Cyclistic Analysis

Introduction

Cyclistic is 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 them to commute to work each day. Moreno is The director of marketing and my 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.

About the company

In 2016, Cyclistic launched a successful bike-share offering. 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 flexibility 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 finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility 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 all-new customers, Moreno believes there is a very good chance 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 marketing analyst 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.

Moreno has assinged me the first question to answer: how annual members and casual riders differ.

Business Task

The purpose of this report is to analyse the data on Cyclistic bike usage for the first quarter of 2023 to understand how annual members and casual users interact with the bike-sharing service. The main goal is to identify usage patterns, station preferences, behavioral differences and opportunities for improvement, to optimize the user experience and encourage long-term subscription by identifying how annual members and casual riders differ.

About the data

I will use Cyclistic's historical trip data to analyze and identify trends. The data has been made available by Motivate International Inc. Bikeshare hereby that grants to me a non-exclusive, royalty-free, limited,

perpetual license to access, reproduce, analyze, copy, modify, distribute in my product or service and use the Data for any lawful purpose ("License"). The dataset are organised by month, and I decided to analyse the first quarter of this year.

Setting the environment

Installing packages and loading libraries

For the following analysis I chose to install the following packages and load the related libraries:

```
install.packages("tidyverse") # For data manipulation and visualisation
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```
install.packages("ggplot2") # For data visualisation
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```
install.packages("dplyr") # For data manipulation
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```
install.packages("lubridate") # For data function
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```
install.packages("geosphere")
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```
install.packages("janitor")
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```r
install.packages("ggmap")
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```r
install.packages("osmdata")
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```r
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- 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
```

```r
library(ggplot2)
library(dplyr)
library(readr)
library(lubridate)
library(geosphere)
```

```
## The legacy packages maptools, rgdal, and rgeos, underpinning the sp package,
## which was just loaded, will retire in October 2023.
## Please refer to R-spatial evolution reports for details, especially
## https://r-spatial.org/r/2023/05/15/evolution4.html.
## It may be desirable to make the sf package available;
## package maintainers should consider adding sf to Suggests:.
## The sp package is now running under evolution status 2
##      (status 2 uses the sf package in place of rgdal)
```

```r
library(janitor)
```

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

```r
library(ggmap)
```

```
## i Google's Terms of Service: <https://mapsplatform.google.com>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```r
library(osmdata)
```

```
## Data (c) OpenStreetMap contributors, ODbL 1.0. https://www.openstreetmap.org/copyright
```

Import the datasets

Now I'll import the 4 data set relating to the first for months of 2023.

```r
list.files()
```

```
##  [1] "chicago_map.html"            "chicago_map.png"
##  [3] "Cyclistic_analysis.html"     "Cyclistic_analysis.Rmd"
##  [5] "Cyclistic-Analysis_files"    "CyclisticAnalysis_prova.log"
##  [7] "CyclisticAnalysis_prova.pdf" "CyclisticAnalysis_prova.Rmd"
##  [9] "CyclisticAnalysis_prova.tex" "dataset"
## [11] "documenti"                   "report_finale"
```

```r
gen <- read_csv("dataset/gennaio.csv")
```

```
## Rows: 190301 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
feb <- read_csv("dataset/febbraio.csv")
```

```
## Rows: 190445 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
mar <- read_csv("dataset/marzo.csv")
```

```
## Rows: 258678 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
apr <- read_csv("dataset/aprile.csv")
```

```
## Rows: 426590 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Wrangle data and combine them into a single dataset

Before I combine the datasets, I need to compare the column names in each of the files and make sure they match perfectly.

```
colnames(gen)
```

```
##  [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"
```

```
colnames(feb)
```

```
##  [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"
```

```
colnames(mar)
```

```
##  [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)
```

```
##  [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"
```

They seems to be pretty consistent.

Inspect the dataframes and look for incongruencies

I compare the dataset and see if there are any inconsistency to possibly eliminate them.

```r
str(gen)
```

```
## spc_tbl_ [190,301 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:190301] "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C907
##  $ rideable_type     : chr [1:190301] "electric_bike" "classic_bike" "electric_bike" "classic_bike"
##  $ started_at        : POSIXct[1:190301], format: "2023-01-21 20:05:42" "2023-01-10 15:37:36" ...
##  $ ended_at          : POSIXct[1:190301], format: "2023-01-21 20:16:33" "2023-01-10 15:46:05" ...
##  $ start_station_name: chr [1:190301] "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western
##  $ start_station_id  : chr [1:190301] "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr [1:190301] "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli P
##  $ end_station_id    : chr [1:190301] "202480.0" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat         : num [1:190301] 41.9 41.8 42 41.8 41.8 ...
##  $ start_lng         : num [1:190301] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num [1:190301] 41.9 41.8 42 41.8 41.8 ...
##  $ end_lng           : num [1:190301] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:190301] "member" "member" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```r
str(feb)
```

```
## spc_tbl_ [190,445 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
```

```
##  $ ride_id           : chr [1:190445] "CBCD0D7777F0E45F" "F3EC5FCE5FF39DE9" "E54C1F27FA9354FF" "3D56
##  $ rideable_type     : chr [1:190445] "classic_bike" "electric_bike" "classic_bike" "electric_bike"
##  $ started_at        : POSIXct[1:190445], format: "2023-02-14 11:59:42" "2023-02-15 13:53:48" ...
##  $ ended_at          : POSIXct[1:190445], format: "2023-02-14 12:13:38" "2023-02-15 13:59:08" ...
##  $ start_station_name: chr [1:190445] "Southport Ave & Clybourn Ave" "Clarendon Ave & Gordon Ter" "S
##  $ start_station_id  : chr [1:190445] "TA1309000030" "13379" "TA1309000030" "TA1309000030" ...
##  $ end_station_name  : chr [1:190445] "Clark St & Schiller St" "Sheridan Rd & Lawrence Ave" "Aberdee
##  $ end_station_id    : chr [1:190445] "TA1309000024" "TA1309000041" "13156" "TA1309000008" ...
##  $ start_lat         : num [1:190445] 41.9 42 41.9 41.9 41.8 ...
##  $ start_lng         : num [1:190445] -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:190445] 41.9 42 41.9 41.9 41.8 ...
##  $ end_lng           : num [1:190445] -87.6 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:190445] "casual" "casual" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(mar)
```

```
## spc_tbl_ [258,678 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:258678] "6842AA605EE9FBB3" "F984267A75B99A8C" "FF7CF57CFE026D02" "6B61
##  $ rideable_type     : chr [1:258678] "electric_bike" "electric_bike" "classic_bike" "classic_bike"
##  $ started_at        : POSIXct[1:258678], format: "2023-03-16 08:20:34" "2023-03-04 14:07:06" ...
##  $ ended_at          : POSIXct[1:258678], format: "2023-03-16 08:22:52" "2023-03-04 14:15:31" ...
##  $ start_station_name: chr [1:258678] "Clark St & Armitage Ave" "Public Rack - Kedzie Ave & Argyle S
##  $ start_station_id  : chr [1:258678] "13146" "491" "620" "TA1306000003" ...
##  $ end_station_name  : chr [1:258678] "Larrabee St & Webster Ave" NA "Clark St & Randolph St" "Sheff
##  $ end_station_id    : chr [1:258678] "13193" NA "TA1305000030" "13154" ...
##  $ start_lat         : num [1:258678] 41.9 42 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:258678] -87.6 -87.7 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:258678] 41.9 42 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:258678] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:258678] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
```

```
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(apr)
```

```
## spc_tbl_ [426,590 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:426590] "8FE8F7D9C10E88C7" "34E4ED3ADF1D821B" "5296BF07A2F77CB5" "4075
##  $ rideable_type     : chr [1:426590] "electric_bike" "electric_bike" "electric_bike" "electric_bike
##  $ started_at        : POSIXct[1:426590], format: "2023-04-02 08:37:28" "2023-04-19 11:29:02" ...
##  $ ended_at          : POSIXct[1:426590], format: "2023-04-02 08:41:37" "2023-04-19 11:52:12" ...
##  $ start_station_name: chr [1:426590] NA NA NA NA ...
##  $ start_station_id  : chr [1:426590] NA NA NA NA ...
##  $ end_station_name  : chr [1:426590] NA NA NA NA ...
##  $ end_station_id    : chr [1:426590] NA NA NA NA ...
##  $ start_lat         : num [1:426590] 41.8 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:426590] -87.6 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:426590] 41.8 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:426590] -87.6 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:426590] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# Convert ride_id and rideable_type to character so that they can stack correctly
gen <- mutate(gen, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type))
feb <- mutate(feb, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type))
mar <- mutate(mar, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type))
apr <- mutate(apr, ride_id = as.character(ride_id)
,rideable_type = as.character(rideable_type))
```

Stack individual quarter's dataframes into one big data frame

I combine the 4 datasets into a single one to make the analysis easier.

```
all_trips <- bind_rows(gen, feb, mar, apr)
```

Process the Data

Inspect the new table that has been created

Let's analyse the new table created in detail.

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

```
##  [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"
```

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

```
## [1] 1066014
```

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

```
## [1] 1066014      13
```

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

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 F96D5A74A3E41399 electric_bike 2023-01-21 20:05:42 2023-01-21 20:16:33
## 2 13CB7EB698CEDB88 classic_bike  2023-01-10 15:37:36 2023-01-10 15:46:05
## 3 BD88A2E670661CE5 electric_bike 2023-01-02 07:51:57 2023-01-02 08:05:11
## 4 C90792D034FED968 classic_bike  2023-01-22 10:52:58 2023-01-22 11:01:44
## 5 3397017529188E8A classic_bike  2023-01-12 13:58:01 2023-01-12 14:13:20
## 6 58E68156DAE3E311 electric_bike 2023-01-31 07:18:03 2023-01-31 07:21:16
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

```
tail(all_trips)
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 A17D800CE963661A classic_bike  2023-04-11 15:46:42 2023-04-11 15:50:03
## 2 8B441A6C436E9900 classic_bike  2023-04-29 21:20:21 2023-04-29 21:30:19
## 3 3980D64BE11540F1 classic_bike  2023-04-24 09:16:05 2023-04-24 09:22:27
## 4 3EF4B49FF7DAA02C classic_bike  2023-04-18 07:53:51 2023-04-18 07:59:16
## 5 210B2ED6583DC231 classic_bike  2023-04-29 07:33:55 2023-04-29 07:38:57
## 6 D29CB39B9E3FC46A electric_bike 2023-04-18 08:00:32 2023-04-18 08:02:35
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

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

```
## tibble [1,066,014 x 13] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1066014] "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C90
##  $ rideable_type     : chr [1:1066014] "electric_bike" "classic_bike" "electric_bike" "classic_bike"
##  $ started_at        : POSIXct[1:1066014], format: "2023-01-21 20:05:42" "2023-01-10 15:37:36" ...
##  $ ended_at          : POSIXct[1:1066014], format: "2023-01-21 20:16:33" "2023-01-10 15:46:05" ...
##  $ start_station_name: chr [1:1066014] "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western
##  $ start_station_id  : chr [1:1066014] "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr [1:1066014] "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli
##  $ end_station_id    : chr [1:1066014] "202480.0" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat         : num [1:1066014] 41.9 41.8 42 41.8 41.8 ...
##  $ start_lng         : num [1:1066014] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num [1:1066014] 41.9 41.8 42 41.8 41.8 ...
##  $ end_lng           : num [1:1066014] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:1066014] "member" "member" "casual" "member" ...
```

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

```
##    ride_id          rideable_type        started_at
##  Length:1066014     Length:1066014     Min.   :2023-01-01 00:01:58.00
##  Class :character   Class :character   1st Qu.:2023-02-12 20:45:29.75
##  Mode  :character   Mode  :character   Median :2023-03-20 16:08:39.00
##                                        Mean   :2023-03-13 00:09:46.90
##                                        3rd Qu.:2023-04-12 21:00:53.75
##                                        Max.   :2023-04-30 23:59:05.00
##
##     ended_at                      start_station_name start_station_id
##  Min.   :2023-01-01 00:02:41.00   Length:1066014     Length:1066014
##  1st Qu.:2023-02-12 21:04:59.50   Class :character   Class :character
##  Median :2023-03-20 16:20:48.00   Mode  :character   Mode  :character
##  Mean   :2023-03-13 00:24:34.89
##  3rd Qu.:2023-04-12 21:17:49.50
##  Max.   :2023-05-03 10:37:12.00
##
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:1066014     Length:1066014     Min.   :41.65   Min.   :-87.84
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.65
```

```
##                                      3rd Qu.:41.93   3rd Qu.:-87.63
##                                      Max.   :42.07   Max.   :-87.52
##
##     end_lat         end_lng       member_casual
##  Min.   :41.63   Min.   :-88.11   Length:1066014
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character
##  Median :41.90   Median :-87.64   Mode  :character
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.08   Max.   :-87.52
##  NA's   :861     NA's   :861
```

Consolidate labels

```
all_trips <- all_trips %>%
mutate(member_casual = recode(member_casual ,"Subscriber" = "member"
,"Customer" = "casual"))

table(all_trips$member_casual)
```

```
##
## casual member
## 292510 773504
```

Add columns that list the date, month, day, and year of each ride

The following step will allow me to aggregate ride data.

```
all_trips <- all_trips %>%
  mutate(year = format(as.Date(started_at), "%Y")) %>% # extract year
  mutate(month = format(as.Date(started_at), "%B")) %>% #extract month
  mutate(date = format(as.Date(started_at), "%d")) %>% # extract date
  mutate(day_of_week = format(as.Date(started_at), "%A")) %>% # extract day of week
  mutate(ride_length = difftime(ended_at, started_at)) %>%
  mutate(start_time = strftime(started_at, "%H"))
```

Now I will add a ride_lenght column to all_trips (in seconds):

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
```

Let's inspect the structure of the columns:

```
str(all_trips)
```

```
## tibble [1,066,014 x 19] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:1066014] "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E670661CE5" "C90
##  $ rideable_type     : chr [1:1066014] "electric_bike" "classic_bike" "electric_bike" "classic_bike"
##  $ started_at        : POSIXct[1:1066014], format: "2023-01-21 20:05:42" "2023-01-10 15:37:36" ...
##  $ ended_at          : POSIXct[1:1066014], format: "2023-01-21 20:16:33" "2023-01-10 15:46:05" ...
```

```
##  $ start_station_name: chr [1:1066014] "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "Western
##  $ start_station_id  : chr [1:1066014] "TA1309000058" "TA1309000037" "RP-005" "TA1309000037" ...
##  $ end_station_name  : chr [1:1066014] "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St" "Valli
##  $ end_station_id    : chr [1:1066014] "202480.0" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat         : num [1:1066014] 41.9 41.8 42 41.8 41.8 ...
##  $ start_lng         : num [1:1066014] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num [1:1066014] 41.9 41.8 42 41.8 41.8 ...
##  $ end_lng           : num [1:1066014] -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:1066014] "member" "member" "casual" "member" ...
##  $ year              : chr [1:1066014] "2023" "2023" "2023" "2023" ...
##  $ month             : chr [1:1066014] "January" "January" "January" "January" ...
##  $ date              : chr [1:1066014] "21" "10" "02" "22" ...
##  $ day_of_week       : chr [1:1066014] "Saturday" "Tuesday" "Monday" "Sunday" ...
##  $ ride_length       : 'difftime' num [1:1066014] 651 509 794 526 ...
##   ..- attr(*, "units")= chr "secs"
##  $ start_time        : chr [1:1066014] "21" "16" "08" "11" ...
```

Now, I need to convert "ride_length" from Factor to numeric so I can run calculations on the data

```
# Is the column ride_lenght a Factor?
is.factor(all_trips$ride_length)
```

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

```
# Converting in numeric
all_trips$ride_length <- as.numeric(as.character(all_trips$ride_length))

# Is the column ride_lenght numeric?
is.numeric(all_trips$ride_length)
```

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

Remove "bad" data

The dataframe includes a few hundred entries when bikes were taken out of docks and checked for quality by Divvy or ride_length was negative. I will create a new version of the dataframe (v2) since data is being removed.

```
all_trips_v2 <- all_trips[!(all_trips$start_station_name == "HQ QR" | all_trips$ride_length<0),]

# Remove NA values in the column ride_length
all_trips_v2 <- all_trips_v2[! is.na(all_trips_v2$ride_length) , ]
```

Analyse the Data

Descriptive analysis on ride_length (all figures in seconds)

All the required information are in one place and ready for exploration. Now I calculate the average, median, maximum and minimum duration for each ride.

```r
# Average ride length
mean(all_trips_v2$ride_length)  #straight average (total ride length / rides)
```

```
## [1] 926.9588
```

```r
# Median ride length
median(all_trips_v2$ride_length) # midpoint number in the ascending array of ride lengths
```

```
## [1] 483
```

```r
# Max ride length
max(all_trips_v2$ride_length) # longest ride
```

```
## [1] 2016224
```

```r
# Min ride length
min(all_trips_v2$ride_length) # shortest ride
```

```
## [1] 0
```

```r
# Condense the four lines above
summary(all_trips_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0     282     483     927     849 2016224
```

As we can see from the results above bout the 'ride_length' data regarding the first quarter of 2023 the longest and shortest rides have extreme values. Due to lack of information about them, it is not possible to find out the reasons behind it, but it need to be analysed further.

Comparison between members and casual users

```r
# Mean of ride length by user
mean_ride_length_by_user <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = mean)
print("Mean of ride length per kind of user:")
```

```
## [1] "Mean of ride length per kind of user:"
```

```r
print(mean_ride_length_by_user)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                1653.4144
## 2                     member                 656.2346
```

```r
# Median of ride length by user
median_ride_length <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = median)
print("Median of ride length by user:")
```

```
## [1] "Median of ride length by user:"
```

```r
print(median_ride_length)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                      609
## 2                     member                      447
```

```r
# Max of ride length by user
max_ride_length <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = max)
print("Max of ride length by user:")
```

```
## [1] "Max of ride length by user:"
```

```r
print(max_ride_length)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                  2016224
## 2                     member                    93580
```

```r
# Min of ride length by user
min_ride_length <- aggregate(all_trips_v2$ride_length ~ all_trips_v2$member_casual, FUN = min)
print("Min of ride lenth by user:")
```

```
## [1] "Min of ride lenth by user:"
```

```r
print(min_ride_length)
```

```
##   all_trips_v2$member_casual all_trips_v2$ride_length
## 1                     casual                        0
## 2                     member                        0
```

From the above data, we can conclude that casual riders have longer rides thatn annual members, as the average ride length and mean ride length is lower that the respective data of casual users.

```r
#  Average ride time by each day for members vs casual users
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                1462.3923
## 2                      member                   Friday                 650.1168
## 3                      casual                   Monday                1603.9670
## 4                      member                   Monday                 619.6024
## 5                      casual                 Saturday                2120.4003
```

```
## 6                       member                Saturday                740.3637
## 7                       casual                  Sunday               1928.2899
## 8                       member                  Sunday                727.3088
## 9                       casual                Thursday               1388.3986
## 10                      member                Thursday                635.5447
## 11                      casual                 Tuesday               1457.7345
## 12                      member                 Tuesday                633.6603
## 13                      casual               Wednesday               1460.7254
## 14                      member               Wednesday                632.9563
```

The days of the week are out of order. Let's fix that:

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

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               1928.2899
## 2                      member                  Sunday                727.3088
## 3                      casual                  Monday               1603.9670
## 4                      member                  Monday                619.6024
## 5                      casual                 Tuesday               1457.7345
## 6                      member                 Tuesday                633.6603
## 7                      casual               Wednesday               1460.7254
## 8                      member               Wednesday                632.9563
## 9                      casual                Thursday               1388.3986
## 10                     member                Thursday                635.5447
## 11                     casual                  Friday               1462.3923
## 12                     member                  Friday                650.1168
## 13                     casual                Saturday               2120.4003
## 14                     member                Saturday                740.3637
```

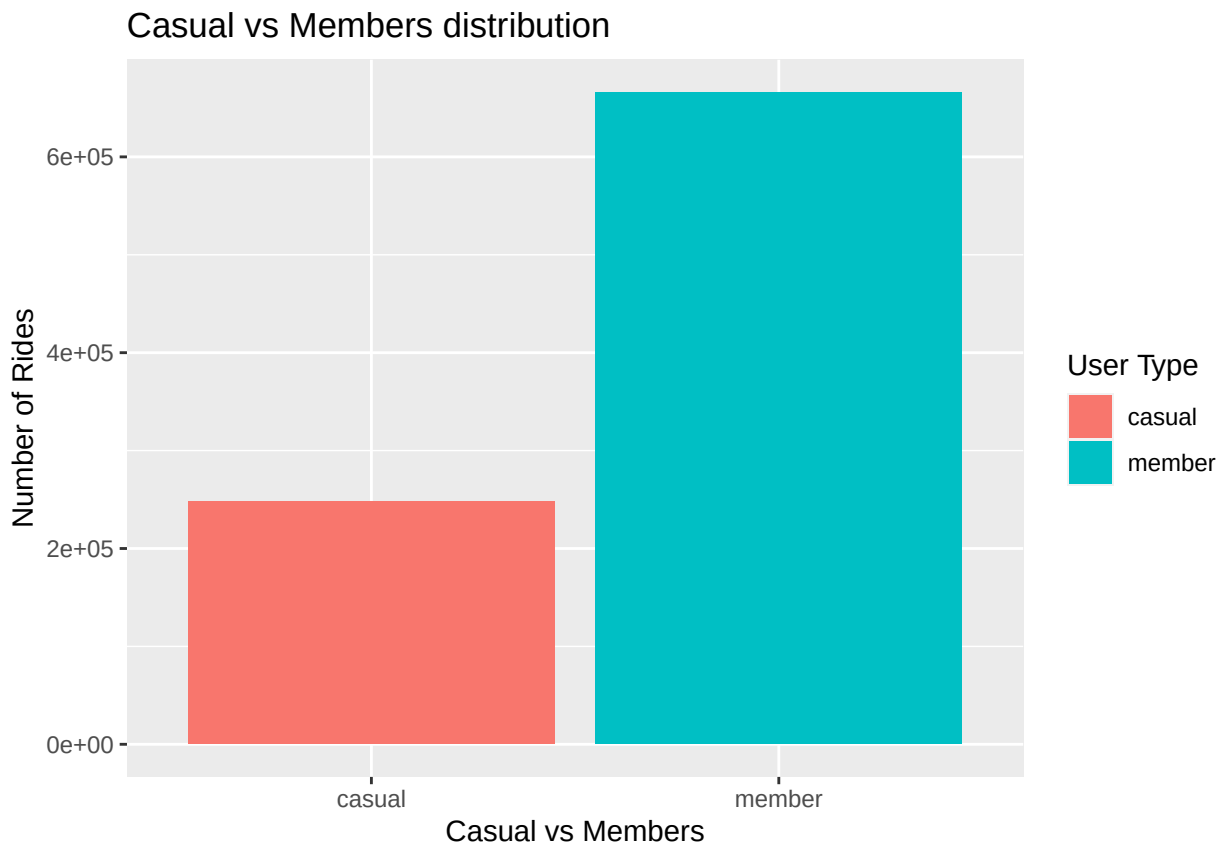
Analyse and visualise the Data

Analysis and visualisation of total rides taken by type of user

```
# Members vs causal riders difference depending on total rides taken
all_trips_v2 %>%
  group_by(member_casual) %>%
  summarise(ride_count =length(ride_id), ride_percentage = (length(ride_id) / nrow(all_trips_v2)) * 100)
```

```
## # A tibble: 2 x 3
##   member_casual ride_count ride_percentage
##   <chr>              <int>           <dbl>
## 1 casual            248167            27.1
## 2 member            665926            72.9
```

```
ggplot(all_trips_v2, aes(x = member_casual, fill = member_casual)) + geom_bar() + labs(x = "Casual vs M
```

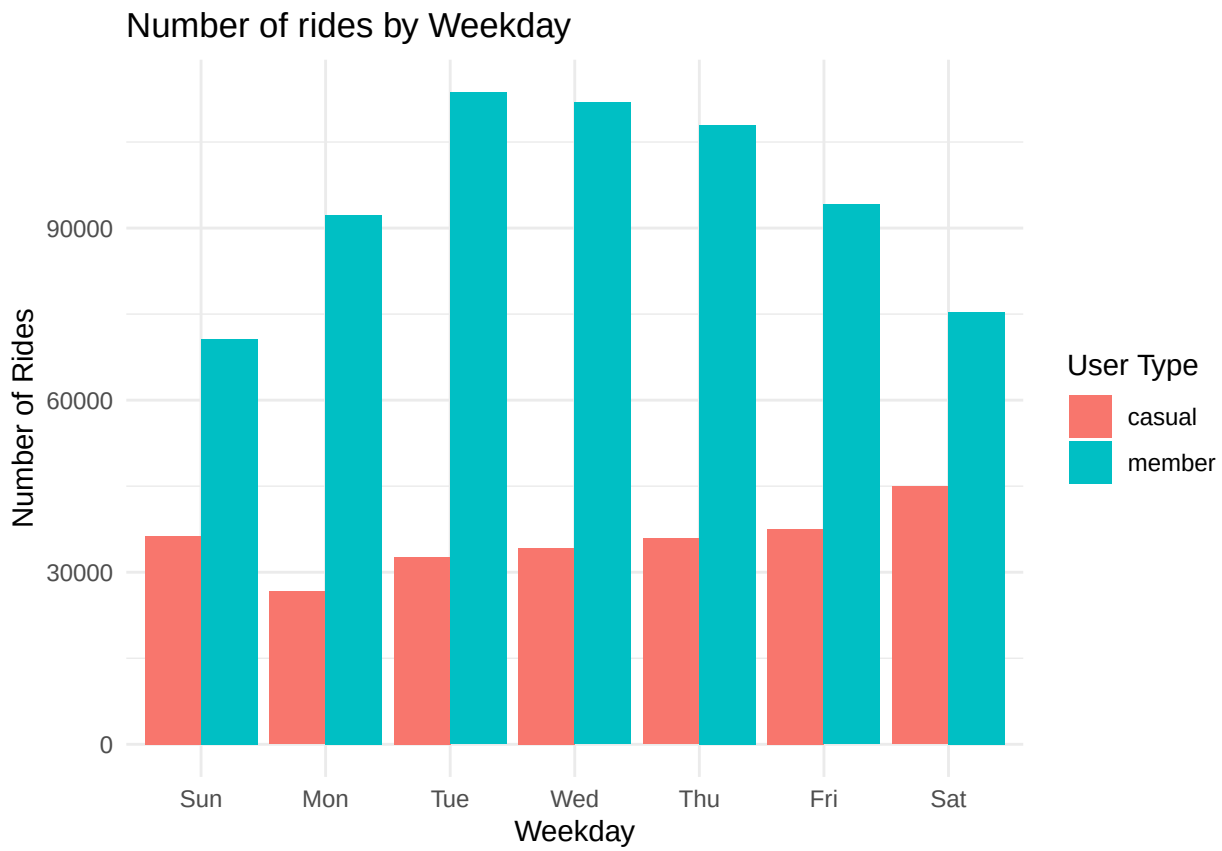


The bar chart above shows the distribution of Casual riders and Annual members depending on the rides taken between January and April. Overall, we can see that Annual Members are the most active users of the bike-sharing service. Casual users have a count of 248.167 rides taken, representing about the 27% of total rides. On the other hand, Annual members have a count more than the double of Casual users with a total of 665.926 representing about the 73% of total rides.

Analysis and visualization of the number of rides per weekday

```
# Visualise the number of rides per user type
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") + l
  ggtitle("Number of rides by Weekday") +
  theme_minimal()
```

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

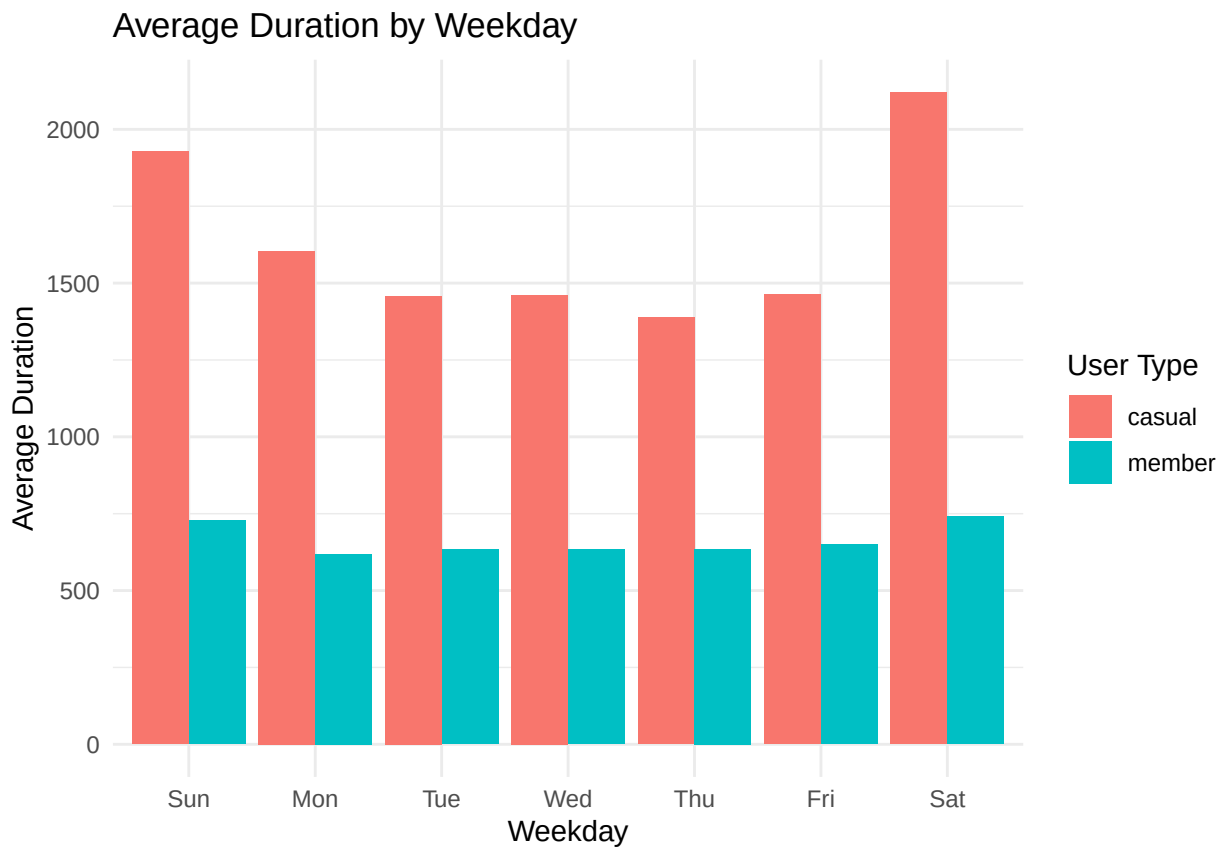


The previous chart represent the total rides taken from Sunday to Saturday in the first quarter of 2023. In general, annual members are the most active along the week, with an positive trend as the week starts, and a negative trend as the week ends, while casual users have pretty constant positive trend from Monday to Saturday, with a slightly negative variation on Sunday. Annual Members have a total number of rides which does not decrease less than about 70.000 rides a day, with an ascending trend till Tuesday, with slightly less than 120.000 rides a day. Instead, casual users have a total of rides that stays between just under 30.000 to 45.000 rides a day, with an increasing usage from Monday to Saturday.

Analysis and visualization for average duration

```
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") +
  ggtitle("Average Duration by Weekday") +
  theme_minimal()
```

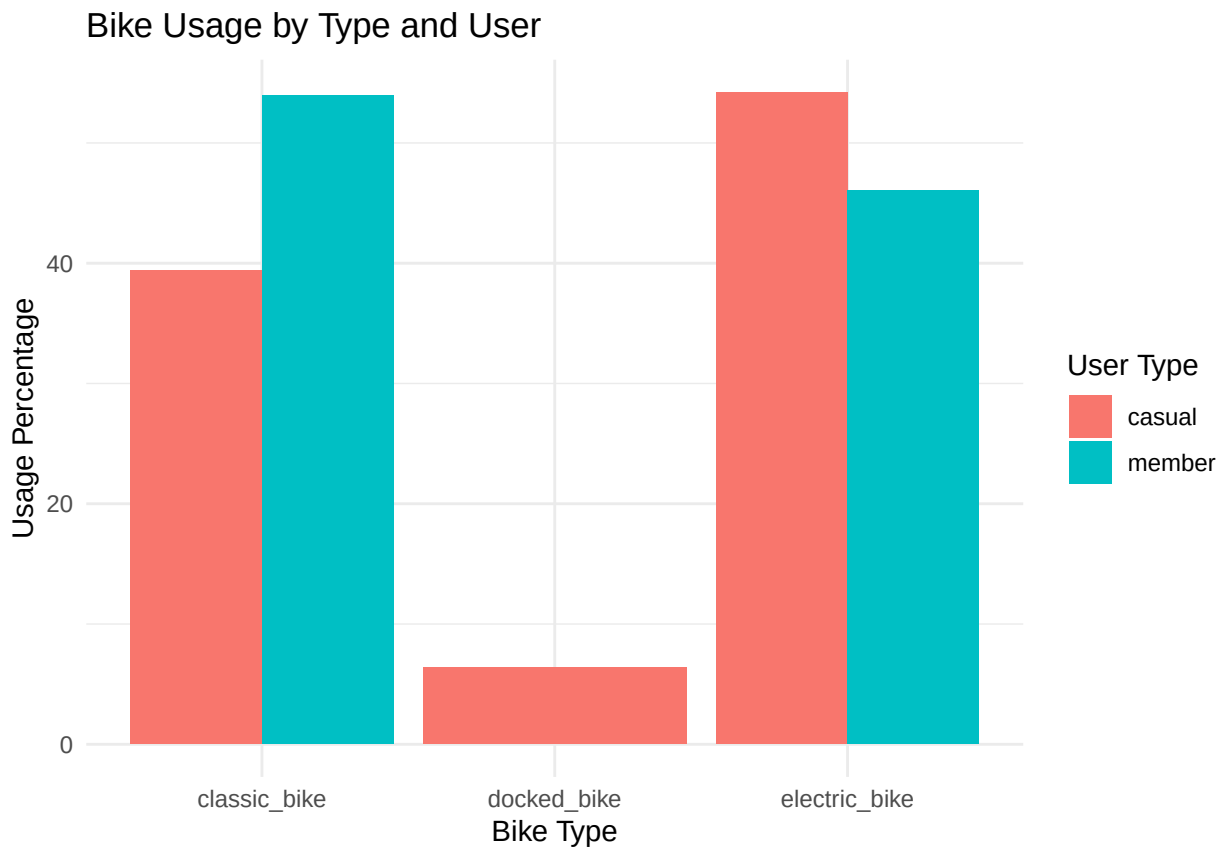
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



This bar chart shows the average ride length during a week in the first quarter. Overall, casual users have an average duration of rides that is more than doubled the average ride time for annual members. Despite annual members are the most active users during the week as we saw on the "Number of rides by Weekday" bar chart, casual users have the most longest rides in terms of duration. Annual members have, during the week, a variation of about 250 seconds of ride lengths with most longest rides on Saturday and Sunday. Casual users have a variation of about 900 seconds with the highest average duration on Saturday, with an average usage of slightly more than 2.000 seconds.

Analysis and visualisation of the frequency of each bike type for members and casual user

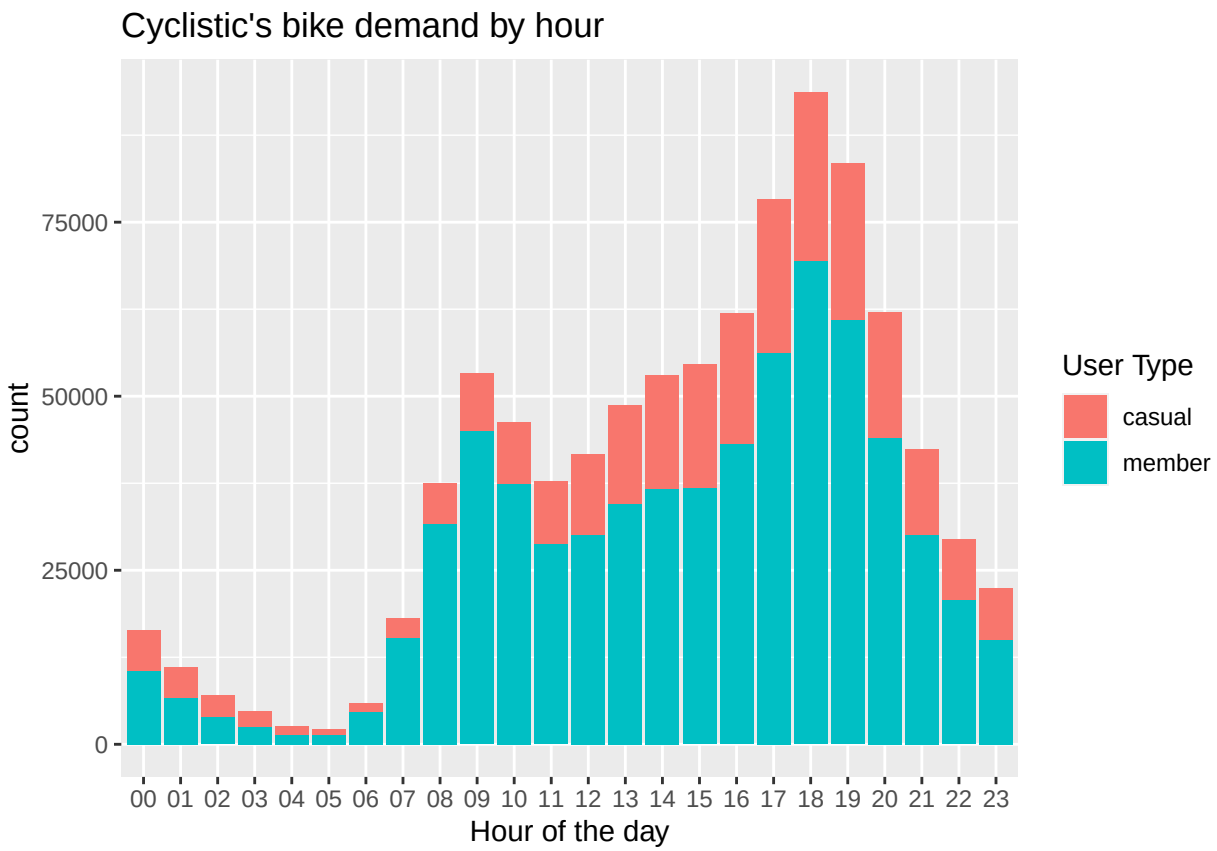
```r
# Calculate the usage frequency of each bike type for members and casual riders
bike_usage <- all_trips_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(frequency = n(), .groups = 'drop') %>%
  group_by(member_casual) %>%
  mutate(total_frequency = sum(frequency)) %>%
  mutate(percentage = (frequency / total_frequency) * 100)

# Create the bar plot
ggplot(bike_usage, aes(x = rideable_type, y = percentage, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "Bike Type", y = "Usage Percentage", fill = "User Type") +
  ggtitle("Bike Usage by Type and User") +
  theme_minimal()
```



This chart demonstrate that casual and annual members differs in the bike type usage also. While annual members prefer classic bikes, they also use electric bikes but don not use docked bikes. On the other hand casual riders prefer electric bikes but they use also classic and docked bike. Annual members use classic bike for the 55% of the times, while they use electric bike for a 45% of the times. Casual members use electric bike 55%, classic bikes 39% and docked bike 6% of the rides.

Analysis and visualisation on Cyclistic's bike demand by hour in a day

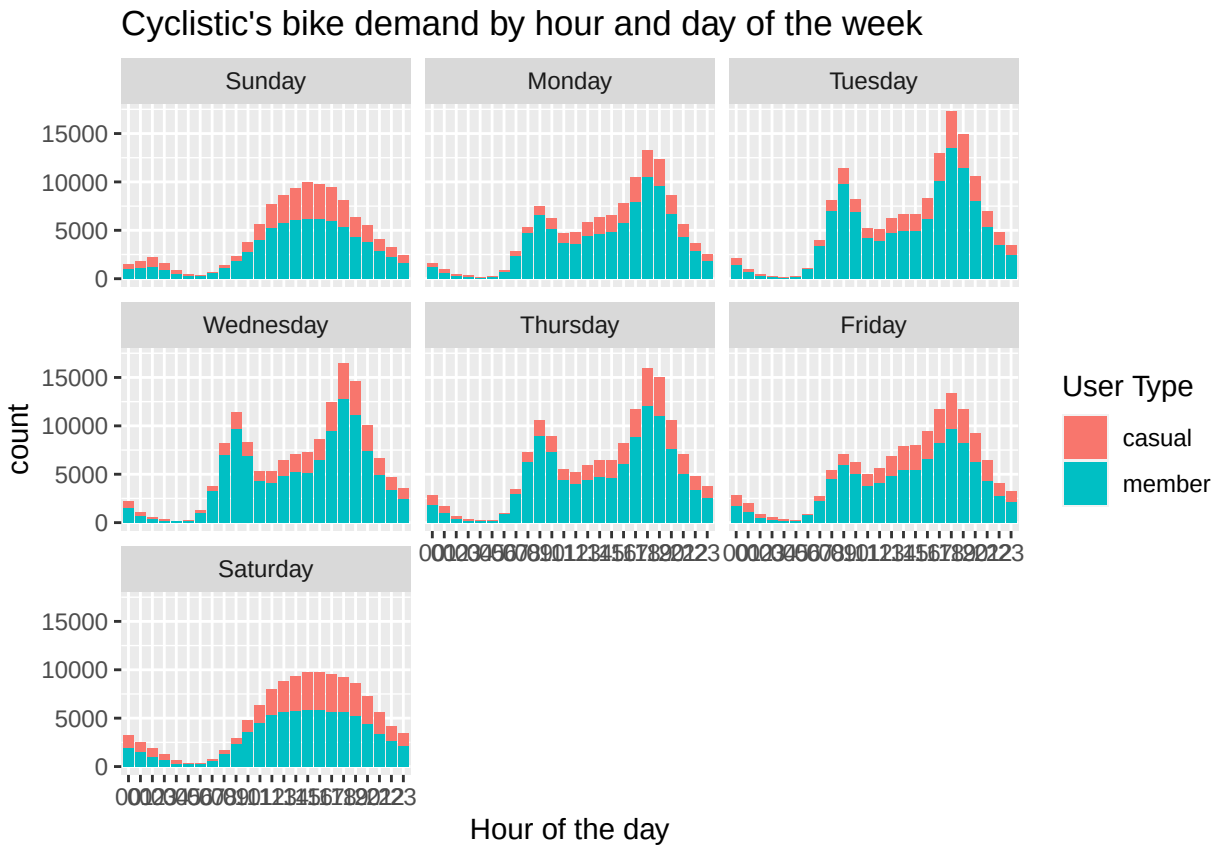
```
all_trips_v2 %>%
  ggplot(aes(x = start_time, fill = member_casual)) + labs(x = "Hour of the day", title = "Cyclistic's
```



This chart compare the bike demand by hour and user type during a day. Overall, annual members request the bike service more than casual riders during the day. As we can see, there are two peaks in usage during the day. The first peak is between 7AM and 10AM (usually the start time of work), and the second peak in the evening between 5PM and 7PM (end time of work). Despite this two peaks, there is a positive trend in usage from 10 AM to 5PM and a descending trend from 8PM to 11PM and in the early morning.

Analysis and visualisation on Cyclistic's bike demand per hour and day of the week

```
all_trips_v2 %>%
  ggplot(aes(x = start_time, fill = member_casual)) + labs(x = "Hour of the day", title = "Cyclistic's
```



These bar charts illustrate how is the distribution in usage by hour and day of the week. We can see two main differences, the first difference is the different trend of use during the week and on weekends. From Monday to Friday we can see (as we analysed before) that there are two main peaks of bike request, respectively the start and end time of work. The second difference is the amount of request of casual riders during the weekend and during week days. In fact, on Saturdays and Sundays the curve is made up of half annual members and the other half of casual users, while during the week most of the curve is made up by annual members.

Analysis and visualisation of the dataset on coordinate basis

```
# Let's create the coordinates data of the rides
#Adding a new data frame
coordinates_df <- all_trips_v2 %>%
  filter(start_lng != end_lng & start_lat != end_lat) %>%
  group_by(start_lng, start_lat, end_lng, end_lat, member_casual, rideable_type) %>%
  summarise(total_rides = n(), .groups = "drop") %>%
  filter(total_rides >100)

# Let's create two different dataframe depending on user type
casual_riders <- coordinates_df %>%
  filter(member_casual == "casual")
member_riders <- coordinates_df %>%
  filter(member_casual == "member")
```

```
##
```

```
## The downloaded binary packages are in
##  /var/folders/f6/_rc2zdsj0hg15tn45f_2chxh0000gp/T//RtmpSroxas/downloaded_packages
```

```
## [1] "chicago_map.png"
```

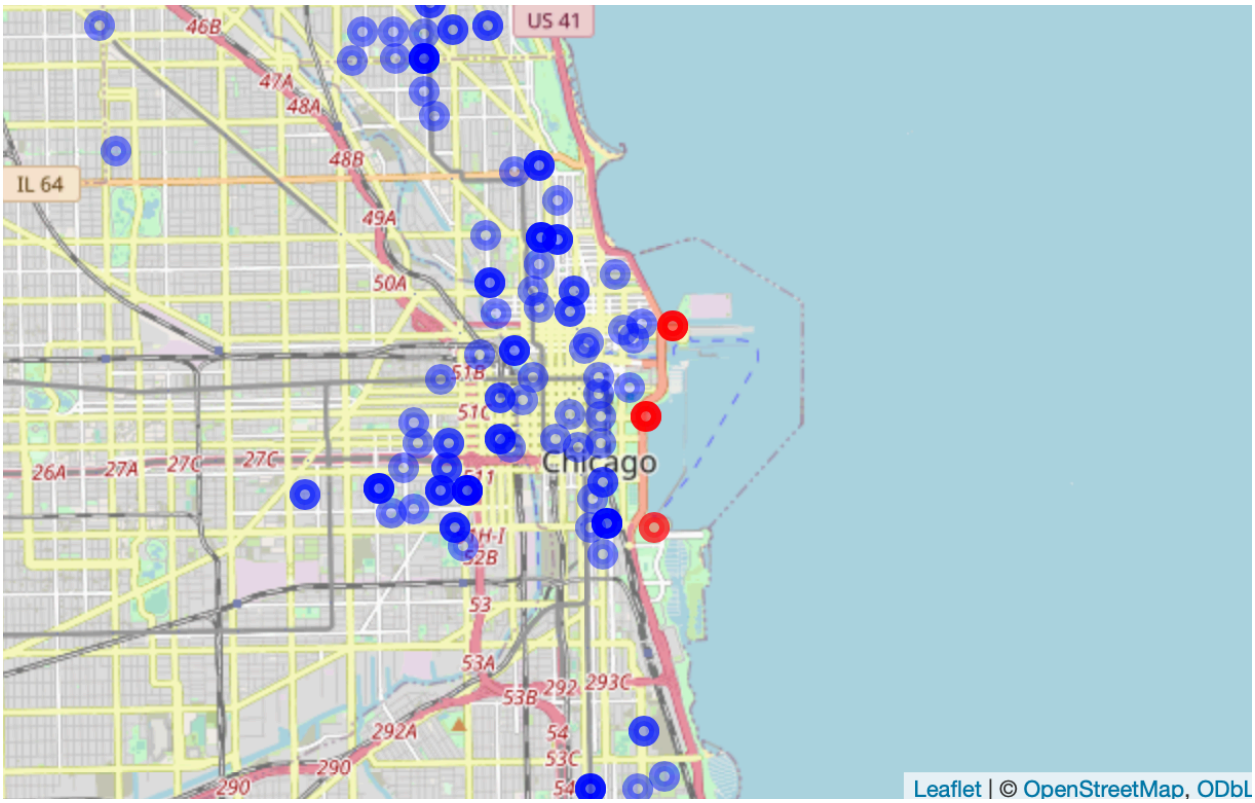


Figure 1: Mappa di Chicago

The map above shows the most common start points (>100) of casual and annual members in Chicago. Overall, annual members have a more fixed use for bikes and their preferred routes are distributed among all the city map. On the other hand, casual riders usually take their bike near the bay area and rarely on the city center.

Share insights

Main insights and finding conclsuions

- Annual members hold the biggest portion of total rides, about 73% of total rides
- Despite the total rides, casual user use the bike sharing service for bigger rides in terms of time duration.

- Annual members use only two of three type of bike (classic and electric). They prefer to use classic bike (55%).
- Casual user prefer electric bike (55%) but they use every type of bike.
- During the week there are two peaks in request during the day in the morning (8AM to 10 AM), and in the evening (5PM to 7PM). This is because 30% use the bike sharing service to commute to work.
- In weekends there is an equal distribution of users between casual and annual members
- Casual members usually start their rides near the bay area, while members have more fixed starting point among all the city.

Act

Conclusions and reccomendations

This three recommendations are data based and with the final purpose of convert casual riders into annual members.
1. Weekend Membership: offer a weekend-only annual membership with a different price than the full annual membership. 2. Targeted Communication: display coupons and discounts in the bay area with a special price or a trial period for casual rider, and highlight cost savings. Otherwise, start a mail marketing campaing. 3. Incentivise Frequent Usage: create a loyalty program that rewards electric bike usage with credits every mile ridden that can be spent to buy the annual membership.

*Note: All ride ids are unique, so we can't conclude if the same rider took several rides. More rider data are required for further analysis.