Package ‘weatheR’

January 4, 2015

Title Download, Transform, and plot NOAA ISD weather data

Version 0.1

Author Michael Piccirilli <mrp2181@columbia.edu>

Maintainer Michael Piccirilli <mrp2181@columbia.edu>

Description A package set up to be used to accompany research being conducted
in the Sustainable Engineering Lab at The Earth Institute at Columbia University.

Depends R (>= 3.1.2),
httr,
ggmap,
FNN,
McSpatial,
data.table,
plyr,
gridExtra

License MIT

R topics documented:

allStations *Get full list of NOAA ISD stations*

Description

This function is included in case the list of stations is updated. The full list of stations is included in the package under the name ’station.list’.

Usage

```
allStations()
```

Value

This function will download the full list of NOAA ISD stations to a variable.

Examples

```
## Not run:
station.list <- allStations()

## End(Not run)
```

getFilteredStationsByCity

Get k-nearest station data, Filtered

Description

This function returns 1 station for each city, after applying four filters: 1) Remove stations with little to no data 2) Remove stations that exceed a maximum distance from each city's reference point 3) Remove stations that exceed a threshold of missing data, including NA values 4) Select closest station remaining for each city, as all remaining stations are deemed adequate

Usage

```
getFilteredStationsByCity(city.list, station.list, k = NULL, begin, end,
  distance = 100, hourly_interval = 3, tolerance = 0.05)
```

Arguments

`city.list`	City of list of Cities. The format should be as follows: "City, State", or "City, Country"
`station.list`	Full list of ISD stations included in the package
`k`	The number of stations to return
`begin`	Start year (4 digits)
`end`	End year (4 digits)
`distance`	Maximum distance allowable from each city's reference point
`hourly_interval`	Minimum hourly interval allowable (1=hourly; 3 = every 3 hours; 6 = every 6 hours, etc..)
`tolerance`	This is the percent, in decimals, of missing data you will allow. (.05 = 5% of total data)

Value

Returns a list of four items. 1) Download status. 2) Number of downlaoded, removed, and kept stations 3) Names of final stations 2) A list of dataframes for each station.

Examples

```
## Not run:
data(stations)
cities <- c("Nairobi, Kenya", "Tema, Ghana", "Accra, Ghana", "Abidjan, Ivory Coast")
get.stations <- getFilteredStationsByCity(cities, station.list, begin = 2012, end = 2013)
get.stations$dl_status
get.stations$removed_rows
get.stations$station_names_final

class(get.stations$station_data)
length(get.stations$station_data)

## End(Not run)
```

`getInterpolatedDataByCity` *Get k-nearest station data, Filtered, Interpolated*

Description

This function applies four filters: 1) Remove stations with little to no data 2) Remove stations that exceed a maximum distance from each city's reference point 3) Remove stations that exceed a threshold of missing data, including NA values 4) Select closest station remaining for each city, as all remaining stations are deemed adequate

Usage

```
getInterpolatedDataByCity(city.list, station.list, k = 5, begin, end,
  distance = 100, hourly_interval = 3, tolerance = 0.05)
```

Arguments

`city.list`	City of list of Cities. The format should be as follows: "City, State", or "City, Country"
`station.list`	Full list of ISD stations included in the package
`k`	The number of stations to return
`begin`	Start year (4 digits)
`end`	End year (4 digits)
`distance`	Maximum distance allowable from each city's reference point
`hourly_interval`	Minimum hourly interval allowable (1=hourly; 3 = every 3 hours; 6 = every 6 hours, etc..)
`tolerance`	This is the percent, in decimals, of missing data you will allow. (.05 = 5% of total data)

Details

It then performs two steps to interpolate missing values: 1) Average over all data points in original dataset to find average hourly observations 2) Linearly interpolate hourly data points for missing observations

Value

Returns a single dataframe with hourly observations (including interpolated) of every city.

Examples

```
## Not run:
data(stations)
cities <- c("Nairobi, Kenya", "Tema, Ghana", "Accra, Ghana", "Abidjan, Ivory Coast")
hourly.data <- getInterpolatedDataByCity(cities, station.list, 5, 2010, 2013, 100, 3, .05)
dim(hourly.data)

## End(Not run)
```

getStationByID *Get ISD station data by USAFID for a range of years*

Description

Get ISD station data by USAFID for a range of years

Usage

```
getStationByID(stationID, station.list, begin, end)
```

Arguments

stationID	USAFID station number
station.list	Full list of ISD stations included in the package
begin	Starting *year* to download data (4 digit)
end	Ending *year* to download data (4 digit)

Value

Returns a list of two items. 1) Status of downloading each year. 2) Weather data in a list

getStationsByCity *Get k-nearest station data*

Description

This function utilizes the kNStation() function to find the k-nearest stations, and downloads all the data for those stations in between a given date range (in years).

Usage

```
getStationsByCity(city.list, station.list, k = 5, begin, end)
```

Arguments

`city.list`	City of list of Cities. The format should be as follows: "City, State", or "City, Country"
`station.list`	Full list of ISD stations included in the package
`k`	The number of stations to return
`begin`	Start year (4 digits)
`end`	End year (4 digits)

Value

Returns a list of two items.

1) Status of downloading each year's data for each station 2) A list of dataframes. Each dataframe is all years data of a particular station.

Examples

```
## Not run:
data(stations)
cities <- c("Nairobi, Kenya", "Tema, Ghana", "Accra, Ghana", "Abidjan, Ivory Coast")
get.stations <- getStationsByCity(cities, station.list, begin = 2012, end = 2013)
get.stations$dl_status
class(get.stations$station_data)
length(get.stations$station_data)

## End(Not run)
```

kNStations *Get the k-nearest weather stations to a city*

Description

Using either an individual city or a list of cities, this will find a reference point for each location and return the k-nearest weather stations to that reference point.

Usage

```
kNStations(city.list, station.list, k = 5)
```

Arguments

`city.list`	List of cities. Format should be: "City, State", or "City, Country"
`station.list`	Full list of ISD stations included in the package
`k`	The number of stations to return

Value

Returns the k-nearest weather stations in the full ISD station.list

Examples

```
## Not run:
data(stations)
cities <- c("Nairobi, Kenya", "Tema, Ghana", "Accra, Ghana", "Abidjan, Ivory Coast")
k.n.stations <- kNStations(cities, station.list, 5)

## End(Not run)
```

plotDailyMax *Plot Daily Max Temp for Multiple Cities*

Description

This function requires that the dataframe fed into the funciton have columns named 'TEMP' for temperature, 'city' for the city or identifier, 'YR' for the year, 'M' for month, and 'D' for the day.

Usage

```
plotDailyMax(hourlyDF)
```

Arguments

`hourlyDF`	Must be a dataframe with columns for TEMP, city, YR, M, and D

Value

Plots the daily max temperateu of one or multiple cities

Examples

```
## Not run:
data(stations)
cities <- c("Nairobi, Kenya", "Tema, Ghana", "Accra, Ghana", "Abidjan, Ivory Coast")
hourly.data <- getInterpolatedDataByCity(cities, station.list, 5, 2010, 2013, 100, 3, .05)
plotDailyMax(hourly.data)

## End(Not run)
```

plotStations *Plot k-nearest stations*

Description

This function will plot each city's reference point, along with the k-nearest stations to that point.

Usage

```
plotStations(city.list, station.list, k = 5)
```

Arguments

`city.list`	City of list of Cities. The format should be as follows: "City, State", or "City, Country"
`station.list`	Full list of ISD stations included in the package
`k`	The number of stations to return

Value

This will produce single, or multple plots for each city

Examples

```
## Not run:
data(stations)
cities <- c("Nairobi, Kenya", "Tema, Ghana", "Accra, Ghana", "Abidjan, Ivory Coast")
plotStations(cities, station.list, 5)

## End(Not run)
```

Index