Coursera_Case_Study

Justin Yee

2024-09-04

#Introduction ##Welcome to the Cyclistic bike-share analysis case study! In this case study, you work for a fictional company, Cyclistic, along with some key team members. In order to answer the business questions, follow the steps of the data analysis process: Ask, Prepare, Process, Analyze, Share, and Act. Along the way, the Case Study Roadmap tables including guiding questions and key tasks — will help you stay on the right path.

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?

```
#Load Packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(janitor)
```

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

```
library(lubridate)
library(ggplot2)
library(readr)
```

```
#Importing the CSV files
jan2023 = read_csv("2023TripData/202301-divvy-tripdata.csv")
```

```
## Rows: 190301 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
feb2023 = read_csv("2023TripData/202302-divvy-tripdata.csv")
```

```
## Rows: 190445 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
mar2023 = read_csv("2023TripData/202303-divvy-tripdata.csv")
```

```
## Rows: 258678 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
apr2023 = read_csv("2023TripData/202304-divvy-tripdata.csv")
```

```
## Rows: 426590 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
may2023 = read_csv("2023TripData/202305-divvy-tripdata.csv")
```

```
## Rows: 604827 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
```

```
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
jun2023 = read_csv("2023TripData/202306-divvy-tripdata.csv")
```

```
## Rows: 719618 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
jul2023 = read_csv("2023TripData/202307-divvy-tripdata.csv")
```

```
## Rows: 767650 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
aug2023 = read_csv("2023TripData/202308-divvy-tripdata.csv")
```

```
## Rows: 771693 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
sep2023 = read_csv("2023TripData/202309-divvy-tripdata.csv")
```

```
## Rows: 666371 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
```

```
##
## 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.
```

```
oct2023 = read_csv("2023TripData/202310-divvy-tripdata.csv")
```

```
## Rows: 537113 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
nov2023 = read_csv("2023TripData/202311-divvy-tripdata.csv")
```

```
## Rows: 362518 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
dec2023 = read_csv("2023TripData/202312-divvy-tripdata.csv")
```

```
## Rows: 224073 Columns: 13
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): ride_id, rideable_type, start_station_name, start_station_id, end_...
## dbl  (4): start_lat, start_lng, end_lat, end_lng
## dttm (2): started_at, ended_at
##
## 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.
```

```
#Getting a preview of the files structure
head(jan2023)
head(feb2023)
head(mar2023)
head(apr2023)
head(may2023)
head(jun2023)
head(jul2023)
head(aug2023)
head(sep2023)
head(oct2023)
head(nov2023)
head(dec2023)
```

```r
#Since they all have the same column names, I can combine them into one date frame
df_2023 = bind_rows(jan2023, feb2023, mar2023, apr2023, may2023, jun2023, jul2023, aug2023, sep2023, oc
#clean up the name of the columns
df_2023 = clean_names(df_2023)
#To check if there are any null values in the data frame
sum(is.na(df_2023))
```

```
## [1] 3624089
```

```r
#Since there are null values and the data is large enough, we would remove them
df_2023 = na.omit(df_2023)
#Double check to make sure there are no more null values
sum(is.na(df_2023))
```

```
## [1] 0
```

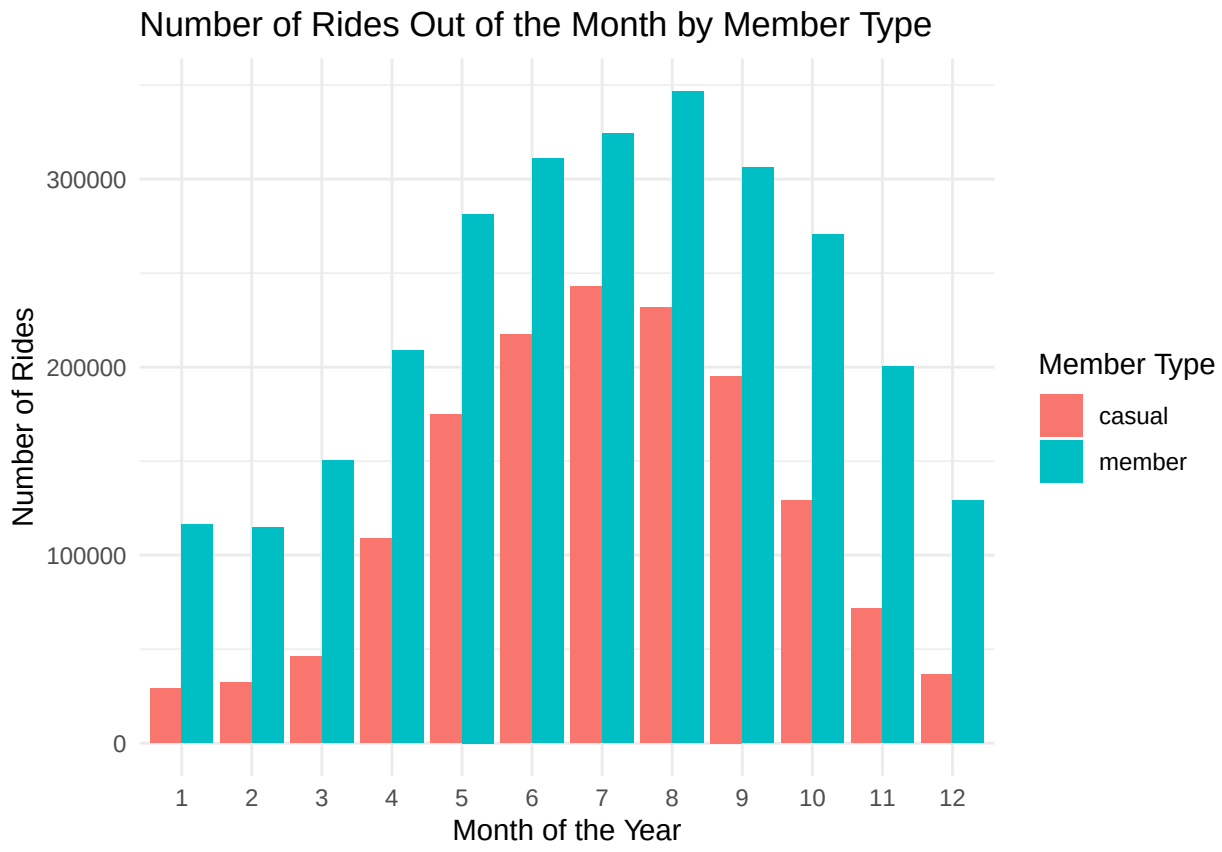
```r
#Now all the data that is left are useful data for us
```

```r
#Adding new columns for analysis usage later
#Also filtering all data where ride time is less than 30 seconds since it may be #invalid data
#Some rows failed to parse due to inconsistent format so I'll remove them
df_2023 = df_2023 %>%
  mutate(ride_time_secs = difftime(ended_at, started_at, units = "secs")) %>%
  filter(ride_time_secs > 30) %>%
  mutate(month = month(ymd_hms(started_at))) %>%
  na.omit(df_2023) %>%
  mutate(day_of_week = wday(started_at, label = TRUE, abbr = FALSE)) %>%
  mutate(hours_of_day = hour(ymd_hms(started_at))) %>%
  mutate(ride_time_minutes = as.numeric(gsub(" secs", "", ride_time_secs)) / 60)
head(df_2023)
```

```
## # A tibble: 6 x 18
##   ride_id          rideable_type started_at          ended_at
##   <chr>            <chr>         <dttm>              <dttm>
## 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
## # i 14 more variables: start_station_name <chr>, start_station_id <chr>,
## #   end_station_name <chr>, end_station_id <chr>, start_lat <dbl>,
## #   start_lng <dbl>, end_lat <dbl>, end_lng <dbl>, member_casual <chr>,
## #   ride_time_secs <drtn>, month <dbl>, day_of_week <ord>, hours_of_day <int>,
## #   ride_time_minutes <dbl>
```

```r
#Visualizations
#To avoid scientific notations
options(scipen = 999)
#For number of rides out of the month by member type
ride_counts <- df_2023 %>%
```

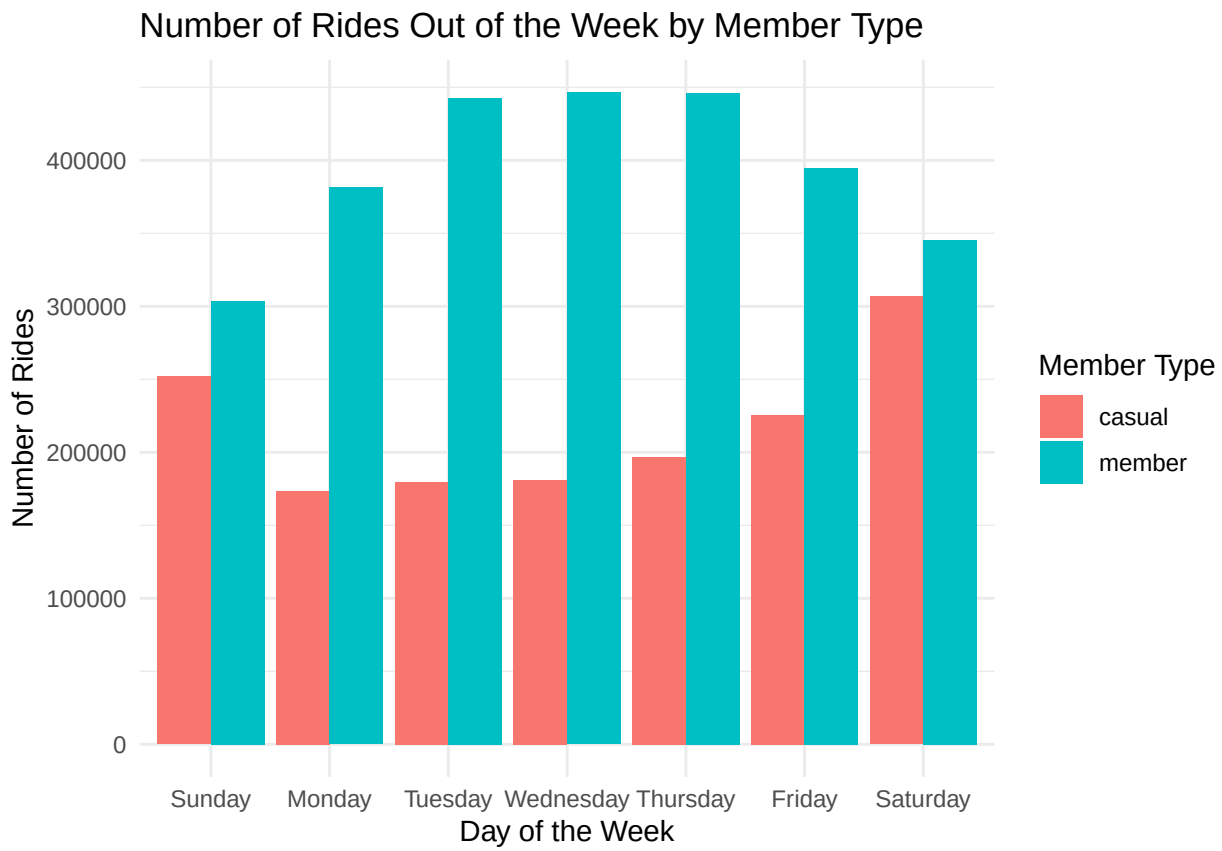
```
  group_by(month, member_casual) %>%
  summarise(number_of_rides = n(), .groups = "drop") %>%
  mutate(month = factor(month, levels = 1:12))


ggplot(data = ride_counts, aes(x = month, y = number_of_rides, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Number of Rides Out of the Month by Member Type",
    x = "Month of the Year",
    y = "Number of Rides",
    fill = "Member Type"
  ) +
  theme_minimal()
```



In this chart, we can see a significant difference between the number of rides in each membership type. Members typically ride more than casual cyclists. This is most likely because members uses bikes as their main method of transportation. We also see an increase in rides during Summer times which may be caused by an potential increase of cars on the road (Students on break, better weather, etc) that leads to more cyclists. Similarly, we see a decrease in rides during Fall and Winter months. This could be caused by shorter daylight hours and colder temperatures.

```
#For number of rides out of the week by member type
ride_counts <- df_2023 %>%
  group_by(day_of_week, member_casual) %>%
  summarise(number_of_rides = n(), .groups = "drop")
```

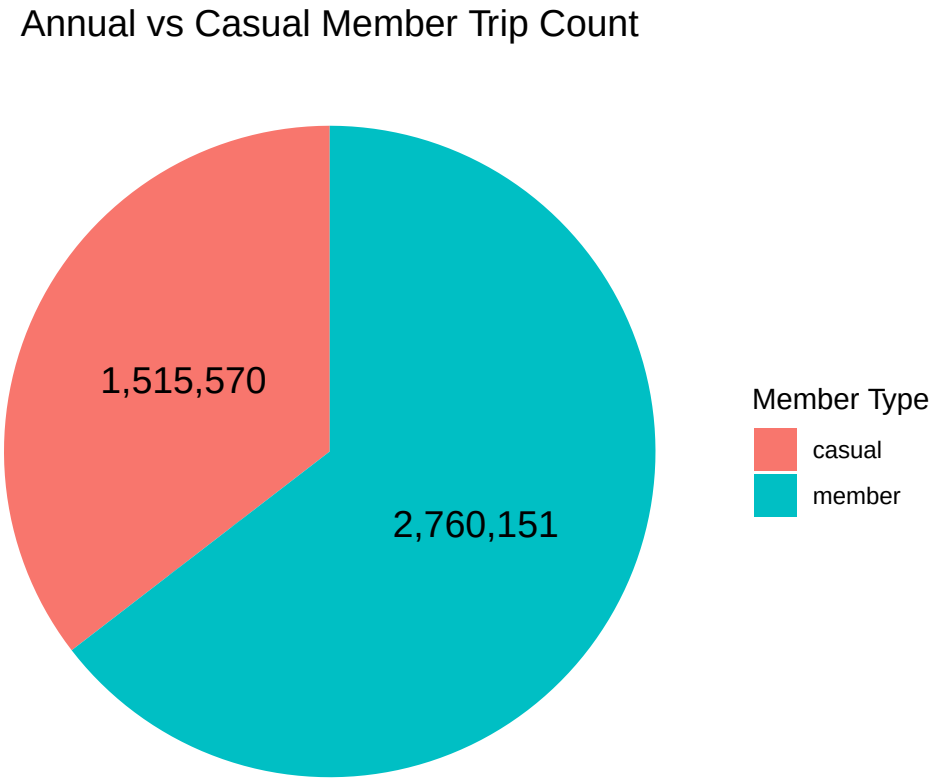
```r
ggplot(data = ride_counts, aes(x = day_of_week, y = number_of_rides, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Number of Rides Out of the Week by Member Type",
    x = "Day of the Week",
    y = "Number of Rides",
    fill = "Member Type"
  ) +
  theme_minimal()
```



This chart gives us information about cyclists' riding pattern during the days of the week. We can see that members consistently take more rides than casual riders, especially on weekday which can also be explained by members use bikes as their main transportation method. This chart also shows that casual riders are more active on the weekends, which may suggest casual cyclists uses bike for recreational purposes.

```r
count_casual_members = df_2023 %>%
  group_by(member_casual) %>%
  summarize(count = n())

ggplot(count_casual_members, aes(x = "", y = count, fill = member_casual)) +
  geom_bar(width = 1, stat = "identity") +
  coord_polar(theta = "y") +  # Convert bar chart to a pie chart
  geom_text(aes(label = scales::comma(count)),
            position = position_stack(vjust = 0.5), size = 5) +  # Add labels to the pie chart
  labs(title = "Annual vs Casual Member Trip Count",
```

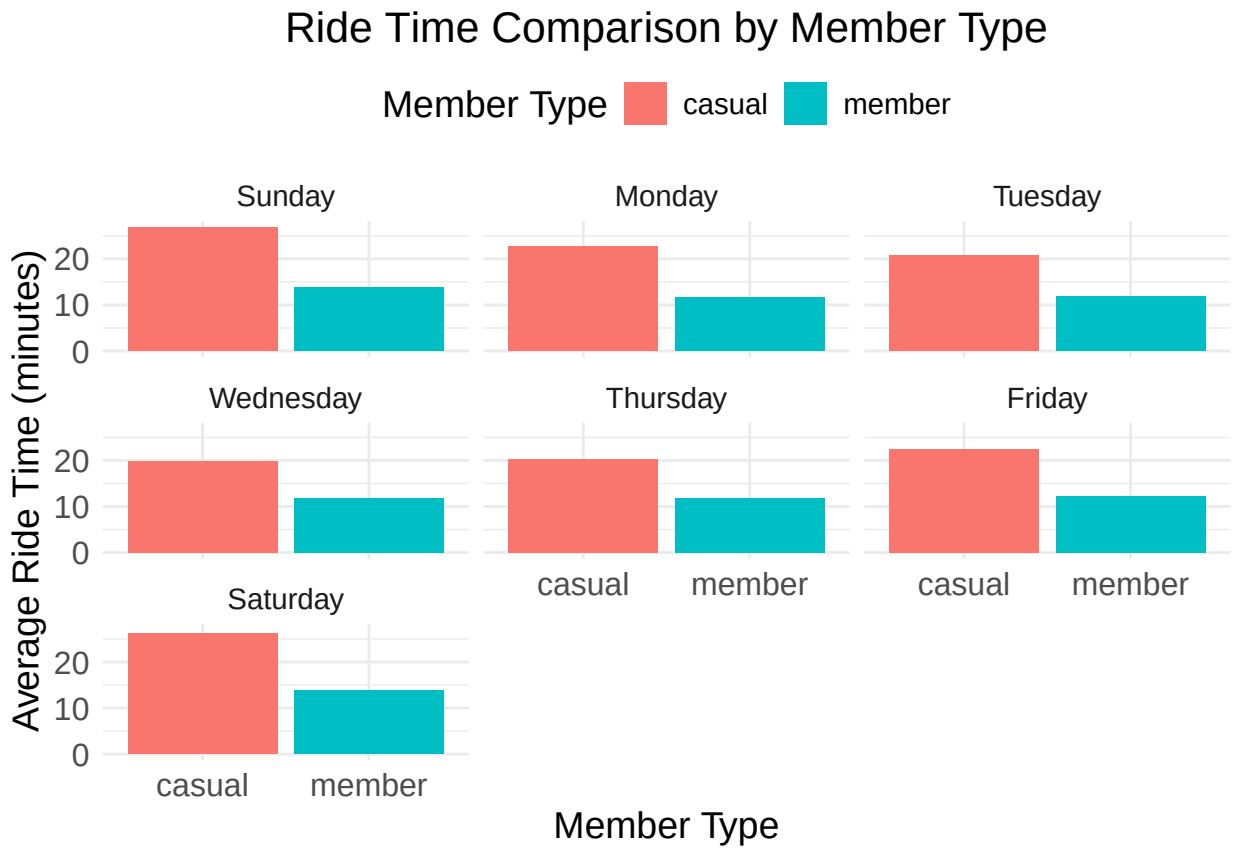
```
      fill = "Member Type") +
  theme_void() +  # Remove background and axis
  theme(
    plot.title = element_text(hjust = 0.5, size = 14),  # Center the title
    legend.position = "right"  # Position the legend
  )
```



This chart tells me that members takes significantly more trips than casual cyclists. This could possibly be explained by members uses bikes as their main transportation methods and that causal cyclists uses bikes more as a leisure activities. It also might be caused by membership incentives that reduces costs of each trip which makes members more inclined to use bikes to travel.

```
avg_trip_duration <- df_2023 %>%
  mutate(ride_time_mins = as.numeric(ride_time_secs) / 60) %>%  # Convert seconds to minutes
  group_by(day_of_week, member_casual) %>%
  summarise(avg_ride_time = mean(ride_time_mins, na.rm = TRUE), .groups = "drop")

# Plot
ggplot(avg_trip_duration, aes(x = member_casual, y = avg_ride_time, fill = member_casual)) +
  geom_bar(stat = "identity", position = "dodge") +
  facet_wrap(~ day_of_week) +  # Facet by day of the week
  labs(
    title = "Ride Time Comparison by Member Type",
    x = "Member Type",
    y = "Average Ride Time (minutes)",
    fill = "Member Type"
```

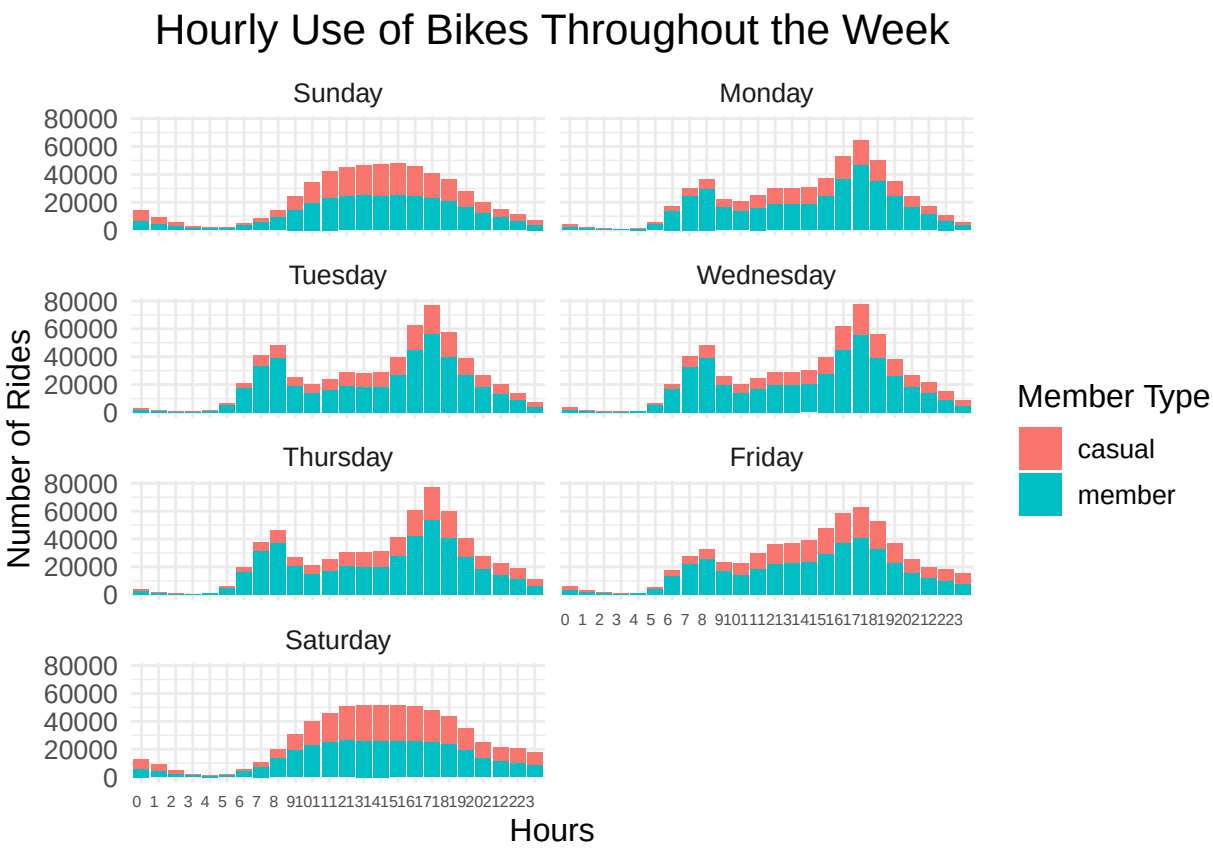
```
  ) +
  theme_minimal() +
  theme(
    text = element_text(size = 14),
    plot.title = element_text(hjust = 0.5, size = 16),  # Center the title and increase size
    axis.text.x = element_text(size = 12),
    axis.text.y = element_text(size = 12),
    legend.position = "top"  # Move the legend to the top
  )
```



Across all days of the week, casual riders (in red) have longer average ride times compared to members (in teal). On average, casual riders take rides lasting 15-25 minutes, while members typically have shorter rides, around 10-15 minutes. The fact that casual riders consistently have longer average ride times, especially on weekends, suggests that they use the service more for leisurely activities like sightseeing, weekend outings, or exercise. These activities typically involve longer ride duration. In contrast, members likely use the service for commuting or short trips, which explains their shorter and more consistent ride times across the week. Members might be taking shorter trips because their memberships offer unlimited or discounted rides. This may incentives members to use bikes for multiple short trips rather than one long trip. Casual riders, who pay per ride, may prefer to get the most out of each trip, resulting in longer ride duration.

```
#For number of rides out of the week by member type
ride_counts <- df_2023 %>%
  group_by(day_of_week, hours_of_day, member_casual) %>%
  summarise(number_of_rides = n(), .groups = "drop") %>%
  mutate(day_of_week = factor(day_of_week,
                              levels = c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday
```

```r
# Create the plot
ggplot(data = ride_counts, aes(x = factor(hours_of_day), y = number_of_rides, fill = member_casual)) +
  geom_bar(stat = "identity", position = "stack") +
  facet_wrap(~ day_of_week, nrow = 4) +  # Creates separate plots for each day of the week
  labs(
    title = "Hourly Use of Bikes Throughout the Week",
    x = "Hours",
    y = "Number of Rides",
    fill = "Member Type"
  ) +
  theme_minimal() +
  theme(
    text = element_text(size = 12),
    plot.title = element_text(hjust = 0.5, size = 16),  # Center and enlarge title
    axis.text.x = element_text(size = 6, hjust = 1),  # Rotate x-axis labels
    legend.position = "right"  # Move the legend to the right
  )
```



On weekdays (Monday to Friday), there is a clear peak in member rides during the morning (around 7-9 AM) and evening (4-6 PM). This pattern suggests that members are using the bikes for commuting to and from work or school.On weekends, the distribution of member rides is more spread out across the day, indicating more leisure-based or flexible usage compared to the structured weekday commuting pattern. Across both groups, bike usage drops dramatically in the late evening and early morning hours (from 9 PM to 6 AM), suggesting limited demand for bike sharing during nighttime hours.