James Li

11.007

Prof. David Hsu

November 30th, 2018

Lab Report 2

Introduction

The purpose of this lab report is to use data scripts in R to clean, display, and compare data from different air quality sensors. I deployed my two sensors ("Norbert," my first sensor, and "Felicia," a standardized sensor built by the class) to a streetpost on Vassar Street, across from the new dorm construction site, from 8:00 to 22:00 UTC November 28th. This location is pictured in Figure 1. I have chosen to compare the data from these two co-located sensors as well as the other class sensors that were active outdoors during the same time: "Jin" (deployed by Austin Floyd) and "DahliaDEQ" (deployed by Nelson Lee). Jin took measurements from Austin's north-facing windowsill on the 6th floor of Simmons Hall, while DahliaDEQ took measurements at the MassDEP air quality monitoring site in Roxbury.

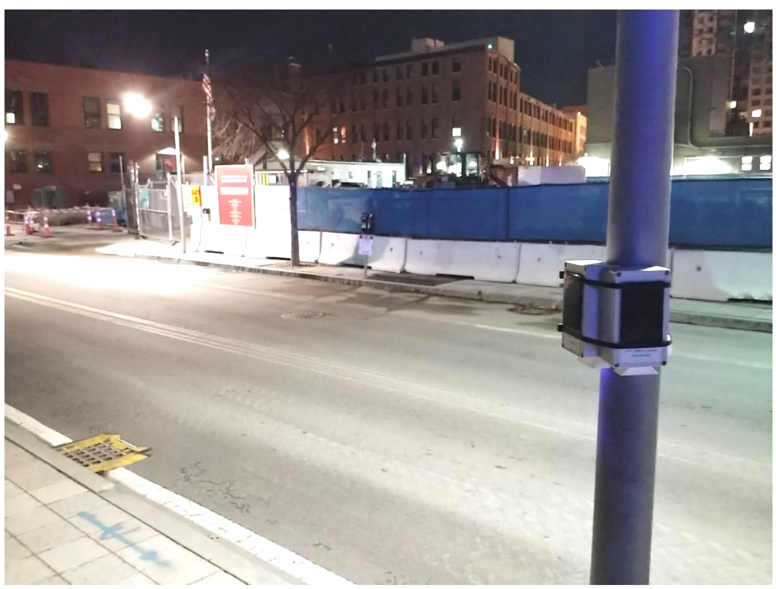

Figure 1: Felicia and Norbert co-located on Vassar Street.

Temperature

Because all four sensors were located outdoors, I expected the temperature data to reflect similar weather conditions that day across the region. While this was generally true, there were several interesting discrepancies in the data. Figure 2 shows the temperature data for all four sensors between 7:00 and midnight UTC November 28th.

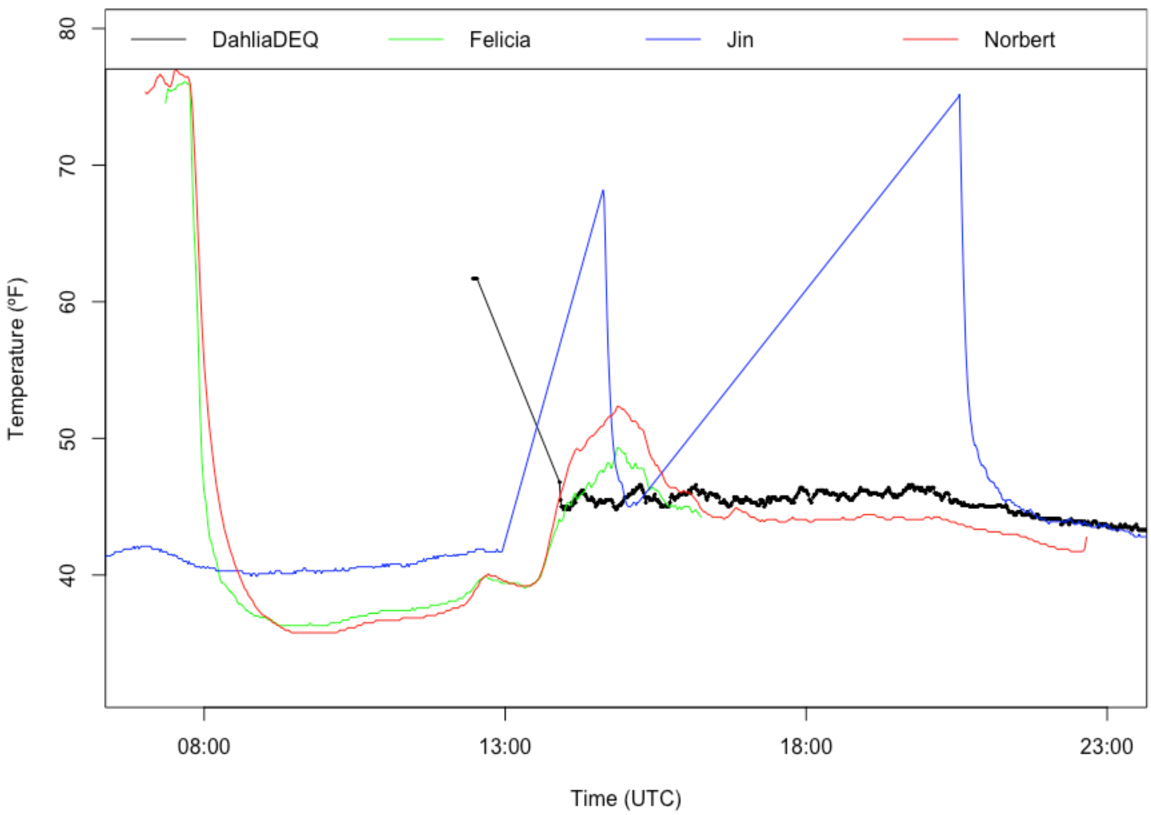

Figure 2: Temperature data over time for all four sensors.

Felicia and Norbert both began reporting data around 07:30; they were located in the warm temperature of my room, and then were moved outside into the cold night air around 08:00 (3:00 AM local time). This can be seen in the steep drop in their temperature measurements around 08:00. Felicia reported data until around 16:00, when an unknown problem caused the sensor to stop working. Norbert continued reporting data until retrieved shortly before 23:00. Jin was already active when Felicia and Norbert were activated, and continued to report data until 13:00. Jin was then restarted two times later in the day; each time, the sensor was restarted indoors and then placed outdoors, as evident in the two temperature spikes in the data. DahliaDEQ was placed at the MassDEP site around 14:00, and continued reporting data through the day. The few initial data points at around 62ºF in the DahliaDEQ data are likely due to the sensor being briefly turned on for testing before being deployed.

One notable discrepancy is visible in the data from Felicia and Norbert, the two co-located sensors, when they were initially moved into the colder outdoor air. Felicia's measurements declined more quickly than Norbert's measurements, but Norbert's measurements eventually reach lower values and remain around 3ºF lower, even as the two datasets follow similar trends. One possible explanation for this discrepancy is the difference in internal design between the two sensors. The DHT22 sensor in Felicia is surrounded by air on all sides, while the DHT22 sensor in Norbert is mounted in a plastic internal wall; maybe the extra thermal mass of the internal wall resulted in a slower temperature response for Norbert. The difference in steady-state measurements may also be due to differences in internal air circulation (could the Arduino have been generating heat?), or simply inconsistency between the DHT22 sensors.

The nighttime temperature measurements from Jin are around 10ºF higher than those from Felicia and Norbert. Because Jin was located on a windowsill, I speculate that heat from the warm interior of the building caused the air temperature at that location to be warmer.

During the daytime, temperature measurements from Felicia and Norbert increased significantly, peaking around 15:00 UTC before decreasing to approximately 44ºF around 17:00 UTC. This is in contrast with the data from DahliaDEQ and Jin, which are in close agreement and steady at approximately 46ºF throughout the day (with small fluctuations). Furthermore, Norbert reported a peak temperature around 3-4ºF higher than Felicia. One possible explanation for this temperature peak in the two co-located sensors is that radiant energy from direct sunlight caused the sensors' enclosures to heat up, thus warming the air inside the sensors, while Jin and DahliaDEQ did not experience this due to being in shaded locations. The higher temperatures reported by Norbert may be due to differences in internal design and air circulation, or difference in mounting orientation (i.e. maybe Norbert received more sunlight). Further testing would be needed to better understand these factors.

Relative Humidity

Trends and discrepancies very similar to those in the temperature data can be seen in the relative humidity data, which is plotted below in Figure 3. The relative humidity data is generally inversely related to the temperature data. There is one outlier (the spike down) towards the end of Norbert's data that is unexplained and may be due to some sort of sensor malfunction or error.

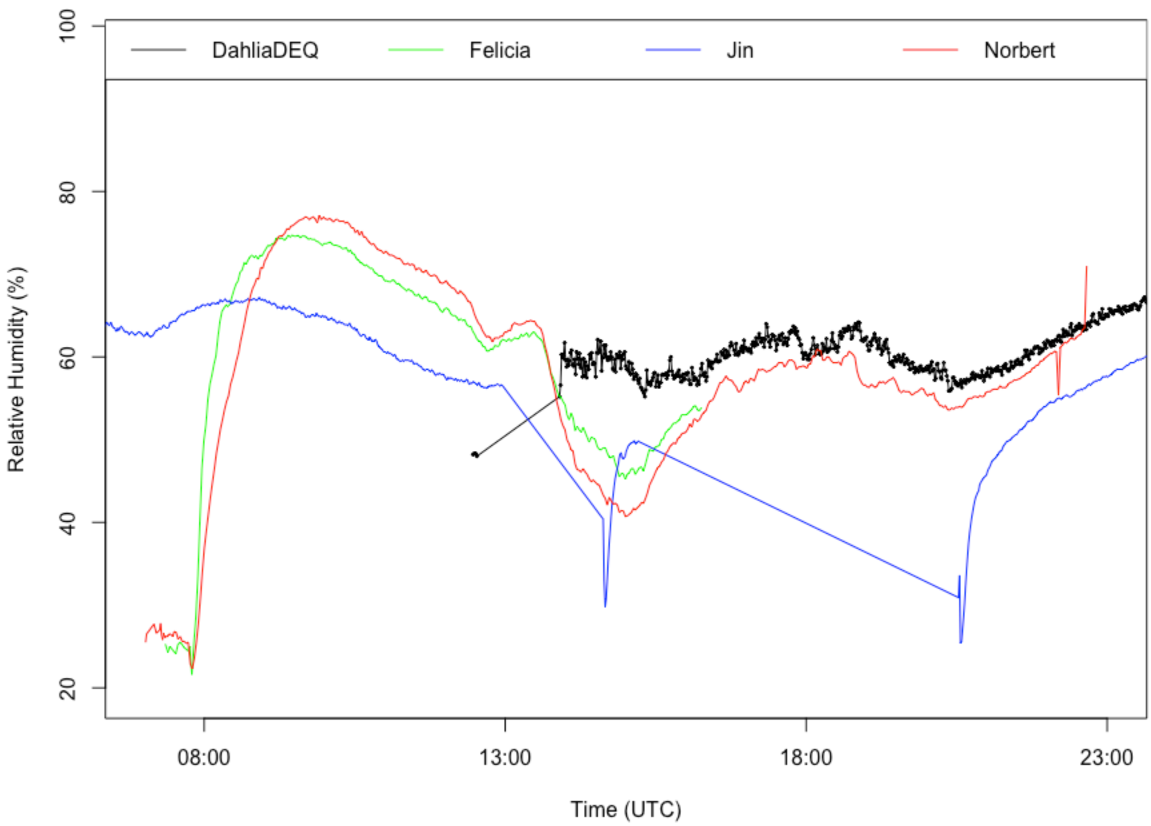

Figure 3: Relative humidity data over time for all four sensors.

Particulate Matter

Although the particulate matter sensors report values for PM1, PM2.5, and PM10, for simplicity I will focus on discussing only the PM2.5 data. The PM2.5 data for all four sensors during the same time period as previously examined (07:00-midnight November 28th) is shown in Figure 4.

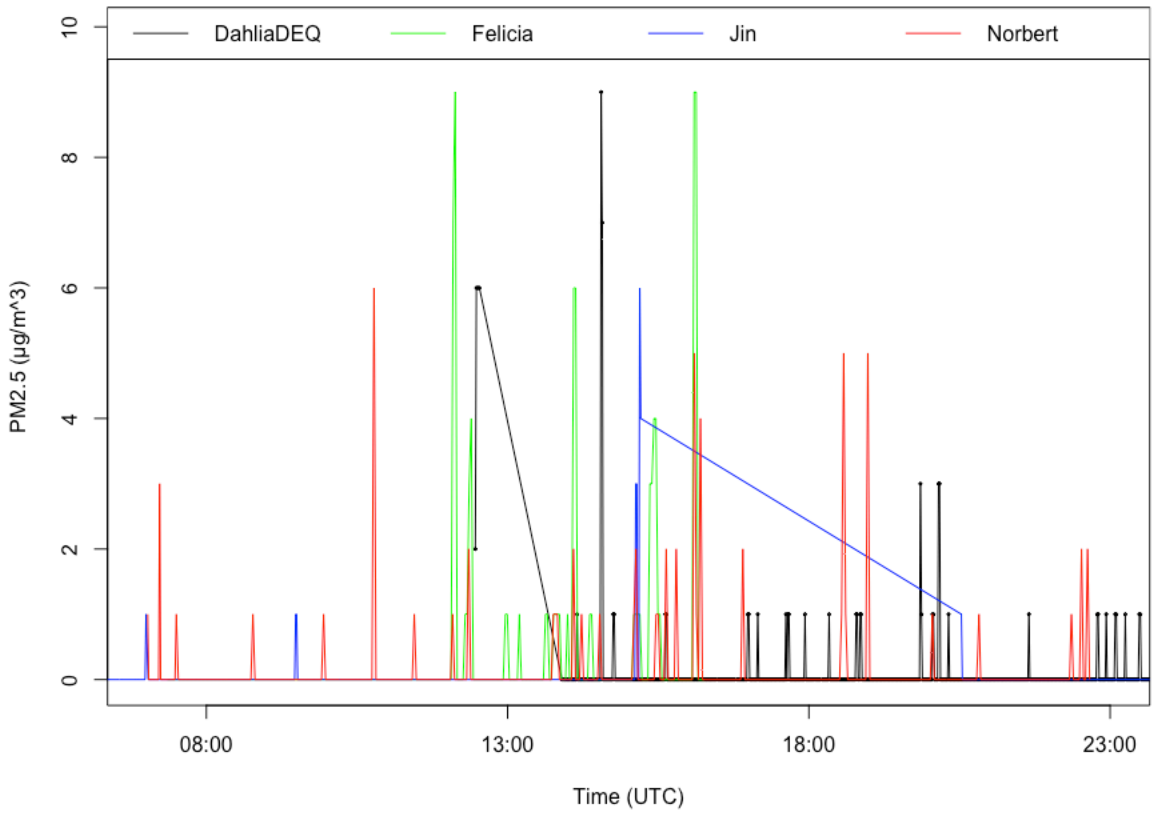

Figure 4: Raw PM2.5 data over time for all four sensors.

Because this data is fairly low in resolution and consists of numerous spikes at discrete values, I thought that it might be useful to apply a moving average to the data. Figure 5 shows a moving average of the PM2.5 data; each data point is averaged over a 20-minute time period. I felt that applying the moving average made it easier to visually identify trends in the otherwise very jumpy PM data.

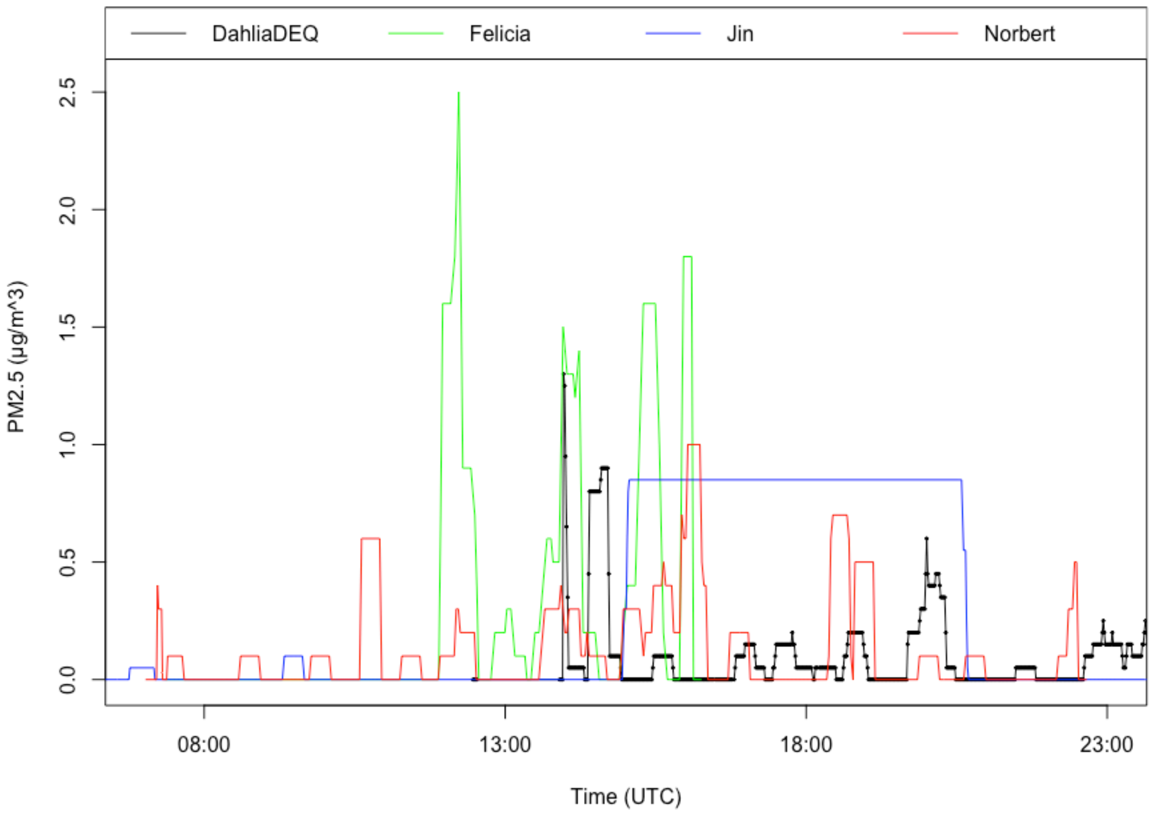

Figure 5: PM2.5 data over time for all four sensors, smoothed with a 20-minute moving average.

Across all four sensors, the overall level of particulate matter recorded is fairly similar, certainly on the same order of magnitude. The raw data show primarily readings of 0 and 1 $\mu g/m^3$, with only occasional spikes never exceeding 10 $\mu g/m^3$. Throughout this set of locations, our sensors were consistent in measuring very low amounts of PM2.5. This seems somewhat surprising, particularly for the co-located sensors measuring near the construction site and at street level – I would have expected the readings to be higher, given the construction activity and vehicle traffic. Unfortunately, without a controlled test, there is no way of testing the accuracy of the PM sensors. If the sensors are accurate, then it would appear that these locations in Boston/Cambridge have fairly low amounts of particulate matter relative to EPA standards.

(The long horizontal line in the data for Jin is due to a break in the data, not a sustained level of PM2.5.)

Because Felicia and Norbert were co-located, comparing their data offers an opportunity to evaluate how repeatable and consistent our sensors may be. Visually examining the data suggests that the agreement between the two sensors is less than great. In the early morning (08:00-12:00 UTC), Norbert repeatedly reports low levels of PM2.5 while Felicia mostly reports none. From around 12:00 UTC through 16:00 UTC, PM2.5 readings from both sensors seem to increase (possibly due to morning traffic?), but the readings from Felicia increase far more than those from Norbert. Possible explanations for this discrepancy include differences in internal airflow in the two sensors and variation between the PMS5003 units.

Future Work

Based on the results discussed in this lab report, there are several opportunities for future work to further explore the accuracy of our sensors. As seen from the Felicia and Norbert datasets, co-location reveals both similarities and discrepancies between different sensors. Co-location of larger numbers of sensors would provide a larger pool of data that could be used to compare measurement accuracy and might provide insight as to the nature of the variations between sensors. A second avenue for future work might be to characterize the accuracy of our sensor in absolute terms, e.g. by comparing it to a known measurement tool sampling the same air. Another way to better understand the differences between sensors would be to isolate the effects of the enclosure, by first comparing measurements from the DHT22 and PMS5003 units without enclosures and then repeating the same test with the sensors in their enclosures. It would also be valuable to leave sensors at street level for longer periods of time, in order to identify longer-term periodic trends and differences between sensors – for example, whether particulate matter increases due to traffic during rush hours, and if certain sensors record these trends differently.

As a result of this data analysis, I feel less confident in data provided by any single air quality sensor and would find datasets from multiple air quality sensors measuring the same source or sample to be more credible. This data analysis suggests that enclosure design may have an impact on the accuracy of readings, but further work would be needed to conclusively assess these effects.

R Script

```
# Look at data for 11.007 Lab Report 2

# clear variables, workspaces
rm(list=ls())
cat('\014')

# libraries we're going to use
library (tidyr)
library(dplyr)
library(zoo) # for moving average function

# set working directory to folder of downloaded data
setwd("~/Dropbox (MIT)/11.007/data/lab report 2")

# load downloaded data
datN <- read.csv("Norbert.csv", sep=",", header=TRUE) # Norbert (James)
datF <- read.csv("MITFelicia_James-20181129-0211.csv", sep=",", header=TRUE) # Felicia
(James)
datJ  <-  read.csv("MITJIN_Austin-20181129-0204.csv",  sep=",",  header=TRUE)  #  Jin
(Austin)
datD   <-   read.csv("MITDahliaDEQ_Nelson-20181129-0210.csv",   sep=",",   header=TRUE)
#DahliaDEQ (Nelson)

# fixtime function from class script
fixtime <- function (ts){
  ts <- gsub("(.*).(..)$","\\1\\2",ts)
  timep <- strptime(ts, "%Y-%m-%d %H:%M:%OS", tz="UTC")
  timep <- as.POSIXct( timep )
  return (timep)
}

# modified treat function from class script for IO data:
treatIO <-function (dat){
  print (head (dat))
  dat[] <- lapply(dat, as.character)
  dat$created_at <- fixtime (dat$created_at)
  dat <- dat %>% separate (value,
                           into = c("temp.text", "temp",
                                    "rh.text", "rh",
                                    "pm1.text", "pm1",
                                    "pm25.text", "pm25",
                                    "pm10.text", "pm10",
                                    "lat.text","lat",
                                    "long.text", "long"),
                           sep=",")

  dat <- dat [, -grep("text", colnames(dat))]
  if  (length(grep("X.lat",  colnames(dat)))  !=  0)  {dat  <-  dat  [,  -grep("X.lat",
colnames(dat))]}

  select.cols <- c("temp", "rh", "pm1", "pm25", "pm10", "lat", "long")
  for (i in 1:length(select.cols)){
```

```
    dat [, select.cols[i]] <- as.numeric (dat [, select.cols[i]])
  }

  # remove the columns that aren't needed for this analysis: id, lat, long, feed_id,
lon, ele
  dat <- subset(dat, select = -c(id, lat, long, feed_id, lon, ele))

  # reorder columns so created_at is first
  dat <- dat[c(6,1,2,3,4,5)]

  print (head (dat))
  print (str (dat))

  return (dat)
}

# treat function for ThingSpeak data:
treatThingSpeak <-function (dat){
  # fix times
  dat$created_at <- fixtime (dat$created_at)
  # remove the entry_id column
  dat <- subset(dat, select = -entry_id)
  # rename columns
  names(dat)[names(dat)=="field1"] <- "temp"
  names(dat)[names(dat)=="field2"] <- "rh"
  names(dat)[names(dat)=="field3"] <- "pm1"
  names(dat)[names(dat)=="field4"] <- "pm25"
  names(dat)[names(dat)=="field5"] <- "pm10"

  return(dat)
}

# treat the data
datD <- treatIO(datD)
datF <- treatIO(datF)
datJ <- treatIO(datJ)
datN <- treatThingSpeak(datN)

# add sensor_id column
datD$sensor_id <- "D"
datF$sensor_id <- "F"
datJ$sensor_id <- "J"
datN$sensor_id <- "N"

# create moving averages for PM data
datD$pm1ma <- rollmean(datD$pm1, 20, fill=c(0,0,0))
datD$pm25ma <- rollmean(datD$pm25, 20, fill=c(0,0,0))
datD$pm10ma <- rollmean(datD$pm10, 20, fill=c(0,0,0))

datF$pm1ma <- rollmean(datF$pm1, 10, fill=c(0,0,0)) # n = half, because read interval
was double
datF$pm25ma <- rollmean(datF$pm25, 10, fill=c(0,0,0))
datF$pm10ma <- rollmean(datF$pm10, 10, fill=c(0,0,0))

datJ$pm1ma <- rollmean(datJ$pm1, 20, fill=c(0,0,0))
```

```
datJ$pm25ma <- rollmean(datJ$pm25, 20, fill=c(0,0,0))
datJ$pm10ma <- rollmean(datJ$pm10, 20, fill=c(0,0,0))

datN$pm1ma <- rollmean(datN$pm1, 10, fill=c(0,0,0))
datN$pm25ma <- rollmean(datN$pm25, 10, fill=c(0,0,0))
datN$pm10ma <- rollmean(datN$pm10, 10, fill=c(0,0,0))

# combine into large dataset
dat <- rbind(datD, datF, datJ, datN)

# list of things we might want to flex
sensor_list <- c("D", "F", "J", "N")
sensor_names <- c("DahliaDEQ", "Felicia", "Jin", "Norbert")
select.cols <- c("temp", "rh", "pm1", "pm25", "pm10", "pm1ma", "pm25ma", "pm10ma")

# list of colors
sensor_colors  <-  c("black",  "green",  "blue",  "red",  "coral",  "cyan",  "gold",
"cadetblue", "orange")


# plot temp for all sensors
# =======================================

# create indices
i <- 1
j <- 1

# limits of entire dataset
xlims <- c(min(dat$created_at), max (dat$created_at))
ylims <- c(min(dat[, select.cols[j]])*0.9, max(dat[, select.cols[j]])*1)
xlims2 <- c("2018-11-28 07:00:00", "2018-11-28 23:00:59")
xlims2 <- strptime(xlims2, format="%Y-%m-%d %H:%M:%OS", tz="UTC")
xlims2 <- as.POSIXct( xlims2 )

# plot for first sensor
small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
plot ( small.dat$created_at, small.dat[, select.cols[j]], cex = 0.25,
       ylab = "Temperature (ºF)", xlab = "Time (UTC)", main = select.cols[j],
       xlim = xlims2, ylim = ylims)
lines (small.dat$created_at, small.dat[, select.cols[j]] )

# add all the other sensors
for (i in 2:length(sensor_list)){
  small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
  lines (small.dat$created_at, small.dat[, select.cols[j]], col = sensor_colors[i])
}
legend ("top", sensor_names, col = sensor_colors, lty=1, cex = 1, horiz= TRUE)


# plot humidity for all sensors
# =======================================

# create indices
i <- 1
j <- 2
```

```
# limits of entire dataset
xlims <- c(min(dat$created_at), max (dat$created_at))
ylims <- c(min(dat[, select.cols[j]])*0.9, max(dat[, select.cols[j]])*1)
xlims2 <- c("2018-11-28 07:00:00", "2018-11-28 23:00:59")
xlims2 <- strptime(xlims2, format="%Y-%m-%d %H:%M:%OS", tz="UTC")
xlims2 <- as.POSIXct( xlims2 )

# plot for first sensor
small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
plot ( small.dat$created_at, small.dat[, select.cols[j]], cex = 0.25,
       ylab = "Relative Humidity (%)", xlab = "Time (UTC)", main = select.cols[j],
       xlim = xlims2, ylim = ylims)
lines (small.dat$created_at, small.dat[, select.cols[j]] )

# add all the other sensors
for (i in 2:length(sensor_list)){
  small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
  lines (small.dat$created_at, small.dat[, select.cols[j]], col = sensor_colors[i])
}
legend ("top", sensor_names, col = sensor_colors, lty=1, cex = 1, horiz= TRUE)


# plot pm2.5 for all sensors
# =======================================

# create indices
i <- 1
j <- 4

# limits of entire dataset
xlims <- c(min(dat$created_at), max (dat$created_at))
ylims <- c(min(dat[, select.cols[j]])*1, max(dat[, select.cols[j]])*1.1)
xlims2 <- c("2018-11-28 07:00:00", "2018-11-28 23:00:59")
xlims2 <- strptime(xlims2, format="%Y-%m-%d %H:%M:%OS", tz="UTC")
xlims2 <- as.POSIXct( xlims2 )

# plot for first sensor
small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
plot ( small.dat$created_at, small.dat[, select.cols[j]], cex = 0.25,
       ylab = "PM2.5 (µg/m^3)", xlab = "Time (UTC)", main = select.cols[j],
       xlim = xlims2, ylim = ylims)
lines (small.dat$created_at, small.dat[, select.cols[j]], col = sensor_colors[i] )

# add all the other sensors
for (i in 2:length(sensor_list)){
  small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
  lines (small.dat$created_at, small.dat[, select.cols[j]], col = sensor_colors[i])
}
legend ("top", sensor_names, col = sensor_colors, lty=1, cex = 1, horiz= TRUE)


# plot pm2.5 moving average for all sensors
# =======================================
```

```
# create indices
i <- 1
j <- 7

# limits of entire dataset
xlims <- c(min(dat$created_at), max (dat$created_at))
ylims <- c(min(dat[, select.cols[j]])*1, max(dat[, select.cols[j]])*1.1)
xlims2 <- c("2018-11-28 07:00:00", "2018-11-28 23:00:59")
xlims2 <- strptime(xlims2, format="%Y-%m-%d %H:%M:%OS", tz="UTC")
xlims2 <- as.POSIXct( xlims2 )

# plot for first sensor
small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
plot ( small.dat$created_at, small.dat[, select.cols[j]], cex = 0.25,
       ylab = "PM2.5 (µg/m^3)", xlab = "Time (UTC)", main = select.cols[j],
       xlim = xlims2, ylim = ylims)
lines (small.dat$created_at, small.dat[, select.cols[j]], col = sensor_colors[i] )

# add all the other sensors
for (i in 2:length(sensor_list)){
  small.dat <- dat[ dat$sensor_id == sensor_list[i], ]
  lines (small.dat$created_at, small.dat[, select.cols[j]], col = sensor_colors[i])
}
legend ("top", sensor_names, col = sensor_colors, lty=1, cex = 1, horiz= TRUE)


# plot different pms against each other
# =====================================

#par(mfrow=c(1,3))
#plot(dat$pm1, dat$pm25)
#plot(dat$pm25, dat$pm10)
#plot(dat$pm1, dat$pm10)
```