Data Wrangling in R

Fall 2022, MATH8050: Homework 1
Your Name, Section XXX

Due September 7, 12:00 PM

General instructions for homeworks: Please follow the uploading file instructions according to the syllabus. You will give the commands to answer each question in its own code block, which will also produce plots that will be automatically embedded in the output file. Each answer must be supported by written statements as well as any code used. Your code must be completely reproducible and must compile.

Advice: Start early on the homeworks and it is advised that you not wait until the day of. While the professor and the TA's check emails, they will be answered in the order they are received and last minute help will not be given.

Commenting code Code should be commented. See the tidyverse style guide for questions regarding commenting or how to write code https://style.tidyverse.org/index.html. No late homework's will be accepted.

R Markdown Test

0. Open a new R Markdown file; set the output to HTML mode and "Knit". This should produce a web page with the knitting procedure executing your code blocks. You can edit this new file to produce your homework submission.

R Working Environment

Please load all the packages used in the following R chunk before the function `sessionInfo()`

```
# load packages
library(tidyverse)
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
library(lubridate)
##
## Attaching package: 'lubridate'
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union

sessionInfo()
## R version 4.1.3 (2022-03-10)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Monterey 12.3.1
## 
## Matrix products: default
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRlapack.dylib
## 
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
## 
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base     
## 
## other attached packages:
##  [1] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.8    
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.6   
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
## 
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2 xfun_0.30        haven_2.5.0      colorspace_2.0-3
##  [5] vctrs_0.4.1      generics_0.1.2   htmltools_0.5.2  yaml_2.3.5      
##  [9] utf8_1.2.2       rlang_1.0.2      pillar_1.7.0     glue_1.6.2      
## [13] withr_2.5.0      DBI_1.1.2        dbplyr_2.1.1     modelr_0.1.8    
## [17] readxl_1.4.0     lifecycle_1.0.1  munsell_0.5.0    gtable_0.3.0    
## [21] cellranger_1.1.0 rvest_1.0.2      evaluate_0.15    knitr_1.38      
## [25] tzdb_0.3.0       fastmap_1.1.0    fansi_1.0.3      broom_0.8.0     
## [29] backports_1.4.1  scales_1.2.0     jsonlite_1.8.0   fs_1.5.2        
## [33] hms_1.1.1        digest_0.6.29    stringi_1.7.6    grid_4.1.3      
## [37] cli_3.2.0        tools_4.1.3      magrittr_2.0.3   crayon_1.5.1    
## [41] pkgconfig_2.0.3  ellipsis_0.3.2   xml2_1.3.3       reprex_2.0.1    
## [45] assertthat_0.2.1 rmarkdown_2.13   httr_1.4.2       rstudioapi_0.13 
## [49] R6_2.5.1         compiler_4.1.3
```

Working with data

Total points on assignment: 10 (reproducibility) + 22 (Q1) + 9 (Q2) + 10 (Q3) + 30 (Q4) + 16 (Q5) + 3 (Q6) = 100 points

Reproducibility component: 10 points.

1. (22 pts total, equally weighted) The data set **rnf6080.dat** records hourly rainfall at a certain location in Canada, every day from 1960 to 1980.

a. Load the data set into R and make it a data frame called `rain.df`. What command did you use?

```
rain.df = read.table("./data/rnf6080.dat")
```

b. How many rows and columns does `rain.df` have? How do you know? (If there are not 5070 rows and 27 columns, you did something wrong in the first part of the problem.)

```
nrow(rain.df)
## [1] 5070
ncol(rain.df)
## [1] 27
```

c. What command would you use to get the names of the columns of `rain.df`? What are those names?

```
colnames(rain.df)
##  [1] "V1"  "V2"  "V3"  "V4"  "V5"  "V6"  "V7"  "V8"  "V9"  "V10" "V11" "V12"
## [13] "V13" "V14" "V15" "V16" "V17" "V18" "V19" "V20" "V21" "V22" "V23" "V24"
## [25] "V25" "V26" "V27"
```

d. What command would you use to get the value at row 2, column 4? What is the value?

```
rain.df[2, 4]
## [1] 0
```

e. What command would you use to display the whole second row? What is the content of that row?

```
rain.df[2, ]
##   V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20 V21
## 2 60  4  2  0  0  0  0  0  0   0   0   0   0   0   0   0   0   0   0   0   0
##   V22 V23 V24 V25 V26 V27
## 2   0   0   0   0   0   0
```

f. What does the following command do?

```
names(rain.df) <- c("year","month","day",seq(0,23))
```

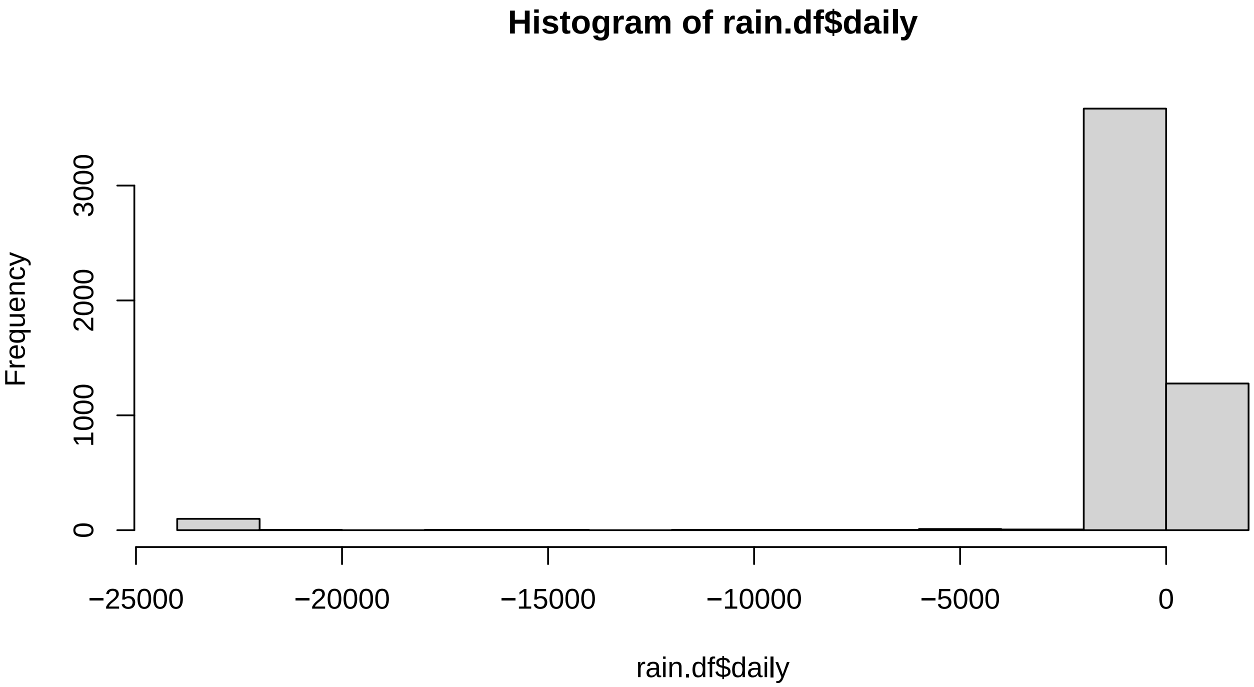
It renames the names of the variables V1 to V27.

g. Create a new column called `daily`, which is the sum of the 24 hourly columns.

```
rain.df$daily = rowSums(rain.df[,as.character(seq(0,23))])
```

h. Give the command you would use to create a histogram of the daily rainfall amounts. Please make sure to attach your figures in your .pdf report.

```
hist(rain.df$daily)
```



i. Explain why that histogram above cannot possibly be right. The rain amount should be at least 0, but the data contains negative values -999, which indicates missing values. In addition, the type of hourly rain amount is `integer` type, which should be converted to `double` type

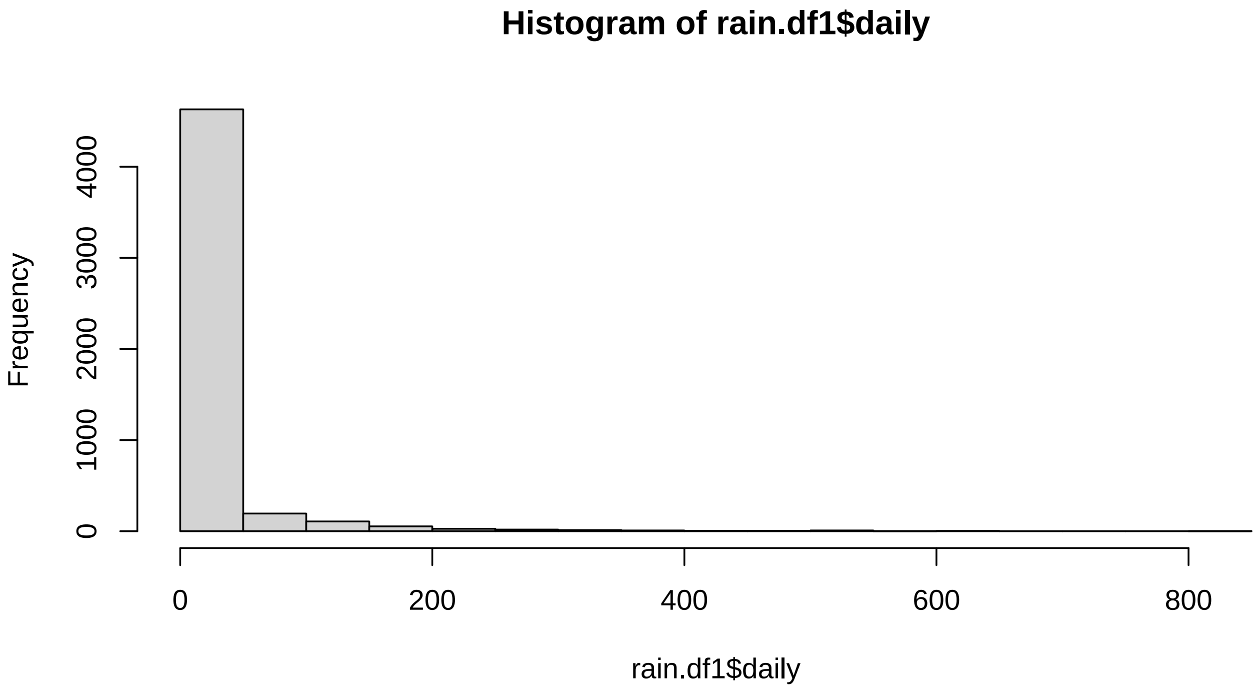
j. Give the command you would use to fix the data frame.

```
rain.df1 = rain.df
# replace all -999 with NA's
rain.df1[rain.df1==-999] = NA
# convert all integer
```

```
rain.df1$`0` = as.numeric(rain.df1$`0`)
rain.df1[,as.character(seq(0,23))] = sapply( rain.df1[,as.character(seq(0,23))], as.numeric)
```

k. Create a corrected histogram and again include it as part of your submitted report. Explain why it is more reasonable than the previous histogram.

```
rain.df1$daily = rowSums(rain.df1[,as.character(seq(0,23))], na.rm=TRUE)
hist(rain.df1$daily)
```



Data types

2. (9 pts total, equally weighted) Make sure your answers to different parts of this problem are compatible with each other.

a. For each of the following commands, either explain why they should be errors, or explain the non-erroneous result.

```
x <- c("5","12","7")
max(x)
sort(x)
sum(x)
```

1. ‘x‘ is binded to a vector of characters instead of numerical values. 2. The functions ‘max‘, ‘sort‘, ‘sum‘ should take numerical values as input, so the values in ‘x‘ are first converted from characters to numerical values implicitly and then the functions apply to these incorrect numerical values

b. For the next two commands, either explain their results, or why they should produce errors.

```
y <- c("5",7,12)
y[2] + y[3]
```

‘y‘ is a vector of characters, which cannot be used for addition.

c. For the next two commands, either explain their results, or why they should produce errors.

```
z <- data.frame(z1="5",z2=7,z3=12)
z[1,2] + z[1,3]
```

‘z‘ is a data frame, but the variable ‘z1‘ is a character while ‘z2‘ and ‘z3‘ are numerical. Adding ‘z2‘ and ‘z3‘ will produce the correct results.

Linear algebra

3. (10 pts total, equally weighted) Consider the linear system $AX = b$, where A is an $n \times n$ positive definite matrix and b is a n-dimensional vector, the unique solution is $X = A^{-1}b$. Please answer the following questions:

a. Write an R function called `my_solver()` such that given inputs A and b, the function `my_solver()` returns the solution of the linear system, i.e., `X <- my_solver(A, b)`.

```
my_solver <- function(A, b){
  R = chol(A)
  x = backsolve(R, backsolve(R, b, transpose=TRUE))
  return(x)
}
```

b. Run the following code to get A and b.

```
n = 100
set.seed(123)
A = rWishart(1, 150, diag(n))[ , ,1]
b = rnorm(n,1)
```

Then use your function `my_solver()` to produce the answer and verify your solution. (hint: AX should be equal to b)

```
n = 100
set.seed(123)
A = rWishart(1, 150, diag(n))[ , ,1]
b = rnorm(n,1)
x1 = my_solver(A, b)
sum((A%*%x1-b)^2)
## [1] 3.564536e-28
```

Working with ggplot2

4. (30 pts total, equally weighted) EPA monitors Air Quality data across the entire U.S. The file **AQSdata.csv** contains daily PM 2.5 concentrations and other information. Please make the following questions using the `ggplot()` function for plotting. In addition make sure that all the x-axis and y-axis labels have 14 font size.

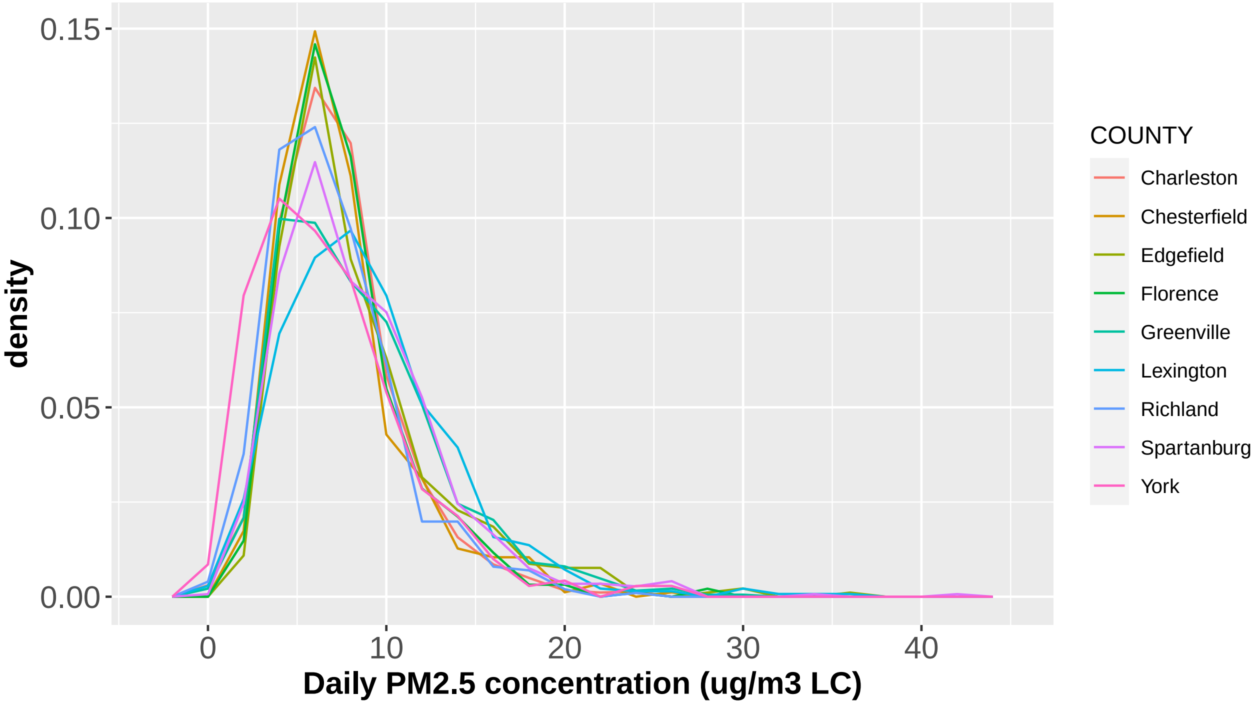
a. Read the data file **AQSdata.csv** into R.

```
df = read_csv("./data/AQSdata.csv")
## Rows: 6404 Columns: 20
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (9): Date, Source, UNITS, Site Name, AQS_PARAMETER_DESC, CBSA_NAME, STA...
## dbl (11): Site ID, POC, Daily Mean PM2.5 Concentration, DAILY_AQI_VALUE, DAI...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(df)
## # A tibble: 6 x 20
##   Date       Source `Site ID`   POC `Daily Mean PM2.5 Co~` UNITS DAILY_AQI_VALUE
##   <chr>      <chr>      <dbl> <dbl>                  <dbl> <chr>           <dbl>
## 1 11/09/2021 AQS    450190020     1                   15.5 ug/m~              58
## 2 11/10/2021 AQS    450190020     1                   13.6 ug/m~              54
## 3 11/11/2021 AQS    450190020     1                    8.1 ug/m~              34
## 4 11/12/2021 AQS    450190020     1                    7.1 ug/m~              30
## 5 11/13/2021 AQS    450190020     1                   10.7 ug/m~              45
## 6 11/14/2021 AQS    450190020     1                    7.5 ug/m~              31
## # ... with 13 more variables: `Site Name` <chr>, DAILY_OBS_COUNT <dbl>,
## #   PERCENT_COMPLETE <dbl>, AQS_PARAMETER_CODE <dbl>, AQS_PARAMETER_DESC <chr>,
## #   CBSA_CODE <dbl>, CBSA_NAME <chr>, STATE_CODE <dbl>, STATE <chr>,
## #   COUNTY_CODE <chr>, COUNTY <chr>, SITE_LATITUDE <dbl>, SITE_LONGITUDE <dbl>
```

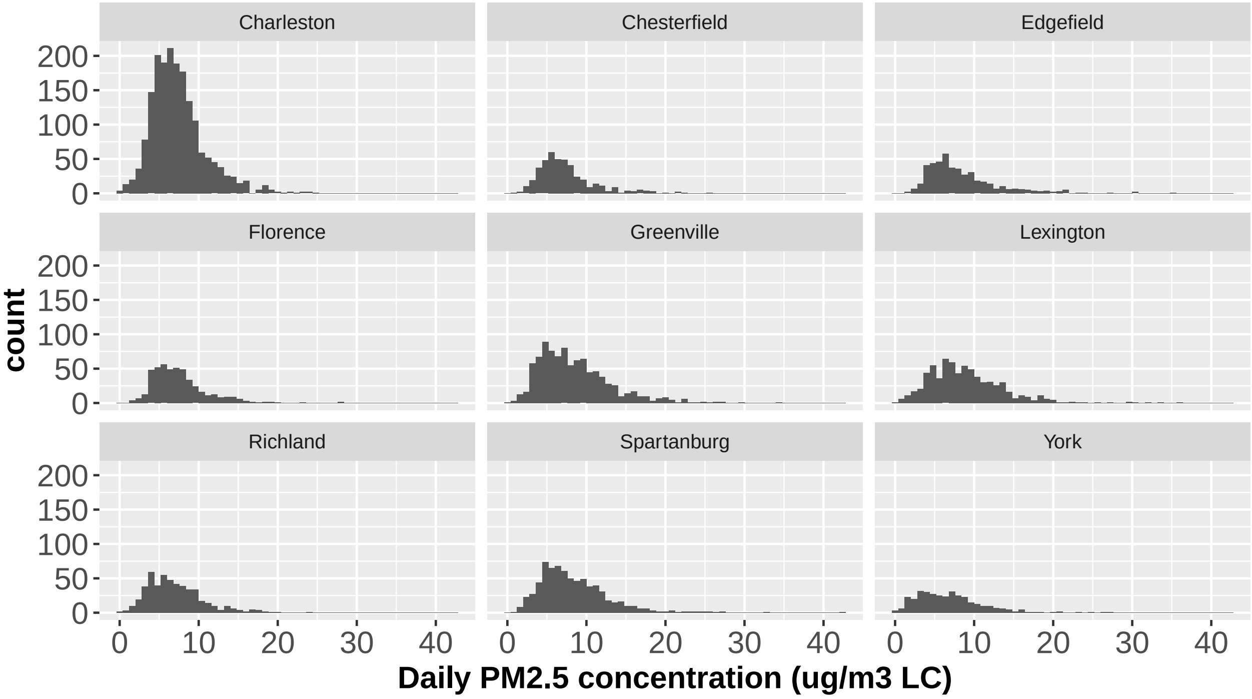
b. Generate density plots of PM2.5 concentrations grouped by `County` in one single panel, where each density should have its own color. What do you find from the figure?

```
theme_mat = theme(axis.text=element_text(size=14),
        axis.title=element_text(size=14,face="bold")
        )
df = rename(df, PM2.5=`Daily Mean PM2.5 Concentration`)
ggplot(df) +
  geom_freqpoly(aes(x=PM2.5, y=..density.., color=COUNTY), binwidth=2) +
  xlab("Daily PM2.5 concentration (ug/m3 LC)") +
  theme_mat
```



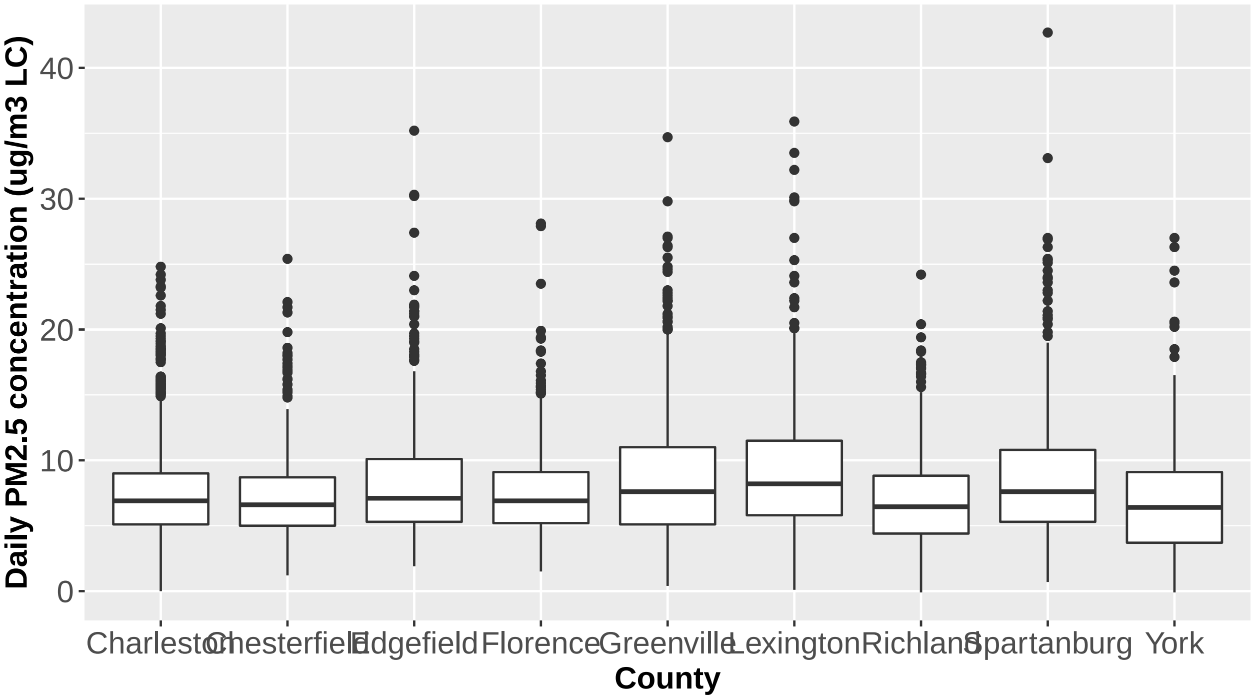
c. Plot histograms of PM2.5 concentrations across different counties with one panel for one histogram.

```
ggplot(df) +
  geom_histogram(aes(x=PM2.5), binwidth=.8) +
  facet_wrap(~COUNTY, ncol=3) +
  xlab("Daily PM2.5 concentration (ug/m3 LC)") +
  theme_mat
```



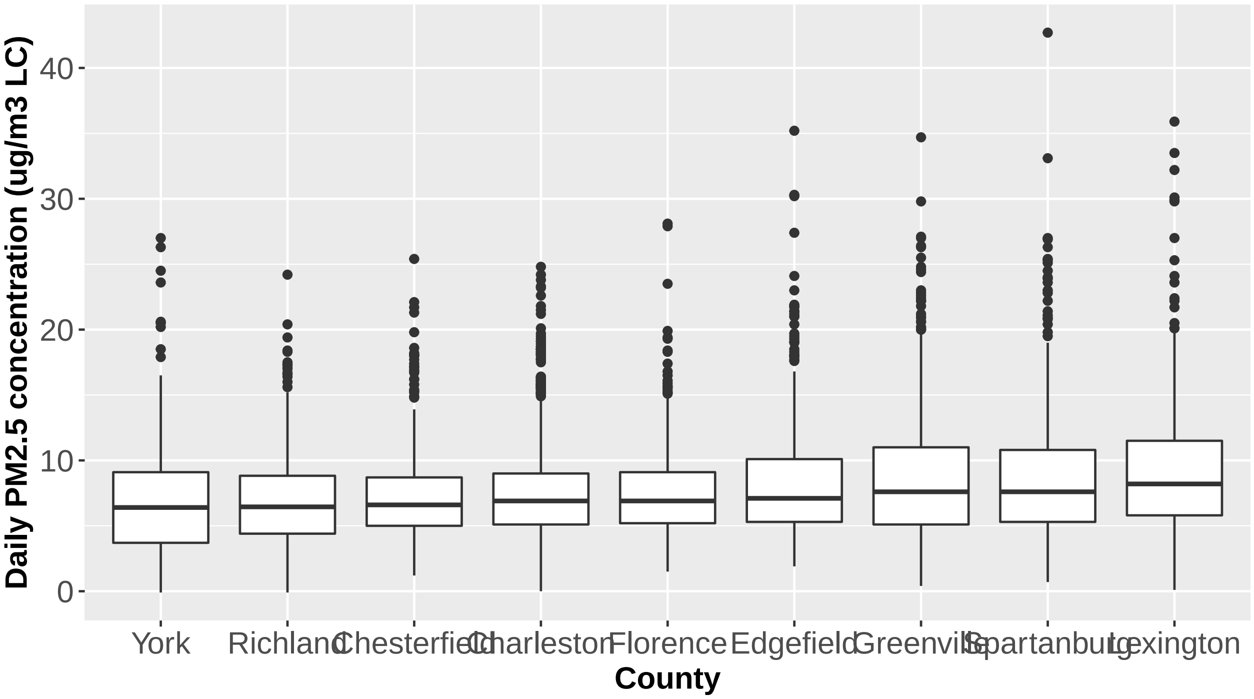
d. Generate boxplots of PM2.5 concentrations by County. What would you say about the distributions?

```
ggplot(df) +
  geom_boxplot(aes(x=COUNTY, y=PM2.5)) +
  xlab("County") +
  ylab("Daily PM2.5 concentration (ug/m3 LC)") +
  theme_mat
```



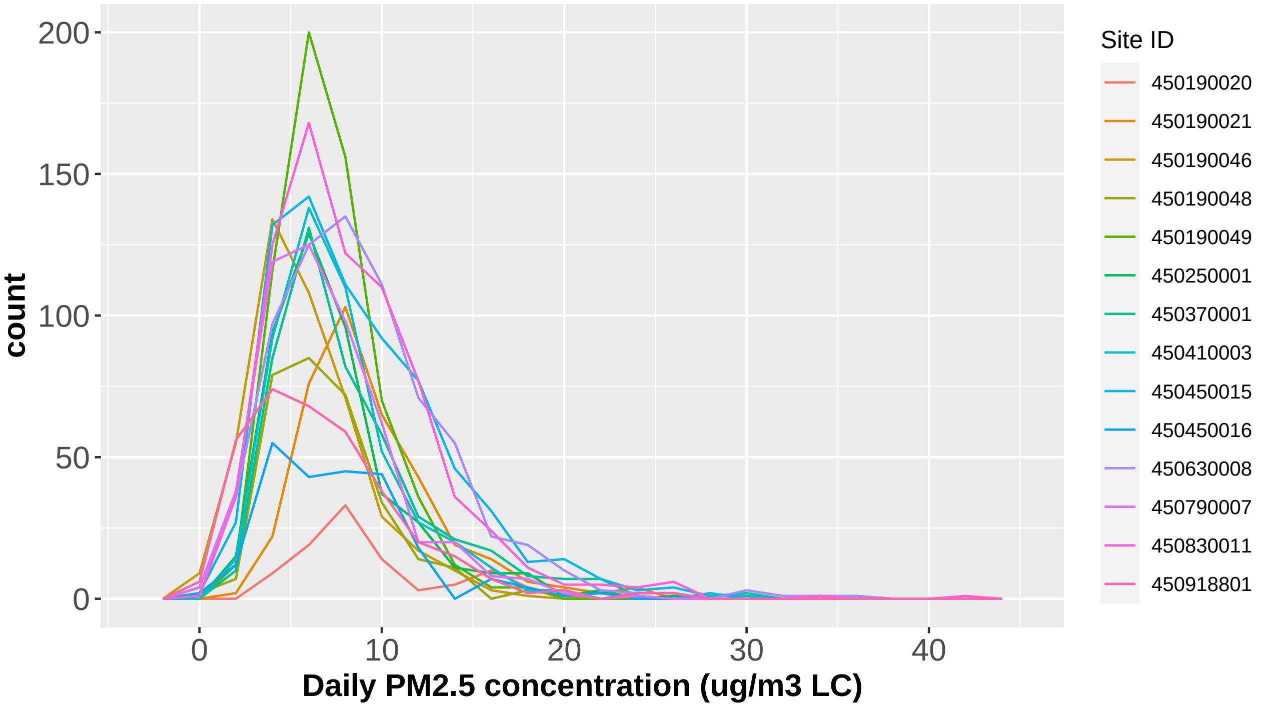
e. Reorder the boxplots above by the median value of PM2.5 concentrations.

```
ggplot(df) +
  geom_boxplot(aes(x=reorder(COUNTY, PM2.5, FUN=median), y=PM2.5)) +
  xlab("County") +
  ylab("Daily PM2.5 concentration (ug/m3 LC)") +
  theme_mat
```



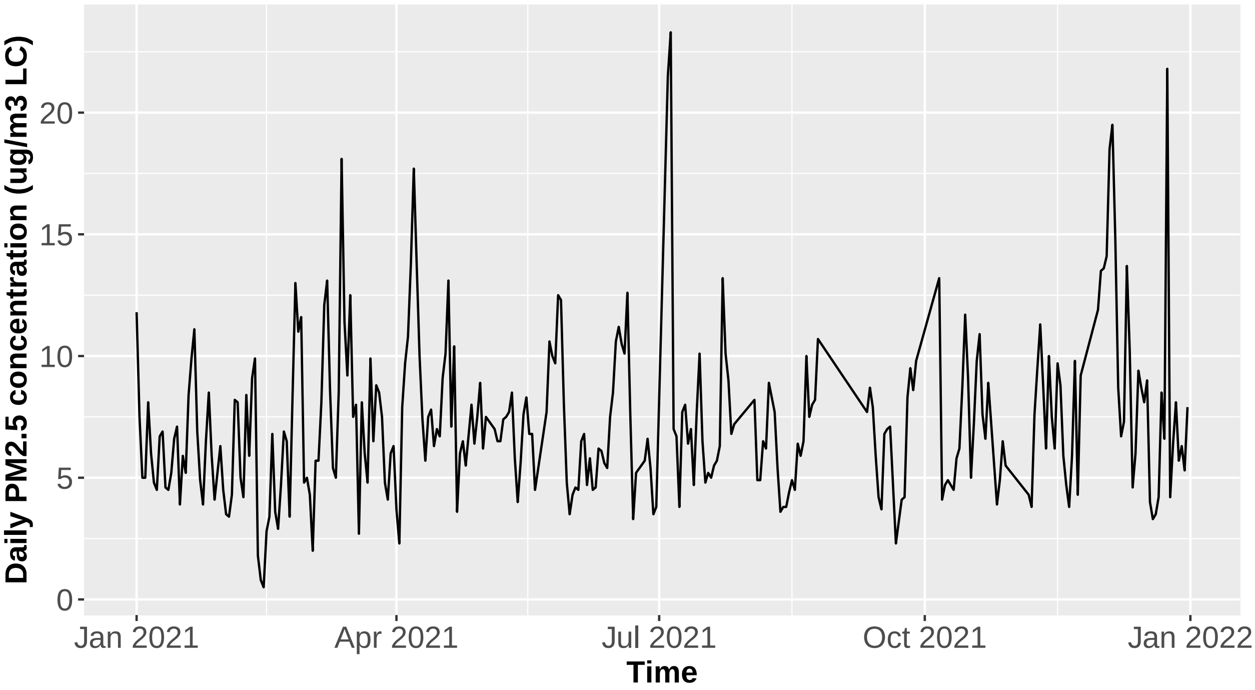
f. Converting the Site ID to a factor and plot the historgram grouped by `Site ID`.

```
df1 = df
df1$`Site ID` =  as.factor(df$`Site ID`)
ggplot(df1) +
  geom_freqpoly(aes(x=PM2.5, color=`Site ID`), binwidth=2) +
  xlab("Daily PM2.5 concentration (ug/m3 LC)") +
  theme_mat
```



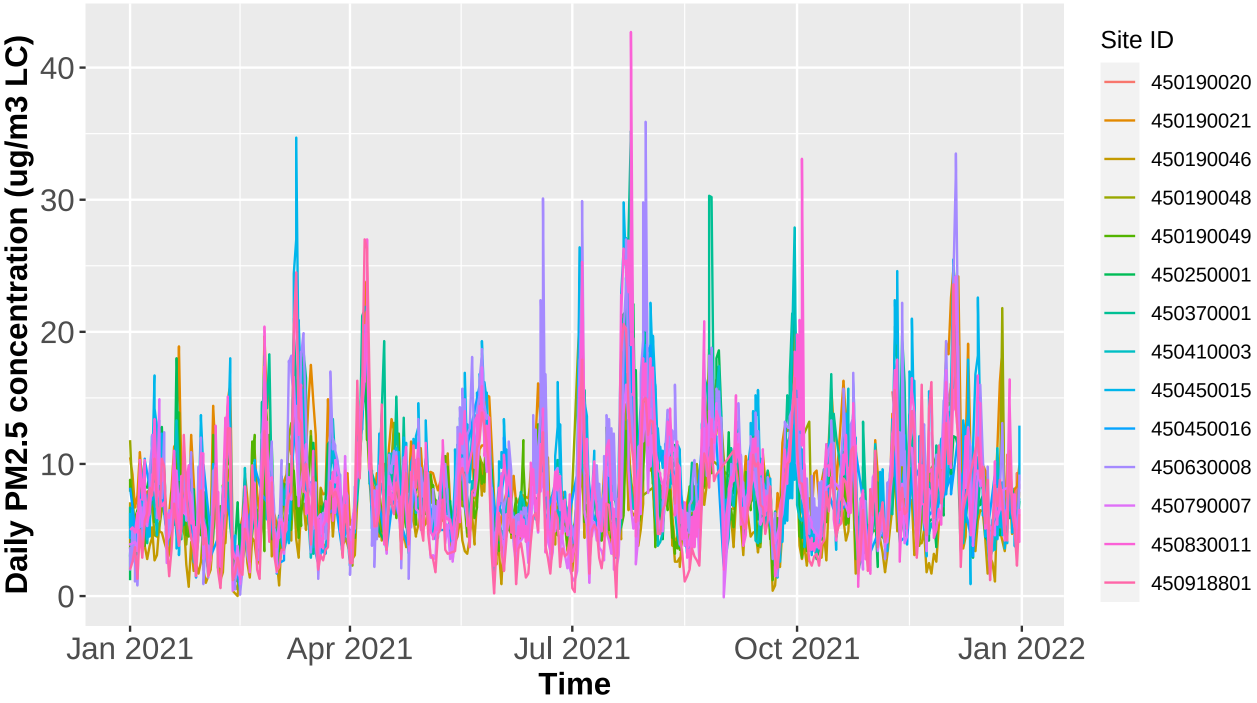
g. Generate the time series plot for the monitoring Site ID 450190048.

```
df1 %>%
  filter(`Site ID`==450190048) %>%
  ggplot() +
  geom_line(aes(x=mdy(Date), y=PM2.5)) +
  labs(
    x = "Time",
    y = "Daily PM2.5 concentration (ug/m3 LC)"
  ) + theme_mat
```



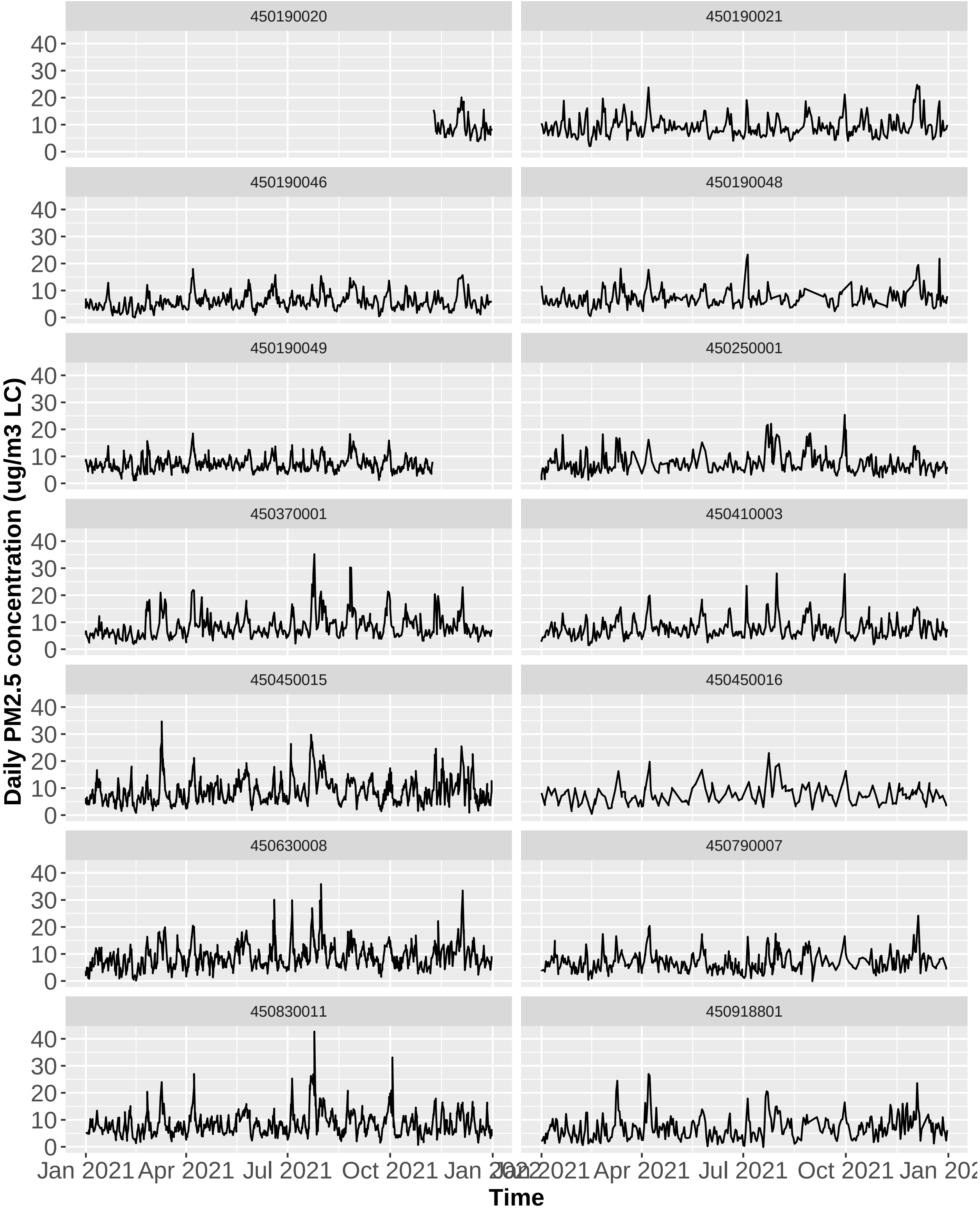
h. Plot time series of PM2.5 concentrations for all monitoring sites in one panel, where each site has its own color

```
df1 %>%
  ggplot() +
  geom_line(aes(x=mdy(Date), y=PM2.5, color=`Site ID`)) +
  labs(
    x = "Time",
    y = "Daily PM2.5 concentration (ug/m3 LC)"
  ) + theme_mat
```



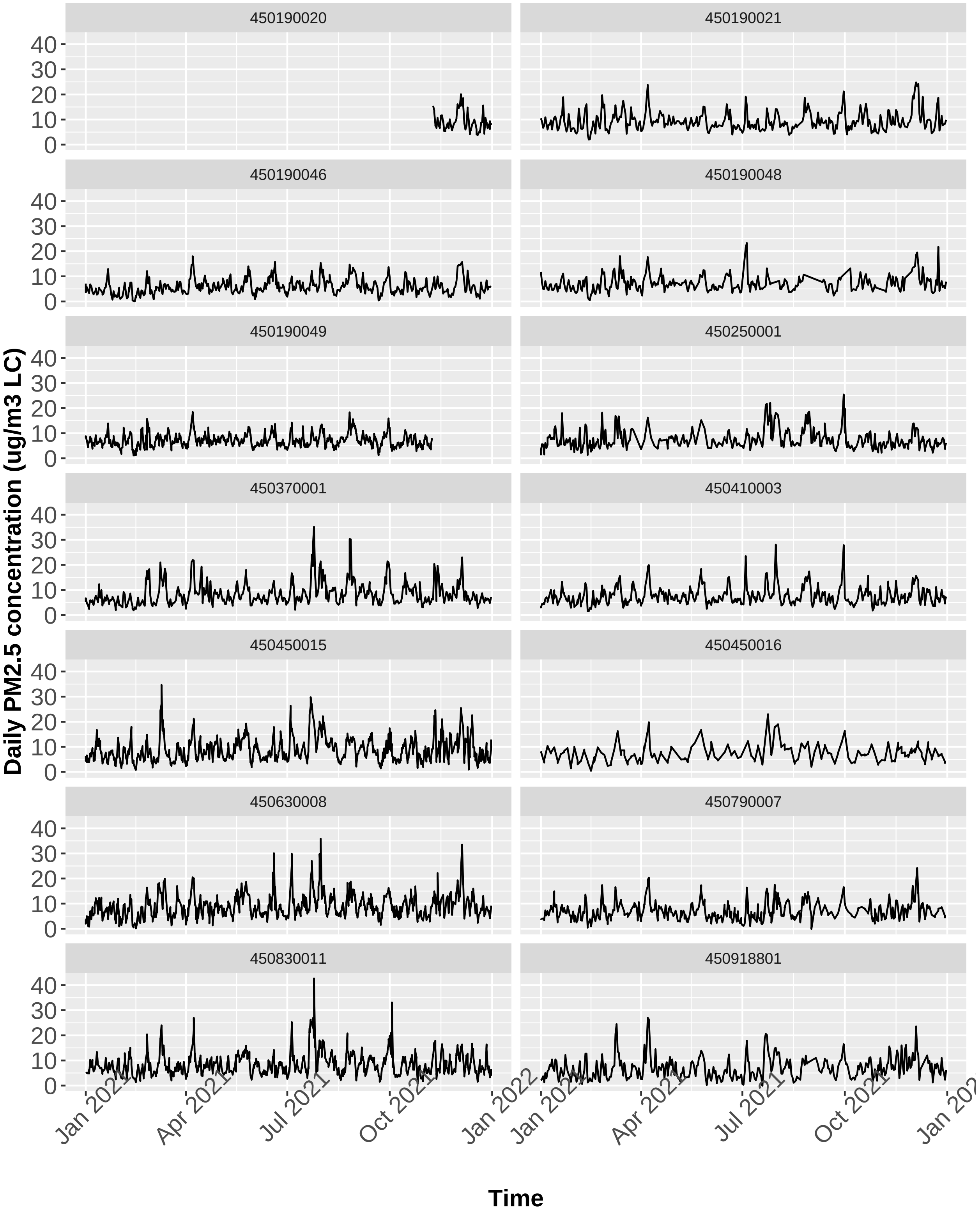
i. Plot time series of PM2.5 concentrations across all monitoring sites in multiple panels, where one panel only has one site, and each row only has two panels.

```
g <- df1 %>%
  ggplot() +
  geom_line(aes(x=mdy(Date), y=PM2.5)) +
  facet_wrap(~`Site ID`, ncol=2) +
  labs(
    x = "Time",
    y = "Daily PM2.5 concentration (ug/m3 LC)"
  ) + theme_mat
print(g)
```



j. In the time series plot, there seems to be not enough space to hold the x-axis labels. One way to avoid this is to rotate the axis labels. Please rotate all the time labels 45 degree.

```
g + theme(axis.text.x = element_text(angle=45))
```



Working with dplyr

5. (16 pts total, equally weighted) Continuing working with the above PM 2.5 data.

a. Filter all the observations in the county Greenville. How many observations are there?

```
df %>%
  filter(COUNTY=="Greenville")
## # A tibble: 937 x 20
##    Date       Source `Site ID`   POC PM2.5 UNITS    DAILY_AQI_VALUE `Site Name`
##    <chr>      <chr>      <dbl> <dbl> <dbl> <chr>              <dbl> <chr>
##  1 01/01/2021 AQS    450450015     1   6.3 ug/m3 LC              26 Greenville ~
##  2 01/02/2021 AQS    450450015     1   6.6 ug/m3 LC              28 Greenville ~
##  3 01/03/2021 AQS    450450015     1   5.4 ug/m3 LC              23 Greenville ~
##  4 01/05/2021 AQS    450450015     1   7.4 ug/m3 LC              31 Greenville ~
##  5 01/06/2021 AQS    450450015     1   7.7 ug/m3 LC              32 Greenville ~
##  6 01/07/2021 AQS    450450015     1   9.5 ug/m3 LC              40 Greenville ~
##  7 01/08/2021 AQS    450450015     1   7.5 ug/m3 LC              31 Greenville ~
##  8 01/09/2021 AQS    450450015     1   5.3 ug/m3 LC              22 Greenville ~
##  9 01/10/2021 AQS    450450015     1  12.1 ug/m3 LC              51 Greenville ~
## 10 01/11/2021 AQS    450450015     1  16.7 ug/m3 LC              61 Greenville ~
## # ... with 927 more rows, and 12 more variables: DAILY_OBS_COUNT <dbl>,
## #   PERCENT_COMPLETE <dbl>, AQS_PARAMETER_CODE <dbl>, AQS_PARAMETER_DESC <chr>,
## #   CBSA_CODE <dbl>, CBSA_NAME <chr>, STATE_CODE <dbl>, STATE <chr>,
## #   COUNTY_CODE <chr>, COUNTY <chr>, SITE_LATITUDE <dbl>, SITE_LONGITUDE <dbl>
```

Clearly, it shows that there are 937 observations in the coun ty Greenville.

b. Filter all the observations in Greenville in August 2021

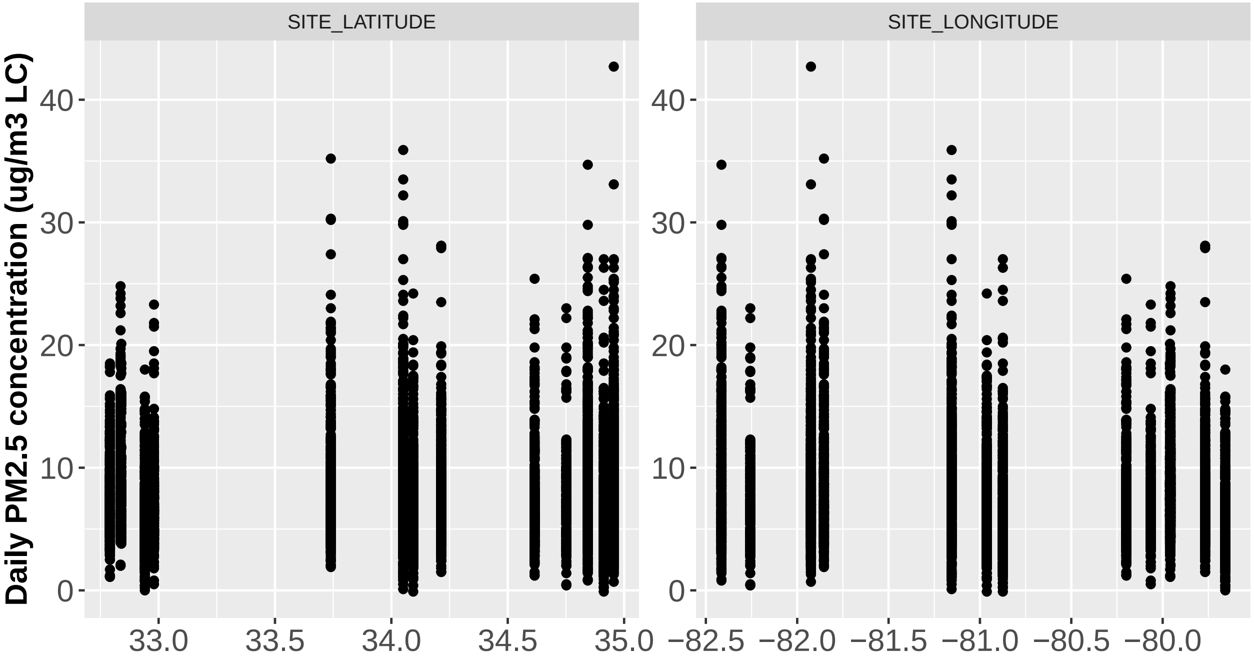
```
df %>%
  mutate(Date = mdy(Date),
         YM = format_ISO8601(Date, precision="ym")
  ) %>%
  filter(COUNTY=="Greenville", YM=="2021-08")
## # A tibble: 82 x 21
##    Date       Source `Site ID`   POC PM2.5 UNITS    DAILY_AQI_VALUE `Site Name`
##    <date>     <chr>      <dbl> <dbl> <dbl> <chr>              <dbl> <chr>
##  1 2021-08-01 AQS    450450015     1  13.8 ug/m3 LC              55 Greenville ~
##  2 2021-08-02 AQS    450450015     1  19   ug/m3 LC              66 Greenville ~
##  3 2021-08-03 AQS    450450015     1  16.9 ug/m3 LC              61 Greenville ~
##  4 2021-08-04 AQS    450450015     1  15.6 ug/m3 LC              58 Greenville ~
##  5 2021-08-05 AQS    450450015     1  11   ug/m3 LC              46 Greenville ~
##  6 2021-08-06 AQS    450450015     1  10.3 ug/m3 LC              43 Greenville ~
##  7 2021-08-07 AQS    450450015     1   9.7 ug/m3 LC              40 Greenville ~
##  8 2021-08-08 AQS    450450015     1  10   ug/m3 LC              42 Greenville ~
##  9 2021-08-09 AQS    450450015     1  12   ug/m3 LC              50 Greenville ~
## 10 2021-08-10 AQS    450450015     1  12.5 ug/m3 LC              52 Greenville ~
## # ... with 72 more rows, and 13 more variables: DAILY_OBS_COUNT <dbl>,
## #   PERCENT_COMPLETE <dbl>, AQS_PARAMETER_CODE <dbl>, AQS_PARAMETER_DESC <chr>,
## #   CBSA_CODE <dbl>, CBSA_NAME <chr>, STATE_CODE <dbl>, STATE <chr>,
## #   COUNTY_CODE <chr>, COUNTY <chr>, SITE_LATITUDE <dbl>, SITE_LONGITUDE <dbl>,
## #   YM <chr>
```

c. Filter all the observations in Greenville in August 2021 and select the variables PM2.5 concentrations, Date, latitude and longitude of sites

```
df %>%
  mutate(Date = mdy(Date),
        YM = format_ISO8601(Date, precision="ym")
  ) %>%
  filter(COUNTY=="Greenville", YM=="2021-08") %>%
  select(PM2.5, Date, SITE_LATITUDE, SITE_LONGITUDE)
## # A tibble: 82 x 4
##    PM2.5 Date       SITE_LATITUDE SITE_LONGITUDE
##    <dbl> <date>             <dbl>          <dbl>
##  1  13.8 2021-08-01          34.8          -82.4
##  2  19   2021-08-02          34.8          -82.4
##  3  16.9 2021-08-03          34.8          -82.4
##  4  15.6 2021-08-04          34.8          -82.4
##  5  11   2021-08-05          34.8          -82.4
##  6  10.3 2021-08-06          34.8          -82.4
##  7   9.7 2021-08-07          34.8          -82.4
##  8  10   2021-08-08          34.8          -82.4
##  9  12   2021-08-09          34.8          -82.4
## 10  12.5 2021-08-10          34.8          -82.4
## # ... with 72 more rows
```

d. Generate scatter plot of PM2.5 against latitude and longitude in two different panels

```
df %>%
  select(PM2.5, SITE_LATITUDE, SITE_LONGITUDE) %>%
  #gather(-PM2.5,  key="variable", value="value") %>%
  pivot_longer(cols=c("SITE_LATITUDE", "SITE_LONGITUDE"),
               names_to="variable",
               values_to="value") %>%
  ggplot(aes(x=value, y=PM2.5)) +
  geom_point() +
  facet_wrap(~variable, scale="free") +
  xlab("") +
  ylab("Daily PM2.5 concentration (ug/m3 LC)") +
  theme_mat
```



6. (3 pts total, equally weighted).

a. What is the point of reproducible code?

Answer Varies: The point is to reproduce my workflow including code, figures, and numerical results by anyone else without making any change in my own code and workflow.

b. Given an example of why making your code reproducible is important for you to know in this class and moving forward.

Answer Varies

c. On a scale of 1 (easy) – 10 (hard), how hard was this assignment. If this assignment was hard (> 5), please state in one sentence what you struggled with.

Answer Varies