# Package 'myRtools'

March 31, 2022

**Type** Package

**Title** Just a package form my general R functions

**Version** 0.3.0

**Author** Thanasis N

**Maintainer** Thanasis N <lapauththanasis@gmail.com>

**Description** These can do many interesting things

**License** GPL-3

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.2

**Imports** arrow, filelock, funr, gdata, pander

## R topics documented:

---

`celsius_to_fahrenheit` *Convert Celsius to Fahrenheit*

---

## Description

Convert Celsius to Fahrenheit

## Usage

```
celsius_to_fahrenheit(T.celsius)
```

## Arguments

| | |
|---|---|
| `T.celsius` | Temperature in Celsius |

## Value

Temperature in Fahrenheit

---

`celsius_to_kelvin` *Convert Celsius to kelvin*

---

## Description

Convert Celsius to kelvin

## Usage

```
celsius_to_kelvin(T.celsius)
```

## Arguments

| | |
|---|---|
| `T.celsius` | Temperature in Celsius |

## Value

Temperature in kelvin

## clean_names *Clean names of an object*

### Description

Clean names of an object

### Usage

```
clean_names(df)
```

### Arguments

| | |
|---|---|
| df | An object that has ‘names()‘ |

### Value

The same object with cleaner names

### Note

This will remove all spaces and dots from files names and will substitute them with underscores.

## cosde *Calculate the cosine of an angle given in degrees*

### Description

Calculate the cosine of an angle given in degrees

### Usage

```
cosde(degrees)
```

### Arguments

| | |
|---|---|
| degrees | Input angle in degrees |

### Value

cos(degrees)

### See Also

Other math: sinde(), tande()

## csR *Rayleigh scattering cross section*

### Description

Rayleigh scattering cross section

### Usage

```
csR(lamda_nm)
```

### Arguments

| | |
|---|---|
| lamda_nm | light wavelength in nanometers |

### Value

Absorption coefficient at wavelength

### Note

by: Hansen - Travis (1974)

## fahrenheit_to_celsius *Convert Fahrenheit to Celsius*

### Description

Convert Fahrenheit to Celsius

### Usage

```
fahrenheit_to_celsius(T.fahrenheit)
```

### Arguments

| | |
|---|---|
| T.fahrenheit | Temperature in Fahrenheit |

### Value

Temperature in Celsius

## fmt_difftime *Get a nice string for the difference of two dates*

### Description

Get a nice string for the difference of two dates

### Usage

```
fmt_difftime(etime, stime, type)
```

### Arguments

| | |
|---|---|
| etime | (POSIX date) The end time. |
| stime | (POSIX date) The start time. |
| type | The output format. "h" "93:29:43.758", "d" "3 21:34:17.072", "s" "336936.990" |

### Details

The difference of dates is computed with difftime in seconds.

### Value

A string of time duration formatted as time

## fmt_secs *Get a nice string for seconds as formatted time*

### Description

Get a nice string for seconds as formatted time

### Usage

```
fmt_secs(seconds, type)
```

### Arguments

| | |
|---|---|
| seconds | Time duration in seconds |
| type | The output format. "h" "93:29:43.758", "d" "3 21:34:17.072", "s" "336936.990" |

### Value

A string of time duration formatted as time

## InHg_to_mbar *Convert pressure from InHg to mbar*

### Description

Convert pressure from InHg to mbar

### Usage

```
InHg_to_mbar(Inches_Hg)
```

### Arguments

| | |
|---|---|
| Inches_Hg | Inches of Hg |

### Value

Pressure in millibars

## InHg_to_Pa *Convert pressure from InHg to Pascals*

### Description

Convert pressure from InHg to Pascals

### Usage

```
InHg_to_Pa(Inches_Hg)
```

### Arguments

| | |
|---|---|
| Inches_Hg | Inches of Hg |

### Value

Pressure in pascals

## In_to_mm *Convert inches to millimeters*

### Description

Convert inches to millimeters

### Usage

```
In_to_mm(Inches)
```

### Arguments

| | |
|---|---|
| Inches | Length in inches |

### Value

Length in millimeters

## MpH_to_mps *Convert Miles per hour to meters per second*

### Description

Convert Miles per hour to meters per second

### Usage

```
MpH_to_mps(Miles_per_hour)
```

### Arguments

| | |
|---|---|
| Miles_per_hour | Speed in Miles per hour |

### Value

Speed in meters per second

## nearest *Find nearest numbers between vectors.*

### Description

Return an array ‘i‘ of indexes into ‘target‘, parallel to array ‘probe‘. For each index ‘j‘ in ‘target‘, ‘probe[i[j]]‘ is nearest to ‘target[j]‘. From: https://stats.stackexchange.com/questions/161379/quickly-finding-nearest-time-observation

Used to get timestamps close to each other between data sets or timeseries.

### Usage

```
nearest(probe, target, ends = c(-Inf, Inf))
```

### Arguments

| | |
|---|---|
| probe | A vector |
| target | A vector |

### Value

Indexes of ‘target‘ matching ‘probe‘ data.

### Examples

```
## Graphical illustration.
set.seed(17)
x <- sort(round(runif(8), 3))
y <- sort(round(runif(12), 1))
i <- nearest(x, y)
plot(c(0,1), c(3/4,9/4), type="n", bty="n", yaxt="n", xlab="Values", ylab="")
abline(v = (y[-1] + y[-length(y)])/2, col="Gray", lty=3)
invisible(apply(rbind(x, y[i]), 2, function(a) arrows(a[1], 1, a[2], 2, length=0.15)))
points(x, rep(1, length(x)), pch=21, bg="Blue")
points(y, rep(2, length(y)), pch=21, bg="Red", cex=sqrt(table(y)[as.character(y)]))
text(c(1,1), c(1,2), c("x","y"), pos=4)
```

## object_metadata *Gather meta data for an R object*

### Description

Gather meta data for an R object

## Usage

```
object_metadata(object, contact = contact, notes = NA)
```

## Arguments

| | |
|---|---|
| object | Object to get metadata for |
| contact | Contact information for the data |
| notes | Notes on the data |

## Value

List with names and values of the metadata

---

# sinde *Calculate the sine of an angle given in degrees*

---

## Description

Calculate the sine of an angle given in degrees

## Usage

```
sinde(degrees)
```

## Arguments

| | |
|---|---|
| degrees | Input angle in degrees |

## Value

sin(degrees)

## See Also

Other math: cosde(), tande()

## std_scriptend_v1 *Standard end for scripts*

### Description

Standard end for scripts

### Usage

```
std_scriptend_v1()
```

### See Also

Other standard scripting: `std_setenv_v1()`

## std_setenv_v1 *Standard starting part in a script*

### Description

Still have to use 'rm(list = (ls()[ls() != ""]))" before this

### Usage

```
std_setenv_v1(ScriptName = "", outdir = "")
```

### Arguments

| | |
|---|---|
| ScriptName | Use this if can not resolve name |
| outdir | Folder for extra files output |

### See Also

Other standard scripting: `std_scriptend_v1()`

## tande *Calculate the tan of an angle given in degrees*

### Description

Calculate the tan of an angle given in degrees

### Usage

```
tande(degrees)
```

### Arguments

| | |
|---|---|
| degrees | Input angle in degrees |

### Value

tan(degrees)

### See Also

Other math: `cosde()`, `sinde()`

## writeDATA *Save data with multiple output format and extra metadata*

### Description

Save data with multiple output format and extra metadata

### Usage

```
writeDATA(
  object,
  file,
  contact = "<lapauththanasis@gmail.com>",
  notes = NA,
  clean = FALSE,
  type = c("Rds")
)
```

### Arguments

| | |
|---|---|
| object | Object with data (not all types will work) |
| file | Prefix of output file. The extension will be added by the type |
| contact | Contact information of data |
| notes | Notes on the data |
| clean | If 'TRUE' don't write .inf.md files |
| type | A vector of types to write. One of c("Rds","dat","prqt") |

### Value

Writes files with data and a corresponding info (md) file.

### See Also

Other write functions: write_RDS(), write_dat(), write_prqt()

---

## write_dat *Default method to write dat files of data*

---

### Description

Default method to write dat files of data

### Usage

```
write_dat(
  object,
  file,
  contact = "<lapauththanasis@gmail.com>",
  notes = NA,
  clean = FALSE
)
```

### Arguments

| | |
|---|---|
| object | Object with data |
| file | File to write to |
| contact | Contact information of data |
| notes | Notes on the data |
| clean | If 'TRUE' don't write .inf.md files |

### Value

Writes a file with data and a corresponding info (md) file.

### Note

It uses the command writeDATA(object = object, file = file, contact = contact, notes = notes, clean = clean, type = "dat")

### See Also

Other write functions: `writeDATA()`, `write_RDS()`, `write_prqt()`

---

`write_prqt` *Default method to write parquet files of data with arrow*

---

### Description

Default method to write parquet files of data with arrow

### Usage

```
write_prqt(
  object,
  file,
  contact = "<lapauththanasis@gmail.com>",
  notes = NA,
  clean = FALSE
)
```

### Arguments

| | |
|---|---|
| `object` | Object with data |
| `file` | File to write to |
| `contact` | Contact information of data |
| `notes` | Notes on the data |
| `clean` | If ‘TRUE‘ don’t write .inf.md files |

### Value

Writes a file with data and a corresponding info (md) file.

### Note

It uses the command writeDATA(object = object, file = file, contact = contact, notes = notes, clean = clean, type = "prqt")

### See Also

Other write functions: `writeDATA()`, `write_RDS()`, `write_dat()`

write_RDS *Write a data frame to an Rds file with extra info*

## Description

Write a data frame to an Rds file with extra info

## Usage

```
write_RDS(
  object,
  file,
  contact = "<lapauththanasis@gmail.com>",
  notes = NA,
  clean = FALSE
)
```

## Arguments

| | |
|---|---|
| object | A data frame to be saved. |
| file | A file name to use without the extension. |
| contact | Contact information for the data set. |
| notes | Notes on the data |
| clean | If 'TRUE' don't write .inf.md files |

## Details

Writes an .Rds file using saveRDS with xz compression and an companion .inf.md file with extra information for the data.

## Note

It uses the command writeDATA(object = object, file = file, contact = contact, notes = notes, clean = clean, type = "Rds")

## See Also

Other write functions: writeDATA(), write_dat(), write_prqt()

# Index