Google Data Analytics Capstone - Case Study 1

Fabian Pang

9/11/2021

Introduction

This is my attempt at the Google Data Analytics Capstone - Case Study 1. The full details to the case study can be found in the course pagelink (https://www.coursera.org/learn/google-data-analytics-capstone?specialization=google-data-analytics) (Google Data Analytics Capstone: Complete a Case Study).

This case study document will adopt the framework as suggested by the course.

The flow of the case study follows the process involving these steps: Ask, Prepare, Process, Analyse, Share and Act

Ask

To begin with, here are some background information to provide the context for this case study.

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.

Characters and Teams:

● Cyclistic: A bike-share program that features more than 5,800 bicycles and 600 docking stations. Cyclistic sets itself apart by also offering reclining bikes, hand tricycles, and cargo bikes, making bike-share more inclusive to people with disabilities and riders who can't use a standard two-wheeled bike. The majority of riders opt for traditional bikes; about 8% of riders use the assistive options. Cyclistic users are more likely to ride for leisure, but about 30% use them to commute to work each day.

● Lily Moreno: The director of marketing and your manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program. These may include email, social media, and other channels.

● Cyclistic marketing analytics team: A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy. You joined this team six months ago and have been busy learning about Cyclistic's mission and business goals — as well as how you, as a junior data analyst, can help Cyclistic achieve them.

● Cyclistic executive team: The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

Business Task

Understanding and identifying differences between casual riders and annual members in Cyclistic bike usage. These insights would then help to enhance the marketing strategies in converting casual riders to annual members.

Prepare

For this case study, the data used will be the 12-month historical ride data of Cyclistic that can be found herelink (https://divvy-tripdata.s3.amazonaws.com/index.html). This project will use historical riding data ranging from August 2020 - July 2021.

The data is organised based on its respective months into separate csv files.

As the data is provided by Cyclistic themselves regarding historical riding data of their own clients, bias and credibility issues should not be present. Cyclistic has their own license over the dataset which does not contain any personal information about their clients. The data is Reliable, Original, Current, Comprehensive and Cited.

While the data may not contain a wide range of information on the riders, the data should contain useful insights towards the bike usage of both casuals and members, which would help tackle the business task.

Process

This portion will include steps taken for data organisation and data cleaning in ensuring data integrity. Due to the large dataset, R will be utilised for this project. The steps are as documented below.

Data Organisation

Loading the relevant libraries

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(janitor)
```

```
##
## Attaching package: 'janitor'
```

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

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)

library(skimr)

library(DescTools)
```

```
## Warning: package 'DescTools' was built under R version 4.1.1
```

```
library(geosphere)
```

```
## Warning: package 'geosphere' was built under R version 4.1.1
```

```
library(ggplot2)
```

Reading the relevant files

```
trips_aug20 <- read_csv("202008-divvy-tripdata.csv")
```

```
## Rows: 622361 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## 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.
```

```
trips_sep20 <- read_csv("202009-divvy-tripdata.csv")
```

```
## Rows: 532958 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## 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.
```

```
trips_oct20 <- read_csv("202010-divvy-tripdata.csv")
```

```
## Rows: 388653 Columns: 13
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## 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.
```

```
trips_nov20 <- read_csv("202011-divvy-tripdata.csv")
```

```
## Rows: 259716 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (5): ride_id, rideable_type, start_station_name, end_station_name, memb...
## dbl  (6): start_station_id, end_station_id, start_lat, start_lng, end_lat, e...
## 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.
```

```
trips_dec20 <- read_csv("202012-divvy-tripdata.csv")
```

```
## Rows: 131573 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.
```

```
trips_jan21 <- read_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_...
## 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.
```

```
trips_feb21 <- read_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_...
## 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.
```

```
trips_mar21 <- read_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_...
## 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.
```

```
trips_apr21 <- read_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_...
## 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.
```

```
trips_may21 <- read_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_...
## 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.
```

```
trips_jun21 <- read_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_...
## 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.
```

```
trips_jul21 <- read_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_...
## 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.
```

Inspecting the data

```
str(trips_aug20)
```

```
## spec_tbl_df [622,361 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:622361] "322BD23D287743ED" "2A3AEF1AB9054D8B" "67DC1D133E8B5816" "C79FBB
D412E578A7" ...
##  $ rideable_type     : chr [1:622361] "docked_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:622361], format: "2020-08-20 18:08:14" "2020-08-27 18:46:04" ...
##  $ ended_at          : POSIXct[1:622361], format: "2020-08-20 18:17:51" "2020-08-27 19:54:51" ...
##  $ start_station_name: chr [1:622361] "Lake Shore Dr & Diversey Pkwy" "Michigan Ave & 14th St" "Columb
us Dr & Randolph St" "Daley Center Plaza" ...
##  $ start_station_id  : num [1:622361] 329 168 195 81 658 658 196 67 153 177 ...
##  $ end_station_name  : chr [1:622361] "Clark St & Lincoln Ave" "Michigan Ave & 14th St" "State St & Ra
ndolph St" "State St & Kinzie St" ...
##  $ end_station_id    : num [1:622361] 141 168 44 47 658 658 49 229 225 305 ...
##  $ start_lat         : num [1:622361] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:622361] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:622361] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:622361] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:622361] "member" "casual" "casual" "casual" ...
##  - 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_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_sep20)
```

```
## spec_tbl_df [532,958 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:532958] "2B22BD5F95FB2629" "A7FB70B4AFC6CAF2" "86057FA01BAC778E" "57F6DC
9A153DB98C" ...
##  $ rideable_type     : chr [1:532958] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:532958], format: "2020-09-17 14:27:11" "2020-09-17 15:07:31" ...
##  $ ended_at          : POSIXct[1:532958], format: "2020-09-17 14:44:24" "2020-09-17 15:07:45" ...
##  $ start_station_name: chr [1:532958] "Michigan Ave & Lake St" "W Oakdale Ave & N Broadway" "W Oakdale
Ave & N Broadway" "Ashland Ave & Belle Plaine Ave" ...
##  $ start_station_id  : num [1:532958] 52 NA NA 246 24 94 291 NA NA NA ...
##  $ end_station_name  : chr [1:532958] "Green St & Randolph St" "W Oakdale Ave & N Broadway" "W Oakdale
Ave & N Broadway" "Montrose Harbor" ...
##  $ end_station_id    : num [1:532958] 112 NA NA 249 24 NA 256 NA NA NA ...
##  $ start_lat         : num [1:532958] 41.9 41.9 41.9 42 41.9 ...
##  $ start_lng         : num [1:532958] -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:532958] 41.9 41.9 41.9 42 41.9 ...
##  $ end_lng           : num [1:532958] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:532958] "casual" "casual" "casual" "casual" ...
##  - 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_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_oct20)
```

```
## spec_tbl_df [388,653 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:388653] "ACB6B40CF5B9044C" "DF450C72FD109C01" "B6396B54A15AC0DF" "44A4AE
E261B9E854" ...
##  $ rideable_type     : chr [1:388653] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:388653], format: "2020-10-31 19:39:43" "2020-10-31 23:50:08" ...
##  $ ended_at          : POSIXct[1:388653], format: "2020-10-31 19:57:12" "2020-11-01 00:04:16" ...
##  $ start_station_name: chr [1:388653] "Lakeview Ave & Fullerton Pkwy" "Southport Ave & Waveland Ave"
"Stony Island Ave & 67th St" "Clark St & Grace St" ...
##  $ start_station_id  : num [1:388653] 313 227 102 165 190 359 313 125 NA 174 ...
##  $ end_station_name  : chr [1:388653] "Rush St & Hubbard St" "Kedzie Ave & Milwaukee Ave" "University
Ave & 57th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : num [1:388653] 125 260 423 256 185 53 125 313 199 635 ...
##  $ start_lat         : num [1:388653] 41.9 41.9 41.8 42 41.9 ...
##  $ start_lng         : num [1:388653] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:388653] 41.9 41.9 41.8 42 41.9 ...
##  $ end_lng           : num [1:388653] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:388653] "casual" "casual" "casual" "casual" ...
##  - 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_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_nov20)
```

```
## spec_tbl_df [259,716 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:259716] "BD0A6FF6FFF9B921" "96A7A7A4BDE4F82D" "C61526D06582BDC5" "E533E8
9C32080B9E" ...
##  $ rideable_type     : chr [1:259716] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:259716], format: "2020-11-01 13:36:00" "2020-11-01 10:03:26" ...
##  $ ended_at          : POSIXct[1:259716], format: "2020-11-01 13:45:40" "2020-11-01 10:14:45" ...
##  $ start_station_name: chr [1:259716] "Dearborn St & Erie St" "Franklin St & Illinois St" "Lake Shore
Dr & Monroe St" "Leavitt St & Chicago Ave" ...
##  $ start_station_id  : num [1:259716] 110 672 76 659 2 72 76 NA 58 394 ...
##  $ end_station_name  : chr [1:259716] "St. Clair St & Erie St" "Noble St & Milwaukee Ave" "Federal St
& Polk St" "Stave St & Armitage Ave" ...
##  $ end_station_id    : num [1:259716] 211 29 41 185 2 76 72 NA 288 273 ...
##  $ start_lat         : num [1:259716] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:259716] -87.6 -87.6 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:259716] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:259716] -87.6 -87.7 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr [1:259716] "casual" "casual" "casual" "casual" ...
##  - 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_double(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_double(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_dec20)
```

```
## spec_tbl_df [131,573 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:131573] "70B6A9A437D4C30D" "158A465D4E74C54A" "5262016E0F1F2F9A" "BE1196
28E44F871E" ...
##  $ rideable_type     : chr [1:131573] "classic_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:131573], format: "2020-12-27 12:44:29" "2020-12-18 17:37:15" ...
##  $ ended_at          : POSIXct[1:131573], format: "2020-12-27 12:55:06" "2020-12-18 17:44:19" ...
##  $ start_station_name: chr [1:131573] "Aberdeen St & Jackson Blvd" NA NA NA ...
##  $ start_station_id  : chr [1:131573] "13157" NA NA NA ...
##  $ end_station_name  : chr [1:131573] "Desplaines St & Kinzie St" NA NA NA ...
##  $ end_station_id    : chr [1:131573] "TA1306000003" NA NA NA ...
##  $ start_lat         : num [1:131573] 41.9 41.9 41.9 41.9 41.8 ...
##  $ start_lng         : num [1:131573] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:131573] 41.9 41.9 41.9 41.9 41.8 ...
##  $ end_lng           : num [1:131573] -87.6 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:131573] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_jan21)
```

```
## spec_tbl_df [96,834 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:96834] "E19E6F1B8D4C42ED" "DC88F20C2C55F27F" "EC45C94683FE3F27" "4FA453A
75AE377DB" ...
##  $ rideable_type     : chr [1:96834] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:96834], format: "2021-01-23 16:14:19" "2021-01-27 18:43:08" ...
##  $ ended_at          : POSIXct[1:96834], format: "2021-01-23 16:24:44" "2021-01-27 18:47:12" ...
##  $ start_station_name: chr [1:96834] "California Ave & Cortez St" "California Ave & Cortez St" "Califo
rnia Ave & Cortez St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:96834] "17660" "17660" "17660" "17660" ...
##  $ end_station_name  : chr [1:96834] NA NA NA NA ...
##  $ end_station_id    : chr [1:96834] NA NA NA NA ...
##  $ start_lat         : num [1:96834] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:96834] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:96834] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:96834] -87.7 -87.7 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:96834] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_feb21)
```

```
## spec_tbl_df [49,622 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:49622] "89E7AA6C29227EFF" "0FEFDE2603568365" "E6159D746B2DBB91" "B32D319
9F1C2E75B" ...
##  $ rideable_type     : chr [1:49622] "classic_bike" "classic_bike" "electric_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:49622], format: "2021-02-12 16:14:56" "2021-02-14 17:52:38" ...
##  $ ended_at          : POSIXct[1:49622], format: "2021-02-12 16:21:43" "2021-02-14 18:12:09" ...
##  $ start_station_name: chr [1:49622] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave" "Clark St &
Lake St" "Wood St & Chicago Ave" ...
##  $ start_station_id  : chr [1:49622] "525" "525" "KA1503000012" "637" ...
##  $ end_station_name  : chr [1:49622] "Sheridan Rd & Columbia Ave" "Bosworth Ave & Howard St" "State St
& Randolph St" "Honore St & Division St" ...
##  $ end_station_id    : chr [1:49622] "660" "16806" "TA1305000029" "TA1305000034" ...
##  $ start_lat         : num [1:49622] 42 42 41.9 41.9 41.8 ...
##  $ start_lng         : num [1:49622] -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ end_lat           : num [1:49622] 42 42 41.9 41.9 41.8 ...
##  $ end_lng           : num [1:49622] -87.7 -87.7 -87.6 -87.7 -87.6 ...
##  $ member_casual     : chr [1:49622] "member" "casual" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_mar21)
```

```
## spec_tbl_df [228,496 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:228496] "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05
AA75A168F2" ...
##  $ rideable_type     : chr [1:228496] "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:228496], format: "2021-03-16 08:32:30" "2021-03-28 01:26:28" ...
##  $ ended_at          : POSIXct[1:228496], format: "2021-03-16 08:36:34" "2021-03-28 01:36:55" ...
##  $ start_station_name: chr [1:228496] "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "S
hields Ave & 28th Pl" "Winthrop Ave & Lawrence Ave" ...
##  $ start_station_id  : chr [1:228496] "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr [1:228496] "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave"
"Halsted St & 35th St" "Broadway & Sheridan Rd" ...
##  $ end_station_id    : chr [1:228496] "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num [1:228496] 41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num [1:228496] -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:228496] 41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num [1:228496] -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:228496] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_apr21)
```

```
## spec_tbl_df [337,230 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:337230] "6C992BD37A98A63F" "1E0145613A209000" "E498E15508A80BAD" "188726
2AD101C604" ...
##  $ rideable_type     : chr [1:337230] "classic_bike" "docked_bike" "docked_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:337230], format: "2021-04-12 18:25:36" "2021-04-27 17:27:11" ...
##  $ ended_at          : POSIXct[1:337230], format: "2021-04-12 18:56:55" "2021-04-27 18:31:29" ...
##  $ start_station_name: chr [1:337230] "State St & Pearson St" "Dorchester Ave & 49th St" "Loomis Blvd
& 84th St" "Honore St & Division St" ...
##  $ start_station_id  : chr [1:337230] "TA1307000061" "KA1503000069" "20121" "TA1305000034" ...
##  $ end_station_name  : chr [1:337230] "Southport Ave & Waveland Ave" "Dorchester Ave & 49th St" "Loomi
s Blvd & 84th St" "Southport Ave & Waveland Ave" ...
##  $ end_station_id    : chr [1:337230] "13235" "KA1503000069" "20121" "13235" ...
##  $ start_lat         : num [1:337230] 41.9 41.8 41.7 41.9 41.7 ...
##  $ start_lng         : num [1:337230] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:337230] 41.9 41.8 41.7 41.9 41.7 ...
##  $ end_lng           : num [1:337230] -87.7 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:337230] "member" "casual" "casual" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_may21)
```

```
## spec_tbl_df [531,633 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:531633] "C809ED75D6160B2A" "DD59FDCE0ACACAF3" "0AB83CB88C43EFC2" "7881AC
6D39110C60" ...
##  $ rideable_type     : chr [1:531633] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:531633], format: "2021-05-30 11:58:15" "2021-05-30 11:29:14" ...
##  $ ended_at          : POSIXct[1:531633], format: "2021-05-30 12:10:39" "2021-05-30 12:14:09" ...
##  $ start_station_name: chr [1:531633] NA NA NA NA ...
##  $ start_station_id  : chr [1:531633] NA NA NA NA ...
##  $ end_station_name  : chr [1:531633] NA NA NA NA ...
##  $ end_station_id    : chr [1:531633] NA NA NA NA ...
##  $ start_lat         : num [1:531633] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:531633] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ end_lat           : num [1:531633] 41.9 41.8 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:531633] -87.6 -87.6 -87.7 -87.7 -87.7 ...
##  $ member_casual     : chr [1:531633] "casual" "casual" "casual" "casual" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_jun21)
```

```
## spec_tbl_df [729,595 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:729595] "99FEC93BA843FB20" "06048DCFC8520CAF" "9598066F68045DF2" "B03C0F
E48C412214" ...
##  $ rideable_type     : chr [1:729595] "electric_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:729595], format: "2021-06-13 14:31:28" "2021-06-04 11:18:02" ...
##  $ ended_at          : POSIXct[1:729595], format: "2021-06-13 14:34:11" "2021-06-04 11:24:19" ...
##  $ start_station_name: chr [1:729595] NA NA NA NA ...
##  $ start_station_id  : chr [1:729595] NA NA NA NA ...
##  $ end_station_name  : chr [1:729595] NA NA NA NA ...
##  $ end_station_id    : chr [1:729595] NA NA NA NA ...
##  $ start_lat         : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
##  $ start_lng         : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ end_lat           : num [1:729595] 41.8 41.8 41.8 41.8 41.8 ...
##  $ end_lng           : num [1:729595] -87.6 -87.6 -87.6 -87.6 -87.6 ...
##  $ member_casual     : chr [1:729595] "member" "member" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
str(trips_jul21)
```

```
## spec_tbl_df [822,410 x 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:822410] "0A1B623926EF4E16" "B2D5583A5A5E76EE" "6F264597DDBF427A" "379B58
EAB20E8AA5" ...
##  $ rideable_type     : chr [1:822410] "docked_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : POSIXct[1:822410], format: "2021-07-02 14:44:36" "2021-07-07 16:57:42" ...
##  $ ended_at          : POSIXct[1:822410], format: "2021-07-02 15:19:58" "2021-07-07 17:16:09" ...
##  $ start_station_name: chr [1:822410] "Michigan Ave & Washington St" "California Ave & Cortez St" "Wab
ash Ave & 16th St" "California Ave & Cortez St" ...
##  $ start_station_id  : chr [1:822410] "13001" "17660" "SL-012" "17660" ...
##  $ end_station_name  : chr [1:822410] "Halsted St & North Branch St" "Wood St & Hubbard St" "Rush St &
Hubbard St" "Carpenter St & Huron St" ...
##  $ end_station_id    : chr [1:822410] "KA1504000117" "13432" "KA1503000044" "13196" ...
##  $ start_lat         : num [1:822410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:822410] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num [1:822410] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:822410] -87.6 -87.7 -87.6 -87.7 -87.7 ...
##  $ member_casual     : chr [1:822410] "casual" "casual" "member" "member" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ride_id = col_character(),
##   ..   rideable_type = col_character(),
##   ..   started_at = col_datetime(format = ""),
##   ..   ended_at = col_datetime(format = ""),
##   ..   start_station_name = col_character(),
##   ..   start_station_id = col_character(),
##   ..   end_station_name = col_character(),
##   ..   end_station_id = col_character(),
##   ..   start_lat = col_double(),
##   ..   start_lng = col_double(),
##   ..   end_lat = col_double(),
##   ..   end_lng = col_double(),
##   ..   member_casual = col_character()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Checking the column names and their data types

This is to check for any mismatch to prevent any complications before combining all of the data into one.

```
compare_df_cols(trips_aug20, trips_sep20, trips_oct20, trips_nov20, trips_dec20, trips_jan21, trips_feb2
1, trips_mar21, trips_apr21, trips_may21, trips_jun21, trips_jul21, return = "mismatch")
```

```
##        column_name trips_aug20 trips_sep20 trips_oct20 trips_nov20 trips_dec20
## 1   end_station_id     numeric     numeric     numeric     numeric   character
## 2 start_station_id     numeric     numeric     numeric     numeric   character
##   trips_jan21 trips_feb21 trips_mar21 trips_apr21 trips_may21 trips_jun21
## 1   character   character   character   character   character   character
## 2   character   character   character   character   character   character
##   trips_jul21
## 1   character
## 2   character
```

Both end_station_id and start_station_id have different data types across the data sets which need to be converted.

Converting both end_station_id and start_station_id into character

```
trips_aug20 <- mutate(trips_aug20, end_station_id = as.character(end_station_id), start_station_id = as.c
haracter(start_station_id))
trips_sep20 <- mutate(trips_sep20, end_station_id = as.character(end_station_id), start_station_id = as.c
haracter(start_station_id))
trips_oct20 <- mutate(trips_oct20, end_station_id = as.character(end_station_id), start_station_id = as.c
haracter(start_station_id))
trips_nov20 <- mutate(trips_nov20, end_station_id = as.character(end_station_id), start_station_id = as.c
haracter(start_station_id))
```

Double checking the columns for any further mismatch

```
compare_df_cols(trips_aug20, trips_sep20, trips_oct20, trips_nov20, trips_dec20, trips_jan21, trips_feb2
1, trips_mar21, trips_apr21, trips_may21, trips_jun21, trips_jul21, return = "mismatch")
```

```
##  [1] column_name trips_aug20 trips_sep20 trips_oct20 trips_nov20 trips_dec20
##  [7] trips_jan21 trips_feb21 trips_mar21 trips_apr21 trips_may21 trips_jun21
## [13] trips_jul21
## <0 rows> (or 0-length row.names)
```

Combining all the datasets into one

```
trips_total <- rbind(trips_aug20, trips_sep20, trips_oct20, trips_nov20, trips_dec20, trips_jan21, trips_
feb21, trips_mar21, trips_apr21, trips_may21, trips_jun21, trips_jul21)

head(trips_total)
```

```
## # A tibble: 6 x 13
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>
## 1 322BD2~ docked_bike   2020-08-20 18:08:14 2020-08-20 18:17:51 Lake Shore Dr &~
## 2 2A3AEF~ electric_bike 2020-08-27 18:46:04 2020-08-27 19:54:51 Michigan Ave & ~
## 3 67DC1D~ electric_bike 2020-08-26 19:44:14 2020-08-26 21:53:07 Columbus Dr & R~
## 4 C79FBB~ electric_bike 2020-08-27 12:05:41 2020-08-27 12:53:45 Daley Center Pl~
## 5 13814D~ electric_bike 2020-08-27 16:49:02 2020-08-27 16:59:49 Leavitt St & Di~
## 6 56349A~ electric_bike 2020-08-27 17:26:23 2020-08-27 18:07:50 Leavitt St & Di~
## # ... with 8 more variables: 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>
```

Data Cleaning

Now that the data is properly organised, data cleaning will commence

Remove duplicates if any

```
trips_totalc <- trips_total[!duplicated(trips_total$ride_id),]

print(paste("Removed", nrow(trips_total) - nrow(trips_totalc), "duplicated rows")) # Number of duplicated
rows
```

```
## [1] "Removed 209 duplicated rows"
```

Checking for entries with NA and removing them

```
sum(!complete.cases(trips_totalc))
```

```
## [1] 563681
```

```
# Removing data entries with NA

trips_totalc <- trips_totalc[complete.cases(trips_totalc), ]
```

Removing entries with started_at greater than ended_at

```
trips_totalc <- trips_totalc %>%
  filter(trips_totalc$started_at < trips_totalc$ended_at)
```

Creating new ride length column

New columns containing the duration of trip in both seconds and minutes will be created

```
trips_totalc <- trips_totalc %>%
  mutate(ride_length = (difftime(trips_totalc$ended_at, trips_totalc$started_at)))

# Convert "ride_length" from Factor to numeric so we can run calculations on the data

is.factor(trips_totalc$ride_length)
```

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

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

is.numeric(trips_totalc$ride_length)
```

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

```
# Creating ride length (minutes) column

trips_totalc$ride_length_min <- (trips_totalc$ride_length/60)
```

Creating separate columns for date elements and Creating day of week column

The elements of the date time columns will be separated into new columns, including a new column showing the day of the week that the ride was taken.

```
trips_totalc$date <- as.Date(trips_totalc$started_at)

trips_totalc$year <- format(as.Date(trips_totalc$started_at), "%Y")

trips_totalc$month <- format(as.Date(trips_totalc$started_at), "%m")

trips_totalc$day <- format(as.Date(trips_totalc$started_at), "%d")

trips_totalc$day_of_week <-paste(format(as.Date(trips_totalc$started_at), "%u"), "-", format(as.Date(trip
s_totalc$started_at), "%a"))
```

Creating column for start hour of trip (might be useful for analysis)

The start hour of the trip might give us some relevant insights as well.

```
trips_totalc$start_hour <- format(trips_totalc$started_at, "%H")
```

Creating column for trip distance

The distance of the trip in kilometers would be calculated based on the coordinates given in the data.

```
trips_totalc <- trips_totalc %>%
  mutate(trip_distance_km = (distHaversine(cbind(trips_totalc$start_lat, trips_totalc$start_lng), cbind(t
rips_totalc$end_lat, trips_totalc$end_lng)))/1000)
```

Checking for bad data, if any

```
sum(trips_totalc$start_station_name == "HQ QR")
```

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

```
sum(trips_totalc$ride_length < 0)
```

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

Inspecting the new dataframe

```
colnames(trips_totalc)
```

```
##  [1] "ride_id"            "rideable_type"      "started_at"        
##  [4] "ended_at"           "start_station_name" "start_station_id"  
##  [7] "end_station_name"   "end_station_id"     "start_lat"         
## [10] "start_lng"          "end_lat"            "end_lng"           
## [13] "member_casual"      "ride_length"        "ride_length_min"   
## [16] "date"               "year"               "month"             
## [19] "day"                "day_of_week"        "start_hour"        
## [22] "trip_distance_km"
```

```
dim(trips_totalc)
```

```
## [1] 4159132      22
```

```
head(trips_totalc)
```

```
## # A tibble: 6 x 22
##   ride_id rideable_type started_at          ended_at            start_station_n~
##   <chr>   <chr>         <dttm>              <dttm>              <chr>           
## 1 322BD2~ docked_bike   2020-08-20 18:08:14 2020-08-20 18:17:51 Lake Shore Dr &~
## 2 2A3AEF~ electric_bike 2020-08-27 18:46:04 2020-08-27 19:54:51 Michigan Ave & ~
## 3 67DC1D~ electric_bike 2020-08-26 19:44:14 2020-08-26 21:53:07 Columbus Dr & R~
## 4 C79FBB~ electric_bike 2020-08-27 12:05:41 2020-08-27 12:53:45 Daley Center Pl~
## 5 13814D~ electric_bike 2020-08-27 16:49:02 2020-08-27 16:59:49 Leavitt St & Di~
## 6 56349A~ electric_bike 2020-08-27 17:26:23 2020-08-27 18:07:50 Leavitt St & Di~
## # ... with 17 more variables: 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>,
## #   ride_length_min <dbl>, date <date>, year <chr>, month <chr>, day <chr>,
## #   day_of_week <chr>, start_hour <chr>, trip_distance_km <dbl>
```

```
str(trips_totalc)
```

```
## tibble [4,159,132 x 22] (S3: tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:4159132] "322BD23D287743ED" "2A3AEF1AB9054D8B" "67DC1D133E8B5816" "C79FB
BD412E578A7" ...
##  $ rideable_type     : chr [1:4159132] "docked_bike" "electric_bike" "electric_bike" "electric_bike"
...
##  $ started_at        : POSIXct[1:4159132], format: "2020-08-20 18:08:14" "2020-08-27 18:46:04" ...
##  $ ended_at          : POSIXct[1:4159132], format: "2020-08-20 18:17:51" "2020-08-27 19:54:51" ...
##  $ start_station_name: chr [1:4159132] "Lake Shore Dr & Diversey Pkwy" "Michigan Ave & 14th St" "Colum
bus Dr & Randolph St" "Daley Center Plaza" ...
##  $ start_station_id  : chr [1:4159132] "329" "168" "195" "81" ...
##  $ end_station_name  : chr [1:4159132] "Clark St & Lincoln Ave" "Michigan Ave & 14th St" "State St & R
andolph St" "State St & Kinzie St" ...
##  $ end_station_id    : chr [1:4159132] "141" "168" "44" "47" ...
##  $ start_lat         : num [1:4159132] 41.9 41.9 41.9 41.9 41.9 ...
##  $ start_lng         : num [1:4159132] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ end_lat           : num [1:4159132] 41.9 41.9 41.9 41.9 41.9 ...
##  $ end_lng           : num [1:4159132] -87.6 -87.6 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr [1:4159132] "member" "casual" "casual" "casual" ...
##  $ ride_length       : num [1:4159132] 577 4127 7733 2884 647 ...
##  $ ride_length_min   : num [1:4159132] 9.62 68.78 128.88 48.07 10.78 ...
##  $ date              : Date[1:4159132], format: "2020-08-20" "2020-08-27" ...
##  $ year              : chr [1:4159132] "2020" "2020" "2020" "2020" ...
##  $ month             : chr [1:4159132] "08" "08" "08" "08" ...
##  $ day               : chr [1:4159132] "20" "27" "26" "27" ...
##  $ day_of_week       : chr [1:4159132] "4 - Thu" "4 - Thu" "3 - Wed" "4 - Thu" ...
##  $ start_hour        : chr [1:4159132] "18" "18" "19" "12" ...
##  $ trip_distance_km  : num [1:4159132] 0.21769 0.02699 0.8933 0.2351 0.00866 ...
```

```
summary(trips_totalc)
```

```
##    ride_id          rideable_type        started_at
##  Length:4159132     Length:4159132     Min.   :2020-08-01 00:00:01
##  Class :character   Class :character   1st Qu.:2020-09-26 16:10:32
##  Mode  :character   Mode  :character   Median :2021-03-27 23:38:08
##                                        Mean   :2021-02-11 16:24:08
##                                        3rd Qu.:2021-06-13 03:10:15
##                                        Max.   :2021-07-31 23:59:57
##     ended_at                    start_station_name start_station_id
##  Min.   :2020-08-01 00:04:41   Length:4159132     Length:4159132
##  1st Qu.:2020-09-26 16:40:24   Class :character   Class :character
##  Median :2021-03-28 00:03:59   Mode  :character   Mode  :character
##  Mean   :2021-02-11 16:48:42
##  3rd Qu.:2021-06-13 04:02:46
##  Max.   :2021-08-12 17:45:41
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:4159132     Length:4159132     Min.   :41.65   Min.   :-87.78
##  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.   :41.65   Min.   :-87.78   Length:4159132     Min.   :      1
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   1st Qu.:    449
##  Median :41.90   Median :-87.64   Mode  :character   Median :    801
##  Mean   :41.90   Mean   :-87.64                      Mean   :   1474
##  3rd Qu.:41.93   3rd Qu.:-87.63                      3rd Qu.:   1462
##  Max.   :42.08   Max.   :-87.52                      Max.   :3356649
##  ride_length_min        date                year               month
##  Min.   :    0.02   Min.   :2020-08-01   Length:4159132     Length:4159132
##  1st Qu.:    7.48   1st Qu.:2020-09-26   Class :character   Class :character
##  Median :   13.35   Median :2021-03-27   Mode  :character   Mode  :character
##  Mean   :   24.57   Mean   :2021-02-11
##  3rd Qu.:   24.37   3rd Qu.:2021-06-13
##  Max.   :55944.15   Max.   :2021-07-31
##      day             day_of_week         start_hour        trip_distance_km
##  Length:4159132     Length:4159132     Length:4159132     Min.   : 0.0000
##  Class :character   Class :character   Class :character   1st Qu.: 0.3685
##  Mode  :character   Mode  :character   Mode  :character   Median : 1.0304
##                                                           Mean   : 1.4363
##                                                           3rd Qu.: 2.0328
##                                                           Max.   :20.0039
```

```
skim(trips_totalc)
```

Data summary

Name	trips_totalc
Number of rows	4159132
Number of columns	22
________________________	
Column type frequency:	
character	12
Date	1
numeric	7
POSIXct	2

Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
ride_id	0	1	16	16	0	4159132	0
rideable_type	0	1	11	13	0	3	0
start_station_name	0	1	10	53	0	731	0
start_station_id	0	1	1	36	0	1281	0
end_station_name	0	1	10	53	0	729	0
end_station_id	0	1	1	36	0	1281	0
member_casual	0	1	6	6	0	2	0
year	0	1	4	4	0	2	0
month	0	1	2	2	0	12	0
day	0	1	2	2	0	31	0
day_of_week	0	1	7	7	0	7	0
start_hour	0	1	2	2	0	24	0

Variable type: Date

skim_variable	n_missing	complete_rate	min	max	median	n_unique
date	0	1	2020-08-01	2021-07-31	2021-03-27	365

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
start_lat	0	1	41.90	0.04	41.65	41.88	41.90	41.93	42.06	▁▁▇▇▁
start_lng	0	1	-87.64	0.02	-87.78	-87.66	-87.64	-87.63	-87.53	▁▁▇▃▁
end_lat	0	1	41.90	0.04	41.65	41.88	41.90	41.93	42.08	▁▁▇▆▁
end_lng	0	1	-87.64	0.02	-87.78	-87.66	-87.64	-87.63	-87.52	▁▁▇▂▁
ride_length	0	1	1474.10	13523.78	1.00	449.00	801.00	1462.00	3356649.00	▇▁▁▁▁
ride_length_min	0	1	24.57	225.40	0.02	7.48	13.35	24.37	55944.15	▇▁▁▁▁
trip_distance_km	0	1	1.44	1.47	0.00	0.37	1.03	2.03	20.00	▇▁▁▁▁

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2020-08-01 00:00:01	2021-07-31 23:59:57	2021-03-27 23:38:08	3580181
ended_at	0	1	2020-08-01 00:04:41	2021-08-12 17:45:41	2021-03-28 00:03:59	3564415

Analyse

Now that the data is properly cleaned and organised, the next step of the process (Analyse) can commence. The aim of this phase would be to answer questions about the dataset and ultimately use those answers to help tackle the business problem at hand.

For this portion, the analysis will be done in two parts. The first part would cover basic descriptive analysis of the data. The second part would cover further analysis of variables that were not covered in the first.

Descriptive Analysis

Proportion of both members and casuals in dataset

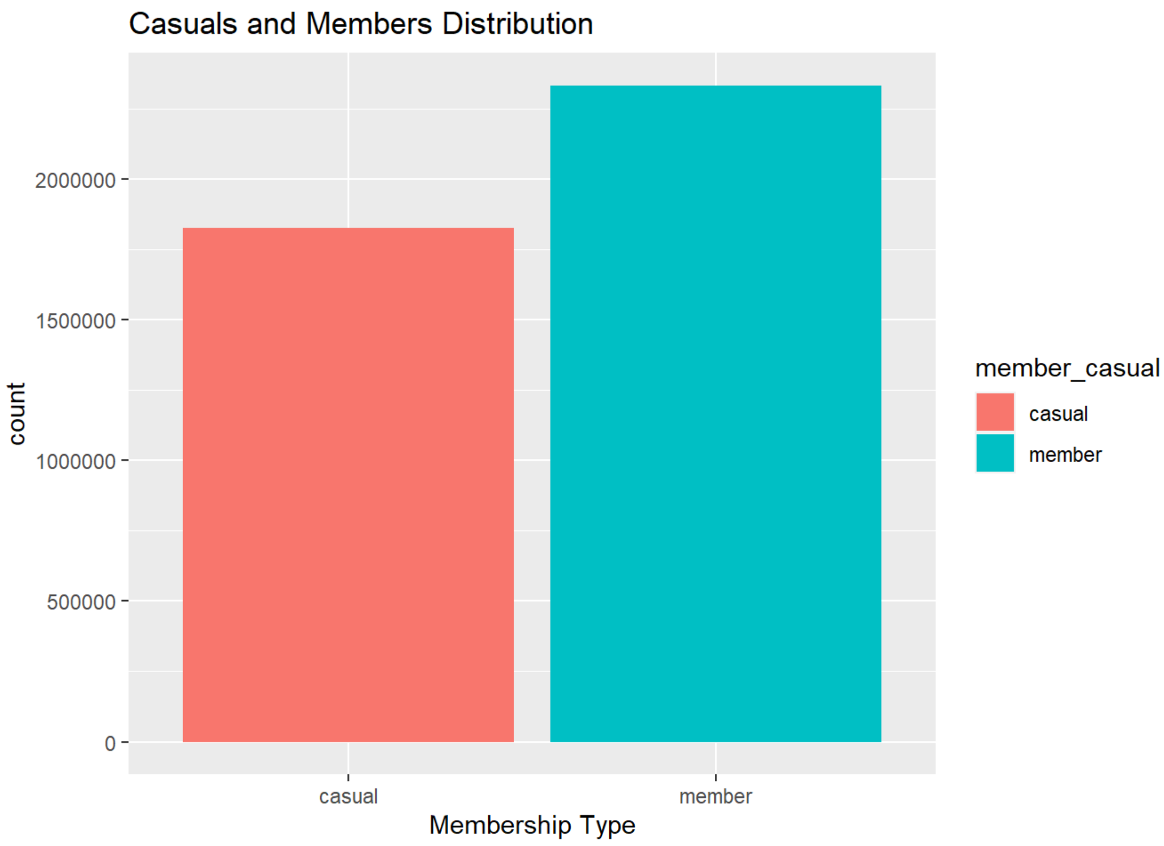
This is to get a better idea of the different proportion of both groups in this dataset.

```
trips_totalc %>%
  group_by(member_casual) %>%
  summarize(count = n(),
            percentage = length(ride_id)/nrow(trips_totalc))
```

```
## # A tibble: 2 x 3
##   member_casual   count percentage
##   <chr>           <int>      <dbl>
## 1 casual        1826043      0.439
## 2 member        2333089      0.561
```

Plotting the distribution of both members and casuals

```
ggplot(trips_totalc, aes(x=member_casual, fill=member_casual))+
  geom_bar()+
  labs(title="Casuals and Members Distribution", x="Membership Type")
```



As shown, members take up a higher proportion in this dataset as compared to casuals.

Descriptive Analysis for ride length

This would cover certain basic descriptive analysis for ride length.

```
# Mean ride_length

trips_totalc %>%
  summarize(mean(ride_length_min))
```

```
## # A tibble: 1 x 1
##   `mean(ride_length_min)`
##                     <dbl>
## 1                    24.6
```

```
# Median ride_length

trips_totalc %>%
  summarize(median(ride_length_min))
```

```
## # A tibble: 1 x 1
##   `median(ride_length_min)`
##                       <dbl>
## 1                      13.4
```

```
# Max ride_length

trips_totalc %>%
  summarize(max(ride_length_min))
```

```
## # A tibble: 1 x 1
##   `max(ride_length_min)`
##                    <dbl>
## 1                 55944.
```

```
# Min ride_length

trips_totalc %>%
  summarize(min(ride_length_min))
```

```
## # A tibble: 1 x 1
##   `min(ride_length_min)`
##                    <dbl>
## 1                 0.0167
```

```
# Alternative method
summary(trips_totalc$ride_length_min)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##     0.02     7.48    13.35    24.57    24.37 55944.15
```

This would give us a better idea of the data and would be further expounded in the later section.

Mode of Day of Week

Finding out the mode of the day of week would inform us of the most frequent day that both groups take bike rides.

```
Mode(trips_totalc$day_of_week)
```

```
## [1] "6 - Sat"
## attr(,"freq")
## [1] 785302
```

Throughout the week, Saturday has the most number of rides taken overall.

Comparing ride length between members and casuals

This would be a brief comparison in terms of ride length between both groups

Mean

```
aggregate(trips_totalc$ride_length_min ~ trips_totalc$member_casual, FUN = mean)
```

```
##   trips_totalc$member_casual trips_totalc$ride_length_min
## 1                     casual                     37.57207
## 2                     member                     14.39054
```

Casuals have a overall greater mean ride length than members.

Median

```
aggregate(trips_totalc$ride_length_min ~ trips_totalc$member_casual, FUN = median)
```

```
##   trips_totalc$member_casual trips_totalc$ride_length_min
## 1                     casual                        18.60
## 2                     member                        10.55
```

The median value for casuals is greater than that of members.

Max value

```
aggregate(trips_totalc$ride_length_min ~ trips_totalc$member_casual, FUN = max)
```

```
##   trips_totalc$member_casual trips_totalc$ride_length_min
## 1                     casual                     55944.15
## 2                     member                     33421.37
```

The maximum ride length for casuals is greater than that of members.

Min value

```
aggregate(trips_totalc$ride_length_min ~ trips_totalc$member_casual, FUN = min)
```

```
##   trips_totalc$member_casual trips_totalc$ride_length_min
## 1                     casual                   0.01666667
## 2                     member                   0.01666667
```

Based on these descriptive analyses, it could be said that casuals have an overall greater ride length as compared to members.

Average ride_length for member and casual riders

```
trips_totalc %>%
  group_by(member_casual) %>%
  summarize(mean(ride_length_min))
```

```
## # A tibble: 2 x 2
##   member_casual `mean(ride_length_min)`
##   <chr>                           <dbl>
## 1 casual                           37.6
## 2 member                           14.4
```

Average ride time by each day for member and casual riders

Breaking down the average ride length between members and casuals for each day of the week can make for easier comparison.

```
aggregate(trips_totalc$ride_length_min ~ trips_totalc$member_casual + trips_totalc$day_of_week, FUN = mea
n)
```

```
##    trips_totalc$member_casual trips_totalc$day_of_week
## 1                      casual                  1 - Mon
## 2                      member                  1 - Mon
## 3                      casual                  2 - Tue
## 4                      member                  2 - Tue
## 5                      casual                  3 - Wed
## 6                      member                  3 - Wed
## 7                      casual                  4 - Thu
## 8                      member                  4 - Thu
## 9                      casual                  5 - Fri
## 10                     member                  5 - Fri
## 11                     casual                  6 - Sat
## 12                     member                  6 - Sat
## 13                     casual                  7 - Sun
## 14                     member                  7 - Sun
##    trips_totalc$ride_length_min
## 1                      36.85736
## 2                      13.86012
## 3                      33.36123
## 4                      13.58315
## 5                      33.38146
## 6                      13.66398
## 7                      32.82248
## 8                      13.40513
## 9                      35.71428
## 10                     14.02672
## 11                     40.30729
## 12                     15.91700
## 13                     43.29739
## 14                     16.50060
```

Generally, ride length increases as the week progresses for both groups and in each day, casuals have longer ride durations than members.

Ridership data by type and weekday

Number of rides and the average duration classified by membership and day of week.

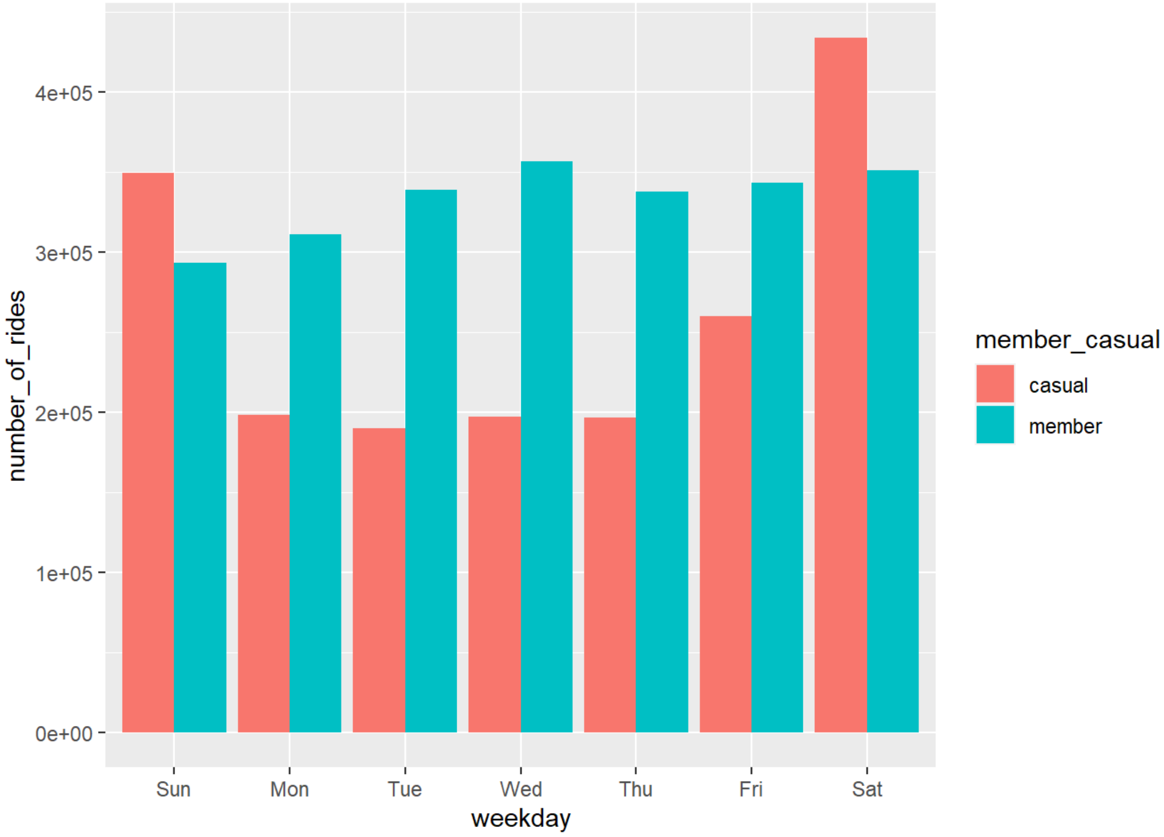
```
trips_totalc %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%  #creates weekday field using wday()
  group_by(member_casual, weekday) %>% #group by member type and weekday
  summarise(number_of_rides = n(),  #calculates number of rides
            average_duration = mean(ride_length)) %>%  #calculate average duration
  arrange(member_casual, weekday) #sorts
```

```
## `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 weekday number_of_rides average_duration
##    <chr>         <ord>             <int>            <dbl>
##  1 casual        Sun              349462            2598.
##  2 casual        Mon              198310            2211.
##  3 casual        Tue              190067            2002.
##  4 casual        Wed              197254            2003.
##  5 casual        Thu              197106            1969.
##  6 casual        Fri              259901            2143.
##  7 casual        Sat              433943            2418.
##  8 member        Sun              293564             990.
##  9 member        Mon              311184             832.
## 10 member        Tue              338988             815.
## 11 member        Wed              356688             820.
## 12 member        Thu              337822             804.
## 13 member        Fri              343484             842.
## 14 member        Sat              351359             955.
```

Visualise number of rides by rider type

```
trips_totalc %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x=weekday, y = number_of_rides, fill = member_casual))+
  geom_col(position = "dodge")
```

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

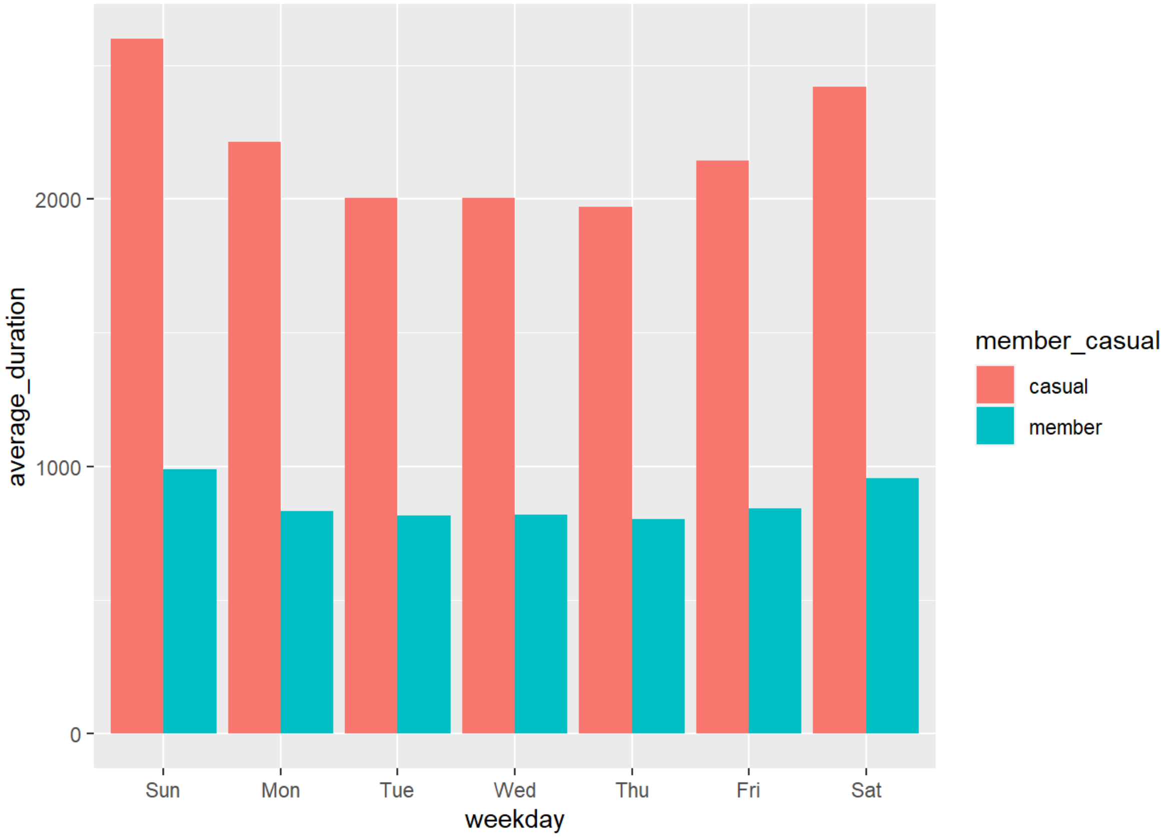


Members take more rides on weekdays. On weekends, however, the trend is reversed.

Visualisation for average duration

```
trips_totalc %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n(),
            average_duration = mean(ride_length)) %>%
  arrange(member_casual, weekday) %>%
  ggplot(aes(x=weekday, y=average_duration, fill=member_casual)) +
  geom_col(position = "dodge")
```

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



Throughout the week, casuals have significantly greater ride lengths as compared to members.

Further Analysis with other variables

Now that we have covered basic descriptive analysis of the dataset, we would move onto further analysis with the other variables of the dataset.

Popular Stations

A possible angle would be to look at stations that riders frequent the most, either to start their rides or to end their rides.

Start Station (Overall)

```
trips_totalc %>%
  group_by(start_station_name) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100)) %>%
  arrange(desc(number_of_rides))
```

```
## # A tibble: 731 x 7
##    start_station_name   number_of_rides percentage member_rides member_percenta~
##    <chr>                          <int>      <dbl>        <int>            <dbl>
##  1 Streeter Dr & Grand~           64987      1.56         13914             21.4
##  2 Lake Shore Dr & Mon~           41836      1.01         10863             26.0
##  3 Theater on the Lake            39520      0.950        17714             44.8
##  4 Clark St & Elm St              38410      0.924        23766             61.9
##  5 Michigan Ave & Oak ~           37666      0.906        12743             33.8
##  6 Lake Shore Dr & Nor~           37555      0.903        15945             42.5
##  7 Millennium Park                35807      0.861         6715             18.8
##  8 Wells St & Concord ~           35330      0.849        19977             56.5
##  9 Wells St & Elm St              31274      0.752        18254             58.4
## 10 Clark St & Armitage~           30022      0.722        16131             53.7
## # ... with 721 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

This list shows the stations where the most number of rides started overall. The top 3 stations are: Streeter Dr & Grand Ave,Lake Shore Dr & Monroe St, Theater on the Lake.

Popular Start Station (Casuals)

```
trips_totalc %>%
  group_by(start_station_name) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100)) %>%
  arrange(desc(casual_rides))
```

```
## # A tibble: 731 x 7
##    start_station_name   number_of_rides percentage member_rides member_percenta~
##    <chr>                          <int>      <dbl>        <int>            <dbl>
##  1 Streeter Dr & Grand~           64987      1.56         13914             21.4
##  2 Lake Shore Dr & Mon~           41836      1.01         10863             26.0
##  3 Millennium Park                35807      0.861         6715             18.8
##  4 Michigan Ave & Oak ~           37666      0.906        12743             33.8
##  5 Theater on the Lake            39520      0.950        17714             44.8
##  6 Lake Shore Dr & Nor~           37555      0.903        15945             42.5
##  7 Shedd Aquarium                 24087      0.579         5094             21.1
##  8 Indiana Ave & Roose~           28445      0.684        12124             42.6
##  9 Wells St & Concord ~           35330      0.849        19977             56.5
## 10 Clark St & Lincoln ~           29599      0.712        14936             50.5
## # ... with 721 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

This list shows the popular stations where most casuals start their rides. The most frequent stations where casuals start their rides are: Streeter Dr & Grand Ave,Lake Shore Dr & Monroe St, Millennium Park.

Popular Start Station (Members)

```
trips_totalc %>%
  group_by(start_station_name) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100)) %>%
  arrange(desc(member_rides))
```

```
## # A tibble: 731 x 7
##    start_station_name   number_of_rides percentage member_rides member_percenta~
##    <chr>                          <int>      <dbl>        <int>            <dbl>
##  1 Clark St & Elm St              38410      0.924        23766             61.9
##  2 Wells St & Concord ~           35330      0.849        19977             56.5
##  3 Kingsbury St & Kinz~           27730      0.667        19354             69.8
##  4 Wells St & Elm St              31274      0.752        18254             58.4
##  5 Theater on the Lake            39520      0.950        17714             44.8
##  6 Dearborn St & Erie ~           28854      0.694        17633             61.1
##  7 Broadway & Barry Ave           27893      0.671        17443             62.5
##  8 St. Clair St & Erie~           26008      0.625        17367             66.8
##  9 Wells St & Huron St            27877      0.670        17077             61.3
## 10 Clark St & Armitage~           30022      0.722        16131             53.7
## # ... with 721 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

This list shows the popular stations where most members start their rides. The most frequent stations where members start their rides are: Clark St & Elm St, Wells St & Concord Ln, Kingsbury St & Kinzie St

Popular End Station (Overall)

```
trips_totalc %>%
  group_by(end_station_name) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100)) %>%
  arrange(desc(number_of_rides))
```

```
## # A tibble: 729 x 7
##    end_station_name     number_of_rides percentage member_rides member_percenta~
##    <chr>                          <int>      <dbl>        <int>            <dbl>
##  1 Streeter Dr & Grand~           67805      1.63         13112             19.3
##  2 Lake Shore Dr & Nor~           41556      0.999        16435             39.5
##  3 Theater on the Lake            41176      0.990        16892             41.0
##  4 Lake Shore Dr & Mon~           40941      0.984        11169             27.3
##  5 Michigan Ave & Oak ~           38829      0.934        12498             32.2
##  6 Clark St & Elm St              38174      0.918        24118             63.2
##  7 Millennium Park                37703      0.907         6982             18.5
##  8 Wells St & Concord ~           36098      0.868        20555             56.9
##  9 Wells St & Elm St              30681      0.738        18150             59.2
## 10 Clark St & Lincoln ~           29818      0.717        14662             49.2
## # ... with 719 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

This list shows the stations where the most number of rides ended overall. The top 3 stations are: Streeter Dr & Grand Ave,Lake Shore Dr & North Blvd, Theater on the Lake.

Popular End Station (Casuals)

```
trips_totalc %>%
  group_by(end_station_name) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100)) %>%
  arrange(desc(casual_rides))
```

```
## # A tibble: 729 x 7
##    end_station_name     number_of_rides percentage member_rides member_percenta~
##    <chr>                          <int>      <dbl>        <int>            <dbl>
##  1 Streeter Dr & Grand~           67805      1.63         13112             19.3
##  2 Millennium Park                37703      0.907         6982             18.5
##  3 Lake Shore Dr & Mon~           40941      0.984        11169             27.3
##  4 Michigan Ave & Oak ~           38829      0.934        12498             32.2
##  5 Lake Shore Dr & Nor~           41556      0.999        16435             39.5
##  6 Theater on the Lake            41176      0.990        16892             41.0
##  7 Shedd Aquarium                 21512      0.517         4831             22.5
##  8 Indiana Ave & Roose~           28292      0.680        11694             41.3
##  9 Wells St & Concord ~           36098      0.868        20555             56.9
## 10 Clark St & Lincoln ~           29818      0.717        14662             49.2
## # ... with 719 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

This list shows the stations where most casuals end their rides. The top 3 stations are: Streeter Dr & Grand Ave, Millennium Park and Lake Shore Dr & Monroe St.

Popular End Stations (Members)

```
trips_totalc %>%
  group_by(end_station_name) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100)) %>%
  arrange(desc(member_rides))
```

```
## # A tibble: 729 x 7
##    end_station_name     number_of_rides percentage member_rides member_percenta~
##    <chr>                          <int>      <dbl>        <int>            <dbl>
##  1 Clark St & Elm St              38174      0.918        24118             63.2
##  2 Wells St & Concord ~           36098      0.868        20555             56.9
##  3 Kingsbury St & Kinz~           27294      0.656        19724             72.3
##  4 Dearborn St & Erie ~           29417      0.707        18198             61.9
##  5 Wells St & Elm St              30681      0.738        18150             59.2
##  6 St. Clair St & Erie~           28081      0.675        18027             64.2
##  7 Broadway & Barry Ave           28538      0.686        17638             61.8
##  8 Theater on the Lake            41176      0.990        16892             41.0
##  9 Lake Shore Dr & Nor~           41556      0.999        16435             39.5
## 10 Wells St & Huron St            26663      0.641        16319             61.2
## # ... with 719 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

The list shows the stations where most members end their rides. The top 3 stations are: Clark St & Elm St, Wells St & Concord Ln and Kingsbury St & Kinzie St.

As shown above, members and casuals do frequent different stations when starting and ending their rides. However, there are also certain overlaps across the various categories. By understanding the popular locations that both demographics frequent during their rides, we would get a better understanding towards their bike usage patterns.

Furthermore, these popular locations might also offer opportunities to capitalise and enhance the marketing strategies.

Year

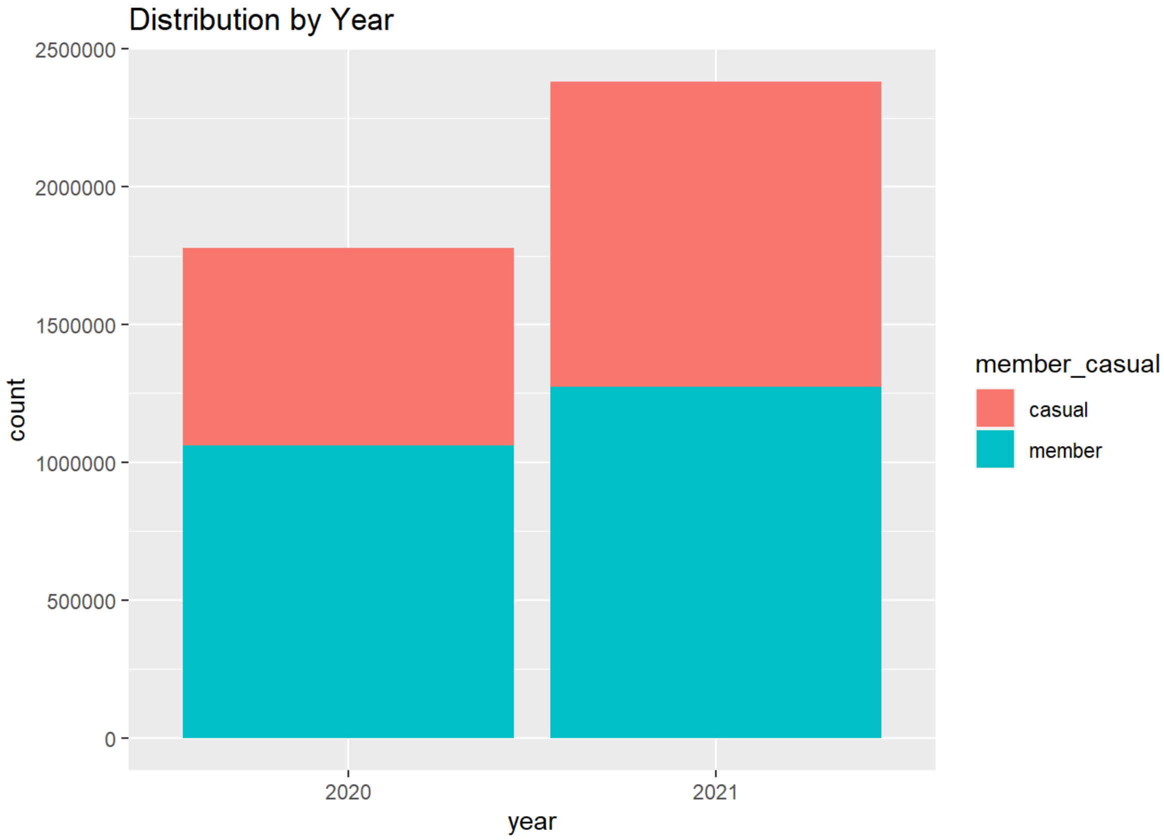
The next variable to examine would be the distribution of the data by year.

```
trips_totalc %>%
  group_by(year) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc)*100),
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))
```

```
## # A tibble: 2 x 7
##   year  number_of_rides percentage member_rides member_percentage casual_rides
##   <chr>           <int>      <dbl>        <int>             <dbl>        <int>
## 1 2020          1776710       42.7      1060437              59.7       716273
## 2 2021          2382422       57.3      1272652              53.4      1109770
## # ... with 1 more variable: casual_percentage <dbl>
```

Visualising the distribution by year

```
ggplot(trips_totalc, aes(x=year, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution by Year")
```



There were more rides in the first half of 2021 (Jan - July) as compared to the second half of 2020 (Aug - Dec). There are both more members and casuals in 2021 (Jan - July) as compared to 2020 (Aug - Dec).

Month

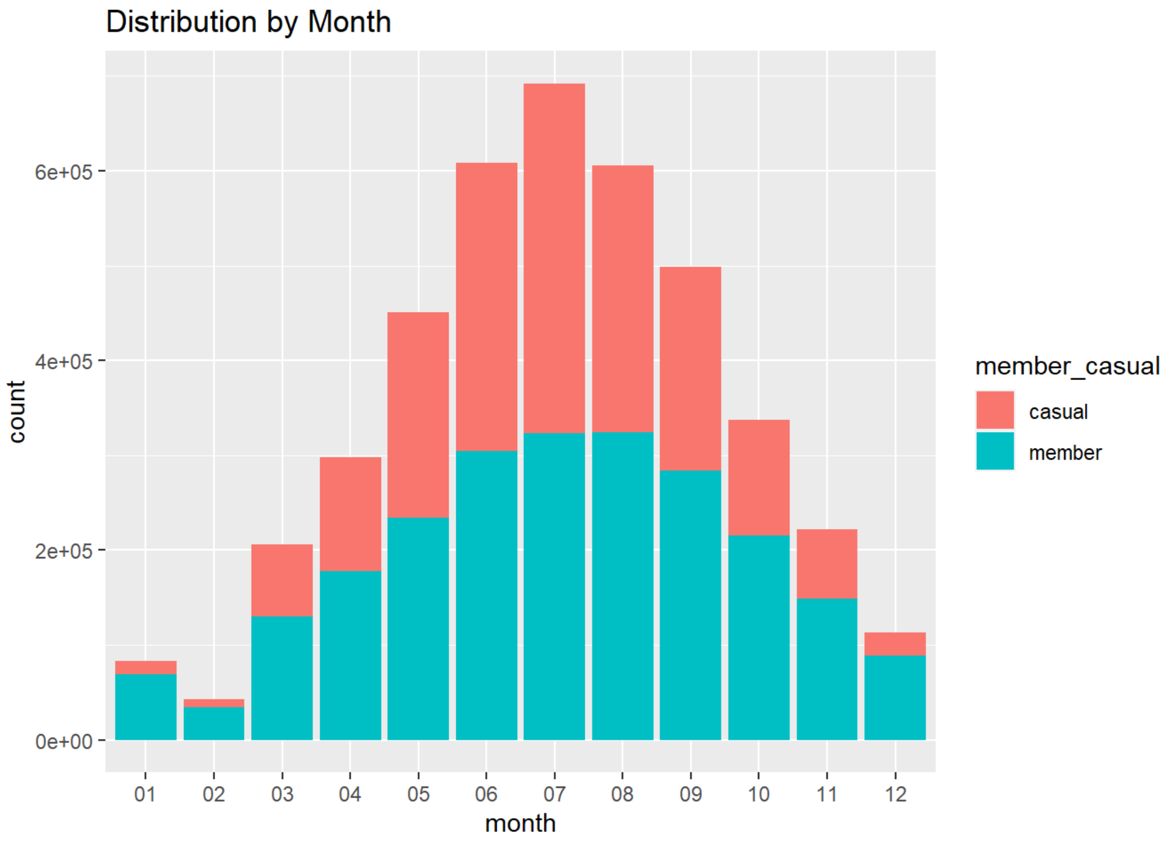
Examining the distribution of data by month.

```
trips_totalc %>%
  group_by(month) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc)*100),
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))
```

```
## # A tibble: 12 x 7
##    month number_of_rides percentage member_rides member_percentage casual_rides
##    <chr>           <int>      <dbl>        <int>             <dbl>        <int>
##  1 01              83508       2.01        68818              82.4        14690
##  2 02              42994       1.03        34381              80.0         8613
##  3 03             205687       4.95       130046              63.2        75641
##  4 04             298199       7.17       177781              59.6       120418
##  5 05             450978      10.8        234155              51.9       216823
##  6 06             608763      14.6        304579              50.0       304184
##  7 07             692293      16.6        322892              46.6       369401
##  8 08             605652      14.6        323707              53.4       281945
##  9 09             498228      12.0        283556              56.9       214672
## 10 10             337375       8.11       215058              63.7       122317
## 11 11             221916       5.34       149069              67.2        72847
## 12 12             113539       2.73        89047              78.4        24492
## # ... with 1 more variable: casual_percentage <dbl>
```

Visualising the distribution by month

```
ggplot(trips_totalc, aes(x=month, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution by Month")
```



Generally, number of rides increase from January to July with a slight dip in February. The number of rides peak in July, before decreasing from July to December.

Both number of casuals and members follow the same general trend throughout the year, with the number of casuals peaking in July and the number of members peaking in August. There were more members than casuals in all months except July. The data demonstrates a cyclical nature towards the monthly distribution of rides.

The lowest period would be from December to February. Given the cyclical nature of the data, a characteristic of these months could have influenced this decrease in number of rides (for instance weather conditions).

Correlational Analysis between monthly rides and weather

The mean monthly temperature of chicago from 1999-2020 was obtained from National Weather Servicelink (https://www.weather.gov/wrh/climate?wfo=lot). A basic correlational analysis would be conducted between the temperature of Chicago and number of rides to investigate the presence of any possible relationship.

```
chicago_mean_temp <- c(-4.2, -2.5, 3.6, 9.6, 15.6, 21.1, 23.9, 22.9, 18.9, 11.8, 5.2, -1.5)

months <- c("01", "02", "03", "04", "05", "06", "07", "08", "09", "10", "11", "12")

chicago_monthly_data <- data.frame(months, chicago_mean_temp)

monthly_data <- trips_totalc %>%
  group_by(month) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc)*100),
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))


cor(chicago_monthly_data$chicago_mean_temp, monthly_data$number_of_rides)
```

```
## [1] 0.9891438
```

There is a significant positive correlation between the mean Chicago temperature and the number of monthly rides. In other words, when one variable increases, the other variable is likely to move in the same direction with a similar magnitude. This could be a possible explanation in accounting for the trend in monthly rides.

However, it is important to note that this is a basic correlational analysis. No form of causation has been established. More data regarding Chicago's climate should be gathered and more in-depth analysis should be carried out before firmly establishing any conclusions. Other factors could also be at play. Chicago's weather is, as of this moment, a possibility.

Day of Week

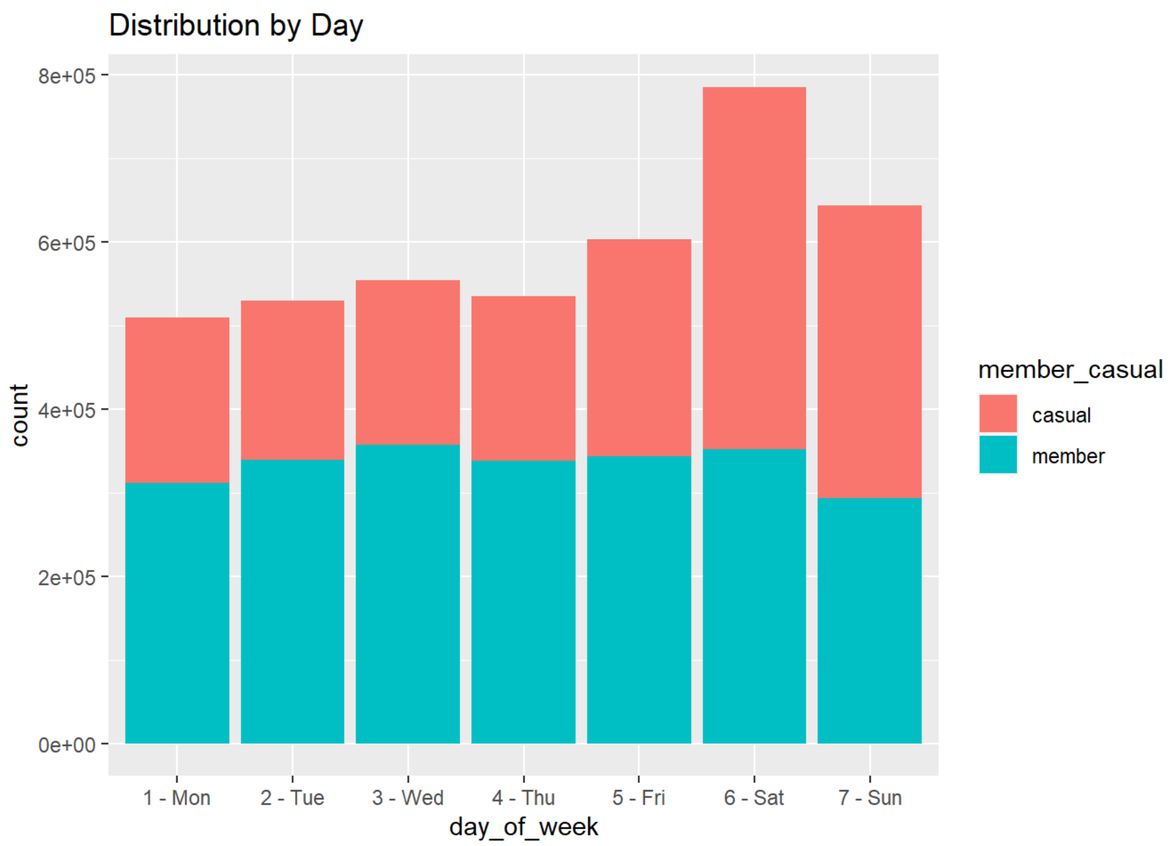
Examining the distribution of rides by day of week.

```
trips_totalc %>%
  group_by(day_of_week) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))
```

```
## # A tibble: 7 x 7
##   day_of_week number_of_rides percentage member_rides member_percentage
##   <chr>                 <int>      <dbl>        <int>             <dbl>
## 1 1 - Mon              509494       12.3       311184              61.1
## 2 2 - Tue              529055       12.7       338988              64.1
## 3 3 - Wed              553942       13.3       356688              64.4
## 4 4 - Thu              534928       12.9       337822              63.2
## 5 5 - Fri              603385       14.5       343484              56.9
## 6 6 - Sat              785302       18.9       351359              44.7
## 7 7 - Sun              643026       15.5       293564              45.7
## # ... with 2 more variables: casual_rides <int>, casual_percentage <dbl>
```

Visualising the distribution of rides by day of week

```
ggplot(trips_totalc, aes(x=day_of_week, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution by Day")
```



Weekends have greater number of rides than weekdays, with Saturday having the most number of rides. Saturday also has the highest number of casuals. On the other hand, wednesday has the highest number of members. On weekdays, there are more members than casuals. However, on weekends, there are more casuals than members.

The number of casuals generally increases as the week progresses, with a sharper increase in number during the weekends. The number of members have a more stable trend with less fluctuations around similar levels.

Start Hour

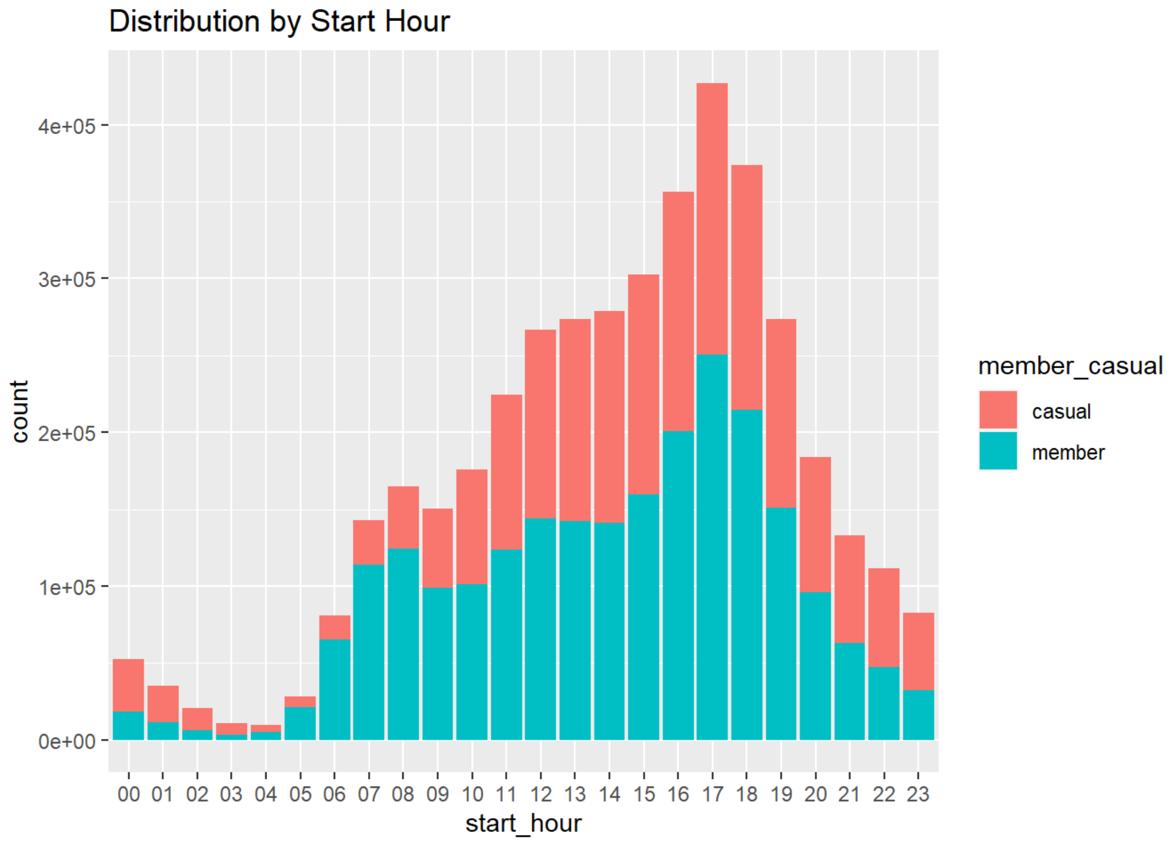
Examining the distribution by the start hour of the trip.

```
trips_totalc %>%
  group_by(start_hour) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))
```

```
## # A tibble: 24 x 7
##    start_hour number_of_rides percentage member_rides member_percentage
##    <chr>                <int>      <dbl>        <int>             <dbl>
##  1 00                   52629      1.27         18502              35.2
##  2 01                   35006      0.842        11393              32.5
##  3 02                   20536      0.494         6149              29.9
##  4 03                   10826      0.260         3451              31.9
##  5 04                    9750      0.234         4692              48.1
##  6 05                   27870      0.670        21063              75.6
##  7 06                   81015      1.95         64950              80.2
##  8 07                  142603      3.43        113814              79.8
##  9 08                  164978      3.97        124512              75.5
## 10 09                  150342      3.61         98511              65.5
## # ... with 14 more rows, and 2 more variables: casual_rides <int>,
## #   casual_percentage <dbl>
```

Visualising the distribution by hour

```
ggplot(trips_totalc, aes(x=start_hour, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution by Start Hour")
```



There is an overall higher number of rides in the afternoon and early hours of the evening, with the number of rides peaking at 5pm.

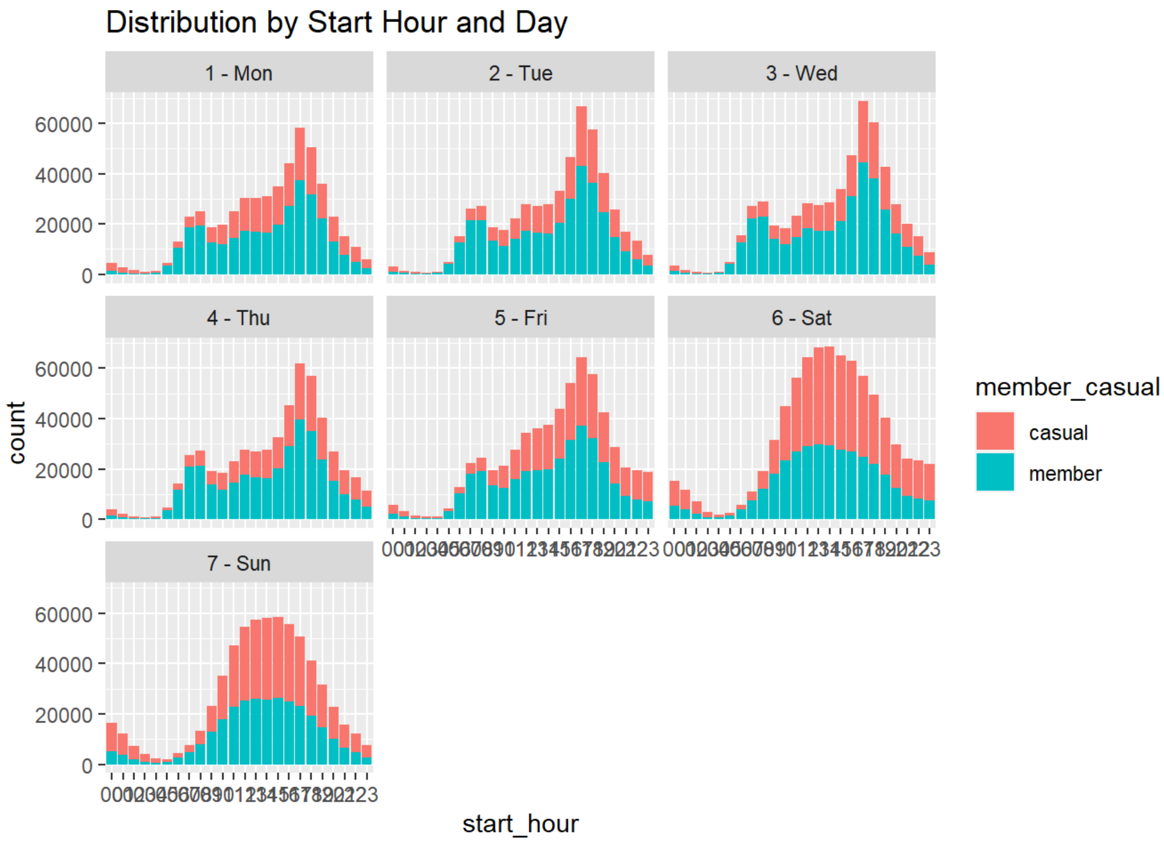
Both casuals and members dominate the ridership at different times of the day.

Between 12am to 4am, there are more casuals than members. From 5am onwards, there are more members than casuals throughout the remaining hours of the day.

Combining Day of Week and Start Hour

By combining both day of week and start hour, one might be able to obtain a more in-depth insight towards the breakdown of ride data.

```
ggplot(trips_totalc, aes(x=start_hour, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution by Start Hour and Day")+
  facet_wrap(~day_of_week)
```



The graphs do show differences between weekdays and weekends.

There is a sharper and more significant increase in casuals during the weekends as compared to the weekdays.

The progression in the number of rides throughout the day is different for both weekends and weekdays. The graphs for weekends have a smoother flow while the graph for weekdays have a steeper progression.

The peak in number of riders differ also between weekdays (~5pm) and weekends (~12pm).

On weekdays, there are also significant increases in number of rides at certain times (eg 4pm-5pm).

By understanding the information given from these data, we can better understand the demographics of the riders from the two groups, their usages and their purposes. For instance, with the significant increases in number of rides at certain time points and higher proportions of members during those times, one could suggest that these members are working adults who utilise the bikes as a means of transportation to work.

Bike Type

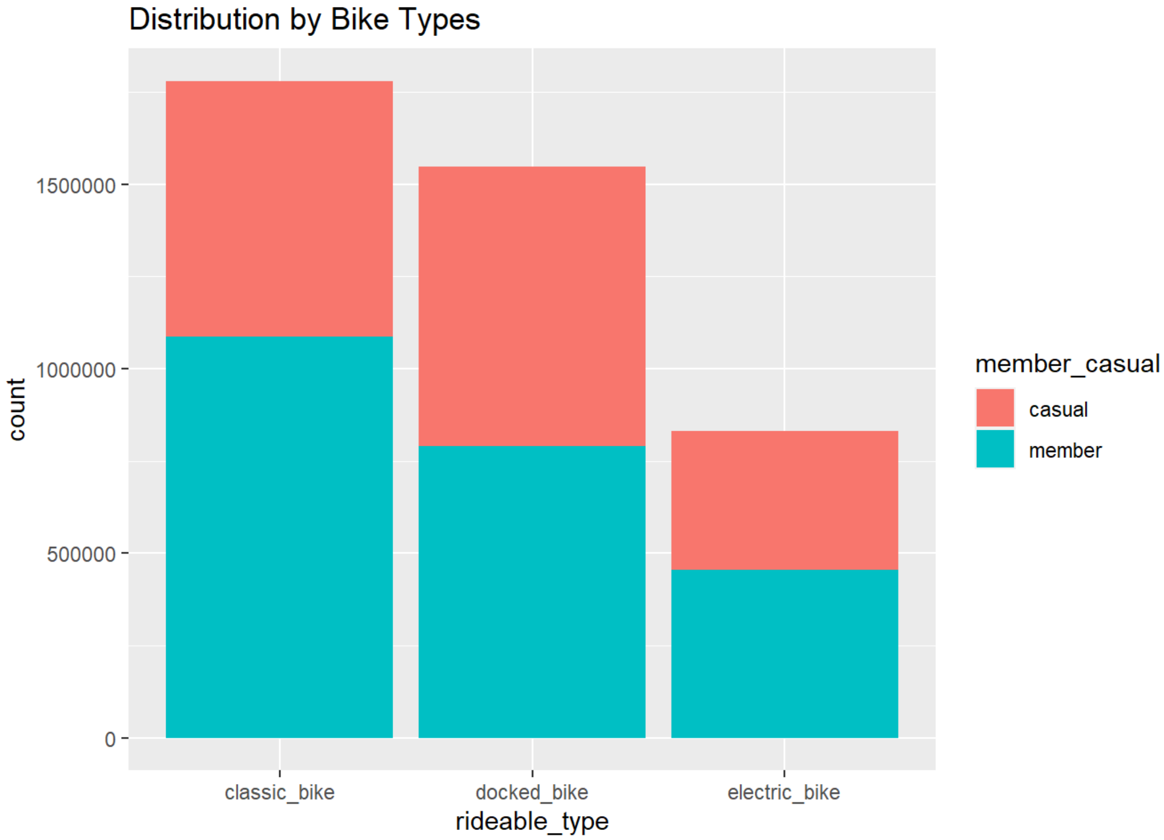
Examining the distribution of rides across the various types of bikes.

```
trips_totalc %>%
  group_by(rideable_type) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))
```

```
## # A tibble: 3 x 7
##   rideable_type number_of_rides percentage member_rides member_percentage
##   <chr>                   <int>      <dbl>        <int>             <dbl>
## 1 classic_bike          1780342       42.8      1087973              61.1
## 2 docked_bike           1547791       37.2       790970              51.1
## 3 electric_bike          830999       20.0       454146              54.7
## # ... with 2 more variables: casual_rides <int>, casual_percentage <dbl>
```

Visualising the distribution across bike types

```
ggplot(trips_totalc, aes(x=rideable_type, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution by Bike Types")
```



Overall, classic bikes have the highest popularity. Majority of the members also have a stronger preference for classic bikes. Casuals, however, have a stronger preference for docked bikes.

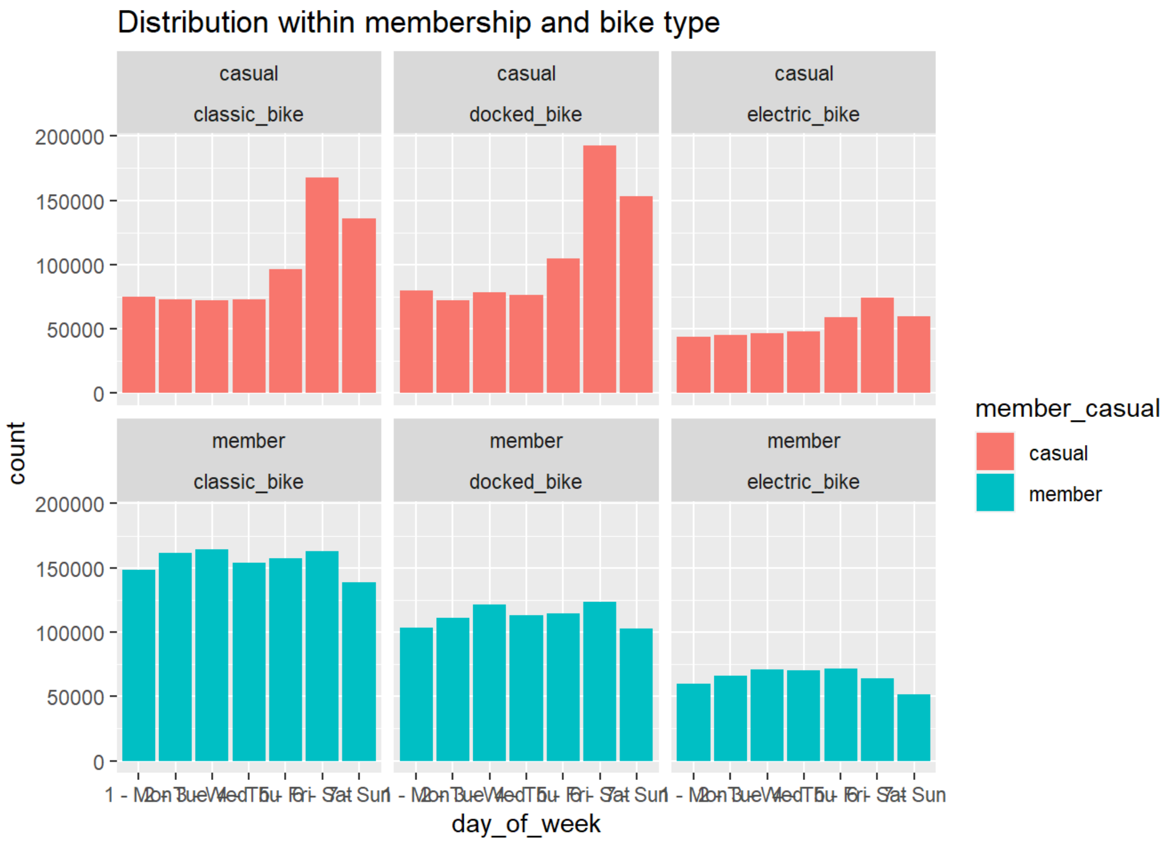
Combining bike type, membership and day of week

Combining the variables of bike type, membership and day of week might give a different perspective from the data.

```
trips_totalc %>%
  group_by(rideable_type, day_of_week) %>%
  summarise(number_of_rides = n(),
            percentage = (number_of_rides/nrow(trips_totalc))*100,
            member_rides = sum(member_casual == "member"),
            member_percentage = ((sum(member_casual == "member")/number_of_rides)*100),
            casual_rides = sum(member_casual == "casual"),
            casual_percentage = ((sum(member_casual == "casual")/number_of_rides)*100))
```

```
## `summarise()` has grouped output by 'rideable_type'. You can override using the `.groups` argument.
## # A tibble: 21 x 8
## # Groups:   rideable_type [3]
##    rideable_type day_of_week number_of_rides percentage member_rides
##    <chr>         <chr>                 <int>      <dbl>        <int>
##  1 classic_bike  1 - Mon              223396       5.37       148475
##  2 classic_bike  2 - Tue              234220       5.63       161545
##  3 classic_bike  3 - Wed              236222       5.68       164279
##  4 classic_bike  4 - Thu              227014       5.46       154196
##  5 classic_bike  5 - Fri              253404       6.09       157211
##  6 classic_bike  6 - Sat              330844       7.95       163251
##  7 classic_bike  7 - Sun              275242       6.62       139016
##  8 docked_bike   1 - Mon              182976       4.40       103154
##  9 docked_bike   2 - Tue              183674       4.42       111381
## 10 docked_bike   3 - Wed              200167       4.81       121763
## # ... with 11 more rows, and 3 more variables: member_percentage <dbl>,
## #   casual_rides <int>, casual_percentage <dbl>
```

```
ggplot(trips_totalc, aes(x=day_of_week, fill=member_casual))+
  geom_bar()+
  labs(title="Distribution within membership and bike type") +
  facet_wrap(~member_casual + rideable_type)
```



Both groups have different usage patterns of each bike throughout the week. Within each membership group, their usage patterns are relatively consistent across each type of bikes.

For casuals, their usage are highest during weekends across all 3 types of bikes, with lower usage during weekdays. There is also a sharp increase in usage from friday to saturday for both classic and docked bikes.

For members, the differences in number of rides across the week are not as stark. Their usage levels on weekdays (such as wednesday) are similar to that of weekends.

Ride Length

Previously, we have conducted basic descriptive analysis on the variable ride length. Now, we would dive deeper into this variable.

We would start by obtaining summary statistics.

```
summary(trips_totalc$ride_length_min)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
##     0.02     7.48    13.35    24.57    24.37 55944.15
```

The maximum value of 55944 minutes (equivalent of 932.4 hours) is unlikely. Furthermore, the minimum value of 0.02 minutes might also be too small. There is a need to check for outliers.

```
quantile(trips_totalc$ride_length_min) # checking the percentile
```

```
##           0%          25%          50%          75%         100% 
## 1.666667e-02 7.483333e+00 1.335000e+01 2.436667e+01 5.594415e+04
```

```
quantile(trips_totalc$ride_length_min, probs = seq(0, 0.05, 0.01)) # breaking down the 1st percentile
```

```
##         0%         1%         2%         3%         4%         5% 
## 0.01666667 0.60000000 1.73333333 2.33333333 2.76666667 3.11666667
```

```
quantile(trips_totalc$ride_length_min, probs = seq(0.95, 1, 0.01)) # breaking down the 95th percentile
```

```
##         95%         96%         97%         98%         99%        100% 
##    64.86667    73.90000    86.23450   105.03333   141.63333 55944.15000
```

```
percentile_duration <- quantile(trips_totalc$ride_length_min, probs = seq(0, 1, 0.01))  # saving the perc
entile values
```

By breaking down both extreme ends of the data, we would be able to obtain a clearer picture. The values between 0-5th percentile are too small and might not be informative towards the analysis. The 100th percentile value is also too large and unlikely.

As such, going forth, these outlier data would be removed and the analysis would be carried out on a subset of data.

```
trips_total_no_outliers <- trips_totalc %>%
  filter(ride_length_min > percentile_duration["5%"]) %>%
  filter(ride_length_min < percentile_duration["99%"])

num_of_rows_v1 <- nrow(trips_total_no_outliers) # saving the number of rows

print(paste("Removed", nrow(trips_totalc)-nrow(trips_total_no_outliers), "rows as outliers")) ## Number o
f rows removed
```

```
## [1] "Removed 251150 rows as outliers"
```

Distribution of ride length by membership

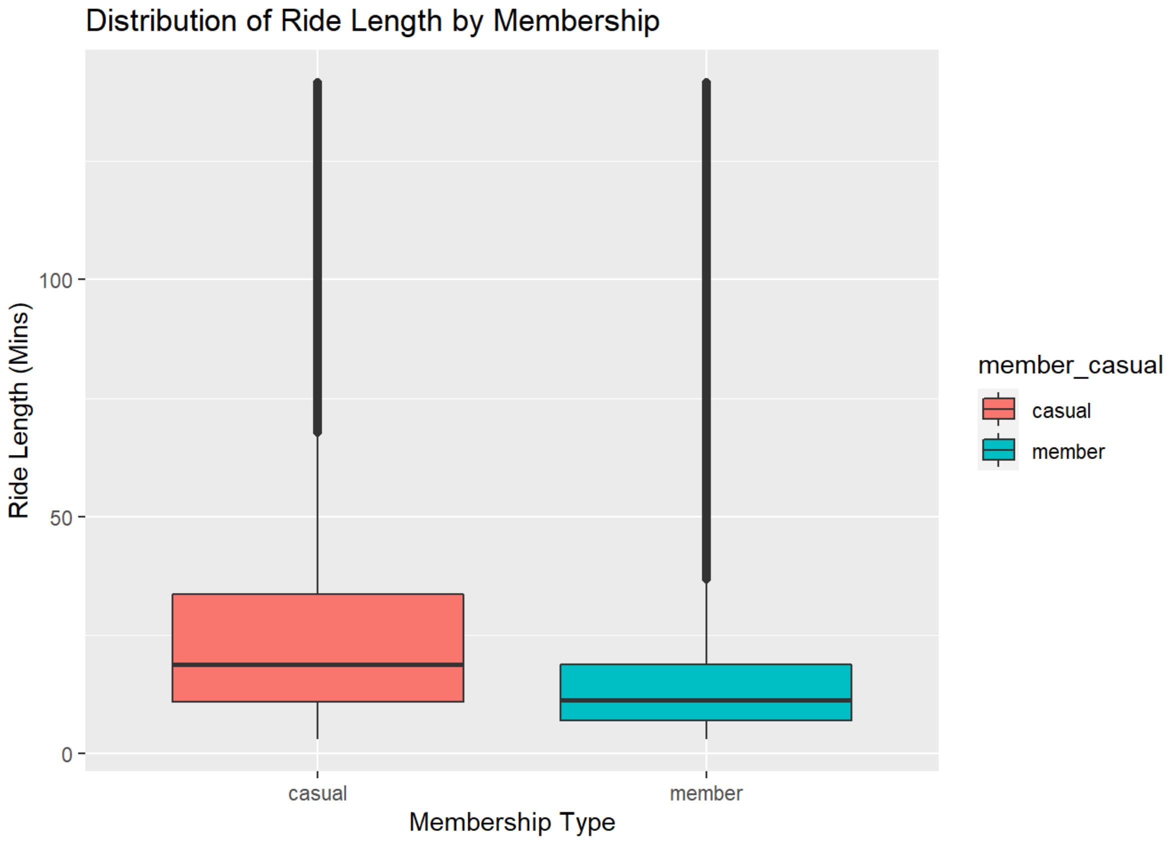
Breaking down the distribution of ride length within each group

```
trips_total_no_outliers %>%
  group_by(member_casual) %>%
  summarise(mean = mean(ride_length_min),
            "first_q" = quantile(ride_length_min, 0.25),
            median = median(ride_length_min),
            "third_q" = quantile(ride_length_min, 0.75),
            IQR = third_q - first_q)
```

```
## # A tibble: 2 x 6
##   member_casual  mean first_q median third_q   IQR
##   <chr>         <dbl>   <dbl>  <dbl>   <dbl> <dbl>
## 1 casual         27.3   10.8    18.8    33.6  22.7
## 2 member         14.7    6.93   11.3    18.9  12.0
```

Plotting the distribution

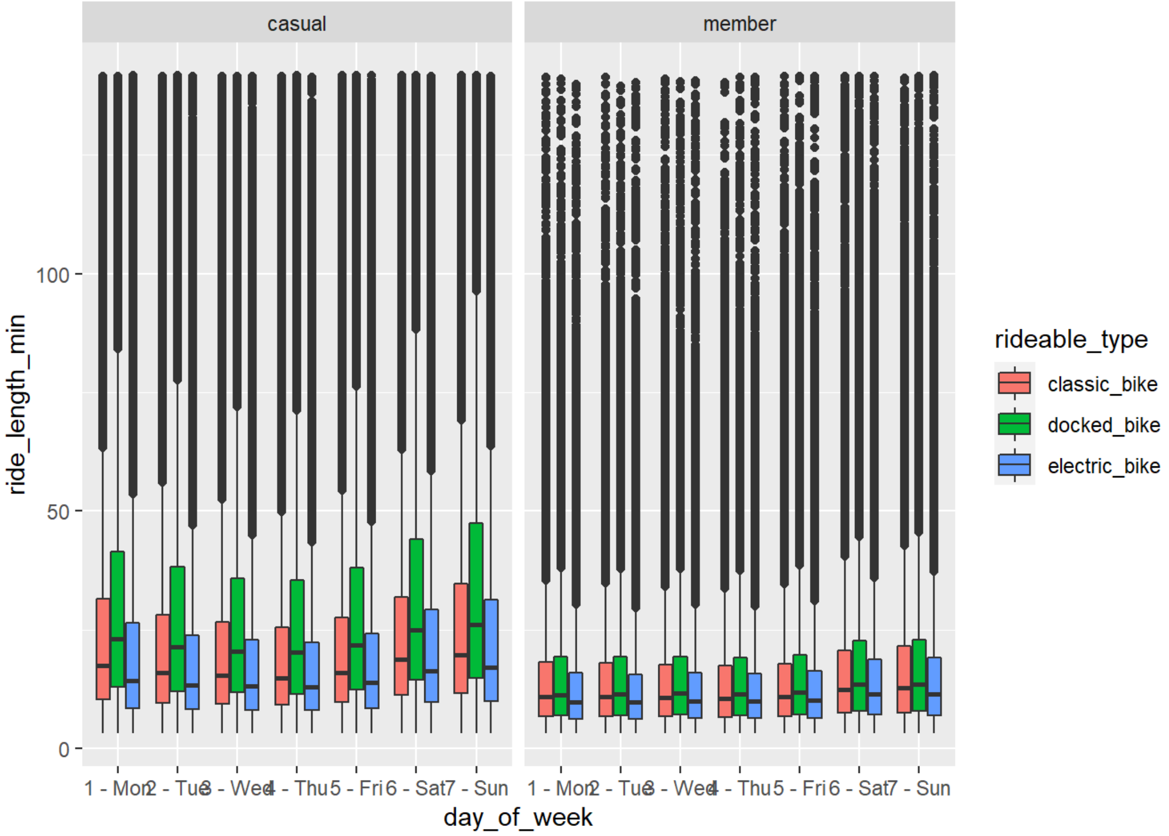
```
ggplot(trips_total_no_outliers, aes(x=member_casual, y=ride_length_min, fill=member_casual))+
  geom_boxplot()+
  labs(title = "Distribution of Ride Length by Membership", x="Membership Type", y="Ride Length (Mins)")
```



Similar to the earlier findings, casuals have a greater riding time as compared to members. Casuals also have a greater mean ride length as well as a greater interquartile range.

Combining Ride length with Day of Week and Membership

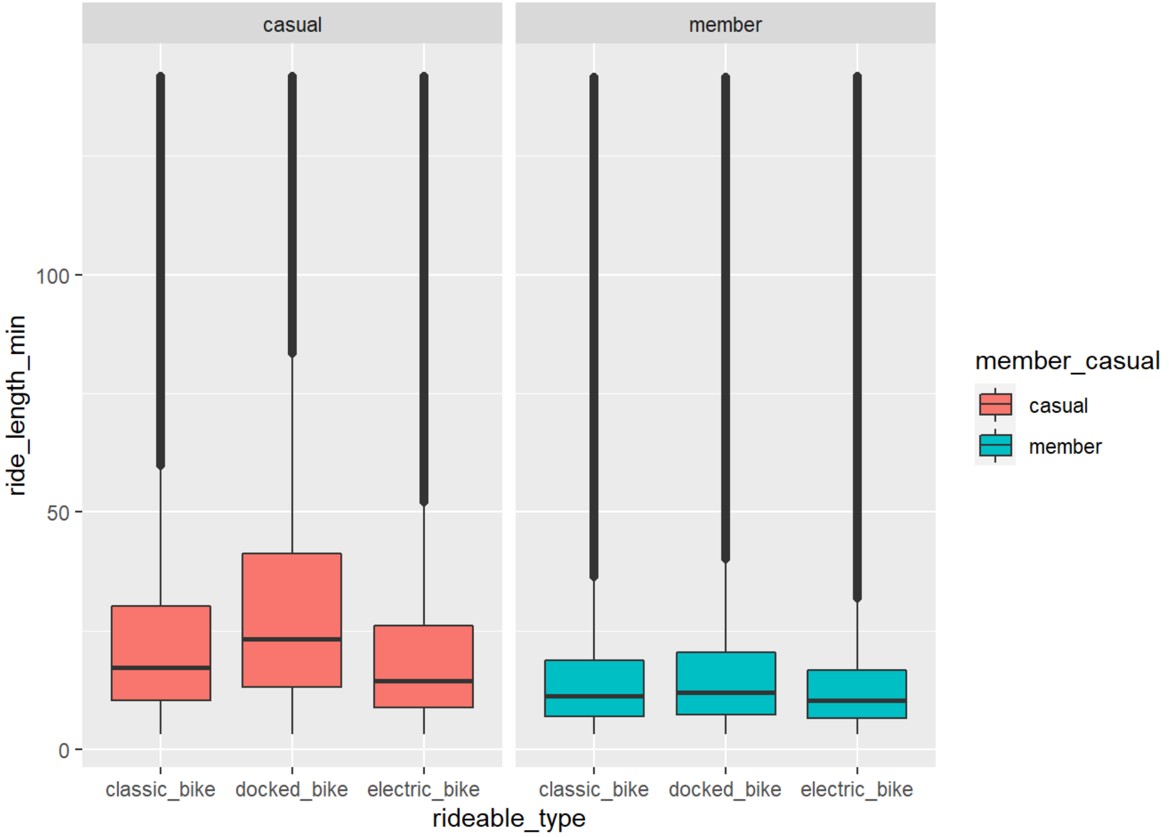
```
ggplot(trips_total_no_outliers, aes(x=day_of_week, y=ride_length_min, fill=rideable_type))+
  geom_boxplot()+
  facet_wrap(~member_casual)
```



Both groups demonstrate different patterns in terms of ride length over the week. For members, ride length is more stable and gradually increases as the week progresses. For casuals, the ride length follows a U shaped curve.

Combining Ride length with Type of Bike and Membership

```
ggplot(trips_total_no_outliers, aes(x=rideable_type, y=ride_length_min, fill=member_casual))+
  geom_boxplot()+
  facet_wrap(~member_casual)
```



Overall, docked bike has the longest ride duration for both groups. Casual has an overall longer duration for all 3 bikes compared to members With greater means and IQRs.

Trip Distance

The next variable to examine would be trip distance. Firstly, we would obtain some summary statistics for this variable.

```
summary(trips_total_no_outliers$trip_distance_km)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##  0.0000  0.4386  1.0988  1.5042  2.1053 20.0039
```

The minimum value of 0km does not make sense. The maximum value of 20km might also be unlikely. There is a need to check for outliers.

```
quantile(trips_total_no_outliers$trip_distance_km, probs = seq(0, 1, 0.05))
```

```
##          0%          5%         10%         15%         20%         25% 
##  0.00000000  0.00000000  0.08103802  0.17545415  0.29977613  0.43856161 
##         30%         35%         40%         45%         50%         55% 
##  0.54582846  0.67082716  0.82342187  0.96933035  1.09876536  1.25662429 
##         60%         65%         70%         75%         80%         85% 
##  1.43630654  1.62728297  1.84157988  2.10529248  2.42156482  2.83910492 
##         90%         95%        100% 
##  3.45158068  4.48526237 20.00393054
```

Need to further breakdown the 5th-10th percentile and 95th-100th percentile for a closer look.

```
# Breaking down the values in the 95th-100th percentile range

quantile(trips_total_no_outliers$trip_distance_km, probs = seq(0.95, 1, 0.01))
```

```
##       95%       96%       97%       98%       99%      100% 
##  4.485262  4.801202  5.214404  5.754631  6.728762 20.003931
```

```
# Breaking down the values in the 5th-10th percentile range

quantile(trips_total_no_outliers$trip_distance_km, probs = seq(0.05, 0.1, 0.01))
```

```
##           5%           6%           7%           8%           9%          10% 
## 0.000000e+00 9.657553e-05 2.286034e-02 4.670914e-02 6.165670e-02 8.103802e-02
```

```
# Saving the percentile values

percentile_dist <- quantile(trips_total_no_outliers$trip_distance_km, probs = seq(0, 1, 0.01))
```

Data on both ends might not be informative, especially those with 0 value. The outlier data would be removed and the analysis would be using a subset of the data.

```
trips_total_no_outliers <- trips_total_no_outliers %>%
  filter(trip_distance_km > percentile_dist["10%"]) %>%
  filter(trip_distance_km < percentile_dist["99%"])

print(paste("Removed", num_of_rows_v1-nrow(trips_total_no_outliers), "rows as outliers")) ## Number of ro
ws removed
```

```
## [1] "Removed 429879 rows as outliers"
```

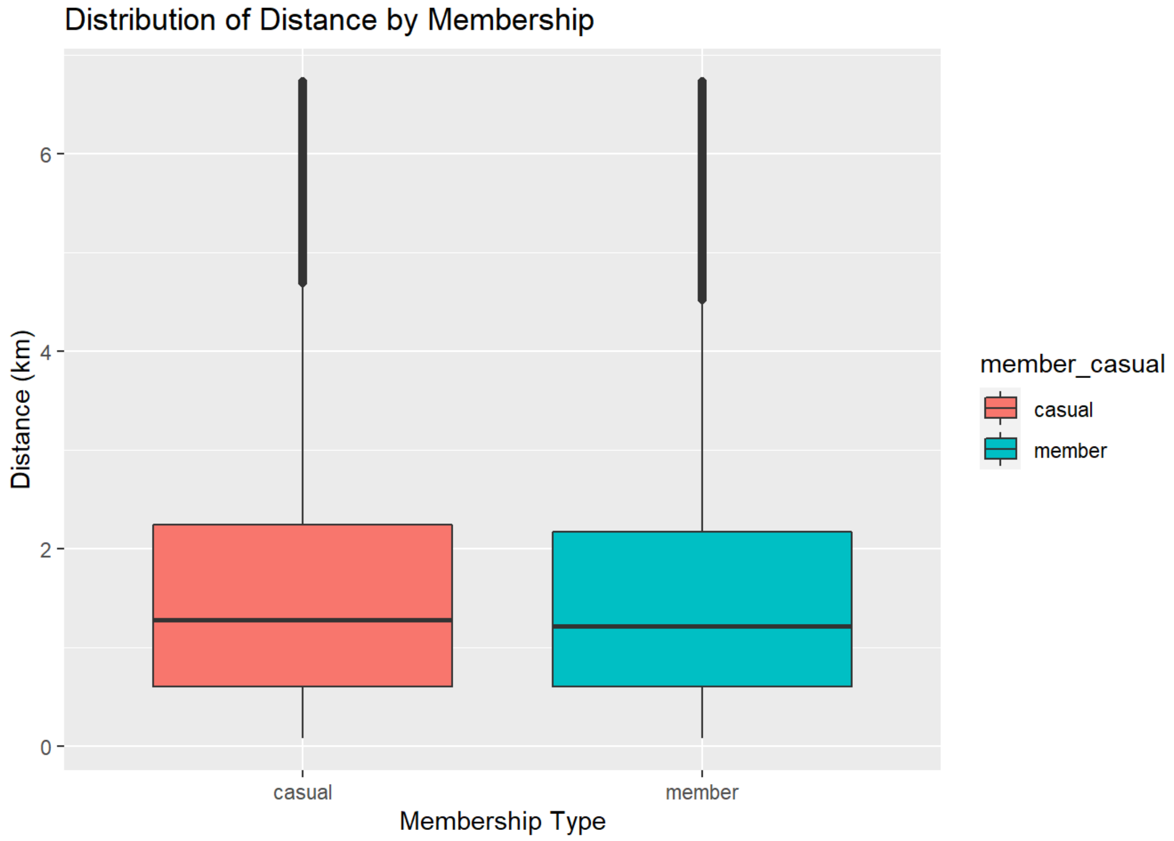
Distribution of trip distance by membership

Breaking down the trip distance within each group

```
trips_total_no_outliers %>%
  group_by(member_casual) %>%
  summarise(mean = mean(trip_distance_km),
            "first_q" = quantile(trip_distance_km, 0.25),
            median = median(trip_distance_km),
            "third_q" = quantile(trip_distance_km, 0.75),
            IQR = third_q - first_q)
```

```
## # A tibble: 2 x 6
##   member_casual  mean first_q median third_q   IQR
##   <chr>         <dbl>   <dbl>  <dbl>   <dbl> <dbl>
## 1 casual         1.62   0.601   1.28    2.24  1.63
## 2 member         1.58   0.597   1.21    2.16  1.57
```

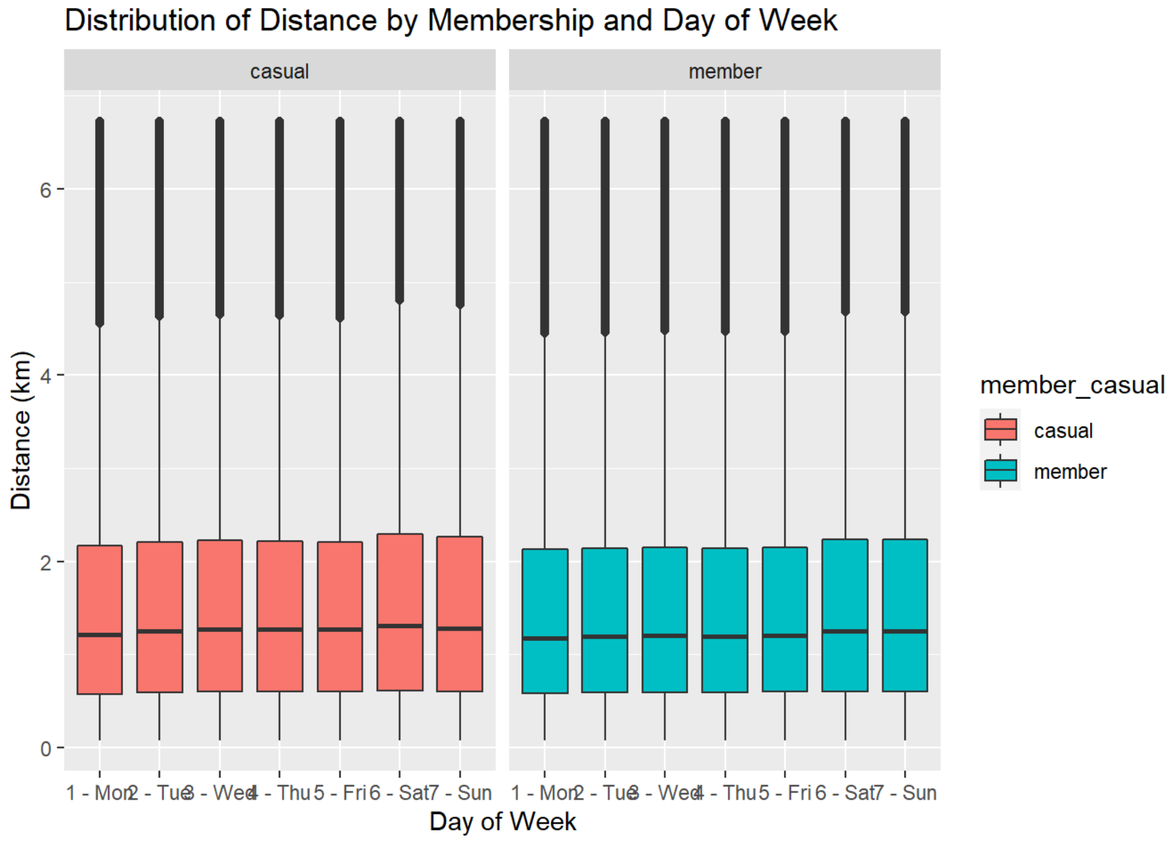
```
ggplot(trips_total_no_outliers, aes(x=member_casual, y=trip_distance_km, fill=member_casual))+
  geom_boxplot()+
  labs(title = "Distribution of Distance by Membership", x="Membership Type", y="Distance (km)")
```



Casuals have a slightly larger trip distance than members.

Combining Trip Distance, Day of week and Membership

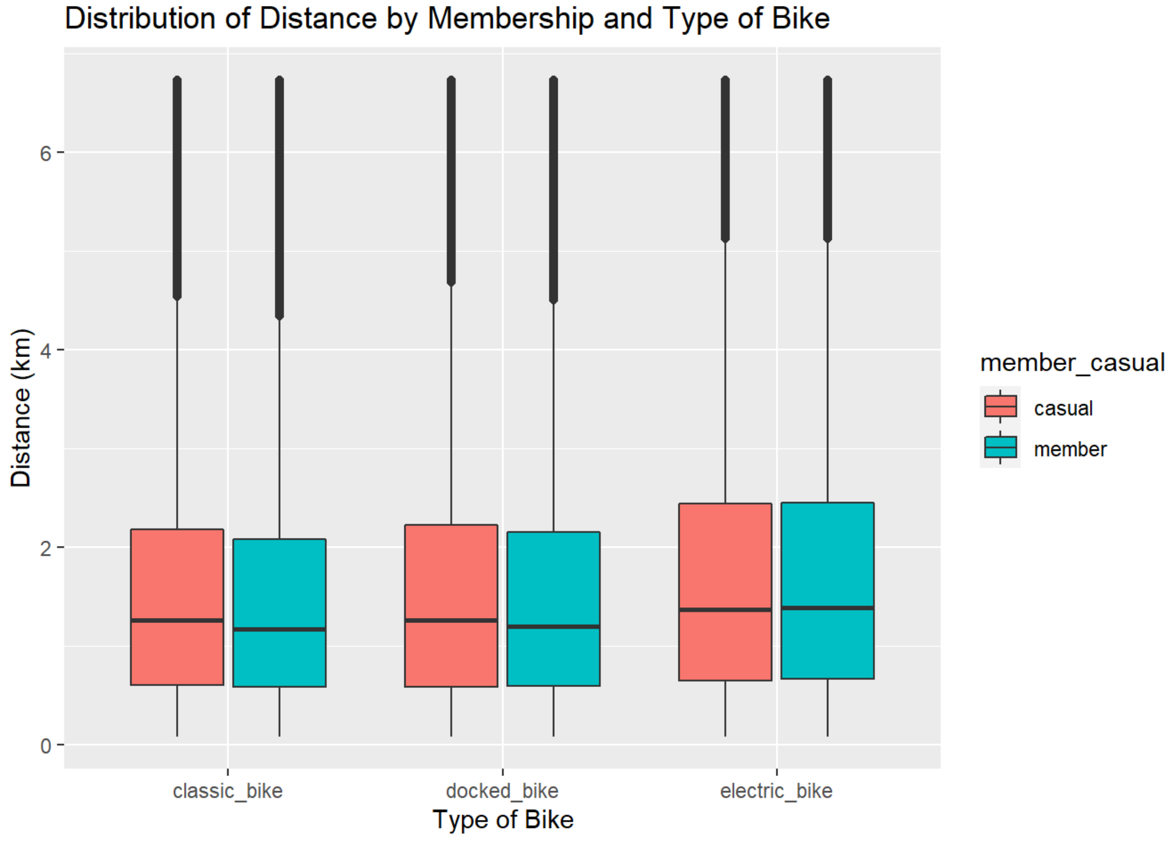
```
ggplot(trips_total_no_outliers, aes(x=day_of_week, y=trip_distance_km, fill=member_casual))+
  geom_boxplot()+
  labs(title = "Distribution of Distance by Membership and Day of Week", x="Day of Week", y="Distance (k
m)")+
  facet_wrap(~member_casual)
```



Both groups have similar patterns in terms of trip distance. Trip distance is quite stable over weekdays, with slight increase on weekends.

Combining Trip Distance, Type of Bike and Membership

```
ggplot(trips_total_no_outliers, aes(x=rideable_type, y=trip_distance_km, fill=member_casual))+
  geom_boxplot()+
  labs(title = "Distribution of Distance by Membership and Type of Bike", x="Type of Bike", y="Distance
 (km)")
```



Electric bike has the greatest trip distance for both groups and it is comparable between both groups.

For other bike types, Casuals have greater trip distance than members.

Exporting file for further analysis

```
alltrips <- trips_total_no_outliers

write.csv(alltrips, file = "all_trips.csv", row.names = FALSE)
```

Summary of Analysis

Before the analysis, the relevant data from each month are combined together into one dataset. The columns also contain the correct data type.

Main Findings

- There are more members than casuals in this dataset.
- There were more rides in the first half of 2021 (Jan - July) as compared to the second half of 2020 (Aug - Dec). + There are both more members and casuals in 2021 (Jan - July) as compared to 2020 (Aug - Dec).
- The number of rides generally increases from January onwards, peaking at July, before decreasing towards the end of the year.
- In all months except July, there were more members than casuals.
- The monthly number of rides tend to be influenced by external factors, such as weather.
- Weekends have greater number of rides than weekdays, with Saturday having the most number of rides.
  - Saturday also has the highest number of casuals.
  - On the other hand, wednesday has the highest number of members.
  - On weekdays, there are more members than casuals. However, on weekends, there are more casuals than members.
- There is an overall higher number of rides in the afternoon and early hours of the evening, with the number of rides peaking at 5pm.
- On weekdays, there are significant increases in number of rides at certain periods of time (eg 4-5pm).
- Members generally prefer classic bikes while casuals generally prefer docked bikes.
- Casuals have greater ride length than members.
- The ride length of casuals are generally lower during weekdays and significantly increase during weekends.
- Members have more stable ride length throughout the week, with ride lengths on weekdays surpassing weekends.

One surprise from this data would be the differences in behaviours between casuals and members in terms of bike usage.

These insights should help to craft separate profiles that would give a deeper understanding of both groups.

Share

The data highlights the main differences between casuals and members. The main story that the data is telling would be both casuals and members belong to different demographics. The various differences highlighted by the data suggests that both members and casuals use bikes for different purposes.

Members incorporate bike usage as part of their daily lives, utilising them for everyday activities such as going to work. This is supported by the findings such as members taking up greater proportions of rides for most of the day, significant increases in rides at certain times (such as 5am - 6am) and stable ride lengths throughout the week.The data suggests a fixed routine usage of bikes by members throughout the week.

On the other hand, casuals would more likely be using bikes for recreational purposes.This is evident from findings such as the huge influx of casuals on weekends, significantly higher ride lengths on weekends than on weekdays, and higher proportions of casuals during odd hours of the day.

By expounding on these behavioural differences from the data, we could build separate profiles for the target groups to tackle the business problem. The main findings from the data can then be visualised and presented in an accessible way to the Cyclistic marketing analytics team, Lily Moreno and Cyclistic executive team.

Act

The marketing team would then utilise these insights and recommendations to further enhance the marketing strategy. The insights could be implemented or incorporated when designing the marketing campaign to convert casuals to members. These insights could also form the foundational elements of the marketing campaign.

Additionally, further information and analysis could be conducted to enhance these findings. For instance, additional demographic information such as gender, age etc. and other information could include Chicago climate data, data on popular routes taken could be used to conduct further analysis.

Recommendations

1. Encourage usage of bikes for recreational activities not just solely on weekends, but on weekdays as well to encourage greater usage of bikes by casuals.
2. Incorporate elements such as demonstrating the many benefits of using Cyclistic bikes for everyday usage such as a means of transportation to work.
3. Provide incentives for existing members to encourage casuals to convert to memberships such as referral codes.
4. Capitalise on the popular stations frequented by casuals and incorporate them into the marketing strategy to further increase exposure.

References

For this project, I have consulted the following references:

Jhelisonuchoa. (2021, June 4). Google data analytics capstone - Case study 1. Kaggle: Your Machine Learning and Data Science Community. https://www.kaggle.com/jhelisonuchoa/google-data-analytics-capstone-case-study-1/notebook (https://www.kaggle.com/jhelisonuchoa/google-data-analytics-capstone-case-study-1/notebook)

US Dept of Commerce National Oceanic and Atmospheric Administration National Weather Service. (n.d.). Climate. National Weather Service. https://www.weather.gov/wrh/climate?wfo=lot (https://www.weather.gov/wrh/climate?wfo=lot)

Tan, W. H. (2021, June 9). Google data analytics course capstone project. LinkedIn. https://www.linkedin.com/pulse/google-data-analytics-course-capstone-project-wen-hao-tan (https://www.linkedin.com/pulse/google-data-analytics-course-capstone-project-wen-hao-tan)