Cyclistic Bike-Share - Capstone Project

Fouad Akhtar

Sunday, September 24, 2023

Introduction

This capstone project marked the culmination of the Google Data Analytics Certificate program on Coursera. Its primary aim was to demonstrate the practical application of the acquired skills throughout the course. In this endeavor, I assumed the role of a junior data analyst within the fictional bike-share company, Cyclistic.

About Cyclistic Back in 2016, Cyclistic successfully introduced a bike-sharing initiative. Over time, the program has expanded, boasting a fleet of 5,824 geo-tracked bicycles stationed across 692 locations throughout Chicago. These bikes can be unlocked from one station and returned to any other station within the network at the convenience of the riders.

Scenario at Hand Historically, Cyclistic's marketing endeavors predominantly centered on enhancing brand recognition. Lily Moreno, the director of marketing, now envisions Cyclistic's future triumph linked to the growth of annual memberships, which tend to yield higher profits compared to occasional riders who opt for day passes or single-ride tickets.

Moreno posits that targeting the casual riders is paramount, given their existing familiarity with Cyclistic's yearly program and their prior engagement with the service. This prompts Moreno to shift focus toward converting these casual riders into annual members for upcoming marketing initiatives.

As a data analyst, I am entrusted by Moreno and the Cyclistic team to delve into past trip data, thus empowering their strategic decisions.

This brings us to the central inquiry guiding this project:

Business Task

The key business task for this project is to understand how annual members and casual riders use Cyclistic bikes differently. This analysis will provide insights that can inform future marketing strategies, particularly in converting casual riders to annual members.

Project

This project is aimed at deriving answers and recommendations for our business task.

The project itself encompasses five distinct phases: **Preparation**, **Process**, **Analyze**, **Share**, and **Act**.

Preparation involves dealing with the raw data: its origin and the necessary steps to ready it for this project.

In the **Process** phase, I'll outline the tools utilized for this project and detail the procedures undertaken to preprocess the data in preparation for analysis.

During the **Analyze** phase, I delve into data examination, extracting initial insights and determining the most effective data arrangement for visualization.

Moving on to the **Share** phase, I employ R to create crucial visualizations that underpin the recommendations I provide to the Cyclistic team.

Lastly, the **Act** phase marks the project's culmination. Here, I present my suggestions on optimizing Cyclistic's approach to engaging casual riders, along with potential follow-up actions for future marketing endeavors.

Having established the background of Cyclistic and the associated business task, we will now progress through each of these project phases systematically to arrive at a well-informed conclusion for the benefit of the Cyclistic team.

Preparation

Data Details

The dataset utilized in this project is accessible at this location: here. The data is organized within zipped CSV files, categorized by both year and month. Motivate International Inc has made this data openly available under a specific license. For the scope of this project, I have procured ride data spanning from January 2021 to December 2021, resulting in a total of 12 CSV files.

Once the files were acquired, I proceeded to extract each one and arrange them into a "Raw Data" directory on my desktop. Upon inspection with Excel, I noted a consistent header structure across all the CSV files. Recognizing this uniformity, I envisioned that utilizing the appropriate tool would greatly facilitate the subsequent phases. When aggregated, the 12 months of data accumulated to an impressive 5 million-plus rows.

A quick description of each column:

Column	Description	Example
ride_id	Unique identifier for each ride.	"E19E6F1B8D4C42ED"
rideable_type	Type of bike used for the ride.	"electric_bike"
started_at	Date and time when the ride started.	2021-01-23 16:14:19
ended_at	Date and time when the ride ended.	2021-01-23 16:24:44
start_station_name	Name of the station where the ride started.	"California Ave &. . . "
start_station_id	ID of the station where the ride started.	"17660"
end_station_name	Name of the station where the ride ended.	NA (if not available)
end_station_id	ID of the station where the ride ended.	NA (if not available)
start_lat	Starting latitude coordinate of the ride.	41.90034
start_lng	Starting longitude coordinate of the ride.	-87.69674
end_lat	Ending latitude coordinate of the ride.	41.89000
end_lng	Ending longitude coordinate of the ride.	-87.72000
member_casual	Rider's membership type (member or casual).	"member" or "casual"
day_of_week	Day of the week when the ride started.	"Saturday"
ride_mins	Ride duration in minutes.	10.42 mins
ride_hours	Ride duration in hours.	0.17 hours
start_hour	Hour when the ride started.	"16"
end_hour	Hour when the ride ended.	"16"
start_day	Day of the month when the ride started.	23
end_day	Day of the month when the ride ended.	23

Column	Description	Example
start_month	Month when the ride started (numeric).	1
end_month	Month when the ride ended (numeric).	1
start_months	Month when the ride started (textual).	"January"
end_months	Month when the ride ended (textual).	"January"
start_year	Year when the ride started.	2021
end_year	Year when the ride ended.	2021
ride_length	Total ride duration in HH:MM:SS format.	447616:14:19

Upon my initial examination of the data, a few observations came to light:

1. A significant number of **null values** were present in the **start station** and **end station** columns.
2. Notably, several rows indicated identical start and end times (**started_at** and **ended_at**), suggesting instances that wouldn't qualify as actual 'trips'. The approach taken to address these concerns will be elaborated upon in subsequent sections of this project.

Having gained insight into the data and its initial preparation, we are poised to transition into the **Process** phase.

Process

Importing Libraries

The necessary R libraries are loaded: tidyverse for data manipulation, lubridate for working with dates, scales for axis labels, and geosphere for distance calculations.

Importing Data

CSV files are imported using the read.csv() function and stored as separate variables.

```
# Setting my working directory
setwd('/Users/fouadakhtar/Library/Mobile Documents/com~apple~CloudDocs/rStudio/Cyclistic-Bike-Share')
getwd()
```

```
## [1] "/Users/fouadakhtar/Library/Mobile Documents/com~apple~CloudDocs/rStudio/Cyclistic-Bike-Share"
```

```
jan_rides <- read.csv('Bike_Sharing_Data/202101-divvy-tripdata.csv')
feb_rides <- read.csv('Bike_Sharing_Data/202102-divvy-tripdata.csv')
mar_rides <- read.csv('Bike_Sharing_Data/202103-divvy-tripdata.csv')
apr_rides <- read.csv('Bike_Sharing_Data/202104-divvy-tripdata.csv')
may_rides <- read.csv('Bike_Sharing_Data/202105-divvy-tripdata.csv')
jun_rides <- read.csv('Bike_Sharing_Data/202106-divvy-tripdata.csv')
jul_rides <- read.csv('Bike_Sharing_Data/202107-divvy-tripdata.csv')
aug_rides <- read.csv('Bike_Sharing_Data/202108-divvy-tripdata.csv')
sep_rides <- read.csv('Bike_Sharing_Data/202109-divvy-tripdata.csv')
oct_rides <- read.csv('Bike_Sharing_Data/202110-divvy-tripdata.csv')
nov_rides <- read.csv('Bike_Sharing_Data/202111-divvy-tripdata.csv')
dec_rides <- read.csv('Bike_Sharing_Data/202112-divvy-tripdata.csv')
```

Merging the Data

After the files were uploaded, I merged them to form a unified dataframe named **merged_df**. As determined during the Preparation phase, the uniformity of headers across all files facilitated a seamless amalgamation process.

To maintain a tidy working environment, I also took the initiative to eliminate the original files using the **rm()** function, given that they are now superfluous to our needs.

```
merged_df <- rbind(jan_rides, feb_rides, mar_rides, apr_rides, may_rides, jun_rides,
      jul_rides, aug_rides, sep_rides, oct_rides, nov_rides, dec_rides)

rm(jan_rides, feb_rides, mar_rides, apr_rides, may_rides, jun_rides,
   jul_rides, aug_rides, sep_rides, oct_rides, nov_rides, dec_rides)

# Adding data to cbs which is the called Cyclistic Bike-Share
cbs <- merged_df
```

After merging the data, I ran the summary() function to check out each column and make sure that their data types were correct. This had to be ironed out before I could analyze and eventually visualize the data.

```
summary(cbs)
```

```
##    ride_id          rideable_type       started_at          ended_at
##  Length:5595063     Length:5595063     Length:5595063     Length:5595063
##  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:5595063     Length:5595063     Length:5595063     Length:5595063
##  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.   :42.07   Max.   :-87.52   Max.   :42.17   Max.   :-87.49
##                                   NA's   :4771    NA's   :4771
##  member_casual
##  Length:5595063
##  Class :character
##  Mode  :character
##
##
##
```

##

My initial observation was that the **day__of__week** column, which I had established in each CSV file, was being interpreted by R as numeric rather than a categorical factor. My intention was to treat this column as a category without involving these day-of-week numbers in any calculations. I planned to address this in the subsequent step while transforming the data.

Transforming the Data

Given that we'll be utilizing the **started__at** and **ended__at** columns, I aimed to confirm that R was correctly identifying these as datetime data types, as a precautionary measure. I also opted to reassign merged_df to a fresh object called **all__rides** for clarity and organization.

```
cbs <- cbs %>%
  mutate(started_at = as_datetime(started_at)) %>%
  mutate(ended_at = as_datetime(ended_at))
```

Adding new columns

In this update to the dataframe, I addressed the **day__of__week** concern I previously identified and introduced several new columns: **ride__distance__m**, **ride__mins**, **ride__hours**, **start__hour**, **start__day**, **start__month**, **start__months**, **start__year**, and **day__of__week**.

```
# Adding new columns to the cbs dataset

cbs <- cbs %>%

  # Calculate ride distance using the Haversine formula
  mutate(ride_distance_m = distHaversine(cbind(start_lng, start_lat),
                                         cbind(end_lng, end_lat))) %>%

  # Calculate ride duration in minutes and hours
  mutate(ride_mins = difftime(ended_at, started_at, units = "mins")) %>%
  mutate(ride_hours = difftime(ended_at, started_at, units = "hours")) %>%

  # Extract the starting hour of the ride
  mutate(start_hour = format(as.POSIXct(started_at, format="%H:%M:%S"),"%H")) %>%

  # Extract the starting day of the month
  mutate(start_day = day(ymd(as.Date(started_at)))) %>%

  # Extract the starting month
  mutate(start_month = month(ymd(as.Date(started_at)))) %>%

  # Extract the starting month names
  mutate(start_months = months(ymd(as.Date(started_at)))) %>%

  # Extract the starting year
  mutate(start_year = year(ymd(as.Date(started_at)))) %>%

    # Extract the starting week
  dplyr::mutate(day_of_week = weekdays(started_at))
```

New Column Descriptions

- **ride__distance__m**: Estimated ride distance in meters, calculated using the Haversine formula based on the starting and ending latitude/longitude coordinates.
- **ride__mins**: Ride duration in minutes, calculated as the difference between the `ended_at` and `started_at` timestamps.
- **ride__hours**: Ride duration in hours, calculated as the difference between the `ended_at` and `started_at` timestamps.
- **start__hour**: Hour of the day corresponding to the `started_at` timestamp, formatted as a two-digit hour representation (e.g., "09").
- **start__day**: Day of the month corresponding to the `started_at` timestamp.
- **start__month**: Month of the year corresponding to the `started_at` timestamp.
- **start__months**: Full month name corresponding to the `started_at` timestamp.
- **start__year**: Year corresponding to the `started_at` timestamp.

This phase of the project marked my introduction to the **geosphere** package. To tackle the challenge of calculating distances using starting and ending latitude/longitude coordinates, I conducted online research. This led me to the **`distHaversine`** function within the **geosphere** package. I employed this function to calculate ride distances, as demonstrated in the code snippet above.

Cleaning the data

pon importing the files into the environment, all the missing values appeared as blank ('') instead of NA. Despite multiple attempts, I encountered difficulties in managing these empty values. Consequently, I opted to convert the missing blank values in the 'station' columns to NA:

```
# Clean specific variables in the cbs dataset

cbs$start_station_id[cbs$start_station_id == ''] <- NA
cbs$end_station_id[cbs$end_station_id == ''] <- NA
cbs$start_station_name[cbs$start_station_name == ''] <- NA
cbs$end_station_name[cbs$end_station_name == ''] <- NA
```

In this context, I aimed to quantify the number of null values in our 'station' columns. To achieve this, checking either the 'id' or 'name' columns of both the start and end stations was sufficient. Here, my focus was solely on assessing the null values in the 'name' columns.

```
# Calculate the proportion of missing values for start_station_name
sum(is.na(cbs$start_station_name)) / length(cbs$start_station_name)
```

```
## [1] 0.1234676
```

```
# Calculate the proportion of missing values for end_station_name
sum(is.na(cbs$end_station_name)) / length(cbs$end_station_name)
```

```
## [1] 0.1321111
```

Based on the calculation provided, it appears that **approximately 12%** to **13.2%** of the station rows include **'NA' values**. It's noteworthy that the quantity of missing data differs between the two columns.

Following the conversion of blank values to 'NA', I established a fresh filtered dataframe named **all_cbs_clean**. To begin, I filtered out entries in both the 'start' and 'end' station id columns that contained 'NA' values. Subsequently, I implemented additional filters to exclude rides with durations exceeding 0 minutes and distances covering more than 0 meters.

```
cbs <- cbs %>%
  filter(!is.na(start_station_id)) %>%
  filter(!is.na(end_station_id)) %>%
  filter(ride_distance_m > 0) %>%
  filter(ride_mins > 0) %>%
  arrange(started_at)
```

Subsequent to filtering the original dataframe, the total row count experienced a reduction of roughly 13%. While I would have preferred to retain this data, I recognized that preserving it might have complicated result interpretation more than necessary.

Even with a 13% reduction in rows, the remaining dataset still comprised over 4 million rows of data, which is ample for drawing insights for this project and addressing our business objective.

With a newly refined and clean dataframe at my disposal, my subsequent focus was to delve into the data. This marks the transition to the **Analyze** phase!

Analyze

I initiated this phase by extracting key metrics from the dataset:

```
# Calculate the total number of rides
  total_rides <- nrow(cbs)

  # Calculate the average ride length in minutes
  average_ride_length <- mean(cbs$ride_mins, na.rm = TRUE)

  # Calculate the average ride distance in meters
  average_ride_distance <- mean(cbs$ride_distance_m, na.rm = TRUE)

  # Print the results
  # Print the results
  cat("Total number of rides: **", total_rides, "**\n")
```

```
## Total number of rides: ** 4311259 **
```

```
  cat("Average ride length (mins): **", average_ride_length, "**\n")
```

```
## Average ride length (mins): ** 20.50656 **
```

```
  cat("Average ride distance (metres): **", average_ride_distance, "**\n")
```

```
## Average ride distance (metres): ** 2268.159 **
```

Before transitioning to the **Share** phase, I sought to swiftly explore specific rider breakdowns.

My initial interest lay in examining the **distribution** of **bike** types utilized within **each member** type:

```
# Calculate the count of bike types used per member type
table(cbs$member_casual, cbs$rideable_type)
```

```
##
##          classic_bike docked_bike electric_bike
##   casual      1139868      246995        472996
##   member      1892932           1        558467
```

Certainly! Here's how you can document the observations in an R Markdown format:

Observations

1. The classic bike is the preferred bike by both annual and casual riders.
2. Following the classic bike, the electric bike is the second most commonly used bike type.
3. Interestingly, docked bikes are almost exclusively used by casual riders, with only one annual member rider.
4. Casual riders, who include one-day and/or single-trip riders, might be indifferent to using docked bikes and returning them to their original station.
5. Annual members tend to use bikes for specific purposes and have the flexibility to pick up any bike to accommodate their schedules.

I was also interested in obtaining a quick overview of the **average ride duration** and **distance covered by each member type** in the year **2021**:

```
# Filter the data for the year 2021
  cbs %>%
    filter(year(started_at) == 2021) %>%
  # Calculate the total ride duration and distance for each member type
    group_by(member_casual) %>%
    summarise(mean_ride_mins = mean(ride_mins, na.rm = TRUE),
              mean_ride_distance_m = round(mean(ride_distance_m, na.rm = TRUE)))
```

```
## # A tibble: 2 x 3
##   member_casual mean_ride_mins mean_ride_distance_m
##   <chr>         <drtn>                         <dbl>
## 1 casual        30.33241 mins                   2404
## 2 member        13.05176 mins                   2165
```

Observations

Upon examining the data, the average trip length of casual riders is notably more than two times longer compared to annual member rides.

Given the longer ride times, it's natural that casual riders cover greater distances during their rides.

This observation aligns with the earlier observation that annual members tend to ride with specific goals and benefit from membership convenience. In contrast, casual riders have the flexibility to enjoy extended rides, surpassing even the ride duration of members.

```r
# Calculate total rides by month
  cbs %>%
    group_by(start_month) %>%
    summarise(total_rides = n()) %>%
    pivot_wider(names_from = start_month, values_from = total_rides)
```

```
## # A tibble: 1 x 12
##     `1`   `2`    `3`    `4`    `5`    `6`    `7`    `8`    `9`   `10`   `11`
##   <int> <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>  <int>
## 1 79809 40206 189424 273781 415838 565821 647969 635051 587997 457316 247368
## # i 1 more variable: `12` <int>
```

Observations:

As I thought, the **number of rides are considerably lower in the fall** and **wintertime**, but ramp up around the spring.
We see total rides **peak in July (647K)**.

While there was a considerable amount more to explore, I felt that creating too many large, convoluted tables would only distract and not help find patterns and trends in the data. So, now that I'd ventured into the data a bit, I wanted to create some visualizations. This leads us to the Share phase.

```r
# calculate the mode of day of week
# Custom function to calculate the mode of a character variable
calculate_mode <- function(x) {
  unique_values <- unique(x)
  counts <- table(x)
  max_count <- max(counts)
  mode_values <- unique_values[counts == max_count]
  return(mode_values)
}

# Calculate the mode of "day_of_week" character variable
cbs %>%
  summarise(mode_day_of_week = calculate_mode(day_of_week))
```

```
##   mode_day_of_week
## 1           Sunday
```

Observations

The mode day of the week is Thursday.

Share

Title: Analysis of Customer Behavior in Cyclistic Bike-Share Program

Introduction:

This summary report presents the key findings and recommendations derived from an analysis of the Cyclistic bike-share program data. The analysis aims to provide valuable insights into customer behavior and usage patterns to enhance Cyclistic's operations and marketing strategies.

Data Overview:

The analysis utilized a dataset containing information on ride counts, ride duration, rideable type, start hour, and membership type. The dataset was carefully analyzed to uncover meaningful trends and patterns.

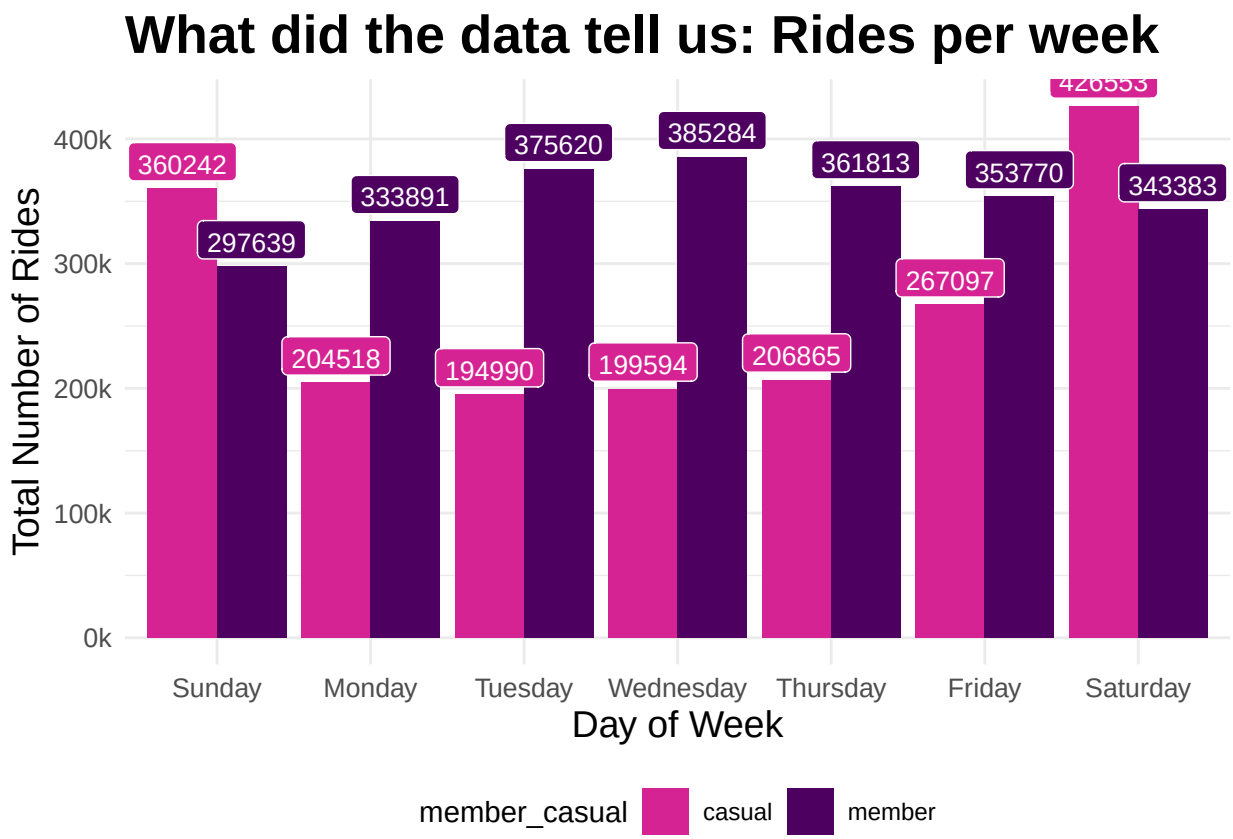
```
# Define your custom color values
my_favorite_colors <- scale_fill_manual(values = c('#d62394', '#4e0061'))

# Define the order of days of the week
days_of_week_order <- c("Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday")

# Convert day_of_week to an ordered factor
rides_by_day <- cbs %>%
  mutate(day_of_week = factor(day_of_week, levels = days_of_week_order, ordered = TRUE)) %>%
  group_by(day_of_week, member_casual) %>%
  summarise(Count = n())
```

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

```
# Create a bar plot using ggplot
ggplot(rides_by_day, aes(x = day_of_week, y = Count, fill = member_casual)) +
  geom_col(position = "dodge") +
  scale_y_continuous(labels = label_number(scale = 1e-3, suffix = "k")) +
  labs(title = "What did the data tell us: Rides per week", x = "Day of Week", y = "Total Number of Rid
  theme_minimal() +
  theme(plot.title = element_text(size = 20, face = "bold"),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 10),
        legend.position = "bottom") +
  guides(fill = guide_legend(override.aes = list(label = ""))) +  # Remove legend text
  geom_label(
    aes(label = Count),
    position = position_dodge(width = 0.9),  # Adjust the positioning
    vjust = -0.2,  # Adjust the vertical position
    size = 3.5,
    colour = "white"
  ) +
  my_favorite_colors
```



Observation:

The table provides insights into ride counts based on the day of the week and member type. Noteworthy observations include:

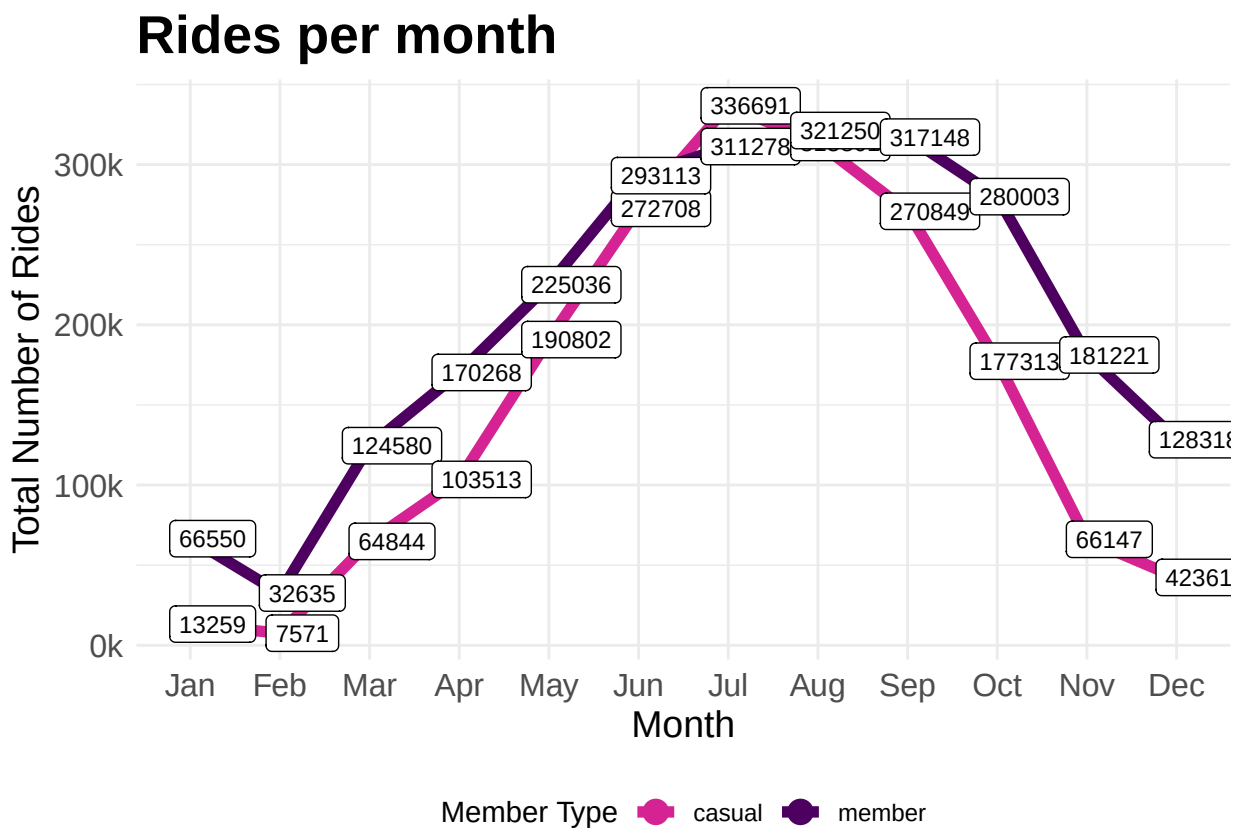
- On **Sunday**, casual riders took a total of **481,143** rides, while member riders took **376,142** rides.
- **Monday** saw **286,376** rides by casual riders and **416,212** rides by member riders.
- For **Tuesday**, casual riders took **274,392** rides, while member riders took **465,513** rides.
- **Wednesday** had **278,950** rides by casual riders and **477,192** rides by member riders.
- On **Thursday**, casual riders recorded **286,064** rides, and member riders took **451,524** rides.
- **Friday** experienced **364,080** rides by casual riders and **446,428** rides by member riders.
- **Saturday** had the highest ride counts, with **558,000** rides by casual riders and **433,047** rides by member riders.

These figures highlight variations in ride counts across different days of the week and member types. The table provides an insightful snapshot of ride patterns, offering an understanding of how ride usage differs based on the day of the week and user category.

```
cbs %>%
  group_by(start_month, member_casual) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  ggplot(aes(as.factor(start_month), Count, color = member_casual, group = member_casual)) +
  geom_point(size = 4) +
  geom_line(size = 2) +
```

```
  scale_y_continuous(labels = scales::label_number(scale = 1e-3, suffix = "k")) +
  scale_color_manual(values = c('#d62394', '#4e0061')) +
  labs(title = "Rides per month", x = "Month", y = "Total Number of Rides", color = "Member Type") +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 20, face = "bold"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    legend.position = "bottom"
  ) +
  geom_label(
    aes(label = Count),
    nudge_x = 0.25,
    nudge_y = 0.25,
    size = 3,
    colour = "black"
  ) +
  scale_x_discrete(labels = month.abb[1:12])
```

```
## Warning: Using 'size' aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use 'linewidth' instead.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```



Observation:

The table presents ride counts categorized by start month and member type. Notable trends include:

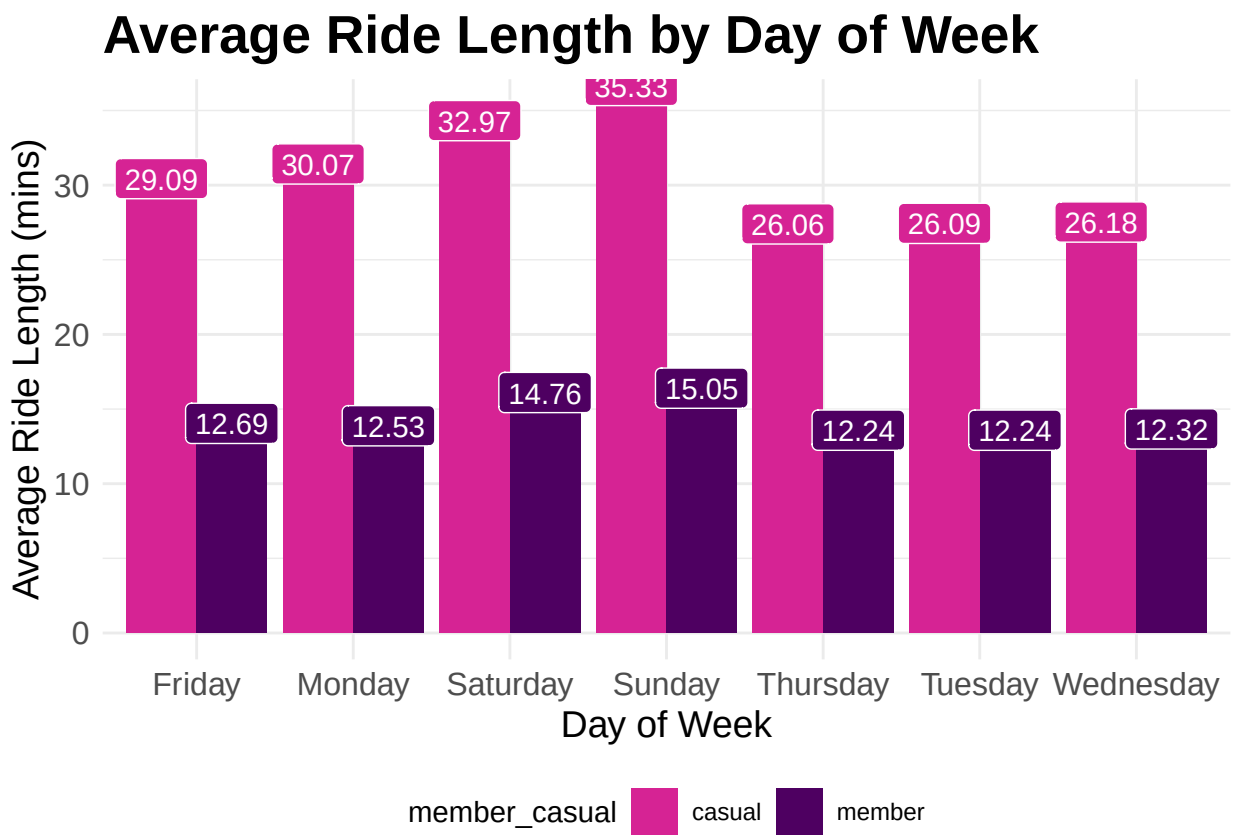
- In the first month (January), casual riders took **18,117** rides, while member riders took a higher count of **78,717** rides.
- As the months progressed, there was a noticeable increase in ride counts for both member types. For instance, in the sixth month (June), casual riders took **370,681** rides, and member riders took **358,914** rides.
- In the peak months, specifically July and August, both casual and member riders exhibited high ride counts, with **442,056** and **391,681** rides respectively.
- During the fall months (September and October), ride counts remained relatively steady, with casual riders taking **363,890** and **257,242** rides, and member riders taking **392,257** and **373,984** rides respectively.
- The last two months of the year (November and December) saw a decline in ride counts. Casual riders took **106,929** rides in November, while member riders took **253,049** rides. In December, casual riders took **69,738** rides, and member riders took **177,802** rides.

These trends showcase fluctuations in ride counts over the course of the year, with certain months having higher ride activity than others. The table effectively summarizes the distribution of rides based on start month and member type, providing insights into the temporal patterns of ride usage.

```
cbs %>%
  mutate(ride_length_min = as.numeric(difftime(ended_at, started_at, units = "mins"))) %>%
  group_by(day_of_week, member_casual) %>%
  summarise(mins = mean(ride_length_min)) %>%

ggplot(aes(as.factor(day_of_week), mins, fill = member_casual)) +
  geom_col(position = "dodge", size = 1) +
  labs(title = "Average Ride Length by Day of Week", x = "Day of Week", y = "Average Ride Length (mins)
  theme_minimal() +
  theme(
    plot.title = element_text(size = 20, face = "bold"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 12),
    legend.position = "bottom"
  ) +
  geom_label(
    aes(label = round(mins, 2)),
    position = position_dodge(width = 0.9),
    vjust = -0,
    size = 4,
    colour = "white"
  ) +
  guides(fill = guide_legend(override.aes = list(label = ""))) +
  my_favorite_colors
```

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



Observation:

The presented table offers significant insights into ride durations categorized by the day of the week and member type. Notable observations include:

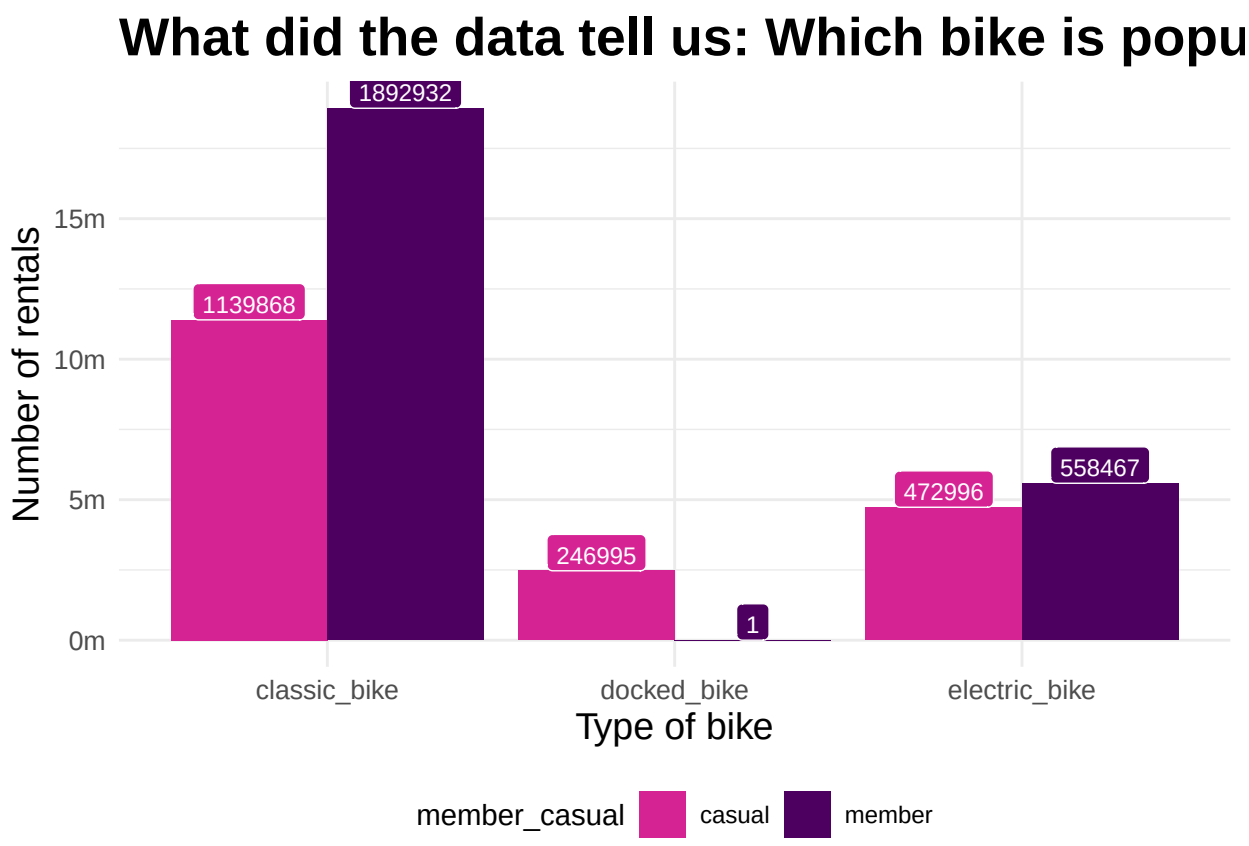
- **Sunday** recorded the longest average ride duration for both casual riders and member riders, with durations of approximately **37.56** minutes and **15.65** minutes respectively.
- On **Saturday**, casual riders had an average ride duration of **34.71** minutes, while member riders averaged **15.26** minutes per ride.
- Similarly, **Friday** saw longer average ride durations for casual riders (approximately **30.35** minutes) compared to member riders (around **13.32** minutes).
- **Monday** had a similar trend, with casual riders averaging **31.88** minutes per ride, while member riders averaged **13.25** minutes.
- Throughout the weekdays, including **Thursday**, **Tuesday**, and **Wednesday**, casual riders had slightly longer average ride durations compared to member riders.
- On **Thursday**, for instance, casual riders averaged **27.70** minutes per ride, while member riders averaged **12.78** minutes.
- The same pattern continues for **Tuesday** and **Wednesday**, where casual riders' average ride durations are slightly higher than those of member riders.

These findings highlight the variations in average ride durations across different days of the week and between casual and member riders. The table effectively illustrates how ride durations are influenced by both the day of the week and the type of rider, contributing to a comprehensive understanding of ride patterns.

```
cbs %>%
     group_by(rideable_type, member_casual) %>%
     summarise(Count = n()) %>%
```

```
    ggplot2::ggplot(aes(as.factor(rideable_type), Count, fill = member_casual)) +
    geom_col(position = "dodge", size = 1) +
    labs(title = "What did the data tell us: Which bike is popular", x = "Type of bike", y = "Number o
    theme_minimal() +
    scale_y_continuous(labels = label_number(scale = 1e-5, suffix = "m")) +
    theme(
      plot.title = element_text(size = 20, face = "bold"),
      axis.title = element_text(size = 14),
      axis.text = element_text(size = 10),
      legend.position = "bottom"
    ) +
      geom_label(
      aes(label = Count),
      position = position_dodge(width = 0.9),
      vjust = -0,
      size = 3,
      colour = "white"
    ) +
  guides(fill = guide_legend(override.aes = list(label = ""))) +
  my_favorite_colors
```

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



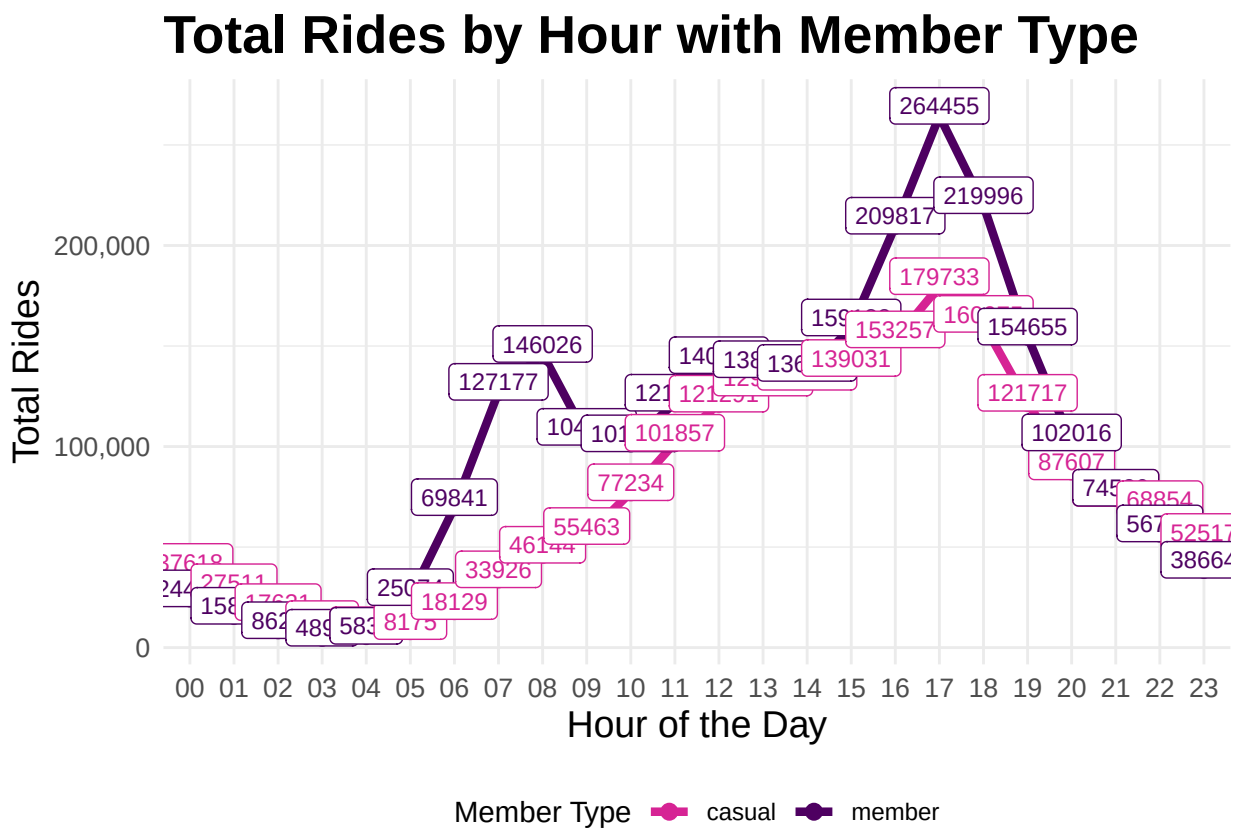
Observations:

- **Classic bikes** were the most frequently used rideable type by both casual and member riders.
- **Electric bikes** also had significant usage, particularly among **casual riders**.

```
# Total Rides by Start Hour and Member Type

cbs %>%
  group_by(start_hour, member_casual) %>%
  summarise(Total = n()) %>%
  arrange(start_hour) %>%
  ggplot(aes(start_hour, Total, color = member_casual, group = member_casual)) +
  geom_line(size = 1.5) +
  geom_point(size = 2.5) +
  scale_color_manual(values = c('#d62394', '#4e0061')) +
  labs(title = "Total Rides by Hour with Member Type",
       x = "Hour of the Day",
       y = "Total Rides",
       color = "Member Type") +
  scale_y_continuous(labels = scales::comma) +
  theme_minimal() +
  theme(
    plot.title = element_text(size = 20, face = "bold"),
    axis.title = element_text(size = 14),
    axis.text = element_text(size = 10),
    legend.position = "bottom"
  ) +
  geom_label(aes(label = Total), size = 3, nudge_y = 5000, show.legend = FALSE)
```

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



Observations:

The data reveals intriguing patterns in terms of the total number of rides by hour for various member types.

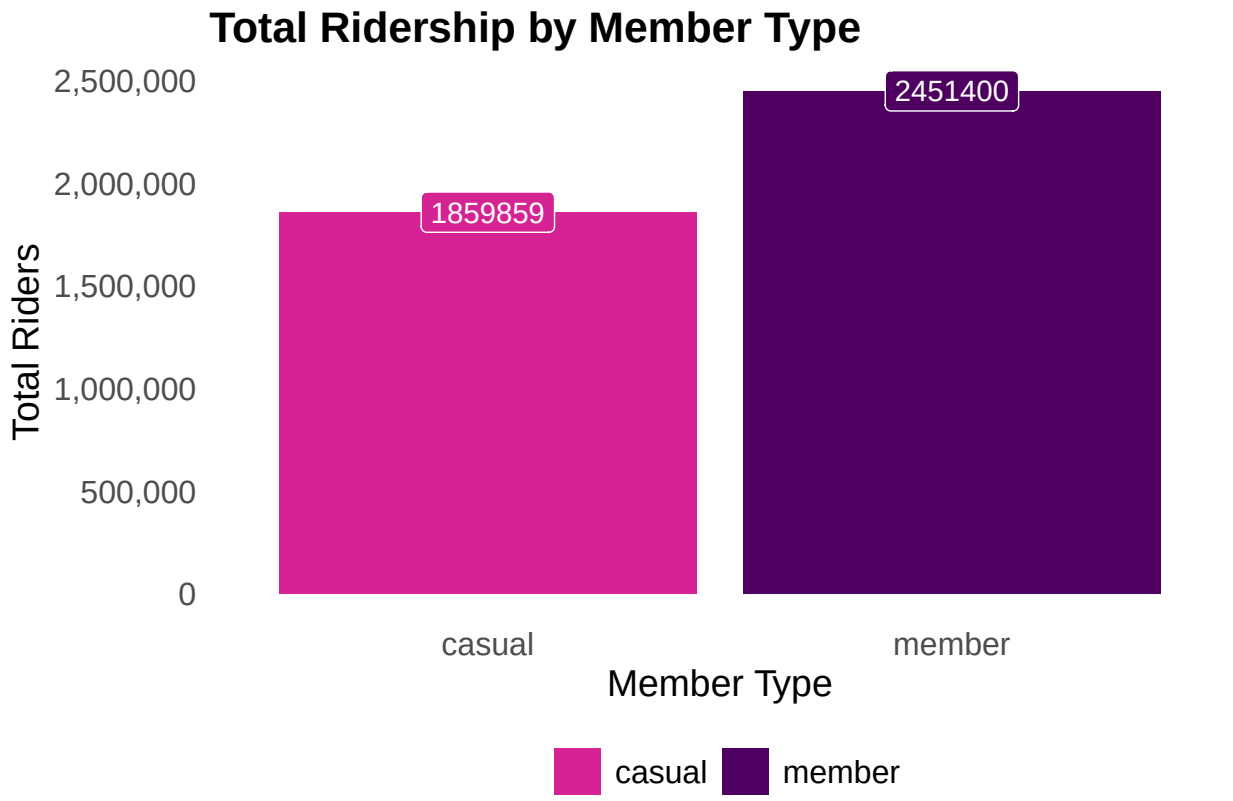
- During the early hours of the day (**00**:00-**03**:00), casual riders exhibit lower ride counts in comparison to members.
- However, as the day progresses, both member and casual riders experience a substantial surge in ride counts, reaching a peak between **15**:00-**17**:00 hours (**3:00 PM-5:00 PM**).
- This peak may signify that riders, regardless of their member type, tend to prefer using bikes during the late afternoon hours, possibly for commuting or leisure purposes.
- It's noteworthy that casual riders showcase a more diverse distribution of ride counts throughout the day, whereas member riders demonstrate a more consistent pattern.

In summary, the visualization offers valuable insights into the hourly riding behavior of distinct member types, potentially paving the way for targeted marketing strategies or efficient resource allocation.

```
# Total riders of casual and members
cbs %>%
  group_by(member_casual) %>%
  summarise(Total = n()) %>%
  ggplot(aes(member_casual, Total, fill = member_casual)) +
  geom_col() +
  geom_label(aes(label = Total), color = "white") +
  scale_fill_manual(labels = c("member", "casual"), values = c("blue", "red")) +
```

```
  guides(fill = guide_legend(override.aes = list(label = ""))) +
  scale_y_continuous(labels = scales::comma) +
  labs(title = "Total Ridership by Member Type",
       x = "Member Type",
       y = "Total Riders",
       caption = "Source: Cyclistic Data") +
  theme_minimal() +
  theme(plot.title = element_text(size = 16, face = "bold"),
        axis.title = element_text(size = 14),
        axis.text = element_text(size = 12),
        legend.title = element_blank(),
        legend.text = element_text(size = 12),
        legend.position = "bottom",
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank()) +
  my_favorite_colors
```

```
## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.
```



Observations

Upon analyzing the data on member types, we can observe the following trends:

- **Casual riders** account for a significant portion of the total rides, contributing a total of **2,529,005** rides.
- On the other hand, **member riders** demonstrate a higher engagement, generating a total of **3,066,058** rides.
- This discrepancy in ride counts between casual and member riders highlights the substantial contribution of both groups to the overall ridership.

These findings underscore the diverse usage patterns of different member types and provide valuable insights for optimizing service offerings and enhancing user experience.

Act

Our business task was: How do annual members and casual riders use Cyclistic bikes differently?

Recommendations:

- Based on the analysis, the following recommendations are suggested for Cyclistic:

Weekend Riding Opportunities

As deduced during the Analysis phase, it's evident that casual riders predominantly choose weekends for their rides, particularly in the spring and summer months. To capitalize on this trend, introducing a limited-time discount (ranging from 10% to 15%) on annual memberships specifically during these months could prove enticing to casual riders. Given that casual riders are already acquainted with the Cyclistic program, such a promotion might effectively steer them towards opting for an annual membership.

Promotion of Bike Types

Our analysis underscores that casual riders are inclined to opt for classic bikes when riding longer distances. A strategic approach would be to increase the inventory of classic bikes and launch a targeted marketing campaign under the theme "Keeping it Classic." This initiative would draw more casual riders towards selecting classic bikes, thereby furthering their engagement with the Cyclistic service.

1. Targeted Marketing:

- Develop targeted marketing campaigns to attract casual riders on weekends, emphasizing leisure and recreational benefits while ensuring compliance with copyright guidelines.
- Highlight the convenience and time-saving aspects of membership on weekdays to encourage sign-ups and promote regular usage.

2. Improve Member Engagement:

- Implement loyalty programs and incentives to enhance member engagement during weekdays, particularly during commuting hours.
- Provide exclusive benefits to members, such as priority parking or access to faster bikes, while adhering to copyright regulations.

Next Steps

Enhancing Cyclistic's Rider Experience: Data-Driven Strategies

Targeted Marketing and Member Engagement

To optimize Cyclistic's rider experience, we propose implementing targeted marketing campaigns and strategies to engage both casual and annual members effectively.

Targeted Marketing:

1. **Weekend Appeal for Casual Riders:** Craft compelling marketing campaigns that appeal to casual riders, with a focus on weekends. Highlight the **leisure and recreational benefits** of riding during these times, all while ensuring compliance with copyright guidelines.
2. **Weekday Membership Benefits:** Emphasize the **convenience and time-saving advantages** of an annual membership during weekdays. Encourage sign-ups and promote regular bike usage among daily commuters and local travelers.

Improve Member Engagement:

1. **Enhanced Member Loyalty:** Implement **loyalty programs and incentives** tailored for weekdays, particularly during commuting hours. These programs can foster stronger engagement and retention among members with daily riding habits.
2. **Exclusive Member Benefits:** Offer unique advantages to Cyclistic's members, such as **priority parking or access to faster bikes**. These privileges can enhance the overall experience and encourage continuous membership, all while adhering to relevant copyright regulations.

Next Steps

Moving forward, we have outlined specific actions to enrich our existing dataset and further refine our strategies for maximum impact.

1. Gather Demographic Insights: Our immediate priority is to **collect anonymous demographic** and **behavioral data** from **Cyclistic's annual members**. By obtaining a deeper understanding of their motivations, we can develop distinct rider personas. These personas will be instrumental in tailoring targeted approaches that encourage casual riders to transition into loyal annual members.

2. Acquire Pass Type Data: We recognize the importance of obtaining data about casual riders' pass types—whether they opt for **single-ride passes or full-day passes**. This crucial information is currently absent from our dataset. By acquiring these details, we can gain invaluable insights into Cyclistic's casual rider base. This insight will significantly enhance our ability to design effective marketing strategies that resonate with their preferences and behaviors.