

# Package: aws.wrfsmn (via r-universe)

September 5, 2024

**Type** Package

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

**Version** 0.0.3

**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

**Roxygen** list(markdown = TRUE)

**LazyData** true

**RoxygenNote** 7.3.1

**Suggests** knitr, rmarkdown, testthat (>= 3.0.0)

**VignetteBuilder** knitr

**Config/testthat/edition** 3

**Repository** <https://gonzalo1985.r-universe.dev>

**RemoteUrl** <https://github.com/gonzalo1985/aws.wrfsmn>

**RemoteRef** HEAD

**RemoteSha** 339586a2445fad43cd54e745477e8f77c9feb0d7

## Contents

eva . . . . .	2
find.nearest.point . . . . .	3
get.wrf.files . . . . .	3
load.by.variable . . . . .	4
mg.evaluation . . . . .	4
multiple.guidance . . . . .	5
ploting . . . . .	6
wrf.download . . . . .	6
%>% . . . . .	7
<b>Index</b>	<b>8</b>

---

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\_1yr\_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      *Temporal series of closest location*

---

**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

---

get.wrf.files      *List of available files for downloading*

---

**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**

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

**Value**

string of the list of elements in the Bucket

---

load.by.variable      *Load and projection of data*

---

### Description

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

### Usage

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

### Arguments

nc.fileNames	netcdf files
variable	name of variable from <a href="https://odp-aws-smn.github.io/documentation_wrf_det/Formato_de_datos/">https://odp-aws-smn.github.io/documentation_wrf_det/Formato_de_datos/</a> 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

---

mg.evaluation      *Evaluation of regression*

---

### 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

*Plot of data*

---

**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

*Download of wrf files*

---

**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

---

%>%

*Daily data to monthly*

---

**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](#)

%>%, [7](#)

daily2monthly (%>%), [7](#)

eva, [2](#)

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

get.wrf.files, [3](#)

load.by.variable, [4](#)

mg.evaluation, [4](#)

multiple.guidance, [5](#)

plotting, [6](#)

wrf.download, [6](#)