p8105_hw3_SZ3319

Shiyu Zhang

2024-10-09

Set up packages

```
library(tidyverse)
library(readxl)
library(haven)
library(tidyr)
library(p8105.datasets)
library(patchwork)
library(knitr)
library(ggridges)
```

Problem 1

Import `ny_noaa` dataset and look at the size and the structure.

```
data("ny_noaa")
str(ny_noaa)
dim(ny_noaa)
head(ny_noaa)
skimr::skim(ny_noaa)
```

This dataset `ny_noaa` is weather observation data from the National Oceanic and Atmospheric Administration (NOAA) in New York, which includes **2595176** observations and **7** variables.

The **7** important variables are: **id, date, prcp, snow, snwd, tmax, tmin**:

- `id` (Weather Station ID): the unique identifier for each observation station. **Type:** Character.
- `date` (Date of Observation): the date of the weather observation in the format YYYY-MM-DD. **Type:** Date.
- `prcp` (Precipitation): the precipitation amount in tenths of a millimeter, indicating the amount of rain that fell on a specific date. **Type:** Integer, which need to be transferred into numeric.
- `snow` (Snowfall): the snowfall amount in millimeters, indicating the depth of new snow that fell on a specific date. **Type:** Integer, which need to be transferred into numeric.
- `snwd` (Snow Depth): the snow depth in millimeters, indicating the thickness of the snowpack on a specific date. **Type:** Integer, which need to be transferred into numeric.

- `tmax` (Maximum Temperature): the maximum temperature in tenths of degrees Celsius, indicating the highest temperature recorded on a specific date. **Type:** Character, which need to be transferred into numeric.
- `tmin` (Minimum Temperature): the minimum temperature in tenths of degrees Celsius, indicating the lowest temperature recorded on a specific date. **Type:** Character, which need to be transferred into numeric.

1.1 Clean the dataset

Clean the dataset

```
ny_noaa =
  data.frame(ny_noaa) |>
  janitor::clean_names() |>
  mutate(
    year = year(date),
    month = month(date),
    day = day(date),
    across(c(year, month, day, prcp, snow, snwd, tmin, tmax), as.numeric),
    prcp = prcp / 10,
    tmax = tmax / 10,
    tmin = tmin / 10
  )
```

View the top 10 snow fall values.

```
common_snow =
  ny_noaa  |>
  filter(snow >= 0)  |>
  group_by(snow) |>
  summarise(count = n()) |>
  arrange(desc(count))  |>
  head(10)

kable(common_snow,
      col.names = c("Snow", "Count"),
      caption = "Top 10 Snowfall Values")
```

Table 1: Top 10 Snowfall Values

Snow	Count
0	2008508
25	31022
13	23095
51	18274
76	10173
8	9962
5	9748
38	9197
3	8790
102	6552

The result indicates that:

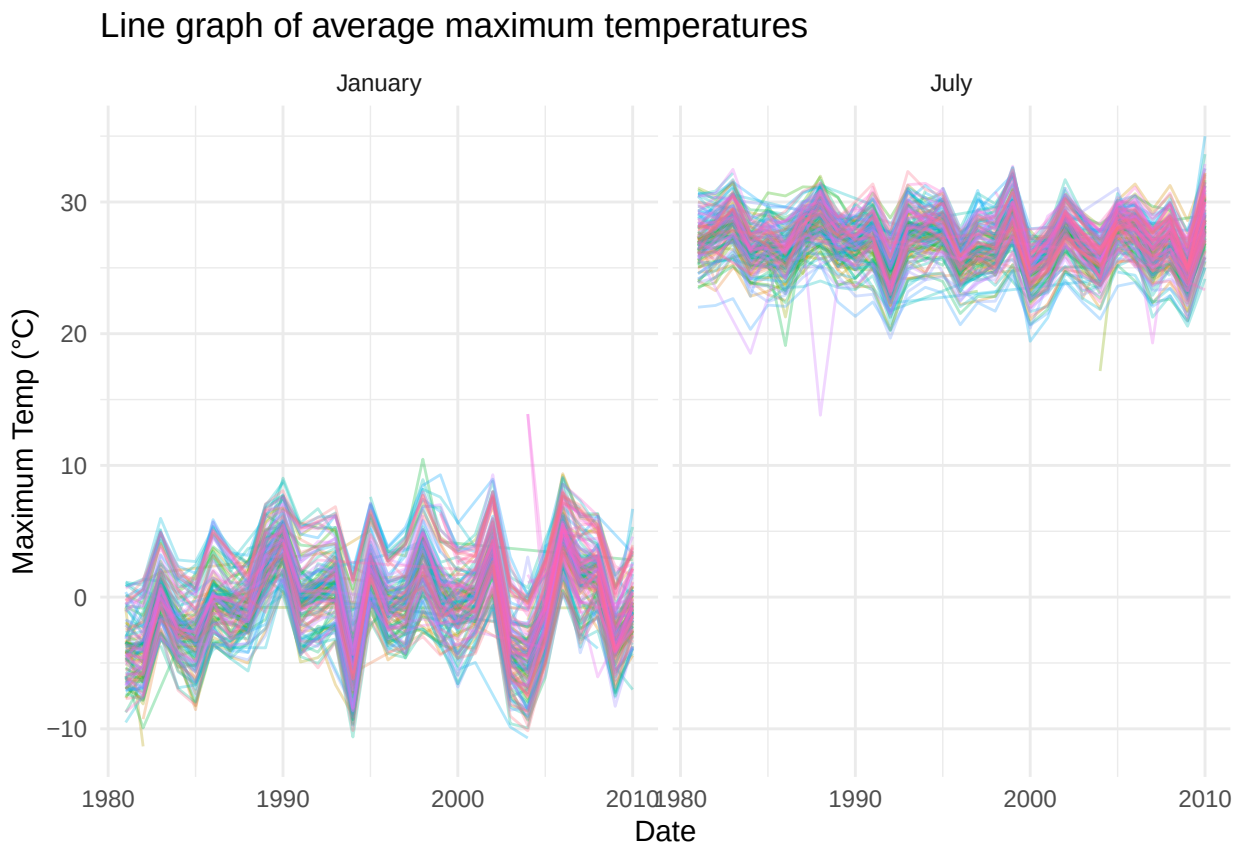
- **0 mm** snowfall is most common, which mens that on the majority of days, there was no snowfall in the New York area.
- Snowfall amounts of 25 mm or less are relatively common, reflecting winter snowfall patterns.
- Although snowfall greater than 100 mm is not common, it can still occur, particularly during extreme weather events.

1.2 Creat line graph of average maximum temperatures

```
ny_noaa_month =
  ny_noaa |>
  filter(
    month == c(1, 7),
    !is.na(tmax)
  ) |>
  mutate(
    month = as.character(month),
    month = case_match(
      month,
      "1" ~ "January",
      "7" ~ "July"
    )
  ) |>
  group_by(id, year, month) |>
  summarize(avg_tmax = mean(tmax, na.rm = TRUE)) |>
  ungroup()
```

Create a plot.

```
ny_noaa_month |>
  ggplot(aes(x = year, y = avg_tmax, colour = id)) +
  geom_line(alpha = .3) +
  labs(
    title = "Line graph of average maximum temperatures",
       x = "Date",
       y = "Maximum Temp (°C)"
  ) +
  facet_wrap(. ~ month) +
  theme_minimal() +
  theme(legend.position = "none")
```



From the plot, it shows that:

- The clear temperature difference between January (winter) and July (summer) is consistent with expected seasonal variation.
- Overall, the trends in average maximum temperatures over the years are consistent across different stations. Individual outliers may be attributed to extreme climatic events occurring at specific weather station locations.
- Both months reflect seasonal temperature patterns typical for winter and summer. While January generally shows lower temperatures due to winter conditions, and July shows higher temperatures due to summer heat, both months may exhibit similar trends in their respective temperature increases or decreases over the years.

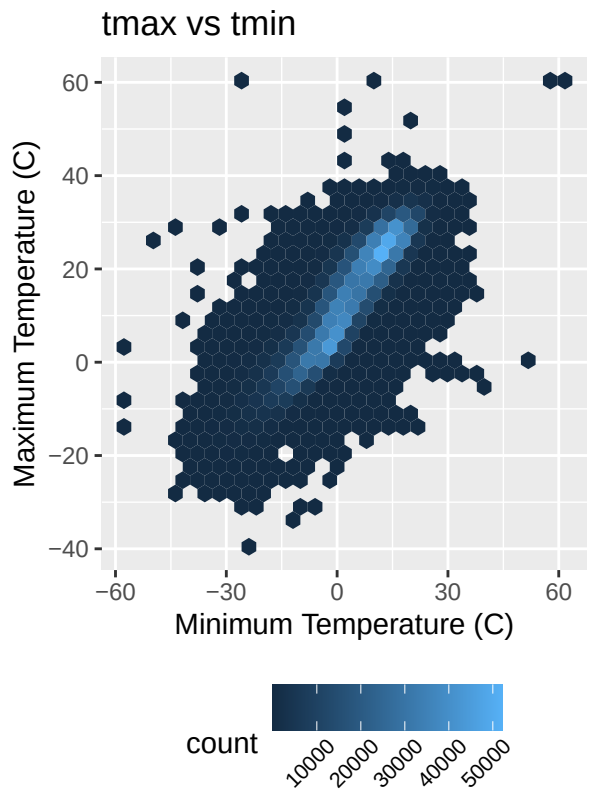
1.3 Create a joint plot

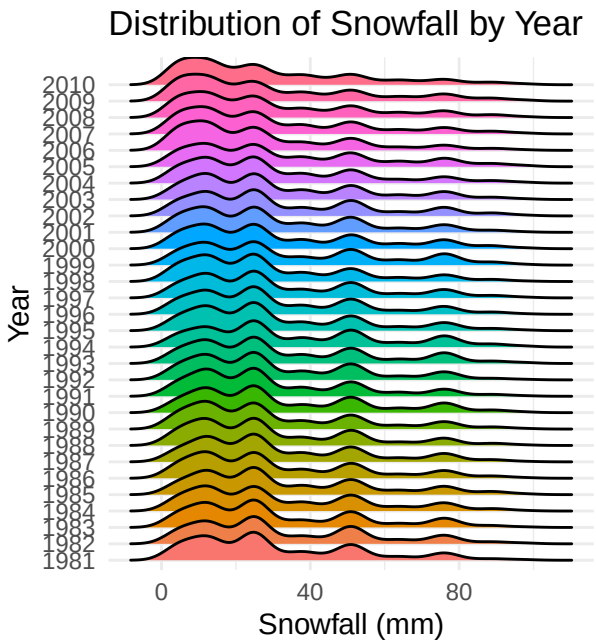
```
ny_noaa_clean = ny_noaa |>
  janitor::clean_names()  |>
  mutate(
    snow = ifelse(snow > 0 & snow < 100, snow, NA)# Keep only snow values between 0 and 100
  ) |>
   filter(
    !is.na(year),
    !is.na(snow)
    )
```

```r
# Panel (i): tmax vs tmin
panel_i =
  ny_noaa |>
  filter(
    !is.na(tmax),
    !is.na(tmin)
    ) |>
  ggplot(aes(x = tmin, y = tmax)) +
  geom_hex() + # Using a hexbin plot instead of scatter
  labs(
    title = "tmax vs tmin",
    x = "Minimum Temperature (C)",
    y = "Maximum Temperature (C)"
  ) +
  theme(legend.position = "bottom", legend.text = element_text(angle = 45, hjust = 0.8, size = 8  ))

# Panel (ii): Distribution of snowfall
panel_ii =
  ny_noaa_clean |>
  ggplot(aes(y = factor(year), x = snow, fill = factor(year))) +
  geom_density_ridges()+
  labs(
    title = "Distribution of Snowfall by Year",
    y = "Year",
    x = "Snowfall (mm)"
  ) +
  theme_minimal() +
  theme(legend.position = "none")

print(panel_i + panel_ii) # combine 2 plots
```





Problem 2

2.0 Load the datesets

```
demo_df =
  read_csv("data/nhanes_covar.csv",
           na = c("NA", "", "."),
           skip = 4) |>
  janitor::clean_names()
str(demo_df)

ac_df =
  read_csv("data/nhanes_accel.csv", na = c("NA", "", ".")) |>
  janitor::clean_names() |>
  na.omit()
```

2.1 Clean the dataset and joint the table

- Include all originally observed variables;
- exclude participants less than 21 years of age, and those with missing demographic data;
- encode data with reasonable variable classes.

```
demo_clean_df =
  demo_df |>
  filter(age >= 21) |>
  na.omit() |>
  mutate(
    education = factor(education,
                       levels = c("1", "2", "3"),
                       labels = c("Less than high school",
                                  "High school equivalent",
                                  "More than high school"),
                       ordered = TRUE),
    sex = factor(sex,
                 levels = c("1", "2"),
                 labels = c("male", "female"),
                 ordered = TRUE)
  )
```

Joint the table

```
total_df =
  demo_clean_df |>
  left_join(ac_df, by = "seqn")
```

2.2 Age table and plot

Create table of number of men and women in dofferent education levels

```
edu_gender_table =
  total_df  |>
  group_by(education, sex)  |>
  summarize(count = n())  |>
  pivot_wider(names_from = sex, values_from = count, values_fill = 0)

kable(edu_gender_table,
      col.names = c("Education Level", "Men", "Women"),
      caption = "Number of Men and Women in Each Education Category")
```

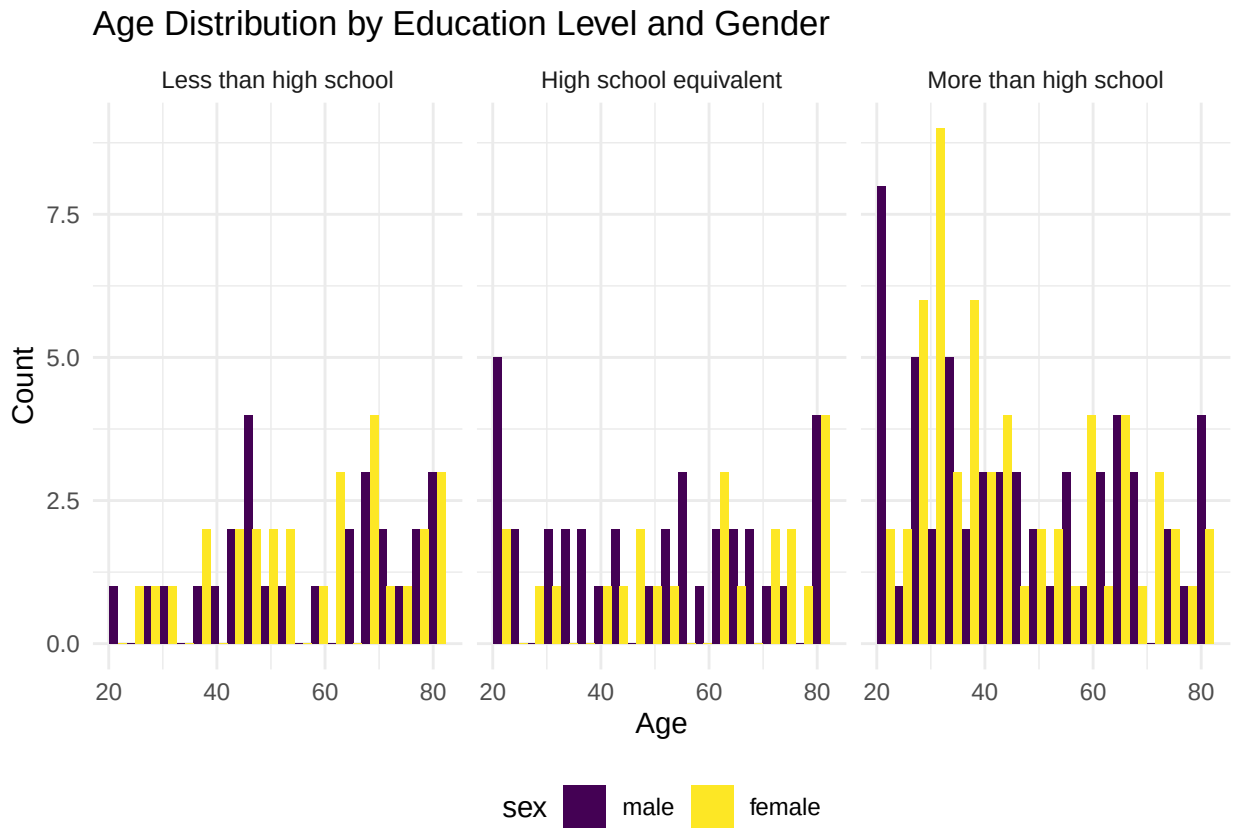
Table 2: Number of Men and Women in Each Education Category

Education Level	Men	Women
Less than high school	27	28
High school equivalent	35	23
More than high school	56	59

Create Age Distribution by Education Level and Gender

```
total_df |>
  ggplot(aes(x = age, fill = sex)) +
  geom_histogram(bins = 20, position = "dodge") +
  facet_wrap(. ~ education) +
```

```
  labs(
    title = "Age Distribution by Education Level and Gender",
    x = "Age",
    y = "Count"
  ) +
  theme_minimal() +
  theme(legend.position = "bottom")
```



* From the table, the highest number of individuals is found in the "More than high school" category for both genders. Notably, in the "High school equivalent" category, the number of women is lower than that of men, while the "Less than high school" category shows a relatively balanced gender distribution.

From the plot:

- Less than high school: The age distribution in this category is relatively uniform, with a close number of men and women, but overall counts are low.
- High school equivalent: This category shows a sparser distribution, with a slightly higher number of men than women, particularly in certain age ranges.
- More than high school: There is a noticeable increase in numbers for ages between 20 and 40, with both men and women displaying similar distribution trends, though men dominate in some specific age ranges.

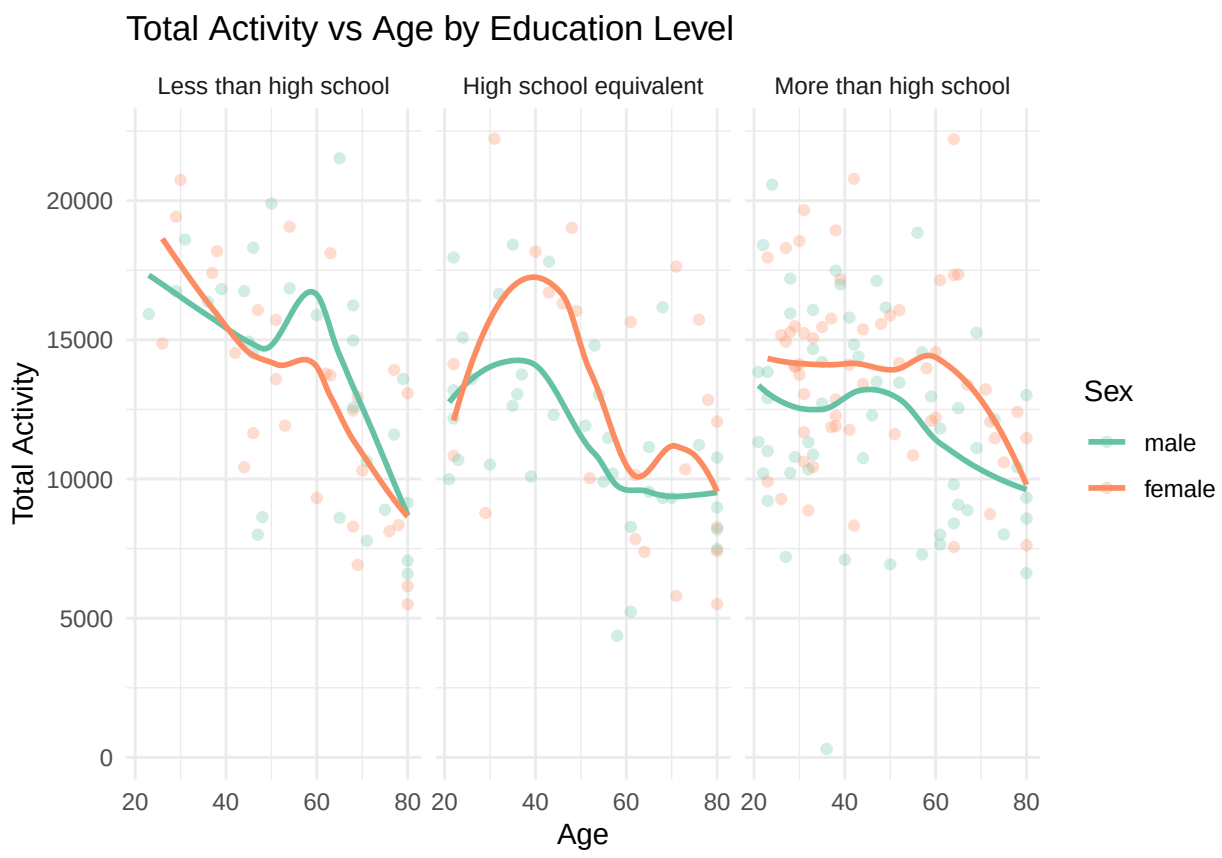
2.3 Activity table and plot

Create the total activity variable

```r
activity_df =
  total_df |>
  mutate(
    total_activity =
      rowSums(select(total_df, starts_with("min")),
              na.rm = TRUE)
  ) |>
  select(seqn, sex, age, bmi, education, total_activity, everything())
```

Plot total activity vs age

```r
activity_df |>
  ggplot(aes(x = age, y = total_activity, color = sex)) +
  geom_point(alpha = .3) +
  geom_smooth(method = "loess", se = FALSE) +
  facet_grid(. ~ education) +
  scale_color_brewer(palette = "Set2") +
  labs(
    title = "Total Activity vs Age by Education Level",
    x = "Age",
    y = "Total Activity",
    color = "Sex"
  ) +
  theme_minimal()
```



The plot suggests that total activity decreases with age for all education levels, with some variations between males and females across different education levels.

2.4 24-Hour Activity Time Course by Education Level and Sex

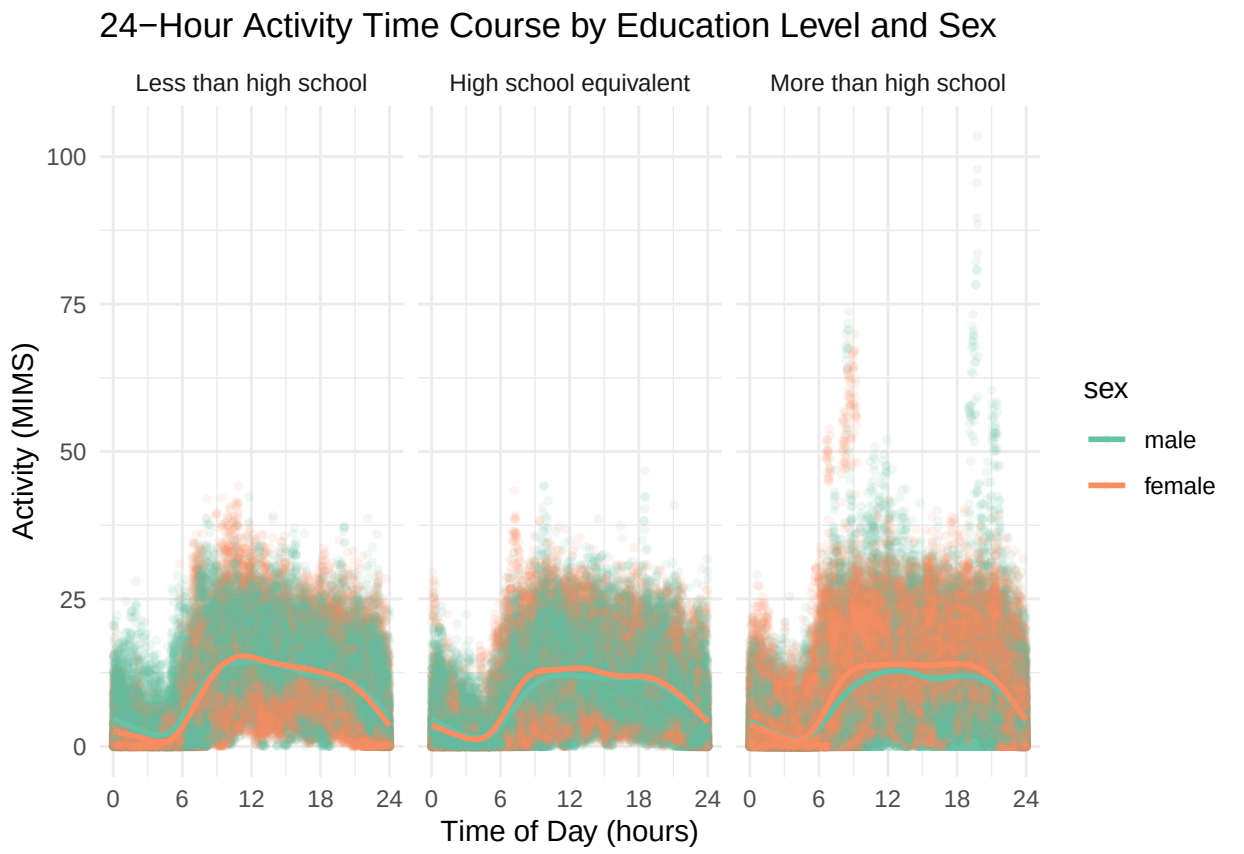
Create long table

```
activity_long_df =
  activity_df |>
  pivot_longer(cols = starts_with("min"), names_to = "minute", values_to = "activity")  |>
  mutate(
    minute = as.numeric(gsub("min", "", minute))
  )
```

24-Hour Activity Time Course by Education Level and Sex

```
activity_p1 =
  activity_long_df |>
  ggplot(aes(x = minute, y = activity, color = sex)) +
  geom_point(alpha = .1, size = 1) +
  geom_smooth(se = FALSE) +
  facet_wrap(~ education) +
  scale_x_continuous(breaks = seq(0, 1440, by = 360), labels = function(x) x / 60) +
  scale_color_brewer(palette = "Set2") +
  labs(
    title = "24-Hour Activity Time Course by Education Level and Sex",
    x = "Time of Day (hours)",
    y = "Activity (MIMS)",
    color = "sex"
  ) +
  theme_minimal()

print(activity_p1)
```



From the plot:

- Most individuals show higher activity levels in the morning and afternoon, with lower activity at night.
- While there are slight differences in the activity patterns between males and females at certain times of the day, the overall trend is similar.
- Education level appears to influence activity patterns, particularly in the "More than high school" group, where females show some activity peaks during the day.

However, P1 has some cons:

- Since all individual data points are displayed, the plot can contain a lot of noise, especially from outliers or extreme values, which might obscure the overall trend.
- the large number of data points can make it difficult to clearly see the overall pattern, as individual variability might overshadow the big picture, especially in densely populated areas of the plot.
- scatter points overlap, making it difficult to accurately assess the overall situation during certain time periods, especially when differences between genders or education levels are subtle.

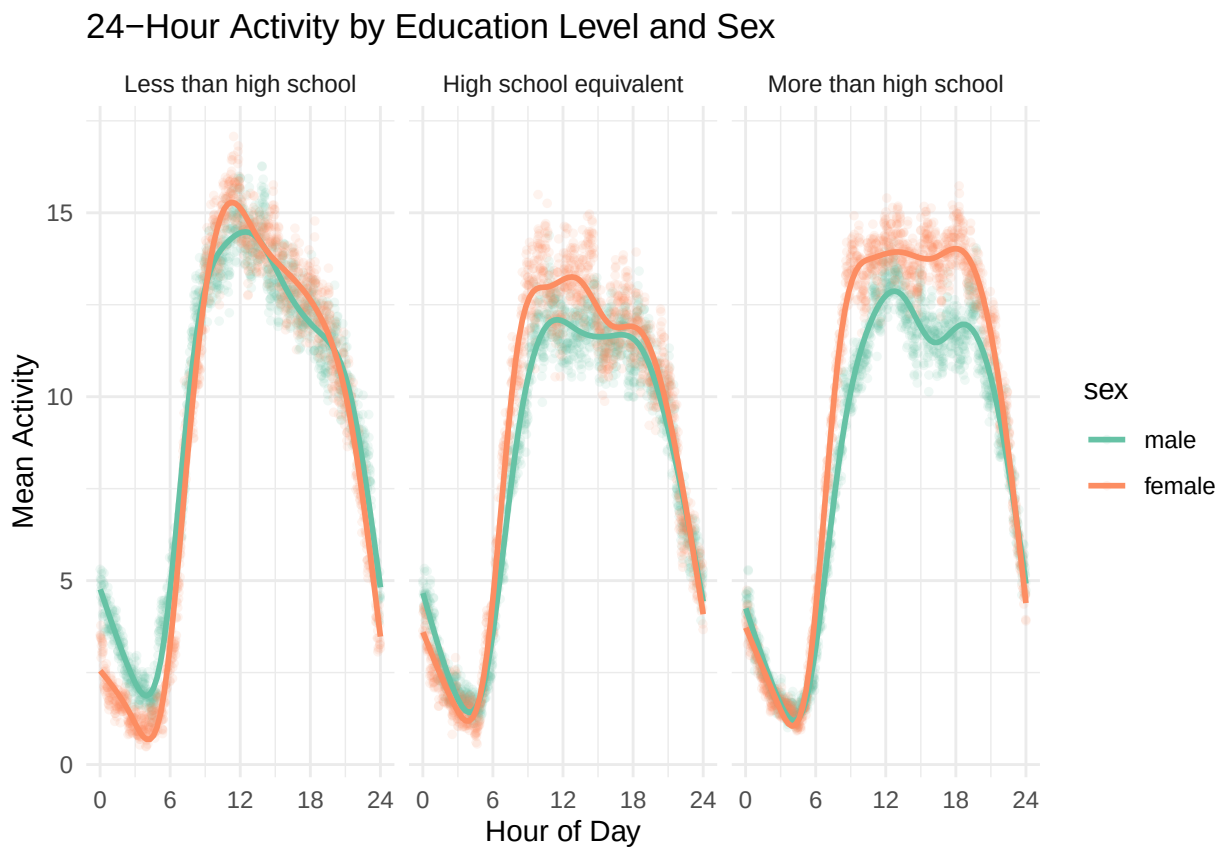
So, I will create a new plot.

24-Hour Mean Activity by Education Level and Sex

```
activity_p2 =
  activity_long_df  |>
  group_by(education, sex, minute)  |>
  summarise(mean_activity = mean(activity, na.rm = TRUE))  |>
  ggplot(aes(x = minute, y = mean_activity, color = sex)) +
```

```r
  geom_point(alpha = .1, size = 1) +
  geom_smooth(se = FALSE) +
  facet_wrap(~education) +
  scale_x_continuous(breaks = seq(0, 1440, by = 360), labels = function(x) x / 60) +
  scale_color_brewer(palette = "Set2") +
  labs(
    title = "24-Hour Activity by Education Level and Sex",
    x = "Hour of Day",
    y = "Mean Activity"
  ) +
  theme_minimal()

activity_p2
```



- Activity peaks in the morning and midday, especially between 7 AM and noon. Afternoon and evening activity decreases, and nighttime activity levels off.
- Females tend to have higher activity levels at certain times of the day, particularly in the "More than high school" group.
- While there are some differences in activity patterns by education level, the overall trends remain quite similar.

Problem 3

3.0 Import the datasets

```
jan20_df =
  read_csv("data/citibike/Jan 2020 Citi.csv", na = c("NA", ".", "")) |>
  janitor::clean_names() |>
  mutate(
    year = 2020,
    month = "January"
  )


july20_df =
  read_csv("data/citibike/July 2020 Citi.csv", na = c("NA", ".", "")) |>
  janitor::clean_names() |>
  mutate(
    year = 2020,
    month = "July"
  )

jan24_df =
  read_csv("data/citibike/Jan 2024 Citi.csv", na = c("NA", ".", "")) |>
   janitor::clean_names() |>
  mutate(
    year = 2024,
    month = "January"
  )

july24_df =
  read_csv("data/citibike/July 2024 Citi.csv", na = c("NA", ".", "")) |>
   janitor::clean_names() |>
  mutate(
    year = 2024,
    month = "July"
  )
```

Combine the data in a total dataframe.

```
citi_total = bind_rows(jan20_df, jan24_df, july20_df, july24_df)

head(citi_total)
str(citi_total)
dim(citi_total)
skimr::skim(citi_total)
```

After cleaning and merging the data, the final dataset consists of 99,485 observations and 9 variables. New variables, “year” and “month,” were created and converted to character type. Key variables in the dataset include “ride_id,” “rideable_type,” “duration,” “member_casual,” among others.

3.1 Ride count table

```
ride_count_table =
  citi_total |>
  drop_na(member_casual) |>
  group_by(year, month, member_casual) |>
  summarise(total_rides = n(), .groups = 'drop') |>
  pivot_wider(names_from = member_casual, values_from = total_rides, values_fill = 0)

kable(ride_count_table, caption = "Total Number of Rides by Year, Month, and User Type")
```

Table 3: Total Number of Rides by Year, Month, and User Type

year	month	casual	member
2020	January	984	11436
2020	July	5637	15411
2024	January	2108	16753
2024	July	10894	36262

- In 2020, both January and July have significantly more rides from members compared to casual riders, with July showing a notable increase in rides for both groups.
- In 2024, the trend continues, with members having more rides than casual riders. However, the gap between casual and member rides seems to narrow compared to 2020, especially in July, where casual rides see a sharp increase.
- In both 2020 and 2024, the number of rides in July is higher than in January for both casual and member users. This is likely due to better weather conditions in summer, leading to more people opting for bike rides.
- Members consistently have more rides than casual users across all months and years, though the difference is more pronounced in 2020. In 2024, casual riders show a significant increase in ride numbers, particularly in July.

3.2 Top 5 stations

```
top_station =
  citi_total |>
  filter( month == "July" & year == 2024) |>
  group_by(start_station_name) |>
  mutate(num_rides = n())|>
  distinct(start_station_name, num_rides) |>
  ungroup() |>
  slice_max(order_by = num_rides, n = 5)


print(top_station)
```

```
## # A tibble: 5 x 2
##   start_station_name       num_rides
```

```
##   <chr>                        <int>
## 1 Pier 61 at Chelsea Piers       163
## 2 University Pl & E 14 St        155
## 3 W 21 St & 6 Ave                152
## 4 West St & Chambers St          150
## 5 W 31 St & 7 Ave                146
```

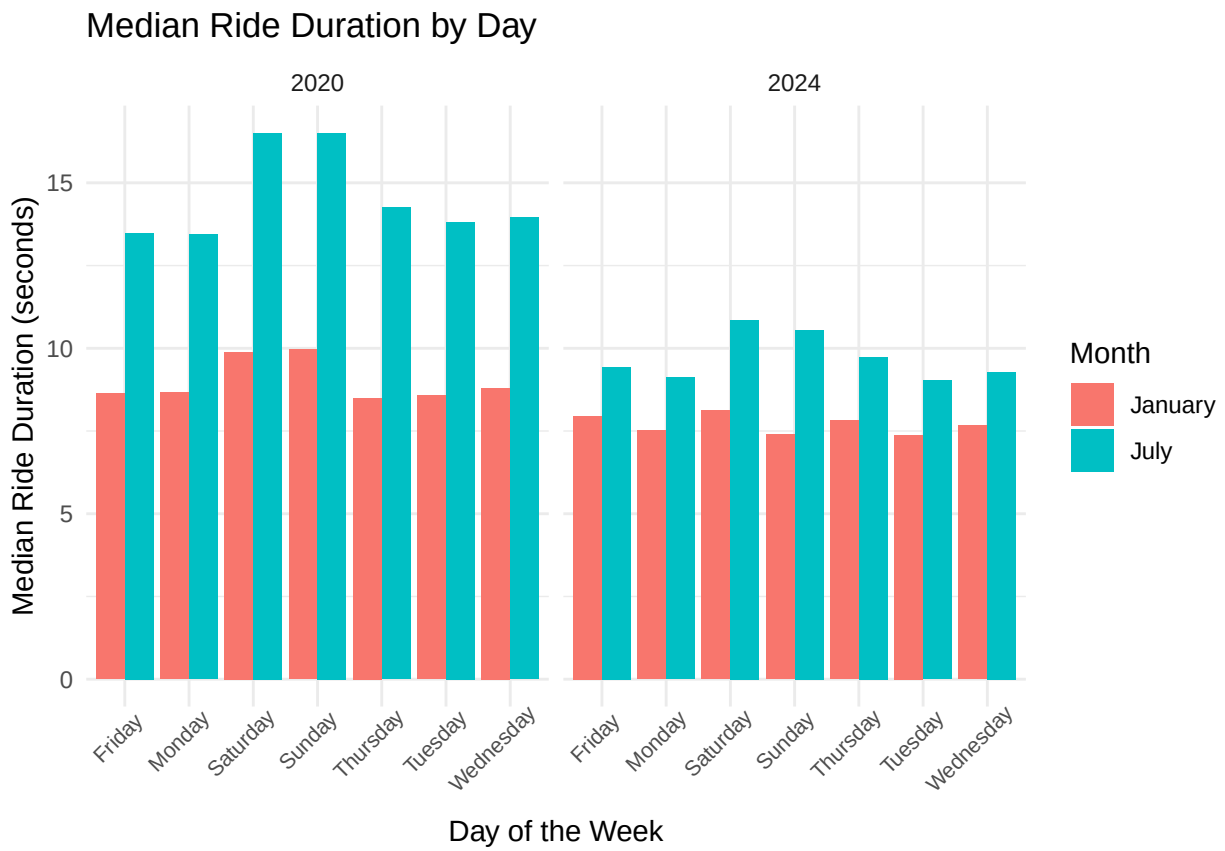
The 5 Most Popular Starting Stations For July 2024.

3.3 Ride durition by day

```
# having ride durition dataframe
ride_duration = citi_total |>
  group_by(year, month, weekdays) |>
  summarise(median_duration = median(duration, na.rm = TRUE), .groups = 'drop')

# making plot of Median Ride Duration by Day of the Week and Month
ride_duration_plot =
  ride_duration |>
  ggplot(aes(x = weekdays, y = median_duration, fill = month)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ year) +
  labs(title = "Median Ride Duration by Day",
       x = "Day of the Week",
       y = "Median Ride Duration (seconds)",
       fill = "Month") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 0.8, size = 8))

print(ride_duration_plot)
```



The plot shows that:

- Ride durations in July (blue bars) are generally longer than in January (red bars) for most days of the week. This could be due to the warmer weather encouraging longer rides, while colder months may deter long trips.
- In 2020, median ride durations for some days (e.g., Saturday and Sunday) are noticeably higher compared to 2024. This may reflect the impact of the pandemic in 2020, as it influenced people's mobility and outdoor activities.
- In both 2020 and 2024, weekend days (e.g., Friday, Saturday, Sunday) tend to have longer ride durations than weekdays. This might be because people have more leisure time on weekends, allowing for longer rides.

3.4 Impact of Month, Membership Status, and Bike Type on Ride(2024)

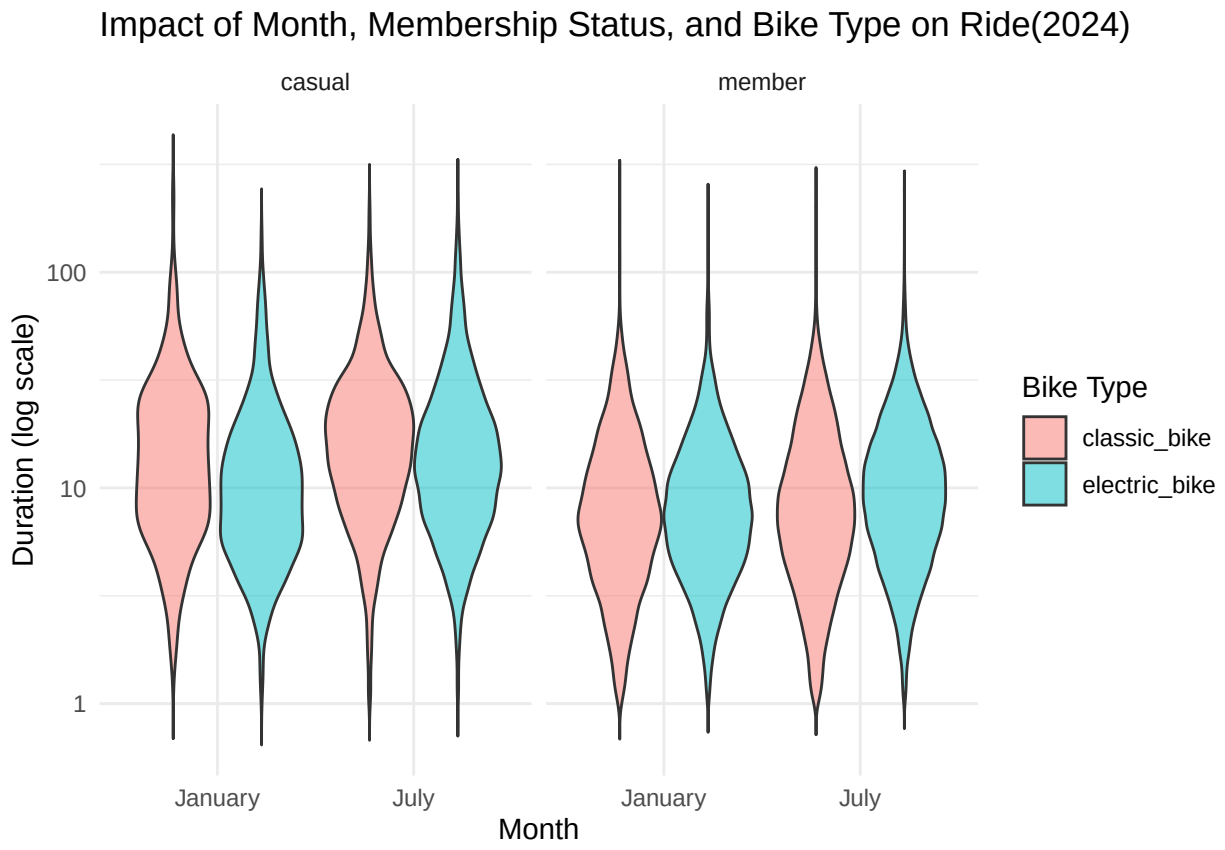
Select data from the total dataset.

```
citi_2024 =
  citi_total |>
  filter(year == 2024)
```

Out put the plot to show the impact of month, membership status, and bike type on duration.

```r
ride_duration_plot =
  citi_2024 |>
  ggplot(aes(x = month, y = duration, fill = rideable_type)) +
  geom_violin(alpha = .5, trim = FALSE) +
  scale_y_log10() +
  facet_grid(.~ member_casual) +
  labs(
    title = "Impact of Month, Membership Status, and Bike Type on Ride(2024)",
    x = "Month",
    y = "Duration (log scale)",
    fill = "Bike Type"
  ) +
  theme_minimal()

print(ride_duration_plot)
```



The plot shows that:

- For casual users, the duration of rides on traditional bikes is consistently higher than on electric bikes, regardless of the month. However, it's noteworthy that ride durations in July are significantly longer than those in January.
- Member users show no notable difference in bike type choice during January, but in July, they tend to have longer ride durations on electric bikes compared to traditional ones. This suggests a seasonal preference for electric bikes during the summer months for longer rides.

- Casual users tend to have longer ride durations compared to member users. This could be because casual users may opt for Citi Bikes for longer trips, whereas shorter distances might be covered by walking or other means.