Contents

getData {airbox}

Selecting airbox data

Description

getData can select the airbox data for a specified period of time.

Usage

getData(data)

Arguments

`data`	A character implies the month of the airbox data which is going to be selected. The character should has 7 digits with the form of **"YYYY-MM"**.

Value

getData returns a data frame of the airbox data for the selected period of time.

Note

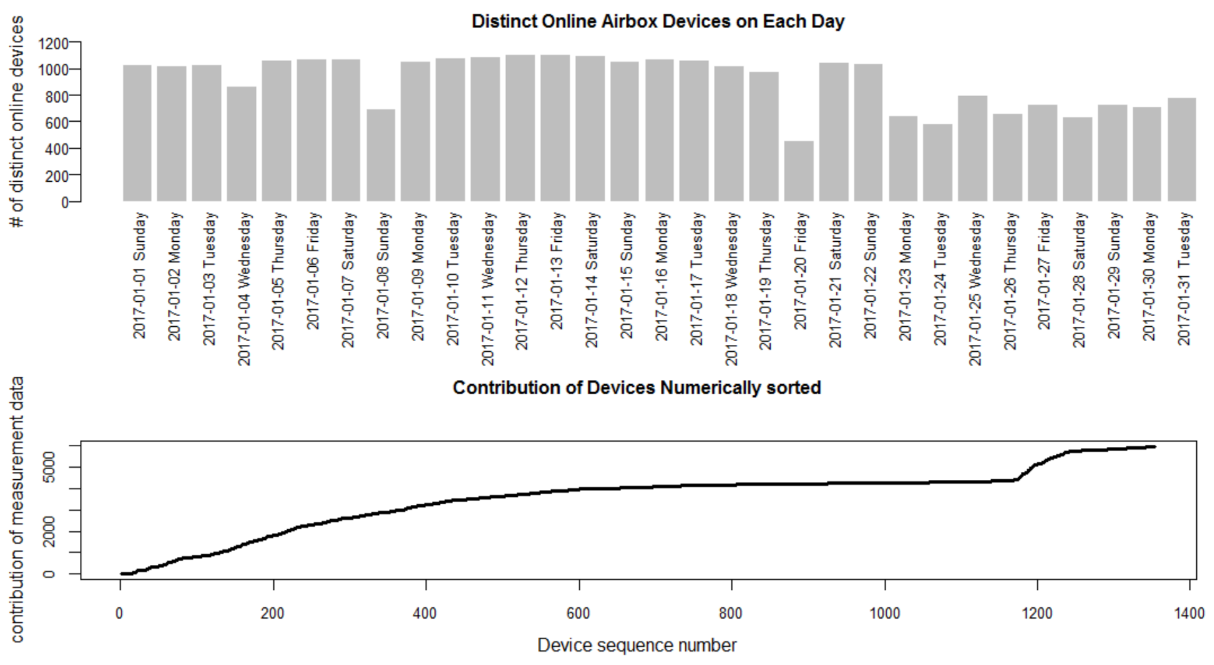
In this airbox package, only 3 months of airbox data, which includes the information from 2017-JAN to 2017-MAR, was provided. Also, the timezone of the time of airbox data is in GMT+8.

Examples

mydata <- getData(data="2017-01")

info {airbox}

Basic analysis of airbox data



Description

info is used to showing the states of online airbox devices. By inputing the data frame produced by getData, this function will output two figures. Figure one shows the number of distinct onlin airbox devices on each day. And figure two shows the number of measurement data for each airbox.

Usage

info(mydata)

Arguments

`mydata`	A data frame that is generated by the function getData or a data frame that has the construcion needed for analyze. The format of the contents and the column names of the data frame should be the same as the data which is provided in the package. The dataset which can be analysis by `info` should include following contents: (1)device_id(the id of airbox devices) (2)Date(in the format of yyyy-mm-dd) (3)Time(in the format of hr:min:sec) (4)concentration of PM2.5, PM10 and PM1 (5)Temperature (6)Humidity (7)location of airbox devices (latitude and longitude)

Details

"Distinct Onlin Airbox Devices on Each Day" Devices that return more than half of the maximum amount of returned data over the course of a day are considered as distinct online airbox devices. The x-axis of the plot is the dates which has been selected by the user. And the y-axis is the number of the distinct online devices. "Contribution of Devices Numerically sorted" The x-axis of this plot is the serial number which are sequence numbers from 1 to the total amount of the airbox devices. And the y-axis is the number of data measured by each airbox device.

Value

info returns two plots. The plot on the top is titled "Distinct Onlin Airbox Devices on Each Day", which shows the number of distinct onlin airbox devices on each day. And the plot on the bottom is titled "Contribution of Devices Numerically sorted", which shows the number of measurement data for each airbox. The details about the explanation of the variables in the plots are given under 'Details'

See Also

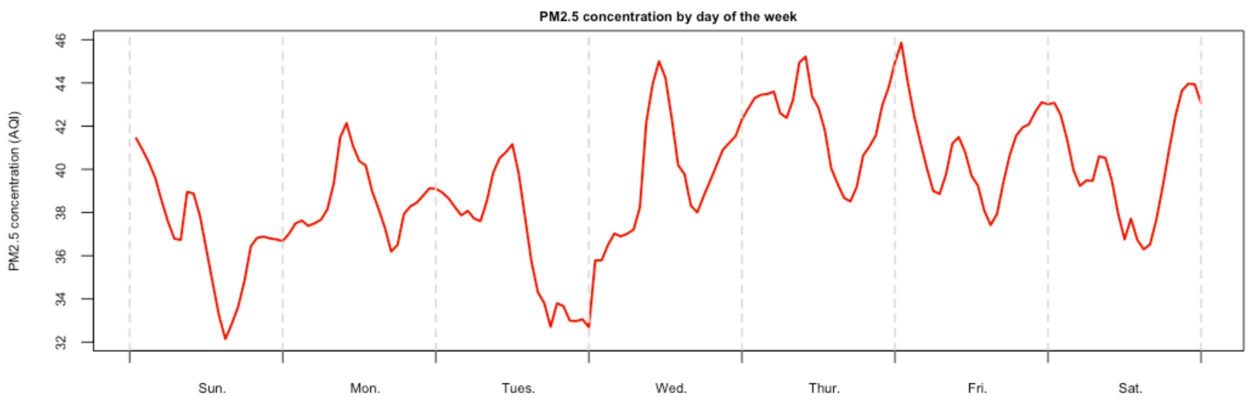
getData to select the airbox data. Or see wholedata which is provide in the package to know the constrction of the dataset. week_day to show the boxplot for the concentration of PM2.5.

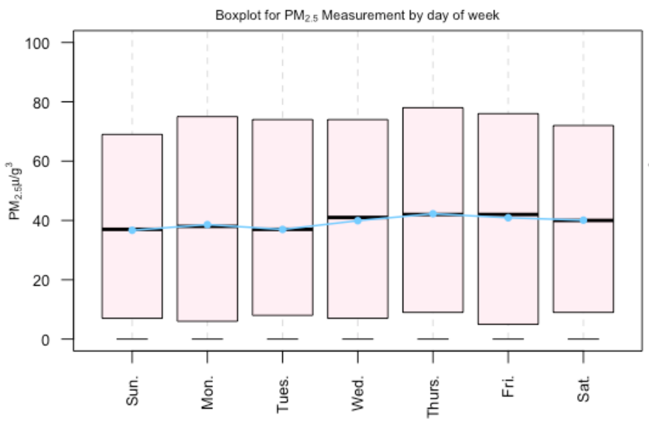
Examples

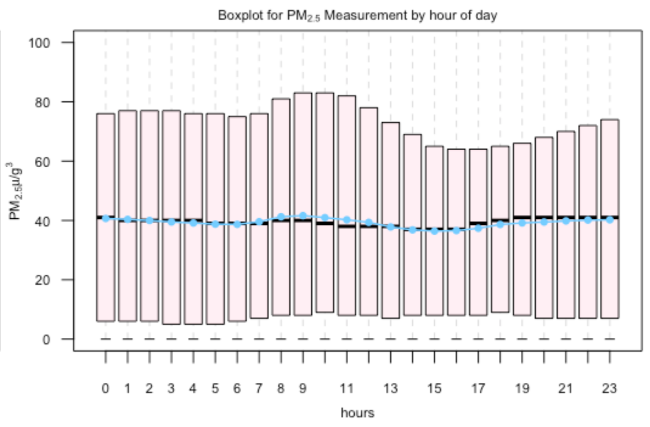
```
mydata <- getData(data="2017-01")
info(mydata = mydata)
```

week_day {airbox}

Present three plots indicating the average concentration of PM2.5, using weekday and hour as variables individually







Description

week_day is a function analyzing PM2.5 data that is input by using temporal factors. Boxplot for PM2.5 Measurement (factored by days in a week) that is returned uses the days in a week as factor, which are Sunday, Monday, Tuesday, Wednesday, Thrusday, Friday, and Saturday. Boxplot for PM2.5 Measurement (factored by hours in a day) that is returned uses the hours in a day as factor, which are the numbers between 0-23. PM2.5 concentration by day of the week is a line chart that is returned uses the hours in a week as factor, which are 24 hours in each day of the week.

Usage

week_day(mydata)

Arguments

`mydata`	A dataset that is either a subset of `wholedata`, or a dataset that has the construcion needed for analyze. A dataset that is able to analyze with `week_day` should contain a POSIXlt form of time with a certain format (please see the Time_1 column in wholedata), and PM2.5 concentration in specific time named PM2.5

Value

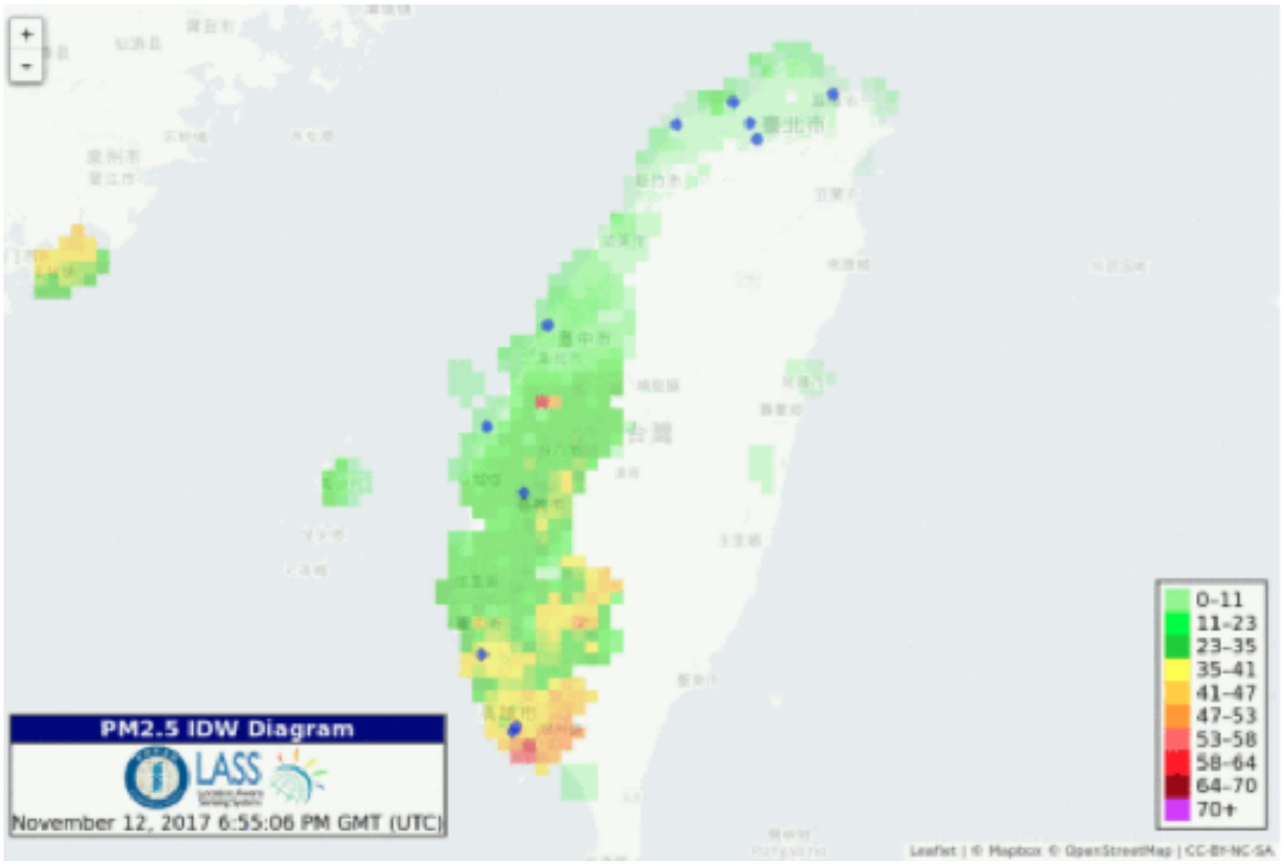
week_day returns three plot. The plot on the bottom left is titled "Boxplot for PM2.5 Measurement", with a subtitle of "(factored by days in a week)", which is a plot indicating the concentration of PM2.5 during a week using days as variables. The plot on the bottom right is titled "Boxplot for PM2.5 Measurement" with a subtitle of "(factored by hours in a day", which is a plot indicating the concentration of PM2.5 during a day using hour as variables. The plot on the top is titled "PM2.5 concentration by day of the week" is a line graph indicating the concentration of PM2.5 during a week using hours of each day as varibles.

Examples

week_day(getData('2017-02'))

animation {airbox}

Download the gif file of the PM2.5 distribution in Taiwan


PM2.5 IDW Diagram
LASS
November 12, 2017 6:55:06 PM GMT (UTC)
0-11
11-23
23-35
35-41
41-47
47-53
53-58
58-64
64-70
70+
Leaflet | © Mapbox © OpenStreetMap | CC-BY-NC-SA

Description

animation is used to download the animated distribution chart from https://data.lass-net.org/GIS/IDW/. The chart shows the concentration of PM2.5 , which is reported from each airbox device in the last 24 hours.

Usage

animation(w = 450, h = 300)

Arguments

w	The value determines the width of the picture. The default value is 450.
h	The value determines the height of the picture. The default value is 300.

Details

The gif picture downlaoded by animation is stored in user file and the picture is named as "taiwan24.gif".

Value

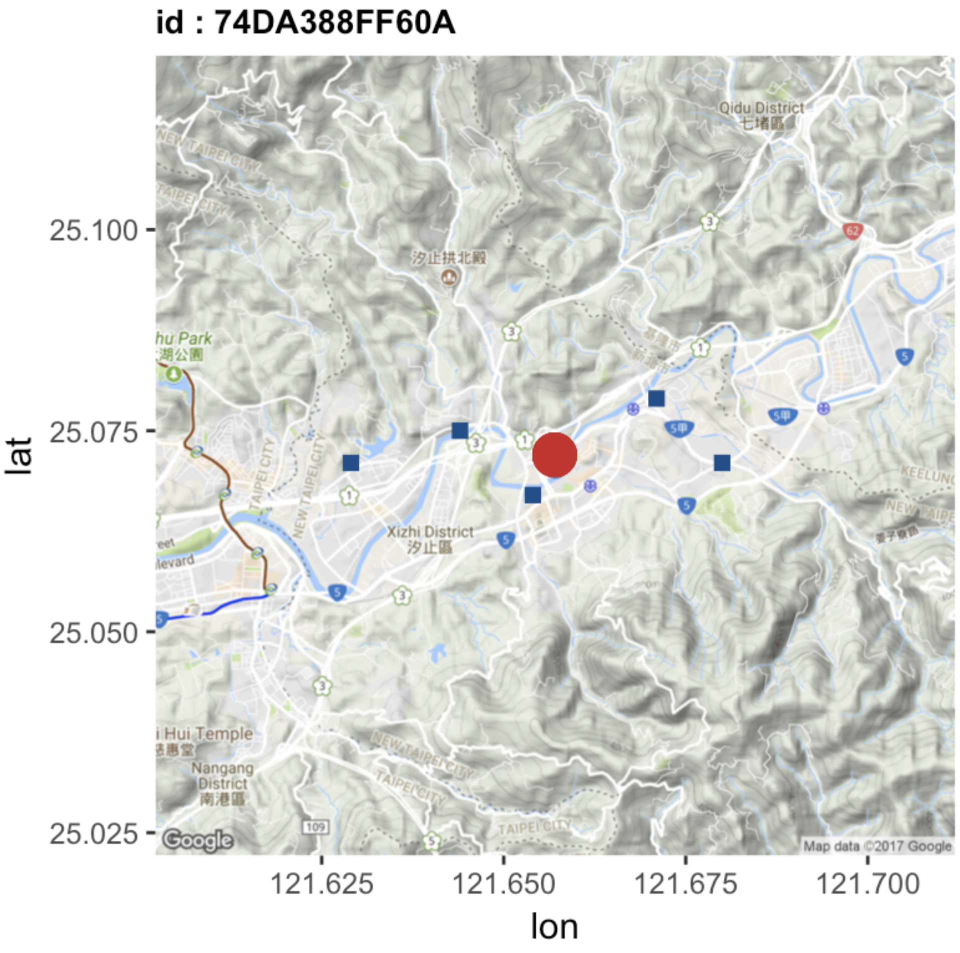
A gif file of the PM2.5 distribution in Taiwan.

Examples

animation(w = 450 , h = 300)

neighbor {airbox}

Return the neighbor devices within selected area



Description

neighbor is used when one wants to know the devices within a selected area with a chosen device being the area's center. The device that is interested in as being the center, and the distance will be input in linkneighbor, and a map will be returned for the neighbor devices.

Usage

neighbor((mydata = getData('2017-01') , id = '74DA388FF60A' , dist = 3000 , unit = 'm' , zoom = 13))

Arguments

`mydata`	A data frame which is generated by the function getData. The contents and the column names of the data frame should be the same as the data which is provided in the package.
`id`	A character implies the device's id that is interested for spatial-anomaly-testing. If the id that is entered does not contain data, the function will stop. The default setting for id is '74DA3895C392'.

`dist`	A numeric implies the distance that is interested to be included in for spatial-anomaly-testing. An unit can be chose for the "unit" argument, and the default setting is a meter measurement.
`unit`	A character that implies the unit that should be attached to the numeric distance. "unit" can be either "m" or "km", which represents meter or kilometer. Other inputs are not available for "unit". The default setting is "m", which represents a meter measuring unit.
`zoom`	map zoom, an integer from 3 (continent) to 21 (building), default value 13. openstreetmaps limits a zoom of 18, and the limit on stamen maps depends on the maptype. "auto" automatically determines the zoom for bounding box specifications, and is defaulted to 10 with center/zoom specifications. maps of the whole world currently not supported. The argument is directly referenced from [Package ggmap version 2.6.1

Value

neighbor returns a map that should contain the device entered, and the neighbor devices in the selected distance. The large red stuffed circle represents the entered device. The blue stuffed squares represent the neighbor devices. The id of the neighbor devices, the longtitude and latitude of the center device would be printed out, as well.

See Also

getData to select the airbox data. week_day to show the boxplot for the concentration of PM2.5. spatial_anomaly to examine for spatial anomaly in PM2.5 concentration.

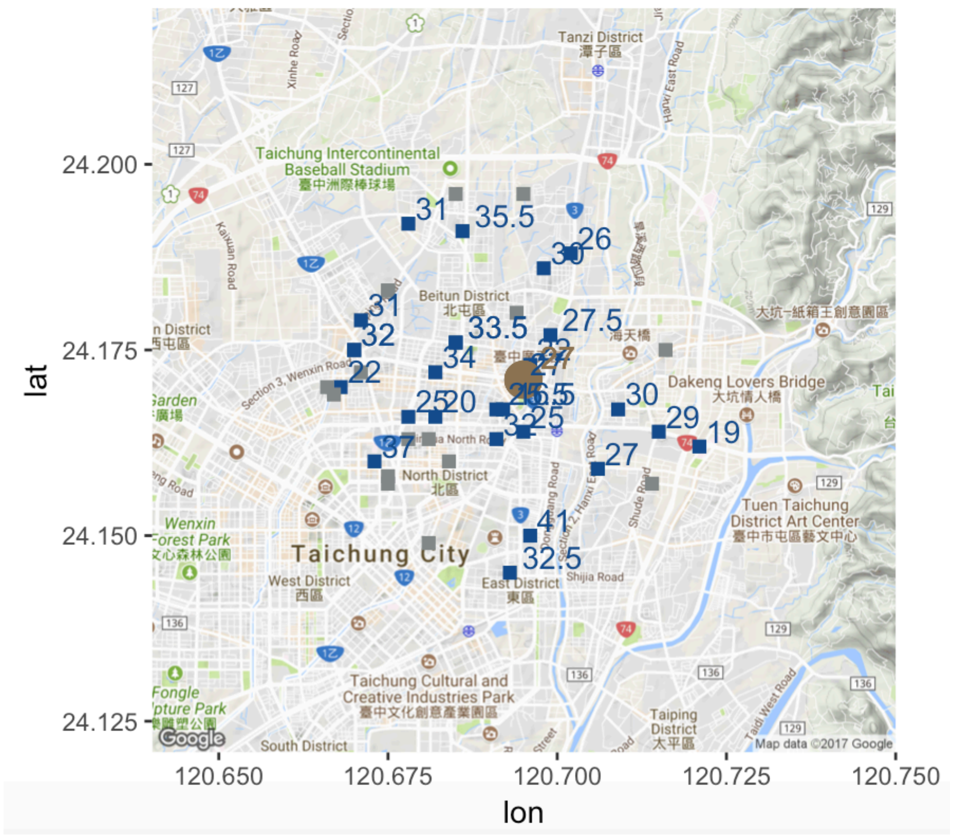
Examples

neighbor((mydata = getData('2017-01') , id = '74DA388FF60A' , dist = 3000 , unit = 'm' , zoom = 13))
neighbor()

spatial_anomaly {airbox}

Determine if there is spatial anomaly for the input device

id : 74DA3895C32E, time : 2017-02-24 09:50:48



```
> spatial_anomaly('74DA3895C32E')
[1] "Adjusted time : 2017-02-24 09:50:48"
 [1] "74DA3895C20A" "74DA3895C3D0" "74DA3895C47A" "74DA3895C310" "74DA3895C400"
 [6] "74DA3895C3D6" "74DA3895C270" "74DA3895C2B4" "74DA3895C32E" "74DA3895C2B8"
[11] "74DA3895C3FA" "74DA3895C216" "74DA3895C4AC" "74DA3895C576" "74DA3895C34A"
[16] "74DA3895C1F8" "74DA3895C3DE" "74DA3895C3DA" "74DA3895C40E" "74DA3895C560"
[21] "74DA3895C3EE" "74DA3895C350" "74DA3895C45E" "74DA3895C490" "74DA3895C392"
[26] "74DA3895C404" "74DA3895C528" "74DA3895C210" "74DA38AF48BE" "74DA3895C35A"
[31] "74DA3895C3D4" "74DA38AF4736" "74DA38AF474A" "74DA3895C42E" "74DA3895C3B4"
[36] "74DA3895C560" "74DA3895C2B4" "74DA38AF496C" "74DA38AF4908" "28C2DDDD45E6"
[41] "74DA38AF48BE" "74DA38AF47C4" "74DA38AF4974" "74DA38AF4960"
[1] "latitude of id:120.695" "longitude of id:24.171"
[1] "Median of PM2.5 concentration in devices in selected area is : 29"
[1] "There is no spatial anomaly in chosen id"
```

Description

spatial_anomaly calculates the median of the input device's PM2.5 concentration in the between 15 minutes prior or after the input time. If there is no specific PM2.5 data under the entered time, an adjusted time will replace the entered time by tracing the time with the least difference either prior or after. The chosen device's median concentration will be compared with the cross-sectional data within all devices appearing in selected distance.

A threshold will be determined in the chosen device's median PM2.5 concentration. There is description of how the threshold is decided in the "remark" section below. If the absolute difference between the chosen device's PM2.5 concentration and the median within neighbor devices in entered distance is larger than the threshold level, an anomaly is determined for the chosen device; and will return a stuffed red circle in a map. In contrast, if the difference of PM2.5 median concentration between chosen device and device within entered distance, there is no abnomaly, which returns with a stuffed brown circle in a map.

Usage

spatial_anomaly(id , time , distance , unit , zoom)

Arguments

`id`	A character implies the device's id that is interested for spatial-anomaly-testing. If the id that is entered does not contain data, the function will stop. The default setting for id is '74DA3895C392'.
`time`	A string of characters that defines the specific time that is interested for spatial-anomaly-testing. The character string should follow the format:"Year(four digits)-month(two digits)-day(two digits) Hour(two digits):Minute(2 digits):Seconds(2 digits)". For instance, "2017-01-25 18:24:56". There should be remark that there is a space between the date and the time, and format in under an 24-hour-setting.
`distance`	A numeric implies the distance that is interested to be included in for spatial-anomaly-testing. An unit can be chose for the "unit" argument, and the default setting is a meter measurement.
`unit`	A character that implies the unit that should be attached to the numeric distance. "unit" can be either "m" or "km", which represents meter or kilometer. Other inputs are not available for "unit". The default setting is "m", which represents a meter measuring unit.
`zoom`	map zoom, an integer from 3 (continent) to 21 (building), default value 13. openstreetmaps limits a zoom of 18, and the limit on stamen maps depends on the maptype. "auto" automatically determines the zoom for bounding box specifications, and is defaulted to 10 with center/zoom specifications. maps of the whole world currently not supported. The argument is directly referenced from [Package ggmap version 2.6.1]

Details

1. If the entered argument for time does not access to a specific data in the chosen device, the time will be adjusted to the closest time either prior or later to the entered time. The adjusted time will be printed out in the console section. 2. Beside the map returned, there will be several things printed out. The adjusted time will be printed out if there is a time adjustment, which happened when there is no directly specific data for the chosen device at the entered time. If the time entered does not need to be adjusted, there will be no "Adjusted time" printed out. In addition, a character list of neighbor device's id will be printed out. Then, latitude and longitude of chosen device will be printed out as well. The median of PM2.5 concentration in the neighbor area (which is the area in the distance

entered) will be printed. Last, there will be either information for "There is no spatial anomaly in chosen id." or "There is spatial anomaly in choson id" being printed. 3. The determination for threshold depends on the chosen device's PM2.5 concentration. When 0 <= PM2.5 concentration < 12, the threshold would be 3.0; when 12 <= PM2.5 concentration < 24, the threshold would be 6.6; when 24 <= PM2.5 concentration < 36, the threshold would be 9.35; when 36 <= PM2.5 concentration < 42, the threshold would be 13.5; when 42 <= PM2.5 concentration < 48, the threshold would be 17.0; when 48 <= PM2.5 concentration < 54, the threshold would be 23.0; when 54 <= PM2.5 concentration < 59, the threshold would be 27.5; when 59 <= PM2.5 concentration < 65, the threshold would be 33.5; when 65 <= PM2.5 concentration < 70, the threshold would be 40.5; when PM2.5 concentration >= 70, the threshold would be 91.5.

Value

sptial_anomaly returns in a map. If a spatial anomaly exists for the chosen device at the entered or adjusted time, the device will imply a brown stuffed circle in its latitude and longitude; if a spatial anomaly does not exist for the chosen device at the entered or adjusted time, the device will imply a red stuffed circle in its latitude and longitude. There will be a number indicating the median PM2.5 concentration between the prior and later 15 minutes of the entered/adjusted time on the upper-right of the circle. The squares in the maps imply the neighbor devices. Blue squares represents those with data recorded between the prior and later 15 minutes of the entered/adjusted time, and the median for the individual device is presented at the upper-right of the squares, as well. The gray squares represent those does not exist recorded data between the prior and later 15 minutes of the entered/adjusted time.

See Also

getData to select the airbox data. week_day to show the boxplot for the concentration of PM2.5. temporal_anomaly to examine for temporal anomaly in PM2.5 concentration.

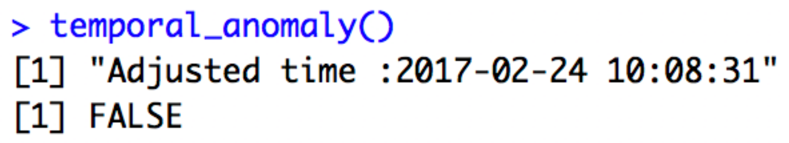
Examples

```
#example for the input time being adjusted (there is no specific PM2.5 data under the entered time)
spatial_anomaly(id = '74DA3895C392' , time = '2017-02-24 10:10:30' , dist = 3000 , unit = 'm' , zoom = 13)
spatial_anomaly()

#example for the input time without being adjusted
spatial_anomaly(id = '74DA3895C392' , time = '2017-02-24 10:08:31' , dist = 3000 , unit = 'm' , zoom = 13)
```

temporal_anomaly {airbox}

Determine if there is temporal anomaly for the input device



```
> temporal_anomaly()
[1] "Adjusted time :2017-02-24 10:08:31"
[1] FALSE
```

Description

temporal_anomaly trace back, and compare with the closet prior data for the chosen device at the certain time entered. A threshold would be determined by the chosen device's PM2.5 concentration at the certain time entered. The absolute difference between the PM2.5 concentration at the entered time and the closest prior concentration for the chosen device would be compared with the threshold. If the threshold is exceeded, a TRUE representing temporal anomaly would be returned. In contrast, if the threshold is not exceeded, a FALSE representing no temporal anomaly would be exceeded.

Usage

temperol_anomaly(id = '74DA3895C392' , time = '2017-02-24 10:10:30')

Arguments

`id`	A character implies the device's id that is interested for spatial-anomaly-testing. If the id that is entered does not contain data, the function will stop. The default setting for id is '74DA3895C392'.
`time`	A string of characters that defines the specific time that is interested for spatial-anomaly-testing. The character string should follow the format:"Year(four digits)-month(two digits)-day(two digits) Hour(two digits):Minute(2 digits):Seconds(2 digits)". For instance, "2017-01-25 18:24:56". There should be remark that there is a space between the date and the time, and format in under an 24-hour-setting.

Details

1. If the entered argument for time does not access to a specific data in the chosen device, the time will be adjusted to the closest time either prior or later to the entered time. The adjusted time will be printed out in the console section. 2. The threshold would be determined by the PM2.5 concentrtion at the entered time for the chosen device. If the PM2.5 concentration is below 30, the threshold would be set as 5; if the PM2.5 concentration exceeds 30, the threshold would be one six of the PM2.5 concentration at the entered time for the chosen device. For example, if the PM2.5 concentration at the entered time for the id entered is 72, the threshold would be 72/6 = 12.

Value

temporal_anomaly returns either TRUE or FALSE. If TRUE is returned, there is temporal anomaly existence for the chosen device. If FALSE is returned, there is no temporal anomaly existence. If there is no recorded PM2.5 concentration at the entered time for the chosen device, an adjusted time will be set by tracking a closest time that contain a record for PM2.5 concentration. The adjusted time may be either prior or later than the time entered, only if it is the nearest time.

See Also

getData to select the airbox data. week_day to show the boxplot for the concentration of PM2.5. spatial_anomaly to examine for spatial anomaly in PM2.5 concentration.

Examples

```
#example for the input time being adjusted (there is no specific PM2.5 data under the entered time)
temporal_anomaly(id = '74DA3895C392' , time = '2017-02-24 10:10:30')
temporal_anomaly()

#example for the input time without being adjusted
temporal_anomaly(id = '74DA3895C392' , time = '2017-02-24 10:08:31')
```