

Google Data Analytics Capstone

Afnan A. Yaqub

2022-08-15

Cyclistic Bike-Share Analysis

Scenario

In this case study my role is as a data analyst for *Cyclistic*, a fictional bike-share company in Chicago.

Cyclistic has two types of users.

- Annual Members
- Casual Riders

The director of marketing, Lily Moreno, believes that Cyclistic's future success depends on increasing the number of annual members. Me and my team have been tasked to identify how Casual Riders and Annual Members use Cyclistic bikes differently by analyzing the historical ride data.

The insights achieved from the data set will be shared with the stakeholders i.e. the director of marketing and the executive team. The marketing team at Cyclistic will design a marketing campaign aimed at converting casual riders to annual members and submit for approval to the stakeholders.

The Analysis Process

The analysis process involves the following steps.

1. **Ask:** Ask the right questions, to define the problem, that the analysis must provide an answer to.
2. **Prepare:** Clean and format the historical data so that it can be processed and analyzed.
3. **Process:** Extract useful information in the form of the variables that correctly represent the answers to the questions.
4. **Analyze:** Perform the analysis to achieve the useful insights into the historical data.
5. **Visualize and Share:** Prepare the visualizations and dashboards for presenting the analysis to the stakeholders.
6. **Act:** Implement marketing campaign based on the outcome of the analysis (Excluded in this capstone).

1. Ask

The following questions must be answered by the analysis for the company to design and run a successful marketing campaign.

- i. How do annual members and casual riders use Cyclistic bikes differently?
- ii. Why would casual riders buy Cyclistic annual memberships?
- iii. How can Cyclistic use digital media to influence casual riders to become members?

2. Prepare

The historical data set of rides since 2013 is provided by the company as downloadable zip files [here](#).

- 2013 dataset is for the complete year.
- 2014-2017 datasets for each year is in two sets of 6 months.
- 2018-2020 datasets from 1st quarter of 2018 to 1st quarter of 2020 is available on quarterly basis.
- 2020-2022 datasets from April 2020 till July 2022 is available on monthly basis.

Since the data is provided by the company itself, it is considered reliable. In order to work with current data I will be using the ride history of past 12 months i.e. August, 2021 to July, 2022. These zip files contain the data in comma separated values (CSV) format.

The programming language used in this analysis is R. First we install the libraries necessary to read the csv files.

```
# setwd("E:/Projects/R/My Data Sources/Cyclistic/")
# install.packages("tidyverse", repos = "http://cran.us.r-project.org")
# install.packages("janitor", repos = "http://cran.us.r-project.org")
# install.packages("dplyr", repos = "http://cran.us.r-project.org")
# install.packages("skimr", repos = "http://cran.us.r-project.org")
# install.packages("kableExtra", repos = "http://cran.us.r-project.org")

library(tidyverse)
library(janitor)
library(dplyr)
library(skimr)
library(lubridate)
library(tibble)
```

The first step of preparation is to load the necessary files into dataframes for initial analysis.

```
tripdata_202108 <- read.csv("202108-divvy-tripdata.csv")
tripdata_202109 <- read.csv("202109-divvy-tripdata.csv")
tripdata_202110 <- read.csv("202110-divvy-tripdata.csv")
tripdata_202111 <- read.csv("202111-divvy-tripdata.csv")
tripdata_202112 <- read.csv("202112-divvy-tripdata.csv")
tripdata_202201 <- read.csv("202201-divvy-tripdata.csv")
```

```

tripdata_202202 <- read.csv("202202-divvy-tripdata.csv")
tripdata_202203 <- read.csv("202203-divvy-tripdata.csv")
tripdata_202204 <- read.csv("202204-divvy-tripdata.csv")
tripdata_202205 <- read.csv("202205-divvy-tripdata.csv")
tripdata_202206 <- read.csv("202206-divvy-tripdata.csv")
tripdata_202207 <- read.csv("202207-divvy-tripdata.csv")

```

Then step by step I visualized each dataframe using the `glimpse()` function from `dplyr` library in `tidyverse` package.

```
glimpse(tripdata_202108)
```

```

## Rows: 804,352
## Columns: 13
## $ ride_id          <chr> "99103BB87CC6C1BB", "EAFCCCFB0A3FC5A1", "9EF4F46C57...
## $ rideable_type    <chr> "electric_bike", "electric_bike", "electric_bike", ...
## $ started_at       <chr> "10/08/2021 17:15", "10/08/2021 17:23", "21/08/2021...
## $ ended_at         <chr> "10/08/2021 17:22", "10/08/2021 17:39", "21/08/2021...
## $ start_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ start_station_id  <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ end_station_name  <chr> "", "", "", "", "", "", "", "", "Clark St & Grace St", ...
## $ end_station_id    <chr> "", "", "", "", "", "", "", "", "TA1307000127", "", "", ...
## $ start_lat         <dbl> 41.77000, 41.77000, 41.95000, 41.97000, 41.79000, 4...
## $ start_lng         <dbl> -87.68000, -87.68000, -87.65000, -87.67000, -87.600...
## $ end_lat           <dbl> 41.77000, 41.77000, 41.97000, 41.95000, 41.77000, 4...
## $ end_lng           <dbl> -87.68000, -87.63000, -87.66000, -87.65000, -87.620...
## $ member_casual    <chr> "member", "member", "member", "member", "member", "..."

```

```
glimpse(tripdata_202109)
```

```

## Rows: 756,147
## Columns: 13
## $ ride_id          <chr> "9DC7B962304CBFD8", "F930E2C6872D6B32", "6EF7213790...
## $ rideable_type    <chr> "electric_bike", "electric_bike", "electric_bike", ...
## $ started_at       <chr> "28/09/2021 16:07", "28/09/2021 14:24", "28/09/2021...
## $ ended_at         <chr> "28/09/2021 16:09", "28/09/2021 14:40", "28/09/2021...
## $ start_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ start_station_id  <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ end_station_name  <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ end_station_id    <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ start_lat         <dbl> 41.89000, 41.94000, 41.81000, 41.80000, 41.88000, 4...
## $ start_lng         <dbl> -87.68000, -87.64000, -87.72000, -87.72000, -87.740...
## $ end_lat           <dbl> 41.89, 41.98, 41.80, 41.81, 41.88, 41.88, 41.74, 41...
## $ end_lng           <dbl> -87.67, -87.67, -87.72, -87.72, -87.71, -87.74, -87...
## $ member_casual    <chr> "casual", "casual", "casual", "casual", "casual", "..."

```

```
glimpse(tripdata_202110)
```

```

## Rows: 631,226
## Columns: 13
## $ ride_id          <chr> "620BC6107255BF4C", "4471C70731AB2E45", "26CA69D43D...
## $ rideable_type    <chr> "electric_bike", "electric_bike", "electric_bike", ...
## $ started_at       <chr> "22/10/2021 12:46", "21/10/2021 9:12", "16/10/2021 ...
## $ ended_at         <chr> "22/10/2021 12:49", "21/10/2021 9:14", "16/10/2021 ...
## $ start_station_name <chr> "Kingsbury St & Kinzie St", "", "", "", "", "", "", ...

```

```
## $ start_station_id <chr> "KA1503000043", "", "", "", "", "", "", "", "", "", "...
## $ end_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "", "...
## $ end_station_id <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "...
## $ start_lat <dbl> 41.88919, 41.93000, 41.92000, 41.92000, 41.89000, 4...
## $ start_lng <dbl> -87.63850, -87.70000, -87.70000, -87.69000, -87.710...
## $ end_lat <dbl> 41.89000, 41.93000, 41.94000, 41.92000, 41.89000, 4...
## $ end_lng <dbl> -87.63000, -87.71000, -87.72000, -87.69000, -87.690...
## $ member_casual <chr> "member", "member", "member", "member", "member", "...
```

```
glimpse(tripdata_202111)
```

```
## Rows: 359,978
## Columns: 13
## $ ride_id <chr> "7C00A93E10556E47", "90854840DFD508BA", "0A7D10CDD1...
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ...
## $ started_at <chr> "2021-11-27 13:27:38", "2021-11-27 13:38:25", "2021...
## $ ended_at <chr> "2021-11-27 13:46:38", "2021-11-27 13:56:10", "2021...
## $ start_station_name <chr> "", "", "", "", "", "Michigan Ave & Oak St", "", "...
## $ start_station_id <chr> "", "", "", "", "", "13042", "", "", "", "", "", "...
## $ end_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "", "...
## $ end_station_id <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", "...
## $ start_lat <dbl> 41.93000, 41.96000, 41.96000, 41.94000, 41.90000, 4...
## $ start_lng <dbl> -87.72000, -87.70000, -87.70000, -87.79000, -87.630...
## $ end_lat <dbl> 41.96, 41.92, 41.96, 41.93, 41.88, 41.90, 41.80, 41...
## $ end_lng <dbl> -87.73, -87.70, -87.70, -87.79, -87.62, -87.63, -87...
## $ member_casual <chr> "casual", "casual", "casual", "casual", "casual", "...
```

```
glimpse(tripdata_202112)
```

```
## Rows: 247,540
## Columns: 13
## $ ride_id <chr> "46F8167220E4431F", "73A77762838B32FD", "4CF4245205...
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ...
## $ started_at <chr> "2021-12-07 15:06:07", "2021-12-11 03:43:29", "2021...
## $ ended_at <chr> "2021-12-07 15:13:42", "2021-12-11 04:10:23", "2021...
## $ start_station_name <chr> "Laflin St & Cullerton St", "LaSalle Dr & Huron St"...
## $ start_station_id <chr> "13307", "KP1705001026", "KA1504000117", "KA1504000...
## $ end_station_name <chr> "Morgan St & Polk St", "Clarendon Ave & Leland Ave"...
## $ end_station_id <chr> "TA1307000130", "TA1307000119", "13137", "KP1705001...
## $ start_lat <dbl> 41.85483, 41.89441, 41.89936, 41.89939, 41.89558, 4...
## $ start_lng <dbl> -87.66366, -87.63233, -87.64852, -87.64854, -87.682...
## $ end_lat <dbl> 41.87197, 41.96797, 41.93758, 41.89488, 41.93125, 4...
## $ end_lng <dbl> -87.65097, -87.65000, -87.64410, -87.63233, -87.644...
## $ member_casual <chr> "member", "casual", "member", "member", "member", "...
```

```
glimpse(tripdata_202201)
```

```
## Rows: 103,770
## Columns: 13
## $ ride_id <chr> "C2F7DD78E82EC875", "A6CF8980A652D272", "BD0F91DFF7...
## $ rideable_type <chr> "electric_bike", "electric_bike", "classic_bike", "...
## $ started_at <chr> "2022-01-13 11:59:47", "2022-01-10 08:41:56", "2022...
## $ ended_at <chr> "2022-01-13 12:02:44", "2022-01-10 08:46:17", "2022...
## $ start_station_name <chr> "Glenwood Ave & Touhy Ave", "Glenwood Ave & Touhy A...
## $ start_station_id <chr> "525", "525", "TA1306000016", "KA1504000151", "TA13...
```

```
## $ end_station_name <chr> "Clark St & Touhy Ave", "Clark St & Touhy Ave", "Gr...
## $ end_station_id <chr> "RP-007", "RP-007", "TA1307000001", "TA1309000021",...
## $ start_lat <dbl> 42.01280, 42.01276, 41.92560, 41.98359, 41.87785, 4...
## $ start_lng <dbl> -87.66591, -87.66597, -87.65371, -87.66915, -87.624...
## $ end_lat <dbl> 42.01256, 42.01256, 41.92533, 41.96151, 41.88462, 4...
## $ end_lng <dbl> -87.67437, -87.67437, -87.66580, -87.67139, -87.627...
## $ member_casual <chr> "casual", "casual", "member", "casual", "member", "...
```

```
glimpse(tripdata_202202)
```

```
## Rows: 115,609
## Columns: 13
## $ ride_id <chr> "E1E065E7ED285C02", "1602DCDC5B30FFE3", "BE7DD2AF4B...
## $ rideable_type <chr> "classic_bike", "classic_bike", "classic_bike", "cl...
## $ started_at <chr> "2022-02-19 18:08:41", "2022-02-20 17:41:30", "2022...
## $ ended_at <chr> "2022-02-19 18:23:56", "2022-02-20 17:45:56", "2022...
## $ start_station_name <chr> "State St & Randolph St", "Halsted St & Wrightwood ...
## $ start_station_id <chr> "TA1305000029", "TA1309000061", "TA1305000029", "13...
## $ end_station_name <chr> "Clark St & Lincoln Ave", "Southport Ave & Wrightwo...
## $ end_station_id <chr> "13179", "TA1307000113", "13011", "13323", "TA13070...
## $ start_lat <dbl> 41.88462, 41.92914, 41.88462, 41.94815, 41.88462, 4...
## $ start_lng <dbl> -87.62783, -87.64908, -87.62783, -87.66394, -87.627...
## $ end_lat <dbl> 41.91569, 41.92877, 41.87926, 41.95283, 41.88584, 4...
## $ end_lng <dbl> -87.63460, -87.66391, -87.63990, -87.64999, -87.635...
## $ member_casual <chr> "member", "member", "member", "member", "member", "...
```

```
glimpse(tripdata_202203)
```

```
## Rows: 284,042
## Columns: 13
## $ ride_id <chr> "47EC0A7F82E65D52", "8494861979B0F477", "EFE527AF80...
## $ rideable_type <chr> "classic_bike", "electric_bike", "classic_bike", "c...
## $ started_at <chr> "2022-03-21 13:45:01", "2022-03-16 09:37:16", "2022...
## $ ended_at <chr> "2022-03-21 13:51:18", "2022-03-16 09:43:34", "2022...
## $ start_station_name <chr> "Wabash Ave & Wacker Pl", "Michigan Ave & Oak St", ...
## $ start_station_id <chr> "TA1307000131", "13042", "13109", "TA1307000131", "...
## $ end_station_name <chr> "Kingsbury St & Kinzie St", "Orleans St & Chestnut ...
## $ end_station_id <chr> "KA1503000043", "620", "15578", "TA1305000025", "13...
## $ start_lat <dbl> 41.88688, 41.90100, 41.97835, 41.88688, 41.91172, 4...
## $ start_lng <dbl> -87.62603, -87.62375, -87.65975, -87.62603, -87.626...
## $ end_lat <dbl> 41.88918, 41.89820, 41.98404, 41.87771, 41.87794, 4...
## $ end_lng <dbl> -87.63851, -87.63754, -87.66027, -87.63532, -87.662...
## $ member_casual <chr> "member", "member", "member", "member", "member", "...
```

```
glimpse(tripdata_202204)
```

```
## Rows: 371,249
## Columns: 13
## $ ride_id <chr> "3564070EEFD12711", "0B820C7FCF22F489", "89EEEE3229...
## $ rideable_type <chr> "electric_bike", "classic_bike", "classic_bike", "c...
## $ started_at <chr> "2022-04-06 17:42:48", "2022-04-24 19:23:07", "2022...
## $ ended_at <chr> "2022-04-06 17:54:36", "2022-04-24 19:43:17", "2022...
## $ start_station_name <chr> "Paulina St & Howard St", "Wentworth Ave & Cermak R...
## $ start_station_id <chr> "515", "13075", "TA1307000121", "13075", "TA1307000...
## $ end_station_name <chr> "University Library (NU)", "Green St & Madison St",...
```

```
## $ end_station_id <chr> "605", "TA1307000120", "TA1307000120", "KA170600500...
## $ start_lat <dbl> 42.01913, 41.85308, 41.87184, 41.85308, 41.87181, 4...
## $ start_lng <dbl> -87.67353, -87.63193, -87.64664, -87.63193, -87.646...
## $ end_lat <dbl> 42.05294, 41.88189, 41.88189, 41.86749, 41.88224, 4...
## $ end_lng <dbl> -87.67345, -87.64879, -87.64879, -87.63219, -87.641...
## $ member_casual <chr> "member", "member", "member", "casual", "member", "...
```

```
glimpse(tripdata_202205)
```

```
## Rows: 634,858
## Columns: 13
## $ ride_id <chr> "EC2DE40644C6B0F4", "1C31AD03897EE385", "1542FBEC83...
## $ rideable_type <chr> "classic_bike", "classic_bike", "classic_bike", "cl...
## $ started_at <chr> "23/05/2022 23:06", "11/05/2022 8:53", "26/05/2022 ...
## $ ended_at <chr> "23/05/2022 23:40", "11/05/2022 9:31", "26/05/2022 ...
## $ start_station_name <chr> "Wabash Ave & Grand Ave", "DuSable Lake Shore Dr & ...
## $ start_station_id <chr> "TA1307000117", "13300", "TA1305000032", "TA1305000...
## $ end_station_name <chr> "Halsted St & Roscoe St", "Field Blvd & South Water...
## $ end_station_id <chr> "TA1309000025", "15534", "13221", "TA1305000030", "...
## $ start_lat <dbl> 41.89147, 41.88096, 41.88224, 41.88224, 41.88224, 4...
## $ start_lng <dbl> -87.62676, -87.61674, -87.64107, -87.64107, -87.641...
## $ end_lat <dbl> 41.94367, 41.88635, 41.90765, 41.88458, 41.88578, 4...
## $ end_lng <dbl> -87.64895, -87.61752, -87.67255, -87.63189, -87.651...
## $ member_casual <chr> "member", "member", "member", "member", "member", "...
```

```
glimpse(tripdata_202206)
```

```
## Rows: 769,204
## Columns: 13
## $ ride_id <chr> "600CFD130D0FD2A4", "F5E6B5C1682C6464", "B6EB6D27BA...
## $ rideable_type <chr> "electric_bike", "electric_bike", "electric_bike", ...
## $ started_at <chr> "30/06/2022 17:27", "30/06/2022 18:39", "30/06/2022...
## $ ended_at <chr> "30/06/2022 17:35", "30/06/2022 18:47", "30/06/2022...
## $ start_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ start_station_id <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ end_station_name <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ end_station_id <chr> "", "", "", "", "", "", "", "", "", "", "", "", "", ...
## $ start_lat <dbl> 41.89, 41.91, 41.91, 41.80, 41.91, 42.03, 41.91, 41...
## $ start_lng <dbl> -87.62, -87.62, -87.65, -87.66, -87.63, -87.71, -87...
## $ end_lat <dbl> 41.91, 41.93, 41.89, 41.80, 41.93, 42.06, 41.92, 41...
## $ end_lng <dbl> -87.62, -87.63, -87.61, -87.65, -87.64, -87.73, -87...
## $ member_casual <chr> "casual", "casual", "casual", "casual", "casual", "...
```

```
glimpse(tripdata_202207)
```

```
## Rows: 823,488
## Columns: 13
## $ ride_id <chr> "954144C2F67B1932", "292E027607D218B6", "5776585258...
## $ rideable_type <chr> "classic_bike", "classic_bike", "classic_bike", "cl...
## $ started_at <chr> "05/07/2022 8:12", "26/07/2022 12:53", "03/07/2022 ...
## $ ended_at <chr> "05/07/2022 8:24", "26/07/2022 12:55", "03/07/2022 ...
## $ start_station_name <chr> "Ashland Ave & Blackhawk St", "Buckingham Fountain ...
## $ start_station_id <chr> "13224", "15541", "15541", "15541", "TA1307000117",...
## $ end_station_name <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th St"...
## $ end_station_id <chr> "KA1503000043", "623", "623", "TA1307000164", "TA13...
```

```
## $ start_lat      <dbl> 41.90707, 41.86962, 41.86962, 41.86962, 41.89147, 4...
## $ start_lng      <dbl> -87.66725, -87.62398, -87.62398, -87.62398, -87.626...
## $ end_lat        <dbl> 41.88918, 41.87277, 41.87277, 41.79526, 41.93625, 4...
## $ end_lng        <dbl> -87.63851, -87.62398, -87.62398, -87.59647, -87.652...
## $ member_casual  <chr> "member", "casual", "casual", "casual", "member", "...

```

First the column names and datatypes are verified for consistency and then the formatting issues are dealt with. It can be seen that dataframes have time stamps in different formats. The dataframes for August, 2021 through October, 2021 and May, 2022 through July, 2022 have dates in Day/Month/Year Hour:Minute format. On the other hand the dataframes for November, 2021 through April, 2022 have dates in Year-Month-Day Hour:Minute:Second format. These data frames have to be made consistent to merge them into a single dataframe.

The started_at and ended_at column values are converted to POSIXct type using lubridate library functions. For dataframes with Day/Month/Year Hour:Minute format dmy_hm() function is used.

```
tripdata_202108$started_at <- dmy_hm(tripdata_202108$started_at)
tripdata_202108$ended_at <- dmy_hm(tripdata_202108$ended_at)

tripdata_202109$started_at <- dmy_hm(tripdata_202109$started_at)
tripdata_202109$ended_at <- dmy_hm(tripdata_202109$ended_at)

tripdata_202110$started_at <- dmy_hm(tripdata_202110$started_at)
tripdata_202110$ended_at <- dmy_hm(tripdata_202110$ended_at)

tripdata_202205$started_at <- dmy_hm(tripdata_202205$started_at)
tripdata_202205$ended_at <- dmy_hm(tripdata_202205$ended_at)

tripdata_202206$started_at <- dmy_hm(tripdata_202206$started_at)
tripdata_202206$ended_at <- dmy_hm(tripdata_202206$ended_at)

tripdata_202207$started_at <- dmy_hm(tripdata_202207$started_at)
tripdata_202207$ended_at <- dmy_hm(tripdata_202207$ended_at)

```

Similarly, for dataframes with Year-Month-Day Hour:Minute:Second format ymd_hms() function was used.

```
tripdata_202111$started_at <- ymd_hms(tripdata_202111$started_at)
tripdata_202111$ended_at <- ymd_hms(tripdata_202111$ended_at)

tripdata_202112$started_at <- ymd_hms(tripdata_202112$started_at)
tripdata_202112$ended_at <- ymd_hms(tripdata_202112$ended_at)

tripdata_202201$started_at <- ymd_hms(tripdata_202201$started_at)
tripdata_202201$ended_at <- ymd_hms(tripdata_202201$ended_at)

tripdata_202202$started_at <- ymd_hms(tripdata_202202$started_at)
tripdata_202202$ended_at <- ymd_hms(tripdata_202202$ended_at)

```

```
tripdata_202203$started_at <- ymd_hms(tripdata_202203$started_at)
tripdata_202203$ended_at <- ymd_hms(tripdata_202203$ended_at)

tripdata_202204$started_at <- ymd_hms(tripdata_202204$started_at)
tripdata_202204$ended_at <- ymd_hms(tripdata_202204$ended_at)
```

3. Process

Once all dataframes are consistent in their value formats, it's time to merge them into a single file for further cleaning and analysis using `rbind()` function.

```
df_12M <- rbind(tripdata_202108,
                tripdata_202109,
                tripdata_202110,
                tripdata_202111,
                tripdata_202112,
                tripdata_202201,
                tripdata_202202,
                tripdata_202203,
                tripdata_202204,
                tripdata_202205,
                tripdata_202206,
                tripdata_202207)
```

After merging the individual dataframes into one the station locations and IDs columns are to be dropped. The columns required for this analysis are the `ride_id`, `rideable_type`, `started_at`, `ended_at` and `member_casual`. Other needed information is to be extracted from the `started_at` and `ended_at` columns. The columns for start and end station information are dropped using following code.

```
df_12M <- df_12M[-c(5:12)]
```

The ride length was calculated as the difference of the `ended_at` and `started_at` column values. Moreover, the year-month, day-of-the-week and hour-of-the-day information was also calculated for purpose of analysis .

```
ride_length = as.numeric(df_12M$ended_at - df_12M$started_at)/60
start_date <- format(as.Date(df_12M$started_at))
start_year_month <- format(as.Date(df_12M$started_at), "%Y-%m")
start_year <- format(as.Date(df_12M$started_at), "%Y")
# start_month <- format(as.Date(df_12M$started_at), "%m")
start_month <- months(as.Date(df_12M$started_at))
start_day <- format(as.Date(df_12M$started_at), "%d")
start_day_of_week <- format(as.Date(df_12M$started_at), "%A")
start_hour <- format(as.POSIXct(df_12M$started_at), format = "%H")

df_12M <- add_column(df_12M, ride_length, .after = "ended_at")
df_12M <- add_column(df_12M, start_date, .after = "ride_length")
```



```
df_12M <- add_column(df_12M, start_year_month, .after = "start_date")
df_12M <- add_column(df_12M, start_year, .after = "start_year_month")
df_12M <- add_column(df_12M, start_month, .after = "start_year")
df_12M <- add_column(df_12M, start_day, .after = "start_month")
df_12M <- add_column(df_12M, start_day_of_week, .after = "start_day")
df_12M <- add_column(df_12M, start_hour, .after = "start_day_of_week")
```

Checking the data using summary() function reveals negative values in the ride_length column.

```
summary(df_12M)

##      ride_id      rideable_type      started_at
## Length:5901463 Length:5901463 Min. :2021-08-01 00:00:00.00
## Class :character Class :character 1st Qu.:2021-09-27 12:35:00.00
## Mode :character Mode :character Median :2022-02-14 14:10:08.00
##                                     Mean :2022-01-31 21:50:20.14
##                                     3rd Qu.:2022-06-05 15:29:00.00
##                                     Max. :2022-07-31 23:59:00.00
##      ended_at      ride_length      start_date
## Min. :2021-08-01 00:03:00.00 Min. : -138.00 Length:5901463
## 1st Qu.:2021-09-27 12:53:30.00 1st Qu.: 6.00 Class :character
## Median :2022-02-14 14:20:23.00 Median : 11.00 Mode :character
## Mean :2022-01-31 22:10:13.52 Mean : 19.89
## 3rd Qu.:2022-06-05 15:54:00.00 3rd Qu.: 20.00
## Max. :2022-08-04 13:53:00.00 Max. :41629.00
##      start_year_month      start_year      start_month      start_day
## Length:5901463 Length:5901463 Length:5901463 Length:5901463
## Class :character Class :character Class :character Class :character
## Mode :character Mode :character Mode :character Mode :character
##
##
##      start_day_of_week      start_hour      member_casual
## Length:5901463 Length:5901463 Length:5901463
## Class :character Class :character Class :character
## Mode :character Mode :character Mode :character
##
##
##
```

The observations with negative values are dropped off from the dataframe.

```
df_12M <- df_12M[!df_12M$ride_length<0,]
```

Finally before we visualize data the data has to be sorted with respect to the values in started_at column and renaming rows.

```
df_12M <- df_12M[order(df_12M$started_at),]
row.names(df_12M) <- NULL

df_Cyclistic <- df_12M
skim(df_Cyclistic)
```

Data summary

Name	df_Cyclistic
Number of rows	5901354
Number of columns	13

Column type frequency:


Character	10
Numeric	1
POSIXct	2

Group variables	None
-----------------	------

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	7	16	0	5901348	0
rideable_type	0	1	11	13	0	3	0
start_date	0	1	10	10	0	365	0
start_year_month	0	1	7	7	0	12	0
start_year	0	1	4	4	0	2	0
start_month	0	1	3	9	0	12	0
start_day	0	1	2	2	0	31	0
start_day_of_week	0	1	6	9	0	7	0
start_hour	0	1	2	2	0	24	0
member_casual	0	1	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
ride_length	0	1	19.89	147.99	0	6	11	20	41629	

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2021-08-01 00:00:00	2022-07-31 23:59:00	2022-02-14 14:14:26	1591909
ended_at	0	1	2021-08-01 00:03:00	2022-08-04 13:53:00	2022-02-14 14:26:40	1592456

4. Analyze

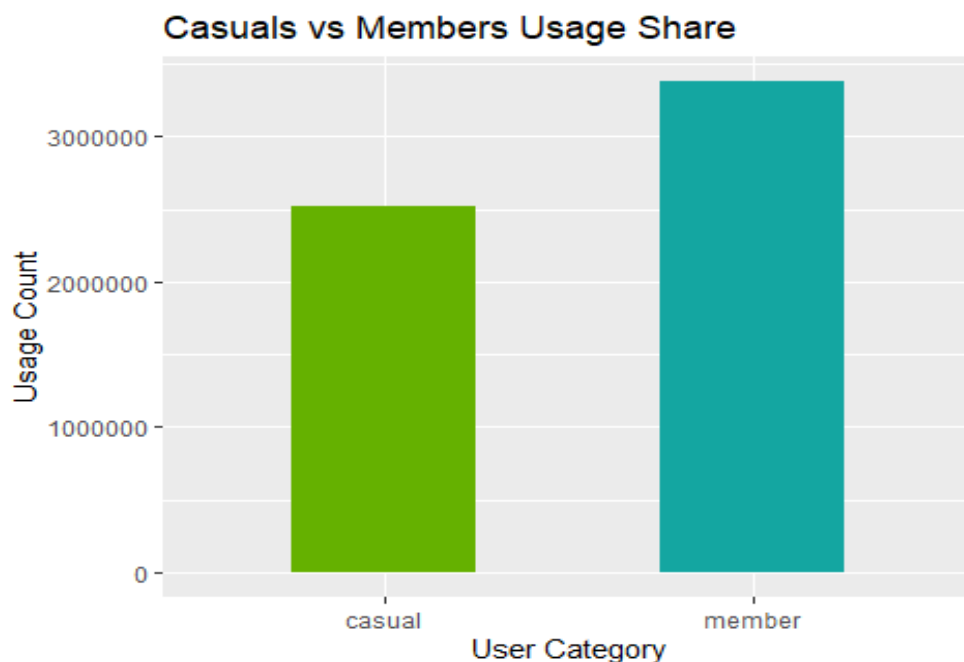
Now that our data is clean, we need to analyze our data and for that we are going to use ggplot library.

First we visualize usage by ride count for both user types.

```
# usage count by user type
usage_share <- df_Cyclistic %>%
  group_by(member_casual) %>%
  summarise(ride_count = length(ride_id),
            ride_percentage = (length(ride_id) / nrow(df_Cyclistic)) * 100)

uservsshare <- ggplot(usage_share) +
  geom_col(mapping=aes(x=member_casual,
                      y=ride_count,
                      fill=member_casual),
           width=0.5,
           position = position_dodge(width=0.5),
           show.legend = FALSE) +
  scale_fill_manual(values = c("casual" = "#65B100",
                              "member" = "#14A6A1"))+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  labs(x="User Category", y="Usage Count",
       title= "Casuals vs Members Usage Share")

uservsshare
```



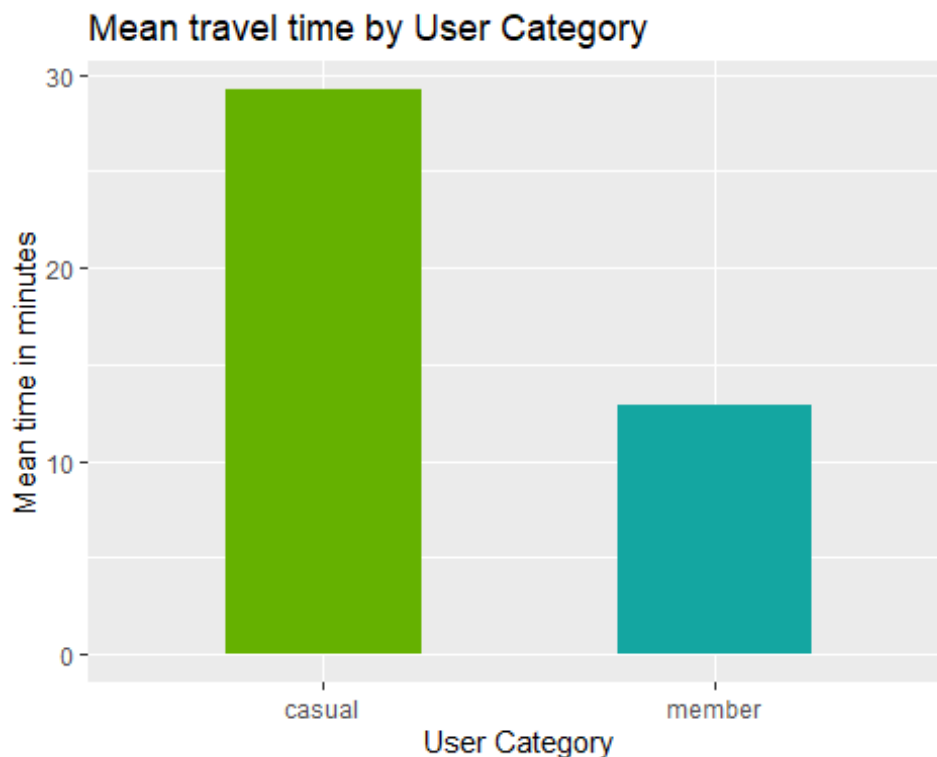
We see that members hold a greater share of Cyclistic bike rides. However there are two and a half million casual riders which is a huge client base that can be brought to the member category.

Next we look at the mean ride length for the two user categories.

```
mean_ride_length <- df_Cyclistic %>%
  group_by(member_casual) %>%
  summarise(mean_time = mean(ride_length))

usertypevstime <- ggplot(mean_ride_length) +
  geom_col(mapping=aes(x=member_casual, y=mean_time,
    fill=member_casual),
    width=0.5,
    position = position_dodge(width=0.5),
    show.legend = FALSE) +
  scale_fill_manual(values = c("casual" = "#65B100", "member" = "#14A6A1")) +
  labs(title = "Mean travel time by User Category", x="User Category", y="Mean
time in minutes")

usertypevstime
```

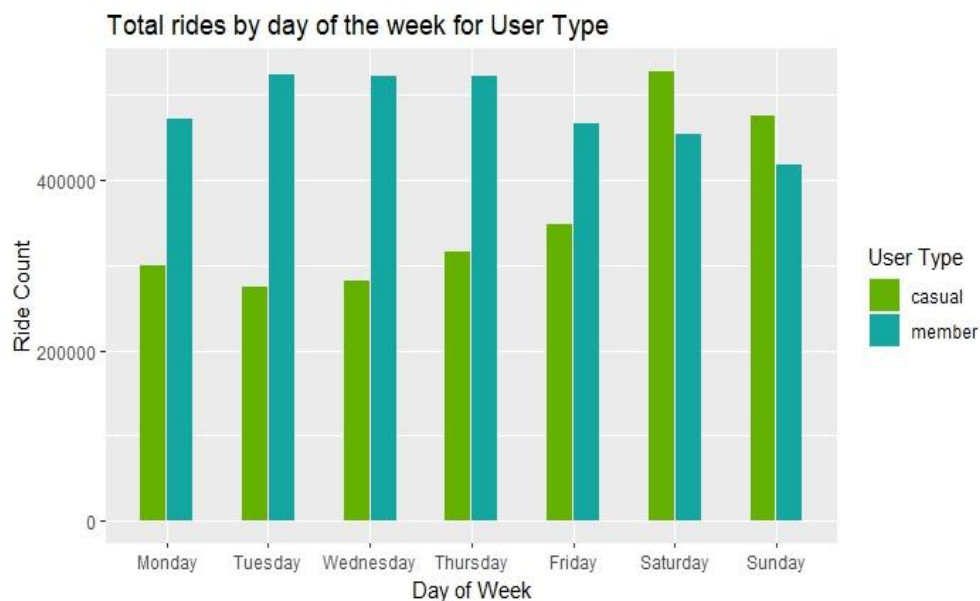


The chart indicates that casual riders spent much greater time on rides as compared to members which is another indicator that Cyclistic can gain much by bringing them to member category.

We need to look at the usage data for day-of-the-week as follows

```
# correcting the days of the week order.
df_Cyclistic$start_day_of_week <- ordered(df_Cyclistic$start_day_of_week,
                                           levels=c("Monday", "Tuesday",
                                                    "Wednesday", "Thursday",
                                                    "Friday", "Saturday",
                                                    "Sunday"))

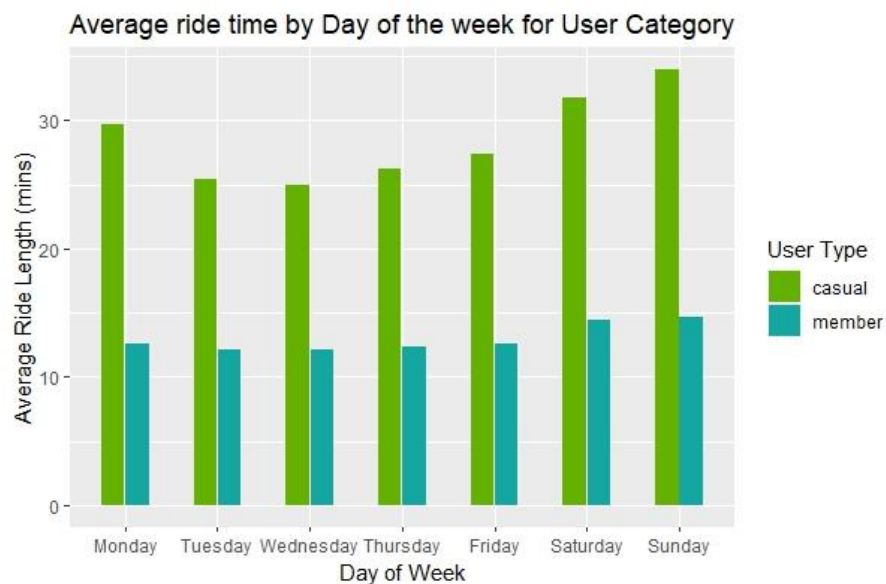
df_Cyclistic %>%
  group_by(member_casual, start_day_of_week) %>%
  summarise(number_of_rides = n(),
            .groups="drop") %>%
  arrange(member_casual, start_day_of_week) %>%
  ggplot(aes(x = start_day_of_week,
            y = number_of_rides,
            fill = member_casual)) +
  labs(title = "Total rides by day of the week for User Type",
       x = "Day of Week",
       y = "Ride Count") +
  geom_col(width=0.5,
           position = position_dodge(width=0.525)) +
  scale_y_continuous(labels = function(x) format(x,
                                                  scientific = FALSE)) +
  scale_fill_manual(values = c("casual" = "#65B100",
                              "member" = "#14A6A1")) +
  guides(fill=guide_legend(title="User Type"))
```



The usage pattern of members vs casual over the week is quite interesting. We see that the usage by members increases through mid-week and then declines over the weekend. On the other hand the casual riders have a much increased usage on the weekends.

Let's take a look at average ride time for each user category.

```
df_Cyclistic %>%
  group_by(member_casual, start_day_of_week) %>%
  summarise(average_ride_length = mean(ride_length),
            .groups="drop") %>%
  ggplot(aes(x = start_day_of_week,
            y = average_ride_length,
            fill = member_casual)) +
  geom_col(width=0.5,
           position = position_dodge(width=0.525)) +
  labs(title = "Average ride time by Day of the week for User Category",
       x = "Day of Week",
       y = "Average Ride Length (mins)") +
  scale_fill_manual(values = c("casual" = "#65B100",
                              "member" = "#14A6A1")) +
  guides(fill=guide_legend(title="User Type"))
```



The average ride length for members is mostly between 10 to 15 minutes throughout the week. This is almost twice for the casual riders.

Next we check the ride count and ride length for both user categories by month.

```
# correct the month order for the past year.
df_Cyclistic$start_month <- ordered(df_Cyclistic$start_month,
                                   levels=c("January", "February",
                                             "March", "April", "May",
                                             "June", "July", "August",
                                             "September", "October",
                                             "November", "December"))

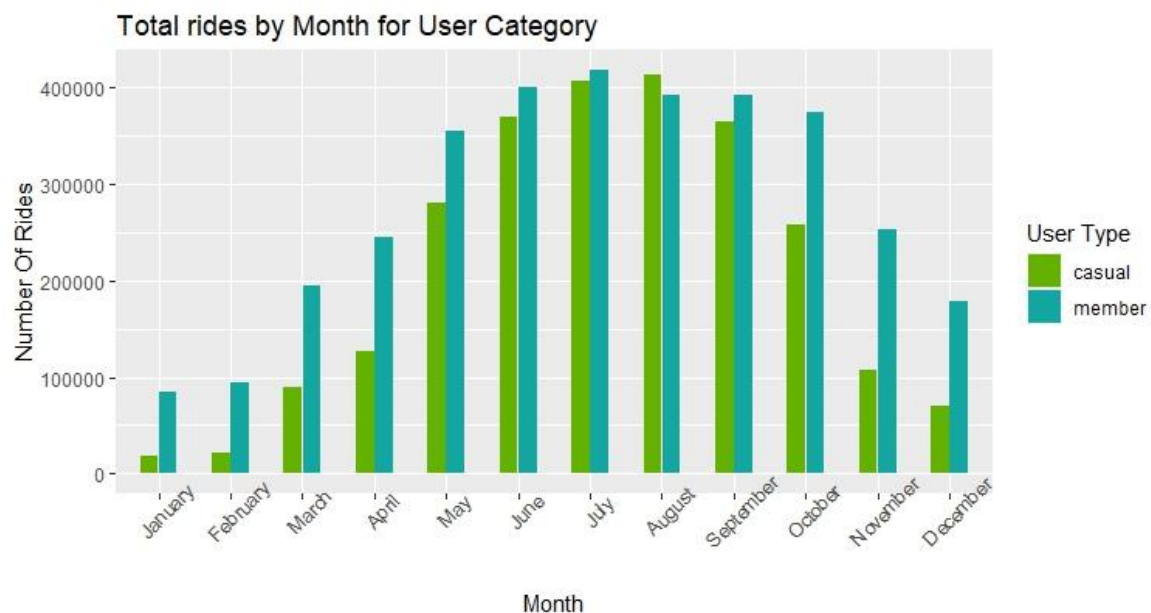
df_Cyclistic %>%
  group_by(member_casual, start_month) %>%
```

```

summarise(number_of_rides = n(),
           .groups="drop") %>%
arrange(member_casual, start_month) %>%
ggplot(aes(x = start_month,
           y = number_of_rides,
           fill = member_casual)) +
geom_col(width=0.50,
         position = position_dodge(width=0.525)) +
labs(title = "Total rides by Month for User Category",
     x = "Month",
     y = "Number Of Rides") +
theme(axis.text.x = element_text(angle = 45)) +

scale_y_continuous(labels = function(x) format(x,
                                                scientific = FALSE))+
scale_fill_manual(values = c("casual" = "#65B100",
                             "member" = "#14A6A1"))+
guides(fill=guide_legend(title="User Type"))

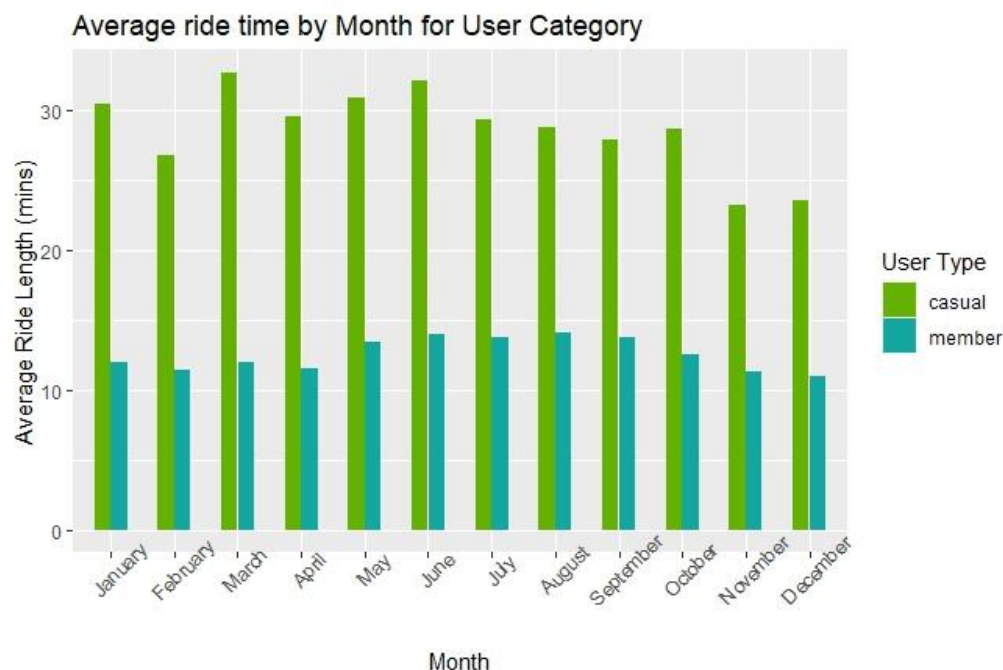
```



Interestingly, summers have a considerably high usage as compared to winters for both types of users. This indicates that it is difficult to ride during cold weather. During summers the number of rides for both user types is in close proximity. However, the usage by casual riders drops to almost one fourth of that of members during winters.

Now we look at the ride length by months for user category.

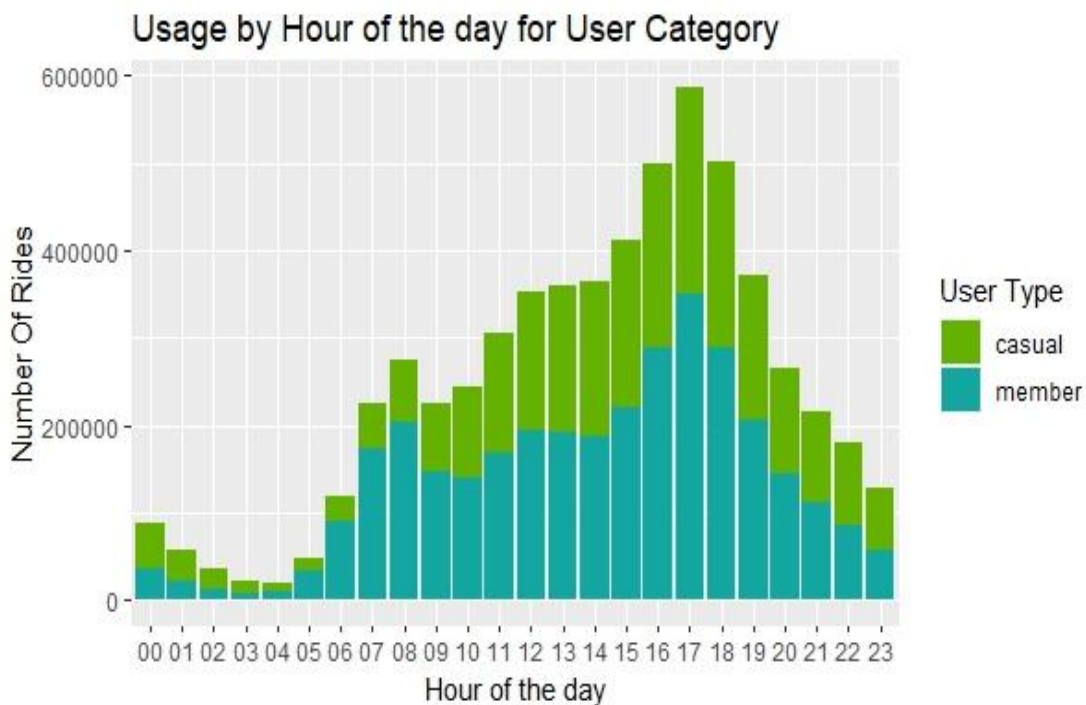
```
df_Cyclistic %>%
  group_by(member_casual, start_month) %>%
  summarise(average_ride_length = mean(ride_length),
            .groups="drop") %>%
  ggplot(aes(x = start_month,
            y = average_ride_length,
            fill = member_casual)) +
  geom_col(width=0.5,
           position = position_dodge(width=0.525)) +
  labs(title = "Average ride time by Month for User Category",
       x = "Month",
       y = "Average Ride Length (mins)") +
  theme(axis.text.x = element_text(angle = 45)) +
  scale_y_continuous(labels = function(x) format(x,
                                                  scientific = FALSE)) +
  scale_fill_manual(values = c("casual" = "#65B100",
                              "member" = "#14A6A1")) +
  guides(fill=guide_legend(title="User Type"))
```



The average monthly ride lengths also indicate that casual riders spend at least twice the time riding as compared to members.

We also need to know the time-of-day difference between the rider categories,

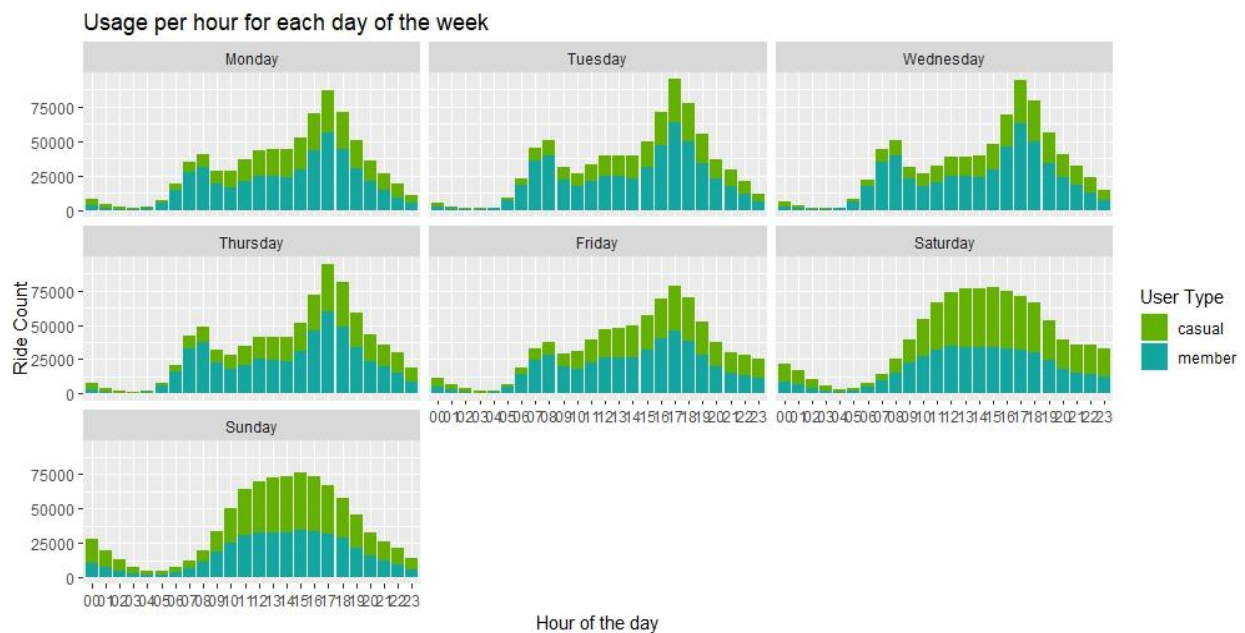
```
df_Cyclistic %>%  
  ggplot(aes(start_hour,  
             fill= member_casual)) +  
  labs(title="Usage by Hour of the day for User Category",  
       x = "Hour of the day",  
       y = "Number Of Rides") +  
  geom_bar()+  
  scale_y_continuous(labels = function(x) format(x,  
                                                  scientific = FALSE))+  
  scale_fill_manual(values = c("casual" = "#65B100",  
                              "member" = "#14A6A1"))+  
  guides(fill=guide_legend(title="User Type"))
```



Members mostly use the service during office times i.e. 0700 hrs to 1900 hrs. Casual riders have a maximum usage from 1100 hrs to 2300 hrs.

Let's observe hourly usage by day of the week

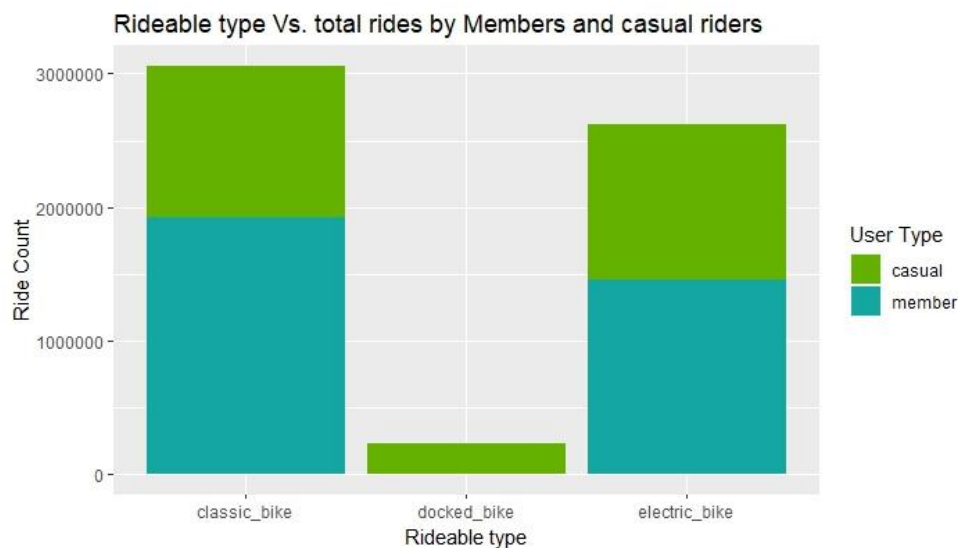
```
df_Cyclistic %>%
  ggplot(aes(start_hour,
             fill=member_casual)) +
  geom_bar() +
  labs(x="Hour of the day",
       y="Ride Count",
       title="Usage per hour for each day of the week") +
  facet_wrap(~ start_day_of_week) +
  scale_y_continuous(labels = function(x) format(x,
                                                  scientific = FALSE))+
  scale_fill_manual(values = c("casual" = "#65B100",
                              "member" = "#14A6A1"))+
  guides(fill=guide_legend(title="User Type"))
```



The difference between weekends and weekdays is clearly visible as the rides increase for casual riders over the weekends. The peak ride count hits between 1500 and 1700 hrs almost always.

The rideable type used by each user category must also be analyzed.

```
ggplot(df_Cyclistic, aes(x=rideable_type, fill=member_casual)) +  
  geom_bar() +  
  labs(x="Rideable type",  
       y="Ride Count",  
       title="Rideable type Vs. total rides by Members and casual riders") +  
  scale_y_continuous(labels = function(x) format(x,  
                                                scientific = FALSE))+  
  scale_fill_manual(values = c("casual" = "#65B100",  
                               "member" = "#14A6A1"))+  
  guides(fill=guide_legend(title="User Type"))
```



Docked bikes are only used by casual riders. Classic bikes have a higher share of usage by members as compared to electric bike. Casual riders almost have equal share in both classic and electric bike rides.

5. Share

The analysis has brought us following insights into the Cyclistic user community.

- i. Members are a greater portion of the Cyclistic users.
- ii. Members mostly use the rides during office hours.
- iii. Members mostly use the rides during weekdays.
- iv. Casual riders are frequent users during afternoons, evenings and nights.
- v. Casual riders are frequent users during weekends.
- vi. Casual riders normally ride for longer time periods.
- vii. Docked bikes are popular with casual riders as compared to members.
- viii. Greater volume of riders is observed in the afternoon.

6. Act

Following are my recommendations based on the analysis.

- i. Membership discounts for riding beyond certain distance as casual riders prefer long rides.
- ii. Offer warm riding gear for members during winter seasons.
- iii. Limited membership periods may be introduced for casual riders such as weekends.
- iv. Membership incentives for casual riders based on the time of the day they mostly prefer to ride.

Acknowledgments

I would like to thank the [course](#) instructors and management for creating a great learning experience. Also the [kaggle](#) community, [github](#) community and the internet users in general deserve much appreciation for sharing their wonderful works.