Cyclistic Bike Share Report

Tauan Oliveira

05-02-2023

1.Introduction

Project Description

In this project, I will perform a case study available by Projeto Final de Data Analytics do Google. In the project, I will be part of a marketing analyst team at Cyclistic, a fictional bike-share company in Chicago. The team will be responsible for understanding how casual riders and annual members use Cyclistic bikes differently. So, I will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act, to gain insights to answer the business task.

Scenario

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

2.Ask

The first step in the data analysis process is the - Ask phase. To solve this problem, we must first define the business task, in other words, the question or problem that the data analysis will deal with, but firstly, we need to understand the whole context where the problem is embedded. In this way, we certify that we're focusing on the problem to be solved.

Understanding the context

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 them to commute to work each day.

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 geo-tracked and locked bikes within a network of 692 stations in Chicago. The bikes can be unlocked from one station and returned to any other station in the system at any time.

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 that purchase single-ride or full-day passes refer to casual riders. Customers that purchase annual memberships are Cyclistic members.

Cyclistic's finance analysts have concluded that annual members are much more profitable than casual passengers. Although the pricing flexibility helps Cyclistic attract customers, Lily Moreno believes that maximizing the number of annual members will be the key to future growth. Rather than creating a marketing campaign that targets new customers, she believes there is a good chance to convert casual passengers into members. She notes that casual cyclists are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.

Stakeholders

- 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 - as well 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.

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.

Three questions will guide the future marketing program:

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

Moreno has assigned you the first question to answer: How do annual members and casual riders use Cyclistic bikes differently?

With the business task defined, I will create a report with the following deliverables:

1. A clear statement of the business task
2. A description of all data sources used
3. Documentation of any cleaning or manipulation of data
4. A summary of your analysis
5. Supporting visualizations and key findings
6. Your top three recommendations based on your analysis

Business task

Moreno has set the following business task: understand how annual members and casual riders use Cyclistic bikes differently, intending to create a new marketing strategy to convert casual riders into annual members.

3.Prepare

Now, we are in the second data analysis phase - Prepare. In this phase, I will collect data to perform the analysis. So, I will gather Cyclistic's trip historical data in the last 12 months.

Download data and store it appropriately

All data collected was stored in my kaggle account. Thus all the data is stored in the cloud to ensure data security.

Identify how data it's organized

The data is in .csv files (comma-separated values) and is organized in rows and columns, which means it is structured. All collected datasets are separate, each representing one specific month of a Cyclistic's trip historical data in the last 12 months.

To make the data more accessible, I implemented naming conventions, foldering, and archiving of old files. But firstly, I met with the team to create a file that describes the project's naming convention for easy reference to avoid any confusion with the team.

Classify and filter the data

Were selected trip data from the last 12 months. In this case, the most recent data available were from March 2022 to February 2023.

Checking the data

The data collected are historical and inclusive. So, we haven't found any problems or biases.

The data I have collected is internal, is proven fit for use, the validity of the data has been certified with the source, have critical information needed to find the solution, and the data is current. Therefore, regarding credibility, the data is reliable, original, comprehensive, actual, and cited.

Examining the datasets, I confirm that the data is presented consistently with correct data types.

The datasets don't have variables like genre, year of birth, and price plans. Having these variables, we could get more complete insights, but that doesn't stop us from performing the analysis.

Data credibility

All the datasets provided by Cyclistic are public data and can be used to explore how different user types use Cyclistic's bikes. Moreover, it's secured that they're trustable, original, comprehensive, actual, and cited.

To ensure data privacy rider's personally identifiable information won't be published, preserving the passenger's id.

A description of all data sources used

The data sources contain historical trip data on Cyclistic (bike-sharing service) for the past 12 months, from March 2022 to February 2023, all located in the city of Chicago. In total, there are 12 datasets, each one representing a specific month with 13 columns. All the data has been made available by Motivate International Inc. by this license. The data are available through this link.

Setting the environment

Installing the packages

```
# installing packages tidyverse, janitor, skirm, lubridate e ggplot
install.packages("tidyverse", repos = "http://cran.us.r-project.org")
install.packages("janitor", repos = "http://cran.us.r-project.org")
install.packages("skimr", repos = "http://cran.us.r-project.org")
install.packages("lubridate", repos = "http://cran.us.r-project.org")
install.packages("ggplot", repos = "http://cran.us.r-project.org")
```

Loading the packages

```
# loading packages tidyverse, janitor, skirm, lubridate e ggplot2
library("tidyverse")
library("janitor")
library("skimr")
library("lubridate")
library("ggplot2")
```

Importing data

```
tripdata_2022_03 <- read.csv("2022-03_tripdata.csv")
tripdata_2022_04 <- read.csv("2022-04_tripdata.csv")
tripdata_2022_05 <- read.csv("2022-05_tripdata.csv")
tripdata_2022_06 <- read.csv("2022-06_tripdata.csv")
tripdata_2022_07 <- read.csv("2022-07_tripdata.csv")
tripdata_2022_08 <- read.csv("2022-08_tripdata.csv")
tripdata_2022_09 <- read.csv("2022-09_tripdata.csv")
tripdata_2022_10 <- read.csv("2022-10_tripdata.csv")
tripdata_2022_11 <- read.csv("2022-11_tripdata.csv")
tripdata_2022_12 <- read.csv("2022-12_tripdata.csv")
tripdata_2023_01 <- read.csv("2023-01_tripdata.csv")
tripdata_2023_02 <- read.csv("2023-02_tripdata.csv")
```

Data source structure

```
str(tripdata_2022_03)

## 'data.frame':    284042 obs. of  13 variables:
##  $ ride_id           : chr  "47EC0A7F82E65D52" "8494861979B0F477" "EFE527AF80B66
109" "9F446FD9DEE3F389" ...
##  $ rideable_type     : chr  "classic_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-03-21 13:45:01" "2022-03-16 09:37:16" "2022-03-23 19:5
2:02" "2022-03-01 19:12:26" ...
##  $ ended_at          : chr  "2022-03-21 13:51:18" "2022-03-16 09:43:34" "2022-03-23 19:
54:48" "2022-03-01 19:22:14" ...
##  $ start_station_name: chr  "Wabash Ave & Wacker Pl" "Michigan Ave & Oak St" "Broa
dway & Berwyn Ave" "Wabash Ave & Wacker Pl" ...
##  $ start_station_id  : chr  "TA1307000131" "13042" "13109" "TA1307000131" ...
##  $ end_station_name  : chr  "Kingsbury St & Kinzie St" "Orleans St & Chestnut St (NEX
T Apts)" "Broadway & Ridge Ave" "Franklin St & Jackson Blvd" ...
##  $ end_station_id    : chr  "KA1503000043" "620" "15578" "TA1305000025" ...
##  $ start_lat         : num  41.9 41.9 42 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 41.9 42 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.7 -87.6 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...

str(tripdata_2022_04)

## 'data.frame':    371249 obs. of  13 variables:
##  $ ride_id           : chr  "3564070EEFD12711" "0B820C7FCF22F489" "89EEEE32293F0
7FF" "84D4751AEB31888D" ...
##  $ rideable_type     : chr  "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-04-06 17:42:48" "2022-04-24 19:23:07" "2022-04-20 19:2
9:08" "2022-04-22 21:14:06" ...
##  $ ended_at          : chr  "2022-04-06 17:54:36" "2022-04-24 19:43:17" "2022-04-20 19:
35:16" "2022-04-22 21:23:29" ...
##  $ start_station_name: chr  "Paulina St & Howard St" "Wentworth Ave & Cermak Rd" "H
alsted St & Polk St" "Wentworth Ave & Cermak Rd" ...
##  $ start_station_id  : chr  "515" "13075" "TA1307000121" "13075" ...
##  $ end_station_name  : chr  "University Library (NU)" "Green St & Madison St" "Green
St & Madison St" "Delano Ct & Roosevelt Rd" ...
##  $ end_station_id    : chr  "605" "TA1307000120" "TA1307000120" "KA1706005007" ...
##  $ start_lat         : num  42 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  42.1 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr  "member" "member" "member" "casual" ...
```

```
str(tripdata_2022_05)

## 'data.frame':    634858 obs. of  13 variables:
##  $ ride_id          : chr  "EC2DE40644C6B0F4" "1C31AD03897EE385" "1542FBEC830415CF" "6FF59852924528F8" ...
##  $ rideable_type    : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at       : chr  "2022-05-23 23:06:58" "2022-05-11 08:53:28" "2022-05-26 18:36:28" "2022-05-10 07:30:07" ...
##  $ ended_at         : chr  "2022-05-23 23:40:19" "2022-05-11 09:31:22" "2022-05-26 18:58:18" "2022-05-10 07:38:49" ...
##  $ start_station_name: chr  "Wabash Ave & Grand Ave" "DuSable Lake Shore Dr & Monroe St" "Clinton St & Madison St" "Clinton St & Madison St" ...
##  $ start_station_id  : chr  "TA1307000117" "13300" "TA1305000032" "TA1305000032" ...
##  $ end_station_name  : chr  "Halsted St & Roscoe St" "Field Blvd & South Water St" "Wood St & Milwaukee Ave" "Clark St & Randolph St" ...
##  $ end_station_id    : chr  "TA1309000025" "15534" "13221" "TA1305000030" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.7 -87.6 -87.7 ...
##  $ member_casual     : chr  "member" "member" "member" "member" ...

str(tripdata_2022_06)

## 'data.frame':    769204 obs. of  13 variables:
##  $ ride_id          : chr  "600CFD130D0FD2A4" "F5E6B5C1682C6464" "B6EB6D27BAD771D2" "C9C320375DE1D5C6" ...
##  $ rideable_type    : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at       : chr  "2022-06-30 17:27:53" "2022-06-30 18:39:52" "2022-06-30 11:49:25" "2022-06-30 11:15:25" ...
##  $ ended_at         : chr  "2022-06-30 17:35:15" "2022-06-30 18:47:28" "2022-06-30 12:02:54" "2022-06-30 11:19:43" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.8 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.8 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
str(tripdata_2022_07)

## 'data.frame':    823488 obs. of  13 variables:
##  $ ride_id           : chr  "954144C2F67B1932" "292E027607D218B6" "57765852588AD6
E0" "B5B6BE44314590E6" ...
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2022-07-05 08:12:47" "2022-07-26 12:53:38" "2022-07-03 13:5
8:49" "2022-07-31 17:44:21" ...
##  $ ended_at          : chr  "2022-07-05 08:24:32" "2022-07-26 12:55:31" "2022-07-03 14:
06:32" "2022-07-31 18:42:50" ...
##  $ start_station_name: chr  "Ashland Ave & Blackhawk St" "Buckingham Fountain (Tem
p)" "Buckingham Fountain (Temp)" "Buckingham Fountain (Temp)" ...
##  $ start_station_id  : chr  "13224" "15541" "15541" "15541" ...
##  $ end_station_name  : chr  "Kingsbury St & Kinzie St" "Michigan Ave & 8th St" "Michig
an Ave & 8th St" "Woodlawn Ave & 55th St" ...
##  $ end_station_id    : chr  "KA1503000043" "623" "623" "TA1307000164" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num  41.9 41.9 41.9 41.8 41.9 ...
##  $ end_lng           : num  -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "member" "casual" "casual" "casual" ...

str(tripdata_2022_08)

## 'data.frame':    785932 obs. of  13 variables:
##  $ ride_id           : chr  "550CF7EFEAE0C618" "DAD198F405F9C5F5" "E6F2BC47B65
CB7FD" "F597830181C2E13C" ...
##  $ rideable_type     : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at        : chr  "2022-08-07 21:34:15" "2022-08-08 14:39:21" "2022-08-08 15:2
9:50" "2022-08-08 02:43:50" ...
##  $ ended_at          : chr  "2022-08-07 21:41:46" "2022-08-08 14:53:23" "2022-08-08 15:
40:34" "2022-08-08 02:58:53" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "" "" "" "" ...
##  $ end_station_id    : chr  "" "" "" "" ...
##  $ start_lat         : num  41.9 41.9 42 41.9 41.9 ...
##  $ start_lng         : num  -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 42 42 41.8 ...
##  $ end_lng           : num  -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
```

```
str(tripdata_2022_09)

## 'data.frame':    701339 obs. of  13 variables:
##  $ ride_id          : chr  "5156990AC19CA285" "E12D4A16BF51C274" "A02B53CD7DB7
2DD7" "C82E05FEE872DF11" ...
##  $ rideable_type    : chr  "electric_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at       : chr  "2022-09-01 08:36:22" "2022-09-01 17:11:29" "2022-09-01 17:1
5:50" "2022-09-01 09:00:28" ...
##  $ ended_at         : chr  "2022-09-01 08:39:05" "2022-09-01 17:14:45" "2022-09-01 17:
16:12" "2022-09-01 09:10:32" ...
##  $ start_station_name: chr  "" "" "" "" ...
##  $ start_station_id  : chr  "" "" "" "" ...
##  $ end_station_name  : chr  "California Ave & Milwaukee Ave" "" "" "" ...
##  $ end_station_id    : chr  "13084" "" "" "" ...
##  $ start_lat        : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng        : num  -87.7 -87.6 -87.6 -87.7 -87.7 ...
##  $ end_lat          : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng          : num  -87.7 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual    : chr  "casual" "casual" "casual" "casual" ...

str(tripdata_2022_10)

## 'data.frame':    558685 obs. of  13 variables:
##  $ ride_id          : chr  "A50255C1E17942AB" "DB692A70BD2DD4E3" "3C02727AAF60
F873" "47E653FDC2D99236" ...
##  $ rideable_type    : chr  "classic_bike" "electric_bike" "electric_bike" "electric_bike" ...
##  $ started_at       : chr  "2022-10-14 17:13:30" "2022-10-01 16:29:26" "2022-10-19 18:5
5:40" "2022-10-31 07:52:36" ...
##  $ ended_at         : chr  "2022-10-14 17:19:39" "2022-10-01 16:49:06" "2022-10-19 19:
03:30" "2022-10-31 07:58:49" ...
##  $ start_station_name: chr  "Noble St & Milwaukee Ave" "Damen Ave & Charleston St" "
Hoyne Ave & Balmoral Ave" "Rush St & Cedar St" ...
##  $ start_station_id  : chr  "13290" "13288" "655" "KA1504000133" ...
##  $ end_station_name  : chr  "Larrabee St & Division St" "Damen Ave & Cullerton St" "W
estern Ave & Leland Ave" "Orleans St & Chestnut St (NEXT Apts)" ...
##  $ end_station_id    : chr  "KA1504000079" "13089" "TA1307000140" "620" ...
##  $ start_lat        : num  41.9 41.9 42 41.9 41.9 ...
##  $ start_lng        : num  -87.7 -87.7 -87.7 -87.6 -87.6 ...
##  $ end_lat          : num  41.9 41.9 42 41.9 41.9 ...
##  $ end_lng          : num  -87.6 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual    : chr  "member" "casual" "member" "member" ...
```

```
str(tripdata_2022_11)

## 'data.frame':    337735 obs. of  13 variables:
##  $ ride_id          : chr  "BCC66FC6FAB27CC7" "772AB67E902C180F" "585EAD07FDEC0152" "91C4E7ED3C262FF9" ...
##  $ rideable_type    : chr  "electric_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at       : chr  "2022-11-10 06:21:55" "2022-11-04 07:31:55" "2022-11-21 17:20:29" "2022-11-25 17:29:34" ...
##  $ ended_at         : chr  "2022-11-10 06:31:27" "2022-11-04 07:46:25" "2022-11-21 17:34:36" "2022-11-25 17:45:15" ...
##  $ start_station_name: chr  "Canal St & Adams St" "Canal St & Adams St" "Indiana Ave & Roosevelt Rd" "Indiana Ave & Roosevelt Rd" ...
##  $ start_station_id  : chr  "13011" "13011" "SL-005" "SL-005" ...
##  $ end_station_name  : chr  "St. Clair St & Erie St" "St. Clair St & Erie St" "St. Clair St & Erie St" "St. Clair St & Erie St" ...
##  $ end_station_id    : chr  "13016" "13016" "13016" "13016" ...
##  $ start_lat        : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng        : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat          : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng          : num  -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual    : chr  "member" "member" "member" "member" ...

str(tripdata_2022_12)

## 'data.frame':    181806 obs. of  13 variables:
##  $ ride_id          : chr  "65DBD2F447EC51C2" "0C201AA7EA0EA1AD" "E0B148CCB358A49D" "54C5775D2B7C9188" ...
##  $ rideable_type    : chr  "electric_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at       : chr  "2022-12-05 10:47:18" "2022-12-18 06:42:33" "2022-12-13 08:47:45" "2022-12-13 18:50:47" ...
##  $ ended_at         : chr  "2022-12-05 10:56:34" "2022-12-18 07:08:44" "2022-12-13 08:59:51" "2022-12-13 19:19:48" ...
##  $ start_station_name: chr  "Clifton Ave & Armitage Ave" "Broadway & Belmont Ave" "Sangamon St & Lake St" "Shields Ave & 31st St" ...
##  $ start_station_id  : chr  "TA1307000163" "13277" "TA1306000015" "KA1503000038" ...
##  $ end_station_name  : chr  "Sedgwick St & Webster Ave" "Sedgwick St & Webster Ave" "St. Clair St & Erie St" "Damen Ave & Madison St" ...
##  $ end_station_id    : chr  "13191" "13191" "13016" "13134" ...
##  $ start_lat        : num  41.9 41.9 41.9 41.8 41.9 ...
##  $ start_lng        : num  -87.7 -87.6 -87.7 -87.6 -87.7 ...
##  $ end_lat          : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng          : num  -87.6 -87.6 -87.6 -87.7 -87.7 ...
##  $ member_casual    : chr  "member" "casual" "member" "member" ...
```

```
str(tripdata_2023_01)

## 'data.frame':    190301 obs. of  13 variables:
##  $ ride_id          : chr  "F96D5A74A3E41399" "13CB7EB698CEDB88" "BD88A2E67066
1CE5" "C90792D034FED968" ...
##  $ rideable_type    : chr  "electric_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at       : chr  "2023-01-21 20:05:42" "2023-01-10 15:37:36" "2023-01-02 07:5
1:57" "2023-01-22 10:52:58" ...
##  $ ended_at         : chr  "2023-01-21 20:16:33" "2023-01-10 15:46:05" "2023-01-02 08:
05:11" "2023-01-22 11:01:44" ...
##  $ start_station_name: chr  "Lincoln Ave & Fullerton Ave" "Kimbark Ave & 53rd St" "We
stern Ave & Lunt Ave" "Kimbark Ave & 53rd St" ...
##  $ start_station_id  : chr  "TA1309000058" "TA1309000037" "RP-005" "TA1309000037"
...
##  $ end_station_name  : chr  "Hampden Ct & Diversey Ave" "Greenwood Ave & 47th St"
"Valli Produce - Evanston Plaza" "Greenwood Ave & 47th St" ...
##  $ end_station_id    : chr  "202480.0" "TA1308000002" "599" "TA1308000002" ...
##  $ start_lat        : num  41.9 41.8 42 41.8 41.8 ...
##  $ start_lng        : num  -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ end_lat          : num  41.9 41.8 42 41.8 41.8 ...
##  $ end_lng          : num  -87.6 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual    : chr  "member" "member" "casual" "member" ...

str(tripdata_2023_02)

## 'data.frame':    190445 obs. of  13 variables:
##  $ ride_id          : chr  "CBCD0D7777F0E45F" "F3EC5FCE5FF39DE9" "E54C1F27FA9
354FF" "3D561E04F739CC45" ...
##  $ rideable_type    : chr  "classic_bike" "electric_bike" "classic_bike" "electric_bike" ...
##  $ started_at       : chr  "2023-02-14 11:59:42" "2023-02-15 13:53:48" "2023-02-19 11:1
0:57" "2023-02-26 16:12:05" ...
##  $ ended_at         : chr  "2023-02-14 12:13:38" "2023-02-15 13:59:08" "2023-02-19 11:
35:01" "2023-02-26 16:39:55" ...
##  $ start_station_name: chr  "Southport Ave & Clybourn Ave" "Clarendon Ave & Gordon
Ter" "Southport Ave & Clybourn Ave" "Southport Ave & Clybourn Ave" ...
##  $ start_station_id  : chr  "TA1309000030" "13379" "TA1309000030" "TA1309000030" ..
.
##  $ end_station_name  : chr  "Clark St & Schiller St" "Sheridan Rd & Lawrence Ave" "Ab
erdeen St & Monroe St" "Franklin St & Adams St (Temp)" ...
##  $ end_station_id    : chr  "TA1309000024" "TA1309000041" "13156" "TA1309000008" .
..
##  $ start_lat        : num  41.9 42 41.9 41.9 41.8 ...
##  $ start_lng        : num  -87.7 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat          : num  41.9 42 41.9 41.9 41.8 ...
##  $ end_lng          : num  -87.6 -87.7 -87.7 -87.6 -87.6 ...
##  $ member_casual    : chr  "casual" "casual" "member" "member" ...
```

Process

In the third data analysis phase - Process, we will process and clean the data, looking for duplicate, inaccurate, and incomplete data to dispose of any errors, inaccuracies, or possible inconsistencies. That ensures integrity in the data before analyzing it.

In this analysis, I will use the RStudio integrated development environment (IDE) (2023.03.0 Build 386) and the programming language R (Version 4.2.2), which is a tool with the ability to work in all data analysis process stages, as well as being a tool that allows work with large amounts of data.

I will also use Tableau for data visualization, creating effective graphs and dashboards.I will also use Tableau, a powerful data visualization and business intelligence tool that helps people see and understand data through beautiful and intuitive dashboards.

Checking data

Firstly, I will check the matching of the columns in each dataset, intending to join all datasets into a single set.

```
compare_df_cols(tripdata_2022_03, tripdata_2022_04, tripdata_2022_05, tripdata_2022_
06, tripdata_2022_07, tripdata_2022_08, tripdata_2022_09, tripdata_2022_10, tripdata_20
22_11, tripdata_2022_12, tripdata_2023_01, tripdata_2023_02)
```

```
##           column_name tripdata_2022_03 tripdata_2022_04 tripdata_2022_05
## 1             end_lat          numeric          numeric          numeric
## 2             end_lng          numeric          numeric          numeric
## 3      end_station_id        character        character        character
## 4    end_station_name        character        character        character
## 5           ended_at        character        character        character
## 6       member_casual        character        character        character
## 7             ride_id        character        character        character
## 8       rideable_type        character        character        character
## 9           start_lat          numeric          numeric          numeric
## 10          start_lng          numeric          numeric          numeric
## 11    start_station_id        character        character        character
## 12 start_station_name        character        character        character
## 13         started_at        character        character        character
##    tripdata_2022_06 tripdata_2022_07 tripdata_2022_08 tripdata_2022_09
## 1           numeric          numeric          numeric          numeric
## 2           numeric          numeric          numeric          numeric
## 3         character        character        character        character
```

```
## 4         character       character       character       character
## 5         character       character       character       character
## 6         character       character       character       character
## 7         character       character       character       character
## 8         character       character       character       character
## 9           numeric         numeric         numeric         numeric
## 10          numeric         numeric         numeric         numeric
## 11        character       character       character       character
## 12        character       character       character       character
## 13        character       character       character       character
##    tripdata_2022_10 tripdata_2022_11 tripdata_2022_12 tripdata_2023_01
## 1           numeric         numeric         numeric         numeric
## 2           numeric         numeric         numeric         numeric
## 3         character       character       character       character
## 4         character       character       character       character
## 5         character       character       character       character
## 6         character       character       character       character
## 7         character       character       character       character
## 8         character       character       character       character
## 9           numeric         numeric         numeric         numeric
## 10          numeric         numeric         numeric         numeric
## 11        character       character       character       character
## 12        character       character       character       character
## 13        character       character       character       character
##    tripdata_2023_02
## 1           numeric
## 2           numeric
## 3         character
## 4         character
## 5         character
## 6         character
## 7         character
## 8         character
## 9           numeric
## 10          numeric
## 11        character
## 12        character
## 13        character
```

All columns from the dataset show correspondence with their data type, proving data consistency. Now that we have the information that there is correspondence in the columns of the datasets, I will join all the datasets into a single one.

Transforming data

Merging data

```
tripdata <- bind_rows(tripdata_2022_03, tripdata_2022_04, tripdata_2022_05, tripdata_20
22_06, tripdata_2022_07, tripdata_2022_08, tripdata_2022_09, tripdata_2022_10, tripdata
_2022_11, tripdata_2022_12, tripdata_2023_01, tripdata_2023_02)
```

Filtering, sorting and adding new columns

Before I create the “ride_length” column to register the length of the trips, I will filter the dataset to confirm that values from started_at are less than values in ended_at.

```
tripdata %>%
  filter(started_at > ended_at) %>%
  summarise(rideable_type, started_at, ended_at)
```

```
##     rideable_type         started_at           ended_at
## 1    classic_bike 2022-03-05 11:00:57 2022-03-05 10:55:01
## 2   electric_bike 2022-03-05 11:38:04 2022-03-05 11:37:57
## 3   electric_bike 2022-05-30 11:06:29 2022-05-30 11:06:17
## 4   electric_bike 2022-06-07 19:15:39 2022-06-07 17:05:37
## 5   electric_bike 2022-06-07 19:14:46 2022-06-07 17:07:45
## 6   electric_bike 2022-06-23 19:22:57 2022-06-23 19:21:46
## 7   electric_bike 2022-06-07 19:14:47 2022-06-07 17:05:42
## 8   electric_bike 2022-06-07 16:18:37 2022-06-07 16:07:28
## 9   electric_bike 2022-06-07 18:47:01 2022-06-07 17:05:41
## 10  electric_bike 2022-06-07 19:11:33 2022-06-07 17:05:24
## 11  electric_bike 2022-06-07 19:06:49 2022-06-07 17:09:43
## 12  electric_bike 2022-06-07 19:13:27 2022-06-07 17:07:57
## 13  electric_bike 2022-06-07 19:23:03 2022-06-07 17:05:38
## 14  electric_bike 2022-06-07 17:05:24 2022-06-07 16:59:53
## 15  electric_bike 2022-06-13 18:31:45 2022-06-13 18:31:27
## 16   classic_bike 2022-07-26 20:07:33 2022-07-26 19:59:34
## 17   classic_bike 2022-07-26 20:08:04 2022-07-26 19:59:34
## 18   classic_bike 2022-07-26 20:20:31 2022-07-26 19:59:34
## 19   classic_bike 2022-07-26 18:35:57 2022-07-26 18:32:30
## 20  electric_bike 2022-07-01 14:35:12 2022-07-01 14:31:50
## 21  electric_bike 2022-07-30 09:36:02 2022-07-30 09:35:53
## 22  electric_bike 2022-07-09 20:31:40 2022-07-09 20:30:17
## 23  electric_bike 2022-07-01 05:35:51 2022-07-01 05:34:15
## 24   classic_bike 2022-07-26 18:32:37 2022-07-26 18:32:30
## 25   classic_bike 2022-07-26 18:35:03 2022-07-26 18:30:15
## 26   classic_bike 2022-07-26 18:40:12 2022-07-26 18:32:30
## 27   classic_bike 2022-07-26 19:59:31 2022-07-26 19:57:59
## 28   classic_bike 2022-07-26 17:38:34 2022-07-26 17:37:38
## 29  electric_bike 2022-07-26 20:22:52 2022-07-26 20:21:19
## 30  electric_bike 2022-07-26 20:24:05 2022-07-26 20:22:49
```

```
## 31   classic_bike 2022-07-26 16:53:01 2022-07-26 16:43:30
## 32  electric_bike 2022-08-01 13:21:10 2022-08-01 13:21:05
## 33  electric_bike 2022-08-22 13:05:10 2022-08-22 13:04:48
## 34  electric_bike 2022-08-11 19:22:32 2022-08-11 19:21:35
## 35   classic_bike 2022-08-05 16:35:21 2022-08-05 16:35:20
## 36  electric_bike 2022-08-25 00:40:51 2022-08-25 00:39:34
## 37  electric_bike 2022-08-27 12:44:27 2022-08-27 12:43:39
## 38  electric_bike 2022-08-09 20:51:36 2022-08-09 20:51:31
## 39  electric_bike 2022-08-25 00:39:58 2022-08-25 00:37:39
## 40  electric_bike 2022-08-27 13:16:02 2022-08-27 13:13:35
## 41  electric_bike 2022-08-27 13:18:18 2022-08-27 13:13:35
## 42  electric_bike 2022-08-27 13:18:54 2022-08-27 13:15:58
## 43  electric_bike 2022-08-27 13:17:51 2022-08-27 13:15:58
## 44  electric_bike 2022-08-27 13:22:25 2022-08-27 13:15:58
## 45  electric_bike 2022-08-27 13:16:39 2022-08-27 13:15:58
## 46  electric_bike 2022-08-25 00:39:40 2022-08-25 00:39:34
## 47  electric_bike 2022-09-30 17:27:30 2022-09-30 17:27:26
## 48   classic_bike 2022-09-14 17:47:44 2022-09-14 17:47:34
## 49  electric_bike 2022-09-08 18:00:55 2022-09-08 18:00:32
## 50  electric_bike 2022-09-05 17:56:42 2022-09-05 17:56:41
## 51  electric_bike 2022-09-28 11:04:32 2022-09-21 06:31:11
## 52  electric_bike 2022-09-08 15:55:54 2022-09-08 15:53:24
## 53  electric_bike 2022-09-10 18:52:40 2022-09-10 18:52:29
## 54  electric_bike 2022-09-18 19:06:32 2022-09-18 19:06:24
## 55  electric_bike 2022-09-08 16:04:03 2022-09-08 16:01:09
## 56  electric_bike 2022-10-03 08:55:01 2022-10-03 08:54:45
## 57  electric_bike 2022-10-21 19:29:00 2022-10-21 19:28:59
## 58   classic_bike 2022-10-13 14:42:10 2022-10-13 11:53:28
## 59  electric_bike 2022-10-24 17:03:29 2022-10-24 17:03:28
## 60  electric_bike 2022-11-06 01:53:12 2022-11-06 01:30:03
## 61  electric_bike 2022-11-06 01:54:17 2022-11-06 01:29:40
## 62  electric_bike 2022-11-06 01:59:33 2022-11-06 01:04:55
## 63  electric_bike 2022-11-06 01:39:08 2022-11-06 01:01:34
## 64   classic_bike 2022-11-06 01:50:49 2022-11-06 01:02:20
## 65  electric_bike 2022-11-06 01:32:51 2022-11-06 01:00:24
## 66   classic_bike 2022-11-06 01:51:43 2022-11-06 01:01:23
## 67  electric_bike 2022-11-18 16:20:53 2022-11-18 16:20:28
## 68  electric_bike 2022-11-06 01:59:10 2022-11-06 01:10:49
## 69   classic_bike 2022-11-06 01:52:29 2022-11-06 01:01:11
## 70  electric_bike 2022-11-06 01:49:12 2022-11-06 01:01:40
## 71   classic_bike 2022-11-06 01:59:53 2022-11-06 01:25:35
## 72  electric_bike 2022-11-06 01:55:06 2022-11-06 01:00:04
## 73  electric_bike 2022-11-06 01:54:30 2022-11-06 01:37:13
## 74  electric_bike 2022-11-06 01:37:09 2022-11-06 01:37:06
## 75  electric_bike 2022-11-06 01:37:55 2022-11-06 01:37:10
## 76  electric_bike 2022-11-06 01:56:47 2022-11-06 01:37:15
## 77  electric_bike 2022-11-06 01:52:40 2022-11-06 01:37:18
## 78  electric_bike 2022-11-06 01:58:11 2022-11-06 01:00:12
## 79  electric_bike 2022-11-06 01:56:40 2022-11-06 01:16:42
## 80   classic_bike 2022-11-06 01:50:30 2022-11-06 01:01:06
## 81   classic_bike 2022-11-06 01:47:22 2022-11-06 01:13:53
```

```
## 82   classic_bike 2022-11-06 01:57:01 2022-11-06 01:21:33
## 83   classic_bike 2022-11-06 01:58:35 2022-11-06 01:05:51
## 84  electric_bike 2022-11-06 01:59:42 2022-11-06 01:10:27
## 85   classic_bike 2022-11-06 01:57:57 2022-11-06 01:06:27
## 86   classic_bike 2022-11-14 00:21:59 2022-11-14 00:17:36
## 87   classic_bike 2022-11-06 01:43:29 2022-11-06 01:12:25
## 88  electric_bike 2022-11-06 01:57:21 2022-11-06 01:02:07
## 89  electric_bike 2022-11-06 01:56:17 2022-11-06 01:12:19
## 90   classic_bike 2022-11-06 01:46:10 2022-11-06 01:06:44
## 91   classic_bike 2022-11-06 01:46:17 2022-11-06 01:05:13
## 92  electric_bike 2022-11-06 01:59:05 2022-11-06 01:02:03
## 93  electric_bike 2022-11-06 01:51:53 2022-11-06 01:18:03
## 94  electric_bike 2022-11-06 01:58:34 2022-11-06 01:03:24
## 95  electric_bike 2022-11-06 01:29:27 2022-11-06 01:12:58
## 96  electric_bike 2022-11-06 01:15:50 2022-11-06 01:01:57
## 97  electric_bike 2022-11-06 01:59:42 2022-11-06 01:04:20
## 98   classic_bike 2022-11-06 01:58:46 2022-11-06 01:11:33
## 99  electric_bike 2022-11-06 01:52:09 2022-11-06 01:04:23
## 100 electric_bike 2022-11-06 01:51:26 2022-11-06 01:13:46
## 101 electric_bike 2023-02-04 13:08:08 2023-02-04 13:04:52
```

Filtering the dataset, we found 101 records where the values from started_at are higher than values from ended_at. However, we can conclude that this is not the right way to represent the time of the trips.

With that in mind, I will create a new dataset only to get these records to analyze them.

```
incorrect_time <- tripdata %>%
  filter(started_at > ended_at)
```

```
incorrect_time %>%
  filter(started_at > ended_at) %>%
  group_by(rideable_type) %>%
  summarise(number_of_rides = n(), percent_of_rides = round(length(rideable_type) / nrow
(incorrect_time) * 100, digits = 2))
```

```
## # A tibble: 2 × 3
##   rideable_type number_of_rides percent_of_rides
##   <chr>                  <int>           <dbl>
## 1 classic_bike              28            27.7
## 2 electric_bike             73            72.3
```

Of 101 records, 73, or 72.28%, were from electric bikes, and 28, or 27.72%, were from classic bikes.

```
incorrect_time %>%
  filter(started_at > ended_at) %>%
  group_by(start_station_name, rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  arrange(desc(number_of_rides)) %>%
  print()
```

```
## # A tibble: 51 × 3
## # Groups:   start_station_name [47]
##    start_station_name           rideable_type number_of_rides
##    <chr>                        <chr>                 <int>
##  1 ""                           electric_bike            24
##  2 "Lincoln Ave & Roscoe St*"   electric_bike            12
##  3 "Lincoln Ave & Roscoe St*"   classic_bike             11
##  4 "McClurg Ct & Ohio St"       electric_bike             5
##  5 "Western Ave & Winnebago Ave" electric_bike            3
##  6 "Ashland Ave & Division St"  classic_bike              1
##  7 "Base - 2132 W Hubbard"      electric_bike             1
##  8 "Broadway & Barry Ave"       classic_bike              1
##  9 "Broadway & Waveland Ave"    electric_bike             1
## 10 "Chicago Ave & Sheridan Rd"  electric_bike             1
## # … with 41 more rows
```

Here, we observe that from 73 electric bikes in that dataset, 24 were without stations' names, and 11 occurred in the same station, "Lincoln Ave & Roscoe St"; While for 28 classic bikes, we also have 12 occurrences in "Lincoln Ave & Roscoe St"; But that whole record for classic bikes we have station's name.

Note: This analysis is not our business task, remembering that our business task is to identify how casual riders and annual members use Cyclistic's bikes differently and not identify how occurred the incorrect input records in trips' time.

To avoid distractions from our business task, I will consider that the error occurred due to software or research imputation errors.

We can consider this as a future analysis because if we continue to get these errors, it can impact the accuracy of travel behavior models and lead to biases in the measures derived from the models.

Considering that it was an error caused by software or research imputation, to solve this problem, I will swap the values between started_at and ended_at so that I correctly have the trip times.

```
for (i in 1:nrow(tripdata)) {
  if(tripdata[i,3] > tripdata[i,4]) {
  x <- tripdata[i,3]
  tripdata[i,3] <- tripdata[i,4]
  tripdata[i,4] <- x
  }
}
```

After performing the data transformation, I will filter the dataset again to verify that I have the correct time values.

```
tripdata %>%
  filter(started_at > ended_at)

##  [1] ride_id           rideable_type     started_at        ended_at
##  [5] start_station_name start_station_id  end_station_name  end_station_id
##  [9] start_lat         start_lng         end_lat           end_lng
## [13] member_casual
## <0 rows> (or 0-length row.names)
```

Now, there are no more incorrect values in started_at and ended_at. After verifying the dataset, I will create the ride_length column.

Now, I will order the data through the variable (started_at), then I will create a column called (date) to record the trip date, and I will add other ones called (month, day, year, day_of_week, ride_length, and start_hour).

Note: The trip duration will be present in seconds.

```
tripdata <- tripdata %>%
  arrange(started_at)

tripdata <- tripdata %>%
  mutate(date = format(as.Date(started_at)))

tripdata <- tripdata %>%
  mutate(month = format(as.Date(date), "%B")) %>%
  mutate(day = format(as.Date(date), "%d")) %>%
  mutate(year = format(as.Date(date), "%Y")) %>%
  mutate(day_of_week = format(as.Date(date), "%A")) %>%
  mutate(ride_length = difftime(ended_at, started_at)) %>%
  mutate(start_hour = strftime(started_at, "%H"))
```

The values in month and day_of_week are in Portuguese. So I will translate them into English.

Converting ride_length column data type

```
tripdata$ride_length <- as.numeric(as.character(tripdata$ride_length))
```

Checking dataset structure

```
str(tripdata)
```

```
## 'data.frame':    5829084 obs. of  20 variables:
##  $ ride_id           : chr  "41557457145715FC" "2CF34B94DEDAF6D1" "ED3DD2C7341F
AF6E" "A3B10F6CF7EF2F01" ...
##  $ rideable_type     : chr  "classic_bike" "electric_bike" "classic_bike" "electric_bike" ...
##  $ started_at        : chr  "2022-03-01 00:00:19" "2022-03-01 00:02:11" "2022-03-01 00:0
3:24" "2022-03-01 00:03:53" ...
##  $ ended_at          : chr  "2022-03-01 00:04:30" "2022-03-01 00:08:49" "2022-03-01 00:
14:35" "2022-03-01 00:05:41" ...
##  $ start_station_name: chr  "Wentworth Ave & Cermak Rd" "State St & Pearson St" "Le
avitt St & Addison St" "" ...
##  $ start_station_id  : chr  "13075" "TA1307000061" "KA1504000143" "" ...
##  $ end_station_name  : chr  "Normal Ave & Archer Ave" "Ogden Ave & Chicago Ave" "S
outhport Ave & Wellington Ave" "" ...
##  $ end_station_id    : chr  "TA1308000014" "TA1305000020" "TA1307000006" "" ...
##  $ start_lat         : num  41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num  -87.6 -87.6 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num  41.8 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num  -87.6 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr  "member" "member" "casual" "member" ...
##  $ date              : chr  "2022-03-01" "2022-03-01" "2022-03-01" "2022-03-01" ...
##  $ month             : chr  "March" "March" "March" "March" ...
##  $ day               : chr  "01" "01" "01" "01" ...
##  $ year              : chr  "2022" "2022" "2022" "2022" ...
##  $ day_of_week       : chr  "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ ride_length       : num  251 398 671 108 790 ...
##  $ start_hour        : chr  "00" "00" "00" "00" ...
```

Checking dataset statistic summary

```
skim_without_charts(tripdata)
```

Data summary

Name	tripdata
Number of rows	5829084
Number of columns	20
_____	
Column type frequency:	
character	15
numeric	5
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	5829084	0
rideable_type	0	1	11	13	0	3	0
started_at	0	1	19	19	0	4891311	0
ended_at	0	1	19	19	0	4904903	0
start_station_name	0	1	0	64	850418	1693	0
start_station_id	0	1	0	37	850550	1315	0
end_station_name	0	1	0	64	909038	1716	0
end_station_id	0	1	0	37	909179	1319	0
member_casual	0	1	6	6	0	2	0
date	0	1	10	10	0	365	0
month	0	1	3	9	0	12	0
day	0	1	2	2	0	31	0
year	0	1	4	4	0	2	0
day_of_week	0	1	6	9	0	7	0
start_hour	0	1	2	2	0	24	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	42.07
start_lng	0	1	-87.65	0.03	-87.84	-87.66	-87.64	-87.63	-87.52
end_lat	5938	1	41.90	0.07	0.00	41.88	41.90	41.93	42.37
end_lng	5938	1	-87.65	0.11	-88.14	-87.66	-87.64	-87.63	0.00
ride_length	0	1	1153.32	1050.48	0.00	344.00	609.00	1094.00	2483235.00

The summary informed us that the dataset contains empty data, missing data and that there is no duplicated data!

The summary also shows that the ride_length column contains a minimum of 0, which does not make sense for a ride duration of 0 seconds. So, in this analysis, I will consider rows just with trip duration starting from 60 seconds.

Making the dataset backup

Before we perform any data cleaning process, I will create a dataset backup to ensure we have an original one.

```
tripdata_v2 <- tripdata
```

Removing values below 60 seconds from ride_length column

```
tripdata_v2 <- tripdata_v2[tripdata_v2$ride_length >= 60,]
```

Using the summary again, we can see that the minimum value from the “ride_length” column is now 60.

```
summary(tripdata_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      60     359     623    1179    1111 2483235
```

Remove irrelevant data

For this analysis, I considered the following variables irrelevant. So, I will remove them.

```
tripdata_v2 <- tripdata_v2 %>%
  select(-c(start_lat, start_lng, end_lat, end_lng))
```

Converting empty data to missing data (NAs)

The statistic summary informed us that the dataset contains empty data. Therefore, I will convert them into NAs to ease the data-cleaning process.

```
tripdata_v2[tripdata_v2 == ''] <- NA
```

Excluding missing data (NAs)

After the data transformation, I will remove all missing data.

```
tripdata_v2 <- na.omit(tripdata_v2)
```

Checking the dataset once again

```
str(tripdata_v2)

## 'data.frame':    4416370 obs. of  16 variables:
##  $ ride_id          : chr  "41557457145715FC" "2CF34B94DEDAF6D1" "ED3DD2C7341F
AF6E" "94FDE8513B6ECF91" ...
##  $ rideable_type    : chr  "classic_bike" "electric_bike" "classic_bike" "classic_bike" ...
##  $ started_at       : chr  "2022-03-01 00:00:19" "2022-03-01 00:02:11" "2022-03-01 00:0
3:24" "2022-03-01 00:04:12" ...
##  $ ended_at         : chr  "2022-03-01 00:04:30" "2022-03-01 00:08:49" "2022-03-01 00:
14:35" "2022-03-01 00:17:22" ...
##  $ start_station_name: chr  "Wentworth Ave & Cermak Rd" "State St & Pearson St" "Le
avitt St & Addison St" "Wells St & Hubbard St" ...
##  $ start_station_id  : chr  "13075" "TA1307000061" "KA1504000143" "TA1307000151" .
..
##  $ end_station_name  : chr  "Normal Ave & Archer Ave" "Ogden Ave & Chicago Ave" "S
outhport Ave & Wellington Ave" "Clark St & Lincoln Ave" ...
##  $ end_station_id    : chr  "TA1308000014" "TA1305000020" "TA1307000006" "13179" .
..
##  $ member_casual     : chr  "member" "member" "casual" "member" ...
##  $ date              : chr  "2022-03-01" "2022-03-01" "2022-03-01" "2022-03-01" ...
##  $ month             : chr  "March" "March" "March" "March" ...
##  $ day               : chr  "01" "01" "01" "01" ...
##  $ year              : chr  "2022" "2022" "2022" "2022" ...
##  $ day_of_week       : chr  "Tuesday" "Tuesday" "Tuesday" "Tuesday" ...
##  $ ride_length       : num  251 398 671 790 1087 ...
```

```
##  $ start_hour        : chr  "00" "00" "00" "00" ...
##  - attr(*, "na.action")= 'omit' Named int [1:1283920] 4 6 9 10 14 15 17 24 25 33 ...
##   ..- attr(*, "names")= chr [1:1283920] "4" "6" "9" "10" ...

skim_without_charts(tripdata_v2)
```

Data summary

Name	tripdata_v2
Number of rows	4416370
Number of columns	16
_____	
Column type frequency:	
character	15
numeric	1
_____	
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	4416370	0
rideable_type	0	1	11	13	0	3	0
started_at	0	1	19	19	0	3837712	0
ended_at	0	1	19	19	0	3851015	0
start_station_name	0	1	7	64	0	1562	0
start_station_id	0	1	3	37	0	1268	0
end_station_name	0	1	10	64	0	1605	0
end_station_id	0	1	3	37	0	1280	0
member_casual	0	1	6	6	0	2	0
date	0	1	10	10	0	365	0
month	0	1	3	9	0	12	0
day	0	1	2	2	0	31	0
year	0	1	4	4	0	2	0
day_of_week	0	1	6	9	0	7	0
start_hour	0	1	2	2	0	24	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
ride_length	0	1	1030.98	2522.86	60	370	640	1142	2061244

Once the data are consistent and complete, it is ready to be analyzed.

Documentation of any cleaning or manipulation of data

In this link we can find the Changelog documentation.

Analyze

In the fourth data analysis process - Analyze, the data are already rightly stored and prepared to be analyzed. In this phase, I will analyze the data to find insights and possible solutions to the business task.

Identify trends and relationship

Descriptive analysis in ride_length column

```
summary(tripdata_v2$ride_length)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      60     370     640    1031    1142 2061244
```

Comparing casual users and annual members

```
tripdata_v2 %>%
  aggregate(ride_length ~ member_casual, FUN = mean)

##   member_casual ride_length
## 1        casual   1443.8168
## 2        member    756.1626

tripdata_v2 %>%
  aggregate(ride_length ~ member_casual, FUN = median)

##   member_casual ride_length
## 1        casual         836
## 2        member         544

tripdata_v2 %>%
  aggregate(ride_length ~ member_casual, FUN = max)

##   member_casual ride_length
## 1        casual     2061244
## 2        member       89872

tripdata_v2 %>%
  aggregate(ride_length ~ member_casual, FUN = min)

##   member_casual ride_length
## 1        casual         60
## 2        member         60
```

Analyzing the number of rides by user type

```
tripdata_v2 %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n(), percent_of_rides = round(length(ride_id) / nrow(tripda
ta_v2), digits = 4) * 100)

## # A tibble: 2 × 3
##   member_casual number_of_rides percent_of_rides
##   <chr>                  <int>           <dbl>
## 1 casual                1764964            40.0
## 2 member                2651406            60.0
```

- From 4.416.370 observations, we have 2.651.406 annual members, which represents 60.04%, and 1.764.964 casual users, which represents 39.96%, meaning that we have more annual members using Cyclistic shared bikes.

Analyzing the number of rides by month

```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(), percent_of_rides = round(length(ride_id) / nrow(tripdata_v2), digits = 4) * 100) %>%
  print(n = 24)
```

```
## # A tibble: 24 × 4
## # Groups:   member_casual [2]
##    member_casual month     number_of_rides percent_of_rides
##    <chr>         <chr>             <int>           <dbl>
##  1 casual        April             90816            2.06
##  2 casual        August           265751            6.02
##  3 casual        December          30979             0.7
##  4 casual        February          32142            0.73
##  5 casual        January           29021            0.66
##  6 casual        July             306618            6.94
##  7 casual        June             287554            6.51
##  8 casual        March             66410             1.5
##  9 casual        May              216938            4.91
## 10 casual        November          72384            1.64
## 11 casual        October          148865            3.37
## 12 casual        September        217486            4.92
## 13 member        April            177723            4.02
## 14 member        August           328581            7.44
## 15 member        December         101634             2.3
## 16 member        February         113712            2.57
## 17 member        January          115445            2.61
## 18 member        July             324312            7.34
## 19 member        June             322257             7.3
## 20 member        March            146497            3.32
## 21 member        May              277162            6.28
## 22 member        November         178778            4.05
## 23 member        October          257461            5.83
## 24 member        September        307844            6.97
```

As we see above, the months are out of order. So, I will sort them.

Sorting the column month

```
tripdata_v2$month <- ordered(tripdata_v2$month, levels=c("March", "April", "May", "June", "July", "August", "September", "October", "November", "December", "January", "February"))
```

I will also sort the column day_of_week.

Sorting the column day_of_week

```
tripdata_v2$day_of_week <- ordered(tripdata_v2$day_of_week, levels=c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))
```

Now with the columns sorted, we can analyze the summary.

Analyzing the number of rides by month

```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(), percent_of_rides = round(length(ride_id) / nrow(tripdata_v2), digits = 4) * 100) %>%
  print(n = 24)
```

```
## # A tibble: 24 × 4
## # Groups:   member_casual [2]
##    member_casual month     number_of_rides percent_of_rides
##    <chr>         <ord>             <int>           <dbl>
##  1 casual        March             66410             1.5
##  2 casual        April             90816            2.06
##  3 casual        May              216938             4.91
##  4 casual        June             287554             6.51
##  5 casual        July             306618             6.94
##  6 casual        August           265751             6.02
##  7 casual        September        217486             4.92
##  8 casual        October          148865             3.37
##  9 casual        November          72384             1.64
## 10 casual        December          30979              0.7
## 11 casual        January           29021             0.66
## 12 casual        February          32142             0.73
## 13 member        March            146497             3.32
## 14 member        April            177723             4.02
## 15 member        May              277162             6.28
## 16 member        June             322257              7.3
## 17 member        July             324312             7.34
## 18 member        August           328581             7.44
## 19 member        September        307844             6.97
## 20 member        October          257461             5.83
```

```
## 21 member      November        178778          4.05
## 22 member      December        101634          2.3
## 23 member      January         115445          2.61
## 24 member      February        113712          2.57
```

- Based on months, annual members were the most frequent users of Cyclistic bike-sharing service in the last 12 months observed.
- June, July, and August, representing the summer period, recorded the highest percentage of trips by annual members, with 7.30%, 7.34%, and 7.44%, respectively.
- The lowest percentages of trips were recorded by casual users in December, January, and February, representing the winter period, with 0.70%, 0.66%, and 0.73%.

Analyzing the number of rides by weekday

```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), percent_of_rides = round(length(ride_id) / nrow(tripda
ta_v2), digits = 4) * 100)
```

```
## # A tibble: 14 × 4
## # Groups:   member_casual [2]
##    member_casual day_of_week number_of_rides percent_of_rides
##    <chr>         <ord>               <int>           <dbl>
##  1 casual        Sunday             304391            6.89
##  2 casual        Monday             211934            4.8
##  3 casual        Tuesday            199973            4.53
##  4 casual        Wednesday          204877            4.64
##  5 casual        Thursday           229837            5.2
##  6 casual        Friday             248018            5.62
##  7 casual        Saturday           365934            8.29
##  8 member        Sunday             303614            6.87
##  9 member        Monday             379785            8.6
## 10 member        Tuesday            425836            9.64
## 11 member        Wednesday          419528            9.5
## 12 member        Thursday           419202            9.49
## 13 member        Friday             363621            8.23
## 14 member        Saturday           339820            7.69
```

- Annual members used shared bikes more frequently on weekdays, while casual users used them most at weekends.
- Tuesday, Wednesday, and Thursday recorded the highest percentages of weekday trips with 9.64%, 9.50%, and 9.49%, respectively, all with annual members exceeding 400,000 rides.

Analyzing the number of rides by hour

```
tripdata_v2 %>%
  group_by(start_hour) %>%
  summarise(number_of_rides = n(),
            member = sum(member_casual == "member"),
            casual = sum(member_casual == "casual"),
            difference = abs(sum(member_casual == "member") - sum(member_casual == "ca
sual"))) %>%
  print(n = 24)

## # A tibble: 24 × 5
##    start_hour number_of_rides member casual difference
##    <chr>              <int>  <int>  <int>      <int>
##  1 00                 58541  25400  33141       7741
##  2 01                 36825  15629  21196       5567
##  3 02                 21273   8616  12657       4041
##  4 03                 12373   5258   7115       1857
##  5 04                 10845   6154   4691       1463
##  6 05                 34608  26043   8565      17478
##  7 06                100020  77697  22323      55374
##  8 07                185261 146461  38800     107661
##  9 08                225240 171956  53284     118672
## 10 09                 170102 114886  55216      59670
## 11 10                 178404 105918  72486      33432
## 12 11                 221431 126476  94955      31521
## 13 12                 258075 145991 112084      33907
## 14 13                 260343 143760 116583      27177
## 15 14                 266393 143003 123390      19613
## 16 15                 311488 175127 136361      38766
## 17 16                 390239 236897 153342      83555
## 18 17                 456795 286353 170442     115911
## 19 18                 376484 226309 150175      76134
## 20 19                 274408 160534 113874      46660
## 21 20                 194524 111552  82972      28580
## 22 21                 157518  86408  71110      15298
## 23 22                 127760  64132  63628        504
## 24 23                  87420  40846  46574       5728
```

- The lowest differences in shared bike use between casual users and annual members were from 10:00 p.m. until 04:00 a.m.
- Annual members have the highest number of trips from 04:00 a.m. to 10:00 p.m., compared to casual users from 11:00 p.m. to 03:00 a.m.
- The highest number of trips was reached at 05:00 p.m. by annual members, which came close to 300.000 trips, and by casual users, which exceeded 150.000.

Analyzing the average ride length by month

```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(average_ride_length = round(mean(ride_length))) %>%
  print(n = 24)
```

```
## # A tibble: 24 × 3
## # Groups:   member_casual [2]
##    member_casual month     average_ride_length
##    <chr>         <ord>                   <dbl>
##  1 casual        March                    1725
##  2 casual        April                    1575
##  3 casual        May                      1687
##  4 casual        June                     1524
##  5 casual        July                     1530
##  6 casual        August                   1419
##  7 casual        September                1328
##  8 casual        October                  1248
##  9 casual        November                 1051
## 10 casual        December                  905
## 11 casual        January                   911
## 12 casual        February                 1081
## 13 member        March                     719
## 14 member        April                     708
## 15 member        May                       812
## 16 member        June                      836
## 17 member        July                      826
## 18 member        August                    802
## 19 member        September                 772
## 20 member        October                   715
## 21 member        November                  662
## 22 member        December                  625
## 23 member        January                   616
## 24 member        February                  642
```

- Looking at the last 12 months of data, we can see that the average ride length was higher for casual users than for annual members in all months.
- In the period observed, annual members did not reach 1.000 seconds, or 16 minutes and 40 seconds of travel time.
- The highest records were in the spring months with casual users, over 1.500 seconds or 25 minutes.

Analyzing the average ride length by weekday

```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_ride_length = round(mean(ride_length)))
```

```
## # A tibble: 14 × 3
## # Groups:   member_casual [2]
##    member_casual day_of_week average_ride_length
##    <chr>         <ord>                     <dbl>
##  1 casual        Sunday                     1643
##  2 casual        Monday                     1490
##  3 casual        Tuesday                    1283
##  4 casual        Wednesday                  1240
##  5 casual        Thursday                   1282
##  6 casual        Friday                     1350
##  7 casual        Saturday                   1619
##  8 member        Sunday                      844
##  9 member        Monday                      729
## 10 member        Tuesday                     713
## 11 member        Wednesday                   719
## 12 member        Thursday                    730
## 13 member        Friday                      742
## 14 member        Saturday                    853
```

- Casual users tend to use shared bikes with an average ride length higher than annual members on all days of the week.
- On all days of the week, the average ride length for annual members was less than 1.000 seconds, or 16 minutes and 40 seconds.
- The highest averages were achieved on Sundays and Saturdays with casual users, exceeding 1.500 seconds or 25 minutes, respectively.

Analyzing the number of rides by bicycle type

```
tripdata_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n(), percent_of_ride = round(length(ride_id) / nrow(tripdat
a_v2) * 100, digits = 2))
```

```
## # A tibble: 5 × 4
## # Groups:   member_casual [2]
##   member_casual rideable_type number_of_rides percent_of_ride
##   <chr>         <chr>                 <int>          <dbl>
## 1 casual        classic_bike         890179           20.2
## 2 casual        docked_bike          174867            3.96
## 3 casual        electric_bike        699918           15.8
## 4 member        classic_bike        1733831           39.3
## 5 member        electric_bike        917575           20.8
```

- From the total of 4.416.370 observations, the annual members didn't use the docked bike, which presented only 3.96% used by casual users.
- Both annual members and casual users prefer classic bicycles: annual members have made more than 1.500.000 trips, representing 39.26% of the total number of trips made, compared with 20.16% for casual users, who made almost 900.000 journeys.
- With electric bikes, annual members correspond to 20.78%, compared to 15.85% from casual users.

Analyzing the average ride length by bicycle type

```
tripdata_v2 %>%
   group_by(member_casual, rideable_type) %>%
   summarise(ride_length = round(mean(ride_length), 2))
```

```
## # A tibble: 5 × 3
## # Groups:   member_casual [2]
##   member_casual rideable_type ride_length
##   <chr>         <chr>               <dbl>
## 1 casual        classic_bike        1479.
## 2 casual        docked_bike         3006.
## 3 casual        electric_bike       1008.
## 4 member        classic_bike         802.
## 5 member        electric_bike        670.
```

- Casual users used docked bikes with longer average trips than other types of bikes, such as classic and electric bikes.

Analyzing the 10 most used stations by casual users

```
tripdata_v2 %>%
  group_by(member_casual, start_station_name) %>%
  filter(member_casual == "casual") %>%
  summarise(number_of_rides = n()) %>%
  arrange(desc(number_of_rides)) %>%
  head(10)
```

```
## # A tibble: 10 × 3
## # Groups:   member_casual [1]
##    member_casual start_station_name              number_of_rides
##    <chr>         <chr>                                   <int>
##  1 casual        Streeter Dr & Grand Ave                 54628
##  2 casual        DuSable Lake Shore Dr & Monroe St       30087
##  3 casual        Millennium Park                         23871
##  4 casual        Michigan Ave & Oak St                   23618
##  5 casual        DuSable Lake Shore Dr & North Blvd      21934
##  6 casual        Shedd Aquarium                          19515
##  7 casual        Theater on the Lake                     17291
##  8 casual        Wells St & Concord Ln                   14870
##  9 casual        Dusable Harbor                          13167
## 10 casual        Indiana Ave & Roosevelt Rd              12731
```

- All 10 of the most used stations by casual users are close to tourist attractions in Chicago.

Analyzing the 10 most used stations by annual members

```
tripdata_v2 %>%
  group_by(member_casual, start_station_name) %>%
  filter(member_casual == "member") %>%
  summarise(number_of_rides = n()) %>%
  arrange(desc(number_of_rides)) %>%
  head(10)
```

```
## # A tibble: 10 × 3
## # Groups:   member_casual [1]
##    member_casual start_station_name          number_of_rides
##    <chr>        <chr>                                <int>
##  1 member       Kingsbury St & Kinzie St             23433
##  2 member       Clark St & Elm St                    20914
##  3 member       Wells St & Concord Ln                19899
##  4 member       Clinton St & Washington Blvd         19555
##  5 member       University Ave & 57th St             18743
##  6 member       Loomis St & Lexington St             18693
##  7 member       Ellis Ave & 60th St                  18616
##  8 member       Clinton St & Madison St              18071
##  9 member       Wells St & Elm St                    17879
## 10 member       Broadway & Barry Ave                 16332
```

- The top 10 stations used by annual members are all near commercial, residential and educational areas in Chicago.

Summary of the Analysis

- From 4.416.370 observations, we have 2.651.406 annual members, which represents 60.04%, and 1.764.964 casual users, which represents 39.96%, meaning that we have more annual members using Cyclistic's shared bikes.
- Annual members have used shared bikes with higher frequency during the days of the week, while casual users used them most during the weekend.
- Casual users usually use shared bikes with an average trip ride length higher than annual members on all days of the week.
- Annual members have the highest trip number from 04:00 a.m. until 10:00 p.m., compared to casual users, from 11:00 p.m. until 03:00 a.m.
- Both Annual members and casual users have a preference for classic bikes. Annual members have realized more than 1.500.000 trips, representing 39.26% of total trips made, compared to 20.16% from casual users who made almost 900.000 journeys.
- All 10 of the most used stations by casual users are close to tourist attractions in Chicago.
- The top 10 stations used by annual members are all near commercial, residential and educational areas in Chicago.

Share

In the fifth data analysis process step - Share, after I have gained the insights in the previous step, I will create compelling data visualizations to share my findings. So I will summarize the results using clear and attractive visuals to help stakeholders understand the problem's solution.

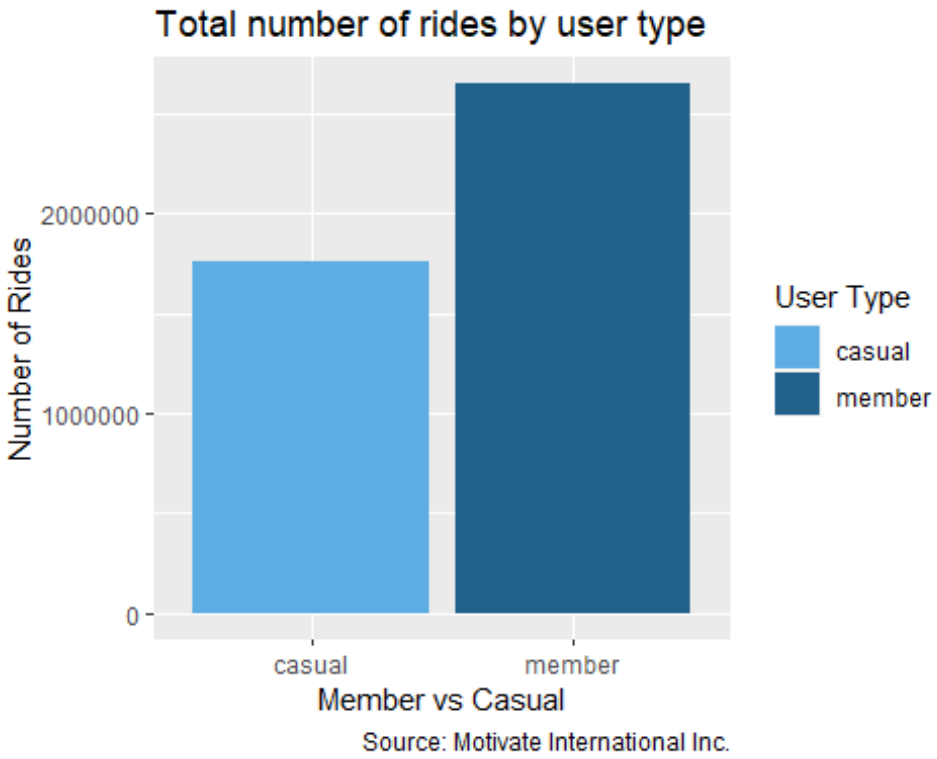
Removing scientific notations

```
options(scipen = 999)
```

Data visualizations

Number of rides by user type

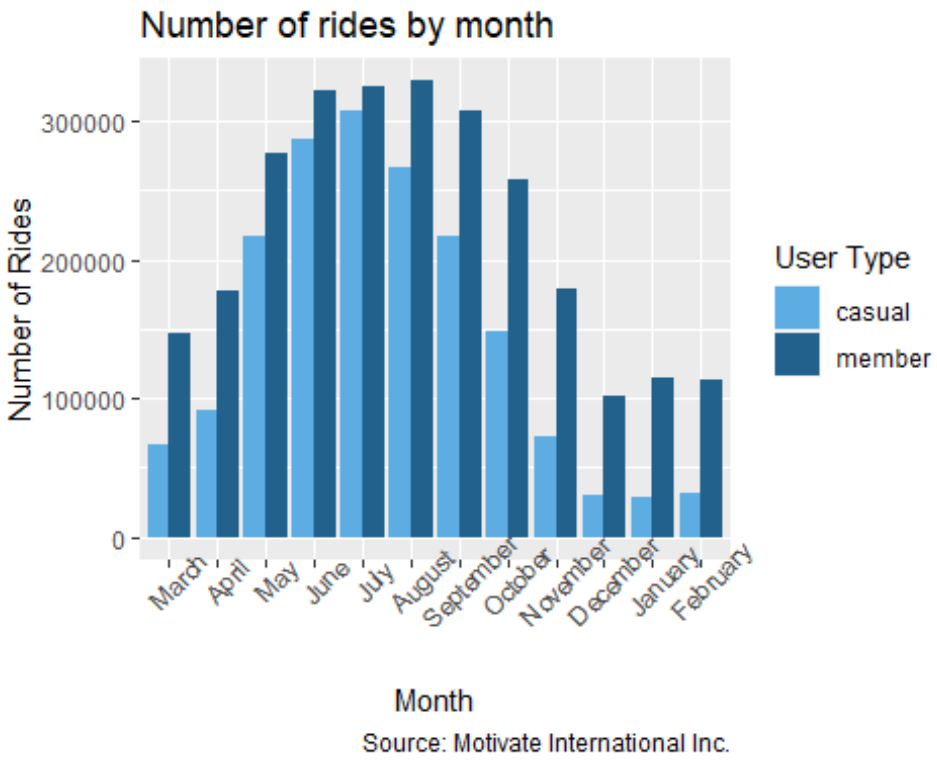
```
tripdata_v2 %>%
  group_by(member_casual) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = member_casual, y = number_of_rides, fill = member_casual)) +
  geom_col(position = 'dodge') +
  labs(title="Total number of rides by user type",
       caption = "Source: Motivate International Inc.",
       x = "Member vs Casual", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- In this graph, we can see that Cyclistic has more annual members than casual users.

Number of rides by month

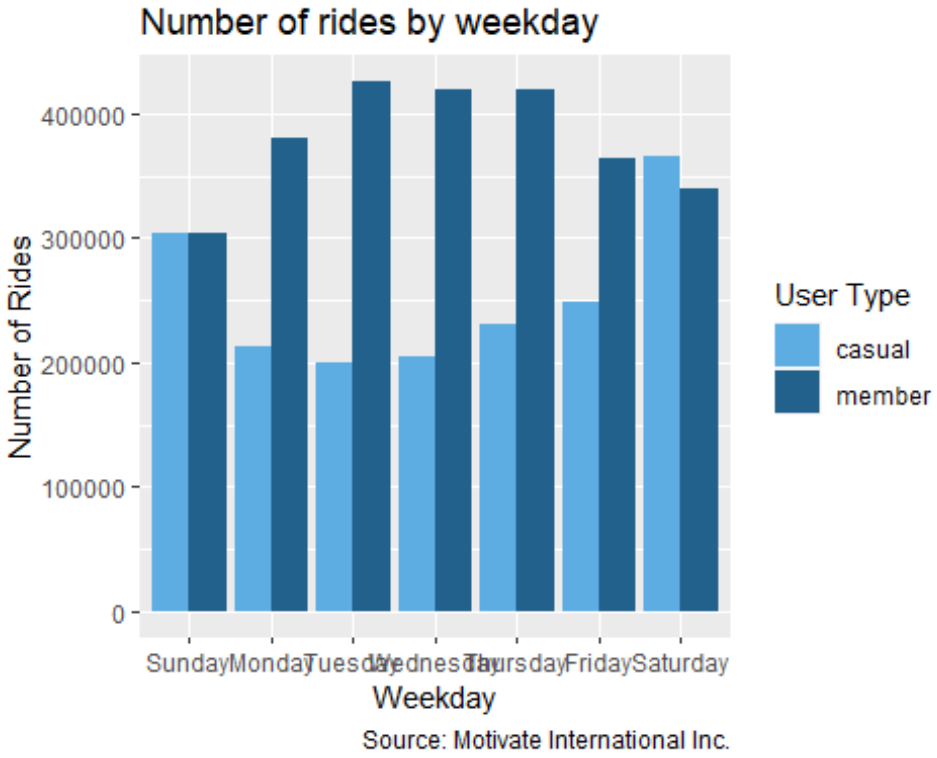
```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, month) %>%
  ggplot(aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title = "Number of rides by month",
       caption = "Source: Motivate International Inc.",
       x = "Month", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- The graphic above shows that annual members were the most frequent users of shared bikes in all months observed.
- The summer period has recorded the highest number of rides.
- The lowest number of rides was in the winter.

Number of rides by weekday

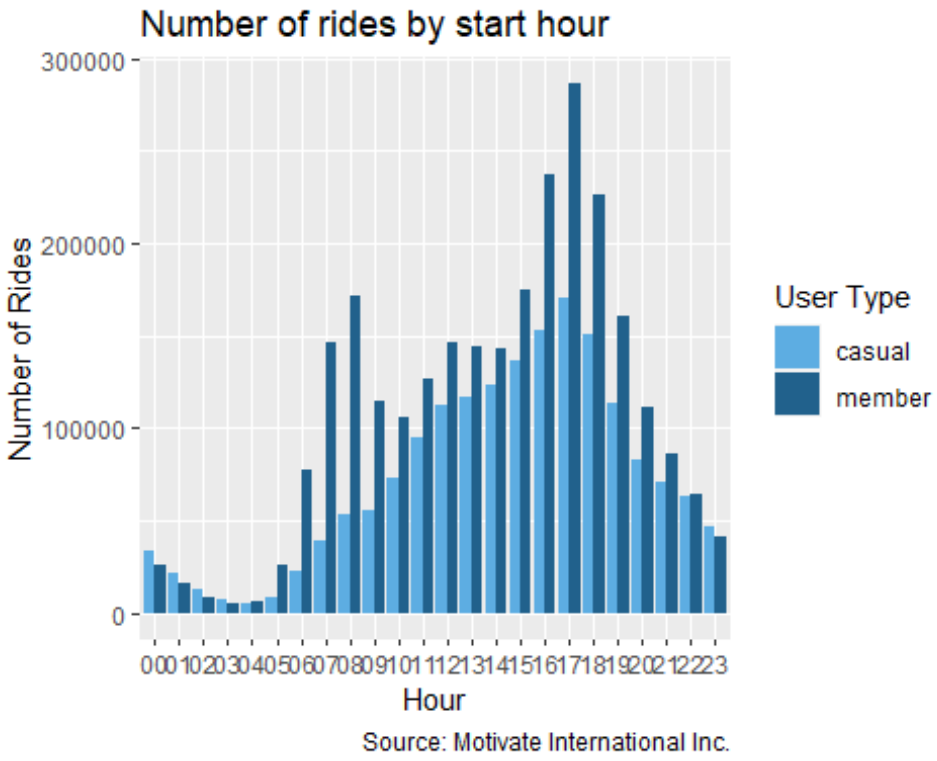
```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Number of rides by weekday",
  caption = "Source: Motivate International Inc.",
  x = "Weekday", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- Regarding weekdays, the graph shows that annual members used shared bikes most often on weekdays.
- Casual users use bikes most frequently on weekends.

Number of rides by hour

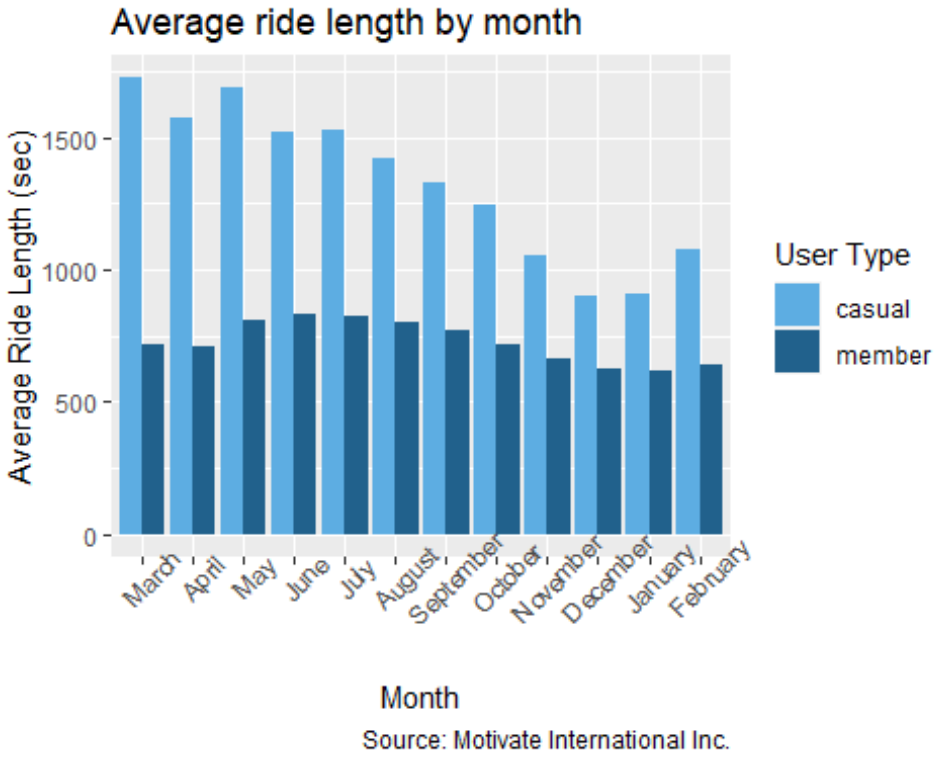
```
tripdata_v2 %>%
  group_by(member_casual, start_hour) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x = start_hour, y = number_of_rides, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Number of rides by start hour",
       caption = "Source: Motivate International Inc.",
       x = "Hour", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- Annual members use the bikes most often from 06:00 a.m. until 09:00 a.m. and from 03:00 p.m. until 07:00 p.m.
- From 11:00 p.m. to 03:00 a.m., casual users outnumbered annual members in the number of trips.

Average ride length by month

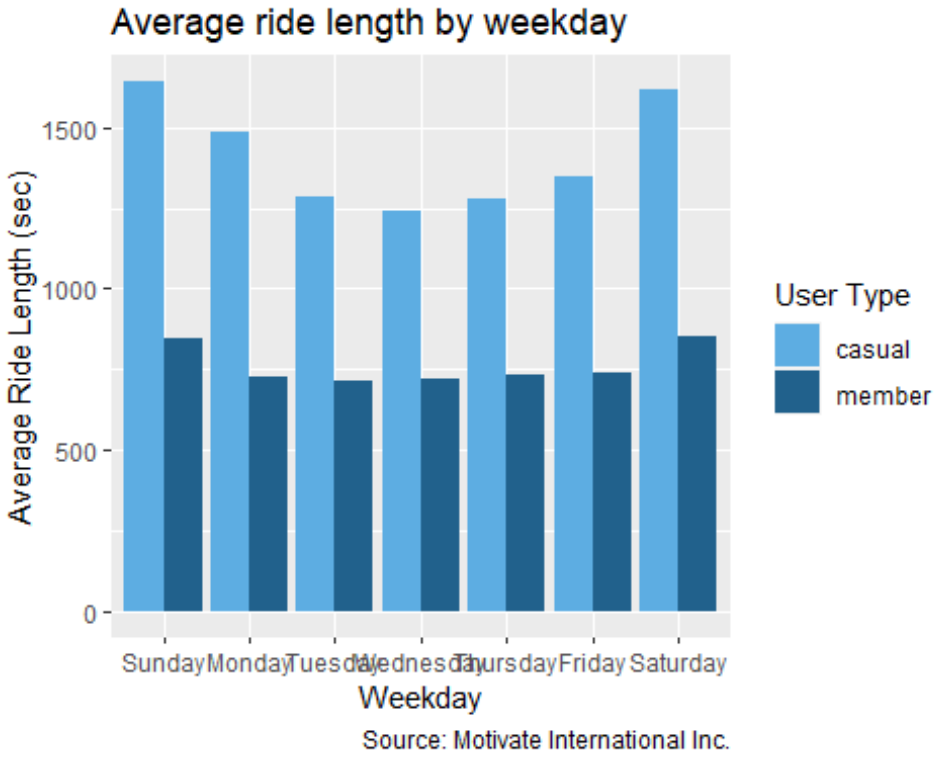
```
tripdata_v2 %>%
  group_by(member_casual, month) %>%
  summarise(average_ride_length = mean(ride_length)) %>%
  ggplot(aes(x = month, y = average_ride_length, fill = member_casual)) +
  geom_col(position = "dodge") +
  theme(axis.text.x = element_text(angle = 45)) +
  labs(title = "Average ride length by month",
       caption = "Source: Motivate International Inc.",
       x = "Month", y = "Average Ride Length (sec)", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- The average ride length made by casual users was higher than that of annual members in all the months observed.
- The spring months have recorded the highest average ride length.

Average ride length by weekday

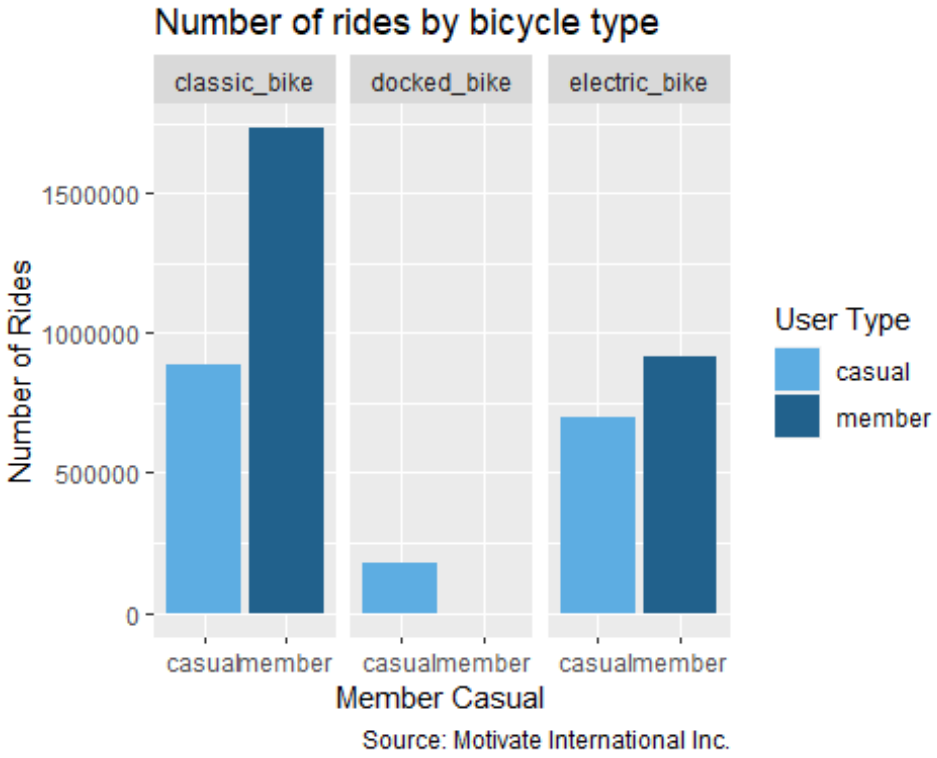
```
tripdata_v2 %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  arrange(member_casual, day_of_week) %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Average ride length by weekday",
  caption = "Source: Motivate International Inc.",
  x = "Weekday", y = "Average Ride Length (sec)", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- In the graph above, casual users used shared bikes for a longer average ride length than annual members on all weekdays.
- The highest average ride length has recorded during the weekends.

Number of rides by bicycle type

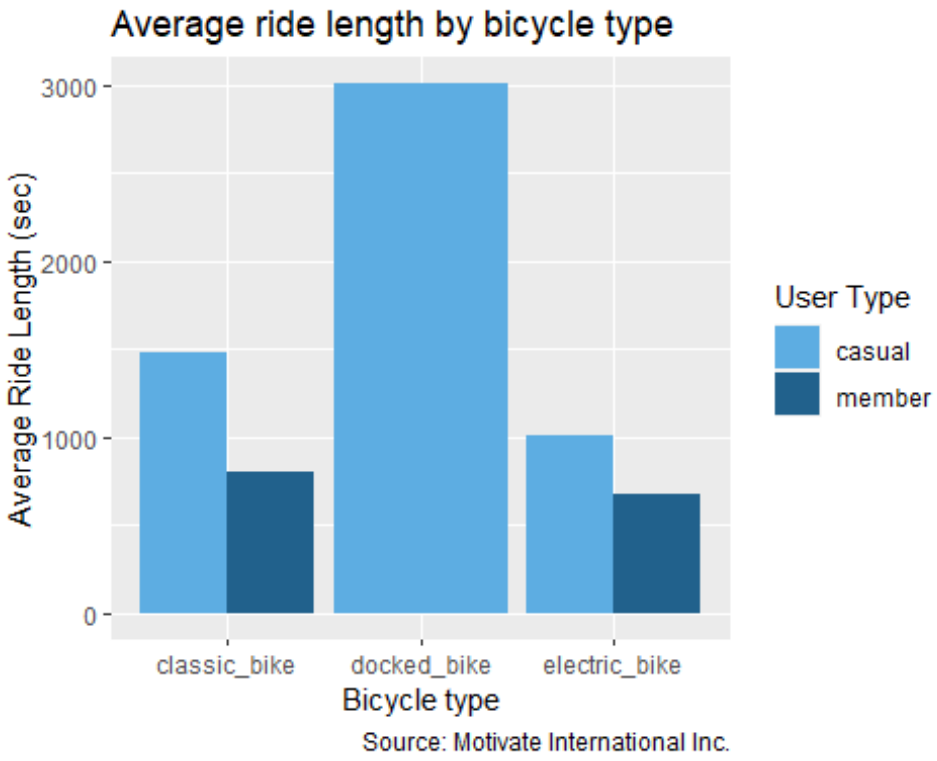
```
tripdata_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(number_of_rides = n(), average_duration = mean(ride_length)) %>%
  ggplot(aes(x=member_casual, y=number_of_rides, fill=member_casual)) +
    geom_col(position = "dodge") +
  facet_wrap(~rideable_type) +
  labs(title = "Number of rides by bicycle type",
       caption = "Source: Motivate International Inc.",
       x = "Member Casual", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2", "member" = "#21618C"))
```



- In this graphic, we can see that only casual users used docked bikes.
- Both casual users and annual members have a preference for classic bikes.

Average ride length by bicycle type

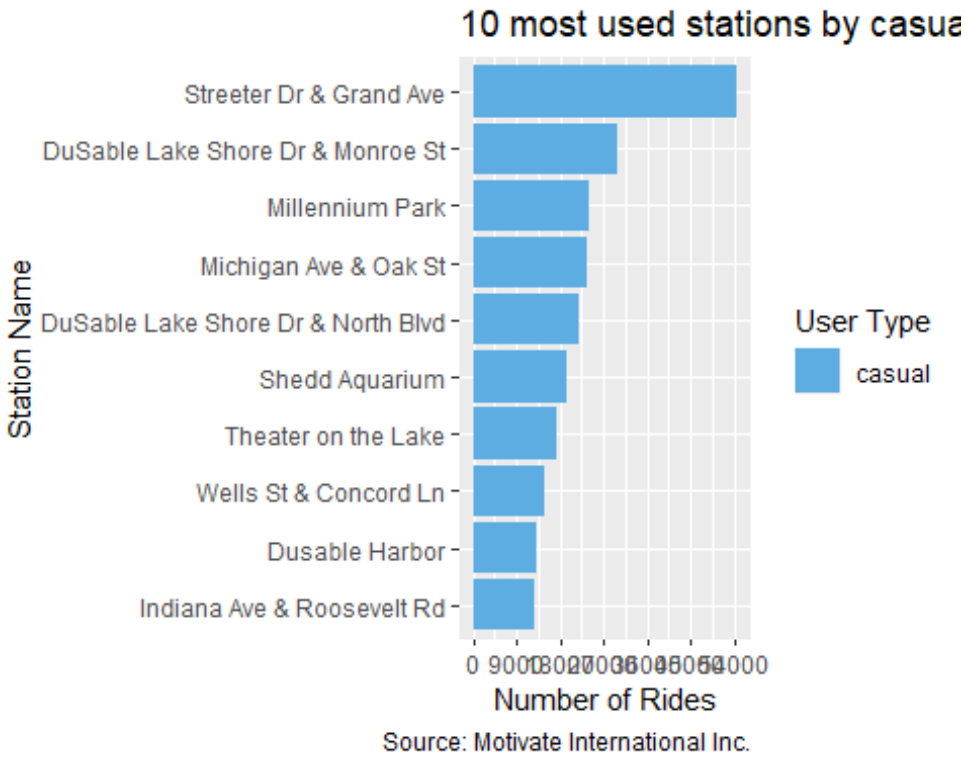
```
tripdata_v2 %>%
  group_by(member_casual, rideable_type) %>%
  summarise(average_ride_length = mean(ride_length)) %>%
  ggplot(aes(x = rideable_type, y = average_ride_length, fill = member_casual)) +
  geom_col(position = "dodge") +
  labs(title = "Average ride length by bicycle type",
       caption = "Source: Motivate International Inc.",
       x = "Bicycle type", y = "Average Ride Length (sec)", fill = "User Type") +
  scale_fill_manual(values = c("casual" = "#5DADE2", "member" = "#21618C"))
```



- Casual users use docked bikes with an average ride length higher than classic and electric bikes.

10 most used stations by casual users

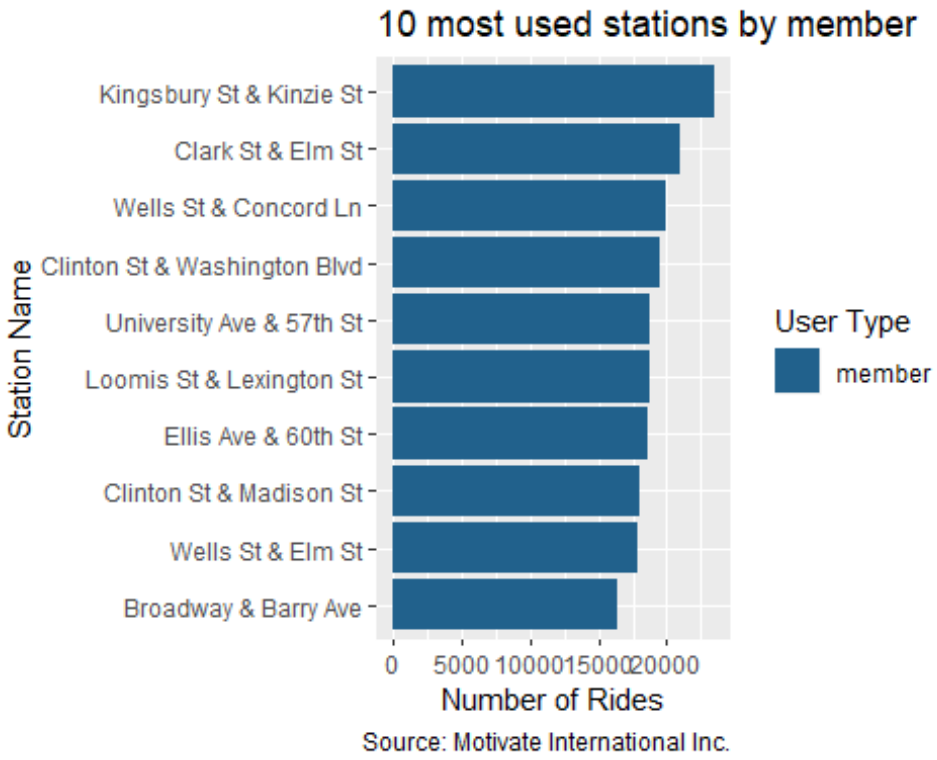
```
tripdata_v2 %>%
  group_by(member_casual, start_station_name) %>%
  filter(member_casual == "casual") %>%
  summarise(number_of_rides = n()) %>%
  arrange(desc(number_of_rides)) %>%
  head(10) %>%
  ggplot(aes(x= reorder(start_station_name, number_of_rides), y=number_of_rides, fill=me
mber_casual)) +
  scale_y_continuous(breaks = seq(0, 60000, 9000)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "10 most used stations by casual",
       caption = "Source: Motivate International Inc.",
       x = "Station Name", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("casual" = "#5DADE2"))
```



- This chart shows that the top 10 stations used by casual users over the last 12 months are all located near tourist attractions in Chicago.

10 most used stations by annual members

```
tripdata_v2 %>%
  group_by(member_casual, start_station_name) %>%
  filter(member_casual == "member") %>%
  summarise(number_of_rides = n()) %>%
  arrange(desc(number_of_rides)) %>%
  head(10) %>%
  ggplot(aes(x= reorder(start_station_name, number_of_rides), y=number_of_rides, fill=me
mber_casual)) +
  scale_y_continuous(breaks = seq(0, 25000, 5000)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "10 most used stations by member",
       caption = "Source: Motivate International Inc.",
       x = "Station Name", y = "Number of Rides", fill = "User Type") +
  scale_fill_manual(values = c ("member" = "#21618C"))
```



- The graph shows that the top 10 stations used by annual members are near commercial, residential, and educational areas in Chicago.

Exporting the clean dataset

I will export my cleaned dataset tripdata_v2, intending to load it in Tableau to visualize our data and create a dashboard.

```
write.csv(tripdata_v2, file = 'path/tripdata_v2.csv')
```

The dashboard is in the following link

Key Findings

What story do the data tell?

According to the analysis, the data tell us:

- Cyclistic has about 60.04% annual members and about 39.96% casual users. However, Cyclistic is mainly composed of Annual members.
- Throughout the 12-month data period, annual members were the most frequent riders, with the highest number of rides in the summer and the lowest in the winter.
- During the week, Cyclistic has more than 500,000 cyclists making trips daily, with the highest number of trips made in the afternoon and the higher average trip duration on weekends.
- Annual members and casual users have a preference for classic bikes.

How do annual members and casual users use Cyclistic bikes differently?

The cyclists differ in the following way:

- Annual members use Cyclistic bikes during the week from 06:00 a.m. to 09:00 a.m. and from 03:00 p.m. to 07:00 p.m. at stations close to commercial, residential, and educational institutions areas such as universities and schools.
- Casual users take longer trips, ride more frequently on weekends, all at stations close to tourist attractions, and use Cyclistic bikes more than annual members, generally from 11:00 p.m. to 03:00 a.m.

Conclusion

These findings allow us to conclude that annual members exhibit behaviors that indicate that they are customers who use Cyclistic bikes to go to work, educational institutions, and their homes.

Regarding casual users, some trends show that they are customers who use Cyclistic bikes for leisure and tourism, visiting Chicago’s major tourist attractions.

Act

In the final step of the data analysis process - Act, I will provide stakeholders with three recommendations based on the findings so that they can make data-driven decisions.

Recommendations

1. Since casual users have two options for pricing plans: single-ride passes and full-day passes, the first recommendation would be to create a new weekend membership plan, as casual users tend to use Cyclistic's bikes more frequently during the weekend.
2. I also recommend creating another member plan for the months in the summer period, a summer plan, as casual users reached the highest number of rides in that period.
3. The third recommendation would make partnerships with hotels, restaurants, and tourist attractions during the summer period, offering benefits for casual users when becoming members.

Extra recommendations:

4. Increasing the number of classic bikes during the summer, as we know, we have a peak in the summer, and casual users and annual members prefer classic bikes.
5. Implement a reward program in the app, especially for members, where members earn points and accumulate them in their account for rewards to redeem later or exchange for extra time. This reward program can encourage casual users to become members as they make longer trips.

Further Analysis

Having carried out the analysis, I'd like to point out that there are possibilities for further exploration, as the dataset did not include variables such as age, genre, and others, and we also have other ways of gaining more complete insights as I will mention below:

- Gather more data: (e.g. age, gender, address, prices) will give us more complete insights.

What is the age distribution of cyclists in the data? How does the plan price affect the choice of users?

- Use external datasets: (e.g. Chicago temperature, Chicago public transportation).

May casual users be more likely to convert to annual members if Cyclistic establishes integration of cycling and public transportation in the member plan?

- Conduct a survey: Ask users relevant questions.

What is your primary reason for cycling? (e.g. commuting, exercise, leisure)? Do you use any other modes of transportation with cycling? How satisfied are you with the cycling infrastructure in your area?

Slide Presentation

In this link you can see the slide presentation

So, I would like to thank Google, Coursera, and others involved in this program, where I've dedicated a considerable time to gaining more knowledge, learning new skills and tools, and increasing my interest in the field, which offers many opportunities for both personal and professional growth. Thank you so much!

Thank you for your time! Your feedback is welcome!