Cyclistic Bikeshare Case Study

Randall Brown

2022-08-22

The Question

I have been tasked with finding the similarities and differences between members and casual customers and how they use the Cyclistic Bikeshare service.

The Data

The data used in this analysis has been provided to me by the company through a secure link that contains all of the company's ride information. Any information shared in this analysis should be considered private.

First, the tidyverse must be installed and loaded.

```r
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## 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()
```

Then the data must be loaded.

```r
trip_data <- read_csv("2022_07_bikeshare_data.csv")
```

```
## Rows: 823488 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (8): member_casual, rideable_type, date, start_station_name, start_stat...
## dbl  (1): day_of_week
## time (2): start_time, end_time
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
glimpse(trip_data)
```

```
## Rows: 823,488
## Columns: 11
```

```
## $ member_casual      <chr> "member", "casual", "casual", "casual", "member", "~
## $ rideable_type      <chr> "classic_bike", "classic_bike", "classic_bike", "cl~
## $ date               <chr> "7/5/2022", "7/26/2022", "7/3/2022", "7/31/2022", "~
## $ start_time         <time> 08:12:00, 12:53:00, 13:58:00, 17:44:00, 19:49:00, ~
## $ end_time           <time> 08:24:00, 12:55:00, 14:06:00, 18:42:00, 20:15:00, ~
## $ start_station_name <chr> "Ashland Ave & Blackhawk St", "Buckingham Fountain ~
## $ start_station_id   <chr> "13224", "15541", "15541", "15541", "TA1307000117",~
## $ end_station_name   <chr> "Kingsbury St & Kinzie St", "Michigan Ave & 8th St"~
## $ end_station_id     <chr> "KA1503000043", "623", "623", "TA1307000164", "TA13~
## $ day_of_week        <dbl> 3, 3, 1, 1, 4, 6, 2, 5, 1, 1, 6, 7, 3, 4, 2, 1, 5, ~
## $ ride_length        <chr> "0:12:00", "0:02:00", "0:08:00", "0:58:00", "0:26:0~
```

```
top_ten_starts <- trip_data %>%
  filter(grepl('Streeter Dr & Grand Ave|DuSable Lake Shore Dr & North Blvd|Michigan Ave & Oak St|DuSabl
top_ten_ends <- trip_data %>%
  filter(grepl('Streeter Dr & Grand Ave|DuSable Lake Shore Dr & North Blvd|Michigan Ave & Oak St|DuSabl
```

Cleaning Process

The data you see in this analysis has been cleaned. Originally, the data contained numerous blank cells that needed to be removed in order to get valid answers. I cleaned the data by going through a few steps: *I downloaded the raw data and saved it locally on my hard drive.* I used Excel to filter and remove all of the rows containing blank cells. *I used SQL to filter the data quickly and show me important subsets of the data.

I have documented my steps in Excel, SQL, and R in order to provide any information that may be needed, some of which will be provided here.

```
sql_starts <- read_csv("top_ten_starts.csv")
```

```
## Rows: 10 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): start_station_name
## dbl (1): f0_
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
view(sql_starts)
sql_ends <- read_csv("top_ten_ends.csv")
```

```
## Rows: 10 Columns: 2
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): end_station_name
## dbl (1): f0_
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
view(sql_ends)
```

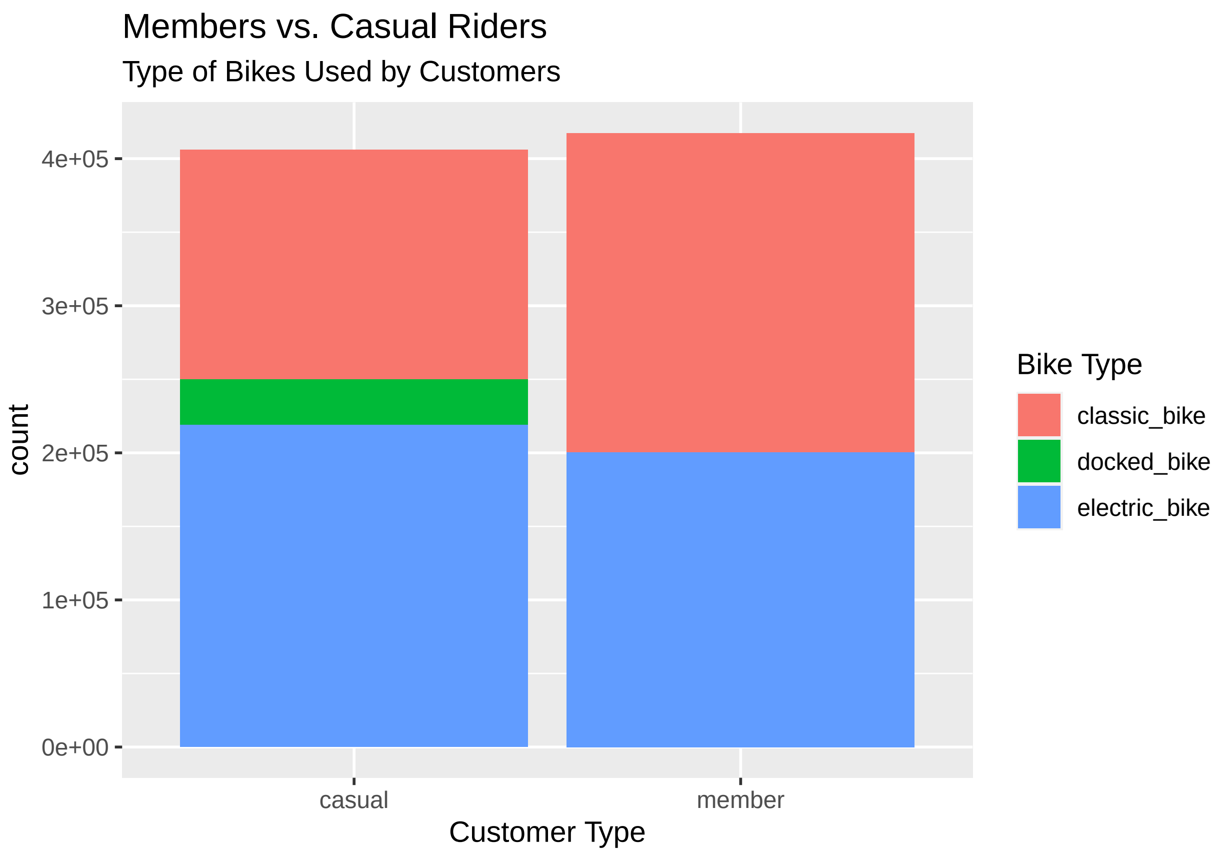
Analysis

I have decided to sort the data by date and compare casual riders to members on 5 criteria; ride length, bike type, days, starting stations, and ending stations.

Plots

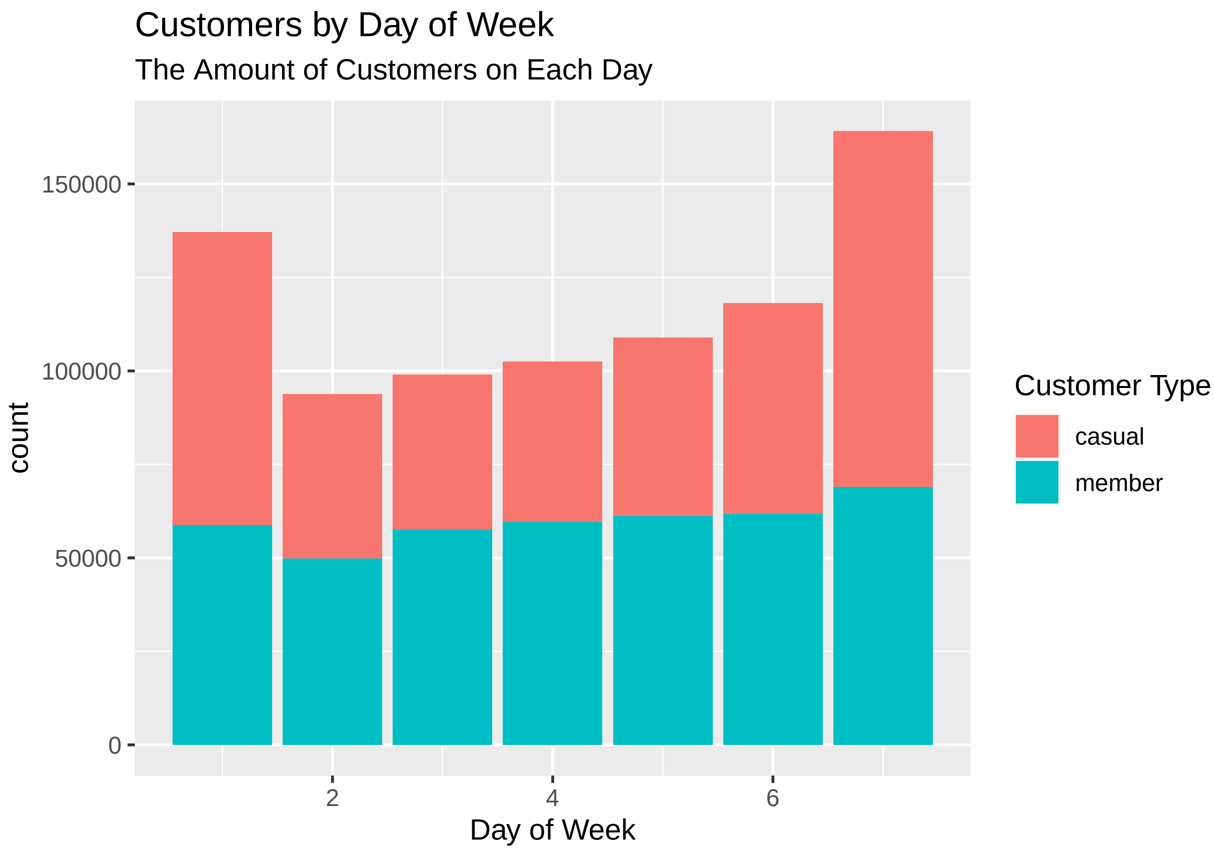
This first plot shows the relationship between riders and their preferred bike types. Members are roughly 50-50, while casual riders lean more toward electric bikes.

```
ggplot(trip_data) +
  geom_bar(aes(x=member_casual, fill=rideable_type)) +
  labs(title="Members vs. Casual Riders",
       subtitle="Type of Bikes Used by Customers",
       fill="Bike Type") +
  xlab("Customer Type")
```



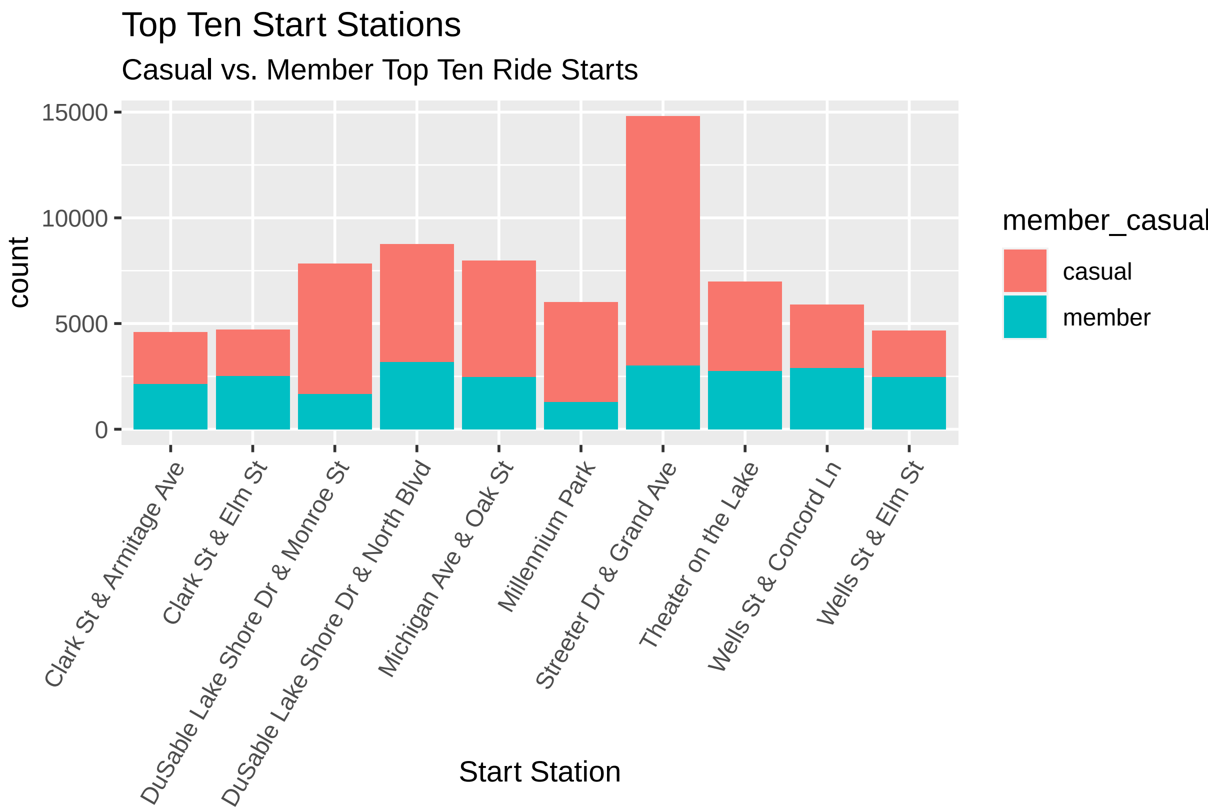
This plot shows customer riding habits by day. (1=Sunday, 2=Monday, and so on.) The graph shows an increase in casual riders on Saturdays and Sundays, and fairly consistent member riders throughout the week.

```
ggplot(trip_data) +
  geom_bar(aes(x=day_of_week, fill=member_casual)) +
  labs(title="Customers by Day of Week",
       subtitle="The Amount of Customers on Each Day",
       fill="Customer Type") +
  xlab("Day of Week")
```



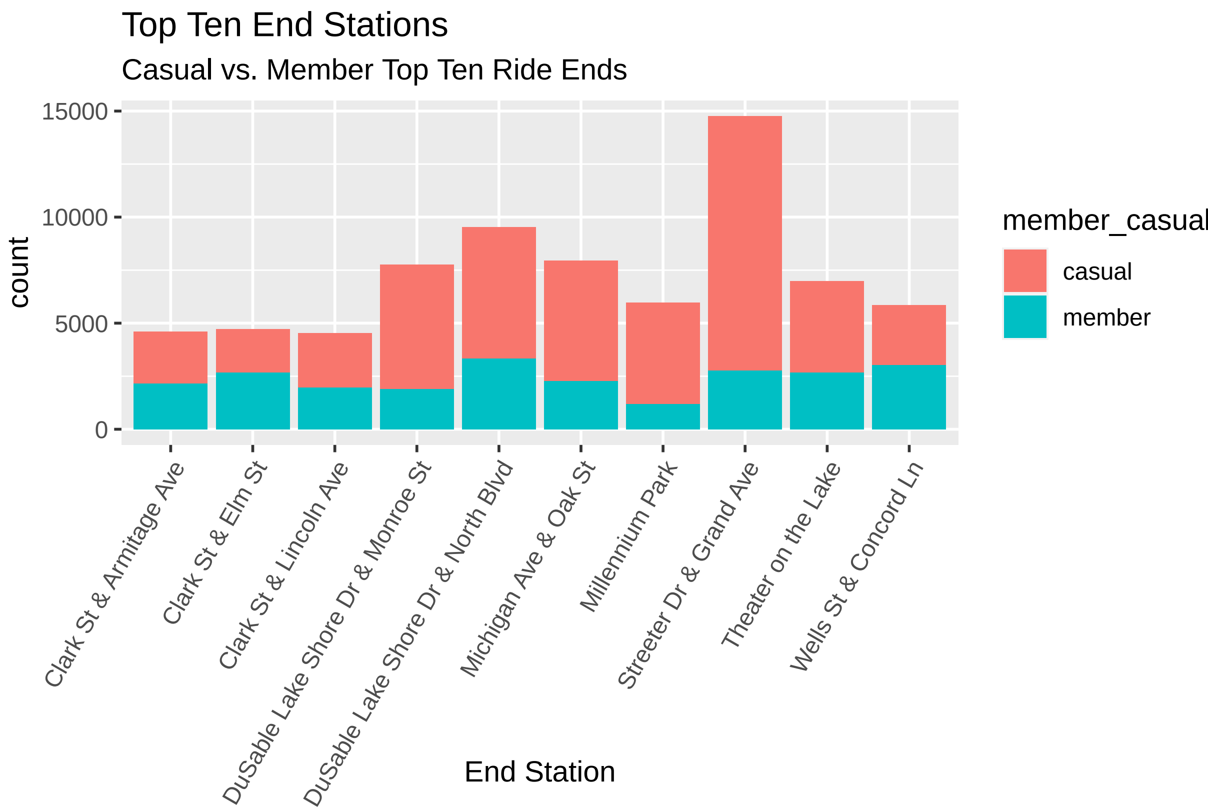
Now it's important to show the most popular starting points.

```
ggplot(top_ten_starts) +
  geom_bar(aes(x=start_station_name, fill=member_casual)) +
  labs(title="Top Ten Start Stations",
       subtitle="Casual vs. Member Top Ten Ride Starts") +
  xlab("Start Station") +
  theme(axis.text.x=element_text(angle=60)) +
  theme(axis.text.x=element_text(vjust=1)) +
  theme(axis.text.x=element_text(hjust=1),
        axis.title.x=element_text(vjust=10))
```



It's also important to show the most popular ending points.

```
ggplot(top_ten_ends) +
  geom_bar(aes(x=end_station_name, fill=member_casual)) +
  labs(title="Top Ten End Stations",
       subtitle="Casual vs. Member Top Ten Ride Ends") +
  xlab("End Station") +
  theme(axis.text.x=element_text(angle=60)) +
  theme(axis.text.x=element_text(vjust=1)) +
  theme(axis.text.x=element_text(hjust=1),
        axis.title.x=element_text(vjust=10))
```



Recommendations

Based on my analysis, it is my recommendation that Cyclistic Bikeshare focus on converting casual customers that start their rides at the Streeter Drive & Grand Avenue location (highest customer base for starts and finishes), as well as the other top 9 starting and ending locations. Weekends would be the best time to campaign in those locations in order to reach the highest percentage of casual riders. Cyclistic should also continue to invest in roughly the an equal amount of both electric bikes and classic bikes as the customer base is almost split 50-50.

If I had more information about membership prices and members' average income, I could be more detailed in who might be most likely to convert from a casual rider to a member.