Package: chcd (via r-universe)

November 3, 2026

Title Access Canadian Historical Climate Data

Version 0.1.1

Description Provides easy access to historical climate data in Canada from R. Search for weather stations and download raw hourly, daily or monthly weather data across Canada from 1840 to present. Implements public API access as detailed at <https://climate.weather.gc.ca>.

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.2

Imports magrittr, readr, dplyr, tibble, stringr, rlang, progress

Depends R (>= 3.5)

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Dan Prisk [aut, cre, cph] (ORCID: <https://orcid.org/0000-0002-3377-6978>)

Maintainer Dan Prisk <dan@prisk.ca>

Config/pak/sysreqs libicu-dev libx11-dev

Repository https://danprisk.r-universe.dev

Date/Publication 2025-07-25 10:10:09 UTC

RemoteUrl https://github.com/cran/chcd

RemoteRef HEAD

RemoteSha be8c8200685424b9f3270edf9a3c75949eca8327

Contents

build_url 2
dl_csv 3
get_climatedata 3

get_station 4
get_timecode 5
stations_near 5
test_climatefile 6

Index **7**

`build_url` *Build a valid URL for downloading a file from CHCD. Follows the format detailed here* https://collaboration.cmc.ec.gc.ca/cmc/climate/Get_More_Data_Plus_de_donnees/.

Description

Build a valid URL for downloading a file from CHCD. Follows the format detailed here https://collaboration.cmc.ec.gc.ca/cmc/climate/Get_More_Data_Plus_de_donnees/.

Usage

```
build_url(station_id, timecode, year, month = NA)
```

Arguments

`station_id`	The ID of the station to be downloaded
`timecode`	The numeric timecode to be downloaded
`year`	The year of data to be loaded
`month`	The month of the year to download data. Setting to NA gets all months.

Value

A URL string

Examples

```
build_url(337,3,1980)

# The timecode can also be a string such as "m" or "month"
build_url(337,"m",1990)

# Setting the month is most useful when downloading hourly data
build_url(337,"h",1980,3)
```

`dl_csv` *Downloads a csv from the internet*

Description

Downloads a csv from the internet

Usage

```
dl_csv(url)
```

Arguments

`url`	A string containing the URL of the CSV to download

Value

The content of the downloaded CSV

`get_climatedata` *Get historical climate data for the specified time and place. The data returned will be raw data as originally reported. As many weather stations historically use manual entry to report this data this is likely to mean that data cleaning will be required for sensible usage.*

Description

Get historical climate data for the specified time and place. The data returned will be raw data as originally reported. As many weather stations historically use manual entry to report this data this is likely to mean that data cleaning will be required for sensible usage.

Usage

```
get_climatedata(place, years, interval, progress = TRUE)
```

Arguments

`place`	The place that data should be downloaded for. This can be a climate station ID, or a text place name. Can take a list of place names.
`years`	The year, or years, to get data for. Can be either a single numeric year or a list of years. All years must be between 1840 and present.
`interval`	The interval that data should be returned for. Must be one of: "h", "hourly", "d", "daily", "m", "monthly". Defaults to monthly.
`progress`	Defines if a progress bar is shown. Can be TRUE or FALSE. Defaults to TRUE.

Value

A single tibble containing all the requested data

Examples

```
# These will return climate data for all stations that include the place string in their name.
get_climatedata("squamish",1980,"monthly")
get_climatedata("toronto",1980,"daily")

# In order to get a specific station you can use its station id (see chcd::get_station())
get_climatedata(337, 1980, "m")

# Can also support lists of places and years
get_climatedata("squamish", c(1980:1985), "m")
get_climatedata(c("squamish","whistler"), 1990, "m")
get_climatedata(c(337,338,339), c(1974:1975), "daily")
```

---

get_station *Find climate station or stations from a given place. This also confirms if a given climate station ID is valid.*

Description

Find climate station or stations from a given place. This also confirms if a given climate station ID is valid.

Usage

```
get_station(place)
```

Arguments

place	Either a numeric station ID or a place name. Case insensitive.

Value

Returns a tibble containing id, name, and location for all valid stations corresponding to place. Or NA if none are found.

Examples

```
get_station("squamish")
get_station("SqUaMiSh")
get_station(337)
```

get_timecode *Converts a text based interval into a CHCD timecode*

Description

Converts a text based interval into a CHCD timecode

Usage

```
get_timecode(interval)
```

Arguments

interval	A text based interval. Must be one of: "h", "hourly", "d", "daily", or "m", "monthly".

Value

A numeric code that can be used in a CHCD URL

stations_near *Finds climate stations near to a given location.*

Description

Finds climate stations near to a given location.

Usage

```
stations_near(longitude, latitude, distance)
```

Arguments

longitude	The longitude of the point
latitude	The latitude of the point
distance	The distance in KM from the point to pull in stations

Value

A tibble containing the id, name, and locations for all valid stations within the radius of distance from point.

`test_climatefile` *Given a file and a list of expected columns this will test to make sure our data looks good. Returns TRUE if all expected columns are present in the file, otherwise returns FALSE.*

Description

Given a file and a list of expected columns this will test to make sure our data looks good. Returns TRUE if all expected columns are present in the file, otherwise returns FALSE.

Usage

```
test_climatefile(file)
```

Arguments

`file`	A file downloaded from CHCD for testing

Value

TRUE if file contains expected columns, FALSE otherwise.

Index

build_url, 2

dl_csv, 3

get_climatedata, 3
get_station, 4
get_timecode, 5

stations_near, 5

test_climatefile, 6