Package ‘flattabler’

September 14, 2023

Title Obtaining a Flat Table from Pivot Tables

Version 2.1.0

Description Transformations that allow obtaining a flat table from reports in text or Excel format that contain data in the form of pivot tables. They can be defined for a single report and applied to a set of reports.

License MIT + file LICENSE

URL https://josesamos.github.io/flattabler/,
https://github.com/josesamos/flattabler

BugReports https://github.com/josesamos/flattabler/issues

Depends R (>= 2.10)

Imports dplyr, readxl, stringr, tibble, utils

Suggests knitr, pander, rmarkdown, testthat, tidyr

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

LazyData true

RoxygenNote 7.2.3

NeedsCompilation no

Author Jose Samos [aut, cre, cph] (https://orcid.org/0000-0002-4457-3439)

Maintainer Jose Samos <jsamos@ugr.es>

Repository CRAN

Date/Publication 2023-09-14 20:30:02 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>define_labels</td>
<td>3</td>
</tr>
<tr>
<td>df_ex</td>
<td>4</td>
</tr>
<tr>
<td>df_ex_compact</td>
<td>4</td>
</tr>
<tr>
<td>df_pivottabler</td>
<td>5</td>
</tr>
<tr>
<td>df_set_h</td>
<td>5</td>
</tr>
<tr>
<td>df_set_h_v</td>
<td>6</td>
</tr>
<tr>
<td>df_set_v</td>
<td>6</td>
</tr>
<tr>
<td>divide</td>
<td>7</td>
</tr>
<tr>
<td>extract_labels</td>
<td>8</td>
</tr>
<tr>
<td>fill_labels</td>
<td>9</td>
</tr>
<tr>
<td>fill_values</td>
<td>10</td>
</tr>
<tr>
<td>flatten_table_list</td>
<td>11</td>
</tr>
<tr>
<td>ft_ex</td>
<td>12</td>
</tr>
<tr>
<td>ft_ex_v2</td>
<td>12</td>
</tr>
<tr>
<td>ft_set</td>
<td>13</td>
</tr>
<tr>
<td>get_col_values</td>
<td>14</td>
</tr>
<tr>
<td>get_page</td>
<td>15</td>
</tr>
<tr>
<td>pf_ex_compact</td>
<td>16</td>
</tr>
<tr>
<td>pivot_table</td>
<td>17</td>
</tr>
<tr>
<td>pt_ex</td>
<td>18</td>
</tr>
<tr>
<td>read_excel_file</td>
<td>18</td>
</tr>
<tr>
<td>read_excel_folder</td>
<td>19</td>
</tr>
<tr>
<td>read_excel_sheet</td>
<td>21</td>
</tr>
<tr>
<td>read_text_file</td>
<td>22</td>
</tr>
<tr>
<td>read_text_folder</td>
<td>23</td>
</tr>
<tr>
<td>remove_agg</td>
<td>24</td>
</tr>
<tr>
<td>remove_bottom</td>
<td>25</td>
</tr>
<tr>
<td>remove_cols</td>
<td>26</td>
</tr>
<tr>
<td>remove_empty</td>
<td>27</td>
</tr>
<tr>
<td>remove_k</td>
<td>28</td>
</tr>
<tr>
<td>remove_left</td>
<td>29</td>
</tr>
<tr>
<td>remove_right</td>
<td>30</td>
</tr>
<tr>
<td>remove_rows</td>
<td>31</td>
</tr>
<tr>
<td>remove_top</td>
<td>32</td>
</tr>
<tr>
<td>replace_dec</td>
<td>33</td>
</tr>
<tr>
<td>set_page</td>
<td>34</td>
</tr>
<tr>
<td>unpivot</td>
<td>35</td>
</tr>
</tbody>
</table>

Index

37
**define_labels**  

*Define the quantity of rows and columns that contain labels*

---

**Description**

A pivot table should only contain label rows and columns, and an array of values, usually numeric data. This function defines the quantity of rows and columns that contain labels.

**Usage**

```r
define_labels(pt, n_col, n_row)
```

```r
## S3 method for class 'pivot_table'
define_labels(pt, n_col, n_row)
```

**Arguments**

- `pt`  
  A pivot_table object.

- `n_col`  
  A number, quantity of columns containing pivot table labels.

- `n_row`  
  A number, quantity of rows containing pivot table labels.

**Value**

A pivot_table object.

**See Also**

pivot_table

Other pivot table definition functions: get_page(), pivot_table(), set_page()

**Examples**

```r
pt <- pt_ex |> define_labels(n_col = 2, n_row = 2)
```
**df_ex**

Pivot table in data frame with with thousands indicator and decimal numbers.

**Usage**

```r
df_ex
```

**Format**

A data frame.

**See Also**

- `pt_ex`

Other pivot table in data frame: `df_ex_compact`, `df_pivottabler`

---

**df_ex_compact**

Pivot table in data frame with a column with data from two label fields

**Description**

Pivot table in data frame in compact table format: with a column with data from two label fields.

**Usage**

```r
df_ex_compact
```

**Format**

A data frame.

**See Also**

- `pf_ex_compact`

Other pivot table in data frame: `df_ex`, `df_pivottabler`
df_pivottabler

Pivot table with basic and subtotal labels in the same column

Description
A dataset containing number of train passengers, generated with the pivottabler package. It contains basic and subtotal labels in the same column.

Usage
df_pivottabler

Format
A data frame.

Source
https://CRAN.R-project.org/package=pivottabler

See Also
Other pivot table in data frame: df_ex_compact, df_ex

df_set_h

Set of pivot tables placed horizontally on one sheet

Description
Set of pivot tables placed horizontally on one sheet.

Usage
df_set_h

Format
A data frame.

See Also
df_ex
Other pivot table set in data frame: df_set_h_v, df_set_v
**df_set_h_v**  
Set of pivot tables on one sheet

---

**Description**  
Example of a set of pivot tables located horizontally and vertically on one sheet.

**Usage**  
df_set_h_v

**Format**  
A data frame.

**See Also**  
df_ex  
Other pivot table set in data frame: df_set_h, df_set_v

---

**df_set_v**  
Set of pivot tables placed vertically on one sheet

---

**Description**  
Set of pivot tables placed vertically on one sheet.

**Usage**  
df_set_v

**Format**  
A data frame.

**See Also**  
df_ex  
Other pivot table set in data frame: df_set_h_v, df_set_h
**divide**

**Divide table**

Description

Divides a table into tables separated by some empty row or column. Returns a `pivot_table` object list.

Usage

```r
divide(pt)
```

## S3 method for class 'pivot_table'

```r
divide(pt)
```

Arguments

- `pt` A `pivot_table` object.

Details

Sometimes multiple pivot tables are placed in a text document, imported as one text table. This operation recursively divides the initial table into tables separated by some empty row or column. Once a division has been made, it tries to divide each part of the result. An object is generated for each indivisible pivot table. Returns a list of `pivot_table` objects.

If individual tables have a header or footer, they should not be separated from the table by empty rows. If they were, objects would be generated from them that must later be removed from the list of objects in the result.

The operation can be applied to tables located horizontally, vertically or in a grid on the initial table. The only requirement to be able to divide it is that there is some empty row or column between them.

Value

A `pivot_table` list.

See Also

- `pivot_table`

Other flat table list functions: `flatten_table_list()`, `get_col_values()`

Examples

```r
pt <- pivot_table(df_set_h_v)
lpt <- pt |> divide()
```
**extract_labels**

---

**Description**

Extract the given set of labels from a table column in compact format to generate a new column in the table.

**Usage**

```r
extract_labels(pt, col, labels)
```

```r
## S3 method for class 'pivot_table'
extract_labels(pt, col = 1, labels = c())
```

**Arguments**

- **pt**  
  A `pivot_table` object.

- **col**  
  A number, column from which labels are extracted.

- **labels**  
  A vector of strings, set of labels to extract.

**Details**

Sometimes a table column includes values of multiple label fields, this is generally known as compact table format. Given a column number and a set of labels, it generates a new column with the labels located at the positions they occupied in the original column and removes them from it.

**Value**

A `pivot_table` object.

**See Also**

- `pivot_table`

Other pivot table transformation functions: `fill_labels()`, `fill_values()`, `remove_agg()`, `remove_bottom()`, `remove_cols()`, `remove_empty()`, `remove_k()`, `remove_left()`, `remove_right()`, `remove_rows()`, `remove_top()`, `replace_dec()`, `unpivot()`

**Examples**

```r
pt <- pivot_table(df_ex_compact) |>  
e xtract_labels(col = 1, labels = c("b1", "b2", "b3", "b4", "Total general"))
```
Description

Fills missing values in row and column labels for a pivot table. By default, columns are filled down and rows are filled right.

Usage

```r
fill_labels(pt, down, right)
```

```r
## S3 method for class 'pivot_table'
fill_labels(pt, down = TRUE, right = TRUE)
```

Arguments

- **pt**: A `pivot_table` object.
- **down**: A boolean, fill down.
- **right**: A boolean, fill right.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data. The row and column closest to the data array are not filled (they must have data defined for each cell).

To correctly carry out this operation, the number of rows and columns that contain labels must be defined, and the table must only contain the pivot table rows and columns.

Value

A `pivot_table` object.

See Also

- `pivot_table`

Other pivot table transformation functions: `extract_labels()`, `fill_values()`, `remove_agg()`, `remove_bottom()`, `remove_cols()`, `remove_empty()`, `remove_k()`, `remove_left()`, `remove_right()`, `remove_rows()`, `remove_top()`, `replace_dec()`, `unpivot()`

Examples

```r
pt <-
  pt_ex |>
  remove_top(1) |>
  define_labels(n_col = 2, n_row = 2) |>
```
fill_labels(down = TRUE, right = TRUE)

---

fill_values  Fill in missing values

Description

Fills with NA missing values in a pivot table value array.

Usage

fill_values(pt)

## S3 method for class 'pivot_table'
fill_values(pt)

Arguments

pt            A pivot_table object.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

To correctly carry out this operation, the number of rows and columns that contain labels must be defined, and the table must only contain the pivot table rows and columns.

Value

A pivot_table object.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_left(), remove_right(), remove_rows(), remove_top(), replace_dec(), unpivot()

Examples

pt <-
  pt_ex |> remove_top(1) |> define_labels(n_col = 2, n_row = 2) |> fill_values()
Transform a pivot_table object list into a flat table

Description
Given a list of pivot_table objects and a transformation function that flattens a pivot_table object, transforms each object using the function and merges the results into a flat table.

Usage
`flatten_table_list(lpt = list(), FUN)`

Arguments
- `lpt`: A list of pivot_table objects.
- `FUN`: A function, transformation function that flattens a pivot_table object (it returns a tibble).

Value
A tibble, a flat table implemented by a tibble.

See Also
- `pivot_table`
- Other flat table list functions: `divide()`, `get_col_values()`

Examples
```r
f <- function(pt) {
  pt |> 
    set_page(1, 1) |> 
    remove_top(1) |> 
    define_labels(n_col = 2, n_row = 2) |> 
    remove_k() |> 
    replace_dec() |> 
    fill_values() |> 
    fill_labels() |> 
    remove_agg() |> 
    unpivot()
}

pt <- pivot_table(df_set_h_v)
lpt <- pt |> divide()
ft <- flatten_table_list(lpt, f)
```
Description
Flat table with page from a pivot table with with thousands indicator and decimal numbers.

Usage
ft_ex

Format
A tibble object.

See Also
df_ex

Other flat table: ft_ex_v2, ft_set

Examples
# Defined by:

ft_ex <- pivot_table(df_ex) |>  
set_page(1, 1) |>  
remove_top(1) |>  
define_labels(n_col = 2, n_row = 2) |>  
remove_k() |>  
replace_dec() |>  
fill_values() |>  
fill_labels() |>  
remove_agg() |>  
unpivot()

Description
Flat table without page from a pivot table with with thousands indicator and decimal numbers.

Usage
ft_ex_v2
ft_set

**Format**

A tibble object.

**See Also**

df_ex

Other flat table: ft_ex, ft_set

**Examples**

# Defined by:

```r
ft_ex_v2 <- pivot_table(df_ex) |>
    set_page(1, 1) |>
    remove_top(1) |>
    define_labels(n_col = 2, n_row = 2) |>
    remove_k() |>
    replace_dec() |>
    fill_values() |>
    fill_labels() |>
    remove_agg() |>
    unpivot(include_page = FALSE, na_rm = FALSE)
```

---

| ft_set | Flat table with page from a pivot table with with thousands indicator and decimal numbers |

**Description**

Flat table with page from a pivot table with with thousands indicator and decimal numbers.

**Usage**

ft_set

**Format**

A tibble object.

**See Also**

df_set_h_v

Other flat table: ft_ex_v2, ft_ex
get_col_values

Examples

# Defined by:

```r
define_labels(n_col = 2, n_row = 2) |
remove_k() |
fill_values() |
fill_labels() |
remove_agg() |
unpivot()
}
```

```r
t <- pivot_table(df_set_h_v)
lpt <- t |> divide()
f <- f <- function(pt) {
  pt |>
    set_page(1, 1) |>
    remove_top(1) |>
    define_labels(n_col = 2, n_row = 2) |>
    remove_k() |
    replace_dec() |
    fill_values() |
    fill_labels() |
    remove_agg() |
    unpivot()
}
```

get_col_values            Get column values

Description

Gets the values of the indicated column of each table in a list of tables, avoiding the rows at the
beginning or the end of each table that are indicated.

Usage

```r
get_col_values(lpt, col = 1, start_row = 2, rows_left = 0)
```

Arguments

- `lpt`  
  pivot_table object list.
- `col`  
  A number, column to consider.
- `start_row`  
  A number, start row in each table.
- `rows_left`  
  A number, rows to ignore at the end of each table.

Details

Sometimes a column includes values of multiple label fields. To facilitate the study of the labels included in the same column of several tables, this function gets the values of the indicated column in a list of tables.
Value

Data frame with two columns: Labels in the column, and the index of the table in the list of tables from which they come.

See Also

pivot_table
Other flat table list functions: divide(), flatten_table_list()

Examples

pt <- pivot_table(df_set_h_v)
lpt <- pt |> divide()
df <- get_col_values(lpt, col = 1, start_row = 4)
labels <- sort(unique(df$label))
See Also

`pivot_table`

Other pivot table definition functions: `define_labels()`, `pivot_table()`, `set_page()`

Examples

```r
page <- pt_ex |> get_page()
```

---

### pf_ex_compact

*Pivot table result of transforming a data frame with a column with data from two label fields*

**Description**

Pivot table result of transforming a data frame in compact table format: with a column with data from two label fields.

**Usage**

```r
pf_ex_compact
```

**Format**

A `pivot_table` object.

**See Also**

`df_ex_compact`

Other pivot table: `pt_ex`

**Examples**

```r
# Defined by:

pf_ex_compact <- pivot_table(df_ex_compact) |> 
  extract_labels(col = 1, 
  labels = c("b1", "b2", "b3", "b4", "Total general"))
```
**pivot_table**

---

### pivot_table

**pivot_table S3 class**

**Description**

Creates a pivot_table object from a data frame. Additional information associated with the pivot table can be indicated. The data frame data is converted to character data type.

**Usage**

```r
pivot_table(
  df,
  page = "",
  page_row = 0,
  page_col = 0,
  n_col_labels = 0,
  n_row_labels = 0
)
```

**Arguments**

- **df**
  
  A data frame, contains one or more pivot tables.

- **page**
  
  A string, additional information associated with the pivot table.

- **page_row, page_col**
  
  A cell (row and column number), page information included in the table.

- **n_col_labels**
  
  A number, number of columns containing pivot table labels.

- **n_row_labels**
  
  A number, number of rows containing pivot table labels.

**Value**

A pivot_table object.

**See Also**

divide

Other pivot table definition functions: define_labels(), get_page(), set_page()

**Examples**

```r
pt <- pivot_table(df_ex)
pt <- pivot_table(df_ex, page = "M4")
pt <- pivot_table(df_ex, page_row = 1, page_col = 1)
pt <- pivot_table(df_ex, page_row = 1, page_col = 1, n_col_labels = 2, n_row_labels = 2)
```
**pt_ex**

*Pivot table with with thousands indicator and decimal numbers*

**Description**

Pivot table with with thousands indicator and decimal numbers.

**Usage**

```r
pt_ex
```

**Format**

A `pivot_table` object.

**See Also**

`df_ex`

Other pivot table: `pf_ex_compact`

**Examples**

```r
# Defined by:
pt_ex <- pivot_table(df_ex)
```

---

**read_excel_file**

*Import Excel file*

**Description**

Reads sheets from an Excel file and creates a `pivot_table` object list, one from each sheet. Each sheet is expected to contain a pivot table. Each line in a sheet corresponds to a row in a table. The file and sheet names are included as part of each object attributes.

**Usage**

```r
read_excel_file(
  file,
  sheetIndexes = NULL,
  sheetNames = NULL,
  define_page = 3,
  page_sep = "::"
)
```
read_excel_folder

Arguments

file
A string, name of an Excel file.

sheetIndexes
A vector of numbers, sheet indexes in the workbook.

sheetNames
A vector of strings, sheet names.

define_page
A integer, 0: no page, 1: file name as page, 2: sheet name as page, 3: file and sheet names as page, separated by the indicated separator.

page_sep
A string, separator to form the page value.

Details

When multiple files or sheets are handled, the file and/or sheet names may contain information associated with the pivot table, they could be the table page information. In order not to lose this information, they are always stored in each pivot_table object.

Value

A pivot_table object list.

See Also

pivot_table

Other import functions: read_excel_folder(), read_excel_sheet(), read_text_file(), read_text_folder()

Examples

# file <- system.file("extdata", "excel/set_sheets.xlsx", package = "flattabler")
# lpt <- read_excel_file(file)
# lpt <- read_excel_file(file, sheetIndexes = 1:4)
# lpt <- read_excel_file(file, sheetNames = c("M1", "M2", "M3", "M4"))

read_excel_folder  Import one sheet from each Excel file in a folder

Description

Reads one sheet (or all sheets) from each of the Excel files in a folder and creates a list of pivot_table objects, one from each sheet. Each sheet is expected to contain a pivot table. Each line in a file corresponds to a row in a table. File and sheet names are included as part of each object attributes.
read_excel_folder

Usage

read_excel_folder(
  folder,
  sheetIndex = 1,
  sheetName = NULL,
  allSheets = FALSE,
  define_page = 3,
  page_sep = ":\"
)

Arguments

folder A string, folder name.
sheetIndex A number, sheet index in the workbook.
sheetName A string, sheet name.
allSheets A boolean.
define_page A integer, 0: no page, 1: file name as page, 2: sheet name as page, 3: file and
sheet names as page, separated by the indicated separator.
page_sep A string, separator to form the page value.

Details

When multiple files or sheets are handled, the file and/or sheet names may contain information
associated with the pivot table, they could be the table page information. In order not to lose this
information, they are always stored in each pivot_table object.

Value

A pivot_table object list.

See Also

pivot_table

Other import functions: read_excel_file(), read_excel_sheet(), read_text_file(), read_text_folder()

Examples

# folder <- system.file("extdata", "excelfolder", package = "flattabler")
# lpt <- read_excel_folder(folder)

# lpt <- read_excel_folder(folder, allSheets = TRUE)
Description

Reads an Excel file sheet and creates a pivot_table object. The sheet is expected to contain one or more pivot tables. Each line in the sheet corresponds to a row in a table. The file and sheet names can be included as part of the object attributes.

Usage

```r
read_excel_sheet(
  file,
  sheetIndex = 1,
  sheetName = NULL,
  define_page = 3,
  page_sep = ":"n
)
```

Arguments

- `file` A string, name of an Excel file.
- `sheetIndex` A number, sheet index in the workbook.
- `sheetName` A string, sheet name.
- `define_page` A integer, 0: no page, 1: file name as page, 2: sheet name as page, 3: file and sheet names as page, separated by the indicated separator.
- `page_sep` A string, separator to form the page value.

Details

When multiple files or sheets are handled, the file and/or sheet names may contain information associated with the pivot table, they could be the table page information. In order not to lose this information, they can be stored in the pivot_table object.

Value

A pivot_table object.

See Also

`pivot_table`

Other import functions: `read_excel_file()`, `read_excel_folder()`, `read_text_file()`, `read_text_folder()`
Examples

```r
# file <- system.file("extdata", "excelfolder/m4.xlsx", package = "flattabler")
# pt <- read_excel_sheet(file)
# pt <- read_excel_sheet(file, sheetName = "Hoja2", define_page = 1)
```

Description

Reads a text file and creates a `pivot_table` object. The file is expected to contain one or more pivot tables. Each line in the file corresponds to a row in a table; within each row, columns are defined by a separator character. The file name is included as part of the object attributes.

Usage

```r
read_text_file(file, sep = ";", encoding = "UTF-8", define_page = TRUE)
```

Arguments

- `file`: A string, name of a text file.
- `sep`: Column separator character.
- `encoding`: A string, encoding to be assumed for input strings.
- `define_page`: A boolean, include file name as `pivot_table` page definition.

Details

When multiple files are handled, the file name may contain information associated with the pivot table, it could be the table page information. In order not to lose this information, it can be stored in the `pivot_table` object.

Value

A `pivot_table` object.

See Also

- `pivot_table`
- Other import functions: `read_excel_file()`, `read_excel_folder()`, `read_excel_sheet()`, `read_text_folder()`

Examples

```r
# file <- system.file("extdata", "csvfolder/m4.csv", package = "flattabler")
# pt <- read_text_file(file)
```
**read_text_folder**

Import all text files in a folder

---

**Description**

Reads all text files in a folder and creates a list of pivot_table objects, one from each file. Each file is expected to contain a pivot table. Each line in a file corresponds to a row in a table; within each row, columns are defined by a separator character. File name is included as part of each object attributes.

**Usage**

```r
read_text_folder(folder, sep = ";", encoding = "UTF-8")
```

**Arguments**

- `folder`: A string, folder name.
- `sep`: Column separator character.
- `encoding`: A string, encoding to be assumed for input strings.

**Details**

When multiple files are handled, the file name may contain information associated with the pivot table, it could be the table page information. In order not to lose this information, it is always stored in each pivot_table object.

**Value**

A pivot_table object list.

**See Also**

- `pivot_table`

Other import functions: `read_excel_file()`, `read_excel_folder()`, `read_excel_sheet()`, `read_text_file()`

**Examples**

```r
# folder <- system.file("extdata", "csvfolder", package = "flattabler")
# lpt <- read_text_folder(folder)
```
The function `remove_agg` is used to remove rows and columns with aggregated data from a pivot table.

**Description**

Aggregated data is recognized because the label of the row or column closest to the array of values is empty.

**Usage**

```r
remove_agg(pt)
```

**Arguments**

- `pt`: A pivot_table object.

**Details**

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

Removes pivot table rows and columns that contain aggregated data. It only checks the value in the row or column closest to the array of values.

To correctly carry out this operation, the number of rows and columns that contain labels must be defined, and the table must only contain the pivot table rows and columns.

**Value**

A pivot_table object.

**See Also**

- `pivot_table`
- Other pivot table transformation functions: `extract_labels()`, `fill_labels()`, `fill_values()`, `remove_bottom()`, `remove_cols()`, `remove_empty()`, `remove_k()`, `remove_left()`, `remove_right()`, `remove_rows()`, `remove_top()`, `replace_dec()`, `unpivot()`

**Examples**

```r
pt <-
  pt_ex |>
  remove_top(1) |>
  define_labels(n_col = 2, n_row = 2) |>
  remove_agg()
```
**remove_bottom**

| remove_bottom | Remove bottom rows from a pivot table |

---

**Description**

Remove bottom rows from the pivot table represented by the object.

**Usage**

```r
remove_bottom(pt, n)
```

```r
# S3 method for class 'pivot_table'
remove_bottom(pt, n)
```

**Arguments**

- **pt**: A `pivot_table` object.
- **n**: A number, number of rows to remove.

**Details**

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

All rows not belonging to the pivot table must be removed. It is common to find rows with footer information, which must be removed.

This function is very useful because it is not necessary to know the number of rows in the table.

**Value**

A `pivot_table` object.

**See Also**

- `pivot_table`

Other pivot table transformation functions: `extract_labels()`, `fill_labels()`, `fill_values()`, `remove_agg()`, `remove_cols()`, `remove_empty()`, `remove_k()`, `remove_left()`, `remove_right()`, `remove_rows()`, `remove_top()`, `replace_dec()`, `unpivot()`

**Examples**

```r
pt <- pt_ex |> remove_bottom(3)
```
remove_cols

Remove columns from a pivot table

Description

Remove the columns whose numbers are indicated from the pivot table represented by the object.

Usage

remove_cols(pt, c)

## S3 method for class 'pivot_table'
remove_cols(pt, c)

Arguments

pt
A pivot_table object.
c
A vector of numbers, column numbers.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

All columns not belonging to the pivot table must be removed.

Value

A pivot_table object.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_empty(), remove_k(), remove_left(), remove_right(), remove_rows(), remove_top(), replace_dec(), unpivot()

Examples

pt <- pt_ex |> remove_cols(7)
pt <- pt_ex |> remove_cols(c(6,7))
Description

Remove rows and columns without data from the pivot table represented by the object.

Usage

remove_empty(pt)

## S3 method for class 'pivot_table'
remove_empty(pt)

Arguments

pt A pivot_table object.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.
All rows and columns not belonging to the pivot table must be removed, including those without data.

Value

A pivot_table object.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(),
remove_agg(), remove_bottom(), remove_cols(), remove_k(), remove_left(), remove_right(),
remove_rows(), remove_top(), replace_dec(), unpivot()

Examples

df <- df_ex
df[seq(from = 1, to = 25, by = 2), ] <- " "
df[, seq(from = 1, to = 7, by = 2)] <- " "
pt <- pivot_table(df)
pt <- pt |> remove_empty()
Description

A pivot table should only contain label rows and columns, and an array of values, usually numeric data. Values, even though they are numbers, are represented as text and sometimes include a thousands separator that can be removed using this function.

Usage

```r
remove_k(pt, sep)
```

```r
## S3 method for class 'pivot_table'
remove_k(pt, sep = ".")
```

Arguments

- `pt`: A `pivot_table` object.
- `sep`: A character, thousands separator to remove.

Details

To correctly carry out this operation, the number of rows and columns that contain labels must be defined, and the table must only contain the pivot table rows and columns.

Value

A `pivot_table` object.

See Also

`pivot_table`

Other pivot table transformation functions: `extract_labels()`, `fill_labels()`, `fill_values()`, `remove_agg()`, `remove_bottom()`, `remove_cols()`, `remove_empty()`, `remove_left()`, `remove_right()`, `remove_rows()`, `remove_top()`, `replace_dec()`, `unpivot()`

Examples

```r
pt <-
  pt_ex |>
  remove_top(1) |>
  define_labels(n_col = 2, n_row = 2) |>
  remove_k()
```
remove_left

Remove left columns from a pivot table

Description
Remove left columns from the pivot table represented by the object.

Usage
remove_left(pt, n)

## S3 method for class 'pivot_table'
remove_left(pt, n)

Arguments
- pt: A pivot_table object.
- n: A number, number of columns to remove.

Details
A pivot table should only contain label rows and columns, and an array of values, usually numeric data.
All columns not belonging to the pivot table must be removed.

Value
A pivot_table object.

See Also
- pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_right(), remove_rows(), remove_top(), replace_dec(), unpivot()

Examples
pt <- pt_ex |> remove_left(3)
remove_right

Remove right columns from a pivot table

Description

Remove right columns from the pivot table represented by the object.

Usage

remove_right(pt, n)

# S3 method for class 'pivot_table'
remove_right(pt, n)

Arguments

pt A pivot_table object.
n A number, number of columns to remove.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.
All columns not belonging to the pivot table must be removed.
This function is very useful because it is not necessary to know the number of columns in the table.

Value

A pivot_table object.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_left(), remove_rows(), remove_top(), replace_dec(), unpivot()

Examples

pt <- pt_ex |> remove_right(3)
Remove rows from a pivot table

Description

Remove the rows whose numbers are indicated from the pivot table represented by the object.

Usage

remove_rows(pt, r)

## S3 method for class 'pivot_table'
remove_rows(pt, r)

Arguments

pt  A pivot_table object.

r   A vector of numbers, row numbers.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

All rows not belonging to the pivot table must be removed. It is common to find rows with header or footer information, which must be removed.

Value

A pivot_table object.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_left(), remove_right(), remove_top(), replace_dec(), unpivot()

Examples

pt <- pt_ex |> remove_rows(1)

pt <- pt_ex |> remove_rows(c(1, 8, 14, 19, 25, 26))
remove_top

Description

Remove top rows from a pivot table

Usage

remove_top(pt, n)

## S3 method for class 'pivot_table'
remove_top(pt, n)

Arguments

pt A pivot_table object.
n A number, number of rows to remove.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

All rows not belonging to the pivot table must be removed. It is common to find rows with header information, which must be removed.

Value

A pivot_table object.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_left(), remove_right(), remove_rows(), replace_dec(), unpivot()

Examples

pt <- pt_ex |> remove_top(3)
replace_dec

Replace decimal separator

Description
A pivot table should only contain label rows and columns, and an array of values, usually numeric data. Values, even though they are numbers, are represented as text and sometimes include a decimal separator different from the one needed; it can be replaced using this function.

Usage
replace_dec(pt, sep)

## S3 method for class 'pivot_table'
replace_dec(pt, sep = ".")

Arguments
- **pt**: A pivot_table object.
- **sep**: A character, new decimal separator to use.

Details
To correctly carry out this operation, the number of rows and columns that contain labels must be defined, and the table must only contain the pivot table rows and columns.
The only decimal separators considered are "." and ",".

Value
A pivot_table object.

See Also
pivot_table
Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_left(), remove_right(), remove_rows(), remove_top(), unpivot()

Examples

pt <-
  pt_ex |> replace_dec()
### set_page

**Set page information to a pivot table**

**Description**

Define the page information associated with a pivot table. Previously existing information is replaced by new information.

**Usage**

```r
set_page(pt, row, col, page)
```

```r
## S3 method for class 'pivot_table'
set_page(pt, row = 0, col = 0, page = "")
```

**Arguments**

- **pt**: A `pivot_table` object.
- **row, col**: A cell (row and column number), page information included in the table.
- **page**: A string, page information.

**Details**

Each pivot table implements a report. The pivot table page represents the context of that report. It is useful when we work with several pivot tables with the same structure: for example, the page can allow us to differentiate their origin, date or author. This information is often included in the file name, sheet name, or a cell attached to the pivot table.

**Value**

A `pivot_table` object.

**See Also**

- `pivot_table`

Other pivot table definition functions: `define_labels()`, `get_page()`, `pivot_table()`

**Examples**

```r
pt <- pt_ex |> set_page(1, 1)
pt <- pt_ex |> set_page(page = "M4")
```
unpivot

Unpivot a pivot table

Description

Transforms a pivot table into a flat table (implemented by a tibble). An additional column with page information can be included. NA values can be excluded from the array of values.

Usage

unpivot(pt, include_page, na_rm, keep_col_names)

## S3 method for class 'pivot_table'
unpivot(pt, include_page = TRUE, na_rm = TRUE, keep_col_names = FALSE)

Arguments

- **pt**: A pivot_table object.
- **include_page**: A boolean, indicates whether a column with the page information is included or not.
- **na_rm**: A boolean, indicates whether NA values from the array of values are removed or not.
- **keep_col_names**: A boolean, if possible, keep the column names.

Details

A pivot table should only contain label rows and columns, and an array of values, usually numeric data.

To correctly carry out this operation, the number of rows and columns that contain labels must be defined, and the table must only contain the pivot table rows and columns.

Value

A tibble.

See Also

pivot_table

Other pivot table transformation functions: extract_labels(), fill_labels(), fill_values(), remove_agg(), remove_bottom(), remove_cols(), remove_empty(), remove_k(), remove_left(), remove_right(), remove_rows(), remove_top(), replace_dec()
Examples

```r
a_tibble <-
  pt_ex |> |
  remove_top(1) |> |
  define_labels(n_col = 2, n_row = 2) |> |
  unpivot(include_page = FALSE)

a_tibble <-
  pt_ex |> |
  set_page(1, 1) |> |
  remove_top(1) |> |
  define_labels(n_col = 2, n_row = 2) |> |
  remove_k() |> |
  replace_dec() |> |
  fill_values() |> |
  fill_labels() |> |
  remove_agg() |> |
  unpivot()
```
## Index

### *datasets*
- `df_ex`, 4
- `df_ex_compact`, 4
- `df_pivottabler`, 5
- `df_set_h`, 5
- `df_set_h_v`, 6
- `df_set_v`, 6
- `ft_ex`, 12
- `ft_ex_v2`, 12
- `ft_set`, 13
- `pf_ex_compact`, 16
- `pt_ex`, 18

### *flat table list functions*
- `divide`, 7
- `flatten_table_list`, 11
- `get_col_values`, 14

### *flat table*
- `ft_ex`, 12
- `ft_ex_v2`, 12
- `ft_set`, 13

### *import functions*
- `read_excel_file`, 18
- `read_excel_folder`, 19
- `read_excel_sheet`, 21
- `read_text_file`, 22
- `read_text_folder`, 23

### *pivot table definition functions*
- `define_labels`, 3
- `get_page`, 15
- `pivot_table`, 17
- `set_page`, 34

### *pivot table in data frame*
- `df_ex`, 4
- `df_ex_compact`, 4
- `df_pivottabler`, 5

### *pivot table set in data frame*
- `df_set_h`, 5
- `df_set_h_v`, 6
- `df_set_v`, 6

### *pivot table transformation functions*
- `extract_labels`, 8
- `fill_labels`, 9
- `fill_values`, 10
- `remove_agg`, 24
- `remove_bottom`, 25
- `remove_cols`, 26
- `remove_empty`, 27
- `remove_k`, 28
- `remove_left`, 29
- `remove_right`, 30
- `remove_rows`, 31
- `remove_top`, 32
- `replace_dec`, 33
- `unpivot`, 35

### *pivot table*
- `pf_ex_compact`, 16
- `pt_ex`, 18

`define_labels`, 3, 16, 17, 34
`df_ex`, 4, 5, 6, 12, 13, 18
`df_ex_compact`, 4, 5, 16
`df_pivottabler`, 4, 5
`df_set_h`, 5, 6
`df_set_h_v`, 5, 6, 13
`df_set_v`, 5, 6, 6
`divide`, 7, 11, 15, 17

`extract_labels`, 8, 9, 10, 24–33, 35
`fill_labels`, 8, 9, 10, 24–33, 35
`fill_values`, 8, 9, 10, 24–33, 35
`flatten_table_list`, 7, 11, 15
`ft_ex`, 12, 13
`ft_ex_v2`, 12, 12, 13
`ft_set`, 12, 13, 13

`get_col_values`, 7, 11, 14
`get_page`, 5, 15, 17, 34

`pf_ex_compact`, 4, 16, 18
pivot_table, 3, 7–11, 15, 16, 17, 19–35
pt_ex, 4, 16, 18

read_excel_file, 18, 20–23
read_excel_folder, 19, 19, 21–23
read_excel_sheet, 19, 20, 21, 22, 23
read_text_file, 19–21, 22, 23
read_text_folder, 19–22, 23
remove_agg, 8–10, 24, 25–33, 35
remove_bottom, 8–10, 24, 25, 26–33, 35
remove_cols, 8–10, 24, 25, 26, 27–33, 35
remove_empty, 8–10, 24–26, 27, 28–33, 35
remove_k, 8–10, 24–27, 28, 29–33, 35
remove_left, 8–10, 24–28, 29, 30–33, 35
remove_right, 8–10, 24–29, 30, 31–33, 35
remove_rows, 8–10, 24–30, 31, 32, 33, 35
remove_top, 8–10, 24–31, 32, 33, 35
replace_dec, 8–10, 24–32, 33, 35

set_page, 3, 16, 17, 34

unpivot, 8–10, 24–33, 35