Package 'tidyplate'

December 7, 2024

Type Package

Title Transform Microplate Data into Tibbles

Version 2.2.0

Maintainer Shubham Dutta <shubhamdutta26@gmail.com>

Description The goal of 'tidyplate' is to help researchers convert
different types of microplates into tibbles which can be used in data
analysis. It accepts xlsx and csv files formatted in a specific way as
input. It supports all types of standard microplate formats such as
6-well, 12-well, 24-well, 48-well, 96-well, 384-well, and, 1536-well
plates.

License MIT + file LICENSE

URL <https://www.shubhamdutta.com/tidyplate/>,
<https://github.com/shubhamdutta26/tidyplate>

BugReports <https://github.com/shubhamdutta26/tidyplate/issues>

Imports data.table, lifecycle, openxlsx, readxl, rlang, tibble, tools,
utils

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

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 7.3.2

NeedsCompilation no

Author Shubham Dutta [aut, cre, cph] (<<https://orcid.org/0000-0001-8484-0717>>)

Repository CRAN

Date/Publication 2024-12-07 00:20:03 UTC

Contents

---

build_plate	*Build a csv or xlsx template for each plate type*

Description

build_plate() helps the user build an empty csv or xlsx file that can be used as a template for storing plate data. Ensure that plate names are unique. Once populated it can be used as an input for tidy_plate() function.

Usage

```
build_plate(
  plate_type = 6,
  n_plates = 1,
  plate_names = NULL,
  file = NULL,
  file_type = NULL
)
```

Arguments

plate_type	A specific integer (6, 12, 24, 48, 96, 384, or 1536) indicating the type of microwell plate.
n_plates	A positive integer indicating the number of plates.
plate_names	A character vector of unique values that will be assigned to each plate. Its length should be equal to the value of n_plates.
file	A character string naming the file.
file_type	**[Deprecated]** A character string of the output file type. It can either be a csv or xlsx file. The default is csv.

Value

A csv or xlsx template file.

See Also

[tidy_plate()](), [generate_plate()]()

Examples

```
temp_file <- tempfile(fileext = ".csv")

build_plate(plate_type = 6, n_plates = 2, file = temp_file)
```

---

check_plate	*Checks whether the input file is valid for use with the* tidy_plate() *function*

Description

check_plate() performs quality checks on the input microwell shaped data and warns the user if there is any discrepancy. The user can either fix the input file or use the build_plate() function to build a template csv or xlsx file.

Usage

```
check_plate(file, well_id = "well", sheet = 1)
```

Arguments

file	A character string containing the path to a csv or excel file.
well_id	A character string that will be the name for the well id column.
sheet	If file type is xlsx this is the sheet name (character) or number (integer).

Value

An message indicating whether the input file is compatible with the tidy_plate() function

See Also

build_plate()

Examples

```
file_path <- system.file(
  "extdata",
  "example_12_well.xlsx",
  package = "tidyplate"
)

check_plate(file = file_path)
```

---

generate_plate   *Generates a microwell plate shaped csv or excel file from tibble or*
                               dataframe

Description

Dataframes or tibbles are not good for visual inspection of microwell plate shaped data. `generate_plate()`
helps the user by transforming dataframe or tibble into a microwell plate data. It does the opposite
of what `tidy_plate()` does to a plate data.

Usage

```
generate_plate(x, well_id, plate_type, file)
```

Arguments

x	A dataframe or tibble.
well_id	A character string or an integer which points to the column containing the well ids.
plate_type	A specific integer (6, 12, 24, 48, 96, 384, 1536) indicating the type of microwell plate.
file	A character string with the filename of the output file with the path and type of exported file. Only csv or xlsx files are supported.

Value

A csv or xlsx file.

See Also

[tidy_plate(),build_plate()](#)

Examples

```
file_path <- system.file("extdata", "tidy_12_well.csv", package = "tidyplate")

tbl <- read.csv(file_path)
temp_file <- tempfile(fileext = ".csv")

plate_12 <- generate_plate(tbl,
                           well_id = "well",
                           plate_type = 12,
                           file = temp_file)
```

---

tidy_plate *Reads and transforms microwell plate to a tibble*

Description

`tidy_plate()` reads a microwell plate shaped csv or excel file and returns a tibble for downstream data analysis. In order to create an template file use the `build_plate()` function.

Usage

```
tidy_plate(file, well_id = "well", sheet = 1)
```

Arguments

`file`	A character string containing the path to a csv or excel file. The format is described below.
`well_id`	A character string that will be the name for the well id column.
`sheet`	A character or integer indicating the excel sheet to be read.

Value

A tibble.

See Also

[build_plate()](#), [generate_plate()](#)

Examples

```
file_path <- system.file("extdata", "example_12_well.xlsx",
  package = "tidyplate"
)

data_12 <- tidy_plate(file = file_path)

head(data_12)
```

---

view_plate_names   *Returns the name of each plate in the file*

Description

view_plate_names() returns the names of all plates in the input file as a character vector. In case of empty or duplicates it diplays a warning and then returns the plate names.

Usage

```
view_plate_names(file, sheet = 1)
```

Arguments

file	This is the path to a xlsx or csv file containing data for the following types of plates: 6, 12, 24, 48, 96, 384, and 1536.
sheet	If file type is xlsx this is the sheet name (character) or number (integer).

Value

A character vector

Examples

```
file_path <- system.file("extdata", "example_12_well.xlsx", package = "tidyplate")

data_12 <- view_plate_names(file = file_path)

data_12
```

Index