

Package: chirps (via r-universe)

July 31, 2026

Type Package

Title Access Climate Hazards Center Datasets

Version 2.0

Description Provides access to Climate Hazards Center (CHC) datasets including CHIRPS v2, CHIRPS v3, CHIRTS-daily, and CHIRTS-ERA5. The 'CHIRPS' data is a quasi-global (50°S – 50°N) 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 – 70°N), high-resolution data set of daily maximum and minimum temperatures. 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 (>= 1.2-10)

Suggests climatrends, knitr, markdown, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

RoxygenNote 7.3.2

Config/pak/sysreqs libabsl-dev cmake libgdal-dev gdal-bin libgeos-dev libssl-dev libproj-dev libsqlite3-dev libudunits2-dev

Repository <https://ropensci.r-universe.dev>

Date/Publication 2026-06-23 08:27:34 UTC

RemoteUrl <https://github.com/ropensci/chirps>

RemoteRef master

RemoteSha 4c4dc3caffae43e7bdfb12f7523a9170a92e2cea

Contents

as.geojson	2
chirps	3
get_chirps	4
get_chirps_raw	6
get_chirts	8
get_chirts_era5_raw	10
get_chirts_raw	11
get_esi	13
get_imerg	14
precip_indices	16
tapajos	17
Index	18

as.geojson	<i>Methods to coerce geographical coordinates into a geojson polygon</i>
------------	--

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, CHIRPS v3, CHIRTS-daily, and CHIRTS-ERA5. The 'CHIRPS' data is a quasi-global (50°S – 50°N) 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 – 70°N), high-resolution data set of daily maximum and minimum temperatures. For more details on 'CHIRPS' and 'CHIRTS' data please visit its 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/JCLI180698.1

Author(s)

Kauê de Sousa and Adam H. Sparks and Aniruddha Ghosh

See Also**Useful links:****"JOSS paper:"** [doi:10.21105/joss.02419](https://doi.org/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 precipitation data from the "Climate Hazards Group". Two server sources are available. The first, "CHC" (default) is recommended for multiple data-points, while "ClimateSERV" is recommended when few data-points are required (~ 50).

Usage

```
get_chirps(object, dates, server, ...)
```

Default S3 method:

```
get_chirps(object, dates, server, 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.matrix = TRUE,
  as.raster = FALSE,
  ...
)
```

S3 method for class 'SpatExtent'

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

S3 method for class 'sf'

```
get_chirps(object, dates, server, as.sf = FALSE, ...)
```

S3 method for class 'geojson'

```
get_chirps(object, dates, server, as.geojson = FALSE, ...)

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

Arguments

<code>object</code>	input, an object of class <code>data.frame</code> (or any other object that can be coerced to <code>data.frame</code>), <code>SpatVector</code> , <code>SpatRaster</code> , <code>sf</code> or <code>geojson</code>
<code>dates</code>	a character of start and end dates in that order in the format "YYYY-MM-DD"
<code>server</code>	a character that represents the server source "CHC" or "ClimateSERV"
<code>...</code>	additional arguments passed to <code>terra</code> or <code>sf</code> methods See details
<code>as.matrix</code>	logical, returns an object of class <code>matrix</code>
<code>as.raster</code>	logical, returns an object of class <code>SpatRaster</code>
<code>as.sf</code>	logical, returns an object of class <code>sf</code>
<code>as.geojson</code>	logical, returns an object of class <code>geojson</code>

Details

Data description at <https://data.chc.ucsb.edu/products/CHIRPS-2.0/README-CHIRPS.txt>

Additional arguments when using server = "CHC"

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

Additional arguments when using server = "ClimateSERV"

dist: numeric, buffer distance for each object coordinate

nQuadSegs: integer, number of segments per buffer quadrant

operation: supported operations for ClimateSERV are:

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

Value

A matrix, raster or a data frame of CHIRPS data:

id the index for the rows in object
dates the dates from which CHIRPS was requested
lon the longitude as provided in object
lat the latitude as provided in object
chirps the CHIRPS value in mm

Note

get_chirps() may return some warning messages given by [sf](#), please look [sf](#) documentation for possible issues.

References

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

Examples

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

# Case 1: return as a data.frame
dates = c("2017-12-15", "2017-12-31")
lonlat = data.frame(lon = c(-55.0281, -54.9857), lat = c(-2.8094, -2.8756))

r1 = get_chirps(lonlat, dates, server = "CHC")

# Case 2: return a matrix
r2 = get_chirps(lonlat, dates, server = "CHC", as.matrix = TRUE)

# Case 3: input SpatVector and return raster
f = system.file("ex/lux.shp", package = "terra")
v = vect(f)
r3 = get_chirps(v, dates, server = "CHC", as.raster = TRUE)

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

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

r4 = get_chirps(area, dates, server = "CHC")

# Case 5: using the server "ClimateSERV"
r5 = get_chirps(lonlat, dates, server = "ClimateSERV")

# Case 6: from "ClimateSERV" and return as a matrix
r6 = get_chirps(lonlat, dates, server = "ClimateSERV", as.matrix = TRUE)
```

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 = "2.0",
  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 "2.0" and "3.0".
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>

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](https://doi.org/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](https://doi.org/10.1038/s41597026070964)

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

Examples

```
if (interactive()) {
  r1 = get_chirps_raw(dates = c("2017-12-15", "2017-12-31"),
                     version = "2.0")

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

get_chirts

Get CHIRTS temperature data

Description

Get daily maximum and minimum temperature data from the "Climate Hazards Group". CHIRTS-daily is a global 2-m temperature product that combines the monthly CHIRTSmax data set with the ERA5 reanalysis to produce routinely updated data to support the monitoring of temperature extreme. Data is currently available from 1983 to 2016. Soon available 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 <code>data.frame</code> (or any other object that can be coerced to a <code>data.frame</code>), <code>SpatVector</code> , or <code>SpatRaster</code>
dates	a character of start and end dates in that order in the format "YYYY-MM-DD"
var	character, A valid variable from the options: "Tmax", "Tmin", "RHum" and "HeatIndex"
...	further arguments passed to <code>terra</code>
as.matrix	logical, returns an object of class <code>matrix</code>
as.raster	logical, returns an object of class <code>SpatRaster</code>

Details

Variable description from <https://data.chc.ucsb.edu/products/CHIRTSdaily/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

Value

A `SpatRaster` object if `as.raster=TRUE`, else `matrix`, `list`, or `data.frame`

Additional arguments

interval: supported intervals are "daily", "pentad", "dekad", "monthly", "2-monthly", "3-monthly", and "annual". Currently hard coded to "daily".

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: input a raster and return raster
f = system.file("ex/lux.shp", package="terra")
v = vect(f)
temp3 = get_chirts(v, dates, var = "Tmax", as.raster = TRUE)
```

```
# Case 4: input a raster and return raster
temp4 = get_chirts(v, dates, var = "Tmax", as.matrix = TRUE)
```

get_chirts_era5_raw *Get raw CHIRTS-ERA5 raster data*

Description

Download CHIRTS-ERA5 daily temperature data in its original raster format.

Usage

```
get_chirts_era5_raw(
  dates,
  var = c("Tmin", "Tmax"),
  use_vsicurl = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

dates	Character vector of length two with start and end dates in "YYYY-MM-DD" format.
var	Character. Variable to download. Supported values are "Tmin" and "Tmax".
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

CHIRTS-ERA5 is a quasi-global (60°S to 70°N), high-resolution ($0.05^\circ \times 0.05^\circ$) dataset of daily maximum and minimum temperatures, heat index, and wet bulb globe temperature from 1980 to near-present.

The dataset combines the Climate Hazards Center Infrared Temperature with Stations (CHIRTS) product (1983–2016) with the regularly updated ERA5 reanalysis. ERA5 fields are downscaled and bias-corrected relative to CHIRTS to provide a temporally consistent climate record.

Users of this dataset should cite:

"CHIRTS-ERA5 Data Repository. <https://doi.org/10.15780/G2F08J>. Data accessed on [DATE]."

Additional information is available at: <https://www.chc.ucsb.edu/data/chirts-era5>

Value

A [SpatRaster](#) with one layer per day.

References

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

Examples

```
if (interactive()) {
  tmin = get_chirts_era5_raw(dates = c("1980-01-01", "1980-01-31"),
                             var = "Tmin")

  tmax = get_chirts_era5_raw(dates = c("1980-01-01", "1980-01-31"),
                             var = "Tmax")
}
```

get_chirts_raw	<i>Get raw CHIRTS raster data</i>
----------------	-----------------------------------

Description

Download CHIRTS daily temperature and humidity data in its original raster format.

Usage

```
get_chirts_raw(
  dates,
  var,
  resolution = 0.05,
  coverage = "global",
  interval = "daily",
  format = "tifs",
  version = "1.0",
  use_vsicurl = FALSE,
  verbose = FALSE,
  ...
)
```

Arguments

dates	Character vector of length two with start and end dates in "YYYY-MM-DD" format.
var	Character. CHIRTS variable. Supported values are "Tmax", "Tmin", "RHum", and "HeatIndex".
resolution	Numeric. Spatial resolution. Default is 0.05.
coverage	Character. Spatial coverage. Default is "global".
interval	Character. Temporal interval. Default is "daily". Currently retained for API consistency.
format	Character. File format folder. Default is "tifs".

version	Character or numeric. CHIRTS 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 ($0.05^\circ \times 0.05^\circ$) dataset of daily maximum and minimum temperatures covering 1983–2016.

CHIRTS-daily combines monthly CHIRTS temperature fields with daily temperature variability from the ERA5 reanalysis to produce a spatially detailed and temporally consistent temperature record suitable for monitoring climate variability and temperature extremes.

Daily maximum temperatures are generated by combining monthly CHIRTS temperature estimates with daily ERA5 anomalies. Daily minimum temperatures are subsequently derived using the ERA5 diurnal temperature range.

Additional information is available at: <https://www.chc.ucsb.edu/data/chirtsdaily>

Value

A [SpatRaster](#) with one layer per day.

Note

CHIRTS-daily provides historical daily temperature estimates for 1983–2016. For an extended record from 1980 to near-present, see [get_chirts_era5_raw](#).

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](https://doi.org/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](https://doi.org/10.1175/JCLID180698.1)

Examples

```
#' if (interactive()) {
  r = get_chirts_raw(dates = c("2015-01-01", "2015-01-31"),
                    var = "Tmax")
}
```

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 <code>data.frame</code> (or any other object that can be coerced to <code>data.frame</code>), <code>SpatVector</code> , <code>SpatRaster</code> , <code>sf</code> or <code>geojson</code>
dates	a character of start and end dates in that order in the format "YYYY-MM-DD"
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 <code>period = 1</code> , twelve weeks <code>= 2</code>
...	additional arguments passed to <code>terra</code> or <code>sf</code> methods See details
as.sf	logical, returns an object of class <code>sf</code>
as.geojson	logical, returns an object of class <code>geojson</code>

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

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), SpatVector , SpatRaster , sf or geojson
dates	a character of start and end dates in that order in the format "YYYY-MM-DD"
operation	optional, an integer that represents which type of statistical operation to perform on the dataset
...	additional arguments passed to terra or sf methods 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 (<i>default value</i>)

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

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	<i>Compute precipitation indices over a time series</i>
----------------	---

Description

Compute precipitation indices over a time series

Usage

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

Arguments

object	an object of class <code>chirps</code> as provided by get_chirps
timeseries	logical, FALSE for a single point time series observation or TRUE for a time series based on <i>intervals</i>
intervals	integer no lower than 5, for the days intervals when <i>timeseries</i> = 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)
R99p	total precipitation when rain > 99th percentile (mm)
Rtotal	total precipitation (mm) in wet days, rain >= 1 (mm)
SDII	simple daily intensity index, total precipitation divided by the number of wet days (mm/days)

References

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

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, server = "ClimateSERV")  
  
# 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)
```

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, [17](#)

- * **utility functions**

- as.geojson, [2](#)

as.geojson, [2](#)

chirps, [3](#)

chirps-package (chirps), [3](#)

data.frame, [5](#), [9](#), [13](#), [15](#)

get_chirps, [4](#), [16](#)

get_chirps_raw, [6](#)

get_chirts, [8](#)

get_chirts_era5_raw, [10](#), [12](#)

get_chirts_raw, [11](#)

get_esi, [13](#)

get_imerg, [14](#)

precip_indices, [16](#)

sf, [2](#), [5](#), [6](#), [13–15](#)

SpatRaster, [5](#), [7](#), [9](#), [10](#), [12](#), [13](#), [15](#)

SpatVector, [5](#), [9](#), [13](#), [15](#)

tapajos, [17](#)

terra, [5](#), [9](#), [13](#), [15](#)