Final Project Preliminary Presentation

Loading libraries and reading in data for subways and buses

```
library(dplyr)
library(tidyr)
library(readr)
library(lubridate)
library(readxl)
library(ggplot2)
library(knitr)
```

```
url1 <- "MTA_Daily_Ridership_Data__Beginning_2020.csv"
df <- read_csv(url1)

colnames(df) %>% kable()
```

x
Date
Subways: Total Estimated Ridership
Subways: % of Comparable Pre-Pandemic Day
Buses: Total Estimated Ridership
Buses: % of Comparable Pre-Pandemic Day
LIRR: Total Estimated Ridership
LIRR: % of Comparable Pre-Pandemic Day
Metro-North: Total Estimated Ridership
Metro-North: % of Comparable Pre-Pandemic Day
Access-A-Ride: Total Scheduled Trips
Access-A-Ride: % of Comparable Pre-Pandemic Day
Bridges and Tunnels: Total Traffic
Bridges and Tunnels: % of Comparable Pre-Pandemic Day
Staten Island Railway: Total Estimated Ridership

x
Staten Island Railway: % of Comparable Pre-Pandemic Day

Creating variable specific data frames

```r
# subways data frame
sub_df <-
  df %>%
  # selecting relevant variables
  select('Date', 'Subways: Total Estimated Ridership',
         'Subways: % of Comparable Pre-Pandemic Day') %>%
  na.omit %>%
  # filtering out any dates in the years 2020 and 2023
  filter(!grepl("2023$", Date),
         !grepl("2020$", Date)) %>%
  # mutating date to convert is from a "char" data type
  # creating a new variable that assigns each date their proper day of the week
  mutate("Date" = mdy(Date),
         "Day of Week" = weekdays(Date)) %>%
  select('Day of Week', 'Date', 'Subways: Total Estimated Ridership',
         'Subways: % of Comparable Pre-Pandemic Day')
colnames(sub_df) %>% kable()
```

x
Day of Week
Date
Subways: Total Estimated Ridership
Subways: % of Comparable Pre-Pandemic Day

```r
# buses data frame
bus_df <-
  df %>%
  # selecting relevant variables
  select('Date', 'Buses: Total Estimated Ridership',
         'Buses: % of Comparable Pre-Pandemic Day') %>%
  na.omit() %>%
  # filtering out any dates in the years 2020 and 2023
  filter(!grepl("2023$", Date),
         !grepl("2020$", Date)) %>%
  # mutating date to convert is from a "char" data type
  # creating a new variable that assigns each date their proper day of the week
  mutate("Date" = mdy(Date),
         "Day of Week" = weekdays(Date)) %>%
  select('Day of Week', 'Date', 'Buses: Total Estimated Ridership',
         'Buses: % of Comparable Pre-Pandemic Day')
colnames(bus_df) %>% kable()
```

x
Day of Week
Date
Buses: Total Estimated Ridership
Buses: % of Comparable Pre-Pandemic Day

Reading in data for weather

```
url2 <- "weather_nyc_2021_2022.xlsx"
weather_df <- read_excel(url2)

colnames(weather_df) %>% kable()
```

x
name
datetime
tempmax
tempmin
temp
feelslikemax
feelslikemin
feelslike
dew
humidity
precip
precipprob
precipcover
preciptype
snow
snowdepth
windgust

x
windspeed
winddir
sealevelpressure
cloudcover
visibility
solarradiation
solarenergy
uvindex
severerisk
sunrise
sunset
moonphase
conditions
description
icon
stations

Data wrangling weather data frame

```r
weather_df <-
  weather_df %>%
  select('datetime', 'tempmax', 'tempmin', 'temp', 'precip', 'snow',
         'snowdepth', 'windspeed', 'conditions', 'icon')
colnames(weather_df) %>% kable()
```

x
datetime
tempmax
tempmin
temp
precip
snow
snowdepth
windspeed
conditions
icon

Joining the weather data frame with both the bus and subway data frame

```
sub_df <-
  sub_df %>%
  # joining by the 'date' and 'datetime' variables
  full_join(weather_df, by = c("Date" = "datetime")) %>%
    mutate(Date = as.Date(Date))
head(sub_df)
```

```
# A tibble: 6 x 13
  Day of~1 Date       Subwa~2 Subwa~3 tempmax tempmin  temp precip  snow snowd~4
  <chr>    <date>       <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl> <dbl>   <dbl>
1 Saturday 2022-12-31 1927101    0.58    54.8    48.7  51    0.272     0       0
2 Friday   2022-12-30 3063480    0.57    60.9    46.6  51.4  0         0       0
3 Thursday 2022-12-29 3039432    0.57    50      40.2  45.3  0         0       0
4 Wednesd~ 2022-12-28 2947028    0.55    47      33.8  40.1  0         0       0
5 Tuesday  2022-12-27 2729514    0.51    34.5    29    31.6  0         0       0
6 Monday   2022-12-26 1812842    0.71    29      18.5  24.2  0         0       0
# ... with 3 more variables: windspeed <dbl>, conditions <chr>, icon <chr>, and
#   abbreviated variable names 1: `Day of Week`,
#   2: `Subways: Total Estimated Ridership`,
#   3: `Subways: % of Comparable Pre-Pandemic Day`, 4: snowdepth
```

```
bus_df <-
  bus_df %>%
  # joining by the 'date' and 'datetime' variables
  full_join(weather_df, by = c("Date" = "datetime")) %>%
  mutate(Date = as.Date(Date))
head(bus_df)
```

```
# A tibble: 6 x 13
  Day of~1 Date       Buses~2 Buses~3 tempmax tempmin  temp precip  snow snowd~4
  <chr>    <date>       <dbl>   <dbl>   <dbl>   <dbl> <dbl>  <dbl> <dbl>   <dbl>
1 Saturday 2022-12-31  651474    0.51    54.8    48.7  51    0.272     0       0
2 Friday   2022-12-30 1087122    0.54    60.9    46.6  51.4  0         0       0
3 Thursday 2022-12-29 1099513    0.55    50      40.2  45.3  0         0       0
4 Wednesd~ 2022-12-28 1088279    0.54    47      33.8  40.1  0         0       0
5 Tuesday  2022-12-27 1027687    0.51    34.5    29    31.6  0         0       0
6 Monday   2022-12-26  644828    0.66    29      18.5  24.2  0         0       0
# ... with 3 more variables: windspeed <dbl>, conditions <chr>, icon <chr>, and
#   abbreviated variable names 1: `Day of Week`,
#   2: `Buses: Total Estimated Ridership`,
#   3: `Buses: % of Comparable Pre-Pandemic Day`, 4: snowdepth
```

EDA

Average Ridership by Day of Week

Subways

```r
avg_sub <-
  sub_df %>%
  group_by(`Day of Week`) %>%
  summarize("Avg Subway Ridership" = mean(`Subways: Total Estimated Ridership`)) %>%
  arrange(desc(`Avg Subway Ridership`))
```

Buses

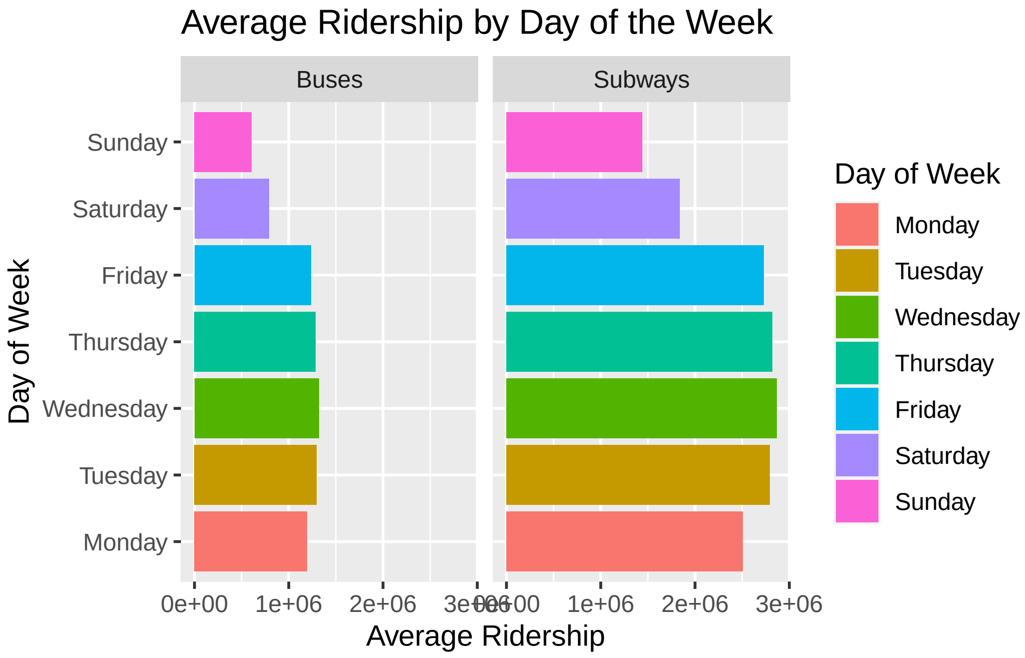
```r
avg_bus <-
  bus_df %>%
  group_by(`Day of Week`) %>%
  summarize("Avg Bus Ridership" = mean(`Buses: Total Estimated Ridership`)) %>%
  arrange(desc(`Avg Bus Ridership`))
```

Joining both dataframes

```r
joint_avg <-
  avg_sub %>%
  full_join(avg_bus) %>%
  rename("Buses" = `Avg Bus Ridership`, "Subways" = `Avg Subway Ridership`) %>%
  pivot_longer(!`Day of Week`,
               names_to = "Category",
               values_to = "Average Ridership")
```

```
Joining, by = "Day of Week"
```

```r
day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
               "Saturday", "Sunday")
joint_avg$`Day of Week` <- factor(joint_avg$`Day of Week`, levels = day_order)

ggplot(joint_avg) +
  geom_bar(aes(x = `Average Ridership`, y = `Day of Week`,
               fill = `Day of Week`), stat = "identity") +
  ggtitle("Average Ridership by Day of the Week") +
  facet_wrap("Category")
```



Total Ridership by Day of Week

Subways

```
total_sub <-
  sub_df %>%
  group_by(`Day of Week`) %>%
  summarize("Total Subway Ridership" = sum(`Subways: Total Estimated Ridership`)) %>%
  arrange(desc(`Total Subway Ridership`))
```

Buses

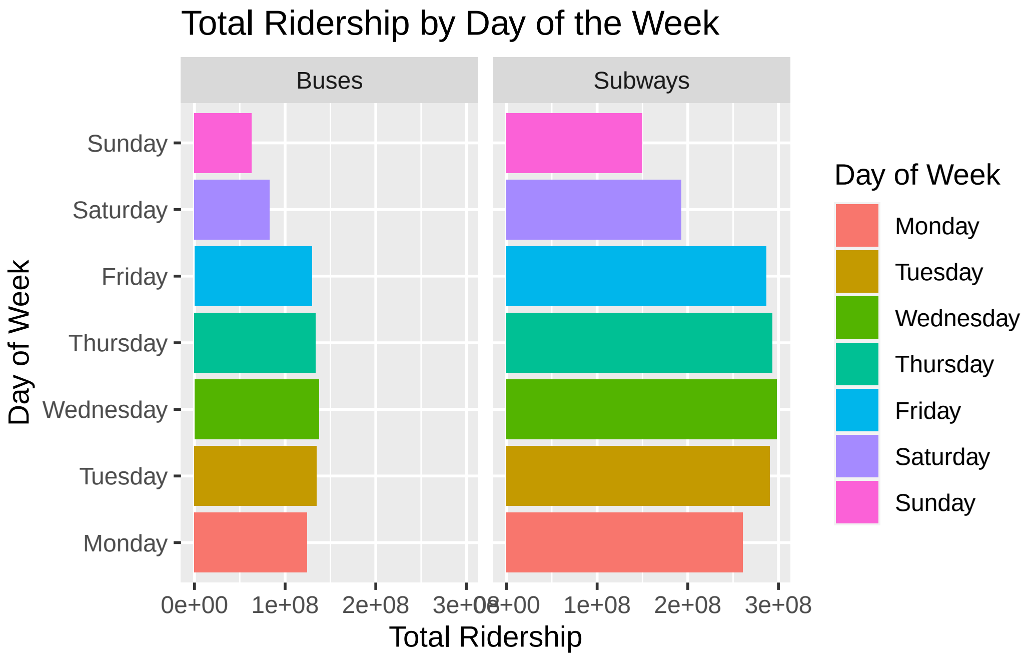
```
total_bus <-
  bus_df %>%
  group_by(`Day of Week`) %>%
  summarize("Total Bus Ridership" = sum(`Buses: Total Estimated Ridership`)) %>%
  arrange(desc(`Total Bus Ridership`))
```

Joining both dataframes

```
joint_total <-
  total_sub %>%
  full_join(total_bus) %>%
  rename("Buses" = `Total Bus Ridership`, "Subways" = `Total Subway Ridership`) %>%
  pivot_longer(!`Day of Week`,
               names_to = "Category",
               values_to = "Total Ridership")
```

```
Joining, by = "Day of Week"
```

```
day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday",
               "Saturday", "Sunday")
joint_total$`Day of Week` <- factor(joint_total$`Day of Week`, levels = day_order)


ggplot(joint_total) +
  geom_bar(aes(x = `Total Ridership`, y = `Day of Week`,
               fill = `Day of Week`), stat = "identity") +
  ggtitle("Total Ridership by Day of the Week") +
  facet_wrap("Category")
```



Data Wrangling

```
# making all the NA values ' '
weather_df$conditions <- ifelse(is.na(weather_df$conditions), "", weather_df$conditions)
weather_df$icon <- ifelse(is.na(weather_df$icon), "", weather_df$icon)

# getting rid of 2020 and 2023 dates as we don't want to include those
df <- df %>%
    filter(!grepl("2023$", Date),
          !grepl("2020$", Date))

# changing column names for simplicity
colnames(df) <- c('Date', 'Subway', 'Subway%', 'Buses', 'Buses%', 'LIRR', 'LIRR%', 'Metro-

# mutating format of date so it can be joined
weather_df <- weather_df %>%
  mutate(datetime = ymd(datetime)) %>%
  mutate(datetime = format(datetime, "%m/%d/%Y"))

# joining weather and transportation df
combined <- weather_df %>%
  full_join(df, by = c('datetime' = 'Date'))
```

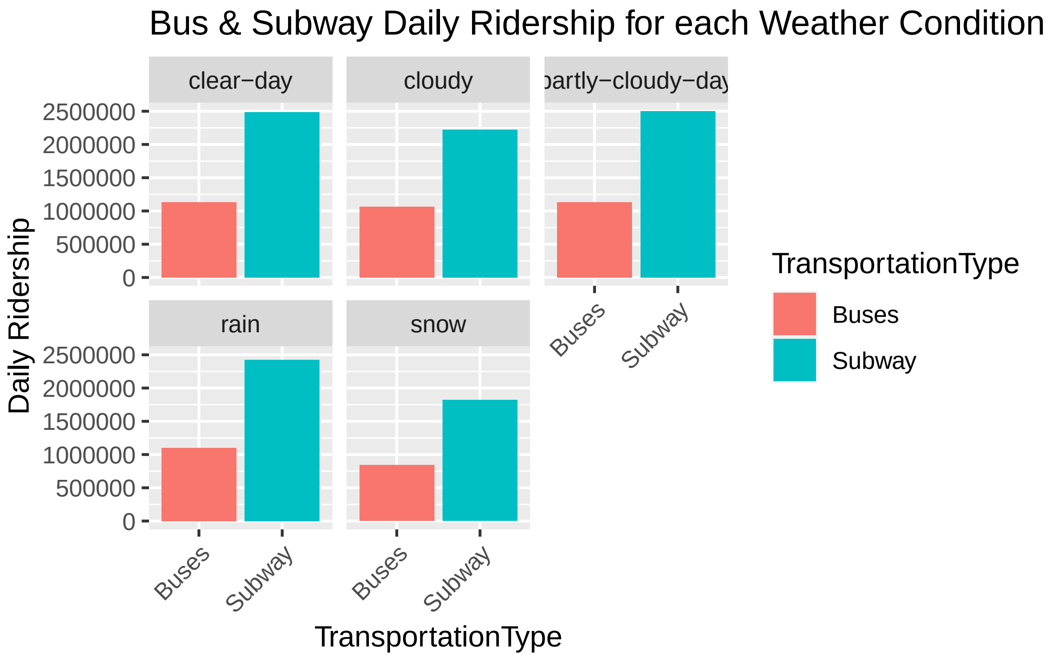
```
# creating table that has Subway ridership per day by icon
SubwayByIcon <- combined %>%
  select(icon, Subway, datetime) %>%
  group_by(icon) %>%
  summarize(RidershipPerIcon = sum(Subway)/n())
```

```
# creating new table that has transportation type so now we can group by type
TransportByIcon <- combined %>%
  select(datetime, Subway, Buses, LIRR, 'Metro-North', AARide, 'Bridges&Tunnels', SIRR, ic
  pivot_longer(cols = c('Subway', 'Buses', 'LIRR', 'Metro-North', 'AARide', 'Bridges&Tunne
  group_by(icon,TransportationType) %>%
  na.omit() %>%
  summarize(Daily = sum(DailyRidership)/n())
```

```
`summarise()` has grouped output by 'icon'. You can override using the
`.groups` argument.
```

Bus and Subway Ridership by Weather Condition

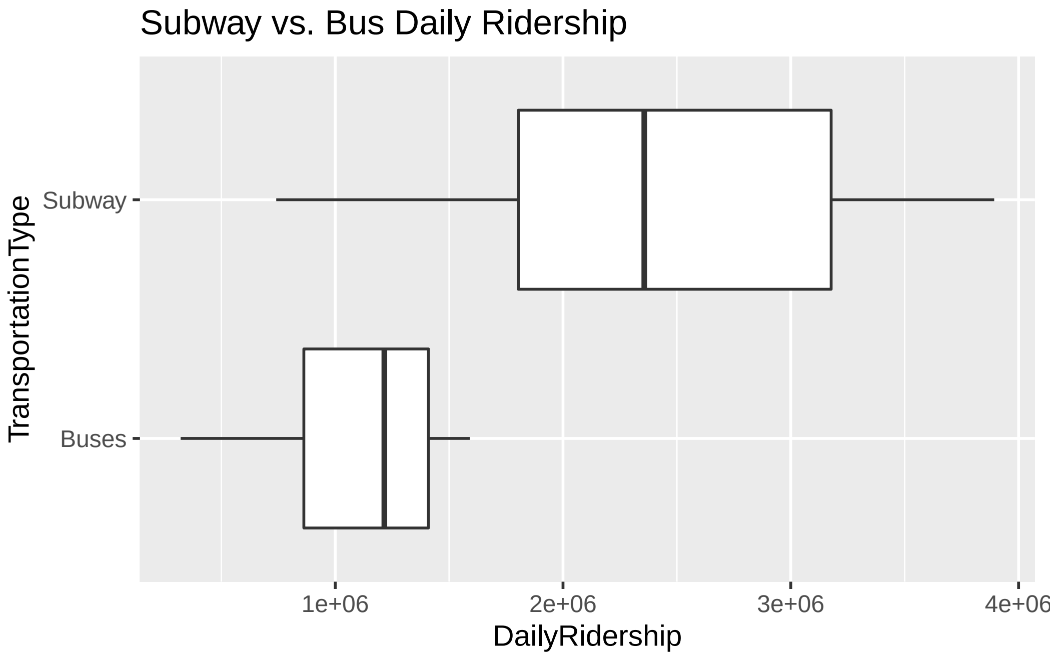
```
# just doing buses vs subways as they are the most popular so we can see diff better
TransportByIcon %>%
  filter(TransportationType == 'Buses'| TransportationType == 'Subway') %>%
  group_by(icon, TransportationType) %>%
  ggplot() +
  geom_bar(aes(x = TransportationType, y = Daily, fill = TransportationType), stat = 'iden
  facet_wrap(~icon) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ylab('Daily Ridership') +
  ggtitle('Bus & Subway Daily Ridership for each Weather Condition')
```



Subway and Bus Daily Ridership Boxplot

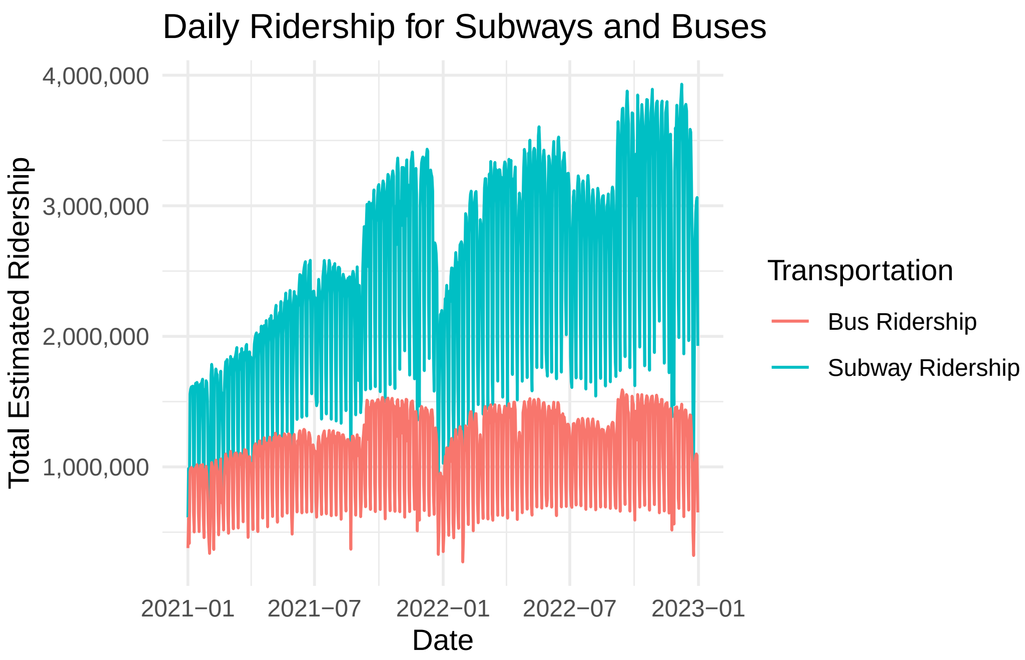
```
longTable <- combined %>%
  select(datetime, Subway, Buses, snowdepth, tempmax, tempmin, temp, precip, snow, windspe
  pivot_longer(cols = c('Subway', 'Buses'), names_to = 'TransportationType', values_to = '
```

```r
# on a clear day, what transportation method is used most (not considering Subway)
longTable %>%
  select(DailyRidership, TransportationType, conditions) %>%
  filter(TransportationType %in% c('Buses', 'Subway')) %>%
  filter(conditions == 'Clear') %>%
  ggplot() +
  geom_boxplot(aes(x = DailyRidership, y = TransportationType)) +
  ggtitle("Subway vs. Bus Daily Ridership")
```



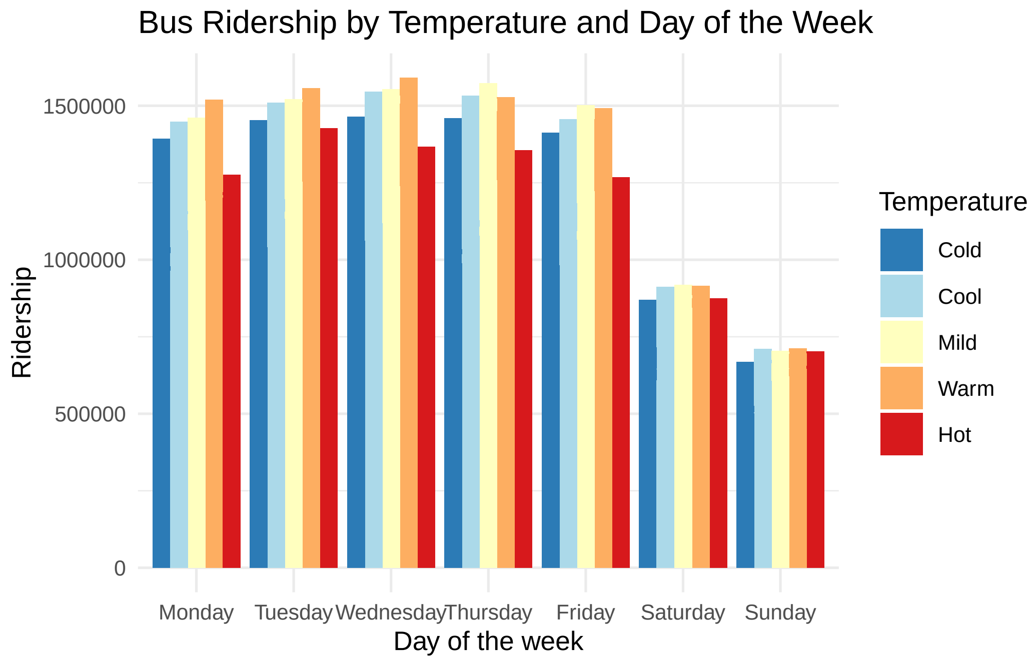
Ridership for Subways and Buses over the Years

```r
ggplot() +
  geom_line(data = sub_df, aes(x = Date, y = `Subways: Total Estimated Ridership`, color =
  geom_line(data = bus_df, aes(x = Date, y = `Buses: Total Estimated Ridership`, color = "
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Daily Ridership for Subways and Buses",
       x = "Date",
       y = "Total Estimated Ridership",
       color = "Transportation") +
  theme_minimal()
```

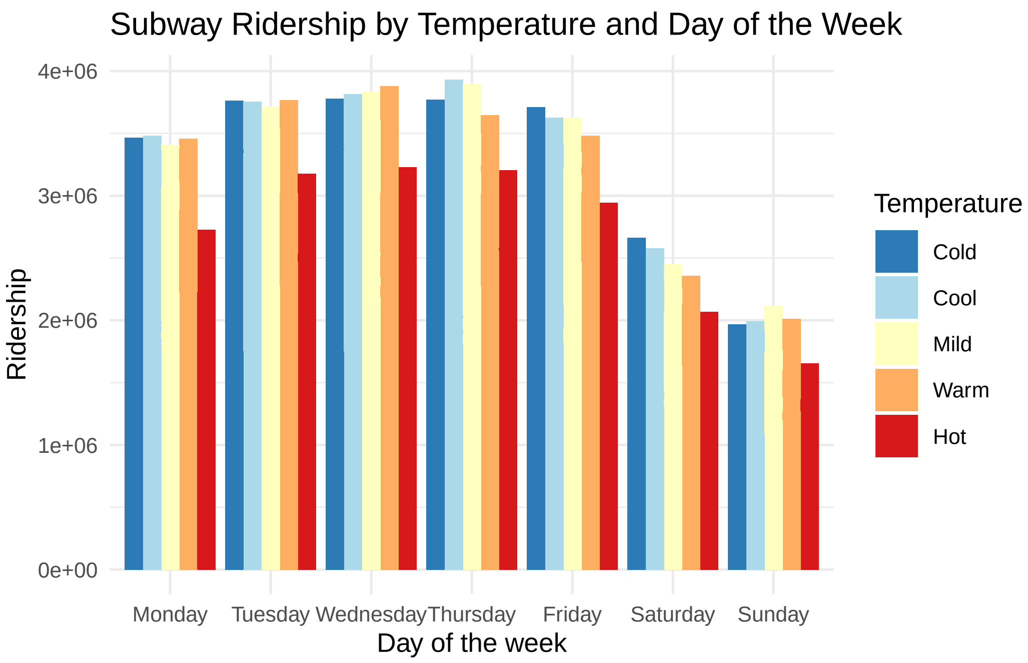


Ridership by Temperature and Day of the Week

```
bus_df$temperature <- cut(sub_df$temp,
                          breaks = c(-Inf, 40, 55, 70, 80, Inf),
                          labels = c("Cold", "Cool", "Mild", "Warm", "Hot"))

day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday
bus_df$`Day of Week` <- factor(bus_df$`Day of Week`, levels = day_order)


sub_df$temperature <- cut(sub_df$temp,
                          breaks = c(-Inf, 40, 55, 70, 80, Inf),
                          labels = c("Cold", "Cool", "Mild", "Warm", "Hot"))

sub_df$`Day of Week` <- factor(sub_df$`Day of Week`, levels = day_order)

#MAIN PLOTS

ggplot(bus_df, aes(x = `Day of Week`, y = `Buses: Total Estimated Ridership`, fill = tempe
  geom_bar(position = "dodge", stat = "identity") +
  scale_fill_brewer(palette = "RdYlBu", direction = -1) +
  labs(x = "Day of the week", y = "Ridership", fill = "Temperature") +
  ggtitle("Bus Ridership by Temperature and Day of the Week") +
  theme_minimal(base_size = 10)
```



```
ggplot(sub_df, aes(x = `Day of Week`, y = `Subways: Total Estimated Ridership`, fill = tem
  geom_bar(position = "dodge", stat = "identity") +
  scale_fill_brewer(palette = "RdYlBu", direction = -1) +
  labs(x = "Day of the week", y = "Ridership", fill = "Temperature") +
  ggtitle("Subway Ridership by Temperature and Day of the Week") +
  theme_minimal(base_size = 10)
```


Subway Ridership by Temperature and Day of the Week
Ridership
4e+06
3e+06
2e+06
1e+06
0e+00
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Sunday
Day of the week
Temperature
Cold
Cool
Mild
Warm
Hot