Package 'pivotea'

July 13, 2024

Title Create Pivot Table Easily

Version 1.0.2

Description Pivot easily by specifying rows, columns, values and split.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

URL https://github.com/matutosi/pivotea,
https://matutosi.github.io/pivotea/

LazyData true

Imports dplyr, purrr, rlang, tidyr

Suggests ggplot2, knitr, rmarkdown, spelling, testthat (>= 3.0.0),
tibble

Config/testthat/edition 3

Language en-US

VignetteBuilder knitr

Depends R (>= 2.10)

NeedsCompilation no

Author Toshikazu Matsumura [aut, cre]
(<https://orcid.org/0000-0003-2306-6355>)

Maintainer Toshikazu Matsumura <matutosi@gmail.com>

Repository CRAN

Date/Publication 2024-07-13 06:40:02 UTC

Contents

`add_group_sub` *Add sub index within group*

Description

Add sub index within group

Usage

```
add_group_sub(df, group, sep = "_", tmp_col = "tmp_col")
```

Arguments

`df`	A dataframe.
`group`	A string or string vector. When vector, the first string will be used for adding sub index.
`sep`	A string for separator.
`tmp_col`	A string of colnames for temporary use.

Value

A dataframe.

Examples

```
library(dplyr)
add_group_sub(mtcars, c("am", "gear"))
add_group_sub(mtcars, c("cyl", "am"))
```

`extract_col` *Helper for na_col_omit()*

Description

Helper for na_col_omit()

Usage

```
extract_col(col, df)
```

Arguments

`col`	A string or string vector.
`df`	A dataframe.

Value

A vector.

Examples

```
library(tidyr)
library(dplyr)
library(purrr)
library(ggplot2)
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "teacher", "room"),
        split = c("house", "grade"))
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "room", "house", "grade"),
        split = c("teacher"))
starwars |>
  pivot(row = "homeworld", col = "species", value = "name", split = "sex")
msleep |>
  pivot(row = "vore", col = "conservation", value = "name") |>
  na2empty() |>
  print(n = Inf)
tibble::as_tibble(Titanic) |>
  pivot(row = "Age", col = c("Sex", "Survived"),
        value = "n", split = "Class")
diamonds |>
  pivot(row = "cut", col = "color", value = "price", split = "clarity")
```

has_col *Detect if df has col*

Description

Detect if df has col

Usage

```
has_col(df, col)
```

Arguments

df	A dataframe.
col	A string or string vector.

Value

A dataframe.

Examples

```
colnames(mtcars)
has_col(mtcars, c("mpg", "cyl"))
has_col(mtcars, c("mpg", "foo"))
```

hogwarts *Timetable in Hogwarts School of Witchcraft and Wizardry.*

Description

Timetable in Hogwarts School of Witchcraft and Wizardry.

Usage

```
hogwarts
```

Format

A data frame with 548 rows and 7 variable:

grade Grades in school.

house Houses. G: Gryffindor, S: Slytherin, R: Ravenclaw, and H: Hufflepuff.

wday Abbreviations of day of the week.

hour Hours.

teacher Teachers.

subject Subjects.

room

Examples

```
data(hogwarts)
hogwarts
```

---

na2empty *replace NA character into ""*

Description

replace NA character into ""

Usage

```
na2empty(df)
```

Arguments

df	A dataframe.

Value

A dataframe.

Examples

```
library(tidyr)
library(dplyr)
library(purrr)
library(ggplot2)
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "teacher", "room"),
        split = c("house", "grade"))
hogwarts |>
  pivot(row = "hour", col = "wday",
```

```
        value = c("subject", "room", "house", "grade"),
        split = c("teacher"))
starwars |>
  pivot(row = "homeworld", col = "species", value = "name", split = "sex")
msleep |>
  pivot(row = "vore", col = "conservation", value = "name") |>
  na2empty() |>
  print(n = Inf)
tibble::as_tibble(Titanic) |>
  pivot(row = "Age", col = c("Sex", "Survived"),
        value = "n", split = "Class")
diamonds |>
  pivot(row = "cut", col = "color", value = "price", split = "clarity")
```

`omit_na_cols` *Remove all NA cols*

Description

Remove all NA cols

Usage

```
omit_na_cols(df)
```

Arguments

`df`	A dataframe.

Value

A dataframe.

Examples

```
library(tidyr)
library(dplyr)
library(purrr)
library(ggplot2)
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "teacher", "room"),
        split = c("house", "grade"))
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "room", "house", "grade"),
        split = c("teacher"))
starwars |>
  pivot(row = "homeworld", col = "species", value = "name", split = "sex")
```

```
msleep |>
  pivot(row = "vore", col = "conservation", value = "name") |>
  na2empty() |>
  print(n = Inf)
tibble::as_tibble(Titanic) |>
  pivot(row = "Age", col = c("Sex", "Survived"),
        value = "n", split = "Class")
diamonds |>
  pivot(row = "cut", col = "color", value = "price", split = "clarity")
```

`omit_na_rows` *Remove all NA rows*

Description

Remove all NA rows

Usage

```
omit_na_rows(df)
```

Arguments

`df`	A dataframe.

Value

A dataframe.

Examples

```
library(tidyr)
library(dplyr)
library(purrr)
library(ggplot2)
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "teacher", "room"),
        split = c("house", "grade"))
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "room", "house", "grade"),
        split = c("teacher"))
starwars |>
  pivot(row = "homeworld", col = "species", value = "name", split = "sex")
msleep |>
  pivot(row = "vore", col = "conservation", value = "name") |>
  na2empty() |>
  print(n = Inf)
```

```
tibble::as_tibble(Titanic) |>
  pivot(row = "Age", col = c("Sex", "Survived"),
        value = "n", split = "Class")
diamonds |>
  pivot(row = "cut", col = "color", value = "price", split = "clarity")
```

`pivot` *Pivot easily by specifying rows, columns, values and split.*

Description

Pivot easily by specifying rows, columns, values and split.

Usage

```
pivot(df, row, col, value, split = NULL, sep = "_", rm_empty_df = TRUE)
```

Arguments

`df`	A dataframe.
`row`, `value`	A string or string vector.
`col`	A string or string vector.
`split`	A string or string vector.
`sep`	A string for separator.
`rm_empty_df`	A logical for removing empty df.

Value

A dataframe.

Examples

```
library(tidyr)
library(dplyr)
library(purrr)
library(ggplot2)
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "teacher", "room"),
        split = c("house", "grade"))
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "room", "house", "grade"),
        split = c("teacher"))
starwars |>
  pivot(row = "homeworld", col = "species", value = "name", split = "sex")
msleep |>
```

```
  pivot(row = "vore", col = "conservation", value = "name") |>
  na2empty() |>
  print(n = Inf)
tibble::as_tibble(Titanic) |>
  pivot(row = "Age", col = c("Sex", "Survived"),
        value = "n", split = "Class")
diamonds |>
  pivot(row = "cut", col = "color", value = "price", split = "clarity")
```

`replace_col` *Replace a col with a data.frame.*

Description

Replace a col with a data.frame.

Usage

```
replace_col(df, replace)
```

Arguments

`df`, `replace`	A dataframe.

Value

A dataframe.

Examples

```
(state <- tibble::tibble(state = state.name, area = state.area))
(abb <- tibble::tibble(state = state.name, abb = state.abb))
replace_col(state, abb)
```

`split_force` *Split by force with "" when split is NULL*

Description

Split by force with "" when split is NULL

Usage

```
split_force(df, split)
```

Arguments

df	A dataframe.
split	A string or string vector.

Value

A dataframe.

Examples

```
split_force(mtcars, split = NULL)
split_force(mtcars, split = c("cyl"))
```

---

validate_col *Validate col*

Description

Validate col

Usage

```
validate_col(df, col)
```

Arguments

df	A dataframe.
col	A string or string vector.

Value

A dataframe.

Examples

```
library(tidyr)
library(dplyr)
library(purrr)
library(ggplot2)
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "teacher", "room"),
        split = c("house", "grade"))
hogwarts |>
  pivot(row = "hour", col = "wday",
        value = c("subject", "room", "house", "grade"),
        split = c("teacher"))
```

```
starwars |>
  pivot(row = "homeworld", col = "species", value = "name", split = "sex")
msleep |>
  pivot(row = "vore", col = "conservation", value = "name") |>
  na2empty() |>
  print(n = Inf)
tibble::as_tibble(Titanic) |>
  pivot(row = "Age", col = c("Sex", "Survived"),
        value = "n", split = "Class")
diamonds |>
  pivot(row = "cut", col = "color", value = "price", split = "clarity")
```

Index