Cyclistic Case Study

2024-08-25

Scenario

You are a junior data analyst working on the marketing analyst team at Cyclistic, a bike-share company in Chicago. The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, your team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, your team will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve your recommendations, so they must be backed up with compelling data insights and professional data visualizations

Business Questions

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?

Data

Our main dataset is the historical bike trip data for the past 12 months from Cyclistic, available here. The data is stored in separate spreadsheets and there are 12 .csv files being used:

- 202301-divvy-tripdata.zip
- 202302-divvy-tripdata.zip
- 202303-divvy-tripdata.zip
- 202304-divvy-tripdata.zip
- 202305-divvy-tripdata.zip
- 202306-divvy-tripdata.zip
- 202307-divvy-tripdata.zip
- 202308-divvy-tripdata.zip
- 202309-divvy-tripdata.zip
- 202310-divvy-tripdata.zip
- 202311-divvy-tripdata.zip
- 202312-divvy-tripdata.zip

This data meets the ROCCC standard.

- Reliable: This data is sourced directly from Cyclistic.
- Original: Same as Reliable - this data is sourced directly from Cyclistic.
- Comprehensive: This is a year's worth of historical data.
- Current: This data is the most recent full year (2023).
- Cited: As stated above, this data is available here.

Libaries Used

Cleaning the data

- Combine the separate data sets into one
- Remove any potential duplicates in ride_id, primary unique ID field
- Remove any entries with NA values
- Remove rows that contain NA values for missing station data

Import the .csv files into RStudio

```
trips_01_2023 <- read.csv("202301-divvy-tripdata.csv")
trips_02_2023 <- read.csv("202302-divvy-tripdata.csv")
trips_03_2023 <- read.csv("202303-divvy-tripdata.csv")
trips_04_2023 <- read.csv("202304-divvy-tripdata.csv")
trips_05_2023 <- read.csv("202305-divvy-tripdata.csv")
trips_06_2023 <- read.csv("202306-divvy-tripdata.csv")
trips_07_2023 <- read.csv("202307-divvy-tripdata.csv")
trips_08_2023 <- read.csv("202308-divvy-tripdata.csv")
trips_09_2023 <- read.csv("202309-divvy-tripdata.csv")
trips_10_2023 <- read.csv("202310-divvy-tripdata.csv")
trips_11_2023 <- read.csv("202311-divvy-tripdata.csv")
trips_12_2023 <- read.csv("202312-divvy-tripdata.csv")
```

After importing the .csv files, we're able to see that the data set has 13 variables. Using January's data as an example:

```
glimpse(trips_01_2023)
```

```
## Rows: 190,301
## 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", "~
```

We're then going to combine the data set into one to make it easier to clean.

```
cyclistic_trips_2023 <- rbind(trips_01_2023,
                              trips_02_2023,
                              trips_03_2023,
                              trips_04_2023,
                              trips_05_2023,
```

```
                            trips_06_2023,
                            trips_07_2023,
                            trips_08_2023,
                            trips_09_2023,
                            trips_10_2023,
                            trips_11_2023,
                            trips_12_2023)
```

This creates a single data set named Cyclistic_Trips_2023 with all of our data. We're then going to get rid of any duplicates from the ride_id.

We're now going to remove any NA values.

When taking a look at the data, we have removed approx. 6990 duplicate entries from the data.

```
glimpse(cyclistic_trips_2023_clean)
```

```
## Rows: 5,712,887
## 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", "~
```

We also need to verify that "casual" and "member" are the only 2 options in member_casual.

```
unique(cyclistic_trips_2023_clean$member_casual)
```

```
## [1] "member" "casual"
```

We are going to do the same check for electric bikes, classic bikes, and docked bikes in the rideable_type column.

```
unique(cyclistic_trips_2023_clean$rideable_type)
```

```
## [1] "electric_bike" "classic_bike"  "docked_bike"
```

We're also going to add a ride length column.

After adding the ride length column, we need to convert the ride_length into hours, minutes and seconds.

```
cyclistic_trips_2023_clean = cyclistic_trips_2023_clean %>%
  mutate(hours = hour(seconds_to_period(cyclistic_trips_2023_clean$ride_length)),
         minutes = minute(seconds_to_period(cyclistic_trips_2023_clean$ride_length)),)
```

We're then going to add the date, then add the trip_month column, as well as the day_of_week column.

```
cyclistic_trips_2023_clean$date <- as.Date(cyclistic_trips_2023_clean$started_at)
cyclistic_trips_2023_clean$month <- format(as.Date(cyclistic_trips_2023_clean$date), "%m")
cyclistic_trips_2023_clean$day_of_week <- format(as.Date(cyclistic_trips_2023_clean$date), "%A")
```

After this, we're going to remove entries that have a negative trip length as well as docked bikes. Docked bikes are not in circulation, so we need to remove them.

```
trip_data = cyclistic_trips_2023_clean[!(cyclistic_trips_2023_clean$rideable_type == "docked_bike" |
                                         cyclistic_trips_2023_clean$ride_length<0),]
glimpse(trip_data)
```

```
## Rows: 5,636,491
## Columns: 19
## $ 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", "~
## $ ride_length        <drtn> 651 secs, 509 secs, 794 secs, 526 secs, 919 secs, ~
## $ hours              <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ~
## $ minutes            <dbl> 10, 8, 13, 8, 15, 3, 14, 9, 12, 12, 9, 9, 7, 9, 4, ~
## $ date               <date> 2023-01-21, 2023-01-10, 2023-01-02, 2023-01-22, 20~
## $ month              <chr> "01", "01", "01", "01", "01", "01", "01", "01", "01~
## $ day_of_week        <chr> "Saturday", "Tuesday", "Monday", "Sunday", "Thursda~
```

We're then going to remove any rows that contain empty strings for the starting station name, end station, or the trip ID.

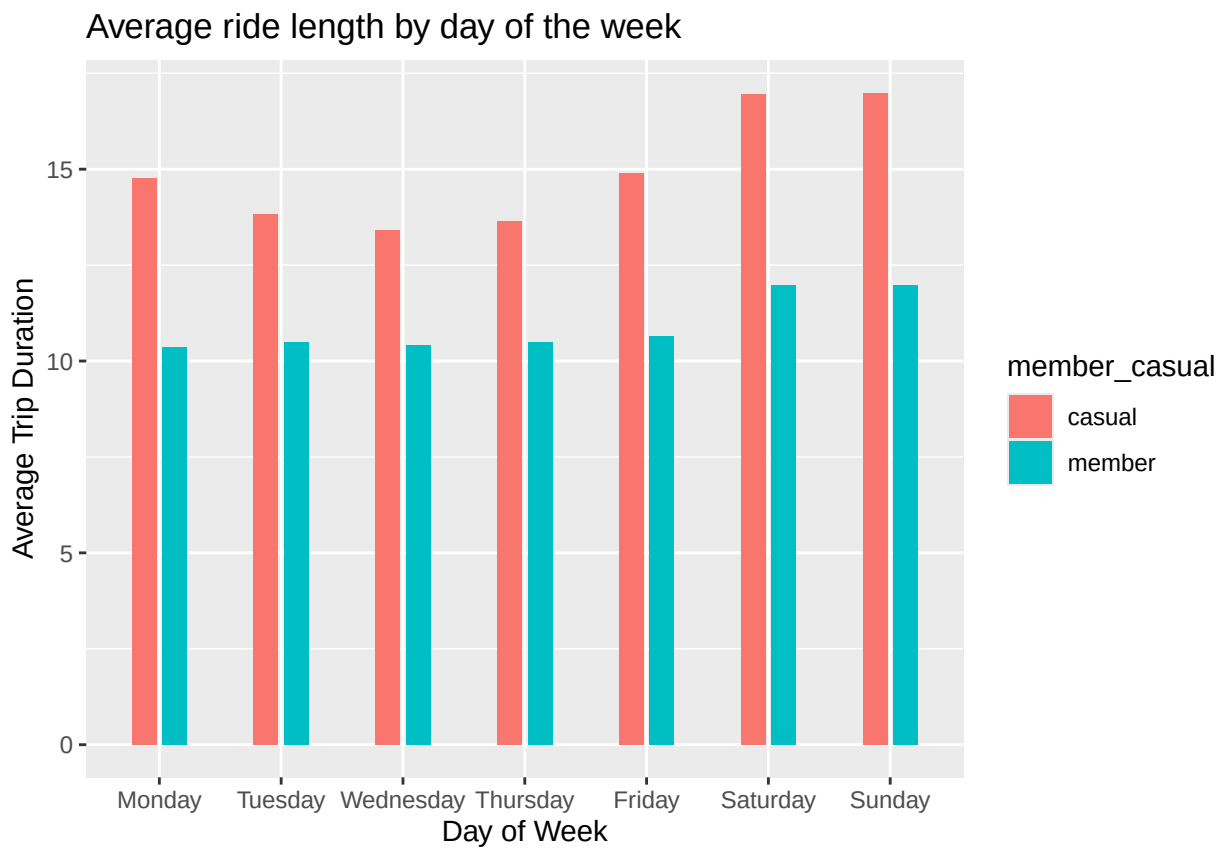
```
trip_data = trip_data[!trip_data$ride_id=="",]
trip_data = trip_data[!trip_data$start_station_name=="",]
trip_data = trip_data[!trip_data$end_station_name=="",]
```

Analyzing the Data

```
trip_data$day_of_week <- factor(trip_data$day_of_week, levels = c("Monday", "Tuesday", "Wednesday", "Thu
```

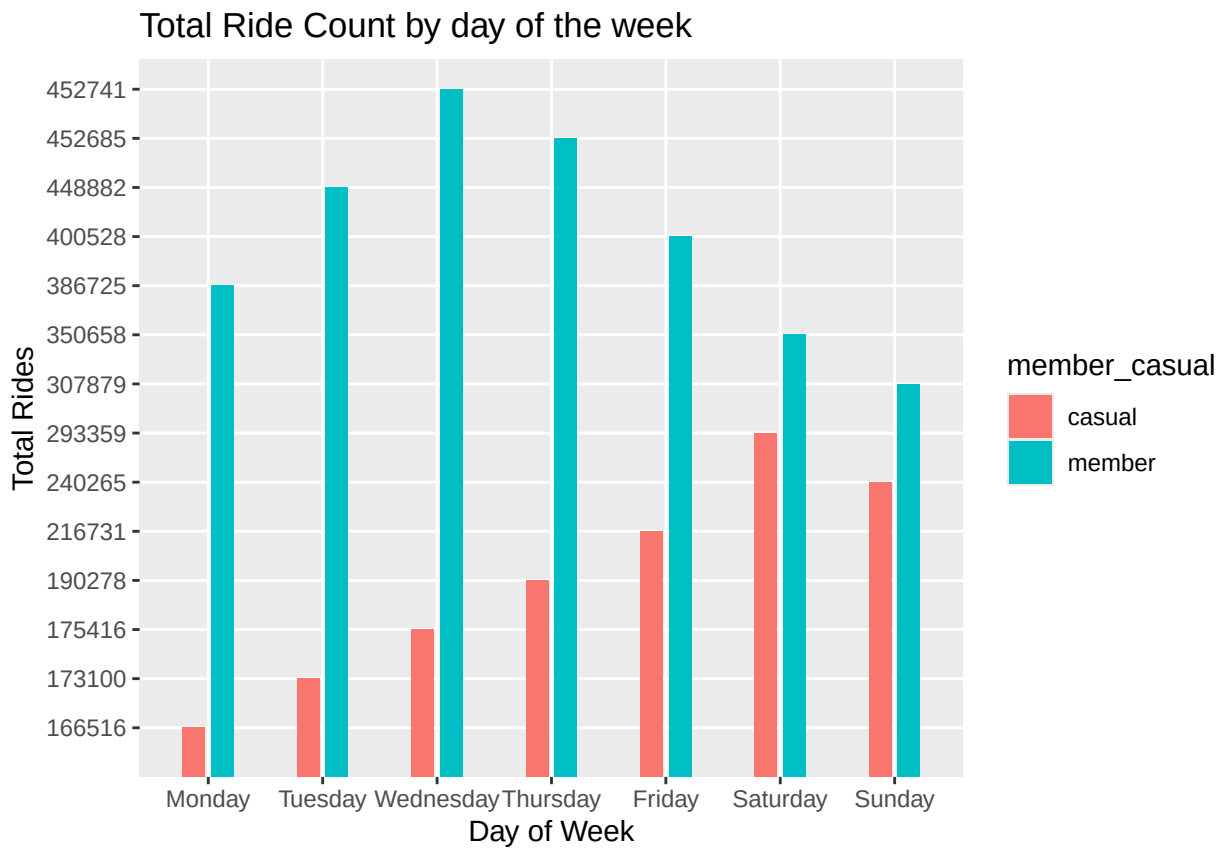
Average ride length by the day of the week

```
trip_data %>%
  group_by(member_casual, day_of_week) %>%
  summarise(average_duration = mean(minutes), .groups = "drop") %>%
  ggplot(aes(x = day_of_week, y = average_duration, fill = member_casual)) +
  geom_col(width = 0.4, position = position_dodge(width = 0.5)) +
  labs(title = "Average ride length by day of the week", x = 'Day of Week', y = 'Average Trip Duration'
```



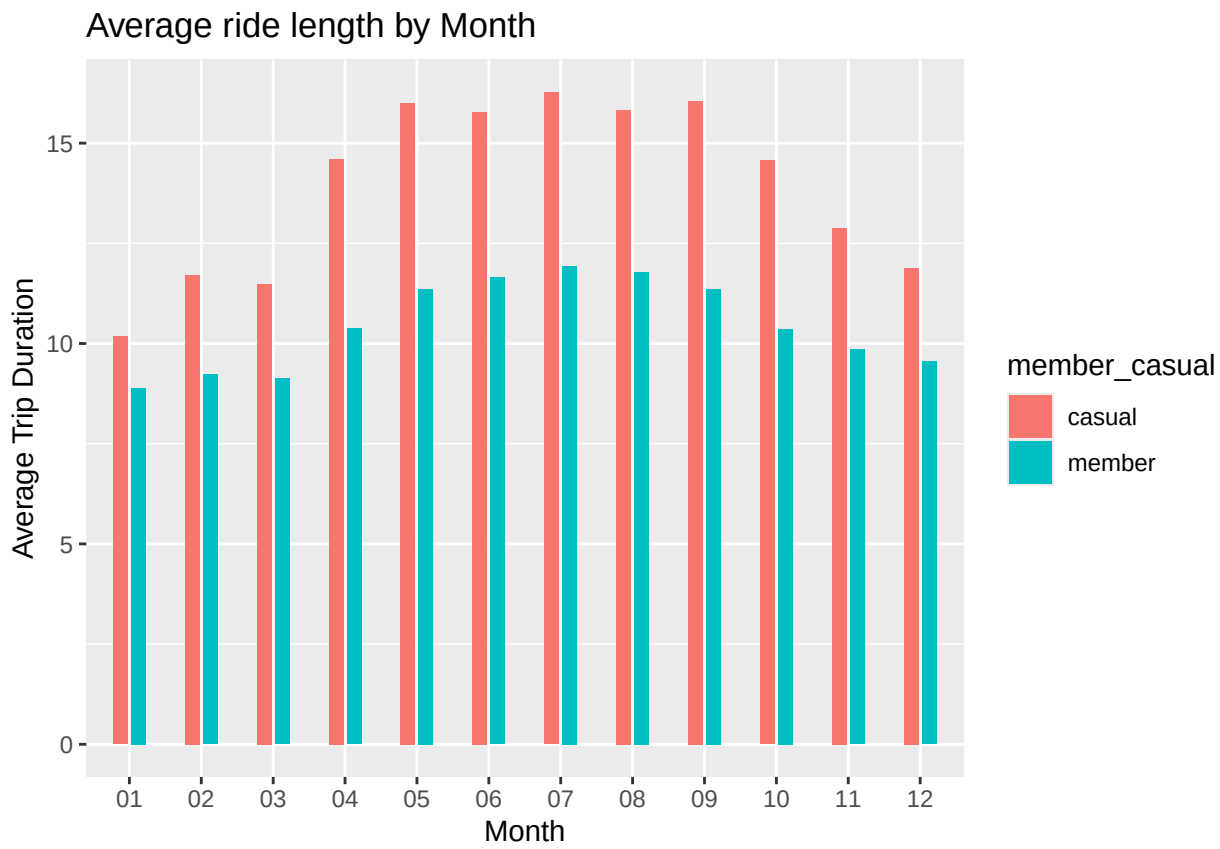
Total ride count by day of the week

```
trip_data %>%
  group_by(member_casual, day_of_week) %>%
  summarise(number_of_rides = n(), .groups = "drop") %>%
  ggplot(aes(x = day_of_week, y = format(number_of_rides, scientific = FALSE), fill = member_casual)) +
  geom_col(width = 0.4, position = position_dodge(width = 0.5)) +
  labs(title = "Total Ride Count by day of the week", x = 'Day of Week', y = 'Total Rides')
```



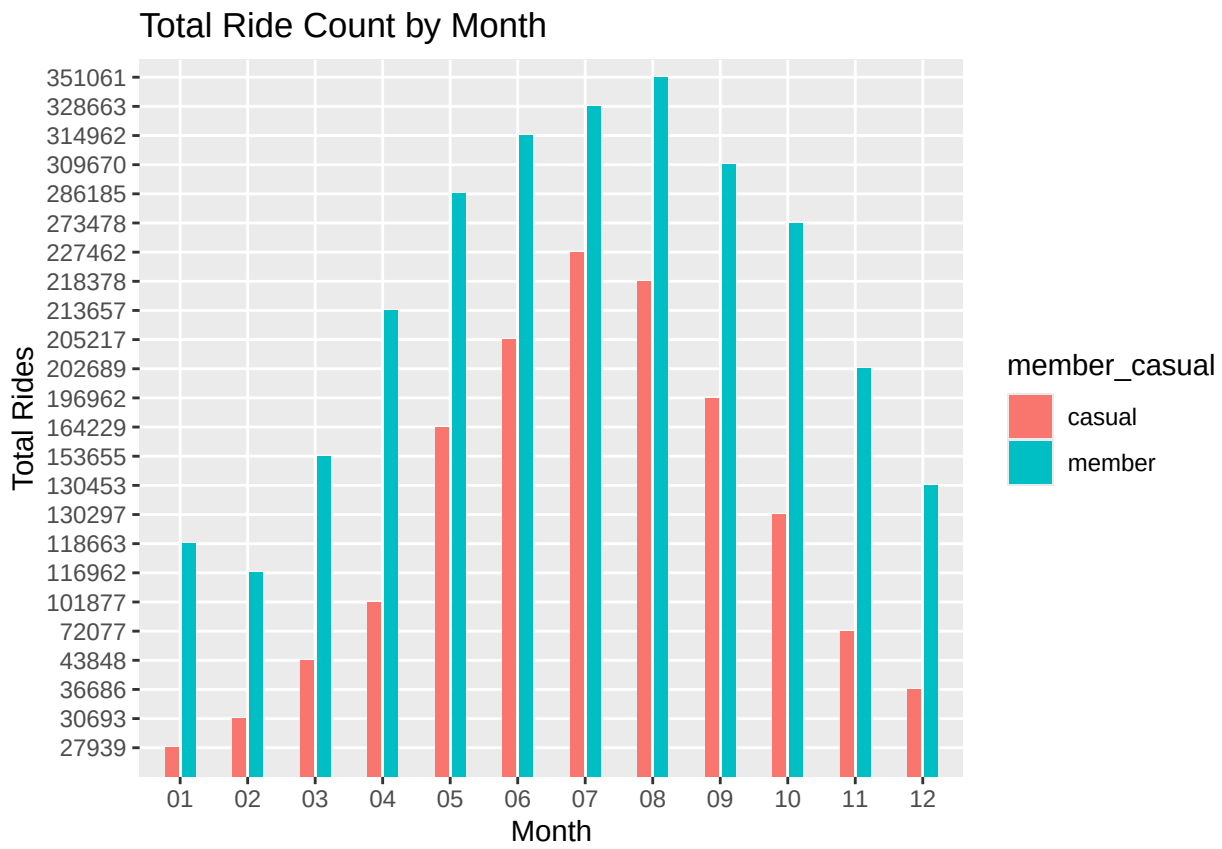
Average ride length by month

```
trip_data %>%
  group_by(member_casual, month) %>%
  summarise(average_duration = mean(minutes), .groups = "drop") %>%
  ggplot(aes(x = month, y = average_duration, fill = member_casual)) +
  geom_col(width = 0.4, position = position_dodge(width = 0.5)) +
  labs(title = "Average ride length by Month", x = 'Month', y = 'Average Trip Duration')
```



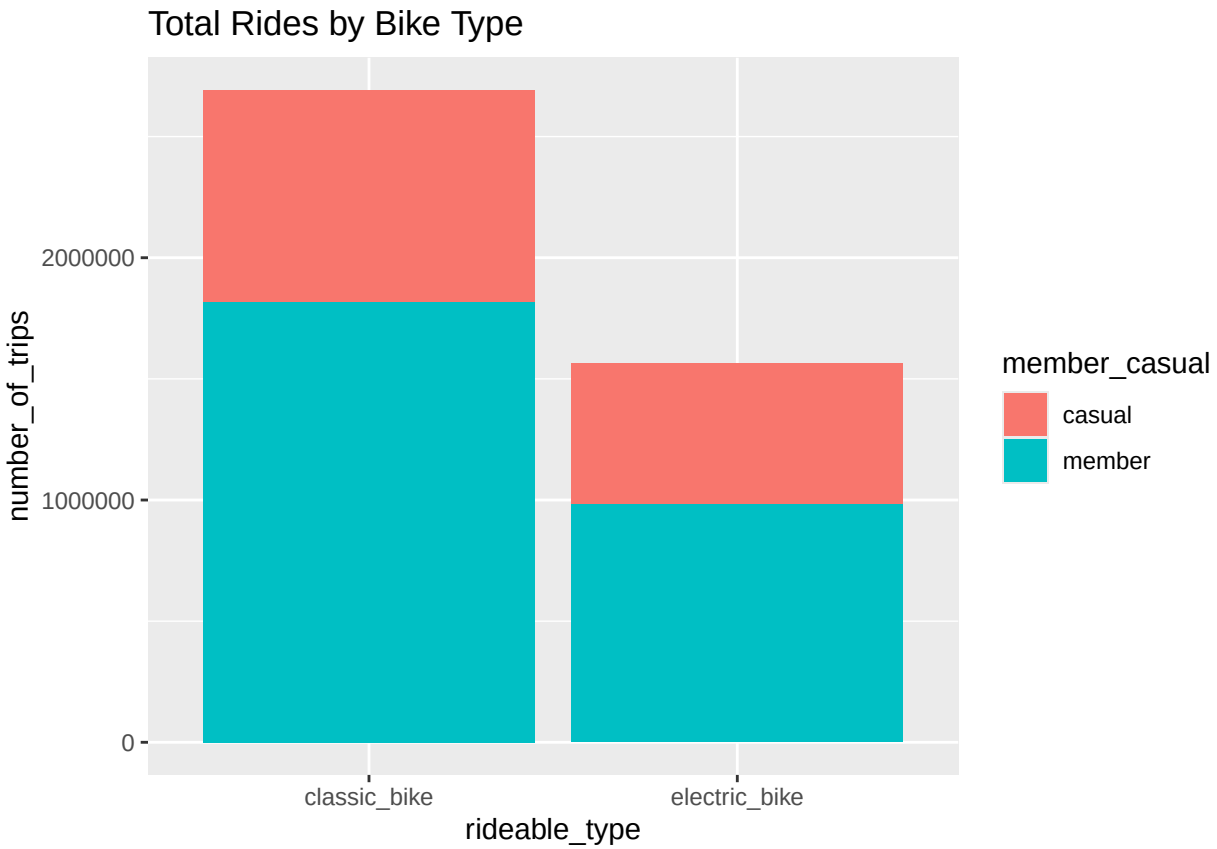
Total ride count by month

```
trip_data %>%
  group_by(member_casual, month) %>%
  summarise(number_of_rides = n(), .groups = "drop") %>%
  ggplot(aes(x = month, y = format(number_of_rides, scientific = FALSE), fill = member_casual)) +
  geom_col(width = 0.4, position = position_dodge(width = 0.5)) +
  labs(title = "Total Ride Count by Month", x = 'Month', y = 'Total Rides')
```



Preferred Bike Type Comparison

```
trip_data %>%
  group_by(rideable_type, member_casual)%>%
  summarize(number_of_trips = n(), .groups = "drop")%>%
  ggplot(aes(x = rideable_type, y = number_of_trips, fill = member_casual)) +
  geom_bar(stat = 'identity') +
  labs(title = "Total Rides by Bike Type") +
  scale_y_continuous(labels = function(x) format(x, scientific = FALSE))
```



Conclusions

Differences between the bike usage of annual members and casual riders:

- Casual members rent bikes more on weekends than on the weekdays.
- Casual members ride for longer but rent less often than annual members.
- Bike usage increases in the summer months and decreases in the colder months for both types of members.
- Classic bikes are much more popular than electric bikes, however a much larger percentage of annual membership riders use the electric bike than casual riders.

Why would casual riders buy Cyclistic annual memberships?

- Casual riders prefer to utilize the bikes on weekends rather than the weekdays. If a lifestyle change occurred to instead prefer the weekdays, the casual members may get an annual membership.

How can Cyclistic use digital media to influence casual riders to become members?

- Promote health and environmental benefits of biking.
- Seasonal membership could lead to an increase of casual members buying it, due to the large increase in the summer months.
- Market and focus on increasing the usage of electric bikes by casual members.