Package 'PreProcessRecordLinkage'

January 20, 2025

Type Package

Title Preprocessing Record Linkage

Version 1.0.1

Date 2023-09-12

Description In this record linkage package, data preprocessing has been meticulously executed to cover a wide range of datasets, ensuring that variable names are standardized using synonyms. This approach facilitates seamless data integration and analysis across various datasets. While users have the flexibility to modify variable names, the system intelligently ensures that changes are only permitted when they do not compromise data consistency or essential variable essence.

License GPL-3

Depends tm, syn, RecordLinkage, data.table

Imports stringr

Maintainer Leila Marvian Mashhad <Leila.marveian@gmail.com>

NeedsCompilation no

Author Hossein Hassani [aut],
Leila Marvian Mashhad [aut, cre]

Repository CRAN

Date/Publication 2023-09-13 09:10:02 UTC

Contents

chzInput 2
create_new_data 3
preproc 4
preprocLinkage 5
selVar 6

Index **8**

chzInput *Consulting User*

Description

After the pre processing of the data sets by `preproc` function, a series of changes were made on the names of the two variables for uniformity. Sometimes these changes of names based on synonyms are not desired by the user. In this function, according to the output of the `preproc` function, the user is asked to tell the program that any change in the name of the variables that he does not want.

Usage

```
chzInput(d1, d2, chz = "NULL")
```

Arguments

`d1`	A data frame.
`d2`	A data frame.
`chz`	the number of the name of the variable that the user does not want to change based on the output of the `preproc` function.

Details

For more details about this function, refer to `preproc` function manual.

Value

A vector of characters. It is a vector of characters that shows the names of the variables of the second data set based on the opinion of the user who said which variable name should not be changed.

Author(s)

Hossein Hassani and and Leila Marvian Mashhad.

See Also

`preproc`

Examples

```
d1 = RLdata500
d2 = RLdata10000
chzInput(d1, d2)
```

`create_new_data` *Final Data Preprocessing*

Description

First, after calling the two data sets, preliminary dat preprocessing is done using `preproc` function. Then, according to its output, the user decides which variables should not be renamed. Then this function performs complementary data preprocessing such as sorting the names of the variables, matching the gender variable with different formats, etc. and produces two new data frames.

Usage

```
create_new_data(d1, d2, chz = "NULL")
```

Arguments

`d1`	A data frame.
`d2`	A data frame.
`chz`	the number of the name of the variable that the user does not want to change based on the output of the `preproc` function.

Value

Two data frames.

Author(s)

Hossein Hassani and and Leila Marvian Mashhad.

See Also

`preproc`

Examples

```
d1 = RLdata500
d2 = RLdata10000
create_new_data(d1, d2)
```

preproc *Preprocessing and Unification of Variable Names of Two Input Data Sets*

Description

In this function data preprocessing has been meticulously executed to cover a wide range of datasets, ensuring that variable names are standardized using synonyms.

Usage

```
preproc(d1, d2)
## S3 method for class 'explain'
print(x,...)
```

Arguments

d1	A data frame.
d2	A data frame.
x	an object of class 'eplain'.
...	further arguments passed to preproc function.

Details

Because we want users to be able to change their names. The output of this function gives the names and classes that have changed in the new version and the previous version, as well as the number of changes in both datasets. Returns the corresponding number for the chz argument in the chzInput function.

Value

preproc an object of class 'explain'.

An object of class 'explain' is a list containing the following components:

Changed variable's names
: Character.

Changed variable's classes
: Character.

Initial variable's names
: Character.

Initial variable's classes
: Character.

A number of changed variable values for the first dataset are
: Data frame.

A number of changed variable values for the second dataset
: Data frame.

Number of changed variable's names
: Vector.

Note

This function has a comprehensible output if changes have been made on the names of the variables for equalization, otherwise it has no specific output and everything is zero.

In addition, it should be noted that the names of the variables of the second data set are matched and the necessary changes are made based on the first data set.

Author(s)

Hossein Hassani and and Leila Marvian Mashhad.

See Also

chzInput

Examples

```
d1 = RLdata500
d2 = RLdata10000
preproc(d1, d2)
```

`preprocLinkage` *Record Linkage with Data Preprocessing*

Description

This function records linkage along with data preprocessing. It has been meticulously executed to cover a wide range of datasets, ensuring that variable names are standardized using synonyms. This approach facilitates seamless data integration and analysis across various datasets.

Usage

```
preprocLinkage(d1,d2,chz="NULL",var=c("age","sex"),threshold=0.9)
```

Arguments

`d1`	A data frame.
`d2`	A data frame.
`chz`	the number of the name of the variable that the user does not want to change based on the output of the `preproc` function.
`var`	The vector of the names of the blocked variables that the user chooses based on the output of the `selVar` function that gives the vector of the names of the common variables between the two data sets.
`threshold`	A numeric value between 0 and 1.

Details

The results are stored in the .csv files, but if the number of records exceeds one million, they are stored in the rdata files.

Value

Two csv files or two rdata files.

Note

Note that, to see the results in the created file, first call the data.table package.

Author(s)

Hossein Hassani and and Leila Marvian Mashhad.

See Also

selVar, chzInput

Examples

```
d1 = RLdata500
d2 = RLdata10000
preprocLinkage(d1, d2, var = "by")
```

---

selVar *Display the names of common variables*

Description

This function displays the names of common variables based on the create_new_data function so that the user can give any variable he/she wants as a blocked variable in the preprocLinkage function.

Usage

```
selVar(d1, d2, chz = "NULL")
```

Arguments

d1	A data frame.
d2	A data frame.
chz	the number of the name of the variable that the user does not want to change based on the output of the preproc function.

Value

Character.

Author(s)

Hossein Hassani and and Leila Marvian Mashhad.

See Also

preproc

Examples

```
d1 = RLdata500
d2 = RLdata10000
selVar(d1, d2)
```

Index

chzInput, 2, *5, 6*
create_new_data, 3

preproc, *2*, *3*, 4, *5*, *7*
preprocLinkage, 5
print.explain *(*preproc*)*, 4

selVar, *6*, 6