Package 'worldmet'

July 6, 2020

Type Package

Title Import Surface Meteorological Data from NOAA Integrated Surface
Database (ISD)

Version 0.9.0

Date 2020-07-04

ByteCompile true

Depends R (>= 3.2.0)

Imports openair, doParallel, parallel, foreach, dplyr, leaflet, tidyr,
readr, zoo

Maintainer David Carslaw <david.carslaw@york.ac.uk>

Description Functions to import data from more than 30,000 surface
meteorological sites around the world managed by the National Oceanic and Atmospheric Ad-
ministration (NOAA) Integrated Surface
Database (ISD, see <https://www.ncdc.noaa.gov/isd>).

License GPL (>= 2)

URL <http://github.com/davidcarslaw/worldmet>

BugReports <http://github.com/davidcarslaw/worldmet/issues>

LazyLoad true

LazyData true

Encoding UTF-8

RoxygenNote 7.1.1

NeedsCompilation no

Author David Carslaw [aut, cre]

Repository CRAN

Date/Publication 2020-07-06 08:00:02 UTC

1

R **topics documented:**

---

exportADMS   *Export a meteorological data frame in ADMS format*

Description

Export a meteorological data frame in ADMS format

Usage

```
exportADMS(dat, out = "./ADMS_met.MET", interp = FALSE, maxgap = 2)
```

Arguments

dat	A data frame imported by importNOAA.
out	A file name for the ADMS file. The file is written to the working directory by default.
interp	Should interpolation of missing values be undertaken? If TRUE linear interpolation is carried out for gaps of up to and including maxgap.
maxgap	The maximum gap in hours that should be interpolated where there are missing data when interp = TRUE. Data with gaps more than maxgap are left as missing.

Value

Writes a text file to a location of the user's choosing.

Examples

```
## Not run:
## import some data then export it
dat <- importNOAA(year = 2012)
exportADMS(dat, file = "~/temp/adms_met.MET")

## End(Not run)
```

---

getMeta *Find a ISD site code and other meta data*

Description

Get information on meteorological sites

Usage

```
getMeta(
  site = "heathrow",
  lat = NA,
  lon = NA,
  country = NA,
  state = NA,
  n = 10,
  end.year = "current",
  plot = TRUE,
  returnMap = FALSE
)
```

Arguments

site	A site name search string e.g. site = "heathrow". The search strings and be partial and can be upper or lower case e.g. site = "HEATHR".
lat	A latitude in decimal degrees to search. Takes the values -90 to 90.
lon	A longitude in decimal degrees to search. Takes values -180 to 180. Negative numbers are west of the Greenwich meridian.
country	The country code. This is a two letter code. For a full listing see `ftp://ftp.ncdc.noaa.gov/pub/data/noaa/country-list.txt`.
state	The state code. This is a two letter code.
n	The number of nearest sites to search based on latitude and longitude.
end.year	To help filter sites based on how recent the available data are. end.year can be "current", "any" or a numeric year such as 2016, or a range of years e.g. 1990:2016 (which would select any site that had an end date in that range. **By default only sites that have some data for the current year are returned**.
plot	If TRUE will plot sites on an interactive leaflet map.
returnMap	Should the leaflet map be returned instead of the meta data? Default is FALSE.

Details

This function is primarily used to find a site code that can be used to access data using `importNOAA`. Sites searches of approximately 30,000 sites can be carried out based on the site name and based on the nearest locations based on user-supplied latitude and logitude.

See also `getMetaLive` to download the all meta data to allow re-use and direct querying.

Value

A data frame is returned with all available meta data, mostly importantly including a code that can be supplied to `importNOAA`. If latitude and longitude searches are made an approximate distance, `dist` in km is also returned.

Author(s)

David Carslaw

Examples

```
## Not run:
## search for sites with name beijing
getMeta(site = "beijing")

## End(Not run)

## Not run:
## search for near a specified lat/lon - near Beijing airport
## returns 'n' nearest by default
getMeta(lat = 40, lon = 116.9)

## End(Not run)
```

---

getMetaLive   *Obtain site meta data from NOAA server*

Description

Obtain site meta data from NOAA server

Usage

```
getMetaLive(...)
```

Arguments

`...`   Currently unused.

Value

A tibble with meta data.

Examples

```
## Not run:
meta <- getMetaLive()
head(meta)

## End(Not run)
```

---

importNOAA   *Import meteorological data*

Description

Main function for importing meteorological data

Usage

```
importNOAA(
  code = "037720-99999",
  year = 2014,
  hourly = TRUE,
  n.cores = 1,
  quiet = FALSE,
  path = NA
)
```

Arguments

code	The identifying code as a character string. The code is a combination of the USAF and the WBAN unique identifiers. The codes are separated by a "-" e.g. code = "037720-99999".
year	The year to import. This can be a vector of years e.g. year = 2000:2005.
hourly	Should hourly means be calculated? The default is TRUE. If FALSE then the raw data are returned.
n.cores	Number of cores to use for parallel processing. Default is 1 and hence no parallelism.
quiet	If FALSE, print missing sites / years to the screen.
path	If a file path is provided, the data are saved as an rds file at the chosen location e.g. path = "C:/Users/David". Files are saved by year and site.

Details

This is the main function to import data from the NOAA Integrated Surface Database (ISD). The ISD contains detailed surface meteorological data from around the world for over 30,000 locations. For general information of the ISD see https://www.ncdc.noaa.gov/isd and the map here https://gis.ncdc.noaa.gov/map/viewer/#app=cdo&cfg=cdo&theme=hourly&layers=1.

Note the following units for the main variables:

date Date/time in POSIXct format. **Note the time zone is GMT (UTC) and may need to be adjusted to merge with other local data. See details below.**

latitude Latitude in decimal degrees (-90 to 90).

longitude Longitude in decimal degrees (-180 to 180). Negative numbers are west of the Greenwich Meridian.

elevation Elevention of site in metres.

wd Wind direction in degrees. 90 is from the east.

ws Wind speed in m/s.

ceil_hgt The height above ground level (AGL) of the lowest cloud or obscuring phenomena layer aloft with 5/8 or more summation total sky cover, which may be predominantly opaque, or the vertical visibility into a surface-based obstruction.

visibility The visibility in metres.

air_temp Air temperature in degrees Celcius.

dew_point The dew point temperature in degrees Celcius.

atmos_pres The sea level pressure in millibars.

RH The relative humidity (%).

cl_1, ..., cl_3 Cloud cover for different layers in Oktas (1-8).

cl Maximum of cl_1 to cl_3 cloud cover in Oktas (1-8).

cl_1_height, ..., cl_3_height Height of the cloud base for each later in metres.

precip_12 12-hour precipitation in mm.

precip_6 6-hour precipitation in mm.

precip Based on the 12 hourly and 6 hourly totals, `precip` spreads the 6-hourly totals across the previous 6-hours to provide an indication of hourly precipitation.

pwc The description of the present weather description (if available).

The data are returned in GMT (UTC). It may be necessary to adjust the time zone when comining with other data. For example, if air quality data were available for Beijing with time zone set to "Etc/GMT-8" (note the negative offset even though Beijing is ahead of GMT. See the `openair` package and manual for more details), then the time zone of the met data can be changed to be the same. One way of doing this would be `attr(met$date,"tzone") <-"Etc/GMT-8"` for a meteorological data frame called `met`. The two data sets could then be merged based on `date`.

Value

Returns a data frame of surface observations. The data frame is consistent for use with the `openair` package. NOTE! the data are returned in GMT (UTC) time zone format. Users may wish to express the data in other time zones e.g. to merge with air pollution data.

Author(s)

David Carslaw

See Also

[getMeta](#) to obtain the codes based on various site search approaches.

Examples

```
## Not run:
## use Beijing airport code (see getMeta example)
dat <- importNOAA(code = "545110-99999", year = 2010:2011)

## End(Not run)
```

---

weatherCodes	*Codes for weather types*

Description

This data frame consists of the weather description codes used in the ISD. It is not of general use to most users.

Examples

```
## basic structure
head(weatherCodes)
```

---

worldmet	*Access surface meteorological data from the NOAA Integrated Surface Database from around the world*

Description

This package contains functions to import surface meteorolgical data from over 30,000 sites around the world. These data are curated by NOAA as part of the Integrated Surface Database (ISD).

Details

If you access these data using the `worldmet` package please give full acknowledgement to NOAA ISD. Users should also take a note of the usage restrictions.

These data work well with the `openair` package that has been developed to analyse air pollution data.

Author(s)

David Carslaw

References

For general information of the ISD see https://www.ncdc.noaa.gov/isd and the map here https://gis.ncdc.noaa.gov/map/viewer/#app=cdo&cfg=cdo&theme=hourly&layers=1.

See Also

See https://github.com/davidcarslaw/openair for information on the related openair package.

Index