Cyclistic

# Helpful Packages

library(tidyverse) #Helps Clean Data

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --  
## v ggplot2 3.3.5 v purrr 0.3.4  
## v tibble 3.1.6 v dplyr 1.0.9  
## 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)#Helps Clean Data

##   
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':  
##   
## date, intersect, setdiff, union

library(skimr) #Summary Data

library(janitor)

##   
## Attaching package: 'janitor'

## The following objects are masked from 'package:stats':  
##   
## chisq.test, fisher.test

# Step 1: Collecting Data

April <- read\_csv("April 2022 Cycle Data (version 1).csv")

## Rows: 284040 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

August <- read\_csv("August 2021 Cycle data.csv")

## Rows: 822397 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

December <- read\_csv("December 2021 Cycle Data.csv")

## Rows: 359925 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

February <- read\_csv("February 2022 Cycle data.csv")

## Rows: 103770 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

January <- read\_csv("Janurary 2022 Cycle Data.csv")

## Rows: 247540 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

July <- read\_csv("July 2021 Cycle data.csv")

## Rows: 729590 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

June <- read\_csv("June 2021 Cycle data.csv")

## Rows: 531631 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

May <- read\_csv("May 2021 Cycle Data.csv")

## Rows: 337225 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

November <- read\_csv("November 2021 Cycle Data.csv")

## Rows: 523189 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

October <- read\_csv("October 2021 Cycle Data.csv")

## Rows: 756111 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

September <- read\_csv("September 2021 Cycle Data.csv")

## Rows: 804323 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

March <- read\_csv("March 2022 Cycle Data.csv")

## Rows: 115609 Columns: 15  
## -- Column specification --------------------------------------------------------  
## Delimiter: ","  
## chr (10): ride\_id, rideable\_type, started\_at, ended\_at, day\_of\_week, start\_...  
## dbl (4): start\_lat, start\_lng, end\_lat, end\_lng  
## time (1): trip\_duration  
##   
## 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.

all\_trips <- bind\_rows(April,August,December,March,February,January,July,June,May,November,October,September)

# Step 2: Cleaning the Data

# Removing any Data with a trip duration < or = to 0   
all\_trips\_v2 <- all\_trips[!(all\_trips$start\_station\_name == "HQ QR" | all\_trips$trip\_duration<0),]  
all\_trips\_v2 <- all\_trips[!(all\_trips$start\_station\_name == "HQ QR" | all\_trips$trip\_duration==0),]

# Step 3: Transforming Data

#Transforming trip duration from String to Numeric  
is.factor(all\_trips$trip\_duration)

## [1] FALSE

all\_trips$trip\_duration <- as.numeric(as.character(all\_trips$trip\_duration))

## Warning: NAs introduced by coercion

is.numeric(all\_trips$trip\_duration)

## [1] TRUE

# Creating Date Fields for Analysis  
all\_trips$date <- as.Date(all\_trips$started\_at, "%m/%d/%y")  
all\_trips$month <- format(as.Date(all\_trips$date), "%m")  
all\_trips$day <- format(as.Date(all\_trips$date), "%d")  
all\_trips$year <- format(as.Date(all\_trips$date), "%Y")  
all\_trips$day\_of\_week <- format(as.Date(all\_trips$date),"%A")

# Step 4: Analyze the Data

## Descriptive Analysis   
mean(all\_trips\_v2$trip\_duration)

## 1161.516 secs

median(all\_trips\_v2$trip\_duration)

## 707 secs

max(all\_trips\_v2$trip\_duration)

## 86397 secs

min(all\_trips\_v2$trip\_duration)

## 1 secs

# Compare members and casual users  
aggregate(all\_trips\_v2$trip\_duration ~ all\_trips\_v2$member\_casual, FUN = mean)

## all\_trips\_v2$member\_casual all\_trips\_v2$trip\_duration  
## 1 casual 1687.9794 secs  
## 2 member 792.5973 secs

aggregate(all\_trips\_v2$trip\_duration ~ all\_trips\_v2$member\_casual, FUN = median)

## all\_trips\_v2$member\_casual all\_trips\_v2$trip\_duration  
## 1 casual 976 secs  
## 2 member 570 secs

aggregate(all\_trips\_v2$trip\_duration ~ all\_trips\_v2$member\_casual, FUN = max)

## all\_trips\_v2$member\_casual all\_trips\_v2$trip\_duration  
## 1 casual 86397 secs  
## 2 member 86397 secs

aggregate(all\_trips\_v2$trip\_duration ~ all\_trips\_v2$member\_casual, FUN = min)

## all\_trips\_v2$member\_casual all\_trips\_v2$trip\_duration  
## 1 casual 1 secs  
## 2 member 1 secs

# See the average ride time by each day for members vs casual users  
all\_trips\_v2$day\_of\_week <- ordered(all\_trips\_v2$day\_of\_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))  
aggregate(all\_trips\_v2$trip\_duration ~ all\_trips\_v2$member\_casual + all\_trips\_v2$day\_of\_week, FUN = mean)

## all\_trips\_v2$member\_casual all\_trips\_v2$day\_of\_week  
## 1 casual Sunday  
## 2 member Sunday  
## 3 casual Monday  
## 4 member Monday  
## 5 casual Tuesday  
## 6 member Tuesday  
## 7 casual Wednesday  
## 8 member Wednesday  
## 9 casual Thursday  
## 10 member Thursday  
## 11 casual Friday  
## 12 member Friday  
## 13 casual Saturday  
## 14 member Saturday  
## all\_trips\_v2$trip\_duration  
## 1 1622.5113 secs  
## 2 757.8981 secs  
## 3 1473.2002 secs  
## 4 745.4221 secs  
## 5 1404.7623 secs  
## 6 749.9582 secs  
## 7 1404.5406 secs  
## 8 749.4862 secs  
## 9 1537.3124 secs  
## 10 786.2941 secs  
## 11 1768.8589 secs  
## 12 877.1346 secs  
## 13 1876.3471 secs  
## 14 887.7408 secs

# Analyze ridership data by type and weekday  
all\_trips %>%   
 group\_by(member\_casual, day\_of\_week) %>% #groups by usertype and weekday  
 summarise(number\_of\_rides = n() #calculates the number of rides and average duration   
 ,average\_duration = mean(trip\_duration)) %>% # calculates the average duration  
 arrange(member\_casual, day\_of\_week) # sorts

## `summarise()` has grouped output by 'member\_casual'. You can override using the  
## `.groups` argument.

## # A tibble: 14 x 4  
## # Groups: member\_casual [2]  
## member\_casual day\_of\_week number\_of\_rides average\_duration  
## <chr> <chr> <int> <dbl>  
## 1 casual Sunday 282444 1622.5113 secs  
## 2 casual Monday 274493 1473.2022 secs  
## 3 casual Tuesday 278461 1404.7623 secs  
## 4 casual Wednesday 286045 1404.5406 secs  
## 5 casual Thursday 366260 1537.3124 secs  
## 6 casual Friday 542023 1768.8589 secs  
## 7 casual Saturday 470887 1876.3471 secs  
## 8 member Sunday 433171 757.8981 secs  
## 9 member Monday 478350 745.4221 secs  
## 10 member Tuesday 482703 746.9582 secs   
## 11 member Wednesday 450227 749.4862 secs  
## 12 member Thursday 439401 786.2941 secs  
## 13 member Friday 434681 877.1346 secs  
## 14 member Saturday 396204 887.7408 secs

# Duration and Number of Trips By Membership   
all\_trips %>%   
 group\_by(member\_casual) %>%   
 summarise(number\_of\_rides = n()   
 ,average\_duration = mean(trip\_duration)) %>%   
 arrange(member\_casual)

## # A tibble: 2 x 3  
## member\_casual number\_of\_rides average\_duration  
## <chr> <int> <dbl>  
## 1 casual 2500613 1623.9821 secs  
## 2 member 3114737 790.2332 secs