Bike-Share Case Study

Adam LaFave

2022-03-22

Introduction

This case study covers bike share data from 03-01-2021 to 02-28-2022 and contains more than 5 million logged rides. A fictional company name: Cyclistic, is used per the the licensing agreement from the data source. Cyclistic is a successful bike share company that has grown to a fleet of 5,824 bikes. The bikes are geotracked and locked into a network of 692 stations across Chicago.

Cyclistic offers different pricing plans and categorizes customers by the type of pricing plan they purchase.

- Customers who purchase *single-ride* or *full-day* passes are referred to as **casual riders**.
- Customers who purchase *annual memberships* are referred to as **members**.

This analysis uses the six stages of the analytics process: **ask**, **prepare**, **process**, **analyze**, **share**, and **act**.

Identifying the Key Stakeholders

- Lily Moreno: Director of marketing and my manager
- Cyclistic Executive Team: notoriously detail-oriented executive team who approves recommended marketing programs

Step 1: Ask

1. How do annual members and causal 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?

- The stakeholders have asked me to focus on the first question above.

Step 2: Prepare

Key Deliverables

- A clear statement of the business task
- A description of all the data sources used
- Documentation of any cleaning or manipulation of data
- A summary of my analysis
- Supporting visualizations and key findings
- My top three recommendations based on my analysis

Task: 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.

Data Source The data has been made available by Motivate International Inc under this licensing agreement.

View the Data Source Here

Bias & Limitations: Does the data **ROCCC**? (Rating Low to High)

- **R**eliable: **High**
- **O**riginal: **High**
- **C**omprehensive: **High**
- **C**urrent: **High**
- **C**ited: **High**

The data source is credible and reliable and is good for making business suggestions. There is a large population sample size of over 5.5 million rides and the data is original, current, and cited.

Step 3: Process

#Load Packages

```
library(tidyverse) #for data import and wrangling
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

```
library(lubridate) #for date functions
```

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

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

```
library(ggplot2)   #for visualization
library(scales)    #for ggplot scaling
```

```
##
## Attaching package: 'scales'

## The following object is masked from 'package:purrr':
##
##     discard

## The following object is masked from 'package:readr':
##
##     col_factor
```

#Loading the CSV files

```
mar_2021 <- read.csv("202103-divvy-tripdata.csv")
apr_2021 <- read.csv("202104-divvy-tripdata.csv")
may_2021 <- read.csv("202105-divvy-tripdata.csv")
```

```r
jun_2021 <- read.csv("202106-divvy-tripdata.csv")
jul_2021 <- read.csv("202107-divvy-tripdata.csv")
aug_2021 <- read.csv("202108-divvy-tripdata.csv")
sep_2021 <- read.csv("202109-divvy-tripdata.csv")
oct_2021 <- read.csv("202110-divvy-tripdata.csv")
nov_2021 <- read.csv("202111-divvy-tripdata.csv")
dec_2021 <- read.csv("202112-divvy-tripdata.csv")
jan_2022 <- read.csv("202201-divvy-tripdata.csv")
feb_2022 <- read.csv("202202-divvy-tripdata.csv")
```

#Checking the colomn names of each dataframe to make sure they match so we can join the data sets with no issues.

```r
colnames(mar_2021)
```

```
##  [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"
```

```r
colnames(apr_2021)
```

```
##  [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"
```

```r
colnames(may_2021)
```

```
##  [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"
```

```r
colnames(jun_2021)
```

```
##  [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"
```

```r
colnames(jul_2021)
```

```
##  [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"
```

```r
colnames(aug_2021)
```

```
##  [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"
```

```r
colnames(sep_2021)
```

```
##  [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"
```

```r
colnames(oct_2021)
```

```
##  [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"
```

```r
colnames(nov_2021)
```

```
##  [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"
```

```r
colnames(dec_2021)
```

```
##  [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"
```

```r
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"
```

```r
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"
```

#Now we can join them.

```r
all_trips <- bind_rows(mar_2021, apr_2021, may_2021, jun_2021, jul_2021, aug_2021,
                       sep_2021, oct_2021, nov_2021, dec_2021, jan_2022, feb_2022)
```

```r

colnames(all_trips)
```

Getting familiar with the 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] "member_casual"
```

#Viewing how many rows(individual rides) the dataset has

```
nrow(all_trips)
```

```
## [1] 5667986
```

#Viewing the first 6 rows of the data set

```
head(all_trips)
```

```
##            ride_id rideable_type          started_at            ended_at
## 1 CFA86D4455AA1030  classic_bike 2021-03-16 08:32:30 2021-03-16 08:36:34
## 2 30D9DC61227D1AF3  classic_bike 2021-03-28 01:26:28 2021-03-28 01:36:55
## 3 846D87A15682A284  classic_bike 2021-03-11 21:17:29 2021-03-11 21:33:53
## 4 994D05AA75A168F2  classic_bike 2021-03-11 13:26:42 2021-03-11 13:55:41
## 5 DF7464FBE92D8308  classic_bike 2021-03-21 09:09:37 2021-03-21 09:27:33
## 6 CEBA8516FD17F8D8  classic_bike 2021-03-20 11:08:47 2021-03-20 11:29:39
##             start_station_name start_station_id
## 1 Humboldt Blvd & Armitage Ave            15651
## 2 Humboldt Blvd & Armitage Ave            15651
## 3        Shields Ave & 28th Pl            15443
## 4  Winthrop Ave & Lawrence Ave     TA1308000021
## 5     Glenwood Ave & Touhy Ave              525
## 6     Glenwood Ave & Touhy Ave              525
##                      end_station_name end_station_id start_lat start_lng
## 1             Stave St & Armitage Ave          13266  41.91751 -87.70181
## 2 Central Park Ave & Bloomingdale Ave          18017  41.91751 -87.70181
## 3                Halsted St & 35th St   TA1308000043  41.84273 -87.63549
## 4              Broadway & Sheridan Rd          13323  41.96881 -87.65766
## 5           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
## 6           Chicago Ave & Sheridan Rd           E008  42.01270 -87.66606
##    end_lat   end_lng member_casual
## 1 41.91774 -87.69139        casual
## 2 41.91417 -87.71676        casual
## 3 41.83066 -87.64717        casual
## 4 41.95283 -87.64999        casual
## 5 42.05049 -87.67782        casual
## 6 42.05049 -87.67782        casual
```

#Making sure there are only two categories of rider types

```
table(all_trips$member_casual)
```

```
##
##  casual  member
## 2540693 3127293
```

#Now checking out the statistical summary of the data

```
summary(all_trips)
```

```
##    ride_id          rideable_type       started_at          ended_at        
##  Length:5667986     Length:5667986     Length:5667986     Length:5667986    
##  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:5667986     Length:5667986     Length:5667986     Length:5667986    
##  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.   :-87.84   Min.   :41.39   Min.   :-88.97  
##  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.   :45.64   Max.   :-73.80   Max.   :42.17   Max.   :-87.49  
##                                   NA's   :4617    NA's   :4617    
##  member_casual     
##  Length:5667986    
##  Class :character  
##  Mode  :character  
##                    
##                    
##                    
##                    
```

A few observations so far

- I notice the 'started_at' and 'ended_at' columns are in character type format and will need to be changed to datetime format for wrangling purposes
- I will want to add a day, month, and trip duration column which will provide additional opportunities to aggregate and visualize the data
- I notice there are some null values in the latitude and longitude columns and will want to make sure to exclude those from any analysis

#Loading lubridate to the library

```
library(reshape2)
```

```
## Warning: package 'reshape2' was built under R version 4.1.3
```

```
## 
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
## 
##     smiths
```

#Casting the 'started_at' column to date type and Creating a date column

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

#creating a day column

```
all_trips$day <- format(as.Date(all_trips$date), "%d")
```

#Creating a month column

```
all_trips$month <- format(as.Date(all_trips$date), "%m")
```

#Creating a column by day of the week column

```
all_trips$day_of_week <- format(as.Date(all_trips$date), "%A")
```

#Calulating the ride length(in seconds)

```
all_trips$ride_length <- difftime(all_trips$ended_at,all_trips$started_at)
```

#Inspecting the structure of the columns

```
str(all_trips)
```

```
## 'data.frame':    5667986 obs. of  18 variables:
##  $ ride_id           : chr  "CFA86D4455AA1030" "30D9DC61227D1AF3" "846D87A15682A284" "994D05AA75A168I
##  $ rideable_type     : chr  "classic_bike" "classic_bike" "classic_bike" "classic_bike" ...
##  $ started_at        : chr  "2021-03-16 08:32:30" "2021-03-28 01:26:28" "2021-03-11 21:17:29" "2021-(
##  $ ended_at          : chr  "2021-03-16 08:36:34" "2021-03-28 01:36:55" "2021-03-11 21:33:53" "2021-(
##  $ start_station_name: chr  "Humboldt Blvd & Armitage Ave" "Humboldt Blvd & Armitage Ave" "Shields A
##  $ start_station_id  : chr  "15651" "15651" "15443" "TA1308000021" ...
##  $ end_station_name  : chr  "Stave St & Armitage Ave" "Central Park Ave & Bloomingdale Ave" "Halsted
##  $ end_station_id    : chr  "13266" "18017" "TA1308000043" "13323" ...
##  $ start_lat         : num  41.9 41.9 41.8 42 42 ...
##  $ start_lng         : num  -87.7 -87.7 -87.6 -87.7 -87.7 ...
##  $ end_lat           : num  41.9 41.9 41.8 42 42.1 ...
##  $ end_lng           : num  -87.7 -87.7 -87.6 -87.6 -87.7 ...
##  $ member_casual     : chr  "casual" "casual" "casual" "casual" ...
##  $ date              : Date, format: "2021-03-16" "2021-03-28" ...
##  $ day               : chr  "16" "28" "11" "11" ...
##  $ month             : chr  "03" "03" "03" "03" ...
##  $ day_of_week       : chr  "Tuesday" "Sunday" "Thursday" "Thursday" ...
##  $ ride_length       : 'difftime' num  244 627 984 1739 ...
##   ..- attr(*, "units")= chr "secs"
```

#Converting the 'ride_length' column we just created from factor to numeric so we can run calculations on the data

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

#Checking to make sure the ride_length column is numeric

```
is.numeric(all_trips$ride_length)
```

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

#Checking out the the average ride time by each day for members vs casual users

```
aggregate(all_trips$ride_length ~ all_trips$member_casual + all_trips$day_of_week, FUN = mean)
```

```
##    all_trips$member_casual all_trips$day_of_week all_trips$ride_length
## 1                   casual                Friday             1810.2679
```

```
## 2                  member               Friday           793.0721
## 3                  casual               Monday          1908.5709
## 4                  member               Monday           783.1895
## 5                  casual             Saturday          2072.7200
## 6                  member             Saturday           905.9669
## 7                  casual               Sunday          2249.7389
## 8                  member               Sunday           929.1869
## 9                  casual             Thursday          1670.6572
## 10                 member             Thursday           760.4469
## 11                 casual              Tuesday          1670.8546
## 12                 member              Tuesday           760.7414
## 13                 casual            Wednesday          1660.6197
## 14                 member            Wednesday           759.0128
```

#Re-ordering the output to order the days of the week

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

#Making sure the output starts with Sunday now

```
aggregate(all_trips$ride_length ~ all_trips$member_casual + all_trips$day_of_week, FUN = mean)
```

```
##    all_trips$member_casual all_trips$day_of_week all_trips$ride_length
## 1                   casual                Sunday             2249.7389
## 2                   member                Sunday              929.1869
## 3                   casual                Monday             1908.5709
## 4                   member                Monday              783.1895
## 5                   casual               Tuesday             1670.8546
## 6                   member               Tuesday              760.7414
## 7                   casual             Wednesday             1660.6197
## 8                   member             Wednesday              759.0128
## 9                   casual              Thursday             1670.6572
## 10                  member              Thursday              760.4469
## 11                  casual                Friday             1810.2679
## 12                  member                Friday              793.0721
## 13                  casual              Saturday             2072.7200
## 14                  member              Saturday              905.9669
```

Step 4 and 5: Analyze & Share Observations

#Taking a look at the categorized data

```
all_trips%>%
  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)
```

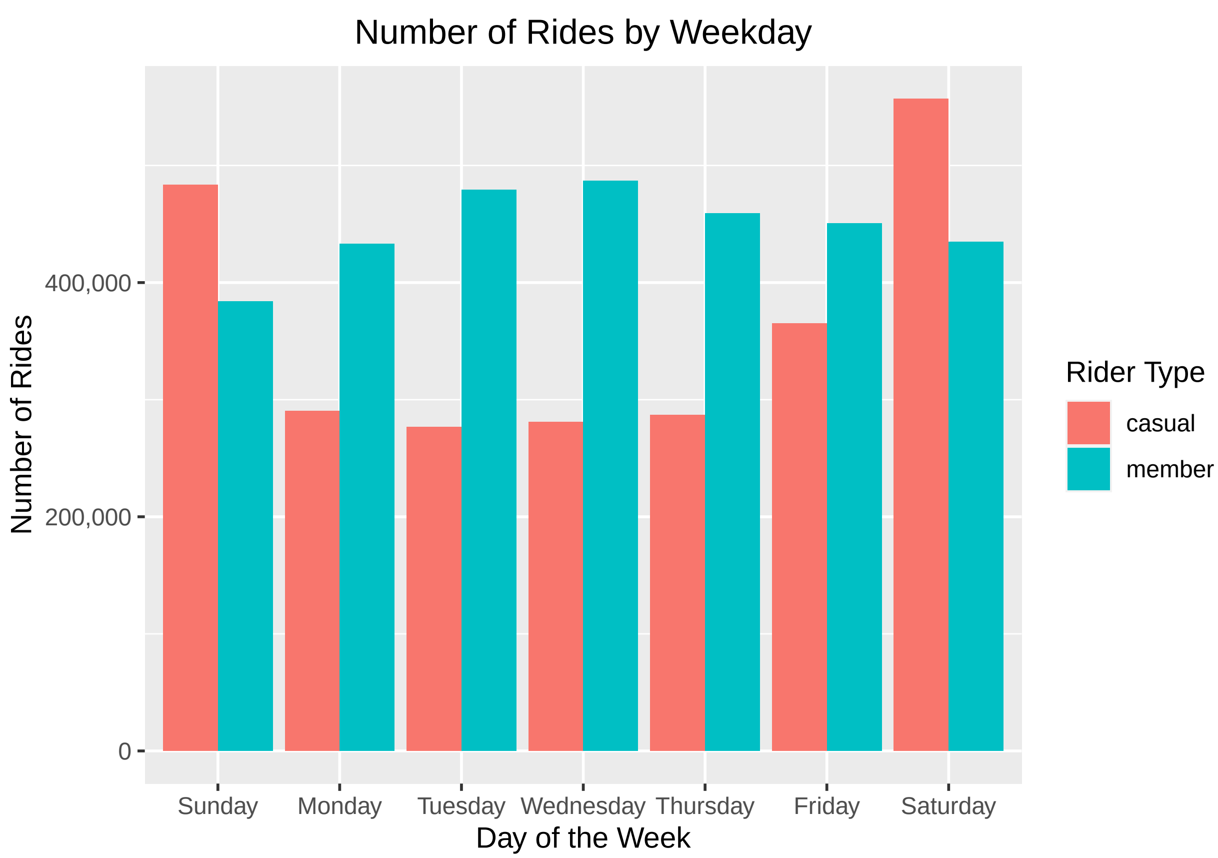
```
## `summarise()` has grouped output by 'member_casual'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 14 x 4
## # Groups:   member_casual [2]
##    member_casual weekday number_of_rides average_duration
```

```
##    <chr>          <ord>            <int>           <dbl>
##   1 casual        Sun             483607           2250.
##   2 casual        Mon             290537           1909.
##   3 casual        Tue             276684           1671.
##   4 casual        Wed             280745           1661.
##   5 casual        Thu             287096           1671.
##   6 casual        Fri             364903           1810.
##   7 casual        Sat             557121           2073.
##   8 member        Sun             383959            929.
##   9 member        Mon             432821            783.
##  10 member        Tue             479230            761.
##  11 member        Wed             486765            759.
##  12 member        Thu             459171            760.
##  13 member        Fri             450501            793.
##  14 member        Sat             434846            906.
```

#Now let's visualize the number of rides by rider type and day of week over the past year

```
all_trips %>%
  ggplot()+
  geom_bar(aes(x=day_of_week, fill = member_casual), position = "dodge") +
  labs(title="Number of Rides by Weekday") +
  xlab("Day of the Week") + ylab("Number of Rides")+
  theme(plot.title = element_text(hjust = 0.5)) +
  guides(fill = guide_legend(title = "Rider Type")) +
  scale_y_continuous(labels = scales::comma)
```



Observations/Key Findings

- We observe that casual rider's trips increase on the weekends, whereas member's trips increase towards the middle of the week. One hypothesis could be that a larger majority of the riders on the weekend are tourist and not locals. Another hypothesis is that it is likely that the members are taking their trips to and from work whereas the tourist may be exploring Chicago on the weekends. If we had access from the data of the rider ID, we could explore further to identify which of the casual riders take recurring trips throughout the year, during the week, and not only on the weekends. Likewise, we could target casual riders that ride every weekend. We could run targeted ads to those riders letting them know about the membership and possibly offering them a special for their previous loyalty.
- Let's explore the data set further and visualize the average trip duration by rider type throughout the week.

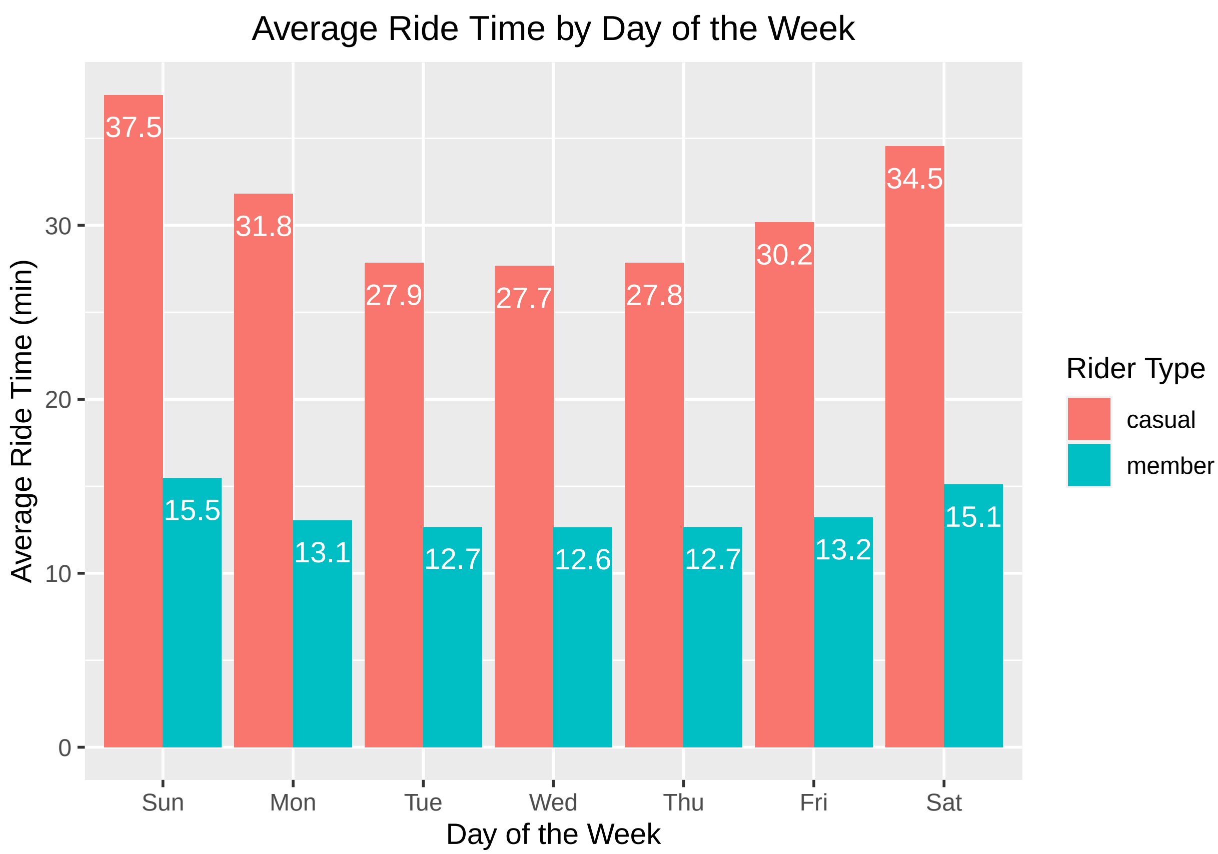
#First we need to create a ride_length column in minutes instead of seconds for visualization purposes

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

#Now onto visualizing the average ride length by rider type throughout the week

```
all_trips %>%
  mutate(weekday = wday(started_at, label = TRUE)) %>%
  group_by(member_casual, weekday) %>%
  summarise(number_of_rides = n()
        ,average_duration = mean(ride_length_min)) %>%
  arrange(member_casual, weekday)  %>%
  ggplot(aes(x = weekday, y = average_duration, fill = member_casual)) +
  geom_col(position = "dodge")+
   labs(title="Average Ride Time by Day of the Week") +
  xlab("Day of the Week") + ylab("Average Ride Time (min)")+
  theme(plot.title = element_text(hjust = 0.5)) +
  guides(fill = guide_legend(title = "Rider Type")) +
  geom_text(aes(label = (sprintf("%0.1f", round(average_duration, digits = 2)))),
           position = position_dodge(0.9), vjust = 2, size = 4, color = "#ffffff")
```

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



Observations/Key Findings

- The average ride time over the last year for casual riders was more than double that of member riders throughout the entire week. For both groups, the ride time increases towards the weekend with Wednesday having the lowest average ride time and Sunday the longest. The average member ride time stays relatively linear with a minimum of 12.6 minutes on Wednesday and a maximum of 15.5 minutes on Sunday. The average casual ride time has a minimum ride of 27.7 minutes on Wednesday and 37.5 minutes on Sunday. Looks like everyone's going out for the casual Sunday Stroll. One hypothesis here is that the membership riders get plenty of riding in and don't want to spend their weekends riding.
- Now let's checkout and visualize the ridership by month and rider type to identify any trends there.

#First, adding a Year Month Column so we can arrange the months sequentially

```
all_trips$YearMonth <- format(as.Date(all_trips$started_at), "%Y-%m")
```

#Adding a column to sum the amount of monthly trips

```
all_trips$rides <- 1
```

#Creating a monthly_trips table by YearMonth

```
 monthly_trips <- all_trips %>%
  group_by(YearMonth) %>%
  summarise(monthlyTrips = sum(rides)) %>%
  arrange(YearMonth)
```

#Creating YearMonth column by rider type

```r
monthly_trips_by_ridertype <- all_trips %>%
  group_by(YearMonth, member_casual) %>%
  summarise(monthly_trips = sum(rides)) %>%
  arrange(YearMonth, member_casual)
```

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

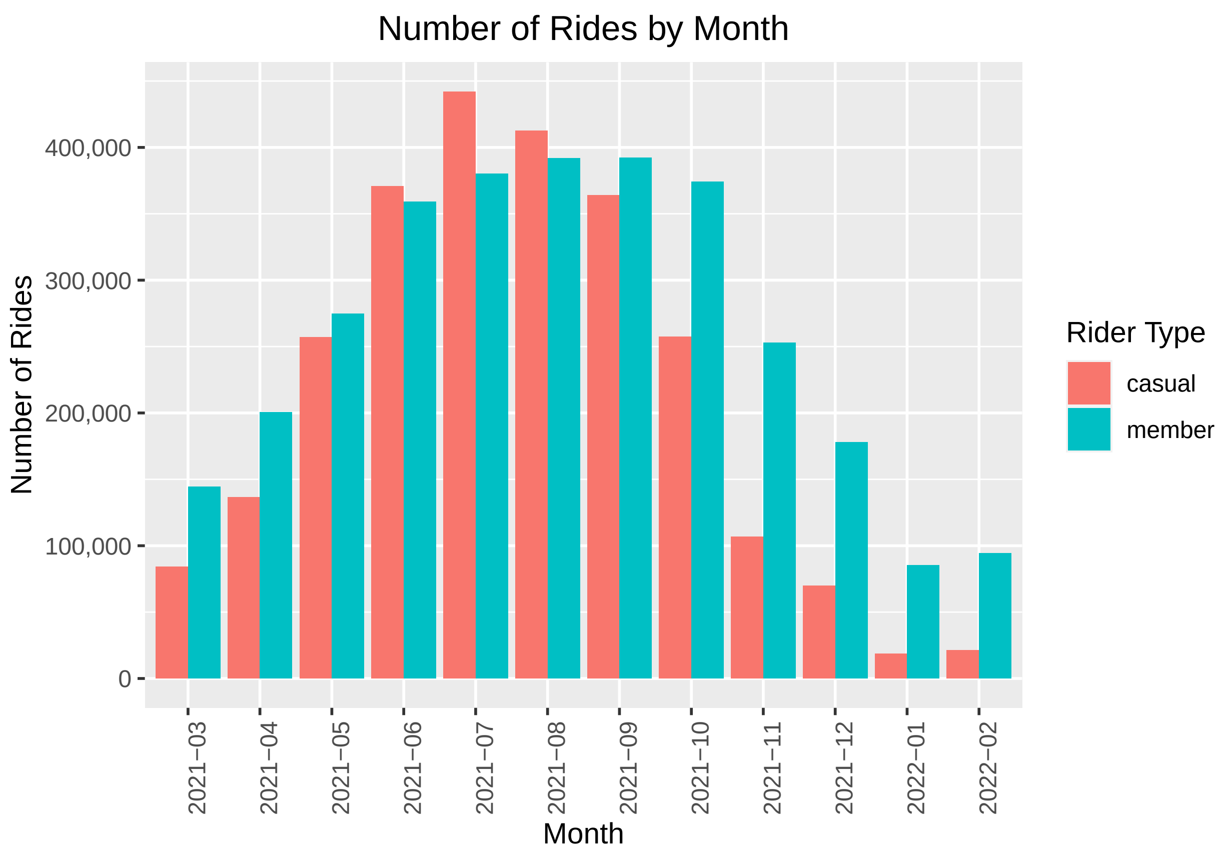
#Viewing our new dataframe before visualizing

```r
head(monthly_trips_by_ridertype)
```

```
## # A tibble: 6 x 3
## # Groups:   YearMonth [3]
##   YearMonth member_casual monthly_trips
##   <chr>     <chr>                 <dbl>
## 1 2021-03   casual                84033
## 2 2021-03   member               144463
## 3 2021-04   casual               136601
## 4 2021-04   member               200629
## 5 2021-05   casual               256916
## 6 2021-05   member               274717
```

#Now we can checkout a visualization of the total trips by month and ridertype over the past year

```r
  ggplot(monthly_trips_by_ridertype,aes( x=YearMonth, y=monthly_trips, fill=member_casual)) +
    geom_col(position="dodge") +
  labs(title = "Number of Rides by Month") +
  xlab("Month") + ylab("Number of Rides") +
  theme(plot.title = element_text(hjust = 0.5)) +
  scale_y_continuous(labels = scales::comma) +
  guides(fill = guide_legend(title = "Rider Type")) +
  theme(axis.text.x = element_text(angle = 90))
```



Observations/Key Findings

- We can observe from this visual that for both rider type groups the number of rides by month reaches a peak during the warmer summer months of July and August and is at it's lowest in the month of January. This makes sense given that Chicago has frigid winter months. A possible suggestion would be to use heavier advertising directed at casual riders during the warmer months while the cold weather is not on their minds. The marketing team could prompt user's in the app during the summer months requesting them to take a short survey about their experience in exchange for one free ride. At the end of the survey they could offer them a membership plan showing them the benefits. An email campaign to rider's shortly after there ride would also be a good recommendation.
- The above three visualizations should help our stakeholder's and marketing team get a good idea as to how casual riders and members use the bikes differently

#We can dig into the station data a bit. Let's checkout the most used stations by station name. #We eliminate any rows that don't contain a start_station_name

```
all_trips %>%
  group_by(start_station_name, end_station_name) %>%
  filter(start_station_name!="NULL") %>%
  summarize(rides = n()) %>%
  ungroup %>%
  top_n(10)
```

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

```
## # A tibble: 10 x 3
##    start_station_name          end_station_name             rides
##    <chr>                       <chr>                        <int>
##  1 ""                          ""                          437912
##  2 "Ellis Ave & 55th St"       "Ellis Ave & 60th St"         4846
##  3 "Ellis Ave & 60th St"       "Ellis Ave & 55th St"         5449
##  4 "Ellis Ave & 60th St"       "University Ave & 57th St"    4266
##  5 "Lake Shore Dr & Monroe St" "Lake Shore Dr & Monroe St"   5038
##  6 "Michigan Ave & Oak St"     "Michigan Ave & Oak St"       6662
##  7 "Millennium Park"           "Millennium Park"             6453
##  8 "Streeter Dr & Grand Ave"   "Streeter Dr & Grand Ave"    13068
##  9 "Theater on the Lake"       "Theater on the Lake"         3985
## 10 "University Ave & 57th St"  "Ellis Ave & 60th St"         4065
```

- Notice that 5 of the top 10 most used starting stations were round trips. One of the major stations that stands out is Streeter Dr and Grand Ave with 13,068 round trips.
- We notice that the ride count(sample size) dropped from over 5 million in our previous analysis examples to a little under 438,000 here. The reason is we eliminated any of the rows that didn't have a start station name in it. 430,000 is still a large sample size so the integrity has changed, but is still reliable.

#Now we'll checkout the most used stations grouping by rider type. #The first two rows indicate the total rides by rider type.

```
all_trips %>%
  group_by(start_station_name, end_station_name,member_casual) %>%
  filter(start_station_name!="NULL") %>%
  summarize(rides = n()) %>%
  ungroup %>%
  top_n(10)
```

```
## 'summarise()' has grouped output by 'start_station_name', 'end_station_name'.
## You can override using the '.groups' argument.
## Selecting by rides
```

```
## # A tibble: 10 x 4
##    start_station_name          end_station_name            member_casual  rides
##    <chr>                       <chr>                       <chr>          <int>
##  1 ""                          ""                          casual        205497
##  2 ""                          ""                          member        232415
##  3 "Ellis Ave & 55th St"       "Ellis Ave & 60th St"       member          3893
##  4 "Ellis Ave & 60th St"       "Ellis Ave & 55th St"       member          4442
##  5 "Ellis Ave & 60th St"       "University Ave & 57th St"  member          3653
##  6 "Lake Shore Dr & Monroe St" "Lake Shore Dr & Monroe St" casual          4585
##  7 "Michigan Ave & Oak St"     "Michigan Ave & Oak St"     casual          5899
##  8 "Millennium Park"           "Millennium Park"           casual          6089
##  9 "Streeter Dr & Grand Ave"   "Streeter Dr & Grand Ave"   casual         11702
## 10 "University Ave & 57th St"  "Ellis Ave & 60th St"       member          3573
```

- We notice that 11,702 of the 13,068 riders for Streeter Dr & Grand Ave that we mentioned above were started by casual riders.

Step 6: Act

- As mentioned above in the summaries under the visualizations. There is very large set of casual riders that can be converted to members.

- Use heavier advertising directed at casual riders during the warmer months while the cold weather is not on their minds. The marketing team could prompt user's in the app during the summer months requesting them to take a short survey about their experience in exchange for one free ride. An email campaign to rider's shortly after there ride would also be a good recommendation.
- If we had access to rider ID and rider demographics, we could explore further to identify which of the casual riders take recurring trips throughout the year, during the week, and not only on the weekends. Likewise, we could target casual riders that ride every weekend. We could run targeted ads to those riders letting them know about the membership and possibly offering them a special for their previous loyalty.
- I would ask the stakeholders for more user specific demographics in order to run a more thorough analysis which would allow me to give a much more detailed recommendation to the marketing department.

Thanks for checking out my analysis!