# lab3b

Andrew Estes

2/9/2022

*The last page has all dashboard graphs on one page*

## 3) "We're interested in the average values for Temperature, Humidity and CO2 when the office is occupied and when it isn't."

```
#creating the table for the dashboard
averages <- matrix(c(
  occ.Temp, occ.Humid, occ.CO2,
  vac.Temp, vac.Humid, vac.CO2,
  ((occ.Temp/vac.Temp)-1)*100, ((occ.Humid/vac.Humid)-1)*100, ((occ.CO2/vac.CO2)-1)*100),
  ncol=3)

colnames(averages) <- c("Occupied Averages", "Vacant Averages", "Percentage Difference")
rownames(averages) <- c("Temperature", "Humidity", "CO2")

averages <- round(averages, 1)
kable(averages)
```

| | Occupied Averages | Vacant Averages | Percentage Difference |
|---|---|---|---|
| Temperature | 21.7 | 20.3 | 6.6 |
| Humidity | 27.1 | 25.3 | 7.1 |
| CO2 | 1037.7 | 490.3 | 111.6 |

This output clearly answeres the building managers questions, although the decimal places should be decreased. It has room for improvement - mainly the formatting. Tables can be transformed into a graphical interface that works with ggplot. This is important because the next chart uses Patchwork which manipulates ggplot. There is another option to use which is plot_ly. Plot_ly has some major benefits such as being able to utilize tables, create the sparkline table, and add interactivity to the graphs. This would be something to consider in the future.

**1) "We'd like a time series plot of Temperature, Humidity and CO2 levels, which will differentiate between when the office is occupied and when it isn't occupied."**

**2) "We start to get complaints of bad air quality and low worker productivity if the CO2 level rises above 1000 ppm. The time series plot should additionally make it clearly visible when this happens."**

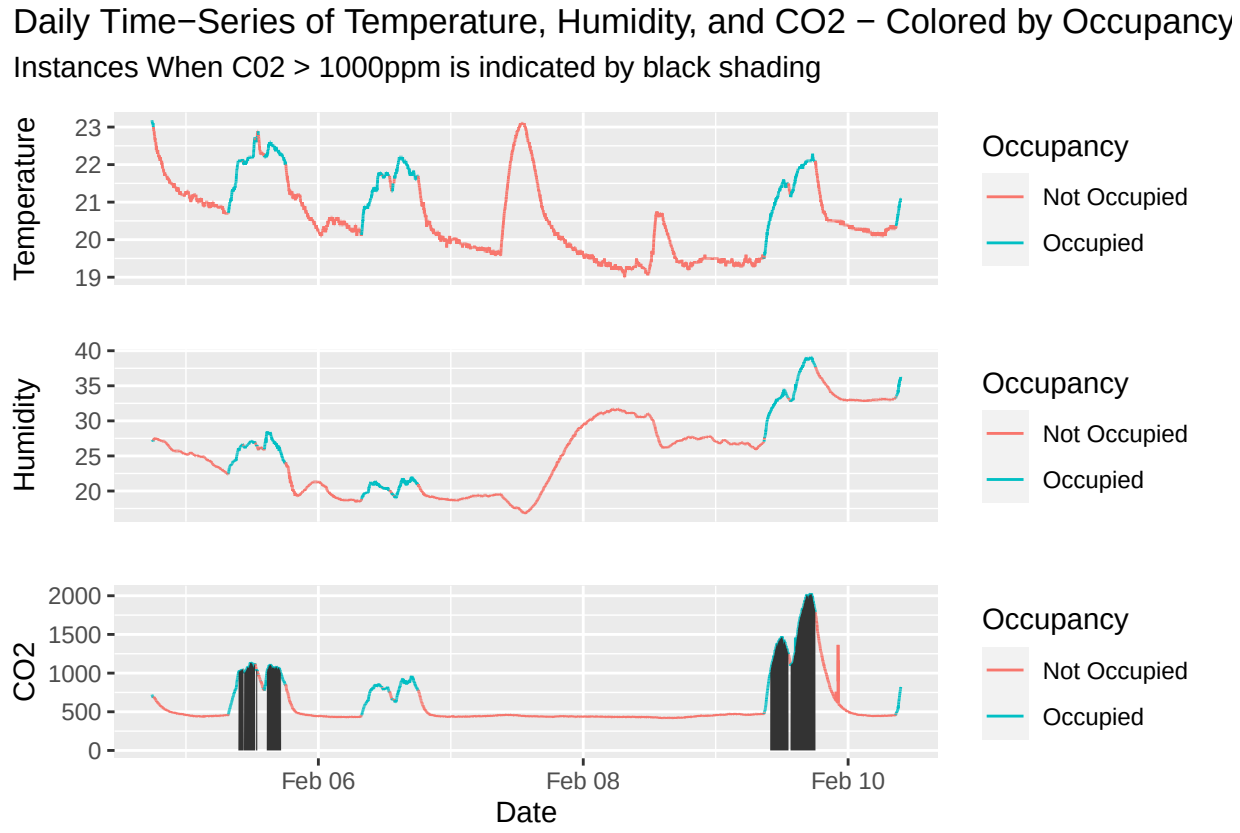
```
a <- ggplot(occ, aes(Date, Temperature)) +
  geom_line_interactive(aes(group=1, colour=Occupancy))

b <- ggplot(occ, aes(Date, Humidity)) +
  geom_line(aes(group=1,  colour=Occupancy))

c <- ggplot(occ, aes(Date, CO2)) +
  geom_line(aes(group=1,  colour=Occupancy)) +
  geom_area(mapping = aes(y = ifelse(CO2>1000 & Occupancy=="Occupied", CO2, 0)))

sparklines <- (a + guides(x="none") + xlab(" ")) / (b + guides(x="none") +xlab(" ")) / c +
  plot_annotation(
    title = "Daily Time-Series of Temperature, Humidity, and CO2 - Colored by Occupancy",
    subtitle = "Instances When CO2 > 1000ppm is indicated by black shading"
  )

sparklines
```




This graph is 80% of what I would like to do. Issues include: * Funky CO2 graph due to if-else statement used to get the shading * Poor line colors between Occupied and Not Occupied * Unnecessary grid lines in the background * Redundant legends (only keep the middle one)

In addition to fixing these issues, some potential add-ons include: * Text hovering * Adding a neat table to the right of the legends that clearly shows Average Temp/Humidity/Co2 When Occupied vs Vacant. Basically in lieu of the legends, put the data in tabular form there.

## 4) "For each day, we'd like the know the percentage of occupied time in which the CO2 level is above 1000."

```
occ.day <- occ %>%
  mutate(Day = day(occ$Date)) %>%
  filter(Occupancy == "Occupied") %>%
  group_by(Day) %>%
  summarise(
    Percentage = round((mean(CO2 > 1000) * 100), 1)
  )

kable(occ.day)
```

| Day | Percentage |
|---|---|
| 4 | 0.0 |
| 5 | 63.5 |
| 6 | 0.0 |
| 9 | 87.6 |
| 10 | 0.0 |

Further research would include the incorporation of local weather. A nearby airport has their data publicly available at https://rp5.ru/Weather_archive_in_Chievres_(airport),_METAR Unfortunately, the ability to download the date range is not functioning.

Due to the short time-frame this dataset covers, I was able to copy+paste the dataset into an excel file and clean it up utilizing VBA sufficiently enough although more work would be necessary to make it truly accessible.

```
#adding weather dataframe and cleaning up slightly
weather <- read_xlsx("belgiumWeather2.xlsx")

weather2 <- weather %>%
  mutate(Hour = hour(weather$Time)) %>%
  mutate(Minute = minute(weather$Time)) %>%
  select(-Time)
```

We need to combine the occ dataframe with the weather2 dataframe. Before we can merge/join them, both dataframes need to be on the same scale. The occ df has 1-minute intervals while weather2 has 60-minute intervals.

We do not want to average the 1-minute intervals over the course of an hour as that will reduce data (particularly the occupancy). Instead we want to put the 60-minute weather observation into the 60 corresponding occupancy observations.

For example, on minutes 1700 - 1759, each row will have the same visability, humidity ratio, mean wind speed, etc values

```
occ.join <- occ %>%
  mutate(Month = month(occ$Date)) %>%
  mutate(Day = day(occ$Date)) %>%
  mutate(Hour = hour(occ$Date)) %>%
  mutate(Minute = minute(occ$Date)) %>%
  group_by(Hour, Day)



join <- merge(occ.join, weather2, by = c("Day", "Hour")) %>%
  select(-c(Month.x, Month.y, Minute.x, Minute.y))


#clarifying names
names(join)[names(join) == 'Temperature.x']  <- "Office Temperature"
names(join)[names(join) == 'Temperature.y']  <- "Outside Temperature"

#removing characters from numeric vectors
join$`Mean Wind Speed 10m Above Ground`<- gsub("[()]", "", join$`Mean Wind Speed 10m Above Ground`)
join$`Mean Wind Speed 10m Above Ground` <-   str_remove(join$`Mean Wind Speed 10m Above Ground`, "m/s")

#change the Wind Type to Factor
join$`Wind Type` <- as.factor(join$`Wind Type`)
```

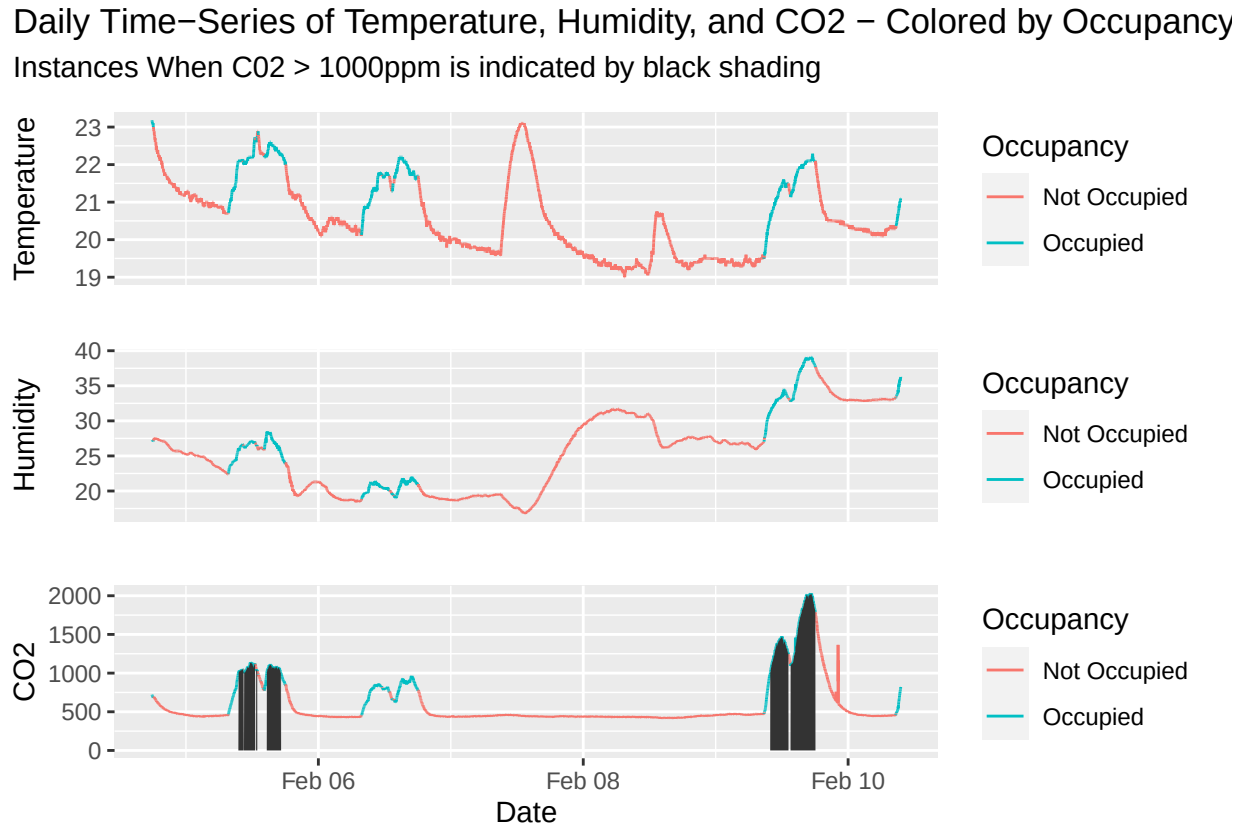
Now that the dataframes have been joined, we can run with some statistical analysis. Ideas for further exploration are commented in the code block:

```
#Predict Occupancy: Random Forest, SVM, Logistic Regression
#Predicting CO2 > 1000: CART, Multiple Linear Regression,
#Multiple Uses: Piecewise regression, Ridge Rregression, Bayesian analysis, Kernels

#Variable Selection/Dimension Reduction: Principal Components,
#Variable Importance Plots, AUC-ROC, Stepwise  Regression, Boltzmann Machine,
#Vapnik-Chervonenkis Dimension

#Future research can including testing for a correlation between the wind direction
#and horizon visability, as the direction might be indicative of an incoming storm.
#Similarly, a time-series analysis of cloud cover and horizon visability
#including lags and other variables could be interesting.

#With respect to our assignment, I wonder what the correlation is between
#several variables, such as the impact of outside humidity on the building's humidity
```




This graph shows the time-series analysis for the 3 metrics asked for: Temperature, Humidity, and CO2. Please note that the values on the left-hand side change dependent on which metric we are looking at. The most important CO2 graph is at the bottom with the shading denoting the time CO2 hit over 1000ppm when Occupied.

| | Occupied Averages | Vacant Averages | Percentage Difference |
|---|---|---|---|
| Temperature | 21.7 | 20.3 | 6.6 |
| Humidity | 27.1 | 25.3 | 7.1 |
| CO2 | 1037.7 | 490.3 | 111.6 |

This chart shows the average metrics when occupied vs vacant. There seems to be little difference with respect to Temperature and Humidity (6% and 7% less when vacant). CO2 is another story 111% more CO2 when Occupied as compared to when Vacant. This makes sense as each breath each human exhales contains 100x more CO2 than typical outside air. (https://www.energy.wsu.edu/Portals/0/Documents/Measuring_CO2_Inside_Buildings-Jan2013.pdf)

| Day | Percentage |
|---|---|
| 4 | 0.0 |
| 5 | 63.5 |
| 6 | 0.0 |
| 9 | 87.6 |
| 10 | 0.0 |

Finally, a breakdown of the frequency when CO2 > 1000ppm and the office is occupied. Without occupancy rates, it is hard to tell why Thursday and Monday were far greater with respect to CO2 issues.