Cyclistic data analysis 2023-2024

Robert Madigan

2024-08-01

## Introduction

This is my attempt at the captstone project for the 6-month Google Data Analytics course on Coursera. The data sets used are from Cyclistic, a bike-share company in Chicago. The data ranges from July, 2023 to June, 2024. The aim was to investigate how casual riders and annual members use Cyclistic bikes differently.

## Scenario

You are a junior data analyst working in the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company’s future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.

## Ask

Three questions will guide the future marketing program:

1. How do annual members and casual riders use Cyclistic bikes differently?
2. Why would casual riders buy Cyclistic annual memberships?
3. How can Cyclistic use digital media to influence casual riders to become members?

Key tasks:

* Identify the business task: Establish an effective marketing campaign for converting casual users into members by analyzing how customers use “Cyclistic” bikes differently.
* Consider key stakeholders: Cyclistic executives, Markering analytics team, Director of Marketing

Deliverable:

* A clear statement of the business task: To find differences between casual users and members

## Prepare

I will use most recent Cyclistic historical trip data (from July, 2023 to June, 2024), available through Motivate International Inc under this [licence](https://divvybikes.com/data-license-agreement).

Key tasks:

* Download data and store it appropriately.
* Identify how it’s organized.
* Sort and filter the data.
* Determine the credibility of the data.

The data are available via this [link](https://divvy-tripdata.s3.amazonaws.com/index.html). The data is stored in .csv files, with 13 columns: “ride\_id”, “rideable\_type”, “started\_at”, “ended\_at”, “start\_station\_name”, “start\_station\_id”, “end\_station\_name”, “end\_station\_id”, “start\_lat”, “start\_lng”, “end\_lat”, “end\_lng”, “member\_casual”.

The most recent data (July, 2023 to June, 2024) is used and will be sorted and filtered in the code segments that follow.

The data is credible as it has been made available by Motivate International Inc. However, riders’ personally identifiable information has been omitted for privacy reasons. Thus, I am unable to verify whether a single user/rider has taken several rides, as all ride ids are unique in this data-set.

Collect and combine data into a single data frame

# Load necessary packages  
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──  
## ✔ dplyr 1.1.4 ✔ readr 2.1.5  
## ✔ forcats 1.0.0 ✔ stringr 1.5.1  
## ✔ ggplot2 3.5.1 ✔ tibble 3.2.1  
## ✔ lubridate 1.9.3 ✔ tidyr 1.3.1  
## ✔ purrr 1.0.2   
## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(janitor)

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

library(ggmap)

## ℹ Google's Terms of Service: <https://mapsplatform.google.com>  
## Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>  
## OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>  
## ℹ Please cite ggmap if you use it! Use `citation("ggmap")` for details.

library(geosphere)  
library(lubridate)  
library(leaflet)  
  
# Importing all relevant datasets  
file\_names <- c("202307-divvy-tripdata.csv", "202308-divvy-tripdata.csv", "202309-divvy-tripdata.csv",  
 "202310-divvy-tripdata.csv", "202311-divvy-tripdata.csv", "202312-divvy-tripdata.csv",  
 "202401-divvy-tripdata.csv", "202402-divvy-tripdata.csv", "202403-divvy-tripdata.csv",  
 "202404-divvy-tripdata.csv", "202405-divvy-tripdata.csv", "202406-divvy-tripdata.csv")  
  
# Reading all data sets into a list  
data\_list <- lapply(file\_names, read.csv)  
  
# Combining all data sets into a single data set  
raw\_data <- do.call(rbind, data\_list)  
  
head(raw\_data)

## ride\_id rideable\_type started\_at ended\_at  
## 1 9340B064F0AEE130 electric\_bike 23/07/2023 20:06 23/07/2023 20:22  
## 2 D1460EE3CE0D8AF8 classic\_bike 23/07/2023 17:05 23/07/2023 17:18  
## 3 DF41BE31B895A25E classic\_bike 23/07/2023 10:14 23/07/2023 10:24  
## 4 9624A293749EF703 electric\_bike 21/07/2023 08:27 21/07/2023 08:32  
## 5 2F68A6A4CDB4C99A classic\_bike 08/07/2023 15:46 08/07/2023 15:58  
## 6 9AEE973E6B941A9C classic\_bike 10/07/2023 08:44 10/07/2023 08:49  
## start\_station\_name start\_station\_id end\_station\_name  
## 1 Kedzie Ave & 110th St 20204 Public Rack - Racine Ave & 109th Pl  
## 2 Western Ave & Walton St KA1504000103 Milwaukee Ave & Grand Ave  
## 3 Western Ave & Walton St KA1504000103 Damen Ave & Pierce Ave  
## 4 Racine Ave & Randolph St 13155 Clinton St & Madison St  
## 5 Clark St & Leland Ave TA1309000014 Montrose Harbor  
## 6 Racine Ave & Randolph St 13155 Sangamon St & Lake St  
## end\_station\_id start\_lat start\_lng end\_lat end\_lng member\_casual  
## 1 877 41.69241 -87.70091 41.69483 -87.65304 member  
## 2 13033 41.89842 -87.68660 41.89158 -87.64838 member  
## 3 TA1305000041 41.89842 -87.68660 41.90940 -87.67769 member  
## 4 TA1305000032 41.88411 -87.65694 41.88275 -87.64119 member  
## 5 TA1308000012 41.96709 -87.66729 41.96398 -87.63818 member  
## 6 TA1306000015 41.88407 -87.65685 41.88578 -87.65102 member  
## ride\_length day\_of\_week  
## 1 00:16:30 1  
## 2 00:13:30 1  
## 3 00:09:36 1  
## 4 00:04:56 6  
## 5 00:11:26 7  
## 6 00:04:54 2

dim(raw\_data)

## [1] 5734381 15

str(raw\_data)

## 'data.frame': 5734381 obs. of 15 variables:  
## $ ride\_id : chr "9340B064F0AEE130" "D1460EE3CE0D8AF8" "DF41BE31B895A25E" "9624A293749EF703" ...  
## $ rideable\_type : chr "electric\_bike" "classic\_bike" "classic\_bike" "electric\_bike" ...  
## $ started\_at : chr "23/07/2023 20:06" "23/07/2023 17:05" "23/07/2023 10:14" "21/07/2023 08:27" ...  
## $ ended\_at : chr "23/07/2023 20:22" "23/07/2023 17:18" "23/07/2023 10:24" "21/07/2023 08:32" ...  
## $ start\_station\_name: chr "Kedzie Ave & 110th St" "Western Ave & Walton St" "Western Ave & Walton St" "Racine Ave & Randolph St" ...  
## $ start\_station\_id : chr "20204" "KA1504000103" "KA1504000103" "13155" ...  
## $ end\_station\_name : chr "Public Rack - Racine Ave & 109th Pl" "Milwaukee Ave & Grand Ave" "Damen Ave & Pierce Ave" "Clinton St & Madison St" ...  
## $ end\_station\_id : chr "877" "13033" "TA1305000041" "TA1305000032" ...  
## $ start\_lat : num 41.7 41.9 41.9 41.9 42 ...  
## $ start\_lng : num -87.7 -87.7 -87.7 -87.7 -87.7 ...  
## $ end\_lat : num 41.7 41.9 41.9 41.9 42 ...  
## $ end\_lng : num -87.7 -87.6 -87.7 -87.6 -87.6 ...  
## $ member\_casual : chr "member" "member" "member" "member" ...  
## $ ride\_length : chr "00:16:30" "00:13:30" "00:09:36" "00:04:56" ...  
## $ day\_of\_week : int 1 1 1 6 7 2 3 6 3 7 ...

## Process

Cleaning and preparing the data for analysis.

Key tasks:

* Check the data for errors.
* Choose your tools.
* Transform the data so you can work with it effectively.
* Document the cleaning process.

# Convert started\_at and ended\_at to POSIXct  
raw\_data <- raw\_data %>%  
 mutate(started\_at = as.POSIXct(started\_at, format = "%d/%m/%Y %H:%M"),  
 ended\_at = as.POSIXct(ended\_at, format = "%d/%m/%Y %H:%M"))  
  
# Adding columns for date, month, year, and day of week.  
trip\_data <- raw\_data %>%   
 mutate(year = format(started\_at, "%Y")) %>% # extract year  
 mutate(month = format(started\_at, "%B")) %>% # extract month  
 mutate(date = format(started\_at, "%d")) %>% # extract date  
 mutate(day\_of\_week = format(started\_at, "%A")) %>% # extract day of week  
 mutate(ride\_length = as.numeric(difftime(ended\_at, started\_at, units = "secs")) / 60) %>% # calculate ride length in minutes  
 mutate(start\_time = format(started\_at, "%H"))  
  
# Convert ride\_length to numeric  
trip\_data <- trip\_data %>%   
 mutate(ride\_length = as.numeric(ride\_length))  
  
# Add ride distance in km  
trip\_data$ride\_distance <- distGeo(matrix(c(trip\_data$start\_lng, trip\_data$start\_lat), ncol = 2), matrix(c(trip\_data$end\_lng, trip\_data$end\_lat), ncol = 2))  
  
trip\_data$ride\_distance <- trip\_data$ride\_distance / 1000 # distance in km  
  
# Remove "bad" data  
# Some entries include when bikes were taken out of docks and checked for quality   
# by Divvy where ride\_length was negative or 'zero'  
clean\_tripdata <- trip\_data[!(trip\_data$ride\_length <= 0),]  
  
# Check the cleaned data frame  
str(clean\_tripdata)

## 'data.frame': 5665293 obs. of 20 variables:  
## $ ride\_id : chr "9340B064F0AEE130" "D1460EE3CE0D8AF8" "DF41BE31B895A25E" "9624A293749EF703" ...  
## $ rideable\_type : chr "electric\_bike" "classic\_bike" "classic\_bike" "electric\_bike" ...  
## $ started\_at : POSIXct, format: "2023-07-23 20:06:00" "2023-07-23 17:05:00" ...  
## $ ended\_at : POSIXct, format: "2023-07-23 20:22:00" "2023-07-23 17:18:00" ...  
## $ start\_station\_name: chr "Kedzie Ave & 110th St" "Western Ave & Walton St" "Western Ave & Walton St" "Racine Ave & Randolph St" ...  
## $ start\_station\_id : chr "20204" "KA1504000103" "KA1504000103" "13155" ...  
## $ end\_station\_name : chr "Public Rack - Racine Ave & 109th Pl" "Milwaukee Ave & Grand Ave" "Damen Ave & Pierce Ave" "Clinton St & Madison St" ...  
## $ end\_station\_id : chr "877" "13033" "TA1305000041" "TA1305000032" ...  
## $ start\_lat : num 41.7 41.9 41.9 41.9 42 ...  
## $ start\_lng : num -87.7 -87.7 -87.7 -87.7 -87.7 ...  
## $ end\_lat : num 41.7 41.9 41.9 41.9 42 ...  
## $ end\_lng : num -87.7 -87.6 -87.7 -87.6 -87.6 ...  
## $ member\_casual : chr "member" "member" "member" "member" ...  
## $ ride\_length : num 16 13 10 5 12 5 7 6 11 12 ...  
## $ day\_of\_week : chr "Sunday" "Sunday" "Sunday" "Friday" ...  
## $ year : chr "2023" "2023" "2023" "2023" ...  
## $ month : chr "July" "July" "July" "July" ...  
## $ date : chr "23" "23" "23" "21" ...  
## $ start\_time : chr "20" "17" "10" "08" ...  
## $ ride\_distance : num 3.99 3.26 1.43 1.32 2.44 ...

# Check summarized details about the cleaned dataset   
summary(clean\_tripdata)

## ride\_id rideable\_type started\_at   
## Length:5665293 Length:5665293 Min. :2023-07-01 00:00:00.00   
## Class :character Class :character 1st Qu.:2023-08-20 13:02:00.00   
## Mode :character Mode :character Median :2023-10-17 12:22:00.00   
## Mean :2023-11-20 22:18:14.07   
## 3rd Qu.:2024-03-07 09:40:00.00   
## Max. :2024-05-31 23:59:00.00   
## NA's :710860   
## ended\_at start\_station\_name start\_station\_id   
## Min. :2023-07-01 00:01:00.00 Length:5665293 Length:5665293   
## 1st Qu.:2023-08-20 13:28:00.00 Class :character Class :character   
## Median :2023-10-17 12:37:00.00 Mode :character Mode :character   
## Mean :2023-11-20 22:36:37.54   
## 3rd Qu.:2024-03-07 09:50:00.00   
## Max. :2024-06-02 00:56:00.00   
## NA's :710860   
## end\_station\_name end\_station\_id start\_lat start\_lng   
## Length:5665293 Length:5665293 Min. :41.6 Min. :-87.9   
## Class :character Class :character 1st Qu.:41.9 1st Qu.:-87.7   
## Mode :character Mode :character Median :41.9 Median :-87.6   
## Mean :41.9 Mean :-87.6   
## 3rd Qu.:41.9 3rd Qu.:-87.6   
## Max. :42.1 Max. :-87.5   
## NA's :710860 NA's :710860   
## end\_lat end\_lng member\_casual ride\_length   
## Min. : 0.0 Min. :-88.0 Length:5665293 Min. : 1.0   
## 1st Qu.:41.9 1st Qu.:-87.7 Class :character 1st Qu.: 6.0   
## Median :41.9 Median :-87.6 Mode :character Median : 10.0   
## Mean :41.9 Mean :-87.6 Mean : 18.4   
## 3rd Qu.:41.9 3rd Qu.:-87.6 3rd Qu.: 17.0   
## Max. :42.2 Max. : 0.0 Max. :98489.0   
## NA's :717655 NA's :717655 NA's :710860   
## day\_of\_week year month date   
## Length:5665293 Length:5665293 Length:5665293 Length:5665293   
## Class :character Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character Mode :character   
##   
##   
##   
##   
## start\_time ride\_distance   
## Length:5665293 Min. : 0.0   
## Class :character 1st Qu.: 0.9   
## Mode :character Median : 1.6   
## Mean : 2.1   
## 3rd Qu.: 2.8   
## Max. :9814.6   
## NA's :717655

## Analyze

Now that the data has been stored and prepared appropriately, it is ready for exploration. Conduct descriptive analysis

Key tasks:

* Aggregate your data so it’s useful and accessible.
* Organize and format your data.
* Perform calculations.
* Identify trends and relationships.

Conduct a descriptive analysis

# descriptive analysis on 'ride\_length'  
# mean = straight average (total ride length / total rides)  
# median = midpoint number of ride length array  
# max = longest ride  
# min = shortest ride  
  
clean\_tripdata %>%   
 summarise(average\_ride\_length = mean(ride\_length, na.rm = TRUE),  
 median\_length = median(ride\_length, na.rm = TRUE),  
 max\_ride\_length = max(ride\_length, na.rm = TRUE),  
 min\_ride\_length = min(ride\_length, na.rm = TRUE))

## average\_ride\_length median\_length max\_ride\_length min\_ride\_length  
## 1 18.39116 10 98489 1

The data shown above depicts data on “ride\_length” from July, 2023 to June, 2024. However, there are strange values for min\_ride\_length and max\_ride\_length.

### Comparing members and casual users

Difference between members and casual riders depending on total rides taken

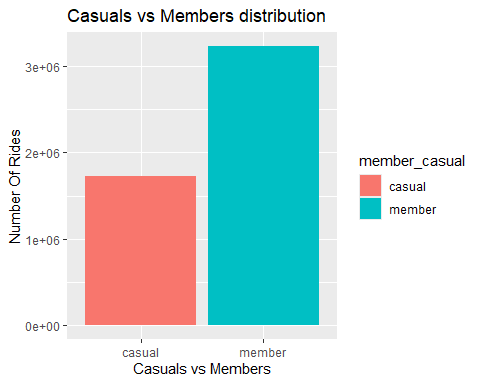
# Need to check member\_casual for NA values  
# Check for NA values in member\_casual  
sum(is.na(clean\_tripdata$member\_casual))

## [1] 710860

# Remove rows with NA values in member\_casual  
clean\_tripdata <- clean\_tripdata %>%  
 filter(!is.na(member\_casual))  
  
# Summarize the data after removing NA values  
summary\_data <- clean\_tripdata %>%   
 group\_by(member\_casual) %>%   
 summarise(ride\_count = length(ride\_id),   
 ride\_percentage = (length(ride\_id) / nrow(clean\_tripdata)) \* 100)  
  
print(summary\_data)

## # A tibble: 2 × 3  
## member\_casual ride\_count ride\_percentage  
## <chr> <int> <dbl>  
## 1 casual 1722723 34.8  
## 2 member 3231710 65.2

# Plot the data  
ggplot(clean\_tripdata, aes(x = member\_casual, fill = member\_casual)) +  
 geom\_bar() +  
 labs(x = "Casuals vs Members", y = "Number Of Rides", title = "Casuals vs Members distribution")



Based on the above graph, there are almost twice as many members than casual users.

Comparison between members and causal riders depending on ride length (mean, median, minimum, maximum)

clean\_tripdata %>%  
 group\_by(member\_casual) %>%   
 summarise(average\_ride\_length = mean(ride\_length), median\_length = median(ride\_length),   
 max\_ride\_length = max(ride\_length), min\_ride\_length = min(ride\_length))

## # A tibble: 2 × 5  
## member\_casual average\_ride\_length median\_length max\_ride\_length  
## <chr> <dbl> <dbl> <dbl>  
## 1 casual 28.4 12 98489  
## 2 member 13.1 9 1560  
## # ℹ 1 more variable: min\_ride\_length <dbl>

The above table indicates that casual users go for longer rides on average than members.

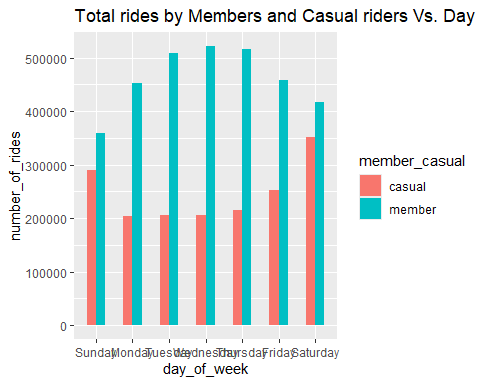
View total rides and average ride time by each day for members vs casual riders

# Fix the days of the week order.  
clean\_tripdata$day\_of\_week <- ordered(clean\_tripdata$day\_of\_week,   
 levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))  
  
clean\_tripdata %>%   
 group\_by(member\_casual, day\_of\_week) %>% #groups by member\_casual  
 summarise(number\_of\_rides = n() #calculates the number of rides and average duration   
 ,average\_ride\_length = mean(ride\_length),.groups="drop") %>% # calculates the average duration  
 arrange(member\_casual, day\_of\_week) #sort

## # A tibble: 14 × 4  
## member\_casual day\_of\_week number\_of\_rides average\_ride\_length  
## <chr> <ord> <int> <dbl>  
## 1 casual Sunday 289268 33.5  
## 2 casual Monday 203685 27.8  
## 3 casual Tuesday 206023 25.4  
## 4 casual Wednesday 205597 24.3  
## 5 casual Thursday 214942 24.7  
## 6 casual Friday 251626 27.2  
## 7 casual Saturday 351582 31.7  
## 8 member Sunday 358505 14.6  
## 9 member Monday 451904 12.6  
## 10 member Tuesday 508427 12.6  
## 11 member Wednesday 521532 12.6  
## 12 member Thursday 516149 12.4  
## 13 member Friday 457824 12.8  
## 14 member Saturday 417369 14.4

Visualize total ride data by type and day of week

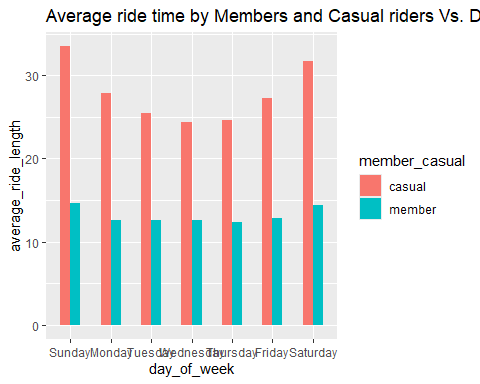
clean\_tripdata %>%   
 group\_by(member\_casual, day\_of\_week) %>%   
 summarise(number\_of\_rides = n(), .groups="drop") %>%   
 arrange(member\_casual, day\_of\_week) %>%   
 ggplot(aes(x = day\_of\_week, y = number\_of\_rides, fill = member\_casual)) +  
 labs(title ="Total rides by Members and Casual riders Vs. Day of the week") +  
 geom\_col(width=0.5, position = position\_dodge(width=0.5)) +  
 scale\_y\_continuous(labels = function(x) format(x, scientific = FALSE))



As shown above, members take more trips during the middle of the week. However, casual users tend to make more trips around the weekend, rising from Friday.

Visualize average ride time data by type and day of week

clean\_tripdata %>%   
 group\_by(member\_casual, day\_of\_week) %>%   
 summarise(average\_ride\_length = mean(ride\_length), .groups="drop") %>%  
 ggplot(aes(x = day\_of\_week, y = average\_ride\_length, fill = member\_casual)) +  
 geom\_col(width=0.5, position = position\_dodge(width=0.5)) +   
 labs(title ="Average ride time by Members and Casual riders Vs. Day of the week")



As shown above, the average ride time tends to be consistent throughout the week for members. For casual users, it tends to be highest around weekends (Saturday/Sunday) and lowest around midweek days (Tuesday/Wednesday/Thursday).

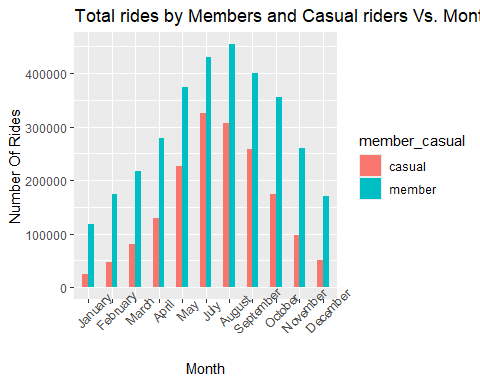
View total rides and average ride time by each month for members vs casual riders

# First lets fix the days of the week order.  
clean\_tripdata$month <- ordered(clean\_tripdata$month,   
 levels=c("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", "December"))  
  
clean\_tripdata %>%   
 group\_by(member\_casual, month) %>%   
 summarise(number\_of\_rides = n(), average\_ride\_length = mean(ride\_length), .groups="drop") %>%   
 arrange(member\_casual, month)

## # A tibble: 22 × 4  
## member\_casual month number\_of\_rides average\_ride\_length  
## <chr> <ord> <int> <dbl>  
## 1 casual January 24065 21.7  
## 2 casual February 46611 25.5  
## 3 casual March 81312 25.3  
## 4 casual April 129474 26.5  
## 5 casual May 226713 28.3  
## 6 casual July 326403 32.8  
## 7 casual August 306949 35.7  
## 8 casual September 258223 25.5  
## 9 casual October 174746 23.2  
## 10 casual November 97181 20.1  
## # ℹ 12 more rows

Visualize total rides data by type and month

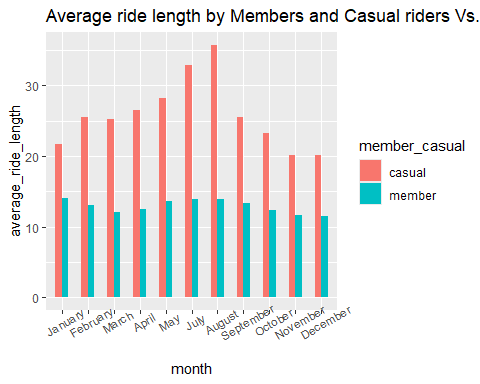
clean\_tripdata %>%   
 group\_by(member\_casual, month) %>%   
 summarise(number\_of\_rides = n(),.groups="drop") %>%   
 arrange(member\_casual, month) %>%   
 ggplot(aes(x = month, y = number\_of\_rides, fill = member\_casual)) +  
 labs(title ="Total rides by Members and Casual riders Vs. Month", x = "Month", y= "Number Of Rides") +  
 theme(axis.text.x = element\_text(angle = 45)) +  
 geom\_col(width=0.5, position = position\_dodge(width=0.5)) +  
 scale\_y\_continuous(labels = function(x) format(x, scientific = FALSE))



The largest number of rides seems to be around Summer time for both casual usersand members, likely because of the better weather. In the colder months, the difference in the number of rides taken by casual users versus those taken by members becomes increasingly large, especially evident for January.

Visualize average ride time data by type and month

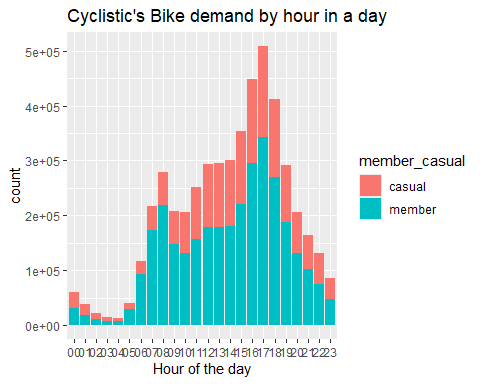
clean\_tripdata %>%   
 group\_by(member\_casual, month) %>%   
 summarise(average\_ride\_length = mean(ride\_length),.groups="drop") %>%  
 ggplot(aes(x = month, y = average\_ride\_length, fill = member\_casual)) +  
 geom\_col(width=0.5, position = position\_dodge(width=0.5)) +   
 labs(title ="Average ride length by Members and Casual riders Vs. Month") +  
 theme(axis.text.x = element\_text(angle = 30))



Average ride length is relatively consistent throughout the year for members. However, casual users see a noticeable increase in average ride length around Summer, peaking in August.

Bike demand by hour in a day

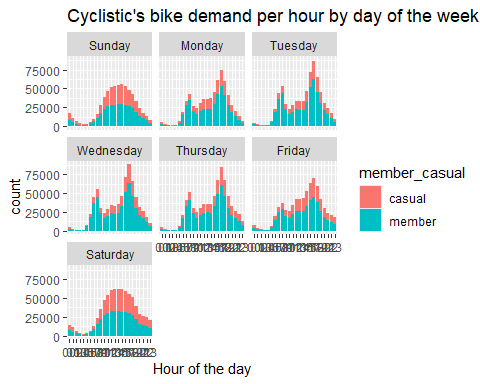
clean\_tripdata %>%  
 ggplot(aes(start\_time, fill= member\_casual)) +  
 labs(x="Hour of the day", title="Cyclistic's Bike demand by hour in a day") +  
 geom\_bar()



There is a similar trend between both members and casual users for bike demand throughout the day. There is a minor peak around 8 am and a larger peak around 5 pm, likely coinciding with commutes to and from work, respectively.

Bike demand per hour by day of the week

clean\_tripdata %>%  
 ggplot(aes(start\_time, fill=member\_casual)) +  
 geom\_bar() +  
 labs(x="Hour of the day", title="Cyclistic's bike demand per hour by day of the week") +  
 facet\_wrap(~ day\_of\_week)



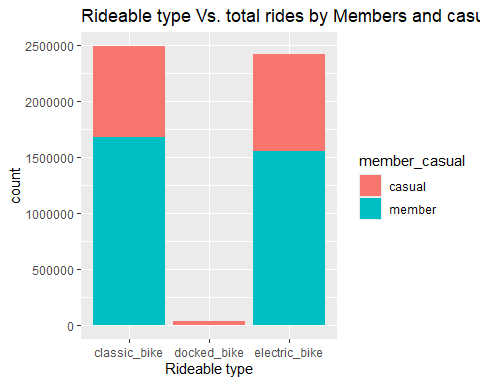
There are notable differences in the patterns of bike demands between weekdays and weekends. On weekends, bike demands steadily increase throughout the day until around 3pm, then steadily drop off for both casual users and members.

Analysis and visualization of rideable type Vs. total rides by members and casual riders

clean\_tripdata %>%  
 group\_by(rideable\_type) %>%   
 summarise(count = length(ride\_id))

## # A tibble: 3 × 2  
## rideable\_type count  
## <chr> <int>  
## 1 classic\_bike 2493934  
## 2 docked\_bike 34205  
## 3 electric\_bike 2426294

ggplot(clean\_tripdata, aes(x=rideable\_type, fill=member\_casual)) +  
 labs(x="Rideable type", title="Rideable type Vs. total rides by Members and casual riders") +  
 geom\_bar()



Classic bikes slightly outperform electric bikes for both casual users and members in terms of total rides. Docked bikes are extremely unpopular, and seem to be favored by casual users.

## Share

This phase should ideally be presented to the stakeholders. However, considering this is merely a capstone project, I will present the findings as a notebook.

Key tasks:

* Determine the best way to share your findings.
* Create effective data visualizations.
* Present your findings.
* Ensure your work is accessible.

Main findings:

* There are almost twice as many members as there are casual users.
* Casual users go for longer rides on average than members.
* Members take more trips during the middle of the week, while casual users tend to make more trips around the weekend.
* Average ride time tends to be consistent throughout the week for members. For casual users, it tends to be highest around weekends (Saturday/Sunday) and lowest around midweek days (Tuesday/Wednesday/Thursday).
* Most activity is during Summer months.
* Average ride length is relatively consistent throughout the year for members. For casual users, it peaks in August.
* Peak bike demand occurs around 8 am and 5 pm on weekdays for both members and casual users.
* Peak bike demand rises steadily from morning until around 3 pm on weekends for both members and casual users, then decreases steadily.
* For both members and casual users, classic bikes are most popular, followed closely by electric bikes. Docked bikes are extremely unpopular.

Possible explanations for these findings:

Both casual users and members likely use bikes to commute to and from work during weekdays, as shown by the increased demand around 8 am and 5 pm, with more rides being taken by members. Colder months seem to show less activity, especially for casual users, likely as conditions become less suitable for cycling. Average trip lengths are noticeably larger for casual users, likely because these customers use the bikes for more leisurely activities such as sightseeing.

## Act

This phase will be conducted by the stakeholders mentioned above.

Deliverable: Your top three recommendations based on your analysis.

Recommendations:

1. Since bike usage peaks during summer months, consider launching summer-specific promotions or campaigns to further boost ridership during this period. Conversely, offer incentives during the off-peak seasons to maintain engagement throughout the year.
2. Increase bike availability during peak times. Perhaps use predictive analytics to forecast demand and optimize bike distribution in real-time.
3. Ensure that there are enough classic and electric bikes available to meet demand. Regularly maintain and update these bikes to ensure high performance and user satisfaction. Consider reducing the number of docked bikes.