Cyclistic Capstone Project

Marwa Rabia

2023-11-27

Introduction

This is my Capstone Project for my Google Data Analytics Course. This project is based on an imaginary company called Cyclistic and all data related to the company has been provided in the course itself.

About the company

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime.

Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes, and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members.

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

Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

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.

Scenario

Cyclistic, a fictional bike-sharing company situated in Chicago, Illinois, enlisted me as a data analyst to contribute insights to the marketing team's strategic initiatives. Within the customer base, Cyclistic caters to two primary segments: annual members and casual riders. The director of marketing recognizes the pivotal role of increasing annual memberships in ensuring the company's future success.

Tasked with crafting a compelling strategy to convert casual riders into annual members, the marketing team seeks data-driven insights into the distinctive usage patterns of these two customer groups. Specifically, we aim to understand how casual riders and annual members differ in their utilization of Cyclistic bikes. In response to the director's directive, the focus of the proposed strategy will be on leveraging digital media channels, such as email, social media, and other online platforms, to effectively communicate and encourage the transition from casual riding to annual memberships.

This Case Study will have 6 steps in total to find solutions to our required Business task. The 6 steps are: **Ask, Prepare, Process, Analyze, Share, Act.**

Step 1 Ask (Business Questions)

The Goal of the Marketing director is to increase the number of annual memberships by converting more and more casual riders to increase profitability of the Company.

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

Our Business Task:

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

Step 2 : Preparing the Data

Information About The Data

- The company Cyclist is a fictional company; hence we are using data of a real world company to answer questions related to our business task.
- The following datasets were provided in the case study only Data link: Divvy trip data public dataset
- We have considered data for Months **January 2023 – June 2023.**

Licensing, Privacy, Reliability

- The data comes from a reliable source, without any bias as it is a real time data of a company which is from recent months.
- This data stands secure in an AWS portal and is an open dataset.
- The data has been made available by Motivate International Inc. under this license.

Tools Used

- I have used R Programming Language for the data cleaning and data processing phase.

Setting up my environment

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.3     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.4     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.0
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(lubridate)
library(ggplot2)
library(janitor)
```

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

```
library(skimr)
library(dplyr)
```

Loading .csv files

```
m1_2023 <- read.csv('Data\\202301-divvy-tripdata.csv')
m2_2023 <- read.csv('Data\\202302-divvy-tripdata.csv')
m3_2023 <- read.csv('Data\\202303-divvy-tripdata.csv')
m4_2023 <- read.csv('Data\\202304-divvy-tripdata.csv')
m5_2023 <- read.csv('Data\\202305-divvy-tripdata.csv')
m6_2023 <- read.csv('Data\\202306-divvy-tripdata.csv')
```

WRANGLE DATA AND COMBINE INTO A SINGLE FILE¶

Compare column names each of the files

```
all(colnames(m1_2023) == colnames(m2_2023)&
    colnames(m1_2023) == colnames(m3_2023)&
    colnames(m1_2023) == colnames(m4_2023)&
    colnames(m1_2023) == colnames(m5_2023)&
    colnames(m1_2023) == colnames(m6_2023))
```

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

Since the column are consistent we can go ahead with the combining

```
all_data <- bind_rows(m1_2023,m2_2023,m3_2023,m4_2023,m5_2023,m6_2023)
```

Save All data as CSV

```
write.csv(all_data, file = "Data_before_cleaning.csv", row.names = FALSE)
```

Understanding The Data

```
head(all_data) # Checking the first few rows of the dataset
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 F96D5A74A3E41399 electric_bike 2023-01-21 20:05:42 2023-01-21 20:16:33
## 2 13CB7EB698CEDB88  classic_bike 2023-01-10 15:37:36 2023-01-10 15:46:05
## 3 BD88A2E670661CE5 electric_bike 2023-01-02 07:51:57 2023-01-02 08:05:11
## 4 C90792D034FED968  classic_bike 2023-01-22 10:52:58 2023-01-22 11:01:44
## 5 3397017529188E8A  classic_bike 2023-01-12 13:58:01 2023-01-12 14:13:20
## 6 58E68156DAE3E311 electric_bike 2023-01-31 07:18:03 2023-01-31 07:21:16
##              start_station_name start_station_id              end_station_name
## 1   Lincoln Ave & Fullerton Ave     TA1309000058      Hampden Ct & Diversey Ave
## 2         Kimbark Ave & 53rd St     TA1309000037        Greenwood Ave & 47th St
## 3        Western Ave & Lunt Ave           RP-005 Valli Produce - Evanston Plaza
## 4         Kimbark Ave & 53rd St     TA1309000037        Greenwood Ave & 47th St
## 5         Kimbark Ave & 53rd St     TA1309000037        Greenwood Ave & 47th St
## 6 Lakeview Ave & Fullerton Pkwy     TA1309000019      Hampden Ct & Diversey Ave
##   end_station_id start_lat start_lng  end_lat   end_lng member_casual
## 1       202480.0  41.92407 -87.64628 41.93000 -87.64000        member
## 2   TA1308000002  41.79957 -87.59475 41.80983 -87.59938        member
## 3            599  42.00857 -87.69048 42.03974 -87.69941        casual
## 4   TA1308000002  41.79957 -87.59475 41.80983 -87.59938        member
## 5   TA1308000002  41.79957 -87.59475 41.80983 -87.59938        member
## 6       202480.0  41.92607 -87.63886 41.93000 -87.64000        member
```

```
summary(all_data) # To Understand statistics about the Data set
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:2390459     Length:2390459     Length:2390459     Length:2390459
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##  start_station_name start_station_id   end_station_name   end_station_id
##  Length:2390459     Length:2390459     Length:2390459     Length:2390459
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##    start_lat       start_lng         end_lat         end_lng
##  Min.   :41.64   Min.   :-87.87   Min.   : 0.00   Min.   :-88.16
##  1st Qu.:41.88   1st Qu.:-87.66   1st Qu.:41.88   1st Qu.:-87.66
##  Median :41.90   Median :-87.64   Median :41.90   Median :-87.64
##  Mean   :41.90   Mean   :-87.65   Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63   3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.07   Max.   :-87.52   Max.   :42.11   Max.   :  0.00
##                                   NA's   :2460    NA's   :2460
##  member_casual
##  Length:2390459
##  Class :character
##  Mode  :character
##
##
##
##
```

```
colnames(all_data) #Column names of the Data set
```

```
##  [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"
```

```
skim_without_charts(all_data)   # 2387999
```

Data summary

Name	all_data
Number of rows	2390459
Number of columns	13

Column type frequency:	
character	9
numeric	4
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	2390459	0
rideable_type	0	1	11	13	0	3	0
started_at	0	1	19	19	0	2049649	0
ended_at	0	1	19	19	0	2055268	0
start_station_name	0	1	0	64	357417	1461	0
start_station_id	0	1	0	35	357549	1388	0
end_station_name	0	1	0	64	380963	1467	0
end_station_id	0	1	0	35	381104	1392	0
member_casual	0	1	6	6	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	42.07
start_lng	0	1	-87.65	0.03	-87.87	-87.66	-87.64	-87.63	-87.52
end_lat	2460	1	41.90	0.06	0.00	41.88	41.90	41.93	42.11
end_lng	2460	1	-87.65	0.08	-88.16	-87.66	-87.64	-87.63	0.00

```
glimpse(all_data) #Summary of the Data set and Data Types of Columns
```

```
## Rows: 2,390,459
## Columns: 13
## $ ride_id            <chr> "F96D5A74A3E41399", "13CB7EB698CEDB88", "BD88A2E670…
## $ rideable_type      <chr> "electric_bike", "classic_bike", "electric_bike", "…
## $ started_at         <chr> "2023-01-21 20:05:42", "2023-01-10 15:37:36", "2023…
## $ ended_at           <chr> "2023-01-21 20:16:33", "2023-01-10 15:46:05", "2023…
## $ start_station_name <chr> "Lincoln Ave & Fullerton Ave", "Kimbark Ave & 53rd …
## $ start_station_id   <chr> "TA1309000058", "TA1309000037", "RP-005", "TA130900…
## $ end_station_name   <chr> "Hampden Ct & Diversey Ave", "Greenwood Ave & 47th …
## $ end_station_id     <chr> "202480.0", "TA1308000002", "599", "TA1308000002", …
## $ start_lat          <dbl> 41.92407, 41.79957, 42.00857, 41.79957, 41.79957, 4…
## $ start_lng          <dbl> -87.64628, -87.59475, -87.69048, -87.59475, -87.594…
## $ end_lat            <dbl> 41.93000, 41.80983, 42.03974, 41.80983, 41.80983, 4…
## $ end_lng            <dbl> -87.64000, -87.59938, -87.69941, -87.59938, -87.599…
## $ member_casual      <chr> "member", "member", "casual", "member", "member", "…
```

Observations about the Data set

- Total number of rows = 2,390,459 and columns = 13.
- Each row represents a unique entry and there are no duplicate values.
- There are 2 unique types of Customers and 3 unique types of bikes.

Problems Noted

- Columns having missing values : start_station_name, start_station_id, end_station_name, end_station_id, end_lat, end_lng.
- Some Columns have to be renamed as they are not intuitive.
- The data can only be aggregated at the ride-level. We will want to add some additional columns of data – such as day, month, year – that provide additional opportunities to aggregate the data.
- We will add a column ride length to understand the amount of time that has been spent by a user in a particular ride.

Step 3 Processing the Data

Data Manipulation

Renaming Column

```
all_data <- rename(all_data, "user_type" = "member_casual")
```

Convert started_at and ended_at to dttm

```
all_data<-mutate(all_data,started_at=ymd_hms(started_at),
                          ended_at =ymd_hms(ended_at))
```

Adding Columns

```
all_data$ride_length<-difftime(all_data$ended_at,all_data$started_at)
all_data$date <- as.Date(all_data$started_at) #The default format is yyyy-mm-dd
all_data$month <- format(as.Date(all_data$date), "%b")
all_data$day <- format(as.Date(all_data$date), "%d")
all_data$year <- format(as.Date(all_data$date), "%Y")
all_data$day_of_week <- format(as.Date(all_data$date), "%A")
```

Changing the Data Type of ride_length to numeric for ease of calculation of data

```
all_data$date <- as.numeric(all_data$ride_length)
```

Clean Data

remove duplicate rows

```
all_data <- distinct(all_data)
```

remove rows with null values

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

- Negative ride_length values and outliers
- ride_length values that are negative are not correct as the difference between end time and start time has to be a positive value.
- If the ride_length values are less than 60 seconds or more than 86,400 seconds (24 hours), then they will be considered outliers as mentioned in the Website.

```
all_clean_data <- filter(all_data, !(ride_length<60), !(ride_length>86400))
```

```
min(all_clean_data$ride_length)
```

```
## Time difference of 60 secs
```

```
max(all_clean_data$ride_length)
```

```
## Time difference of 86146 secs
```

Checking the changes that we made to our Data

```
skim_without_charts(all_clean_data)
```

Data summary

Name	all_clean_data
Number of rows	2315029

Number of columns	19
_____	
Column type frequency:	
character	11
difftime	1
numeric	5
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	2315029	0
rideable_type	0	1	11	13	0	3	0
start_station_name	0	1	0	64	336188	1454	0
start_station_id	0	1	0	35	336313	1381	0
end_station_name	0	1	0	64	350091	1454	0
end_station_id	0	1	0	35	350229	1381	0
user_type	0	1	6	6	0	2	0
month	0	1	3	3	0	6	0
day	0	1	2	2	0	31	0
year	0	1	4	4	0	1	0
day_of_week	0	1	6	9	0	7	0

Variable type: difftime

skim_variable	n_missing	complete_rate	min	max	median	n_unique
ride_length	0	1	60 secs	86146 secs	569 secs	15042

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
start_lat	0	1	41.90	0.05	41.64	41.88	41.90	41.93	42.07
start_lng	0	1	-87.65	0.03	-87.87	-87.66	-87.64	-87.63	-87.52

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
end_lat	0	1	41.90	0.05	0.00	41.88	41.90	41.93	42.11
end_lng	0	1	-87.65	0.06	-88.16	-87.66	-87.64	-87.63	0.00
date	0	1	906.51	1825.66	60.00	331.00	569.00	1008.00	86146.00

Variable type: POSIXct

skim_variable	n_missing	complete_rate	min	max	median	n_unique
started_at	0	1	2023-01-01 00:02:06	2023-06-30 23:59:56	2023-05-09 01:49:51	1993158
ended_at	0	1	2023-01-01 00:07:23	2023-07-01 18:26:01	2023-05-09 02:23:25	1998671

```
colnames(all_clean_data)
```

```
##  [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] "user_type"          "ride_length"        "date"
## [16] "month"              "day"                "year"
## [19] "day_of_week"
```

```
summary(all_clean_data)
```

```
##    ride_id          rideable_type        started_at
##  Length:2315029     Length:2315029     Min.   :2023-01-01 00:02:06.00
##  Class :character   Class :character   1st Qu.:2023-03-28 08:21:52.00
##  Mode  :character   Mode  :character   Median :2023-05-09 01:49:51.00
##                                        Mean   :2023-04-27 07:36:30.43
##                                        3rd Qu.:2023-06-05 17:02:55.00
##                                        Max.   :2023-06-30 23:59:56.00
##    ended_at                      start_station_name start_station_id
##  Min.   :2023-01-01 00:07:23.00   Length:2315029     Length:2315029
##  1st Qu.:2023-03-28 08:32:38.00   Class :character   Class :character
##  Median :2023-05-09 02:23:25.00   Mode  :character   Mode  :character
##  Mean   :2023-04-27 07:51:36.94
##  3rd Qu.:2023-06-05 17:18:30.00
##  Max.   :2023-07-01 18:26:01.00
##  end_station_name   end_station_id       start_lat       start_lng
##  Length:2315029     Length:2315029     Min.   :41.64   Min.   :-87.87
##  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   :-87.65
##                                        3rd Qu.:41.93   3rd Qu.:-87.63
##                                        Max.   :42.07   Max.   :-87.52
##    end_lat         end_lng        user_type         ride_length
##  Min.   : 0.00   Min.   :-88.16   Length:2315029     Length:2315029
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character   Class :difftime
##  Median :41.90   Median :-87.64   Mode  :character   Mode  :numeric
##  Mean   :41.90   Mean   :-87.65
##  3rd Qu.:41.93   3rd Qu.:-87.63
##  Max.   :42.11   Max.   :  0.00
##       date            month               day                year
##  Min.   :   60.0   Length:2315029     Length:2315029     Length:2315029
##  1st Qu.:  331.0   Class :character   Class :character   Class :character
##  Median :  569.0   Mode  :character   Mode  :character   Mode  :character
##  Mean   :  906.5
##  3rd Qu.: 1008.0
##  Max.   :86146.0
##  day_of_week
##  Length:2315029
##  Class :character
##  Mode  :character
##
##
##
```

```
sample_n(all_clean_data,50)
```

```
##             ride_id rideable_type          started_at            ended_at
## 1  01E41A6DB55CEBA2  classic_bike 2023-04-27 20:49:26 2023-04-27 20:53:11
## 2  DE7CB1FFA56897CD electric_bike 2023-06-04 14:38:21 2023-06-04 14:46:49
## 3  75FAC1AE935412B3 electric_bike 2023-03-06 21:26:36 2023-03-06 21:33:23
## 4  65C54A3EA073186C  classic_bike 2023-05-24 06:46:31 2023-05-24 06:59:42
## 5  1530C9538003120A  classic_bike 2023-04-05 17:50:26 2023-04-05 17:53:53
## 6  C34DE632C08E19F6 electric_bike 2023-04-19 17:22:28 2023-04-19 17:35:22
## 7  F33EC185BCA8A2F7  classic_bike 2023-02-06 21:56:00 2023-02-06 21:58:30
## 8  34F245E602CEDA9E electric_bike 2023-02-19 18:10:04 2023-02-19 18:55:25
## 9  FA0682417321F4DF electric_bike 2023-05-08 23:14:49 2023-05-08 23:18:31
## 10 0955E13F66E53D7E electric_bike 2023-06-22 08:02:43 2023-06-22 08:14:53
## 11 2806A72127E55D16 electric_bike 2023-03-24 16:54:03 2023-03-24 16:59:24
## 12 75ADB276AD5CC2FF  classic_bike 2023-05-24 06:35:03 2023-05-24 06:36:47
## 13 F14EC65532DAE6B6  classic_bike 2023-06-27 06:25:38 2023-06-27 10:36:40
## 14 AF062678ECF716DC electric_bike 2023-03-08 09:07:22 2023-03-08 09:12:48
## 15 D4EEFE4781A7AFF7 electric_bike 2023-04-13 17:34:05 2023-04-13 17:50:04
## 16 F673CC9EA9ACA303 electric_bike 2023-03-21 18:27:50 2023-03-21 18:30:04
## 17 624442A839E62E9D  classic_bike 2023-02-13 08:22:39 2023-02-13 08:29:57
## 18 D2971100A8C1AD76  classic_bike 2023-06-21 07:41:47 2023-06-21 08:03:02
## 19 FA3190245EAD852A electric_bike 2023-05-19 03:23:45 2023-05-19 03:48:43
## 20 5BAAC9235ED11472  classic_bike 2023-02-20 19:46:31 2023-02-20 19:58:41
## 21 2BBB741C10CD23F1 electric_bike 2023-05-22 09:05:28 2023-05-22 09:11:13
## 22 C12585BCEE77A6B7  classic_bike 2023-05-14 11:47:54 2023-05-14 11:57:34
## 23 8DB5783A84A5B7D2  classic_bike 2023-05-20 23:07:29 2023-05-20 23:29:14
## 24 50F1909CE6A9EA2D electric_bike 2023-06-27 16:32:43 2023-06-27 16:37:49
## 25 D0D082D2EC6CE242 electric_bike 2023-03-24 18:12:12 2023-03-24 18:27:27
## 26 571726D635EF1E34 electric_bike 2023-05-22 17:26:21 2023-05-22 17:32:32
## 27 0C8BC6980286E24A  classic_bike 2023-06-06 17:55:22 2023-06-06 18:19:55
## 28 7EF7908F3BC94DEC electric_bike 2023-05-20 13:13:43 2023-05-20 13:20:55
## 29 78DF55E55BA85C7E electric_bike 2023-04-17 06:27:57 2023-04-17 06:30:24
## 30 CBE1618B29543664 electric_bike 2023-03-21 15:56:17 2023-03-21 16:07:45
## 31 D20653FD55CD07DE electric_bike 2023-03-08 08:35:59 2023-03-08 08:39:50
## 32 B32FEFFD47B81AE5  classic_bike 2023-06-25 17:33:08 2023-06-25 17:37:13
## 33 E32E74D7B5AAEF74  classic_bike 2023-06-18 19:16:25 2023-06-18 19:46:02
## 34 988F77B3EAB61C65  classic_bike 2023-04-19 10:13:34 2023-04-19 10:19:54
## 35 1100F2E8868561FF  classic_bike 2023-02-07 07:52:09 2023-02-07 07:59:50
## 36 67FFBDD03354AD9E  classic_bike 2023-06-21 21:41:17 2023-06-21 22:08:07
## 37 10BD7D3774D3960D  classic_bike 2023-01-12 19:01:01 2023-01-12 19:04:14
## 38 DC406F9EA051606B   docked_bike 2023-05-03 13:20:47 2023-05-03 14:56:45
## 39 A241623493A2BDD9 electric_bike 2023-05-10 16:17:10 2023-05-10 16:37:13
## 40 8AF655832D45E8CD  classic_bike 2023-05-16 20:01:51 2023-05-16 20:13:41
## 41 C37F8D47D15E3120  classic_bike 2023-01-31 19:08:08 2023-01-31 19:18:04
## 42 4D18A94D3FD168DF electric_bike 2023-05-22 11:27:12 2023-05-22 11:44:09
## 43 4398D1732A46DA84 electric_bike 2023-06-02 19:33:06 2023-06-02 19:49:35
## 44 DBE8FE2025A9A4FF  classic_bike 2023-03-04 10:30:34 2023-03-04 10:37:39
## 45 9321F3B6272352EB electric_bike 2023-06-23 22:33:57 2023-06-23 22:40:58
## 46 1A01EDDADCA62B16 electric_bike 2023-04-28 19:32:43 2023-04-28 19:38:38
## 47 1E47D013D3B0706C  classic_bike 2023-04-07 07:21:36 2023-04-07 07:24:59
## 48 575742BF423D04BA electric_bike 2023-02-12 16:25:26 2023-02-12 16:47:37
## 49 C79616FB6BF53EF1 electric_bike 2023-06-10 14:00:39 2023-06-10 14:03:24
## 50 085CD9A0DBC9A517  classic_bike 2023-06-03 16:51:40 2023-06-03 17:03:31
##                       start_station_name start_station_id
```

```
## 1                    McClurg Ct & Ohio St     TA1306000029
## 2
## 3                     Perry Ave & 69th St     KA1503000047
## 4               Larrabee St & Menomonee St     TA1306000007
## 5                     Ellis Ave & 60th St     KA1503000014
## 6                      Shore Dr & 55th St     TA1308000009
## 7              Wilton Ave & Diversey Pkwy*     chargingstx0
## 8                   Clark St & Berwyn Ave     KA1504000146
## 9               Greenview Ave & Jarvis Ave              520
## 10              Ashland Ave & Blackhawk St            13224
## 11              Racine Ave & Fullerton Ave     TA1306000026
## 12              Aberdeen St & Jackson Blvd             13157
## 13           Sangamon St & Washington Blvd             13409
## 14              Kedzie Ave & Milwaukee Ave             13085
## 15                Elston Ave & Cortland St     TA1305000039
## 16     Mies van der Rohe Way & Chestnut St             15529
## 17                Carpenter St & Huron St             13196
## 18               Sedgwick St & Webster Ave             13191
## 19                   Wells St & Concord Ln     TA1308000050
## 20                   Wells St & Hubbard St     TA1307000151
## 21                Franklin St & Monroe St     TA1309000007
## 22                Halsted St & Maxwell St     TA1309000001
## 23             Sheffield Ave & Webster Ave     TA1309000033
## 24
## 25
## 26                   Broadway & Argyle St             13108
## 27                    Rush St & Hubbard St     KA1503000044
## 28                  State St & Randolph St     TA1305000029
## 29               Sheffield Ave & Willow St     TA1306000032
## 30                   Wells St & Hubbard St     TA1307000151
## 31               Western Ave & Fillmore St              644
## 32                    McClurg Ct & Ohio St     TA1306000029
## 33                   Wabash Ave & Adams St     KA1503000015
## 34                   Ashland Ave & Lake St             13073
## 35 Wolcott (Ravenswood) Ave & Montrose Ave     TA1307000144
## 36               Halsted St & Clybourn Ave              331
## 37                  Clinton St & Tilden St             13037
## 38                Lake Park Ave & 35th St     KA1503000004
## 39
## 40               Franklin St & Illinois St              RN-
## 41                   LaSalle Dr & Huron St     KP1705001026
## 42                         Millennium Park             13008
## 43
## 44                Western Ave & Walton St     KA1504000103
## 45                   Long Ave & North Ave              375
## 46                Wood St & Milwaukee Ave             13221
## 47                 Clinton St & Tilden St             13037
## 48      DuSable Lake Shore Dr & North Blvd            LF-005
## 49                Clark St & Winnemac Ave     TA1309000035
## 50               Desplaines St & Kinzie St     TA1306000003
##                      end_station_name end_station_id start_lat start_lng
## 1  Mies van der Rohe Way & Chestnut St          15529  41.89259 -87.61729
```

```
## 2        Lakeview Ave & Fullerton Pkwy   TA1309000019  41.93000 -87.65000
## 3                                                     41.76926 -87.62822
## 4               Rush St & Superior St          15530  41.91468 -87.64332
## 5                 Ellis Ave & 55th St   KA1504000076  41.78510 -87.60107
## 6                 Ellis Ave & 55th St   KA1504000076  41.79523 -87.58089
## 7          Lincoln Ave & Diversey Pkwy   TA1307000064  41.93242 -87.65270
## 8              State St & Van Buren St   TA1305000035  41.97801 -87.66801
## 9           Greenview Ave & Jarvis Ave            520  42.01602 -87.66868
## 10              Sangamon St & Lake St   TA1306000015  41.90704 -87.66723
## 11           Clark St & Wrightwood Ave   TA1305000014  41.92557 -87.65855
## 12            Peoria St & Jackson Blvd          13158  41.87773 -87.65479
## 13          LaSalle St & Washington St          13006  41.88316 -87.65110
## 14                                                    41.92965 -87.70801
## 15                                                    41.91646 -87.66667
## 16              State St & Pearson St   TA1307000061  41.89842 -87.62241
## 17        Clinton St & Washington Blvd         WL-012  41.89456 -87.65345
## 18             Dearborn St & Monroe St   TA1305000006  41.92217 -87.63889
## 19               Emerald Ave & 28th St   TA1307000153  41.91213 -87.63484
## 20                   Clark St & Elm St   TA1307000039  41.88991 -87.63427
## 21         Desplaines St & Randolph St          15535  41.88045 -87.63516
## 22           Blue Island Ave & 18th St          13135  41.86488 -87.64707
## 23         Sheffield Ave & Webster Ave   TA1309000033  41.92154 -87.65382
## 24                                                    41.91000 -87.65000
## 25                                                    41.92000 -87.67000
## 26          Sheridan Rd & Montrose Ave   TA1307000107  41.97367 -87.65985
## 27           Halsted St & Clybourn Ave            331  41.89017 -87.62619
## 28             Orleans St & Hubbard St            636  41.88462 -87.62792
## 29                                                    41.91389 -87.65286
## 30             Sedgwick St & North Ave   TA1307000038  41.88990 -87.63424
## 31         Western Ave & Congress Pkwy          15668  41.86865 -87.68627
## 32 Mies van der Rohe Way & Chicago Ave          13338  41.89259 -87.61729
## 33              Normal Ave & Archer Ave   TA1308000014  41.87947 -87.62569
## 34                Wood St & Chicago Ave            637  41.88592 -87.66717
## 35         Oakley Ave & Irving Park Rd   KA1504000158  41.96141 -87.67617
## 36      Southport Ave & Irving Park Rd   TA1309000043  41.90967 -87.64813
## 37                 Canal St & Taylor St          15550  41.87588 -87.64079
## 38              Lake Park Ave & 35th St   KA1503000004  41.83127 -87.60880
## 39                  Ellis Ave & 58th St   TA1309000011  41.80000 -87.58000
## 40              Ogden Ave & Chicago Ave   TA1305000020  41.89102 -87.63548
## 41                Franklin St & Lake St   TA1307000111  41.89488 -87.63233
## 42              Streeter Dr & Grand Ave          13022  41.88116 -87.62410
## 43            Ashland Ave & Division St          13061  41.94000 -87.69000
## 44            Ashland Ave & Division St          13061  41.89842 -87.68660
## 45                                                    41.90916 -87.76096
## 46                                                    41.90760 -87.67247
## 47              Clinton St & Madison St   TA1305000032  41.87588 -87.64079
## 48                       Dusable Harbor   KA1503000064  41.91171 -87.62678
## 49                                                    41.97334 -87.66780
## 50                    Daley Center Plaza   TA1306000010  41.88872 -87.64445
##     end_lat   end_lng user_type ride_length  date month day year day_of_week
## 1  41.89859 -87.62192    casual    225 secs   225   Apr  27 2023    Thursday
## 2  41.92586 -87.63897    casual    508 secs   508   Jun  04 2023      Sunday
```

```
## 3  41.77000 -87.64000    casual    407 secs   407   Mar  06 2023      Monday
## 4  41.89576 -87.62591    member    791 secs   791   May  24 2023   Wednesday
## 5  41.79430 -87.60145    casual    207 secs   207   Apr  05 2023   Wednesday
## 6  41.79430 -87.60145    member    774 secs   774   Apr  19 2023   Wednesday
## 7  41.93223 -87.65862    member    150 secs   150   Feb  06 2023      Monday
## 8  41.87718 -87.62784    member   2721 secs  2721   Feb  19 2023      Sunday
## 9  42.01596 -87.66857    casual    222 secs   222   May  08 2023      Monday
## 10 41.88578 -87.65102    member    730 secs   730   Jun  22 2023    Thursday
## 11 41.92955 -87.64312    member    321 secs   321   Mar  24 2023      Friday
## 12 41.87764 -87.64962    member    104 secs   104   May  24 2023   Wednesday
## 13 41.88266 -87.63253    casual  15062 secs 15062   Jun  27 2023     Tuesday
## 14 41.93000 -87.73000    casual    326 secs   326   Mar  08 2023   Wednesday
## 15 41.95000 -87.68000    member    959 secs   959   Apr  13 2023    Thursday
## 16 41.89745 -87.62872    member    134 secs   134   Mar  21 2023     Tuesday
## 17 41.88338 -87.64117    member    438 secs   438   Feb  13 2023      Monday
## 18 41.88132 -87.62952    casual   1275 secs  1275   Jun  21 2023   Wednesday
## 19 41.84358 -87.64537    casual   1498 secs  1498   May  19 2023      Friday
## 20 41.90297 -87.63128    member    730 secs   730   Feb  20 2023      Monday
## 21 41.88462 -87.64457    member    345 secs   345   May  22 2023      Monday
## 22 41.85756 -87.66154    member    580 secs   580   May  14 2023      Sunday
## 23 41.92154 -87.65382    casual   1305 secs  1305   May  20 2023    Saturday
## 24 41.92000 -87.65000    member    306 secs   306   Jun  27 2023     Tuesday
## 25 41.95000 -87.68000    member    915 secs   915   Mar  24 2023      Friday
## 26 41.96167 -87.65464    member    371 secs   371   May  22 2023      Monday
## 27 41.90967 -87.64813    member   1473 secs  1473   Jun  06 2023     Tuesday
## 28 41.89003 -87.63662    casual    432 secs   432   May  20 2023    Saturday
## 29 41.91000 -87.65000    casual    147 secs   147   Apr  17 2023      Monday
## 30 41.91139 -87.63868    casual    688 secs   688   Mar  21 2023     Tuesday
## 31 41.87475 -87.68645    member    231 secs   231   Mar  08 2023   Wednesday
## 32 41.89694 -87.62176    casual    245 secs   245   Jun  25 2023      Sunday
## 33 41.84953 -87.64059    member   1777 secs  1777   Jun  18 2023      Sunday
## 34 41.89563 -87.67207    member    380 secs   380   Apr  19 2023   Wednesday
## 35 41.95434 -87.68608    member    461 secs   461   Feb  07 2023     Tuesday
## 36 41.95418 -87.66436    member   1610 secs  1610   Jun  21 2023   Wednesday
## 37 41.87026 -87.63947    member    193 secs   193   Jan  12 2023    Thursday
## 38 41.83127 -87.60880    casual   5758 secs  5758   May  03 2023   Wednesday
## 39 41.78875 -87.60133    casual   1203 secs  1203   May  10 2023   Wednesday
## 40 41.89636 -87.65406    casual    710 secs   710   May  16 2023     Tuesday
## 41 41.88584 -87.63550    casual    596 secs   596   Jan  31 2023     Tuesday
## 42 41.89228 -87.61204    casual   1017 secs  1017   May  22 2023      Monday
## 43 41.90345 -87.66775    member    989 secs   989   Jun  02 2023      Friday
## 44 41.90345 -87.66775    casual    425 secs   425   Mar  04 2023    Saturday
## 45 41.91000 -87.76000    casual    421 secs   421   Jun  23 2023      Friday
## 46 41.92000 -87.69000    casual    355 secs   355   Apr  28 2023      Friday
## 47 41.88275 -87.64119    member    203 secs   203   Apr  07 2023      Friday
## 48 41.88698 -87.61281    member   1331 secs  1331   Feb  12 2023      Sunday
## 49 41.97000 -87.66000    casual    165 secs   165   Jun  10 2023    Saturday
## 50 41.88424 -87.62963    casual    711 secs   711   Jun  03 2023    Saturday
```

Save data after cleaning

```
write.csv(all_clean_data, file = "Data_after_cleaning.csv", row.names = FALSE)
```

Step 4 : Analyzing and Visualising the Data

- Descriptive Analysis

```
summary(all_clean_data$ride_length)
```

```
##   Length    Class     Mode
##  2315029 difftime  numeric
```

- Average ride_length of all rides is 905.5 secs (~15 mins)
- Minimum ride_length of all rides is 60 secs (1 min)
- Maximum ride_length of all rides is 86146 secs (~24 hrs)

Data for Average ride_length by user_type

```
all_clean_data%>%
  group_by(user_type) %>%
  summarise(average_ride_length = mean(ride_length))
```

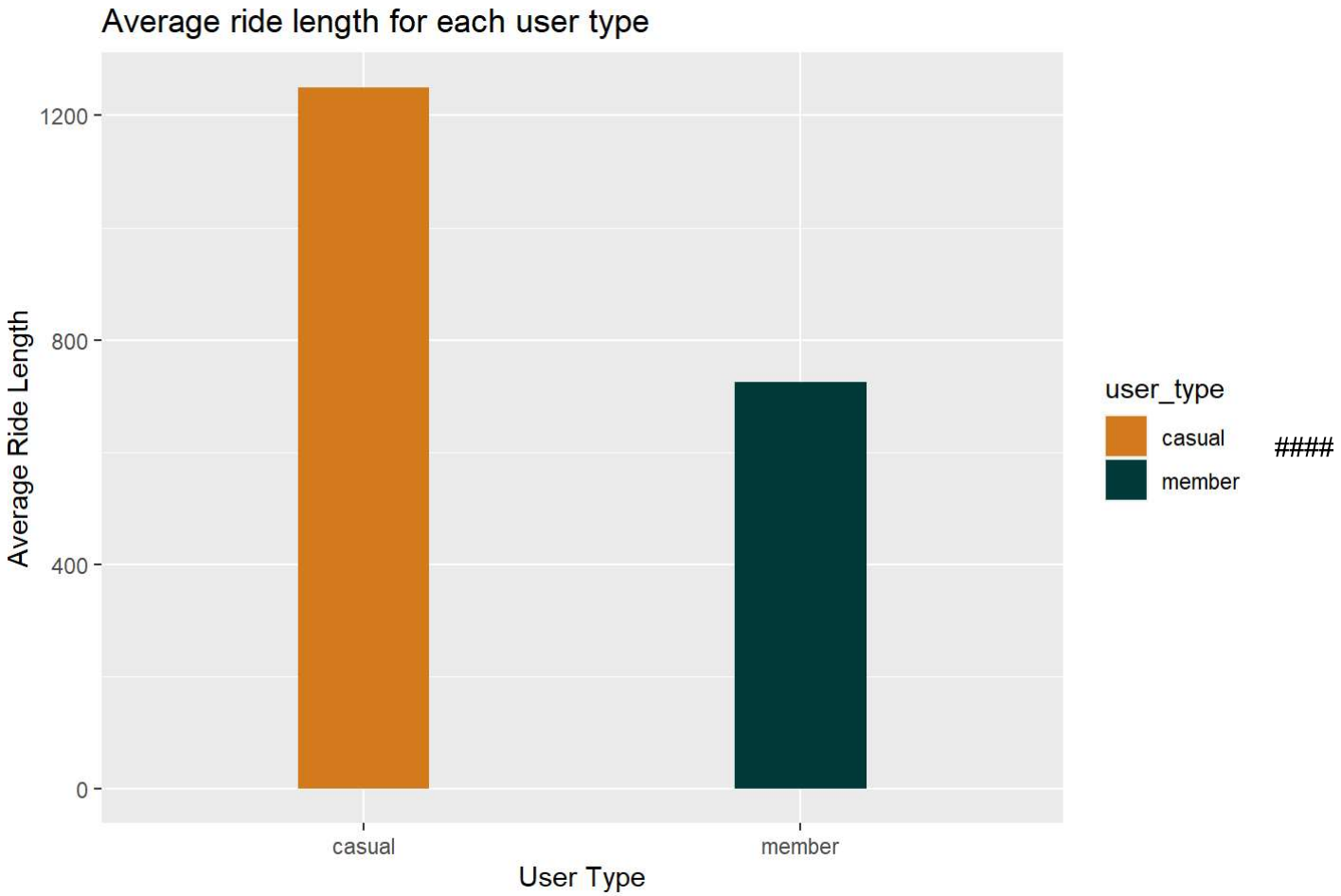
```
## # A tibble: 2 × 2
##   user_type average_ride_length
##   <chr>     <drtn>
## 1 casual    1250.6332 secs
## 2 member     724.2051 secs
```

- Average Ride Length
  - **Member** 12.04 minutes
  - **Casual** 20.84 minutes

Visualizing Average ride_length

```
all_clean_data %>%
  group_by(user_type) %>%
  summarise(average_ride_length = mean(ride_length)) %>%
  ggplot() + geom_bar(mapping = aes(x = user_type, y = average_ride_length, fill = user_type), s
tat = 'identity', width = 0.3) + labs(title = "Average ride length for each user type", x = "Use
r Type", y = "Average Ride Length") + scale_fill_manual(values = c("#d37a1c", "#023a3a"))
```

```
## Don't know how to automatically pick scale for object of type <difftime>.
## Defaulting to continuous.
```



Key Insights * Average ride_length or ride time of casual rides is 1250.63 secs(~21 mins.) while that of member rides is 724.2051 secs (12 mins.)

Total Number of Rides by User Type and Type of the bike

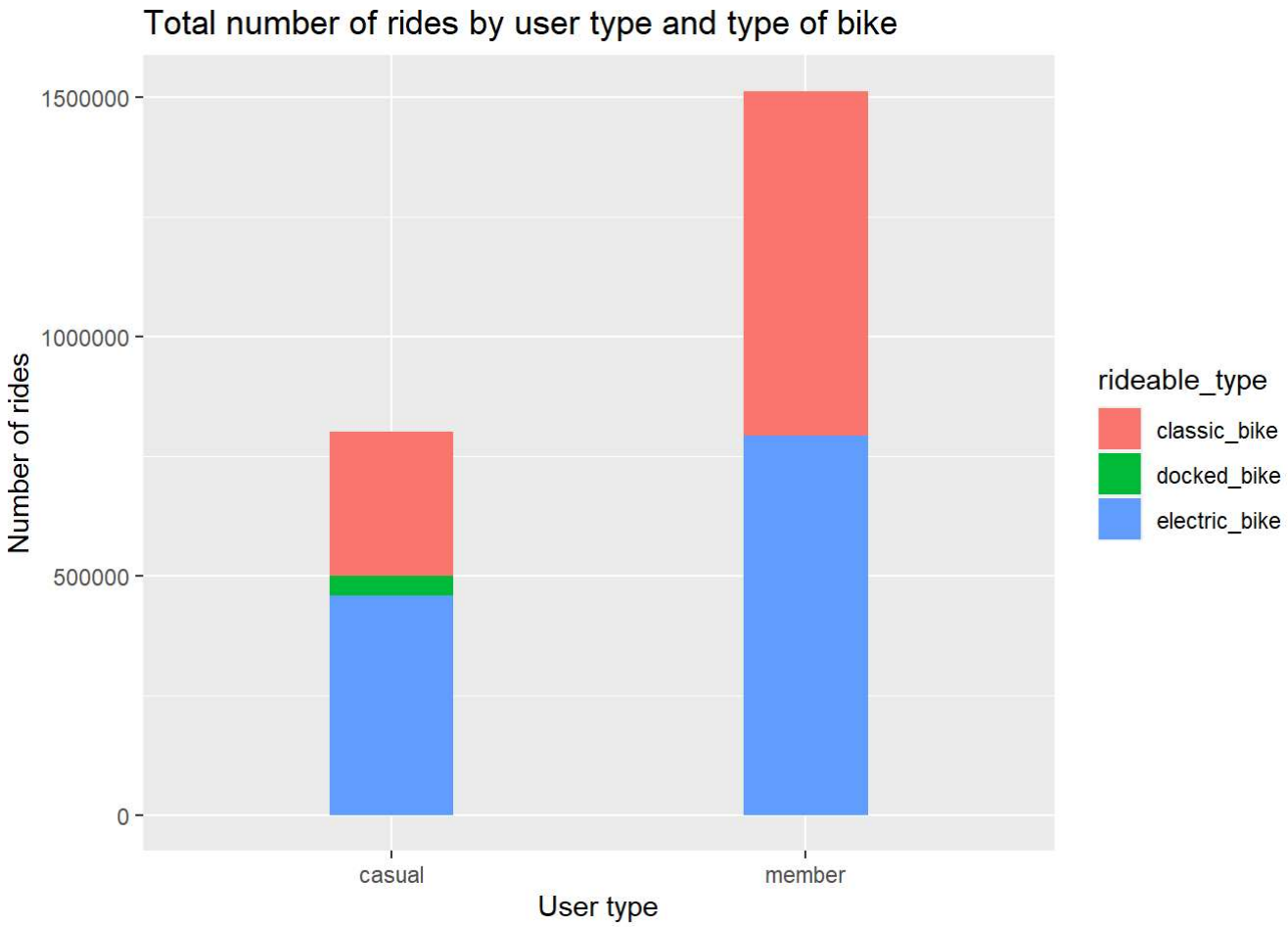
```
all_clean_data %>%
  group_by(user_type, rideable_type) %>%
  summarise(number_of_rides = n())
```

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

```
## # A tibble: 5 × 3
## # Groups:   user_type [2]
##   user_type rideable_type number_of_rides
##   <chr>     <chr>                  <int>
## 1 casual    classic_bike          301333
## 2 casual    docked_bike            42377
## 3 casual    electric_bike         457993
## 4 member    classic_bike          721162
## 5 member    electric_bike         792164
```

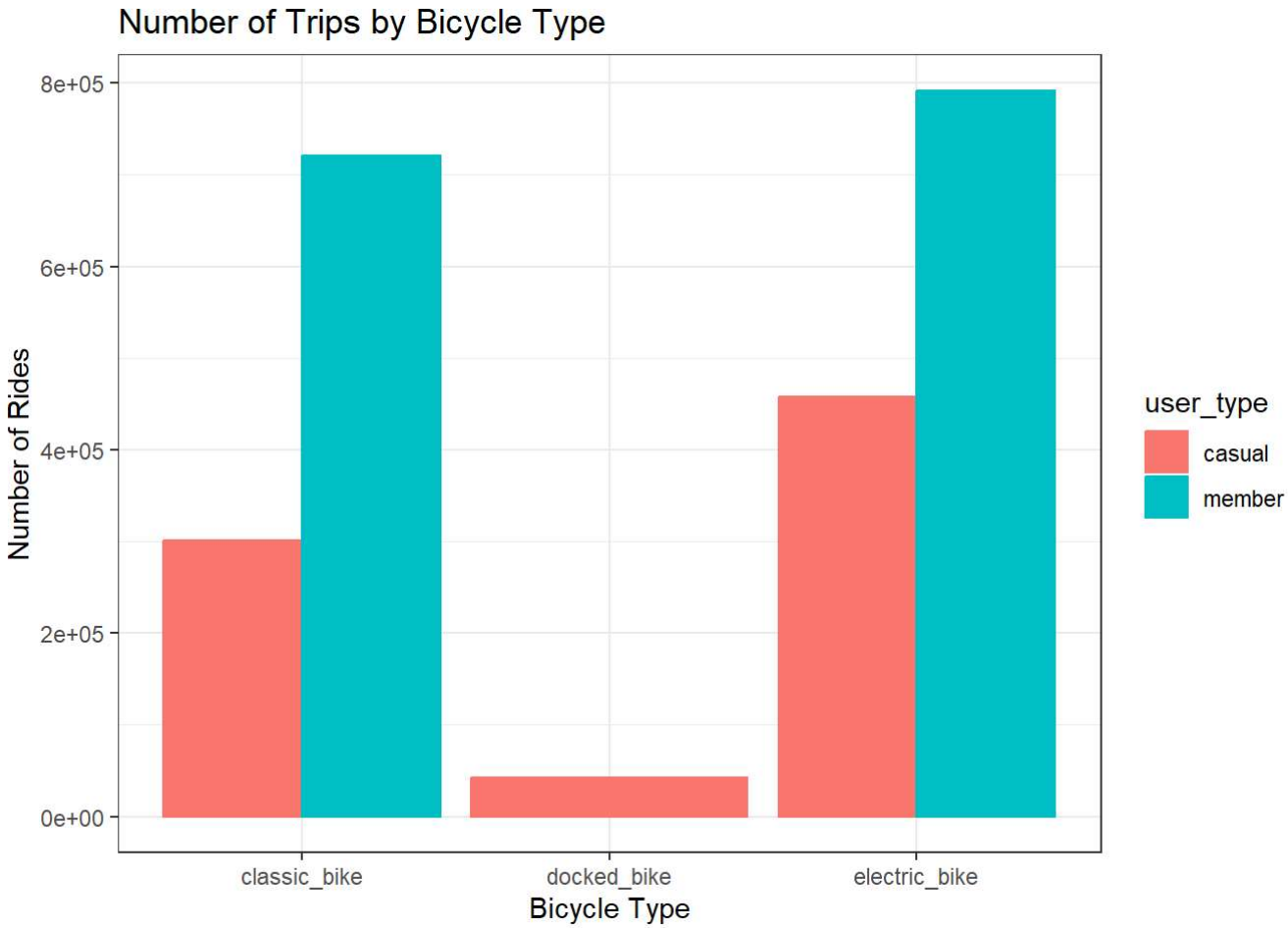
```
all_clean_data %>%
  group_by(user_type, rideable_type) %>%
  summarise(number_of_rides = n()) %>%
  ggplot() + geom_bar(mapping = aes(x=user_type, y=number_of_rides, fill=rideable_type), stat='i
dentity', width = 0.3) + labs(title = "Total number of rides by user type and type of bike", x
="User type", y="Number of rides")
```

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




```
all_clean_data %>%
  group_by(rideable_type,user_type) %>%
  dplyr::summarize(count_trips = n()) %>%
  ggplot(aes(x=rideable_type, y=count_trips, fill=user_type, color=user_type)) +
  geom_bar(stat='identity', position='dodge') +
  theme_bw()+
  labs(title="Number of Trips by Bicycle Type", x="Bicycle Type", y="Number of Rides")
```

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



Key Insights * Docked bike is not used by anyone who is a member and is not that popular as an option among casual riders as well. * Classic and electric bike are a popular choice among casual as well as member riders.

Total number of riders for each user type by months

```
all_clean_data %>%
  group_by(user_type, month) %>%
  summarise(number_of_rides = n()) %>%
  arrange(user_type)
```

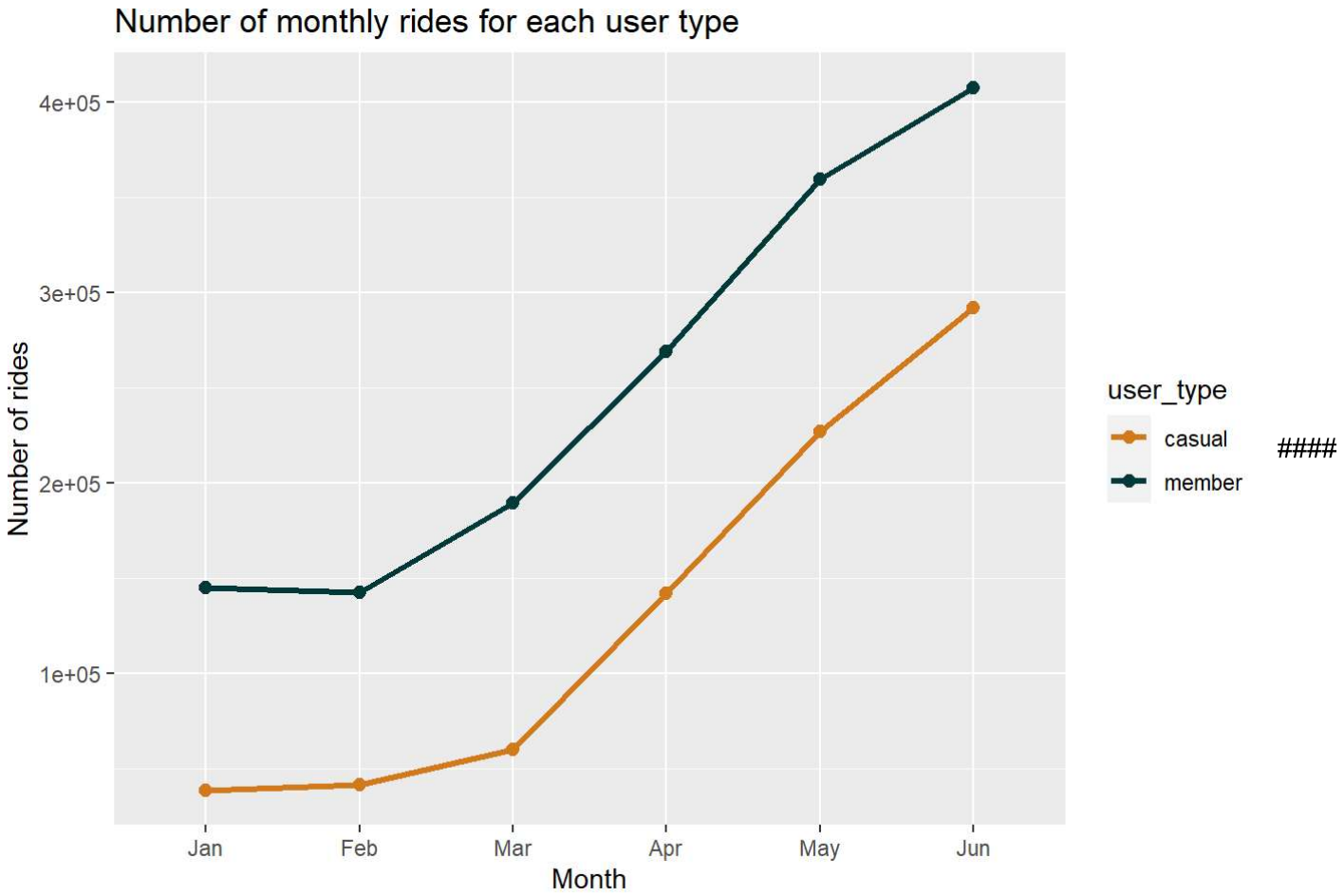
```
## `summarise()` has grouped output by 'user_type'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 12 × 3
## # Groups:   user_type [2]
##    user_type month number_of_rides
##    <chr>     <chr>           <int>
##  1 casual    Apr            142121
##  2 casual    Feb             41723
##  3 casual    Jan             38731
##  4 casual    Jun            292232
##  5 casual    Mar             60125
##  6 casual    May            226771
##  7 member    Apr            269042
##  8 member    Feb            142601
##  9 member    Jan            145246
## 10 member    Jun            407688
## 11 member    Mar            189289
## 12 member    May            359460
```

```
all_clean_data <- all_clean_data %>%
  mutate(month =factor(month, levels = c( "Jan", "Feb", "Mar","Apr", "May", "Jun")))
all_clean_data %>%
  group_by(user_type, month) %>%
  summarise(number_of_rides = n()) %>%
  ggplot(aes(x=month, y=number_of_rides, group=user_type, color=user_type)) + geom_line(size=1.
1) + geom_point(size=2.2)+ labs(title = "Number of monthly rides for each user type", x="Month",
y="Number of rides") + scale_color_manual(values = c("#d37a1c", "#023a3a"))
```

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

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



Total number of riders for each user type by Days

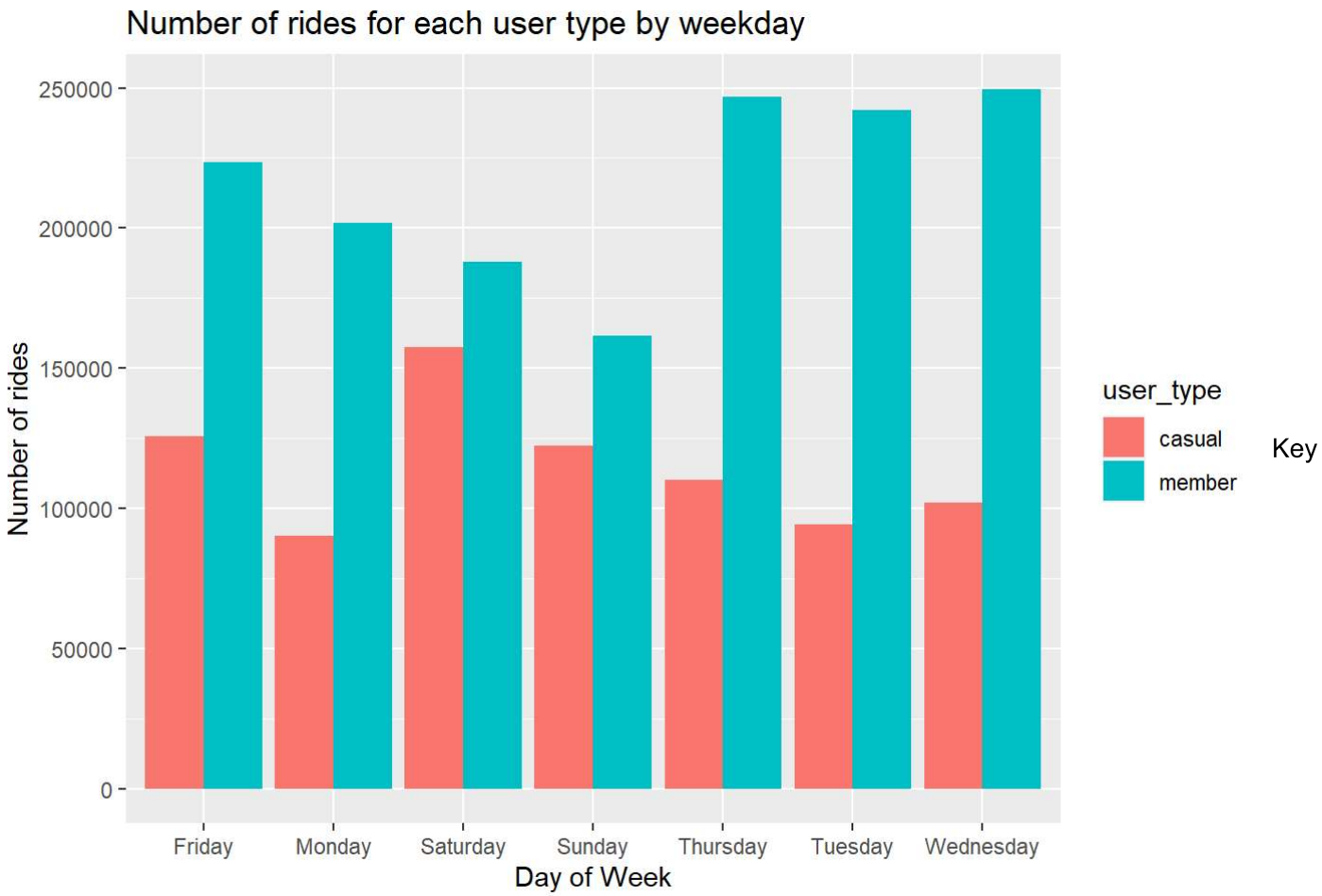
```
all_clean_data %>%
  group_by(user_type, day_of_week) %>%
  summarise(number_of_rides = n()) %>%
  arrange(user_type)
```

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

```
## # A tibble: 14 × 3
## # Groups:   user_type [2]
##    user_type day_of_week number_of_rides
##    <chr>     <chr>                 <int>
##  1 casual    Friday               125553
##  2 casual    Monday                90337
##  3 casual    Saturday             157465
##  4 casual    Sunday               122190
##  5 casual    Thursday             110043
##  6 casual    Tuesday               94076
##  7 casual    Wednesday            102039
##  8 member    Friday               223626
##  9 member    Monday               201811
## 10 member    Saturday             188011
## 11 member    Sunday               161558
## 12 member    Thursday             246649
## 13 member    Tuesday              242030
## 14 member    Wednesday            249641
```

```
all_clean_data %>%
  group_by(user_type, day_of_week) %>%
  summarise(number_of_rides = n()) %>%
  ggplot() + geom_col(mapping = aes(x=day_of_week, y=number_of_rides, fill=user_type), position
= "dodge") +
  labs(title = "Number of rides for each user type by weekday", x="Day of Week", y="Number of ri
des")
```

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



Insights * The number of Member riders are maximum on on Thursday, Tuesday and Wednesday. * The number of Casual riders are almost constant throughout the week.

Find popular start station for member riders

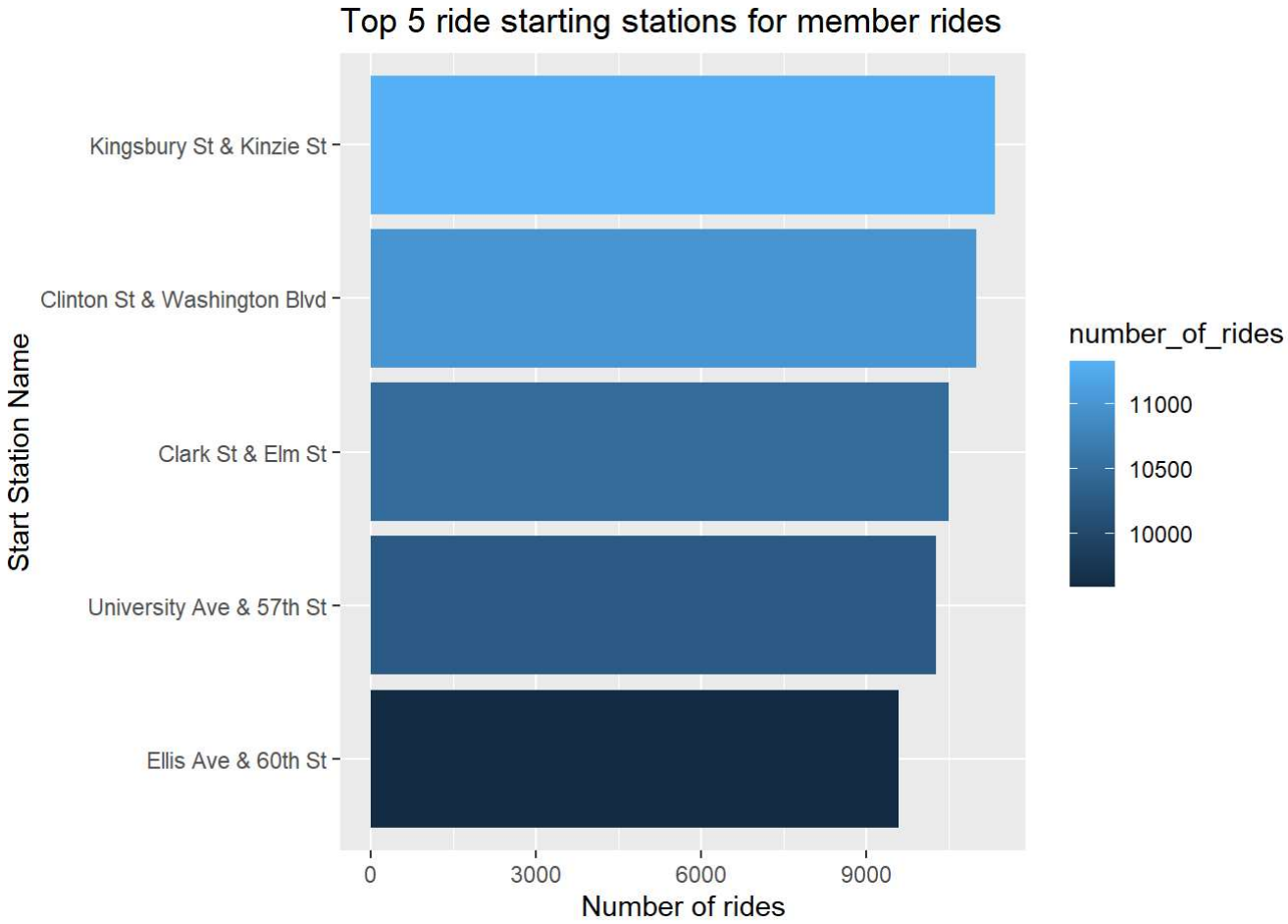
```
all_clean_data %>%
  group_by(user_type,start_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(start_station_name != "", "member" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)
```

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

```
## # A tibble: 5 × 3
## # Groups:   user_type [1]
##   user_type start_station_name           number_of_rides
##   <chr>     <chr>                                  <int>
## 1 member    Kingsbury St & Kinzie St               11335
## 2 member    Clinton St & Washington Blvd           11004
## 3 member    Clark St & Elm St                      10492
## 4 member    University Ave & 57th St               10268
## 5 member    Ellis Ave & 60th St                     9595
```

```
#Find popular start station for member riders
all_clean_data %>%
  group_by(user_type,start_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(start_station_name != "", "member" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)  %>%
  mutate(start_station_name= fct_reorder(start_station_name, number_of_rides)) %>%
  ggplot() + geom_bar(mapping = aes(x=start_station_name, y=number_of_rides, fill=number_of_ride
s), stat = 'identity') + coord_flip() + labs(title = "Top 5 ride starting stations for member ri
des", x="Start Station Name", y="Number of rides")
```

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



Find popular start station for casual riders

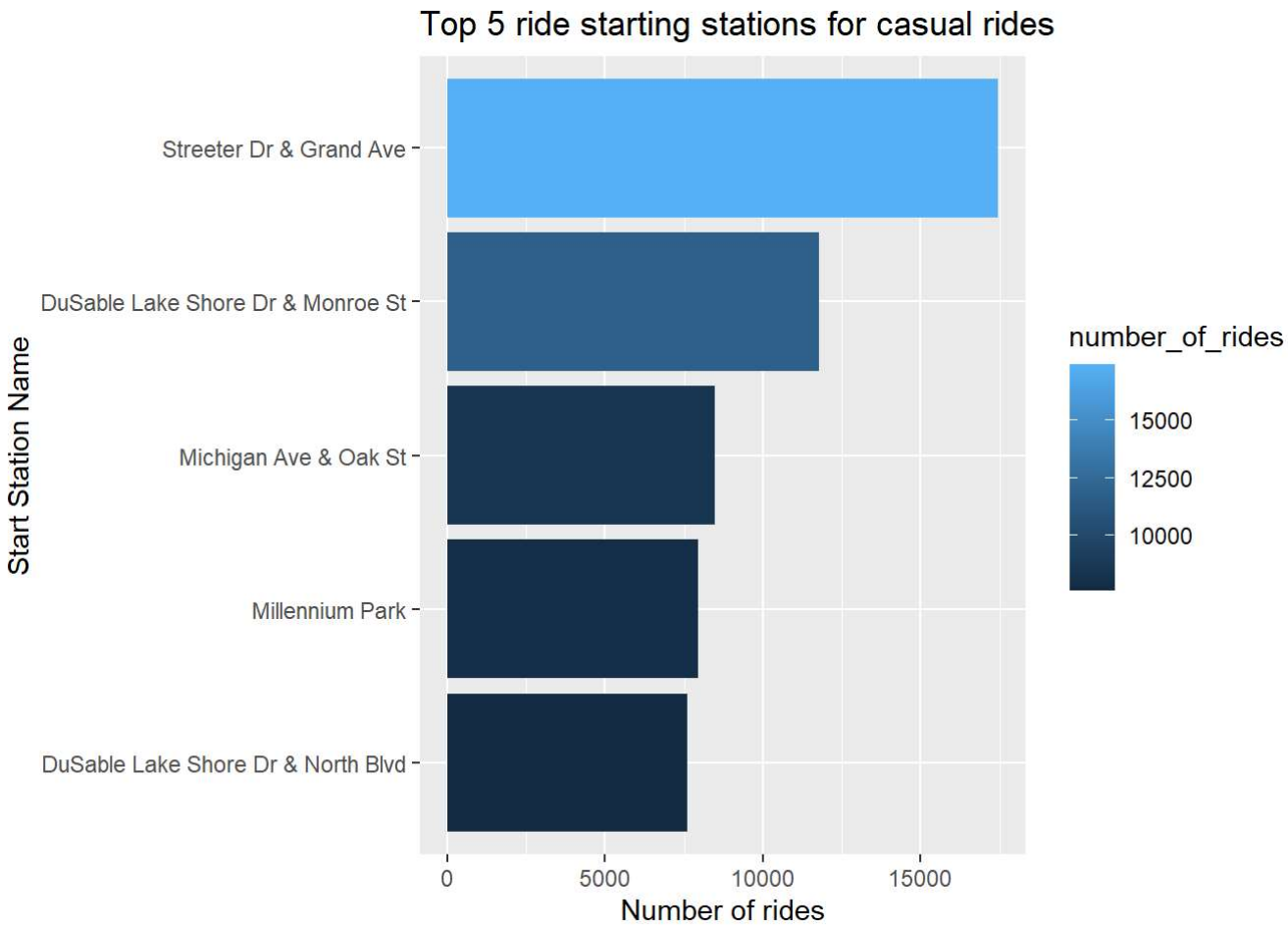
```
all_clean_data %>%
  group_by(user_type,start_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(start_station_name != "", "casual" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)
```

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

```
## # A tibble: 5 × 3
## # Groups:   user_type [1]
##   user_type start_station_name                number_of_rides
##   <chr>     <chr>                                       <int>
## 1 casual    Streeter Dr & Grand Ave                     17458
## 2 casual    DuSable Lake Shore Dr & Monroe St           11772
## 3 casual    Michigan Ave & Oak St                        8479
## 4 casual    Millennium Park                              7963
## 5 casual    DuSable Lake Shore Dr & North Blvd           7618
```

```
all_clean_data %>%
  group_by(user_type,start_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(start_station_name != "", "casual" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)  %>%
  mutate(start_station_name= fct_reorder(start_station_name, number_of_rides)) %>%
  ggplot() + geom_bar(mapping = aes(x=start_station_name, y=number_of_rides, fill=number_of_ride
s), stat = 'identity') + coord_flip() + labs(title = "Top 5 ride starting stations for casual ri
des", x="Start Station Name", y="Number of rides")
```

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



Find popular end station for member riders

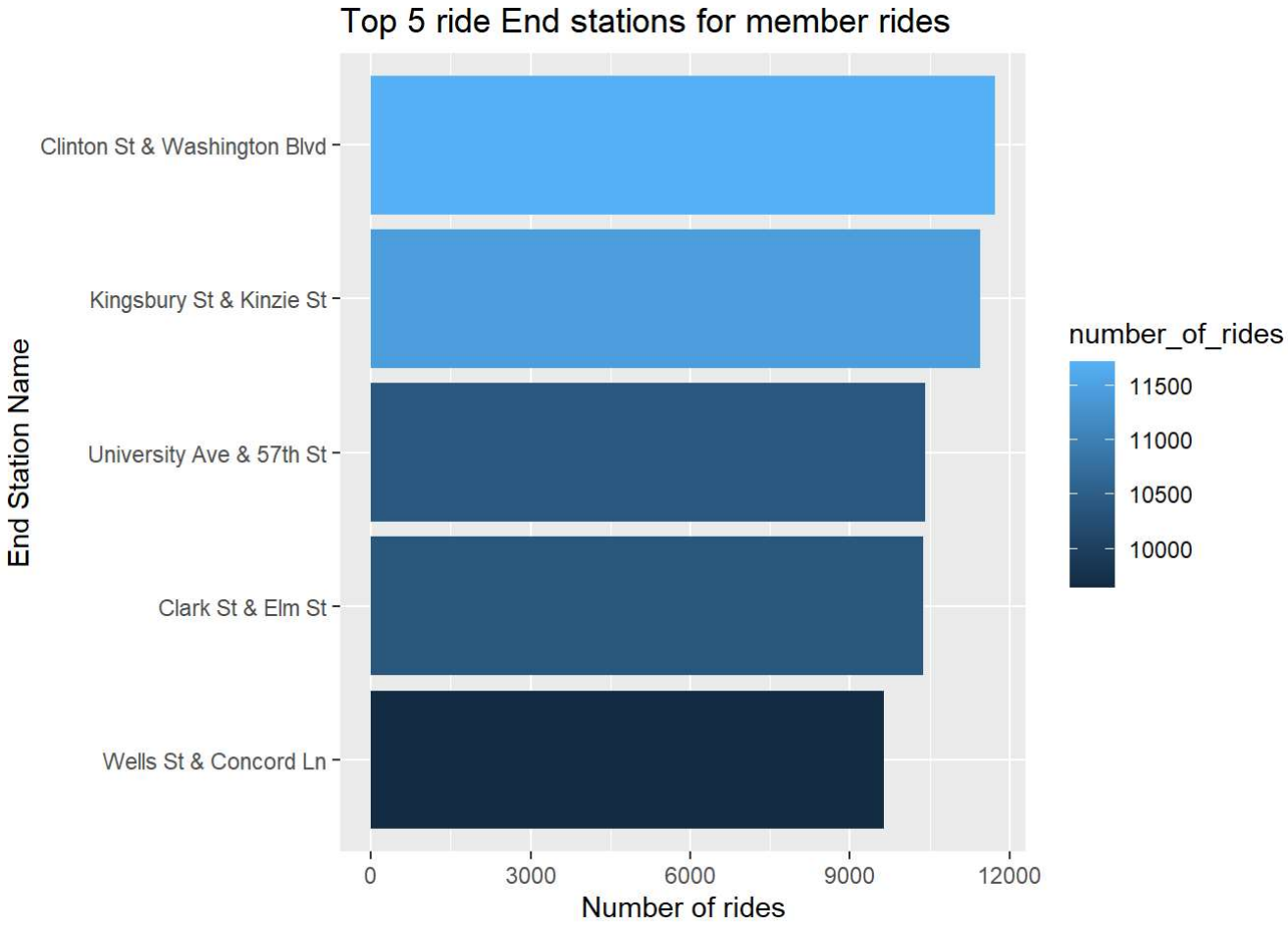
```
all_clean_data %>%
  group_by(user_type,end_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(end_station_name != "", "member" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)
```

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

```
## # A tibble: 5 × 3
## # Groups:   user_type [1]
##   user_type end_station_name             number_of_rides
##   <chr>     <chr>                                  <int>
## 1 member    Clinton St & Washington Blvd           11722
## 2 member    Kingsbury St & Kinzie St               11449
## 3 member    University Ave & 57th St               10409
## 4 member    Clark St & Elm St                      10380
## 5 member    Wells St & Concord Ln                   9647
```

```
all_clean_data %>%
  group_by(user_type,end_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(end_station_name != "", "member" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)  %>%
  mutate(end_station_name= fct_reorder(end_station_name, number_of_rides)) %>%
  ggplot() + geom_bar(mapping = aes(x=end_station_name, y=number_of_rides, fill=number_of_ride
s), stat = 'identity') + coord_flip() + labs(title = "Top 5 ride End stations for member rides",
x="End Station Name", y="Number of rides")
```

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



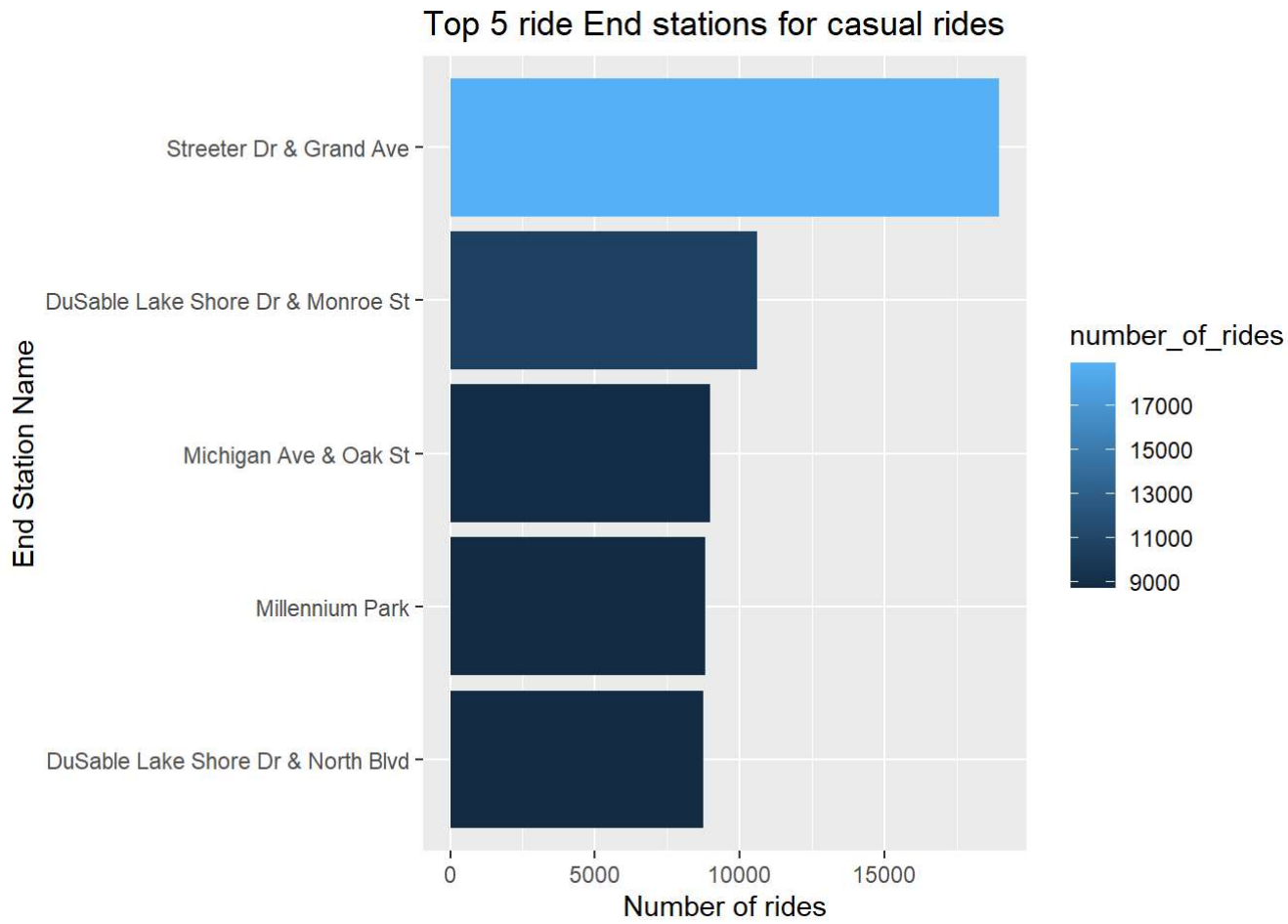

```
all_clean_data %>%
  group_by(user_type,end_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(end_station_name != "", "casual" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)
```

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

```
## # A tibble: 5 × 3
## # Groups:   user_type [1]
##   user_type end_station_name                  number_of_rides
##   <chr>     <chr>                                       <int>
## 1 casual    Streeter Dr & Grand Ave                     18974
## 2 casual    DuSable Lake Shore Dr & Monroe St           10613
## 3 casual    Michigan Ave & Oak St                        8996
## 4 casual    Millennium Park                              8797
## 5 casual    DuSable Lake Shore Dr & North Blvd           8744
```

```
all_clean_data %>%
  group_by(user_type,end_station_name) %>%
  dplyr::summarise(number_of_rides = n()) %>%
  filter(end_station_name != "", "casual" == user_type) %>%
  arrange(-number_of_rides) %>%
  head(n=5)  %>%
  mutate(end_station_name= fct_reorder(end_station_name, number_of_rides)) %>%
  ggplot() + geom_bar(mapping = aes(x=end_station_name, y=number_of_rides, fill=number_of_ride
s), stat = 'identity') + coord_flip() + labs(title = "Top 5 ride End stations for casual rides",
x="End Station Name", y="Number of rides")
```

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



EXPORT SUMMARY FILE FOR FURTHER ANALYSIS

```
write.csv(all_clean_data, file = "yearly_trips_report.csv")
```

Step 6 : Act

Conclusions from the analysis

- 1-The average ride length for casual rides is 20 minutes which is more than member rides - 12 minutes.
- 2-Analysis based on Months :
  - For Casual riders the highest number of rides are in the months of May and Jun. The months with the lowest number of rides are January and February.
  - For Member riders the highest number of rides are in the months of May and Jun. The months with the lowest number of rides are January and February.
- 3-Analysis based on Days :
  - Member rides are highest on Thursday, Tuesday and Wednesday.
  - Casual rides are lowest on Monday and Tuesday.
- 4-Analysis based on Location :
  - The top 5 ride starting stations for Member users are- Kingsbury St & Kinzie St,Clinton St & Washington Blvd, Clark St & Elm St, University Ave & 57th St, Ellis Ave & 60th St.
  - The top 5 ride starting stations for Casual users are- Streeter Dr & Grand Ave,DuSable Lake Shore Dr & Monroe St,Michigan Ave & Oak St,Millennium Park, DuSable Lake Shore Dr & North Blvd.

    - The top 5 ride Ending stations for Member users are-Clinton St & Washington Blvd,Kingsbury St & Kinzie St, University Ave & 57th St, Clark St & Elm St, Wells St & Concord Ln.
    - The top 5 ride Ending stations for Casual users are- Streeter Dr & Grand Ave, DuSable Lake Shore Dr & Monroe St, Michigan Ave & Oak St, Millennium Park, DuSable Lake Shore Dr & North Blvd

Conclusions

- Casual riders take less trips/rides but for longer durations of time vs members

Recommendations¶

Members and casual riders differ in how long they use the bikes and how often they use the bikes. In order to get more casual riders to buy a membership my top recommendations are:

- Give discounts for longer rides when you have a membership
- Longer rides can get some type of rewards program when they become members