Visualization of air quality in Buffalo and the effect factors

Xiaohan Rui

November 2, 2017

Introduction

Each people are concerned about air quality because we need to breathe all the time. And with the air quality index data, we can know how clean or polluted our outdoor or indoor air is, along with associated health effects that may be of concern. And these AQI data uses different colors to present levels which tell people when to take action to protect their health.

However, many people like vivid images more than cold numbers. Based on this situation, I would like to make our statistic results more understandable and interesting for people to look and read. So calendar heatmap may be a good choice. People can see clearly from the heatmap how many unhealthy air days in the past months or years.

Additionally, it is not enough to just know what is air quality. So learning the possible infuencing factors is also important. I want to use daily temperature and wind data from the GHCND data we used in Geo503 chapter 8 class. With the calendar heatmap, we can add in those possible influential data to find if direct relationships exsist. Also some statistics should be done to proof the relationships. If the results show that low AQI days may be effected by low wind speed and high daily temperature, it would be a powerful reminder to let people make preventive measures in such weather, such as wearing a mask.

Materials and methods

Data: 1. AQI data is downloaded from EPA AIR data website: (You need to choose the geographic positons and the pollutant, such as PM2.5) Here is the website link: https://www.epa.gov/outdoor-air-quality-data (https://www.epa.gov/outdoor-air-quality-data)

2. GHCND variables: AWND: Average daily wind speed (meters per second or miles per hour as per user preference); TMAX: Maximum temperature; TMIN: Minimum temperature;

Here's my first code chunk. Load any required packages in a code chunk (you may need to install some packages):

```
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(openair)
library(ggplot2)
library(readxl)
library(rnoaa)
library(ggmap)
library(openair)
library(climdex.pcic)
```

```
## Loading required package: PCICt
```

Change the direction of the project and load the PM2.5 air quality data in Buffalo in 2016:(some functions in the packages can only recognize “date” as the column name so we first need to change the name.)

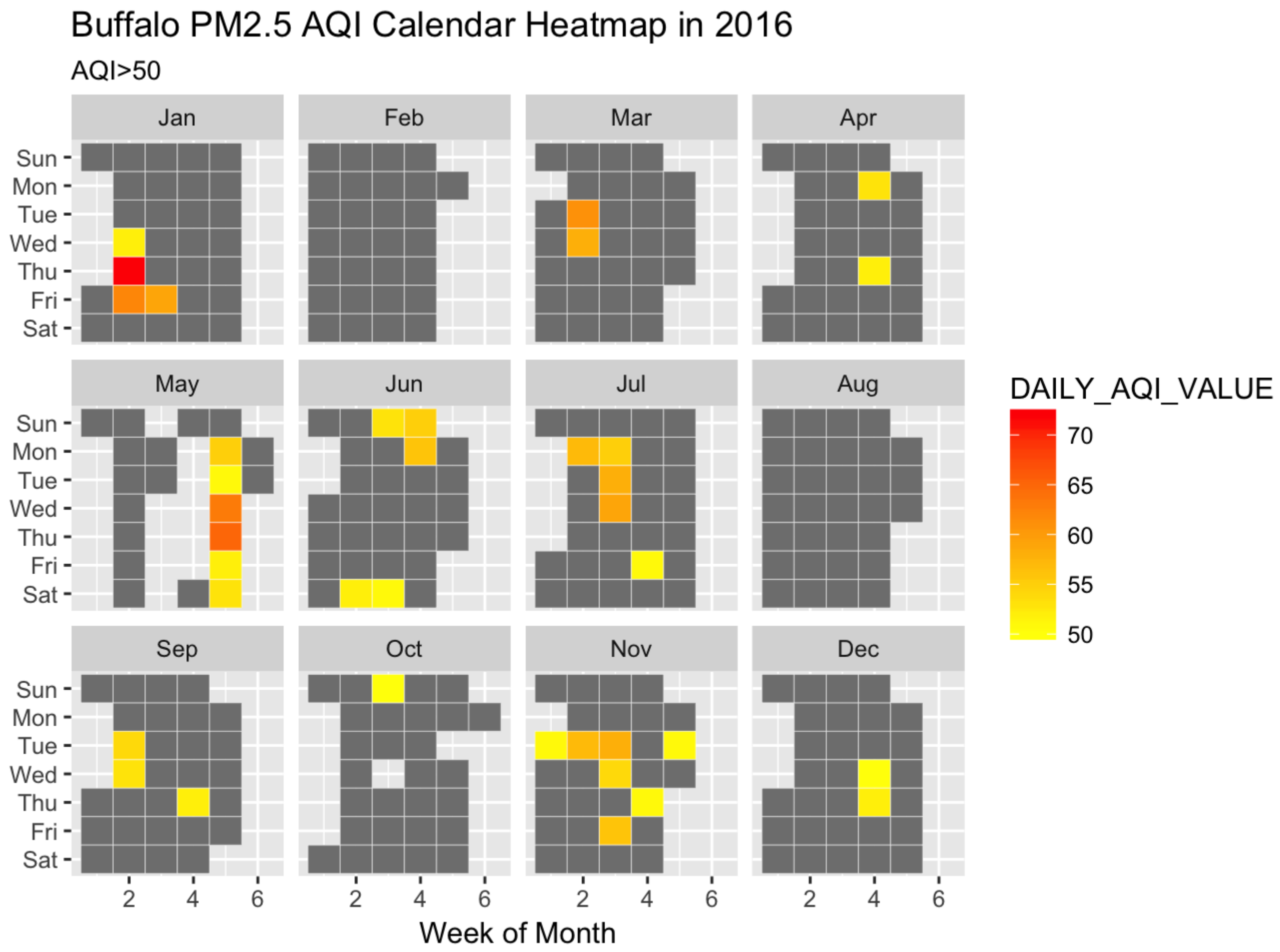
```
setwd("/Users/xh/Desktop")
mytable=read_excel("Data/Buffalo_PM2.5_AQI.xlsx")
names(mytable)[1]<-"date"
```

According to the method mentioned in reference[1], I set some parameters to make the plots below more easier and understandable to be shown:

```
dat=mytable
dat$month<-as.numeric(as.POSIXlt(dat$date)$mon+1)
dat$monthf<-
factor(dat$month,levels=as.character(1:12),labels=c("Jan","Feb","Mar","Apr","May","Jun","J
ul","Aug","Sep","Oct","Nov","Dec"),ordered=TRUE)
dat$weekday<-as.POSIXlt(dat$date)$wday
dat$weekdayf<-factor(dat$weekday,levels=rev(0:6),labels=rev(c("Sun","Mon","Tue","Wed","T
hu","Fri","Sat")),ordered=TRUE)
dat$week <- as.numeric(format(dat$date,"%W"))
dat<-ddply(dat,.(monthf),transform,monthweek=1+week-min(week))
```

This can take a few seconds to show the plot about Calendar Heatmap of PM2.5 air quality in Buffalo:(in the plot I filter days which have AQI more then 50, just because in Air Quality Index(AQI) leves, more than 50 but less than 100 means moderate, more than 100 but less than 150 means unhealthy for sensitive groups.)

```
ggplot(dat,aes(monthweek,weekdayf,fill=DAILY_AQI_VALUE))+

  geom_tile(colour='white') +

  facet_wrap(~monthf ,nrow=3) +

  scale_fill_gradient(space="Lab",limits=c(50, max(dat$DAILY_AQI_VALUE)),low="yellow", high="red") +

  labs(title="Buffalo PM2.5 AQI Calendar Heatmap in 2016",subtitle="AQI>50",x="Week of Month",y="")
```



Then turns to influential factors:

```
datadir="data"
st = ghcnd_stations()
st=dplyr::filter(st,element%in%c("TMAX","TMIN","AWND"))
coords=as.matrix(geocode("Buffalo, NY"))
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Buffalo,%20NY&sensor=false
```

```
dplyr::filter(st,
              grepl("BUFFALO",name)&
              between(latitude,coords[2]-1,coords[2]+1) &
              between(longitude,coords[1]-1,coords[1]+1)&
         element=="TMAX")
```

```
## # A tibble: 3 x 11
##            id latitude longitude elevation state       name gsn_flag
##         <chr>    <dbl>     <dbl>     <dbl> <chr>      <chr>    <chr>
## 1 USC00301010  42.8833  -78.8833    -999.9    NY    BUFFALO
## 2 USC00301018  42.9333  -78.9000     177.1    NY BUFFALO #2
## 3 USW00014733  42.9486  -78.7369     211.2    NY    BUFFALO
## # ... with 4 more variables: wmo_id <chr>, element <chr>,
## #   first_year <int>, last_year <int>
```

```
d=meteo_tidy_ghcnd("USW00014733",
                   var = c("TMAX","TMIN","AWND"),
                   keep_flags=T)
head(d)
```

```
## # A tibble: 6 x 14
##            id       date  awnd mflag_awnd mflag_tmax mflag_tmin qflag_awnd
##         <chr>     <date> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 USW00014733 1938-05-01  <NA>       <NA>                             <NA>
## 2 USW00014733 1938-05-02  <NA>       <NA>                             <NA>
## 3 USW00014733 1938-05-03  <NA>       <NA>                             <NA>
## 4 USW00014733 1938-05-04  <NA>       <NA>                             <NA>
## 5 USW00014733 1938-05-05  <NA>       <NA>                             <NA>
## 6 USW00014733 1938-05-06  <NA>       <NA>                             <NA>
## # ... with 7 more variables: qflag_tmax <chr>, qflag_tmin <chr>,
## #   sflag_awnd <chr>, sflag_tmax <chr>, sflag_tmin <chr>, tmax <dbl>,
## #   tmin <dbl>
```

Then convert correct units of temperatures (convert to degrees C)

```
d_filtered=d%>%
  mutate(tmax=ifelse(qflag_tmax!=" "|tmax==-9999,NA,tmax/10))%>%
  mutate(tmin=ifelse(qflag_tmin!=" "|tmin==-9999,NA,tmin/10))%>%
  mutate(awnd=ifelse(qflag_tmin!=" "|awnd==-9999,NA,awnd))%>%
  arrange(date)
```

Then filter data in 2016 to match the 2016's AQI data in Buffalo:

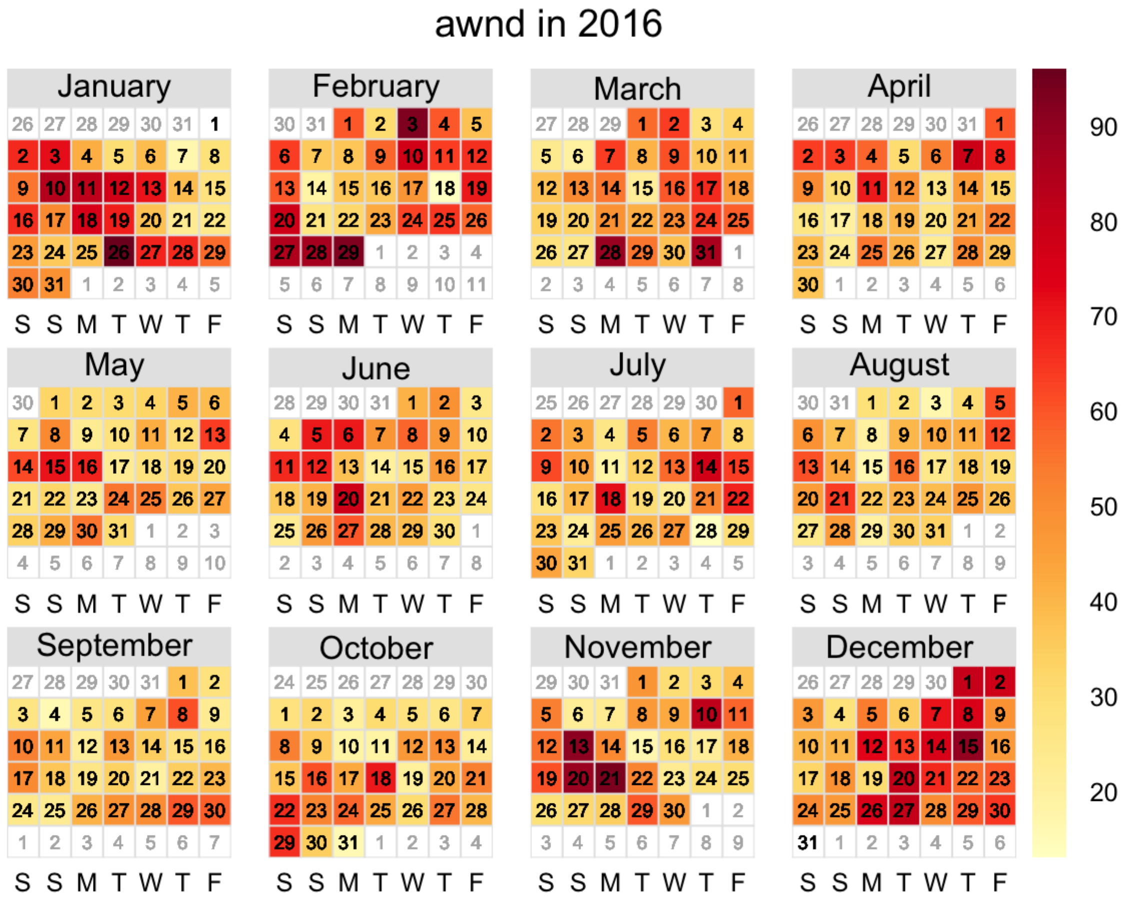
```
d_filtered_2016=filter(d_filtered,date>as.Date("2016-01-01")&date<as.Date("2016-12-31"))

head(d_filtered_2016)
```

```
## # A tibble: 6 x 14
##            id       date  awnd mflag_awnd mflag_tmax mflag_tmin qflag_awnd
##         <chr>     <date> <chr>      <chr>      <chr>      <chr>      <chr>
## 1 USW00014733 2016-01-02    67
## 2 USW00014733 2016-01-03    72
## 3 USW00014733 2016-01-04    42
## 4 USW00014733 2016-01-05    30
## 5 USW00014733 2016-01-06    39
## 6 USW00014733 2016-01-07    17
## # ... with 7 more variables: qflag_tmax <chr>, qflag_tmin <chr>,
## #   sflag_awnd <chr>, sflag_tmax <chr>, sflag_tmin <chr>, tmax <dbl>,
## #   tmin <dbl>
```

Plots the daily average wind speed data:

```
calendarPlot(d_filtered_2016, pollutant = "awnd", year=2016)
```

```
## Warning in bind_rows_(x, .id): Unequal factor levels: coercing to character
## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector

## Warning in bind_rows_(x, .id): binding character and factor vector,
## coercing into character vector
```



Results

Show tables, plots, etc. and describe them.

Conclusions

What have you learned? Are there any broader implications?

References

[1]http://margintale.blogspot.com/2012/04/ggplot2-time-series-heatmaps.html (http://margintale.blogspot.com/2012/04/ggplot2-time-series-heatmaps.html)