Package: weathercan (via r-universe)

October 30, 2026

Type Package

Title Download Weather Data from Environment and Climate Change Canada

Version 1.0.1

Description Provides means for downloading historical weather data from the Environment and Climate Change Canada website (<https://climate.weather.gc.ca/historical_data/search_historic_data_e.html>). Data can be downloaded from multiple stations and over large date ranges and automatically processed into a single dataset. Tools are also provided to identify stations either by name or proximity to a location.

License GPL-3

Language en-CA

BugReports https://github.com/ropensci/weathercan/issues/

LazyData TRUE

URL https://docs.ropensci.org/weathercan/, https://github.com/ropensci/weathercan/

Depends R (>= 4.1.0)

Imports dplyr (>= 1.0.0), lubridate (>= 1.7.1), memoise (>= 2.0.0), purrr (>= 1.0.0), rlang (>= 0.1.4), readr (>= 2.0.0), stringi (>= 1.1.2), stringr (>= 1.4.0), tidyr (>= 1.1.3), rappdirs (>= 0.3.3), httr2 (>= 1.2.2), cli (>= 3.6.6)

RoxygenNote 7.3.3

Roxygen list(markdown = TRUE)

Suggests devtools, ggplot2, htmltools, knitr, lutz, mockery, rmarkdown, sf, testthat (>= 3.0.0), withr

VignetteBuilder knitr

Encoding UTF-8

Config/testthat/edition 3

Config/pak/sysreqs libicu-dev libssl-dev libx11-dev

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

Date/Publication 2026-08-31 15:31:44 UTC

RemoteUrl https://github.com/ropensci/weathercan

RemoteRef HEAD

RemoteSha 7869a45c4ca551c1160bed16d312e4361bd0e2d4

Contents

cache_dir . . . 2
cache_remove . . . 3
check_eccc . . . 3
codes . . . 4
finches . . . 4
flags . . . 5
glossary . . . 5
glossary_normals . . . 6
kamloops . . . 6
kamloops_day . . . 8
normals_dl . . . 9
normals_file . . . 12
normals_measurement_types . . . 12
normals_measurements . . . 13
pg . . . 13
stations . . . 15
stations_dl . . . 16
stations_meta . . . 17
stations_search . . . 17
variables_normals_new . . . 19
variables_normals_old . . . 19
weather_dl . . . 20
weather_interp . . . 22

Index **24**

`cache_dir` *Cache Directory*

Description

Location of weathercan cache (or where the cache will be if created).

Usage

```
cache_dir()
```

Value

Directory path

Examples

```
cache_dir()
```

`cache_remove` *Remove cached directory and contents*

Description

This is a helper function to remove weathercan's cache directory. This folder contains updated stations inventory lists and the full dataset of Climate normals for 1991-2010.

Usage

```
cache_remove()
```

Value

nothing but removes directory

Examples

```
cache_remove()
```

`check_eccc` *Check access to ECCC*

Description

Checks if whether there is internet access, weather data, normals data, and eccc sites are available and accessible, and whether we're NOT running on cran

Usage

```
check_eccc()
```

Value

FALSE if not, TRUE if so

Examples

```
check_eccc()
```

codes *Meaning of climate normal 'codes'*

Description

A reference dataset containing codes matched to their meaning. Data downloaded using the normals_dl() function contains columns indicating code. These are presented here for interpretation.

Usage

```
codes
```

Format

A data frame with 4 rows and 2 variables:

code Code

meaning Explanation of the code

finches *RFID Data on finch visits to feeders*

Description

RFID Data on finch visits to feeders

Usage

```
finches
```

Format

An example dataset of finch RFID data for interpolation:

bird_id Bird ID number

time Time

feeder_id feeder ID

species Species

lat Latitude of station location in degree decimal format

lon Longitude of station location in degree decimal format

flags *Meaning of coded 'flags'*

Description

A reference dataset containing 'flags' matched to their meaning. Data downloaded using the `weather_dl()` function contains columns indicating 'flags' these codes are presented here for interpretation.

Usage

```
flags
```

Format

A data frame with 16 rows and 2 variables:

code Flag code

meaning Explanation of the code

glossary *Glossary of units and terms*

Description

A reference dataset matching information on columns in data downloaded using the `weather_dl()` function. Indicates the units of the data, and contains a link to the ECCC glossary page explaining the measurement.

Usage

```
glossary
```

Format

A data frame with 77 rows and 5 variables:

interval Data interval type, 'hour', 'day', or 'month'.

ECCC Original column name when downloaded directly from ECCC

weathercan R-compatible name given when downloaded with the `weather_dl()` function using the default argument `format = TRUE`.

units Units of the measurement.

ECCC_ref Link to the glossary or reference page on the ECCC website.

glossary_normals *Glossary of terms for Climate Normals*

Description

A reference dataset matching information on general columns in older climate normals (pre 1991-2020) data downloaded using the `normals_dl()` function. Indicates the names and descriptions of different data measurements.

Usage

```
glossary_normals
```

Format

A data frame with 18 rows and 3 variables:

ECCC Original measurement type from ECCC

weathercan R-compatible name given when downloaded with the `normals_dl()` function

description Description of the measurement type from ECCC

kamloops *Hourly weather data for Kamloops*

Description

Downloaded with `weather()`. Terms are more thoroughly defined here https://climate.weather.gc.ca/glossary_e.html

Usage

```
kamloops
```

Format

An example dataset of hourly weather data for Kamloops:

station_name Station name

station_id Environment Canada's station ID number. Required for downloading station data.

prov Province

lat Latitude of station location in degree decimal format

lon Longitude of station location in degree decimal format

date Date

time Time

year Year

month Month

day Day

hour Hour

qual Data quality

weather The state of the atmosphere at a specific time.

hmdx Humidex

hmdx_flag Humidex data flag

pressure Pressure (kPa)

pressure_flag Pressure data flag

rel_hum Relative humidity

rel_hum_flag Relative humidity data flag

temp Temperature

temp_dew Dew Point Temperature

temp_dew_flag Dew Point Temperature flag

visib Visibility (km)

visib_flag Visibility data flag

wind_chill Wind Chill

wind_chill_flag Wind Chill flag

wind_dir Wind Direction (10's of degrees)

wind_dir_flag wind Direction Flag

wind_spd Wind speed km/hr

wind_spd_flag Wind speed flag

elev Elevation (m)

climate_id Climate identifier

WMO_id World Meteorological Organization Identifier

TC_id Transport Canada Identifier

Source

https://climate.weather.gc.ca/index_e.html

`kamloops_day` *Daily weather data for Kamloops*

Description

Downloaded with `weather()`. Terms are more thoroughly defined here https://climate.weather.gc.ca/glossary_e.html

Usage

```
kamloops_day
```

Format

An example dataset of daily weather data for Kamloops:

station_name Station name

station_id Environment Canada's station ID number. Required for downloading station data.

prov Province

lat Latitude of station location in degree decimal format

lon Longitude of station location in degree decimal format

date Date

year Year

month Month

day Day

cool_deg_days Cool degree days

cool_deg_days_flag Cool degree days flag

dir_max_gust Direction of max wind gust

dir_max_gust_flag Direction of max wind gust flag

heat_deg_days Heat degree days

heat_deg_days_flag Heat degree days flag

max_temp Maximum temperature

max_temp_flag Maximum temperature flag

mean_temp Mean temperature

mean_temp_flag Mean temperature flag

min_temp Minimum temperature

min_temp_flag Minimum temperature flag

snow_grnd Snow on the ground (cm)

snow_grnd_flag Snow on the ground flag

spd_max_gust Speed of the max gust km/h

spd_max_gust_flag Speed of the max gust flag

total_precip Total precipitation (any form)

total_precip_flag Total precipitation flag

total_rain Total rain (any form)

total_rain_flag Total rain flag

total_snow Total snow (any form)

total_snow_flag Total snow flag

elev Elevation (m)

climate_id Climate identifier

WMO_id World Meteorological Organization Identifier

TC_id Transport Canada Identifier

Source

https://climate.weather.gc.ca/index_e.html

`normals_dl` *Download climate normals from Environment and Climate Change Canada*

Description

Downloads climate normals from Environment and Climate Change Canada (ECCC) for one or more stations (defined by `climate_ids`). For details and units, see the `glossary_normals`, `variables_normals_old`, and `variables_normals_new` included data sets and/or the `glossary_normals` vignette: `vignette("glossary_normals", package = "weathercan")`.

Usage

```
normals_dl(
  climate_ids,
  normals_years = "current",
  format = TRUE,
  measurement_type = NULL
)
```

Arguments

`climate_ids`	Character. A vector containing the Climate ID(s) of the station(s) you wish to download data from. See the `stations()` data frame or the `stations_search()` function to find Climate IDs.
`normals_years`	Character. The year range for which you want climate normals. Default `current` (i.e. 1991-2020). One of `current`, `1991-2020`, `1981-2010`, or `1971-2000`. `current` returns only stations from the most recent *complete* normals year range (i.e. `1991-2020`).

`format` Logical. If TRUE (default) formats measurements to numeric and date accordingly. Unlike `weather_dl()`, `normals_dl()` will always format column headings as normals data from ECCC cannot be directly made into a data frame without doing so.

`measurement_type` Character vector. Measurement types (called element groups in original ECCC data) to include in normals data (only relevant for new normals >= `1991-2020`). Will return only the measurements included in the these groups. If `NULL` (default) returns all normals measurements. See `normals_measurement_types` of a list of types and which measurements are included.

Details

The format and method of downloading climate normals from ECCC varies by year span.

Regardless of year, each normals measurement column has a corresponding `_code` column which reflects the data quality of that measurement (see the 1991-2020, 1981-2010, or 1971-2000 for more details) ECCC calculation documents.

Newer normals (1991-2020):

Newer normals from ECCC are provided in one bulk downloaded which weathercan will fetch and store in a local cache directory (`cache_dir()`). Then `normals_dl()` will read, filter, format, and return the climate normals in a data frame easier to work with in R than the original data.

These normals are also provided in a single table, so both 'normals' and 'frost' data are combined in one.

Newer climate normals are downloaded from the url stored in option `weathercan.urls.normals_1991_2020`. To change this location use: `options(weathercan.urls.normals_1991_2020 = "your_new_url")`.

Older normals (1981-2010 and earlier):

Older normals from ECCC are provided by individual file downloads which weathercan will fetch, format and return as requested (no local on-disk cache storage).

These older normals also include two separate data types: averages by month for a variety of measurements as well as data relating to the frost-free period. Because these two data sources are quite different, we return them as nested data so the user can extract them as they wish. See examples for how to use the `unnest()` function from the `tidyr` package to extract the two different datasets.

The data also returns a column called `meets_wmo` this reflects whether or not the climate normals for this station met the WMO standards for temperature and precipitation (i.e. both have code >= A).

Older climate normals are downloaded from the url stored in option `weathercan.urls.normals`. To change this location use: `options(weathercan.urls.normals = "your_new_url")`.

@inheritSection weather_dl Verbosity

Value

For new climate normals, a tibble of normals. For older climate normals, a tibble with *nested* normals and first/last frost data.

Examples

```
# Find the climate_id
stations_search("Brandon A", normals_years = "current")

# Download climate normals 1991-2020 ("current" normals)
n <- normals_dl(climate_ids = "5010480")
n

# Download multiple climate Ids - But only one location!
# - 1990-2010 normals use composite stations
stations_search("Winnipeg", normals_years = "current")
n <- normals_dl(climate_ids = c("502S001", "5023227", "5023222"))
unique(dplyr::select(n, "location_name", "composite_stations"))

# Download multiple climate Ids
n <- normals_dl(climate_ids = c("5010480", "5023222"))
unique(dplyr::select(n, "location_name", "composite_stations"))

# Download climate normals 1981-2010
# - Note: Very different data format from current normals!
n <- normals_dl(climate_ids = "5010480", normals_years = "1981-2010")

# Pull out last frost data *with* station information
library(tidyr)
f <- unnest(n, frost)
f

# Pull out normals *with* station information
nm <- unnest(n, normals)
nm

# Download climate normals 1971-2000
n <- normals_dl(climate_ids = "5010480", normals_years = "1971-2000")
n

# Note that some do not have last frost dates
n$frost

# Download multiple stations for 1981-2010,
n <- normals_dl(
  climate_ids = c("301C3D4", "301FFNJ", "301N49A"),
  normals_years = "1981-2010"
)
unnest(n, frost)

# Note, putting both normals and frost data into the same data set can be
# done, but makes for a very unwieldy dataset (there is lots of repetition).
nm <- unnest(n, normals) |>
  unnest(frost)
```

normals_file *Location of the cached normals data*

Description

Returns the *expected* file path of the location of cached normals data. Only the most current normals data are provided as a full data download by ECCC, so only these normals are cached locally. Note that if you haven't downloaded the normals files yet (call `normals_dl(...)`) these files will not exist.

Usage

```
normals_file(normals_years = "1991-2020", type = "normals")
```

Arguments

`normals_years`	Character. Years to load. Currently only `1991-2020` is available.
`type`	Character. Data type to load, one of "normals", or "meta" (the composite station inventory).

Value

Character file path.

Examples

```
normals_file()
```

normals_measurement_types *List of climate normals measurements and types for each set of normals*

Description

A data frame listing the climate normals measurements classified by measurement_type available for each set of climate normals. This is very similar to `normals_measurements` and just omits the stations.

Usage

```
normals_measurement_types
```

Format

A data frame with 158 rows and 3 variables:

normals Year range of climate normals

measurement_type Type of measurement (relevant only for 1990-2020)

measurement Climate normals measurement available for this station

---

normals_measurements *List of climate normals measurements for each station*

Description

A data frame listing the climate normals measurements available for each station.

Usage

```
normals_measurements
```

Format

A data frame with 113,325 rows and 5 variables:

prov Province

station_name Station Name

climate_id Climate ID

normals Year range of climate normals

measurement_type Type of measurement (relevant only for 1990-2020)

measurement Climate normals measurement available for this station

---

pg *Hourly weather data for Prince George*

Description

Downloaded with `weather()`. Terms are more thoroughly defined here https://climate.weather.gc.ca/glossary_e.html

Usage

```
pg
```

Format

An example dataset of hourly weather data for Prince George:

station_name Station name
station_id Environment Canada's station ID number. Required for downloading station data.
prov Province
lat Latitude of station location in degree decimal format
lon Longitude of station location in degree decimal format
date Date
time Time
year Year
month Month
day Day
hour Hour
qual Data quality
weather The state of the atmosphere at a specific time.
hmdx Humidex
hmdx_flag Humidex data flag
pressure Pressure (kPa)
pressure_flag Pressure data flag
rel_hum Relative humidity
rel_hum_flag Relative humidity data flag
temp Temperature
temp_dew Dew Point Temperature
temp_dew_flag Dew Point Temperatureflag
visib Visibility (km)
visib_flag Visibility data flag
wind_chill Wind Chill
wind_chill_flag Wind Chill flag
wind_dir Wind Direction (10's of degrees)
wind_dir_flag wind Direction Flag
wind_spd Wind speed km/hr
wind_spd_flag Wind speed flag
elev Elevation (m)
climate_id Climate identifier
WMO_id World Meteorological Organization Identifier
TC_id Transport Canada Identifier

Source

https://climate.weather.gc.ca/index_e.html

`stations` *Access Station data downloaded from Environment and Climate Change Canada*

Description

This function access the built-in stations data frame. You can update this data frame with `stations_dl()` which will update the locally stored data.

Usage

```
stations()
```

Format

A data frame:

prov Province

station_name Station name

station_id Environment Canada's station ID number. Required for downloading station data.

climate_id Climate ID number

WMO_id Climate ID number

TC_id Climate ID number

lat Latitude of station location in degree decimal format

lon Longitude of station location in degree decimal format

elev Elevation of station location in metres

tz Local timezone excluding any Daylight Savings

interval Interval of the data measurements ('hour', 'day', 'month')

start Starting year of data record

end Ending year of data record

normals Whether *any* climate normals are available for that station (new behaivour)

normals_1991_2020 Whether 1991-2020 climate normals are available for that station. **Note** that even if available, these are not yet downloadable via weathercan.

normals_1981_2010 Whether 1981-2010 climate normals are available for that station

normals_1971_2000 Whether 1971-2000 climate normals are available for that station

Details

You can check when this was last updated with `stations_meta()`.

A dataset containing station information downloaded from Environment and Climate Change Canada. Note that a station may have several station IDs, depending on how the data collection has changed over the years. Station information can be updated by running `stations_dl()`.

Source

https://climate.weather.gc.ca/index_e.html

Examples

```
stations()
stations_meta()

# Which Manitoba stations have *any* climate normals?
# Note `normals` is TRUE or FALSE, so we can included it as is for normals == TRUE

library(dplyr)
filter(stations(), interval == "hour", normals, prov == "MB")
```

---

stations_dl *Get available stations*

Description

This function can be used to download a Station Inventory CSV file from Environment and Climate Change Canada. This is only necessary if the station you're interested was only recently added. The 'stations' data set included in this package contains station data downloaded when the package was last compiled. This function may take a few minutes to run.

Usage

```
stations_dl(skip = NULL)
```

Arguments

skip	Numeric. Number of lines to skip at the beginning of the csv. If NULL, automatically derived.

Details

The stations list is downloaded from the url stored in the option `weathercan.urls.stations`. To change this location use `options(weathercan.urls.stations = "your_new_url")`.

The list of which stations have climate normals is downloaded from the url stored in the option `weathercan.urls.stations.normals`. To change this location use `options(weathercan.urls.normals = "your_new_url")`.

Currently there are two sets of climate normals available: 1981-2010 and 1971-2000. Whether a station has climate normals for a given year range is specified in `normals_1981_2010` and `normals_1971_2000`, respectively.

The column `normals` represents the most current year range of climate normals (i.e. currently 1981-2010)

@inheritSection weather_dl Verbosity

Examples

```
# Update stations data frame
stations_dl()

# Updated stations data frame is now automatically used
stations_search("Winnipeg")
```

`stations_meta` *Show stations list meta data*

Description

Date of ECCC update and date downloaded via weathercan.

Usage

```
stations_meta()
```

Examples

```
stations_meta()
```

`stations_search` *Search for stations by name or location*

Description

Returns stations that match the name provided OR which are within `dist` km of the location provided. This is designed to provide the user with information with which to decide which station to then get weather data from.

Usage

```
stations_search(
  name = NULL,
  coords = NULL,
  dist = 10,
  interval = c("hour", "day", "month"),
  normals_years = NULL,
  normals_only = NULL,
  starts_latest = NULL,
  ends_earliest = NULL
)
```

Arguments

name	Character. A vector of length 1 or more with text against which to match. Will match station names that contain all components of name, but they can be in different orders and separated by other text.
coords	Numeric. A vector of length 2 with latitude and longitude of a place to match against. Overrides lat and lon if also provided.
dist	Numeric. Match all stations within this many kilometres of the coords.
interval	Character. Return only stations with data at these intervals. Must be any of "hour", "day", "month".
normals_years	Character. One of NULL (default), current, 1991-2020, 1981-2010, or 1971-2000. current returns only stations from the most recent *complete* normals year range (i.e. 1981-2010). Default NULL does not filter by climate normals. Specific year ranges return stations with normals in that period. See Details for more specifics.
normals_only	DEPRECATED. Logical. Return only stations with climate normals?
starts_latest	Numeric. Restrict results to stations with data collection beginning in or before the specified year.
ends_earliest	Numeric. Restrict results to stations with data collection ending in or after the specified year.

Details

To search by coordinates, users must make sure they have the sp package installed.

The current, most recent, climate normals year range is 1981-2010.

Value

Returns a subset of the stations data frame which match the search parameters. If the search was by location, an extra column 'distance' shows the distance in kilometres from the location to the station. If no stations are found withing dist, the closest 10 stations are returned.

Examples

```
stations_search(name = "Kamloops")
stations_search(name = "Kamloops", interval = "hour")

stations_search(name = "Ottawa", starts_latest = 1950, ends_earliest = 2010)

stations_search(name = "Ottawa", normals_years = "current")   # 1981-2010
stations_search(name = "Ottawa", normals_years = "1981-2010") # Same as above
stations_search(name = "Ottawa", normals_years = "1971-2000") # 1971-2010

if(requireNamespace("sf")) {
  stations_search(coords = c(53.915495, -122.739379))
}
```

variables_normals_new *Index of variables for new Climate Normals*

Description

An index matching variables named in weathercan and downloaded with the `normals_dl()` function to those in the original *new* (1991-2020) climate normals data from ECCC.

Usage

```
variables_normals_new
```

Format

A data frame with 18 rows and 3 variables:

measurement_type Measurement category

ECCC Original variable name from ECCC

weathercan R-compatible name given when formatting the data with the `normals_dl()` function

variables_normals_old *Index of variables for new Climate Normals*

Description

An index matching variables named in weathercan and downloaded with the `normals_dl()` function to those in the original *old* (1981-2010 and 1971-2000) climate normals data from ECCC.

Usage

```
variables_normals_old
```

Format

A data frame with 18 rows and 3 variables:

ECCC Original variable name from ECCC

weathercan R-compatible name given when formatting the data with the `normals_dl()` function

`weather_dl` *Download weather data from Environment and Climate Change Canada*

Description

Downloads data from Environment and Climate Change Canada (ECCC) for one or more stations. For details and units, see the glossary vignette (`vignette("glossary", package = "weathercan")`) or the glossary online https://climate.weather.gc.ca/glossary_e.html.

Usage

```
weather_dl(
  station_ids,
  start = NULL,
  end = NULL,
  interval = "hour",
  months = NULL,
  trim = TRUE,
  trim_by_stn = FALSE,
  format = TRUE,
  string_as = NA,
  time_disp = "none",
  encoding = "UTF-8",
  list_col = FALSE
)
```

Arguments

`station_ids`	Numeric/Character. A vector containing the ID(s) of the station(s) you wish to download data from. See the `stations` data frame or the `stations_search` function to find IDs.
`start`	Date/Character. The start date of the data in YYYY-MM-DD format (applies to all stations_ids). Defaults to start of range.
`end`	Date/Character. The end date of the data in YYYY-MM-DD format (applies to all station_ids). Defaults to end of range.
`interval`	Character. Interval of the data, one of "hour", "day", "month".
`months`	Numeric vector. Can supply 1-12 to optionally filter the data to only specific months. For "hour" interval, this selectively downloads data by month so can speed up downloads. For intervals of "day" and "month" this only filters the data after full years or full data ranges have been downloaded.
`trim`	Logical. Trim missing values from the start and end of the weather dataframe. Only applies if `format = TRUE`
`trim_by_stn`	Logical. Data from different stations are generally padded with NAs to have the same date range. If this isn't desirable, use `trim = TRUE` and `trim_by_stn =`

	TRUE to trim NAs from the start and end of each station. trim_by_stn = FALSE (default), only the sides of the entire range are trimmed.
format	Logical. If TRUE, formats data for immediate use. If FALSE, returns data exactly as downloaded from Environment and Climate Change Canada. Useful for dealing with changes by Environment Canada to the format of data downloads.
string_as	Character. What value to replace character strings in a numeric measurement with. See Details.
time_disp	Character. Either "none" (default) or "UTC". See details.
encoding	Character. Text encoding for download.
list_col	Logical. Return data as nested data set? Defaults to FALSE. Only applies if format = TRUE

Details

Data can be returned 'raw' (format = FALSE) or can be formatted. Formatting transforms dates/times to date/time class, renames columns, and converts data to numeric where possible. If character strings are contained in traditionally numeric fields (e.g., weather speed may have values such as "< 30"), they can be replaced with a character specified by string_as. The default is NA. Formatting also replaces data associated with certain flags with NA (M = Missing), if they are not already marked as NA.

Start and end date can be specified, but if not, it will default to the start and end date of the range (this could result in downloading a lot of data!).

For hourly data, timezones are always marked "UTC", but the actual times are either local time (default; time_disp = "none"), or UTC (time_disp = "UTC"). When time_disp = "none", times reflect the local time without daylight savings. This means that relative measures of time, such as "nighttime", "daytime", "dawn", and "dusk" are comparable among stations in different timezones. This is useful for comparing daily cycles. When time_disp = "UTC" the times are transformed into UTC timezone. Thus midnight in Kamloops would register as 08:00:00 (Pacific time is 8 hours behind UTC). This is useful for tracking weather events through time, but will result in odd 'daily' measures of weather (e.g., data collected in the afternoon on Sept 1 in Kamloops will be recorded as being collected on Sept 2 in UTC).

Files are downloaded from the url stored in getOption("weathercan.urls.weather"). To change this location use options(weathercan.urls.weather = "your_new_url").

Data is downloaded from ECCC as a series of files which are then bound together. Each file corresponds to a different month, or year, depending on the interval. Metadata (station name, lat, lon, elevation, etc.) is extracted from the start of the most recent file (i.e. most recent dates) for a given station. Note that important data (i.e. station name, lat, lon) is unlikely to change between files (i.e. dates), but some data may or may not be available depending on the date of the file (e.g., station operator was added as of April 1st 2018, so will be in all data which includes dates on or after April 2018).

Value

A tibble with station ID, name and weather data.

Verbosity

Verbosity (how 'chatty' weathercan is) can be specified using the option `weathercan.verbosity`. Which takes "standard" (default), "quiet" (suppress all messages including those regarding missing data, etc.), or "verbose" (extra progress messages).

Examples

```
kam <- weather_dl(station_ids = 51423,
                  start = "2016-01-01", end = "2016-02-15")

stations_search("Kamloops A$", interval = "hour")
stations_search("Prince George Airport", interval = "hour")

kam.pg <- weather_dl(station_ids = c(48248, 51423),
                     start = "2016-01-01", end = "2016-02-15")

library(ggplot2)

ggplot(data = kam.pg, aes(x = time, y = temp,
                          group = station_name,
                          colour = station_name)) +
       geom_line()

# Download only January and December
kam <- weather_dl(
  station_ids = 51423,
  start = "2016-01-01",
  end = "2018-02-15",
  months = c(1, 10)
)
```

`weather_interp` *Interpolate and add weather data to a dataframe*

Description

When data and the weather measurements do not perfectly line up, perform a linear interpolation between two weather measurements and merge the results into the provided dataset. Only applies to numerical weather columns (see `weather` for more details).

Usage

```
weather_interp(data, weather, cols = "all", interval = "hour", na_gap = 2)
```

Arguments

data	Dataframe. Data with dates or times to which weather data should be added.
weather	Dataframe. Weather data downloaded with weather which should be interpolated and added to data.
cols	Character. Vector containing the weather columns to add or 'all' for all relevant columns. Note that some measure are omitted because they cannot be linearly interpolated (e.g., wind direction).
interval	What interval is the weather data recorded at? "hour" or "day".
na_gap	How many hours or days (depending on the interval) is it acceptable to skip over when interpolating over NAs (see details).

Details

Dealing with NA values If there are NAs in the weather data, na_gap can be used to specify a tolerance. For example, a tolerance of 2 with an interval of "hour", means that a two hour gap in data can be interpolated over (i.e. if you have data for 9AM and 11AM, but not 10AM, the data between 9AM and 11AM will be interpolated. If, however, you have 9AM and 12PM, but not 10AM or 11AM, no interpolation will happen and data between 9AM and 12PM will be returned as NA.)

@inheritSection weather_dl Verbosity

Examples

```
# Weather data only
kamloops

# Data about finch observations at RFID feeders in Kamloops, BC
finches

# Match weather to finches

# First line up the timezones
# - Finches are in Pacific Time (inc. Daylight savings),
#   Kamloops is in Pacific Time *without* daylight savings, but is marked as UTC for
#   simplicity (see ?weather_dl for details)
# - First we convert finches to remove daylight savings, then we mark as UTC
finches <- dplyr::mutate(finches, time = lubridate::with_tz(time, "Etc/GMT+8"))
finches <- dplyr::mutate(finches, time = lubridate::force_tz(time, "UTC"))

# Then interpolate over the first 30 observations
finch_weather <- weather_interp(data = finches[1:30,], weather = kamloops)
```

Index

∗ **datasets**
  codes, 4
  finches, 4
  flags, 5
  glossary, 5
  glossary_normals, 6
  kamloops, 6
  kamloops_day, 8
  normals_measurement_types, 12
  normals_measurements, 13
  pg, 13
  variables_normals_new, 19
  variables_normals_old, 19

add_weather *(*weather_interp*)*, 22

cache_dir, 2
cache_dir(), *10*
cache_remove, 3
check_eccc, 3
codes, 4

finches, 4
flags, 5

glossary, 5
glossary_normals, 6, *9*

kamloops, 6
kamloops_day, 8

normals_dl, 9
normals_file, 12
normals_measurement_types, 12
normals_measurements, 13

pg, 13

stations, 15, *20*
stations(), *9*
stations_dl, 16
stations_meta, 17
stations_search, 17, *20*
stations_search(), *9*

variables_normals_new, *9*, 19
variables_normals_old, *9*, 19

weather, *6*, *8*, *13*, *23*
weather *(*weather_dl*)*, 20
weather_dl, 20
weather_interp, 22