Package 'GHCNr'

January 7, 2025

Title Download Weather Station Data from GHCNd

Version 1.4.5

Description The goal of 'GHCNr' is to provide a fast and friendly interface with the Global Historical Climatology Network daily (GHCNd) database, which contains daily summaries of weather station data worldwide (<https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily>). GHCNd is accessed through the web API <https://www.ncei.noaa.gov/access/services/data/v1>. 'GHCNr' main functionalities consist of downloading data from GHCNd, filter it, and to aggregate it at monthly and annual scales.

License MIT + file LICENSE

Imports tibble, dplyr, tidyr, readr, tidyselect, httr2, terra, utils, rlang, curl

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

RoxygenNote 7.3.2

Depends R (>= 2.10)

LazyData true

VignetteBuilder knitr

NeedsCompilation no

Author Emilio Berti [aut, cre] (<https://orcid.org/0000-0001-9286-011X>)

Maintainer Emilio Berti <emilio.berti@idiv.de>

Repository CRAN

Date/Publication 2025-01-07 13:00:06 UTC

Contents

.add_variables *Add Columns to Handle Summarize*

Description

Add Columns to Handle Summarize

Usage

```
.add_variables(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Value

Table with number of days in the months.

.api_error *Handles API Errors*

Description

Handles API Errors

Usage

```
.api_error(resp)
```

Arguments

resp	Object of class httr2_response.

Value

NULL, called for side effects.

.check_flags *Check Flags Columns*

Description

Check Flags Columns

Usage

```
.check_flags(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Value

NULL, called for side effects

.daily_request *Request Daily Summaries*

Description

Request Daily Summaries

Usage

```
.daily_request(url)
```

Arguments

url	Character, URL of the request.

Value

Body of the JSON request.

.daily_url *Create Request URL for Daily Summaries*

Description

Create Request URL for Daily Summaries

Usage

```
.daily_url(station_id, start_date, end_date, variables)
```

Arguments

station_id	Character, station id(s).
start_date	Character, start date.
end_date	Character, end date.
variables	Character, vector of the variables to include.

Details

station_id can be a vector with multiple stations. Dates should be given in YYYY-mm-dd format.

Value

Character string with the API URL.

.drop_flags *Drop Flags Columns*

Description

Drop Flags Columns

Usage

```
.drop_flags(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Value

The original objects without flags column.

.elevation_url *The GHCNd Station URL with Elevation*

Description

The GHCNd Station URL with Elevation

Usage

```
.elevation_url()
```

Value

The URL of the GHCNd stations.

.extract_flag *Extract GHCNd Flags*

Description

Extract GHCNd Flags

Usage

```
.extract_flag(x)
```

Arguments

x	Character, vector of the flag as returned by the GHCNd API call.

Details

https://www.ncei.noaa.gov/products/land-based-station/global-historical-climatology-network-daily

Value

Character of the flag.

.flags *GHCNd Flags*

Description

GHCNd Flags

Usage

```
.flags(strict)
```

Arguments

strict	Logical, if to include all flags (TRUE) or not (FALSE).

Details

doi:10.1175/2010JAMC2375.1

Value

Table with flags.

.inventory_url *The GHCNd Inventory URL*

Description

The GHCNd Inventory URL

Usage

```
.inventory_url()
```

Value

The URL of the GHCNd inventory.

.max *Calculate Maximum*

Description

Calculate Maximum

Usage

```
.max(x)
```

Arguments

x	Numeric vector

Value

Numeric.

.mean *Calculate Mean*

Description

Calculate Mean

Usage

```
.mean(x)
```

Arguments

x	Numeric vector

Value

Numeric.

.min *Calculate Minimum*

Description

Calculate Minimum

Usage

```
.min(x)
```

Arguments

x	Numeric vector

Value

Numeric.

.missing_variables *Check Which Variables Are Absent*

Description

Check Which Variables Are Absent

Usage

```
.missing_variables(x)
```

Arguments

x	Object of class `ghcn_daily`.

Value

Character vector

.s3_annual *Annual Class Constructor*

Description

Annual Class Constructor

Usage

```
.s3_annual(data = tibble::tibble())
```

Arguments

data	A data frame or tibble to be used as the underlying data.

Details

Creates a new object of class ghcn_annual.

Value

An object of class ghcn_annual.

.s3_anomaly *Anomaly Constructor*

Description

Anomaly Constructor

Usage

```
.s3_anomaly(data = tibble::tibble())
```

Arguments

data	A data frame or tibble to be used as the underlying data.

Details

Creates a new object of class ghcn_anomaly.

Value

An object of class ghcn_anomaly.

.s3_daily *Daily Class Constructor*

Description

Daily Class Constructor

Usage

```
.s3_daily(data = tibble::tibble())
```

Arguments

data	A data frame or tibble to be used as the underlying data.

Details

Creates a new object of class ghcn_daily.

Value

An object of class ghcn_daily.

.s3_monthly *Monthly Class Constructor*

Description

Monthly Class Constructor

Usage

```
.s3_monthly(data = tibble::tibble())
```

Arguments

data	A data frame or tibble to be used as the underlying data.

Details

Creates a new object of class ghcn_monthly.

Value

An object of class ghcn_monthly.

.s3_quarterly *Annual Quarter Constructor*

Description

Annual Quarter Constructor

Usage

```
.s3_quarterly(data = tibble::tibble())
```

Arguments

data	A data frame or tibble to be used as the underlying data.

Details

Creates a new object of class ghcn_quarterly.

Value

An object of class ghcn_quarterly.

.sum *Calculate Sum*

Description

Calculate Sum

Usage

```
.sum(x)
```

Arguments

x	Numeric vector

Value

Numeric.

annual *Calculate Annual Timseries*

Description

annual() aggregates the daily timeseries into an annual one. Aggregation is done differently for TMIN, TMAX, and PRCP.

Usage

```
annual(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Details

Aggregation is done as:

TMAX Maximum temperature recorded in the year

TMIN Minimum temperature recorded in the year

PRCP Total (cumulative) precipitation amount in the year

Value

A tibble with the annual timeseries at the stations.

Examples

```
annual(CA003076680)
```

annual_coverage *Calculate Annual Coverage*

Description

annual_coverage() calculates how many days have been recorded for each year in the time period.

Usage

```
annual_coverage(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Details

To calculate the coverage, a full daily time range is full joined to the timeseries. Missing days are set to NA. Coverage is then calculated as the number of values that are not NAs over the number of NAs.

Value

A table with annual coverage.

Examples

```
cleaned <- remove_flagged(CA003076680)
cover <- annual_coverage(cleaned)
cover
```

---

anomaly *Temperature Anomaly*

Description

`anomaly()` calculates the temperature anomalies compared to a baseline reference period. Anomalies are the difference between annual temperature extremes and the average across the baseline period.

If `aggregate_stations = TRUE`, anomalies are averaged across all weather stations.

Usage

```
anomaly(x, cutoff, aggregate_stations = FALSE)
```

Arguments

`x`	Object of class `ghcn_daily` or `ghcn_annual`. See `daily()` and `annual()` for details.
`cutoff`	Numeric, last year of the baseline period (inclusive).
`aggregate_stations`	Logical, if anomaly should be calculated aggregating data from all weather stations.

Details

`cutoff` must be a character with the date, e.g. "2000-01-01".

Value

A tibble with the anomaly timeseries at the stations.

Examples

```
x <- USC00010655
x <- remove_flagged(x)
cover <- annual_coverage(x)
years <- cover$year[cover$"annual_coverage_tmax" > .99 & cover$"annual_coverage_tmin" > .99]
years <- setdiff(years, 2024)
x$years <- as.numeric(format(x$date, "%Y"))
x <- x[x$years %in% years, ]
a <- annual(x)
anom <- anomaly(a, cutoff = 2012)
plot(anom)
```

`as_daily` *Cast Table to Daily*

Description

Cast Table to Daily

Usage

```
as_daily(data)
```

Arguments

`data`	A data frame or tibble to be used as the underlying data.

Value

An object of class `ghcn_daily`.

Examples

```
## Not run:
df <- read.csv(...)
df <- as_daily(df)

## End(Not run)
```

CA003076680 *Daily data for Station CA003076680*

Description

Daily data for Station CA003076680

Usage

```
CA003076680
```

Format

`CA003076680`**:**
A 'ghcn-daily' object, i.e. table 7,574 x 8:

date Date of measurment
station Station name, i.e. 'CA003076680'
tmax Maximum temperature
tmin Minimum temperature
prcp Total precipitation
***_flag** Flags for the measurments

Source

https://www.countrycallingcodes.com/iso-country-codes/europe-codes.php

country_codes *Countries ISO Codes*

Description

Countries ISO Codes

Usage

```
country_codes
```

Format

`europe_codes`**:**
A table 253 x 2:

name Country name
iso3 3 letter ISO country code

coverage *Calculate Coverage of Daily Summaries*

Description

coverage() calculates the temporal coverage of the time series. See also monthly_coverage(), annual_coverage(), and period_coverage().

Usage

```
coverage(x, graph = FALSE)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.
graph	Logical, if to show a graph of annual coverage.

Details

Returns a table with:

- mothly_coverage The proportion of the days with records in the month
- annual_coverage The proportion of the days with records in the year
- annual_coverage The proportion of the years with records in the reference period

Value

A table with coverage.

Examples

```
cleaned <- remove_flagged(CA003076680)
cover <- coverage(cleaned)
cover[cover$month == 1, ]
```

daily *Download Daily Summaries*

Description

Download Daily Summaries

Usage

```
daily(station_id, start_date, end_date, variables = c("tmax", "tmin", "prcp"))
```

Arguments

station_id	Character, station id(s).
start_date	Character, start date.
end_date	Character, end date.
variables	Character, vector of the variables to include.

Details

station_id can be a vector with multiple stations. Dates should be given in YYYY-mm-dd format. Available *variables* can be found at https://www.ncei.noaa.gov/pub/data/ghcn/daily/readme.txt.

Value

A tibble with the daily timeseries at the stations.

Examples

```
## Not run:
CA003076680 <- daily("CA003076680", "1990-01-01", "2024-12-31")

## End(Not run)
```

---

download_inventory *Download GHCNd Inventory File*

Description

Download GHCNd Inventory File

Usage

```
download_inventory(filename)
```

Arguments

filename	Character of the filename of the inventory, if already downloaded.

Details

Download the inventory from <"https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-inventory.txt">.

Value

Character, the location of the file where the inventory has been saved.

Examples

```
## Not run:
download_inventory(...)

## End(Not run)
```

`elevation_stations` *Get GHCNd Station Elevation*

Description

Get GHCNd Station Elevation

Usage

```
elevation_stations()
```

Value

The table with the elevation of GHCNd stations.

Examples

```
## Not run:
el <- elevation_stations()

## End(Not run)
```

`filter_stations` *Spatial Filtering of Stations*

Description

Spatial Filtering of Stations

Usage

```
filter_stations(stations, roi)
```

Arguments

`stations`	the table with station data. See `stations()`.
`roi`	the geometry of the region of interest. See `get_country()`.

Value

Table with filtered stations.

Examples

```
## Not run:
inventory <- stations()
roi <- get_country("ITA")
s <- filter_stations(inventory, roi)

## End(Not run)
```

`get_countries` *Download multiple countries' shapefiles from geoBoundaries*

Description

Download multiple countries' shapefiles from geoBoundaries

Usage

```
get_countries(countries_code, simplified = TRUE)
```

Arguments

`countries_code`	Vector of three letter ISO code.
`simplified`	Logical.

Details

https://github.com/wmgeolab/geoBoundaries.

Value

A shapefile.

Examples

```
## Not run:
eu <- get_countries(country_code$iso3, simplified = TRUE)

## End(Not run)
```

get_country *Download country shapefile from geoBoundaries*

Description

Download country shapefile from geoBoundaries

Usage

```
get_country(country_code, simplified = TRUE)
```

Arguments

country_code	Three letter ISO code.
simplified	Logical.

Details

https://github.com/wmgeolab/geoBoundaries.

Value

A shapefile.

Examples

```
## Not run:
italy <- get_country("ITA")

## End(Not run)
```

monthly *Calculate Monthly Summaries*

Description

Calculate Monthly Summaries

Usage

```
monthly(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Details

x is the table returned from `daily()` or `remove_flagged()` or any subset of them.

Value

A tibble with the monthly timeseries at the stations.

Examples

```
monthly(CA003076680)
```

`monthly_coverage` *Calculate Monthly Coverage*

Description

`monthly_coverage()` calculates how many days have been recorded for each month in the time period.

Usage

```
monthly_coverage(x)
```

Arguments

`x`	Object of class `ghcn_daily`. See `daily()` for details.

Details

To calculate the coverage, a full daily time range is full joined to the timeseries. Missing days are set to NA. Coverage is then calculated as the number of values that are not NAs over the number of NAs.

Value

A table with mothly coverage.

Examples

```
cleaned <- remove_flagged(CA003076680)
cover <- monthly_coverage(cleaned)
cover[cover$year == 2020, ]
```

period_coverage *Calculate Period Coverage*

Description

`period_coverage()` calculates how many days have been recorded for the whole time period.

Usage

```
period_coverage(x)
```

Arguments

`x`	Object of class `ghcn_daily`. See `daily()` for details.

Details

To calculate the coverage, a full daily time range is full joined to the timeseries. Missing days are set to NA. Coverage is then calculated as the number of values that are not NAs over the number of NAs. Period coverage is a constant value for each station in the `ghcn_daily` object.

Value

A table with period coverage.

Examples

```
cleaned <- remove_flagged(CA003076680)
cover <- period_coverage(cleaned)
cover
```

plot.ghcn_annual *Plot GHCNd Timeseries*

Description

Plot GHCNd Timeseries

Usage

```
## S3 method for class 'ghcn_annual'
plot(x, variable, ...)
```

Arguments

x	Object of class ghcn_annual. See annual() for details.
variable	Name of the variable to plot.
...	additional arguments to be passed to stats::interaction.plot().

Value

NULL, called for side effects.

Examples

```
plot(annual(CA003076680), "tmax")
```

---

plot.ghcn_anomaly *Plot GHCN Timeseries*

Description

Plot GHCN Timeseries

Usage

```
## S3 method for class 'ghcn_anomaly'
plot(x, ...)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.
...	additional arguments to be passed to stats::interaction.plot().

Value

NULL, called for side effects.

Examples

```
plot(anomaly(remove_flagged(CA003076680), 2015))
```

plot.ghcn_daily *Plot GHCNd Timeseries*

Description

Plot GHCNd Timeseries

Usage

```
## S3 method for class 'ghcn_daily'
plot(x, variable, ...)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.
variable	Name of the variable to plot.
...	additional arguments to be passed to stats::interaction.plot().

Value

NULL, called for side effects.

Examples

```
plot(CA003076680, "tmax")
```

plot.ghcn_monthly *Plot GHCNd Timeseries*

Description

Plot GHCNd Timeseries

Usage

```
## S3 method for class 'ghcn_monthly'
plot(x, variable, ...)
```

Arguments

x	Object of class ghcn_monthly. See monthly() for details.
variable	Name of the variable to plot.
...	additional arguments to be passed to stats::interaction.plot().

Value

NULL, called for side effects.

Examples

```
plot(monthly(CA003076680), "tmax")
```

`plot.ghcn_quarterly` *Plot GHCNd Timeseries*

Description

Plot GHCNd Timeseries

Usage

```
## S3 method for class 'ghcn_quarterly'
plot(x, variable, ...)
```

Arguments

`x`	Object of class `ghcn_quarterly`. See `daily()` for details.
`variable`	Name of the variable to plot.
`...`	additional arguments to be passed to `stats::interaction.plot()`.

Value

NULL, called for side effects.

Examples

```
plot(monthly(CA003076680), "tmax")
```

`quarterly` *Calculate Quarterly Timseries*

Description

`quarterly()` aggregates the daily timeseries into a quarterly one. Aggregation is done differently for TMIN, TMAX, and PRCP.

Usage

```
quarterly(x)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.

Details

Quarters are defined as:

first January to March
second April to June
third July to September
fourth October to December

Aggregation is done as:

TMAX Maximum temperature recorded in the quarter
TMIN Minimum temperature recorded in the quarter
PRCP Total (cumulative) precipitation amount in the quarter

Value

A tibble with the quarterly timeseries at the stations.

Examples

```
quarterly(CA003076680)
```

---

remove_flagged *Remove Flagged Recrods*

Description

Remove Flagged Recrods

Usage

```
remove_flagged(x, strict = TRUE)
```

Arguments

x	Object of class ghcn_daily. See daily() for details.
strict	Logical, if to remove also looser flags.

Value

x without flagged records.

Examples

```
remove_flagged(CA003076680)
```

stations *Get GHCNd Inventory*

Description

Get GHCNd Inventory

Usage

```
stations(
  filename,
  variables = c("tmin", "tmax", "prcp"),
  first_year,
  last_year
)
```

Arguments

filename	Character, the filename of the inventory, if already downloaded.
variables	Character, vector of the variables to include.
first_year	Integer, the year since when data should be recorded.
last_year	Integer, the year until when data should be recorded.

Details

If *filename* is not provided, this will download the inventory from <"https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-inventory.txt">. In alternative, you can download the invetory yourself and load it (see examples).

Value

The table with the GHCNd stations.

Examples

```
## Not run:
dest <- tempfile()
download_inventory(dest)
s <- stations(dest)

## End(Not run)
```

USC00010655 *Daily data for Station USC00010655*

Description

Daily data for Station USC00010655

Usage

```
USC00010655
```

Format

USC00010655**:**
A 'ghcn-daily' object, i.e. table 7,809 x 8:

date Date of measurment
station Station name, i.e. 'USC00010655'
tmax Maximum temperature
tmin Minimum temperature
prcp Total precipitation
***_flag** Flags for the measurments

Source

https://www.countrycallingcodes.com/iso-country-codes/europe-codes.php

Index