

# Package ‘BrazilMet’

July 21, 2025

**Type** Package

**Title** Download and Processing of Automatic Weather Stations (AWS) Data of INMET-Brazil

**Version** 0.4.0

**Language** en-US

**Description** A collection of functions for downloading and processing automatic weather station (AWS) data from INMET (Brazil’s National Institute of Meteorology), designed to support the estimation of reference evapotranspiration (ET<sub>o</sub>). The package facilitates streamlined access to meteorological data and aims to simplify analyses in agricultural and environmental contexts.

**License** GPL-3

**Encoding** UTF-8

**Depends** R (>= 4.1.0)

**Imports** stringr, readxl, dplyr(>= 0.3.0.1), tibble, lubridate, sf, stringi, terra

**BugReports** <https://github.com/FilgueirasR/BrazilMet/issues>

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Roberto Filgueiras [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-0186-8907>>),

Luan P. Venancio [aut] (ORCID: <<https://orcid.org/0000-0002-5544-8588>>),

Catariny C. Aleman [aut] (ORCID:

<<https://orcid.org/0000-0002-3894-3077>>),

Fernando F. da Cunha [aut] (ORCID:

<<https://orcid.org/0000-0002-1671-1021>>),

Arthur T. Calegario [ctb] (ORCID:

<<https://orcid.org/0000-0002-1921-5234>>)

**Maintainer** Roberto Filgueiras <betofilgueiras@gmail.com>

**Repository** CRAN

**Date/Publication** 2025-05-23 20:02:07 UTC

## Contents

correction_etp_thornwaite . . . . .	2
daily_eto_FAO56 . . . . .	3
design_eto . . . . .	4
download_AWS_INMET_daily . . . . .	5
download_climate_normals . . . . .	6
ea_dew_calculation . . . . .	7
ea_rh_calculation . . . . .	7
es_calculation . . . . .	8
es_ea_calculation . . . . .	9
eto_hs . . . . .	10
etp_thorntwaite . . . . .	10
get_max_eto_at_location . . . . .	11
hourly_weather_station_download . . . . .	12
max_eto_grid_download . . . . .	13
Patm . . . . .	14
psy_const . . . . .	15
radiation_conversion . . . . .	15
ra_calculation . . . . .	17
rh_calculation . . . . .	17
rn1_calculation . . . . .	18
rns_calculation . . . . .	19
rn_calculation . . . . .	20
rso_calculation_1 . . . . .	20
rso_calculation_2 . . . . .	21
rs_nearby_calculation . . . . .	22
see_stations_info . . . . .	23
selectAWSstations . . . . .	23
sr_ang_calculation . . . . .	24
sr_tair_calculation . . . . .	25
u2_calculation . . . . .	25
<b>Index</b>	<b>27</b>

---

correction\_etp\_thornwaite

*Correction for Thorntwaite - Potential evapotranspiration*

---

### Description

Correction for Thorntwaite - Potential evapotranspiration

### Usage

correction\_etp\_thornwaite(etp, date, lat)

**Arguments**

etp	A column of a dataframe containing Thorntwaite potential evapotranspiration (ETp) data without adjustments for sunlight hours and the number of days in the month."
date	A column of dataframe with date (i.e: 2025-01-02).
lat	A column of dataframe with latitude in degrees.

**Value**

Returns a vector object with the Thorntwaite ETp corrected for sunlight hours and the number of days in the month.

**Author(s)**

Roberto Filgueiras.

**Examples**

```
## Not run:
etp_cor <- correction_etp_thornwaite(etp, date, lat)

## End(Not run)
```

---

daily_eto_FAO56	<i>ETo calculation based on FAO-56 Penman-Monteith methodology, with data from automatic weather stations (AWS) downloaded and processed in function *daily_download_AWS_INMET*</i>
-----------------	---

---

**Description**

This function will calculate the reference evapotranspiration (ETo) based on FAO-56 (Allen et al., 1998) with the automatic weather stations (AWS) data, downloaded and processed in function \*daily\_download\_AWS\_INMET\*.

**Usage**

```
daily_eto_FAO56(lat, tmin, tmax, tmean, Rs, u2, Patm, RH_max, RH_min, z, date)
```

**Arguments**

lat	A numeric value of the Latitude of the AWS (decimal degrees).
tmin	A dataframe with Minimum daily air temperature (Celsius).
tmax	A dataframe with Maximum daily air temperature (Celsius).
tmean	A dataframe with Mean daily air temperature (Celsius).
Rs	A dataframe with mean daily solar radiation (MJ m-2 day-1).

u2	A dataframe with Wind speed at two meters high (m s-2).
Patm	A dataframe with atmospheric Pressure (mB).
RH_max	A dataframe with Maximum relative humidity (percentage).
RH_min	A dataframe with Minimum relative humidity (percentage).
z	A numeric value of the altitude of AWS (m).
date	A data.frame with the date information (YYYY-MM-DD).

**Value**

Returns a data.frame with the AWS data requested

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:
eto <- daily_eto_FA056(lat, tmin, tmax, tmean, Rs, u2, Patm, RH_max, RH_min, z, date)

## End(Not run)
```

---

design\_eto

*Design reference evapotranspiration (Design ETo)*

---

**Description**

Function to calculate the reference evapotranspiration for irrigation design.

**Usage**

```
design_eto(eto_daily_data, percentile = 0.8, grouping = NULL)
```

**Arguments**

eto_daily_data	A data frame containing daily reference evapotranspiration values (mm day-1) named as "ETo", and other column with dates named "date". To estimate the design reference evapotranspiration, it is recommended to use more than 10 years of historical data.
percentile	The relative position of a value within the data distribution. The recommendation to desing irrigating is greater than 80 percent (0.8). For instance: The 90th (0.90) percentile of reference evapotranspiration (ETo) represents a value that is surpassed in only 10 percent of the observed period.
grouping	specifies the column used to group the data, typically by station. Use this parameter to calculate the design ETo separately for each station, based on the corresponding station code.

**Value**

Returns a numeric object with the desing ETo for the respective data and percentile.

**Author(s)**

Roberto Filgueiras

**Examples**

```
## Not run:  
design_eto_value <- design_eto(eto, percentile)  
  
## End(Not run)
```

---

download\_AWS\_INMET\_daily

*Download of hourly data from automatic weather stations (AWS) of INMET-Brazil in daily aggregates*

---

**Description**

This function will download the hourly AWS data of INMET and it will aggregate the data in a daily time scale, based on the period of time selected (start\_date and end\_date).

**Usage**

```
download_AWS_INMET_daily(stations, start_date, end_date)
```

**Arguments**

stations	The stations code (ID - WMO code) for download. To see the station ID, please see the function <i>*see_stations_info*</i> .
start_date	Date that start the investigation, should be in the following format (1958-01-01 /Year-Month-Day)
end_date	Date that end the investigation, should be in the following format (2017-12-31 /Year-Month-Day)

**Value**

Returns a data.frame with the AWS data requested

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

### Examples

```
## Not run:
df <- download_AWS_INMET_daily(
  stations = c("A001", "A042"),
  start_date = "2016-01-01",
  end_date = "2018-12-31"
)

## End(Not run)
```

---

download\_climate\_normals

*Download climatological normals from Conventional weather stations (CWS) of Inmet*

---

### Description

This function will download the climatological normals from CWS stations available in Inmet site.

### Usage

```
download_climate_normals(variable, range_time)
```

### Arguments

variable	The variables available to download. The available variables are: rainfall_norm, t2m_norm, rh_norm, ws_norm and etp_norm.
range_time	The range of time which the climatological normals were computed. The available range of time is: "1991-2020".

### Value

Returns a data.frame with climatological normal required.

### Author(s)

Roberto Filgueiras

### Examples

```
## Not run:
df <- download_climate_normals(
  variable = "rainfall_norm",
  range_time = "1991-2020"
)

## End(Not run)
```

---

ea\_dew\_calculation      *Actual vapour pressure (ea) derived from dewpoint temperature*

---

**Description**

Actual vapour pressure (ea) derived from dewpoint temperature

**Usage**

```
ea_dew_calculation(tdew)
```

**Arguments**

tdew                    A dataframe with dewpoint temperature (Celsius).

**Value**

Returns a data.frame object with the ea from dewpoint data.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha.

**Examples**

```
## Not run:  
ea <-ea_dew_calculation(tdew).  
  
## End(Not run)
```

---

ea\_rh\_calculation      *Actual vapour pressure (ea) derived from relative humidity data*

---

**Description**

Actual vapour pressure (ea) derived from relative humidity data

**Usage**

```
ea_rh_calculation(tmin, tmax, rh_min, rh_mean, rh_max)
```

**Arguments**

tmin                    A dataframe with minimum daily air temperature (Celsius)  
tmax                    A dataframe with maximum daily air temperature (Celsius)  
rh\_min                  A dataframe with minimum daily relative air humidity (percentage).  
rh\_mean                 A dataframe with mean daily relative air humidity (percentage).  
rh\_max                  A dataframe with maximum daily relative air humidity (percentage).

**Value**

Returns a data.frame object with the with ea from relative humidity data.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
ea <- ea_rh_calculation(tmin, tmax, rh_min, rh_mean, rh_max)  
  
## End(Not run)
```

---

es_calculation	<i>Mean saturation vapour pressure (es)</i>
----------------	---

---

**Description**

Mean saturation vapour pressure (es)

**Usage**

```
es_calculation(tmin, tmax)
```

**Arguments**

tmin	A dataframe with Minimum daily air temperature (Celsius).
tmax	A dataframe with Maximum daily air temperature (Celsius).

**Value**

Returns a data.frame object with the es data.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha.

**Examples**

```
## Not run:  
es <- es_calculation(tmin, tmax)  
  
## End(Not run)
```



---

es_ea_calculation	<i>Vapour pressure deficit (es - ea)</i>
-------------------	--

---

### Description

Vapour pressure deficit (es - ea)

### Usage

```
es_ea_calculation(tmin, tmax, tdew, rh_min, rh_mean, rh_max, ea_method)
```

### Arguments

tmin	A dataframe with minimum daily air temperature (Celsius).
tmax	A dataframe with maximum daily air temperature (Celsius).
tdew	A dataframe with dewpoint temperature (Celsius).
rh_min	A dataframe with minimum daily relative air humidity (percentage).
rh_mean	A dataframe with mean daily relative air humidity (percentage).
rh_max	A dataframe with maximum daily relative air humidity (percentage).
ea_method	The methodology to calculate the actual vapour pressure. Assume the "rh" (default) for relative humidity procedure and "dew" for dewpoint temperature procedure.

### Value

Returns a data.frame object with the ea from relative humidity data.

### Author(s)

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

### Examples

```
## Not run:  
ea <- es_ea_calculation(tmin, tmax, tdew, rh_min, rh_mean, rh_max, ea_method)  
  
## End(Not run)
```

---

eto\_hs                      *Hargreaves - Samani ETo*

---

**Description**

Hargreaves - Samani ETo

**Usage**

```
eto_hs(tmin, tmean, tmax, ra)
```

**Arguments**

tmin	A dataframe with Maximum daily air temperature (Celsius)
tmean	A dataframe with Mean daily air temperature (Celsius)
tmax	A dataframe with Maximum daily air temperature (Celsius)
ra	A dataframe of extraterrestrial radiation (MJ m <sup>-2</sup> day <sup>-1</sup> )

**Value**

Returns a data.frame object with the ETo HS data

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
eto_hs <- eto_hs(tmin, tmean, tmax, ra)  
  
## End(Not run)
```

---

etp\_thorntwaite            *Thorntwaite - Potential evapotranspiration*

---

**Description**

Thorntwaite - Potential evapotranspiration

**Usage**

```
etp_thorntwaite(tmean)
```

**Arguments**

tmean                    A dataframe with Mean monthly air temperature (Celsius)

**Value**

Returns a data.frame object with the Thorntwaite ETp data

**Author(s)**

Roberto Filgueiras.

**Examples**

```
## Not run:
etp <- etp_thorntwaite(tmean)

## End(Not run)
```

---

get\_max\_eto\_at\_location

*Get Max Reference Evapotranspiration Values by Geographic Location*

---

**Description**

Extract maximum reference evapotranspiration (max ETo) values for one or more locations of interest, using data from the dataset provided by Dias (2018).

**Usage**

```
get_max_eto_at_location(img, lat, long)
```

**Arguments**

img                    SpatRaster with the maximum reference evapotranspiration (ETo) grid downloaded from max\_eto\_grid\_download.

lat                    Numeric. Latitude of the location in decimal degrees.

long                   Numeric. Longitude of the location in decimal degrees.

**Value**

A ‘data.frame’ object containing the maximum reference evapotranspiration (ETo) values per pair of coordinates.

**Author(s)**

Roberto Filgueiras.

## Examples

```
## Not run:  
# Visualize Brazilian states (optional)  
  
max_eto_df <- get_max_reference_eto_by_location(img,  
                                              long = c(-51.95, -43.23),  
                                              lat = c(-23.52, -21.34))  
  
## End(Not run)
```

---

hourly\_weather\_station\_download

*Download of hourly data from automatic weather stations (AWS) of  
INMET-Brazil*

---

## Description

This function will download the hourly AWS data of INMET for whatever station of interest, based on the period of time selected (start\_date and end\_date) and station code.

## Usage

```
hourly_weather_station_download(stations, start_date, end_date)
```

## Arguments

stations	The station code (ID - WMO code) for download. To see the station ID, please see the function <code>*see_stations_info*</code> .
start_date	Date that start the investigation, should be in the following format (1958-01-01 /Year-Month-Day)
end_date	Date that end the investigation, should be in the following format (2017-12-31 /Year-Month-Day)

## Value

Returns a data.frame with the AWS data requested

## Author(s)

Roberto Filgueiras

## Examples

```
## Not run:
df <- hourly_weather_station_download(
  stations = c("A001", "A042"),
  start_date = "2022-08-12",
  end_date = "2022-08-16")

## End(Not run)
```

---

max\_eto\_grid\_download *Download maximum reference evapotranspiration (ETo) grids for Brazil*

---

## Description

Downloads maximum reference evapotranspiration (ETo) grids for Brazil, intended for irrigation design purposes. The dataset was developed by Dias (2018).

## Usage

```
max_eto_grid_download(dir_out, product = "max_12_months")
```

## Arguments

dir_out	Character. Directory where the downloaded raster file will be saved.
product	Character. Specifies which maximum ETo product to download. Available options include: <ul style="list-style-type: none"><li>• max_12_months: maximum ETo over the full year.</li><li>• max_jan to max_dec: monthly maximum ETo for each respective month (January to December).</li></ul>

## Value

A ‘SpatRaster’ object containing the downloaded maximum reference evapotranspiration (ETo) grid.

## Author(s)

Roberto Filgueiras.

## References

Dias, S. H. B. (2018). \*Evapotranspiração de referência para projeto de irrigação no Brasil utilizando o produto MOD16\*. Dissertação (Mestrado) – Universidade Federal de Viçosa.

**Examples**

```
## Not run:  
# Visualize Brazilian states (optional)  
see_brazil_states()  
  
# Download maximum ETo grid (annual)  
img_max_eto <- max_eto_grid_download(dir_out = "data/", product = "max_12_months")  
  
## End(Not run)
```

---

Patm

*Atmospheric pressure (Patm)*

---

**Description**

Atmospheric pressure (Patm)

**Usage**

Patm(z)

**Arguments**

z                    Elevation above sea level (m)

**Value**

Returns a data.frame object with the atmospheric pressure calculated.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
Patm <- Patm(z)  
  
## End(Not run)
```

---

psy_const	<i>Psychrometric constant</i>
-----------	-------------------------------

---

**Description**

Psychrometric constant (kPa/Celsius) is calculated in this function.

**Usage**

```
psy_const(Patm)
```

**Arguments**

Patm                    Atmospheric pressure (kPa)

**Value**

A data.frame object with the psychrometric constant calculated.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
psy_df <- psy_const(Patm)  
  
## End(Not run)
```

---

radiation_conversion	<i>Conversion factors for radiation</i>
----------------------	---

---

**Description**

Function to convert the radiation data. The conversion name can be understand as follow:

- conversion\_1 = MJ m<sup>-2</sup> day<sup>-1</sup> to J cm<sup>-2</sup> day<sup>-1</sup>;
- conversion\_2 = MJ m<sup>-2</sup> day<sup>-1</sup> to cal cm<sup>-2</sup> day<sup>-1</sup>;
- conversion\_3 = MJ m<sup>-2</sup> day<sup>-1</sup> to W m<sup>-2</sup>;
- conversion\_4 = MJ m<sup>-2</sup> day<sup>-1</sup> to mm day<sup>-1</sup>;
- conversion\_5 = cal cm<sup>-2</sup> day<sup>-1</sup> to MJ m<sup>-2</sup> day<sup>-1</sup>;
- conversion\_6 = cal cm<sup>-2</sup> day<sup>-1</sup> to J cm<sup>-2</sup> day<sup>-1</sup>;
- conversion\_7 = cal cm<sup>-2</sup> day<sup>-1</sup> to W m<sup>-2</sup>;

- conversion\_8 = cal cm-2 day-1 to mm day-1;
- conversion\_9 = W m-2 to MJ m-2 day-1;
- conversion\_10 = W m-2 to J cm-2 day-1;
- conversion\_11 = W m-2 to cal cm-2 day-1;
- conversion\_12 = W m-2 to mm day-1;
- conversion\_13 = mm day-1 to MJ m-2 day-1;
- conversion\_14 = mm day-1 to J cm-2 day-1;
- conversion\_15 = mm day-1 to cal cm-2 day-1;
- conversion\_16 = mm day-1 to W m-2.

### Usage

```
radiation_conversion(data_to_convert, conversion_name)
```

### Arguments

data\_to\_convert

A data.frame with radiation values to convert.

conversion\_name

A character with the conversion\_name summarize in the description of this function.

### Value

A data.frame object with the converted radiation.

### Author(s)

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

### Examples

```
## Not run:  
radiation_conversion_df <- radiation_conversion(  
  data_to_convert = df$rad,  
  conversion_name = "conversion_1"  
)  
  
## End(Not run)
```



---

ra_calculation	<i>Extraterrestrial radiation for daily periods (ra)</i>
----------------	--

---

**Description**

ra is expressed in MJ m<sup>-2</sup> day<sup>-1</sup>

**Usage**

```
ra_calculation(latitude, date)
```

**Arguments**

latitude	A dataframe with latitude in decimal degrees that you want to calculate the ra.
date	A dataframe with the dates that you want to calculate the ra.

**Value**

A data.frame with the extraterrestrial radiation for daily periods

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
ra <- ra_calculation(latitude, date)  
  
## End(Not run)
```

---

rh_calculation	<i>Relative humidity (rh) calculation</i>
----------------	---

---

**Description**

Relative humidity is calculated in this function based on minimum air temperature of the day and the air temperature of the moment.

**Usage**

```
rh_calculation(tmin, tmean)
```

**Arguments**

tmin	A dataframe with minimum daily air temperature (Celsius)
tmean	A dataframe with mean air temperature (Celsius) that you want to calculate the relative humidity.

**Value**

A data.frame object with the relative humidity calculated

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
rh <- rh_calculation(tmin, tmean)  
  
## End(Not run)
```

---

rnl\_calculation      *Net longwave radiation (rnl)*

---

**Description**

Net outgoing longwave radiation is calculate with this function

**Usage**

```
rnl_calculation(tmin, tmax, ea, rs, rso)
```

**Arguments**

tmin	A dataframe with Minimum daily air temperature (Celsius)
tmax	A dataframe with Maximum daily air temperature (Celsius)
ea	A dataframe with the actual vapour pressure (KPa).
rs	A dataframe with the incoming solar radiation (MJ m-2 day-1).
rso	A dataframe with the clear-sky radiation (MJ m-2 day-1)

**Value**

A data.frame object with the net longwave radiation.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
rnl_df <- rnl_calculation(tmin, tmax, ea, rs, rso)  
  
## End(Not run)
```

---

rns_calculation	<i>Net solar or net shortwave radiation (rns)</i>
-----------------	---

---

**Description**

The rns results form the balance between incoming and reflected solar radiation (MJ m<sup>-2</sup> day<sup>-1</sup>).

**Usage**

```
rns_calculation(albedo, rs)
```

**Arguments**

albedo	Albedo or canopy reflectance coefficient. The 0.23 is the value used for hypothetical grass reference crop (dimensionless).
rs	The incoming solar radiation (MJ m <sup>-2</sup> day <sup>-1</sup> ).

**Value**

A data.frame object with the net solar or net shortwave radiation data.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
ra <- rns_calculation(albedo, rs)  
  
## End(Not run)
```

---

rn_calculation	<i>Net radiation (rn)</i>
----------------	---------------------------

---

**Description**

The net radiation (MJ m<sup>-2</sup> day<sup>-1</sup>) is the difference between the incoming net shortwave radiation (rns) and the outgoing net longwave radiation (rnl).

**Usage**

```
rn_calculation(rns, rnl)
```

**Arguments**

rns	The incoming net shortwave radiation (MJ m <sup>-2</sup> day <sup>-1</sup> ).
rnl	The outgoing net longwave radiation (MJ m <sup>-2</sup> day <sup>-1</sup> ).

**Value**

A data.frame object with the net radiation data.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
rn <- rn_calculation(rns, rnl)  
  
## End(Not run)
```

---

rso_calculation_1	<i>Clear-sky solar radiation with calibrated values available</i>
-------------------	---

---

**Description**

Clear-sky solar radiation is calculated in this function for near sea level or when calibrated values for as and bs are available.

**Usage**

```
rso_calculation_1(as, bs, ra)
```

**Arguments**

as	A dataframe with latitude in decimal degrees that you want to calculate the ra. The values of as = 0.25 is recommended by Allen et al. (1998).
bs	A dataframe with the dates that you want to calculate the ra. The values of bs = 0.50 is recommended by Allen et al. (1998).
ra	Extraterrestrial radiation for daily periods (ra).

**Value**

A data.frame object with the clear-sky radiation data

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
rso_df <- rso_calculation_1(as, bs, ra)  
  
## End(Not run)
```

---

rso\_calculation\_2      *Clear-sky solar radiation when calibrated values are not available*

---

**Description**

Clear-sky solar radiation is calculated in this function for near sea level or when calibrated values for as and bs are available.

**Usage**

```
rso_calculation_2(z, ra)
```

**Arguments**

z	Station elevation above sea level (m)
ra	Extraterrestrial radiation for daily periods (ra).

**Value**

A data.frame object with the clear-sky solar radiation

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
rso_df <- rso_calculation_2(z, ra)  
  
## End(Not run)
```

---

rs\_nearby\_calculation *Solar radiation data from a nearby weather station*

---

**Description**

The solar radiation data is calculated based in a nearby weather station.

**Usage**

```
rs_nearby_calculation(rs_reg, ra_reg, ra)
```

**Arguments**

rs_reg	A dataframe with the solar radiation at the regional location (MJ m <sup>-2</sup> day <sup>-1</sup> ).
ra_reg	A dataframe with the extraterrestrial radiation at the regional location (MJ m <sup>-2</sup> day <sup>-1</sup> ).
ra	A dataframe with the extraterrestrial radiation for daily periods (ra).

**Value**

A data.frame object with the Solar radiation data based on a nearby weather station

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
rs_nearby_df <- rs_nearby_calculation(rs_reg, ra_reg, ra)  
  
## End(Not run)
```

---

see\_stations\_info      *Localization of the automatic weather station of INMET*

---

**Description**

Function to see the localization of the automatic weather station of INMET.

**Usage**

```
see_stations_info()
```

**Value**

A data.frame with informations of OMM code, latitude, longitude and altitude of all AWS stations available in INMET.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
see_stations_info()  
  
## End(Not run)
```

---

selectAWSstations      *Select Automatic Weather Stations*

---

**Description**

Select automatic weather stations of INMET based on sf object.

**Usage**

```
selectAWSstations(aoi, as_sf = FALSE)
```

**Arguments**

aoi                      sf object. Area of interest.  
as\_sf                    logical. Inmet stations inventory result should be export as sf object. Default = FALSE.

**Value**

a vector or sf object with station located at area of interest (aoi)

**Examples**

```
## Not run:

es = geobr::read_municipality(code_muni = "ES")

esStations = selectStations(aoi = es, as_sf = F); es

## End(Not run)
```

---

sr\_ang\_calculation      *Solar radiation based in Angstrom formula (sr\_ang)*

---

**Description**

If global radiation is not measure at station, it can be estimated with this function.

**Usage**

```
sr_ang_calculation(latitude, date, n, as, bs)
```

**Arguments**

latitude	A dataframe with latitude in decimal degrees that you want to calculate the ra.
date	A dataframe with the dates that you want to calculate the ra.
n	The actual duration of sunshine. This variable is recorded with Campbell-Stokes sunshine recorder.
as	A dataframe with latitude in decimal degrees that you want to calculate the ra. The values of as = 0.25 is recommended by Allen et al. (1998).
bs	A dataframe with the dates that you want to calculate the ra. The values of bs = 0.50 is recommended by Allen et al. (1998).

**Value**

A data.frame object with solar radiation data

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:
sr_ang <- sr_ang_calculation(latitude, date, n, as, bs)

## End(Not run)
```



---

sr\_tair\_calculation     *Solar radiation data derived from air temperature differences*

---

**Description**

If global radiation is not measure at station, it can be estimated with this function.

**Usage**

```
sr_tair_calculation(latitude, date, tmax, tmin, location_krs)
```

**Arguments**

latitude	A dataframe with latitude in decimal degrees that you want to calculate the ra.
date	A dataframe with the dates that you want to calculate the ra.
tmax	A dataframe with Maximum daily air temperature (Celsius)
tmin	A dataframe with Minimum daily air temperature (Celsius)
location_krs	Adjustment coefficient based in location. Please decide between "coastal or "interior". If coastal the krs will be 0.19, if interior the krs will be 0.16.

**Value**

A data.frame object with solar radiation data

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
sr_tair <- sr_tair_calculation(latitude, date, tmax, tmin, location_krs)  
  
## End(Not run)
```

---

u2\_calculation     *Wind speed at 2 meters high*

---

**Description**

Wind speed at two meters high can be calculated with this function.

**Usage**

```
u2_calculation(uz, z)
```

**Arguments**

uz                    measured wind speed at z meters above ground surface  
z                      height of measurement above ground surface.

**Value**

A data.frame with the wind speed at 2 meters high calculated.

**Author(s)**

Roberto Filgueiras, Luan P. Venancio, Catariny C. Aleman and Fernando F. da Cunha

**Examples**

```
## Not run:  
u2_df <- u2_calculation(uz, z)  
  
## End(Not run)
```

# Index

correction\_etc\_thornwaite, [2](#)

daily\_eto\_FAO56, [3](#)  
design\_eto, [4](#)  
download\_AWS\_INMET\_daily, [5](#)  
download\_climate\_normals, [6](#)

ea\_dew\_calculation, [7](#)  
ea\_rh\_calculation, [7](#)  
es\_calculation, [8](#)  
es\_ea\_calculation, [9](#)  
eto\_hs, [10](#)  
etc\_thorntwaite, [10](#)

get\_max\_eto\_at\_location, [11](#)

hourly\_weather\_station\_download, [12](#)

max\_eto\_grid\_download, [13](#)

Patm, [14](#)  
psy\_const, [15](#)

ra\_calculation, [17](#)  
radiation\_conversion, [15](#)  
rh\_calculation, [17](#)  
rn\_calculation, [20](#)  
rnl\_calculation, [18](#)  
rns\_calculation, [19](#)  
rs\_nearby\_calculation, [22](#)  
rso\_calculation\_1, [20](#)  
rso\_calculation\_2, [21](#)

see\_stations\_info, [23](#)  
selectAWSstations, [23](#)  
sr\_ang\_calculation, [24](#)  
sr\_tair\_calculation, [25](#)

u2\_calculation, [25](#)