

Package: datefixR (via r-universe)

June 28, 2024

Title Standardize Dates in Different Formats or with Missing Data

Version 1.6.1.9000

Description There are many different formats dates are commonly represented with: the order of day, month, or year can differ, different separators ('`-', ``/`, or whitespace) can be used, months can be numerical, names, or abbreviations and year given as two digits or four. 'datefixR' takes dates in all these different formats and converts them to R's built-in date class. If 'datefixR' cannot standardize a date, such as because it is too malformed, then the user is told which date cannot be standardized and the corresponding ID for the row. 'datefixR' also allows the imputation of missing days and months with user-controlled behavior.

License GPL (>= 3)

URL <https://docs.ropensci.org/datefixR/>,
<https://github.com/ropensci/datefixR>

BugReports <https://github.com/ropensci/datefixR/issues>

Depends R (>= 4.1.0)

Imports lifecycle, Rcpp, rlang, stringr

Suggests DT, htmltools, knitr, parsedate, pkgbuild, png, readr, readxl, rmarkdown, shiny, shinytest2, spelling, testthat (>= 3.0.0), withr

LinkingTo Rcpp

VignetteBuilder knitr

Config/testthat/edition 3

Config/testthat/parallel true

Encoding UTF-8

Language en-US

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3
Repository https://nathansam.r-universe.dev
RemoteUrl https://github.com/ropensci/datefixR
RemoteRef HEAD
RemoteSha 39f1a6706aa768cd8a2fb9da49d0f4d8ef325a41

Contents

exampledates	2
fix_date_app	3
fix_date_char	3
fix_date_df	5
Index	7

exampledates	<i>Example dataset of dates in different formats</i>
--------------	--

Description

A toy dataset to use with datefixR functions.

Usage

exampledates

Format

- A data frame with 5 rows and 3 variables:
- id** Row ID (numeric).
 - some.dates** Dates in different formats (character).
 - some.more.dates** Additional dates in different formats (character).

fix_date_app

*Shiny application standardizing date data in csv of excel files***Description**

A shiny application which allows users to standardize dates using a graphical user interface (GUI). Most features of datefixR are supported including imputing missing date data. Data can be provided as CSV (comma-separated value) or XLSX (Excel) files. Processed datasets can be downloaded as CSV files. Please note, the dependencies for this app (DT, htmltools, readxl, and shiny) are not installed alongside datefixR. This allows datefixR to be installed on secure systems where these packages may not be allowed. If one of these dependencies is not installed on the system when this function is called, then the user will have the option of installing them.

Usage

```
fix_date_app(theme = "datefixR")
```

Arguments

theme	Color theme for shiny app. Either "datefixR" (datefixR colors) or "none" (default shiny app styling).
-------	---

Value

A shiny app.

See Also

The [shiny](#) package.

Examples

```
## Not run:
fix_date_app()

## End(Not run)
```

fix_date_char

*Convert improperly formatted date to R's Date class***Description**

Converts a character vector (or single character object) from improperly formatted dates to R's Date class. Supports numerous separators including /, -, or white space. Supports all-numeric, abbreviation or long-hand month notation. Where day of the month has not been supplied, the first day of the month is imputed by default. Either DMY or YMD is assumed by default. However, the US system of MDY is supported via the format argument.

Usage

```
fix_date_char(
  dates,
  day.impute = 1,
  month.impute = 7,
  format = "dmy",
  excel = FALSE,
  roman.numeral = FALSE
)
```

Arguments

dates	Character vector to be converted to R's date class.
day.impute	Integer. Day of the month to be imputed if not available. defaults to 1. Maximum value of 31. If day.impute is greater than the number of days for a given month, then the last day of that month will be imputed. If day.impute = NA, then NA will be imputed for the date instead and a warning will be raised. If day.impute = NULL then instead of imputing the day of the month, the function will fail.
month.impute	Integer. Month to be imputed if not available. Defaults to 7 (July). If month.impute = NA then NA will be imputed for the date instead and a warning will be raised. If month.impute = NULL then instead of imputing the month, the function will fail.
format	Character. The format which a date is mostly likely to be given in. Either "dmy" (default) or "mdy". If year appears to have been given first, then YMD is assumed for the subject (format argument is not used for these observations)
excel	Logical. If a date is given as only numbers (no separators), and is more than four digits, should the date be assumed to be from Excel which counts the number of days from 1900-01-01? In most programming languages (including R), days are instead calculated from 1970-01-01 and this is the default for this function (excel = FALSE)
roman.numeral	[Experimental] Logical. If TRUE, months detected to have been given as Roman numerals will be converted. Months are given in Roman numerals in some database systems and biological records. Defaults to FALSE as this may occasionally interfere with months in other formats.

Value

A vector of elements belonging to R's built in Date class with the following format yyyy-mm-dd.

See Also

[fix_date_df](#) which is similar to `fix_date_char()` except is applicable to columns of a data frame.

Examples

```
bad.date <- "02 03 2021"
```

```
fixed.date <- fix_date_char(bad.date)
fixed.date
```

fix_date_df

Clean up messy date columns

Description

Tidies a dataframe object which has date columns entered via a free-text box (possibly by different users) and are therefore in a non-standardized format. Supports numerous separators including /,-, or space. Supports all-numeric, abbreviation, or long-hand month notation. Where day of the month has not been supplied, the first day of the month is imputed. Either DMY or YMD is assumed by default. However, the US system of MDY is supported via the format argument.

Usage

```
fix_date_df(
  df,
  col.names,
  day.impute = 1,
  month.impute = 7,
  id = NULL,
  format = "dmy",
  excel = FALSE,
  roman.numeral = FALSE
)
```

Arguments

df	A dataframe or tibble object with messy date column(s)
col.names	Character vector of names of columns of messy date data
day.impute	Integer. Day of the month to be imputed if not available. defaults to 1. Maximum value of 31. If day.impute is greater than the number of days for a given month, then the last day of that month will be imputed. If day.impute = NA, then NA will be imputed for the date instead and a warning will be raised. If day.impute = NULL then instead of imputing the day of the month, the function will fail.
month.impute	Integer. Month to be imputed if not available. Defaults to 7 (July). If month.impute = NA then NA will be imputed for the date instead and a warning will be raised. If month.impute = NULL then instead of imputing the month, the function will fail.
id	Name of column containing row IDs. By default, the first column is assumed.
format	Character. The format which a date is mostly likely to be given in. Either "dmy" (default) or "mdy". If year appears to have been given first, then YMD is assumed for the subject (format argument is not used for these observations)

excel	Logical. If a date is given as only numbers (no separators), and is more than four digits, should the date be assumed to be from Excel which counts the number of days from 1900-01-01? In most programming languages (including R), days are instead calculated from 1970-01-01 and this is the default for this function (excel = FALSE)
roman.numeral	[Experimental] Logical. If TRUE, months detected to have been given as Roman numerals will be converted. Months are given in Roman numerals in some database systems and biological records. Defaults to FALSE as this may occasionally interfere with months in other formats.

Value

A dataframe or tibble object. Dependent on the type of df. Selected columns are of type Date with the following format yyyy-mm-dd

See Also

[fix_date_char](#) which is similar to fix_date_df() except can only be applied to character vectors.

Examples

```
data(exampledates)
fixed.df <- fix_date_df(exampledates, c("some.dates", "some.more.dates"))
fixed.df
```

Index

* **datasets**

 exampledates, [2](#)

exampledates, [2](#)

fix_date_app, [3](#)

fix_date_char, [3](#), [6](#)

fix_date_df, [4](#), [5](#)

shiny, [3](#)