Package 'DataMetProcess'

May 7, 2026

Type Package

Title Meteorological Data Processing

Version 1.0.8

Maintainer Wagner Martins dos Santos <wagnnerms97@gmail.com>

Description Set of tools aimed at processing meteorological data, converting hourly recorded data to daily, monthly and annual data.

License GPL (>= 3)

Encoding UTF-8

URL https://github.com/wagnnerms97/DataMetProcess

BugReports https://github.com/wagnnerms97/DataMetProcess/issues

Depends R (>= 4.1)

RoxygenNote 7.3.3

Imports dplyr, tidyr, lubridate, rlang, utils, base, shiny, stringr

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Wagner Martins dos Santos [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3584-1323>),
Hoi Leong Lee [aut],
Edimir Xavier Leal Ferraz [aut] (ORCID:
<https://orcid.org/0000-0002-3151-8916>),
Abelardo Antônio de Assunção Montenegro [aut],
Lady Daiane Costa de Sousa Martins [aut] (ORCID:
<https://orcid.org/0000-0002-0942-4673>),
Alan Cézar Bezerra [aut],
Ênio Farias de França e Silva [aut],
Thieres George Freire da Silva [aut] (ORCID:
<https://orcid.org/0000-0002-8355-4935>),
João L.M.P. de Lima [aut],
Xuguang Tang [aut],
Alexandre Maniçoba da Rosa Ferraz Jardim [aut] (ORCID:
<https://orcid.org/0000-0001-7094-3635>)

Repository CRAN

Date/Publication 2026-01-20 08:30:14 UTC

Contents

`adjustDate` *Fix the time zone*

Description

Allows you to correct the timezone based on a date column and another time column

Usage

```
adjustDate(data = NULL, col_date = NULL, col_hour = NULL, fuso = NULL)
```

Arguments

`data`	Data frame containing the data
`col_date`	Column containing the dates
`col_hour`	Column containing the time. It must be in the format "hh", "hh:mm", or "hh:mm:ss"; only the hours "hh" will be used for conversion.
`fuso`	Time zone for correction. Query OlsonNames()

Value

Data frame with the corrected timezone

Examples

```
address <-
 base::system.file("extdata",
                   "ex1_inmet.CSV",
                   package = "DataMetProcess")

df <-
  read.table(
    address,
    h=TRUE,
```

```
    sep = ";",
    dec = ",",
    skip = 8,
    na.strings = -9999,
    check.names = FALSE
  )

df$Data = as.Date(df$Data,format = "%d/%m/%Y")


df <-
  adjustDate(df,
             colnames(df)[1],
             colnames(df)[2],
             fuso = "America/Bahia")

head(df[1:2])
```

`calculateDMY` *Calculation of daily, monthly and annual scales*

Description

Performs data processing on an hourly scale for daily, monthly or annual scales

Usage

```
calculateDMY(
  data = NULL,
  col_date = NULL,
  col_sum = NULL,
  col_mean = NULL,
  col_max = NULL,
  col_min = NULL,
  n.round = 2,
  type = c("Daily", "Monthly", "Yearly")
)
```

Arguments

`data`	Data frame containing the data
`col_date`	String with the column of data containing the date (R default date: "%Y-%m-%d")
`col_sum`	String with the column of data to apply the sum process
`col_mean`	String with the column of data to apply the averaging process
`col_max`	String with data column to find maximum

col_min	String with data column to find minimum
n.round	Integer, number of decimal places
type	string, receives "Daily", "Monthly" or "Yearly" ("Daily" default). Defines the scale of processing to be performed

Value

Data frame with the defined scale

Examples

```
address <-
 base::system.file("extdata",
                   "ex1_inmet.CSV",
                   package = "DataMetProcess")

df <-
read.table(
  address,
  h=TRUE,
  sep = ";",
  dec = ",",
  skip = 8,
  na.strings = -9999,
  check.names = FALSE
)

df$Data = as.Date(df$Data,format = "%d/%m/%Y")

df.d <-
  calculateDMY(
    data = df,
    col_date = "Data",
    col_sum = colnames(df)[c(3,7)],
    col_mean = colnames(df)[-c(1,2,3,7)],
    type = "Daily"
  )

df.m <-
  calculateDMY(
    data = df.d,
    col_date = "Date",
    col_sum = colnames(df.d)[c(2)],
    col_mean = colnames(df.d)[-c(1,2)],
    type = "Monthly"
  )

df.a <-
  calculateDMY(
    data = df.m,
    col_date = "Date",
    col_sum = colnames(df.m)[c(2)],
```

```
  col_mean = colnames(df.m)[-c(1,2)],
  type = "Yearly"
)
```

`calculateETrefPM` *FAO Penman-Monteith method for daily reference evapotranspiration*

Description

Calculation of daily reference evapotranspiration using the FAO-56 Penman-Monteith method for a dataset stored in a data.frame.

Usage

```
calculateETrefPM(
  data,
  lat,
  alt,
  za,
  DAP = 1,
  date,
  Ta = NULL,
  Tmin = NULL,
  Tmax = NULL,
  RH = NULL,
  RHmin = NULL,
  RHmax = NULL,
  Rg,
  AP,
  WS,
  G = NULL,
  Kc = NULL
)
```

Arguments

`data`	Data frame containing the data
`lat`	Numeric, latitude in decimal degrees
`alt`	Numeric, altitude in meters
`za`	Numeric, anemometer height in meters
`DAP`	Numeric, days after planting for the first date
`date`	String with the column name containing date records
`Ta`	Optional. String with the column name containing mean air temperature (degC)

Tmin	Optional. String with the column name containing minimum air temperature (degC)
Tmax	Optional. String with the column name containing maximum air temperature (degC)
RH	String with the column name containing mean relative humidity (percent)
RHmin	Optional. String with the column name containing minimum relative humidity (percent)
RHmax	Optional. String with the column name containing maximum relative humidity (percent)
Rg	String with the column name containing global radiation (MJ/m2/day)
AP	String with the column name containing atmospheric pressure (hPa)
WS	String with the column name containing wind speed (m/s)
G	Optional. If NULL, soil heat flux is assumed to be zero (MJ/m2/day)
Kc	Optional. Crop coefficient column name

DMPshiny *Launch DataMetProcess Shiny Application*

Description

The 'DMPshiny' function is used to start the Shiny application of the 'DataMetProcess' package. It allows configuring the host address, port, whether to launch the browser automatically, and the maximum upload size.

Usage

```
DMPshiny(
  host = "127.0.0.1",
  port = NULL,
  launch.browser = TRUE,
  maxUploadSize = 200
)
```

Arguments

host	Character. The host address where the application will run. Default is "127.0.0.1".
port	Integer. The port on which the application will run. If NULL, a random port will be used.
launch.browser	Logical. Indicates whether the browser should be launched automatically. Default is TRUE.
maxUploadSize	Numeric. Maximum upload file size in megabytes. Default is 200.

Details

The function sets Shiny options, such as the maximum upload size, and then runs the Shiny application located in the 'DataMetProcess_Shiny/App.R' directory of the package.

Value

This function does not return a value. It starts the Shiny server and opens the application in the specified browser.

Examples

```
## Not run:
  DMPshiny()

## End(Not run)
```

`list_inmet` *List of data available at INMET by year*

Description

Collects the available files for the year and returns a list containing: 1) a table containing the addresses of each file inside the zip for later extraction by the down_inmet() function, 2) Yearther structured table with the information available in the file name (e,g, city, station code, year, date of start and end date) and 3) the address of the zip file.

Usage

```
list_inmet(year = NULL, filename = NULL)
```

Arguments

`year`	year for download in the INMET database
`filename`	string containing the path and name of the file with the extension ".zip", if NULL (default) it will be saved in a temporary file

Value

List containing: 1) a table containing the addresses of each file inside the zip for later extraction by the unzip() function of the utils package, 2) Yearther structured table with the information available in the file name (e,g, city, station code, year, date of start and end date) and 3) the address of the zip file.

Examples

```
file.down <- tempfile()
file.save <- tempfile()

info.inmet <-
  DataMetProcess::list_inmet(year="2000",file.down)

unzip.file <-
  utils::unzip(
    zipfile = file.down, #or info.inmet$Saved
    exdir = file.save
  )

unzip.file
```

Index