CYCLISTIC BIKE SHARE DATA ANALYSIS

(A Google data analysis capstone project)

DHANUSHWR K

JUNIOR DATA ANALYST

Table of Contents

1. Introduction

2. Background of the company

3. Project Overview

4. Ask

5. Prepare

6. Process

7. Analyse

8. Share

9. Act

1. Introduction:

This report presents a comprehensive bike share data analysis for a fictional company called “*Cyclistic”* to understand the differences between casual and annual members. The goal is to inform a marketing strategy aimed at converting casual riders into annual members.

As a junior data analyst, I have applied the skills gained from the ***Google Data Analytics Professional Certificate*** to this project. Utilizing Excel, SQL, Big Query for data manipulation, and Tableau for visualization, I have explored the dataset to uncover valuable insights.

The analysis follows a structured approach, encompassing data exploration, comparison of user behaviour, and identification of potential marketing opportunities. The findings will be presented clearly and concisely to support data-driven decision-making.

2. Background of the company:

Cyclistic is a bike-sharing service operating in Chicago. With a network of over 5,800 bicycles and 600 docking stations, it offers convenient access to bike rentals for residents and visitors alike. To cater to a diverse range of users, Cyclistic provides a variety of bike types, including traditional bicycles, reclining bikes, hand tricycles, and cargo bikes.

Launched in 2016, Cyclistic has experienced significant growth and has become a popular mode of transportation within the city. While many customers utilize the service for leisure activities, approximately 30% of riders rely on Cyclistic for their daily commute.

To accommodate different usage patterns, Cyclistic offers flexible pricing options. Customers can choose from single-ride passes for occasional use, full-day passes for extended trips, or annual memberships for frequent riders. Customers purchasing single-ride or full-day passes are classified as casual riders, while those with annual memberships are considered Cyclistic members. Financial analysis has revealed that annual members contribute substantially more to the company's revenue than casual riders.

3. Project overview

Recognizing the potential for growth, Cyclistic aims to increase the number of annual members. The marketing team believes that a deeper understanding of how casual riders and annual members utilize the service will inform a targeted marketing strategy to convert casual riders into loyal subscribers.

As a junior data analyst, I will analyse historical bike trip data to identify key differences between these two customer segments. The insights derived from this analysis will be instrumental in developing effective marketing campaigns. Ultimately, the goal is to present compelling data-driven recommendations to the Cyclistic executive team to secure approval for the proposed marketing initiatives.

4. ASK

Three questions will guide the future marketing program:

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?

Moreno, director of marketing and manager has assigned me the question to answer: ***How do annual members and casual riders use Cyclistic bikes differently?***

Business objective: To analyse the behavioural differences between casual and annual Cyclistic members to inform the development of a marketing strategy aimed at converting casual riders to annual memberships.

5. PREPARE

This analysis utilizes Cyclistic's historical bike trip data spanning from June 2023 to June 2024. The dataset, publicly available from Motivate International Inc. under license, provides insights into customer usage patterns. Due to privacy regulations, personally identifiable rider information is excluded.

The data is structured into monthly files, each containing detailed trip information. Key variables include ride identifiers, bike type, start and end times, station details**, geographic coordinates, and customer type (member or casual rider).**

6. PROCESS

- **Data quality:** Initial exploration of the dataset revealed several data quality issues. Notably, a significant proportion of records contained missing values for station identifiers, names, and minimal values for geographic coordinates. Additionally, inconsistencies were observed in ride start and end times, with some instances indicating start times occurring after the corresponding end times.
- **Data Transformation:** New columns were created using excel individually for every month to facilitate analysis. This included ride duration, start and end hours, and day of the week.

FORMULA'S:

1) ride_length: =ABS(ended_at – started_at) → Formatted to time (HH: MM: SS).
2) start_hour & end_hour: =HOUR(started_at) & =HOUR(ended_at)
3) day_of_week: = WEEKDAY(started_at)

Finally, blanks were filled with "null" (Ctrl + G → "Go to" dialogue box → special → blanks → ok → type "null" in one blank cell → Ctrl + D: Automatic filling)

- The substantial size of the Cyclistic dataset, exceeding five million rows, necessitated the use of a scalable data platform. Given the limitations of Microsoft Excel, Google BigQuery was selected for data management and processing. To accommodate the dataset's size, the original files were divided into smaller subsets to comply with BigQuery's upload restrictions (Limit 100 MB). Subsequently, these individual files were consolidated into a unified dataset for comprehensive analysis. This process ensured efficient handling of the large volume of data and facilitated subsequent data exploration and analysis.

SQL query employed for combining files to a unified dataset:

- *This query unifies distinct columns (thereby removing duplicates)*
- *Day of week column converted to string by changing the number of the day to name*

```
SELECT
  t1.ride_id,
  t1.member_casual,
  t1.rideable_type,
  t2.start_station_id,
  t2.start_station_name,
  t2.start_lat,
  t2.start_lng,
  t1.started_at,
  t2.end_station_id,
  t2.end_station_name,
  t1.ended_at,
  SAFE_CAST(t2.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t2.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t1.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t1.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t1.start_hour IS NULL THEN NULL
         ELSE CAST(t1.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t1.end_hour IS NULL THEN NULL
         ELSE CAST(t1.end_hour AS INT64)
    END AS end_hour,
    CASE
    WHEN t1.day_of_week IS NULL THEN NULL
    WHEN t1.day_of_week = 1 THEN "Sunday"
    WHEN t1.day_of_week = 2 THEN "Monday"
    WHEN t1.day_of_week = 3 THEN "Tuesday"
    WHEN t1.day_of_week = 4 THEN "Wednesday"
    WHEN t1.day_of_week = 5 THEN "Thursday"
    WHEN t1.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week
FROM bike-ridership-analysis.trip_data.01_24_nllsi AS t1
FULL JOIN bike-ridership-analysis.trip_data.01_24_sil AS t2
ON t1.ride_id = t2.ride_id

UNION ALL

SELECT
  t3.ride_id,
  t3.member_casual,
  t3.rideable_type,
  t4.start_station_id,
  t4.start_station_name,
  t4.start_lat,
  t4.start_lng,
  t3.started_at,
  t4.end_station_id,
```

```
  t4.end_station_name,
  t3.ended_at,
  SAFE_CAST(t4.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t4.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t3.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t3.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t3.start_hour IS NULL THEN NULL
         ELSE CAST(t3.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t3.end_hour IS NULL THEN NULL
         ELSE CAST(t3.end_hour AS INT64)
    END AS end_hour,
    CASE
    WHEN t3.day_of_week IS NULL THEN NULL
    WHEN t3.day_of_week = 1 THEN "Sunday"
    WHEN t3.day_of_week = 2 THEN "Monday"
    WHEN t3.day_of_week = 3 THEN "Tuesday"
    WHEN t3.day_of_week = 4 THEN "Wednesday"
    WHEN t3.day_of_week = 5 THEN "Thursday"
    WHEN t3.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.02_24_nllsi AS t3
FULL JOIN bike-ridership-analysis.trip_data.02_24_sil AS t4
ON t3.ride_id = t4.ride_id

UNION ALL

SELECT
  t5.ride_id,
  t5.member_casual,
  t5.rideable_type,
  t6.start_station_id,
  t6.start_station_name,
  t6.start_lat,
  t6.start_lng,
  t5.started_at,
  t6.end_station_id,
  t6.end_station_name,
  t5.ended_at,
  SAFE_CAST(t6.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t6.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t5.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t5.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t5.start_hour IS NULL THEN NULL
         ELSE CAST(t5.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t5.end_hour IS NULL THEN NULL
         ELSE CAST(t5.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t5.day_of_week IS NULL THEN NULL
    WHEN t5.day_of_week = 1 THEN "Sunday"
    WHEN t5.day_of_week = 2 THEN "Monday"
    WHEN t5.day_of_week = 3 THEN "Tuesday"
```

```
    WHEN t5.day_of_week = 4 THEN "Wednesday"
    WHEN t5.day_of_week = 5 THEN "Thursday"
    WHEN t5.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.03_24_nllsi AS t5
FULL JOIN bike-ridership-analysis.trip_data.03_24_sil AS t6
ON t5.ride_id = t6.ride_id

UNION ALL

SELECT
  t7.ride_id,
  t7.member_casual,
  t7.rideable_type,
  t8.start_station_id,
  t8.start_station_name,
  t8.start_lat,
  t8.start_lng,
  t7.started_at,
  t8.end_station_id,
  t8.end_station_name,
  t7.ended_at,
  SAFE_CAST(t8.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t8.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t7.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t7.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t7.start_hour IS NULL THEN NULL
         ELSE CAST(t7.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t7.end_hour IS NULL THEN NULL
         ELSE CAST(t7.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t7.day_of_week IS NULL THEN NULL
    WHEN t7.day_of_week = 1 THEN "Sunday"
    WHEN t7.day_of_week = 2 THEN "Monday"
    WHEN t7.day_of_week = 3 THEN "Tuesday"
    WHEN t7.day_of_week = 4 THEN "Wednesday"
    WHEN t7.day_of_week = 5 THEN "Thursday"
    WHEN t7.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.04_24_nllsi AS t7
FULL JOIN bike-ridership-analysis.trip_data.04_24_sil AS t8
ON t7.ride_id = t8.ride_id

UNION ALL

SELECT
  t9.ride_id,
  t9.member_casual,
  t9.rideable_type,
  t10.start_station_id,
  t10.start_station_name,
```

```sql
  t10.start_lat,
  t10.start_lng,
  t9.started_at,
  t10.end_station_id,
  t10.end_station_name,
  t9.ended_at,
  SAFE_CAST(t10.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t10.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t9.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t9.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t9.start_hour IS NULL THEN NULL
         ELSE CAST(t9.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t9.end_hour IS NULL THEN NULL
         ELSE CAST(t9.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t9.day_of_week IS NULL THEN NULL
    WHEN t9.day_of_week = 1 THEN "Sunday"
    WHEN t9.day_of_week = 2 THEN "Monday"
    WHEN t9.day_of_week = 3 THEN "Tuesday"
    WHEN t9.day_of_week = 4 THEN "Wednesday"
    WHEN t9.day_of_week = 5 THEN "Thursday"
    WHEN t9.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.05_24_nllsi AS t9
FULL JOIN bike-ridership-analysis.trip_data.05_24_sil AS t10
ON t9.ride_id = t10.ride_id

UNION ALL

SELECT
  t11.ride_id,
  t11.member_casual,
  t11.rideable_type,
  t12.start_station_id,
  t12.start_station_name,
  t12.start_lat,
  t12.start_lng,
  t11.started_at,
  t12.end_station_id,
  t12.end_station_name,
  t11.ended_at,
  SAFE_CAST(t12.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t12.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t11.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t11.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t11.start_hour IS NULL THEN NULL
         ELSE CAST(t11.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t11.end_hour IS NULL THEN NULL
         ELSE CAST(t11.end_hour AS INT64)
    END AS end_hour,
  CASE
```

```
    WHEN t11.day_of_week IS NULL THEN NULL
    WHEN t11.day_of_week = 1 THEN "Sunday"
    WHEN t11.day_of_week = 2 THEN "Monday"
    WHEN t11.day_of_week = 3 THEN "Tuesday"
    WHEN t11.day_of_week = 4 THEN "Wednesday"
    WHEN t11.day_of_week = 5 THEN "Thursday"
    WHEN t11.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.06_24_nllsi AS t11
FULL JOIN bike-ridership-analysis.trip_data.06_24_sil AS t12
ON t11.ride_id = t12.ride_id

UNION ALL

SELECT
  t13.ride_id,
  t13.member_casual,
  t13.rideable_type,
  t14.start_station_id,
  t14.start_station_name,
  t14.start_lat,
  t14.start_lng,
  t13.started_at,
  t14.end_station_id,
  t14.end_station_name,
  t13.ended_at,
  SAFE_CAST(t14.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t14.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t13.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t13.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t13.start_hour IS NULL THEN NULL
         ELSE CAST(t13.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t13.end_hour IS NULL THEN NULL
         ELSE CAST(t13.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t13.day_of_week IS NULL THEN NULL
    WHEN t13.day_of_week = 1 THEN "Sunday"
    WHEN t13.day_of_week = 2 THEN "Monday"
    WHEN t13.day_of_week = 3 THEN "Tuesday"
    WHEN t13.day_of_week = 4 THEN "Wednesday"
    WHEN t13.day_of_week = 5 THEN "Thursday"
    WHEN t13.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.06_23_nllsi AS t13
FULL JOIN bike-ridership-analysis.trip_data.06_23_sil AS t14
ON t13.ride_id = t14.ride_id

UNION ALL

SELECT
  t15.ride_id,
```

```
  t15.member_casual,
  t15.rideable_type,
  t16.start_station_id,
  t16.start_station_name,
  t16.start_lat,
  t16.start_lng,
  t15.started_at,
  t16.end_station_id,
  t16.end_station_name,
  t15.ended_at,
  SAFE_CAST(t16.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t16.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t15.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t15.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t15.start_hour IS NULL THEN NULL
         ELSE CAST(t15.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t15.end_hour IS NULL THEN NULL
         ELSE CAST(t15.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t15.day_of_week IS NULL THEN NULL
    WHEN t15.day_of_week = 1 THEN "Sunday"
    WHEN t15.day_of_week = 2 THEN "Monday"
    WHEN t15.day_of_week = 3 THEN "Tuesday"
    WHEN t15.day_of_week = 4 THEN "Wednesday"
    WHEN t15.day_of_week = 5 THEN "Thursday"
    WHEN t15.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.07_23_nllsi AS t15
FULL JOIN bike-ridership-analysis.trip_data.07_23_sil AS t16
ON t15.ride_id = t16.ride_id

UNION ALL

SELECT
  t17.ride_id,
  t17.member_casual,
  t17.rideable_type,
  t18.start_station_id,
  t18.start_station_name,
  t18.start_lat,
  t18.start_lng,
  t17.started_at,
  t18.end_station_id,
  t18.end_station_name,
  t17.ended_at,
  SAFE_CAST(t18.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t18.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t17.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t17.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t17.start_hour IS NULL THEN NULL
         ELSE CAST(t17.start_hour AS INT64)
    END AS start_hour,
```

```
  CASE WHEN t17.end_hour IS NULL THEN NULL
         ELSE CAST(t17.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t17.day_of_week IS NULL THEN NULL
    WHEN t17.day_of_week = 1 THEN "Sunday"
    WHEN t17.day_of_week = 2 THEN "Monday"
    WHEN t17.day_of_week = 3 THEN "Tuesday"
    WHEN t17.day_of_week = 4 THEN "Wednesday"
    WHEN t17.day_of_week = 5 THEN "Thursday"
    WHEN t17.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.08_23_nllsi AS t17
FULL JOIN bike-ridership-analysis.trip_data.08_23_sil AS t18
ON t17.ride_id = t18.ride_id

UNION ALL

SELECT
  t19.ride_id,
  t19.member_casual,
  t19.rideable_type,
  t20.start_station_id,
  t20.start_station_name,
  t20.start_lat,
  t20.start_lng,
  t19.started_at,
  t20.end_station_id,
  t20.end_station_name,
  t19.ended_at,
  SAFE_CAST(t20.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t20.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t19.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t19.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t19.start_hour IS NULL THEN NULL
         ELSE CAST(t19.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t19.end_hour IS NULL THEN NULL
         ELSE CAST(t19.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t19.day_of_week IS NULL THEN NULL
    WHEN t19.day_of_week = 1 THEN "Sunday"
    WHEN t19.day_of_week = 2 THEN "Monday"
    WHEN t19.day_of_week = 3 THEN "Tuesday"
    WHEN t19.day_of_week = 4 THEN "Wednesday"
    WHEN t19.day_of_week = 5 THEN "Thursday"
    WHEN t19.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.09_23_nllsi AS t19
FULL JOIN bike-ridership-analysis.trip_data.09_23_sil AS t20
ON t19.ride_id = t20.ride_id
```

```sql
UNION ALL

SELECT
  t21.ride_id,
  t21.member_casual,
  t21.rideable_type,
  t22.start_station_id,
  t22.start_station_name,
  t22.start_lat,
  t22.start_lng,
  t21.started_at,
  t22.end_station_id,
  t22.end_station_name,
  t21.ended_at,
  SAFE_CAST(t22.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t22.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t21.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t21.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t21.start_hour IS NULL THEN NULL
         ELSE CAST(t21.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t21.end_hour IS NULL THEN NULL
         ELSE CAST(t21.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t21.day_of_week IS NULL THEN NULL
    WHEN t21.day_of_week = 1 THEN "Sunday"
    WHEN t21.day_of_week = 2 THEN "Monday"
    WHEN t21.day_of_week = 3 THEN "Tuesday"
    WHEN t21.day_of_week = 4 THEN "Wednesday"
    WHEN t21.day_of_week = 5 THEN "Thursday"
    WHEN t21.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.10_23_nllsi AS t21
FULL JOIN bike-ridership-analysis.trip_data.10_23_sil AS t22
ON t21.ride_id = t22.ride_id

UNION ALL

SELECT
  t23.ride_id,
  t23.member_casual,
  t23.rideable_type,
  t24.start_station_id,
  t24.start_station_name,
  t24.start_lat,
  t24.start_lng,
  t23.started_at,
  t24.end_station_id,
  t24.end_station_name,
  t23.ended_at,
  SAFE_CAST(t24.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t24.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t23.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t23.ride_length_hhmmss AS TIME)
```

```
    END AS ride_length,
  CASE WHEN t23.start_hour IS NULL THEN NULL
         ELSE CAST(t23.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t23.end_hour IS NULL THEN NULL
         ELSE CAST(t23.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t23.day_of_week IS NULL THEN NULL
    WHEN t23.day_of_week = 1 THEN "Sunday"
    WHEN t23.day_of_week = 2 THEN "Monday"
    WHEN t23.day_of_week = 3 THEN "Tuesday"
    WHEN t23.day_of_week = 4 THEN "Wednesday"
    WHEN t23.day_of_week = 5 THEN "Thursday"
    WHEN t23.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week

FROM bike-ridership-analysis.trip_data.11_23_nllsi AS t23
FULL JOIN bike-ridership-analysis.trip_data.11_23_sil AS t24
ON t23.ride_id = t24.ride_id

UNION ALL

SELECT
  t25.ride_id,
  t25.member_casual,
  t25.rideable_type,
  t26.start_station_id,
  t26.start_station_name,
  t26.start_lat,
  t26.start_lng,
  t25.started_at,
  t26.end_station_id,
  t26.end_station_name,
  t25.ended_at,
  SAFE_CAST(t26.end_lat AS FLOAT64) AS end_lat,
  SAFE_CAST(t26.end_lng AS FLOAT64) AS end_lng,
  CASE WHEN t25.ride_length_hhmmss IS NULL THEN NULL
         ELSE CAST(t25.ride_length_hhmmss AS TIME)
    END AS ride_length,
  CASE WHEN t25.start_hour IS NULL THEN NULL
         ELSE CAST(t25.start_hour AS INT64)
    END AS start_hour,
  CASE WHEN t25.end_hour IS NULL THEN NULL
         ELSE CAST(t25.end_hour AS INT64)
    END AS end_hour,
  CASE
    WHEN t25.day_of_week IS NULL THEN NULL
    WHEN t25.day_of_week = 1 THEN "Sunday"
    WHEN t25.day_of_week = 2 THEN "Monday"
    WHEN t25.day_of_week = 3 THEN "Tuesday"
    WHEN t25.day_of_week = 4 THEN "Wednesday"
    WHEN t25.day_of_week = 5 THEN "Thursday"
    WHEN t25.day_of_week = 6 THEN "Friday"
    ELSE "Saturday"
  END AS day_of_week
```

```
FROM bike-ridership-analysis.trip_data.12_23_nllsi AS t25
FULL JOIN bike-ridership-analysis.trip_data.12_23_sil AS t26
ON t25.ride_id = t26.ride_id
```

SQL query employed for ensuring blanks were treated as “null”:

```
SELECT
       CASE
         WHEN ride_id = "null" THEN NULL
         WHEN ride_id =  "" THEN NULL
        ELSE ride_id
        END AS ride_id,
        CASE
          WHEN member_casual = "null" THEN NULL
          WHEN member_casual =  "" THEN NULL
          ELSE member_casual
        END AS member_casual,
        CASE
          WHEN rideable_type = "null" THEN NULL
          WHEN rideable_type =  "" THEN NULL
          ELSE rideable_type
        END AS rideable_type,
        CASE
          WHEN start_station_id = "null" THEN NULL
          WHEN start_station_id =  "" THEN NULL
          ELSE start_station_id
        END AS start_station_id,
        CASE
          WHEN start_station_name = "null" THEN NULL
          WHEN start_station_name =  "" THEN NULL
          ELSE start_station_name
        END AS start_station_name,
        started_at,
        CASE
          WHEN start_lat IS NULL THEN CAST(NULL AS FLOAT64)
          WHEN CAST(start_lat AS STRING) = '' THEN NULL
          ELSE start_lat
        END AS start_lat,
        CASE
          WHEN start_lng IS NULL THEN CAST(NULL AS FLOAT64)
          WHEN CAST(start_lng AS STRING) = '' THEN NULL
          ELSE start_lng
        END AS start_lng,
                CASE
          WHEN end_station_id = "null" THEN NULL
          WHEN end_station_id =  "" THEN NULL
          ELSE end_station_id
        END AS end_station_id,
        CASE
          WHEN end_station_name = "null" THEN NULL
          WHEN end_station_name =  "" THEN NULL
          ELSE end_station_name
        END AS end_station_name,
        ended_at,
        CASE
          WHEN end_lat IS NULL THEN CAST(NULL AS FLOAT64)
          WHEN CAST(end_lat AS STRING) = '' THEN NULL
```

```
      ELSE end_lat
    END AS end_lat,
    CASE
      WHEN end_lng IS NULL THEN CAST(NULL AS FLOAT64)
      WHEN CAST(end_lng AS STRING) = '' THEN NULL
      ELSE end_lng
    END AS end_lng,
    ride_length,
    start_hour,
    end_hour,
    day_of_week
FROM  bike-ridership-analysis.trip_data.divvy_bikeridership_prelim2
```

Final dataset:

23_24_divvy_bike_ridership_details QUERY SHARE COPY SNAPSHOT DELETE EXPORT

SCHEMA DETAILS PREVIEW TABLE EXPLORER PREVIEW INSIGHTS PREVIEW LINEAGE DATA PROFILE DATA QUALITY

Filter Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
ride_id	STRING	NULLABLE	-	-	-	-	-
member_casual	STRING	NULLABLE	-	-	-	-	-
rideable_type	STRING	NULLABLE	-	-	-	-	-
start_station_id	STRING	NULLABLE	-	-	-	-	-
start_station_name	STRING	NULLABLE	-	-	-	-	-
started_at	TIMESTAMP	NULLABLE	-	-	-	-	-
start_lat	FLOAT	NULLABLE	-	-	-	-	-
start_lng	FLOAT	NULLABLE	-	-	-	-	-
end_station_id	STRING	NULLABLE	-	-	-	-	-
end_station_name	STRING	NULLABLE	-	-	-	-	-
ended_at	TIMESTAMP	NULLABLE	-	-	-	-	-
end_lat	FLOAT	NULLABLE	-	-	-	-	-
end_lng	FLOAT	NULLABLE	-	-	-	-	-
ride_length	TIME	NULLABLE	-	-	-	-	-
start_hour	INTEGER	NULLABLE	-	-	-	-	-
end_hour	INTEGER	NULLABLE	-	-	-	-	-
day_of_week	STRING	NULLABLE	-	-	-	-	-

Number of missing values were counted from the final dataset:

QUERY:

```
SELECT
   COUNTIF (ride_id IS NULL) AS ride_id,
   COUNTIF (rideable_type IS NULL) AS rideable_type,
   COUNTIF (started_at IS NULL) AS started_at,
   COUNTIF (ended_at IS NULL) AS ended_at,
   COUNTIF (start_station_name IS NULL) AS start_station_name,
   COUNTIF (start_station_id IS NULL) AS start_station_id,
   COUNTIF (end_station_name IS NULL) AS end_station_name,
   COUNTIF (end_station_id IS NULL) AS end_station_id,
   COUNTIF (start_lat IS NULL) AS start_lat,
   COUNTIF (start_lng IS NULL) AS start_lng,
   COUNTIF (end_lat IS NULL) AS end_lat,
```

```
    COUNTIF (end_lng IS NULL) AS end_lng,
    COUNTIF (member_cASual IS NULL) AS member_cASual
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
```

RESULTS:

ride_id	0
rideable_type	0
started_at	0
ended_at	0
start_station_name	1049407
start_station_id	1049407
end_station_name	1104699
end_station_id	1104699
start_lat	0
start_lng	0
end_lat	8769
end_lng	8769
member_casual	0

7. ANALYSE

7.1 Membership Analysis:

Number of casual and Cyclistic members were counted.

QUERY:

```
SELECT
member_casual,
COUNT(member_casual) AS count
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
GROUP BY member_casual
```

RESULTS:

member_casual	count
casual	2350877
member	4103939

7.2 Ride length:

Total ride duration by mins were calculated for both membership types.

QUERY:

```
SELECT
    member_casual,
    SUM(TIMESTAMP_DIFF(ended_at, started_at, MINUTE)) AS total_ride_length
FROM
    bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details
GROUP BY
    member_casual
```

RESULTS:

member_casual	total_ride_length
member	53358596
casual	65979228

7.3 Bikes utilization

- Number of times bikes used were counted.

QUERY:

```
SELECT
rideable_type,
COUNT(rideable_type) AS count
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
GROUP BY rideable_type
```

RESULTS:

rideable_type	count
classic_bike	3167818
docked_bike	49367
electric_bike	3237631

- Number of times bikes used by both casual and members were counted.

QUERY:

```
SELECT
rideable_type,
member_casual,
COUNT(rideable_type) AS count
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
GROUP BY rideable_type, member_casual
ORDER BY rideable_type, member_casual
```

RESULTS:

rideable_type	member_casual	count
classic_bike	casual	1081141
classic_bike	member	2086677
docked_bike	casual	49367
electric_bike	casual	1220369
electric_bike	member	2017262

7.4 Annual Bike Ridership

- Initially month column was added to the dataset

QUERY:

```
SELECT
  *,
  CASE
    WHEN EXTRACT(MONTH FROM started_at) = 1 THEN 'January'
    WHEN EXTRACT(MONTH FROM started_at) = 2 THEN 'February'
    WHEN EXTRACT(MONTH FROM started_at) = 3 THEN 'March'
    WHEN EXTRACT(MONTH FROM started_at) = 4 THEN 'April'
    WHEN EXTRACT(MONTH FROM started_at) = 5 THEN 'May'
    WHEN EXTRACT(MONTH FROM started_at) = 6 THEN 'June'
    WHEN EXTRACT(MONTH FROM started_at) = 7 THEN 'July'
    WHEN EXTRACT(MONTH FROM started_at) = 8 THEN 'August'
    WHEN EXTRACT(MONTH FROM started_at) = 9 THEN 'September'
    WHEN EXTRACT(MONTH FROM started_at) = 10 THEN 'October'
    WHEN EXTRACT(MONTH FROM started_at) = 11 THEN 'November'
    WHEN EXTRACT(MONTH FROM started_at) = 12 THEN 'December'
  END AS month
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
```

- Then, number of rides during the period of June 2023 to June 2024 were counted.

QUERY:

```
SELECT
month,
member_casual
COUNT(month) AS ride_count
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
GROUP BY month
ORDER BY CASE WHEN month = "January" THEN 1
 WHEN month = "February" THEN 2
 WHEN month = "March" THEN 3
 WHEN month = "April" THEN 4
 WHEN month = "May" THEN 5
```

```
  WHEN month = "June" THEN 6
  WHEN month = "July" THEN 7
  WHEN month = "August" THEN 8
  WHEN month = "September" THEN 9
  WHEN month = "October" THEN 10
  WHEN month = "November" THEN 11
  WHEN month = "December" THEN 12
  END,
  Member_casual;
```

RESULTS:

month	member_casual	ride_count
January	casual	24461
January	member	120416
February	casual	47165
February	member	176009
March	casual	82552
March	member	219149
April	casual	131817
April	member	283242
May	casual	231047
May	member	378526
June	casual	602329
June	member	828061
July	casual	331428
July	member	436390
August	casual	311235
August	member	460682
September	casual	261670
September	member	404811
October	casual	177100
October	member	360095
November	casual	98396
November	member	264142
December	casual	51677
December	member	172416

7.5 Weekly Bike Ridership

- Number of rides during weekdays were counted.

QUERY:

```
SELECT
 day_of_week,
 member_casual,
```

```
 COUNT(day_of_week) AS ride_count
 FROM `bike-ridership-analysis.trip_data.bike_ridership_details_23_24`
 GROUP BY day_of_week
 ORDER BY CASE WHEN day_of_week = "Sunday" THEN 1
  WHEN day_of_week = "Monday" THEN 2
  WHEN day_of_week = "Tuesday" THEN 3
  WHEN day_of_week = "Wednesday" THEN 4
  WHEN day_of_week = "Thursday" THEN 5
  WHEN day_of_week = "Friday" THEN 6
  WHEN day_of_week = "Saturday" THEN 7
  END,
  member_casual;
```

RESULTS:

day_of_week	member_casual	ride_count
Sunday	casual	399998
Sunday	member	466807
Monday	casual	269454
Monday	member	564732
Tuesday	casual	264140
Tuesday	member	623990
Wednesday	casual	281705
Wednesday	member	653081
Thursday	casual	296563
Thursday	member	658496
Friday	casual	351709
Friday	member	594101
Saturday	casual	487308
Saturday	member	542732

7.6 Hourly Bike Ridership

- Number of rides during 00:00 to 23:00 hrs were counted.

QUERY:

```
SELECT
start_hour,
member_casual,
COUNT(start_hour) as ride_count
FROM bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details
GROUP BY start_hour, member_casual
ORDER BY start_hour, member_casual
```

RESULTS:

start_hour	member_casual	ride_count
0	casual	41423
0	member	39211

1	casual	26848
1	member	23186
2	casual	16371
2	member	13262
3	casual	9096
3	member	8912
4	casual	6764
4	member	9971
5	casual	12802
5	member	38440
6	casual	32442
6	member	117645
7	casual	57924
7	member	217480
8	casual	79618
8	member	272915
9	casual	80261
9	member	186303
10	casual	100546
10	member	167465
11	casual	128248
11	member	198219
12	casual	152516
12	member	225724
13	casual	158598
13	member	224949
14	casual	164616
14	member	226508
15	casual	182036
15	member	275571
16	casual	208129
16	member	372393
17	casual	225601
17	member	434343
18	casual	195785
18	member	343851
19	casual	145053
19	member	243208
20	casual	106444
20	member	172177
21	casual	87494
21	member	132152
22	casual	77338
22	member	98005
23	casual	54924
23	member	62049

7.7 Roundtrip analysis

- Initially sub dataset was prepared.

QUERY:

```
(SELECT
CONCAT(start_station_name, " to ", end_station_name) AS biker_routes,
member_casual
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
WHERE start_station_name IS NOT NULL
AND end_station_name IS NOT NULL
AND start_station_name = end_station_name)

UNION ALL

(SELECT
  CONCAT(
    COALESCE(start_station_name, CONCAT(start_lat, ',', start_lng)),
    ' to ',
    COALESCE(end_station_name, CONCAT(end_lat, ',', end_lng))
  ) AS biker_routes,
  member_casual
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
WHERE start_station_name IS NULL
AND end_station_name IS NULL
AND start_lat = end_lat
AND end_lng = start_lng)
```

- Then, top routes were counted.

QUERY:

```
SELECT *,
COUNT(*) AS trip_count
FROM `bike-ridership-analysis.trip_data.routetrip_membertype`
GROUP BY biker_routes
ORDER BY -trip_count
LIMIT 10
```

RESULTS:

biker_routes	trip_count
Streeter Dr & Grand Ave to Streeter Dr & Grand Ave	11452
DuSable Lake Shore Dr & Monroe St to DuSable Lake Shore Dr & Monroe St	9251
Michigan Ave & Oak St to Michigan Ave & Oak St	6061
41.89,-87.63 to 41.89,-87.63	4716
Millennium Park to Millennium Park	4421

Dusable Harbor to Dusable Harbor	4013
Montrose Harbor to Montrose Harbor	3927
DuSable Lake Shore Dr & North Blvd to DuSable Lake Shore Dr & North Blvd	3327
Theater on the Lake to Theater on the Lake	3197
41.79,-87.6 to 41.79,-87.6	3098

(Totally 2067 distinct round trips were identified)

- Then, number of round trips were made by members and casual riders were counted.

QUERY:

```
SELECT
member_casual,
COUNT(member_casual) AS count
FROM `bike-ridership-analysis.trip_data.routetrip_membertype`
GROUP BY member_casual
```

RESULTS:

member_casual	count
Casual	233782
Member	188434

7.8 Departure hotspots analysis

- Initially distinct start locations were listed.

QUERY:

```
SELECT
CONCAT(start_lat, "," ,start_lng) AS start_coordinates,
member_casual
FROM `bike-ridership-analysis.trip_data.23_24_divvy_bike_ridership_details`
```

- Then associated trips were counted.

QUERY:

```
SELECT *,
COUNT(member_casual) AS count
FROM `bike-ridership-analysis.trip_data.startcoordinates_membertype`
GROUP BY start_coordinates, member_casual
```

RESULTS: Totally 2190724 distinct start loactions were identified.

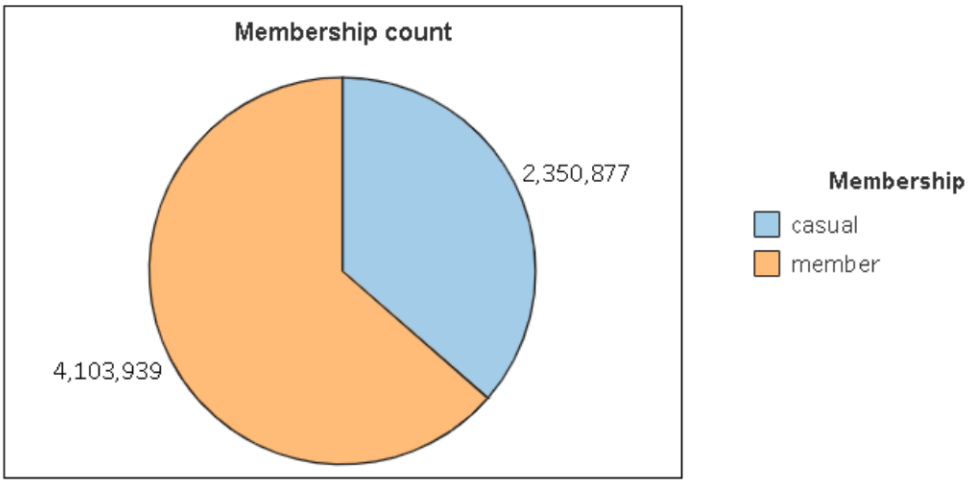
- Hotspots (where more than 5000 casual riders start their journey) were determined.

QUERY:

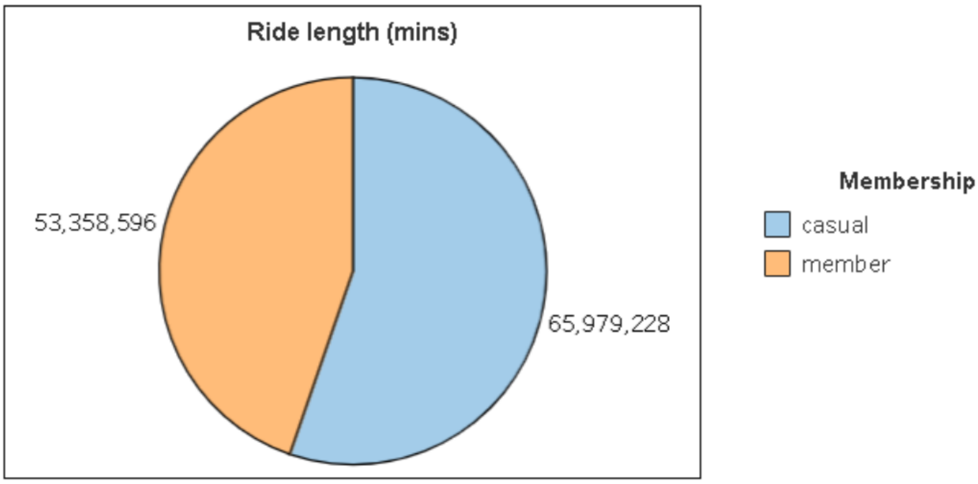
```
SELECT
start_coordinates,
count
FROM `bike-ridership-analysis.trip_data.startcoordinates_membertype_count`
WHERE member_casual = "casual"
AND count > 5000
```

RESULTS: Totally 62 hotspots were identified.

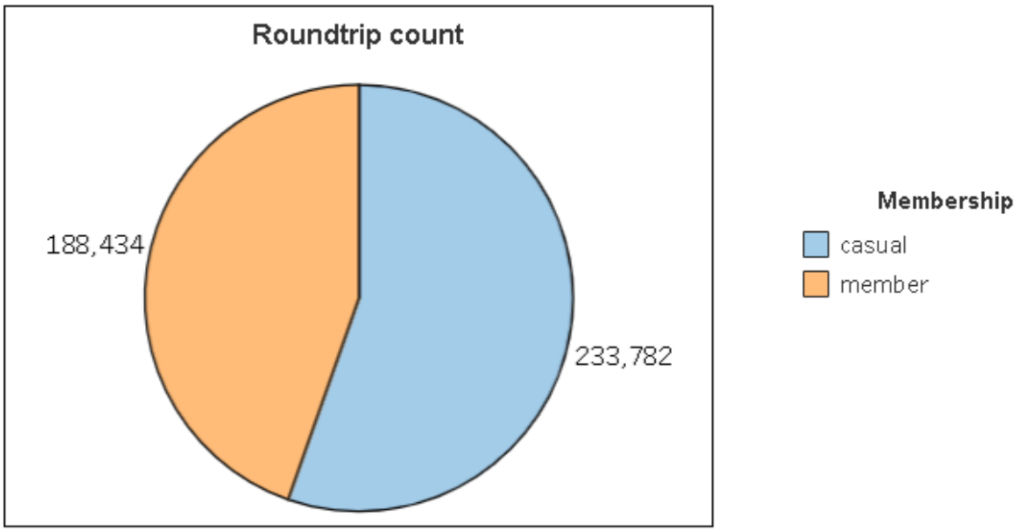
VISUALIZATION:



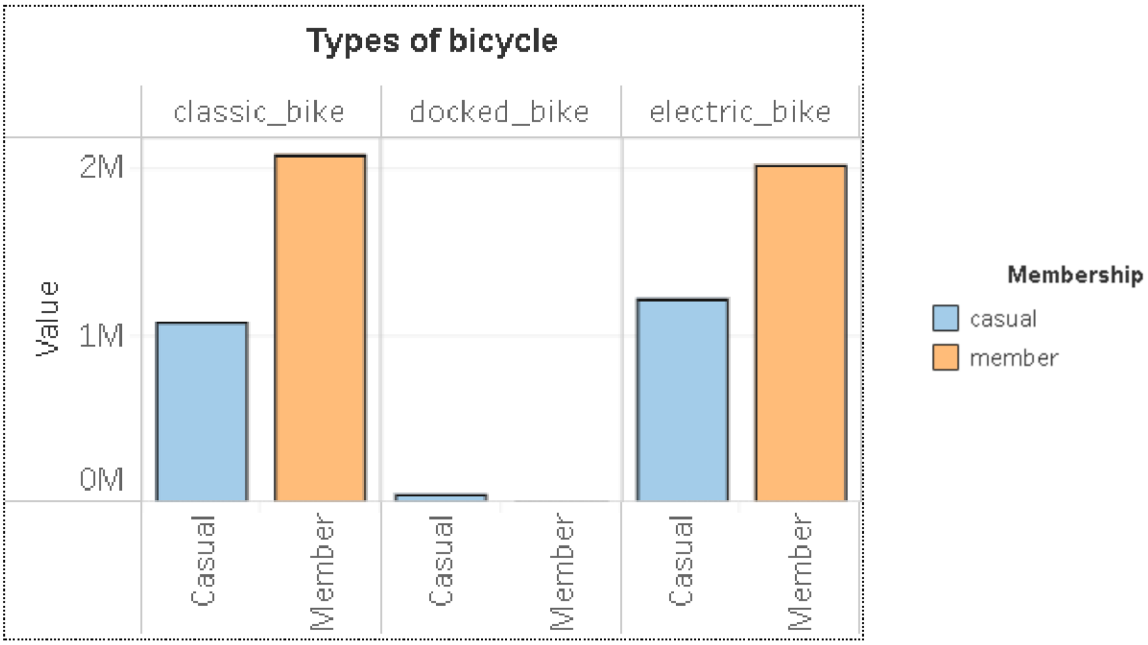
About, 36.4 % of constitutes casual riders and 63.6 % make up for members.



Despite a larger customer base, casual riders logged significantly more total ride time than members, accumulating approximately 65 million minutes compared to 53 million minutes for members.



Casual riders undertook approximately 23.3 million round trips, surpassing member round trips by 10.8%. Members completed roughly 18.8 million round trips.

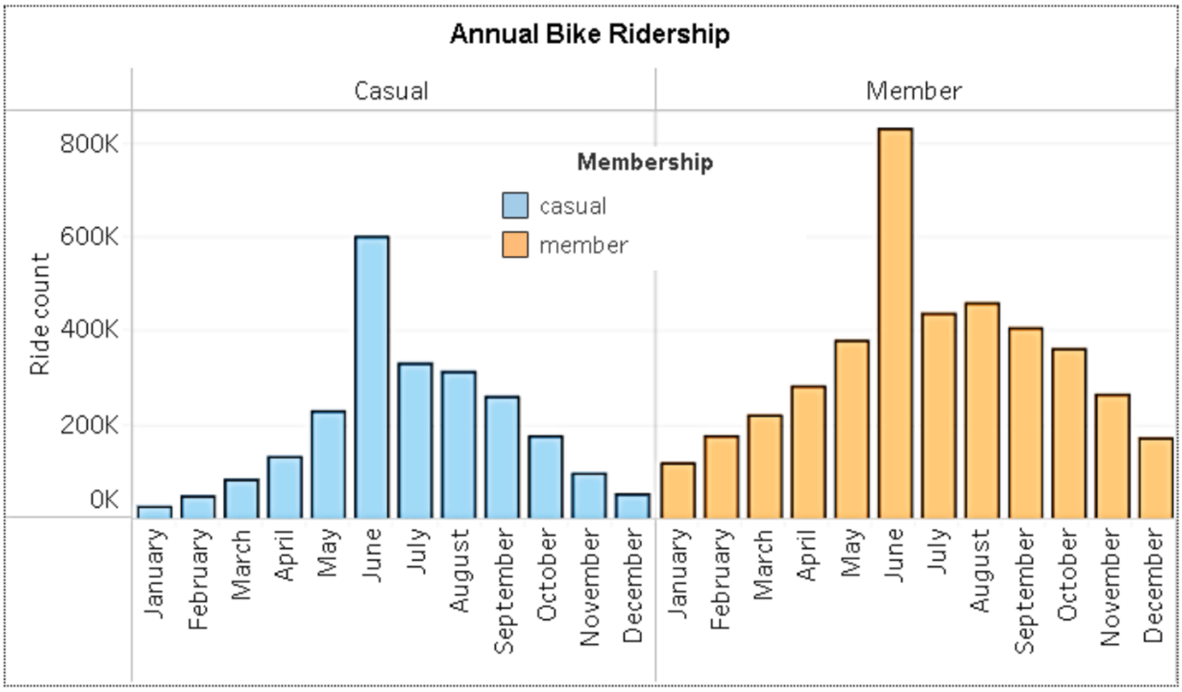


The data indicates a clear preference for electric and classic bikes among both member and casual riders. While electric bikes were the most popular choice overall, classic bikes were also widely utilized. Notably, docked bikes were predominantly used by casual riders. This suggests that docked bikes may be less conveniently located or desirable for longer rides compared to electric and classic bikes.

Electric Bikes: Both member and casual riders Favor electric bikes, accounting for 50.6% (3,237,631) of total rides. However, members utilize electric bikes slightly more at 2,017,262 compared to casual riders at 1,220,369.

Classic Bikes: Classic bikes are another popular choice, representing 49.08% (3,167,818) of total rides. Interestingly, members use classic bikes (2,086,677) slightly more than electric bikes (2,017,262) compared to casual riders (1,081,141).

Docked Bikes: Docked bikes are used almost exclusively by casual riders, constituting 0.3% (49,367) of total rides. Members didn't use docked bikes, suggesting a potential preference for specific bike types or locations where docked bikes are less prevalent.

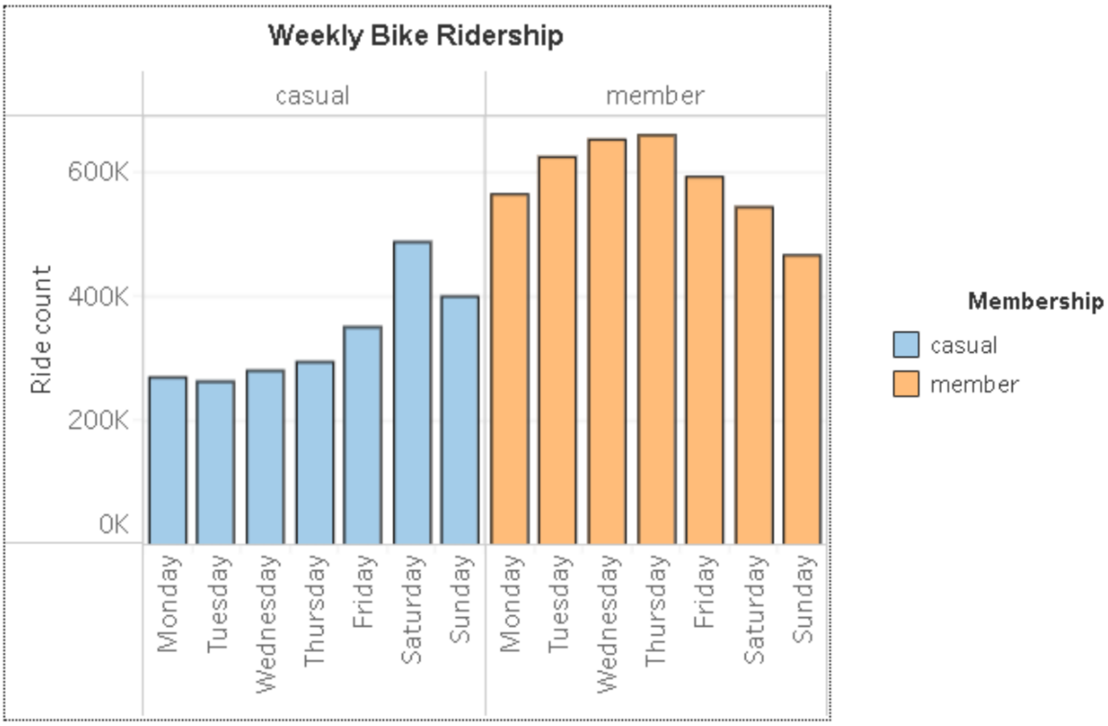


Both casual and member ridership exhibit a pronounced seasonal pattern, with peak usage occurring during the warmer months (April to October) and a significant decline during the colder months (November to March).

Casual Riders: Casual riders demonstrate a more pronounced seasonal variation. Their ridership increases rapidly from January to June, peaking in June at 602,329 rides. A slight decline is observed in July and August, followed by a more substantial decrease from September to December.

Members: Member ridership also follows a seasonal pattern but with less pronounced fluctuations. While there is an increase from January to June, the peak is less sharp compared to casual riders. Member usage remains relatively high during the summer months and declines gradually from September to December.

Overall, the data suggests a strong correlation between weather conditions and bike-sharing demand, particularly for casual riders. Understanding these seasonal patterns can be crucial for optimizing bike-sharing operations, inventory management, and marketing strategies.

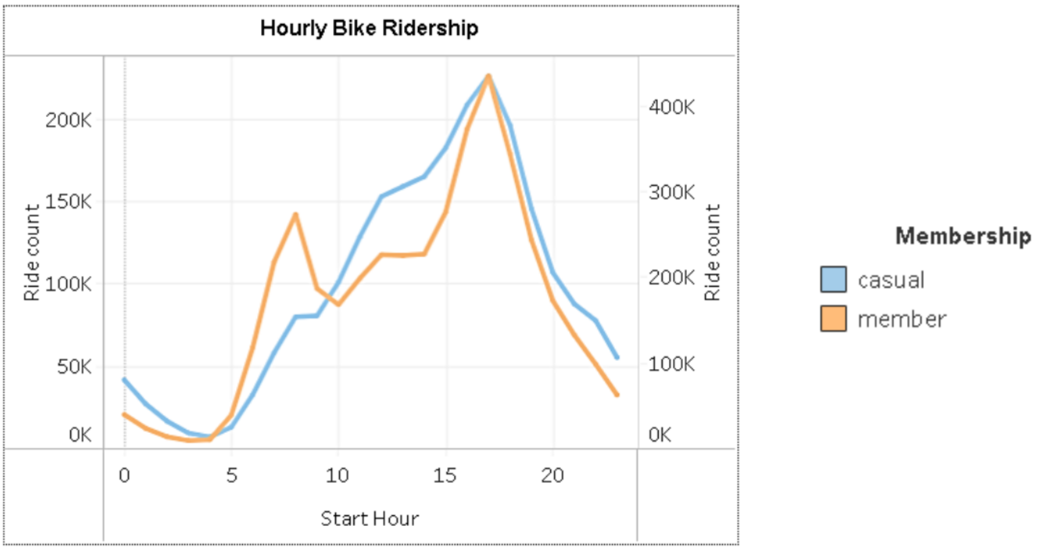


A distinct contrast emerges between casual and member riders in terms of weekday and weekend usage.

Members exhibit a consistent usage pattern throughout the week, with higher ridership on weekdays (Monday to Friday) compared to weekends. This suggests a significant portion of member rides are likely for commuting purposes. Casual riders demonstrate a pronounced preference for weekend usage, with Saturday and Sunday witnessing the highest ridership numbers. This indicates that recreational activities and leisure are primary motivations for casual riders.

Both member and casual riders exhibit peak usage on Friday and Saturday, suggesting potential overlap in usage patterns during these days.

This data highlights complementary usage patterns between the two customer segments. Understanding these differences is crucial for optimizing bike-sharing operations, resource allocation, and marketing strategies. Additionally, insights into weekday and weekend usage can inform targeted marketing campaigns and pricing strategies.



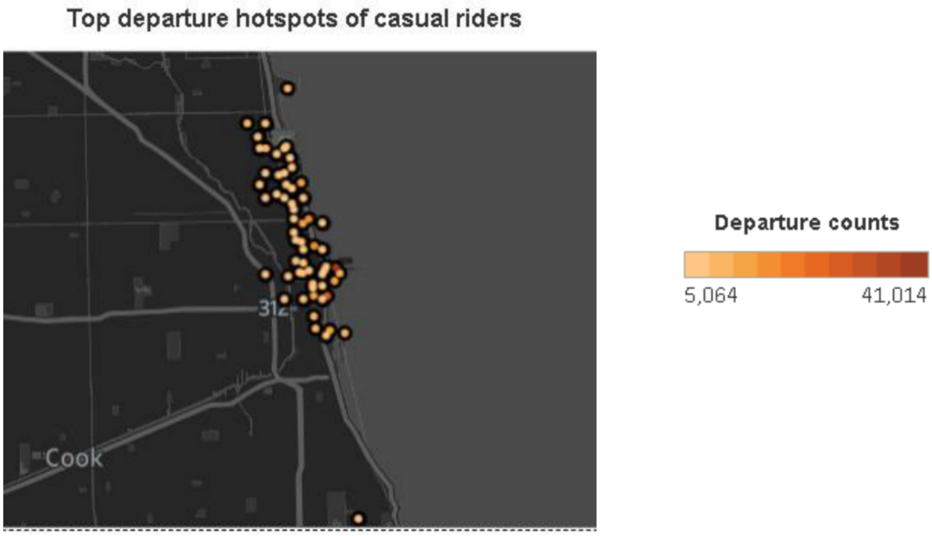
The data clearly indicates a bimodal distribution of ridership for both casual and member users. There are two distinct peak periods.

- **Morning Peak:** Between 6 AM and 9 AM, ridership increases significantly for both groups, suggesting a strong correlation with commuting patterns. Members exhibit a more pronounced peak during this period, likely due to a larger commuter base.
- **Evening Peak:** Between 4 PM and 7 PM, both groups experience another peak, suggesting leisure and recreational activities as primary motivators. Casual riders tend to have a more pronounced evening peak compared to members.

Nighttime hours (12 AM to 5 AM) consistently exhibit the lowest ridership for both groups, indicating minimal bike usage during these periods.

While not explicitly shown in the provided data, it is reasonable to assume that weekday ridership would be higher during peak hours for members, reflecting commuting patterns. Conversely, casual riders might exhibit higher weekend usage, especially during the evening peak.

By understanding peak usage hours, operators can strategically **distribute bikes across stations to ensure availability during high-demand periods**. Implementing **dynamic pricing** based on hourly demand can maximize revenue and optimize resource utilization. Targeted marketing campaigns can be developed to encourage off-peak usage or promote specific usage occasions (e.g., commuting, leisure).



The data reveals distinct clusters of high-departure locations, suggesting **popular starting points** for bike share journeys. These clusters likely correspond to areas with high population density, employment centres, or popular recreational destinations.

By identifying these high-departure areas, bike share operators can **focus marketing efforts** on casual riders in these specific locations. Tailored promotions and incentives can be offered to encourage increased usage and potential conversion to annual memberships.

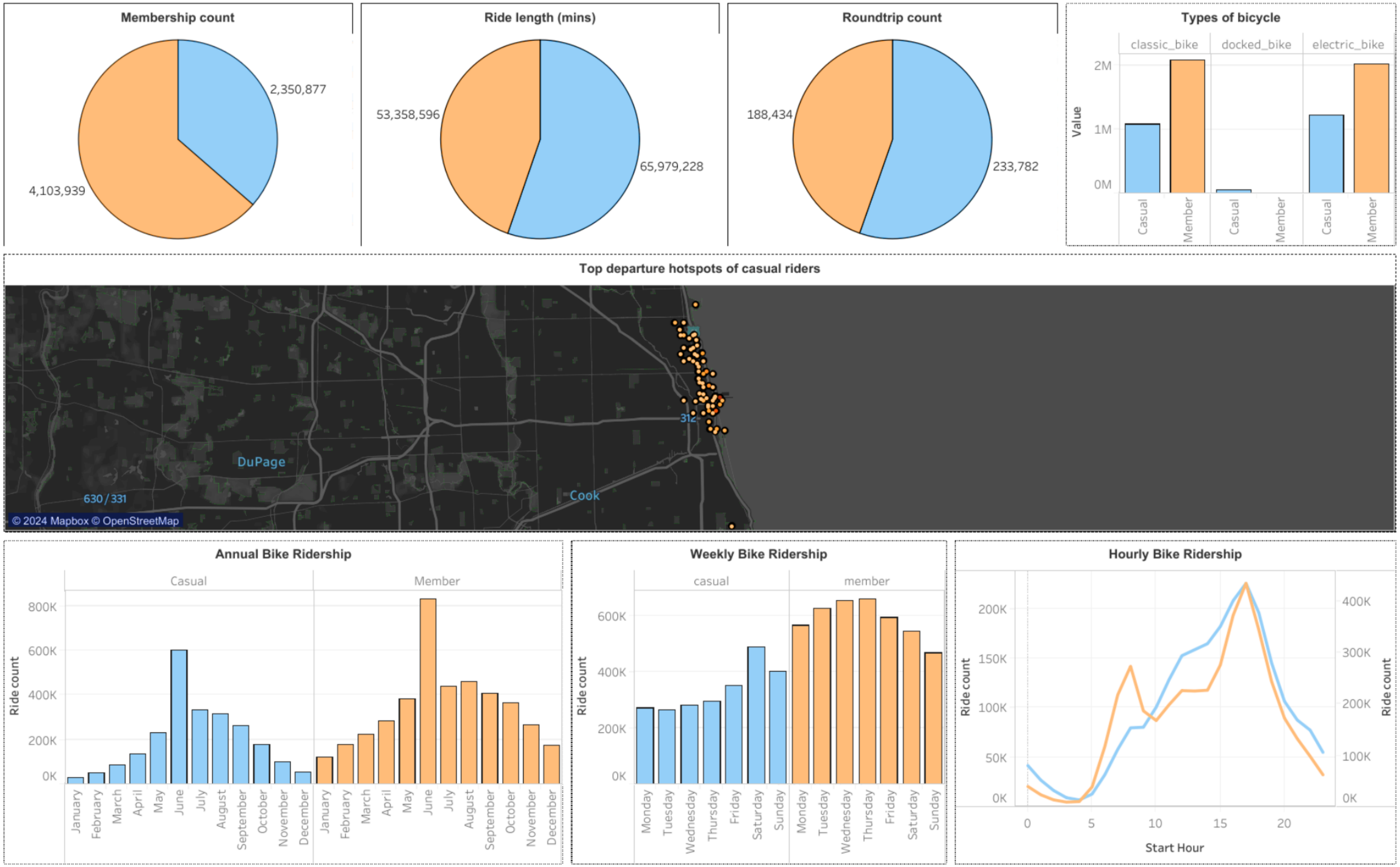
Implications for Targeted Marketing

- **Geo-fencing:** Implementing geo-fencing technology to target users within these high-departure areas with location-specific promotions and offers.
- **Partnerships:** Collaborating with local businesses and attractions in these areas to create bundled packages or exclusive offers for bike share users.
- **Data-Driven Campaign Optimization:** Continuously monitoring departure patterns to refine marketing efforts and allocate resources effectively.

A granular understanding of high-departure locations is essential for optimizing bike share operations and marketing strategies. By focusing on these areas, bike share operators can enhance customer experience, increase ridership, and drive revenue growth.

8 SHARE

CYCLISTIC BIKE RIDERSHIP DASHBOARD (Tableau)



CYCLISTIC BIKE RIDERSHIP PRESENTATION (PowerPoint)

Cyclistic

BIKE SHARE RIDERSHIP ANALYSIS

Dhanushwr K
Junior Data Analyst
28 July 2024

AGENDA

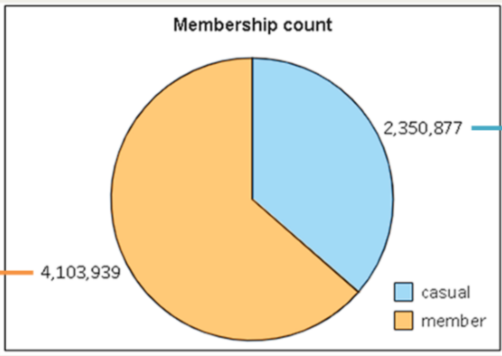
- Objective
- Membership analysis
- Ride duration analysis
- Roundtrip analysis
- Bike type analysis
- Annual bike ridership analysis
- Weekly bike ridership analysis
- Hourly bike ridership analysis
- Departure hotspots analysis
- Final insights
- Marketing strategies

OBJECTIVE

Identify behavioral disparities between casual and annual Cyclistic members to inform a **targeted marketing strategy** aimed at **member acquisition** and retention from existing casual riders.

How do annual members and casual riders use Cyclistic bikes differently?

MEMBERSHIP ANALYSIS

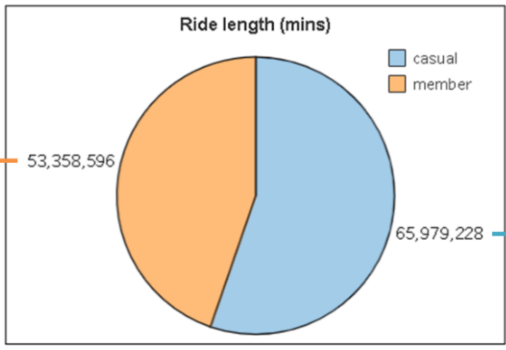


Casual riders comprise the remaining **36.4%**

Members account for **63.6%** of Cyclistic's customer base

Cyclistic's customer base is divided into two primary segments

RIDE DURATION ANALYSIS



Members account for **53 million minutes** approx.

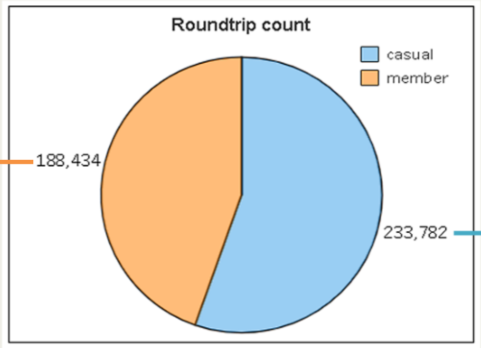
Casual riders logged approx. **65 million mins**

Despite a larger customer base,

casual riders logged significantly more total ride time than members

ROUNDTRIP ANALYSIS

Members completed roughly around **18.8 million trips**

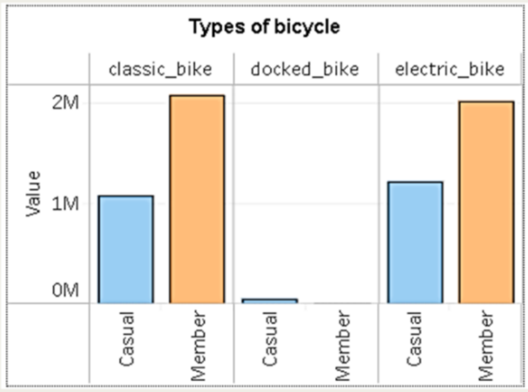


Casual riders took **23.3 million trips** approx.

Despite a larger customer base,

casual riders surpasses member round trips by 10.8%.

BIKETYPE ANALYSIS

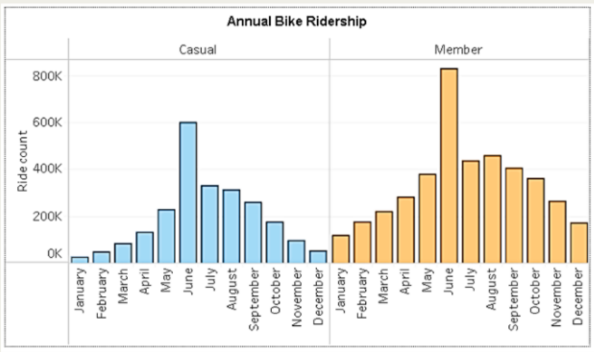


A clear preference for electric and classic bikes

among both member and casual riders

ANNUAL BIKE RIDERSHIP ANALYSIS

Peak usage occurring during the **warmer months (April to October)**



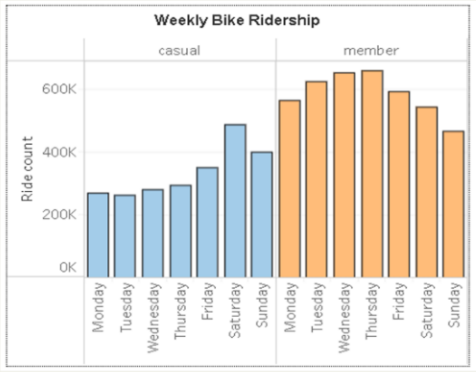
Significant decline during the **colder months (November to March).**

Both casual and member ridership

exhibit a pronounced seasonal pattern.

WEEKLY BIKE RIDERSHIP ANALYSIS

Casual riders demonstrate a pronounced **preference for weekend usage** (Saturday and Sunday).



Members exhibit **higher ridership on weekdays** (Monday to Friday) compared to weekends.

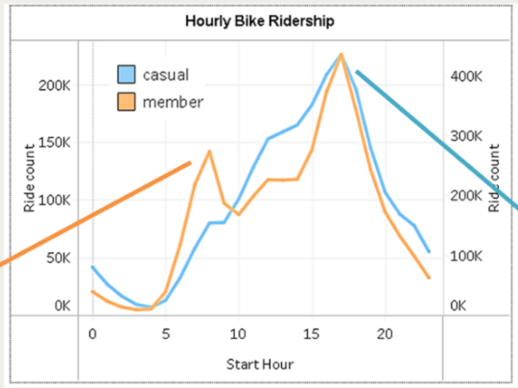
Complementary usage patterns

between the two customer segments

HOURLY BIKE RIDERSHIP ANALYSIS

Between 6 AM and 9 AM, **ridership increases significantly** for both groups

Members **exhibit a more pronounced peak** during this period.

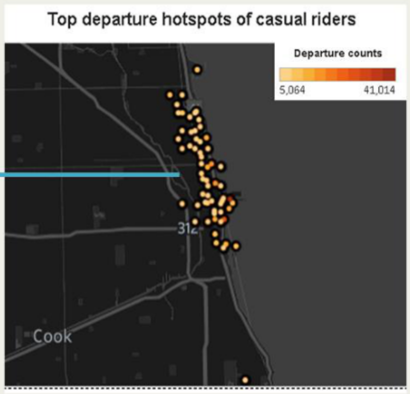


Between 4 PM and 7 PM, both groups **experience another peak**

Casual riders tend to have a **more pronounced evening peak** compared to members.

DEPARTURE HOTSPOTS ANALYSIS

Distinct clusters of **high-departure** locations



These clusters likely correspond to areas with high population density, employment centres, or popular recreational destinations.

Key differences

FINAL INSIGHTS

Customer *profile*

Casual riders represent **36.4%** of the customer base.

Members, constitutes about **63.6%** of the customer base

Ride *behavior*

Casual riders exhibit ***higher ride frequency and longer durations.***

Members display ***lower ride frequency but longer individual rides.***

Bike *preferences*

Both segments favour electric and classic bikes, with a slight inclination towards classic bikes for members.

Docked bikes are predominantly used by casual riders.

Departure *hotspots*

High-departure locations ***offer opportunities for targeted marketing*** to casual riders.

Marketing strategies

MARKETING STRATEGIES

Dynamic
Pricing

Implement membership fees are ***lower during off-peak hours and higher during peak hours.***

Optimize
Bike distribution

Align bike distribution with seasonal and daily usage patterns **to ensure availability** during peak periods.

Membership
Benefits

Advantages of membership, such as ***cost savings, exclusive access, and convenience.*** **Offer additional perks** for longer memberships, such as ***bike accessories, priority bike access, or exclusive events.***

Innovating
App functionality

Develop features that make it ***easier for casual users to transition to members***, such as one-click upgrades or personalized recommendations.

Thank you!

Dhanushwr K
Junior Data Analyst
28 July 2024

Key insights:

- **Customer Profile:** Casual riders represent 36.4% of the customer base but contribute significantly to ride volume and trip frequency. Members, while constituting 63.6%, exhibit more consistent usage patterns.
- **Ride Behaviour:** Casual riders exhibit higher annual ride frequency, while members demonstrate longer average ride durations. Casual riders demonstrate longer ride durations and a higher propensity for round trips. Members exhibit a stronger preference for weekday usage, indicative of commuting patterns. Casual riders demonstrate a leisure-oriented pattern with heightened weekend activity.
- **Bike Preferences:** Both segments Favor electric and classic bikes, with a slight inclination towards classic bikes for members. Docked bikes are predominantly used by casual riders.
- **Departure Hotspots:** High-departure locations offer opportunities for targeted marketing to casual riders.

9 ACT

Marketing strategies:

- **Convert Casual Riders:** Focus marketing efforts on casual riders, particularly those in high-departure areas, to encourage annual memberships. Offer incentives for longer ride durations and weekday usage.
- **Dynamic Pricing:** Implement a dynamic pricing model where membership fees are lower during off-peak hours and higher during peak hours.
- **Peak Hour Discounts:** Offer significant discounts for members who use bikes during peak hours.
- **Optimize Bike Distribution:** Align bike distribution with seasonal and daily usage patterns to ensure availability during peak periods.
- **Improved App Functionality:** Develop features that make it easier for casual users to transition to members, such as one-click upgrades or personalized recommendations.
- **Personalized Offers:** Use data to create targeted marketing campaigns based on user behaviour.

- **Highlight Membership Benefits:** Clearly communicate the advantages of membership, such as cost savings, exclusive access, and convenience.
- **Loyalty Programs:** Reward frequent casual users with discounts or free rides for upgrading to membership.
- **Additional Membership Perks:** Offer additional perks for longer memberships, such as bike accessories, priority bike access, or exclusive events.

Leverage Data: Continuously analyse ride data to identify emerging trends and refine marketing strategies.

Performance indicators:

After implementing marketing strategies monitor:

- Conversion rate of casual riders to members
- Average ride duration for both segments
- Bike utilization rates by type and location
- Customer retention rates for both segments

to measure the impact of the actions.