Package 'chirps'

November 4, 2026

Type Package

Title Access Climate Hazards Center Datasets

Version 2.0.0

Description Provides access to Climate Hazards Center (CHC) datasets including CHIRPS v3, CHIRPS v2, CHIRTS-daily and CHIRTS-ERA5, read directly from the CHC server. 'CHIRPS' is a quasi-global (60°S to 60°N for v3) high-resolution (0.05 arc-degrees) rainfall data set, which incorporates satellite imagery and in-situ station data to create gridded rainfall time series for trend analysis and seasonal drought monitoring. 'CHIRTS' is a quasi-global (60°S to 70°N), high-resolution data set of daily maximum and minimum temperatures. Data are returned as rasters or as time series for points and polygons, ready for climate indices with the 'climatrends' package. For more details on CHC datasets please visit their official home page <https://www.chc.ucsb.edu/data>.

License MIT + file LICENSE

URL https://docs.ropensci.org/chirps/

BugReports https://github.com/ropensci/chirps/issues

Depends methods, R (>= 3.5.0)

Imports httr, jsonlite, sf, stats, terra (>= 1.7-0), utils

Suggests climatrends, curl, knitr, markdown, rmarkdown, testthat (>= 3.2.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

LazyData true

Language en-GB

RoxygenNote 7.3.2

NeedsCompilation no

Author Kauê de Sousa [aut, cre] (ORCID:
<https://orcid.org/0000-0002-7571-7845>),
Adam H. Sparks [aut] (ORCID: <https://orcid.org/0000-0002-0061-8359>),
Aniruddha Ghosh [aut] (ORCID: <https://orcid.org/0000-0003-3667-8019>),
Pete Peterson [ctb] (API Client implementation),
William Ashmall [ctb] (API Client implementation),
Jacob van Etten [ths] (ORCID: <https://orcid.org/0000-0001-7554-2558>),
Svein Ø. Solberg [ths] (ORCID: <https://orcid.org/0000-0002-4491-4483>),
Emanuele Cordano [aut] (ORCID: <https://orcid.org/0000-0002-3508-5898>)

Maintainer Kauê de Sousa <desousa.kaue@gmail.com>

Repository CRAN

Date/Publication 2026-11-04 00:10:16 UTC

Contents

as.geojson 2
chirps 3
get_chirps 4
get_chirps_raw 7
get_chirts 9
get_chirts_raw 11
get_esi 13
get_imerg 15
precip_indices 17
tapajos 18

Index **20**

`as.geojson` *Methods to coerce geographical coordinates into a geojson polygon*

Description

Take single points from geographical coordinates and coerce into a geojson of geometry 'Polygon'

Usage

```
as.geojson(lonlat, dist = 1e-05, nQuadSegs = 2L, ...)

## Default S3 method:
as.geojson(lonlat, dist = 1e-05, nQuadSegs = 2L, ...)

## S3 method for class 'sf'
as.geojson(lonlat, dist = 1e-05, nQuadSegs = 2L, ...)
```

Arguments

lonlat	a data.frame or matrix with geographical coordinates 'lonlat', in that order, or an object of class sf with geometry type 'POINT' or 'POLYGON'
dist	numeric, buffer distance for all lonlat
nQuadSegs	integer, number of segments per quadrant
...	further arguments passed to sf methods

Value

An object of class 'geosjon' for each row in lonlat

Examples

```
# Default S3 Method
# random geographic points within bbox(10, 12, 45, 47)
library("sf")

set.seed(123)
lonlat = data.frame(lon = runif(1, 10, 12),
                    lat = runif(1, 45, 47))

gjson = as.geojson(lonlat)

#################

# S3 Method for objects of class 'sf'
# random geographic points within bbox(10, 12, 45, 47)
library("sf")

set.seed(123)
lonlat = data.frame(lon = runif(5, 10, 12),
                    lat = runif(5, 45, 47))

lonlat = st_as_sf(lonlat, coords = c("lon","lat"))

gjson = as.geojson(lonlat)
```

---

chirps *API Client for CHIRPS and CHIRTS*

Description

Provides access to Climate Hazards Center (CHC) datasets including CHIRPS v3, CHIRPS v2, CHIRTS-daily and CHIRTS-ERA5, read directly from the CHC server. 'CHIRPS' is a quasi-global (60°S to 60°N for v3) high-resolution (0.05 arc-degrees) rainfall data set, which incorporates satellite imagery and in-situ station data to create gridded rainfall time series for trend analysis and seasonal drought monitoring. 'CHIRTS' is a quasi-global (60°S to 70°N), high-resolution data set

of daily maximum and minimum temperatures. Data are returned as rasters or as time series for points and polygons, ready for climate indices with the 'climatrends' package. For more details on CHC datasets please visit their official home page https://www.chc.ucsb.edu/data.

Note

While **chirps** does not redistribute the data or provide it in any way, we encourage users to cite Funk et al. (2015) when using CHIRPS and Funk et al. (2019) when using CHIRTS.

Funk et al. (2015). Scientific Data, 2, 150066. doi:10.1038/sdata.2015.66

Funk et al. (2019). Journal of Climate, 32(17), 5639-5658. doi:10.1175/JCLID180698.1

Author(s)

Kauê de Sousa and Adam H. Sparks and Aniruddha Ghosh

See Also

Useful links:

"JOSS paper:" doi:10.21105/joss.02419

"Development repository:" https://github.com/ropensci/chirps

"Static documentation:" https://docs.ropensci.org/chirps/

"Report bugs:" https://github.com/ropensci/chirps/issues

"CHC website:" https://www.chc.ucsb.edu

---

get_chirps *Get CHIRPS precipitation data*

Description

Get daily rainfall from CHIRPS, produced by the Climate Hazards Center (CHC), for points, polygons or an area. The data are read from the CHC server, which is the main source and serves both CHIRPS v3 (default) and CHIRPS v2.

Usage

```
get_chirps(object, dates, server = "CHC", ...)

## Default S3 method:
get_chirps(object, dates, server = "CHC", as.matrix = FALSE, ...)

## S3 method for class 'SpatVector'
get_chirps(object, dates, server = "CHC", as.raster = TRUE, ...)

## S3 method for class 'SpatRaster'
get_chirps(object, dates, server = "CHC", as.raster = TRUE, ...)
```

```
## S3 method for class 'SpatExtent'
get_chirps(object, dates, server = "CHC", as.raster = TRUE, ...)

## S3 method for class 'sf'
get_chirps(object, dates, server = "CHC", as.sf = FALSE, ...)

## S3 method for class 'sfc'
get_chirps(object, dates, server = "CHC", as.sf = FALSE, ...)

## S3 method for class 'geojson'
get_chirps(object, dates, server = "CHC", as.geojson = FALSE, ...)
```

Arguments

`object`	input, an object of class `data.frame` (or any other object that can be coerced to `data.frame`) with longitude and latitude in the first two columns, `SpatVector`, `SpatRaster`, `SpatExtent`, `sf`, `sfc` or `geojson`
`dates`	a character or Date vector "YYYY-MM-DD", with one date for a single day or two dates for the start and end of the period
`server`	a character, "CHC" (default and recommended) or "ClimateSERV". See the section on servers
`...`	additional arguments passed to `get_chirps_raw` and `extract`. See details
`as.matrix`	logical, returns a `matrix` with one row per location and one column per day, named "YYYY-MM-DD"
`as.raster`	logical, returns an object of class `SpatRaster`
`as.sf`	logical, returns an object of class `sf`
`as.geojson`	logical, returns an object of class `geojson`

Details

Additional arguments

version: CHIRPS version, "3.0" (default) or "2.0". CHC retires CHIRPS v2 after the December 2026 release

type: CHIRPS v3 product, "rnl" (default) or "sat". See `get_chirps_raw`

resolution: numeric, resolution of CHIRPS v2, 0.05 (default) or 0.25 degrees

operation: for polygons, the function used to summarise the cells of each polygon on each day

operation		**value**
max	=	0
min	=	1
median	=	2
sum	=	4
average	=	5 (*default value*)

Coverage: CHIRPS v3 covers 60°S to 60°N and CHIRPS v2 covers 50°S to 50°N, from 1981 to near present. Final data for a month arrive some weeks after it ends.

Polygons: with `sf`, `SpatVector` or `geojson` polygons, each day is summarised over the cells that touch the polygon, with the function set by **operation**. The `lon` and `lat` columns give the centroid of each polygon.

Messages: each request states what it downloads, here daily rainfall from CHIRPS, with the period, so it is not mistaken for temperature from `get_chirts`. Set `options(chirps.quiet = TRUE)` to hide these messages.

Indices: for rainfall indices (dry spells, heavy rain days, intensity) use `climatrends::rainfall()`. The matrix returned with `as.matrix = TRUE` is its input, see the examples.

Value

A data frame of CHIRPS data in the long format, ordered by `id` and `date`, or a matrix, raster, `sf` or `geojson` object as requested:

id the index for the rows in `object`

date the date of each value

lon the longitude as provided in `object`

lat the latitude as provided in `object`

chirps the CHIRPS value in mm

Servers

CHC (default) reads the CHIRPS files from `https://data.chc.ucsb.edu` through **terra**. It serves CHIRPS v3 and v2, gives the value of the cell that holds each point, and handles many points and long periods in one call. It is the recommended source.

ClimateSERV is kept for backward compatibility. It serves CHIRPS v2 only, averaged over a small buffer around each point, so its values can differ from the CHC values. It sends several requests per location and is slow for more than a few points. ClimateSERV support for CHIRPS may be removed in a future release. Its extra arguments are **dist**, the buffer distance around each point, **nQuadSegs**, the number of segments per buffer quadrant, and **operation** as above.

`SpatVector`, `SpatRaster` and `SpatExtent` input always uses CHC.

References

Funk, C., Peterson, P., Harrison, L. et al. (2026). Scientific Data, 13, 718.
doi:10.1038/s41597026070964

Funk C. et al. (2015). Scientific Data, 2, 150066.
doi:10.1038/sdata.2015.66

See Also

`get_chirps_raw` for the rasters, `get_chirts` for temperature, and the **climatrends** package for rainfall and temperature indices

Examples

```
library("chirps")
library("terra")

dates = c("2017-12-15", "2017-12-31")
lonlat = data.frame(lon = c(-55.0281, -54.9857), lat = c(-2.8094, -2.8756))

# Case 1: return a data.frame in the long format
r1 = get_chirps(lonlat, dates)

# Case 2: a single day, from the CHIRPS v3 product based on IMERG
r2 = get_chirps(lonlat, "2017-12-15", type = "sat")

# Case 3: return a matrix, one column per day
r3 = get_chirps(lonlat, dates, as.matrix = TRUE)

# Case 4: input SpatVector and return raster
f = system.file("ex/lux.shp", package = "terra")
v = vect(f)
r4 = get_chirps(v, dates, as.raster = TRUE)

# Case 5: input SpatExtent and return a raster within the extent
area = ext(c(-66, -64, -6, -4))
r5 = get_chirps(area, dates)

# Case 6: total rainfall per polygon and day
r6 = get_chirps(v, dates, as.raster = FALSE, operation = 4)

# Case 7: rainfall indices with climatrends
if (requireNamespace("climatrends", quietly = TRUE)) {
  climatrends::rainfall(r3, day.one = dates[1], span = ncol(r3))
}
```

---

get_chirps_raw *Get raw CHIRPS raster data*

Description

Download Global CHIRPS precipitation data in its original raster format.

Usage

```
get_chirps_raw(
  dates,
  version = "3.0",
  resolution = 0.05,
  type = c("rnl", "sat"),
  verbose = FALSE,
```

```
  ...
)
```

Arguments

`dates`	Character vector of length two with start and end dates in `"YYYY-MM-DD"` format.
`version`	Character or numeric. CHIRPS version. Supported values are `"3.0"` (default) and `"2.0"`. CHC retires CHIRPS v2 after the December 2026 release.
`resolution`	Numeric. Spatial resolution for CHIRPS v2. Supported values are `0.05` and `0.25`. Ignored for CHIRPS v3.
`type`	Character. CHIRPS v3 product type. Supported values are `"rnl"` and `"sat"`. Ignored for CHIRPS v2.
`verbose`	Logical. If `TRUE`, prints the URLs used.
`...`	Additional arguments reserved for future use.

Details

This function returns CHIRPS raster layers as a `SpatRaster` without extracting values, reshaping the data, or converting it to a data frame. It is intended for users who need direct access to the original raster product for custom spatial workflows.

CHIRPS v3 (Climate Hazards Center Infrared Precipitation with Stations) is a high-resolution (0.05°), quasi-global rainfall dataset spanning 60°N to 60°S from 1981 to near-present.

CHIRPS v3 combines satellite-based thermal infrared rainfall estimates with in-situ station observations to produce gridded precipitation estimates over land.

Two daily products are available:

rnl Daily values derived using ECMWF ERA5 precipitation to partition pentadal CHIRPS totals into daily amounts.

sat Daily values derived using NASA IMERG precipitation to partition pentadal CHIRPS totals into daily amounts.

Additional information is available at: `https://www.chc.ucsb.edu/data/chirps3`

Messages: each request states what it downloads, here daily rainfall from CHIRPS, with the period, so it is not mistaken for temperature from `get_chirts`. Set `options(chirps.quiet = TRUE)` to hide these messages.

Daily product documentation: `https://data.chc.ucsb.edu/products/CHIRPS/v3.0/daily/readme.txt`

Value

A `SpatRaster` with one layer per day.

References

Climate Hazards Center Infrared Precipitation with Stations version 3 (CHIRPS3) Data Repository. doi:10.15780/G2JQ0P

Version 3 Funk, C., Peterson, P., Harrison, L. et al. (2026). The Climate Hazards Center Infrared Precipitation with Stations, Version 3. Scientific Data, 13, 718. doi:10.1038/s41597026070964

Version 2 Funk C. et al. (2015). Scientific Data, 2, 150066. doi:10.1038/sdata.2015.66

Examples

```
if (interactive()) {
r1 = get_chirps_raw(dates = c("2017-12-15", "2017-12-31"),
                    version = "3.0",
                    type = "rnl")

r2 = get_chirps_raw(dates = c("2017-12-15", "2017-12-31"),
                    version = "3.0",
                    type = "sat")
}
```

`get_chirts` *Get CHIRTS temperature data*

Description

Get daily temperature from the Climate Hazards Center (CHC) for points, polygons or an area. Two products are available. CHIRTS-daily (default) covers 1983 to 2016 and has maximum and minimum temperature, relative humidity and heat index. CHIRTS-ERA5 extends maximum and minimum temperature from 1980 to near present.

Usage

```
get_chirts(object, dates, var, ...)

## Default S3 method:
get_chirts(object, dates, var, as.matrix = FALSE, ...)

## S3 method for class 'SpatVector'
get_chirts(object, dates, var, as.raster = TRUE, ...)

## S3 method for class 'SpatRaster'
get_chirts(object, dates, var, as.raster = TRUE, ...)

## S3 method for class 'SpatExtent'
get_chirts(object, dates, var, as.raster = TRUE, ...)
```

Arguments

object	an object of class data.frame (or any other object that can be coerced to a data.frame) with longitude and latitude in the first two columns, sf points, SpatVector, SpatRaster or SpatExtent
dates	a character or Date vector "YYYY-MM-DD", with one date for a single day or two dates for the start and end of the period
var	character, A valid variable from the options: "Tmax", "Tmin", "RHum" and "HeatIndex". CHIRTS-ERA5 has "Tmax" and "Tmin" only
...	further arguments passed to get_chirts_raw and extract. See details
as.matrix	logical, returns a matrix with one row per location and one column per day, named "YYYY-MM-DD"
as.raster	logical, returns an object of class SpatRaster

Details

Variable description from https://data.chc.ucsb.edu/products/CHIRTS/daily/aaa.Readme.txt

Tmax Daily average maximum air temperature at 2 m above ground

Tmin Daily average minimum air temperature at 2 m above ground

RHum Daily average relative humidity

HeatIndex Daily average heat index

Additional arguments

product: "CHIRTS-daily" (default, 1983 to 2016) or "CHIRTS-ERA5" (1980 to near present, Tmax and Tmin)

operation: for polygons, the function used to summarise the cells of each polygon on each day, with the codes listed in get_chirps (average by default)

Coverage: both products cover 60°S to 70°N.

Messages: each request states what it downloads, here daily temperature or humidity from CHIRTS, with the period, so it is not mistaken for rainfall from get_chirps. Set options(chirps.quiet = TRUE) to hide these messages.

Indices: for temperature indices (growing degree days, crop sensitive indices, frost and heat days) use the **climatrends** package, which takes the matrices returned with as.matrix = TRUE.

Value

A data frame in the long format, ordered by id and date, with the value in the column chirts. A SpatRaster if as.raster = TRUE, or a matrix with one column per day, named "YYYY-MM-DD", if as.matrix = TRUE

See Also

get_chirts_raw

Examples

```
library("chirps")
library("terra")

# Case 1: input a data frame return a data frame in the long format
dates = c("2010-12-15","2010-12-31")
lonlat = data.frame(lon = c(-55.0281,-54.9857),
                    lat = c(-2.8094, -2.8756))

temp1 = get_chirts(lonlat, dates, var = "Tmax")

# Case 2: input a data frame return a matrix
temp2 = get_chirts(lonlat, dates, "Tmax", as.matrix = TRUE)

# Case 3: recent dates from CHIRTS-ERA5
temp3 = get_chirts(lonlat, c("2024-01-01", "2024-01-31"), var = "Tmin",
                   product = "CHIRTS-ERA5")

# Case 4: input a SpatVector and return raster
f = system.file("ex/lux.shp", package="terra")
v = vect(f)
temp4 = get_chirts(v, dates, var = "Tmax", as.raster = TRUE)

# Case 5: average per polygon and day
temp5 = get_chirts(v, dates, var = "Tmax", as.raster = FALSE)

# Case 6: indices with climatrends
if (requireNamespace("climatrends", quietly = TRUE)) {
  tmin = get_chirts(lonlat, dates, var = "Tmin", as.matrix = TRUE)
  temp = array(c(temp2, tmin), dim = c(dim(temp2), 2),
               dimnames = list(NULL, colnames(temp2), NULL))
  climatrends::temperature(temp, day.one = dates[1], span = ncol(temp2))
}
```

`get_chirts_raw` *Get raw CHIRTS raster data*

Description

Download CHIRTS daily temperature data in its original raster format, from CHIRTS-daily (default) or CHIRTS-ERA5.

Usage

```
get_chirts_raw(
  dates,
  var,
  product = c("CHIRTS-daily", "CHIRTS-ERA5"),
```

```
  resolution = 0.05,
  coverage = "global",
  interval = "daily",
  format = "tifs",
  version = "1.0",
  use_vsicurl = TRUE,
  verbose = FALSE,
  ...
)
```

Arguments

dates	Character or Date vector "YYYY-MM-DD", with one date for a single day or two dates for the start and end of the period.
var	Character. CHIRTS variable. Supported values are "Tmax", "Tmin", "RHum", and "HeatIndex". CHIRTS-ERA5 has "Tmax" and "Tmin" only.
product	Character. "CHIRTS-daily" (default, 1983 to 2016) or "CHIRTS-ERA5" (1980 to near present).
resolution	Numeric. Spatial resolution. Default is 0.05. CHIRTS-daily only.
coverage	Character. Spatial coverage. Default is "global". CHIRTS-daily only.
interval	Character. Temporal interval. Default is "daily". Currently retained for API consistency.
format	Character. File format folder. Default is "tifs". CHIRTS-daily only.
version	Character or numeric. CHIRTS-daily version. Default is "1.0".
use_vsicurl	Logical. If TRUE, prepends "/vsicurl" to remote URLs.
verbose	Logical. If TRUE, prints the URLs used.
...	Additional arguments reserved for future use.

Details

This function returns CHIRTS raster layers as a SpatRaster without extracting values, reshaping the data, or converting it to a data frame.

CHIRTS-daily (Climate Hazards Center Infrared Temperature with Stations) is a quasi-global (60°S to 70°N), high-resolution (0.05 degrees) dataset of daily maximum and minimum temperatures, relative humidity and heat index covering 1983-2016. It combines monthly CHIRTS temperature fields with daily temperature variability from the ERA5 reanalysis. Daily minimum temperatures are derived using the ERA5 diurnal temperature range. See https://www.chc.ucsb.edu/data/chirtsdaily.

CHIRTS-ERA5 is a quasi-global (60°S to 70°N), high-resolution (0.05 degrees) dataset of daily maximum and minimum temperatures from 1980 to near present. It combines CHIRTS (1983-2016) with the regularly updated ERA5 reanalysis, downscaled and bias-corrected relative to CHIRTS. It is an experimental CHC product. Users should cite "CHIRTS-ERA5 Data Repository. https://doi.org/10.15780/G2F08J. Data accessed on [DATE]." See https://www.chc.ucsb.edu/data/chirts-era5.

Messages: each request states what it downloads, here daily temperature or humidity from CHIRTS, with the period, so it is not mistaken for rainfall from get_chirps. Set options(chirps.quiet = TRUE) to hide these messages.

Value

A SpatRaster with one layer per day.

References

Verdin, A., Funk, C., Peterson, P., Landsfeld, M., Tuholske, C., and Grace, K. (2020). Development and validation of the CHIRTS-daily quasi-global high-resolution daily temperature data set. Scientific Data, 7, 303. doi:10.1038/s41597020006437

Funk, C., Peterson, P., Peterson, S., Shukla, S., Davenport, F., Michaelsen, J., et al. (2019). A High-Resolution 1983-2016 Tmax Climate Data Record Based on Infrared Temperatures and Stations by the Climate Hazards Center. Journal of Climate, 32, 5639-5658. doi:10.1175/JCLID180698.1

Climate Hazards Center (CHC). CHIRTS-ERA5 Data Repository. doi:10.15780/G2F08J

Examples

```
if (interactive()) {
  tmax = get_chirts_raw(dates = c("2015-01-01", "2015-01-31"),
                        var = "Tmax")

  # recent dates from CHIRTS-ERA5
  tmin = get_chirts_raw(dates = c("2024-01-01", "2024-01-31"),
                        var = "Tmin",
                        product = "CHIRTS-ERA5")
}
```

---

get_esi *Get evaporative stress index (ESI) data*

Description

Get evaporative stress index (ESI) from SERVIR Global via ClimateSERV API Client. ESI is available every four (or twelve) weeks from 2001 to present. The dataset may contain cloudy data which is returned as NAs. ClimateSERV works with 'geojson' of type 'Polygon'. The input `object` is then transformed into polygons with a small buffer area around the point.

Usage

```
get_esi(object, dates, operation = 5, period = 1, ...)

## Default S3 method:
get_esi(object, dates, operation = 5, period = 1, ...)

## S3 method for class 'sf'
get_esi(object, dates, operation = 5, period = 1, as.sf = FALSE, ...)

## S3 method for class 'geojson'
get_esi(object, dates, operation = 5, period = 1, as.geojson = FALSE, ...)
```

Arguments

`object`	input, an object of class `data.frame` (or any other object that can be coerced to `data.frame`) with longitude and latitude in the first two columns, `SpatVector`, `SpatRaster`, `SpatExtent`, `sf`, `sfc` or `geojson`
`dates`	a character or Date vector "YYYY-MM-DD", with one date for a single day or two dates for the start and end of the period
`operation`	optional, an integer that represents which type of statistical operation to perform on the dataset
`period`	an integer value for the period of ESI data, four weeks period = 1, twelve weeks = 2
`...`	additional arguments passed to `get_chirps_raw` and `extract`. See details
`as.sf`	logical, returns an object of class `sf`
`as.geojson`	logical, returns an object of class `geojson`

Details

operation: supported operations are:

operation		**value**
max	=	0
min	=	1
median	=	2
sum	=	4
average	=	5 (*default value*)

dist: numeric, buffer distance for each `object` coordinate

nQuadSegs: integer, number of segments per buffer quadrant

Value

A data frame of ESI data:

`id`	the index for the rows in `object`
`dates`	the dates from which ESI was requested
`lon`	the longitude as provided in `object`
`lat`	the latitude as provided in `object`
`esi`	the ESI value

Future deprecation

`get_esi()` reads from the ClimateSERV API, which **chirps** no longer recommends. `get_esi()` will be deprecated in a future release, and a notice says so once per session. CHIRPS rainfall is available from the CHC server with `get_chirps()`.

Note

get_esi() may return some warning messages given by sf, please check the sf documentation for possible issues.

Examples

```
lonlat = data.frame(lon = c(-55.0281,-54.9857),
                    lat = c(-2.8094, -2.8756))

dates = c("2017-12-15","2018-06-20")

# by default the function sets a very small buffer around the points which
# can return NAs due to cloudiness in ESI data

dt = get_esi(lonlat, dates = dates)

# the argument dist passed through sf increase the buffer area

dt = get_esi(lonlat, dates = dates, dist = 0.1)
```

---

get_imerg *Get Integrated Multisatellite Retrievals for GPM (IMERG) data*

Description

The IMERG dataset provides near-real time global observations of rainfall at 10km resolution, which can be used to estimate total rainfall accumulation from storm systems and quantify the intensity of rainfall and flood impacts from tropical cyclones and other storm systems. IMERG is a daily precipitation dataset available from 2015 to present within the latitudes 70 and -70 degrees.

Usage

```
get_imerg(object, dates, operation = 5, ...)

## Default S3 method:
get_imerg(object, dates, operation = 5, ...)

## S3 method for class 'sf'
get_imerg(object, dates, operation = 5, as.sf = FALSE, ...)

## S3 method for class 'geojson'
get_imerg(object, dates, operation = 5, as.geojson = FALSE, ...)
```

Arguments

`object`	input, an object of class `data.frame` (or any other object that can be coerced to `data.frame`) with longitude and latitude in the first two columns, `SpatVector`, `SpatRaster`, `SpatExtent`, `sf`, `sfc` or `geojson`
`dates`	a character or Date vector "YYYY-MM-DD", with one date for a single day or two dates for the start and end of the period
`operation`	optional, an integer that represents which type of statistical operation to perform on the dataset
`...`	additional arguments passed to `get_chirps_raw` and `extract`. See details
`as.sf`	logical, returns an object of class `sf`
`as.geojson`	logical, returns an object of class `geojson`

Details

operation: supported operations are:

operation		**value**
max	=	0
min	=	1
median	=	2
sum	=	4
average	=	5 (*default value*)

dist: numeric, buffer distance for each `object` coordinate

nQuadSegs: integer, number of segments per buffer quadrant

Value

A data frame of IMERG data:

`id`	the index for the rows in `object`
`dates`	the dates from which imerg was requested
`lon`	the longitude as provided in `object`
`lat`	the latitude as provided in `object`
`imerg`	the IMERG value

Future deprecation

`get_imerg()` reads from the ClimateSERV API, which **chirps** no longer recommends. `get_imerg()` will be deprecated in a future release, and a notice says so once per session. CHIRPS rainfall is available from the CHC server with `get_chirps()`.

Examples

```
lonlat = data.frame(lon = c(-55.0281,-54.9857),
                    lat = c(-2.8094, -2.8756))

dates = c("2017-12-15", "2017-12-31")

dt = get_imerg(lonlat, dates)

dt
```

`precip_indices` *Compute simple precipitation indices over a time series*

Description

A simple example of how to compute rainfall indices from the data returned by `get_chirps` or `get_imerg`. For analysis, use the **climatrends** package instead. `climatrends::rainfall()` computes the same indices with tested definitions and flexible periods per location, and climatrends also has temperature, growing degree days, evapotranspiration and crop sensitive indices. See the examples for how to pass CHIRPS data to it.

Usage

```
precip_indices(object, timeseries = FALSE, intervals = NULL)
```

Arguments

`object`	an object of class `chirps` as provided by `get_chirps`
`timeseries`	logical, `FALSE` for a single point time series observation or `TRUE` for a time series based on *intervals*
`intervals`	integer no lower than 5, for the days intervals when *timeseries* = `TRUE`

Value

A data frame with precipitation indices:

`MLDS`	maximum length of consecutive dry day, rain < 1 mm (days)
`MLWS`	maximum length of consecutive wet days, rain >= 1 mm (days)
`R10mm`	number of heavy precipitation days 10 >= rain < 20 mm (days)
`R20mm`	number of very heavy precipitation days rain >= 20 (days)
`Rx1day`	maximum 1-day precipitation (mm)
`Rx5day`	maximum 5-day precipitation (mm)
`R95p`	total precipitation when rain > 95th percentile (mm), with the percentile taken over all days in the period, dry days included

R99p	total precipitation when rain > 99th percentile (mm), with the percentile taken over all days in the period, dry days included
Rtotal	total precipitation (mm) in wet days, rain >= 1 (mm)
SDII	simple daily intensity index, total precipitation in wet days divided by the number of wet days (mm/days)

References

Aguilar E., et al. (2005). Journal of Geophysical Research, 110(D23), D23107.

de Sousa K., van Etten J., Neby M., Solberg S. Ø. (2023). Climate variability indices for ecological and crop models in R: the climatrends package. Journal of Open Source Software, 8(85), 4405. doi:10.21105/joss.04405

Kehel Z., et al. (2016). In: Applied Mathematics and Omics to Assess Crop Genetic Resources for Climate Change Adaptive Traits (eds Bari A., Damania A. B., Mackay M., Dayanandan S.), pp. 151-174. CRC Press.

See Also

`climatrends::rainfall()`

Examples

```
lonlat = data.frame(lon = c(-55.0281,-54.9857),
                    lat = c(-2.8094, -2.8756))

dates = c("2017-12-15", "2017-12-31")

dt = get_chirps(lonlat, dates)

# take the indices for the entire period
precip_indices(dt, timeseries = FALSE)

# take the indices for periods of 7 days
precip_indices(dt, timeseries = TRUE, intervals = 7)

# recommended, the same data in climatrends
if (requireNamespace("climatrends", quietly = TRUE)) {
  rain = get_chirps(lonlat, dates, as.matrix = TRUE)
  climatrends::rainfall(rain, day.one = dates[1], span = ncol(rain))
}
```

---

tapajos *Tapajós National Forest*

Description

Geometries for the Tapajós National Forest, a protected area in the Brazilian Amazon

Usage

```
tapajos
```

Format

An object of class 'sfc_POLYGON' within the bounding box xmin: -55.41127 ymin: -4.114584 xmax: -54.7973 ymax: -2.751706

Source

The data was provided by the Chico Mendes Institute for Biodiversity Conservation (ICMBio) via https://www.protectedplanet.net/en

Index

∗ **datasets**
    tapajos, 18
∗ **utility functions**
    as.geojson, 2

as.geojson, 2

chirps, 3
chirps-package *(*chirps*)*, 3

data.frame, *5*, *10*, *14*, *16*

extract, *5*, *10*, *14*, *16*

get_chirps, 4, *10*, *12*, *14*, *16, 17*
get_chirps_raw, *5, 6*, 7, *14*, *16*
get_chirts, *6*, *8*, 9
get_chirts_raw, *10*, 11
get_esi, 13
get_imerg, 15, *17*

precip_indices, 17

sf, *3*, *5*, *10*, *14–16*
SpatExtent, *5*, *10*, *14*, *16*
SpatRaster, *5*, *8*, *10*, *12–14*, *16*
SpatVector, *5*, *10*, *14*, *16*

tapajos, 18