Package 'bpa'

October 12, 2022

Type Package

Title Basic Pattern Analysis

Version 0.1.1

Date 2016-04-03

Description Run basic pattern analyses on character sets, digits, or combined input containing both characters and numeric digits. Useful for data cleaning and for identifying columns containing multiple or nonstandard formats.

Depends base

Imports magrittr, plyr

Suggests testthat, knitr, rmarkdown

License GPL (>= 2)

URL https://github.com/bgreenwell/bpa

BugReports https://github.com/bgreenwell/bpa/issues

RoxygenNote 5.0.1

VignetteBuilder knitr

NeedsCompilation no

Author Brandon Greenwell [aut, cre]

Maintainer Brandon Greenwell <greenwell.brandon@gmail.com>

Repository CRAN

Date/Publication 2016-04-04 08:37:03

R topics documented:

get_pattern *Basic Pattern Analysis*

Description

Perform a basic pattern analysis

Usage

```
get_pattern(x, show_ws = TRUE, ws_char = "w")

basic_pattern_analysis(x, unique_only = FALSE, show_ws = TRUE,
  ws_char = "w", useNA = c("no", "ifany", "always"), ...)

## Default S3 method:
basic_pattern_analysis(x, unique_only = FALSE,
  show_ws = TRUE, ws_char = "w", useNA = c("no", "ifany", "always"), ...)

## S3 method for class 'data.frame'
basic_pattern_analysis(x, unique_only = FALSE,
  show_ws = TRUE, ws_char = "w", useNA = c("no", "ifany", "always"), ...)

bpa(x, ...)
```

Arguments

x	A data frame or character vector.
show_ws	Logical indicating whether or not to show whitespace using a special character. Default is TRUE.
ws_char	Character string to use to depict whitespace when show_ws = TRUE.
unique_only	Logical indicating whether or not to only show the unique patterns. Default is TRUE.
useNA	Logical indicating whether to include NA values in the table. See table for details.
...	Additional optional arguments to be passed onto llply.

Examples

```
basic_pattern_analysis(iris)
basic_pattern_analysis(iris, unique_only = TRUE)
```

match_pattern *Pattern Matching*

Description

Extract values from a vector that match a particular pattern.

Usage

```
match_pattern(x, pattern, unique_only = FALSE, ...)
```

Arguments

x	A vector, typically of class "character".
pattern	Character string specifying the particular pattern to match.
unique_only	Logical indicating whether or not to only return unique values. Default is FALSE.
...	Additional optional arguments to ba passed onto get_pattern.

Details

The pattern specified by the required argument pattern must be a valid pattern produced by the get_pattern function. That is, all digits should be represented by a "9", lowercase/uppercase letters by a "a"/"A", etc.

Examples

```
phone <- c("123-456-7890", "456-7890", "123-4567", "456-7890")
match_pattern(phone, pattern = "999-9999")
match_pattern(phone, pattern = "999-9999", unique_only = TRUE)
```

messy *Simulated Data*

Description

Simulated (messy) data set to help illustrate some of the uses of basic pattern analysis.

Format

A data frame with 1000 rows and 3 variables

Details

- Gender Gender in various formats.
- Date Dates in various formats.
- Phone Phone numbers in various formats.

Examples

```
data(messy)
bpa(messy, unique_only = TRUE, ws_char = " ")
```

`trim_ws` *Remove Leading/Trailing Whitespace*

Description

Remove leading and/or trailing whitespace from character strings.

Usage

```
trim_ws(x, which = c("both", "left", "right"))
```

Arguments

`x`	A data frame or vector.
`which`	A character string specifying whether to remove both leading and trailing whitespace (default), or only leading (`"left"`) or trailing (`"right"`). Can be abbreviated.

Examples

```
# Toy example
d <- data.frame(x = c(" a ", "b ", "c"),
                y = c("   1 ", "2", " 3"),
                z = c(4, 5, 6))
print(d)  # print data as is
trim_ws(d)  # print data with whitespace trimmed off
sapply(trim_ws(d), class)  # check that column types are preserved
```

Index