

# PDAT 613G Module 6 Assignment

Jessica Lopez

Description: After an analysis of the office building's Temperature, Humidity and CO2 levels, CO2 did present an issue as the data is showing it's over 1,000ppm about 47.75% of the time over the occupied time of the three day period. The line graphs show that all three variables drop or decrease in level when the building is not occupied, and the variables rise when the building is occupied. The average values for each variable is in the bar graphs. Feb 3 has the worst average CO2 level and it was at 1,074ppm, which is above 1,000ppm.

For question 3, the average CO2 level for the three not occupied days is 608ppm and the occupied level is 966.7ppm. The average temperature for not occupied days is 21.2 Celsius. For occupied, the temperature is 22.5 Celsius.

For Humidity, the not occupied days average is 24.4RH%. The occupied days average is 26.8RH%.

To answer question four, the CO2 level above 1,000ppm for the occupied time for each day was 45% on March 2, 69% on March 3 and 29% on day March 4. Therefore the CO2 level during the occupied time over 1000ppm for the three days was about 47.75% of the time.

Load Libraries and Dataset

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'ggplot2' was built under R version 4.5.2  
  
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v forcats 1.0.0    v readr  2.1.5  
## v ggplot2  4.0.1    v stringr 1.5.1  
## v lubridate 1.9.4    v tibble  3.3.0  
## v purrr    1.1.0    v tidyr   1.3.1  
  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```

library(ggplot2)
library(lubridate)

data <- read.csv("occupancy_detection/datatest.txt")

# Ensure date is valid POSIXct in knit
data$date <- as.POSIXct(data$date, format = "%Y-%m-%d %H:%M:%S", tz = "America/New_York")

# Drop rows with invalid dates (NA/Inf)
data <- data[!is.na(data$date) & is.finite(as.numeric(data$date)), ]

```

Graph by Temperature

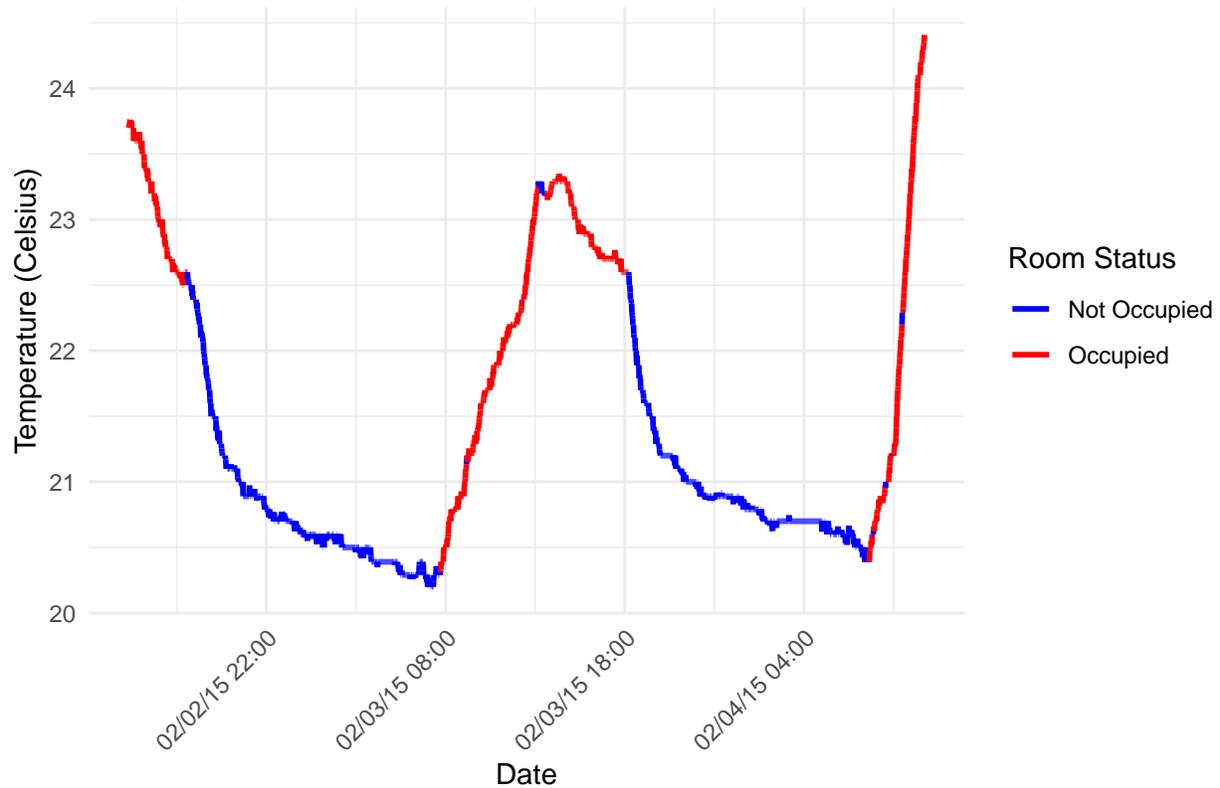
```

data$date <- as.POSIXct(data$date, format = "%D %H %M")

# Plot
ggplot(data, aes(x = date, y = Temperature)) +
  geom_line(aes(color = factor(Occupancy), group = 1),
            linewidth = 1) +
  scale_color_manual(
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "Not Occupied",
              "1" = "Occupied")
  ) +
  scale_x_datetime(
    date_breaks = "10 hour",
    date_labels = "%D %H:%M"
  ) +
  labs(title = "Temperature by Occupancy Status",
       x = "Date",
       y = "Temperature (Celsius)",
       color = "Room Status") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

```

## Temperature by Occupancy Status



## Bar Graph of Average Temperature

```
temp_avg <- data %>%
  mutate(day = as.Date(date)) %>%
  group_by(day, Occupancy) %>%
  summarise(temp_avg = mean(Temperature, na.rm = TRUE), .groups = "drop")

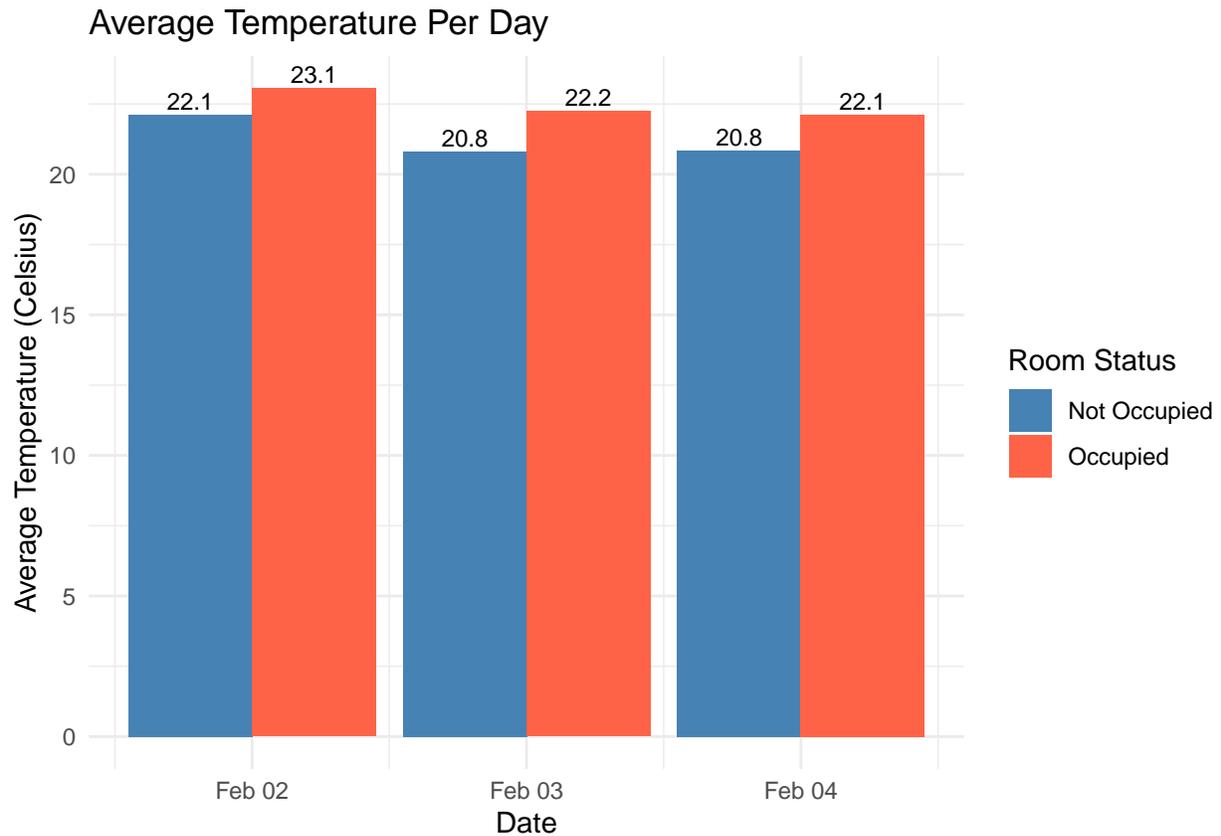
ggplot(temp_avg, aes(x = day, y = temp_avg, fill = factor(Occupancy))) +
  geom_col(position = position_dodge(width = 0.9)) +

  # value labels on bars
  geom_text(aes(label = round(temp_avg, 1)),
            position = position_dodge(width = 0.9),
            vjust = -0.3,
            size = 3) +

  scale_fill_manual(
    values = c("0" = "steelblue", "1" = "tomato"),
    labels = c("0" = "Not Occupied", "1" = "Occupied")
  ) +

  labs(title = "Average Temperature Per Day",
        x = "Date",
        y = "Average Temperature (Celsius)",
        fill = "Room Status") +
```

```
theme_minimal()
```

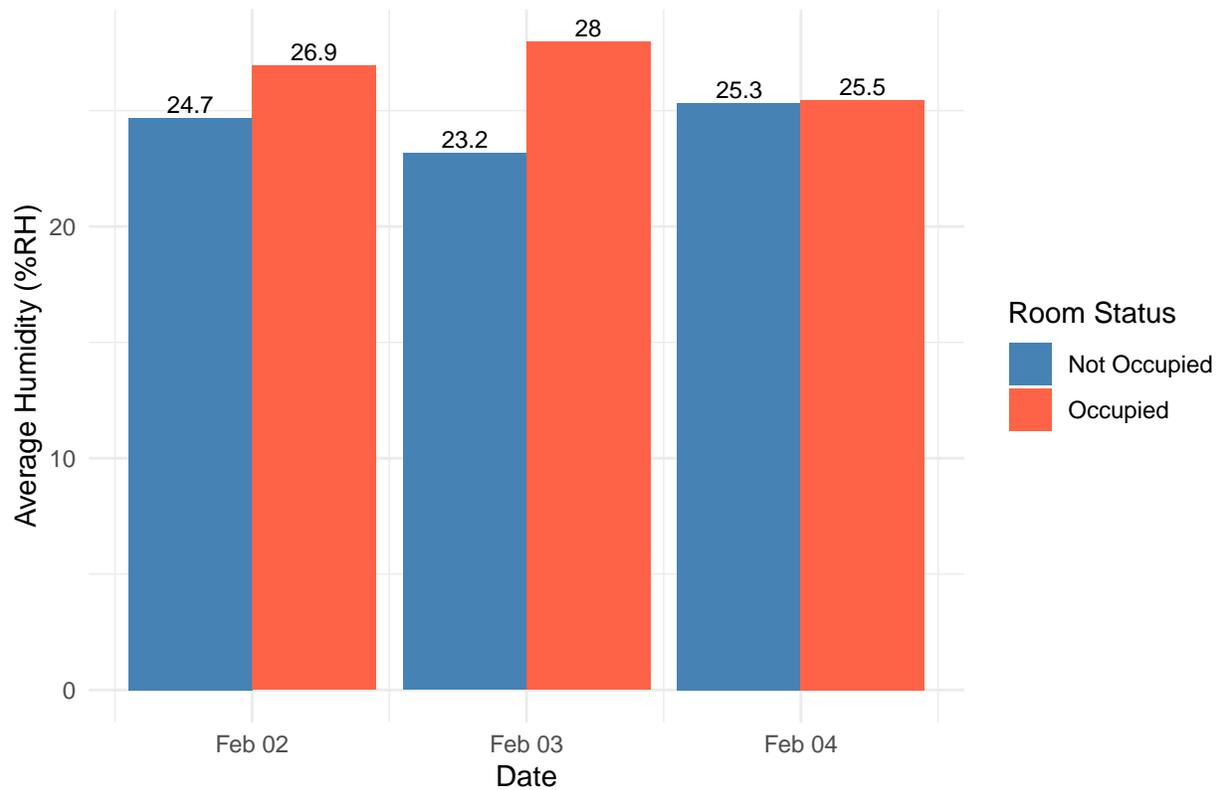


Graph of Humidity

```
# Plot
ggplot(data, aes(x = date, y = Humidity)) +
  geom_line(aes(color = factor(Occupancy), group = 1),
            linewidth = 1) +
  scale_color_manual(
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "Not Occupied",
               "1" = "Occupied")
  ) +
  scale_x_datetime(
    date_breaks = "10 hour",
    date_labels = "%D %H:%M"
  ) +
  labs(title = "Humidity by Occupancy Status",
       x = "Date",
       y = "Humidity (%RH)",
       color = "Room Status") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



Average Humidity Per Day



Graph by CO2 Level

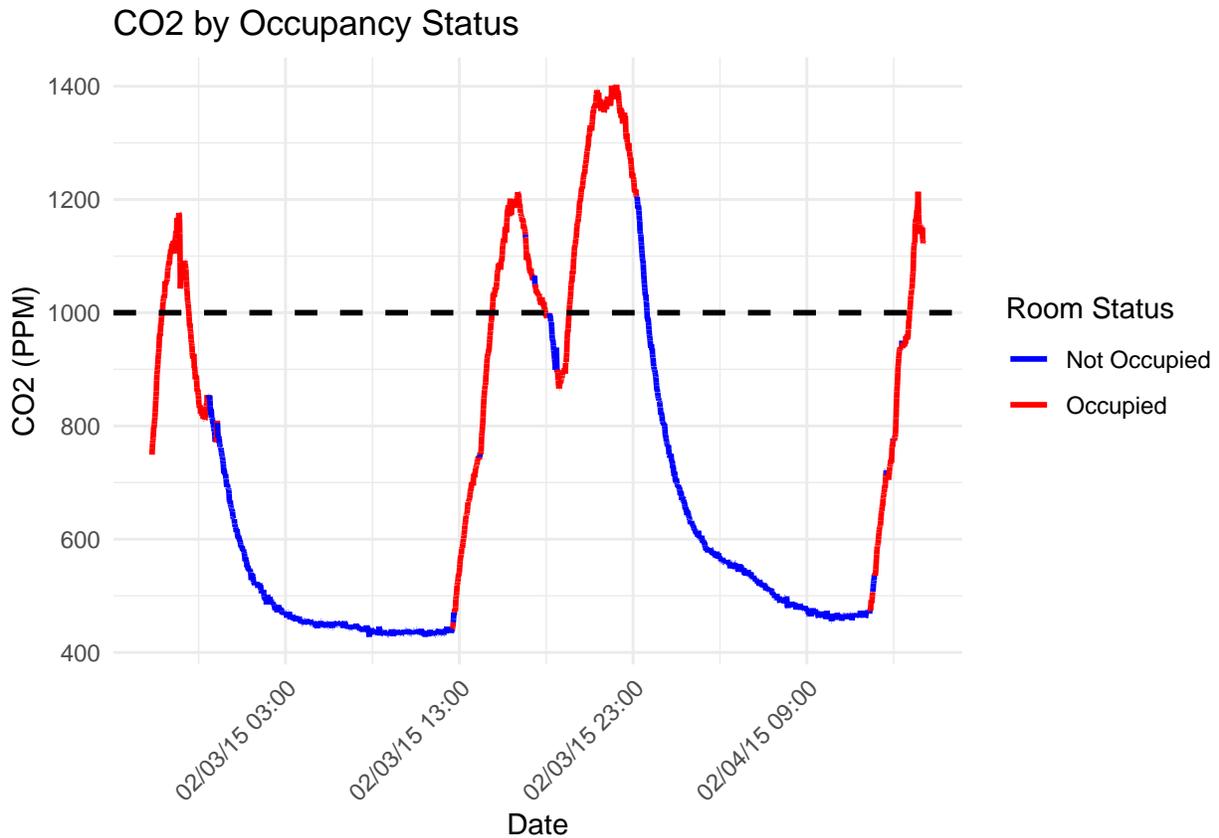
```
data$date <- as.POSIXct(data$date,
                        format = "%D %H:%M",
                        tz = "UTC")

# Plot
ggplot(data, aes(x = date, y = CO2)) +
  geom_line(aes(color = factor(Occupancy), group = 1),
            linewidth = 1) +
  geom_hline(yintercept = 1000,
             linetype = "dashed",
             color = "black",
             linewidth = 1) +
  scale_color_manual(
    values = c("0" = "blue", "1" = "red"),
    labels = c("0" = "Not Occupied",
              "1" = "Occupied")
  ) +
  scale_x_datetime(
    date_breaks = "10 hour",
    date_labels = "%D %H:%M"
  ) +
  labs(title = "CO2 by Occupancy Status",
       x = "Date",
       y = "CO2 (PPM)",
```

```

color = "Room Status") +
theme_minimal() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))

```



Bar graph of average CO2 level

```

CO2_avg <- data %>%
  mutate(day = as.Date(date)) %>%
  group_by(day, Occupancy) %>%
  summarise(CO2_avg = mean(CO2, na.rm = TRUE), .groups = "drop")

ggplot(CO2_avg, aes(x = day, y = CO2_avg, fill = factor(Occupancy))) +
  geom_col(position = position_dodge(width = 0.9)) +

  # value labels
  geom_text(aes(label = round(CO2_avg, 0)),
            position = position_dodge(width = 0.9),
            vjust = -0.3,
            size = 3) +

  scale_fill_manual(
    values = c("0" = "steelblue", "1" = "tomato"),
    labels = c("0" = "Not Occupied", "1" = "Occupied")
  ) +

  geom_hline(yintercept = 1000,

```

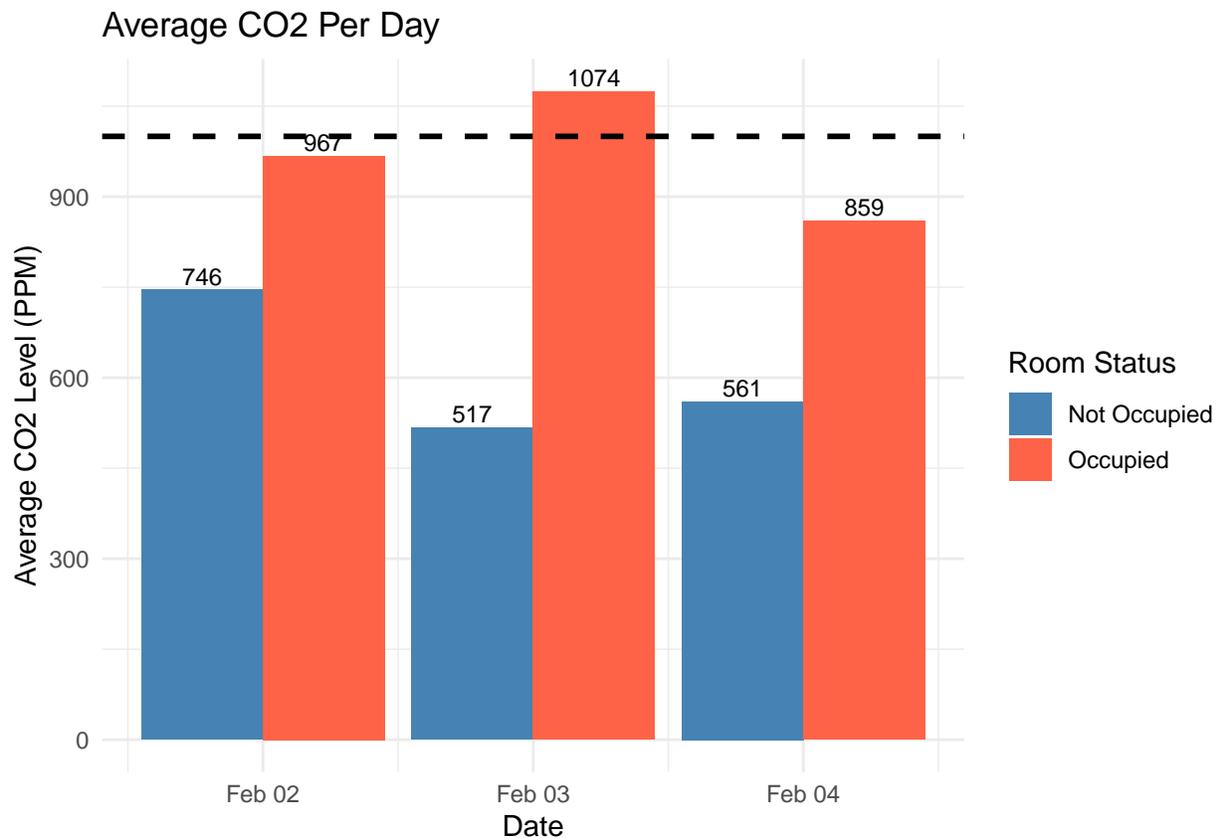
```

linetype = "dashed",
color = "black",
linewidth = 1) +

labs(title = "Average CO2 Per Day",
x = "Date",
y = "Average CO2 Level (PPM)",
fill = "Room Status") +

theme_minimal()

```



```

daily_pct_occ_above_1000 <- data %>%
  mutate(
    date = as.POSIXct(date, format = "%Y-%m-%d %H:%M:%S"),
    day = as.Date(date)
  ) %>%
  filter(Occupancy == 1) %>%
  group_by(day) %>%
  summarise(
    pct_above_1000 = 100 * mean(CO2 > 1000, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  select(day, pct_above_1000)

```

```
daily_pct_occ_above_1000
```

```
## # A tibble: 3 x 2
##   day      pct_above_1000
##   <date>      <dbl>
## 1 2015-02-02      45.3
## 2 2015-02-03      69.1
## 3 2015-02-04      28.8
```

The percentage of time that the CO2 level is above 1,000 is about 4.65%.