Package 'DataMetProcess'

January 20, 2025

Type Package

Title Meteorological Data Processing

Version 1.0.3

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

RoxygenNote 7.3.2

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

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Author Wagner Martins dos Santos [aut, cre]
(<https://orcid.org/0000-0002-3584-1323>),
Edimir Xavier Leal Ferraz [aut]
(<https://orcid.org/0000-0002-3151-8916>),
Lady Daiane Costa de Sousa Martins [aut]
(<https://orcid.org/0000-0002-0942-4673>)

Repository CRAN

Date/Publication 2024-10-03 17:40:01 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` *The FAO Penman–Monteith for calculating daily reference evapotranspiration*

Description

Calculation of daily reference evapotranspiration using the PM method for a dataset stored in a data.frame (Allen et al., 1998).

Usage

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

Arguments

data	Data frame containing the data
lat	Numeric, latitude in decimals
alt	Numeric, altitude in meters
za	Numeric, anemometer height in meters
DAP	Numeric, days after planting for the first column date
date	String with the column name containing date records (R default date: "%Y-%m-%d")
Ta	String with the column name containing temperature records in °C
G	Optional, if NULL will be considered as zero. String with the column name containing soil heat flux (MJ/m²/day)
RH	String with the column name containing relative humidity records in %
Rg	String with the column name containing global radiation records in MJ/m²
AP	String with the column name containing atmospheric pressure records in hPa
WS	String with the column name containing wind speed records in m/s
Kc	Optional, when not NULL the crop evapotranspiration ETc is calculated based on ETref. String with the column name containing crop coefficient (Kc) records

Details

The FAO Penman–Monteith method:

$$ETrefPM = \frac{0.408\Delta(Rn - G) + \gamma\frac{900}{T+273}u_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

where: ETref - reference evapotranspiration (mm/dia), delta - slope of the saturated water–vapor-pressure curve (kPA/°C), Rn - net radiation (MJ/m²/dia), G - soil heat flux (MJ/m²/day), y - psychrometric constant (kPA/°C), T - average daily air temperature (°C), u2 - wind speed at 2m height (m/s), es - saturation vapor pressure (kPa), e ea - actual vapor pressure (kPa)

Value

Data frame with: date; etref - reference evapotranspiration (mm/dia); dj - julian day; DAP - days after planting; es - saturation vapor pressure (kPa); ea - actual vapor pressure (kPa); delta - slope of the saturated water–vapor-pressure curve (kPA/°C); y - psychrometric constant (kPA/°C); rn - net radiation (MJ/m²/dia); etc - crop evapotranspiration (mm/dia) (depends on supply of Kc)

References

Allen, R.G., Pereira, L.S., Raes, D., Smith, M., 1998. Crop evapotranspiration – guidelines for computing crop water requirements – FAO Irrigation and Drainage Paper 56. FAO, 1998. ISBN 92-5-104219-5.

Examples

```
address <-
 base::system.file("extdata",
                   "ex2_daily.CSV",
                   package = "DataMetProcess")

df <- read.table(
address,
h = TRUE,
sep = ";"
)

#converting to Mj/m
df$radiacao_global_kj_m <- df$radiacao_global_kj_m/1000
colnames(df)[3] <- "radiacao_global_mj_m"

df.Eto <-
  calculateETrefPM(
    data = df,
    lat = -21.980353,
    alt = 859.29,
    za = 10,
    DAP = 1,
    date = colnames(df)[1],
    Ta = colnames(df)[7],
    G = NULL,
```

```
  RH = colnames(df)[15],
  Rg = colnames(df)[3],
  AP = colnames(df)[4],
  WS = colnames(df)[18],
  Kc = NULL
)
```

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