

Package ‘tableSMY’

June 17, 2019

Type Package

Title A Simple Toolbox that Allows Quick Visualization of your Matrix or Dataframe

Version 1.0

Date 2019-03-26

Author Peter I-Fan Wu

Maintainer Peter I-Fan Wu <peterwu19881230@gmail.com>

Description This tools contain 5 functions that allow users to quickly visualize their matrix/dataframe/datatable and remove incomplete cells.

License GPL

Imports grid, pheatmap

R topics documented:

tableSMY-package	1
anyIncomplete	2
changeNames	3
checkDuplicates_vect	4
filterTable	5
graphTable	6
Index	8

tableSMY-package	<i>A Simple Toolbox that Allows Quick Visualization of your Matrix or Dataframe</i>
------------------	---

Description

This tools contain 5 functions that allow users to quickly visualize their matrix/dataframe/datatable and remove incomplete cells.

Details

The DESCRIPTION file: This package was not yet installed at build time.

Index: This package was not yet installed at build time.

Author(s)

Peter I-Fan Wu

Maintainer: Peter I-Fan Wu <peterwu19881230@gmail.com>

Examples

```
mat=matrix(c(1,2,3,4,5,6),ncol=2)
graphTable(mat)

set.seed(101)
random.matrix=matrix(runif(500, min = -1, max = 1), nrow = 50)
graphTable(random.matrix)

set.seed(101)
random.matrix[sample(1:50,10),sample(1:10,2)]=NA
graphTable(random.matrix)

anyIncomplete(random.matrix)

filtered_random.matrix=filterTable(random.matrix)
str(filtered_random.matrix)

checkDuplicates_vect(c(1,1,2,3,4,4,4,5,6,7,8,9,10))

Table=matrix(rnorm(2*3),ncol=2,nrow=3)
rownames(Table)=c("one", "two", "three")
colnames(Table)=c("col_one", "col_two")
Table

rowNameForTable=matrix(c("two", "one", "three", "TWO", "ONE", "THREE"),ncol=2,byrow=FALSE)
colNameForTable=matrix(c("col_two", "col_one", "COL_TWO", "COL_ONE"),ncol=2,byrow=FALSE)

#newTable=changeNames(rowOrCol="test",Table,nameForTable) #test the error message of the function
newTable=changeNames(rowOrCol="row",Table,rowNameForTable) #test rownames
newTable=changeNames(rowOrCol="col",Table,colNameForTable) #test colnames
```

anyIncomplete

Check Incompletion

Description

Check if your matrix/dataframe/datatable has any Incompletion (NA, NULL, NaN, "") and return the statistics of them

Usage

```
anyIncomplete(table)
```

Arguments

table A matrix/dataframe/datatable

Author(s)

Peter I-Fan Wu

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (table)
{
  if (!(class(table) %in% c("matrix", "data.frame", "data.table")))
    stop("Input is not a matrix, data frame or data table")
  out = list()
  out$dimension = paste("Dimension: ", dim(table)[1], " rows * ",
    dim(table)[2], " columns", sep = "")
  na = apply(table, 2, FUN = function(column) {
    any = sum(is.na(column))
    return(any)
  })
  out$na = na[na >= 1]
  null = apply(table, 2, FUN = function(column) {
    any = sum(is.null(column))
    return(any)
  })
  out$null = null[null >= 1]
  nan = apply(table, 2, FUN = function(column) {
    any = sum(is.nan(column))
    return(any)
  })
  out$nan = nan[nan >= 1]
  empty = apply(table, 2, FUN = function(column) {
    any = sum(column == "", na.rm = T)
    return(any)
  })
  out$empty = empty[empty >= 1]
  total = sum(c(na, null, nan, empty))
  out$completeness = paste(total, " of NA, NAN, NULL, or empty character is found from ",
    dim(table)[1] * dim(table)[2], " data points. They constitute ",
    total/(dim(table)[1] * dim(table)[2]) * 100, "%", sep = "")
  return(out)
}
```

changeNames

*Change row/col names***Description**

Change rownames or colnames of a matrix/dataframe/datatable based on another matrix/dataframe/datatable

Usage

```
changeNames(rowOrCol, Table, nameForTable)
```

Arguments

rowOrCol	Whether it's the row or col names that need to be changed
Table	A matrix/dataframe/datatable
nameForTable	A 2 column matrix/dataframe/datatable: 1st column/row: original names; 2nd column/row: new names

Author(s)

Peter I-Fan Wu

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (rowOrCol, Table, nameForTable)
{
  if (rowOrCol == "row") {
    IndexForNewName = match(rownames(Table), nameForTable[,
      1])
    rownames(Table) = nameForTable[, 2][IndexForNewName]
  }
  else if (rowOrCol == "col") {
    IndexForNewName = match(colnames(Table), nameForTable[,
      1])
    colnames(Table) = nameForTable[, 2][IndexForNewName]
  }
  else {
    stop("Enter either \"row\" or \"col\"")
  }
  return(Table)
}
```

checkDuplicates_vect *Check items that occur more than once*

Description

Check if a vector has any items that occur more than once and return a frequency table.

Usage

```
checkDuplicates_vect(vect)
```

Arguments

vect	Any types of vector in R
------	--------------------------

Author(s)

Peter I-Fan Wu

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (vect)
{
  if (sum(duplicated(vect)) >= 1) {
    print("Some duplicates are found:")
    table(vect)
  }
  else {
    return("Everything in this vector is unique")
  }
}
```

filterTable

*Check and remove incompleteness of your matrix/dataframe/datatable***Description**

Check for rows and columns that contain any incompleteness (NA, NULL, NaN, "") in your matrix/dataframe/datatable and removes them.

Usage

```
filterTable(table)
```

Arguments

table A matrix/dataframe/datatable

Author(s)

Peter I-Fan Wu

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (table)
{
  if (!(class(table) %in% c("matrix", "data.frame", "data.table")))
    stop("Input is not a matrix, data frame or data table")
  colToBeRemoved = apply(table, 2, FUN = function(col) {
    ifelse(sum(is.na(col) + is.nan(col) + is.null(col) +
      sapply(col, FUN = function(point) {
        identical(as.character(point), "")
      }) >= 1, T, F)
  })
```

```

    })
    new_table = table[, !colToBeRemoved]
    return(new_table)
  }

```

graphTable

Draw a heat map of your matrix/dataframe/datatable

Description

Draw a heat map that gives intuitive overview of your matrix/dataframe/datatable.

Usage

```
graphTable(table)
```

Arguments

table A matrix/dataframe/datatable

Author(s)

Peter I-Fan Wu

Examples

```

##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function (table)
{
  if (!(class(table) %in% c("matrix", "data.frame", "data.table")))
    stop("Input is not a matrix, data frame or data table")
  if (class(table) == "matrix" && typeof(table) %in% c("integer",
    "single", "double")) {
    table[] = sapply(table, FUN = function(point) as.numeric(point))
    print("From all data in the table")
    print(summary(as.numeric(table)))
  }
  else table[] = sapply(table, FUN = function(point) as.numeric(point))
  colnames(table) = NULL
  rownames(table) = NULL
  listOfPackages = c("grid", "pheatmap")
  new_pack = listOfPackages[!(listOfPackages %in% installed.packages()[,
    "Package"])]
  if (length(new_pack)) {
    print(paste(new_pack, " is required and being installed...",
      sep = ""))
    install.packages(new_pack)
  }

  q = quantile(table, na.rm = T)

```

```
q1 = q["25%"]
med = q["50%"]
q3 = q["75%"]
iqr = q3 - q1
lower = q1 - 3 * iqr
upper = q3 + 3 * iqr
palette.breaks = c(0:50 * (med - lower)/50 + lower, 1:50 *
  (upper - med)/50 + med)
pheatmap::pheatmap(table, cluster_rows = F, cluster_cols = F, breaks = palette.breaks,
  show_rownames = F, show_colnames = F)
}
```

Index

*Topic **package**

tableSMY-package, [1](#)

anyIncomplete, [2](#)

changeNames, [3](#)

checkDuplicates_vect, [4](#)

filterTable, [5](#)

graphTable, [6](#)

tableSMY (tableSMY-package), [1](#)

tableSMY-package, [1](#)