

# Package ‘NOAAWeather’

March 21, 2018

**Type** Package

**Title** Get Weather Data from NOAA Weather Stations

**Version** 0.1.0

**Date** 2018-03-15

**Author** Sagar Ganapaneni <123saga@gmail.com>, Vikramjeet Singh <vikram.dceian@gmail.com>

**Maintainer** Sagar Ganapaneni <123saga@gmail.com>

**Description** Provides functions to retrieve real-time weather data from all NOAA (National Oceanic and Atmospheric Administration) stations and plot time series, boxplot, calendar heatmap and geospatial maps to analyze trends. NOAA is an American scientific agency within the United States Department of Commerce that focuses on the conditions of the oceans and the atmosphere. Stations are situated at 235 locations across 50 states within United States and are used to capture weather data. NOAA offers web services <<https://www.ncdc.noaa.gov/cdo-web/api/v2/>> that provide access to data collected from stations.

**Depends** R (>= 3.4),

**Imports** RCurl, jsonlite, tidyr, dplyr, lubridate, ggplot2, ggmap, ggExtra, grid, gridExtra, tcR, scales, stats, utils

**License** GPL

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**Suggests** knitr, rmarkdown, testthat

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2018-03-21 12:11:12 UTC

## R topics documented:

Distance_data_master . . . . .	2
getAllLocations . . . . .	2
getMetrics . . . . .	3
getSpatialPlotData . . . . .	4
getWeatherData . . . . .	4
Locations . . . . .	5
Metrics . . . . .	6
plot_boxplot . . . . .	6
plot_calhmap . . . . .	7
plot_spatial . . . . .	8
plot_tseries . . . . .	9
weather_data . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

Distance\_data\_master    *Locations data with geo spatial distance*

---

### Description

A data frame consisting of all available locations along with spatial distance between all combinations

### Usage

```
data(Distance_data_master)
```

---

getAllLocations    *Get All locations of NOAA Weather Stations*

---

### Description

get all locations information using API call

### Usage

```
getAllLocations(online = TRUE, state = NA)
```

### Arguments

online	if TRUE location data is pulled from the API, if FALSE, data is pulled from offline database
state	filter the final output based on the state code specified, default to all US states

**Value**

Dataframe with Location ID, Location name and State

**Author(s)**

Vikramjeet Singh (vjs)

**Examples**

```
locations <- getAllLocations (online=FALSE,state="CA")
```

---

<code>getMetrics</code>	<i>get all Available Metrics</i>
-------------------------	----------------------------------

---

**Description**

get all Available Metrics from NOAA Web API along with descriptions and units

**Usage**

```
getMetrics(online = TRUE)
```

**Arguments**

online            if TRUE metrics data is pulled from the API, if FALSE, data is pulled from offline database

**Value**

All metrics Information

**Author(s)**

Vikramjeet Singh (vjs)

---

```
getSpatialPlotData    generate data for plot_spatial()
```

---

### Description

generate data for plot\_spatial()

### Usage

```
getSpatialPlotData(online = TRUE, date = "2017-01-01",
  metric = "t_official")
```

### Arguments

online	if TRUE(default),realtime data is pulled from the API. if FALSE, data is pulled from sample offline database
date	date for which spatial plot is needed. Default: 2017-01-01
metric	metric for which spatial plot is needed. Default: "t_official"

### Value

a dataframe of values for the given metric and date

### Author(s)

Sagar Ganapaneni (sagar123)

---

```
getWeatherData    get Weather data
```

---

### Description

get Weather data from NOAA Web API for a given location, time period, and distance range in miles

Available metrics: "p\_official" : calculated Geonor precip total for hour (mm) "rh\_std" : RH std dev for hour (percent) "solarad" : Average solar radiation for the hour (W m-2) "t\_max" : Calculated maximum temp for hour (deg\_C) "t\_min" : Calculated minimum temp for hour (deg\_C) "t\_official" : calculated average temp for hour (deg\_C) "windspd" : average 1.5m wind speed for the hour (m s-1) "ws\_max" : 10sec maximum 1.5m wind speed for the hour (m s-1)

### Usage

```
getWeatherData(online = TRUE, location, state, from = "2017-01-01",
  to = "2017-01-02", range = NA)
```

**Arguments**

online	if TRUE(default),realtime data is pulled from the API. if FALSE, data is pulled from sample offline database
location	Valid location supported by NOAA, use <a href="#">getAllLocations</a> for view available locations.
state	Valid two letter code of US state
from	Beginning of date range. Default: 2017-01-01
to	End of date rage. Default: 2017-01-02
range	Select other available locations within the radial distance of specified location

**Value**

a dataframe of with all the metrics mentioned at hourly granularity

**Author(s)**

Sagar Ganapaneni (sagar123)

**Examples**

```
## Not run:
data <- getWeatherData(online=TRUE,
  location="Austin",state="TX",from="2017-01-10",
  to="2017-01-11")
data <- getWeatherData(online=FALSE,
  location="Austin",state="TX",from="2017-01-10",
  to="2017-01-11",range=120)

## End(Not run)
```

---

Locations

*Locations data*

---

**Description**

A data frame consisting of all available locations

**Usage**

```
data(Locations)
```

---

Metrics	<i>Metrics available via NOAA Weather API</i>
---------	---

---

**Description**

A data frame consisting of all metrics (with definitions, units) details

**Usage**

```
data(Metrics)
```

---

plot_boxplot	<i>box plot of weather data</i>
--------------	---------------------------------

---

**Description**

generate box plot for a given time period, measure, location, and state

**Usage**

```
plot_boxplot(online = TRUE, from = "2017-01-01", to = "2017-01-31",
  measure = "t_max", location = "Wolf Point", state = "MT")
```

**Arguments**

online	if TRUE(default),realtime data is pulled from the API. if FALSE, data is pulled from sample offline database
from	Beginning of date range. Default: 2017-01-01
to	End of date range. Default: 2017-01-31
measure	metric for which box plot is needed. Default: "t_max"
location	Valid location supported by NOAA, use <a href="#">getAllLocations</a> for view available locations. Default: "Wolf Point"
state	Valid two letter code of US state. Default: "MT"

**Value**

box plot for a given time period, measure, location, and state

**Author(s)**

Vikramjeet Singh (vjs)

## Examples

```
## Not run:
plot_boxplot(online=FALSE,from="2017-01-01",
to="2017-01-03", measure="t_max",
location='Austin', state='TX')
plot_boxplot(online=TRUE,from="2017-01-01",
to="2017-01-03", measure="t_max",
location='Austin', state='TX')

## End(Not run)
```

---

plot\_calhmap                      *calendar heat map of weather data*

---

## Description

generate calendar heat map for a given time period, measure, location, and state

## Usage

```
plot_calhmap(online = TRUE, from = "2017-01-01", to = "2017-01-31",
measure = "t_max", location = "Wolf Point", state = "MT", text = TRUE)
```

## Arguments

online	if TRUE(default),realtime data is pulled from the API. if FALSE, data is pulled from sample offline database
from	Beginning of date range. Default: 2017-01-01
to	End of date range. Default: 2017-01-31
measure	metric for which time series plot is needed. Default: "t_max"
location	Valid location supported by NOAA, use <a href="#">getAllLocations</a> for view available locations. Default: "Wolf Point"
state	Valid two letter code of US state. Default: "MT"
text	if TRUE, print the values of each day on heatmap, if FALSE values are not printed

## Value

calendar heat map for a given time period, measure, location, and state

## Author(s)

Vikramjeet Singh (vjs)

## Examples

```
## Not run:
plot_calhmap(online=FALSE,from="2017-01-01",
to="2017-01-03", measure="t_max",
location='Austin', state='TX')

## End(Not run)
```

---

plot\_spatial

*geo-spatial heat map of weather data*

---

## Description

generate geo-spatial heat map for a given date and measure across all NOAA weather stations

## Usage

```
plot_spatial(online = TRUE, Date = "2017-01-01", measure = "t_max")
```

## Arguments

online	if TRUE(default),realtime data is pulled from the API. if FALSE, data is pulled from sample offline database
Date	Beginning of date range. Default: 2017-01-01
measure	metric for which geo-spatial heat map is needed. Default: "t_max"

## Value

geo-spatial heat map for a given date and measure across all NOAA weather stations

## Author(s)

Vikramjeet Singh (vjs)

## Examples

```
## Not run:
plot_spatial(online=FALSE,
Date="2017-01-10",
measure="t_max")
plot_spatial(online=TRUE,
Date="2017-01-10",
measure="t_official")

## End(Not run)
```



---

plot\_tseries                    *time-series plot of weather data*

---

### Description

generate time series plot for a given time period, measure, location, and state

### Usage

```
plot_tseries(online = TRUE, from = "2017-01-01", to = "2017-01-31",  
             measure = "t_max", location = "Wolf Point", state = "MT")
```

### Arguments

online	if TRUE(default),realtime data is pulled from the API. if FALSE, data is pulled from sample offline database
from	Beginning of date range. Default: 2017-01-01
to	End of date range. Default: 2017-01-31
measure	metric for which time series plot is needed. Default: "t_max"
location	Valid location supported by NOAA, use <a href="#">getAllLocations</a> for view available locations. Default: "Wolf Point"
state	Valid two letter code of US state. Default: "MT"

### Value

time series plot for a given time period, measure, location, and state

### Author(s)

Vikramjeet Singh (vjs)

### Examples

```
## Not run:  
plot_tseries(online=FALSE,from="2017-01-01",  
             to="2017-01-02", measure="t_max",  
             location='Austin', state='TX')  
  
## End(Not run)
```

---

`weather_data`*Weather data for 2017 January*

---

**Description**

A data frame consisting of hourly data for metrics: ("p\_official", "rh\_std", "solarad", "t\_max", "t\_min", "t\_official", "windspd", "v

**Usage**

```
data(weather_data)
```

**Author(s)**

Sagar Ganapaneni (sagar123)

# Index

Distance\_data\_master, 2

getAllLocations, 2, 5–7, 9

getMetrics, 3

getSpatialPlotData, 4

getWeatherData, 4

Locations, 5

Metrics, 6

plot\_boxplot, 6

plot\_calhmap, 7

plot\_spatial, 8

plot\_tseries, 9

weather\_data, 10