Package ‘MergeGUI’

February 19, 2015

Type Package

Title A GUI for Merging Datasets in R

Version 0.2-1

Date 2014-01-27

Author Xiaoyue Cheng, Dianne Cook, Heike Hofmann

Maintainer Xiaoyue Cheng <xycheng@iastate.edu>

Description A GUI for merging datasets in R using gWidgets.

Depends gWidgetsRGtk2, ggplot2

Imports cairoDevice, rpart

License GPL (>= 2.0)

Collate 'mergeGUI.R' 'zzz.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2014-01-27 22:44:16

R topics documented:

`intersect2` *Obtain the intersection of a list of vectors. Function "intersect" in the base package can only intersect two vectors. The function "intersect2" is designed to obtain the intersection and the difference for more than two vectors. The input should be a list whose elements are the vectors, and the outputs include the intersection of all vectors and a list whose elements are the input vectors substracting the intersection. Besides, intersect2 allows the labels of the vectors. If a list of labels is given in the input, then the outputs will also include a matrix of labels which match the intersection for the vectors, and a list of labels which match the left part of the vectors.*

Description

Obtain the intersection of a list of vectors. Function "intersect" in the base package can only intersect two vectors. The function "intersect2" is designed to obtain the intersection and the difference for more than two vectors. The input should be a list whose elements are the vectors, and the outputs include the intersection of all vectors and a list whose elements are the input vectors substracting the intersection. Besides, intersect2 allows the labels of the vectors. If a list of labels is given in the input, then the outputs will also include a matrix of labels which match the intersection for the vectors, and a list of labels which match the left part of the vectors.

Usage

```
intersect2(vname, simplifiedname = vname)
```

Arguments

`vname` A list of labels.

`simplifiedname` A list of vectors to make the intersection. Each element in the list has the same length as the corresponding element in vname. Default to be vname. If simplifiedname is not vname, then it works as the real vectors to match, and vname is like the labels of simplifiedname. If simplifiedname is the same as vname, then the returned value simpleuniq=uniq.

Value

The outputs are 'public', 'individual', 'uniq', and 'simpleuniq'. 'public' is a vector of the intersection of 'simplifiedname'. 'individual' is a matrix with the original colnames matched to 'public' in all files. 'simpleuniq' is a list of the left part of 'simplifiedname' if we pick 'public' out. 'uniq' is a list of the left part of 'vname' if we pick 'individual' out.

Author(s)

Xiaoyue Cheng <<xycheng@iastate.edu>>

Examples

```
a = list(x1 = c("label11", "label12"), x2 = c("label21", "label22", "label23"),
    x3 = c("label31", "label32"))
b = list(x1 = c(1, 2), x2 = c(3, 1, 2), x3 = c(2, 1))
intersect2(a, b)
```

`MergeGUI` *The Merging GUI. This function will start with an starting interface, allowing 1) selecting several data files; 2) doing the next command with more than one files. There are two commands which could be selected: match the variables, match the cases by the key variable. In the matching-variable interface the user can 1) check the matching of the variables among files and switch the variable names if they are wrongly matched; 2) look at the numerical and graphical summaries for the selected variables, or the dictionary for selected factor varibles; 3) observe the misclassification rate, KS-test p-values and Chi-square test p-values for each variable, which helps to determine whether any transformation is needed for the variable; (For each variable, the user may want to know whether it could distinguish the sources correctly. So the misclassification rate is calculated through the tree model. KS-test is used to check whether any variable has different distributions for different sources. And the Chi-square test is useful when the user is interested in the pattern of missing values among the sources.) 4) change the name or class for any variable; 5) export the merged dataset and the summary for it. In the matching-case interface the user can determine a primary key for each data file and then merge the cases by the key.*

Description

The merging GUI consists of four tabs. In the preferences tab, user can choose whether the numerical p-values or the flag symbols are displayed in the summary tab; whether the y-scales are free for different data files when drawing the plots faceted by the sources. In the checking tab, each data file has a list of variable names, and the GUI will automatically arrange the order of variable names to align the same names in one row. The user can switch the order of the variables in one file's list. It is possible to undo, redo, or reset the matching. In the summary tab, there is a list of variable names on the left which corresponds to the checking tab. The misclassification rate, KS-test p-values and Chi-square test p-values for each variable may also be presented with the variable names. On the top right there are three buttons: Numeric summary, Graphical summary, and Dictionary. And the results could be shown below the buttons. For the graphical summary, histogram or barchart will be shown if a single variable is selected. A scatterplot will be drawn if two numeric or two factor varaibles are chosen. Side-by-side boxplots will be presented when one numeric and one factor varaibles are selected. A parallel coordinate plot is shown when all the variables selected are numeric and there are more than two variables. If more than two variables are chosen but the classes of the variables are mixed, i.e. some are numeric, some are factor or character, then histograms and barcharts will be drawn individually. All the plots are facetted by the source. In the export tab the user could select all or none variables by click the buttons or choose several varaibles by

Ctrl+Click. Then the export button will export the merged data and the numeric summaries of the selected variables into two csv files.

Usage

```
MergeGUI(..., filenames = NULL, unit = TRUE, distn = TRUE, miss = TRUE)
```

Arguments

`...`	names of the data frames to read
`filenames`	A vector of csv file names with their full paths.
`unit`	whether the test of the difference among the group centers is on or off
`distn`	whether the test of the difference among the group distributions is on or off
`miss`	whether the test of the difference among the group missing patterns is on or off

Value

NULL

Author(s)

Xiaoyue Cheng <<`xycheng@iastate.edu`>>

Examples

```
if (interactive()) {
    MergeGUI()

    csvnames = list.files(system.file("doc", package = "MergeGUI"),
        pattern = "\\.csv$")
    files = system.file("doc", csvnames, package = "MergeGUI")
    MergeGUI(filenames = files)

    data(iris)
    setosa = iris[iris$Species == "setosa", 1:4]
    versicolor = iris[iris$Species == "versicolor", 1:4]
    virginica = iris[iris$Species == "virginica", 1:4]
    MergeGUI(setosa, versicolor, virginica)
}
```

scale_kstest

Compute the p-values of the Kolmogorov-Smirnov tests between different sources for each variable. This function is used to detect whether the matched variables from different files have different distributions. For each variable, it will compute the pairwise KS-test p-values among the sources, then report the lowest p-value as the indice for this variable.

Description

Compute the p-values of the Kolmogorov-Smirnov tests between different sources for each variable. This function is used to detect whether the matched variables from different files have different distributions. For each variable, it will compute the pairwise KS-test p-values among the sources, then report the lowest p-value as the indice for this variable.

Usage

```
scale_kstest(nametable.class, dataset.class, name.class, varclass = NULL)
```

Arguments

`nametable.class`
: A matrix of the matched variable names. The number of columns is equal to the number of files. Each row represents a variable that is going to be merged. Any elements except NA in nametable.class must be the variable names in dataset.class.

`dataset.class`
: The dataset list. The length of the list is equal to the number of files, and the order of the list is the same as the order of columns in nametable.class.

`name.class`
: A character vector of variable names. The length of the vector must be equal to the number of rows in nametable.class. Since the variable names in nametable.class may not be consistent, name.class is needed to name the variables.

`varclass`
: A character vector of variable classes. The length of the vector must be equal to the number of rows in nametable.class. All the classes should be in "numeric", "integer", "factor", and "character". Default to be null, then it will be determined by `var.class`.

Value

A vector of p-values from the KS-test for each variable.The p-values are between 0 and 1, or equal to 9 if one of more groups only have NA's.

Author(s)

Xiaoyue Cheng <`<xycheng@iastate.edu>`>

Examples

```
a = data.frame(aa = 1:5, ab = LETTERS[6:2], ac = as.logical(c(0, 1, 0, NA, 0)))
b = data.frame(b1 = letters[12:14], b2 = 3:1)
dat = list(a, b)
name = matrix(c("ab", "aa", "ac", "b1", "b2", NA), ncol = 2)
colnames(name) = c("a", "b")
newname = c("letter", "int", "logic")
scale_kstest(name, dat, newname)
```

`scale_missing` *Chi-square tests for the counts of missing and non-missing. This function is used to detect whether the matched variables from different files have different missing patterns. For each variable, it will firstly count the missing and non-missing values among the sources, and then form a contingency table. The p-value of Chi-square test is computed from the contingency table and finally reported for the variable.*

Description

Chi-square tests for the counts of missing and non-missing. This function is used to detect whether the matched variables from different files have different missing patterns. For each variable, it will firstly count the missing and non-missing values among the sources, and then form a contingency table. The p-value of Chi-square test is computed from the contingency table and finally reported for the variable.

Usage

```
scale_missing(nametable.class, dataset.class, name.class)
```

Arguments

`nametable.class`
A matrix of the matched variable names. The number of columns is equal to the number of files. Each row represents a variable that is going to be merged. Any elements except NA in nametable.class must be the variable names in dataset.class.

`dataset.class` The dataset list. The length of the list is equal to the number of files, and the order of the list is the same as the order of columns in nametable.class.

`name.class` A character vector of variable names. The length of the vector must be equal to the number of rows in nametable.class. Since the variable names in nametable.class may not be consistent, name.class is needed to name the variables.

Value

A vector of p-values from the Chisquare-test for the missings of each variable. The p-values are between 0 and 1.

Author(s)

Xiaoyue Cheng <`<xycheng@iastate.edu>`>

Examples

```
a = data.frame(aa = 1:5, ab = LETTERS[6:2], ac = as.logical(c(0, 1, 0, NA, 0)))
b = data.frame(b1 = letters[12:14], b2 = 3:1)
dat = list(a, b)
```

```
name = matrix(c("ab", "aa", "ac", "b1", "b2", NA), ncol = 2)
colnames(name) = c("a", "b")
newname = c("letter", "int", "logic")
scale_missing(name, dat, newname)
```

---

scale_rpart

Compute the misclassification rate for each variable. When merging data from several datasets, it is meaningful to detect whether the matched variables from different files have different centers. The function computes the misclassification rate variable by variable using classification tree (the rpart package). It will firstly merge the dataset by the given nametable.class, then use rpart for each variable to seperate the data without any covariates and compute the misclassification rate.

Description

Compute the misclassification rate for each variable. When merging data from several datasets, it is meaningful to detect whether the matched variables from different files have different centers. The function computes the misclassification rate variable by variable using classification tree (the rpart package). It will firstly merge the dataset by the given nametable.class, then use rpart for each variable to seperate the data without any covariates and compute the misclassification rate.

Usage

```
scale_rpart(nametable.class, dataset.class, name.class, varclass = NULL)
```

Arguments

`nametable.class`
A matrix of the matched variable names. The number of columns is equal to the number of files. The column names are required. Each row represents a variable that is going to be merged. Any elements except NA in nametable.class must be the variable names in dataset.class.

`dataset.class`
The dataset list. The length of the list is equal to the number of files, and the order of the list is the same as the order of columns in nametable.class.

`name.class`
A character vector of variable names. The length of the vector must be equal to the number of rows in nametable.class. Since the variable names in nametable.class may not be consistent, name.class is needed to name the variables.

`varclass`
A character vector of variable classes. The length of the vector must be equal to the number of rows in nametable.class. All the classes should be in "numeric", "integer", "factor", and "character". Default to be null, then it will be determined by `var.class`.

Value

A vector of the misclassification rate. The rate is between 0 and 1, or equal to 9 if one of more groups only have NA's.

Author(s)

Xiaoyue Cheng <<xycheng@iastate.edu>>

Examples

```
a = data.frame(aa = 1:5, ab = LETTERS[6:2], ac = as.logical(c(0, 1, 0, NA, 0)))
b = data.frame(b1 = letters[12:14], b2 = 3:1)
dat = list(a, b)
name = matrix(c("ab", "aa", "ac", "b1", "b2", NA), ncol = 2)
colnames(name) = c("a", "b")
newname = c("letter", "int", "logic")
scale_rpart(name, dat, newname)
```

---

simplifynames

Short the names from a template. The merging GUI is designed to merge data from different files. But sometimes the file names are too long to be displayed in the GUI. Hence this function is used to short the basenames by removing the same beginning letters of each name. Hence the output is a character vector whose elements will not start with the same letter.

Description

Short the names from a template. The merging GUI is designed to merge data from different files. But sometimes the file names are too long to be displayed in the GUI. Hence this function is used to short the basenames by removing the same beginning letters of each name. Hence the output is a character vector whose elements will not start with the same letter.

Usage

```
simplifynames(namevector)
```

Arguments

`namevector`	A character vector.

Value

A character vector which cuts the first several same letters from the input.

Author(s)

Xiaoyue Cheng <<xycheng@iastate.edu>>

Examples

```
simplifynames(c("abc234efg.csv", "abc234hfg.csv"))
simplifynames(c("12345", "54321"))
simplifynames(c("aeiou", "aerial"))
```

var.class *Detect the classes of the variables.*

Description

This function gives an initial guess of the classes of each variable in the merged data.

Usage

```
var.class(nametable.class, dataset.class)
```

Arguments

`nametable.class`
A matrix of the matched variable names. The number of columns is equal to the number of files. Each row represents a variable that is going to be merged. Any elements except NA in nametable.class must be the variable names in dataset.class.

`dataset.class`
The dataset list. The length of the list is equal to the number of files, and the order of the list is the same as the order of columns in nametable.class.

Value

A vector matching the rows of 'nametable.class'. The value includes NA if any variable are only NA's.

Author(s)

Xiaoyue Cheng <<xycheng@iastate.edu>>

Examples

```
a = data.frame(aa = 1:5, ab = LETTERS[6:2], ac = as.logical(c(0, 1, 0, NA, 0)))
b = data.frame(b1 = letters[12:14], b2 = 3:1)
dat = list(a, b)
name = matrix(c("ab", "aa", "ac", "b1", "b2", NA), ncol = 2)
var.class(name, dat)
```

Index