Package 'fy'

October 13, 2022

Title Utilities for Financial Years

Version 0.3.0

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 7.1.1

Depends R (>= 3.1.0)

Imports fastmatch, data.table, hutils, utils

Suggests testthat (>= 2.1.0), withr, rlang, zoo, covr

NeedsCompilation no

Author Hugh Parsonage [aut, cre]

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

Repository CRAN

Date/Publication 2020-09-15 06:20:03 UTC

R topics documented:

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, validate = TRUE)

fy2date(x, validate = 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 getOption("fy.assume1901_2100", TRUE).
x	A character vector suspected to be a financial year.
validate	If TRUE, the default, inputs that are expected to be financial years are first validated. Validation should be very fast, though some use-cases may require this be skipped.
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

See valid-fys for allowed forms of x.

Value

For `is_fy`, a logical, whether its argument is a financial year. The following forms are allowed: `2012-13`, `201213`, `2012 13`, as well as `2012<dash>13` for some dash symbols. For `fy.year`, `yr2fy`, and `date2fy`, the financial year. For the inverses, a numeric corresponding to the year.

`fy.year` was an alias for `yr2fy`, and is now defunct.

`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` *Next and previous financial years*

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 *Verifying validity of financial years*

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",
  .retain_fmatches = FALSE
)
```

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.
`.retain_fmatches`	If `TRUE`, the function may retain an attribute `fy_fmatches` an integer vector of the matches against the financial years `"1900-01"` to `"2099-00"`. A trade-off between memory and runtime from not recalculating matches.

Details

The preferred form is `"2012-13"`, and this function returns all elements of `to_verify` in this form. That is, it does not preserve the input form.

Other forms that are recognized (and converted) are:

- "201213"
- "2012 13"
- "2012\u201113"
- "2012\u201213"
- "2012\u201313"
- "2012\u201413"
- "2012-2013"

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`.

`fy_fmatches` An integer vector, the matches with the prebuilt financial years.

Benchmarks

```
x <- rep_len(yr2fy(2004L), 1e9)
bench::system_time(validate_fys_permitted(x))
#> process    real
#>  3.578s  3.576s
x <- rep_len(yr2fy(1980:2016), 1e9)

bench::system_time(validate_fys_permitted(x))
#> process    real
#>  3.766s  3.753s
```

Index