Google Capstone Project - Cyclistic Case Study

R Welsh

2024-09-25

The Background

Scenario

You are a junior data analyst working on 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 compeling data insights and professional data visualizations.

The Stakeholders and Teams

Cyclistic: A bike-share program that features more than 5,800 bicycles and 600 docking stations. Cyclistic sets itself apart by also offering reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike. The majority of riders opt for traditional bikes; about 8% of riders use the assistive options. Cyclistic users are more likely to ride for leisure, but about 30% use the bikes to commute to work each day.

Lily Moreno: The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program.These may include email, social media, and other channels.

Cyclistic marketing analytics team: A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic's mission and business goals— aswell as how you, as a junior data analyst, can help Cyclistic achieve them.

Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

Company Overview

In 2016, Cyclistic launched a successful bike-share ordering service. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.

Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the exibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders.Customers who purchase annual memberships are Cyclistic members. Cyclistic's analysts have concluded that annual members are much more profitable than casual riders. Although the pricing exibility helps Cyclistic attract more customers, Moreno believes that maximizing the

number of annual members will be key to future growth. Rather than creating a marketing campaign that targets new customers, Moreno believes there is a solid opportunity to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.

Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

My Role

Moreno has assigned me to see how annual members and casual members differ in their use of the service.

How will I do this

First thing I will need to do is get a dataset that will be used to help my derive analysis to therfore make a report on my findings.

Gathering the datasets required for my analyis

As Cyclistic is a "mock" company there is no first party data so I gathered the datasets required for this project from this link https://divvy-tripdata.s3.amazonaws.com/index.html. This data has been made available from Motivate International Inc Under the license https://divvybikes.com/data-license-agreement The datasets I have selected are the previous 12 months (2023-09-divvy-tripdata - 2024-08-2024-divvy-tripdata) these datasets cover recent data which will give me more insight into the current consumer market which will be more benefical to the company as my recommendations will be based on more reliable data than if I used a past dataset from a previous year i.e 2005. This will also give a better indication to the marketing team for what type of indicators they will need to target.

Preparing and Cleaning the data

After downloading and unzipping the datasets required I then converted all file types from CSV to XLS and then opened each spreadsheet in Excel and checked for duplicated values among the data, No duplicated values were found therefore the remove duplicates function didn't need to be used.

Individual Sheet Analysis

Start of the analysis

I went through all of the datasets and decided to find the mean, the mode, the max and the min of the rides.

I then added a weekday function in order to view the day the rides were made as this could be a way to differentiate how the types of user use the service. I then added a mode function to see what day was the most common.

I then made pivot tables, visualisations and a overall Dashboard for each dataset to find and showcase the difference between Annual and Casual users. The 5 metrics I wanted to view was average ride length between the two members, the average ride length by day, the amount of use per member per day, The most common bike type by member and the average ride length by member per bike type.

The spreadsheets are able to be viewed on my portfolio

Using R to Merge and Produce further analysis

First thing to do when opening R is to open the packages I will need to provide analysis on the data

```
library(tidyverse)
```

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

```
library(readxl)
library(janitor)
```

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

```
library(skimr)
library(flexdashboard)
```

```
library(tinytex)
```

To start my analysis in R I have to import the data so I will change my working directory in R to where the raw csv files are held

```
setwd("C:/Documents/Google Capstone Project/Cyclistic Case Study")
```

Then I imported the datasets

```
sep_23 <- read.csv("202309-divvy-tripdata.csv")
```

```
oct_23 <- read.csv("202310-divvy-tripdata.csv")
```

```
nov_23 <- read.csv("202311-divvy-tripdata.csv")
```

```
dec_23 <- read.csv("202312-divvy-tripdata.csv")
```

```r
jan_24 <- read.csv("202401-divvy-tripdata.csv")
```

```r
feb_24 <- read.csv("202402-divvy-tripdata.csv")
```

```r
mar_24 <- read.csv("202403-divvy-tripdata.csv")
```

```r
apr_24 <- read.csv("202404-divvy-tripdata.csv")
```

```r
may_24 <- read.csv("202405-divvy-tripdata.csv")
```

```r
jun_24 <- read.csv("202406-divvy-tripdata.csv")
```

```r
jul_24 <- read.csv("202407-divvy-tripdata.csv")
```

```r
aug_24 <- read.csv("202408-divvy-tripdata.csv")
```

Next Steps

View the datasets to see if they've imported correctly

```r
View(sep_23)
```

```r
View(oct_23)
```

```r
View(nov_23)
```

```r
View(dec_23)
```

```r
View(jan_24)
```

```r
View(feb_24)
```

```r
View(mar_24)
```

```r
View(apr_24)
```

```r
View(may_24)
```

```r
View(jun_24)
```

```r
View(jul_24)
```

```r
View(aug_24)
```

check the column names of the datasets

```r
colnames(sep_23)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(oct_23)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(nov_23)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(dec_23)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(jan_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(feb_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(mar_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(apr_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(may_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(jun_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(jul_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
colnames(aug_24)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

```r
head(sep_23)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 011C1903BF4E2E28  classic_bike 2023-09-23 00:27:50 2023-09-23 00:33:27
## 2 87DB80E048A1BF9F  classic_bike 2023-09-02 09:26:43 2023-09-02 09:38:19
## 3 7C2EB7AF669066E3 electric_bike 2023-09-25 18:30:11 2023-09-25 18:41:39
## 4 57D197B010269CE3  classic_bike 2023-09-13 15:30:49 2023-09-13 15:39:18
## 5 8A2CEA7C8C8074D8  classic_bike 2023-09-18 15:58:58 2023-09-18 16:05:04
## 6 03F7044D1304CD58 electric_bike 2023-09-15 20:19:25 2023-09-15 20:30:27
##               start_station_name start_station_id
## 1    Halsted St & Wrightwood Ave     TA1309000061
## 2         Clark St & Drummond Pl     TA1307000142
## 3  Financial Pl & Ida B Wells Dr           SL-010
## 4         Clark St & Drummond Pl     TA1307000142
## 5    Halsted St & Wrightwood Ave     TA1309000061
## 6 Southport Ave & Wrightwood Ave     TA1307000113
##                 end_station_name end_station_id start_lat start_lng  end_lat
## 1 Sheffield Ave & Wellington Ave   TA1307000052  41.92914 -87.64908 41.93625
## 2     Racine Ave & Fullerton Ave   TA1306000026  41.93125 -87.64434 41.92557
## 3           Racine Ave & 15th St          13304  41.87506 -87.63314 41.86127
## 4       Racine Ave & Belmont Ave   TA1308000019  41.93125 -87.64434 41.93974
## 5     Racine Ave & Fullerton Ave   TA1306000026  41.92914 -87.64908 41.92557
## 6                                                41.92884 -87.66387 41.90000
##     end_lng member_casual
## 1 -87.65266        member
## 2 -87.65842        member
## 3 -87.65663        member
## 4 -87.65887        member
## 5 -87.65842        member
## 6 -87.64000        member
```

```r
head(oct_23)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 4449097279F8BBE7  classic_bike 2023-10-08 10:36:26 2023-10-08 10:49:19
## 2 9CF060543CA7B439 electric_bike 2023-10-11 17:23:59 2023-10-11 17:36:08
## 3 667F21F4D6BDE69C electric_bike 2023-10-12 07:02:33 2023-10-12 07:06:53
## 4 F92714CC6B019B96  classic_bike 2023-10-24 19:13:03 2023-10-24 19:18:29
## 5 5E34BA5DE945A9CC  classic_bike 2023-10-09 18:19:26 2023-10-09 18:30:56
## 6 F7D7420AFAC53CD9 electric_bike 2023-10-04 17:10:59 2023-10-04 17:25:21
##                     start_station_name start_station_id
## 1 Orleans St & Chestnut St (NEXT Apts)              620
## 2            Desplaines St & Kinzie St     TA1306000003
## 3 Orleans St & Chestnut St (NEXT Apts)              620
## 4            Desplaines St & Kinzie St     TA1306000003
## 5            Desplaines St & Kinzie St     TA1306000003
## 6 Orleans St & Chestnut St (NEXT Apts)              620
##               end_station_name end_station_id start_lat start_lng  end_lat
## 1 Sheffield Ave & Webster Ave   TA1309000033  41.89820 -87.63754 41.92154
## 2 Sheffield Ave & Webster Ave   TA1309000033  41.88864 -87.64441 41.92154
## 3       Franklin St & Lake St   TA1307000111  41.89807 -87.63751 41.88584
## 4       Franklin St & Lake St   TA1307000111  41.88872 -87.64445 41.88584
```

```
## 5       Franklin St & Lake St   TA1307000111  41.88872 -87.64445 41.88584
## 6 Sheffield Ave & Webster Ave   TA1309000033  41.89812 -87.63753 41.92154
##     end_lng member_casual
## 1 -87.65382        member
## 2 -87.65382        member
## 3 -87.63550        member
## 4 -87.63550        member
## 5 -87.63550        member
## 6 -87.65382        member
```

```
head(nov_23)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 4EAD8F1AD547356B electric_bike 2023-11-30 21:50:05 2023-11-30 22:13:27
## 2 6322270563BF5470 electric_bike 2023-11-03 09:44:02 2023-11-03 10:17:15
## 3 B37BDE091ECA38E0 electric_bike 2023-11-30 11:39:44 2023-11-30 11:40:08
## 4 CF0CA5DD26E4F90E  classic_bike 2023-11-08 10:01:45 2023-11-08 10:27:05
## 5 EB8381AA641348DB  classic_bike 2023-11-03 16:20:25 2023-11-03 16:54:25
## 6 B8CF14EA423D6886 electric_bike 2023-11-30 16:15:53 2023-11-30 16:39:52
##              start_station_name start_station_id              end_station_name
## 1              Millennium Park            13008 Pine Grove Ave & Waveland Ave
## 2        Broadway & Sheridan Rd            13323        Broadway & Sheridan Rd
## 3         State St & Pearson St     TA1307000061         State St & Pearson St
## 4           Theater on the Lake     TA1308000001           Theater on the Lake
## 5           Theater on the Lake     TA1308000001           Theater on the Lake
## 6 Pine Grove Ave & Waveland Ave     TA1307000150               Millennium Park
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   TA1307000150  41.88110 -87.62408 41.94947 -87.64645        member
## 2          13323  41.95287 -87.65003 41.95283 -87.64999        member
## 3   TA1307000061  41.89753 -87.62869 41.89745 -87.62872        member
## 4   TA1308000001  41.92628 -87.63083 41.92628 -87.63083        member
## 5   TA1308000001  41.92628 -87.63083 41.92628 -87.63083        member
## 6          13008  41.94942 -87.64638 41.88103 -87.62408        member
```

```
head(dec_23)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 C9BD54F578F57246 electric_bike 2023-12-02 18:44:01 2023-12-02 18:47:51
## 2 CDBD92F067FA620E electric_bike 2023-12-02 18:48:19 2023-12-02 18:54:48
## 3 ABC0858E52CBFC84 electric_bike 2023-12-24 01:56:32 2023-12-24 02:04:09
## 4 F44B6F0E8F76DC90 electric_bike 2023-12-24 10:58:12 2023-12-24 11:03:04
## 5 3C876413281A90DF electric_bike 2023-12-24 12:43:16 2023-12-24 12:44:57
## 6 28C0D6EFB81E1769 electric_bike 2023-12-24 13:59:57 2023-12-24 14:10:57
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                                         41.92
## 2                                                                         41.92
## 3                                                                         41.89
## 4                                                                         41.95
## 5                                                                         41.92
## 6                                                                         41.91
##   start_lng end_lat end_lng member_casual
## 1    -87.66   41.92  -87.66        member
## 2    -87.66   41.89  -87.64        member
```

```
## 3    -87.62   41.90  -87.64        member
## 4    -87.65   41.94  -87.65        member
## 5    -87.64   41.93  -87.64        member
## 6    -87.63   41.88  -87.65        member
```

```
head(jan_24)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 C1D650626C8C899A electric_bike 2024-01-12 15:30:27 2024-01-12 15:37:59
## 2 EECD38BDB25BFCB0 electric_bike 2024-01-08 15:45:46 2024-01-08 15:52:59
## 3 F4A9CE78061F17F7 electric_bike 2024-01-27 12:27:19 2024-01-27 12:35:19
## 4 0A0D9E15EE50B171  classic_bike 2024-01-29 16:26:17 2024-01-29 16:56:06
## 5 33FFC9805E3EFF9A  classic_bike 2024-01-31 05:43:23 2024-01-31 06:09:35
## 6 C96080812CD285C5  classic_bike 2024-01-07 11:21:24 2024-01-07 11:30:03
##           start_station_name start_station_id          end_station_name
## 1          Wells St & Elm St     KA1504000135  Kingsbury St & Kinzie St
## 2          Wells St & Elm St     KA1504000135  Kingsbury St & Kinzie St
## 3          Wells St & Elm St     KA1504000135  Kingsbury St & Kinzie St
## 4     Wells St & Randolph St     TA1305000030 Larrabee St & Webster Ave
## 5 Lincoln Ave & Waveland Ave            13253  Kingsbury St & Kinzie St
## 6          Wells St & Elm St     KA1504000135  Kingsbury St & Kinzie St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   KA1503000043  41.90327 -87.63474 41.88918 -87.63851        member
## 2   KA1503000043  41.90294 -87.63444 41.88918 -87.63851        member
## 3   KA1503000043  41.90295 -87.63447 41.88918 -87.63851        member
## 4          13193  41.88430 -87.63396 41.92182 -87.64414        member
## 5   KA1503000043  41.94880 -87.67528 41.88918 -87.63851        member
## 6   KA1503000043  41.90322 -87.63432 41.88918 -87.63851        member
```

```
head(feb_24)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 FCB05EB1758F85E8  classic_bike 2024-02-03 14:14:18 2024-02-03 14:21:00
## 2 7FB986AD5D3DE9D6  classic_bike 2024-02-05 21:10:06 2024-02-05 21:15:44
## 3 40CA13E15B5B470D electric_bike 2024-02-05 15:10:44 2024-02-05 15:12:32
## 4 D47A1660919E8861  classic_bike 2024-02-15 12:40:34 2024-02-15 12:44:24
## 5 4CD173D11BA019F8  classic_bike 2024-02-14 12:28:36 2024-02-14 12:36:59
## 6 DA5032C0CA737AF5 electric_bike 2024-02-16 00:54:48 2024-02-16 01:01:47
##             start_station_name start_station_id             end_station_name
## 1        Clark St & Newport St              632 Southport Ave & Waveland Ave
## 2 Michigan Ave & Washington St            13001       Wabash Ave & Grand Ave
## 3    Leavitt St & Armitage Ave     TA1309000029 Milwaukee Ave & Wabansia Ave
## 4 Southport Ave & Waveland Ave            13235  Southport Ave & Belmont Ave
## 5      Wentworth Ave & 35th St     KA1503000005        Shields Ave & 31st St
## 6   Sheridan Rd & Lawrence Ave     TA1309000041        Clark St & Newport St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1          13235  41.94454 -87.65468 41.94815 -87.66394        member
## 2   TA1307000117  41.88398 -87.62468 41.89147 -87.62676        member
## 3          13243  41.91760 -87.68250 41.91262 -87.68139        member
## 4          13229  41.94815 -87.66394 41.93948 -87.66375        member
## 5   KA1503000038  41.83078 -87.63250 41.83846 -87.63541        casual
## 6            632  41.96942 -87.65479 41.94454 -87.65468        member
```

```
head(mar_24)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 64FBE3BAED5F29E6 electric_bike 2024-03-05 18:33:11 2024-03-05 18:51:48
## 2 9991629435C5E20E electric_bike 2024-03-06 17:15:14 2024-03-06 17:16:04
## 3 E5C9FECD5B71BEBD electric_bike 2024-03-06 17:16:36 2024-03-06 17:19:28
## 4 4CEA3EC8906DAEA8 electric_bike 2024-03-03 22:55:54 2024-03-03 22:58:08
## 5 77266B408503C55F electric_bike 2024-03-17 11:15:18 2024-03-17 11:31:18
## 6 E81C25D251767135 electric_bike 2024-03-29 14:40:49 2024-03-29 14:58:11
##   start_station_name start_station_id end_station_name end_station_id start_lat
## 1                                                                        41.94
## 2                                                                        41.91
## 3                                                                        41.91
## 4                                                                        41.90
## 5                                                                        41.93
## 6                                                                        41.93
##   start_lng end_lat end_lng member_casual
## 1    -87.65   41.96  -87.65        member
## 2    -87.64   41.91  -87.64        member
## 3    -87.64   41.92  -87.64        member
## 4    -87.63   41.89  -87.63        member
## 5    -87.70   41.93  -87.72        member
## 6    -87.70   41.95  -87.68        member
```

```
head(apr_24)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 743252713F32516B  classic_bike 2024-04-22 19:08:21 2024-04-22 19:12:56
## 2 BE90D33D2240C614 electric_bike 2024-04-11 06:19:24 2024-04-11 06:22:21
## 3 D47BBDDE7C40DD61  classic_bike 2024-04-20 11:13:13 2024-04-20 11:29:31
## 4 6684E760BF9EA9B5  classic_bike 2024-04-04 18:39:20 2024-04-04 18:43:06
## 5 CA9EFC0D24C24A27 electric_bike 2024-04-19 19:30:20 2024-04-19 20:07:42
## 6 AA64319F52336324  classic_bike 2024-04-10 16:27:08 2024-04-10 16:32:16
##           start_station_name start_station_id              end_station_name
## 1 Aberdeen St & Jackson Blvd            13157   Desplaines St & Jackson Blvd
## 2 Aberdeen St & Jackson Blvd            13157   Desplaines St & Jackson Blvd
## 3 Sheridan Rd & Montrose Ave     TA1307000107 Ashland Ave & Belle Plaine Ave
## 4 Aberdeen St & Jackson Blvd            13157   Desplaines St & Jackson Blvd
## 5 Sheridan Rd & Montrose Ave     TA1307000107   Stetson Ave & South Water St
## 6 Aberdeen St & Jackson Blvd            13157       Loomis St & Lexington St
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1          15539  41.87773 -87.65479 41.87812 -87.64395        member
## 2          15539  41.87772 -87.65496 41.87812 -87.64395        member
## 3          13249  41.96167 -87.65464 41.95606 -87.66884        member
## 4          15539  41.87773 -87.65479 41.87812 -87.64395        member
## 5   TA1308000029  41.96161 -87.65461 41.88683 -87.62232        member
## 6          13332  41.87773 -87.65479 41.87223 -87.66136        member
```

```
head(may_24)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 7D9F0CE9EC2A1297  classic_bike 2024-05-25 15:52:42 2024-05-25 16:11:50
```

```
## 2 02EC47687411416F  classic_bike 2024-05-14 15:11:51 2024-05-14 15:22:00
## 3 101370FB2D3402BE  classic_bike 2024-05-30 17:46:04 2024-05-30 18:09:16
## 4 E97E396331ED6913 electric_bike 2024-05-17 20:21:54 2024-05-17 20:40:32
## 5 674EDE311C543165  classic_bike 2024-05-22 18:52:20 2024-05-22 18:59:04
## 6 2E3EA4C19F0341A6 electric_bike 2024-05-25 19:32:12 2024-05-25 19:36:17
##            start_station_name start_station_id              end_station_name
## 1     Streeter Dr & Grand Ave            13022              Clark St & Elm St
## 2 Sheridan Rd & Greenleaf Ave     KA1504000159       Sheridan Rd & Loyola Ave
## 3     Streeter Dr & Grand Ave            13022            Wabash Ave & 9th St
## 4     Streeter Dr & Grand Ave            13022 Sheffield Ave & Wellington Ave
## 5   Larrabee St & Division St     KA1504000079              Clark St & Elm St
## 6 Sheridan Rd & Greenleaf Ave     KA1504000159       Sheridan Rd & Loyola Ave
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1   TA1307000039  41.89228 -87.61204 41.90297 -87.63128        casual
## 2         RP-009  42.01059 -87.66241 42.00104 -87.66120        casual
## 3   TA1309000010  41.89228 -87.61204 41.87077 -87.62573        member
## 4   TA1307000052  41.89227 -87.61195 41.93625 -87.65266        member
## 5   TA1307000039  41.90349 -87.64335 41.90297 -87.63128        casual
## 6         RP-009  42.01057 -87.66246 42.00104 -87.66120        casual
```

```
head(jun_24)
```

```
##            ride_id rideable_type              started_at
## 1 CDE6023BE6B11D2F electric_bike 2024-06-11 17:20:06.289
## 2 462B48CD292B6A18 electric_bike 2024-06-11 17:19:21.567
## 3 9CFB6A858D23ABF7 electric_bike 2024-06-11 17:25:27.089
## 4 6365EFEB64231153 electric_bike 2024-06-11 11:53:50.769
## 5 BA0323C33134CBA8 electric_bike 2024-06-11 00:11:08.237
## 6 DE26F0D728517B77 electric_bike 2024-06-11 00:12:38.396
##                  ended_at start_station_name start_station_id end_station_name
## 1 2024-06-11 17:21:39.464
## 2 2024-06-11 17:19:36.377
## 3 2024-06-11 17:30:13.035
## 4 2024-06-11 12:08:13.382
## 5 2024-06-11 00:11:22.998
## 6 2024-06-11 00:12:57.813
##   end_station_id start_lat start_lng end_lat end_lng member_casual
## 1                    41.89    -87.65   41.89  -87.65        casual
## 2                    41.89    -87.65   41.89  -87.65        casual
## 3                    41.93    -87.65   41.94  -87.65        casual
## 4                    41.88    -87.64   41.88  -87.64        casual
## 5                    41.94    -87.64   41.94  -87.64        casual
## 6                    41.94    -87.64   41.94  -87.64        casual
```

```
head(jul_24)
```

```
##            ride_id rideable_type              started_at
## 1 2658E319B13141F9 electric_bike 2024-07-11 08:15:14.784
## 2 B2176315168A47CE electric_bike 2024-07-11 15:45:07.851
## 3 C2A9D33DF7EBB422 electric_bike 2024-07-11 08:24:48.192
## 4 8BFEA406DF01D8AD electric_bike 2024-07-11 08:46:06.864
## 5 ECD3EF02E5EB73B6 electric_bike 2024-07-11 18:18:16.588
## 6 A3C62391BBBAC107 electric_bike 2024-07-11 16:03:59.708
```

```
##                  ended_at start_station_name start_station_id end_station_name
## 1 2024-07-11 08:17:56.335
## 2 2024-07-11 16:06:04.243
## 3 2024-07-11 08:28:05.237
## 4 2024-07-11 09:14:11.664
## 5 2024-07-11 18:30:20.288
## 6 2024-07-11 16:32:38.635
##   end_station_id start_lat start_lng end_lat end_lng member_casual
## 1                    41.80    -87.59   41.79  -87.59        casual
## 2                    41.79    -87.60   41.80  -87.59        casual
## 3                    41.79    -87.59   41.79  -87.60        casual
## 4                    41.88    -87.64   41.90  -87.67        casual
## 5                    41.95    -87.64   41.91  -87.62        casual
## 6                    41.70    -87.61   41.70  -87.61        casual
```

```
head(aug_24)
```

```
##            ride_id rideable_type              started_at
## 1 BAA154388A869E64  classic_bike 2024-08-02 13:35:14.403
## 2 8752245932EFF67A electric_bike 2024-08-02 15:33:13.965
## 3 44DDF9F57A9A161F  classic_bike 2024-08-16 15:44:06.233
## 4 44AAAF069B0C78C3 electric_bike 2024-08-19 18:47:11.855
## 5 77138D500A6B7B4B  classic_bike 2024-08-03 20:34:20.560
## 6 F6F581F31A9C9BC2 electric_bike 2024-08-03 20:08:09.067
##                  ended_at       start_station_name start_station_id
## 1 2024-08-02 13:48:24.426   State St & Randolph St     TA1305000029
## 2 2024-08-02 15:55:23.865  Franklin St & Monroe St     TA1309000007
## 3 2024-08-16 15:57:52.109  Franklin St & Monroe St     TA1309000007
## 4 2024-08-19 18:56:33.269        Clark St & Elm St     TA1307000039
## 5 2024-08-03 20:46:29.305 Western Ave & Leland Ave     TA1307000140
## 6 2024-08-03 20:44:53.847        Clark St & Elm St     TA1307000039
##                 end_station_name end_station_id start_lat start_lng  end_lat
## 1            Wabash Ave & 9th St   TA1309000010  41.88462 -87.62783 41.87077
## 2        Damen Ave & Cortland St          13133  41.88032 -87.63519 41.91598
## 3             Clark St & Elm St   TA1307000039  41.88032 -87.63519 41.90297
## 4           McClurg Ct & Ohio St   TA1306000029  41.90297 -87.63128 41.89259
## 5 Ashland Ave & Belle Plaine Ave          13249  41.96640 -87.68870 41.95606
## 6   Stetson Ave & South Water St   TA1308000029  41.90297 -87.63128 41.88683
##     end_lng member_casual
## 1 -87.62573        member
## 2 -87.67733        member
## 3 -87.63128        member
## 4 -87.61729        member
## 5 -87.66884        casual
## 6 -87.62232        casual
```

I still need to amend and delete some column's in order to make the files workable when I merge them together.

Merging the datasets

Now it is time to merge the two datasets together into a new dataframe

```
all_trips <- bind_rows(sep_23, oct_23, nov_23, dec_23, jan_24, feb_24, mar_24, apr_24, may_24, jun_24, j
```

This has now created a new dataset called all_trips which I will view

```
View(all_trips)
```

check the columns are correct

```
colnames(all_trips)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"
##  [4] "ended_at"           "start_station_name" "start_station_id"
##  [7] "end_station_name"   "end_station_id"     "start_lat"
## [10] "start_lng"          "end_lat"            "end_lng"
## [13] "member_casual"
```

And check the rows and columns have stacked correctly

```
head(all_trips)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 011C1903BF4E2E28  classic_bike 2023-09-23 00:27:50 2023-09-23 00:33:27
## 2 87DB80E048A1BF9F  classic_bike 2023-09-02 09:26:43 2023-09-02 09:38:19
## 3 7C2EB7AF669066E3 electric_bike 2023-09-25 18:30:11 2023-09-25 18:41:39
## 4 57D197B010269CE3  classic_bike 2023-09-13 15:30:49 2023-09-13 15:39:18
## 5 8A2CEA7C8C8074D8  classic_bike 2023-09-18 15:58:58 2023-09-18 16:05:04
## 6 03F7044D1304CD58 electric_bike 2023-09-15 20:19:25 2023-09-15 20:30:27
##               start_station_name start_station_id
## 1    Halsted St & Wrightwood Ave     TA1309000061
## 2         Clark St & Drummond Pl     TA1307000142
## 3  Financial Pl & Ida B Wells Dr           SL-010
## 4         Clark St & Drummond Pl     TA1307000142
## 5    Halsted St & Wrightwood Ave     TA1309000061
## 6 Southport Ave & Wrightwood Ave     TA1307000113
##                 end_station_name end_station_id start_lat start_lng  end_lat
## 1 Sheffield Ave & Wellington Ave   TA1307000052  41.92914 -87.64908 41.93625
## 2     Racine Ave & Fullerton Ave   TA1306000026  41.93125 -87.64434 41.92557
## 3           Racine Ave & 15th St          13304  41.87506 -87.63314 41.86127
## 4       Racine Ave & Belmont Ave   TA1308000019  41.93125 -87.64434 41.93974
## 5     Racine Ave & Fullerton Ave   TA1306000026  41.92914 -87.64908 41.92557
## 6                                                41.92884 -87.66387 41.90000
##     end_lng member_casual
## 1 -87.65266        member
## 2 -87.65842        member
## 3 -87.65663        member
## 4 -87.65887        member
## 5 -87.65842        member
## 6 -87.64000        member
```

As you can see some columns mainly start and end station are only referenced in some datasets and some columns I will not need to produce my report therefore I will need to clean this new dataframe before beginning analysis.

```
all_trips <- all_trips %>%
  select(-c(start_lat, start_lng, end_lat, end_lng, start_station_name, start_station_id, end_station_na
```

Removing Columns I will not need

Time to Inspect my New Dataframe

Now it's time to inspect my new Dataframe.

```
View(all_trips)
```

```
colnames(all_trips)  #List of column names
```

```
## [1] "ride_id"       "rideable_type" "started_at"    "ended_at"
## [5] "member_casual"
```

```
nrow(all_trips)  #How many rows are in data frame?
```

```
## [1] 5699639
```

```
dim(all_trips)  #Dimensions of the data frame?
```

```
## [1] 5699639       5
```

```
head(all_trips)  #See the first 6 rows of data frame.
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 011C1903BF4E2E28  classic_bike 2023-09-23 00:27:50 2023-09-23 00:33:27
## 2 87DB80E048A1BF9F  classic_bike 2023-09-02 09:26:43 2023-09-02 09:38:19
## 3 7C2EB7AF669066E3 electric_bike 2023-09-25 18:30:11 2023-09-25 18:41:39
## 4 57D197B010269CE3  classic_bike 2023-09-13 15:30:49 2023-09-13 15:39:18
## 5 8A2CEA7C8C8074D8  classic_bike 2023-09-18 15:58:58 2023-09-18 16:05:04
## 6 03F7044D1304CD58 electric_bike 2023-09-15 20:19:25 2023-09-15 20:30:27
##   member_casual
## 1        member
## 2        member
## 3        member
## 4        member
## 5        member
## 6        member
```

```
str(all_trips)  #See list of columns and data types (numeric, character, etc)
```

```
## 'data.frame':    5699639 obs. of  5 variables:
##  $ ride_id      : chr  "011C1903BF4E2E28" "87DB80E048A1BF9F" "7C2EB7AF669066E3" "57D197B010269CE3" .
##  $ rideable_type: chr  "classic_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at   : chr  "2023-09-23 00:27:50" "2023-09-02 09:26:43" "2023-09-25 18:30:11" "2023-09-13
##  $ ended_at     : chr  "2023-09-23 00:33:27" "2023-09-02 09:38:19" "2023-09-25 18:41:39" "2023-09-13
##  $ member_casual: chr  "member" "member" "member" "member" ...
```

```
tail(all_trips)
```

```
##                  ride_id rideable_type              started_at
## 5699634 B8F5251CAD532E7B electric_bike 2024-08-09 14:31:52.567
## 5699635 71F72E76C0BD298A electric_bike 2024-08-09 14:34:08.222
## 5699636 CC9B1538EE80EEF5 electric_bike 2024-08-09 20:23:23.237
## 5699637 0F2CB3D2FDC5C124 electric_bike 2024-08-18 00:36:26.939
## 5699638 FAAD5D34100D74DF electric_bike 2024-08-30 11:22:50.179
## 5699639 5CB8C2D9C5C72EBA electric_bike 2024-08-30 17:01:57.243
##                        ended_at member_casual
## 5699634 2024-08-09 14:31:54.880        member
## 5699635 2024-08-09 14:51:31.936        member
## 5699636 2024-08-09 20:37:30.107        member
## 5699637 2024-08-18 00:38:45.112        member
## 5699638 2024-08-30 11:47:10.750        member
## 5699639 2024-08-30 17:20:12.001        member
```

```
summary(all_trips)  #Statistical summary of data. Mainly for numerics
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5699639     Length:5699639     Length:5699639     Length:5699639
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##  member_casual
##  Length:5699639
##  Class :character
##  Mode  :character
```

Further Actions

I need to introduce columns I had for the individual spreadsheets (Weekday, Ride Length, Mean Ride, Max Ride, Min Ride) as well as create new columns i.e Day, Month, Year so that I am able to further aggregate the data.

Adding the Date,Month,Year,Day to each ride

The reason I want to add this to the Dataframe is in order to be able to gain more opportunities to aggragate the data.

```
all_trips$date <- as.Date(all_trips$started_at) #The default format is yyyy-mm-dd
```

```
all_trips$month <- format(as.Date(all_trips$date), "%m") # Month
```

```
all_trips$day <- format(as.Date(all_trips$date), "%d") # Day
```

```
all_trips$year <- format(as.Date(all_trips$date), "%Y") # Year
```

```
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A") # Day of Week
```

Now my Dataframe looks like this,

```
head(all_trips)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 011C1903BF4E2E28  classic_bike 2023-09-23 00:27:50 2023-09-23 00:33:27
## 2 87DB80E048A1BF9F  classic_bike 2023-09-02 09:26:43 2023-09-02 09:38:19
## 3 7C2EB7AF669066E3 electric_bike 2023-09-25 18:30:11 2023-09-25 18:41:39
## 4 57D197B010269CE3  classic_bike 2023-09-13 15:30:49 2023-09-13 15:39:18
## 5 8A2CEA7C8C8074D8  classic_bike 2023-09-18 15:58:58 2023-09-18 16:05:04
## 6 03F7044D1304CD58 electric_bike 2023-09-15 20:19:25 2023-09-15 20:30:27
##   member_casual       date month day year day_of_week
## 1        member 2023-09-23    09  23 2023    Saturday
## 2        member 2023-09-02    09  02 2023    Saturday
## 3        member 2023-09-25    09  25 2023      Monday
## 4        member 2023-09-13    09  13 2023   Wednesday
## 5        member 2023-09-18    09  18 2023      Monday
## 6        member 2023-09-15    09  15 2023      Friday
```

Adding the Ride Length (in Seconds)

```
all_trips$started_at <- ymd_hms(all_trips$started_at)
all_trips$ended_at <- ymd_hms(all_trips$ended_at)


# Calculate diff time
all_trips$ride_length <- difftime(
    all_trips$ended_at,
    all_trips$started_at,
    units = "secs"
    )
```

Then I will inspect the structure of the column

```
str(all_trips)
```

```
## 'data.frame':    5699639 obs. of  11 variables:
##  $ ride_id      : chr  "011C1903BF4E2E28" "87DB80E048A1BF9F" "7C2EB7AF669066E3" "57D197B010269CE3" .
##  $ rideable_type: chr  "classic_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at   : POSIXct, format: "2023-09-23 00:27:50" "2023-09-02 09:26:43" ...
##  $ ended_at     : POSIXct, format: "2023-09-23 00:33:27" "2023-09-02 09:38:19" ...
##  $ member_casual: chr  "member" "member" "member" "member" ...
##  $ date         : Date, format: "2023-09-23" "2023-09-02" ...
##  $ month        : chr  "09" "09" "09" "09" ...
##  $ day          : chr  "23" "02" "25" "13" ...
##  $ year         : chr  "2023" "2023" "2023" "2023" ...
##  $ day_of_week  : chr  "Saturday" "Saturday" "Monday" "Wednesday" ...
##  $ ride_length  : 'difftime' num  337 696 688 509 ...
##   ..- attr(*, "units")= chr "secs"
```

I need to make sure the column for ride_length isn't in factor form, I need this to be numeric if I wish to make calculations such as mean,median, mode later on.

```
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))
is.numeric(all_trips$ride_length)
```

```
## [1] TRUE
```

Removing Bad Data

As I had previously found from the individual spreadsheets there are rides that have either negative or had 0 sec rides, These rides are either Quality Control on behalf of Cyclistic or errors, I will remove these rides for my analysis

```
all_trips_v2 <- all_trips[!(all_trips$ride_length<1),] # This will remove all rides that are negative o
```

Time to Anaylse the Data

Descriptive analysis on ride_length (all figures in seconds)

```
summary(all_trips_v2) # Overall Summary of the data
```

```
##    ride_id          rideable_type        started_at
##  Length:5697848     Length:5697848     Min.   :2023-09-01 00:00:44.00
##  Class :character   Class :character   1st Qu.:2023-11-15 18:52:07.75
##  Mode  :character   Mode  :character   Median :2024-04-29 17:56:55.50
##                                        Mean   :2024-03-23 06:16:16.68
##                                        3rd Qu.:2024-07-04 14:55:59.21
##                                        Max.   :2024-08-31 23:58:30.89
##     ended_at                       member_casual           date
##  Min.   :2023-09-01 00:03:06.00   Length:5697848     Min.   :2023-09-01
##  1st Qu.:2023-11-15 19:04:56.50   Class :character   1st Qu.:2023-11-15
##  Median :2024-04-29 18:11:34.50   Mode  :character   Median :2024-04-29
##  Mean   :2024-03-23 06:33:50.03                      Mean   :2024-03-22
##  3rd Qu.:2024-07-04 15:21:06.00                      3rd Qu.:2024-07-04
##  Max.   :2024-08-31 23:59:53.88                      Max.   :2024-08-31
##     month                day                year           day_of_week
##  Length:5697848     Length:5697848     Length:5697848     Length:5697848
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##   ride_length
##  Min.   :    1
##  1st Qu.:  334
##  Median :  585
##  Mean   : 1053
##  3rd Qu.: 1039
##  Max.   :93596
```

```r
mean(all_trips_v2$ride_length) # Mean of the ride length
```

```
## [1] 1053.355
```

```r
total_ride_length <- sum(all_trips_v2$ride_length, na.rm = TRUE)

number_of_rides <- nrow(all_trips_v2)

average_ride_length <- total_ride_length / number_of_rides

print(average_ride_length) # This chuck of code will get the Average ride length
```

```
## [1] 1053.355
```

```r
median(all_trips_v2$ride_length) # Median ride length
```

```
## [1] 585
```

```r
max(all_trips_v2$ride_length) # Longest ride
```

```
## [1] 93596
```

```r
min(all_trips_v2$ride_length) # Shortest ride
```

```
## [1] 1
```

Time to compare Members and Casuals

Now it's time to see the difference using these metrics between the two userbases

```r
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean) # The mean between the two
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1546.7155
## 2                     member                 777.1539
```

```r
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                      733
## 2                     member                      523
```

```r
# The Median between the two userbases
```

```r
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                    93596
## 2                     member                    93588
```

```
# The longest ride between the two userbases
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                        1
## 2                     member                        1
```

```
# The shortest ride between the two userbases
```

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Friday                1509.8972
## 2                      member                   Friday                 757.9903
## 3                      casual                   Monday                1492.9274
## 4                      member                   Monday                 741.8198
## 5                      casual                 Saturday                1712.6496
## 6                      member                 Saturday                 858.0694
## 7                      casual                   Sunday                1813.5691
## 8                      member                   Sunday                 870.8032
## 9                      casual                 Thursday                1351.0120
## 10                     member                 Thursday                 743.1314
## 11                     casual                  Tuesday                1325.5858
## 12                     member                  Tuesday                 744.1502
## 13                     casual                Wednesday                1378.0317
## 14                     member                Wednesday                 757.3989
```

The days of the week are out of order making this data harder to understand I need to write the below code to fix this issues

```
all_trips_v2$day_of_week <- ordered(all_trips_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "
```

Now I will try to find the average ride time by userbase on a daily basis

```
aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual + all_trips_v2$day_of_week, FUN = mean)
```

```
##    all_trips_v2$member_casual all_trips_v2$day_of_week all_trips_v2$ride_length
## 1                      casual                   Sunday                1813.5691
## 2                      member                   Sunday                 870.8032
## 3                      casual                   Monday                1492.9274
## 4                      member                   Monday                 741.8198
## 5                      casual                  Tuesday                1325.5858
## 6                      member                  Tuesday                 744.1502
## 7                      casual                Wednesday                1378.0317
## 8                      member                Wednesday                 757.3989
## 9                      casual                 Thursday                1351.0120
## 10                     member                 Thursday                 743.1314
## 11                     casual                   Friday                1509.8972
## 12                     member                   Friday                 757.9903
## 13                     casual                 Saturday                1712.6496
## 14                     member                 Saturday                 858.0694
```

This looks much clearer to read and understand.

Lets Analyse the Userbase by Weekday

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>%  #groups by usertype and weekday
  summarise(number_of_rides = n()                              #calculates the number of rides and average
            ,average_duration = mean(ride_length)) %>%      # calculates the average duration
  arrange(member_casual, weekday)   # sorts the data
```

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

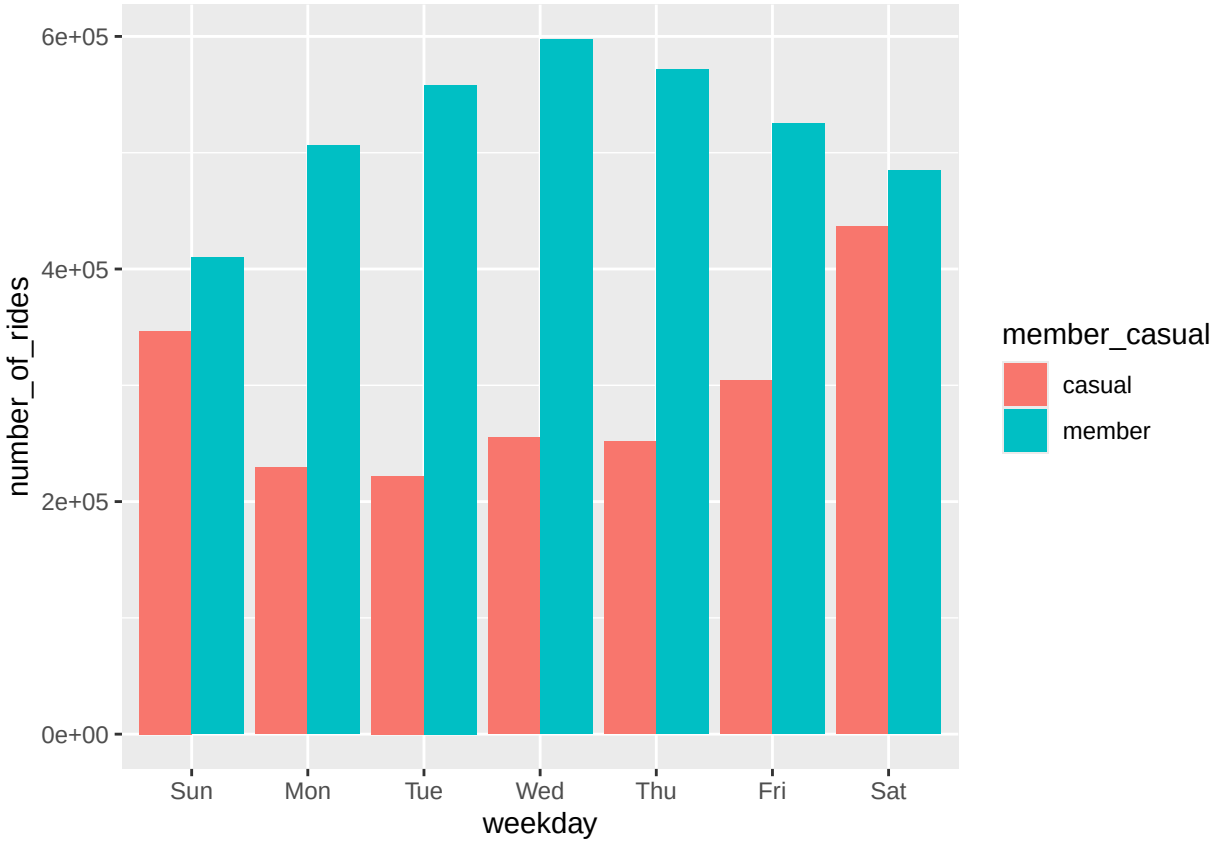
```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              346556            1814.
##  2 casual        Mon              229179            1493.
##  3 casual        Tue              221880            1326.
##  4 casual        Wed              255023            1378.
##  5 casual        Thu              251839            1351.
##  6 casual        Fri              304126            1510.
##  7 casual        Sat              436397            1713.
##  8 member        Sun              409816             871.
##  9 member        Mon              506179             742.
## 10 member        Tue              558123             744.
## 11 member        Wed              597490             757.
## 12 member        Thu              571273             743.
## 13 member        Fri              525143             758.
## 14 member        Sat              484824             858.
```

Time to Visualise this

The Below Graph shows the number of rides by userbase

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

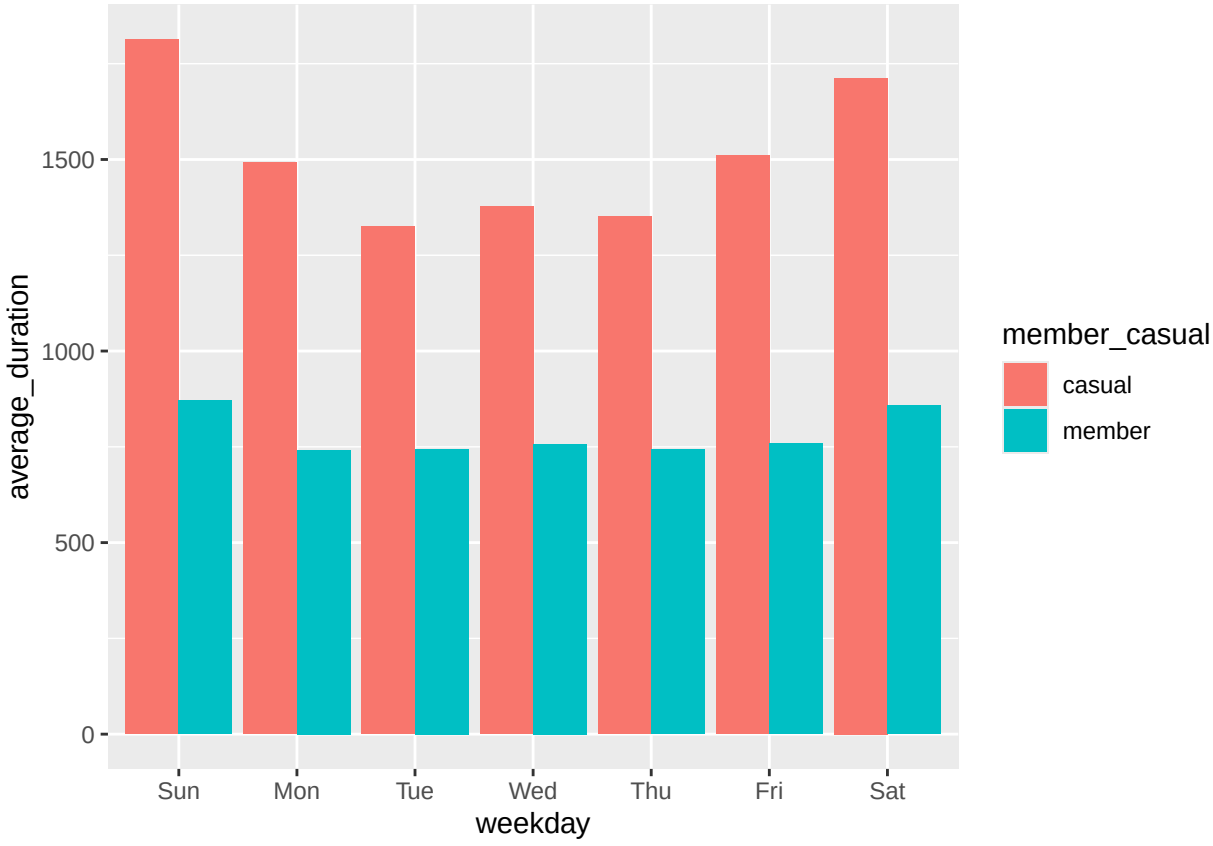
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



The Below graph shows the average duration by userbase

```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

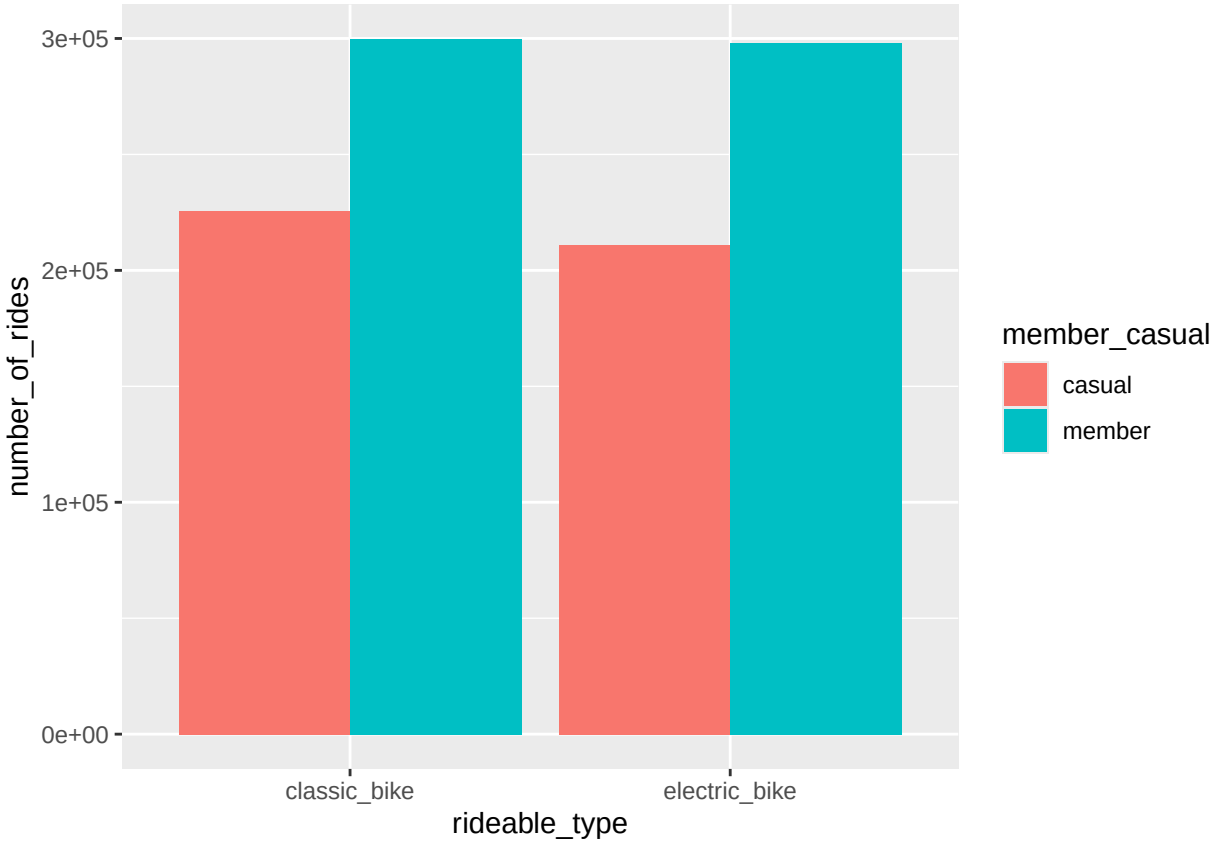
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



The Rideable Type by Member Type

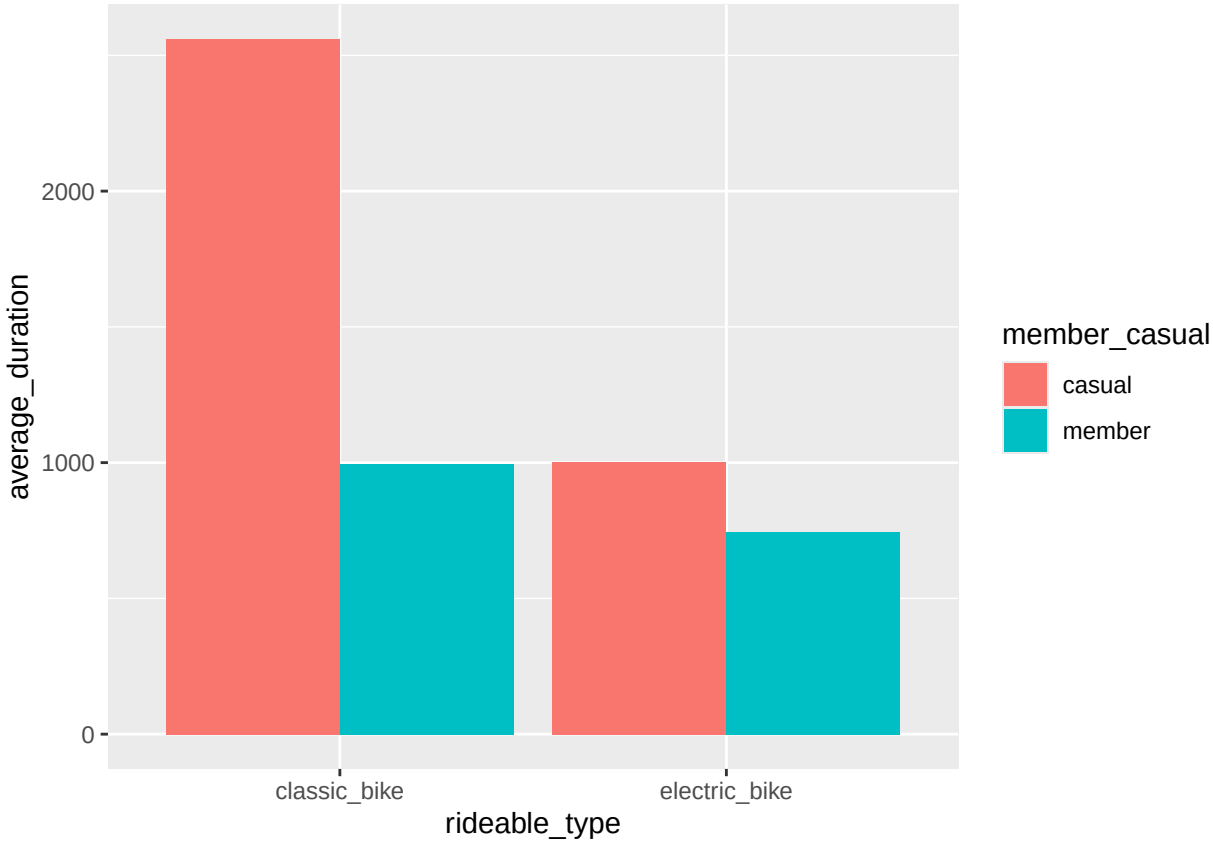
```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday, rideable_type) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x = rideable_type, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge")
```

```
## `summarise()` has grouped output by 'member_casual', 'weekday'. You can
## override using the `.groups` argument.
```




```
all_trips_v2 %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday, rideable_type) %>%
  summarise(number_of_rides = n()
            ,average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = rideable_type, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")
```

```
## `summarise()` has grouped output by 'member_casual', 'weekday'. You can
## override using the `.groups` argument.
```



Summary

The Graphs currently displayed show the annual user use of the service is higher than the causal user on a daily basis.

However you can see an underlying trend between the two types.

The annual users seems to have an upward trajectory through the working week with a tail off on the weekend,This is most likely due to the trips took by annual members are commutes to work or to events.

The casual users use is lower through the week with the highest peak of users being on the weekend. The rides of a casual user is on average longer than a annual rider regardless of what day the ride is made. this is most likely less frequent but long bike rides from a user when they are off work.

The type of bike that is ridden seems to be equal between the two users but the average duration on a docked bike by a casual user seems to point that a causal user prefers to use this type of bike when doing a long bike ride on the weekend.

Suggestions Based on this

For an Weekend only annual membership. this may have get casual users who only use the service on a weekend to be more inclined to a subscription model.

Monthly or Bi Monthly passes with a set number of hours linked to the pass i.e 1 Month 35 Hours, 2 Months 80 Hours. This may get causal users who will use the service for long durations of time on a weekend to pay upfront but not have the fear of being in an annual subscription model initially. this may In turn make them use the service during the week which may make them reconsider an annual subscription.

A long rider membership which will only be applicable to docked bikes as these seem to be the choice of bike casual users prefer when going for long rides.

Further Visualisation / Analysis

I am going to create a CSV file with this dataframe that I then perform more visualisations using tableau https://public.tableau.com/app/profile/ryan.welsh6016/vizzes