Sofia air measurement

Akila

October 24, 2018

### Project Goal

Goal of this project is to predict PM10 pollutant in Sofia in next 24 hours.

### Reading the data into environment

In here I am loading the required R packages into R session.

library(lubridate)  
 library(dplyr)  
 library(geohash)  
 library(rworldmap)  
 library(ggmap)

Here I am importing required datasets to the environment.

airtube2017 <- read.csv("Air Tube/data\_bg\_2017.csv", stringsAsFactors = FALSE)  
airtube2018 <- read.csv("Air Tube/data\_bg\_2018.csv", stringsAsFactors = FALSE)  
topography <- read.csv("TOPO-DATA/sofia\_topo.csv")

Lets look at the structure of the imported data.

str(airtube2017)

## 'data.frame': 651492 obs. of 7 variables:  
## $ time : chr "2017-09-06T20:00:00Z" "2017-09-06T20:00:00Z" "2017-09-06T20:00:00Z" "2017-09-06T20:00:00Z" ...  
## $ geohash : chr "sx8d5r7wmxr" "sx8d6zjg5h8" "sx8dk3k2wr6" "sx2rj28e0gs" ...  
## $ P1 : int 9 9 8 5 1 29 15 12 21 14 ...  
## $ P2 : int 8 8 7 5 1 15 11 10 16 13 ...  
## $ temperature: int 14 0 15 18 11 14 14 19 19 18 ...  
## $ humidity : int 55 0 52 47 68 63 54 67 55 10 ...  
## $ pressure : int 0 0 92655 97448 94823 93785 94841 101342 101163 0 ...

str(airtube2018)

## 'data.frame': 2958654 obs. of 7 variables:  
## $ time : chr "2018-01-01T00:00:00Z" "2018-01-01T00:00:00Z" "2018-01-01T00:00:00Z" "2018-01-01T00:00:00Z" ...  
## $ geohash : chr "sx3wvzu7f6h" "sx3wypu7fdn" "sx86yxxv72r" "sx2qvybw9tb" ...  
## $ P1 : int 55 103 265 227 187 23 42 371 493 420 ...  
## $ P2 : int 34 51 130 94 110 11 24 208 223 235 ...  
## $ temperature: int 4 9 0 -1 4 3 7 4 1 1 ...  
## $ humidity : int 64 53 78 93 67 69 64 66 75 73 ...  
## $ pressure : int 99161 99109 94085 96814 100114 97352 98556 95532 95527 95155 ...

str(topography)

## 'data.frame': 196 obs. of 3 variables:  
## $ Lat : num 42.6 42.6 42.6 42.6 42.6 ...  
## $ Lon : num 23.2 23.2 23.2 23.3 23.3 ...  
## $ Elev: num 1184 1333 1505 1586 1533 ...

You can see the time variable in airtube data imported as “character” data. Its not usable. Lets convert the character data into time data field.

airtube2017$time <- ymd\_hms(airtube2017$time)  
airtube2018$time <- ymd\_hms(airtube2018$time)

Get the geohashes not in 2017 but not in 2018. Because if the data collection is discontinued after 2017 is not usable for analysis.

listA <- setdiff(airtube2017$geohash,airtube2018$geohash)  
listA

## [1] "sx2rj28e0gs" "sx2rj3nke6c" "sx2qvy4xrkm" "sx8duqvvuyt" "sx8e1s09n61"  
## [6] "sx8ddxesuy1" "sx8dfe5esgc" "sx8dezkrjet" "m-2105171" "sx8d9nfjsk2"  
## [11] "sx8dfy4ytuc"

there are 11 geolocations not in 2018 but in 2017. Lets remove these locations from 2017 dataset.

airtube2017\_selected <- subset(airtube2017, !(geohash %in% listA))

Now we can combine both 2017 filtered dataset and 2018 dataset.

airtube\_All<-bind\_rows(airtube2017\_selected,airtube2018)

Lets check how many geolocations are missing from the dataset.

sum(is.na(airtube\_All$geohash))

## [1] 0

It seems no geohash has “NA” for their ID.

Final objective here is to predict forecast to next 24 hrs in each geo location. Threfore lets get summarized values for each geo location.

airtube\_All\_summary <- airtube\_All %>% group\_by(geohash) %>%  
 summarise(n = n(), tmax = max(temperature), tmin = min(temperature), Days = max(time) - min(time)) %>%  
 arrange(Days)  
airtube\_All\_summary

## # A tibble: 1,254 x 5  
## geohash n tmax tmin Days  
## <chr> <int> <dbl> <dbl> <time>  
## 1 sx8dc3dv5bs 1 0 0 0.000000 hours  
## 2 sx8dc74gegu 1 28 28 0.000000 hours  
## 3 sx6ntd094dt 32 10 2 1.291667 hours  
## 4 sx86yynqthy 34 7 0 1.375000 hours  
## 5 sx8dmum9hkd 41 28 14 2.000000 hours  
## 6 sx8em0tftye 58 8 -1 2.375000 hours  
## 7 sx8dev3x03p 53 27 3 2.666667 hours  
## 8 sxe99peg8tq 25 13 0 2.875000 hours  
## 9 4 15 14 3.000000 hours  
## 10 sx8dtyqgpmw 4 20 19 3.000000 hours  
## # ... with 1,244 more rows

In here each geohashes grouped and get details about each group.You can see number of observations in each group, highest temparature and lowest temparature and duration of the time period which the data collected.

Lets convert geohash into latitude and longitude. Then we can use the location details on a map later.

airtube\_All\_extended = airtube\_All\_summary %>%  
 bind\_cols(gh\_decode(airtube\_All\_summary$geohash))  
  
head(airtube\_All\_extended)

## # A tibble: 6 x 9  
## geohash n tmax tmin Days lat lng  
## <chr> <int> <dbl> <dbl> <time> <dbl> <dbl>  
## 1 sx8dc3dv5bs 1 0 0 0.000000 hours 42.68000 23.26200  
## 2 sx8dc74gegu 1 28 28 0.000000 hours 42.68800 23.26200  
## 3 sx6ntd094dt 32 10 2 1.291667 hours 41.93500 25.55500  
## 4 sx86yynqthy 34 7 0 1.375000 hours 42.70500 23.15700  
## 5 sx8dmum9hkd 41 28 14 2.000000 hours 42.60655 23.46355  
## 6 sx8em0tftye 58 8 -1 2.375000 hours 42.76200 23.43100  
## # ... with 2 more variables: lat\_error <dbl>, lng\_error <dbl>

You can see the lat and lon information inserted into the dataset.

In this dataset we have a record without geohash code.

subset(airtube\_All\_extended, geohash =="")

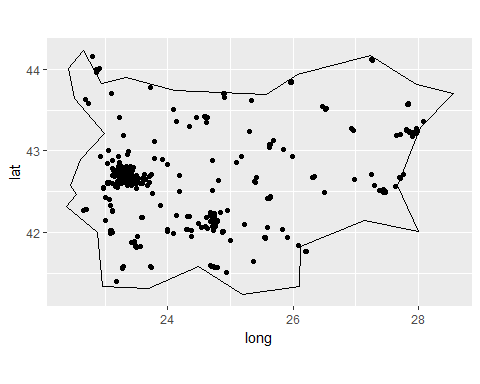
## # A tibble: 1 x 9  
## geohash n tmax tmin Days lat lng lat\_error lng\_error  
## <chr> <int> <dbl> <dbl> <time> <dbl> <dbl> <dbl> <dbl>  
## 1 4 15 14 3 hours 0 0 90 180

Lets remove it from dataset

airtube\_All\_extended <- airtube\_All\_extended %>% filter(geohash != "")

Lets create a map of data collection locations in Bulgaria map.

world\_map <- getMap()  
Bulgaria <- map\_data(world\_map, region = "bulgaria")  
  
plot1 <- ggplot() + geom\_polygon(data = Bulgaria, aes(x = long, y = lat, group = group),   
 fill = NA, col = "black") + coord\_fixed(1.3) +  
 geom\_point(data =airtube\_All\_extended, aes(x = lng, y = lat) )  
plot1

 As per the map, data collection points distributed all over the country but concentrated on main 3 different towns.

**### Filter rules**

For the analysis , small time durations of observations will not be sufficient. Therefore I am removing date duration less than 2 weeks.

airtube\_All\_extended <- airtube\_All\_extended %>% filter(Days >= 14)  
  
airtube\_All\_extended

## # A tibble: 1,203 x 9  
## geohash n tmax tmin Days lat lng  
## <chr> <int> <dbl> <dbl> <time> <dbl> <dbl>  
## 1 sx3x74rjsjg 338 24 1 14.04167 hours 42.06900 24.75100  
## 2 sx8dc2vj9z1 338 26 -3 14.04167 hours 42.67600 23.26500  
## 3 sx8f4vc441d 338 22 -4 14.04167 hours 42.57100 23.67700  
## 4 sx9w8h35mg7 338 23 -4 14.04167 hours 43.35400 24.61100  
## 5 sx8e6q4zm1m 341 22 -2 14.16667 hours 42.79300 23.30600  
## 6 sx8dcru746x 294 23 1 14.37500 hours 42.71400 23.26400  
## 7 sxd43wztqd6 371 23 5 15.41667 hours 42.62100 25.38900  
## 8 sx8d6wr7c71 384 30 0 15.95833 hours 42.61800 23.32300  
## 9 sx8ds1b4d0v 17 4 1 16.00000 hours 42.63700 23.37900  
## 10 sx8deg2bjfr 394 0 0 16.37500 hours 42.64482 23.36921  
## # ... with 1,193 more rows, and 2 more variables: lat\_error <dbl>,  
## # lng\_error <dbl>

We only consider the data points related to Sofia