

The Cyclistic Bike-share Analysis Case Study

Albert Joonyoung Park

2023-02-05

Contents

1	Introduction	1
2	Scenario	1
2.1	Characters and teams	2
2.2	About the company	2
2.3	Project deliverables	2
3	Ask	3
4	Prepare	3
5	Process	4
6	Analyze	7
7	Share	18
8	Act	22
8.1	Key findings	22
8.2	Three recommendations	22

1 Introduction

This document serves as the case study capstone project for the Google Data Analytics Professional Certificate. Throughout this case study, I undertook various real-world tasks as a junior data analyst. I was employed at Cyclistic, a fictional company, where I had the opportunity to collaborate with diverse characters and team members. The project aimed to address key business questions by following the structured data analysis process, which included the steps of **asking** questions, **preparing** data, **processing** information, **analyzing** insights, **sharing** findings, and taking appropriate **actions**.

2 Scenario

Cyclistic is a fictional bike-share company based in Chicago, dedicated to optimizing its future success by maximizing the number of annual memberships. As the director of marketing, you recognize the importance of understanding the distinct usage patterns between casual riders and annual members in order to develop

an effective marketing strategy. The goal is to convert casual riders into loyal annual members, ultimately driving the company's growth.

2.1 Characters and teams

- **Cyclistic:** A bike-share program that features more than 5,800 bicycles and 600 docking stations. Cyclistic sets itself apart by also offering reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike. The majority of riders opt for traditional bikes; about 8% of riders use the assistive options. Cyclistic users are more likely to ride for leisure, but about 30% use them to commute to work each day.
- **Lily Moreno:** The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.
- **Cyclistic marketing analytics team:** A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic's mission and business goals — 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.

2.2 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.

2.3 Project deliverables

Produce 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.

3 Ask

Key tasks:

1. Identify the business task

The business objective is to convert occasional riders into annual members, which is crucial for driving the company's future growth. The primary purpose of this analysis is to identify trends and differences in riding behavior between casual riders and annual members. The insights derived from this analysis will serve as a foundation for informing the company's marketing strategies and efforts aimed at converting casual riders into loyal annual members.

2. Key stakeholders:

Cyclistic executive team, Director of Marketing (Lily Moreno), Marketing Analytics team

**** Statement of the business task ****

The primary objective of this project is to analyze the historical bike trip data of Cyclistic in order to identify and understand the differences between casual riders and annual members. By examining the data, we aim to uncover trends and patterns that distinguish the riding behaviors of these two user groups.

4 Prepare

Key tasks:

1. Download data and store it appropriately.

The previous 12 months of Cyclistic trip data is available for download [here](#). They are made available by Motivate International Inc., under [license](#). 12 CSV files have been downloaded and saved in the folder C:\Users\Joon\git\albert-joonyoung-park.github.io\data.

2. Identify how it's organized

The data files are in CSV format and are organized into individual monthly files for the past 12-month period.

3. Sort and filter the data.

A spreadsheet was used to conduct an initial inspection of the data files, revealing several issues that require data cleaning and transformation.

- Many columns exhibit missing values, including: `started_at`, `ended_at`, `start_station_name`, `end_station_name`, `end_station_id`, `start_lat`, and `end_lat`.
- The `ride_id` column contains inconsistent data types, with a mixture of character and numeric data.
- When calculating the 'ride length' based on the 'started_at' and 'ended_at' columns, some results indicate unrealistic durations, such as trips exceeding 24 hours or trips with durations of just a few minutes or zero.

- All files share identical column names and have the same number of columns.
- Each data file contains a substantial number of rows, ranging from 180,000 to 800,000+ rows.

4. Determine the credibility of the data.

ROCCC measurement was applied to check the credibility of the data.

- **Reliable (R):** The dataset provides a complete 12-month bike trip data, but it may have some limitations or biases.
- **Original (O):** The dataset is sourced directly from Cyclistic, the company in question, making it a first-party and publicly available dataset.
- **Comprehensive (C):** The dataset contains sufficient information to address the specific business task at hand.
- **Current (C):** The dataset is up-to-date, encompassing the full-year bike trip data for the current period.
- **Cited (C):** The dataset is publicly available and properly cited online, providing clear references and license information from both the company and the municipality.

**** A description of all data sources used ****

The analysis utilized 12 bike trip CSV data files obtained from Cyclistic, covering the period from May 1, 2022, to April 30, 2023. These data files were sourced as first-party public data from Motivate International Inc. The original files used for analysis are named as follows:

File name
202205-divvy-tripdata.csv
202206-divvy-tripdata.csv
202207-divvy-tripdata.csv
202208-divvy-tripdata.csv
202209-divvy-publictripdata.csv
202210-divvy-tripdata.csv
202211-divvy-tripdata.csv
202212-divvy-tripdata.csv
202301-divvy-tripdata.csv
202302-divvy-tripdata.csv
202303-divvy-tripdata.csv
202304-divvy-tripdata.csv

5 Process

Key tasks

- 1. Check the data for errors.
- 2. Choose your tools.
- 3. Transform the data so you can work with it effectively.
- 4. Document the cleaning process.

For the tasks of data inspection, cleaning, and manipulation, the R programming language was selected as the tool of choice. The versatility and powerful data manipulation capabilities of R make it well-suited

for these tasks. In the subsequent sections, I present a series of code blocks that outline the step-by-step process for cleaning and preparing the data for future analysis.

**** Prepare the local workspace by setting up the necessary environment and loading the required libraries for efficient data processing. ****

```
# Set up working directory in my environment.
setwd("C:/Users/Joon/git/albert-joonyoung-park.github.io/reports")

# Load required libraries for data processing.
library(tidyverse)
library(dplyr)
library(lubridate)
library(skimr)
library(ggplot2)
library(knitr)
library(scales)
library(leaflet)
library(kableExtra)
```

**** Import the data files, Remove rows with missing values and duplicates, Transform the dataset to increase aggregation level for analysis. ****

```
# List all the CSV data file names to be imported.
file_names <- list.files(path = "../data/",
                        pattern = "*.csv", full.names = FALSE,
                        recursive = FALSE)

# Import csv data files one at a time and clean, transform the data
total_num_rows <- 0
total_num_rows_dropped <- 0
total_num_rows_duplicated <- 0

for (name in file_names) {

  # Import a csv file
  data <- read_csv(paste("../data/", name, sep = ""))
  num_rows <- nrow(data)
  #print(paste(num_rows, "rows imported from", name)) # number of rows imported
  total_num_rows <- total_num_rows + num_rows

  # Cleaning - drop rows with missing values - NA
  data_na_dropped <- drop_na(data)
  num_rows_dropped <- num_rows - nrow(data_na_dropped)
  #print(paste(num_rows_dropped, "rows with missing value dropped from", name))
  total_num_rows_dropped <- total_num_rows_dropped + num_rows_dropped

  # Cleaning - remove duplicates
  data_unique_na_dropped <- distinct(data_na_dropped)
  num_rows_duplicated <- nrow(data_unique_na_dropped) - nrow(data_na_dropped)
```

```

total_num_rows_duplicated <- total_num_rows_duplicated + num_rows_duplicated

# Transformation:
# add columns to increase aggregation levels for analysis.
# limit ride_length values
trip_month_data_clean <-
  data_unique_na_dropped |>
    mutate(
      start_date = as.Date(started_at),
      start_hour = format(started_at, "%H"),
      start_month = format(start_date, "%m"),
      start_day = format(start_date, "%d"),
      start_year = format(start_date, "%Y"),
      #start_weekday = format(start_date, "%A"),
      start_weekday = wday(start_date, week_start = 7),
      ride_length = difftime(ended_at, started_at, units = "secs"),
      .after = started_at
    ) |>
    filter(ride_length > 0 & ride_length <= 3600) # limit ride_length

# Save cleaned data files as a CSV file
write_csv(trip_month_data_clean,
          paste("../data/cleaned_datasets/", name, sep = ""))
}

# print some row number changes and structure of cleaned data being saved
total_num_rows # 5859061 rows

```

```
[1] 5859061
```

```
total_num_rows_dropped # 1325062 rows
```

```
[1] 1325062
```

```
total_num_rows_duplicated # 0 row
```

```
[1] 0
```

```
glimpse(trip_month_data_clean)
```

```

Rows: 315,757
Columns: 20
$ ride_id          <chr> "5B6500E1E58655C0", "AA65D25D69AF771F", "079FB2C196~
$ rideable_type    <chr> "classic_bike", "classic_bike", "electric_bike", "c~
$ started_at       <dtm> 2023-04-10 17:34:35, 2023-04-12 12:29:46, 2023-04--
$ start_date       <date> 2023-04-10, 2023-04-12, 2023-04-13, 2023-04-29, 20~
$ start_hour       <chr> "17", "12", "17", "20", "17", "22", "21", "11", "11~
$ start_month      <chr> "04", "04", "04", "04", "04", "04", "04", "04", "04~
$ start_day        <chr> "10", "12", "13", "29", "20", "14", "07", "05", "05~
$ start_year       <chr> "2023", "2023", "2023", "2023", "2023", "2023", "20~

```

```

$ start_weekday      <dbl> 2, 4, 5, 7, 5, 6, 6, 4, 4, 2, 7, 3, 1, 4, 3, 1, 2, ~
$ ride_length        <drtm> 1681 secs, 1454 secs, 94 secs, 3 secs, 1307 secs, ~
$ ended_at           <dtm> 2023-04-10 18:02:36, 2023-04-12 12:54:00, 2023-04-~
$ start_station_name <chr> "Avenue O & 134th St", "Cottage Grove Ave & 51st St~
$ start_station_id   <chr> "20214", "TA13090000067", "TA13060000002", "TA1309000~
$ end_station_name   <chr> "Avenue O & 134th St", "Cottage Grove Ave & 51st St~
$ end_station_id     <chr> "20214", "TA13090000067", "TA13060000002", "TA1309000~
$ start_lat          <dbl> 41.65187, 41.80304, 41.86243, 41.80304, 41.90303, 4~
$ start_lng          <dbl> -87.53967, -87.60662, -87.65115, -87.60662, -87.697~
$ end_lat            <dbl> 41.65187, 41.80304, 41.86238, 41.80304, 41.92269, 4~
$ end_lng            <dbl> -87.53967, -87.60662, -87.65106, -87.60662, -87.697~
$ member_casual      <chr> "member", "member", "member", "member", "casual", "~

```

6 Analyze

Key tasks

- 1. Aggregate your data so it's useful and accessible.
- 2. Organize and format your data.
- 3. Perform calculations.
- 4. Identify trends and relationships.

Deliverable

A summary of your analysis

**** Step 1: Aggregate the individual cleaned data files and create a single duplicate-free dataset for analysis ****

```

#-----
# Aggregate the cleaned dataset into a single data frame and inspect further
#-----

# Get all the cleaned data file names for aggregation
file_names <- list.files(path = "../data/cleaned_datasets/",
                        pattern = "*.csv", full.names = FALSE,
                        recursive = FALSE)

# Import the first monthly cleaned csv file, so it constructs the data frame
# for full year aggregation
full_year_trips <- read_csv(paste("../data/cleaned_datasets/", file_names[1], sep = ""))

# Aggregate the remainig csv files, starting from the second
for(name in file_names[2:length(file_names)]) {
  # Read csv file
  trip_month_data_clean <- read_csv(paste("../data/cleaned_datasets/",
                                          name, sep = ""))

  # Aggregate the monthly data
  full_year_trips <- rbind(full_year_trips, trip_month_data_clean)
}

```

```

# Drop duplicates
full_year_trips <- drop_na(full_year_trips)

# Save monthly trip data into a aggregated csv file
write_csv(trip_month_data_clean, "../data/analyzed_datasets/full_year_trips.csv",
          append = TRUE)
}

# Check the overview of the aggregated dataset
# skim(full_year_trips) # 4393697

```

**** Step 2: Descriptive analysis on ride length and frequencies for full year, and quarter, to save the summary and visualizations ****

```

#-----
# Analyze the aggregated data and create summary to find trends
# Use question, data summary, visualization format
# Save each summary
#-----

# !. Basic descriptive analysis on ride length per member and casual groups.

# Total ride frequency and percentage per member and casual group
total_ride = nrow(full_year_trips)
ride_frequency_membership <-
  full_year_trips |>
  group_by(membership = member_casual) |>
  summarize(
    ride_frequency = n(),
    ride_freq_pct = n() / total_ride * 100
  )

# visualization of total ride frequency, percentage per member and casual group

p <-
ride_frequency_membership |>
  ggplot(aes(x = membership, y = ride_frequency, fill = membership)) +
  geom_col() +
  geom_text(aes(label = comma(ride_frequency)), size = 3, vjust = -0.3) +
  geom_text(aes(label = paste(round(ride_freq_pct, digits = 2), "%")), size = 6, vjust = 5.0, color = "v
  labs(title = "Total ride frequency by membership",
       x = "Membership", y = "Ride Frequency (2022 - 2023)", fill = "Membership" ) + scale_y_continuous

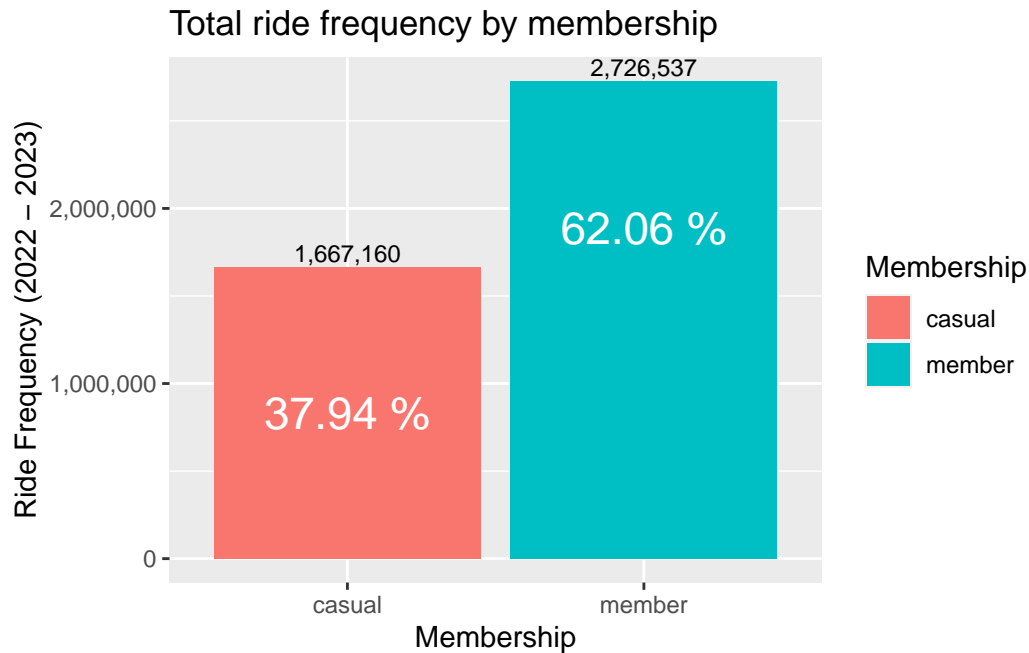
# Save the summary data
write_csv(ride_frequency_membership, "../data/analyzed_datasets/ride_frequency_membership.csv")

# Save the plot
plot_name <- deparse(substitute(ride_frequency_membership))

```



```
ggsave(paste("../data/analyzed_datasets/", plot_name, ".png"), p, width = 6, height = 4, dpi = 300)
p
```



```
# ride_frequency by quarter per member and casual group
ride_frequency_q_membership <-
  full_year_trips |>
  group_by(qtr = paste("QTR", quarter(start_date)),
           membership = member_casual) |>
  summarize(
    ride_frequency = n()
  )

# Save the summary data
write_csv(ride_frequency_q_membership,
          "../data/analyzed_datasets/ride_frequency_q_membership.csv")

# Mean, median, max and min ride length per member and casual groups.
ride_length_membership <-
  full_year_trips |>
  group_by(membership = member_casual) |>
  summarize(
    mean = round(mean(ride_length / 60)),
    median = round(median(ride_length / 60)),
    max = round(max(ride_length / 60)),
    min = round(min(ride_length / 60))
  )
```

```

    )

ride_length_membership <-
  ride_length_membership |>
    pivot_longer(cols = c(mean, median, max, min),
                  names_to = "statistic",
                  values_to = "ride_length")

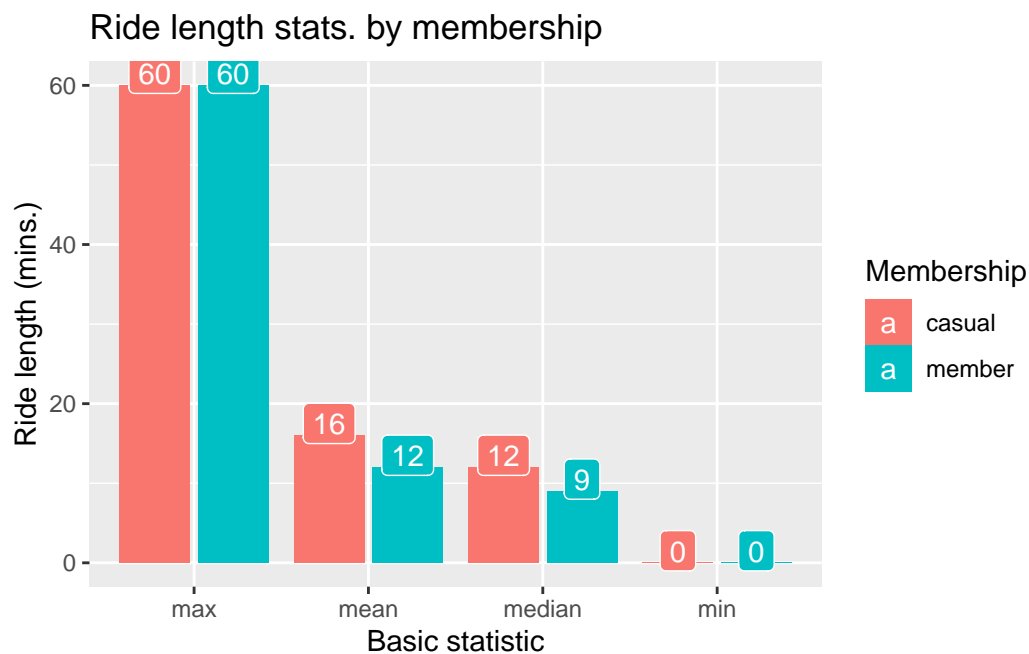
p <-
  ride_length_membership |>
    ggplot(aes(x = statistic, y = ride_length, fill = membership)) +
    geom_col(position = "dodge2") +
    geom_label(
      aes(label = ride_length),
      size = 4,
      color = "white",
      position = position_dodge(width = 0.9), vjust = 0.2
    ) +
    labs(title = "Ride length stats. by membership", size = 14) +
    labs(x = "Basic statistic", y = "Ride length (mins.)", fill = "Membership")

# Save the summary data
write_csv(ride_length_membership,
          "../data/analyzed_datasets/ride_length_membership.csv")

# Save the plot
plot_name <- deparse(substitute(ride_length_membership))
ggsave(paste("../data/analyzed_datasets/", plot_name, ".png"), p, width = 6, height = 4, dpi = 300)

p

```



**** Step 3: Descriptive analysis on ride length for day of the week, to save the summary and visualizations ****

```
# 2. Average ride length of day of week per casual and member group
ride_length_weekday_avg <-
  full_year_trips |>
  group_by(start_weekday, member_casual) |>
  summarize(avg_ride_length = round(mean(ride_length) / 60)) |>
  arrange(start_weekday) |>
  mutate(start_weekday = recode(start_weekday,
                                "1" = "Sunday",
                                "2" = "Monday",
                                "3" = "Tuesday",
                                "4" = "Wednesday",
                                "5" = "Thursday",
                                "6" = "Friday",
                                "7" = "Saturday")
  )
ride_length_weekday_avg
```

A tibble: 14 x 3

Groups: start_weekday [7]

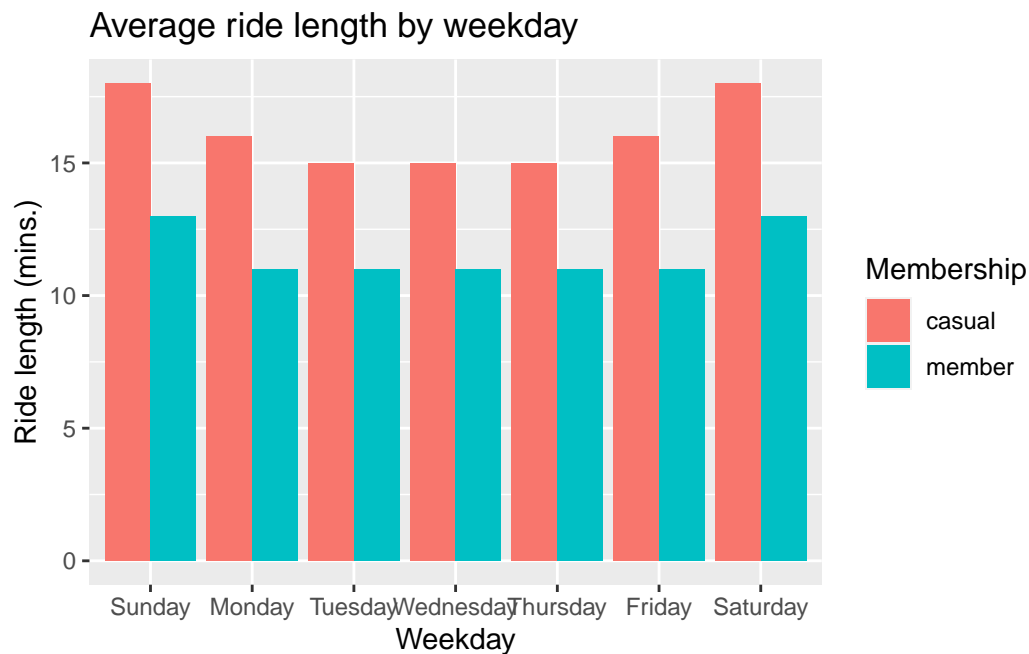
	start_weekday	member_casual	avg_ride_length
	<chr>	<chr>	<dbl>
1	Sunday	casual	18
2	Sunday	member	13
3	Monday	casual	16
4	Monday	member	11

5	Tuesday	casual	15
6	Tuesday	member	11
7	Wednesday	casual	15
8	Wednesday	member	11
9	Thursday	casual	15
10	Thursday	member	11
11	Friday	casual	16
12	Friday	member	11
13	Saturday	casual	18
14	Saturday	member	13

```
p <-
ride_length_weekday_avg |>
ggplot() +
  geom_col(
    aes(x = factor(start_weekday,
                    levels=c("Sunday",
                             "Monday",
                             "Tuesday",
                             "Wednesday",
                             "Thursday",
                             "Friday",
                             "Saturday")
                    ),
        y = avg_ride_length,
        fill = member_casual),
    position = "dodge"
  ) +
  labs(title = "Average ride length by weekday", size = 14) +
  labs(x = "Weekday", y = "Ride length (mins.)", fill = "Membership")

# Save the summary data
write_csv(ride_length_weekday_avg,
          "../data/analyzed_datasets/ride_length_weekday_avg.csv")

# Save the plot
plot_name <- deparse(substitute(ride_length_weekday_avg))
ggsave(paste("../data/analyzed_datasets/", plot_name, ".png"), p, width = 6, height = 4, dpi = 300)
p
```



**** Step 4: Descriptive analysis on ride frequencies for day of the week, to save the summary and visualizations ****

```
# 3. Ride count for day of week per member vs casual group.
```

```
ride_count_weekday <-
  full_year_trips |>
  mutate(weekday = wday(started_at, label = TRUE)) |>
  group_by(member_casual, weekday) |>
  summarise(
    number_of_rides = n(),
    avg_ride_length = as.integer(mean(ride_length)/60)
  ) |>
  arrange(member_casual, weekday)
```

```
ride_count_weekday
```

```
# A tibble: 14 x 4
```

```
# Groups:   member_casual [2]
```

	member_casual	weekday	number_of_rides	avg_ride_length
	<chr>	<ord>	<int>	<int>
1	casual	Sun	274808	17
2	casual	Mon	194524	16
3	casual	Tue	192448	14
4	casual	Wed	201808	14
5	casual	Thu	225061	14
6	casual	Fri	245557	15
7	casual	Sat	332954	17

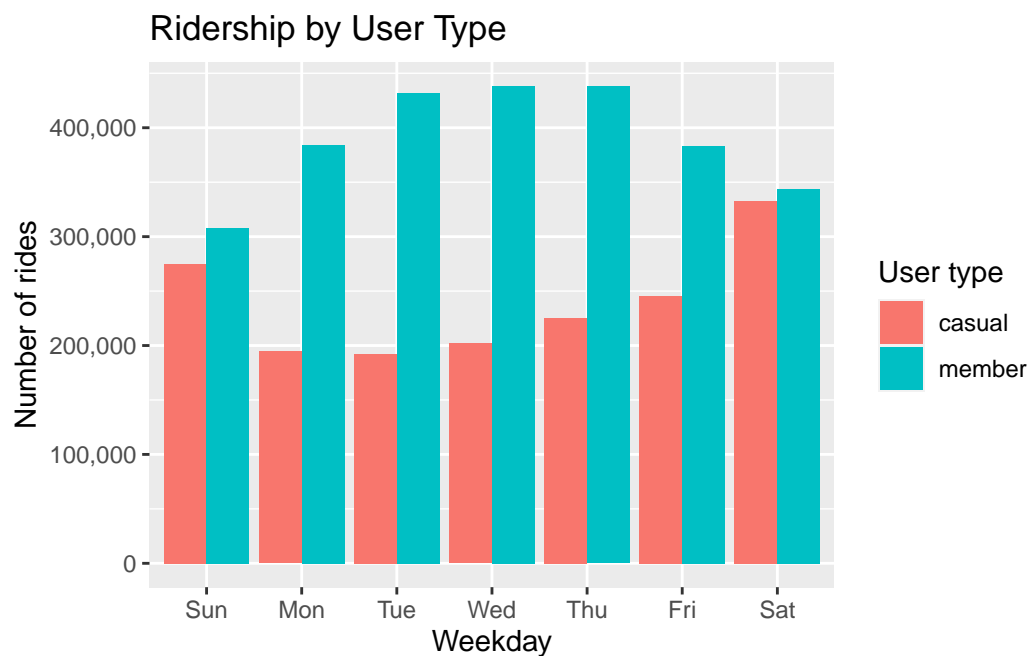
8 member	Sun	307754	12
9 member	Mon	383896	11
10 member	Tue	431802	11
11 member	Wed	438440	11
12 member	Thu	437776	11
13 member	Fri	383060	11
14 member	Sat	343809	12

```
# Visualize the number of rides by rider type
```

```
p <-
ride_count_weekday |>
  ggplot(aes(x = weekday, y = number_of_rides, fill = member_casual)) +
    geom_col(position = "dodge") +
    labs(title = "Ridership by User Type", size = 14) +
    labs(x = "Weekday", y = "Number of rides", fill = "User type") +
    scale_y_continuous(labels = comma)

# Save the summary data
write_csv(ride_count_weekday,
          "../data/analyzed_datasets/ride_count_weekday.csv")

# Save the plot
plot_name <- deparse(substitute(ride_count_weekday))
ggsave(paste("../data/analyzed_datasets/", plot_name, ".png"), p, width = 6, height = 4, dpi = 300)
p
```



**** Step 5: Descriptive analysis on rideable type usage, to save the summary and visualizations**

```
# 4. Rideable type frequencies for day of week per member vs casual group.
```

```
dfs_rideable_type_member_casual <-  
  full_year_trips |>  
  group_by(member_casual, rideable_type) |>  
  summarize(ride_count = n()) |>  
  arrange(rideable_type, member_casual, ride_count)
```

```
dfs_rideable_type_member_casual
```

```
# A tibble: 5 x 3
```

```
# Groups:   member_casual [2]
```

	member_casual	rideable_type	ride_count
	<chr>	<chr>	<int>
1	casual	classic_bike	824392
2	member	classic_bike	1739554
3	casual	docked_bike	131036
4	casual	electric_bike	711732
5	member	electric_bike	986983

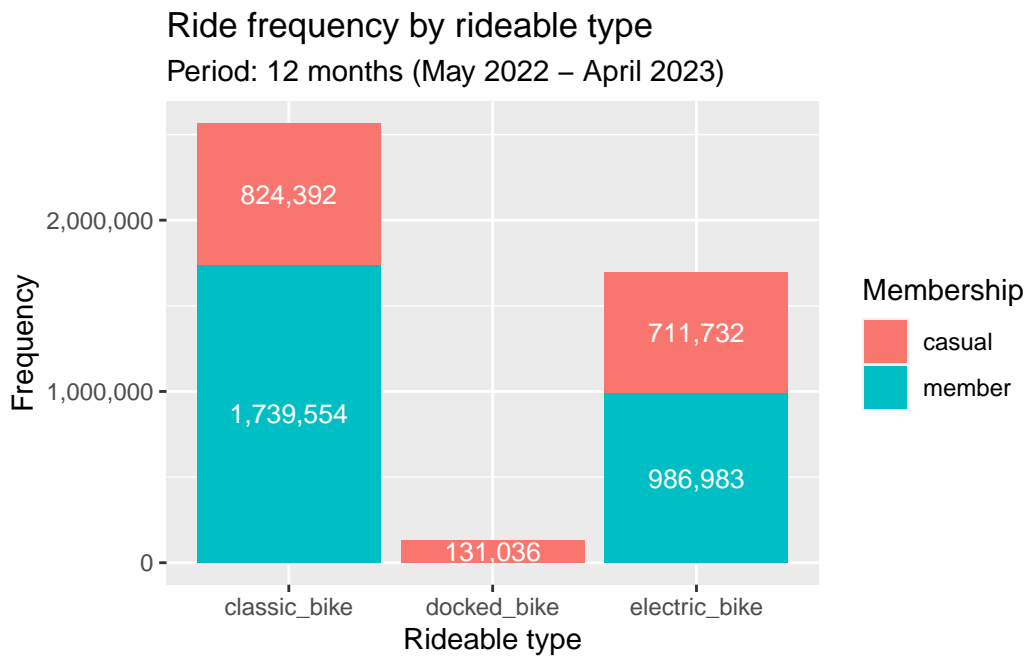
```
p <-  
dfs_rideable_type_member_casual |>  
  ggplot(aes(x = rideable_type, y = ride_count, fill = member_casual )) +  
  geom_bar(stat = "identity") +  
  geom_text(  
    aes(label = comma(ride_count)),  
    size = 3.5,  
    position = position_stack(vjust = 0.5),  
    colour = "white"  
  ) +  
  scale_y_continuous(labels = comma) +  
  labs(title = "Ride frequency by rideable type", size = 16) +  
  labs(subtitle = "Period: 12 months (May 2022 - April 2023)") +  
  labs(x = "Rideable type", y = "Frequency", fill = "Membership")
```

```
# Save the summary data
```

```
write_csv(dfs_rideable_type_member_casual,  
  "../data/analyzed_datasets/dfs_rideable_type_member_casual.csv")
```

```
# Save the plot
```

```
plot_name <- deparse(substitute(ride_count_weekday))  
ggsave(paste("../data/analyzed_datasets/", plot_name, ".png"), p, width = 6, height = 4, dpi = 300)  
p
```



**** Step 6: Descriptive analysis on top 20 most popular start stations by casual riders, to save the summary and visualizations ****

```
# 6. What stations are most frequently used by casual riders in the past 12 months?

# Retrieve frequent station information used by casual riders in the past 12 months
dfs_ride_start_loc_casual <-
  full_year_trips |>
  filter(member_casual == "casual") |>
  group_by(start_station_id, start_station_name, member_casual) |>
  summarize(
    ride_frequency = n(),
    .groups = "drop" # need to set "drop" option to assign ranking properly later
  )

dfs_ride_start_loc_casual <-
  dfs_ride_start_loc_casual |>
  arrange(desc(ride_frequency)) |>
  mutate(rank = rank(-ride_frequency, ties.method = 'min')) |>
  select(rank, start_station_id, start_station_name, ride_frequency)

dfs_ride_start_loc_casual_ranked <-
  dfs_ride_start_loc_casual |>
  rename(
    "rank" = "rank",
    "station_id" = "start_station_id",
    "station_name" = "start_station_name",
```



```

    "ride_count" = "ride_frequency"
  )

# List top 20 stations

dfs_ride_start_loc_casual_ranked_20 <- dfs_ride_start_loc_casual_ranked[1:20,]

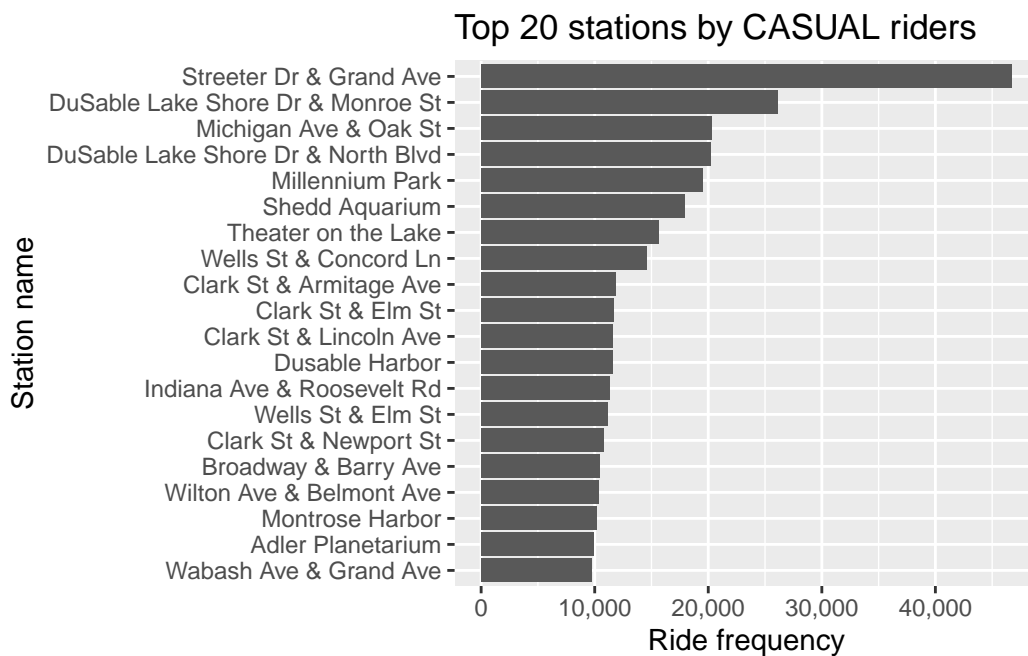
# Set ordering
dfs_ride_start_loc_casual_ranked_20$station_name <- factor(dfs_ride_start_loc_casual_ranked_20$station_name, levels = dfs_ride_start_loc_casual_ranked_20$station_name[order(dfs_ride_start_loc_casual_ranked_20$ride_count)])

p <-
  dfs_ride_start_loc_casual_ranked_20 |>
  ggplot(aes(x = ride_count, y = station_name)) +
    geom_bar(stat = "identity") +
    labs(title = "Top 20 stations by CASUAL riders", size = 14) +
    labs(x = "Ride frequency", y = "Station name") +
    scale_x_continuous(labels = comma)

# Save the summary data
write_csv(dfs_ride_start_loc_casual_ranked_20,
  "../data/analyzed_datasets/dfs_ride_start_loc_casual_ranked_20.csv")

# Save the plot
plot_name <- deparse(substitute(dfs_ride_start_loc_casual_ranked_20))
ggsave(paste("../data/analyzed_datasets/", plot_name, ".png"), p, width = 6, height = 4, dpi = 300)
p

```



```
# Create a table with proper number format

dfs_ride_start_loc_casual_ranked_20$ride_count <-
  format(dfs_ride_start_loc_casual_ranked_20$ride_count, big.mark = ",")

dfs_ride_start_loc_casual_ranked_20
```

A tibble: 20 x 4

	rank	station_id	station_name	ride_count
	<int>	<chr>	<fct>	<chr>
1	1	13022	Streeter Dr & Grand Ave	"46,750"
2	2	13300	DuSable Lake Shore Dr & Monroe St	"26,117"
3	3	13042	Michigan Ave & Oak St	"20,291"
4	4	LF-005	DuSable Lake Shore Dr & North Blvd	"20,202"
5	5	13008	Millennium Park	"19,517"
6	6	15544	Shedd Aquarium	"17,959"
7	7	TA1308000001	Theater on the Lake	"15,606"
8	8	TA1308000050	Wells St & Concord Ln	"14,555"
9	9	13146	Clark St & Armitage Ave	"11,879"
10	10	TA1307000039	Clark St & Elm St	"11,665"
11	11	13179	Clark St & Lincoln Ave	"11,580"
12	12	KA1503000064	Dusable Harbor	"11,555"
13	13	SL-005	Indiana Ave & Roosevelt Rd	"11,319"
14	14	KA1504000135	Wells St & Elm St	"11,121"
15	15	632	Clark St & Newport St	"10,816"
16	16	13137	Broadway & Barry Ave	"10,458"
17	17	TA1307000134	Wilton Ave & Belmont Ave	"10,336"
18	18	TA1308000012	Montrose Harbor	"10,129"
19	19	13431	Adler Planetarium	" 9,890"
20	20	TA1307000117	Wabash Ave & Grand Ave	" 9,716"

7 Share

The findings from the descriptive analysis have been visualized and shared on Tableau Public for easy access and interpretation. You can explore the summarized data through the following [link](#).

Ride Frequency: Past 12 mo.
(May 2022 and April 2023)

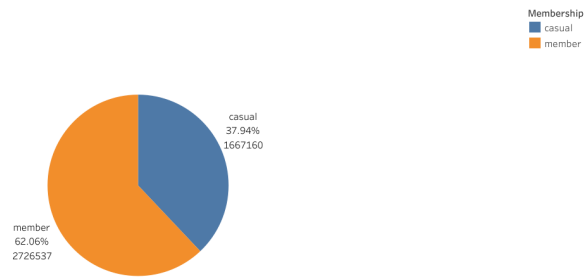


Figure 1: Ride Frequency: Past 12 mo.

Basic Ride Length Stats.

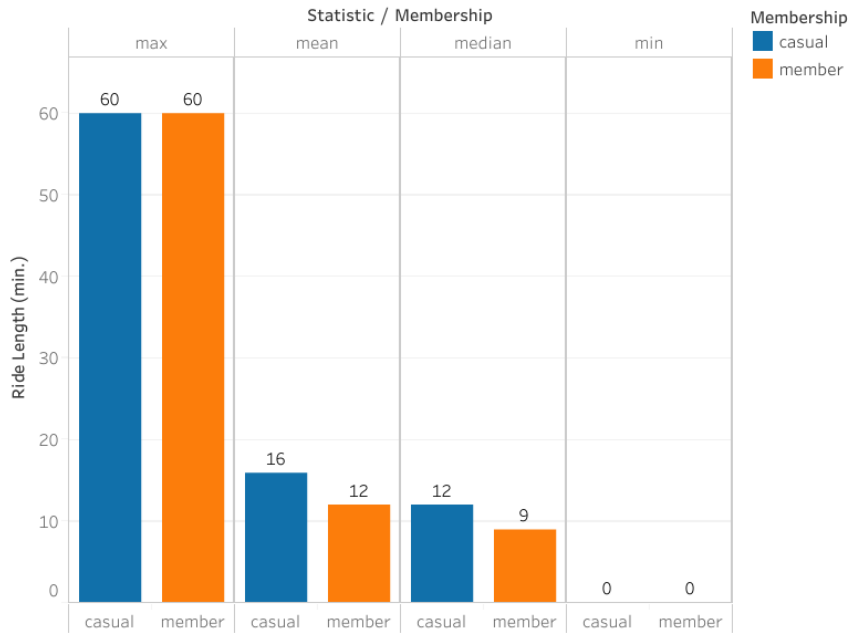


Figure 2: Basic Ride Length Stats.

Average Ride Length per Day of the Week - Casual vs. Member

The period between May 2022 and April 2023

(In minutes)

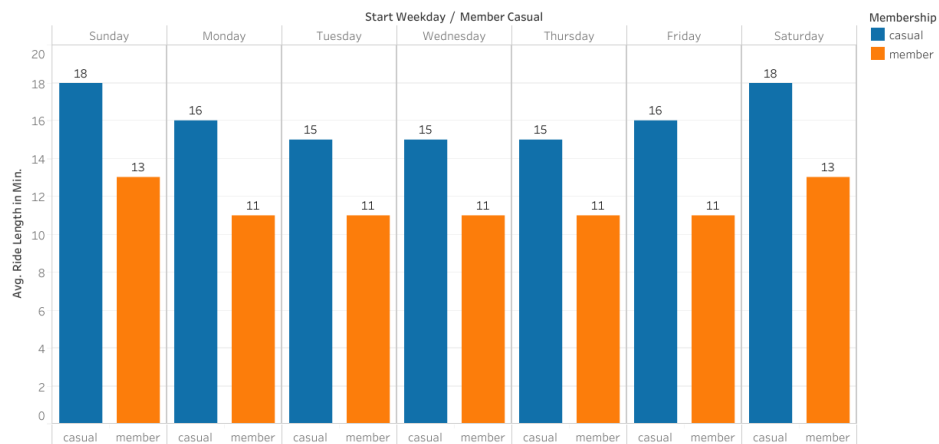
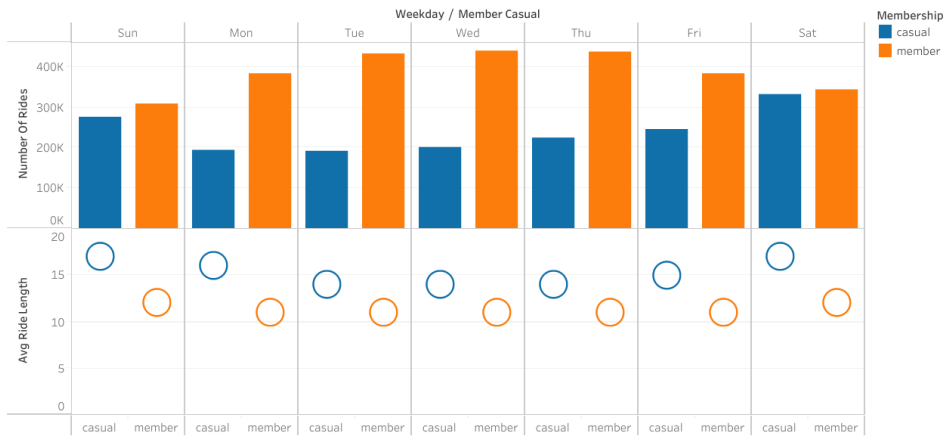


Figure 3: Average Ride Length per Day of the Week

Weekly Ride Count/Avg. Length



Ride Frequency per Yearly Quarter

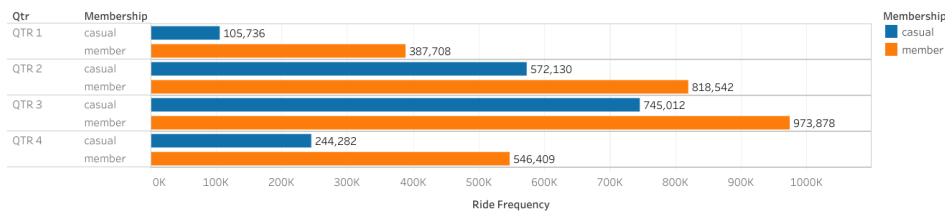


Figure 4: Ride Frequency per Yearly Quarter

Ride Frequency by Rideable Type - Causal vs. Member

Period between May 2022 and April 2023

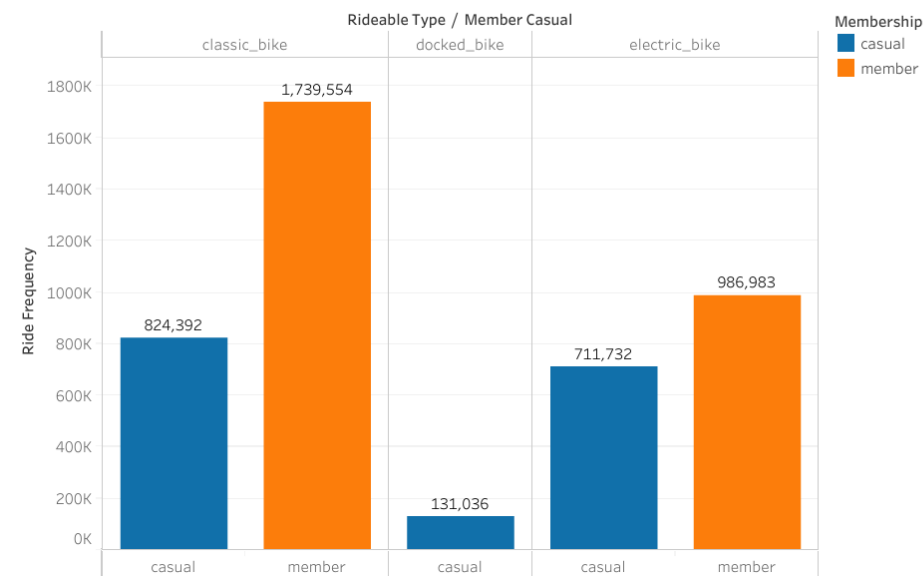


Figure 5: Ride Frequency by Rideable Type

Top 20 Start Stations by Casual Riders

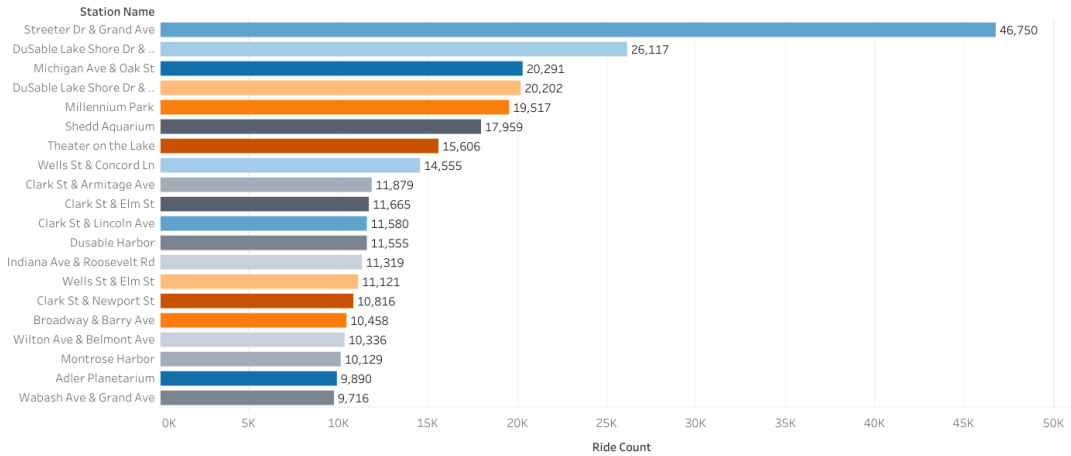


Figure 6: Top 20 Start Stations by Casual Riders

8 Act

8.1 Key findings

- The bikes were used by members more frequently than casual users, with a 24.12% higher usage rate in the last 12 months.
- However, casual riders had longer ride duration compared to members, with an average difference of 4-7 minutes.
- From a quarterly perspective, the third and second quarters of the year were the most popular seasons for casual riders.
- Casual riders exhibited a preference for riding bikes on Fridays, Saturdays, and Sundays, and they had longer trip duration during these days.
- Among the rideable types, classic bikes were the most popular choice among casual riders.
- Casual riders were the only user group that utilized docked bikes.
- “Streeter Dr & Grand Ave” and “DuSable Lake Shore Dr & Monroe St” were the most common starting stations for casual users.

8.2 Three recommendations

1. Create a marketing campaign that educates casual riders on their long-ride trips and invites them to a membership that can cater to their riding needs year-round. The campaign can be advertised both online and offline, with a focus on regions where the most popular stations for casual riders are located. This recommendation is supported by the data analysis, which highlights the long ride durations of casual riders and emphasizes the value of a membership that can provide continuous benefits.
2. Consider creating a specialty membership specifically designed for docked-bike users. This targeted approach can be an effective way to engage casual riders who prefer docked bikes. However, instead of offering a separate membership, it may be more beneficial to incorporate docked-bike usage as an additional feature within the existing full-scale membership. This way, casual riders can experience the benefits of membership without creating a distinct membership category.

3. Develop a special introductory membership for the months between April and September or for the days of the week (Friday to Sunday). This limited-time membership can serve as a trial period, offering casual riders the opportunity to experience the benefits of a membership during the most popular seasons or days. Following the introductory period, you can then transition these riders to a regular yearly membership.