Package ‘ThomasJeffersonUniv’

April 21, 2024

Type Package

Title Handy Tools for TJU/TJUH Employees

Version 0.1.1

Date 2024-04-21

Description Functions for admin needs of employees of
Thomas Jefferson University and Thomas Jefferson
University Hospital, Philadelphia, PA.

License GPL-2

Encoding UTF-8

Imports zoo, timeDate, lubridate

Language en-US

Depends R (>= 4.3.0)

RoxygenNote 7.3.1

NeedsCompilation no

Author Tingting Zhan [aut, cre, cph] (<https://orcid.org/0000-0001-9971-4844>)

Maintainer Tingting Zhan <tingtingzhan@gmail.com>

Repository CRAN

Date/Publication 2024-04-21 13:52:35 UTC

R topics documented:

`asDifftime` *Create Time Differences, Extended*

Description

To create difftime object ?with additional time units `'months'` and `'years'`.

Usage

```
asDifftime(
  tim,
  units = names(timeUnits()),
 negative_do = stop(sQuote(deparse1(substitute(tim))), " has negative value!"),
  ...
)
```

Arguments

`tim`	numeric or difftime object, similar usage as in function as.difftime
`units`	character scalar, similar usage as in function as.difftime, but with additional options `'months'` and `'years'`
`negative_do`	exception handling if input `tim` contains negative element(s). Default is an error stop
`...`	additional parameters, currently not in use

Details

Function asDifftime improves function as.difftime in terms that

- If input `tim` is a difftime object, function units_difftime<- is called and the unit of `tim` is changed. In function as.difftime, `tim` is returned directly, i.e., `units` is ignored
- Additional time units `'months'` and `'years'` are supported, in addition to `'secs'`, `'mins'`, `'hours'`, `'days'`, `'weeks'` supported in function as.difftime. Moreover, partial matching (i.e., function match.arg) is called, while function as.difftime requires exact matching.
- End user may choose to stop if `tim` contains negative values. Function as.difftime does not check for negative `tim`.

Value

Function asDifftime returns a difftime object.

Note

Potential name clash as_difftime

checkCount *checkCount*

Description

..

Usage

```
checkCount(x)
```

Arguments

x	logical vector

Value

Function checkCount returns a character scalar.

checkDuplicated *checkDuplicated*

Description

..

Usage

```
checkDuplicated(data, f, file_duplicated = tempfile(fileext = ".txt"), ...)
```

Arguments

data	data.frame
f	formula
file_duplicated	character scalar
...	additional parameters, currently not in use

Value

Function checkDuplicated returns a data.frame.

Examples

```
(d1 = data.frame(A = c(1, 1), B = c(NA_character_, 'text')))


(d2 = data.frame(A = c(1, 2), B = c(NA_character_, 'text')))
```

---

matchDF *Match Rows of data.frames*

Description

To match the rows of one data.frame to the rows of another data.frame.

Usage

```
matchDF(
  x,
  table = unique.data.frame(x),
  by = names(x),
  by.x = character(),
  by.table = character()
)
```

Arguments

`x`	data.frame, the rows of which to be matched.
`table`	data.frame, the rows of which to be matched *against*.
`by`	character scalar or vector
`by.x, by.table`	character scalar or vector

Value

Function matchDF returns a integer vector

Examples

```
DF <- swiss[sample(nrow(swiss), size = 100, replace = TRUE), ]
matchDF(DF)
```

mergeDF *An Alternative Merge Operation*

Description

..

Usage

```
mergeDF(e1, e2, by = character(), by1 = character(), by2 = character())
```

Arguments

`e1` data.frame, on which new columns will be added. All rows of `e1` will be retained in the returned object, *in their original order*.

`e2` data.frame, columns of which will be added to `e1`. Not all rows of `e2` will be included in the returned object

`by` character scalar or vector

`by1, by2` character scalar or vector

Value

Function mergeDF returns a data.frame.

Note

We avoid merge.data.frame as much as possible, because it's slow and even `sort = FALSE` may not completely retain the original order of input `x`.

Examples

```
# examples inspired by ?merge.data.frame

(authors = data.frame(
 surname = c('Tukey', 'Venables', 'Tierney', 'Ripley', 'McNeil'),
 nationality = c('US', 'Australia', 'US', 'UK', 'Australia'),
 deceased = c('yes', rep('no', 4))))
(books = data.frame(
 name = c('Tukey', 'Venables', 'Tierney', 'Ripley',
  'Ripley', 'McNeil', 'R Core', 'Diggle'),
 title = c(
  'Exploratory Data Analysis',
  'Modern Applied Statistics',
  'LISP-STAT', 'Spatial Statistics', 'Stochastic Simulation',
  'Interactive Data Analysis', 'An Introduction to R',
  'Analysis of Longitudinal Data'),
 other.author = c(
  NA, 'Ripley', NA, NA, NA, NA, 'Venables & Smith',
  'Heagerty & Liang & Scott Zeger')))

(m = mergeDF(books, authors, by1 = 'name', by2 = 'surname'))
attr(m, 'nomatch')
```

phone10 *10-digit US phone number*

Description

..

Usage

```
phone10(x, sep = "")
```

Arguments

x	character vector
sep	character scalar

Details

Function phone10 converts all US and Canada (+1) phone numbers to 10-digit.

Value

Function phone10 returns a character vector of nchar-10.

Examples

```
x = c('+16108885188', '15108581787', '8588889426')
phone10(x)
phone10(x, sep = '-')
```

sourcePath *Source All R Files under a Directory*

Description

source all `*.R` and `*.r` files under a directory.

Usage

```
sourcePath(path, ...)
```

Arguments

`path`	character scalar, parent directory of `.R` files
`...`	additional parameters of source

Value

Function sourcePath does not have a returned value

splitDF *Split data.frame by Row*

Description

split.data.frame into individual rows.

Usage

```
splitDF(x)
```

Arguments

`x`	data.frame

Value

Function splitDF returns a list of nrow-1 data.frames.

Note

We use split.data.frame with argument `f` being `attr(x, which = 'row.names', exact = TRUE)` instead of `seq_len(.row_names_info(x, type = 2L))`, not only because the former is faster, but also .rowNamesDF<- enforces that row.names.data.frame must be unique.

Examples

```
splitDF(head(mtcars)) # data.frame with rownames
splitDF(head(warpbreaks)) # data.frame without rownames
splitDF(data.frame()) # exception
```

`TJU_Cayuse` *Award & Effort from Cayuse*

Description

Print out grant and effort from Cayuse.

Usage

```
aggregateAwards(path = "~/Downloads", fiscal.year = year(Sys.Date()))

viewProposal(path = "~/Downloads", fiscal.year = year(Sys.Date()))

viewAward(path = "~/Downloads")

award2LaTeX(path = "~/Downloads")
```

Arguments

`path`	character scalar, directory of downloaded award `.csv` file. Default is the download directory `'~/Downloads'`
`fiscal.year`	integer scalar

Details

- go to `https://jefferson.cayuse424.com/sp/index.cfm`
- My Proposals -> Submitted Proposals. Lower-right corner of screen, 'Export to CSV'. Downloaded file has name pattern `'^proposals_.*\\.csv'`
- My Awards -> Awards (*not* 'Active Projects'). Lower-right corner of screen, 'View All', then 'Export to CSV'. Downloaded file has name pattern `'^Awards_.*\\.csv'`
- My Awards -> Awards. Click into each project, under 'People' tab to find my 'Sponsored Effort'

Function `aggregateAwards()` aggregates grant over different period (e.g. from Axx-xx-001, Axx-xx-002, Axx-xx-003 to Axx-xx). Then we need to manually added in our 'Sponsored Effort' in the returned `.csv` file.

Value

..

Examples

```
if (FALSE) {
aggregateAwards()
viewAward()
viewProposal()
award2LaTeX()
}
```

`TJU_Fiscal_Year` *TJU Fiscal Year*

Description

..

Usage

```
TJU_Fiscal_Year(x)
```

Arguments

`x`	integer scalar

Value

Function `TJU_Fiscal_Year()` returns a length-two Date vector, indicating the start (July 1 of the previous calendar year) and end date (June 30) of a fiscal year.

Examples

```
TJU_Fiscal_Year(2022L)
```

`TJU_SchoolTerm` *TJU School Term*

Description

..

Usage

```
TJU_SchoolTerm(x)
```

Arguments

`x`	Date object

Value

TJU_SchoolTerm returns a character vector

Examples

```
TJU_SchoolTerm(as.Date(c('2021-03-14', '2022-01-01', '2022-05-01')))
```

`TJU_Workday` *Thomas Jefferson University Workdays*

Description

To summarize the number of workdays, weekends, holidays and vacations in a given time-span (e.g., a month or a quarter of a year).

Usage

```
TJU_Workday(x, vacations)
```

Arguments

`x`	character scalar or vector (e.g., `'2021-01'` for January 2021, `'2021 Q1'` for 2021 Q1 (January to March)), or integer scalar or vector (e.g., `2021L` for year 2021); The time-span to be summarized. Objects of classes yearqtr and yearmon are also accepted.
`vacations`	Date vector, vacation days

Details

Function `TJU_Workday()` summarizes the workdays, weekends, Jefferson paid holidays (New Year's Day, Martin Luther King, Jr. Day, Memorial Day, Fourth of July, Labor Day, Thanksgiving and Christmas) and your vacation (e.g., sick, personal, etc.) days (if any), in a given time-span.

Per Jefferson policy (source needed), if a holiday is on Saturday, then the preceding Friday is considered to be a weekend day. If a holiday is on Sunday, then the following Monday is considered to be a weekend day.

Value

Function `TJU_Workday()` returns a factor.

Examples

```
table(TJU_Workday(c('2021-01', '2021-02')))

tryCatch(TJU_Workday(c('2019-10', '2019-12')), error = identity)
table(c(TJU_Workday('2019-10'), TJU_Workday('2019-12'))) # work-around

table(TJU_Workday('2022-12'))

table(TJU_Workday('2022 Q1', vacations = seq.Date(
 from = as.Date('2022-03-14'), to = as.Date('2022-03-18'), by = 1)))

table(TJU_Workday('2022 Q2', vacations = as.Date(c(
 '2022-05-22', '2022-05-30', '2022-06-01', '2022-07-04'))))

table(TJU_Workday(2021L))
```

---

zip5 *5-digit US Zip Code*

Description

..

Usage

```
zip5(x)
```

Arguments

x	character vector

Details

Function zip5 converts all US zip codes to 5-digit.

Value

Function zip5 returns a character vector of nchar-5.

Index