Package: climex (via r-universe)

November 23, 2026

Type Package

Title Climate Data of Mexico from the NASA POWER API

Version 0.1.0

Description Climate data of Mexico from the NASA POWER API (daily, monthly or climatological temperature, precipitation, humidity, wind and radiation) for the 2,465 municipalities of Mexico or any set of points. One request per point with all the period, up to five at a time, with retries; the download can be stopped and resumed, and the result has one row per point and date.

License MIT + file LICENSE

URL https://github.com/fdzul/climex, https://fdzul.github.io/climex/

BugReports https://github.com/fdzul/climex/issues

Depends R (>= 4.1.0)

Imports curl, jsonlite, utils

Suggests knitr, rmarkdown, sf, testthat (>= 3.1.7)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en

LazyData false

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.1

Config/pak/sysreqs libssl-dev

Repository https://fdzul.r-universe.dev

Date/Publication 2026-11-23 03:18:31 UTC

RemoteUrl https://github.com/fdzul/climex

RemoteRef HEAD

RemoteSha 51b94ffa292163fab75950b1df1e368f3f4b6366

Contents

build_url 2
centroids 3
download_nasa_power_data 3
municipalities 5
parse_response 6

Index **7**

`build_url` *URL of one NASA POWER request for a point*

Description

URL of one NASA POWER request for a point

Usage

```
build_url(
  latitude,
  longitude,
  parameters,
  temporal_resolution = "daily",
  community = "RE",
  start = NULL,
  end = NULL
)
```

Arguments

`latitude`, `longitude`	Coordinates of the point (decimal degrees, WGS84).
`parameters`	NASA POWER parameters, e.g. `c("T2M", "PRECTOTCORR")`.
`temporal_resolution`	`"daily"`, `"monthly"` or `"climatology"`.
`community`	`"RE"`, `"AG"` or `"SB"`.
`start`, `end`	Dates as `"YYYYMMDD"` (for `"monthly"` only the year is used; not used for `"climatology"`).

Value

A character string with the URL.

Examples

```
build_url(20.97, -89.62, c("T2M", "PRECTOTCORR"), "daily", "AG",
          "20240101", "20241231")
```

centroids *Points inside polygons, to download their climate*

Description

Uses the representative point of each polygon (`sf::st_point_on_surface()`), which is always inside it; the centroid of a curved municipality can fall outside.

Usage

```
centroids(x, id_col)
```

Arguments

`x`	An `sf` object with polygons, e.g. the AGEB or localities of rgeomex.
`id_col`	Column with the identifier of each polygon (e.g. `"CVEGEO"`).

Value

A data frame with `id_col`, `latitude` and `longitude` (WGS84), ready for `download_nasa_power_data()`.

download_nasa_power_data *Download NASA POWER climate data for the points of a table*

Description

One request per point with all the period, up to five at a time. Each point is saved as `<identifier>.csv` in `output_folder`; a point whose CSV already has all the period (with data in every row) is not downloaded again, so an interrupted download can be run again and continues where it stopped, and with a later `end_date` (an update) the point is downloaded again. Busy answers of the API (429, 5xx) and network errors are retried; the points that still fail get a second round at the end.

Usage

```
download_nasa_power_data(
  df,
  lat_col = "latitude",
  lon_col = "longitude",
  start_date = "20150101",
  end_date = "20150305",
  parameters = c("T2M", "T2M_MAX", "T2M_MIN"),
  temporal_resolution = "daily",
  community = "RE",
  processes = 5,
```

```
  output_folder = "./nasa_power_data",
  start_col = NULL,
  end_col = NULL,
  id_col = NULL,
  return_consolidated = TRUE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`df`	Data frame with one row per point, e.g. `municipalities()`.
`lat_col`, `lon_col`	Columns with the coordinates (decimal degrees, WGS84).
`start_date`, `end_date`	Period as `"YYYYMMDD"`; for `"monthly"` only the year is used.
`parameters`	NASA POWER parameters. Temperature: `T2M`, `T2M_MAX`, `T2M_MIN`, `T2M_RANGE`, `T2MDEW`, `T2MWET`, `TS`; precipitation: `PRECTOTCORR`; humidity: `RH2M`, `QV2M`; wind: `WS2M`, `WS10M`; radiation: `ALLSKY_SFC_SW_DWN`; pressure: `PS`. Up to 20 per request.
`temporal_resolution`	`"daily"`, `"monthly"` or `"climatology"`.
`community`	`"RE"` (renewable energy), `"AG"` (agroclimatology) or `"SB"` (sustainable buildings); it changes the units of some parameters.
`processes`	Simultaneous requests (NASA asks for at most 5).
`output_folder`	Folder for the CSV of each point.
`start_col`, `end_col`	Optional columns with a different period for each point.
`id_col`	Optional column with the identifier of each point (else `idx_<row>`); its values must be unique.
`return_consolidated`	If `TRUE`, return the data of all the points in one table; if `FALSE`, the result of each download.
`overwrite`	Download again every point, even if its CSV already has the period.
`verbose`	Show the progress and the final report.

Value

A data frame with `identifier`, `latitude`, `longitude`, the date (`date`; `year` and `month`; or `period`) and one column per parameter, only for the period requested and with one row per point and date, sorted by point and date. Missing values (-999 in NASA POWER) are `NA`.

Examples

```
## Not run:
yuc <- municipalities("31")
```

```
clima <- download_nasa_power_data(yuc, id_col = "CVEGEO",
                                  parameters = c("T2M", "PRECTOTCORR"),
                                  start_date = "20240101", end_date = "20241231",
                                  community = "AG")

## End(Not run)
```

`municipalities` *One point per municipality of Mexico (INEGI 2019), all or by state*

Description

The table comes with the package (2,465 municipalities), so no download is needed; it is ready for `download_nasa_power_data()` with `id_col = "CVEGEO"`.

Usage

```
municipalities(cve_edo = NULL, point = c("inside", "centroid"))
```

Arguments

`cve_edo`	State code(s) (`"31"`, `31` or `c("14", "31")`). `NULL` returns the 2,465 municipalities of the country.
`point`	`"inside"` (default) uses a point that is always inside the municipality (its representative point); `"centroid"` uses the geometric centroid, which falls outside 67 curved or fragmented municipalities (column `centroid_inside`).

Value

A data frame with `CVEGEO`, `CVE_ENT`, `state`, `CVE_MUN`, `municipality`, `latitude`, `longitude` (WGS84), `centroid_lat`, `centroid_lon`, `centroid_inside` and `area_km2`.

Examples

```
yuc <- municipalities("31")        # the 106 municipalities of Yucatán
head(yuc[, c("CVEGEO", "municipality", "latitude", "longitude")])
nrow(municipalities())             # all the country
```

parse_response *Table from the JSON answer of NASA POWER*

Description

Table from the JSON answer of NASA POWER

Usage

```
parse_response(js, temporal_resolution = "daily")
```

Arguments

js	The JSON answer, as text or as the list of `jsonlite::fromJSON()`.
temporal_resolution	`"daily"`, `"monthly"` or `"climatology"`.

Value

A data frame with one row per day (`date`), year and month (`year`, `month`; the annual value 13 is dropped) or period (`period`), and one column per parameter. Missing values (-999) are `NA`.

Index

build_url, 2

centroids, 3

download_nasa_power_data, 3
download_nasa_power_data(), *3*, *5*

jsonlite::fromJSON(), *6*

municipalities, 5
municipalities(), *4*

parse_response, 6