Cyclistic Bike Share : Case study with R

Rounak Saha

2024-07-18

Introduction

This is a capstone project as a part my Google Data Analytics Professional Certificate course. In this project i have used R programming language and R studio IDE as a data analysis tool, for the simplicity of its data analysis and data visualization.

For this project following steps should be followed:

- Ask
- Prepare
- Process
- Analyze
- Share
- 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.

Scenario In the given scenario, I am 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, my team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, our team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve my recommendations, so they must be backed up with compelling data insights and professional data visualizations.

My report will include the following deliverables:

- A clear summary of the business task
- A description of all data sources used
- Documentation of any cleaning or manipulation of data
- A summary of your analysis
- Supporting visualizations and key findings
- Your top high-level content recommendations based on your analysis

Step 1: Ask

Business Task 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.

Key stakeholders

- **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.

Questions to explore for the analysis

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

Step 2: Prepare

I will use Cyclistic's historical trip data (July 2023 to June 2024) to analyze and identify trends. The data has been made available by Motivate International Inc. under the license. The dataset is available here.

The data set contains 12 CSV files organized in long format. Below is a breakdown of the data using the ROCCC approach:

- **Reliable**- Yes, not biased, but not sure if vetted, as its given by a certification course
- **Original**- Yes, located at original public data
- **Comprehensive**- Yes, not missing any important information
- **Current**- Yes, the data is updated monthly
- **Cited**- Yes

Step 3: Process

First, we will install and load the required packages

```r
## Installing the requird packages
install.packages("tidyverse", repos="https://cloud.r-project.org/")
install.packages("readr", repos="https://cloud.r-project.org/")
install.packages("dplyr", repos="https://cloud.r-project.org/")
install.packages("tidyr", repos="https://cloud.r-project.org/")
install.packages("ggplot2", repos="https://cloud.r-project.org/")
install.packages("lubridate",repos="https://cloud.r-project.org/")
install.packages("geosphere",repos="https://cloud.r-project.org/")
install.packages("ggmap",repos="https://cloud.r-project.org/")
install.packages("sqldf",repos="https://cloud.r-project.org/")
install.packages("scales", repos="https://cloud.r-project.org/")
```

```r
## Loading the packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

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

```
## i Google's Terms of Service: <https://mapsplatform.google.com>
##   Stadia Maps' Terms of Service: <https://stadiamaps.com/terms-of-service/>
##   OpenStreetMap's Tile Usage Policy: <https://operations.osmfoundation.org/policies/tiles/>
## i Please cite ggmap if you use it! Use `citation("ggmap")` for details.
```

```r
library(sqldf)
```

```
## Loading required package: gsubfn
## Loading required package: proto

## Warning in doTryCatch(return(expr), name, parentenv, handler): unable to load shared object '/Library
##   dlopen(/Library/Frameworks/R.framework/Resources/modules//R_X11.so, 0x0006): Library not loaded: /
##   Referenced from: <B3716E5A-BF4D-3CA3-B8EB-89643DB72A04> /Library/Frameworks/R.framework/Versions/4
##   Reason: tried: '/opt/X11/lib/libSM.6.dylib' (no such file), '/System/Volumes/Preboot/Cryptexes/OS/

## tcltk DLL is linked to '/opt/X11/lib/libX11.6.dylib'
## Could not load tcltk.  Will use slower R code instead.
## Loading required package: RSQLite
```

```
## Warning: package 'RSQLite' was built under R version 4.3.3
```

```
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

We will now load the dataframes using `read_csv()` function

```
## Import data into R Studio
jul23 <- read_csv("data-June2023:June2024/202307-divvy-tripdata.csv")
```

```
## Rows: 767650 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
aug23 <- read_csv("data-June2023:June2024/202308-divvy-tripdata.csv")
```

```
## Rows: 771693 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
sep23 <- read_csv("data-June2023:June2024/202309-divvy-tripdata.csv")
```

```
## Rows: 666371 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
oct23 <- read_csv("data-June2023:June2024/202310-divvy-tripdata.csv")
```

```
## Rows: 537113 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
nov23 <- read_csv("data-June2023:June2024/202311-divvy-tripdata.csv")
```

```
## Rows: 362518 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
dec23 <- read_csv("data-June2023:June2024/202312-divvy-tripdata.csv")
```

```
## Rows: 224073 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
jan24 <- read_csv("data-June2023:June2024/202401-divvy-tripdata.csv")
```

```
## Rows: 144873 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
feb24 <- read_csv("data-June2023:June2024/202402-divvy-tripdata.csv")
```

```
## Rows: 223164 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
mar24 <- read_csv("data-June2023:June2024/202403-divvy-tripdata.csv")
```

```
## Rows: 301687 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
apr24 <- read_csv("data-June2023:June2024/202404-divvy-tripdata.csv")
```

```
## Rows: 415025 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
may24 <- read_csv("data-June2023:June2024/202405-divvy-tripdata.csv")
```

```
## Rows: 609493 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
jun24 <- read_csv("data-June2023:June2024/202406-divvy-tripdata.csv")
```

```
## Rows: 710721 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

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

Examining each of the datasets

```
glimpse(jul23)
```

```
## Rows: 767,650
## Columns: 13
## $ ride_id            <chr> "9340B064F0AEE130", "D1460EE3CE0D8AF8", "DF41BE31B8~
## $ rideable_type      <chr> "electric_bike", "classic_bike", "classic_bike", "e~
## $ started_at         <dttm> 2023-07-23 20:06:14, 2023-07-23 17:05:07, 2023-07-~
## $ ended_at           <dttm> 2023-07-23 20:22:44, 2023-07-23 17:18:37, 2023-07-~
## $ start_station_name <chr> "Kedzie Ave & 110th St", "Western Ave & Walton St",~
## $ start_station_id   <chr> "20204", "KA1504000103", "KA1504000103", "13155", "~
## $ end_station_name   <chr> "Public Rack - Racine Ave & 109th Pl", "Milwaukee A~
## $ end_station_id     <chr> "877", "13033", "TA1305000041", "TA1305000032", "TA~
## $ start_lat          <dbl> 41.69241, 41.89842, 41.89842, 41.88411, 41.96709, 4~
## $ start_lng          <dbl> -87.70091, -87.68660, -87.68660, -87.65694, -87.667~
## $ end_lat            <dbl> 41.69483, 41.89158, 41.90940, 41.88275, 41.96398, 4~
## $ end_lng            <dbl> -87.65304, -87.64838, -87.67769, -87.64119, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(aug23)
```

```
## Rows: 771,693
## Columns: 13
## $ ride_id            <chr> "903C30C2D810A53B", "F2FB18A98E110A2B", "D0DEC7C94E~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2023-08-19 15:41:53, 2023-08-18 15:30:18, 2023-08-~
## $ ended_at           <dttm> 2023-08-19 15:53:36, 2023-08-18 15:45:25, 2023-08-~
## $ start_station_name <chr> "LaSalle St & Illinois St", "Clark St & Randolph St~
## $ start_station_id   <chr> "13430", "TA1305000030", "TA1305000030", "KA1504000~
## $ end_station_name   <chr> "Clark St & Elm St", NA, NA, NA, NA, NA, NA, NA, NA~
## $ end_station_id     <chr> "TA1307000039", NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lat          <dbl> 41.89072, 41.88451, 41.88498, 41.90310, 41.88555, 4~
## $ start_lng          <dbl> -87.63148, -87.63155, -87.63079, -87.63467, -87.632~
## $ end_lat            <dbl> 41.90297, 41.93000, 41.91000, 41.90000, 41.89000, 4~
## $ end_lng            <dbl> -87.63128, -87.64000, -87.63000, -87.62000, -87.680~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(sep23)
```

```
## Rows: 666,371
## Columns: 13
## $ ride_id            <chr> "011C1903BF4E2E28", "87DB80E048A1BF9F", "7C2EB7AF66~
## $ rideable_type      <chr> "classic_bike", "classic_bike", "electric_bike", "c~
## $ started_at         <dttm> 2023-09-23 00:27:50, 2023-09-02 09:26:43, 2023-09-~
```

```
## $ ended_at           <dttm> 2023-09-23 00:33:27, 2023-09-02 09:38:19, 2023-09-~
## $ start_station_name <chr> "Halsted St & Wrightwood Ave", "Clark St & Drummond~
## $ start_station_id   <chr> "TA1309000061", "TA1307000142", "SL-010", "TA130700~
## $ end_station_name   <chr> "Sheffield Ave & Wellington Ave", "Racine Ave & Ful~
## $ end_station_id     <chr> "TA1307000052", "TA1306000026", "13304", "TA1308000~
## $ start_lat          <dbl> 41.92914, 41.93125, 41.87506, 41.93125, 41.92914, 4~
## $ start_lng          <dbl> -87.64908, -87.64434, -87.63314, -87.64434, -87.649~
## $ end_lat            <dbl> 41.93625, 41.92557, 41.86127, 41.93974, 41.92557, 4~
## $ end_lng            <dbl> -87.65266, -87.65842, -87.65663, -87.65887, -87.658~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(oct23)
```

```
## Rows: 537,113
## Columns: 13
## $ ride_id            <chr> "4449097279F8BBE7", "9CF060543CA7B439", "667F21F4D6~
## $ rideable_type      <chr> "classic_bike", "electric_bike", "electric_bike", "~
## $ started_at         <dttm> 2023-10-08 10:36:26, 2023-10-11 17:23:59, 2023-10-~
## $ ended_at           <dttm> 2023-10-08 10:49:19, 2023-10-11 17:36:08, 2023-10-~
## $ start_station_name <chr> "Orleans St & Chestnut St (NEXT Apts)", "Desplaines~
## $ start_station_id   <chr> "620", "TA1306000003", "620", "TA1306000003", "TA13~
## $ end_station_name   <chr> "Sheffield Ave & Webster Ave", "Sheffield Ave & Web~
## $ end_station_id     <chr> "TA1309000033", "TA1309000033", "TA1307000111", "TA~
## $ start_lat          <dbl> 41.89820, 41.88864, 41.89807, 41.88872, 41.88872, 4~
## $ start_lng          <dbl> -87.63754, -87.64441, -87.63751, -87.64445, -87.644~
## $ end_lat            <dbl> 41.92154, 41.92154, 41.88584, 41.88584, 41.88584, 4~
## $ end_lng            <dbl> -87.65382, -87.65382, -87.63550, -87.63550, -87.635~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(nov23)
```

```
## Rows: 362,518
## Columns: 13
## $ ride_id            <chr> "4EAD8F1AD547356B", "6322270563BF5470", "B37BDE091E~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2023-11-30 21:50:05, 2023-11-03 09:44:02, 2023-11-~
## $ ended_at           <dttm> 2023-11-30 22:13:27, 2023-11-03 10:17:15, 2023-11-~
## $ start_station_name <chr> "Millennium Park", "Broadway & Sheridan Rd", "State~
## $ start_station_id   <chr> "13008", "13323", "TA1307000061", "TA1308000001", "~
## $ end_station_name   <chr> "Pine Grove Ave & Waveland Ave", "Broadway & Sherid~
## $ end_station_id     <chr> "TA1307000150", "13323", "TA1307000061", "TA1308000~
## $ start_lat          <dbl> 41.88110, 41.95287, 41.89753, 41.92628, 41.92628, 4~
## $ start_lng          <dbl> -87.62408, -87.65003, -87.62869, -87.63083, -87.630~
## $ end_lat            <dbl> 41.94947, 41.95283, 41.89745, 41.92628, 41.92628, 4~
## $ end_lng            <dbl> -87.64645, -87.64999, -87.62872, -87.63083, -87.630~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(dec23)
```

```
## Rows: 224,073
## Columns: 13
## $ ride_id            <chr> "C9BD54F578F57246", "CDBD92F067FA620E", "ABC0858E52~
```

```
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2023-12-02 18:44:01, 2023-12-02 18:48:19, 2023-12-~
## $ ended_at           <dttm> 2023-12-02 18:47:51, 2023-12-02 18:54:48, 2023-12-~
## $ start_station_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_station_id   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lat          <dbl> 41.92, 41.92, 41.89, 41.95, 41.92, 41.91, 41.99, 42~
## $ start_lng          <dbl> -87.66, -87.66, -87.62, -87.65, -87.64, -87.63, -87~
## $ end_lat            <dbl> 41.92, 41.89, 41.90, 41.94, 41.93, 41.88, 42.00, 41~
## $ end_lng            <dbl> -87.66, -87.64, -87.64, -87.65, -87.64, -87.65, -87~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(jan24)
```

```
## Rows: 144,873
## Columns: 13
## $ ride_id            <chr> "C1D650626C8C899A", "EECD38BDB25BFCB0", "F4A9CE7806~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-01-12 15:30:27, 2024-01-08 15:45:46, 2024-01-~
## $ ended_at           <dttm> 2024-01-12 15:37:59, 2024-01-08 15:52:59, 2024-01-~
## $ start_station_name <chr> "Wells St & Elm St", "Wells St & Elm St", "Wells St~
## $ start_station_id   <chr> "KA1504000135", "KA1504000135", "KA1504000135", "TA~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Kingsbury St & Kinzie ~
## $ end_station_id     <chr> "KA1503000043", "KA1503000043", "KA1503000043", "13~
## $ start_lat          <dbl> 41.90327, 41.90294, 41.90295, 41.88430, 41.94880, 4~
## $ start_lng          <dbl> -87.63474, -87.63444, -87.63447, -87.63396, -87.675~
## $ end_lat            <dbl> 41.88918, 41.88918, 41.88918, 41.92182, 41.88918, 4~
## $ end_lng            <dbl> -87.63851, -87.63851, -87.63851, -87.64414, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(feb24)
```

```
## Rows: 223,164
## Columns: 13
## $ ride_id            <chr> "FCB05EB1758F85E8", "7FB986AD5D3DE9D6", "40CA13E15B~
## $ rideable_type      <chr> "classic_bike", "classic_bike", "electric_bike", "c~
## $ started_at         <dttm> 2024-02-03 14:14:18, 2024-02-05 21:10:06, 2024-02-~
## $ ended_at           <dttm> 2024-02-03 14:21:00, 2024-02-05 21:15:44, 2024-02-~
## $ start_station_name <chr> "Clark St & Newport St", "Michigan Ave & Washington~
## $ start_station_id   <chr> "632", "13001", "TA1309000029", "13235", "KA1503000~
## $ end_station_name   <chr> "Southport Ave & Waveland Ave", "Wabash Ave & Grand~
## $ end_station_id     <chr> "13235", "TA1307000117", "13243", "13229", "KA15030~
## $ start_lat          <dbl> 41.94454, 41.88398, 41.91760, 41.94815, 41.83078, 4~
## $ start_lng          <dbl> -87.65468, -87.62468, -87.68250, -87.66394, -87.632~
## $ end_lat            <dbl> 41.94815, 41.89147, 41.91262, 41.93948, 41.83846, 4~
## $ end_lng            <dbl> -87.66394, -87.62676, -87.68139, -87.66375, -87.635~
## $ member_casual      <chr> "member", "member", "member", "member", "casual", "~
```

```
glimpse(mar24)
```

```
## Rows: 301,687
```

```
## Columns: 13
## $ ride_id            <chr> "64FBE3BAED5F29E6", "9991629435C5E20E", "E5C9FECD5B~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-03-05 18:33:11, 2024-03-06 17:15:14, 2024-03-~
## $ ended_at           <dttm> 2024-03-05 18:51:48, 2024-03-06 17:16:04, 2024-03-~
## $ start_station_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_station_id   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lat          <dbl> 41.94, 41.91, 41.91, 41.90, 41.93, 41.93, 41.94, 41~
## $ start_lng          <dbl> -87.65, -87.64, -87.64, -87.63, -87.70, -87.70, -87~
## $ end_lat            <dbl> 41.96, 41.91, 41.92, 41.89, 41.93, 41.95, 41.95, 41~
## $ end_lng            <dbl> -87.65, -87.64, -87.64, -87.63, -87.72, -87.68, -87~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(apr24)
```

```
## Rows: 415,025
## Columns: 13
## $ ride_id            <chr> "743252713F32516B", "BE90D33D2240C614", "D47BBDDE7C~
## $ rideable_type      <chr> "classic_bike", "electric_bike", "classic_bike", "c~
## $ started_at         <dttm> 2024-04-22 19:08:21, 2024-04-11 06:19:24, 2024-04-~
## $ ended_at           <dttm> 2024-04-22 19:12:56, 2024-04-11 06:22:21, 2024-04-~
## $ start_station_name <chr> "Aberdeen St & Jackson Blvd", "Aberdeen St & Jackso~
## $ start_station_id   <chr> "13157", "13157", "TA1307000107", "13157", "TA13070~
## $ end_station_name   <chr> "Desplaines St & Jackson Blvd", "Desplaines St & Ja~
## $ end_station_id     <chr> "15539", "15539", "13249", "15539", "TA1308000029",~
## $ start_lat          <dbl> 41.87773, 41.87772, 41.96167, 41.87773, 41.96161, 4~
## $ start_lng          <dbl> -87.65479, -87.65496, -87.65464, -87.65479, -87.654~
## $ end_lat            <dbl> 41.87812, 41.87812, 41.95606, 41.87812, 41.88683, 4~
## $ end_lng            <dbl> -87.64395, -87.64395, -87.66884, -87.64395, -87.622~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```
glimpse(may24)
```

```
## Rows: 609,493
## Columns: 13
## $ ride_id            <chr> "7D9F0CE9EC2A1297", "02EC47687411416F", "101370FB2D~
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "el~
## $ started_at         <dttm> 2024-05-25 15:52:42, 2024-05-14 15:11:51, 2024-05-~
## $ ended_at           <dttm> 2024-05-25 16:11:50, 2024-05-14 15:22:00, 2024-05-~
## $ start_station_name <chr> "Streeter Dr & Grand Ave", "Sheridan Rd & Greenleaf~
## $ start_station_id   <chr> "13022", "KA1504000159", "13022", "13022", "KA15040~
## $ end_station_name   <chr> "Clark St & Elm St", "Sheridan Rd & Loyola Ave", "W~
## $ end_station_id     <chr> "TA1307000039", "RP-009", "TA1309000010", "TA130700~
## $ start_lat          <dbl> 41.89228, 42.01059, 41.89228, 41.89227, 41.90349, 4~
## $ start_lng          <dbl> -87.61204, -87.66241, -87.61204, -87.61195, -87.643~
## $ end_lat            <dbl> 41.90297, 42.00104, 41.87077, 41.93625, 41.90297, 4~
## $ end_lng            <dbl> -87.63128, -87.66120, -87.62573, -87.65266, -87.631~
## $ member_casual      <chr> "casual", "casual", "member", "member", "casual", "~
```

```r
glimpse(jun24)
```

```
## Rows: 710,721
## Columns: 13
## $ ride_id            <chr> "CDE6023BE6B11D2F", "462B48CD292B6A18", "9CFB6A858D~
## $ rideable_type      <chr> "electric_bike", "electric_bike", "electric_bike", ~
## $ started_at         <dttm> 2024-06-11 17:20:06, 2024-06-11 17:19:21, 2024-06-~
## $ ended_at           <dttm> 2024-06-11 17:21:39, 2024-06-11 17:19:36, 2024-06-~
## $ start_station_name <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_station_id   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_name   <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ end_station_id     <chr> NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA, NA,~
## $ start_lat          <dbl> 41.89, 41.89, 41.93, 41.88, 41.94, 41.94, 41.94, 41~
## $ start_lng          <dbl> -87.65, -87.65, -87.65, -87.64, -87.64, -87.64, -87~
## $ end_lat            <dbl> 41.89000, 41.89000, 41.94000, 41.88000, 41.94000, 4~
## $ end_lng            <dbl> -87.65000, -87.65000, -87.65000, -87.64000, -87.640~
## $ member_casual      <chr> "casual", "casual", "casual", "casual", "casual", "~
```

Column names for all the dataframes are consistent and are in correct data format. Now, combining 12 dataframes into one large dataframe

```r
trip_data <- rbind(jul23,aug23,sep23,oct23,nov23,dec23,jan24,feb24,mar24,apr24,may24,jun24)

# Removing individual dataframes
rm(jul23,aug23,sep23,oct23,nov23,dec23,jan24,feb24,mar24,apr24,may24,jun24)
```

Check the structure and get a glimpse of the consolidated data

```r
glimpse(trip_data)
```

```
## Rows: 5,734,381
## Columns: 13
## $ ride_id            <chr> "9340B064F0AEE130", "D1460EE3CE0D8AF8", "DF41BE31B8~
## $ rideable_type      <chr> "electric_bike", "classic_bike", "classic_bike", "e~
## $ started_at         <dttm> 2023-07-23 20:06:14, 2023-07-23 17:05:07, 2023-07-~
## $ ended_at           <dttm> 2023-07-23 20:22:44, 2023-07-23 17:18:37, 2023-07-~
## $ start_station_name <chr> "Kedzie Ave & 110th St", "Western Ave & Walton St",~
## $ start_station_id   <chr> "20204", "KA1504000103", "KA1504000103", "13155", "~
## $ end_station_name   <chr> "Public Rack - Racine Ave & 109th Pl", "Milwaukee A~
## $ end_station_id     <chr> "877", "13033", "TA1305000041", "TA1305000032", "TA~
## $ start_lat          <dbl> 41.69241, 41.89842, 41.89842, 41.88411, 41.96709, 4~
## $ start_lng          <dbl> -87.70091, -87.68660, -87.68660, -87.65694, -87.667~
## $ end_lat            <dbl> 41.69483, 41.89158, 41.90940, 41.88275, 41.96398, 4~
## $ end_lng            <dbl> -87.65304, -87.64838, -87.67769, -87.64119, -87.638~
## $ member_casual      <chr> "member", "member", "member", "member", "member", "~
```

```r
str(trip_data,give.attr = FALSE)
```

```
## spc_tbl_ [5,734,381 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5734381] "9340B064F0AEE130" "D1460EE3CE0D8AF8" "DF41BE31B895A25E" "962
##  $ rideable_type     : chr [1:5734381] "electric_bike" "classic_bike" "classic_bike" "electric_bike"
```

```
##  $ started_at        : POSIXct[1:5734381], format: "2023-07-23 20:06:14" "2023-07-23 17:05:07" ...
##  $ ended_at          : POSIXct[1:5734381], format: "2023-07-23 20:22:44" "2023-07-23 17:18:37" ...
##  $ start_station_name: chr [1:5734381] "Kedzie Ave & 110th St" "Western Ave & Walton St" "Western Ave
##  $ start_station_id  : chr [1:5734381] "20204" "KA1504000103" "KA1504000103" "13155" ...
##  $ end_station_name  : chr [1:5734381] "Public Rack - Racine Ave & 109th Pl" "Milwaukee Ave & Grand A
##  $ end_station_id    : chr [1:5734381] "877" "13033" "TA1305000041" "TA1305000032" ...
##  $ start_lat         : num [1:5734381] 41.7 41.9 41.9 41.9 42 ...
##  $ start_lng         : num [1:5734381] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:5734381] 41.7 41.9 41.9 41.9 42 ...
##  $ end_lng           : num [1:5734381] -87.7 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:5734381] "member" "member" "member" "member" ...
```

```r
head(trip_data)
```

```
## # A tibble: 6 x 13
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 9340B064F0AEE130 electric_bike 2023-07-23 20:06:14 2023-07-23 20:22:44
## 2 D1460EE3CE0D8AF8 classic_bike  2023-07-23 17:05:07 2023-07-23 17:18:37
## 3 DF41BE31B895A25E classic_bike  2023-07-23 10:14:53 2023-07-23 10:24:29
## 4 9624A293749EF703 electric_bike 2023-07-21 08:27:44 2023-07-21 08:32:40
## 5 2F68A6A4CDB4C99A classic_bike  2023-07-08 15:46:42 2023-07-08 15:58:08
## 6 9AEE973E6B941A9C classic_bike  2023-07-10 08:44:47 2023-07-10 08:49:41
## # i 9 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>
```

```r
# Check rideable_type and member_casual column for any discrepancies

unique(trip_data$rideable_type)
```

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

```r
unique(trip_data$member_casual)
```

```
## [1] "member" "casual"
```

```r
# Check for duplicates in ride_id
sum(duplicated(trip_data$ride_id))
```

```
## [1] 211
```

```r
# Removing duplicates
trip_data <- trip_data[!duplicated(trip_data$ride_id),]
```

```r
# Check for NA's
sum(is.na(trip_data))
```

```
## [1] 3842670
```

```r
# Removing NA's
trip_data <- drop_na(trip_data)
```

```r
# Checking for test stations
unique(trip_data$start_station_name[grep("TEST", trip_data$start_station_name)])
```

```
## character(0)
```

```r
unique(trip_data$start_station_name[grep("Test", trip_data$start_station_name)])
```

```
## character(0)
```

Adding column for Month, Year, day of week, hour of ride, ride length, ride distance

```r
# ride length
trip_data$ride_length <- round(difftime(trip_data$ended_at, trip_data$started_at, units = "mins"),1)

# Date, Month, Year, Day of Week, Hour of ride
trip_data <- trip_data %>%
  mutate(month = format(as.Date(started_at), "%B")) %>%
  mutate(year = format(as.Date(started_at), "%Y")) %>%
  mutate(day_of_week = format(as.Date(started_at), "%A")) %>%
  mutate(hour = format(as_datetime(started_at), "%H"))

# Calculate the ride distance in Kms
trip_data <- trip_data %>%
  rowwise() %>%
  mutate(ride_distance = distGeo(c(start_lng,start_lat), c(end_lng,end_lat))/ 1000)

# Convert ride length to numeric
trip_data$ride_length <- as.numeric(trip_data$ride_length)
is.numeric(trip_data$ride_length)
```

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

```r
# Removing the trips where ride_length <= 0 or more than 24hrs (24*60= 1440 mins)
trip_data <- trip_data[!(trip_data$ride_length <= 0 | trip_data$ride_length> 1440),]
```

Step 4: Analyze

Now, all the required information are in one place to begin our analysis phase

```r
# Lets first check our cleaned dataframe
str(trip_data, give.attr = FALSE)
```

```
## rowws_df [4,266,629 x 19] (S3: rowwise_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:4266629] "9340B064F0AEE130" "D1460EE3CE0D8AF8" "DF41BE31B895A25E" "962
##  $ rideable_type     : chr [1:4266629] "electric_bike" "classic_bike" "classic_bike" "electric_bike"
##  $ started_at        : POSIXct[1:4266629], format: "2023-07-23 20:06:14" "2023-07-23 17:05:07" ...
```

```
##  $ ended_at          : POSIXct[1:4266629], format: "2023-07-23 20:22:44" "2023-07-23 17:18:37" ...
##  $ start_station_name: chr [1:4266629] "Kedzie Ave & 110th St" "Western Ave & Walton St" "Western Ave
##  $ start_station_id  : chr [1:4266629] "20204" "KA1504000103" "KA1504000103" "13155" ...
##  $ end_station_name  : chr [1:4266629] "Public Rack - Racine Ave & 109th Pl" "Milwaukee Ave & Grand A
##  $ end_station_id    : chr [1:4266629] "877" "13033" "TA1305000041" "TA1305000032" ...
##  $ start_lat         : num [1:4266629] 41.7 41.9 41.9 41.9 42 ...
##  $ start_lng         : num [1:4266629] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:4266629] 41.7 41.9 41.9 41.9 42 ...
##  $ end_lng           : num [1:4266629] -87.7 -87.6 -87.7 -87.6 -87.6 ...
##  $ member_casual     : chr [1:4266629] "member" "member" "member" "member" ...
##  $ ride_length       : num [1:4266629] 16.5 13.5 9.6 4.9 11.4 4.9 7 6 11 11.9 ...
##  $ month             : chr [1:4266629] "July" "July" "July" "July" ...
##  $ year              : chr [1:4266629] "2023" "2023" "2023" "2023" ...
##  $ day_of_week       : chr [1:4266629] "Sunday" "Sunday" "Sunday" "Friday" ...
##  $ hour              : chr [1:4266629] "20" "17" "10" "08" ...
##  $ ride_distance     : num [1:4266629] 3.99 3.26 1.43 1.32 2.44 ...
```

```
head(trip_data)
```

```
## # A tibble: 6 x 19
## # Rowwise:
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 1 9340B064F0AEE130 electric_bike 2023-07-23 20:06:14 2023-07-23 20:22:44
## 2 D1460EE3CE0D8AF8 classic_bike  2023-07-23 17:05:07 2023-07-23 17:18:37
## 3 DF41BE31B895A25E classic_bike  2023-07-23 10:14:53 2023-07-23 10:24:29
## 4 9624A293749EF703 electric_bike 2023-07-21 08:27:44 2023-07-21 08:32:40
## 5 2F68A6A4CDB4C99A classic_bike  2023-07-08 15:46:42 2023-07-08 15:58:08
## 6 9AEE973E6B941A9C classic_bike  2023-07-10 08:44:47 2023-07-10 08:49:41
## # i 15 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>,
## #   ride_length <dbl>, month <chr>, year <chr>, day_of_week <chr>, hour <chr>,
## #   ride_distance <dbl>
```

```
summary(trip_data)
```

```
##    ride_id          rideable_type        started_at
##  Length:4266629     Length:4266629     Min.   :2023-07-01 00:00:00.00
##  Class :character   Class :character   1st Qu.:2023-08-27 09:49:48.00
##  Mode  :character   Mode  :character   Median :2023-11-07 06:46:25.00
##                                        Mean   :2023-12-14 08:20:35.01
##                                        3rd Qu.:2024-04-18 16:11:22.00
##                                        Max.   :2024-06-30 23:54:52.22
##     ended_at                      start_station_name start_station_id
##  Min.   :2023-07-01 00:03:30.00   Length:4266629     Length:4266629
##  1st Qu.:2023-08-27 10:06:55.00   Class :character   Class :character
##  Median :2023-11-07 06:55:22.00   Mode  :character   Mode  :character
##  Mean   :2023-12-14 08:37:03.69
##  3rd Qu.:2024-04-18 16:24:53.00
##  Max.   :2024-06-30 23:59:57.93
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:4266629     Length:4266629     Min.   :41.65   Min.   :-87.84
```

```
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64
##                                        Mean   :41.90   Mean   :-87.64
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.06   Max.   :-87.53
##     end_lat         end_lng       member_casual       ride_length
##  Min.   : 0.00   Min.   :-87.84   Length:4266629     Min.   :   0.10
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   1st Qu.:   5.80
##  Median :41.90   Median :-87.64   Mode  :character   Median :  10.10
##  Mean   :41.90   Mean   :-87.64                      Mean   :  16.48
##  3rd Qu.:41.93   3rd Qu.:-87.63                      3rd Qu.:  18.00
##  Max.   :42.06   Max.   :  0.00                      Max.   :1439.90
##     month               year          day_of_week           hour
##  Length:4266629     Length:4266629     Length:4266629     Length:4266629
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##  ride_distance
##  Min.   :   0.000
##  1st Qu.:   0.872
##  Median :   1.532
##  Mean   :   2.081
##  3rd Qu.:   2.696
##  Max.   :9814.574
```

Conduct descriptive analysis: Compare Members vs casual on total number of rides taken

```
trip_data %>%
  group_by(member_casual) %>%
  summarise(no_of_rides = n(), ride_percentage = (no_of_rides = n()/ nrow(trip_data))*100)
```

```
## # A tibble: 2 x 3
##   member_casual no_of_rides ride_percentage
##   <chr>               <int>           <dbl>
## 1 casual            1501566            35.2
## 2 member            2765063            64.8
```

Descriptive analysis of ride length(Min,Max,Median,Mean)

```
# min = shortest ride in mins
aggregate(trip_data$ride_length ~ trip_data$member_casual, FUN = min)
```

```
##   trip_data$member_casual trip_data$ride_length
## 1                  casual                   0.1
## 2                  member                   0.1
```

```
# max = longest ride in mins
aggregate(trip_data$ride_length ~ trip_data$member_casual, FUN = max)
```

```
##   trip_data$member_casual trip_data$ride_length
## 1                  casual                1439.8
## 2                  member                1439.9
```

```r
# median = midpoint of ride length array in asc order
aggregate(trip_data$ride_length ~ trip_data$member_casual, FUN = median)
```

```
##   trip_data$member_casual trip_data$ride_length
## 1                  casual                  13.3
## 2                  member                   8.8
```

```r
# mean= straight average of ride length in mins
aggregate(trip_data$ride_length ~ trip_data$member_casual, FUN = mean)
```

```
##   trip_data$member_casual trip_data$ride_length
## 1                  casual              23.81375
## 2                  member              12.49406
```

Next, we need to compare the average ride length and number of rides takes by members and casuals on each day of the week

```r
# We need to order the day of week first
trip_data$day_of_week <- ordered(trip_data$day_of_week,
                                 levels = c("Monday", "Tuesday", "Wednesday",
                                            "Thursday", "Friday", "Saturday", "Sunday"))

# Compare the average ride length and number of rides takes by members and casuals
#on each day of the week
trip_data %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  arrange(day_of_week)
```

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

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual day_of_week average_ride_length no_of_rides_taken
##    <chr>         <ord>                     <dbl>             <int>
##  1 casual        Monday                     23.4            175344
##  2 member        Monday                     11.9            394317
##  3 casual        Tuesday                    21.2            173487
##  4 member        Tuesday                    12.1            436589
##  5 casual        Wednesday                  20.8            177964
##  6 member        Wednesday                  12.0            447099
##  7 casual        Thursday                   20.3            182505
##  8 member        Thursday                   11.8            440136
##  9 casual        Friday                     22.9            213159
## 10 member        Friday                     12.2            384624
## 11 casual        Saturday                   26.7            311072
## 12 member        Saturday                   13.9            349157
## 13 casual        Sunday                     27.6            268035
## 14 member        Sunday                     14.1            313141
```

Next, we need to compare the average ride length and number of rides taken by members and casuals on each Month and Year

```
# We need to order the month first
trip_data$month <- ordered(trip_data$month,
                           levels = c("July","August","September","October",
                                      "November","December","January","February",
                                      "March","April","May","June"))

# Average ride length and number of rides taken each month
trip_data %>%
  group_by(member_casual, month) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  arrange(month)
```

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

```
## # A tibble: 24 x 4
## # Groups:   member_casual [2]
##    member_casual month     average_ride_length no_of_rides_taken
##    <chr>         <ord>                   <dbl>             <int>
##  1 casual        July                     25.2            244938
##  2 member        July                     13.4            327914
##  3 casual        August                   24.2            233533
##  4 member        August                   13.3            350270
##  5 casual        September                23.5            196698
##  6 member        September                12.7            308998
##  7 casual        October                  21.3            130112
##  8 member        October                  11.7            272901
##  9 casual        November                 17.8             71972
## 10 member        November                 11.1            202293
## # i 14 more rows
```

```
# Average ride length and number of rides taken each Year
trip_data %>%
  group_by(member_casual, year) %>%
  summarise(no_of_rides_taken = n()) %>%
  arrange(year)
```

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

```
## # A tibble: 4 x 3
## # Groups:   member_casual [2]
##   member_casual year  no_of_rides_taken
##   <chr>         <chr>             <int>
## 1 casual        2023             913908
## 2 member        2023            1592588
## 3 casual        2024             587658
## 4 member        2024            1172475
```

Next, we need to compare the number of rides takes by members and casuals with each type of bike

```
# Compare the average ride length and number of rides
#takes by members and casuals with each type of bike
trip_data %>%
  group_by(member_casual, rideable_type) %>%
  summarise(no_of_rides_taken = n()) %>%
  arrange(rideable_type,member_casual)
```

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

```
## # A tibble: 5 x 3
## # Groups:   member_casual [2]
##   member_casual rideable_type no_of_rides_taken
##   <chr>         <chr>                     <int>
## 1 casual        classic_bike             957896
## 2 member        classic_bike            1884025
## 3 casual        docked_bike               33261
## 4 casual        electric_bike            510409
## 5 member        electric_bike            881038
```

Comparison between Members and Casual riders depending on ride distance

```
# Avg ride distance in kms
trip_data %>%
  group_by(member_casual) %>%
  summarise(average_ride_distance = mean(ride_distance))
```

```
## # A tibble: 2 x 2
##   member_casual average_ride_distance
##   <chr>                         <dbl>
## 1 casual                         2.12
## 2 member                         2.06
```

Analyzing the top 10 most travelled routes for Member and Casual riders

```
trip_data_v1 <- trip_data %>%
  unite("most_common_travel_route", start_station_name,end_station_name, sep = "--")

#top 10 most travelled routes for Member
r_member <- sqldf("SELECT most_common_travel_route, count (ride_id) AS total_number_of_ride
                  FROM trip_data_v1
                  WHERE member_casual = 'member'
                  GROUP BY most_common_travel_route
                  ORDER BY total_number_of_ride DESC
                  LIMIT 10")
r_member
```

```
##                         most_common_travel_route total_number_of_ride
## 1      Calumet Ave & 33rd St--State St & 33rd St                 5678
```

```
## 2      State St & 33rd St--Calumet Ave & 33rd St                5674
## 3  Ellis Ave & 60th St--University Ave & 57th St                4074
## 4  University Ave & 57th St--Ellis Ave & 60th St                4067
## 5       Ellis Ave & 60th St--Ellis Ave & 55th St                3830
## 6       Ellis Ave & 55th St--Ellis Ave & 60th St                3575
## 7  Loomis St & Lexington St--Morgan St & Polk St                2993
## 8  Morgan St & Polk St--Loomis St & Lexington St                2797
## 9        MLK Jr Dr & 29th St--State St & 33rd St                2396
## 10       State St & 33rd St--MLK Jr Dr & 29th St                2262
```

```
#top 10 most travelled routes for Casual riders
r_casual <- sqldf("SELECT most_common_travel_route, count (ride_id) AS total_number_of_ride
                  FROM trip_data_v1
                  WHERE member_casual = 'casual'
                  GROUP BY most_common_travel_route
                  ORDER BY total_number_of_ride DESC
                  LIMIT 10")
r_casual
```

```
##                                                 most_common_travel_route
## 1                      Streeter Dr & Grand Ave--Streeter Dr & Grand Ave
## 2  DuSable Lake Shore Dr & Monroe St--DuSable Lake Shore Dr & Monroe St
## 3            DuSable Lake Shore Dr & Monroe St--Streeter Dr & Grand Ave
## 4                         Michigan Ave & Oak St--Michigan Ave & Oak St
## 5                                    Millennium Park--Millennium Park
## 6                                      Dusable Harbor--Dusable Harbor
## 7                                    Montrose Harbor--Montrose Harbor
## 8            Streeter Dr & Grand Ave--DuSable Lake Shore Dr & Monroe St
## 9                               Dusable Harbor--Streeter Dr & Grand Ave
## 10                                 Adler Planetarium--Adler Planetarium
##    total_number_of_ride
## 1                  8705
## 2                  7294
## 3                  4807
## 4                  4378
## 5                  3437
## 6                  3015
## 7                  2624
## 8                  2487
## 9                  2199
## 10                 2182
```

Step 5: Share

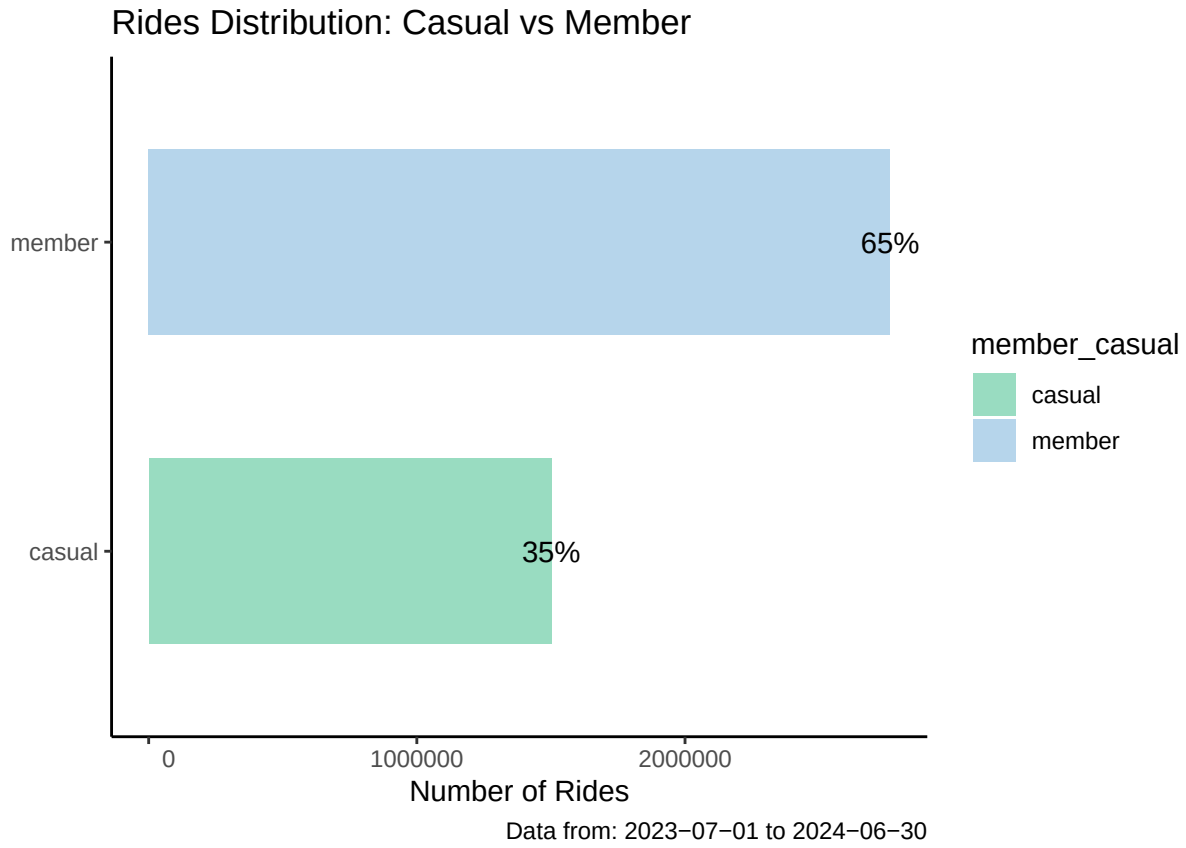
In this step we will visualize the trends and relationship between different variables

Viz 1- Total number of rides taken by member and casual riders

```
mindate<- as.Date(min(trip_data$started_at))
maxdate<- as.Date(max(trip_data$started_at))

# Rides Distribution: Casual vs Member
trip_data %>%
```

```
  group_by(member_casual) %>%
  summarise(no_of_rides = n(),
            ride_percentage = (no_of_rides = n()/ nrow(trip_data))*100) %>%
  ggplot() + geom_col(mapping = aes(x= member_casual,
                                    y = no_of_rides,
                                    fill = member_casual),width=0.6)+
  theme_classic()+
  scale_fill_manual(values = c("casual"= "#99dcc1", "member"= "#B6D5EB"))+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  labs(x= '', y= 'Number of Rides' ,
       title = 'Rides Distribution: Casual vs Member',
  caption = paste("Data from:", mindate, "to", maxdate)) +
  geom_text(aes(x= member_casual, y = no_of_rides,
                label = percent(ride_percentage/100), hjust= 0.5))+
   coord_flip()
```



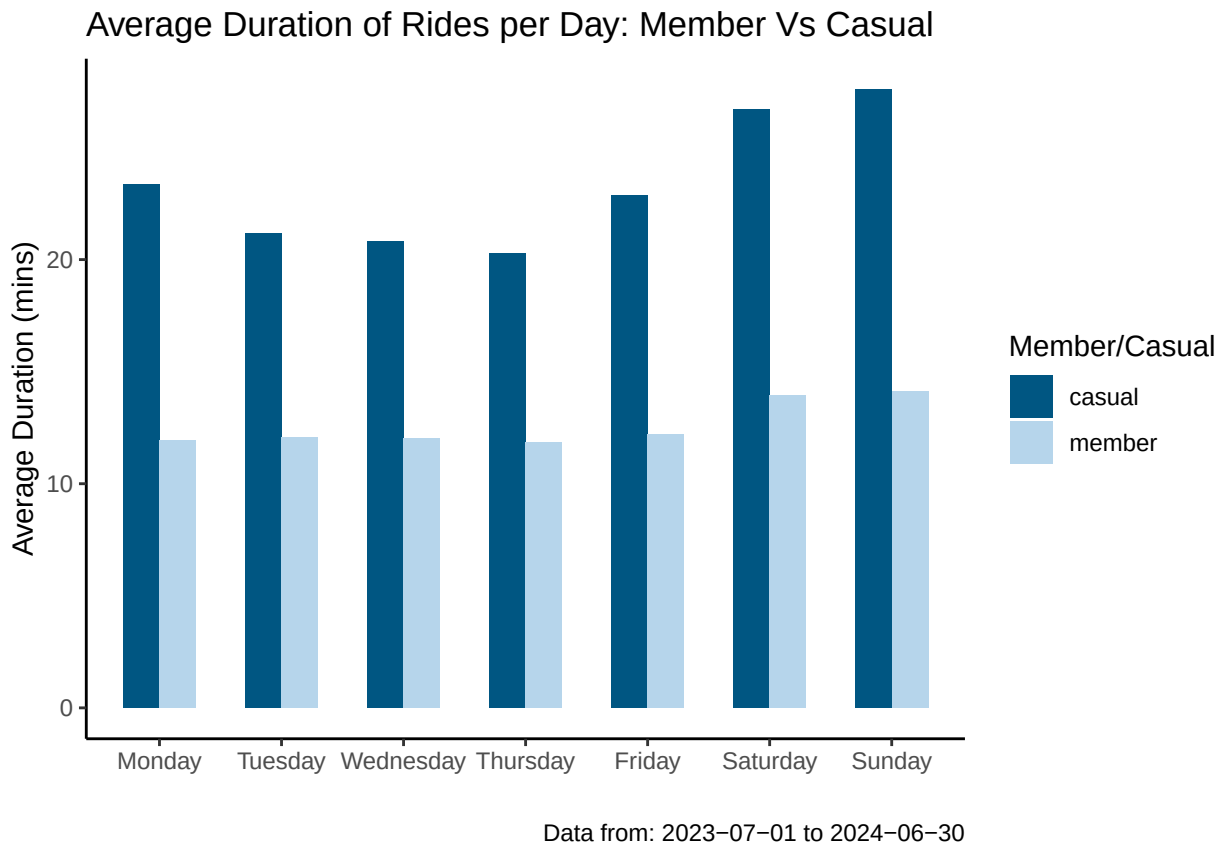
- We can see on the member vs casual ride distribution chart, 65% of the rides are taken by members and 35% of the rides were taken by casual rider. So its clearly visible that in July 23 to June 24, members used ride share ~30% more than the casual riders

Viz 2 & 3- Average ride duration and total number of rides taken per day between member and casual riders

```
# Average ride length per day
trip_data %>%
```

```
  group_by(member_casual, day_of_week) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  ggplot() + geom_col(mapping = aes(x= day_of_week, y= average_ride_length,
                                    fill= member_casual),position = "dodge",
                      width = 0.6)+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= '', y= 'Average Duration (mins)' ,
       title = 'Average Duration of Rides per Day: Member Vs Casual',
  caption = paste("Data from:", mindate, "to", maxdate), fill= "Member/Casual")
```

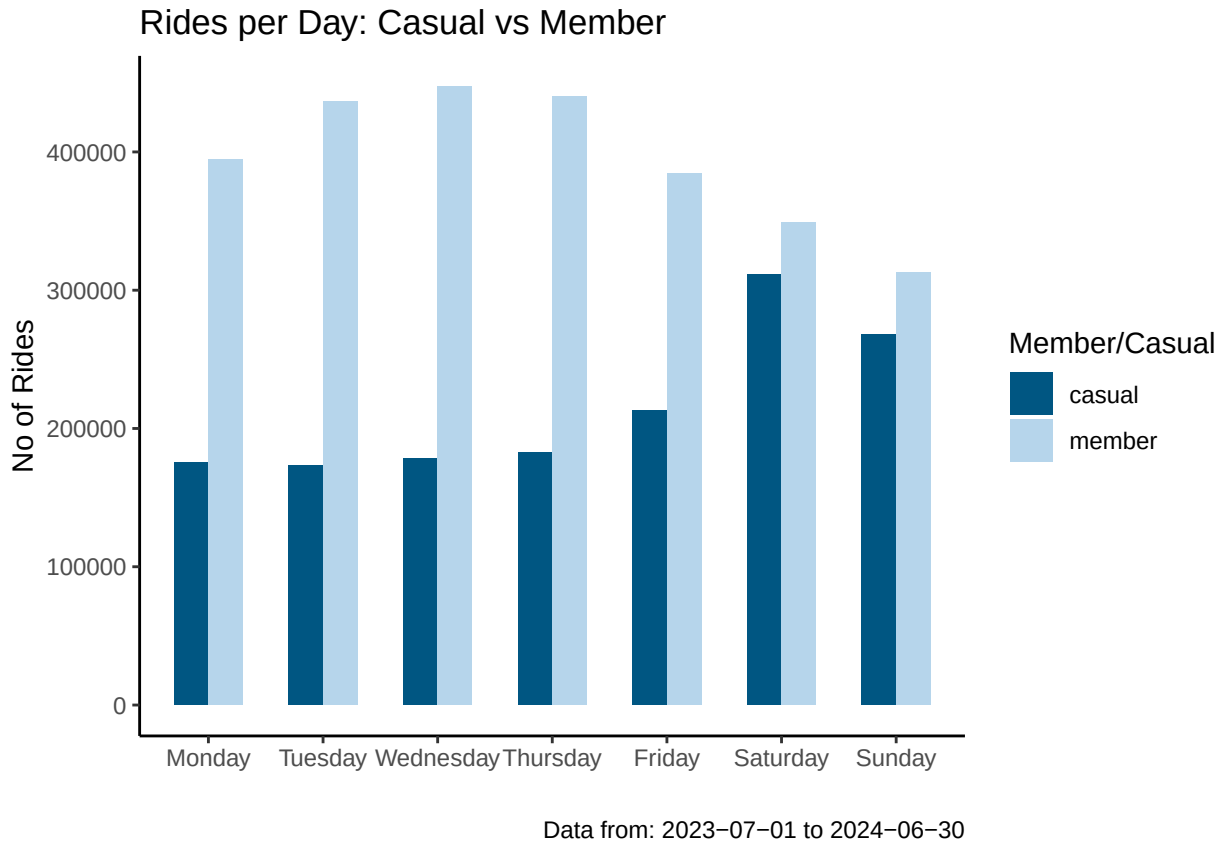
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```




```
# Number of rides taken per day
trip_data %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  ggplot() + geom_col(mapping = aes(x= day_of_week, y= no_of_rides_taken,
                                    fill= member_casual),position = "dodge", width = 0.6)+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= '', y= 'No of Rides' , title = 'Rides per Day: Casual vs Member',
  caption = paste("Data from:", mindate, "to", maxdate), fill= "Member/Casual")+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
```

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

```
## `.groups` argument.
```

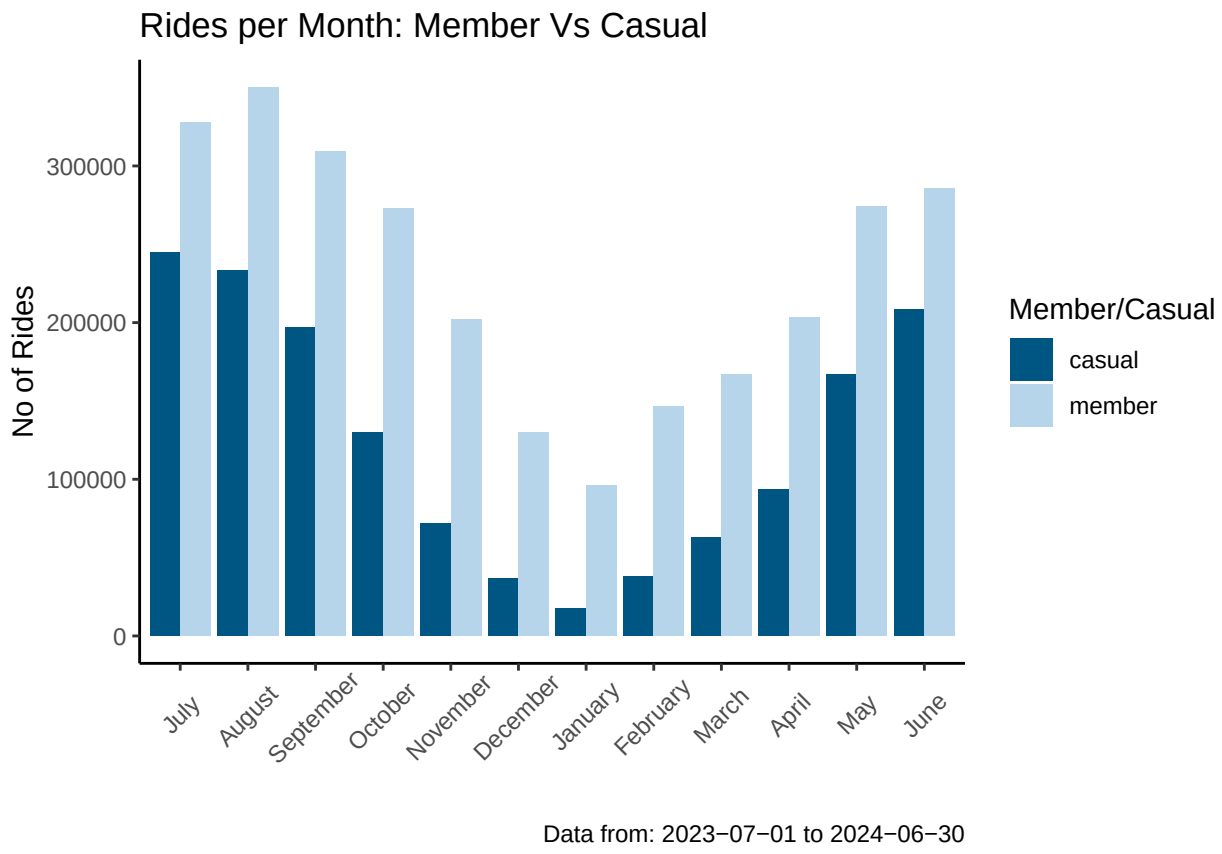


- From the first chart above, average ride duration for casual riders are much more than the members. It can be seen weekend rides has more duration is higher for casual riders. It is also seen a consistent ride duration for the member through the week (having less than 15 mins)
- Members took more rides on weekdays, starting a decline on weekends. Whereas, the casual riders took rides more on Friday, Saturday & Sunday than the other days.

Viz 4 & 5- Average ride duration and total number of rides taken each month between member and casual riders

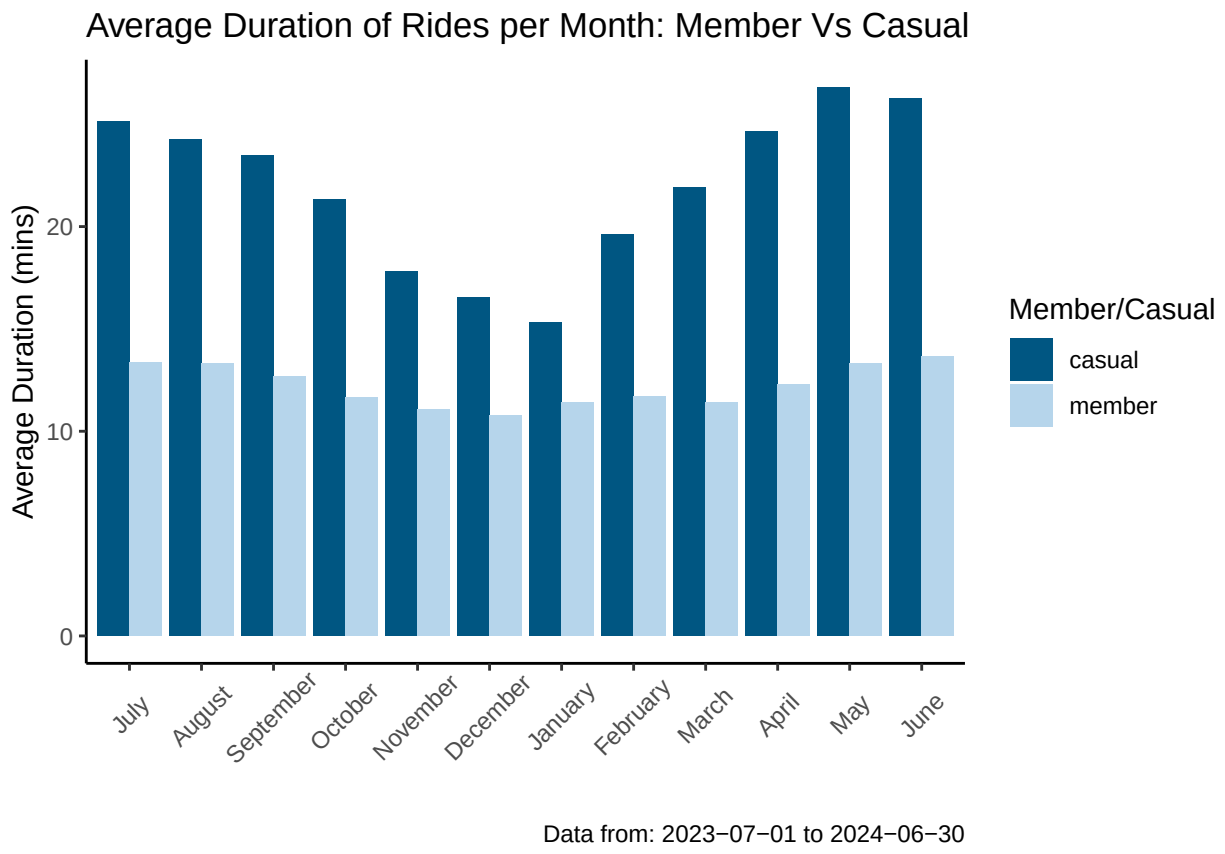
```
# Number of rides taken each month
trip_data %>%
  group_by(member_casual, month) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  ggplot() + geom_col(mapping = aes(x= month, y= no_of_rides_taken,
                                    fill = member_casual), position = "dodge")+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  theme(axis.text.x = element_text(angle = 45, hjust = 0.55, vjust = 0.65))+
  labs(x= "", y= "No of Rides",
       title = "Rides per Month: Member Vs Casual",
       fill= "Member/Casual", caption =  paste("Data from:", mindate, "to", maxdate))+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
```

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




```
# Average ride length each month
trip_data %>%
  group_by(member_casual, month) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  ggplot() + geom_col(mapping = aes(x= month, y= average_ride_length,
                                    fill = member_casual), position = "dodge")+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  theme(axis.text.x = element_text(angle = 45, hjust = 0.55, vjust = 0.65))+
  labs(x= "", y= "Average Duration (mins)",
       title = "Average Duration of Rides per Month: Member Vs Casual",
       fill= "Member/Casual", caption = paste("Data from:", mindate, "to", maxdate))
```

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

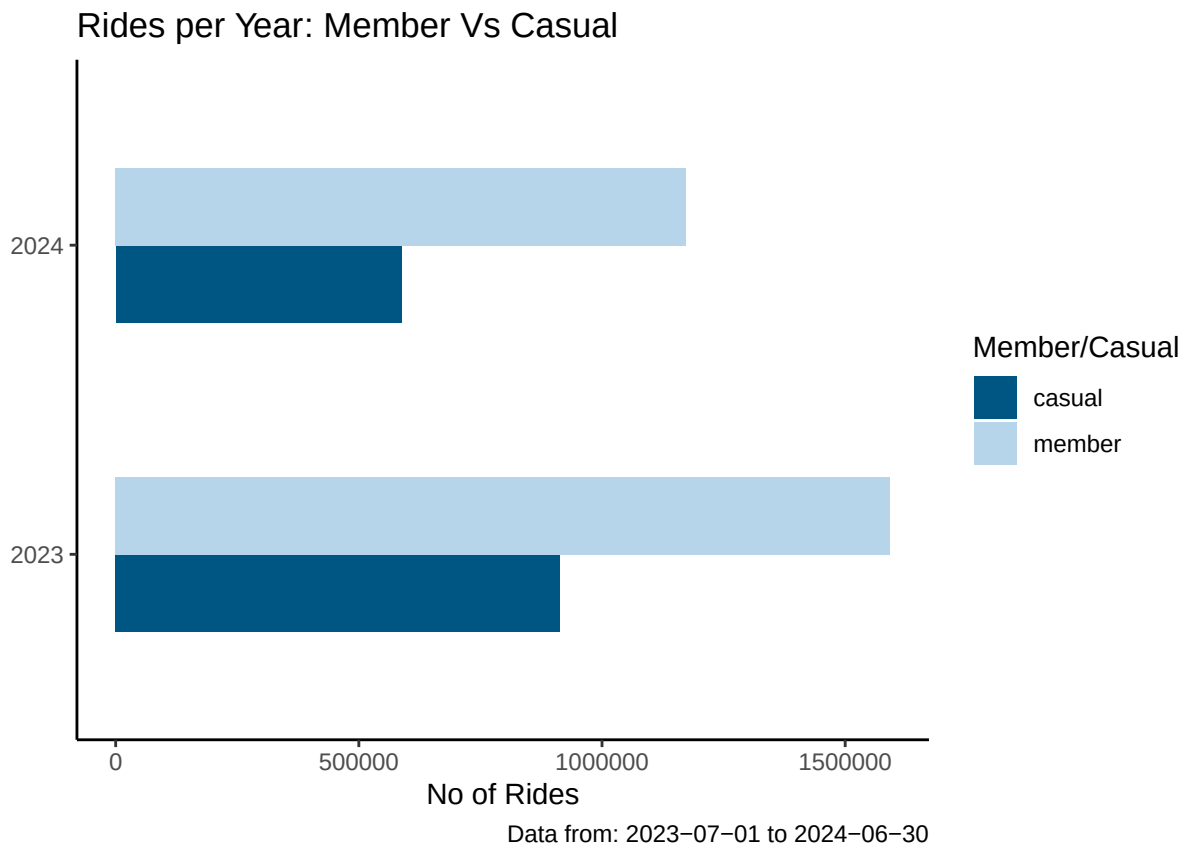


- From the first viz, we can see the number of rides taken by both is decreasing from July to Jan and increasing from Feb to June, with member have more rides than casual riders. It is possible due to winter there is significant drop in the rides.
- Members still have a ride duration less than 15 mins. The trend of the chart is similar to the previous chart.

Viz 6- Total number of rides taken per year between member and casual riders

```
# Number of rides taken each Year
trip_data %>%
  group_by(member_casual, year) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  arrange(year) %>%
  ggplot() + geom_col(mapping = aes(x= year, y= no_of_rides_taken,
                                    fill = member_casual), width = 0.5, position = "dodge")+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= "", y= "No of Rides", title = "Rides per Year: Member Vs Casual",
       fill= "Member/Casual", caption = paste("Data from:", mindate, "to", maxdate))+
  coord_flip()
```

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

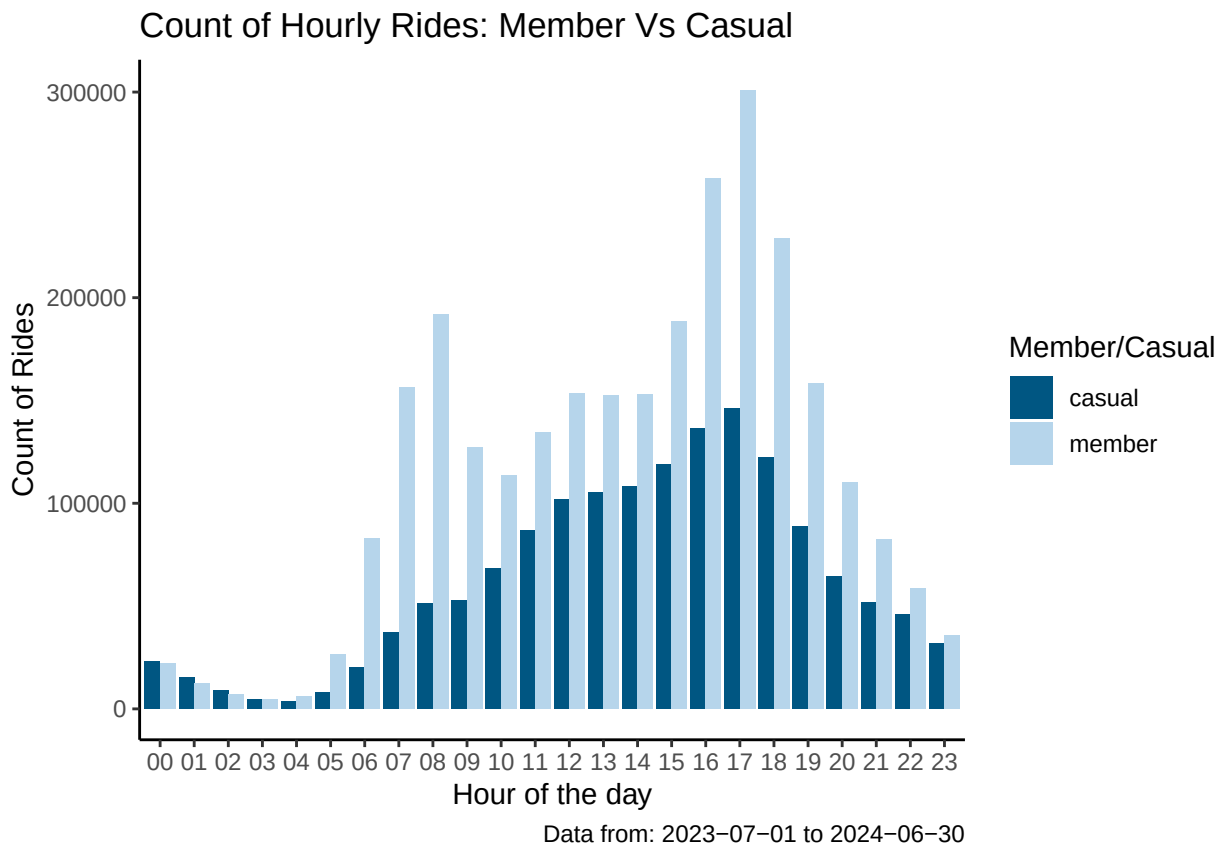


- Overall rides taken in 2023 was more than the total number of rides taken in 2024, with member having more ride share than casual riders.

Viz 7- Total number of rides taken per hour between member and casual riders

```
# Count of Hourly Rides
trip_data %>%
  group_by(member_casual, hour) %>%
  summarise(average_ride_length = mean(ride_length), no_of_rides_taken = n()) %>%
  ggplot() + geom_col(mapping = aes(x= hour, y= no_of_rides_taken, fill = member_casual), position = "d
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= "Hour of the day", y= "Count of Rides",
       title = "Count of Hourly Rides: Member Vs Casual", fill= "Member/Casual",
       caption = paste("Data from:", mindate, "to", maxdate))+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
```

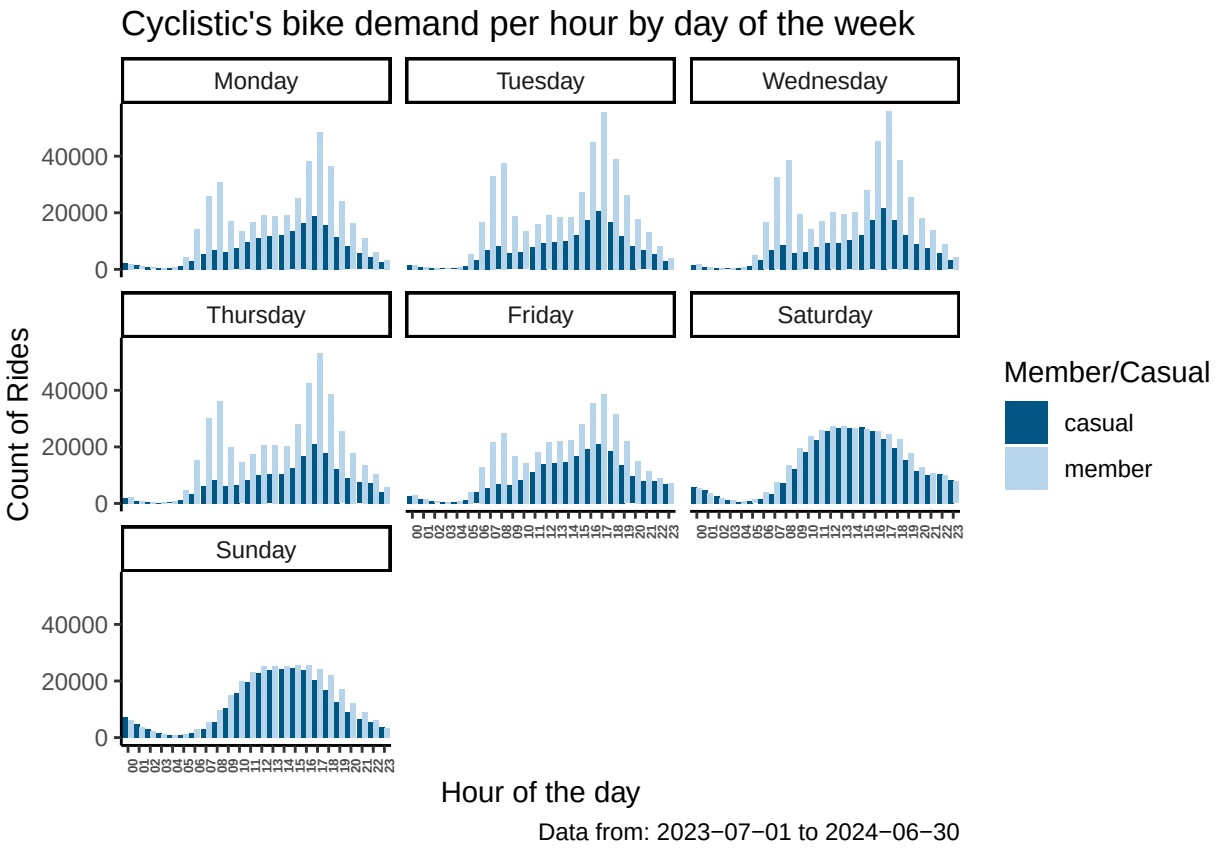
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



- From the above graph, we can see that the number of rides taken increases from 10am to 5pm for casual riders. Also there is a bigger volume rise from 3pm to 6pm for the member. This need to check further on daily basis.

Viz 8- Hourly ride distribution each weekday between member and casual riders

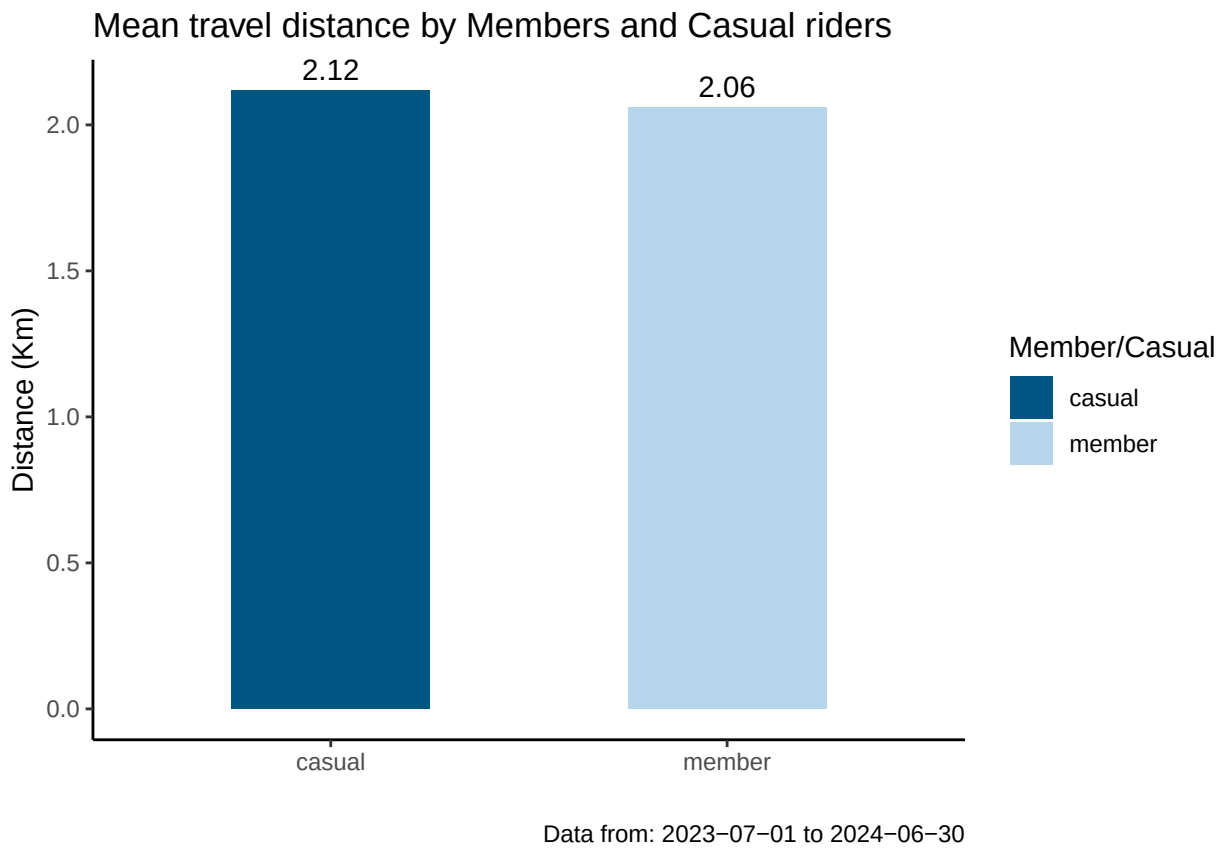
```
#Analysis and visualization on cyclistic's bike demand per hour by day of the week
trip_data %>%
  ggplot() + geom_bar(mapping = aes(hour, fill = member_casual), position = "dodge")+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= "Hour of the day", y= "Count of Rides",
       title = "Cyclistic's bike demand per hour by day of the week", fill= "Member/Casual",
       caption = paste("Data from:", mindate, "to", maxdate))+
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))+
  facet_wrap(~day_of_week)+ theme(axis.text.x = element_text(angle = 90, size = 5, face = "bold"))
```



- There is alot of difference between the weekdays and weekends. A big volume of rides increase on weekdays from 6am to 9am and another big volume increase from 4pm to 7pm. This can be due to riders going to office in the morning and returning back at the evening. Whereas, on weekends most of the rides are taken between 9am to 6pm, from this we can hypothesize that both the riders use their bike share for leisure purpose on weekends.

Viz 9- Comparison between Members and Casual riders depending on ride distance

```
# Avg ride distance in kms
trip_data %>%
  group_by(member_casual) %>%
  summarise(average_ride_distance = mean(ride_distance)) %>%
   ggplot() + geom_col(mapping = aes(x= member_casual, y= average_ride_distance,
                                     fill = member_casual), width = 0.5, position = "dodge")+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= "", y= "Distance (Km)",
       title = "Mean travel distance by Members and Casual riders",
       fill= "Member/Casual", caption = paste("Data from:", mindate, "to", maxdate))+
  geom_text(aes(x= member_casual, y = average_ride_distance,
                label = round(average_ride_distance,2), vjust = -0.5))
```

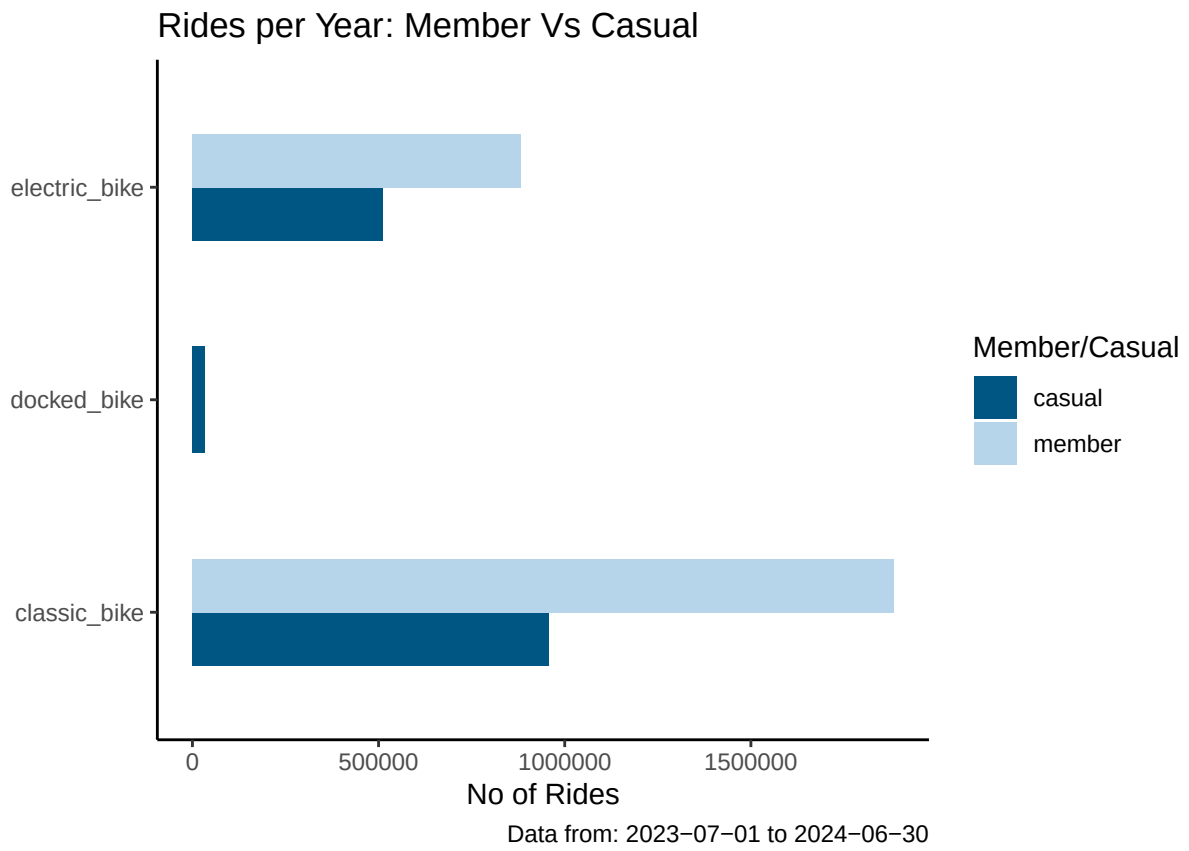


- No major difference in distance can be seen between members and casual riders.

Viz 10- Usage of different bike by rider type

```
trip_data %>%
  group_by(member_casual,rideable_type) %>%
  summarise(no_of_rides_taken = n()) %>%
  arrange(rideable_type,member_casual) %>%
  ggplot() + geom_col(mapping = aes(x= rideable_type, y= no_of_rides_taken,
                                    fill = member_casual), width = 0.5, position = "dodge")+
  theme_classic()+ scale_fill_manual(values = c("casual"= "#005682", "member"= "#B6D5EB"))+
  labs(x= "", y= "No of Rides", title = "Rides per Year: Member Vs Casual",
       fill= "Member/Casual", caption = paste("Data from:", mindate, "to", maxdate))+
  coord_flip()
```

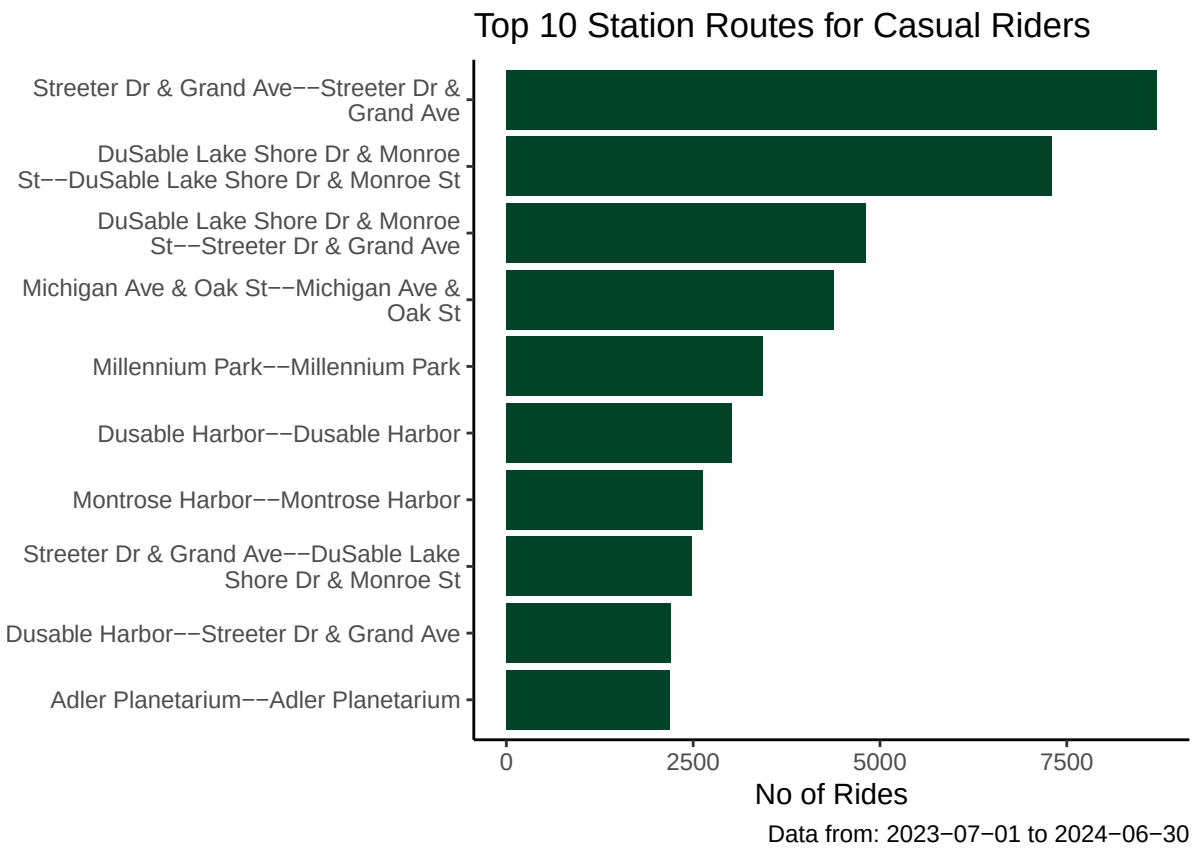
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```



- From the above chart we can see, members and casual riders mostly use classic bikes, followed by electric bikes. Docked bikes are only used by casual riders.

Viz 11- Top 10 Station routes taken by Casual Riders

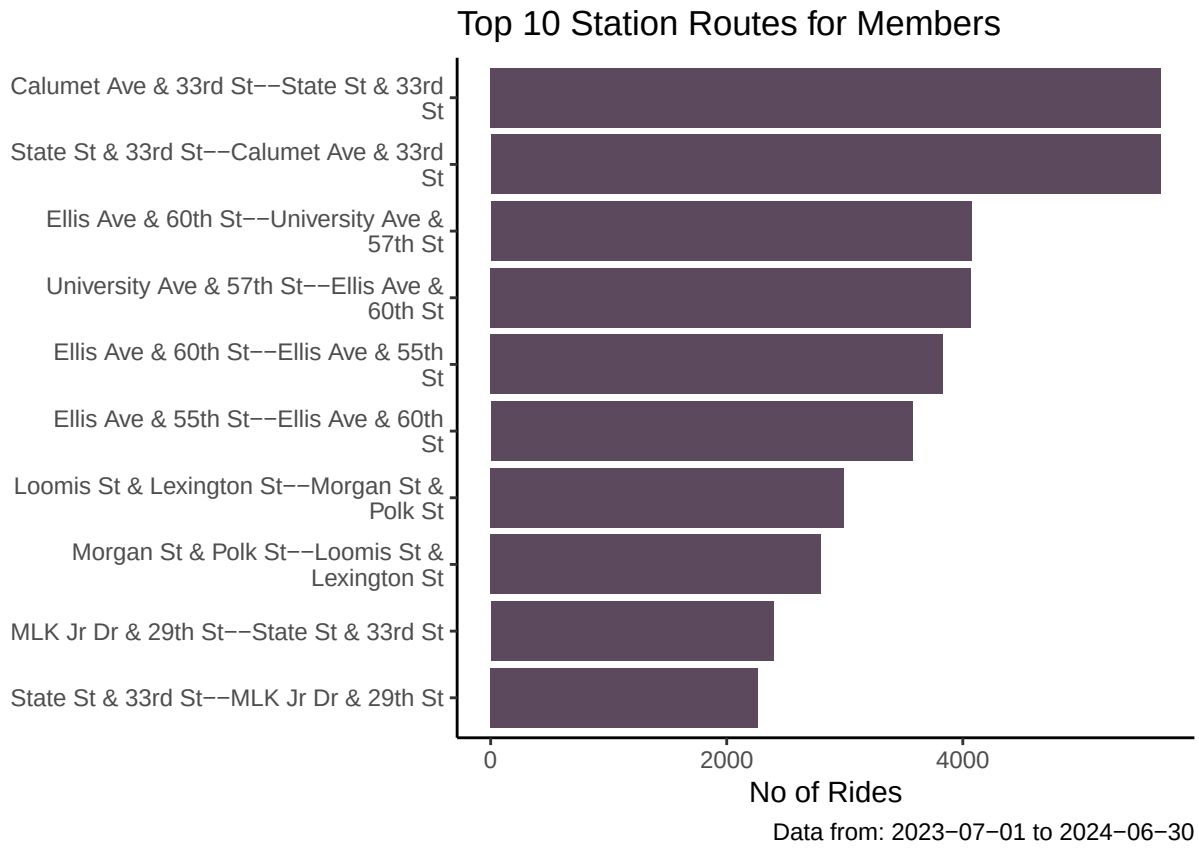
```
# top 10 routes for casual riders
ggplot(r_casual)+ geom_col(mapping = aes(x= reorder(most_common_travel_route,total_number_of_ride),
                                         y= total_number_of_ride),fill = "#004328")+
  coord_flip()+ theme_classic()+
  labs(x= "", y= "No of Rides", title = "Top 10 Station Routes for Casual Riders",
       fill= "Member/Casual", caption = paste("Data from:", mindate, "to", maxdate))+
  scale_x_discrete(labels = wrap_format(40))
```



- From the above viz, Streeter Dr & Grand Ave–Streeter Dr & Grand Ave is the most popular route for the casual riders.

Viz 12- Top 10 Station routes taken by members

```
# top 10 routes for members
r_member %>%
 ggplot()+ geom_col(mapping = aes(x= reorder(most_common_travel_route,total_number_of_ride),
                                  y= total_number_of_ride),fill = "#5d495d")+
  coord_flip()+ theme_classic()+
  labs(x= "", y= "No of Rides", title = "Top 10 Station Routes for Members",
       fill= "Member/Casual", caption = paste("Data from:", mindate, "to", maxdate))+
   scale_x_discrete(labels = wrap_format(40))
```



* From the above viz, Calumet Ave & 33rd St–State St & 33rd St is the most popular route taken by members

Key findings:

- Members hold biggest proportion of total rides, ~30% more than the casual riders.
- Casual riders have more number of rides on weekends, whereas, members have more on the weekdays. The avg ride duration for casual is significantly more than the members. And avg ride time increases slightly on the weekends.
- Number of rides taken and ride duration decreases during cold season.
- We have more rides from 6am to 9am and 4pm to 7pm on weekdays. On weekdays the maximum rides are taken from 9am to 6pm.
- Top 3 routes taken by casual riders are 1) Streeter Dr & Grand Ave–Streeter Dr & Grand Ave 2) DuSable Lake Shore Dr & Monroe St–DuSable Lake Shore Dr & Monroe St 3) DuSable Lake Shore Dr & Monroe St–Streeter Dr & Grand Ave.
- Similar to members, casual riders used classic bike more frequently. Docked bikes are the least popular bike types.

Step 6: Act

Three recommendations:

- Since the members hold the biggest proportion of total rides, the marketing team can provide promotional offers and discount for Yearly and monthly memberships.
- The marketing team should give additional discounts on summer months and run campaigns from May to October, which can help converting one-time customers to loyal members and increase visibility to

attract new customers. Moreover, the team can target the stations for the most route taken by casual riders for running advertisement and campaigns.

- The company should increase the number of electric and docked bikes and reduce the price for those passes. This could benefit the company as electric bikes are already in trends. More docking stations can be introduced for more organised pick-up and dropping system for the users.