Cyclistic Bikeshare Case Study

Austin Broadbent

2025-03-29

1. Introduction

This case study was completed as part of Capstone for the Google Data Analytics Professional Certificate. The purpose of this project is to showcase technical skills (SQL, R, and Python) as well as soft skills (including problem-solving, attention to detail, and data communication). Alongside this notebook, I created a Google Slides presentation for a less technical communication of the main insights.

1.1 Project Background & Objectives

Cyclistic is a (fictional) bike-share program that features more than 5,800 bicycles and 600 docking stations around Chicago. The company offers single-ride passes, full-day passes, and annual memberships. Within the company, customers who purchase single-ride or full-day passes are referred to as casual riders, while customers who purchase annual memberships are Cyclistic members. More info on the case study can be found here.

This analysis will focus on the past 12 months of Cyclistic historical bike ride data (March 2024 - February 2025. The analysis aims to understand the differences in how casual riders and annual members use Cyclistic bikes. The results of this analysis will be used to design marketing strategies aimed at converting casual riders into annual members.

1.2 Data Source & Description

The data source consists of Cyclistic's historical trip data from March 2024 to February 2025. The data for each month of the time period was originally contained within a CSV file (12 CSVs in total). Key variables for our analysis include:

- `ride_id`: a unique ID for each ride (primary key)
- `started_at`: start time of the ride
- `ended_at`: end time of the ride
- `member_casual`: whether the rider is a member or a casual rider
- `rideable_type`: whether the bike is a classic bike, electric bike, or an electric scooter

1.3 Key Research 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?

---

2. Data Cleaning and Preprocessing

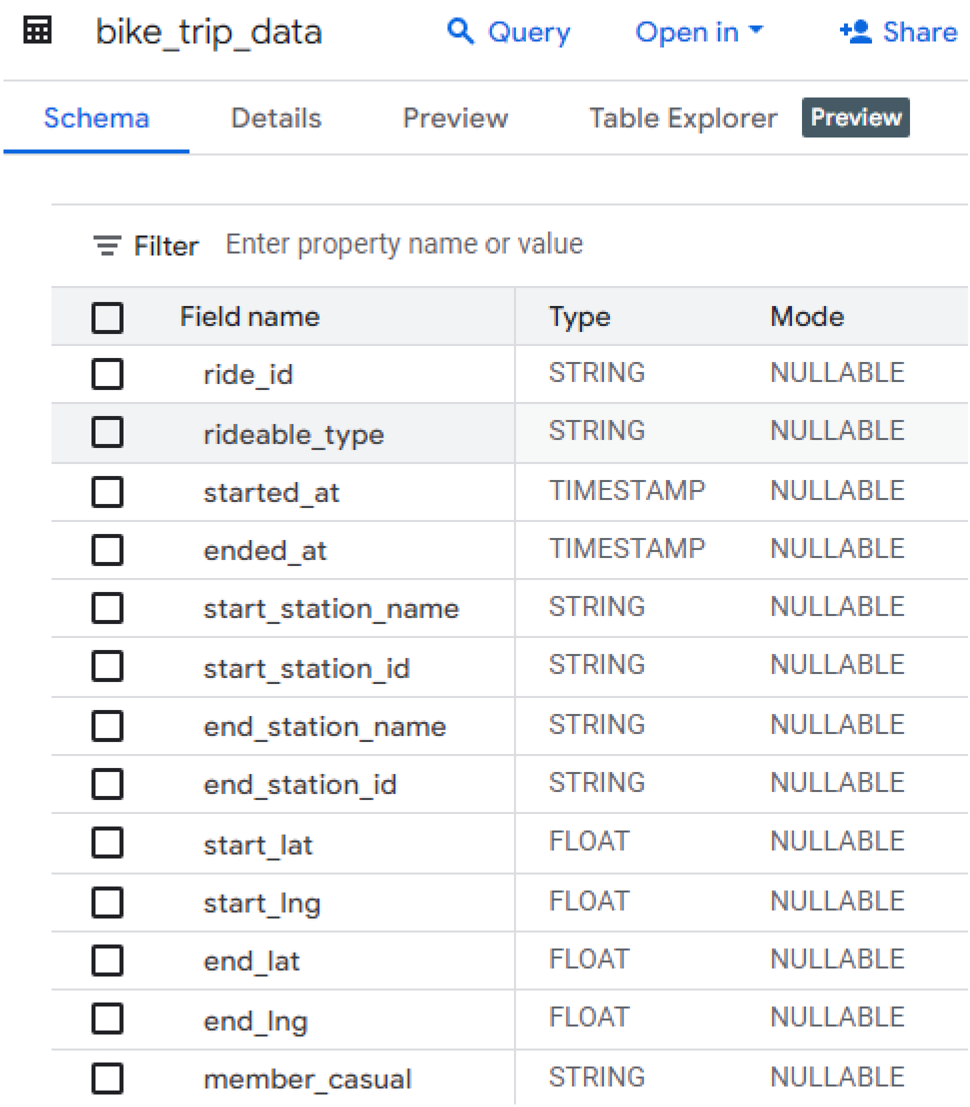
The entirety of this analysis could be completed in RStudio, but in order to practice writing SQL queries I first uploaded the data to BigQuery for cleaning and preprocessing.

2.0 Data Upload

As stated, the original data was split into 12 CSV files: one for each month. I created a table in BigQuery called `bike_trip_data`. My goal was to append each CSV to the end of the table, aggregating all of the CSVs into one queryable object. The first snag that I encountered was that the free version of BigQuery only allows files up to 100MB to be uploaded. So I went into Python and wrote a function to split the larger CSVs into multiple files:

```python
import csv
import os

def split_csv(input_filename, lines_per_file):
    """Splits a CSV file into smaller files.

    Args:
        input_filename (str): The name of the input CSV file.
        lines_per_file (int): The maximum number of data lines per output file.
    """
    # Get the filename and extension
    head, tail = os.path.split(input_filename)
    name, ext = os.path.splitext(tail)

    with open(input_filename, 'r', newline='') as infile:
        reader = csv.reader(infile)
        header = next(reader)  # Read the header row

        file_number = 1
        for i, row in enumerate(reader):
            if i % lines_per_file == 0:
                output_filename = f'{name}-pt{file_number}.csv'
                with open(output_filename, 'w', newline='') as outfile:
                    writer = csv.writer(outfile)
                    writer.writerow(header)  # Write the header to each file
                    writer.writerow(row)
                file_number += 1
            else:
                with open(output_filename, 'a', newline='') as outfile:
                    writer = csv.writer(outfile)
                    writer.writerow(row)
```

I verified that there was no loss of data when running the `split_csv()` function or during upload (Kaggle Notebook). The resulting table contained 5,783,100 observations and 13 columns:



□ Field name	Type	Mode
□ ride_id	STRING	NULLABLE
□ rideable_type	STRING	NULLABLE
□ started_at	TIMESTAMP	NULLABLE
□ ended_at	TIMESTAMP	NULLABLE
□ start_station_name	STRING	NULLABLE
□ start_station_id	STRING	NULLABLE
□ end_station_name	STRING	NULLABLE
□ end_station_id	STRING	NULLABLE
□ start_lat	FLOAT	NULLABLE
□ start_lng	FLOAT	NULLABLE
□ end_lat	FLOAT	NULLABLE
□ end_lng	FLOAT	NULLABLE
□ member_casual	STRING	NULLABLE

2.1 Missing Value Handling

Now that the data was aggregated and in one place, the first thing I noticed was that there were many missing values. Missing values existed in `start_station_name`, `start_station_id`, `end_station_name`, `end_station_id`, `end_longitude`, and `end_latitude`.

```
SELECT
    COUNT(CASE WHEN ride_id IS NULL THEN 1 END) AS missing_ride_id,
    COUNT(CASE WHEN rideable_type IS NULL THEN 1 END) AS missing_rideable_type,
    COUNT(CASE WHEN started_at IS NULL THEN 1 END) AS missing_start_time,
    COUNT(CASE WHEN ended_at IS NULL THEN 1 END) AS missing_end_time,
    COUNT(CASE WHEN start_station_name IS NULL THEN 1 END) AS missing_start_station_name,
    COUNT(CASE WHEN start_station_id IS NULL THEN 1 END) AS missing_start_station_id,
    COUNT(CASE WHEN end_station_name IS NULL THEN 1 END) AS missing_end_station_name,
    COUNT(CASE WHEN end_station_id IS NULL THEN 1 END) AS missing_end_station_id,
    COUNT(CASE WHEN start_lat IS NULL THEN 1 END) AS missing_start_latitude,
    COUNT(CASE WHEN start_lng IS NULL THEN 1 END) AS missing_start_longitude,
    COUNT(CASE WHEN end_lat IS NULL THEN 1 END) AS missing_end_latitude,
    COUNT(CASE WHEN end_lng IS NULL THEN 1 END) AS missing_end_longitude,
    COUNT(CASE WHEN member_casual IS NULL THEN 1 END) AS missing_member_type,
FROM
    ####.cyclistic_bikeshare.bike_trip_data;
```

missing_ride_id	0
missing_rideable_type	0
missing_start_time	0
missing_end_time	0
missing_start_station_name	1080148
missing_start_station_id	1080148
missing_end_station_name	1110075
missing_end_station_id	1110075
missing_start_latitude	0
missing_start_longitude	0
missing_end_latitude	6744
missing_end_longitude	6744
missing_member_type	0

Since all of these columns are variables dealing with location data, I was suspicious that there might be relationships within this missing data. After investigating, I found the following relationships:

1. When the `start_station_name` was missing from an observation, the `start_station_id` was also missing. (1,080,148 observations)
2. The same was true for `end_station_name` and `end_station_id`. (1,110,075 observations)
3. If `end_longitude`, and `end_latitude` were missing, `end_station_name` and `end_station_id` were also missing. In other words, the missing GPS data was a subset of the missing end station data. (6,744 observations)
4. There were 527,697 rows missing both start and end station data. None of these include the subset of missing GPS data.

In a real scenario, these findings could be investigated further to find the cause of the missing data. There could be errors in recording the station information, problems with the data collection process, GPS errors, software/hardware malfunctions, etc.

For this case study, a key part of my analysis will rest on the trip duration. Therefore, I felt that the missing location data called into question the reliability of these observations. Therefore I decided to filter out these observations and save the results into a new table. This way, the missing data is still contained in the original table available for later analysis.

```
CREATE OR REPLACE TABLE ####.cyclistic_bikeshare.found_bike_trip_data AS
SELECT * FROM ####.cyclistic_bikeshare.bike_trip_data
WHERE end_lat IS NOT NULL AND start_station_name IS NOT NULL AND end_station_name IS NOT NULL;
```

Running the original SQL query on the new table showed that there were now 4,120,571 observations with no missing values.

2.2 Duplicate Removal

Next, I checked for duplicate `ride_id`. Since this variable is supposed to be unique for each observation.

```
SELECT ride_id, COUNT(*) as count
FROM ####.cyclistic_bikeshare.found_gps_bike_trip_data
GROUP BY ride_id
HAVING COUNT(*) > 1;
```

ride_id	count
F8A9257D1DD04F43	2
1EF6CBC15814F5DA	2
43637BA11F2DAA42	2
C8279789821AE094	2
C4348E6A5DF57407	2
0AF69D1A20891AD2	2

There were 121 duplicates. I noticed that all of the duplicates had a count of 2. Upon further investigation, the duplicated instances were identical, EXCEPT that the start and end times for one instance were more precise than the other instance. Below is an example case:

ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_name	end_station_id	start_lat	start_lng	end_lat	end_lng	member_casual
F8A9257D1DD04F43	classic_bike	2024-05-31 23:43:04.939000 UTC	2024-06-01 00:02:46.881000 UTC	Sheffield Ave & Wrightwood Ave	TA1309000023	Halsted St & Fulton St	23003	41.92871	-87.65383	41.88687	-87.64809	member
F8A9257D1DD04F43	classic_bike	2024-05-31 23:43:04.000000 UTC	2024-06-01 00:02:46.000000 UTC	Sheffield Ave & Wrightwood Ave	TA1309000023	Halsted St & Fulton St	23003	41.92871	-87.65383	41.88687	-87.64809	member

I decided that the more precise time was more likely to be accurate. For each pair of duplicates, I kept the record with the more detailed time stamp and filtered out the other, making a third table.

```
CREATE OR REPLACE TABLE ####.cyclistic_bikeshare.dedepulicate_bike_trip_data AS
WITH ranked_rows AS (
  SELECT
    *,
    ROW_NUMBER() OVER (PARTITION BY ride_id ORDER BY started_at DESC) as rn
  FROM
    ####.cyclistic_bikeshare.found_bike_trip_data
)
SELECT * EXCEPT (rn) FROM ranked_rows WHERE rn = 1;
```

After verifying that all duplicates were removed from the data, there were 4,120,450 observations.

2.3 Duration Outliers & Inconsistencies

Next, I wanted to check for inconsistencies with the start and end times of the rides. I added a new column to calculate the ride duration in minutes. This needed two SQL queries: an `ALTER TABLE` query to add the column, and an `UPDATE` to populate it with the calculation.

```
ALTER TABLE
  ####.cyclistic_bikeshare.dedepulicate_bike_trip_data
ADD COLUMN
  ride_duration_minutes FLOAT64;

UPDATE
  ####.cyclistic_bikeshare.dedepulicate_bike_trip_data
SET
  ride_duration_minutes = TIMESTAMP_DIFF(ended_at, started_at, SECOND) / 60
WHERE TRUE;
```

Now that I could easily check how long each ride lasted, I checked for rides that didn't make logical sense (i.e. the start time was after the end time). I found and removed 267 instances like this:

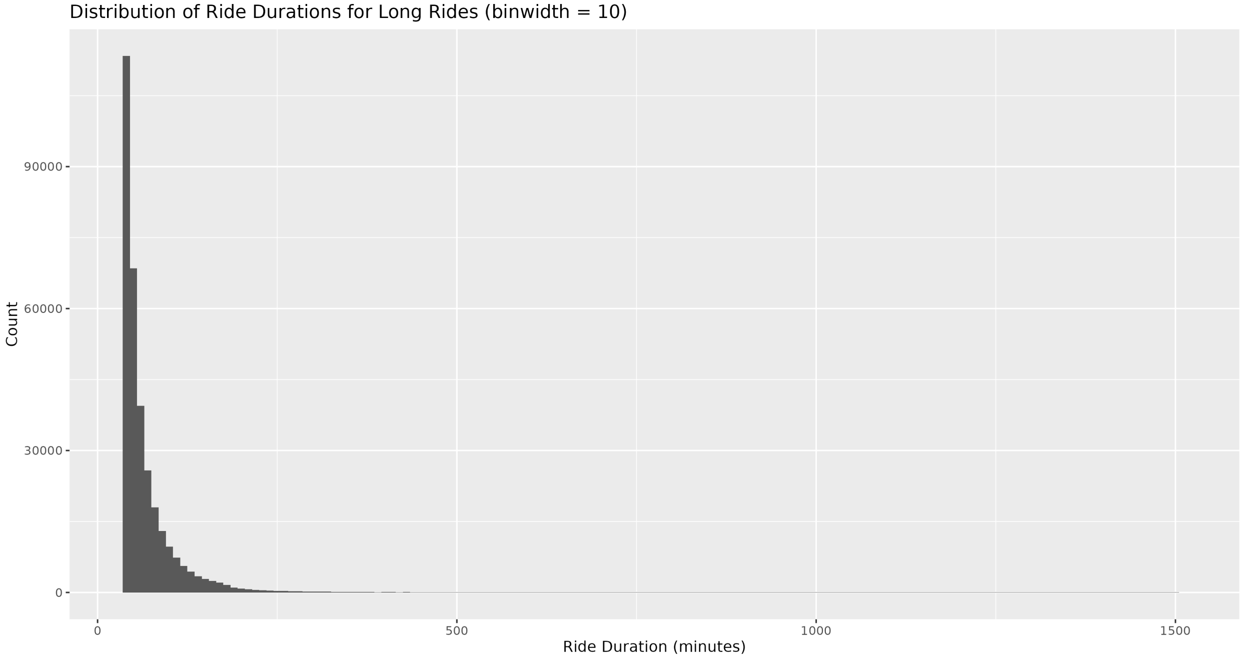
```
SELECT
    ride_id,
    started_at,
    ended_at,
    ride_duration_minutes
FROM
    ####.cyclistic_bikeshare.dedepulicate_bike_trip_data
WHERE
    ride_duration_minutes <= 0;

DELETE FROM
    ####.cyclistic_bikeshare.dedepulicate_bike_trip_data
WHERE
    ride_duration_minutes <= 0;
```

Even after removing the rides that were obviously illogical, I was still suspicious of outliers. I uploaded the data set to RStudio to do some quick exploratory data analysis. I calculated some summary statistics and made a quick histogram (`ggplot(df, aes(x=ride_duration_minutes)) + geom_histogram()`). This analysis revealed that the ride durations had the vast majority of points close to zero, but a large tail that extended over 1000 minutes.

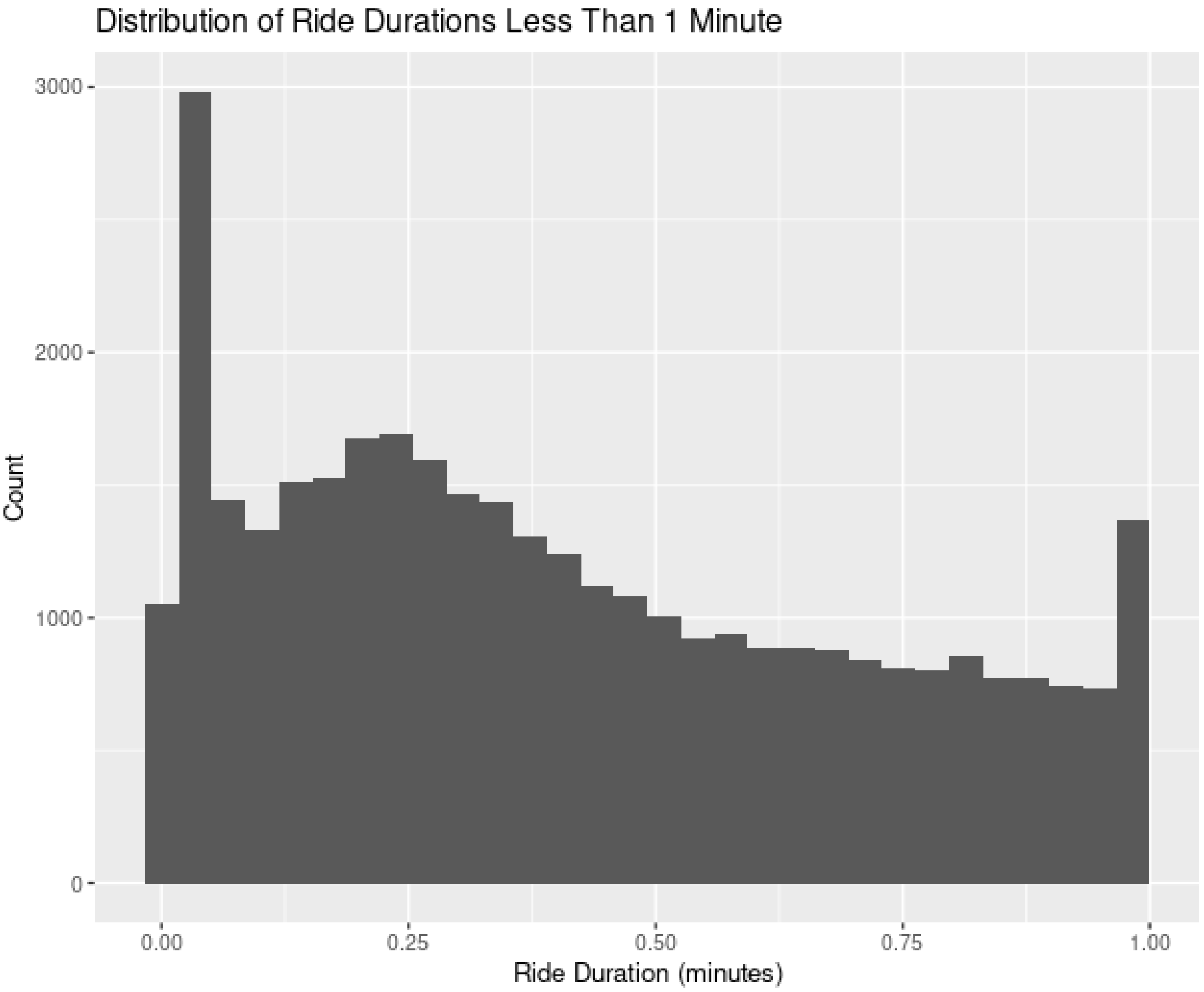
I investigated both short and long rides. I calculated long rides by using the Interquartile Rule ($1.5 * IQR + Q3$). Because of the distribution of ride duration, using the Interquartile Rule on the other end led to a negative ride duration for short rides. Instead, I used rides that were less than or equal to 1 minute. For both short and long rides, I made 3 visualizations to familiarize myself with the outliers:

1. Histogram (included below)
2. Scatter plot of Start Time vs Ride Duration
3. Hourly/Daily groupings
4. Box Plot of Ride Duration by Rider Type

The most important insight I gained from this is shown in the histograms below. The long rides histogram has a huge tail that extends to over 1500 minutes (over 1 day!). That seems like a data collection issue.



The short ride histogram has the bulk of rides less than 0.25 minutes (or 15 seconds). This also seems suspicious.



Just as with the missing location data, further investigation could be done to determine a cause for these outliers. As is, I found some interesting anomalies. Between 4-5AM, there was a huge increase in extremely short rides (<15 seconds), as well as rides that are abnormally long (>400 minutes). This suggest there could be a system error occurring at this time that affects the correct logging of start/end times.

As I lacked enough information, I decided to be conservative in my approach. I filtered out rides that were less than 15 seconds (0.25 minutes), and those that were greater than 16 hours (960 minutes or the time that most people are awake in a day). This was done using the same SQL query as removing negative ride durations, just adjusting the integer.

In total, this filtered an additional 15,016 observations out of the table. There were now 4,105,434 observations remaining to be analyzed.

2.4 Other Inconsistencies

I also wanted to check for inconsistencies in the other variables before beginning my analysis. A quick rundown of these checks is listed below:

- Trimmed whitespace on all string variables

- Checked that all ride_ids follow a consistent pattern and are 16 characters in length
- Checked for misspellings in `rideable_type` and `member_casual` columns
- Checked that the GPS data contains locations within the service area of Chicago
- Checked that all start/end times are in 2024 or 2025
- Checked that each column contains the correct data type

2.5 Data Limitations & Potential Biases

2.5.1 Data Loss Due to Filtering

A significant amount of data was removed during this process. To summarize:

- Rows with missing values in `end_lat/end_lng`, `start_station_name/start_station_id`, and `end_station_name/end_station_id`
- Rows with duplicate `ride_id` and less precise start/end times
- Rows with illogical ride durations
- Rows with outliers in ride duration (extremely short and long rides)

This was done to avoid unreliable data, but could result in bias if the removed rows were not randomly distributed.

2.5.2 Assumptions About Data Accuracy

As stated, data was removed with justification to avoid unreliable data in the analysis. But, without further information, it is hard to be 100% sure that rows were removed for legitimate reasons.

Additionally, the rest of this analysis assumes that the remaining data is reliable. But, if there were errors in data collection or recording of some rows, there could be undetectable errors in the remaining rows, too. An example of this is the following issue I discovered:

According to the cast study, Cyclistic has 692 stations across Chicago, but there are 1746 distinct start station ids and 1754 distinct end station ids in the table. A similar problem exists with start_station_names (1805) and end_station_names (1814). In addition, Several instances were found where the same station id has 2 different station names, and vice versa.

However without a definitive list of Cyclistic station IDs and names, I found that I could not reliably make changes. Therefore, I ultimately decided to not use spatial data in my analysis. These rows are still included in the data, so it is assumed that the other variables associated with the observation are accurate.

2.5.3 Time Frame of Data

This analysis is limited to one year of data: March 2024 - February 2025. Longer term trends in the data might not be captured in this data.

2.5.4 Other Factors

As stated, with more information the spatial data could be studied more deeply. However, there could be many other outside factors that affect rider's behavior:

- Weather conditions
- Local festivals and events
- Road construction and closures
- Marketing campaigns

---

3. Exploratory Data Analysis

Once the data was ready, I uploaded it to RStudio because R make data visualization easy. Since I decided to that the location data was not reliable enough to include in the analysis, I filtered the data frame to only contain the relevant columns:

```r
df <- read_csv("C:/Users/Administrator/Downloads/cleaned_bikeshare_data.csv", show_col_types = FALSE)
df <- df |> select(c(ride_id, member_casual, rideable_type, started_at, ended_at, ride_duration_minutes)
knitr::kable(head(df))
```

ride_id	member_casual	rideable_type	started_at	ended_at	ride_duration_minutes
C9E22923AE7D4CFC	casual	classic_bike	2024-11-15 00:34:41.598 UTC	2024-11-15 00:50:16.346 UTC	15.566667
7B1DC201622DE078	member	classic_bike	2024-08-30 16:20:55.581 UTC	2024-08-30 16:36:47.45 UTC	15.850000
D502D4C85A2AE88C	casual	electric_bike	2024-08-30 21:31:33.63 UTC	2024-08-30 21:42:03.136 UTC	10.483333
01EC0A48ABF5C3E8	casual	electric_bike	2024-09-20 07:57:32.958 UTC	2024-09-20 08:23:45.826 UTC	26.200000
EE44579C007B7C01	casual	classic_bike	2024-08-09 16:42:37.606 UTC	2024-08-09 16:47:09.52 UTC	4.516667
665E71D52FB52693	member	electric_bike	2024-06-07 05:59:19.036 UTC	2024-06-07 06:03:09.764 UTC	3.833333

I began to do Exploratory Data Analysis to find connections within the data. My methodology included the following:

- **Summary Statistics:** Calculating descriptive statistics such as mean, median, standard deviation, minimum, and maximum values to understand the distribution and range of key variables (e.g., ride duration).
- **Grouping Data:** Grouping and summarizing data by rider type (`member_casual`), hour of the day, day of the week, and month to identify trends and patterns in ride frequency and duration.
- **Data Visualization:** Creating charts and graphs to visually represent the data. The visualizations included:
  - Bar charts for comparing ride counts.
  - Line charts for visualizing trends over time (hourly, daily, monthly).
  - Histograms to understand the distribution of ride durations.
  - Grouped bar charts to compare rideable type preferences.

3.1 Data Transformations

To make these visualizations easier to accomplish, I used the `started_at` column to create new features `hour`, `weekday`, and `month`. After creating the `weekday` column, I wanted the weekdays to be plotted in order starting with Monday to more easily compare weekends vs. weekdays. So I converted the column to an ordered factor.

```r
df$started_at <- ymd_hms(df$started_at, tz="UTC")

df <- df |>
  mutate(
    hour = hour(started_at),
    weekday = weekdays(started_at),
    month = month(started_at)
  )

day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday")
df$weekday <- factor(df$weekday, levels = day_order, ordered = TRUE)

knitr::kable(head(df))
```

ride_id	member_casual	rideable_type	started_at	ended_at	ride_duration_minutes	hour	weekday	month
C9E22923AE7D4CFC	casual	classic_bike	2024-11-15 00:34:41	2024-11-15 00:50:16.346 UTC	15.566667	0	Friday	11
7B1DC201622DE078	member	classic_bike	2024-08-30 16:20:55	2024-08-30 16:36:47.45 UTC	15.850000	16	Friday	8
D502D4C85A2AE88C	casual	electric_bike	2024-08-30 21:31:33	2024-08-30 21:42:03.136 UTC	10.483333	21	Friday	8
01EC0A48ABE5C3E8	casual	electric_bike	2024-09-20 07:57:32	2024-09-20 08:23:45.826 UTC	26.200000	7	Friday	9
EE44579C007B7C01	casual	classic_bike	2024-08-09 16:42:37	2024-08-09 16:47:09.52 UTC	4.516667	16	Friday	8
665E71D52FB52693	member	electric_bike	2024-06-07 05:59:19	2024-06-07 06:03:09.764 UTC	3.833333	5	Friday	6

3.2 Descriptive Statistics

Descriptive statistics were calculated to provide a summary of `ride_duration_minutes` for casual riders and annual members. This helps to see at a quick glance the basic distribution of each group:

```r
duration_stats <- df |>
  group_by(member_casual) |>
  summarize(
    n = n(),
    avg_duration = mean(ride_duration_minutes),
    median_duration = median(ride_duration_minutes),
    sd_duration = sd(ride_duration_minutes),
    min_duration = min(ride_duration_minutes),
    max_duration = max(ride_duration_minutes),
    q25_duration = quantile(ride_duration_minutes, 0.25),
    q75_duration = quantile(ride_duration_minutes, 0.75)
  )

knitr::kable(t(duration_stats))
```

member_casual	casual	member
n	1495902	2609532
avg_duration	23.09728	12.26361
median_duration	13.433333	8.816667
sd_duration	36.18510	15.93525
min_duration	0.2666667	0.2666667
max_duration	958.7333	959.6500
q25_duration	7.533333	5.216667
q75_duration	25.65	15.00

3.3 Visualizations

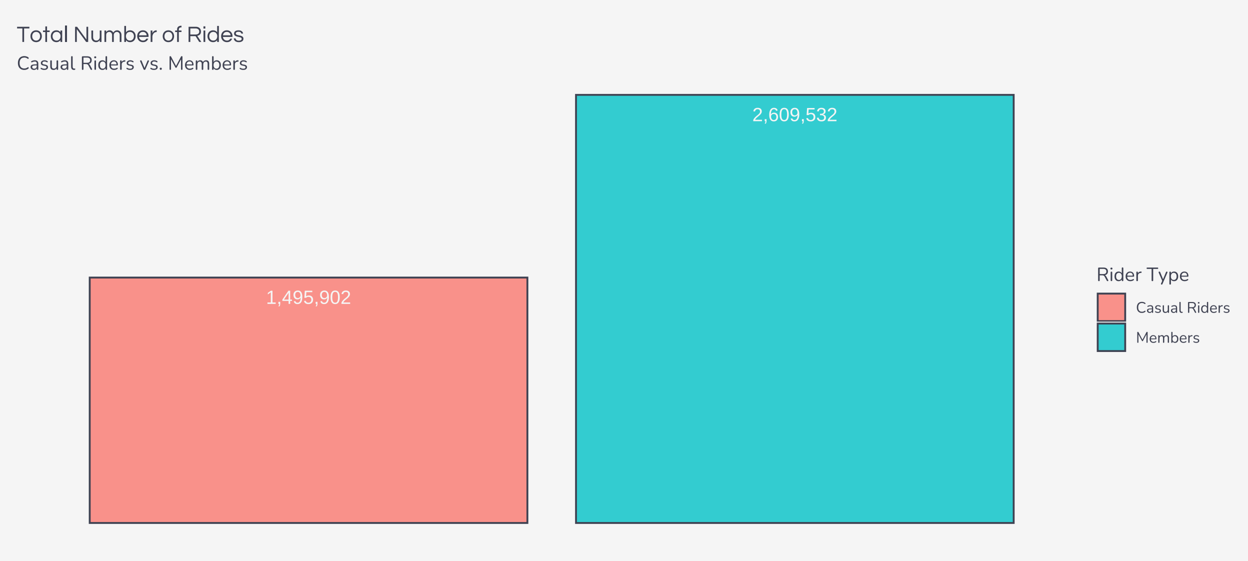
This section presents the visualizations created to explore the patterns in Cyclistic's ride data. I will provide a caption to explain the purpose and key insights gained from each plot. (Note: the visual design choices were made to match the theme of the Google Slides presentation accompanying this document. See the link above for the final results of that presentation.)

```
font_add_google("Questrial", "questrial") #Fonts
font_add_google("Nunito", "nunito")
showtext_auto()

background_color <- "#f5f5f5" #Plot design colors
text_color <- "#3f4252"

rider_labels <- c("Casual Riders", "Members") #Member_casual plot labels and colors
rider_colors <- c("casual" = "#F9918A", "member" = "#33CCD0")


bike_types <- c("electric_scooter", "electric_bike", "classic_bike") #Rideable_type plot labels and col
bike_colors <- c("electric_scooter" = "#FF7F50", "electric_bike" = "#33C860", "classic_bike" = "#81B0FF

hour_labels <- setNames(format(strptime(0:23, format = "%H"), "%l %p"), 0:23) #Formatting for x-axis of
breaks_seq <- seq(0, 23, by = 4)
labels_seq <- hour_labels[as.character(breaks_seq)]
bike_labels <- c("Classic Bike", "Electric Bike", "Electric Scooter") # Labels for rideable type plot
```

3.3.1 Rider Type Comparison

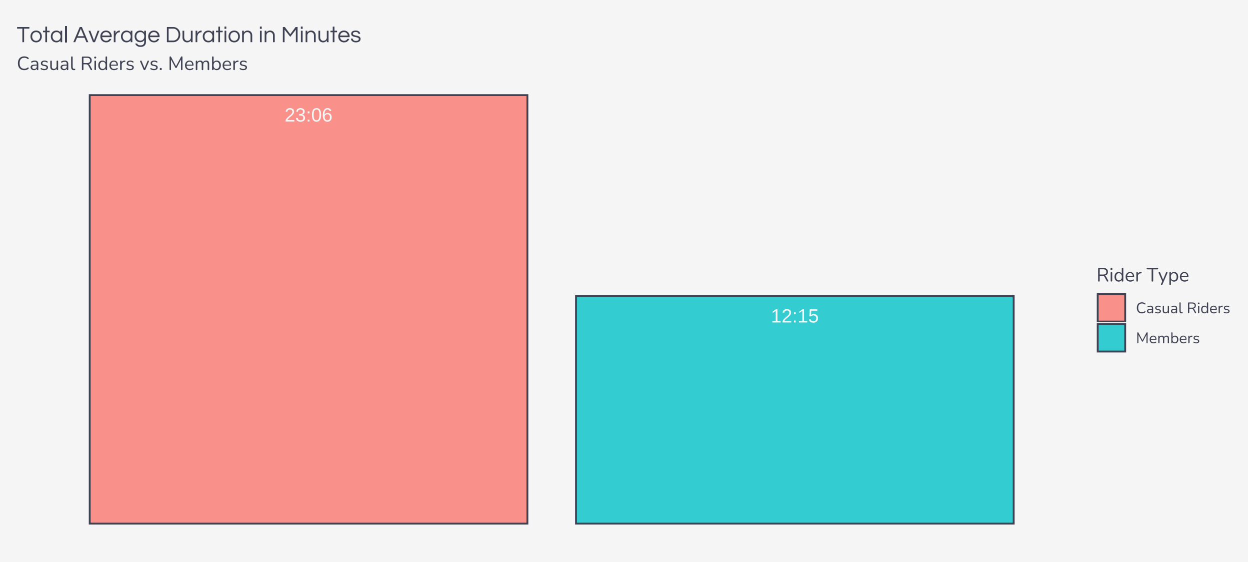
Total Number of Rides This bar chart compares the total number of rides taken by casual riders and annual members during the past 12 months. It shows that members take the majority of rides (63.56%, in fact).

```
#Rider Type
rider_type <- df |>
  group_by(member_casual) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

ggplot(rider_type, aes(x = member_casual, y = ride_count, fill = member_casual)) +
  geom_bar(stat = "identity", color = text_color) +
  labs(
    title = "Total Number of Rides",
    subtitle = "Casual Riders vs. Members",
    fill = "Rider Type"
  ) +
  geom_text(aes(label = c("1,495,902","2,609,532"), vjust = 2), color = background_color) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  theme_void() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color)
  )
```



Average Ride Duration This bar chart compares the average ride duration (in minutes) for casual riders and annual members. It helps to understand the typical trip length for each rider group. While members take more trips, casual riders take longer trips, on average. (almost twice as long!)

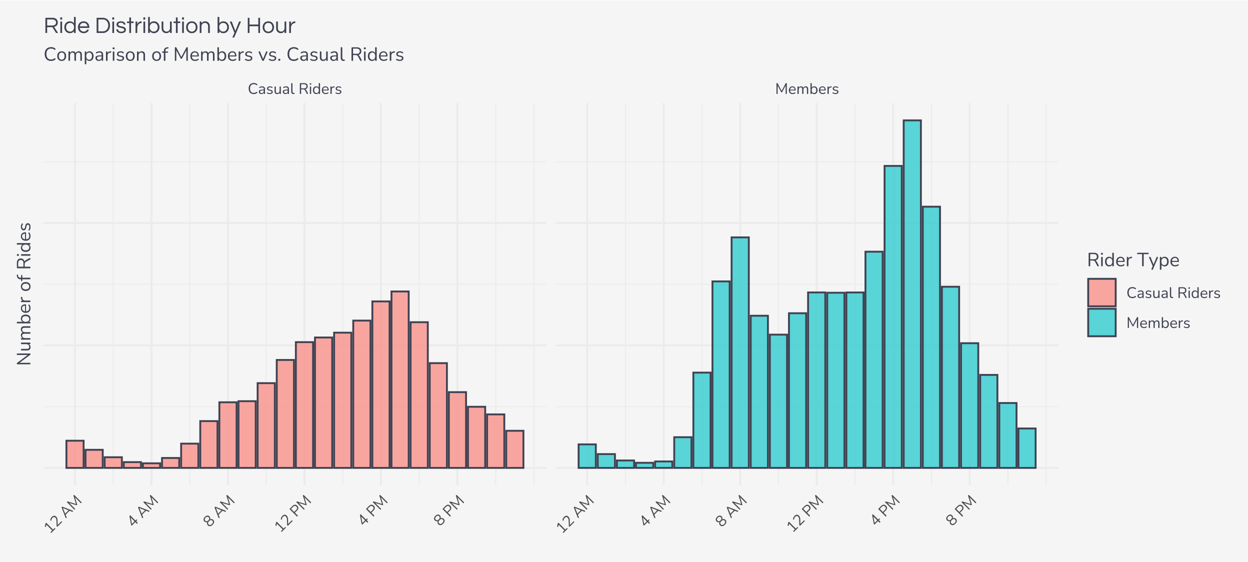
```
ggplot(rider_type, aes(x = member_casual, y = avg_duration, fill = member_casual)) +
  geom_bar(stat = "identity", color = text_color) +
  labs(
    title = "Total Average Duration in Minutes",
    subtitle = "Casual Riders vs. Members",
    fill = "Rider Type"
  ) +
  geom_text(aes(label = c("23:06", "12:15"), vjust = 2), color = background_color) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  theme_void() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color)
  )
```



3.3.2 Hourly Patterns

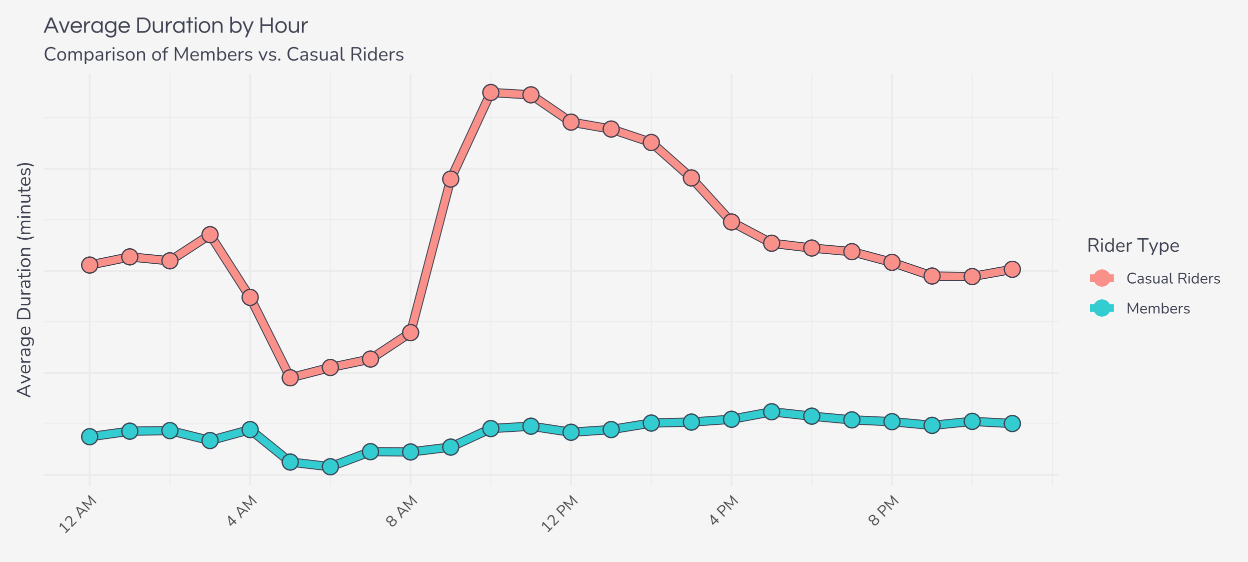
Ride Distribution by Hour These side by side histograms compare the time of day when each rider type takes their rides. It shows that members have peak usage during commuting hours, while casual riders during midday and afternoon hours.

```
hourly_rider_type <- df |>
  group_by(member_casual, hour) |>
  filter(!is.na(hour)) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

ggplot(hourly_rider_type, aes(x = hour, y = ride_count, fill = member_casual)) +
  geom_col(position = "dodge", color = text_color, alpha = 0.8) +
  labs(
    title = "Ride Distribution by Hour",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Number of Rides",
    fill = "Rider Type"
  ) +
  scale_x_continuous(breaks = breaks_seq, labels = labels_seq) +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  facet_wrap(~member_casual,labeller = as_labeller(c("casual" = "Casual Riders", "member" = "Members"))
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    strip.text = element_text(colour = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
    )
```



Average Duration by Hour This line chart shows the average ride duration by hour of the day. It shows that casual riders have longer averages overall (which matches our expectations from the first bar chart). But also, casual riders have more varied ride times throughout the day, while members' average ride duration stays fairly steady.

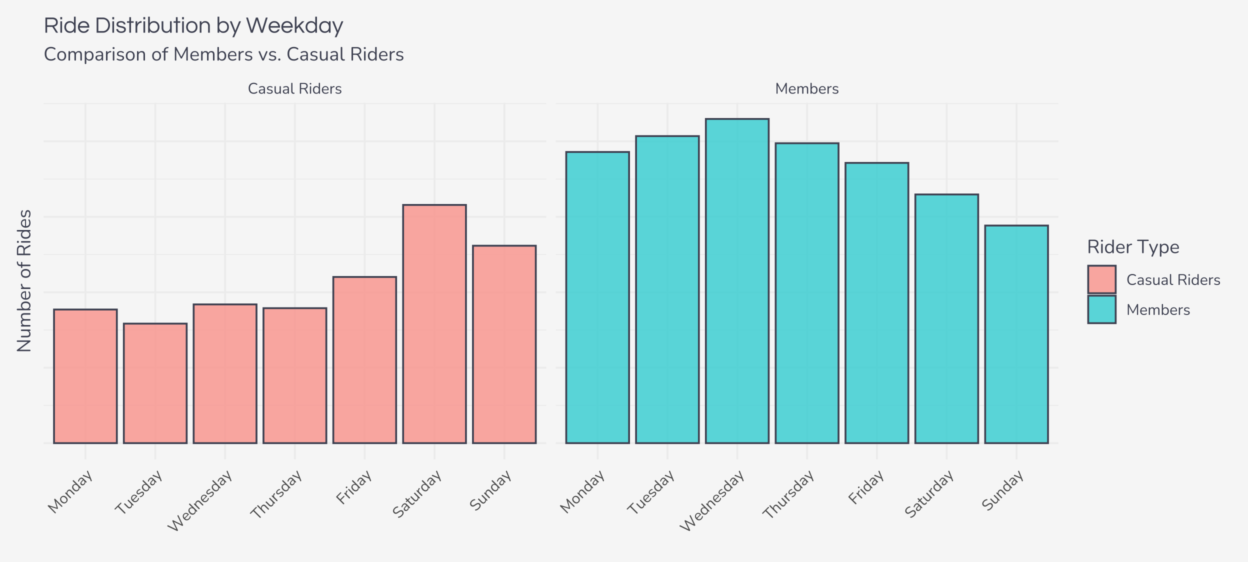
```
ggplot(hourly_rider_type, aes(x = hour, y = avg_duration, color = member_casual)) +
  geom_line(linewidth = 2.5, color = text_color, aes(group = member_casual)) +
  geom_line(linewidth = 2, aes(group = member_casual)) +
  geom_point(size = 4) +
  geom_point(size = 4, color = text_color, shape = 1) +
  labs(
    title = "Average Duration by Hour",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Average Duration (minutes)",
    color = "Rider Type"
  ) +
  scale_x_continuous(breaks = breaks_seq, labels = labels_seq) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = rider_colors, labels = rider_labels) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )
```



3.3.3 Weekly Patterns

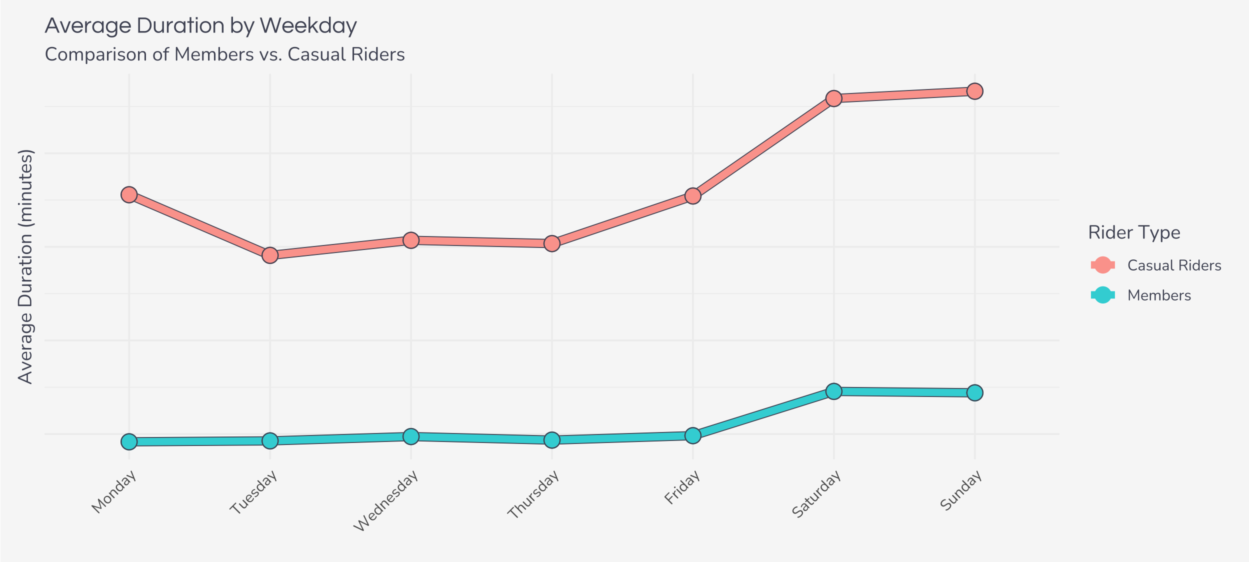
Ride Distribution by Weekday These side by side histograms show the distribution of rides by day of the week for each group. We can see an uptick in the number of rides by casual riders on the weekend, while the members have the highest usage during the weekdays.

```
weekday_rider_type <- df |>
  group_by(member_casual, weekday) |>
  filter(!is.na(member_casual)) |>
  filter(!is.na(weekday)) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

ggplot(weekday_rider_type, aes(x = weekday, y = ride_count, fill = member_casual)) +
  geom_col(position = "dodge", color = text_color, alpha = 0.8) +
  labs(
    title = "Ride Distribution by Weekday",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Number of Rides",
    fill = "Rider Type") +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  facet_wrap(. ~ member_casual,labeller = as_labeller(c("casual" = "Casual Riders", "member" = "Members
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    strip.text = element_text(colour = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )
```



Average Duration by Weekday This line chart shows the average ride duration by day of the week. It shows that both types of riders also have longer average ride times on weekends than weekdays, but this effect is more pronounced for casual riders.

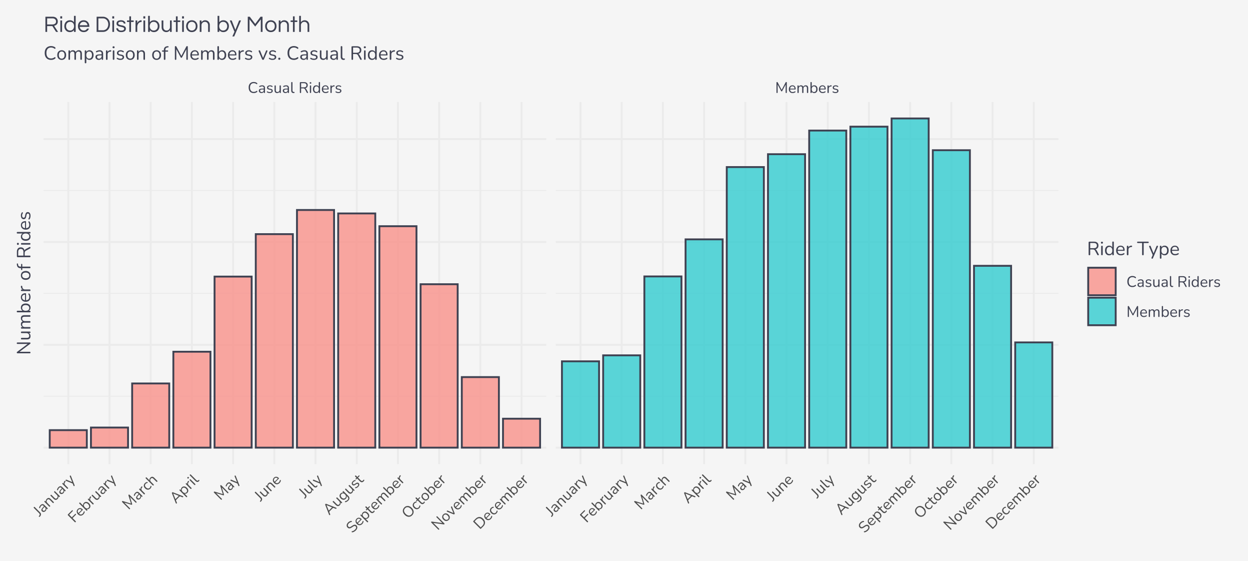
```
ggplot(weekday_rider_type, aes(x = weekday, y = avg_duration, color = member_casual)) +
  geom_line(linewidth = 2.5, color = text_color, aes(group = member_casual)) +
  geom_line(linewidth = 2, aes(group = member_casual)) +
  geom_point(size = 4) +
  geom_point(size = 4, color = text_color, shape = 1) +
  labs(
    title = "Average Duration by Weekday",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Average Duration (minutes)",
    color = "Rider Type"
  ) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = rider_colors, labels = rider_labels) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )
```



3.3.4 Monthly Patterns

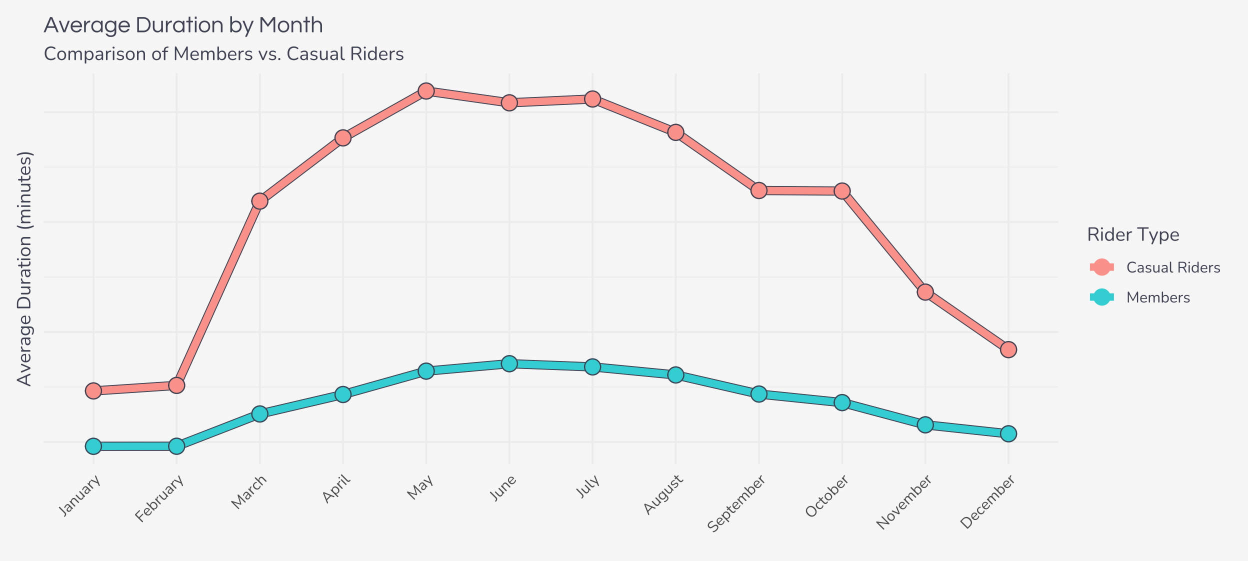
Ride Distribution by Month These side by side histograms show the distributions of rides by month. It can help us see the seasonal trends in rider usage. It shows that both groups use the service more in warmer months, from about May until October.

```
monthly_rider_type <- df |>
  group_by(member_casual, month) |>
  filter(!is.na(month)) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

ggplot(monthly_rider_type, aes(x = month, y = ride_count, fill = member_casual)) +
  geom_col(position = "dodge", color = text_color, alpha = 0.8) +
  labs(
    title = "Ride Distribution by Month",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Number of Rides",
    fill = "Rider Type") +
  scale_x_discrete(limits = month.name[1:12], labels = month.name[1:12]) +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  facet_wrap(~member_casual,labeller = as_labeller(c("casual" = "Casual Riders", "member" = "Members"))
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    strip.text = element_text(colour = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )
```



Average Duration by Month This line chart shows the average ride duration by month. It shows that both members use the service for longer in the warmer months, but as with weekdays, this effect is more pronounced in casual riders.

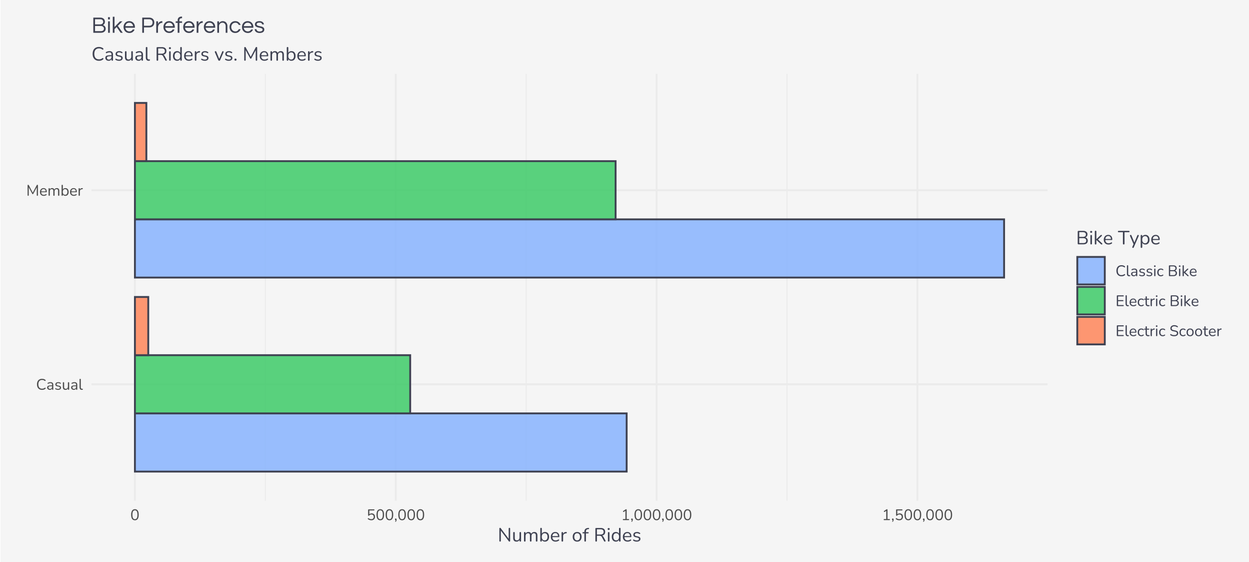
```
ggplot(monthly_rider_type, aes(x = month, y = avg_duration, color = member_casual)) +
  geom_line(linewidth = 2.5, color = text_color, aes(group = member_casual)) +
  geom_line(linewidth = 2, aes(group = member_casual)) +
  geom_point(size = 4) +
  geom_point(size = 4, color = text_color, shape = 1) +
  labs(
    title = "Average Duration by Month",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Average Duration (minutes)",
    color = "Rider Type"
  ) +
  scale_x_discrete(limits = month.name[1:12], labels = month.name[1:12]) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = rider_colors, labels = rider_labels) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )
```



3.3.5 Rideable Type Patterns

This grouped bar chart shows the number of rides by rider type and rideable type. It shows the preferences for bike type for each group. Both groups prefer classic bikes, and there doesn't seem to be much of a difference in usage patterns when it comes to rideable type.

```
ride_preference <- df |>
  group_by(member_casual, rideable_type) |>
  summarize(ride_count = n())

ggplot(ride_preference, aes(fill = rideable_type, y = member_casual, x = ride_count)) +
  geom_bar(position = "dodge", stat = "identity", color = text_color, alpha = 0.8) +
  labs(
    title = "Bike Preferences",
    subtitle = "Casual Riders vs. Members",
    fill = "Bike Type",
    y = element_blank(),
    x = "Number of Rides"
  ) +
  scale_fill_manual(values = bike_colors, labels = bike_labels) +
  scale_x_continuous(labels = scales::comma,) +
  scale_y_discrete(labels = c("Casual", "Member")) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color)
  )
```



4. Analysis Results

This section presents the key findings from the exploratory data analysis, focusing on the business question: How do annual members and casual riders use Cyclistic bikes differently?

4.1 Total Number of Rides and Average Duration Analysis

- **Total Rides:** Annual members take significantly more rides than casual riders. Members took 2,609,532 rides, while casual riders took 1,495,902 rides.
- **Average Ride Duration:** Casual riders have a significantly longer average ride duration than annual members. The average ride duration for casual riders is 23 minutes and 6 seconds, while for annual members, it is 12 minutes and 15 seconds.
- **Key Insight:** Annual members are more frequent users, but casual riders take longer trips.

4.2 Rideable Type Preferences

- Casual riders primarily use classic bikes (942,638 rides) and electric bikes (527,689 rides), with a small number using electric scooters (25,575 rides).

- Annual members also primarily use classic bikes (1,666,181 rides) and electric bikes (921,430 rides), with even fewer using electric scooters (21,921 rides).

- Both rider groups prefer classic and electric bikes over electric scooters. There doesn't seem to be much difference between casual riders and annual members in this area.

4.3 Temporal Patterns

- **Hourly Patterns:**
  - Members have peak usage during commuting hours.
  - Casual riders have peak usage during midday and afternoon hours.
  - Casual riders have longer and more varied ride times throughout the day.
- **Daily Patterns:**
  - Members use bikes more on weekdays.
  - Casual riders use bikes more on weekends.
  - Both have longer average ride times on weekends, but more pronounced for casual riders.
- **Monthly Patterns:**
  - Both user types show increased usage during warmer months.
  - The increase in usage during warmer months is more pronounced for casual riders.
  - Casual riders have much longer average ride times during warmer months.

4.4 Casual Rider Behavior Insights

- Casual riders seem to use Cyclistic for recreational purposes.
- Casual riders use Cyclistic for leisure, exploring the city, and enjoying longer rides.
- Casual riders' peak usage is on weekends, during warmer months, and during midday/afternoon hours.
- Casual riders prefer longer rides, suggesting they value the journey and the experience.

5. Discussion

This section interprets the analysis results, discusses their implications, and addresses the limitations of the analysis.

5.1 Interpretation of Results

The analysis reveals distinct differences in how casual riders and annual members utilize Cyclistic bikes.

- Annual members are frequent users, likely incorporating Cyclistic into their daily routine. This is supported by the higher number of rides taken by members and the peak usage during commuting hours on weekdays.
- Casual riders, on the other hand, appear to use Cyclistic for recreational purposes. They take longer trips, particularly on weekends and during warmer months, suggesting they use the service for fun and for exploration.
- The preference of both rider groups for classic and electric bikes over electric scooters suggests that Cyclistic should ensure there are plenty of these bike types available.

5.2 Comparison of Casual Rider and Member Behavior

Feature	Casual Riders	Annual Members
Ride Frequency	Lower	Higher
Average Duration	Longer	Shorter
Peak Usage Time	Midday/Afternoon, Weekends	Commuting Hours, Weekdays

Feature	Casual Riders	Annual Members
Seasonal Variation	Stronger increase in warmer months	Increase in warmer months
Purpose	Recreational, Leisure	Commuting, Daily Routines
Bike Preference	Classic and Electric Bikes	Classic and Electric Bikes

5.3 Potential Conversion Opportunities

The differences in usage patterns suggest several opportunities to convert casual riders into annual members:

- **Targeted Memberships:** Offer memberships focused on casual rider behavior, such as "Weekend Warrior Pass," "Summer Fun Pass," or "Explorer Membership".
- **Promotional Offers:** Offer promotional offers that align with casual rider preferences, such as weekend discounts, seasonal bundles, or offering the first ride for free.
- **Marketing Focus:** Emphasize the recreational benefits of Cyclistic in marketing campaigns, highlighting the enjoyment of longer rides and city exploration. These marketing campaigns can take place on social media, on sponsored blog posts, or through encouraging user-generated content.

5.4 Limitations of the Analysis

The analysis is subject to the following limitations:

- **Missing Data:** A significant number of records had missing data for start and end station names and IDs, preventing geographic analysis.
- **Imprecise GPS Data:** A subset of the data had imprecise latitude and longitude data, which could introduce bias. This was further reason why the geographical analysis was not performed.
- **Outliers:** The data contained a large number of extremely short and extremely long rides, which were likely due to system errors and required filtering.
- **Data Inconsistencies:** There were inconsistencies in station names and IDs, which could not be resolved without an official list from Cyclistic.

5.5 Suggestions for Future Research

Further research could explore the following:

- **Geographical Analysis:** With more precise GPS data, could do geographical analysis to identify popular routes and inform targeted marketing further.
- **Outlier Analysis:** Investigate the causes of extremely short and extremely long rides to identify potential system errors or unusual usage patterns.
- **External Factors:** Analyze the impact of other factors such as weather conditions, local events, and infrastructure changes on Cyclistic usage.

6. Conclusion

This section summarizes the key findings of the analysis and restates the main insights regarding Cyclistic rider behavior.

6.1 Summary of Key Findings

- Annual members take more rides, suggesting Cyclistic is part of their daily routine, perha[s in commuting to work.
- Casual riders take longer rides, showing a preference for recreational use.
- Member have maximum usage during commuting hours on weekdays, while casual rider have maximum usage during midday and afternoons, especially on weekends.
- Both rider groups show increased usage in warmer months.
- Both rider groups prefer classic and electric bikes.

6.2 Restatement of Main Insights

- Cyclistic identifies two user groups with different needs and usage patterns.
- Annual members use the service for commuting, while casual riders use it for leisure and recreation.
- The differences found in this analysis are to be used for developing effective marketing strategies to convert casual riders into annual members.
- Targeted marketing strategies and digital media campaigns can effectively target casual riders and highlight the benefits of annual memberships.

7. Appendix

This section provides supplementary materials to support the analysis.

7.1 Complete R Code Used for Analysis

```
library(tidyverse)
library(showtext)

####Formatting & Values --------------------------------------------------------
font_add_google("Questrial", "questrial") #Fonts need to be downloaded from Google Fonts
font_add_google("Nunito", "nunito")
showtext_auto()

background_color <- "#f5f5f5" #Plot design colors
text_color <- "#3f4252"

rider_colors <- c("casual" = "#F9918A", "member" = "#33CCD0")
rider_labels <- c("Casual Riders", "Members")

bike_types <- c("electric_scooter", "electric_bike", "classic_bike")
bike_colors <- c("electric_scooter" = "#FF7F50", "electric_bike" = "#33C860", "classic_bike" = "#81B0FF

hour_labels <- setNames(format(strptime(0:23, format = "%H"), "%l %p"), 0:23) #Formatting for x-axis of
breaks_seq <- seq(0, 23, by = 4)
labels_seq <- hour_labels[as.character(breaks_seq)]

day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday") #Set day o
bike_labels <- c("Classic Bike", "Electric Bike", "Electric Scooter") # Labels for rideable type plot

#### Dataframe Creation --------------------------------------------------------
```

```
#Full dataframe
df <- read_csv("C:/Users/Administrator/Downloads/cleaned_bikeshare_data.csv")
df <- df |> select(c(ride_id, member_casual, rideable_type, started_at, ended_at, ride_duration_minutes

df$started_at <- ymd_hms(df$started_at, tz="UTC") #Extract time components
df <- df |>
  mutate(
    hour = hour(started_at),
    weekday = weekdays(started_at),
    month = month(started_at)
  )
day_order <- c("Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday") #Order wee
df$weekday <- factor(df$weekday, levels = day_order, ordered = TRUE)

#Rider Type
rider_type <- df |>
  group_by(member_casual) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

#Ride Preferences
ride_preference <- df |>
  group_by(member_casual, rideable_type) |>
  summarize(ride_count = n())

#Hourly
hourly_rider_type <- df |>
  group_by(member_casual, hour) |>
  filter(!is.na(hour)) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

#Weekday
weekday_rider_type <- df |>
  group_by(member_casual, weekday) |>
  filter(!is.na(hour)) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes)) |>
  mutate(start_weekday = factor(start_weekday, levels = day_order, ordered = TRUE))

#Monthly
monthly_rider_type <- df |>
  group_by(member_casual, month) |>
  filter(!is.na(hour)) |>
  summarize(ride_count = n(),
            avg_duration = mean(ride_duration_minutes))

rm(df) #save space by removing the original dataframe from memory

#### Summary Statistics -------------------------------------------------------

duration_stats <- df |>
  group_by(member_casual) |>
  summarize(
    n = n(),
    avg_duration = mean(ride_duration_minutes),
    median_duration = median(ride_duration_minutes),
    sd_duration = sd(ride_duration_minutes),
    min_duration = min(ride_duration_minutes),
    max_duration = max(ride_duration_minutes),
    q25_duration = quantile(ride_duration_minutes, 0.25),
    q75_duration = quantile(ride_duration_minutes, 0.75)
  )

#### Visualizations -------------------------------------------------------

#Total Number of Rides
ggplot(rider_type, aes(x = member_casual, y = ride_count, fill = member_casual)) +
  geom_bar(stat = "identity", color = text_color) +
  labs(
    title = "Total Number of Rides",
    subtitle = "Casual Riders vs. Members",
    fill = "Rider Type"
  ) +
  geom_text(aes(label = c("1,495,902","2,609,532"), vjust = 2), color = background_color, size = 5) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  theme_void() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color)
  )

#Total Average Duration
ggplot(rider_type, aes(x = member_casual, y = avg_duration, fill = member_casual)) +
  geom_bar(stat = "identity", color = text_color) +
  labs(
    title = "Total Average Duration in Minutes",
    subtitle = "Casual Riders vs. Members",
    fill = "Rider Type"
  ) +
  geom_text(aes(label = c("23:06", "12:15"), vjust = 2), color = background_color, size = 5) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  theme_void() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color)
  )

#Rideable Type
ggplot(ride_preference, aes(fill = rideable_type, y = member_casual, x = ride_count)) +
  geom_bar(position = "dodge", stat = "identity", color = text_color, alpha = 0.8) +
  labs(
    title = "Bike Preferences",
    subtitle = "Casual Riders vs. Members",
    fill = "Bike Type",
    y = element_blank(),
    x = "Number of Rides"
  ) +
  scale_fill_manual(values = bike_colors, labels = bike_labels) +
  scale_x_continuous(labels = scales::comma,) +
  scale_y_discrete(labels = c("Casual", "Member")) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color)
  )

#Hourly Ride Count
ggplot(hourly_rider_type, aes(x = hour, y = ride_count, fill = member_casual)) +
  geom_col(position = "dodge", color = text_color, alpha = 0.8) +
  labs(
    title = "Ride Distribution by Hour",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Number of Rides",
    fill = "Rider Type"
  ) +
  scale_x_continuous(breaks = breaks_seq, labels = labels_seq) +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  facet_wrap(~member_casual,labeller = as_labeller(c("casual" = "Casual Riders", "member" = "Members"))
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    strip.text = element_text(colour = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
    )

#Hourly Average Duration
ggplot(hourly_rider_type, aes(x = hour, y = avg_duration, color = member_casual)) +
  geom_line(linewidth = 2.5, color = text_color, aes(group = member_casual)) +
  geom_line(linewidth = 2, aes(group = member_casual)) +
  geom_point(size = 4) +
  geom_point(size = 4, color = text_color, shape = 1) +
  labs(
    title = "Average Duration by Hour",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Average Duration (minutes)",
    color = "Rider Type"
  ) +
  scale_x_continuous(breaks = breaks_seq, labels = labels_seq) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = rider_colors, labels = rider_labels) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )

#Weekday Ride Count
ggplot(weekday_rider_type, aes(x = start_weekday, y = ride_count, fill = member_casual)) +
  geom_col(position = "dodge", color = text_color, alpha = 0.8) +
  labs(
    title = "Ride Distribution by Weekday",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Number of Rides",
    fill = "Rider Type") +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  facet_wrap(~member_casual,labeller = as_labeller(c("casual" = "Casual Riders", "member" = "Members"))
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    strip.text = element_text(colour = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )

#Weekday Average Duration
ggplot(weekday_rider_type, aes(x = start_weekday, y = avg_duration, color = member_casual)) +
  geom_line(linewidth = 2.5, color = text_color, aes(group = member_casual)) +
  geom_line(linewidth = 2, aes(group = member_casual)) +
  geom_point(size = 4) +
  geom_point(size = 4, color = text_color, shape = 1) +
  labs(
    title = "Average Duration by Weekday",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Average Duration (minutes)",
    color = "Rider Type"
  ) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = rider_colors, labels = rider_labels) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )

#Monthly Ride Count
ggplot(monthly_rider_type, aes(x = month, y = ride_count, fill = member_casual)) +
  geom_col(position = "dodge", color = text_color, alpha = 0.8) +
  labs(
    title = "Ride Distribution by Month",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Number of Rides",
    fill = "Rider Type") +
  scale_x_discrete(limits = month.name[1:12], labels = month.name[1:12]) +
  scale_y_continuous(labels = scales::comma) +
  scale_fill_manual(values = rider_colors, labels = rider_labels) +
  facet_wrap(~member_casual,labeller = as_labeller(c("casual" = "Casual Riders", "member" = "Members"))
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    strip.text = element_text(colour = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )

#Monthly Average Duration
ggplot(monthly_rider_type, aes(x = month, y = avg_duration, color = member_casual)) +
  geom_line(linewidth = 2.5, color = text_color, aes(group = member_casual)) +
  geom_line(linewidth = 2, aes(group = member_casual)) +
  geom_point(size = 4) +
  geom_point(size = 4, color = text_color, shape = 1) +
  labs(
    title = "Average Duration by Month",
    subtitle = "Comparison of Members vs. Casual Riders",
    x = element_blank(),
    y = "Average Duration (minutes)",
    color = "Rider Type"
  ) +
  scale_x_discrete(limits = month.name[1:12], labels = month.name[1:12]) +
  scale_y_continuous(labels = scales::comma) +
  scale_color_manual(values = rider_colors, labels = rider_labels) +
  theme_minimal() +
  theme(
    plot.margin = margin(t = 10, r = 10, b = 10, l = 10),
    plot.title = element_text(family = "questrial", face = "bold", color = text_color),
    text = element_text(family = "nunito", color = text_color),
    plot.background = element_rect(fill = background_color, color = background_color),
    panel.background = element_rect(fill = background_color, color = background_color),
    axis.text.x = element_text(angle = 45, hjust = 1),
    axis.text.y = element_blank()
  )
```

7.2 Detailed Data Dictionaries

1. Original Dataframe (cleaned_bikeshare_data.csv)

- **ride_id:**
  - Description: Unique identifier for each ride.
  - Data Type: Character (string).
  - Usage: Used as a unique identifier.
- **member_casual:**
  - Description: Indicates the type of rider (member or casual).
  - Data Type: Character (string).
  - Possible Values: "member", "casual".
  - Usage: Categorical variable for rider segmentation.
- **rideable_type:**
  - Description: Type of bike used for the ride.
  - Data Type: Character (string).
  - Possible Values: "electric_scooter", "electric_bike", "classic_bike".
  - Usage: Categorical variable for bike type analysis.
- **started_at:**
  - Description: Date and time when the ride started.
  - Data Type: POSIXct (date-time).
  - Usage: Used to extract hour, weekday, and month.
- **ended_at:**
  - Description: Date and time when the ride ended.
  - Data Type: POSIXct (date-time).
  - Usage: Used to calculate ride duration.
- **ride_duration_minutes:**
  - Description: Duration of the ride in minutes.
  - Data Type: Numeric.
  - Usage: Used for duration analysis.

2. Engineered Features in `df` Dataframe

- **hour:**
  - Description: Hour of the day when the ride started (0-23).
  - Data Type: Integer.

  - Usage: Used for hourly analysis.

- **weekday:**
  - Description: Day of the week when the ride started.
  - Data Type: Factor (ordered).
  - Possible Values: "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday".
  - Usage: Used for weekday analysis.

- **month:**
  - Description: Month of the year when the ride started.
  - Data Type: Integer.
  - Possible Values: 1-12.
  - Usage: Used for monthly analysis

3. Summary Dataframes

- **rider_type:**
  - **member_casual:** Rider type ("member", "casual").
  - **ride_count:** Total number of rides for each rider type.
  - **avg_duration:** Average ride duration in minutes for each rider type.

- **ride_preference:**
  - **member_casual:** Rider type ("member", "casual").
  - **rideable_type:** Type of bike ("electric_scooter", "electric_bike", "classic_bike").
  - **ride_count:** Number of rides for each rider type and bike type combination.

- **hourly_rider_type:**
  - **member_casual:** Rider type ("member", "casual").
  - **hour:** Hour of the day (0-23).
  - **ride_count:** Number of rides for each rider type and hour combination.
  - **avg_duration:** Average ride duration in minutes for each rider type and hour combination.

- **weekday_rider_type:**
  - **member_casual:** Rider type ("member", "casual").
  - **weekday:** Day of the week.
  - **ride_count:** Number of rides for each rider type and weekday combination.
  - **avg_duration:** Average ride duration in minutes for each rider type and weekday combination.
  - **start_weekday:** Same as weekday, but as factor ordered.

- **monthly_rider_type:**
  - **member_casual:** Rider type ("member", "casual").
  - **month:** Month of the year.
  - **ride_count:** Number of rides for each rider type and month combination.
  - **avg_duration:** Average ride duration in minutes for each rider type and month combination.

- **duration_stats:**
  - **member_casual:** Rider type ("member", "casual").
  - **n:** Number of rides.
  - **avg_duration:** Mean ride duration.
  - **median_duration:** Median ride duration.
  - **sd_duration:** Standard deviation of ride duration.
  - **min_duration:** Minimum ride duration.

    - **max_duration:** Maximum ride duration.
    - **q25_duration:** 25th percentile of ride duration.
    - **q75_duration:** 75th percentile of ride duration.

4. R Script Variables

- **background_color:** Hex color code for plot background.
- **text_color:** Hex color code for plot text.
- **rider_colors:** Named vector of colors for rider types.
- **rider_labels:** Named vector of labels for rider types.
- **bike_types:** Vector of bike types.
- **bike_colors:** Named vector of colors for bike types.
- **hour_labels:** Named vector of formatted hour labels.
- **breaks_seq:** Sequence of hour breaks for x-axis.
- **labels_seq:** Labels for hour breaks.
- **day_order:** Vector of weekdays in desired order.
- **bike_labels:** Vector of bike type labels.

5. Libraries Used

- **tidyverse:** For data manipulation and visualization.
- **showtext:** For using custom fonts.

7.3 Supplementary Materials

- Case Study Background Information
- Original Data Source
- Google Slides Presentation