Package ‘chirps’

July 13, 2020

Type Package
Title API Client for CHIRPS
Version 0.1.2
URL https://docs.ropensci.org/chirps/
BugReports https://github.com/ropensci/chirps/issues
Description API Client for the Climate Hazards Group InfraRed Precipitation with Station Data 'CHIRPS'. The 'CHIRPS' data is a 35+ year quasi-global rainfall data set, which incorporates 0.05 arc-degrees resolution satellite imagery, and in-situ station data to create gridded rainfall time series for trend analysis and seasonal drought monitoring. For more details on 'CHIRPS' data please visit its official home page <https://www.chc.ucsb.edu/data/chirps>. Requests from large time series (> 10 years) and large geographic coverage (global scale) may take several minutes.
License MIT + file LICENSE
Encoding UTF-8
LazyData true
Depends R (>= 3.5.0), methods
Imports crul, jsonlite, sf, stats
Suggests knitr, markdown, rmarkdown, testthat (>= 2.1.0), vcr (>= 0.5)
Language en-GB
RoxygenNote 7.1.1
VignetteBuilder knitr
NeedsCompilation no
Author Kauê de Sousa [aut, cre] (<https://orcid.org/0000-0002-7571-7845>), Adam H. Sparks [aut] (<https://orcid.org/0000-0002-0061-8359>), William Ashmall [ctb] (API Client implementation), Jacob van Etten [ths] (<https://orcid.org/0000-0001-7554-2558>), Svein Ø. Solberg [ths] (<https://orcid.org/0000-0002-4491-4483>)
Maintainer Kauê de Sousa <kaue.desousa@inn.no>
Repository CRAN
Date/Publication 2020-07-13 16:30:03 UTC


\textbf{R topics documented:}

\begin{verbatim}
as.geojson .................................................. 2
chirps .................................................. 3
get_chirps .................................................. 4
get_esi .................................................. 6
get_imerg .................................................. 8
precip_indices ............................................. 9
tapajos .................................................. 11
\end{verbatim}

\section*{Description}

Take single points from geographical coordinates and coerce it into a geojson 'Polygon' string

\section*{Usage}

\begin{verbatim}
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, ...)
\end{verbatim}

\section*{Arguments}

\begin{verbatim}
lonlat a data.frame or matrix with geographical coordinates lonlat, in that order, or an
object of class 'sf' and 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
\end{verbatim}

\section*{Value}

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

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

set.seed(123)
lonlat <- data.frame(lon = runif(2, 10, 12),
                     lat = runif(2, 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  

---

**chirps**  

*API Client for CHIRPS*

---

**Description**

API Client for the Climate Hazards Group InfraRed Precipitation with Station Data 'CHIRPS'. The 'CHIRPS' data is a 35+ year quasi-global rainfall data set, which incorporates 0.05 arc-degrees resolution satellite imagery, and in-situ station data to create gridded rainfall time series for trend analysis and seasonal drought monitoring. For more details on 'CHIRPS' data please visit its official home page <https://www.chc.ucsb.edu/data/chirps>. Requests from large time series (> 10 years) and large geographic coverage (global scale) may take several minutes.

**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 the CHIRPS data.

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

**Author(s)**

Kauê de Sousa and Adam H. Sparks
get_chirps

See Also

Useful links:

• JOSS paper: 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
• CHIRPS website: https://www.chc.ucsb.edu/data/chirps
• ClimateSERV website: https://climateserv.servirglobal.net/

get_chirps  Get CHIRPS precipitation data

Description

Get daily precipitation data from the "Climate Hazards Group InfraRed Precipitation with Station Data" via ClimateSERV API client. 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_chirps(object, dates, operation = 5, ...)  
## Default S3 method:
get_chirps(object, dates, operation = 5, ...)

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

## S3 method for class 'geojson'
get_chirps(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), geojson or sf
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
...  further arguments passed to 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:
**get_chirps**

<table>
<thead>
<tr>
<th>operation</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>0</td>
</tr>
<tr>
<td>min</td>
<td>1</td>
</tr>
<tr>
<td>median</td>
<td>2</td>
</tr>
<tr>
<td>sum</td>
<td>4</td>
</tr>
<tr>
<td>average</td>
<td>5 ( (default) value)</td>
</tr>
</tbody>
</table>

dist: numeric, buffer distance for each object coordinate

nQuadSegs: integer, number of segments per buffer quadrant

**Value**

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**

https://doi.org/10.1038/sdata.2015.66

ClimateSERV https://climateserv.servirglobal.net

**Examples**

```r
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)

dt
```
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), geojson or sf

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 period = 1, twelve weeks = 2

...    further arguments passed to 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:

<table>
<thead>
<tr>
<th>operation</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>0</td>
</tr>
<tr>
<td>min</td>
<td>1</td>
</tr>
<tr>
<td>median</td>
<td>2</td>
</tr>
<tr>
<td>sum</td>
<td>4</td>
</tr>
<tr>
<td>average</td>
<td>5 (default value)</td>
</tr>
</tbody>
</table>
get_esi

**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 look *sf* documentation for possible issues.

**References**

ClimateSERV [https://climateserv.servirglobal.net](https://climateserv.servirglobal.net)

**Examples**

```r
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 set 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 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.

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), geojson or sf  

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  

...: further arguments passed to 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:

<table>
<thead>
<tr>
<th>operation</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>max</td>
<td>0</td>
</tr>
<tr>
<td>min</td>
<td>1</td>
</tr>
<tr>
<td>median</td>
<td>2</td>
</tr>
<tr>
<td>sum</td>
<td>4</td>
</tr>
<tr>
<td>average</td>
<td>5 (default value)</td>
</tr>
</tbody>
</table>
precip_indices

**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

**References**

ClimateSERV [https://climateserv.servirglobal.net](https://climateserv.servirglobal.net)

NASA IMERG [https://disasters.nasa.gov/instruments/imerg](https://disasters.nasa.gov/instruments/imerg)

**Examples**

```r
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 precipitation indices over a time series.*

**Description**

Compute precipitation indices over a time series.

**Usage**

```r
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 dataframe 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

https://doi.org/10.1029/2005JD006119


Examples

```r
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)
```
Description

Geometries for the Tapajos National Forest, a protected area in the Brazilian Amazon http://www.icmbio.gov.br/flonatapajos/

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 via https://www.protectedplanet.net/.
Index

* datasets
  tapajos, 11
* utility functions
  as.geojson, 2

as.geojson, 2
chirps, 3
chirps-package (chirps), 3
data.frame, 4, 6, 8
get_chirps, 4, 9
get_esi, 6
get_imerg, 8
precip_indices, 9
sf, 2, 4–8	

tapajos, 11