Code ▾

Google Data Analytics Capstone Project: Cyclistic Bikeshare Case Study

Skills and tools used:

MS Excel: Skimming and getting familiar with the data, processing

R : Data processing , transformation, cleaning , Analysis, Visualization, Documentation

SQL (Postgres) : Extensive Analysis

Tableau : Visualization , Dashboard

MS PowerPoint : Making power point slides

INTRODUCTION & 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 future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations.”

This project will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

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.

Ask

Business Task

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

Key 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

Prepare

Data Located: Here is the data link https://divvy-tripdata.s3.amazonaws.com/index.html (https://divvy-tripdata.s3.amazonaws.com/index.html)

Data Organization: 12 months data used from Jan 2021 to Dec 2021 for the analysis.

Bias and Credibility : The data satisfies the ROCCC standard which means that it is reliable, original, comprehensive, current and cited.

Data Limitations: Some null valus were presented in the data , they were deleted.

Process Phase

The process phase check check for error , cleaning and transforming the data and to do all this I used RStudio as my tool. My initial task was to install the relevant packages and loading the data sets.

Here are the data cleaning and transforming steps in bullet point

- Install and loading packages
- Loading all 12 months data
- Combined all the data and make a single data frame
- Extracted new column week_day, month, day, year, date from started_at column.
- Created a new column as ride_length by subtracting ended_at, started_at
- Changed the data type of ride_length to numeric
- Deleted null values
- Deleted negative values
- Removed latitude and longitude type columns from main data frame and created a sub data frame using them for separate visualizations
- Created a new column ride_routes combining started_station_at and end ended_station_at column
- Sorted by month(January) and week name(Monday) of month and week_days column

Installing Packages

Hide

```
library(tidyverse)
```

```
── Attaching core tidyverse packages ──────────────────────────────────────────────────────
──────────────────── tidyverse 2.0.0 ──
✔ dplyr     1.1.0     ✔ readr     2.1.4
✔ forcats   1.0.0     ✔ stringr   1.5.0
✔ ggplot2   3.4.1     ✔ tibble    3.1.8
✔ lubridate 1.9.2     ✔ tidyr     1.3.0
✔ purrr     1.0.1     ── Conflicts ─────────────────────────────────────────────────────
────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the  ]8;;http://conflicted.r-lib.org/ conflicted package ]8;;  to force all conflict
s to become errors
```

Hide

```
library(lubridate)
```

Hide

```
install.packages("skimr")
```

```
trying URL 'https://cran.rstudio.com/bin/macosx/big-sur-arm64/contrib/4.2/skimr_2.1.5.tg
z'
Content type 'application/x-gzip' length 1221909 bytes (1.2 MB)
==================================================
downloaded 1.2 MB
```

```


The downloaded binary packages are in
    /var/folders/5h/_glh7qw931v_d44w4mck59b80000gn/T//RtmpxmjJxE/downloaded_packages
```

Hide

```
library(skimr)
```

```
Registered S3 methods overwritten by 'htmltools':
  method               from
  print.html           tools:rstudio
  print.shiny.tag      tools:rstudio
  print.shiny.tag.list tools:rstudio
```

Loading datasets

Data set loaded using the month name of the year 2021 from January to December

Hide

```
jan<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202101-divvy-tripdata.csv")
```

```
Rows: 96834 Columns: 13── Column specification ──────────────────────────────────────────
────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
feb<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202102-divvy-tripdata.csv")
```

```
Rows: 49622 Columns: 13── Column specification ──────────────────────────────────────────
────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
mar<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202103-divvy-tripdata.csv")
```

```
Rows: 228496 Columns: 13── Column specification ─────────────────────────────────────────
─────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
apr<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202104-divvy-tripdata.csv")
```

```
Rows: 337230 Columns: 13── Column specification ────────────────────────────────────────
──────────────────────────────────────────────────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
may<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202105-divvy-tripdata.csv")
```

```
Rows: 531633 Columns: 13── Column specification ────────────────────────────────────────
──────────────────────────────────────────────────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
june<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/dat
as/CSV/202106-divvy-tripdata.csv")
```

```
Rows: 729595 Columns: 13── Column specification ────────────────────────────────────────
──────────────────────────────────────────────────────────────
Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
july<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/dat
as/CSV/202107-divvy-tripdata.csv")
```

```
Rows: 822410 Columns: 13── Column specification ────────────────────────────────────────
──────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
aug<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202108-divvy-tripdata.csv")
```

```
Rows: 804352 Columns: 13── Column specification ────────────────────────────────────────
──────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
sep<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202109-divvy-tripdata.csv")
```

```
Rows: 756147 Columns: 13── Column specification ────────────────────────────────────────
──────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
oct<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202110-divvy-tripdata.csv")
```

```
Rows: 631226 Columns: 13── Column specification ────────────────────────────────────────
────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
nov<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202111-divvy-tripdata.csv")
```

```
Rows: 359978 Columns: 13── Column specification ────────────────────────────────────────
────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Hide

```
dec<- read_csv("/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/data
s/CSV/202112-divvy-tripdata.csv")
```

```
Rows: 247540 Columns: 13── Column specification ────────────────────────────────────────
────────────────────────────────────────────────────────────

Delimiter: ","
chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_station_nam
e, end_station_id, member_casual
dbl  (4): start_lat, start_lng, end_lat, end_lng
dttm (2): started_at, ended_at
ℹ Use `spec()` to retrieve the full column specification for this data.
ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Binding the data

Hide

```
trip_datas<-bind_rows(jan, feb, mar, apr, may, june, july, aug, sep, oct,nov, dec)
```

Showing few rows using head() function

Hide

```
head(trip_datas)
```

ride_id <chr>	rideable_type <chr>	started_at <S3: POSIXct>	ended_at <S3: POSIXct>	start_station_ <chr>
E19E6F1B8D4C42ED	electric_bike	2021-01-23 16:14:19	2021-01-23 16:24:44	California Ave &
DC88F20C2C55F27F	electric_bike	2021-01-27 18:43:08	2021-01-27 18:47:12	California Ave &
EC45C94683FE3F27	electric_bike	2021-01-21 22:35:54	2021-01-21 22:37:14	California Ave &
4FA453A75AE377DB	electric_bike	2021-01-07 13:31:13	2021-01-07 13:42:55	California Ave &
BE5E8EB4E7263A0B	electric_bike	2021-01-23 02:24:02	2021-01-23 02:24:45	California Ave &
5D8969F88C773979	electric_bike	2021-01-09 14:24:07	2021-01-09 15:17:54	California Ave &

6 rows | 1-5 of 13 columns

Using str() function we can check the overall structure of th data

Hide

```
str(trip_datas)
```

```
spc_tbl_ [5,595,063 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ ride_id           : chr [1:5595063] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683
FE3F27" "4FA453A75AE377DB" ...
 $ rideable_type     : chr [1:5595063] "electric_bike" "electric_bike" "electric_bike"
"electric_bike" ...
 $ started_at        : POSIXct[1:5595063], format: "2021-01-23 16:14:19" "2021-01-27 18:
43:08" "2021-01-21 22:35:54" "2021-01-07 13:31:13" ...
 $ ended_at          : POSIXct[1:5595063], format: "2021-01-23 16:24:44" "2021-01-27 18:
47:12" "2021-01-21 22:37:14" "2021-01-07 13:42:55" ...
 $ start_station_name: chr [1:5595063] "California Ave & Cortez St" "California Ave & Co
rtez St" "California Ave & Cortez St" "California Ave & Cortez St" ...
 $ start_station_id  : chr [1:5595063] "17660" "17660" "17660" "17660" ...
 $ end_station_name  : chr [1:5595063] NA NA NA NA ...
 $ end_station_id    : chr [1:5595063] NA NA NA NA ...
 $ start_lat         : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
 $ start_lng         : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
 $ end_lat           : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
 $ end_lng           : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
 $ member_casual     : chr [1:5595063] "member" "member" "member" "member" ...
 - attr(*, "spec")=
  .. cols(
  ..   ride_id = col_character(),
  ..   rideable_type = col_character(),
  ..   started_at = col_datetime(format = ""),
  ..   ended_at = col_datetime(format = ""),
  ..   start_station_name = col_character(),
  ..   start_station_id = col_character(),
  ..   end_station_name = col_character(),
  ..   end_station_id = col_character(),
  ..   start_lat = col_double(),
  ..   start_lng = col_double(),
  ..   end_lat = col_double(),
  ..   end_lng = col_double(),
  ..   member_casual = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

lets see the summary of the data set

Hide

```
skim_without_charts(trip_datas)
```

```
── Data Summary ────────────────────────
                           Values
Name                       trip_datas
Number of rows             5595063
Number of columns          13
_______________________
Column type frequency:
  character                7
  numeric                  4
  POSIXct                  2
________________________
Group variables            None
```

	skim_variable <chr>	n_missing <int>	complete_rate <dbl>	m.. <int>	m.. <int>	em... <int>	n_unique <int>	whitespace <int>
1	ride_id	0	1.0000000	16	16	0	5595063	0
2	rideable_type	0	1.0000000	11	13	0	3	0
3	start_station_name	690809	0.8765324	3	53	0	847	0
4	start_station_id	690806	0.8765329	3	36	0	834	0
5	end_station_name	739170	0.8678889	10	53	0	844	0
6	end_station_id	739170	0.8678889	3	36	0	832	0
7	member_casual	0	1.0000000	6	6	0	2	0

7 rows

	skim_variable <chr>	n_missing <int>	complete_rate <dbl>	mean <dbl>	sd <dbl>	p0 <dbl>	p25 <dbl>	p50 <dbl>
1	start_lat	0	1.0000000	41.90107	0.04608635	41.64	41.88155	41.89897
2	start_lng	0	1.0000000	-87.64605	0.02872111	-87.84	-87.66000	-87.64182
3	end_lat	4771	0.9991473	41.90134	0.04618916	41.39	41.88184	41.89964
4	end_lng	4771	0.9991473	-87.64626	0.02889959	-88.97	-87.66000	-87.64275

4 rows | 1-10 of 10 columns

	skim_variable <chr>	n_missing <int>	complete_rate <dbl>	min <S3: POSIXct>	max <S3: POSIXct>
1	started_at	0	1	2021-01-01 00:02:05	2021-12-31 23:59:48
2	ended_at	0	1	2021-01-01 00:08:39	2022-01-03 17:32:18

2 rows | 1-6 of 7 columns

Extracting month, day, year, week days from started at column to get better insights from the dataset

Hide

```
trip_datas$date <- as.Date(trip_datas$started_at)
trip_datas$month <- format(as.Date(trip_datas$date),"%B")
trip_datas$day <- format(as.Date(trip_datas$date),"%d")
trip_datas$year <- format(as.Date(trip_datas$date),"%Y")
trip_datas$week_day <- weekdays(trip_datas$date)
```

Now lets make another column to show the ride length of each trip to do that we will subtract strated_at column from ended_at column. And the units using as minutes

Hide

```
trip_datas$ride_length <- difftime(trip_datas$ended_at,trip_datas$started_at,units = "mi
n")
```

Now lets round the vaue to 2 decimal and make the column numeric using below function

Hide

```
trip_datas$ride_length <- round(trip_datas$ride_length,2)
trip_datas$ride_length <- as.numeric(as.character(trip_datas$ride_length))
str(trip_datas)
```

```
spc_tbl_ [5,595,063 × 19] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ ride_id           : chr [1:5595063] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683
FE3F27" "4FA453A75AE377DB" ...
 $ rideable_type     : chr [1:5595063] "electric_bike" "electric_bike" "electric_bike"
"electric_bike" ...
 $ started_at        : POSIXct[1:5595063], format: "2021-01-23 16:14:19" "2021-01-27 18:
43:08" "2021-01-21 22:35:54" "2021-01-07 13:31:13" ...
 $ ended_at          : POSIXct[1:5595063], format: "2021-01-23 16:24:44" "2021-01-27 18:
47:12" "2021-01-21 22:37:14" "2021-01-07 13:42:55" ...
 $ start_station_name: chr [1:5595063] "California Ave & Cortez St" "California Ave & Co
rtez St" "California Ave & Cortez St" "California Ave & Cortez St" ...
 $ start_station_id  : chr [1:5595063] "17660" "17660" "17660" "17660" ...
 $ end_station_name  : chr [1:5595063] NA NA NA NA ...
 $ end_station_id    : chr [1:5595063] NA NA NA NA ...
 $ start_lat         : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
 $ start_lng         : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
 $ end_lat           : num [1:5595063] 41.9 41.9 41.9 41.9 41.9 ...
 $ end_lng           : num [1:5595063] -87.7 -87.7 -87.7 -87.7 -87.7 ...
 $ member_casual     : chr [1:5595063] "member" "member" "member" "member" ...
 $ date              : Date[1:5595063], format: "2021-01-23" "2021-01-27" "2021-01-21"
"2021-01-07" ...
 $ month             : chr [1:5595063] "January" "January" "January" "January" ...
 $ day               : chr [1:5595063] "23" "27" "21" "07" ...
 $ year              : chr [1:5595063] "2021" "2021" "2021" "2021" ...
 $ week_day          : chr [1:5595063] "Saturday" "Wednesday" "Thursday" "Thursday" ...
 $ ride_length       : num [1:5595063] 10.42 4.07 1.33 11.7 0.72 ...
 - attr(*, "spec")=
  .. cols(
  ..   ride_id = col_character(),
  ..   rideable_type = col_character(),
  ..   started_at = col_datetime(format = ""),
  ..   ended_at = col_datetime(format = ""),
  ..   start_station_name = col_character(),
  ..   start_station_id = col_character(),
  ..   end_station_name = col_character(),
  ..   end_station_id = col_character(),
  ..   start_lat = col_double(),
  ..   start_lng = col_double(),
  ..   end_lat = col_double(),
  ..   end_lng = col_double(),
  ..   member_casual = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

Cleaning the data

For cleaning data lets frop all the null values first

Hide

```
trip_datas <- drop_na(trip_datas)
dim(trip_datas)
```

```
[1] 4588302      19
```

Now lets see if we have any negative value in our ride length column

Hide

```
filter(trip_datas,ride_length<0)
```

ride_id	rideable_type	started_at	ended_at ▶
<chr>	<chr>	<S3: POSIXct>	<S3: POSIXct>
F79335E3A77A57B5	electric_bike	2021-03-29 15:41:21	2021-03-29 15:41:20
5D2797A8FFA71B49	classic_bike	2021-03-13 18:02:58	2021-03-13 18:02:57
BC53ECCBC76278FD	classic_bike	2021-04-07 16:11:33	2021-04-07 16:11:26
6E81034B446FC2FD	electric_bike	2021-04-23 09:43:39	2021-04-23 09:43:29
318DD838369AEA61	classic_bike	2021-04-30 10:56:32	2021-04-30 10:56:30
8ADD13BD8F6A7567	classic_bike	2021-04-17 12:43:36	2021-04-17 12:43:27
3EC1B5A4D4B9AB99	classic_bike	2021-05-05 16:10:04	2021-05-05 16:09:51
732D84DAD2CC9B73	classic_bike	2021-06-20 10:52:26	2021-06-20 10:52:25
A18D39992AA99793	classic_bike	2021-06-15 20:58:03	2021-06-15 20:54:51
24C4FC421D642C22	classic_bike	2021-06-28 13:18:26	2021-06-28 13:18:25

1-10 of 116 rows | 1-4 of 19 columns   Previous **1** 2 3 4 5 6 ... 12 Next

We have 116 rows of negative ride length, lets remove those and make a new data frame where ride length are > 0

Hide

```
trip_datas<- filter(trip_datas, ride_length>0)
```

Now lets make a separate data frame as lat_long to make visualization of this later after that extract lat and long columns from trip_datas

Hide

```
lat_lng1 <- select(trip_datas,start_station_name,end_station_name,start_lat,start_lng,en
d_lat,end_lng,member_casual)

# Extracting latitude and longitude from trip_datas
trip_datas <- trip_datas %>%
  select (-c(start_lat, start_lng, end_lat, end_lng))
```

Analysis

In the analysis phase the goal is to identify the relationship and trends in data which will help to solve the business task. Lets explore further.

Hide

```
# Percentage of annual member and casual member
count_members <- table(trip_datas$member_casual)
sum(count_members)
```

```
[1] 4588104
```

Hide

```
percentage <- 100*count_members/sum(count_members)
percentage
```

```


  casual   member 
44.64376 55.35624 
```

There are 44.6% Casual member and 55.4% Annual member

Types of bikes

Hide

```
# Type of bikes
unique(trip_datas$rideable_type)
```

```
[1] "classic_bike"  "electric_bike" "docked_bike"
```

3 types of bike

Hide

```
# types of bike riden by each member type
trip_datas %>% group_by(member_casual) %>% count(rideable_type,sort=FALSE)
```

member_casual	**rideable_type**	**n**
<chr>	<chr>	<int>
casual	classic_bike	1261508
casual	docked_bike	312040
casual	electric_bike	474754
member	classic_bike	1980328
member	docked_bike	1

member_casual	rideable_type	n
<chr>	<chr>	<int>
member	electric_bike	559473

6 rows

For both Casual and member most favorite is classic_bike and also we can see that for annual member they don’t like docked bike that much.

Let’s run few calculation to find out max, mean , min ride length for both member type

Hide

```
#mean
trip_datas %>% group_by(member_casual) %>% summarise(rides_number=n(),mean(ride_length))
```

member_casual	rides_number	mean(ride_length)
<chr>	<int>	<dbl>
casual	2048302	32.51015
member	2539802	13.18401

2 rows

Hide

```
#max
trip_datas %>% group_by(member_casual) %>% summarise(rides_number=n(),max(ride_length))
```

member_casual	rides_number	max(ride_length)
<chr>	<int>	<dbl>
casual	2048302	55944.15
member	2539802	1495.63

2 rows

Hide

```
#min
trip_datas %>% group_by(member_casual) %>% summarise(rides_number=n(),min(ride_length))
```

member_casual	rides_number	min(ride_length)
<chr>	<int>	<dbl>
casual	2048302	0.02
member	2539802	0.02

2 rows

Hide

```
# summary statistics of ride length across members
summary(trip_datas$ride_length)
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
    0.02     6.95    12.20    21.81    22.12 55944.15 
```

As our business task is mainly focus on converting the Casual members to Annual member lets do our further analysis focused on Casual members

But first lets sort out the month and week column from January to December and Monday to Sunday

Hide

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

#week
trip_datas$week_day <- ordered(trip_datas$week_day, levels=
                                 c("Monday","Tuesday","Wednesday","Thursday","Friday","S
aturday","Sunday"))
```

Stations

Hide

```
#most used start stations by casual riders
most_used_stations <- trip_datas %>%
  group_by(start_station_name,member_casual) %>%
  summarise(number_count=n()) %>%
  arrange(desc(number_count)) %>%
  filter(member_casual== "casual") %>%
  select(start_station_name,number_count)
```

```
`summarise()` has grouped output by 'start_station_name'. You can override using the `.g
roups` argument.
```

Hide

```
most_used_stations
```

start_station_name <chr>	number_count <int>
Streeter Dr & Grand Ave	64446
Millennium Park	32185
Michigan Ave & Oak St	28661
Shedd Aquarium	22544
Theater on the Lake	20620
Lake Shore Dr & Monroe St	19085
Wells St & Concord Ln	18879
Clark St & Lincoln Ave	16290
Indiana Ave & Roosevelt Rd	15814
Wells St & Elm St	15809

1-10 of 835 rows Previous **1** 2 3 4 5 6 ... 84 Next

Hide

```
#least used start stations casual riders
least_used_start_stations <- trip_datas %>%
  group_by(start_station_name,member_casual) %>%
  summarise(number_count=n()) %>%
  arrange(number_count) %>%
  filter(member_casual== "casual") %>%
  select(start_station_name,number_count)
```

```
`summarise()` has grouped output by 'start_station_name'. You can override using the `.g
roups` argument.
```

Hide

```
least_used_start_stations
```

start_station_name <chr>	number_count <int>
351	1
Francisco Ave & Hollywood Ave	1
Indiana Ave & 133rd St	1
Lincoln Ave & Peterson Ave	1
Olive Harvey College	1
Washtenaw Ave & Madison St	1

start_station_name <chr>	number_count <int>
California Ave & 29th St	2
Campbell Ave & 51st St	2
Francisco Ave & Bloomingdale Ave	2
Halsted St & 63rd St	2

1-10 of 835 rows Previous **1** 2 3 4 5 6 ... 84 Next

Hide

```
#most used end stations by casual riders
most_used_end_stations <- trip_datas %>%
  group_by(end_station_name,member_casual) %>%
  summarise(number_count=n()) %>%
  arrange(desc(number_count)) %>%
  filter(member_casual== "casual") %>%
  select(end_station_name,number_count)
```

```
`summarise()` has grouped output by 'end_station_name'. You can override using the `.gro
ups` argument.
```

Hide

```
most_used_end_stations
```

end_station_name <chr>	number_count <int>
Streeter Dr & Grand Ave	67524
Millennium Park	33744
Michigan Ave & Oak St	30364
Theater on the Lake	22307
Shedd Aquarium	21158
Wells St & Concord Ln	18618
Lake Shore Dr & Monroe St	17932
Lake Shore Dr & North Blvd	17072
DuSable Lake Shore Dr & North Blvd	16653
Wabash Ave & Grand Ave	16575

1-10 of 834 rows Previous **1** 2 3 4 5 6 ... 84 Next

Hide

```
#least used end stations by casual riders
least_used_end_stations <- trip_datas %>%
  group_by(end_station_name,member_casual) %>%
  summarise(number_count=n()) %>%
  arrange((number_count)) %>%
  filter(member_casual== "casual") %>%
  select(end_station_name,number_count)
```

```
`summarise()` has grouped output by 'end_station_name'. You can override using the `.gro
ups` argument.
```

Hide

```
least_used_end_stations
```

end_station_name <chr>	number_count <int>
California Ave & 29th St	1
Francisco Ave & 47th St	1
Indiana Ave & 133rd St	1
N Hampden Ct & W Diversey Ave	1
Pulaski & Ann Lurie Pl	1
Washtenaw Ave & Polk St	1
Whipple St & Irving Park Rd	1
Augusta Blvd & Laramie Ave	2
California Ave & Marquette Rd	2
Fairfield Ave & 44th St	2

1-10 of 834 rows Previous **1** 2 3 4 5 6 ... 84 Next

By Month

Hide

```
#mostly used start stations by each month by casual riders
most_used_start_stations_by_month <- trip_datas %>%
  group_by(member_casual,month,start_station_name) %>%
  summarise(number_count=n()) %>%
  arrange(desc(number_count)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
most_used_start_stations_by_month
```

member_casual <chr>	month <ord>	start_station_name <chr>	number_count <int>
casual	July	Streeter Dr & Grand Ave	13911
casual	August	Streeter Dr & Grand Ave	11350
casual	June	Streeter Dr & Grand Ave	11185
casual	September	Streeter Dr & Grand Ave	8998
casual	May	Streeter Dr & Grand Ave	8171
casual	July	Millennium Park	6104
casual	July	Michigan Ave & Oak St	5838
casual	August	DuSable Lake Shore Dr & North Blvd	5768
casual	August	Michigan Ave & Oak St	5526
casual	June	Lake Shore Dr & North Blvd	5331

1-10 of 8,301 rows Previous **1** 2 3 4 5 6 ... 100 Next

Hide

```
#mostly used end stations by each month by casual riders
most_used_end_stations_by_month <- trip_datas %>%
  group_by(member_casual,month,end_station_name) %>%
  summarise(number_count=n()) %>%
  arrange(desc(number_count)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
most_used_end_stations_by_month
```

member_casual <chr>	month <ord>	end_station_name <chr>	number_count <int>
casual	July	Streeter Dr & Grand Ave	14488
casual	August	Streeter Dr & Grand Ave	11604
casual	June	Streeter Dr & Grand Ave	11542
casual	September	Streeter Dr & Grand Ave	9197
casual	May	Streeter Dr & Grand Ave	8936
casual	August	DuSable Lake Shore Dr & North Blvd	6915
casual	June	Lake Shore Dr & North Blvd	6740
casual	July	Millennium Park	6289
casual	July	Michigan Ave & Oak St	6110
casual	August	Michigan Ave & Oak St	5715

1-10 of 8,301 rows Previous **1** 2 3 4 5 6 ... 100 Next

Streeter Dr & Grand Ave is the most popular station(start & end) in summer season.

Hide

```
#mostly rided months by casual riders
most_rided_months_by_casual <- trip_datas %>%
  group_by(member_casual,month) %>%
  summarise(number_count=n()) %>%
  arrange(desc(number_count)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
most_rided_months_by_casual
```

member_casual <chr>	month <ord>	number_count <int>
casual	July	369401
casual	August	341463
casual	June	304184

member_casual	month	number_count
<chr>	<ord>	<int>
casual	September	292920
casual	May	216823
casual	October	189115
casual	April	120418
casual	March	75641
casual	November	69958
casual	December	45076

1-10 of 12 rows Previous **1** 2 Next

Hide

```
NA
```

From April the no of rides increases where July takes the peak count. On the other hand November takes significantly drop and continues till March where February has the lowest count.

Hide

```
# mean ride length by casual riders each months
mean_ride_length_by_month <- trip_datas %>%
  group_by(member_casual,month) %>%
  summarise(number_count=n(),mean=mean(ride_length)) %>%
  arrange(desc(mean)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
mean_ride_length_by_month
```

member_casual	month	number_count	mean
<chr>	<ord>	<int>	<dbl>
casual	February	8613	47.13714
casual	May	216823	39.63541
casual	June	304184	38.52030
casual	March	75641	38.48047

member_casual <chr>	month <ord>	number_count <int>	mean <dbl>
casual	April	120418	38.44172
casual	July	369401	33.28581
casual	August	341463	28.56455
casual	September	292920	28.08850
casual	January	14690	26.36998
casual	October	189115	26.31429

1-10 of 12 rows Previous **1** 2 Next

February has the highest average of ride length because of low number of rides

By Week

Hide

```
#mostly rided week days by casual riders
most_rided_weekDays_by_casual <- trip_datas %>%
  group_by(member_casual,week_day) %>%
  summarise(number_count=n()) %>%
  arrange(desc(number_count)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
most_rided_weekDays_by_casual
```

member_casual <chr>	week_day <ord>	number_count <int>
casual	Saturday	468318
casual	Sunday	403754
casual	Friday	290034
casual	Monday	228931
casual	Thursday	224204
casual	Wednesday	218129
casual	Tuesday	214932

7 rows

Hide

```
# mean ride length by casual riders each week
mean_ride_length_by_week <- trip_datas %>%
  group_by(member_casual,week_day) %>%
  summarise(number_count=n(),mean=mean(ride_length)) %>%
  arrange(desc(mean)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
mean_ride_length_by_week
```

member_casual <chr>	week_day <ord>	number_count <int>	mean <dbl>
casual	Sunday	403754	37.60360
casual	Saturday	468318	34.87240
casual	Monday	228931	32.63724
casual	Friday	290034	30.91907
casual	Tuesday	214932	28.80336
casual	Wednesday	218129	28.26963
casual	Thursday	224204	28.01100

7 rows

Saturday and Sunday has the highest number of ride count and mean ride length count by the casual members. Reason for this is holiday. It could be a good target marketing for this two days.

Mean peak hours by week days

Hide

```
#Mean peak hours of week days by casual rider
peak_hour <- trip_datas %>%
  group_by(week_day,member_casual) %>% summarize(mean(started_at),mean(ended_at)) %>% fi
lter(member_casual=="casual")
```

```
`summarise()` has grouped output by 'week_day'. You can override using the `.groups` arg
ument.
```

Hide

```
peak_hour
```

week_day <ord>	member_casual <chr>	mean(started_at) <S3: POSIXct>	mean(ended_at) <S3: POSIXct>
Monday	casual	2021-07-21 15:20:44	2021-07-21 15:53:22
Tuesday	casual	2021-07-21 11:52:24	2021-07-21 12:21:12
Wednesday	casual	2021-07-26 17:32:02	2021-07-26 18:00:19
Thursday	casual	2021-07-29 00:50:53	2021-07-29 01:18:53
Friday	casual	2021-07-25 05:03:17	2021-07-25 05:34:12
Saturday	casual	2021-07-21 17:16:34	2021-07-21 17:51:26
Sunday	casual	2021-07-21 01:20:38	2021-07-21 01:58:15

7 rows

Now lets combined start and end stations name to make route and save it as trip_datas_v2

Hide

```
trip_datas_v2 <- (unite(trip_datas, "ride_routes", start_station_name, end_station_name,
sep= " to "))
```

Top 10

Hide

```
# top_10 routes
top_10_routes <- trip_datas_v2 %>%
  group_by(ride_routes) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>% head(10)
top_10_routes
```

ride_routes <chr>	count <int>
Streeter Dr & Grand Ave to Streeter Dr & Grand Ave	13027
Michigan Ave & Oak St to Michigan Ave & Oak St	6674
Millennium Park to Millennium Park	6470
Lake Shore Dr & Monroe St to Lake Shore Dr & Monroe St	5132
Ellis Ave & 60th St to Ellis Ave & 55th St	5098
Ellis Ave & 55th St to Ellis Ave & 60th St	4613

ride_routes <chr>	count <int>
Theater on the Lake to Theater on the Lake	4020
Montrose Harbor to Montrose Harbor	3776
Ellis Ave & 60th St to University Ave & 57th St	3685
Buckingham Fountain to Buckingham Fountain	3625

1-10 of 10 rows

Hide

```
# top_10 routes by number of rides by casual member
top_10_routes_casual <- trip_datas_v2 %>%
  group_by(ride_routes,member_casual) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual")
```

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

Hide

```
head(top_10_routes_casual,10)
```

ride_routes <chr>	member_c <chr>
Streeter Dr & Grand Ave to Streeter Dr & Grand Ave	casual
Millennium Park to Millennium Park	casual
Michigan Ave & Oak St to Michigan Ave & Oak St	casual
Lake Shore Dr & Monroe St to Lake Shore Dr & Monroe St	casual
Buckingham Fountain to Buckingham Fountain	casual
Streeter Dr & Grand Ave to Millennium Park	casual
Theater on the Lake to Theater on the Lake	casual
DuSable Lake Shore Dr & Monroe St to DuSable Lake Shore Dr & Monroe St	casual
Montrose Harbor to Montrose Harbor	casual
Shedd Aquarium to Shedd Aquarium	casual

1-10 of 10 rows

As we see that saturday and sunday has highest ride count by casual member lets see their top 10 route this two days

Hide

```
# top_10 routes by number of rides by casual member On Saturday
top_10_routes_casual_saturday <- trip_datas_v2 %>%
  group_by(ride_routes,member_casual,week_day) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual", week_day=="Saturday")
```

```
`summarise()` has grouped output by 'ride_routes', 'member_casual'. You can override usi
ng the `.groups` argument.
```

Hide

```
head(top_10_routes_casual_saturday,10)
```

ride_routes	member_c
<chr>	<chr>
Streeter Dr & Grand Ave to Streeter Dr & Grand Ave	casual
Millennium Park to Millennium Park	casual
Michigan Ave & Oak St to Michigan Ave & Oak St	casual
Lake Shore Dr & Monroe St to Lake Shore Dr & Monroe St	casual
DuSable Lake Shore Dr & Monroe St to DuSable Lake Shore Dr & Monroe St	casual
Streeter Dr & Grand Ave to Millennium Park	casual
Millennium Park to Streeter Dr & Grand Ave	casual
DuSable Lake Shore Dr & Monroe St to Streeter Dr & Grand Ave	casual
Theater on the Lake to Theater on the Lake	casual
Lake Shore Dr & Monroe St to Streeter Dr & Grand Ave	casual

1-10 of 10 rows | 1-3 of 4 columns

Hide

```
# top_10 routes by number of rides by casual member on Sunday
top_10_routes_casual_sunday <- trip_datas_v2 %>%
  group_by(ride_routes,member_casual,week_day) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual", week_day=="Sunday")
```

```
`summarise()` has grouped output by 'ride_routes', 'member_casual'. You can override usi
ng the `.groups` argument.
```

Hide

```
head(top_10_routes_casual_sunday,10)
```

ride_routes <chr>	member_c <chr>
Streeter Dr & Grand Ave to Streeter Dr & Grand Ave	casual
Millennium Park to Millennium Park	casual
Michigan Ave & Oak St to Michigan Ave & Oak St	casual
Lake Shore Dr & Monroe St to Lake Shore Dr & Monroe St	casual
Buckingham Fountain to Buckingham Fountain	casual
Theater on the Lake to Theater on the Lake	casual
Montrose Harbor to Montrose Harbor	casual
Streeter Dr & Grand Ave to Millennium Park	casual
DuSable Lake Shore Dr & Monroe St to DuSable Lake Shore Dr & Monroe St	casual
Fort Dearborn Dr & 31st St to Fort Dearborn Dr & 31st St	casual

1-10 of 10 rows | 1-3 of 4 columns

Top 10 start and End station Saturday and Sunday by casual member

Saturday

Hide

```
#top 10 start station saturday
top_10_start_station_casual_saturday <- trip_datas %>%
  group_by(start_station_name,member_casual,week_day) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual", week_day=="Saturday")
```

```
`summarise()` has grouped output by 'start_station_name', 'member_casual'. You can overr
ide using the `.groups` argument.
```

Hide

```
head(top_10_start_station_casual_saturday,10)
```

start_station_name <chr>	member_casual <chr>	week_day <ord>	count <int>
Streeter Dr & Grand Ave	casual	Saturday	17039
Millennium Park	casual	Saturday	8021
Michigan Ave & Oak St	casual	Saturday	7450
Theater on the Lake	casual	Saturday	5370
Shedd Aquarium	casual	Saturday	5366
Lake Shore Dr & Monroe St	casual	Saturday	5026
Clark St & Lincoln Ave	casual	Saturday	4879
Wells St & Concord Ln	casual	Saturday	4862
DuSable Lake Shore Dr & Monroe St	casual	Saturday	4557
Clark St & Armitage Ave	casual	Saturday	4188

1-10 of 10 rows

Hide

```
# top 10 end station saturday
top_10_end_station_casual_saturday <- trip_datas %>%
  group_by(end_station_name,member_casual,week_day) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual", week_day=="Saturday")
```

```
`summarise()` has grouped output by 'end_station_name', 'member_casual'. You can overrid
e using the `.groups` argument.
```

Hide

```
head(top_10_end_station_casual_saturday,10)
```

end_station_name <chr>	member_casual <chr>	week_day <ord>	count <int>
Streeter Dr & Grand Ave	casual	Saturday	17677
Millennium Park	casual	Saturday	8455

end_station_name <chr>	member_casual <chr>	week_day <ord>	count <int>
Michigan Ave & Oak St	casual	Saturday	7882
Theater on the Lake	casual	Saturday	5822
Shedd Aquarium	casual	Saturday	5089
Wells St & Concord Ln	casual	Saturday	4938
Clark St & Lincoln Ave	casual	Saturday	4924
Lake Shore Dr & Monroe St	casual	Saturday	4833
DuSable Lake Shore Dr & Monroe St	casual	Saturday	4336
Clark St & Armitage Ave	casual	Saturday	4102

1-10 of 10 rows

Sunday

Hide

```
#top 10 start station Sunday
top_10_start_station_casual_sunday <- trip_datas %>%
  group_by(start_station_name,member_casual,week_day) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual", week_day=="Sunday")
```

```
`summarise()` has grouped output by 'start_station_name', 'member_casual'. You can overr
ide using the `.groups` argument.
```

Hide

```
head(top_10_start_station_casual_sunday,10)
```

start_station_name <chr>	member_casual <chr>	week_day <ord>	count <int>
Streeter Dr & Grand Ave	casual	Sunday	14678
Millennium Park	casual	Sunday	7492
Michigan Ave & Oak St	casual	Sunday	6797
Theater on the Lake	casual	Sunday	5203
Shedd Aquarium	casual	Sunday	4618
Lake Shore Dr & Monroe St	casual	Sunday	4502
Wells St & Concord Ln	casual	Sunday	3648

start_station_name <chr>	member_casual <chr>	week_day <ord>	count <int>
Lake Shore Dr & North Blvd	casual	Sunday	3550
DuSable Lake Shore Dr & Monroe St	casual	Sunday	3509
Indiana Ave & Roosevelt Rd	casual	Sunday	3468

1-10 of 10 rows

Hide

```
# top 10 end station sunday
top_10_end_station_casual_sunday <- trip_datas %>%
  group_by(end_station_name,member_casual,week_day) %>%
  summarise(count=n()) %>%
  arrange(desc(count)) %>%
  filter(member_casual== "casual", week_day=="Sunday")
```

```
`summarise()` has grouped output by 'end_station_name', 'member_casual'. You can overrid
e using the `.groups` argument.
```

Hide

```
head(top_10_end_station_casual_sunday,10)
```

end_station_name <chr>	member_casual <chr>	week_day <ord>	count <int>
Streeter Dr & Grand Ave	casual	Sunday	15444
Millennium Park	casual	Sunday	7584
Michigan Ave & Oak St	casual	Sunday	6905
Theater on the Lake	casual	Sunday	5633
Lake Shore Dr & Monroe St	casual	Sunday	4499
Shedd Aquarium	casual	Sunday	4461
Lake Shore Dr & North Blvd	casual	Sunday	3749
Wabash Ave & Grand Ave	casual	Sunday	3610
Wells St & Concord Ln	casual	Sunday	3557
Indiana Ave & Roosevelt Rd	casual	Sunday	3478

1-10 of 10 rows

Key Finding after analysis

- There are 44.6% Casual member and 55.4% Annual member

- 3 types of bike “classic_bike” “electric_bike” “docked_bike”
- For both Casual and member most favorite is classic_bike
- Min. 1st Qu. Median Mean 3rd Qu. Max. 0.02 6.95 12.20 21.81 22.12 55944.15

Now Focusing on Casual riders as our goal is to convert casual riders to annual member

- Streeter Dr & Grand Ave is the most popular station(start & end) in summer season for casual rider
- From May the no of rides increases where July takes the peak count. On the other hand November takes significantly drop and continues till March where February has the lowest count.
- February has the highest average of ride length because of low number of rides
- Saturday and Sunday has the highest number of ride count and mean ride length count by the casual members. Reason for this is holiday. It could be a good target marketing for this two days.

Hide

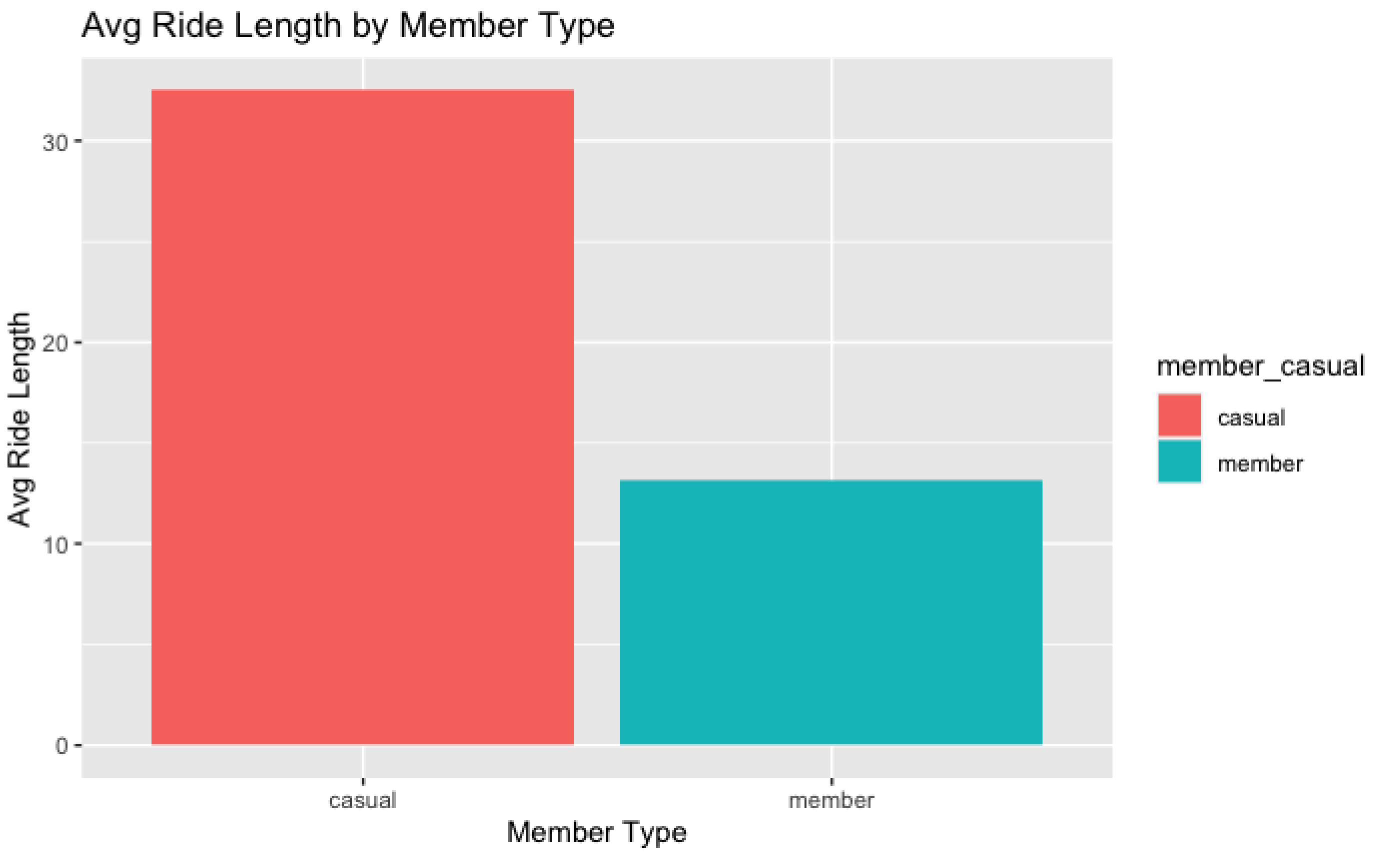
```
library(scales)
```

```


Attaching package: ‘scales’

The following object is masked from ‘package:purrr’:

    discard

The following object is masked from ‘package:readr’:

    col_factor
```

Visualization

Hide

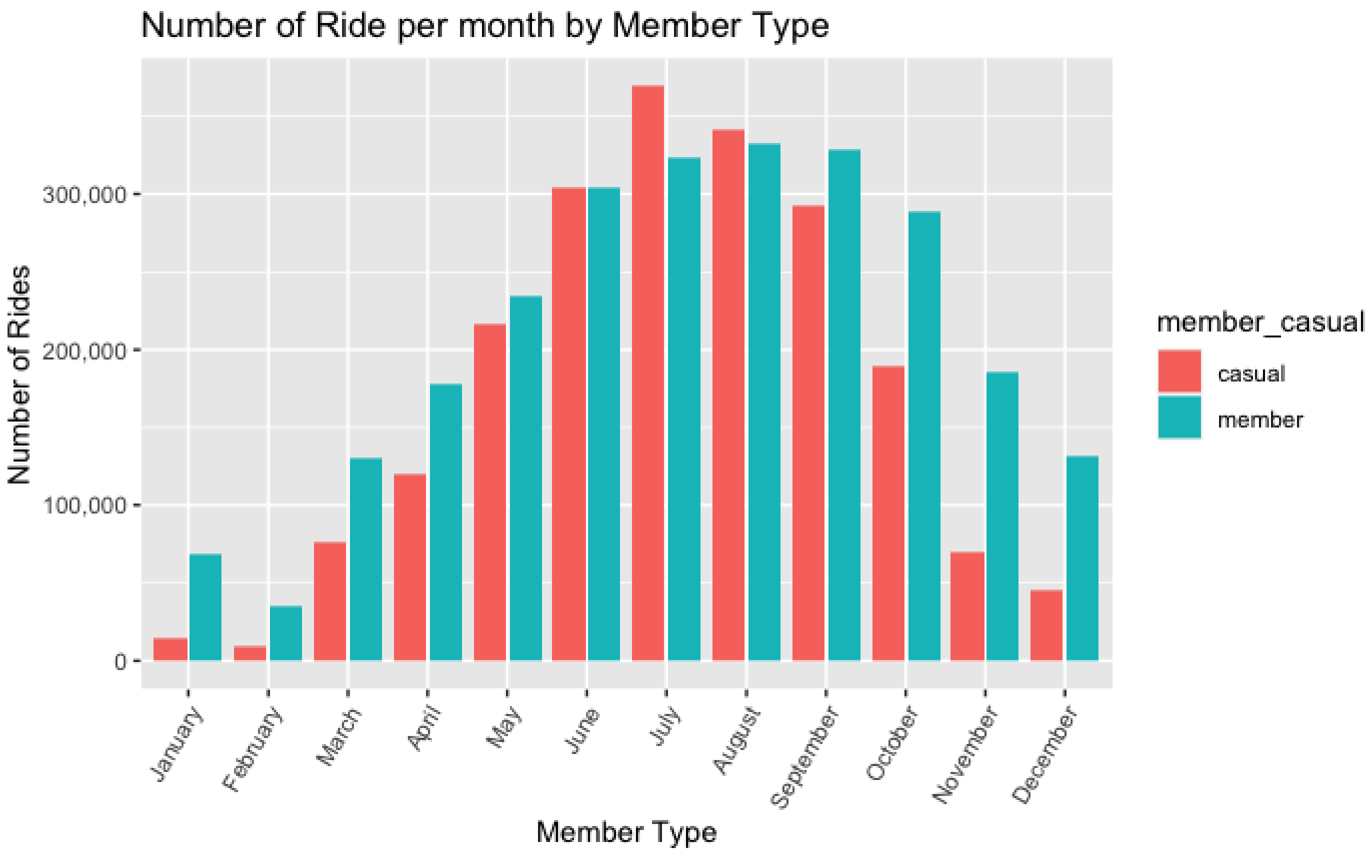
```
#Avg Ride Length by Member Type
trip_datas_v2 %>% group_by(member_casual) %>%
  summarise(avg_ride_length=mean(ride_length)) %>%
              ggplot(aes(x=member_casual, y= avg_ride_length, fill=member_casual)) +
              geom_col() +
              labs(title = "Avg Ride Length by Member Type", x="Member Type", y="Avg Rid
e Length")
```



Hide

```
#Number of rides per month by Member type
trip_datas_v2 %>% group_by(member_casual,month) %>%
  summarise(ride_count=n()) %>%
  ggplot(aes(x=month, y= ride_count, fill=member_casual)) +
  geom_col(position= "dodge2") +
  labs(title = "Number of Ride per month by Member Type", x="Member Type", y="Number of
Rides")+scale_y_continuous(labels = comma) + theme(axis.text.x = element_text(angle = 6
0, hjust=1))
```

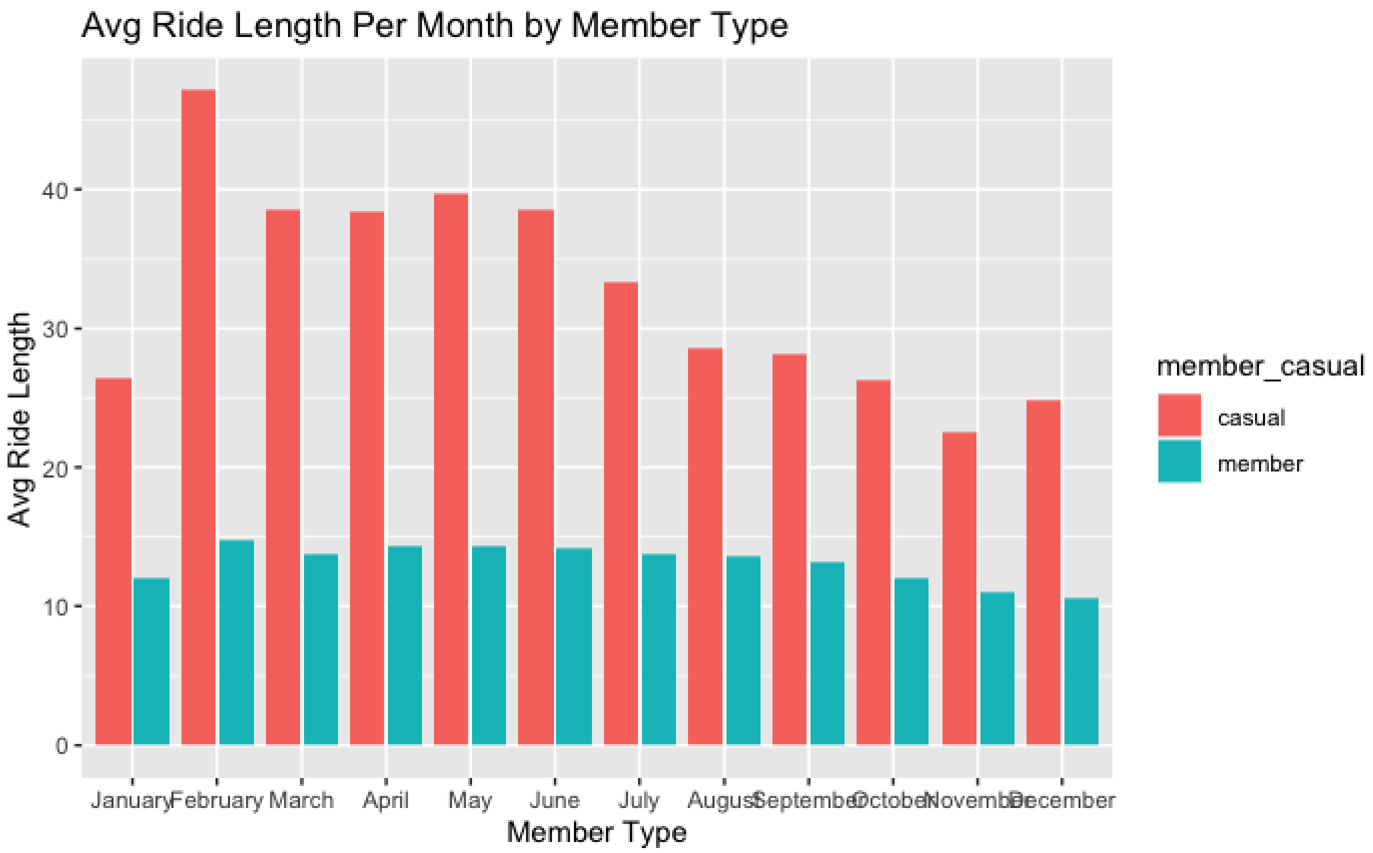
```
`summarise()` has grouped output by 'member_casual'. You can override using the `.groups
` argument.
```



Hide

```
# Avg ride length per month by customer type
trip_datas_v2 %>% group_by(member_casual,month) %>%
  summarise(avg_ride_length= mean(ride_length)) %>%
  ggplot(aes(x=month, y= avg_ride_length, fill=member_casual)) +
  geom_col(position= "dodge2") +
  labs(title = "Avg Ride Length Per Month by Member Type", x="Member Type", y="Avg Ride
Length")
```

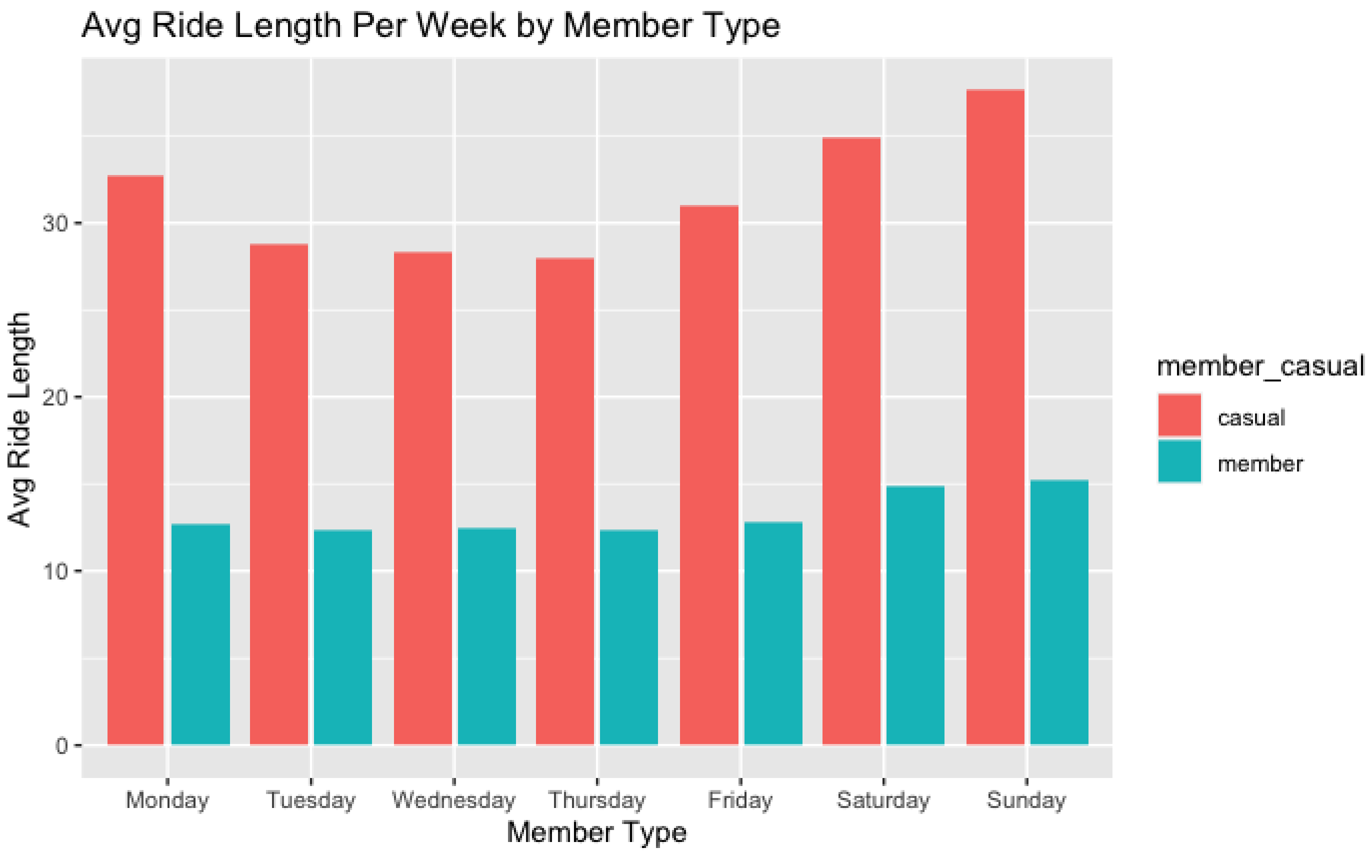
```
`summarise()` has grouped output by 'member_casual'. You can override using the `.groups
` argument.
```



Hide

```
# Avg ride length per Week by customer type
trip_datas_v2 %>% group_by(member_casual,week_day) %>%
  summarise(avg_ride_length= mean(ride_length)) %>%
  ggplot(aes(x=week_day, y= avg_ride_length, fill=member_casual)) +
  geom_col(position= "dodge2") +
  labs(title = "Avg Ride Length Per Week by Member Type", x="Member Type", y="Avg Ride L
ength")
```

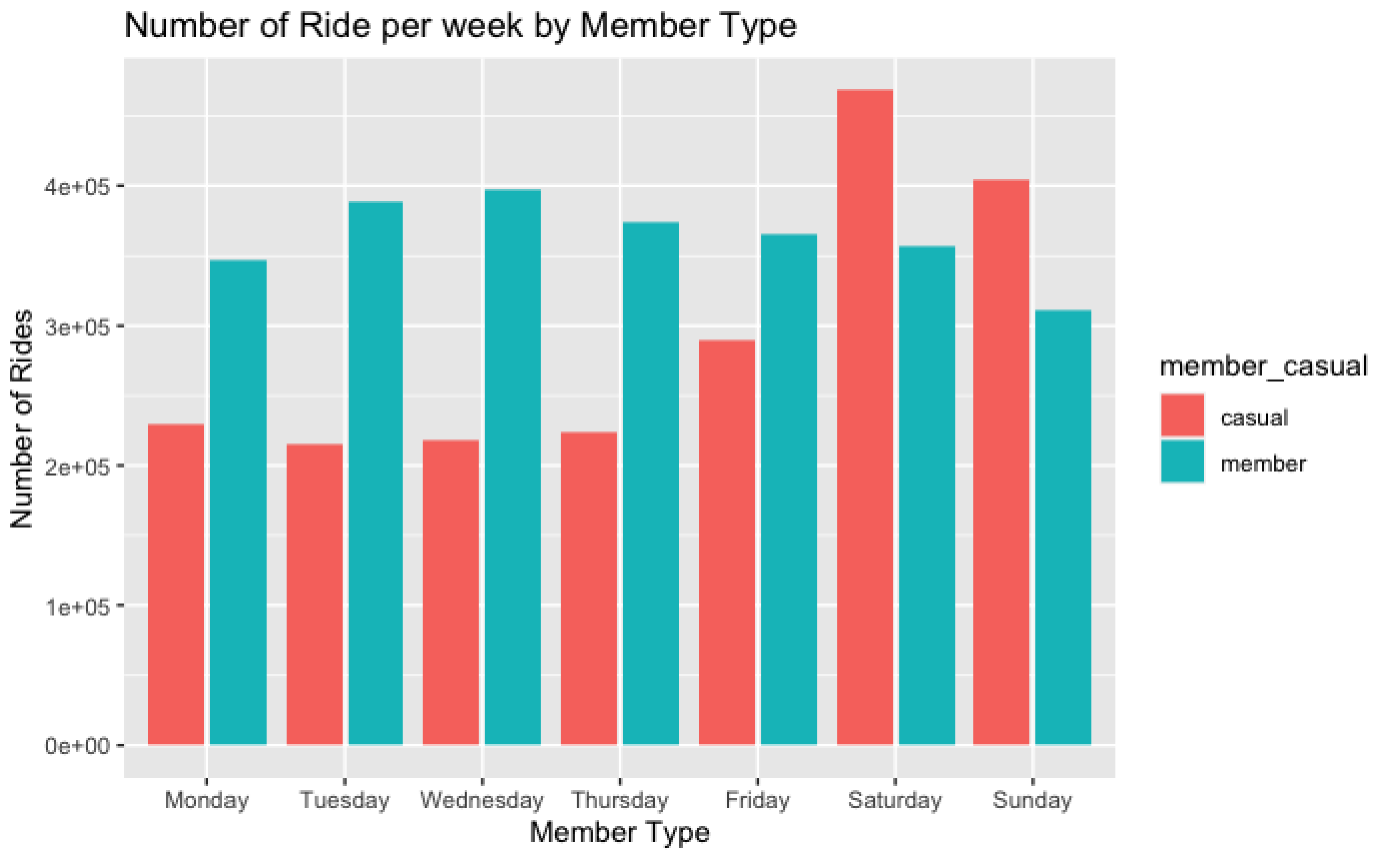
```
`summarise()` has grouped output by 'member_casual'. You can override using the `.groups
` argument.
```



Hide

```
#Number of rides per week by Member type
trip_datas_v2 %>% group_by(member_casual,week_day) %>%
  summarise(ride_count=n()) %>%
  ggplot(aes(x=week_day, y= ride_count, fill=member_casual)) +
  geom_col(position= "dodge2") +
  labs(title = "Number of Ride per week by Member Type", x="Member Type", y="Number of R
ides")
```

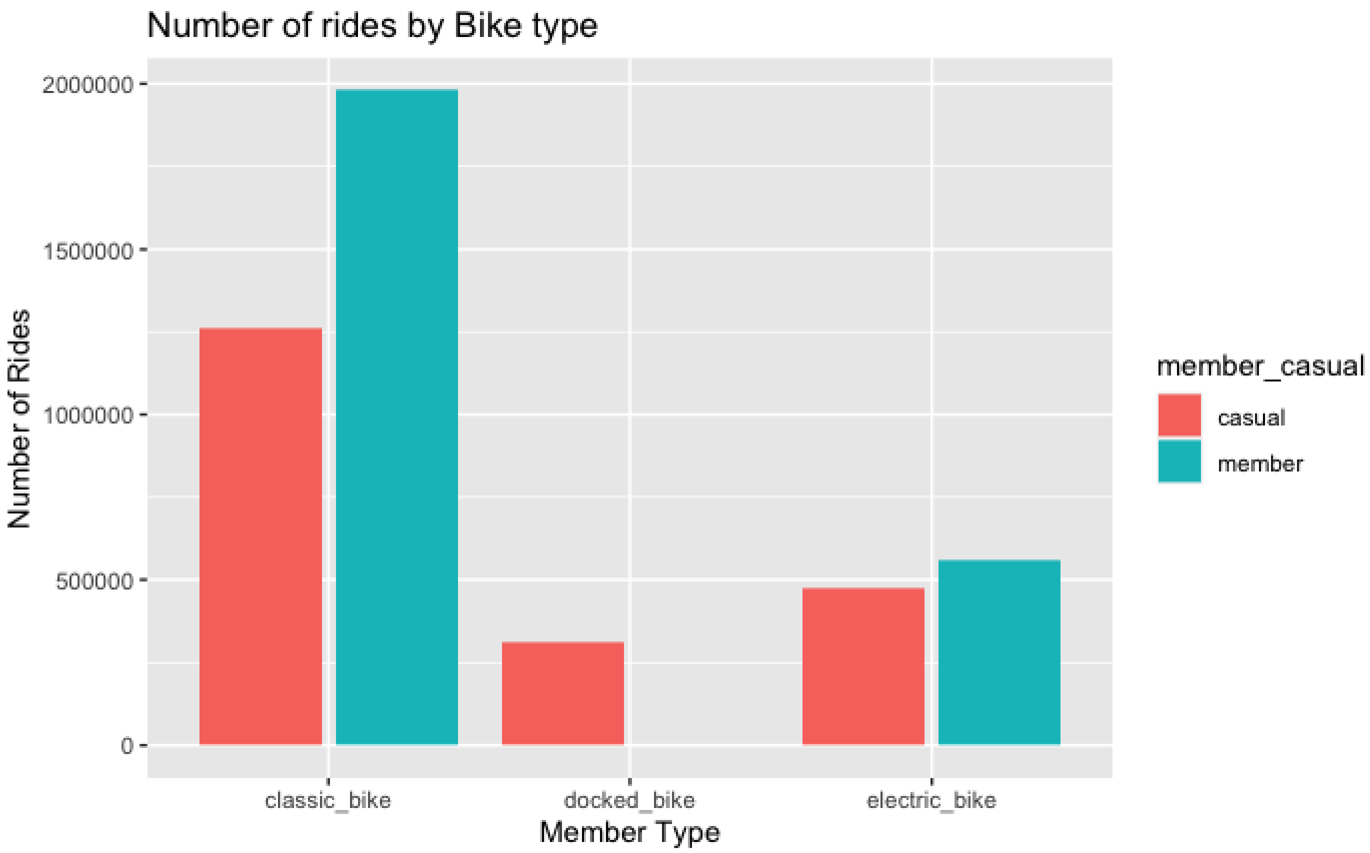
```
`summarise()` has grouped output by 'member_casual'. You can override using the `.groups
` argument.
```



Hide

```
#Number of rides by Bike type
trip_datas_v2 %>% group_by(rideable_type,member_casual) %>%
  summarise(ride_count=n()) %>%
  ggplot(aes(x=rideable_type, y= ride_count, fill=member_casual)) +
  geom_col(position= "dodge2") +
  labs(title = "Number of rides by Bike type", x="Member Type", y="Number of Rides")
```

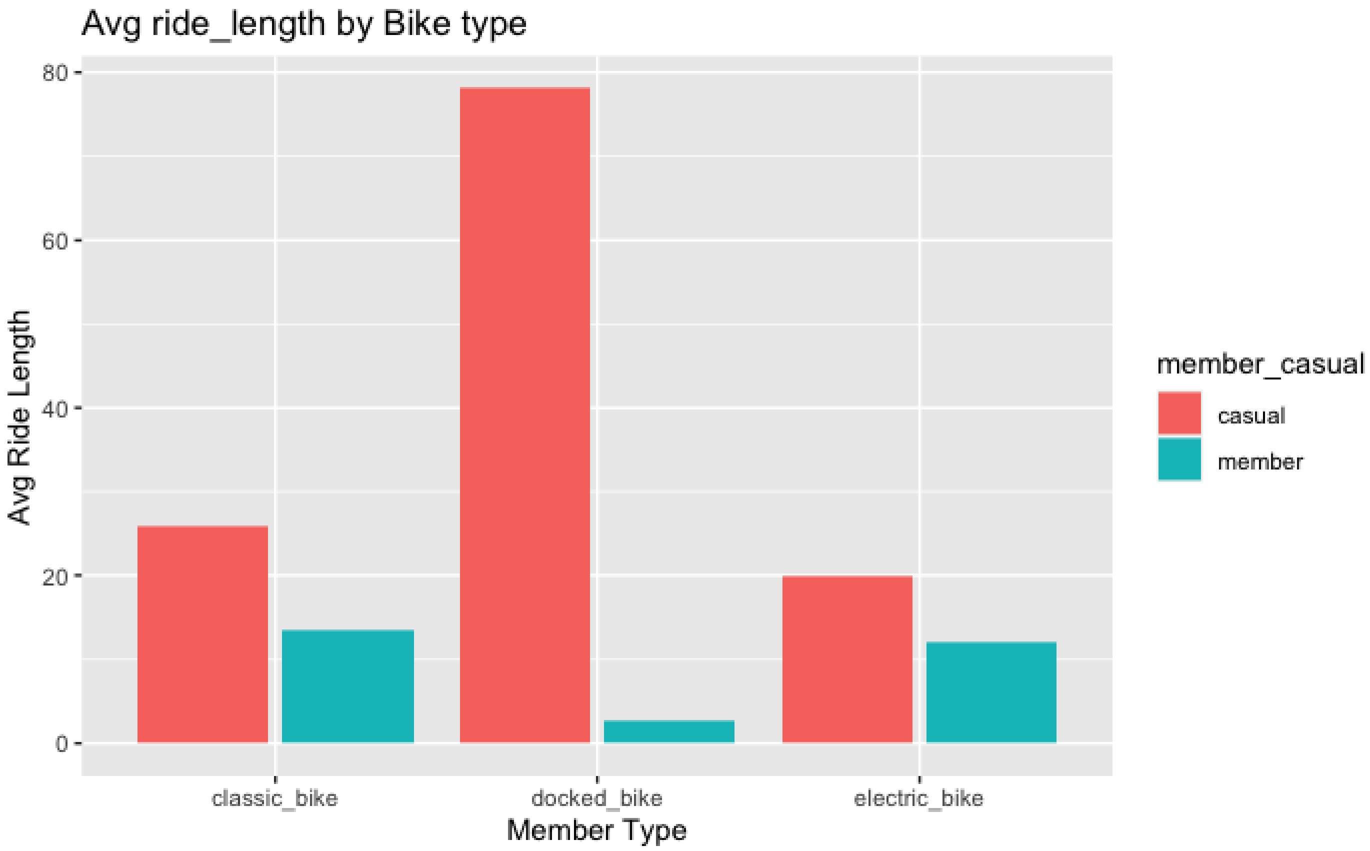
```
`summarise()` has grouped output by 'rideable_type'. You can override using the `.groups
` argument.
```



Hide

```
#Avg ride_length by Bike type
trip_datas_v2 %>% group_by(rideable_type,member_casual) %>%
  summarise(avg_ride_length= mean(ride_length)) %>%
  ggplot(aes(x=rideable_type, y= avg_ride_length, fill=member_casual)) +
  geom_col(position= "dodge2") +
  labs(title = "Avg ride_length by Bike type", x="Member Type", y="Avg Ride Length")
```

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



Hide

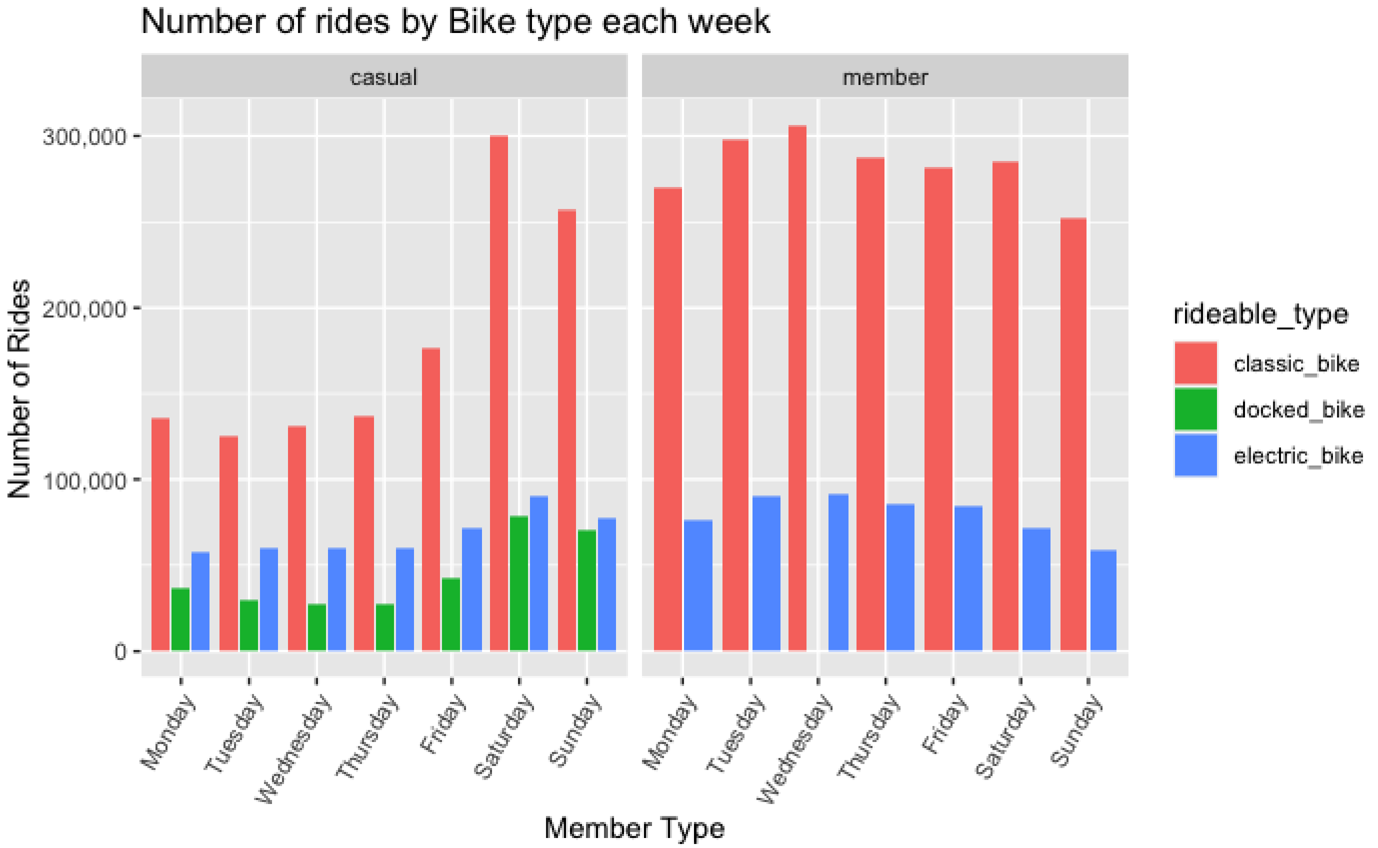
```
#Number of rides by Bike type each month
trip_datas_v2 %>% group_by(rideable_type,member_casual,month) %>%
  summarise(ride_count=n()) %>%
  ggplot(aes(x=month, y= ride_count, fill=rideable_type)) +
  geom_col(position= "dodge2") +
  facet_wrap(~member_casual)+
  labs(title = "Number of rides by Bike type each month", x="Member Type", y="Number of
Rides") + theme(axis.text.x = element_text(angle = 60, hjust=1))+ scale_y_continuous(lab
els = comma)
```

```
`summarise()` has grouped output by 'rideable_type', 'member_casual'. You can override u
sing the `.groups` argument.Error in check_breaks_labels(breaks, labels) : object 'comm
a' not found
```

Hide

```
#Number of rides by Bike type each week
trip_datas_v2 %>% group_by(rideable_type,member_casual,week_day) %>%
  summarise(ride_count=n()) %>%
  ggplot(aes(x=week_day, y= ride_count, fill=rideable_type)) +
  geom_col(position= "dodge2") +
  facet_wrap(~member_casual)+
  labs(title = "Number of rides by Bike type each week", x="Member Type", y="Number of R
ides") + scale_y_continuous(labels = comma) + theme(axis.text.x = element_text(angle = 6
0, hjust=1))
```

```
`summarise()` has grouped output by 'rideable_type', 'member_casual'. You can override u
sing the `.groups` argument.
```



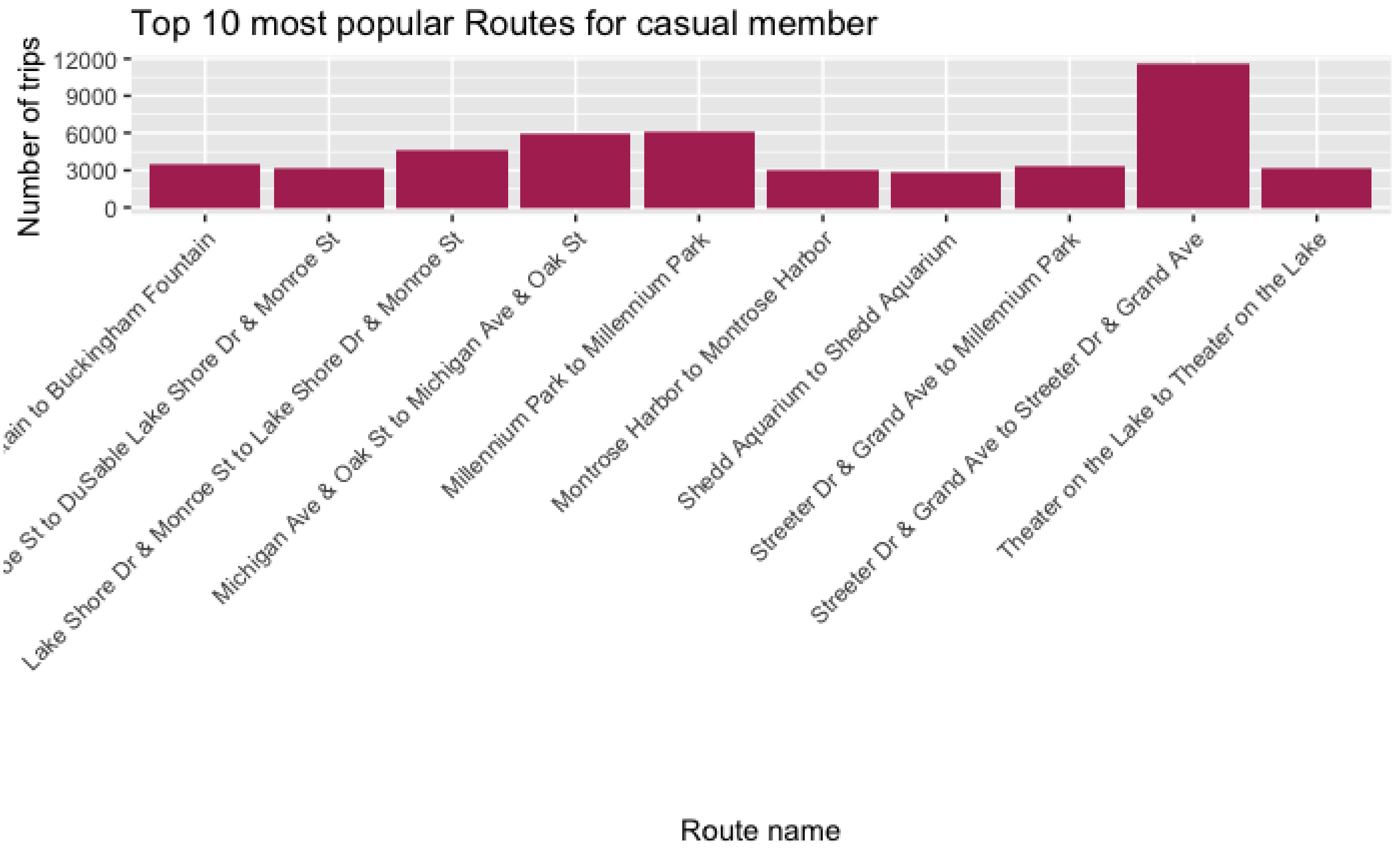
Hide

```
# top_10 routes per number of rides by Casual member
top_10_routes <- trip_datas_v2 %>%
  group_by(ride_routes,member_casual) %>%
  summarise(count=n()) %>%
  filter(member_casual=="casual") %>%
  arrange(desc(count)) %>% head(10) %>%
  ggplot(aes(x=ride_routes, y=count)) + geom_col(fill="maroon")  +
  labs(title="Top 10 most popular Routes for casual member", x="Route name", y="Number o
f trips")+theme(axis.text.x = element_text(angle = 45, hjust=1))
```

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

Hide

```
top_10_routes
```



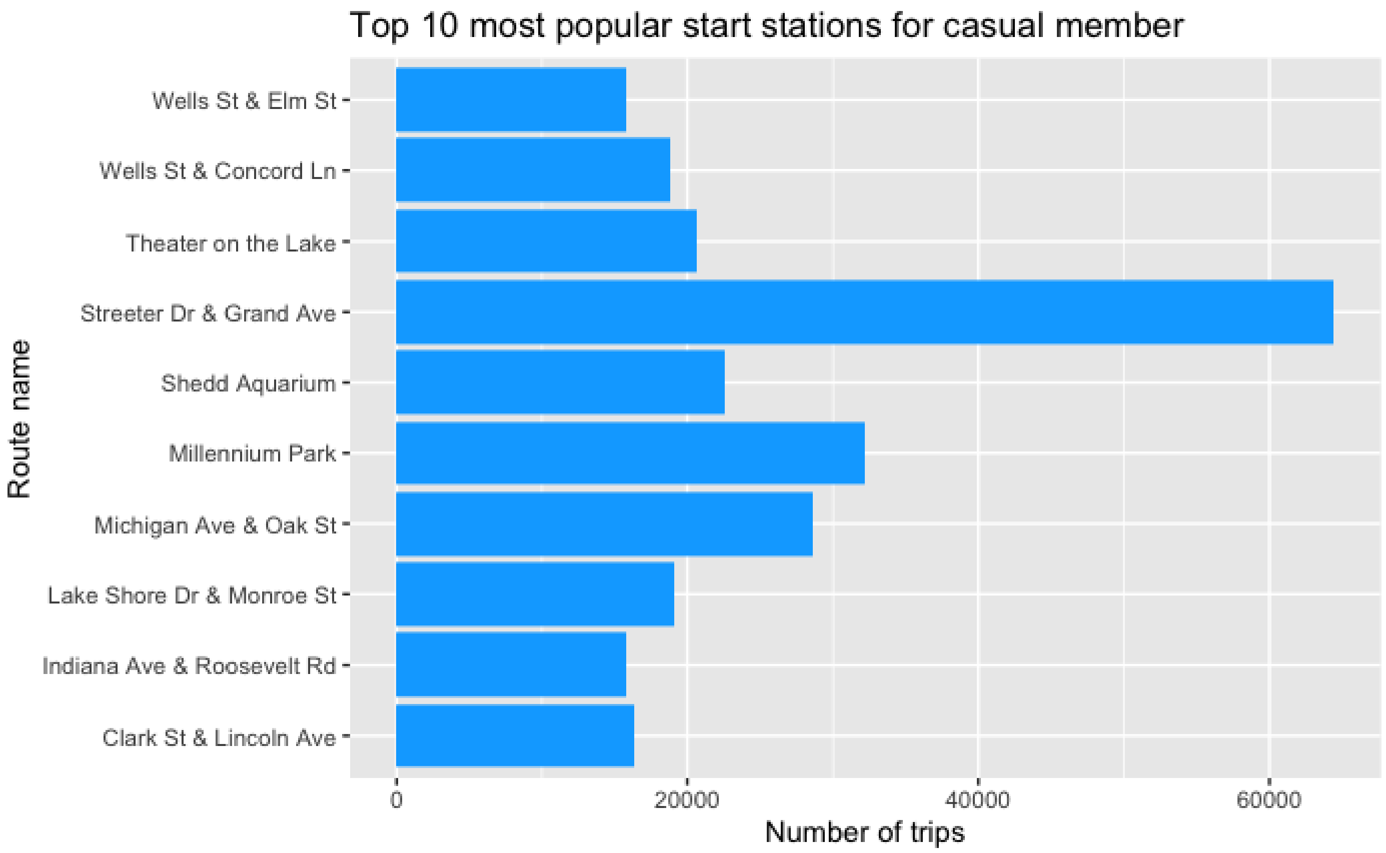
Hide

```
# top_10 start stations per number of rides by Casual member
top_10_start_stations <- trip_datas %>%
  group_by(start_station_name,member_casual) %>%
  summarise(count=n()) %>%
  filter(member_casual=="casual") %>%
  arrange(desc(count)) %>% head(10) %>%
  ggplot(aes(x=start_station_name, y=count)) + geom_col(fill="#00abff")  + coord_flip()+
  labs(title="Top 10 most popular start stations for casual member", x="Route name", y
="Number of trips")
```

```
`summarise()` has grouped output by 'start_station_name'. You can override using the `.g
roups` argument.
```

Hide

```
top_10_start_stations
```



Top 10 most popular start stations for casual member

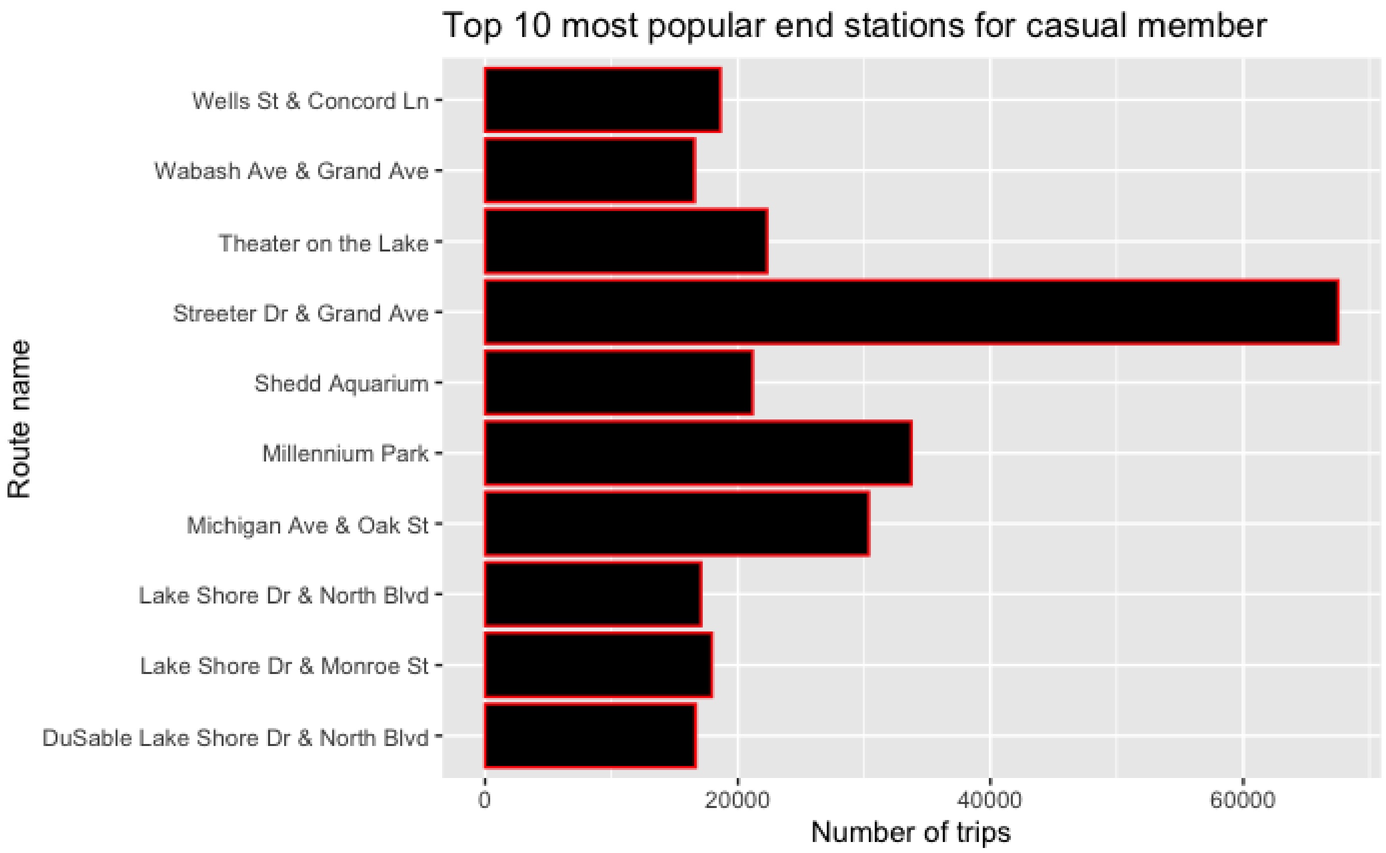
Hide

```
# top_10 end stations per number of rides by Casual member
top_10_end_stations <- trip_datas %>%
  group_by(end_station_name,member_casual) %>%
  summarise(count=n()) %>%
  filter(member_casual=="casual") %>%
  arrange(desc(count)) %>% head(10) %>%
  ggplot(aes(x=end_station_name, y=count)) + geom_col(fill="black",color="red")  + coord
_flip()+
  labs(title="Top 10 most popular end stations for casual member", x="Route name", y="Nu
mber of trips")
```

```
`summarise()` has grouped output by 'end_station_name'. You can override using the `.gro
ups` argument.
```

Hide

```
top_10_end_stations
```



Hide

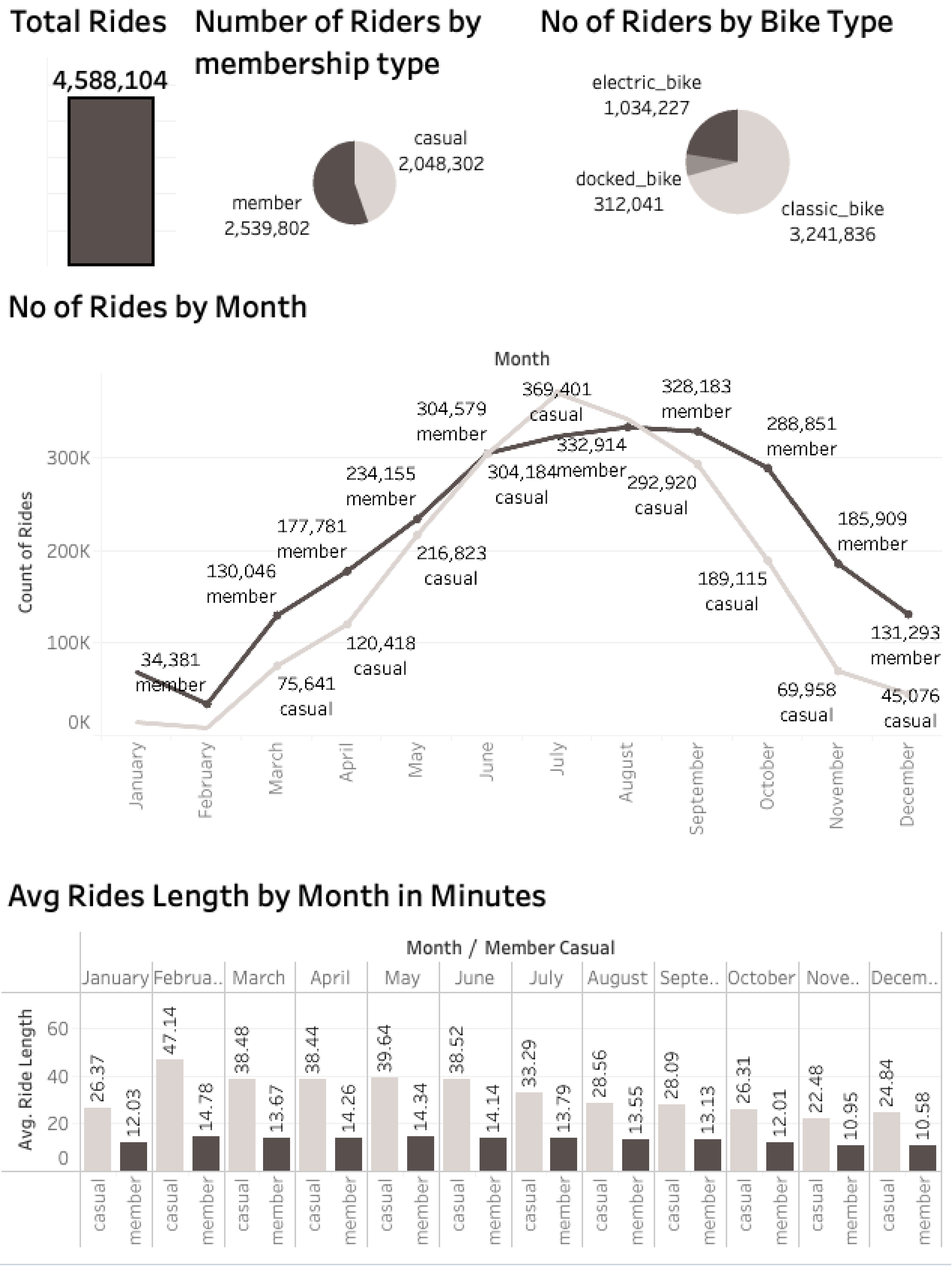
```
write.csv(trip_datas,"/Users/ikbalhossain/Documents/projects/Google_Analytics /Case Study/trip_datas.csv", row.names=FALSE)
```

Share

Further visualization being done in tableau. In Tableau it shows the key findings and their visualization

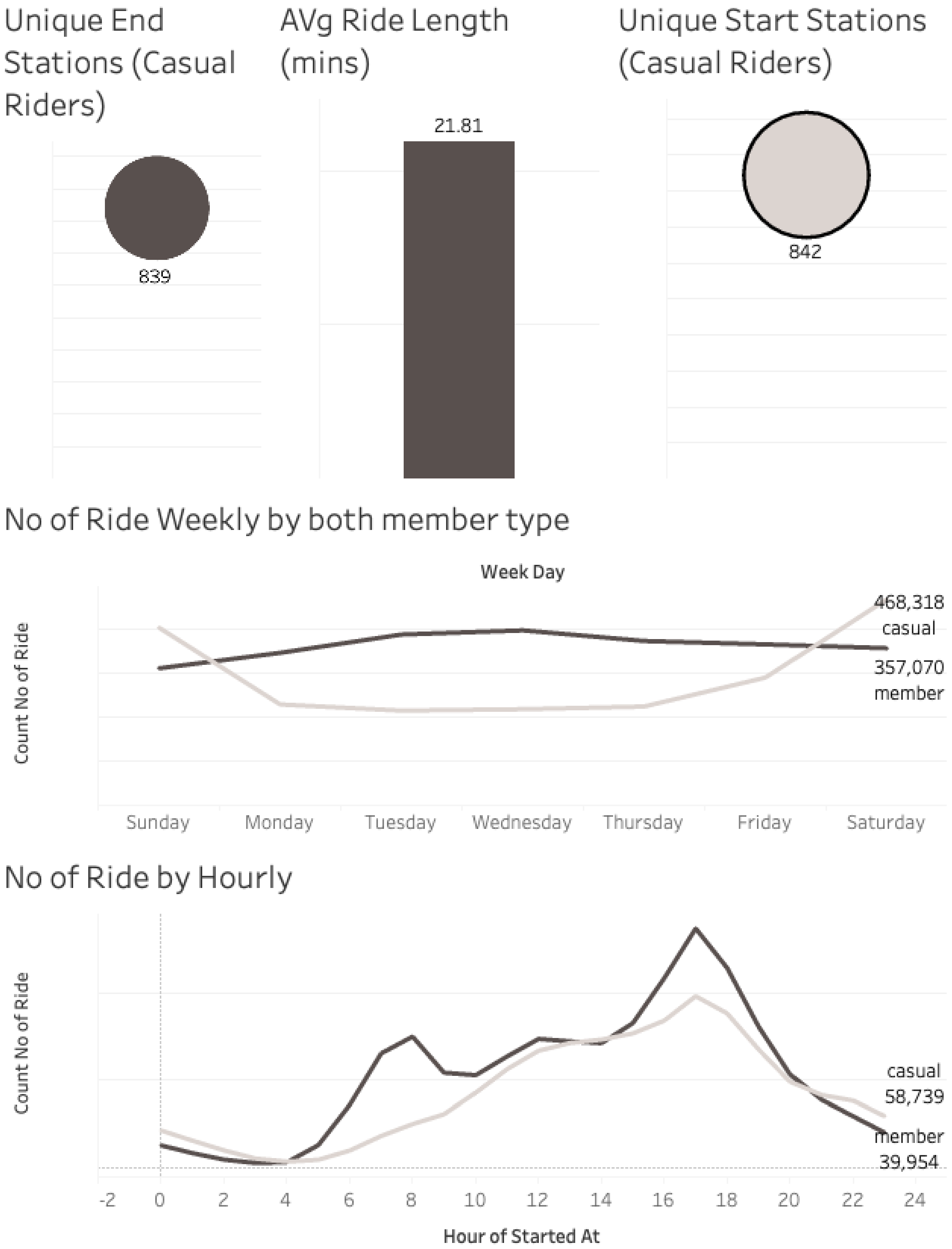
Hide

```
knitr::include_graphics("/Users/ikbalhossain/Desktop/Screen Shot 2023-03-09 at 20.39.09 PM.png")
```



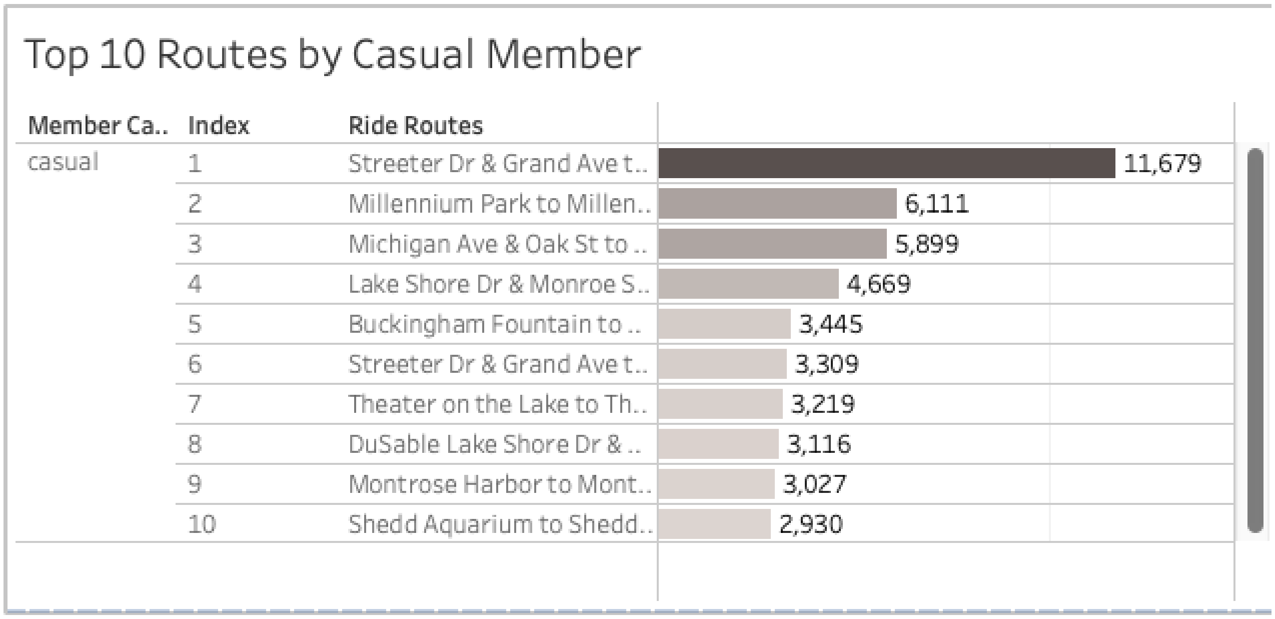
Hide

```
knitr::include_graphics("/Users/ikbalhossain/Desktop/Screen Shot 2023-03-09 at 20.39.01
PM.png")
```



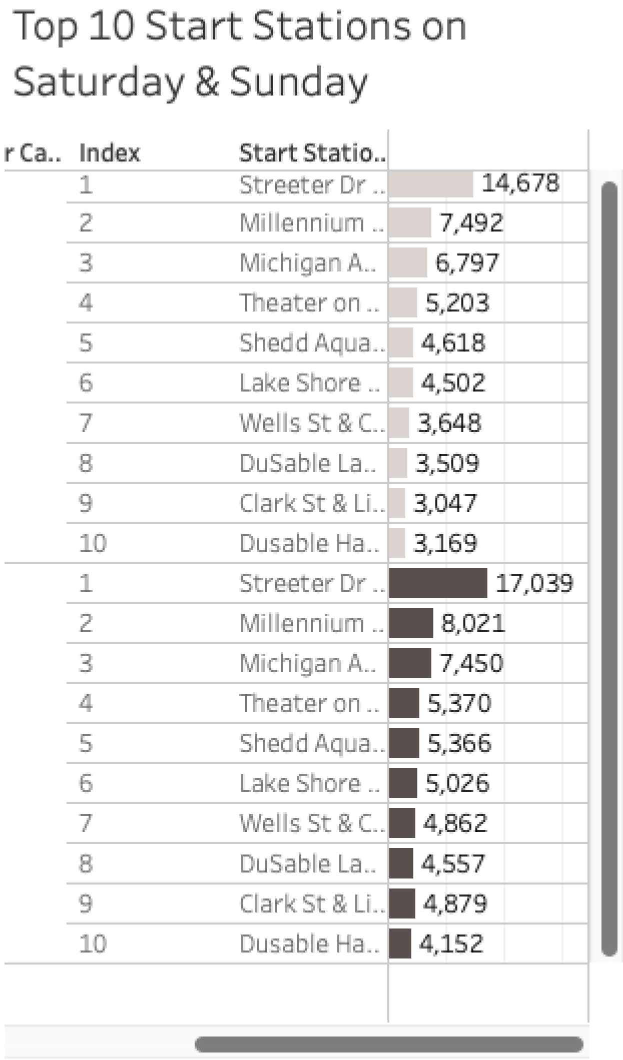
Hide

```
knitr::include_graphics("/Users/ikbalhossain/Desktop/Screen Shot 2023-03-09 at 20.39.20
PM.png")
```



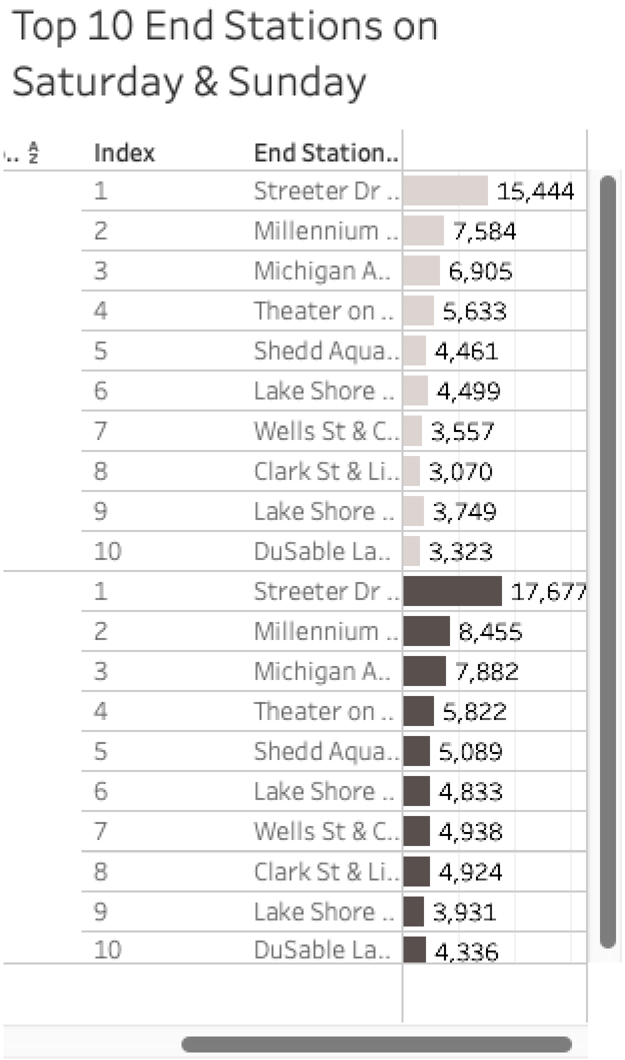
Top 10 Routes by Casual Member

Member Ca..	Index	Ride Routes	
casual	1	Streeter Dr & Grand Ave t..	11,679
	2	Millennium Park to Millen..	6,111
	3	Michigan Ave & Oak St to ..	5,899
	4	Lake Shore Dr & Monroe S..	4,669
	5	Buckingham Fountain to ..	3,445
	6	Streeter Dr & Grand Ave t..	3,309
	7	Theater on the Lake to Th..	3,219
	8	DuSable Lake Shore Dr & ..	3,116
	9	Montrose Harbor to Mont..	3,027
	10	Shedd Aquarium to Shedd..	2,930



Top 10 Start Stations on Saturday & Sunday

r Ca..	Index	Start Statio..	
	1	Streeter Dr ..	14,678
	2	Millennium ..	7,492
	3	Michigan A..	6,797
	4	Theater on ..	5,203
	5	Shedd Aqua..	4,618
	6	Lake Shore ..	4,502
	7	Wells St & C..	3,648
	8	DuSable La..	3,509
	9	Clark St & Li..	3,047
	10	Dusable Ha..	3,169
	1	Streeter Dr ..	17,039
	2	Millennium ..	8,021
	3	Michigan A..	7,450
	4	Theater on ..	5,370
	5	Shedd Aqua..	5,366
	6	Lake Shore ..	5,026
	7	Wells St & C..	4,862
	8	DuSable La..	4,557
	9	Clark St & Li..	4,879
	10	Dusable Ha..	4,152



Top 10 End Stations on Saturday & Sunday

..	Index	End Station..	
	1	Streeter Dr ..	15,444
	2	Millennium ..	7,584
	3	Michigan A..	6,905
	4	Theater on ..	5,633
	5	Shedd Aqua..	4,461
	6	Lake Shore ..	4,499
	7	Wells St & C..	3,557
	8	Clark St & Li..	3,070
	9	Lake Shore ..	3,749
	10	DuSable La..	3,323
	1	Streeter Dr ..	17,677
	2	Millennium ..	8,455
	3	Michigan A..	7,882
	4	Theater on ..	5,822
	5	Shedd Aqua..	5,089
	6	Lake Shore ..	4,833
	7	Wells St & C..	4,938
	8	Clark St & Li..	4,924
	9	Lake Shore ..	3,931
	10	DuSable La..	4,336

Act

After Ask, prepare, process, analyze, share, now we are at act phase. After analyzing the data here is my top 3 recommendations.

- 1. Cyclistic company could consider implementing a promotional campaign aimed at casual members who frequently ride on Saturdays and Sundays. In order to encourage more rides on weekdays, the company may want to offer a new membership discount specifically for those days, focusing on the most frequently used start and end stations. This could help to increase the number of rides taken by casual members on weekdays, while also incentivizing continued weekend use.
- 2. To encourage more ridership during the low seasons of winter and spring, Cyclistic company could consider offering a seasonal membership with discounts. A potential promotion to attract casual riders might include a free 7-day membership trial, highlighting the benefits and convenience of becoming a member. By leveraging the lower demand of these seasons, Cyclistic could potentially increase its membership base and rider loyalty.
- 3. Cyclistic company may want to focus on increasing ridership for its electric bikes by offering targeted membership promotions to its most frequent casual riders. This could include providing bonuses and coupons to incentivize riders to try out the electric bikes and potentially become members. By highlighting the benefits of electric bikes, such as convenience and sustainability, Cyclistic could increase ridership and encourage more members to choose electric bikes as their preferred mode of transportation.