MECH481A6: Engineering Data Analysis in R

Chapter 4 Homework: Visualizing Ozone Data

Ethan Rutledge

26 September, 2023

Note: If you haven't installed *LaTex*, change the output mode in the above YAML to `html_document` for ease of knitting and homework submission.

This R Markdown (.Rmd) file is a template for your Chapter 4 Homework. Do everything within this file. Make it your own, but be careful not to change the code-figure-text integration I set up with the code appendix and the global options. If you have used R Markdown before and are comfortable with the extra options, feel free to customize to your heart's desire. In the end, we will grade the **knitted** PDF or HTML document from within your private GitHub repository. Remember to make regular, small commits (e.g., at least one commit per question) to save your work. We will grade the latest knit, as long as it occurs *before* the start of the class in which we advance to the next chapter. As always, reach out with questions via GitHub Issues or during office hours.

Ozone data

The corresponding data file (.csv) contains *hourly* ozone data from two sites in Fort Collins. You should already have this file in your `/data` folder.

Preparation

You completed the following steps in your Chapter 3 Homework. If correct, you should copy-paste the code into this R Markdown document; FYI, you cannot `source()` R Markdown files for use of its output in another file because they are intended to be self-contained and reproducible. Therefore, you need to copy-paste parts of your Chapter 3 Homework into this document and adjust the pathnames, if needed.

Load R packages

Import, select, and clean data

Recreate the pipe of `dplyr` functions that you used to import the data, select and rename the variables listed below, drop missing observations, and assign the output as a `tibble` (*not in that particular order*).

- `sample_measurement` renamed as `ozone_ppm` (ozone measurement in ppm)
- `datetime` (date in YYYY-MM-DD format and time of measurement in HH:MM:SS)

Examine Data

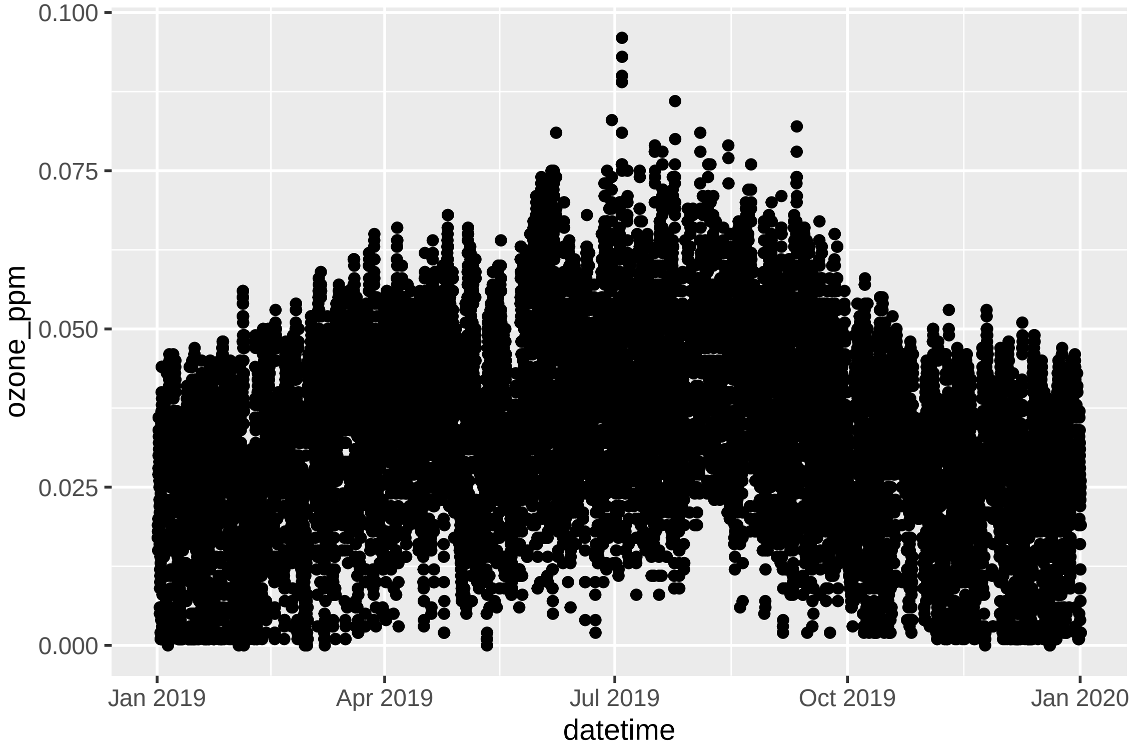
Examine the structure and contents of the dataframe to confirm the file imported and was manipulated properly.

```
##    ozone_ppm          datetime
##  Min.   :0.00000   Min.   :2019-01-01 07:00:00.00
##  1st Qu.:0.02300   1st Qu.:2019-03-30 21:00:00.00
##  Median :0.03300   Median :2019-06-27 12:30:00.00
##  Mean   :0.03305   Mean   :2019-07-01 06:37:00.78
##  3rd Qu.:0.04300   3rd Qu.:2019-10-03 22:45:00.00
##  Max.   :0.09600   Max.   :2020-01-01 06:00:00.00
```

```
## [1] 0
```

Question 1: `ggplot2` time series

Using `ggplot` and the corresponding `geom`, create a time series of ozone measurement across time. Warning: This plot will have a very poor ink-to-information ratio. Ugly plots are okay when you are just exploring data.



Question 2: Base R equivalent

For comparison, what function could you use to create a time series of these data in base R? How does the syntax of this function compare to that of `ggplot()`?

The comporable function in base R is plot(). The syntax for this would be: plot(x=ozone_data$datetime,y=ozone_data$ozon

Question 3: ggplot object

Excluding the geom, assign the plot from Question 1 as a ggplot object with a descriptive name.

Question 4: geom

Now, geom_point() to the ggplot object using the following syntax: object_name + geom_point(). Remember, you have already defined the aes() in the ggplot object in Question 3.

Question 5: ggplot layers

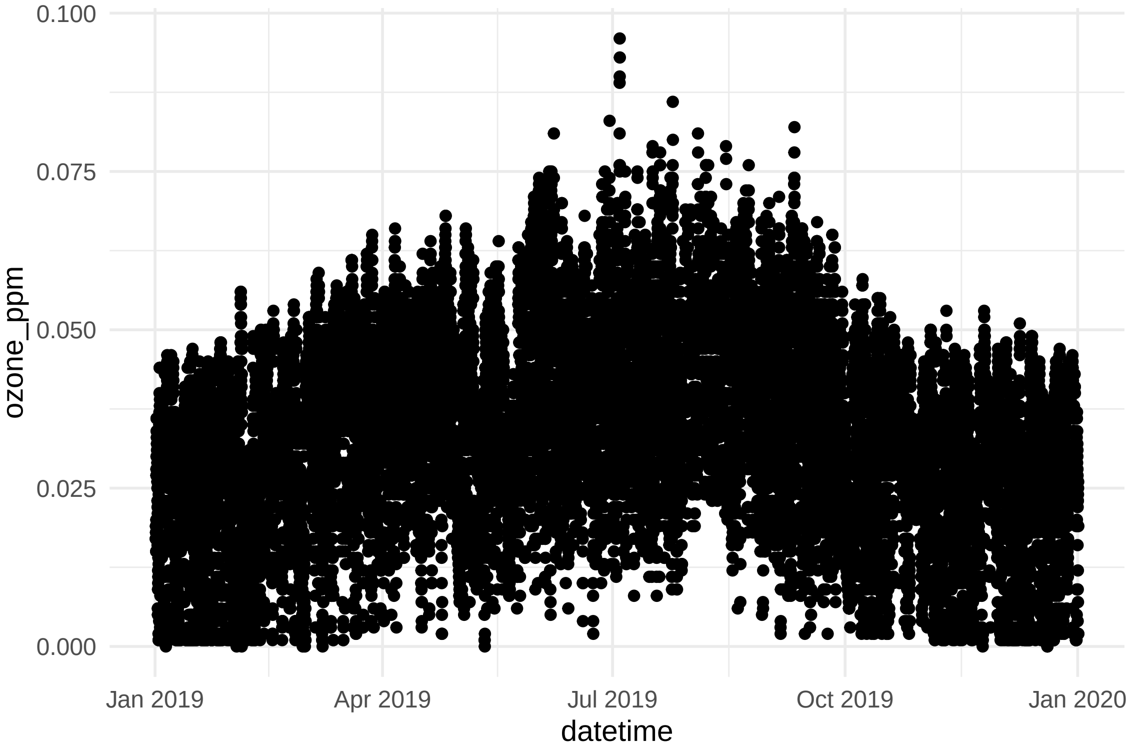
Call and examine object within the R Markdown, Console, or using View. How many layers does this ggplot object contain? Why?

The object only has one layer this is because only one layer has been added. In the line below you can see there is one layer and it is of type geom_point

```
## [[1]]
## geom_point: na.rm = FALSE
## stat_identity: na.rm = FALSE
## position_identity
```

Question 6: theme

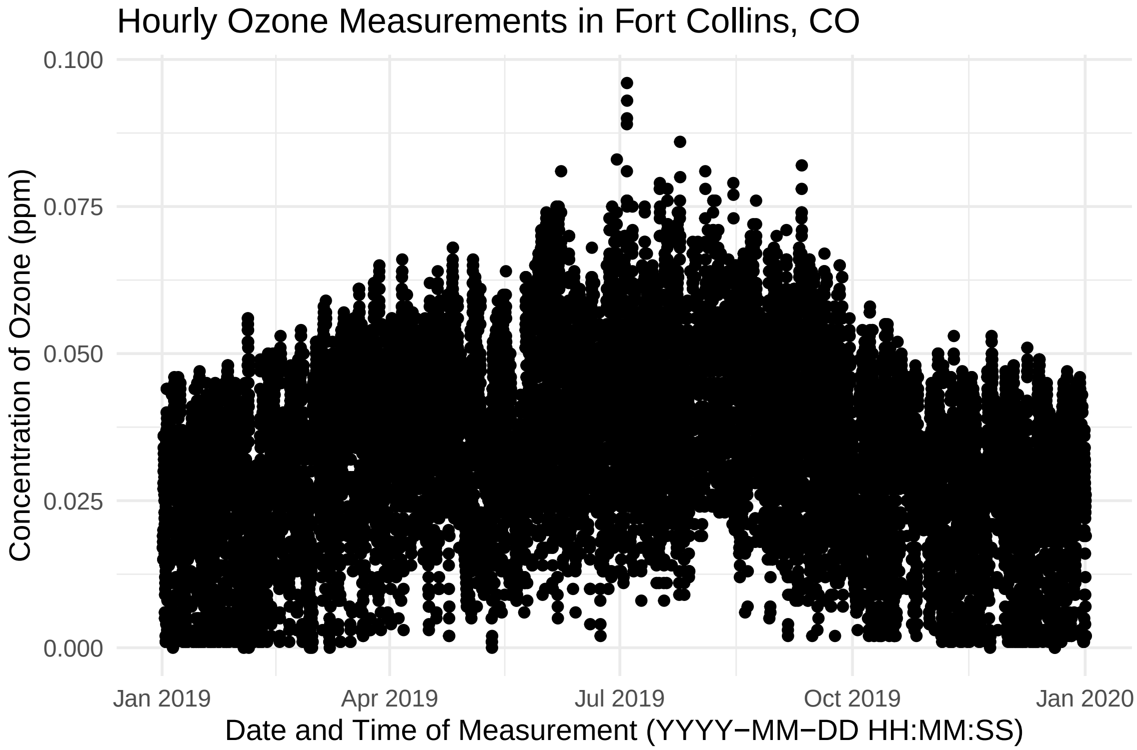
Next, add the ggplot2 theme of your choice to the ggplot object with theme_*() function prefix.



Question 7: additions

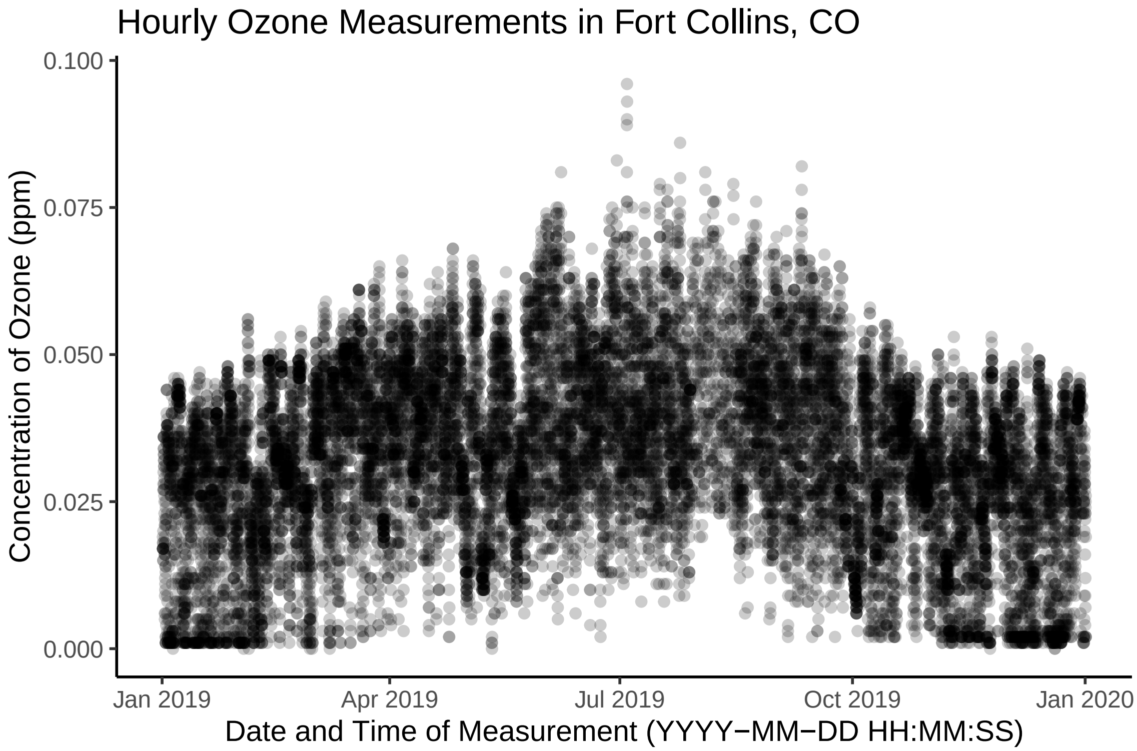
In addition to assigning a plot as a `ggplot` object, one can also assign aspects of the figure such as axis labels and titles to an object for later use. For example:

Using this technique and the same additive approach (`ggplot_object + ... + title`) from Questions 5 and 6, add a title and revise the axis labels.

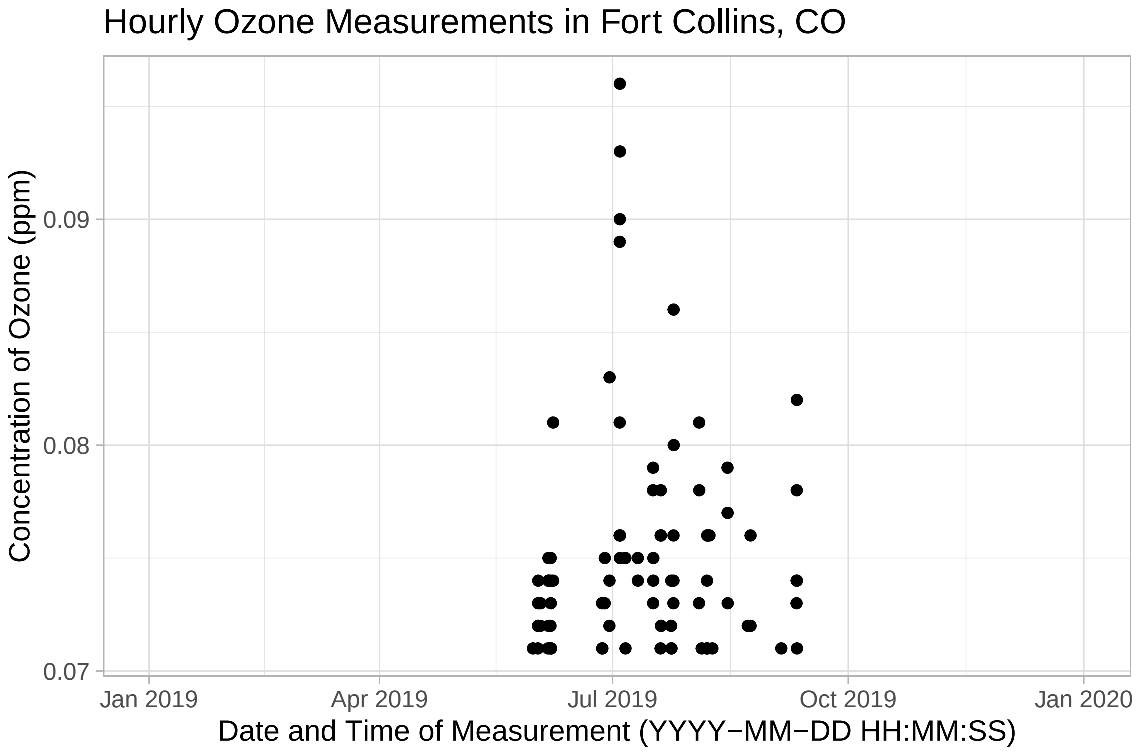


Question 7: ways to see more granularity

The time series from the previous questions does not look nice. It is hard to discern granular patterns in the data because of their sheer density; there are too many hourly measurements over time. We could look at the data on different time scales, but, because we have not discussed how to manipulate dates and times, we will instead focus first on adding transparency to the data points using the `alpha =` aesthetic. Try recreating the time-series plot with `alpha = 0.2`.



Alternatively, we could examine just the ozone measurements that exceed the threshold (0.070 ppm) set by the Environmental Protection Agency. Filter the dataset to the ozone levels exceeding 0.070 ppm, and use these data to construct a time series plot with time of measurement on the x-axis and ozone concentration measurement on the y-axis. Remember to add the relevant `geom`, title, subtitle, axis labels, and your choice of theme.



```
## # A tibble: 85 x 2
##    ozone_ppm datetime
##        <dbl> <dttm>
##  1     0.073 2019-06-02 21:00:00
##  2     0.075 2019-06-05 23:00:00
##  3     0.074 2019-06-07 19:00:00
##  4     0.081 2019-07-03 19:00:00
##  5     0.09  2019-07-03 20:00:00
##  6     0.075 2019-07-03 21:00:00
##  7     0.073 2019-07-16 20:00:00
##  8     0.074 2019-07-24 19:00:00
##  9     0.072 2019-08-23 21:00:00
## 10     0.074 2019-09-10 23:00:00
## # i 75 more rows
```

Question 8: seasonality

Based on the time series, do you see any seasonal pattern for higher levels of ozone? Describe what you see.

There is an increase in ozone concentration during Summer months (July & August).

Question 9: proportion

What proportion of ozone measurements exceed the EPA guidelines? Instead of plugging in the actual values, make R figure out the length of each vector and the corresponding proportion in one line of code.

```
## [1] 0.005025423
```

Question 10: `ggplot2` extensions

Navigate to this website and browse the `ggplot2` extensions. These themes can be very useful, so it's good to be aware of them. Which theme would be appropriate for your own research or senior project, and why? How would you use it? Briefly describe your data and why the extension would improve the data visualization and communication.

The theme 'ggflowchart' could be very useful for the senior design project. There are many deliverables that require flowcharts from both timeline tracking as well as system design and idea/concept development. This theme could develop the information in a clear and readable manner.

Appendix

```
# set global options for figures, code, warnings, and messages
knitr::opts_chunk$set(fig.width = 6, fig.height = 4, fig.path = "../figs/",
                      echo = FALSE, warning = FALSE, message = FALSE)
# load packages for current R session
library(tidyverse)
library(ggplot2)
# ozone: import, select, drop missing observations, rename
# use relative pathname
ozone_data <- read_csv("../data/ftc_o3.csv")%>%
 # select needed variables
  select(sample_measurement, datetime)%>%
 # drop missing observations
  na.omit()%>%
 # rename main variable
  rename(ozone_ppm = sample_measurement)
# examine dataframe object
summary(ozone_data)
sum(is.na(ozone_data$ozone_ppm))
# create basic time series using ggplot2 package
ozone_data %>%
  ggplot(aes(x=datetime,
             y=ozone_ppm))+
  geom_point()
# create base layer of ozone time series (no geom) and save to object
ozone_data_ggplot <- ozone_data %>%
  ggplot(aes(x=datetime,
             y=ozone_ppm))
# add layer to ggplot object
ozone_data_ggplot <- ozone_data_ggplot + geom_point()
head(ozone_data_ggplot$layers)
# add theme to ggplot object
ozone_data_ggplot <- ozone_data_ggplot + theme_minimal()

ozone_data_ggplot
# create new object with ggplot labels
ozone_labels <- labs(x = "Time of Measurement (YYYY-MM-DD HH:MM:SS)",
                     y = "Ozone Concentration (ppm)",
                     title = "Hourly Ozone Measurements in Fort Collins, CO")
# add title and axis labels to time series
ozone_data_ggplot <- ozone_data_ggplot + labs(x = "Date and Time of Measurement (YYYY-MM-DD HH:MM:SS)",
                                              y = "Concentration of Ozone (ppm)",
                                              title = "Hourly Ozone Measurements in Fort Collins, CO")

ozone_data_ggplot
# add alpha aesthetic to the geom_point()
ozone_plot_alpha <- ozone_data_ggplot
ozone_plot_alpha$layers[1] <- NULL
ozone_plot_alpha <-  ozone_plot_alpha +
  geom_point(aes(x = datetime,
             y = ozone_ppm),
             alpha = 0.2)+
```

```r
  theme_classic()

ozone_plot_alpha

# filter data to ozone concentration measurements exceeding 0.070 ppm
ozone_plot_thresh <- ozone_data %>%
  filter(ozone_ppm > 0.07)
# time series of high ozone measurements
ozone_plot_thresh %>%
  ggplot(aes(x=datetime,
             y=ozone_ppm))+
  geom_point() + theme_light() +
  labs(x = "Date and Time of Measurement (YYYY-MM-DD HH:MM:SS)",
       y = "Concentration of Ozone (ppm)",
       title = "Hourly Ozone Measurements in Fort Collins, CO") +
  xlim(min(ozone_data$datetime),max(ozone_data$datetime))

ozone_plot_thresh
# calculate proportion of ozone measurements that exceed 0.070 ppm
proportion <- length(ozone_plot_thresh$ozone_ppm)/length(ozone_data$ozone_ppm)
print(proportion)
```