

# Package ‘GHCNr’

December 20, 2024

**Title** Download Weather Station Data from GHCN

**Version** 1.0.0

**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, tidymodels, 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](mailto:emilio.berti@idiv.de)>

**Repository** CRAN

**Date/Publication** 2024-12-20 12:30:06 UTC

## Contents

.add_variables . . . . .	2
.api_error . . . . .	3
.check_flags . . . . .	3

.daily_request . . . . .	4
.daily_url . . . . .	4
.drop_flags . . . . .	5
.extract_flag . . . . .	5
.flags . . . . .	6
.inventory_url . . . . .	6
.max . . . . .	7
.mean . . . . .	7
.min . . . . .	8
.missing_variables . . . . .	8
.s3_annual . . . . .	9
.s3_daily . . . . .	9
.s3_monthly . . . . .	10
.sum . . . . .	10
annual . . . . .	11
annual_coverage . . . . .	11
as_daily . . . . .	12
CA003076680 . . . . .	13
country_codes . . . . .	13
coverage . . . . .	14
daily . . . . .	14
download_inventory . . . . .	15
filter_stations . . . . .	16
get_countries . . . . .	16
get_country . . . . .	17
monthly . . . . .	18
monthly_coverage . . . . .	18
period_coverage . . . . .	19
plot.ghcn_annual . . . . .	20
plot.ghcn_daily . . . . .	20
plot.ghcn_monthly . . . . .	21
remove_flagged . . . . .	21
stations . . . . .	22
stations_elevation . . . . .	23
USC00010655 . . . . .	23

**Index** **25**

---

<code>.add_variables</code>	<i>Add Columns to Handle Summarize</i>
-----------------------------	--

---

**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

---

<code>.daily_request</code>	<i>Request Daily Summaries</i>
-----------------------------	--------------------------------

---

**Description**

Request Daily Summaries

**Usage**

```
.daily_request(url)
```

**Arguments**

<code>url</code>	Character, URL of the request.
------------------	--------------------------------

**Value**

Body of the JSON request.

---

<code>.daily_url</code>	<i>Create Request URL for Daily Summaries</i>
-------------------------	---

---

**Description**

Create Request URL for Daily Summaries

**Usage**

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

**Arguments**

<code>station_id</code>	Character, station id(s).
<code>start_date</code>	Character, start date.
<code>end_date</code>	Character, end date.
<code>variables</code>	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.

---

`.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.

---

<code>.flags</code>	<i>GHCNd Flags</i>
---------------------	--------------------

---

**Description**

GHCNd Flags

**Usage**

`.flags(strict)`

**Arguments**

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

**Details**

[doi:10.1175/2010JAMC2375.1](https://doi.org/10.1175/2010JAMC2375.1)

**Value**

Table with flags.

---

<code>.inventory_url</code>	<i>The GHCNd Inventory URL</i>
-----------------------------	--------------------------------

---

**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 ghc\_n\_annual.

**Value**

An object of class ghc\_n\_annual.

---

.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 ghc\_n\_daily.

**Value**

An object of class ghc\_n\_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`.

---

`.sum`*Calculate Sum*

---

**Description**

Calculate Sum

**Usage**

```
.sum(x)
```

**Arguments**

`x` Numeric vector

**Value**

Numeric.

---

annual	<i>Calculate Annual Timeseries</i>
--------	------------------------------------

---

**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	<i>Calculate Annual Coverage</i>
-----------------	----------------------------------

---

**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
```

---

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 ghc\_n\_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 measurement

**station** Station name, i.e. 'CA003076680'

**tmax** Maximum temperature

**tmin** Minimum temperature

**prcp** Total precipitation

**\*\_flag** Flags for the measurements

**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	<i>Calculate Coverage of Daily Summaries</i>
----------	--

---

### 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 ghcndaily. See <a href="#">daily()</a> for details.
graph	Logical, if to show a graph of annual coverage.

### Details

Returns a table with:

- monthly\_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	<i>Download Daily Summaries</i>
-------	---------------------------------

---

### 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	<i>Download GHCNd Inventory File</i>
--------------------	--------------------------------------

---

**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)
```

---

filter_stations	<i>Spatial Filtering of Stations</i>
-----------------	--------------------------------------

---

**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	<i>Download multiple countries' shapefiles from geoBoundaries</i>
---------------	---

---

**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	<i>Calculate Monthly Summaries</i>
---------	------------------------------------

---

**Description**

Calculate Monthly Summaries

**Usage**

```
monthly(x)
```

**Arguments**

x                    Object of class ghcndaily. 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	<i>Calculate Monthly Coverage</i>
------------------	-----------------------------------

---

**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 ghcndaily. 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 monthly coverage.

**Examples**

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

---

period_coverage	<i>Calculate Period Coverage</i>
-----------------	----------------------------------

---

**Description**

period\_coverage() calculates how many days have been recorded for the whole time period.

**Usage**

```
period_coverage(x)
```

**Arguments**

x                    Object of class ghcndaily. 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 ghcndaily object.

**Value**

A table with period coverage.

**Examples**

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

---

plot.ghcn\_annual      *Plot GHCN Timeseries*

---

**Description**

Plot GHCN Timeseries

**Usage**

```
## S3 method for class 'ghcn_annual'
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(annual(CA003076680), "tmax")
```

---

plot.ghcn\_daily      *Plot GHCN Timeseries*

---

**Description**

Plot GHCN 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 GHCN Timeseries*

---

**Description**

Plot GHCN Timeseries

**Usage**

```
## S3 method for class 'ghcn_monthly'
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(monthly(CA003076680), "tmax")
```

---

*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)
```

---

<code>stations</code>	<i>Get GHCNd Inventory</i>
-----------------------	----------------------------

---

**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 inventory 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)
```

---

stations\_elevation      *Elevation of the GHCNd Stations*

---

**Description**

Elevation of the GHCNd Stations

**Usage**

```
stations_elevation
```

**Format**

europa\_codes:  
A table 128,024 x 2:  
**station** Station name  
**elevation** Elevation in meters

**Source**

<https://www.ncei.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.csv>

---

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 measurement

**station** Station name, i.e. 'USC00010655'

**tmax** Maximum temperature

**tmin** Minimum temperature

**prcp** Total precipitation

**\*\_flag** Flags for the measurements

**Source**

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

# Index

## \* datasets

- CA003076680, 13
- country\_codes, 13
- stations\_elevation, 23
- USC00010655, 23
- .add\_variables, 2
- .api\_error, 3
- .check\_flags, 3
- .daily\_request, 4
- .daily\_url, 4
- .drop\_flags, 5
- .extract\_flag, 5
- .flags, 6
- .inventory\_url, 6
- .max, 7
- .mean, 7
- .min, 8
- .missing\_variables, 8
- .s3\_annual, 9
- .s3\_daily, 9
- .s3\_monthly, 10
- .sum, 10

- annual, 11
- annual\_coverage, 11
- annual\_coverage(), 14
- as\_daily, 12

- CA003076680, 13
- country\_codes, 13
- coverage, 14

- daily, 14
- daily(), 3, 5, 11, 14, 18–22
- download\_inventory, 15

- filter\_stations, 16

- get\_countries, 16
- get\_country, 17
- get\_country(), 16

- monthly, 18
- monthly\_coverage, 18
- monthly\_coverage(), 14
- period\_coverage, 19
- period\_coverage(), 14
- plot.ghcn\_annual, 20
- plot.ghcn\_daily, 20
- plot.ghcn\_monthly, 21
- remove\_flagged, 21
- stations, 22
- stations(), 16
- stations\_elevation, 23
- stats::interaction.plot(), 20, 21
- USC00010655, 23