Package 'daymetr'

January 5, 2023

Title Interface to the 'Daymet' Web Services

Version 1.7

Description Programmatic interface to the 'Daymet' web services (<http://daymet.ornl.gov>). Allows for easy downloads of 'Daymet' climate data directly to your R workspace or your computer. Routines for both single pixel data downloads and gridded (netCDF) data are provided.

Depends R (>= 3.6)

Imports sf, terra, ncdf4, httr, tidyr, tibble, tools, utils

License AGPL-3

LazyData true

ByteCompile true

RoxygenNote 7.2.1

Suggests ggplot2, dplyr, knitr, markdown, covr, testthat

VignetteBuilder knitr

URL https://github.com/bluegreen-labs/daymetr

BugReports https://github.com/bluegreen-labs/daymetr/issues

NeedsCompilation no

Author Koen Hufkens [aut, cre] (<https://orcid.org/0000-0002-5070-8109>), BlueGreen Labs [cph, fnd]

Maintainer Koen Hufkens <koen.hufkens@gmail.com>

Repository CRAN

Date/Publication 2023-01-05 21:50:47 UTC

R topics documented:

calc_nd 2
daymetr 3
daymet_grid_agg 4

daymet_grid_offset 5
daymet_grid_tmean 5
download_daymet 6
download_daymet_batch 8
download_daymet_ncss 10
download_daymet_tiles 11
nc2tif 12
read_daymet 13
tile_outlines 14

Index **16**

`calc_nd` *Count days meeting set criteria*

Description

Function to count the number of days in a given time period that meet a given set of criteria. This can be used to extract indices such as Growing Degree Days (tmin > 0), or days with precipitation (prcp != 0).

Usage

```
calc_nd(
  file,
  start_doy = 1,
  end_doy = 365,
  criteria,
  value,
  internal = FALSE,
  path = tempdir()
)
```

Arguments

`file`	path of a file containing the daily gridded Daymet data
`start_doy`	numeric day-of-year at which counting should begin. (default = 1)
`end_doy`	numeric day of year at which counting should end. (default = 365)
`criteria`	logical expression (">=",">","<=","<","==", "!=") to evaluate
`value`	the value that the criteria is evaluated against
`internal`	return to workspace (`TRUE`) or write to disk (`FALSE`) (default = `FALSE`)
`path`	path to which to write data to disk (default = tempdir())

Value

A raster object in the R workspace or a file on disk with summary statistics for every pixel which meet the predefined criteria. Output files if written to file will be named nd_YYYY.tif (with YYYY the year of the processed tile or ncss netCDF file).

Examples

```
## Not run:
# download daily gridded data
# using default settings (data written to tempdir())
download_daymet_ncss()

# read in the Daymet file and report back the number
# of days in a year with a minimum temperature lower
# than 15 degrees C
r <- calc_nd(file.path(tempdir(),"tmin_daily_1980_ncss.nc"),
            criteria = "<",
            value = 15,
            internal = TRUE)

# plot the output
terra::plot(r)

## End(Not run)
```

---

daymetr *Interface to the 'Daymet' Web Services*

Description

Programmatic interface to the 'Daymet' web services (https://daymet.ornl.gov). Allows for easy downloads of Daymet climate data directly to your R workspace or your computer. Routines for both single pixel data downloads and gridded (netCDF) data are provided.

daymetr functions

download_daymet
download_daymet_batch
download_daymet_tiles
download_daymet_ncss
daymet_grid_tmean
daymet_grid_offset
daymet_grid_agg
nc2tif
read_daymet
calc_nd

Author(s)

Koen Hufkens

daymet_grid_agg *Aggregate daily Daymet data*

Description

Aggregates daily Daymet data by time interval to create convenient seasonal datasets for data exploration or modelling.

Usage

```
daymet_grid_agg(
  file,
  int = "seasonal",
  fun = "mean",
  internal = FALSE,
  path = tempdir()
)
```

Arguments

file	The name of the file to be processed. Use daily gridded Daymet data.
int	Interval to aggregate by. Options are "monthly", "seasonal" or "annual". Seasons are defined as the astronomical seasons between solstices and equinoxes (default = "seasonal")
fun	Function to be used to aggregate data. Genertic R functions can be used. "mean" and "sum" are suggested. na.rm = TRUE by default. (default = "mean")
internal	logical If FALSE, write the output to a tif file using the Daymet file format protocol.
path	path to a directory where output files should be written. Used only if internal = FALSE (default = tempdir())

Value

aggregated daily Daymet data as a tiff file written to disk or a raster stack when data is returned to the workspace.

Examples

```
## Not run:
# This code calculates the average minimum temperature by
# season for a subset region.

# download default ncss tiled subset for 1980
# (daily tmin values only), works on tiles as well
download_daymet_ncss()
```

```
 # Finally, run the function
 daymet_grid_agg(
  file = file.path(tempdir(),"/tmin_daily_1980_ncss.nc"),
  int = "seasonal",
  fun = "mean"
 )

## End(Not run)
```

---

daymet_grid_offset *Returns a time shifted (offset) dataset*

Description

Returns an offset dataset with data running from offset DOY in year - 1 to offset DOY in the current year. Two years of data (730 data layers) are required for this function to work. The output serves as input for further data processing and / or ecosystem modelling efforts.

Usage

```
daymet_grid_offset(data, offset = 264)
```

Arguments

data	rasterStack or rasterBrick of 730 layers (2 consecutive years)
offset	offset of the time series in DOY (default = 264, sept 21)

Examples

```
## Not run:
my_subset <- daymet_gridded_offset(mystack, offset = 264)

## End(Not run)
```

---

daymet_grid_tmean *Averages tmax and tmin 'Daymet' gridded products*

Description

Combines data into a single mean daily temperature (tmean) gridded output (geotiff) for easy post processing and modelling. Optionally a raster object is returned to the current workspace.

Usage

```
daymet_grid_tmean(path = tempdir(), product, year, internal = FALSE)
```

Arguments

path	full path location of the daymet tiles (default = tempdir())
product	either a tile number or a ncss product name
year	which year to process
internal	TRUE / FALSE (if FALSE, write the output to file) using the Daymet file format protocol.

Examples

```
## Not run:
# This code calculates the mean temperature
# for all daymet tiles in a user provided
# directory. In this example we first
# download tile 11935 for tmin and tmax

# download a tile
download_daymet_tiles(tiles = 11935,
                      start = 1980,
                      end = 1980,
                      param = c("tmin","tmax"),
                      path = tempdir())

# calculate the mean temperature and export
# the result to the R workspace (internal = TRUE)
# If internal = FALSE, a file tmean_11935_1980.tif
# is written into the source path (path_with_daymet_tiles)
tmean <- daymet_grid_tmean(path = tempdir(),
                           tile = 11935,
                           year = 1980,
                           internal = TRUE)

## End(Not run)
```

---

download_daymet *Function to download single location 'Daymet' data*

Description

Function to download single location 'Daymet' data

Usage

```
download_daymet(
  site = "Daymet",
  lat = 36.0133,
  lon = -84.2625,
  start = 2000,
```

```
  end = as.numeric(format(Sys.time(), "%Y")) - 2,
  path = tempdir(),
  internal = TRUE,
  silent = FALSE,
  force = FALSE,
  simplify = FALSE
)
```

Arguments

site	the site name.
lat	latitude (decimal degrees)
lon	longitude (decimal degrees)
start	start of the range of years over which to download data
end	end of the range of years over which to download data
path	set path where to save the data if internal = FALSE (default = NULL)
internal	TRUE or FALSE, if TRUE returns a list to the R workspace if FALSE puts the downloaded data into the current working directory (default = FALSE)
silent	TRUE or FALSE (default), to provide verbose output
force	TRUE or FALSE (default), override the conservative end year setting
simplify	output tidy data (tibble), logical FALSE or TRUE (default = TRUE)

Value

Daymet data for a point location, returned to the R workspace or written to disk as a csv file.

Examples

```
## Not run:
# The following commands download and process Daymet data
# for 10 years of the >30 year of data available since 1980.
daymet_data <- download_daymet("testsite_name",
                lat = 36.0133,
                lon = -84.2625,
                start = 2000,
                end = 2010,
                internal = TRUE)

# We can now quickly calculate and plot
# daily mean temperature. Also, take note of
# the weird format of the header. This format
# is not altered as to keep compatibility
# with other ways of acquiring Daymet data
# through the ORNL DAAC website.

# The below command lists headers of
# the downloaded nested list.
```

```
# This data includes information on the site
# location etc. The true climate data is stored
# in the "data" part of the nested list.
# In this case it can be accessed through
# daymet_data$data. Other attributes include
# for example the tile location (daymet_data$tile),
# the altitude (daymet_data$altitude), etc.
str(daymet_data)

# load the tidyverse (install if necessary)
if(!require(tidyverse)){install.package(tidyverse)}
library(tidyverse)

# Calculate the mean temperature from min
# max temperatures and convert the year and doy
# to a proper date format.
daymet_data$data <- daymet_data$data %>%
 mutate(tmean = (tmax..deg.c. + tmin..deg.c.)/2,
        date = as.Date(paste(year, yday, sep = "-"), "%Y-%j"))

# show a simple graph of the mean temperature
plot(daymet_data$data$date,
     daymet_data$data$tmean,
     xlab = "Date",
     ylab = "mean temperature")

# For other practical examples consult the included
# vignette.

## End(Not run)
```

`download_daymet_batch` *This function downloads 'Daymet' data for several single pixel location, as specified by a batch file.*

Description

This function downloads 'Daymet' data for several single pixel location, as specified by a batch file.

Usage

```
download_daymet_batch(
  file_location = NULL,
  start = 1980,
  end = as.numeric(format(Sys.time(), "%Y")) - 1,
  internal = TRUE,
  force = FALSE,
  silent = FALSE,
  path = tempdir(),
  simplify = FALSE
)
```

Arguments

file_location	file with several site locations and coordinates in a comma delimited format: site, latitude, longitude
start	start of the range of years over which to download data
end	end of the range of years over which to download data
internal	assign or FALSE, load data into workspace or save to disc
force	TRUE or FALSE (default), override the conservative end year setting
silent	suppress the verbose output (default = FALSE)
path	set path where to save the data if internal = FALSE (default = tempdir())
simplify	output tidy data (tibble), logical FALSE or TRUE (default = TRUE)

Value

Daymet data for point locations as a nested list or data written to csv files

Examples

```
## Not run:
# The download_daymet_batch() routine is a wrapper around
# the download_daymet() function. It queries a file with
# coordinates to easily download a large batch of daymet
# pixel locations. When internal = TRUE, the data is stored
# in a structured list in an R variable. If FALSE, the data
# is written to disk.

# create demo locations (two sites)
locations <- data.frame(site = c("site1", "site2"),
                        lat = rep(36.0133, 2),
                        lon = rep(-84.2625, 2))

# write data to csv file
write.table(locations, paste0(tempdir(),"/locations.csv"),
            sep = ",",
            col.names = TRUE,
            row.names = FALSE,
            quote = FALSE)

# download data, will return nested list of daymet data
df_batch <- download_daymet_batch(file_location = paste0(tempdir(),
                                                         "/locations.csv"),
                                      start = 1980,
                                      end = 1980,
                                      internal = TRUE,
                                      silent = TRUE)

# For other practical examples consult the included
# vignette.

## End(Not run)
```

`download_daymet_ncss` *Function to geographically subset 'Daymet' regions exceeding tile limits*

Description

Function to geographically subset 'Daymet' regions exceeding tile limits

Usage

```
download_daymet_ncss(
  location = c(34, -82, 33.75, -81.75),
  start = 1980,
  end = 1980,
  param = "tmin",
  frequency = "daily",
  mosaic = "na",
  path = tempdir(),
  silent = FALSE,
  force = FALSE,
  ssl = TRUE
)
```

Arguments

`location`	location of a bounding box c(lat, lon, lat, lon) defined by a top left and bottom-right coordinates
`start`	start of the range of years over which to download data
`end`	end of the range of years over which to download data
`param`	climate variable you want to download vapour pressure (vp), minimum and maximum temperature (tmin,tmax), snow water equivalent (swe), solar radiation (srad), precipitation (prcp) , day length (dayl). The default setting is ALL, this will download all the previously mentioned climate variables.
`frequency`	frequency of the data requested (default = "daily", other options are "monthly" or "annual".
`mosaic`	which tile mosiac to source from (na = Northern America, hi = Hawaii, pr = Puerto Rico), defaults to "na".
`path`	directory where to store the downloaded data (default = tempdir())
`silent`	suppress the verbose output
`force`	`TRUE` or `FALSE` (default), override the conservative end year setting
`ssl`	`TRUE` (default) or `FALSE`, override default SSL settings in case of CA issues

Value

netCDF data file of an area circumscribed by the location bounding box

Examples

```
## Not run:
# The following call allows you to subset gridded
# Daymet data using a bounding box location. This
# is an alternative way to query gridded data. The
# routine is particularly helpful if you need certain
# data which stradles boundaries of multiple tiles
# or a smaller subset of a larger tile. Keep in mind
# that there is a 6GB upper limit to the output file
# so querying larger regions will result in an error.
# To download larger areas use the download_daymet_tiles()
# function.

# Download a subset of a / multiple tiles
# into your current working directory.
download_daymet_ncss(location = c(34, -82, 33.75, -81.75),
                     start = 1980,
                     end = 1980,
                     param = "tmin",
                     path = tempdir())

# For other practical examples consult the included
# vignette.

## End(Not run)
```

`download_daymet_tiles` *Function to batch download gridded 'Daymet' data tiles*

Description

Function to batch download gridded 'Daymet' data tiles

Usage

```
download_daymet_tiles(
  location = c(18.9103, -114.6109),
  tiles,
  start = 1980,
  end = 1980,
  path = tempdir(),
  param = "ALL",
  silent = FALSE,
  force = FALSE
)
```

Arguments

location	location of a point c(lat, lon) or a bounding box defined by a top left and bottom-right coordinates c(lat, lon, lat, lon)
tiles	which tiles to download, overrides geographic constraints
start	start of the range of years over which to download data
end	end of the range of years over which to download data
path	where should the downloaded tiles be stored (default = tempdir())
param	climate variable you want to download vapour pressure (vp), minimum and maximum temperature (tmin,tmax), snow water equivalent (swe), solar radiation (srad), precipitation (prcp) , day length (dayl). The default setting is ALL, this will download all the previously mentioned climate variables.
silent	suppress the verbose output
force	TRUE or FALSE (default), override the conservative end year setting

Value

downloads netCDF tiles as defined by the Daymet tile grid

Examples

```
## Not run:
Download a single tile of minimum temperature
download_daymet_tiles(location = c(18.9103, -114.6109),
                      start = 1980,
                      end = 1980,
                      param = "tmin")

# For other practical examples consult the included
# vignette.

## End(Not run)
```

---

nc2tif *Converts netCDF (nc) files to geotiff*

Description

Conversion to .tif to simplify workflows if the data that has been downloaded is to be handled in other software (e.g. QGIS).

Usage

```
nc2tif(path = tempdir(), files = NULL, overwrite = FALSE, silent = FALSE)
```

Arguments

path	a character string showing the path to the directory containing Daymet .nc files (default = tempdir())
files	a character vector containing the name of one or more files to be converted (optional)
overwrite	a logical controlling whether all files will be written, or whether files will not be written in the event that there is already a .tif of that file. (default = NULL)
silent	limit verbose output (default = FALSE)

Value

Converted geotiff files of all netCDF data in the provided directory (path).

Examples

```
## Not run:

 # The below command converts all netCDF data in
 # the provided path to geotiff files. Existing
 # files will be overwritten. If set to FALSE,
 # files will not be overwritten.

 # download the data
 download_daymet_ncss(param = "tmin",
                      frequency = "annual",
                      path = tempdir(),
                      silent = TRUE))

 # convert files from nc to tif
 nc2tif(path = tempdir(),
 overwrite = TRUE)

 # print converted files
 print(list.files(tempdir(), "*.tif"))


## End(Not run)
```

---

read_daymet *Read Single Pixel Daymet data*

Description

Reads Single Pixel Daymet data into a nested list or tibble, preserving header data and critical file name information.

Usage

```
read_daymet(file, site, skip_header = FALSE, simplify = TRUE)
```

Arguments

file	a Daymet Single Pixel data file
site	a sitename (default = NULL)
skip_header	do not ingest header meta-data, logical FALSE or TRUE (default = FALSE)
simplify	output tidy data (tibble), logical FALSE or TRUE (default = TRUE)

Value

A nested data structure including site meta-data, the full header and the data as a 'data.frame()'.

Examples

```
## Not run:
# download the data
download_daymet(site = "Daymet",
                start = 1980,
                end = 1980,
                internal = FALSE,
                silent = TRUE)

# read in the Daymet file
df <- read_daymet(paste0(tempdir(),"/Daymet_1980_1980.csv"))

# print data structure
print(str(df))

## End(Not run)
```

---

tile_outlines *tile_outlines*

Description

Large simple feature collection containing the outlines of all the Daymet tiles available as well as projection information. This data was converted from a shapefile as provided on the Daymet main website.

Usage

```
tile_outlines
```

Format

SpatialPolygonDataFrame

TileID tile ID number

XMin minimum longitude

XMax maximum longitude

YMin minimum latitude

YMax maximum latitude

Index

∗ **datasets**
tile_outlines, 14
∗ **package**
daymetr, 3

calc_nd, 2, *3*

daymet_grid_agg, *3*, 4
daymet_grid_offset, *3*, 5
daymet_grid_tmean, *3*, 5
daymetr, 3
download_daymet, *3*, 6
download_daymet_batch, *3*, 8
download_daymet_ncss, *3*, 10
download_daymet_tiles, *3*, 11

nc2tif, *3*, 12

read_daymet, *3*, 13

tile_outlines, 14