Package: dfexpand (via r-universe)

September 13, 2024

Title Automatically Expand Delimited Column Values into Multiple Binary Columns with 'dfexpand'

Description Implements an algorithm to effortlessly split a column in an R data frame filled with multiple values separated by delimiters. This automates the process of creating separate columns for each unique value, transforming them into binary outcomes.

Version 0.0.2

Imports stringr

License GPL (>= 3)

URL `https://github.com/jlpainter/dfexpand`

BugReports `https://github.com/jlpainter/dfexpand/issues/`

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Repository https://jlpainter.r-universe.dev

RemoteUrl https://github.com/jlpainter/dfexpand

RemoteRef HEAD

RemoteSha 069c6bb2c1dd7833614be554c205f1d4b2ae05f4

Contents

`expand_column` *Expand a single column containing delimited values into multiple binary columns*

Description

Expand a single column containing delimited values into multiple binary columns

Usage

```
expand_column(
  dataframe,
  colname = NULL,
  delimiter = ";",
  trim = TRUE,
  ignore_case = FALSE,
  colnumber = NULL
)
```

Arguments

`dataframe` The data frame containing the column we want to expand

`colname` The name of the column to split on.

`delimiter` A single character to split the string on.

`ignore_case` Boolean flag if you want the split values to ignore case

`colnumber` You can provide the column number in the dataframe to expand, rather than the name

`trim` Boolean field to trim white space when searching for unique values

Value

A list of distinct values found in the entry string

Examples

```
library('dfexpand')
myDelimiter = ";"

# Create some fake data with duplicates
rows = c(
    c("a;b"), c("a;b;c"), c("b;c"), c("d"), c("d")
)

# Add to a dataframe
df = data.frame(rows)

colnames(df) <- c("myvar")
```

```
 #
 # The default behavior is to trim extra whitespace from the extracted values,
# but not to alter or change the case of the values. So 'Alpha' is distinct from 'alpha'
 # but ' beta ' is the same as 'beta'. You can override this behavior with
 # the trim and ignore case flags.
 #
 expanded_df = expand_column(df, "myvar", myDelimiter)
```

`getDistinctValues` *dfexpand*

Description

Methods to auto-expand a delimited string into a list of unique values

Usage

```
getDistinctValues(entry, delimiter, trim = TRUE, ignore_case = FALSE)
```

Arguments

`entry` A string to parse.

`delimiter` A single character to split the string on.

`trim` Boolean flag to signify if the leading and trailing whitespace should be trimmed for each value found.

`ignore_case` Boolean flag to indicate if the unique values extracted should ignore case differences or not.

Value

list

A list of distinct values found in the entry string

Examples

```
values <- getDistinctValues("a;b;c", ';')
```

Index