Unit4: For Live Session

DS6306
Garrity

PART 1A

Number of Sushi restaurants in the Baltimore area.

```
> df_sushi <- data.frame(name = sushi_name, zipcode = sushi_zipcode, council=sushi_council)
> str(df_sushi)

'data.frame':     1327 obs. of  3 variables:
 $ name   : Factor w/ 1277 levels "#1 CHINESE KITCHEN",..: 9 3 992 1 2 4 5 6 7 8 ...
 $ zipcode: Factor w/ 32 levels "-21226","21201",..: 5 27 21 10 20 17 4 10 4 27 ...
 $ council: Factor w/ 14 levels "1","10","11",..: 7 1 1 6 14 6 5 12 5 1 …

> grep("sushi", ignore.case = TRUE, df_sushi$name)
[1]   17   90  249  250  391  457  537  725 1137
> length(grep("sushi", ignore.case = TRUE, df_sushi$name))
[1] 9
```

There appears to be 9 sushi restaurants in the Baltimore area.

PART 1B

Number of Sushi restaurants in downtown Baltimore.

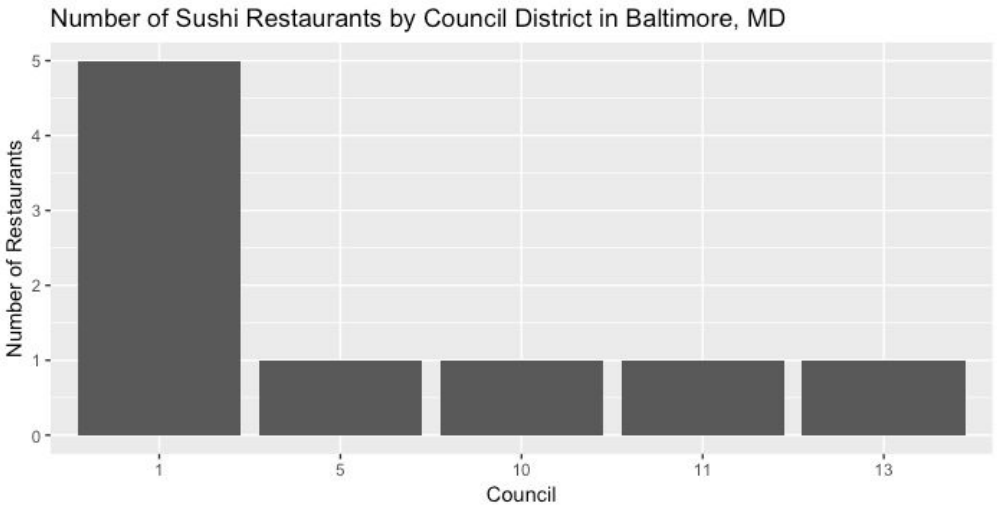
```
> df_sushi_11 <- df_sushi %>% filter(council == 11)

> length(grep("sushi", ignore.case = TRUE, df_sushi_11$name))
[1] 1

> df_sushi_11[grep("sushi", ignore.case = TRUE, df_sushi_11$name),]
        name zipcode council
81 EDO SUSHI   21202      11
```

There is one sushi restaurant (Edo Sushi) in downtown Baltimore.

PART 1C

Plot of the number of sushi restaurants in each council district.

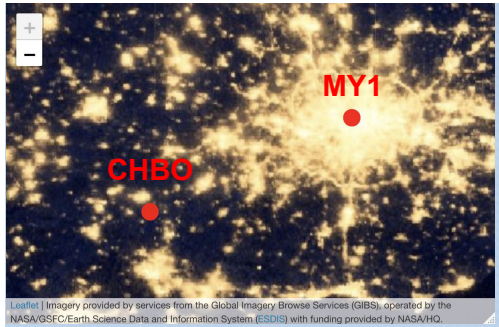
```
sushi_by_council_summarize <- sushi_by_council %>% group_by(council) %>% summarize(count = n())

sushi_by_council_summarize$council =
  factor(sushi_by_council_summarize$council,
         level = sushi_by_council_summarize$council[order(as.numeric(as.character(sushi_by_council_summarize$council)))])

sushi_by_council_summarize %>% ggplot(aes(x = council, y = count)) + geom_col() +
  labs(title="Number of Sushi Restaurants by Council District in Baltimore, MD", y="Number of Restaurants", x ="Council")
```



OpenAir is an API providing access to current and historic air quality observations from 244 locations across the UK. It is extremely easy to us and comes with several helpful functions. I used the API to compare one year (2018) of air quality observations from an urban location in London (MY1) to a rural area to the west of London (CHBO). http://davidcarslaw.github.io/openair/index.html

```
library(openair)

#load metadata for all sites:

aurn_detailed <- importMeta(source =
"aurn", all = TRUE)

#plot site locations:

library(leaflet)

m2 <- leaflet() %>%
  addProviderTiles(providers$Wikimedia)
%>%
addCircleMarkers(lng=aurn_detailed$longit
ude, lat=aurn_detailed$latitude, radius =
6, color = "navy", stroke = FALSE,
fillOpacity = 0.2, popup =
aurn_detailed$code) %>%
addCircleMarkers(lng=aurn_detailed$longit
ude[aurn_detailed$code == "CHBO" |
aurn_detailed$code == "MY1"],
lat=aurn_detailed$latitude[aurn_detailed$
code == "CHBO" |aurn_detailed$code ==
"MY1"],    radius = 6, color = "red",
stroke = FALSE, fillOpacity = 0.5, popup
= aurn_detailed$code[aurn_detailed$code
== "CHBO" | aurn_detailed$code == "MY1"])

#display map:

m2
```

Air quality observation stations across the UK



London area at night

OPENAIR

Data wrangling

```
# list all air quality parameters
str(aurn_detailed$parameter)

# list all urban sites
urban_background_sites <- filter(aurn_detailed, site.type == "Urban Background")
nrow(urban_background_sites)

# list all rural sites
rural_background_sites <- filter(aurn_detailed, site.type == "Rural Background")
nrow(rural_background_sites)

# download data for CHBO (rural) and MY1 (urban)
chbo <- importAURN(site= "CHBO", year = 2018)
str(chbo)
my1 <- importAURN(site = "MY1", year = 2018)
str(my1)

# combine data frames, filling empty columns with NAs
require(plyr)
alldata = rbind.fill(my1,chbo)

# create a factor with four levels from the continuous wind direction variable
alldata$windCardinal = cut(alldata$wd, breaks = c(0,90,180,270,360), labels = c("NE","SE","SW","NW"))
```

OPENAIR

Data wrangling (tidy data)

```
# combine data frames, filling empty columns with NAs

require(plyr)

alldata = rbind.fill(my1,chbo)
```

```
> head(alldata)
                 date code                    site       o3       no2       co     so2 pm10      nox        no pm2.5 nv10  v10 nv2.5 v2.5  ws
wd
1 2018-01-01 00:00:00  MY1 London Marylebone Road 24.48059  79.05960 0.298901 4.92942 24.6 187.4120  70.66562  12.7 25.2 -0.6  14.5 -1.8 5.5
263.3
2 2018-01-01 01:00:00  MY1 London Marylebone Road 13.96990  97.05587 0.261539 6.75514 18.7 310.9482 139.49697  11.6 17.8  0.9  11.4  0.2 5.0
256.4
3 2018-01-01 02:00:00  MY1 London Marylebone Road 18.21076 101.07355 0.308242 6.43564 17.2 273.9910 112.77380   7.6 15.2  2.0   6.1  1.5 4.8
251.0
4 2018-01-01 03:00:00  MY1 London Marylebone Road 24.84647  81.56557 0.224176 5.54560 10.8 211.2405  84.57176   2.3 11.0 -0.2   4.3 -2.0 4.8
246.8
5 2018-01-01 04:00:00  MY1 London Marylebone Road 29.03744  59.85638 0.252198 5.06635 12.4 175.8730  75.66403   4.8 10.7  1.7   3.6  1.2 5.3
248.4
6 2018-01-01 05:00:00  MY1 London Marylebone Road 35.62325  51.55781 0.112088 3.90246 12.6 134.8788  54.34053   2.7 10.9  1.7   2.0  0.7 5.3
248.0
```

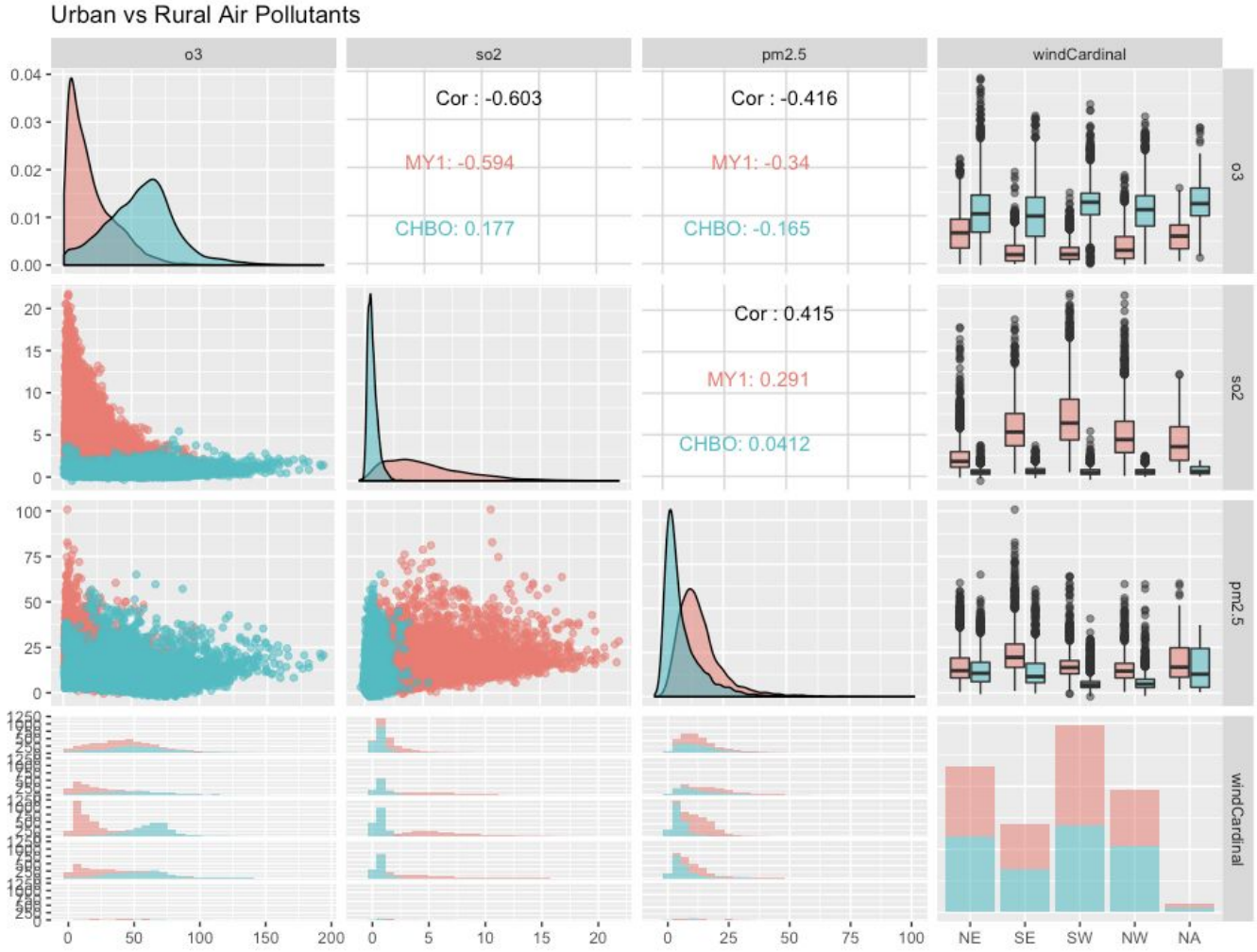
```
> tail(alldata)
                    date code                     site       o3     no2 co     so2   pm10     nox      no pm2.5 nv10 v10 nv2.5 v2.5  ws    wd
17515 2018-12-31 18:00:00 CHBO Chilbolton Observatory 76.38542 1.85279 NA 0.33295 11.125 2.42172 0.37105 6.132   NA  NA    NA   NA 4.8 284.2
17516 2018-12-31 19:00:00 CHBO Chilbolton Observatory 76.63488 1.64820 NA 0.46613  9.875 2.37230 0.47225 5.590   NA  NA    NA   NA 4.3 281.5
17517 2018-12-31 20:00:00 CHBO Chilbolton Observatory 75.18800 1.64820 NA 0.46613  9.150 2.37230 0.47225 5.590   NA  NA    NA   NA 4.1 279.1
17518 2018-12-31 21:00:00 CHBO Chilbolton Observatory 69.74972 3.33778 NA 0.46613 11.775 3.85499 0.33732 8.326   NA  NA    NA   NA 4.5 289.5
17519 2018-12-31 22:00:00 CHBO Chilbolton Observatory 71.14671 3.19181 NA 0.19977  9.700 3.65730 0.30359 6.344   NA  NA    NA   NA 4.8 296.3
17520 2018-12-31 23:00:00 CHBO Chilbolton Observatory 69.15101 2.94929 NA 0.39955  9.075 3.31134 0.23612 6.179   NA  NA    NA   NA 4.9 303.2
```

OPENAIR

openAir EDA

```
require(GGally)

alldata %>% select(o3, so2,
pm2.5, windCardinal) %>%
ggpairs(aes(color=alldata$code,
alpha=0.2)) + ggtitle("Urban vs
Rural Air Pollutants")
```

There appears to be some fairly stark differences in O_3, SO_2, and $PM_{2.5}$ between the urban (MY1) and rural (CHBO) sites. It also appears that concentrations of each pollutant vary somewhat with wind direction.



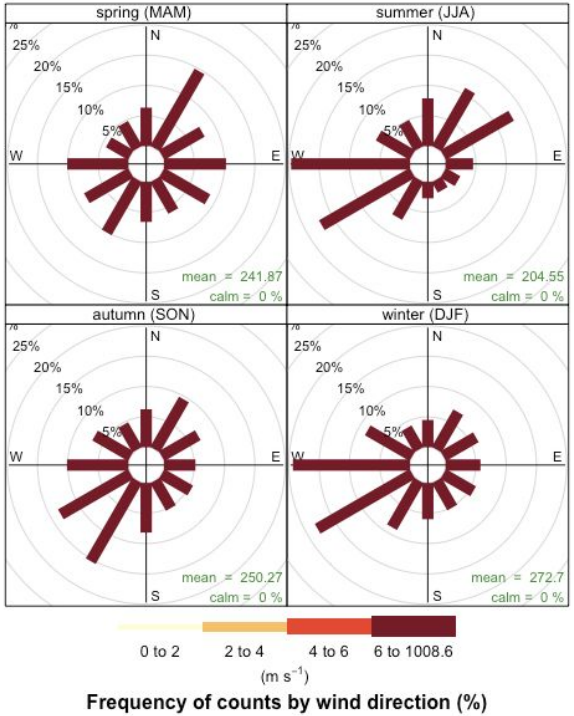
OPENAIR

openAir windrose plots of NO_x concentration by season

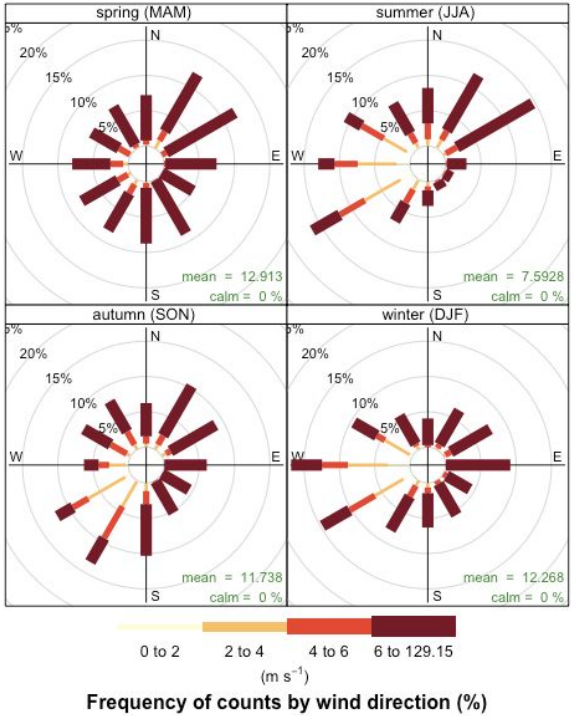
```
# plot seasonal windrose
windRose(my1, type =
"season", pollutant = "nox",
cols="heat", width=0.5,
offset=15)

windRose(chbo, type =
"season", pollutant = "nox",
cols="heat", width=0.5,
offset=15)
```

The windrose shows both frequency of wind direction at each site, as well as NO_x concentrations associated with each wind direction. Seasonal differences in wind direction are obvious for both sites, but only CHBO exhibits directional dependency for NO_x concentration, which varies somewhat by season.

MY1 (urban)



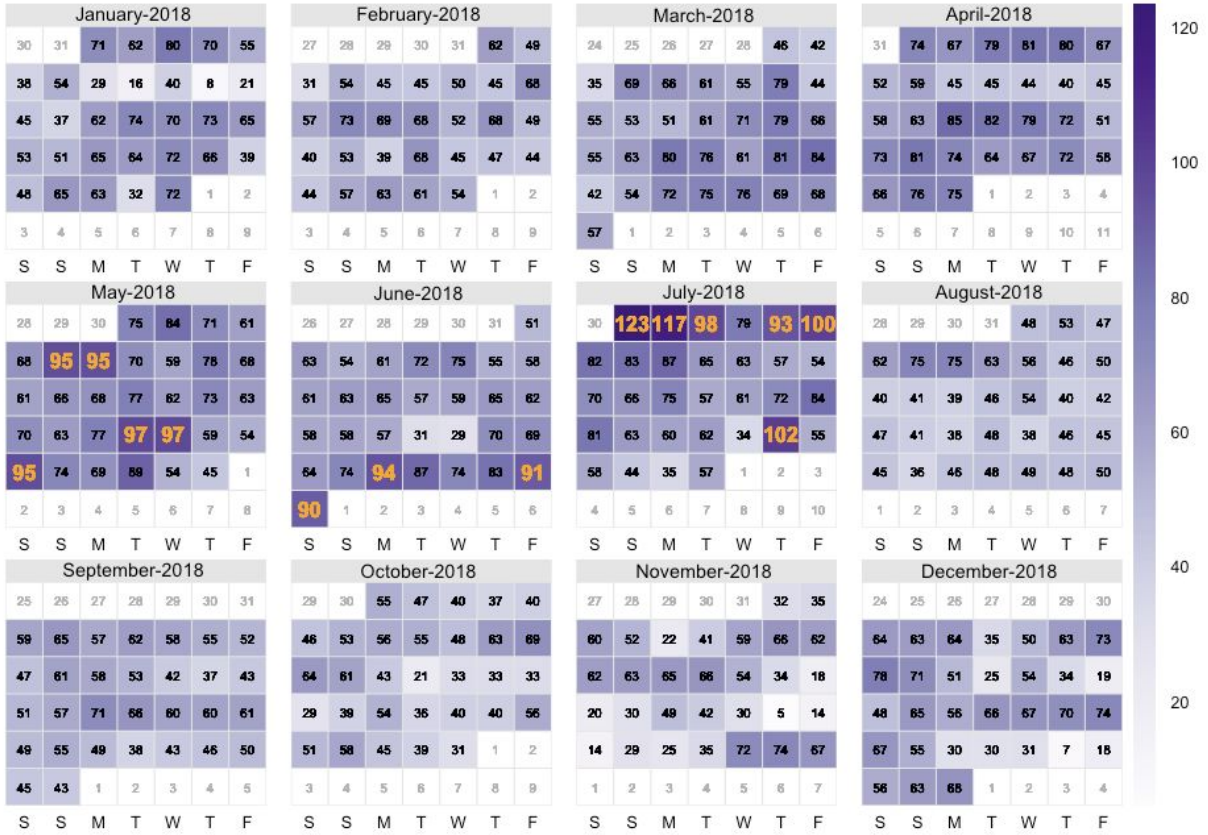
CHBO (rural)



OPENAIR

Too cool not to include!

```
calendarPlot(chbo, pollutant =
"o3", annotate = "value", lim =90,
cols = "Purples", col.lim =
c("black", "orange"),
layout = c(4, 3))
```



Takeaways & Questions

Why didn't the following code work?

```
> df_sushi %>% filter(council == 11) %>% grep("sushi", ignore.case = TRUE, name)
Error in grep(., "sushi", ignore.case = TRUE, name) :
  object 'name' not found
```

I had trouble with the *twitteR* package but found that *rtweet* worked fine and has most of the same functionality as *twitteR*.

Access to data has been a limiting factor for me in the past. The ability to scrape data and use APIs opens up a whole new world of information to ingest and process. This is definitely one of the tools I was hoping to add to the toolbelt through this program. Exciting!