

Package ‘aws.wrfsmn’

July 22, 2025

Type Package

Title Data Processing of SMN Hi-Res Weather Forecast from 'AWS'

Version 0.1.0

Description Exploration of Weather Research & Forecasting ('WRF') Model data of Servicio Meteorologico Nacional (SMN) from Amazon Web Services (<https://registry.opendata.aws/smn-ar-wrf-dataset/>) cloud. The package provides the possibility of data downloading, processing and correction methods. It also has map management and series exploration of available meteorological variables of 'WRF' forecast.

License GPL (>= 3)

Depends R (>= 4.1.0)

Imports aws.s3 (>= 0.3.21), lubridate (>= 1.9.3), terra (>= 1.7-65), dplyr (>= 1.1.4), ggplot2 (>= 3.4.4), hydroGOF (>= 0.5-4), stats (>= 4.1.2), magrittr (>= 2.0.3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

NeedsCompilation no

Author Gonzalo Diaz [cre, aut]

Maintainer Gonzalo Diaz <gonzalomartindiaz22@gmail.com>

Repository CRAN

Date/Publication 2025-07-22 18:30:54 UTC

Contents

daily.data.fields	2
eva	2

find.nearest.point	3
get.wrf.files	4
ith	4
load.by.variable	5
mg.evaluation	5
multiple.guidance	6
ploting	7
wrf.download	7
%>%	8
Index	9

daily.data.fields	<i>Daily data is obtained from hourly data</i>
-------------------	--

Description

....

Usage

daily.data.fields(raster.list, aggregate)

Arguments

- | | |
|-------------|--|
| raster.list | Spat Raster variable with several times for a unique variable (T2 or HR2 or ...) |
| aggregate | Type of aggregation (sum, mean, min, max) |

Value

Spat Raster with daily information

eva	<i>Evaporation data (observation and model)</i>
-----	---

Description

Data of evaporation from in-situ observation and several soil model outputs

Usage

data(eva)

Format

- An object of class "data.frame".
- Dates** 1st column with dates
- evapo_obs** 2nd column with evaporation observation
- OUT_PREC** Precipitation
- OUT_EVAP** Evaporation
- OUT_RUNOFF** Runoff
- OUT_BASEFLOW** Baseflow
- OUT_SOIL_MOIST_lyr_1** Soil moisture from 1st layer
- OUT_EVAP_CANOP** Evaporation from canopy
- OUT_SURF_TEMP** Surface temperature

References

Diaz et al. (2024) AAGG 2024 Not yet published

Examples

data(eva)

find.nearest.point	<i>Temporal series of closest location</i>
--------------------	--

Description

Location of nearest point to lon/lat and temporal serie of location

Usage

find.nearest.point(data.spat.raster = data.spat.raster, lon = lon, lat = lat)

Arguments

- data.spat.raster
Spat Raster of WRF SMN (only one or several)
- lon
Longitude location of nearest point to find
- lat
Latitude location of nearest point to find

Value

a vector with the nearest location (lon/lat) and time serie of that location

<code>get.wrf.files</code>	<i>List of available files for downloading</i>
----------------------------	--

Description

Character string with the filenames of WRF SMN located in AWS Bucket

Usage

```
get.wrf.files(year = year, month = month, day = day, cycle = cycle, time = time)
```

Arguments

<code>year</code>	character or numeric of year
<code>month</code>	character or numeric of month
<code>day</code>	character or numeric of day
<code>cycle</code>	cycle of forecast, "00", "06", "12" or "18"
<code>time</code>	selection of datasets, "01H", "24H" or "10M"

Value

string of the list of elements in the Bucket

<code>ith</code>	<i>Calculation of ITH index</i>
------------------	---------------------------------

Description

ITH index calculation is made from gridded observational or model data. If the data is needed in lat/lon projection is better to use first the `load.by.variable` function to change projection

The index is calculated as:

$$ITH = 1.8 * T(C) + 32 - (((0.55 - (0.55 * RelHum(\%))) / 100) * ((1.8 * T(C)) - 26))$$

where $T(^{\circ}C)$ is the temperature in celsius degrees and $RelHum(\%)$ is the relative humidity in percentage

Usage

```
ith(raster.list = raster.list)
```

Arguments

raster.list	Spat Raster variable with several variables and times or only one Spat Raster field
-------------	---

Value

Spat Raster with ITH calculation for each time

load.by.variable	<i>Load and projection of data</i>
------------------	------------------------------------

Description

Open of netcdf files of WRF SMN from AWS and optional projection

Usage

```
load.by.variable(nc.filesnames, variable, transform, method)
```

Arguments

nc.filesnames	netcdf files
variable	name of variable from https://odp-aws-smn.github.io/documentation_wrf_det/Formato_de_datos/ as character
transform	TRUE to project data to longlat datum=WGS84
method	if transform is set TRUE define projection method taken from project function of terra package

Value

Spat Raster with the chosen variable (optional: with the projection changed)

mg.evaluation	<i>Evaluation of regression</i>
---------------	---------------------------------

Description

Evaluation of the linear multiple regression obtained from the multiple.guidance function

Usage

```
mg.evaluation(
  input.data = input.data,
  predictand = predictand,
  predictors = predictors,
  var.model = var.model,
  lmodel = lmodel
)
```

Arguments

input.data	Data frame with first column as a "POSIXct" class and one or more columns with data values. The predictand and predictors variables should be located in these columns
predictand	Character with column name of the predictand variable
predictors	Character array with one or more elements of the predictors chosen by the user
var.model	Character with column name of the modeled predicting variable
lmodel	Element of class lm obtained from multiple.guidance output function

Value

List of two elements. First element is a matrix with the columns of observed data, modeled data and corrected data. Second element is a data frame of the statistical results of the modeled and corrected data versus observed data

multiple.guidance	<i>Multiple lineal regression of data</i>
-------------------	---

Description

Definition of linear multiple regression adjustment based on predictor variables that fit a predicting variable

Usage

```
multiple.guidance(
  input.data = input.data,
  predictand = predictand,
  predictors = predictors
)
```

Arguments

input.data	Data frame with first column as a "POSIXct" class and one or more columns with data values. The predictand and predictors variables should be located in these columns
predictand	Character with column name of the predictand variable
predictors	Character array with one or more elements of the predictors chosen by the user

Value

an element of class lm

ploting	<i>Plot of data</i>
---------	---------------------

Description

Plot of observed, modeled and corrected guidance series

Usage

```
ploting(data = data)
```

Arguments

data Data frame from daily2monthly output function or any other temporal series

Value

ggplot element

wrf.download	<i>Download of wrf files</i>
--------------	------------------------------

Description

Download of WRF SMN data from AWS

Usage

```
wrf.download(wrf.name = wrf.name)
```

Arguments

wrf.name list of names to download from get.wrf.files. e.g.: "DATA/WRF/DET/2024/01/01/18/WRFDETAR_24H_

Value

downloaded netcdf files

%>%	<i>Daily data to monthly</i>
-----	------------------------------

Description

Data transformation from daily to monthly scale

Usage

```
daily2monthly(data = data)
```

Arguments

data matrix with daily data from mg.evaluation output function

Value

Data frame with monthly data

Index

* datasets

eva, [2](#)

%>%, [8](#)

daily.data.fields, [2](#)

daily2monthly (%>%), [8](#)

eva, [2](#)

find.nearest.point, [3](#)

get.wrf.files, [4](#)

ith, [4](#)

load.by.variable, [5](#)

mg.evaluation, [5](#)

multiple.guidance, [6](#)

ploting, [7](#)

wrf.download, [7](#)