

Cyclist User Behavior Analyst 2024

Fachreza Dewanto

2025-07-17

Contents

Project Introduction	1
1. Data Preparation & Overview	1
2. Descriptive Analysis	2
3. Revenue Insight	9
4. Outliers Profiling: Ride Length	14
5. Rides Duration Behavior	15
6. Station Insight	20
7. Insight Recommendation	21
Interactive Dashboard	26
Limitations & Future Works	26
References	26
About the Author	26

Project Introduction

This report presents an analysis of Cyclist ride data spanning Q1–Q4 2024, prepared for the Marketing Department to support future business decisions. The goal is to uncover user behavior patterns, highlight seasonal trends, and estimate ride-level revenue based on membership type and bike preference. The analysis is conducted as part of the Google Data Analytics Professional Certificate capstone, using a hypothetical company (“Cyclist”) and real-world ride data from Divvy Bikes—a bike-sharing service based in Chicago.

***Data source:** The dataset is publicly available under the [Divvy Data License Agreement](#). This analysis is for educational purposes only and is not affiliated with or endorsed by Divvy, Lyft, or the City of Chicago.*

1. Data Preparation & Overview

1.1 Data Cleaning in SQL BigQuery

Twelve monthly datasets from January to December 2024 were extracted using SQL in Google BigQuery. Null values were filtered from key columns `start_station_name` and `end_station_id` to ensure data

integrity. No duplicate entries were detected, and all types were consistent across tables.

Each cleaned monthly dataset was exported as CSV and merged into a single dataset for analysis in R. The final combined file contains distinct 4,208,309 observations.

```
# SQL Sample of dataset cleaning
SELECT *
FROM 'capstone-project-202507.Cyclist.2024-divvy-tripdata'
WHERE start_station_name IS NOT NULL
AND end_station_id IS NOT NULL
```

1.2 Dataset Description and Import

The dataset contains the following columns:

1. `ride_id`: Unique identifier for each ride
2. `rideable_type`: Type of bike used for the ride
3. `started_at`: Start time of the ride
4. `ended_at`: End time of the ride
5. `start_station_name`: Name of the station where the ride started
6. `end_station_name`: Name of the station where the ride ended
7. `start_station_id`: Unique identifier for the start station
8. `end_station_id`: Unique identifier for the end station
9. `start_lat`: Latitude of the start station
10. `start_lng`: Longitude of the start station
11. `end_lat`: Latitude of the end station
12. `end_lng`: Longitude of the end station
13. `member_casual`: Type of membership (member or casual)

```
# Create sample for Github
sample_clean <- combined_clean_data %>%
  slice_sample(n = 4092)

#sample_clean will be used for this structure and logic flow for data analyst. This sample size is 80%
```

We found the dataset for `started_at` and `ended_at` character columns, and convert them to POSIXct date-time format for easier analysis.

```
# Change the started_at and ended_at columns to Date Time format
sample_clean$started_at <- as.POSIXct(sample_clean$started_at, format="%Y-%m-%d %H:%M:%S")
sample_clean$ended_at <- as.POSIXct(sample_clean$ended_at, format="%Y-%m-%d %H:%M:%S")
```

2. Descriptive Analysis

2.a Member Type Distribution

We began by comparing total ride counts between annual members and casual users. A simple bar plot highlights strong seasonal engagement from casual users and consistent usage from members.

```

# Find overall composisiton of type of member, annual & casual
member_composition <- sample_clean %>%
  group_by(member_casual) %>%
  summarise(total_rides = n())

```

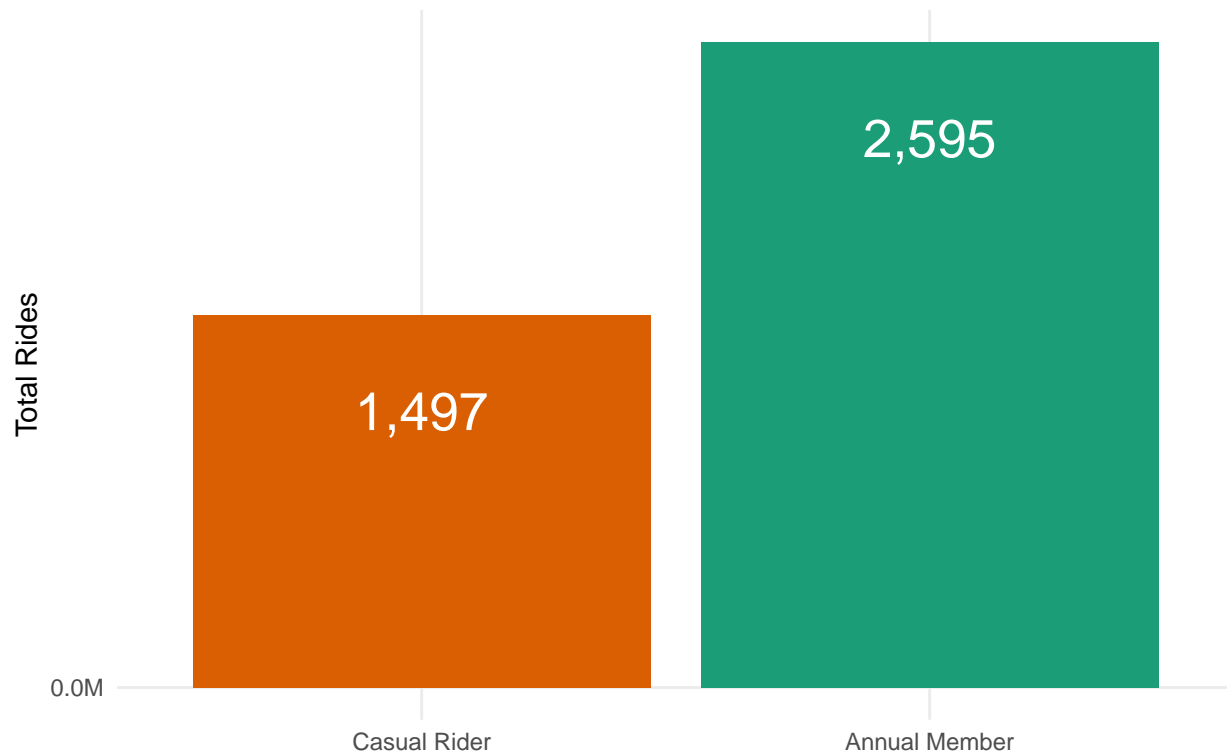
```

# Graph for total rides by membership type
p2a <- ggplot(member_composition,
  aes(x = member_casual,
    y = total_rides,
    fill = member_casual)) +
  geom_col() +
  scale_fill_manual(name = "Membership Type",
    values = c(member = "#1b9e77",
      casual = "#d95f02")) +
  geom_text(aes(label = label_comma()(total_rides)),
    colour = "white", vjust = 3, size = 7) +
  labs(
    title = "Membership Types in 2024",
    subtitle = "Visualizing member type distribution",
    x = NULL,
    y = "Total Rides") +
  scale_x_discrete(labels = c(member = "Annual Member",
    casual = "Casual Rider")) +
  scale_y_continuous(
    breaks = seq(0, max(member_composition$total_rides), by = 500000),
    labels = label_number(scale = 1e-6, suffix = "M", accuracy = 0.1)) +
  theme_minimal() +
  theme(legend.position = "none")
print(p2a)

```

Membership Types in 2024

Visualizing member type distribution



2.b Rideable Type Preference

To explore user preferences, we created a summary of bike types (classic_bike, electric_bike, and electric_scooter) grouped by membership. The resulting plot reveals that casual riders show a stronger preference for classic bikes—especially during weekends.

```
# Find overall distribution ride type based on member type
ride_type_summary <- sample_clean %>%
  group_by(member_casual, rideable_type) %>%
  summarise(total_ride_type = n(), .groups = "drop")
```

```
# Bar chart comparison bike type use by member type
p2b <- ggplot(ride_type_summary,
  aes(x = rideable_type,
    y = total_ride_type,
    fill = member_casual)) +
  geom_col(position = position_dodge()) +
  scale_fill_manual(name = "Membership Type",
    values = c(member = "#1b9e77", casual = "#d95f02")) +
  geom_text(aes(label = label_comma()(total_ride_type),
    position = position_dodge2(width = 0.9), vjust = -0.2, fontface = "bold") +
  labs(title = "Bike Type Usage Distribution by Member Type",
    subtitle = "Visualizing bike type used by users",
    x = NULL, y = "Total Rides") +
  scale_x_discrete(labels = c(classic_bike = "Classic Bike",
```

```

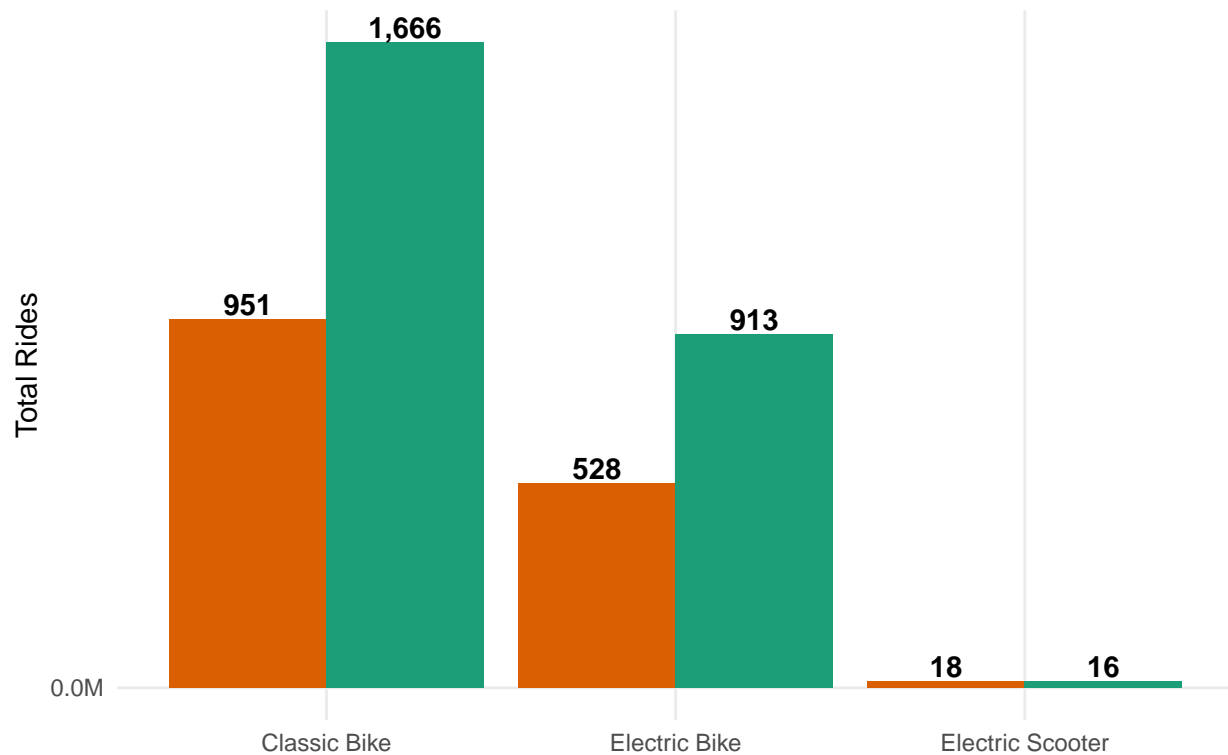
        electric_bike = "Electric Bike",
        electric_scooter = "Electric Scooter")) +
scale_y_continuous(breaks = seq(0, max(ride_type_summary$total_ride_type), by = 500000),
                  labels = label_number(scale = 1e-6, suffix = "M", accuracy = 0.1)) +
theme_minimal() +
theme(legend.position = "none")

print(p2b)

```

Bike Type Usage Distribution by Member Type

Visualizing bike type used by users



2.c Monthly Ride Patterns and Seasonality

We analyzed ride trends across each month of 2024, segmented by membership type. Seasonal variation was evident: casual users showed strong engagement during summer months, while annual members maintained consistent ride activity year-round. A faceted bar chart displays ride volume per month, with seasonal bands for context.

```

#Summarize monthly totals
monthly_summary <- sample_clean %>%
  mutate(month_num = month(started_at),
         month = month(started_at, label = TRUE, abbr = TRUE)) %>%
  group_by(month_num, month, member_casual) %>%
  summarise(total_rides = n(), .groups = "drop")

```

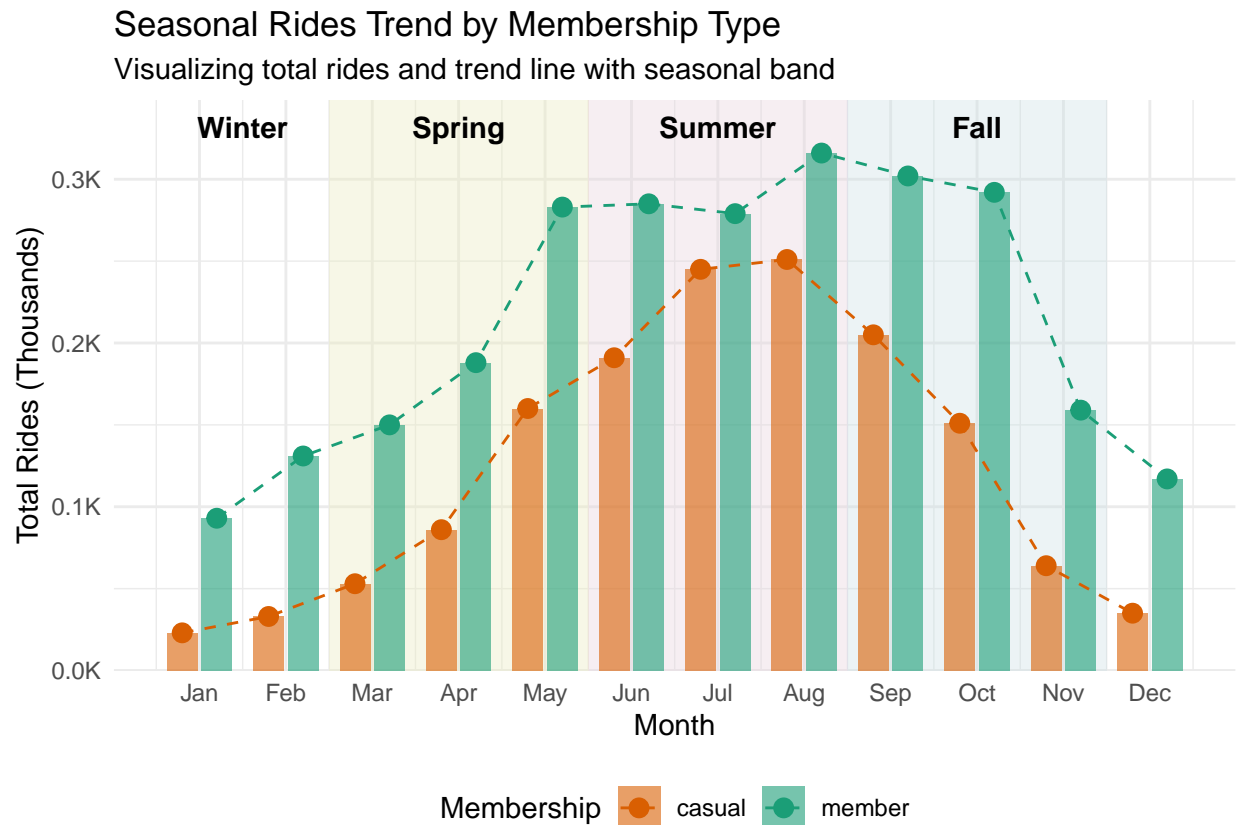
```

#Define seasons and their numeric ranges
seasons <- tibble( season   = c("Winter", "Spring", "Summer", "Fall"),
                  start_mon = c(12, 3, 6, 9),
                  end_mon   = c(2, 5, 8, 11),
                  x_label   = c(1.5, 4, 7, 10)) # midpoint for label

#Compute peak y so we can place season labels above data
y_max <- max(monthly_summary$total_rides)
#Plot: season bands, bars, lines/points, labels
p2c <- ggplot() +
  # Season bands (semi-transparent rectangles)
  geom_rect(data = seasons, inherit.aes = FALSE,
            aes(xmin = start_mon - 0.5, xmax = end_mon + 0.5,
                ymin = -Inf, ymax = Inf),
            fill = c("#D3E4CD", "#FFF3B0", "#F6C6EA", "#C4DEF6"),
            alpha = 0.2) +
  # Bars (dodge by membership)
  geom_col(data = monthly_summary,
            aes(x = month_num,
                y = total_rides,
                fill = member_casual),
            position = position_dodge(width = 0.8), width = 0.7, alpha = 0.6) +
  # Trend lines and points
  geom_line(data = monthly_summary,
            aes(x = month_num,
                y = total_rides,
                color = member_casual,
                group = member_casual),
            position = position_dodge(width = 0.8),
            linetype = "dashed") +
  geom_point(data = monthly_summary,
             aes(x = month_num,
                 y = total_rides,
                 color = member_casual),
             position = position_dodge(width = 0.8), size = 3) +
  # Season labels
  geom_text(data = seasons, inherit.aes = FALSE,
            aes(x = x_label, y = y_max * 1.05, label = season),
            size = 4, fontface = "bold") +
  # Scales, labs, theme
  scale_x_continuous(breaks = 1:12,
                     labels = month.abb) +
  scale_y_continuous(labels = label_number(scale = 1e-3, suffix = "K"),
                     expand = expansion(mult = c(0, 0.05))) +
  scale_fill_manual(name = "Membership",
                    values = c(member = "#1b9e77", casual = "#d95f02")) +
  scale_color_manual(name = "Membership",
                     values = c(member = "#1b9e77", casual = "#d95f02")) +
  labs(title = "Seasonal Rides Trend by Membership Type",
       subtitle = "Visualizing total rides and trend line with seasonal band",
       x = "Month",
       y = "Total Rides (Thousands)") +
  theme_minimal() +
  theme(legend.position = "bottom")

```

```
print(p2c)
```



2.d Ride Duration Analysis

We explored how much time users spend on rides by grouping ride lengths into behavioral buckets (e.g., 0–5 min, 5–15 min, etc.). After we create bar plot we found out most of user fall-in under 30 minutes internal use.

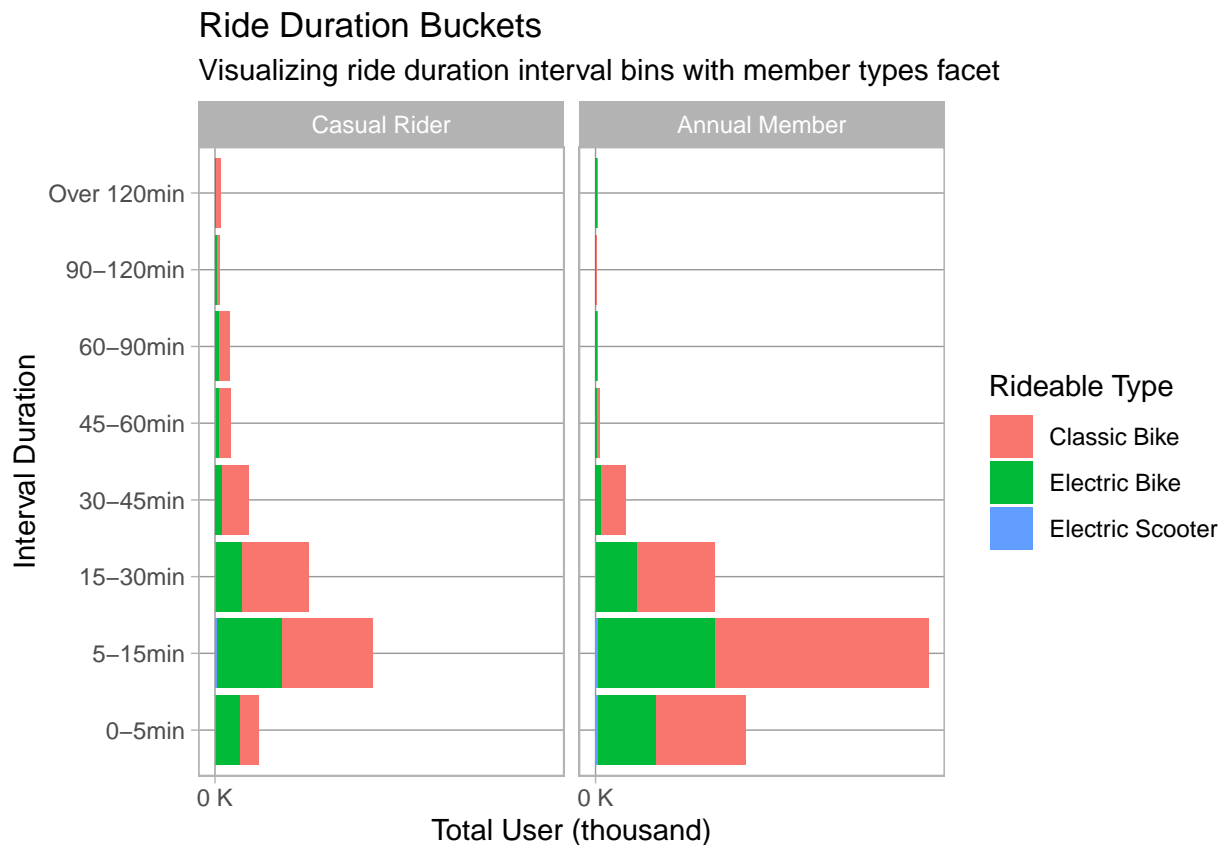
```
# Create table with weekday, ride_length and day_type variables
ride_duration <- sample_clean %>%
  mutate(weekday = wday(started_at, label = TRUE),
         ride_length = as.numeric(difftime(ended_at, started_at, units = "mins")),
         day_type = if_else(weekday(started_at) %in% c(1, 7), "Weekend", "Weekday")
  )

user_rides_interval <- ride_duration %>%
  mutate(duration_interval = cut(ride_length,
                                breaks = c(0, 5, 15, 30, 45, 60, 90, 120, Inf),
                                labels = c("0-5min", "5-15min", "15-30min", "30-45min", "45-60min",
                                           "60-90min", "90-120min", "Over 120min")),
         rideable_type = recode(rideable_type, classic_bike = "Classic Bike",
                                electric_bike = "Electric Bike",
                                electric_scooter = "Electric Scooter")) %>%
  filter(!is.na(duration_interval)) %>%
  count(member_casual, weekday, rideable_type, duration_interval, name = "total_user")
```

```

# Bar chart user rides interval
p2d <- ggplot(user_rides_interval,
  aes(x = total_user, y = duration_interval, fill = rideable_type )) +
  geom_col() +
  scale_x_continuous(labels = label_number(scale = 1e-3, suffix = " K"),
    breaks = seq(0,1500000, by = 300000 ),
    minor_breaks = seq(0, 1500000, by = 100000))+
  labs(title = "Ride Duration Buckets",
    subtitle = "Visualizing ride duration interval bins with member types facet",
    x = "Total User (thousand)",
    y = "Interval Duration",
    fill = "Rideable Type") +
  theme_light()+
  theme(panel.grid = element_line(color = "grey60"),
    panel.grid.minor = element_line(color = "grey90"))+
  facet_grid(~member_casual,
    labeller = labeller(member_casual = c(casual = "Casual Rider",
      member = "Annual Member"))))
print(p2d)

```



3. Revenue Insight

Using reference rates and unlock fees from [Divvy Pricing] (<https://divvybikes.com/pricing>), we summarize duration category to real life pricing plan.

Revenue calculations in this segment exclude capped pricing rules for member e-bike rides between 31–45 minutes and do not account for day-pass packages due to data limitations.

Annual Member revenue only count for their extra time using after 45 minutes for classic bike and after 30 minutes with e-bike.

```
# Dataframe for assign_price_tier by creating 3 variable
# free_threshold for each plan tier classic 0-30min, e-bike
# per_min_rate count extra charge by rate
# unlock_fee free for member and other will $1
assign_price_tier <- ride_duration %>%
  mutate(
    free_threshold = case_when(
      rideable_type == "classic_bike" & member_casual == "member" ~ 45, #Not charge until 45min
      rideable_type == "classic_bike" & member_casual == "casual" ~ 0,
      rideable_type == "electric_bike" ~ 0,
      rideable_type == "electric_scooter" ~ 0),
    per_min_rate = case_when(
      rideable_type == "classic_bike" ~ 0.19,
      rideable_type == "electric_bike" & member_casual == "member" ~ 0.19,
      rideable_type == "electric_bike" & member_casual == "casual" ~ 0.44,
      rideable_type == "electric_scooter" & member_casual == "member" ~ 0.31,
      rideable_type == "electric_scooter" & member_casual == "casual" ~ 0.44),
    unlock_fee = case_when(
      member_casual == "member" ~ 0,
      TRUE ~ 1), #Others tier will be charge $1
    extra_minutes = pmax(ride_length - free_threshold, 0),
    ride_revenue = unlock_fee + extra_minutes * per_min_rate
  )

# Dataframe for revenue summary
revenue_summary <- assign_price_tier %>%
  group_by(member_casual, rideable_type, day_type) %>%
  summarise(total_rides = n(),
            breach_rate = mean(extra_minutes > 0), #percentile
            avg_revenue = mean(ride_revenue), #in USD
            total_revenue = sum(ride_revenue), #in USD
            .groups = "drop")

knitr::kable(revenue_summary,
             caption = "Revenue Summary")
```

Table 1: Revenue Summary

member_casual	rideable_type	day_type	total_rides	breach_rate	avg_revenue	total_revenue
casual	classic_bike	Weekday	574	1.0000000	5.9312503	3404.537667
casual	classic_bike	Weekend	377	1.0000000	6.5095296	2454.092667

member_casual	rideable_type	day_type	total_rides	breach_rate	avg_revenue	total_revenue
casual	electric_bike	Weekday	361	1.0000000	7.4790711	2699.944667
casual	electric_bike	Weekend	167	1.0000000	9.9026228	1653.738000
casual	electric_scooter	Weekday	12	1.0000000	4.6678889	56.014667
casual	electric_scooter	Weekend	6	1.0000000	12.1332222	72.799333
member	classic_bike	Weekday	1251	0.0151878	0.3120496	390.374000
member	classic_bike	Weekend	415	0.0096386	0.0141012	5.852000
member	electric_bike	Weekday	725	1.0000000	2.0652520	1497.307667
member	electric_bike	Weekend	188	1.0000000	2.6178564	492.157000
member	electric_scooter	Weekday	15	1.0000000	2.3174222	34.761333
member	electric_scooter	Weekend	1	1.0000000	3.9318333	3.931833

breach_rate showing the rate of member type to breach the free time threshold in percentile.

3.a Monthly Revenue Trends

We summarized total revenue per month across bike types and membership status. Revenue spikes in July and August correlated with increased casual usage of classic bikes, suggesting summer promotions could be optimized around this times.

```
# Summarize monthly revenue
revenue_monthly <- assign_price_tier %>%
  mutate(month_num = month(started_at),
         month = month(started_at, label = TRUE, abbr = TRUE)) %>%
  group_by(month_num, month, member_casual, rideable_type) %>%
  summarise(revenue_by_month = sum(ride_revenue, na.rm = TRUE),
            .groups = "drop")
```

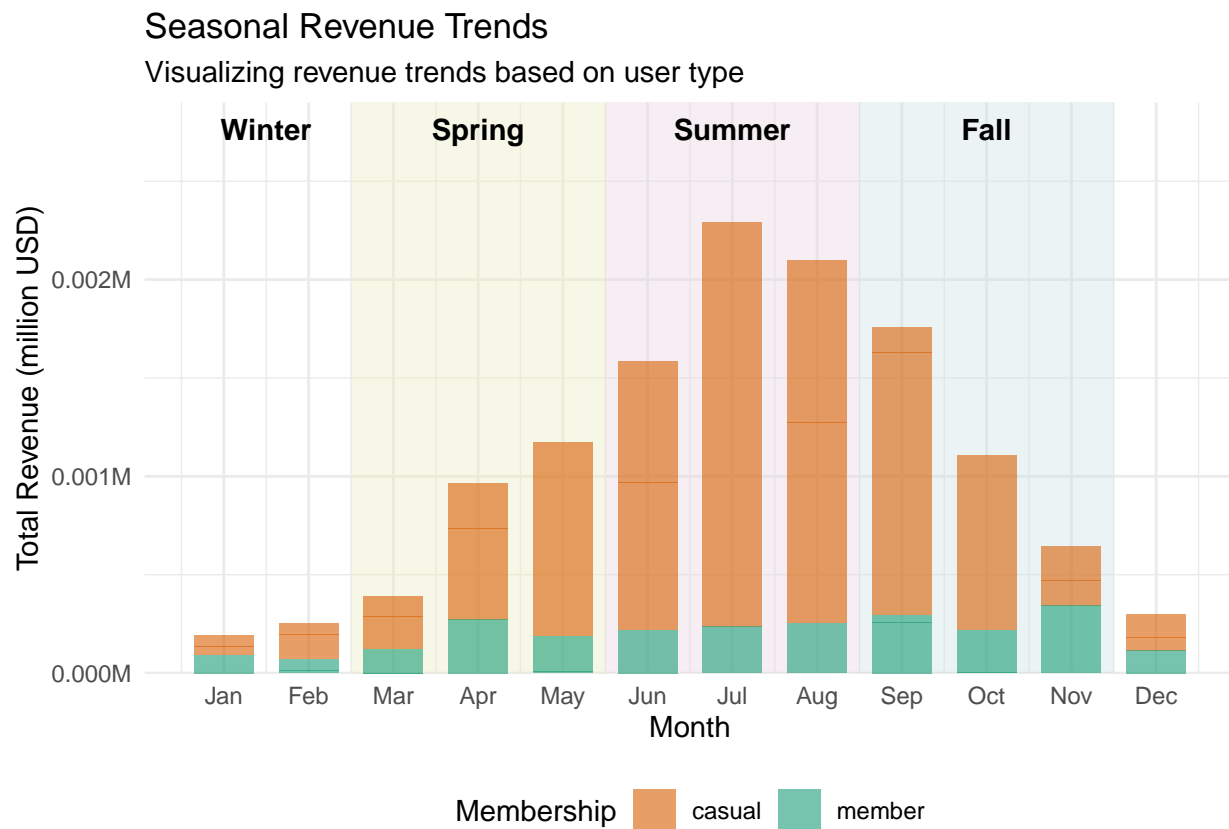
```
# Compute peak y so we can place season labels above data
y_max_revenue <- max(revenue_monthly$revenue_by_month)

#Plot: season bands + bars + labels
p3a1 <- ggplot() +
  # Season bands
  geom_rect( data = seasons, inherit.aes = FALSE,
            aes(xmin = start_mon - 0.5,
                xmax = end_mon + 0.5,
                ymin = -Inf,
                ymax = Inf),
            fill = c("#D3E4CD", "#FFF3B0", "#F6C6EA", "#C4DEF6"),
            alpha = 0.2) +
  geom_col(data = revenue_monthly,
          aes(x = month_num,
              y = revenue_by_month,
              fill = member_casual,
              group = member_casual),
          width = 0.7, alpha = 0.6) +
  #Season labels
  geom_text(data = seasons, inherit.aes = FALSE,
          aes(x = x_label, y = y_max_revenue * 2, label = season),
          size = 4, fontface = "bold") +
  #Scales, labs, theme
```

```

scale_x_continuous(breaks = 1:12,
  labels = month.abb, "Month") +
scale_y_continuous(labels = label_number(scale = 1e-6, suffix = "M"),
  expand = expansion(mult = c(0, 0.05))) +
scale_fill_manual(name = "Membership",
  values = c(member = "#1b9e77", casual = "#d95f02")) +
labs(title = "Seasonal Revenue Trends",
  subtitle = "Visualizing revenue trends based on user type",
  y = "Total Revenue (million USD)") +
theme_minimal() +
theme(legend.position = "bottom")
print(p3a1)

```



```

# Faceting by rideable_type
ridetype_facetlabel <- c(classic_bike = "Classic Bike",
  electric_bike = "Electric Bike",
  electric_scooter = "Electric Scooter")

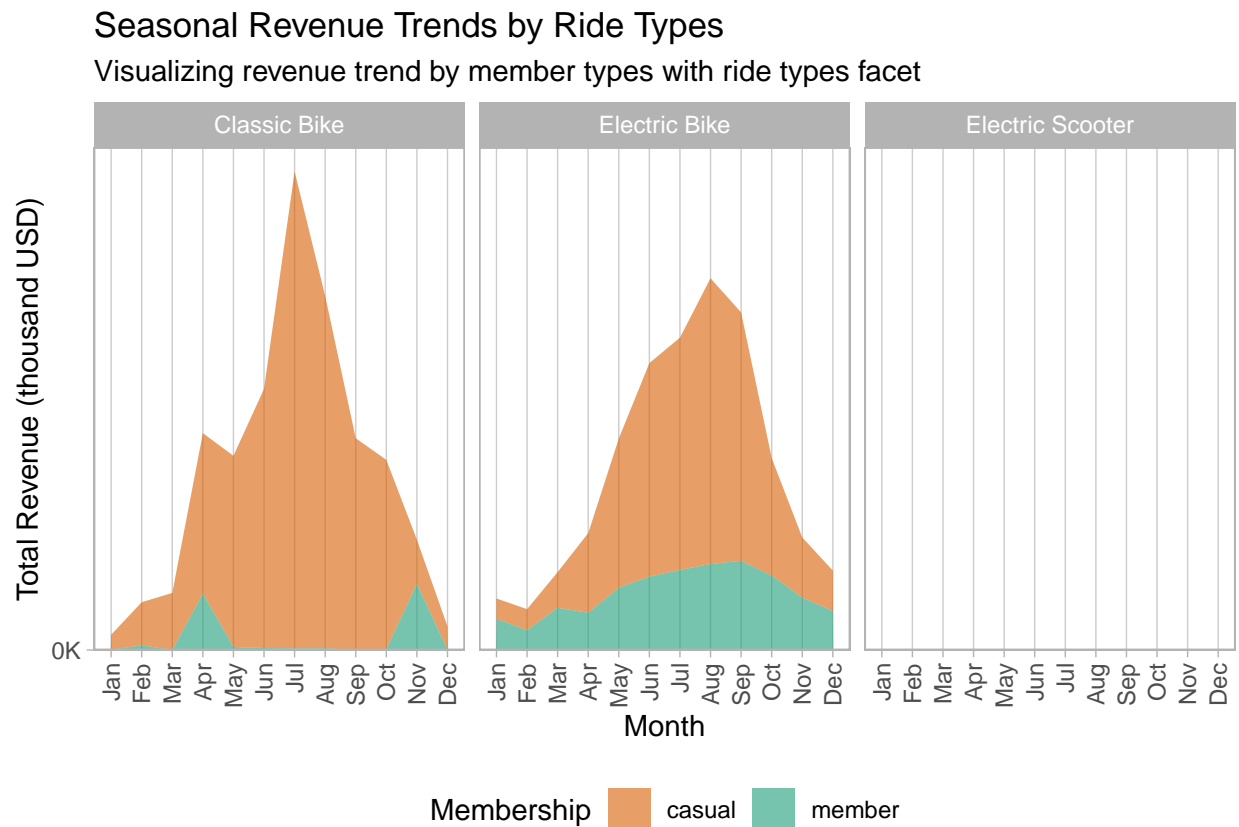
p3a2 <- ggplot(revenue_monthly,
  aes(x = month_num,
    y = revenue_by_month,
    fill = member_casual)) +
geom_area(alpha = 0.6) +
facet_grid(~ rideable_type,
  labeller = labeller(rideable_type = ridetype_facetlabel)) +

```

```

labs(title = "Seasonal Revenue Trends by Ride Types",
     subtitle = "Visualizing revenue trend by member types with ride types facet",
     x = "Month",
     y = "Total Revenue (thousand USD)") +
scale_x_continuous(breaks = 1:12,
                  labels = month.abb,
                  guide = guide_axis(angle = 90),
                  minor_breaks = NULL) +
scale_y_continuous(labels = label_number(scale = 1e-3, suffix = "K"),
                  expand = expansion(mult = c(0, 0.05)),
                  breaks = c(0, 2e+5, 4e+5, 6e+5, 8e+5, 10e+5)) +
scale_fill_manual(name = "Membership",
                 values = c(member = "#1b9e77", casual = "#d95f02")) +
theme_light() +
theme(panel.grid = element_line(color = "grey80"),
      panel.grid.minor = element_line(color = "grey95")) +
theme(legend.position = "bottom")
print(p3a2)

```



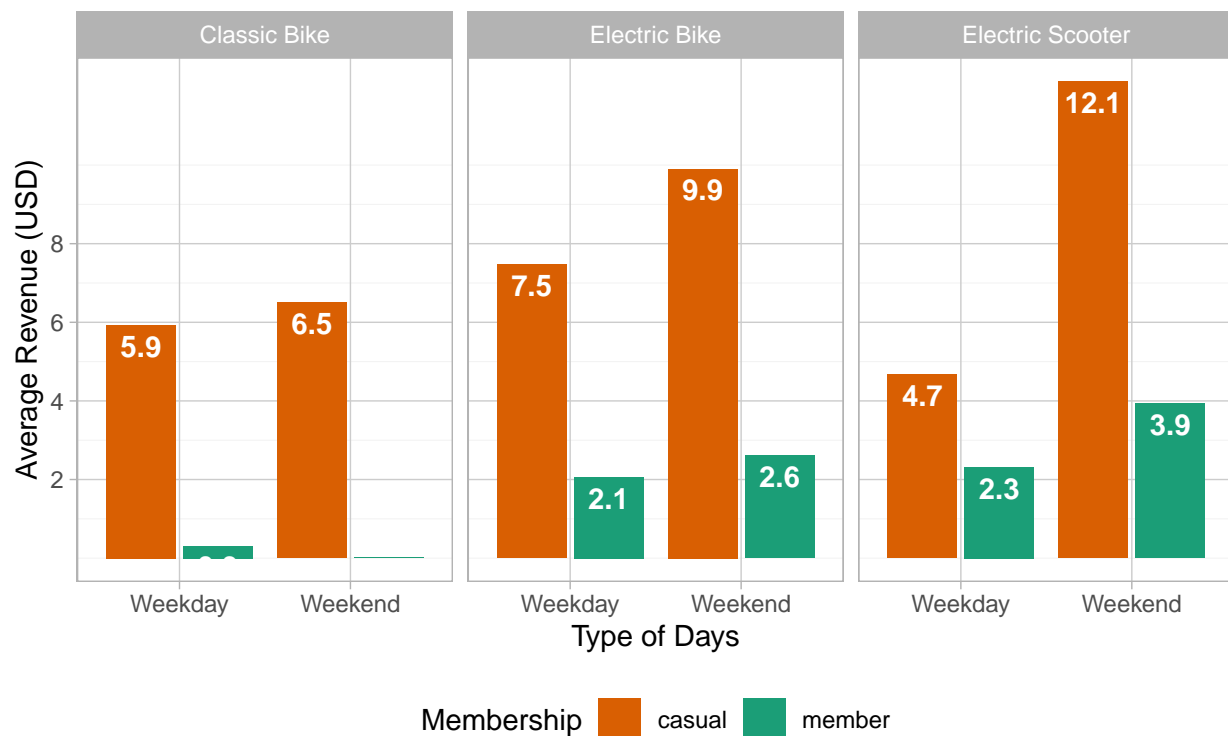
3.b Weekday vs Weekend Revenue

By comparing average revenue on weekdays and weekends, we observed that casual users generate more revenue on weekends, particularly on e-bikes. While members primarily use classic bikes within their free time window.

```
# Plot average revenue based on day type
p3b<- ggplot(revenue_summary,
  aes(x = day_type,
      y = avg_revenue,
      fill = member_casual)) +
  geom_col(position = position_dodge2(0.8)) +
  geom_text(aes(label = label_number(accuracy = 0.1)(avg_revenue)),
    position = position_dodge2(0.9), vjust = 1.5,
    fontface = "bold", color = "white") +
  facet_grid(~ rideable_type,
    labeller = labeller(rideable_type = ridetype_facetlabel)) +
  labs(title = "Weekday vs Weekend Revenue",
    subtitle = "Visualizing average revenue by member types with ride types facet",
    x = "Type of Days",
    y = "Average Revenue (USD)") +
  scale_y_continuous(breaks = c(2, 4, 6, 8)) +
  scale_fill_manual(name = "Membership",
    values = c(member = "#1b9e77", casual = "#d95f02")) +
  theme_light() +
  theme(panel.grid = element_line(color = "grey80"),
    panel.grid.minor = element_line(color = "grey95")) +
  theme(legend.position = "bottom")
print(p3b)
```

Weekday vs Weekend Revenue

Visualizing average revenue by member types with ride types facet



Outliers retained for descriptive analysis but excluded from duration-based summaries.

4. Outliers Profiling: Ride Length

Before conducting average-based analyses, we examined ride duration to detect and remove extreme values that fall outside typical user behavior.

```
ride_duration <- sample_clean %>%
  mutate(weekday = wday(started_at, label = TRUE),
         ride_length = as.numeric(difftime(ended_at, started_at, units = "mins")),
         day_type = if_else(wday(started_at) %in% c(1, 7), "Weekend", "Weekday")
  )
```

4.a Define Threshold Outliers

When we explore for duration of rides, we observe two extreme in ride_length that outside typical user behavior: rides under 1 minute and rides over 120 minute This threshold based on business and user behavior logic: - Is unlikely user will use service under 1 minuter and over 2 hours biking continuously - It flag potential error, special case or apps bug/test

We created a summary of flagged outliers grouped by member_casual. This helped reveal whether casual or member users are more prone to unusual trip lengths.

```
# Outlier filter dataframe for ride_length under 1 minutes and over 120 minutes
ride_outlier <- ride_duration %>%
  mutate(ride_length = as.numeric(difftime(ended_at, started_at, units = "mins")),
         duration_outlier = case_when(
           short_rides = ride_length < 1 ~ "Under 1 min",
           long_rides = ride_length > 120 ~ "Over 120 min",
           TRUE ~ "Valid Ride")) %>%
  count(member_casual, duration_outlier, name = "total_outlier") %>%
  filter(duration_outlier != "Valid Ride")

knitr::kable(ride_outlier,
             caption = "Outliers Summary")
```

Table 2: Outliers Summary

member_casual	duration_outlier	total_outlier
casual	Over 120 min	26
casual	Under 1 min	15
member	Over 120 min	7
member	Under 1 min	30

4.b Hypotheses of Outliers

1. ride_length under 1 minutes

- Oops moment by user (unfamiliar with apps, weathers change)
- Apps testing or bugs
- Apps exploitation (check-in & out for rewards)

2. ride_length over than 120 minutes

- Inattentive user during rides or error when finishing trip/docking
- Apps testing or glitches
- Long ride event or group event rides

4.c Outliers Cleanup

Outliers were removed prior to conducting duration-based analysis. The filtered dataset `ride_duration_clean` includes only rides between 1 and 120 minutes.

```
# Clean outlier in ride_length under 1 min and over 120min
ride_duration_clean <- ride_duration %>%
  filter(ride_length >= 1, ride_length <= 120)
```

5. Rides Duration Behavior

5.a Weekday vs Weekend Behavior

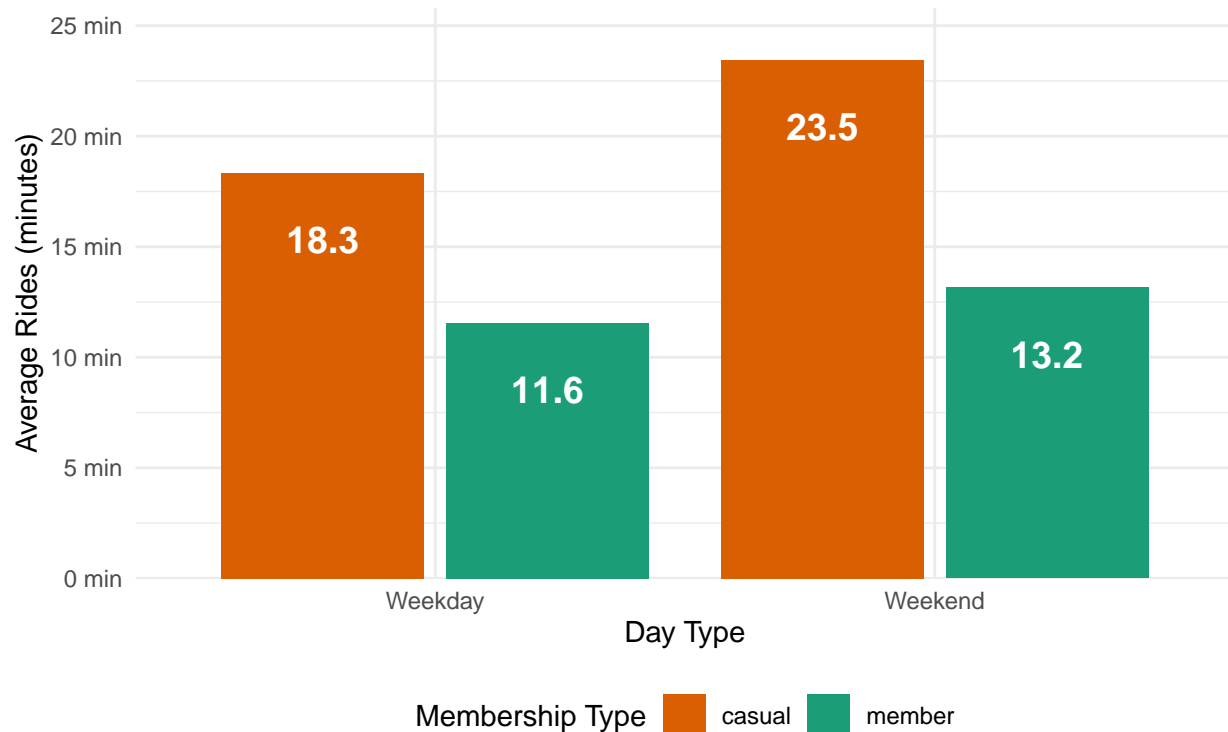
We found that casual riders consistently had longer ride duration than members, especially on weekends. This suggests a leisure-oriented use case, while members tend to commute or use the service more efficiently.

```
# Average duration of rides by membership type based on weekday and weekend
mean_duration_summary <- ride_duration_clean %>%
  mutate(day_of_week = wday(started_at, label = TRUE, abbr = TRUE),
         day_type = ifelse(day_of_week %in% c("Sat", "Sun"), "Weekend", "Weekday")
  ) %>%
  group_by(member_casual, day_type) %>%
  summarise(mean_duration = as.numeric(mean(difftime(
    ended_at, started_at, units = "mins"))),
    .groups = "drop")
```

```
p5a <- ggplot(mean_duration_summary,
  aes(x = day_type, y = mean_duration, fill = member_casual)) +
  geom_col(position = position_dodge2(width = 0.6)) +
  scale_fill_manual(name = "Membership Type",
    values = c(member = "#1b9e77", casual = "#d95f02")) +
  scale_y_continuous(labels = label_number(suffix = " min"),
    expand = expansion(mult = c(0, 0.1))) +
  geom_text(aes(label = label_number(accuracy = 0.1)(mean_duration)),
    position = position_dodge2(width = 0.9),
    color = "white", vjust = 3, size = 5, fontface = "bold") +
  labs(title = "Average Ride Duration Weekday vs Weekend",
    subtitle = "Visualizing compariosn average ride duration by member types",
    x = "Day Type",
    y = "Average Rides (minutes)") +
  theme_minimal() +
  theme(legend.position = "bottom")
print(p5a)
```

Average Ride Duration Weekday vs Weekend

Visualizing comparisn average ride duration by member types



5.b Rides Duration by Day of Week

We further broke down average ride duration by weekday. Casual riders peaked on Saturdays and Sundays, while member usage remained steady, showing their behavior is less time-dependent.

```
# Average duration of rides by membership type based days of the week
mean_duration_eachday <- ride_duration_clean %>%
  mutate(weekday = wday(started_at, label = TRUE),
         ride_length = as.numeric(difftime(ended_at, started_at, units = "mins"))
  ) %>%
  group_by(member_casual, weekday) %>%
  summarise(mean_duration = as.numeric(mean(ride_length, na.rm = TRUE)),
            .groups = "drop")
```

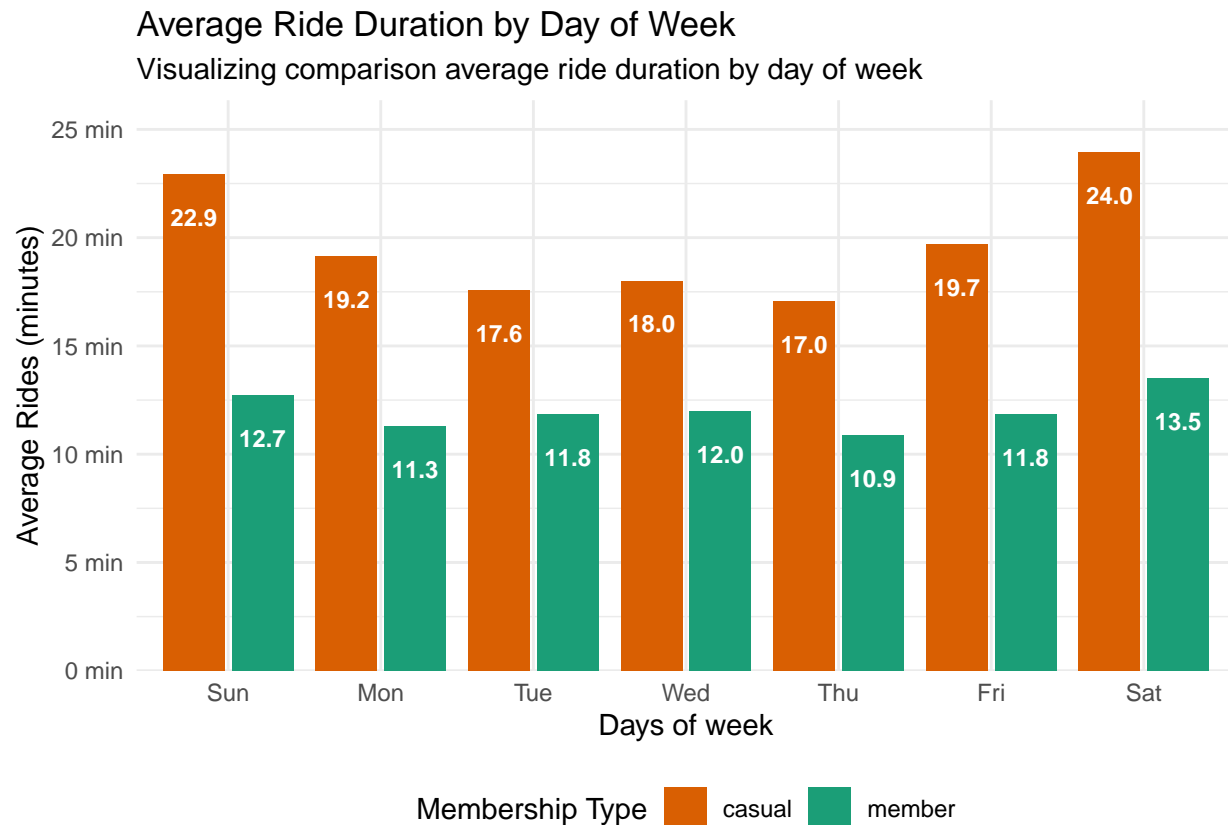
```
p5b <- ggplot(mean_duration_eachday,
  aes(x = weekday, y = mean_duration, fill = member_casual)) +
  geom_col(position = position_dodge2(width = 0.6)) +
  scale_fill_manual(name = "Membership Type",
    values = c(member = "#1b9e77", casual = "#d95f02")) +
  scale_y_continuous(labels = label_number(suffix = " min"),
    expand = expansion(mult = c(0, 0.1))) +
  geom_text(aes(label = label_number(accuracy = 0.1)(mean_duration)),
    position = position_dodge2(width = 0.9),
    color = "white", vjust = 3, size = 3, fontface = "bold") +
  labs(title = "Average Ride Duration by Day of Week",
```



```

    subtitle = "Visualizing comparison average ride duration by day of week",
    x = "Days of week",
    y = "Average Rides (minutes)" +
    theme_minimal() +
    theme(legend.position = "bottom")
print(p5b)

```



5.c Rides Duration by Ride Type

A faceted chart displays average ride duration segmented by `rideable_type`. Classic bikes showed the longest average duration for casual users, while electric bikes were favored for shorter, time-efficient rides.

```

# Average ride duration based on bike types
biketype_duration_summary <- ride_duration_clean %>%
  mutate(day_of_week = wday(started_at, label = TRUE, abbr = TRUE),
         day_type = ifelse(day_of_week %in% c("Sat", "Sun"), "Weekend", "Weekday")) %>%
  group_by(member_casual, rideable_type, day_type) %>%
  summarise(mean_duration = as.numeric(mean(difftime(
    ended_at, started_at, units = "mins"))), .groups = "drop")

```

```

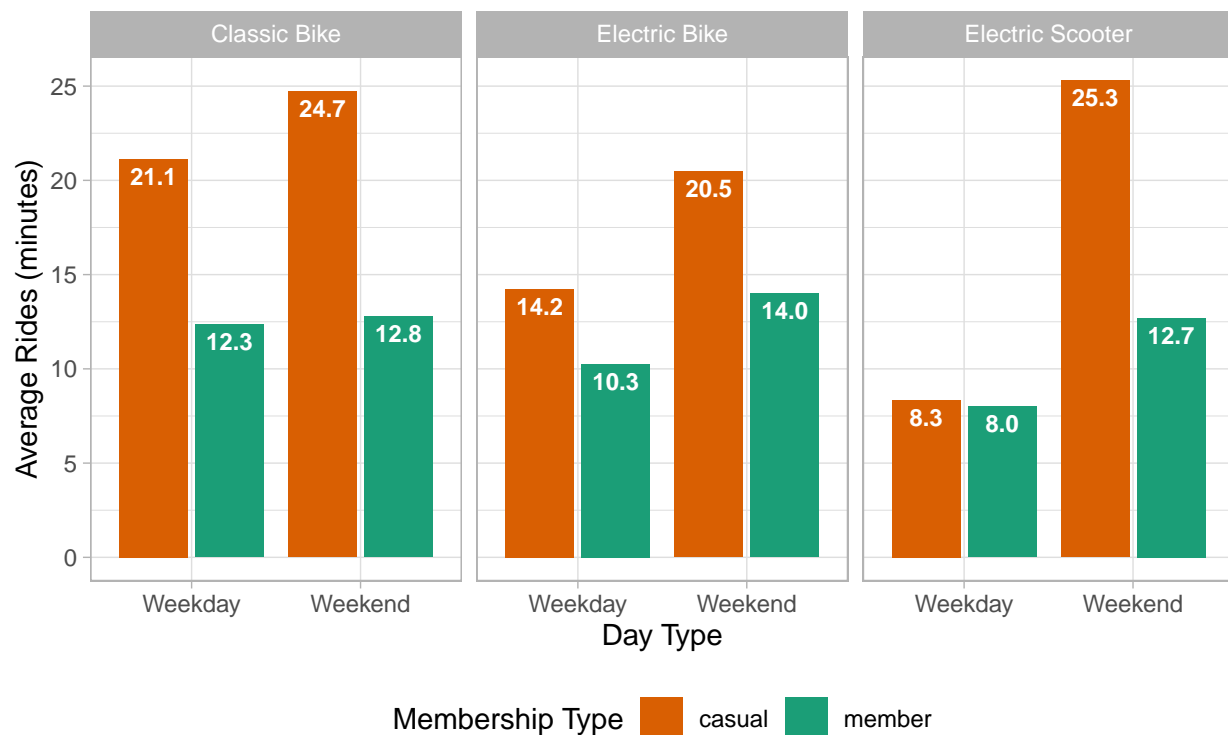
#Create custom label for facet warp rideable_type
biketype_labels <- c(classic_bike = "Classic Bike",
                     electric_bike = "Electric Bike",
                     electric_scooter = "Electric Scooter")

```

```
# Bar chart for average duration based on bike types
p5c <- ggplot(biketype_duration_summary,
  aes(x = day_type, y = mean_duration, fill = member_casual)) +
  geom_col(position = position_dodge2(width = 0.6)) +
  facet_wrap(~rideable_type,
    labeller = labeller(rideable_type = biketype_labels)) +
  scale_fill_manual(name = "Membership Type",
    values = c(member = "#1b9e77", casual = "#d95f02")) +
  geom_text(aes(label = label_number(accuracy = 0.1)(mean_duration)),
    position = position_dodge2(width = 0.9),
    color = "white", vjust = 1.5, size = 3, fontface = "bold") +
  labs(title = "Average Ride Duration by Ride Type",
    subtitle = "Visualizing average ride duration by member type with bike type facet",
    x = "Day Type",
    y = "Average Rides (minutes)") +
  theme_light() +
  theme(legend.position = "bottom")
print(p5c)
```

Average Ride Duration by Ride Type

Visualizing average ride duration by member type with bike type facet



5.d Peak Hour by Member Type

A stacked plot with member type show the pattern of our users:

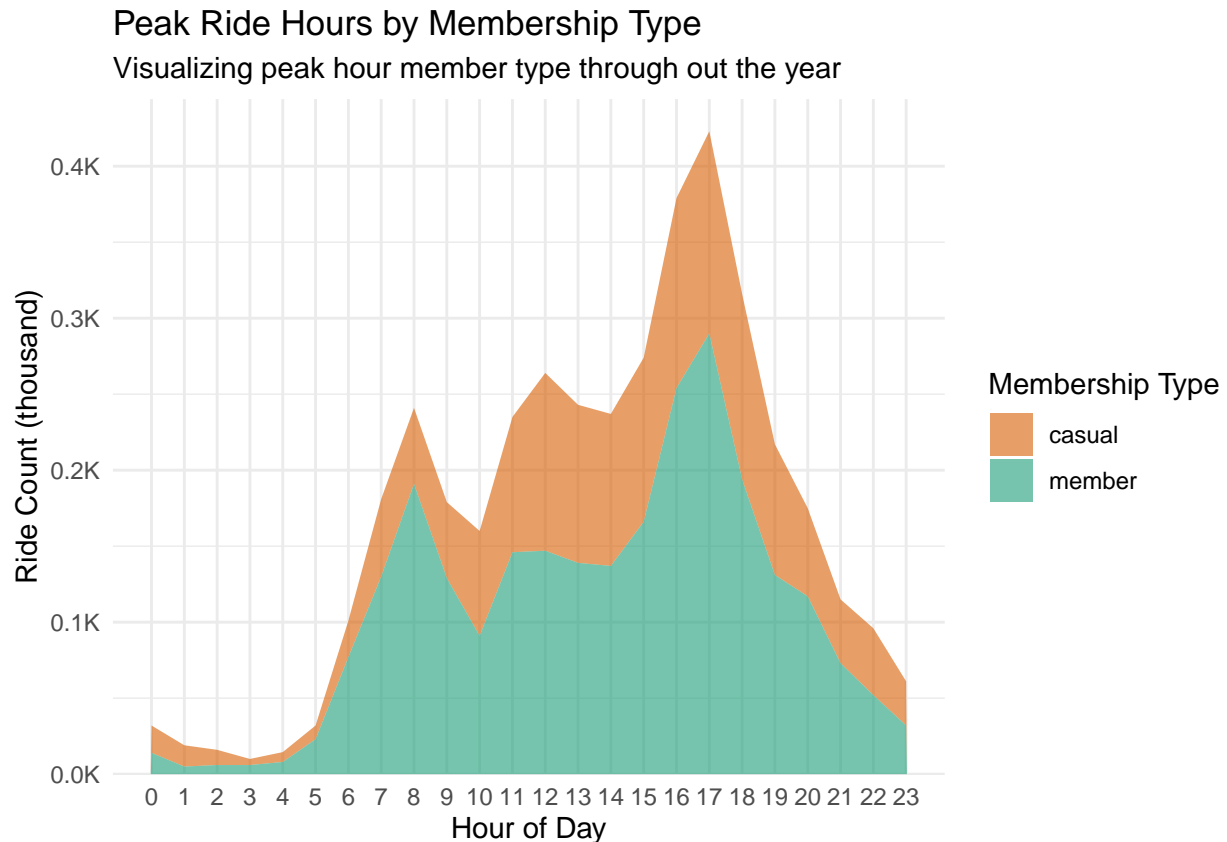
- Morning peak (6–9 AM): commuter usage, likely Annual Member

- Afternoon peak (3–8 PM): return commutes
- Midday or evening spikes: leisure behavior, likely Casual

```
#peak hour summary & user tendencies
ride_hour_summary <- ride_duration_clean %>%
  mutate(hour = hour(started_at))

peak_by_user <- ride_hour_summary %>%
  group_by(member_casual, hour) %>%
  summarise(total_rides = n(), .groups = "drop")

#Plot Peak ride hour
p5d <- ggplot(peak_by_user, aes(x = hour, y = total_rides, fill = member_casual)) +
  geom_area(alpha = 0.6) +
  scale_fill_manual(name = "Membership Type",
                    values = c(member = "#1b9e77", casual = "#d95f02")) +
  scale_x_continuous(breaks = seq(0, 23, by = 1),
                    minor_breaks = NULL) +
  scale_y_continuous(labels = label_number(scale = 1e-3, suffix = "K"),
                    expand = expansion(mult = c(0, 0.05))) +
  labs(title = "Peak Ride Hours by Membership Type",
       subtitle = "Visualizing peak hour member type through out the year",
       x = "Hour of Day", y = "Ride Count (thousand)") +
  theme_minimal()
print(p5d)
```



6. Station Insight

We identified the top 10 starting and ending stations by ride volume. These high-traffic locations, often downtown or near tourist zones, indicate key operational touch-points for fleet allocation and promotional targeting

```
top_started_station <- ride_duration_clean %>%
  group_by(start_station_name) %>%
  drop_na(start_station_name) %>%
  summarise(total_started = (count= n()), .groups = "drop") %>%
  arrange(desc(total_started)) %>%
  slice_head(n = 10)
knitr::kable(top_started_station,
  caption = "Top 10 Start Station")
```

Table 3: Top 10 Start Station

start_station_name	total_started
Streeter Dr & Grand Ave	55
DuSable Lake Shore Dr & Monroe St	41
Wells St & Concord Ln	37
Kingsbury St & Kinzie St	35
Clark St & Elm St	34
Millennium Park	32
Daley Center Plaza	31
Wells St & Huron St	31
Michigan Ave & Oak St	30
Clinton St & Madison St	28

```
top_ended_station <- ride_duration_clean %>%
  group_by(end_station_name) %>%
  drop_na(end_station_name) %>%
  summarise(total_end = (count= n()), .groups = "drop") %>%
  arrange(desc(total_end)) %>%
  slice_head(n = 10)
knitr::kable(top_ended_station,
  caption = "Top 10 End Station")
```

Table 4: Top 10 End Station

end_station_name	total_end
Streeter Dr & Grand Ave	49
Kingsbury St & Kinzie St	45
Clark St & Elm St	38
DuSable Lake Shore Dr & Monroe St	38
Michigan Ave & Oak St	38
DuSable Lake Shore Dr & North Blvd	33
Clinton St & Madison St	30
Millennium Park	30
Wabash Ave & Grand Ave	30

end_station_name	total_end
LaSalle St & Illinois St	28

7. Insight Recommendation

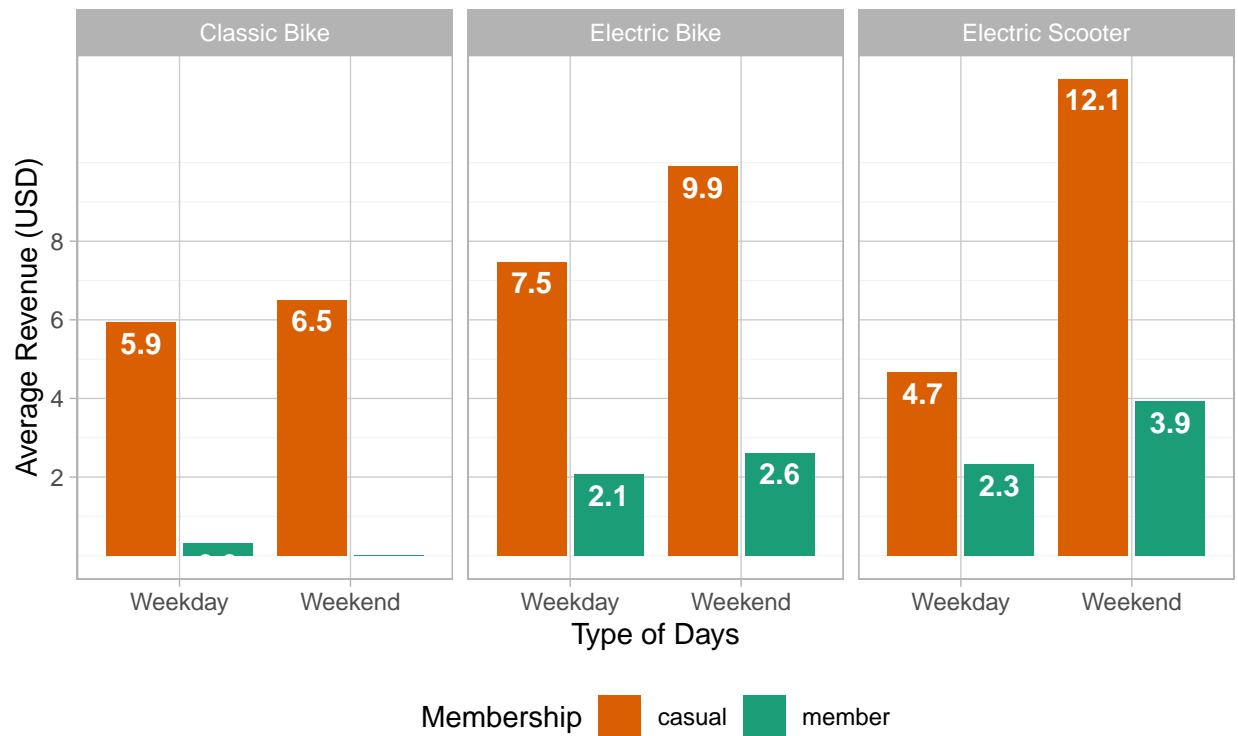
7.a User Behavior Finding

- Casual rider have preference using our service on warmth season (May - October), Annual Member showing loyalty through the cold season.
- Annual Member have consistent rides duration and within in their free-time windows (classic bikes), shows our price plan are work with Annual Member. Based on revenue faceted by ride type (see *plot Seasonal Revenue by Ride Type* facet Electric Bike), Annual Member tendencies have a longer ride with e-bike – breaching the free-time windows (charge after 31 minutes rides).

p3b

Weekday vs Weekend Revenue

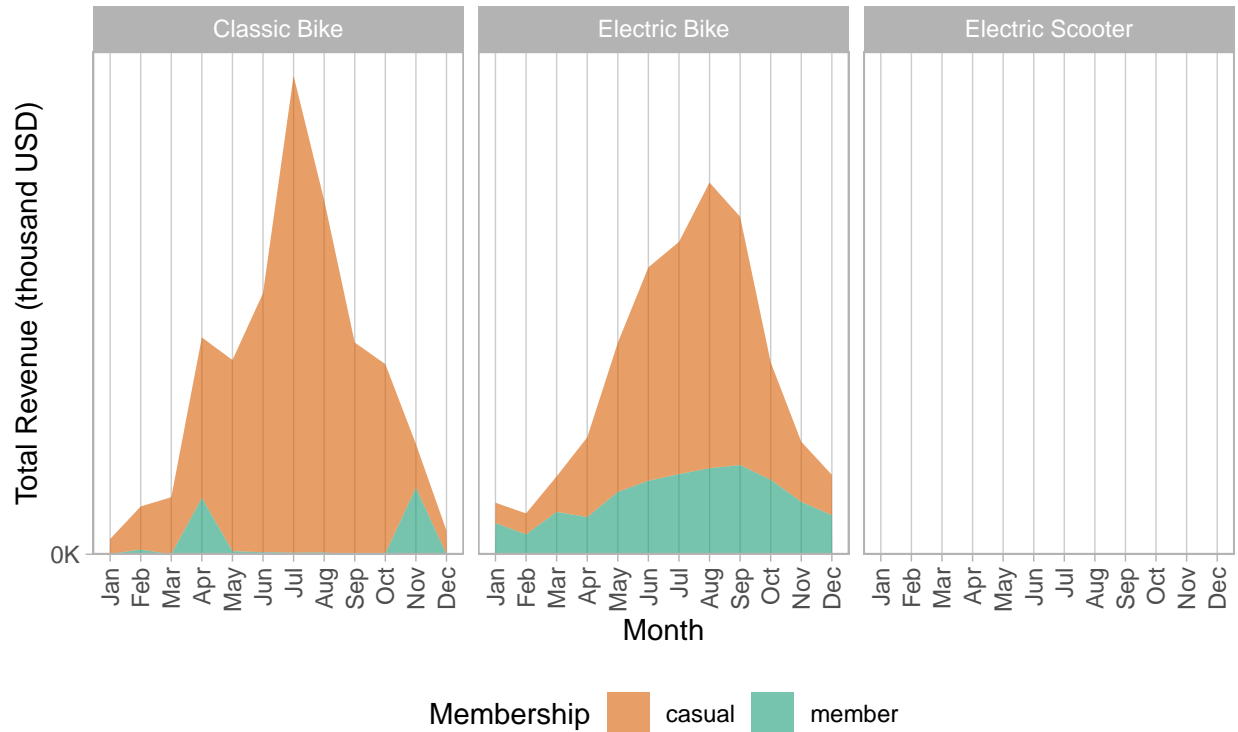
Visualizing average revenue by member types with ride types facet



p3a2

Seasonal Revenue Trends by Ride Types

Visualizing revenue trend by member types with ride types facet

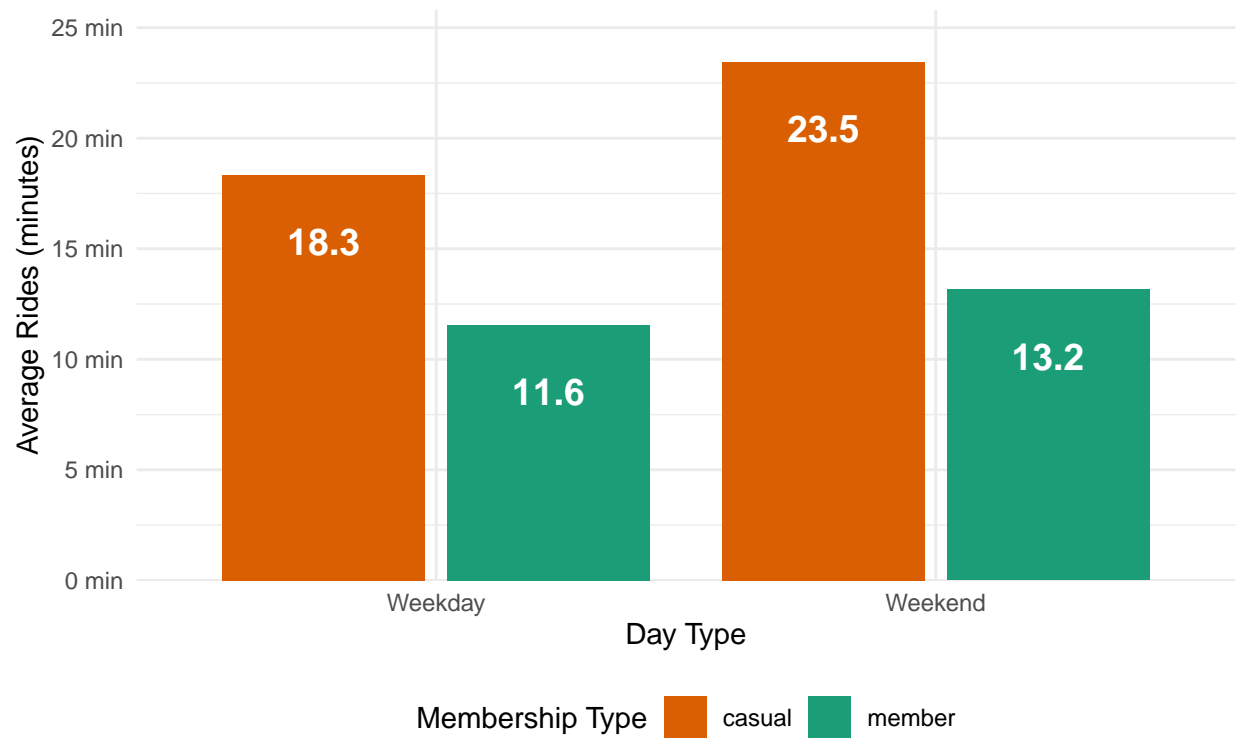


- Electric Scooter have promising stream of revenue with casual rider, but due too data limitation (only 2 months recorded for electric scooter) it still hypothesis.
- Casual rider shown have higher intention to use our service longer than our Annual Member based on their Average Ride Duration. On the weekend casual rider tend to have longer rides, while Annual Member is less time dependent.

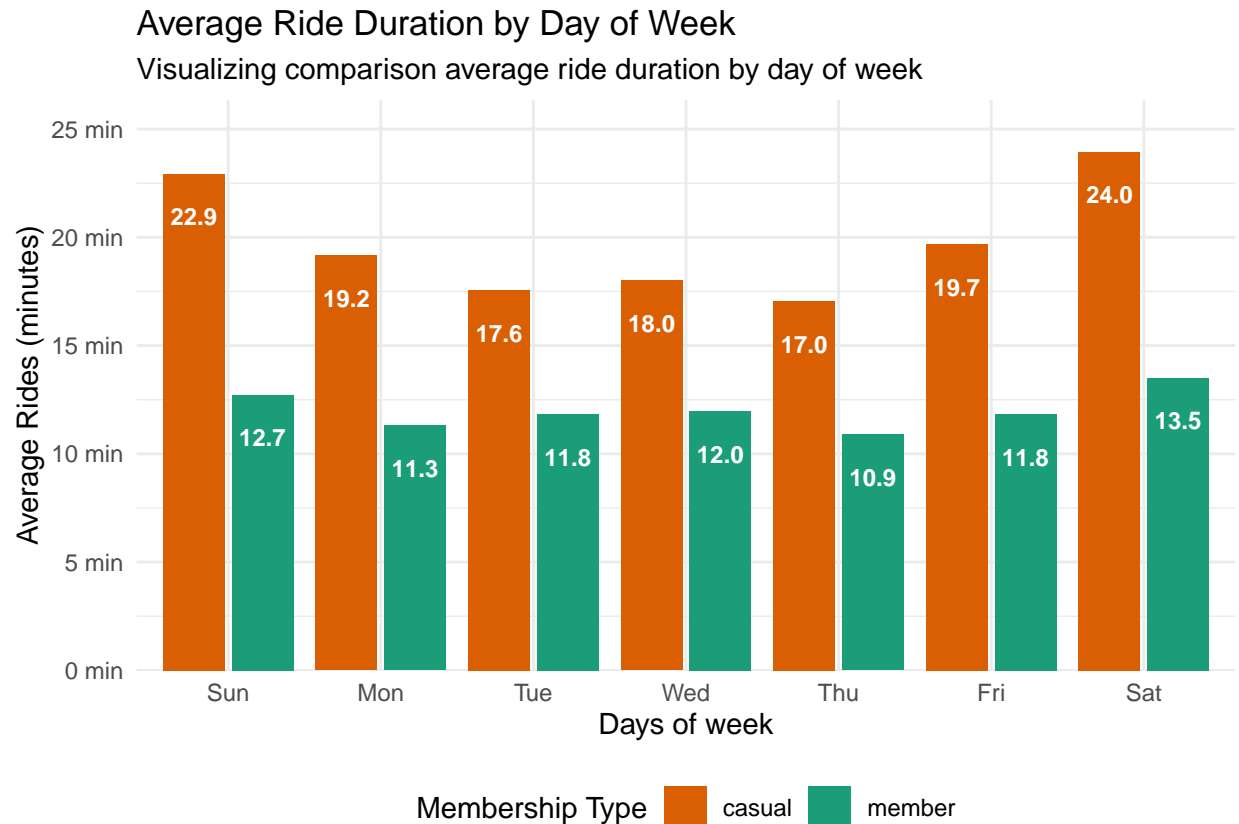
p5a

Average Ride Duration Weekday vs Weekend

Visualizing compariosn average ride duration by member types



p5b



7.b Marketing Strategies Suggestion

7.b.i Casual Rider Conversion With Casual Rider have high intention in weekend and summers, we can create focus time marketing program. Few marketing program suggestion and implication:

1. Summer Classic Bike Deals (July - August)

- Offers for Casual rider 3 months unlimited time classic bike rides with discount when converting to Annual Member.
- Promoting via in-app (push notification) & email marketing, targeting Casual rider have history renting classic bike more than 15-20 rides in a months.
- KPI to track: Program sign-up, Upgrade conversion.

2. In-App Notification at 20min

- Offers for Casual rider who rides more than 20 minutes rides using classic bikes for Annual membership programs.
- Trigger push notification or at the end trip notification. Targeting casual rider with 5 days in a row using classic bikes more than 30 minutes per day (assuming the user use it for commutes). Sample of copy notification:
Enjoy our classic bike? Unlock your first year membership for just \$143.9 and ride free up to 45 minutes.
- KPI to track: Click through rate, Sign-up rate.

7.b.ii Annual Membership Growth and Retention Annual member are our ambassador for our services, since they are the one who keep steady revenue and word-of-mouth recommendation for our services. Maintaining the growth and retention our loyal friend are a must. Don't trick them to bought services they don't need but a genuine celebration for their commitment to our services.

1. Early Renewal Incentive

- Offers 10%-15% discount on next year Annual Membership if renewed two-months earlier.
- Promoting via personal email marketing and apps notification starting at month ten, give reminder 14 days after non responds in email renewal.
- KPI to track: Renewal sign-up. Revenue track between renewal and lapsed member.

2. Tiered Loyalty Program

- Keep annual member engagement via in-app loyalty program that rewards user, introducing gamify tiered step for their activities (e.g. referral membership, social media activity, ride counts, milestone minutes). The rewards should related to bike activities that worth to use/wear by them (e.g. winter jacket, gloves rides, collapse-able helm, portable bottle)
- Loyalty point tracker that in-app integration via their basic activities in-app (rides, share rides, etc).
- KPI to track: Rewards redemption, Incremental rate loyalty trackers.

7.c Operational Optimization

Below are targeted recommendations to streamline our day-to-day operations, grounded in the data patterns we observed.

1. Seasonal Maintenance Scaling

From May - October increasing in total rides, driving higher wear and tear of our bikes. Without proper maintenance, bike downtime (due maintenance) and service interruption will rise, harming user experience and revenue.

Recommendation:

Partner with an external bike-service provider to supplement in-house repairs during peak season.

2. First Ride On-boarding & Apps Stability

Casual riders generate most "Under 1 min" ride outliers, which often indicate accidental unlocks or confusion with the app interface. Misuse increases operational costs (unreported rentals, city fines) and damages the brand if users feel penalized for honest mistakes.

Recommendation:

Implement mandatory "How-to Ride" tutorial in-app after user scan their 1st ride, segment tutorial based on ride journey (start, before rides-safety, docking, & billings)

3. Busiest Station Management

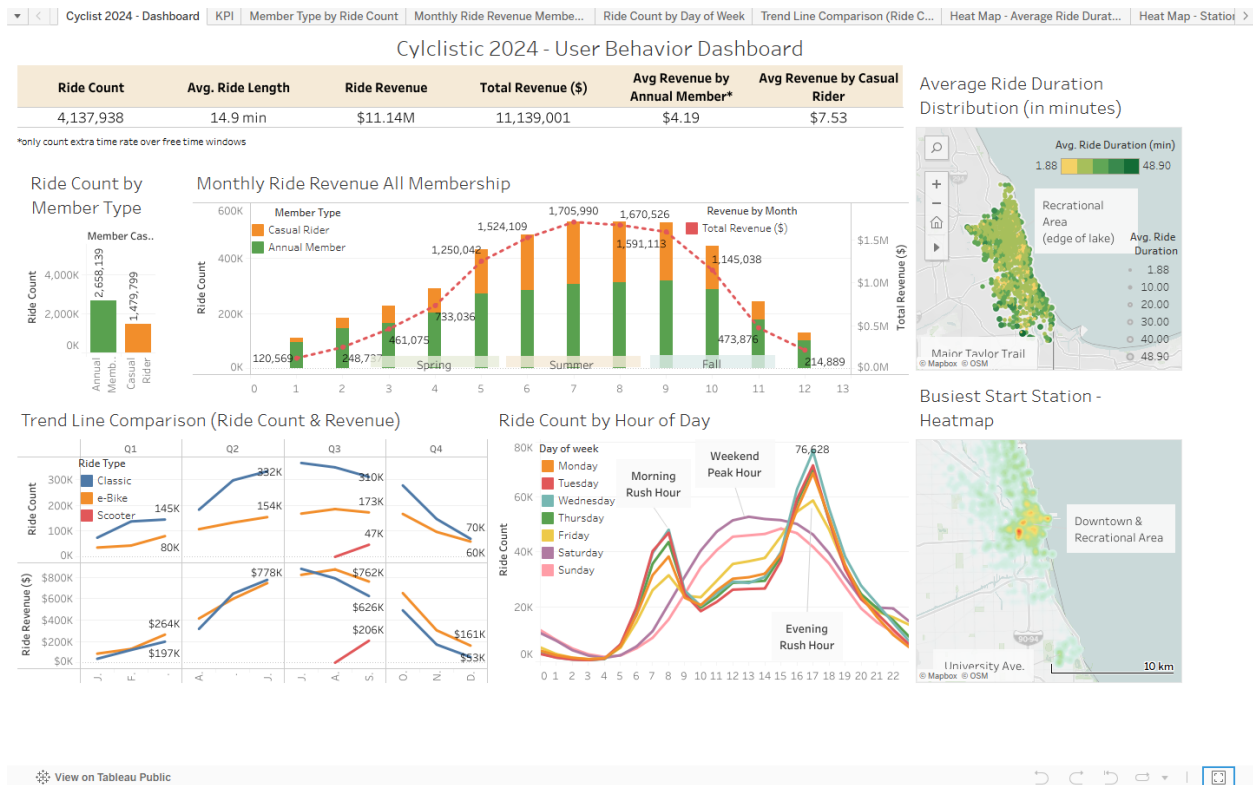
The top 10 busiest stations fall into two user persona—downtown commuter hubs on weekdays and recreational hotspots on weekends. Stock imbalances at key station frustrate users, leading to lost rides and diminished trust in the network's reliability.

Recommendation:

Deploy re-balancing van before morning peak time (6–9 AM and (3–8 PM) aim for 80% bike availability during this peak time. Introducing a user incentive program that rewards trips ending at lower-capacity stations near busy stations.

Interactive Dashboard

Explore the full dashboard on Tableau Public.



This dashboard visualizes seasonal trends, user types, trend line revenue comparison (bike type vs revenue), busiest station heatmap and ride duration distribution map with interactive filters.

Limitations & Future Works

- Data-set provided are limited, each ride_id not represent the user and channel of purchase. All observations are intended for educational purposes, not real-life business recommendation.
- Revenue calculations in this report exclude capped pricing rules for member e-bike rides between 31–45 minutes and do not account for day-pass packages due to data limitations. Annual Member revenue only count for their extra time using after 45 minutes for classic bike and after 30 minutes with e-bike.
- Authors will explore and learn geospatial data visualization for creating heat-map of Top 10 Station, while learn better structure and writing style in R.

References

1. Divvy Trip Dataset from (Jan - Dec 2024)
2. R packages (tidyveres, skimr, janitor, scales, patchwork)

About the Author

Fachreza Dewanto

Creative by Experience, Analytical by Curiosity | Learn, Adapt, Evolve

Creative professional with 13+ years of experience in multimedia production, content marketing, and campaign strategy. Skilled in transforming creative ideas into measurable digital outcomes, with a strong foundation in video production, social media, and stakeholder collaboration. Currently transitioning into data-driven creative roles, actively upskilling with the Google Data Analytics Certification.

GitHub: fac-dew / LinkedIn: Fachreza Dewanto / Kaggle: facdew