

Package ‘fy’

May 16, 2019

Title Utilities for Financial Years

Version 0.1.0

Description In Australia, a financial year (or fiscal year) is the period from 1 July to 30 June of the following calendar year. As such, many databases need to represent and validate financial years efficiently. While the use of integer years with a convention that they represent the year ending is common, it may lead to ambiguity with calendar years. On the other hand, string representations may be too inefficient and do not easily admit arithmetic operations. This package tries to make validation of financial years quicker while retaining clarity.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Depends R (>= 3.1.0)

Imports fastmatch, data.table, hutils, utils

Suggests testthat, withr, rlang, zoo, covr

NeedsCompilation no

Author Hugh Parsonage [aut, cre]

Maintainer Hugh Parsonage <hugh.parsonage@gmail.com>

Repository CRAN

Date/Publication 2019-05-16 16:00:03 UTC

R topics documented:

is_fy	2
next_fy	3
validate_fys_permitted	3

Index	5
--------------	----------

is_fy

*Convenience functions for dealing with financial years***Description**

Convenience functions for dealing with financial years

Usage

```
yr2fy(yr_ending, assume1901_2100 = .getOption("fy.assume1901_2100",
  .getOption("grattan.assume1901_2100", TRUE)))
```

```
.yr2fy(yr_ending)
```

```
fy2yr(x, na_error = TRUE)
```

```
fy2date(x, na_error = TRUE)
```

```
date2fy(date)
```

```
qtr2fy(yq)
```

Arguments

yr_ending	An integer representing a year.
assume1901_2100	For yr2fy, assume that yr_ending is between 1901 and 2100, for performance. By default, set to <code>getOption("fy.assume1901_2100", TRUE)</code> .
x, fy.yr	A character vector suspected to be a financial year.
na_error	If an input expects a financial year and na_error is TRUE then the function exits with an error.
date	A string or date for which the financial year is desired. Note that yr2fy does not check its argument is an integer.
yq	A character vector representing year quarters in 1066-Q2 format.

Details

The following forms are permitted: 2012-13, 201213, 2012 13, only. However, the 2012-13 form is preferred and will improve performance.

Value

For is_fy, a logical, whether its argument is a financial year. The following forms are allowed: 2012-13, 201213, 2012 13, only. For fy.year, yr2fy, and date2fy, the financial year. For the inverses, a numeric corresponding to the year.

fy.year is a deprecated alias for yr2fy, the latter is slightly more efficient, as well as more declarative.

fy2yr converts a financial year to the year ending: fy2yr("2016-17") returns 2017. yr2fy is the inverse: yr2fy(fy2yr("2016-17")) == "2016-17".

fy2date converts a financial year to the 30 June of the financial year ending.

date2fy converts a date to the corresponding financial year.

Examples

```
is_fy("2012-13")
is_fy("2012-14")
yr2fy(2012)
fy2yr("2015-16")
date2fy("2014-08-09")
```

next_fy	<i>Next and previous financial years</i>
---------	--

Description

Next and previous financial years

Usage

```
next_fy(fy, h = 1L)
```

```
prev_fy(fy, h = 1L)
```

Arguments

fy	A financial year as a character vector.
h	An integer, the "horizon" to go forward (for next_fy) or backward (for prev_fy).

validate_fys_permitted	<i>Verifying validity of financial years</i>
------------------------	--

Description

Many functions expect financial years. Determining that they are validly entered is often quite computationally costly, relative to the core calculations. These internal functions provide mechanisms to check validity quickly, while still providing clear, accurate error messages.

Usage

```
validate_fys_permitted(to_verify, permitted_fys = NULL, min.yr = NULL,
  max.yr = NULL, deparsed = deparse(substitute(to_verify)),
  allow.projection = TRUE,
  earliest_permitted_financial_year = "earliest permitted financial year",
  latest_permitted_financial_year = "latest permitted financial year")
```

Arguments

`to_verify` A user-provided value, purporting to be character vector of financial years.

`permitted_fys` A character vector of valid financial years.

`min.yr`, `max.yr` Integers specifying the range of `to_verify`. If `NULL`, no restriction on the upper or lower bound of the range.

`deparsed` A string indicating the argument that the user provided. Should generally be provided explicitly as the default is unlikely to be user-friendly.

`allow.projection`
If `FALSE` emit a different error message.

`earliest_permitted_financial_year`, `latest_permitted_financial_year`
Text for earliest/latest permitted financial year when `min.yr`/`max.yr` condition is violated.

Value

If `to_verify` contains valid financial years they are returned all in the form 2013-14. If they were already in that form, they obtain the following attributes:

`fy_all_fy` `TRUE` if all the financial years are valid.

`fy_min_yr` An integer, the earliest year ending in `to_verify`.

`fy_max_yr` An integer, the latest year ending in `to_verify`.

Index

[.yr2fy \(is_fy\), 2](#)

[date2fy \(is_fy\), 2](#)

[fy.year \(is_fy\), 2](#)

[fy2date \(is_fy\), 2](#)

[fy2yr \(is_fy\), 2](#)

[is_fy, 2](#)

[next_fy, 3](#)

[prev_fy \(next_fy\), 3](#)

[qtr2fy \(is_fy\), 2](#)

[validate_fys_permitted, 3](#)

[yr2fy \(is_fy\), 2](#)