Package: meteor (via r-universe)

September 8, 2024

Type Package

Title Meteorological Data Manipulation

Version 0.4-6

LinkingTo Rcpp

Imports methods, Rcpp (>= 0.12.4)

Suggests terra

Date 2023-08-26

Maintainer Robert J. Hijmans <r.hijmans@gmail.com>

Description A set of functions for weather and climate data manipulation, and other helper functions, to support dynamic ecological modeling, particularly crop and crop disease modeling.

License GPL-3

LazyLoad yes

Repository https://cropmodels.r-universe.dev

RemoteUrl https://github.com/cropmodels/meteor

RemoteRef HEAD

RemoteSha 660add047f644c8059b0cef3ef304b363a8d1c0b

Contents

`meteor-package` *The meteor package*

Description

This pacakges contains of a number of meteorological data manipulation functions. Some of these are also available in other R packages. The context of this package is to make the functions available from dynamic simulation models of crops and crop diseases.

`dailyToHourly` *Estimate hourly values from daily values*

Description

Estimate hourly temperature from daily minimum and maximum temperature, or hourly relative humidity from average relative humidity and minimum and maximum temperature.

The functions require the day of the year and latitude to compute the photoperiod.

Usage

```
hourlyFromDailyTemp(tmin, tmax, doy, latitude)
hourlyFromDailyRelh(relh, tmin, tmax, doy, latitude)
```

Arguments

`tmin`	numeric. minimum temperature (must be in C for hourlyFromDailyRelh)
`tmax`	numeric. maximum temperature (must be in C for hourlyFromDailyRelh)
`relh`	relative humidity (percent)
`doy`	integer. Day of the year (between 1 and 365)
`latitude`	numeric. Latitude

Value

matrix

Examples

```
hourlyFromDailyTemp(c(20,22), c(28,34), c(150,151), 52)
hourlyFromDailyRelh(80, c(20,22), c(28,34), c(150,151), 52)
```

dates *date manipulation*

Description

Helper functions for manipulation of dates, includding conversion between (day of year) (DOY) to date and back, and extraction of parts of a date.

Usage

```
dateFromDoy(doy, year)
doyFromDate(date)
dayFromDate(date)
monthFromDate(date)
yearFromDate(date)
isLeapYear(year)
daysInYear(year)
```

Arguments

doy	integer. Day of the year (1..365) or (1..366) for leap years
year	integer. Year, e.g. 1982
date	Date object or character formatted 'yyyy-mm-dd', e.g. '1982-11-23'

Value

integer or Date

Examples

```
doy <- 88
year <- 1970
date <- dateFromDoy(doy, year)
date
dateFromDoy(-15, 2000)
doyFromDate(date)
isLeapYear(2000)
daysInYear(2000)
daysInYear(1999)
```

dayTemp *Estimate the temperature during the day*

Description

Estimate the mean temperature during the day (between sunrise and sunset) from daily minimum and maximum temperature.

The function requires the day of the year and latitude to compute the photoperiod.

Usage

```
dayTemp(tmin, tmax, doy, latitude)
```

Arguments

tmin	numeric. minimum temperature (any unit)
tmax	numeric. maximum temperature (any unit)
doy	integer. Day of the year (between 1 and 365)
latitude	numeric. Latitude

Value

numeric

Examples

```
dayTemp(c(20,22), c(28,34), c(150,151), 52)
```

evapotranspiration *Reference evapo-transpiration*

Description

Functions to compute the reference evapotranspiration (ET0) from meteorological data. ET0 is a representation of the atmospheric water demand. The equations estimate the evapotranspiration rate of a short green crop (grass), completely shading the ground, of uniform height and with adequate water status in the soil profile. Actual evapotranspiration is equal reference evapotranspiration when there is ample water, but taller crops could have an evapotranspiration rate that is higher than ET0.

Usage

```
ET0_PenmanMonteith(temp, relh, atmp, Rn, G, ra, rs)
ET0_PriestleyTaylor(temp, relh, atmp, Rn, G)
ET0_Makkink(temp, relh, atmp, Rs)
ET0_ThornthwaiteWilmott(temp, doy, latitude)
ET0_ThornthwaiteWilmottCamargo(tmin, tmax, doy, latitude, Pereira=FALSE)
```

Arguments

`temp`	temperature (degrees C)
`relh`	relative humidity (percent)
`atmp`	air pressure (hPa
`Rn`	net radiation (J m-2 day-1)
`Rs`	incoming solar radiation (J m-2 day-1)
`G`	soil heat flux (J m-2 day-1)
`ra`	aerodynamic resistance (s m-1)
`rs`	surface resistance (s m-1)
`doy`	integer. Day of the year (between 1–365)
`latitude`	numeric. Latitude
`tmin`	numeric. minimum temperature (C)
`tmax`	numeric. maximum temperature (C)
`Pereira`	logical. If `TRUE`, the Pereira adjustment for photoperiod is used

Value

vector with evaporation values (mm)

Author(s)

Robert Hijmans, partly based on Python evapolib by Maarten J. Waterloo http://python.hydrology-amsterdam.nl/

References

Allen, R.G., L.S. Pereira, D. Raes and M. Smith, 1998. Crop evapotranspiration. Guidelines for computing crop water requirements. FAO Irrigation and drainage paper 56. FAO - Food and Agriculture Organization of the United Nations, Rome, 1998. (http://www.fao.org/docrep/x0490e/x0490e07.htm)

Thornthwaite, C.W., 1948. An approach toward a rational classification of climate. Geogr. Rev. 38:55-94.

Willmott, C.J., Rowe, C.M. and Mintz, Y., 1985. Climatology of the terrestrial seasonal water cycle. J. Climatol. 5:589-606.

Camargo, A.P., Marin, F.R., Sentelhas, P.C. and Picini, A.G., 1999. Adjust of the Thornthwaite's method to estimate the potential evapotranspiration for arid and superhumid climates, based on daily temperature amplitude. Rev. Bras. Agrometeorol. 7(2):251-257

Pereira, A.R. and W.O. Pruitt, 2004. Adaptation of the Thornthwaite scheme for estimating daily reference evapotranspiration. Agricultural Water Management 66: 251-257

Examples

```
ET0_PenmanMonteith(21.67, 67, 1013, 14100000, 500000, 104, 70)
ET0_PriestleyTaylor(21.65, 67, 1013, 18200000, 600000)
ET0_Makkink(21.65, 67, 1013, 24200000)
```

ExtraTerrestrialRadiation *Extra-terrestrial Radiation*

Description

Compute incoming radiation (J day-1 m-2) at the top of the atmosphere and photoperiod (daylength, sunshine duration).

Usage

```
ExtraTerrestrialRadiation(doy, latitude, sc=1367.7, FAO=FALSE)
```

Arguments

`doy`	integer. Day of the year
`latitude`	numeric. Latitude
`sc`	numeric. The solar constant
`FAO`	logical. If `TRUE` the algorithm described by Allen et al (1998) is used. If `FALSE` the approach by Goudriaan and Van Laar (1995) is used

Value

matrix with incoming radiation (J/day) and

Author(s)

Robert Hijmans, based on Python meteolib by Maarten J. Waterloo and J. Delsman http://python.hydrology-amsterdam.nl/

References

Goudriaan and Van Laar, 1995.

R.G. Allen, L.S. Pereira, D. Raes and M. Smith (1998). Crop Evaporation - Guidelines for computing crop water requirements. Irrigation and drainage paper 56. FAO, Rome, Italy. https://www.fao.org/3/x0490e/x0490e07.htm

Examples

```
ExtraTerrestrialRadiation(50, 60)
ExtraTerrestrialRadiation(50, 60, FAO=TRUE)
```

from *Date manipulation*

Description

Helper functions for extracting information from dates. Or to create a Date from a day number.

Usage

```
fromDate(x, v)
fromYear(y, v)
fromDoy(doy, y)
```

Arguments

x	Date or POSIX
y	integer (year)
v	character. Output variable. With `fromDate` it can be one of: "year", "month", "week", "day", "doy". With `fromYear` it can be one of "leap", "ndays", "days"
doy	integer (day of the year)

Value

integer or Date

Examples

```
d <- as.Date("1999-12-30") + 1:2
d
fromDate(d, "month")
fromDate(d, "doy")

fromDoy(10, 2000)
```

FSE weather *Read FSE formatted weather data*

Description

Read or write FSE formatted weather data

Usage

```
readFSEwth(f)
writeFSEwth(w, country='AAA', station=1, lon=0, lat=0, elev=0,  path=".")

example_weather()
```

Arguments

`f`	character. filename
`w`	data.frame with daily weather data. Must include the following variables: "date", "srad", "tmin", "tmax", "wind", "prec", "vapr". The data must be sorted by date in ascending order. "date" must be a `Date`, the other variables must be numeric
`country`	character code for a country (up to three letters)
`station`	positive integer. Station number for the country
`lon`	numeric. Longitude of the weather station (not used by the models)
`lat`	numeric. Latitude of the weather station
`elev`	numeric. Elevation of the weather station
`path`	character. Folder where you want to write the files. It must exist

Value

`readFSEwth`: data.frame

`writeFSEwth`: character (invisibly) with the filenames

`generics` *Generic functions*

Description

These are generic functions that are declared in this package but have no implementation here.

`globe` *Globe temperature*

Description

Globe temperatures (Tg, Tnwb, WBGT).

The can be computed for either a `data.frame` or a `SpatRasterDataset`. These must have variables "temp" (C), "rhum" (%), "wind" (m s-1), and "srad" (J s-1 m-2). The `data.frame` must also have a variable "date".

Usage

```
## S4 method for signature 'data.frame'
Tg(x, latitude)
## S4 method for signature 'SpatRasterDataset'
Tg(x, filename="", overwrite=FALSE, ...)

## S4 method for signature 'data.frame'
Tnwb(x, latitude, kelvin=FALSE)
## S4 method for signature 'SpatRasterDataset'
Tnwb(x, kelvin=FALSE, filename="", overwrite=FALSE, ...)

## S4 method for signature 'data.frame'
WBGT(x, latitude, kelvin=FALSE)
## S4 method for signature 'SpatRasterDataset'
WBGT(x, kelvin=FALSE, mask=NULL, filename="", overwrite=FALSE, ...)
```

Arguments

`x`	data.frame or SpatRasterDataset
`latitude`	numeric
`filename`	character. Output filename
`overwrite`	logical. If `TRUE`, `filename` is overwritten
`...`	additional arguments for writing files as in `writeRaster`
`kelvin`	logical. Set to `TRUE` of the units of temperature are in Kelvin
`mask`	NULL of SpatRaster. if a SpatRaster is used, it should have one layer. No computations are done for cells that are NA, and these are set to NA in the output

Value

numeric or SpatRaster

Examples

```
wd <- data.frame(date=as.Date("2003-08-28") + 1:3,
  temp=c(19.1, 20.6, 19.4),
  rhum=c(66,71,73),
  wind=c(3.3, 1.9, 1.1),
  srad=c(168, 178, 125))
Tg(wd, 40.96)

Tnwb(wd, 40.96)

WBGT(wd, 40.96)


library(terra)
r <- rast(ncol=2, nrow=2, nlyr=1)
```

```
temp <- setValues(r, 21:24)
time(temp) <- as.Date("2000-01-01")
rhum <- setValues(r, 81:84)
wind <- setValues(r, 9:12)
srad <- setValues(r, 100:103)
s <- sds(list(temp=temp, rhum=rhum, wind=wind, srad=srad))

x <- Tg(s)
y <- WBGT(s)
```

photoperiod *photoperiod*

Description

Compute photoperiod (daylength, sunshine duration) at a given latitude and day of the year.

Usage

```
## S4 method for signature 'Date'
photoperiod(x, latitude)

## S4 method for signature 'data.frame'
photoperiod(x)

## S4 method for signature 'SpatRaster'
photoperiod(x, filename="", overwrite=FALSE, ...)
```

Arguments

`x`	Date, integer (day of the year), or data.frame (with variables "date" and "latitude", or SpatRaster
`latitude`	numeric. Latitude
`filename`	character. Output filename
`overwrite`	logical. If `TRUE`, `filename` is overwritten
`...`	additional arguments for writing files as in `writeRaster`

Value

double. Photoperiod in hours

References

Forsythe, W.C., E.J. Rykiel Jr., R.S. Stahl, H. Wu, R.M. Schoolfield, 1995. A model comparison for photoperiod as a function of latitude and day of the year. Ecological Modeling 80: 87-95.

Examples

```
photoperiod(50, 52)
photoperiod(50, 5)
photoperiod(180, 55)

p <- photoperiod(1:365, 52)
d <- dateFromDoy(1:365, 2001)
plot(d, p)
```

---

power_weather — *Global weather data estimated from satellite data and models.*

Description

This functions returns a data.frame with weather data from the NASA POWER database. It has the date, incoming solar radiation (srad, kJ m-2 day-1), minimum temperature (tmin, degrees C) and maximum temperature (tmax, degrees C), vapor pressure (vapr, Pa), precipitation (prec, mm), and windspeed (wind, m/s)

The data are from 1983-01-01 to 2016-12-31

Missing values for radiation (Jan to June 1983 and ...) and a few inbetween were replaced by the long term averages.

There are no precipitation values before 1997-01-01. Missing values for precipitaion after that date were estimated as the long term average (i.e., not a particularly good method).

The data are at 1 degree spatial resolution. That is, they are the average for a large grid cell.

These are estimates. They can give a good general impression, but they are not ground observations.

Data are downloaded as-needed by tile. By default to folder called "power" in your working directory.

Usage

```
power_weather(lon, lat, folder=file.path(getwd(), 'power'), tiles=FALSE, ...)
```

Arguments

`lon`	numeric
`lat`	numeric
`folder`	character
`tiles`	logical. Download by tile?
`...`	additional arguments

Value

data.frame

Examples

```
## Not run:
w <- power_weather(5, 50)
w$srad <- w$srad * 1000
wth <- subset(w, date > as.Date('2012-01-01'))
head(wth)

## End(Not run)
```

---

pwc *pwc*

Description

pwc

Usage

```
## S4 method for signature 'numeric'
pwc(x, input="wbgt", adjust=TRUE)

## S4 method for signature 'SpatRaster'
pwc(x, input="wbgt", adjust=TRUE, filename="", overwrite=FALSE, ...)
```

Arguments

`x`	numeric or SpatRaster
`input`	character. One of "wbgt" or "utci"
`adjust`	logical. If `TRUE`, the Smallcombe et al. (2022) adjustment for a 7-hour workday is used
`filename`	character. Output filename
`overwrite`	logical. If `TRUE`, `filename` is overwritten
`...`	additional arguments for writing files as in `writeRaster`

Value

numeric or SpatRaster

References

Smallcombe et al.. 2022

Foster et al., 2022

Examples

```
pwc(25)
```

`utci` *utci*

Description

utci

Usage

```
## S4 method for signature 'data.frame'
utci(x)
```

Arguments

`x`	data.frame

Value

numeric or SpatRaster

Examples

```
d <- data.frame(temp=20, rhum=80, tglb=22, wind=10)
utci(d)
```

`vaporpressure` *Vapor pressure*

Description

Functions to compute the saturated vapor pressure (SVP), actual vapor pressure (VP), and vapor pressure deficit (VPD) in Pascal or the dew-point temperature in C.

For temperature < 0C the saturation vapour pressure equation for ice is used according to Goff and Gratch (1946), whereas for temperature >=0C that of Goff (1957) is used.

Usage

```
SVP(temp)
VP(temp, relh)
VPD(temp, relh)
tDew(temp, relh)
```

Arguments

`temp`	numeric. Temperature in degrees C
`relh`	relative humidity (percent)

Value

numeric vector (Pascal).

Author(s)

Robert Hijmans, partly based on Python meteolib by Maarten J. Waterloo and J. Delsman http://python.hydrology-amsterdam.nl/

References

Goff, J.A.,and S. Gratch, 1946. Low-pressure properties of water from -160 to 212 F. Transactions of the American society of heating and ventilating engineers, p. 95-122, presented at the 52nd annual meeting of the American society of heating and ventilating engineers, New York, USA.

Goff, J. A. 1957. Saturation pressure of water on the new Kelvin temperature scale, Transactions of the American society of heating and ventilating engineers, pp 347-354, presented at the semi-annual meeting of the American society of heating and ventilating engineers, Murray Bay, Quebec, Canada.

Examples

```
temperature <- seq(-10,30,10)
SVP(temperature)
VP(temperature, 60)
VPD(temperature, 60)
tDew(temperature, 60)
```

`Weather-class` *Weather class*

Description

Weather data

Objects from the Class

Objects can be created by calls of the form `new("Weather", ...)`, or with the helper functions such as `weather`.

Slots

Slots of Weather objects

`data`: data.frame with the weather data

`ID`: character

`name`: character

`country`: character

`longitude`: numeric

`latitude`: numeric

`elevation`: numeric

Examples

```
showClass("Weather")
```

Index