.mhda.1, 6
Res.mhda.1-class *(*Res.mhda.1*)*, 6
Res.mhda.2, 6
Res.mhda.2-class *(*Res.mhda.2*)*, 6

Steps.analysis, 7