

Package ‘weatherData’

February 4, 2014

Type Package

Title Get Weather Data from the Web

Version 0.3

Date 2014-01-31

Author Ram Narasimhan

Imports plyr

Maintainer Ram Narasimhan <ramnarasimhan@gmail.com>

Description Functions that help in fetching weather data from websites. Given a location and a date range, these functions help fetch weather data (temperature, pressure etc.) for any weather related analysis.

LazyData TRUE

License GPL

Depends R (>= 2.10)

NeedsCompilation no

Repository CRAN

Date/Publication 2014-02-04 10:42:18

R topics documented:

weatherData-package	2
checkDataAvailability	3
checkDataAvailabilityForDateRange	3
getCurrentTemperature	4
getDailyMinMaxTemp	5
getStationCode	6
getWeatherData	7
getWeatherForDate	8

getWeatherForYear	9
IntlWxStations	10
IsStationTypeInvalid	10
London2013	11
Mumbai2013	11
NewYork2013	12
SFO2012	12
SFO2013	13
USAirportWeatherStations	13

Index	14
--------------	-----------

weatherData-package	<i>Get Weather & Temperature data from the Web</i>
---------------------	--

Description

The package has functions that can fetch weather data.

Details

Package: weatherData
 Type: Package
 Version: 0.1
 Date: 2014-02-03
 License: GPL

These functions don't use APIs. They rely on reading URL's instead. Given a valid city and a date (or date range), the functions in weatherData can fetch them as a nice R data frame.

These functions are useful for anyone interested in doing analysis using weather data.

Author(s)

Ram Narasimhan

Maintainer: Ram Narasimhan <ramnarasimhan@gmail.com>

Examples

```
getWeatherForDate("LAX", "2013-01-01")
checkDataAvailabilityForDateRange("BWI", "2013-12-13", "2014-01-01", "airportCode")
getStationCode("Chennai")
```

checkDataAvailability *Check if WeatherUnderground has Data for given station and date*

Description

Use this function to check if data is available for station and date. If the station code or the date is invalid, function will return 0.

Usage

```
checkDataAvailability(station_id, check_date, station_type = "airportCode")
```

Arguments

station_id	is a valid airport code or a valid Weather Station ID
check_date	is a valid string representing a date in the past (string "YYYY-MM-DD")
station_type	is either airportCode or id

Value

1 if the station does have weather records for input date, 0 if no records were found

References

For a list of valid Weather Stations, try this format <http://www.wunderground.com/weatherstation/ListStations.asp?selectedCountry=United+States> and replace with your country of interest

checkDataAvailabilityForDateRange

Quick Check to see if WeatherUnderground has Weather Data for given station for a range of dates

Description

Before we attempt to fetch the data for a big time interval of dates, this function is useful to see if the data even exists.

Usage

```
checkDataAvailabilityForDateRange(station_id, start_date, end_date,  
    station_type = "airportCode")
```

Arguments

station_id	is a valid 3-letter airport code or a valid Weather Station ID
station_type	is either airportCode or id
start_date	is a valid string representing a date in the past (YYYY-MM-DD, all numeric)
end_date	is a valid string representing a date in the past (YYYY-MM-DD, all numeric) and is greater than start_date

Details

This functions checks for just the first and the last date in the interval, not the days in between

Value

1 if the Station did have weather records, 0 if nothing was found

getCurrentTemperature *Get the latest recorded temperature for a location*

Description

A wrapper for getWeatherData(), it returns the last record in the web page. Uses Sys.Date() to get current time.

Usage

```
getCurrentTemperature(station_id)
```

Arguments

station_id	is a valid Weather Station ID (example: "BUF", "ORD", "VABB" for Mumbai). Valid Weather Station "id" values: "KFLMIAMI75" or "IMOSCOWO2" You can look these up at wunderground.com. You can get station_id's for a given location by calling getStationCode()
------------	---

Value

A one row data frame containing:

- Date and Time stamp (for when the latest temperature reading was recorded)
- Temperature for the station in Farenheit (or Celcius)

References

For a list of valid Weather Stations, try this format <http://www.wunderground.com/weatherstation/ListStations.asp?selectedCountry=United+States> and replace with your country of interest

Examples

```
getCurrentTemperature(station = "HNL")
```

getDailyMinMaxTemp	<i>Get the daily minimum (maximum) temperatures for a given weather stations</i>
--------------------	--

Description

Given a StationID and a set of dates, this function returns the Daily Minimum and/or Maximum temperatures recorded, along with timestamps

Usage

```
getDailyMinMaxTemp(station_id, start_date, end_date = NULL,
                    daily_min = TRUE, daily_max = TRUE, station_type = "airportCode",
                    opt_write_to_file = FALSE)
```

Arguments

station_id	is a valid 3- or 4-letter Airport code or a valid Weather Station ID (example: "BUF", "ORD", "VABB" for Mumbai). Valid Weather Station "id" values: "KFLMIAMI75" or "IMOSCOWO2" You can look these up at wunderground.com
start_date	is a valid string representing a date in the past (YYYY-MM-DD, all numeric)
end_date	(optional) If an interval is to be specified, end_date is a a valid string representing a date in the past (YYYY-MM-DD, all numeric) and greater than start_date
daily_min	A boolean indicating if the Minimum Temperatures are desired
daily_max	A boolean indicating if the Maximum Temperatures are desired Both daily_min and daily_max can be TRUE, but at least one of them should be TRUE.
station_type	= "airportCode" or "ID" (Wx call Sign)
opt_write_to_file	If TRUE, the resulting dataframe will be stored in a CSV file. Default is FALSE

Details

This functions fetches all the records for each date specified, but it only retains the min and/or max record, along with the timestamp.

Value

A data frame with each row containing:

- Date and Time stamp (for when that day's minimum temperature was recorded)
- Minimum Temperature for the station in Farenheit (or Celcius)
- Date and Time stamp (for when that day's maximum temperature was recorded)
- Maximum Temperature for the station in Farenheit (or Celcius)

Examples

```
## Not run:
dat <- getDailyMinMaxTemp("KIAH", "2013-08-10", 2013-08-31", daily_max=TRUE)
dat <- getDailyMinMaxTemp("KBIL", "2013-08-10", daily_max=T)
dat <- getDailyMinMaxTemp("EGLL", "2013-08-10", daily_max=T, daily_min=TRUE)

## End(Not run)
```

getStationCode	<i>Gets the Weather Station code for a location (in the US)</i>
----------------	---

Description

This function goes through the USAirportWeatherStations dataset and looks for matches. Usually, the 4 letter airportCode is what you are after.

Usage

```
getStationCode(stationName, region = NULL)
```

Arguments

stationName	String that you want to get the weatherStation code for
region	A qualifier about the station's location. It could be a continent or a country. If in the US, region is a two-letter state abbreviation. Ex. "AK" for Alaska

Value

A one row data frame containing:

- A string of Station Name that matched
- the region. (two-letter state abbreviation if in the US)
- The 4-letter weather station ID. (This is the string you use when calling getWeatherData())

References

For a world-wide list of possible stations, be sure to look at <http://weather.rap.ucar.edu/surface/stations.txt> The ICAO (4-letter code is what needs to be input to getWeatherData())

Examples

```
getStationCode("Denver")
```

getWeatherData	<i>Gets weather data for a single date</i>
----------------	--

Description

Given a valid station and a single date this function will return a dataframe of time-stamped weather data

Usage

```
getWeatherData(station, date, station_type = "airportCode",  
               opt_temperature_only = T, opt_compress_output = FALSE,  
               opt_verbose = FALSE, opt_warnings = TRUE)
```

Arguments

station	is a valid 3-letter airport code or a valid Weather Station ID
date	is a valid string representing a date in the past (YYYY-MM-DD)
station_type	can be airportCode which is the default, or it can be id which is a weather-station ID
opt_temperature_only	Boolean flag to indicate only Temperature data is to be returned (default TRUE)
opt_compress_output	Boolean flag to indicate if a compressed output is preferred. If this option is set to be TRUE, only every other record is returned
opt_verbose	Boolean flag to indicate if verbose output is desired
opt_warnings	Boolean flag to turn off warnings. Default value is TRUE, to keep the warnings on.

Value

A data frame with each row containing:

- Date and Time stamp for the date specified
- Temperature and/or other weather columns

A data frame containing the Date & Time stamp and Weather data columns

getWeatherForDate	<i>Getting data for a range of dates</i>
-------------------	--

Description

This function will return a (fairly large) data frame. If you are going to be using this data for future analysis, you can store the results in a CSV file by setting `opt_write_to_file` to be `TRUE`

Usage

```
getWeatherForDate(station_id, start_date, end_date = NULL,
  daily_min = FALSE, daily_max = FALSE, station_type = "airportCode",
  opt_write_to_file = FALSE)
```

Arguments

<code>station_id</code>	is a valid 3- or 4-letter Airport code or a valid Weather Station ID (example: "BUF", "ORD", "VABB" for Mumbai). Valid Weather Station "id" values: "KFLMIAMI75" or "IMOSCOWO2" You can look these up at wunderground.com
<code>start_date</code>	is a valid string representing a date in the past (YYYY-MM-DD, all numeric)
<code>end_date</code>	(optional) If an interval is to be specified, <code>end_date</code> is a a valid string representing a date in the past (YYYY-MM-DD, all numeric) and greater than <code>start_date</code>
<code>daily_min</code>	A boolean indicating if only the Minimum Temperatures are desired
<code>daily_max</code>	A boolean indicating if only the Maximum Temperatures are desired
<code>station_type</code>	= "airportCode" (3- or 4-letter airport code) or "ID" (Wx call Sign)
<code>opt_write_to_file</code>	If <code>TRUE</code> , the resulting dataframe will be stored in a CSV file. Default is <code>FALSE</code>

Details

For each day in the date range, this function fetches Weather Data. Internally, it makes multiple calls to `getWeatherData`.

Value

A data frame with each row containing:

- Date and Time stamp (for each date specified)
- Temperature and/or other weather columns sought

References

For a list of valid Weather Stations, try this format <http://www.wunderground.com/weatherstation/ListStations.asp?selectedCountry=United+States> and replace with your country of interest

Examples

```
## Not run:
dat <- getWeatherForDate("PHNL", "2013-08-10", 2013-08-31")

## End(Not run)
```

getWeatherForYear	<i>Get weather data for one full year</i>
-------------------	---

Description

Function will return a data frame with all the records for a given station_id and year. If the current year is supplied, it will returns records until the current Sys.Date() ("today")

Usage

```
getWeatherForYear(station_id, year, station_type = "airportCode",
  opt_write_to_file = FALSE)
```

Arguments

station_id	is a valid Weather Station ID (example: "BUF", "ORD", "VABB" for Mumbai). Valid Weather Station "id" values: "KFLMIAMI75" or "IMOSCOWO2" You can look these up at wunderground.com. You can get station_id's for a given location by calling getStationCode()
year	is a valid year in the past (numeric, YYYY format)
station_type	= "airportCode" (3 or 4 letter airport code) or "ID" (Wx call Sign)
opt_write_to_file	If TRUE, the resulting dataframe will be stored in a CSV file. Default is FALSE

Details

Note that this function is a light wrapper for getWeatherForDate with the two end dates being Jan-01 and Dec-31 of the given year.

Value

A data frame with each row containing:

- Date and Time stamp (for each date specified)
- Temperature and/or other weather columns sought

References

For a list of valid Weather Stations, try this format <http://www.wunderground.com/weatherstation/ListStations.asp?selectedCountry=United+States> and replace with your country of interest

Examples

```
## Not run:
dat <- getWeatherForYear("KLGA", 2013)

## End(Not run)
```

IntlWxStations

Data - International Weather Stations

Description

This is a data frame of the 1602 stations in Weather Underground's database. The 4-letter "ICAO" is used by the functions in this package to check and get the weather data. Note that not all the stations have weather data.

Usage

```
data(IntlWxStations)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

This data frame has been created by <http://weather.rap.ucar.edu/surface/stations.txt> maintained by Greg Thompson of NCAR.

IsStationTypeInvalid

Check if the station type is airportCode or id

Description

We are checking if a valid station type was given to the function.

Usage

```
IsStationTypeInvalid(station_type)
```

Arguments

`station_type` can be `airportCode` which is the default, or it can be `id` which is a weather-station ID

See Also

`getStationCode`

London2013*Data - Ambient Temperature for the City of London for all of 2013*

Description

This is a data frame of Ambient temperature data, extracted from Weather Underground. Each row has two entries (columns). The Timestamp (YYYY-MM-DD HH:MM:SS) and the Temperature (in degrees F)

Usage

```
data(London2013)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

<http://www.wunderground.com/history/airport/EGLL/2013/1/1/DailyHistory.html?format=1>

Mumbai2013*Data - Ambient Temperature for the City of Mumbai, India for all of 2013*

Description

This is a data frame of Ambient temperature data, extracted from Weather Underground. Each row has two entries (columns). The Timestamp (YYYY-MM-DD HH:MM:SS) and the Temperature (in degrees F)

Usage

```
data(Mumbai2013)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

<http://www.wunderground.com/history/airport/VABB/2014/1/1/DailyHistory.html?format=1>

NewYork2013*Data - Ambient Temperature for New York City for all of 2013*

Description

This is a data frame of Ambient temperature data, extracted from Weather Underground. Each row has two entries (columns). The Timestamp (YYYY-MM-DD HH:MM:SS) and the Temperature (in degrees F)

Usage

```
data(NewYork2013)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

<http://www.wunderground.com/history/airport/KLGA/2013/1/1/DailyHistory.html?format=1>

SF02012*Data - Ambient Temperature for the City of San Francisco for all of 2012*

Description

This is a data frame of Ambient temperature data, extracted from Weather Underground. Each row has two entries (columns). The Timestamp (YYYY-MM-DD HH:MM:SS) and the Temperature (in degrees F)

Usage

```
data(SF02012)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

<http://www.wunderground.com/history/airport/KSFO/2012/1/1/DailyHistory.html?format=1>

SFO2013	<i>Data - Ambient Temperature for the City of San Francisco for all of 2013</i>
---------	---

Description

This is a data frame of Ambient temperature data, extracted from Weather Underground. Each row has two entries (columns). The Timestamp (YYYY-MM-DD HH:MM:SS) and the Temperature (in degrees F)

Usage

```
data(SFO2013)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

<http://www.wunderground.com/history/airport/KSFO/2013/1/1/DailyHistory.html?format=1>

USAirportWeatherStations	<i>Data - US Weather Stations ID's</i>
--------------------------	--

Description

This is a data frame of the 1602 stations in Weather Underground's database. The 4-letter "airport-Code" is used by functions to check and get the weather data.

Usage

```
data(USAirportWeatherStations)
```

Author(s)

Ram Narasimhan <ramnarasimhan@gmail.com>

References

http://www.wunderground.com/about/faq/US_cities.asp

Index

*Topic **data**

IntlWxStations, [10](#)

London2013, [11](#)

Mumbai2013, [11](#)

NewYork2013, [12](#)

SF02012, [12](#)

SF02013, [13](#)

USAirportWeatherStations, [13](#)

*Topic **package**

weatherData-package, [2](#)

checkDataAvailability, [3](#)

checkDataAvailabilityForDateRange, [3](#)

getCurrentTemperature, [4](#)

getDailyMinMaxTemp, [5](#)

getStationCode, [6](#)

getWeatherData, [7](#)

getWeatherForDate, [8](#)

getWeatherForYear, [9](#)

IntlWxStations, [10](#)

IsStationTypeInvalid, [10](#)

London2013, [11](#)

Mumbai2013, [11](#)

NewYork2013, [12](#)

SF02012, [12](#)

SF02013, [13](#)

USAirportWeatherStations, [13](#)

weatherData (weatherData-package), [2](#)

weatherData-package, [2](#)