Package 'AWAPer'

February 15, 2018

Type Package

Title Extract daily climate data derived from the Bureau of Meteorology Australian Water Availability Project.

Version 0.0.0.2000

Authors person(``Tim'', ``Peterson'', email = ``timjp@unimelb.edu.au'', role = c(``aut'', ``cre''))

Description This package builds netCDF files of the Bureau of Meteorology Australian Water Availability Project daily national climate grids and allows efficient extraction of daily points and catchment average precipitation, Tmin, Tmax, vapour pressure, solar radiation and then estimation of areal potential evaporation (Morton's). For details on the source climate data see http://www.bom.gov.au/jsp/awap/.

Depends R (>= 3.2.3), Evapotranspiration (>= 1.12)

Imports ncdf4, R.utils, raster, chron, maptools, sp

BugReports https://github.com/peterson-tim-j/AWAPer/issues

URL https://github.com/peterson-tim-j/AWAPer

License GPL3 (or later)

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

R topics documented:

catchments *Example catchment boundary polygons.*

Description

Two example catchment boundaries as a SpatialPolygonsDataFrame. The catchments are Creswick Creek (ID 407214, Vic., Australia, see http://www.bom.gov.au/water/hrs/#id=407214) and Bet Bet Creek (ID 407220, Vic., Australia, see http://www.bom.gov.au/water/hrs/#id=407220). The catchments can be used to extract catchment average climate data usng `extractCatchmentData`

Usage

```
catchments
```

Format

An object of class `SpatialPolygonsDataFrame` with 2 rows and 1 columns.

See Also

`extractCatchmentData` for extracting catchment average climate data.

Examples

```
# Load required packages.
library(sp);library(raster);library(chron);library(AWAPer);library(ncdf4);
library(maptools);library(Evapotranspiration)

# Load example cacthment boundaries.
data("catchments")

# Load the 9 second Australian DEM.
data("DEM_9s")

# Plot the catchment boundaries.
image(DEM_9s, xlab='Long.',ylab='Lat.')
plot(catchments,add=T)
```

extractCatchmentData *Extracts daily catchment mean and variance from netCDF files of the Bureau of Meteorology (Australia) national gridded climate data.*

Description

`extractCatchmentData` extracts catchment average climate data from netCDF files containing Australian climate data.

Usage

```
extractCatchmentData(ncdfFilename = file.path(getwd(), "AWAP.nc"),
  ncdfSolarFilename = file.path(getwd(), "AWAP_solar.nc"),
  extractFrom = as.Date("1900-01-01", "%Y-%m-%d"),
  extractTo = as.Date(Sys.Date(), "%Y-%m-%d"), getPrecip = TRUE,
  getTmin = TRUE, getTmax = TRUE, getVprp = TRUE, getSolarrad = TRUE,
  getMortonsPET = TRUE, DEM = "", catchments = "")
```

Arguments

ncdfFilename	is a full file name (as string) to the netCDF file.
ncdfSolarFilename	is the full file name (as string) to the netCDF file.
extractFrom	is a date string specifying the start date for data extraction. The default is "1900-1-1".
extractTo	is a date string specifying the end date for the data extraction. The default is today's date as YYYY-MM-DD.
getPrecip	logical variable for extracting precipitation. Default is TRUE.
getTmin	logical variable for extracting Tmin. Default is TRUE.
getTmax	logical variable for extracting Tmax. Default is TRUE.
getVprp	logical variable for extracting vapour pressure. Default is TRUE.
getSolarrad	logical variable for extracting solar radiation. Default is TRUE.
getMortonsPET	logical variable for calculating Morton's potential ET. Note, to calculate set getTmin=T, getTmax=T, getVprp=T and getSolarrad=T. Default is TRUE.
DEM	is either the full file name to a ESRI ASCII grid (as lat/long and using GDA94) or a raster class grid object. The DEM is used for the calculation of Morton's PET. The Australian 9 second DEM can be loaded using data(DEM_9s). For details see https://www.data.gov.au/dataset/geodata-9-second-dem-and-d8-digital-elevation-mo
catchments	is either the full file name to an ESRI shape file of points or polygons (latter assumed to be catchment boundaries) or a shape file already imported using readShapeSpatial(). Either way the shape file must be in long/lat (i.e. not projected) use the ellipsoid GRS 80.

Details

This function extracts the AWAP climate data for each point or polygon, and for the latter the daily spatial mean and variance of each climate metric are calculated. The calculation of the spatial mean uses the fraction of each AWAP grid cell within the polygon. The variance calculation does not use the fraction of the grid cell and returns NA if there are <2 grid cells in the catchment boundary. Prior to the catchment averaging and variance, Morton's areal potential ET (PET) is also calculated; after which the mean and variance PET is calculated. Morton's PET is calculated using the ET.MortonCRAE() function from the Evapotranspiration package at a monthly time-step and using the AWAP solar radiation. For both points and polygons, the monthly PET estimate is then interpolated using a spline to a daily time step (using zoo:na.spline()) and then constrained to >=0.

Also, when "catchments" is points (not polygons), then the netCDF grids are interpolate using bilinear interpolation of the closest 4 grid cells.

Lastly, data is extracted for all time points and no temporal infilling is undertaken if the grid cells are blank.

Value

When "catchments" are polygons, the returned variable is list variables containing two data.frames, one of the catchment average daily climate metrics and another of the catchment variance daily climate metrics.

When "catchments" are points, the returned variable is a data.frame containing daily climate data at each point.

See Also

`makeNetCDF_file` for building the NetCDF files of daily climate data.

Examples

```
# Load required packages.
library(sp);library(raster);library(chron);library(ncdf4);
library(maptools);library(Evapotranspiration);library(AWAPer)

# Download the 9 second Australian DEM.
DEM_9s = getDEM()

# Load example cacthment boundaries.
data("catchments")

# Extract all climate data and calculate Morton's PET.
# Note, the input "catchments" can also be a file to a ESRI shape file.
climateData = extractCatchmentData(catchments=catchments,DEM=DEM_9s)

# Extract the daily catchment average data.
climateDataAvg = climateData$catchmentAvg

# Extract the daily catchment variance data.
climateDataVar = climateData$catchmentVar

# Export data to .csv files
write.csv(climateDataAvg,'warrionClimateAvg.csv')
write.csv(climateDataVar,'warrionClimateVar.csv')
```

`getDEM` *Downloads and imports Geoscience Australia 9s DEM.*

Description

getDEM get Australian 9s DEM.

Usage

```
getDEM(workingFolder = getwd(),
  urlDEM = "https://s3-ap-southeast-2.amazonaws.com/elvis.ga.gov.au/elevation/9secPackagedData/DEM-9
  keepFiles = F)
```

Arguments

workingFolder	is the file path (as string) in which to download the zip file. The default is getwd().
urlDEM	URL to the folder containing the Geoscience Australia 9s DEM. The default is https://s3-ap-southeast-2.amazonaws.com/elvis.ga.gov.au/elevation/9secPackagedData/DEM-9S_ESRI.zip.
keepFiles	is a logical scalar to keep the downloaded zip file and extracted DEM ASCII file. The default is FALSE.

Details

This function downlaod the Geoscience Australia 9 second DEM and then imports the grid. The DEM is required for the calculation of evaportranspiration within extractCatchmentData. For details of the DEM see https://www.data.gov.au/dataset/geodata-9-second-dem-and-d8-digital-elevation-mode

Value

A RasterLayer DEM for Asutralia.

See Also

extractCatchmentData for extracting catchment daily average and variance data.

Examples

```
# Load required packages.
library(sp);library(raster);library(chron);library(ncdf4);
library(maptools);library(Evapotranspiration);library(AWAPer)

# Download the DEM.
DEM_9s = getDEM()

# Plot the DEM.
image(DEM_9s, xlab='Long.',ylab='Lat.')

# Save DEM for next time it is needed.
save(DEM_9s,file="DEM.RData" )
```

makeNetCDF_file *Build netCDF files of the Bureau of Meteorology (Australia) national gridded climate data.*

Description

makeNetCDF_file builds two netCDF files containing Australian climate data.

Usage

```
makeNetCDF_file(ncdfFilename = file.path(getwd(), "AWAP.nc"),
  ncdfSolarFilename = file.path(getwd(), "AWAP_solar.nc"),
  updateFrom = as.Date("1900-01-01", "%Y-%m-%d"),
  updateTo = as.Date(Sys.Date(), "%Y-%m-%d"), workingFolder = getwd(),
  keepFiles = FALSE,
 urlPrecip = "http://www.bom.gov.au/web03/ncc/www/awap/rainfall/totals/daily/grid/0.05/history/nat/
 urlTmin = "http://www.bom.gov.au/web03/ncc/www/awap/temperature/minave/daily/grid/0.05/history/nat
 urlTmax = "http://www.bom.gov.au/web03/ncc/www/awap/temperature/maxave/daily/grid/0.05/history/nat
 urlVprp = "http://www.bom.gov.au/web03/ncc/www/awap/vprp/vprph15/daily/grid/0.05/history/nat/",
 urlSolarrad = "http://www.bom.gov.au/web03/ncc/www/awap/solar/solarave/daily/grid/0.05/history/nat
```

Arguments

ncdfFilename	is a file path (as string) and name to the netCDF file. The default is file.path(getwd(),'AWAP.nc').
ncdfSolarFilename	is the file path (as string) and name to the netCDF file. The default is file.path(getwd(),'AWAP_solar.
updateFrom	is a date string specifying the start date for the AWAP data. If ncdfFilename and ncdfSolarFilename are specified and exist, then the netCDF grids will be updated with new data from updateFrom. To update the files from the end of the last day in the file set updateFrom=NA. The default is "1900-1-1".
updateTo	is a date string specifying the end date for the AWAP data. If ncdfFilename and ncdfSolarFilename are specified and exist, then the netCDF grids will be updated with new data to updateFrom. The default is today's date as YYYY-MM-DD.
workingFolder	is the file path (as string) in which to download the AWAP grid files. The default is getwd().
keepFiles	is a logical scalar to keep the downloaded AWAP grid files. The default is FALSE.
urlPrecip	URL to the folder containing the AWAP daily precipittaion grids. The default is http://www.bom.gov.au/web03/ncc/www/awap/rainfall/totals/daily/grid/0.05/history/nat/.
urlTmin	URL to the folder containing the AWAP daily minimum temperature grids. The default is http://www.bom.gov.au/web03/ncc/www/awap/temperature/minave/daily/grid/0.05/history/nat/.

urlTmax — URL to the folder containing the AWAP daily maximum temperature grids. The default is http://www.bom.gov.au/web03/ncc/www/awap/temperature/maxave/daily/grid/0.05/history/nat/.

urlVprp — URL to the folder containing the AWAP daily vapour pressure grids. The default is http://www.bom.gov.au/web03/ncc/www/awap/vprp/vprph15/daily/grid/0.05/history/nat/.

urlSolarrad — URL to the folder containing the AWAP daily solar radiation grids. The default is http://www.bom.gov.au/web03/ncc/www/awap/solar/solarave/daily/grid/0.05/history/nat/.

Details

This function creates two netCDF files of daily climate data. One file contains precipitation, minimum daily temperature, maximum daily temperature and vappour pressure. It should span from 1/1/1900 to today and be ~360GB. The second file contains the solar radiation and started from 1/1/1990 and be ~24GB and spatial gaps are infilled using a 3x3 moving average repeated 3 times. The climate data is sourced from the Bureau of Meteorology Australian Water Availability Project (http://www.bom.gov.au/jsp/awap/. For details see Jones et al. (2009).

The output from this function is required for all data extraction functions within this package and must be ran prior.

The function can be used to build netCDF files from stratch or to update existng netCDF files previously derived from this function. To not build or update a variable, set its respective URL to NA.

References

David A. Jones, William Wang and Robert Fawcett, (2009), High-quality spatial climate data-sets for Australia, Australian Meteorological and Oceanographic Journal, 58 , p233-248.

See Also

extractCatchmentData for extracting catchment daily average and variance data.

Examples

```
# Load required packages.
library(sp);library(raster);library(chron);library(ncdf4);
library(maptools);library(Evapotranspiration);library(AWAPer)

# Example 1. Build netCDF grids for all existing time points.
makeNetCDF_file()

# Example 2. Build netCDF grids for ONLY precipitation data at all existing time points.
makeNetCDF_file(urlTmin = NA,urlTmax = NA, urlVprp = NA, urlSolarrad = NA)

# Example 3. Build netCDF grids for all data but only over a defined time period.
# Note if the netCDFs have already been created, then the netCDF files will be updated.
makeNetCDF_file(updateFrom='2000-1-1',updateTo='2002-12-31')
```

```
# Example 4. Update the netCDF grids (from example 3) from the end dates within the netCDF files to the current da
makeNetCDF_file(updateFrom=NA)
```

Index