Google Data Analysis Capstone Project

Alemayehu Desta

2023-02-24

Introduction

This project is part of the google data analytics professional certificate final capstone project. It is about a fictional bike sharing company named Cyclistic based in Chicago. In order to answer the key business questions, the following steps of data analysis process are incorporated:

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

Scenario

The director of marketing of the company believes the company’s future success depends on maximizing the number of annual memberships. Therefore, the company team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights,the company team will design a new marketing strategy to convert casual riders into annual members. visualizations to approve

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.

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.

I have installed and loaded the necessary r base functions that I beleive are relevant to do data cleaning, analysis and visualization.

```
library(tidyverse)
```

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

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

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

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
library(readr)
library(ggplot2)
library(dplyr) #calculations
library("dplyr")
library(rmarkdown)
```

Ask

Three questions will guide the future marketing program:

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

Prepare

The project used Cyclistic's historical trip data to analyze and identify trends. Annua trip data of 2022 is downloaded from here (https://divvy-tripdata.s3.amazonaws.com/index.html).The data has been made available by Motivate International Inc. under this license.) This is public data that you can use to explore how different customer types are using Cyclistic bikes. But note that data-privacy issues prohibit you from using riders' personally identifiable information. This means that you won't be able to connect pass purchases to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes. Now, prepare your data for analysis using the following Case Study Roadmap as a guide

Guiding questions * Where is your data located? * How is the data organized? * Are there issues with bias or credibility in this data? * How are you addressing licensing, privacy, security, and accessibility? * How did you verify the data's integrity? * How does it help you answer your question? * Are there any problems with the data

Process

Then, process your data for analysis using the following Case Study Roadmap as a guide

** gudiing Questions ** * What tools are you choosing and why? * Have you ensured your data's integrity? * What steps have you taken to ensure that your data is clean? * How can you verify that your data is clean and ready to analyze? * Have you documented your cleaning process so you can review and share those results

Import and collect data

Data wrangling, cleaning and validation

```
colnames(Jan_2022)
```

```
##  [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"
```

```
colnames(Feb_2022)
```

```
##  [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"
```

```
colnames(Mar_2022)
```

```
##  [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"
```

```
colnames(Apr_2022)
```

```
##  [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"
```

```
colnames(May_2022)
```

```
##  [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"
```

```
colnames(Jun_2022)
```

```
##  [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"
```

```
colnames(Jul_2022)
```

```
##  [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"
```

```
colnames(Aug_2022)
```

```
##  [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"
```

```
colnames(Sep_2022)
```

```
##  [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"
```

```
colnames(Oct_2022)
```

```
##  [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"
```

```
colnames(Nov_2022)
```

```
##  [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"
```

```
colnames(Dec_2022)
```

```
##  [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"
```

#Total number of rows

```
sum(nrow(Jan_2022) + nrow(Feb_2022) 
    + nrow(Mar_2022) + nrow(Apr_2022) + nrow(May_2022) 
    + nrow(Jun_2022) + nrow(Jul_2022) + nrow(Aug_2022)
    + nrow(Sep_2022) + nrow(Oct_2022) + nrow(Nov_2022)+nrow(Dec_2022))
```

```
## [1] 5667717
```

#Combine every months data to a year for a complete picture

```
annual_trip<-rbind(Jan_2022,Feb_2022,Mar_2022,
                          Apr_2022,May_2022,Jun_2022,
                          Jul_2022,Aug_2022,Sep_2022,
                          Oct_2022,Nov_2022, Dec_2022)
```

#Final data validation

```
str(annual_trip)
```

```
## spc_tbl_ [5,667,717 × 13] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ride_id           : chr [1:5667717] "C2F7DD78E82EC875" "A6CF8980A652D272" "BD0F91DFF741C66
D" "CBB80ED419105406" ...
##  $ rideable_type     : chr [1:5667717] "electric_bike" "electric_bike" "classic_bike" "classi
c_bike" ...
##  $ started_at        : POSIXct[1:5667717], format: "2022-01-13 11:59:47" "2022-01-10 08:41:5
6" ...
##  $ ended_at          : POSIXct[1:5667717], format: "2022-01-13 12:02:44" "2022-01-10 08:46:1
7" ...
##  $ start_station_name: chr [1:5667717] "Glenwood Ave & Touhy Ave" "Glenwood Ave & Touhy Ave"
"Sheffield Ave & Fullerton Ave" "Clark St & Bryn Mawr Ave" ...
##  $ start_station_id  : chr [1:5667717] "525" "525" "TA1306000016" "KA1504000151" ...
##  $ end_station_name  : chr [1:5667717] "Clark St & Touhy Ave" "Clark St & Touhy Ave" "Greenvi
ew Ave & Fullerton Ave" "Paulina St & Montrose Ave" ...
##  $ end_station_id    : chr [1:5667717] "RP-007" "RP-007" "TA1307000001" "TA1309000021" ...
##  $ start_lat         : num [1:5667717] 42 42 41.9 42 41.9 ...
##  $ start_lng         : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ end_lat           : num [1:5667717] 42 42 41.9 42 41.9 ...
##  $ end_lng           : num [1:5667717] -87.7 -87.7 -87.7 -87.7 -87.6 ...
##  $ member_casual     : chr [1:5667717] "casual" "casual" "member" "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>
```

```
dim(annual_trip)
```

```
## [1] 5667717      13
```

```
summary(annual_trip)
```

```
##    ride_id          rideable_type        started_at                    
##  Length:5667717     Length:5667717     Min.   :2022-01-01 00:00:05.00  
##  Class :character   Class :character   1st Qu.:2022-05-28 19:21:05.00  
##  Mode  :character   Mode  :character   Median :2022-07-22 15:03:59.00  
##                                        Mean   :2022-07-20 07:21:18.74  
##                                        3rd Qu.:2022-09-16 07:21:29.00  
##                                        Max.   :2022-12-31 23:59:26.00  
##                                                                        
##     ended_at                      start_station_name start_station_id  
##  Min.   :2022-01-01 00:01:48.00   Length:5667717     Length:5667717    
##  1st Qu.:2022-05-28 19:43:07.00   Class :character   Class :character  
##  Median :2022-07-22 15:24:44.00   Mode  :character   Mode  :character  
##  Mean   :2022-07-20 07:40:45.33                                        
##  3rd Qu.:2022-09-16 07:39:03.00                                        
##  Max.   :2023-01-02 04:56:45.00                                        
##                                                                        
##  end_station_name   end_station_id       start_lat       start_lng     
##  Length:5667717     Length:5667717     Min.   :41.64   Min.   :-87.84  
##  Class :character   Class :character   1st Qu.:41.88   1st Qu.:-87.66  
##  Mode  :character   Mode  :character   Median :41.90   Median :-87.64  
##                                        Mean   :41.90   Mean   :-87.65  
##                                        3rd Qu.:41.93   3rd Qu.:-87.63  
##                                        Max.   :45.64   Max.   :-73.80  
##                                                                        
##     end_lat         end_lng       member_casual     
##  Min.   : 0.00   Min.   :-88.14   Length:5667717    
##  1st Qu.:41.88   1st Qu.:-87.66   Class :character  
##  Median :41.90   Median :-87.64   Mode  :character  
##  Mean   :41.90   Mean   :-87.65                     
##  3rd Qu.:41.93   3rd Qu.:-87.63                     
##  Max.   :42.37   Max.   :  0.00                     
##  NA's   :5858    NA's   :5858
```

```
names(annual_trip)
```

```
##  [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"
```

```
started_at=c("date","time")
date=factor(started_at)
as.numeric(date)
```

```
## [1] 1 2
```

#Rename data frame and remove unnecessary variables

```
clean_annual_trip <- annual_trip %>%
  select(-c(start_station_id,ride_id, end_station_id,start_lat,start_lng,end_lat,end_lng))
```

#Remove duplicates

```
clean_annual_trip <- distinct(clean_annual_trip)
```

#Check Cleaned data

```
colSums(is.na(clean_annual_trip))
```

```
##      rideable_type         started_at           ended_at start_station_name 
##                  0                  0                  0             833060 
##   end_station_name      member_casual 
##             892728                  0
```

#Rename column for better context

```
clean_annual_trip <- rename(clean_annual_trip, customer_type = member_casual, bike_type = rideable_type)
```

#Create column for date, day, month and year

```
clean_annual_trip$date <- as.Date(clean_annual_trip$started_at)
clean_annual_trip$week_day <- format(as.Date(clean_annual_trip$date), "%A")
clean_annual_trip$month <- format(as.Date(clean_annual_trip$date), "%m")
clean_annual_trip$year <- format(clean_annual_trip$date, "%Y")
```

#Create column for time

```
clean_annual_trip$time <- as.POSIXct(clean_annual_trip$started_at, format = "%Y-%m-%d %H:%M:%S")
clean_annual_trip$time <- format(clean_annual_trip$time, format = "%H:%M")
```

#Add ride_length column

```
clean_annual_trip$ride_length <- difftime(clean_annual_trip$ended_at, clean_annual_trip$started_at, units = "mins")
```

#Select relevant data only

```
clean_annual_trip <- clean_annual_trip %>%
  select(bike_type,customer_type,start_station_name, end_station_name,started_at, ended_at, month, year, time, week_day, ride_length)
```

#Remove data with greater start_at than end_at

```
clean_annual_trip<- clean_annual_trip %>%
  filter(started_at < ended_at)
```

Data Analysis and Visualization

#Descriptive analysis on ride_length (all figures in seconds)

```
mean(clean_annual_trip$ride_length) #average
```

```
## Time difference of 19.44613 mins
```

```
median(clean_annual_trip$ride_length) #Center point
```

```
## Time difference of 10.28333 mins
```

```
max(clean_annual_trip$ride_length) #Longest ride
```

```
## Time difference of 41387.25 mins
```

```
min(clean_annual_trip$ride_length) #shortest ride
```

```
## Time difference of 0.01666667 mins
```

#members and casual users comparision

```
aggregate(clean_annual_trip$ride_length ~clean_annual_trip$customer_type, FUN = mean)
```

```
##   clean_annual_trip$customer_type clean_annual_trip$ride_length
## 1                          casual                 29.14518 mins
## 2                          member                 12.71487 mins
```

```
aggregate(clean_annual_trip$ride_length ~ clean_annual_trip$customer_type, FUN = median)
```

```
##   clean_annual_trip$customer_type clean_annual_trip$ride_length
## 1                          casual                13.000000 mins
## 2                          member                 8.833333 mins
```

```
aggregate(clean_annual_trip$ride_length ~ clean_annual_trip$customer_type, FUN = max)
```

```
##   clean_annual_trip$customer_type clean_annual_trip$ride_length
## 1                          casual                 41387.25 mins
## 2                          member                  1559.90 mins
```

```
aggregate(clean_annual_trip$ride_length ~ clean_annual_trip$customer_type, FUN = min)
```

```
##   clean_annual_trip$customer_type clean_annual_trip$ride_length
## 1                          casual               0.01666667 mins
## 2                          member               0.01666667 mins
```

#Now, let’s run the average ride time by each day for members vs casual users

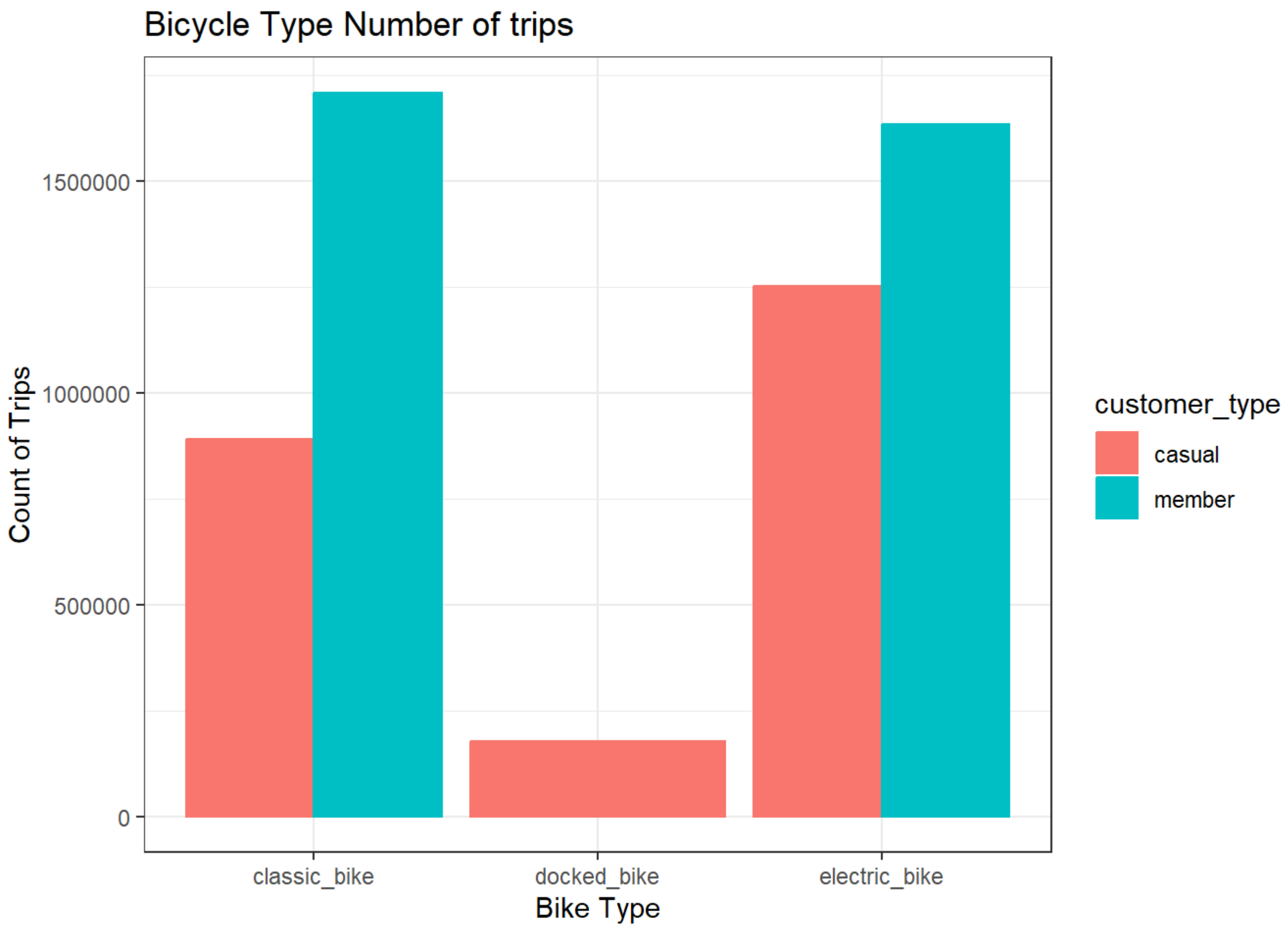
```
aggregate(clean_annual_trip$ride_length ~ clean_annual_trip$customer_type + clean_annual_trip$we
ek_day, FUN = mean)
```

```
##    clean_annual_trip$customer_type clean_annual_trip$week_day
## 1                           casual                     Friday
## 2                           member                     Friday
## 3                           casual                     Monday
## 4                           member                     Monday
## 5                           casual                   Saturday
## 6                           member                   Saturday
## 7                           casual                     Sunday
## 8                           member                     Sunday
## 9                           casual                   Thursday
## 10                          member                   Thursday
## 11                          casual                    Tuesday
## 12                          member                    Tuesday
## 13                          casual                  Wednesday
## 14                          member                  Wednesday
##    clean_annual_trip$ride_length
## 1                  28.04689 mins
## 2                  12.53164 mins
## 3                  29.18980 mins
## 4                  12.27087 mins
## 5                  32.60232 mins
## 6                  14.14097 mins
## 7                  34.06077 mins
## 8                  14.03233 mins
## 9                  25.55043 mins
## 10                 12.29366 mins
## 11                 25.82534 mins
## 12                 12.13029 mins
## 13                 24.75194 mins
## 14                 12.10560 mins
```

#Bike type per number of rides

```
clean_annual_trip %>%
  group_by(bike_type, customer_type) %>%
  dplyr::summarize(count_trips = n()) %>%
  ggplot(aes(x= bike_type, y=count_trips, fill=customer_type, color=customer_type)) +
  geom_bar(stat='identity', position = 'dodge') +
  theme_bw()+
  labs(title ="Bicycle Type Number of trips", x = "Bike Type", y = "Count of Trips")
```

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



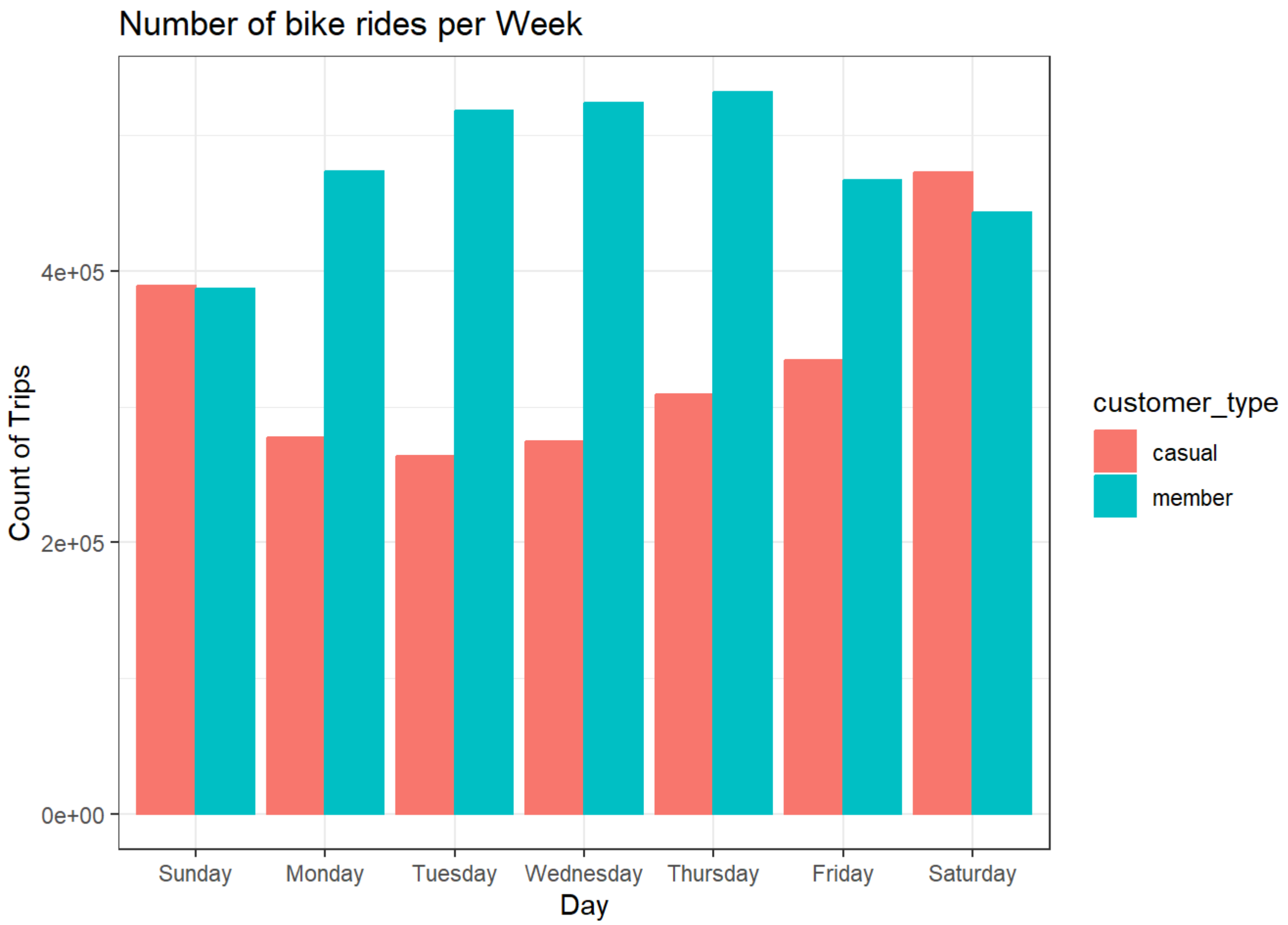
#Arranges the weekdays in order Sunday to Saturday.

```
clean_annual_trip$week_day <- ordered(clean_annual_trip$week_day, levels=c("Sunday", "Monday",
"Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"))
```

#Number bike rides per week BAR CHART

```
clean_annual_trip %>%
  group_by(customer_type,week_day) %>%
  dplyr::summarize(count_trips = n()) %>%
  ggplot(aes(x= week_day, y=count_trips, fill=customer_type, color=customer_type)) +
  geom_bar(stat='identity', position = 'dodge') +
  theme_bw()+
  labs(title ="Number of bike rides per Week", x = "Day", y = "Count of Trips")
```

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

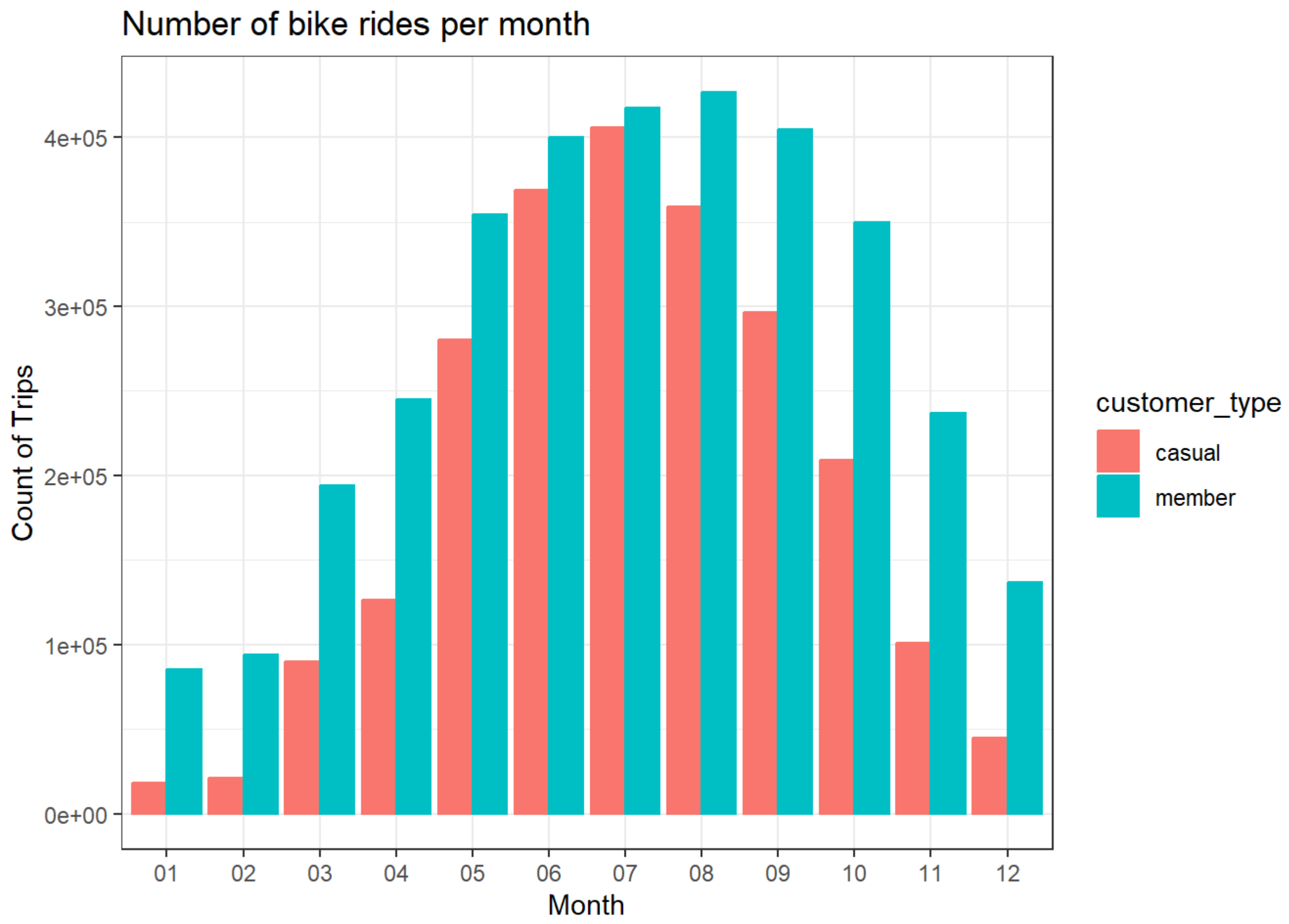


```
clean_annual_trip$month <- ordered(clean_annual_trip$month, levels=c("01", "02", "03", "04", "0
5", "06", "07","08","09","10","11","12"))
```

#Number of bike rides per month BAR CHART

```
clean_annual_trip %>%
  group_by(customer_type,month) %>%
  dplyr::summarize(count_trips = n()) %>%
  ggplot(aes(x= month, y=count_trips, fill=customer_type, color=customer_type)) +
  geom_bar(stat='identity', position = 'dodge') +
  theme_bw() +
  labs(title ="Number of bike rides per month", x = "Month", y = "Count of Trips")
```

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



#Popular start stations per casual customer type

```
clean_annual_trip %>%
  group_by(customer_type,start_station_name) %>%
  dplyr::summarise(number_of_ride = n()) %>%
  filter(start_station_name != "", "casual"== customer_type) %>%
  arrange(-number_of_ride) %>%
  head(n=10) %>%
  select(-customer_type)
```

```
## `summarise()` has grouped output by 'customer_type'. You can override using the
## `.groups` argument.
## Adding missing grouping variables: `customer_type`
## # A tibble: 10 × 3
## # Groups:   customer_type [1]
##    customer_type start_station_name                 number_of_ride
##    <chr>         <chr>                                       <int>
##  1 casual        Streeter Dr & Grand Ave                     58082
##  2 casual        DuSable Lake Shore Dr & Monroe St           31860
##  3 casual        Millennium Park                             25523
##  4 casual        Michigan Ave & Oak St                       25264
##  5 casual        DuSable Lake Shore Dr & North Blvd          23655
##  6 casual        Shedd Aquarium                              20265
##  7 casual        Theater on the Lake                         18450
##  8 casual        Wells St & Concord Ln                       16214
##  9 casual        Dusable Harbor                              14101
## 10 casual        Clark St & Armitage Ave                     13802
```

#Popular start stations per memebr customer type

```
clean_annual_trip %>%
  group_by(customer_type,start_station_name) %>%
  dplyr::summarise(number_of_ride = n()) %>%
  filter(start_station_name != "", "member"== customer_type) %>%
  arrange(-number_of_ride) %>%
  head(n=10) %>%
  select(-customer_type)
```

```
## `summarise()` has grouped output by 'customer_type'. You can override using the
## `.groups` argument.
## Adding missing grouping variables: `customer_type`
```

```
## # A tibble: 10 × 3
## # Groups:   customer_type [1]
##    customer_type start_station_name           number_of_ride
##    <chr>         <chr>                                 <int>
##  1 member        Kingsbury St & Kinzie St              24936
##  2 member        Clark St & Elm St                     22037
##  3 member        Wells St & Concord Ln                 21298
##  4 member        University Ave & 57th St              19949
##  5 member        Clinton St & Washington Blvd          19827
##  6 member        Ellis Ave & 60th St                   19501
##  7 member        Loomis St & Lexington St              19127
##  8 member        Wells St & Elm St                     18986
##  9 member        Clinton St & Madison St               18931
## 10 member        Broadway & Barry Ave                  17756
```

Recommendations and conclusion

- Members use classic bike and electric bike.
- Finding from the data shows that members use bike sharing on weekdays while casual tends to use bike on weekends.

- Both members and casual uses bike share from May to October the most.
- The top five start station for casual riders are Streeter Dr & Grand Ave, DuSable Lake Shore Dr & Monroe St,Millennium Park, Michigan Ave & Oak St,DuSable Lake Shore Dr & North Blvd in order.
- To convert casual riders to member riders a marketing strategy has to focus on the top starting stations on weekend targeting classic and electrical bike riders. Summer edition membership campaign can is also recommended.