# Case Study: How Does a Bike-Share Navigate Speedy Success?

Name: Md Sahil

Date: June 2022

## Introduction:

This case study is a part of the Google Data Analytics Professional Certification Capstone project. In this case study, I am a junior data analyst working in the marketing analyst team at Cyclistic, a fictional bike-share company in Chicago. The director of the marketing team of Cyclistic believes that the future growth of the company depends on maximizing the number of annual memberships as annual members are much more profitable than casual riders. Therefore, the data analyst team has to identify how differently Cyclistic bikes are used by casual riders and annual members. From these insights, the team of Cyclistic will develop a new marketing strategy for converting casual riders into annual members.

### About the Company:

The bike-share offering program launched by Cyclistic in 2016. There are 5,824 bicycles with geotrack that are locked into a network of 692 stations across Chicago. The bikes can be unlocked from any of one station and returned to any other station in the system anytime.

Cyclistic also offers reclining bikes, hand tricycles, and cargo bikes. The majority of riders in the city choose Cyclistic's traditional bikes; about 8% of riders use the other assistive options. It is found that most of the Cyclistic users ride for leisure, but about 30% of users ride to commute to work each day.

There are 3 flexible pricing plans –

- single-ride passes
- full-day passes
- annual memberships

Customers who purchase single-ride or full-day passes are considered as casual riders. Customers who buy annual memberships are considered as Cyclistic members.

It is found that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs.

# PHASE 1: Asking the right question and Identifying business task

## Stakeholders:

- **Lily Moreno (The director of marketing and manage of analytics team):** responsible for the development of campaigns and initiatives to promote the bike-share program.
- **Cyclistic marketing analytics team:** responsible for collecting, analyzing, and reporting data.
- **Cyclistic executive team:** decide whether to approve the recommended marketing program.

## Goal:

The Goal of this analysis is to finding answers that helps to converting casual riders into annual members.

## Key Business tasks:

1) Key business tasks include finding answer of following 3 given question –
2) How do annual members and casual riders use Cyclistic bikes differently?
3) Why would casual riders buy Cyclistic annual memberships?

How can Cyclistic use digital media to influence casual riders to become members?

From the above 3 tasks my assigned task is finding the answer of the first question that is

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

Therefore, This case study only focuses on the first question.

# PHASE 2: Preparing the Data

## Data Source:

The data has been made available by Motivate International Inc. under this license.

Data source link: https://divvy-tripdata.s3.amazonaws.com/index.html

This data source contained previous 12 months of Cyclistic trip data of 2021.

This is public data that anyone can use to explore how different customer types are using Cyclistic bikes. But note that data-privacy issues prohibit using riders' personally identifiable information. This means that one won't be able to connect pass purchases to credit card numbers to determine if casual riders live in the Cyclistic service area or if they have purchased multiple single passes.

**Data details:**

There are 12 Zipped CSV file of 12 months.

There are nine columns in each CSV file containing data about all rides that took place in the year 2021:

**ride_id:** unique id for each ride
**rideable_type**: the type of the bike (docked_bike, electric_bike and classic_bike)
**started_at**: the date and time the ride started at
**ended_at**: the date and time the ride ended at
**start_station_name**: station's name the ride started at
**start_station_id**: station's ID the ride started at
**end_station_name**: station's name the ride ended at
**end_station_id**: station's ID the ride ended at
**member_casual**: casual rider or member rider

## PHASE 3 & 4: Process and Analyzing the Data

For the cleaning and transforming process I used SQL in SNOWFLAKE Data Platform. The following steps were taken:

**[for Full Transform and EDA follow the link:**

https://github.com/mdsahilmca20/PortfolioProjects/blob/main/Cyclistic%20Rider%20Analysis/CYCLISTIC_ANALYSIS_EDA.sql**]**

1. Creating database for this study in following way:

```
CREATE DATABASE CYCLISTIC_DATABASE_2021;
```

2. Creating 12 tables for each month as TRIP1, TRIP2, TRIP3, ……, TRIP12 respectively. Example of creating TRIP1 table given below:

```
-- CREATING TABLE FOR JANUARY 2021 TRIP DATA
CREATE TABLE TRIP1(
ride_id varchar(100) primary key,
rideable_type varchar(100),
started_at varchar(100),
ended_at varchar(100),
start_station_name varchar(100),
start_station_id varchar(100),
end_station_name varchar(100),
end_station_id varchar(100),
start_lat varchar(100),
start_lng varchar(100),
end_lat varchar(100),
```

```
end_lng varchar(100),
member_casual varchar(100)
);
```

3. Combining 12 tables as a whole in table called TRIP in following way:

```
-- COMBINING TABLES TRIP1, TRIP2, TRIP3, ......,TRIP12  FOR 2021 TRIP DATA AS A
WHOLE

CREATE TABLE TRIP_2021 AS SELECT * FROM
(SELECT * FROM TRIP1)
UNION
(SELECT * FROM TRIP2)
UNION
(SELECT * FROM TRIP3)
UNION
(SELECT * FROM TRIP4)
UNION
(SELECT * FROM TRIP5)
UNION
(SELECT * FROM TRIP6)
UNION
(SELECT * FROM TRIP7)
UNION
(SELECT * FROM TRIP8)
UNION
(SELECT * FROM TRIP9)
UNION
(SELECT * FROM TRIP10)
UNION
(SELECT * FROM TRIP11)
UNION
(SELECT * FROM TRIP12);
```

4. checked for and removed duplicates.

5. I made sure there are no extra unneeded spaces. For this I used the **TRIM** function.

6. Observe data where trip duration is 0 by calculating trip duration which is calculated by difference between ENDED_AT and STARTED_AT and also calculating distance in miles.

```
------------------------------- OBSERVE DATA WHERE TRIP DURATION IS 0


--SELECT * FROM TRIP_2021 WHERE
TIMESTAMPDIFF(minute,STARTED_AT,ENDED_AT)=0;
```

```
SELECT *,
TIMESTAMPDIFF(second,STARTED_AT,ENDED_AT) AS DURATION_SECOND,
SQRT(POWER(69.1 * ( END_LAT - START_LAT),  2) + POWER(69.1 * ( START_LNG  -
END_LNG )  * COS(END_LAT / 57.3), 2)) AS DISTANCE_MILES
FROM TRIP_2021
WHERE STARTED_AT = ENDED_AT;
```

7. Found many columns which have 0 duration, so we don't consider and discard.

8. changing type of STARTED_AT column to timestamp format where time format is 24-hour format.

9. Selecting records which is required for visualization and reporting purpose quarter wise and in some cases month and other factor wise( As in some quarter, result volume is too high to download). Example of Quarter-1 given below:

```
---------------------------------------- RETREIVING ONLY VALUABLE INFORMATION FOR
DIFFEREN ----------QUARTER, MONTH AND OTHER BASIC (FOR VISUALIZATION)
----- QUARTER1 RECORD
SELECT
RIDE_ID,
RIDEABLE_TYPE AS BIKE_TYPE,
STARTED_AT AS START_DATE_TIME,
TIMESTAMPDIFF(second,STARTED_AT,ENDED_AT)/60 AS DURATION_MINUTE,
START_STATION_NAME,
END_STATION_NAME,
SQRT(POWER(69.1 * ( END_LAT - START_LAT),  2) + POWER(69.1 * ( START_LNG  -
END_LNG )  * COS(END_LAT / 57.3), 2)) AS DISTANCE_MILES,
MEMBER_CASUAL AS RIDER_TYPE
FROM TRIP_2021
WHERE QUARTER(TO_TIMESTAMP(STARTED_AT)) = 1
AND STARTED_AT != ENDED_AT;
```

10. download previous step query results in CSV file.

## **Analyze data:**

---------- RETREIVE MONTHLY RIDE COUNT

```
SELECT MONTHNAME(TO_TIMESTAMP(STARTED_AT)) AS MONTH, COUNT(*) AS TOTAL_RIDE

FROM TRIP_2021 GROUP BY MONTHNAME(TO_TIMESTAMP(STARTED_AT)) ORDER BY COUNT(*) DESC;
```

**Output:**

Results | Data Preview | Open History

Query ID | SQL | 3.19s | 12 rows

Filter result... | Copy | Columns

| Row | MONTH | TOTAL_RIDE |
|---|---|---|
| 1 | Jul | 822410 |
| 2 | Aug | 804352 |
| 3 | Sep | 756147 |
| 4 | Jun | 729595 |
| 5 | Oct | 631226 |
| 6 | May | 531633 |
| 7 | Nov | 359978 |
| 8 | Apr | 337230 |
| 9 | Dec | 247540 |
| 10 | Mar | 228496 |
| 11 | Jan | 96834 |
| 12 | Feb | 49622 |

---------- RETREIVE RIDE COUNT DAYNAME WISE

```
SELECT DAYNAME(TO_TIMESTAMP(STARTED_AT)) AS DAY, COUNT(*) AS TOTAL_RIDE

FROM TRIP_2021 GROUP BY DAYNAME(TO_TIMESTAMP(STARTED_AT)) ORDER BY COUNT(*) DESC;
```

**Output:**

Results | Data Preview | Open History

Query ID | SQL | 1.81s | 7 rows

Filter result... | Copy | Columns

| Row | DAY | TOTAL_RIDE |
|---|---|---|
| 1 | Sat | 991047 |
| 2 | Sun | 857285 |
| 3 | Fri | 810508 |
| 4 | Wed | 756142 |
| 5 | Tue | 739905 |
| 6 | Thu | 737588 |
| 7 | Mon | 702588 |

---------- RETREIVE RIDE COUNT QUARTER WISE

```
SELECT QUARTER(TO_TIMESTAMP(STARTED_AT)) AS QUARTER, COUNT(*) AS TOTAL_RIDE

FROM TRIP_2021 GROUP BY QUARTER(TO_TIMESTAMP(STARTED_AT)) ORDER BY COUNT(*) DESC;
```

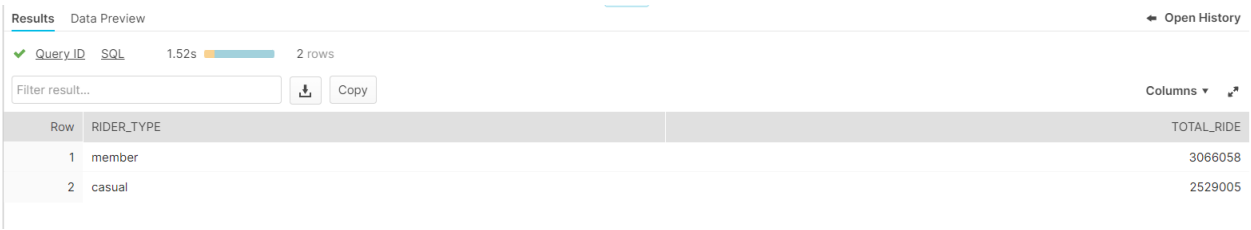
**Output:**

| Row | QUARTER | TOTAL_RIDE |
|---|---|---|
| 1 | 3 | 2382909 |
| 2 | 2 | 1598458 |
| 3 | 4 | 1238744 |
| 4 | 1 | 374952 |

---------- RETREIVE RIDE COUNT RIDER TYPE WISE

```
SELECT MEMBER_CASUAL AS RIDER_TYPE, COUNT(*) AS TOTAL_RIDE FROM TRIP_2021 GROUP BY MEMBER_CASUAL ORDER BY COUNT(*) DESC;
```

**Output:**




---------- RETREIVE RIDE COUNT BIKE TYPE WISE

```
SELECT RIDEABLE_TYPE AS BIKE_TYPE, COUNT(*) AS TOTAL_RIDE FROM TRIP_2021 GROUP BY RIDEABLE_TYPE ORDER BY COUNT(*) DESC;
```

**Output:**

Results Data Preview Open History

Query ID SQL 388ms 3 rows

Filter result... Copy Columns

| Row | BIKE_TYPE | TOTAL_RIDE |
|---|---|---|
| 1 | classic_bike | 3251028 |
| 2 | electric_bike | 2031692 |
| 3 | docked_bike | 312343 |

---------- RETRIEVE TRIP DURATION

```
SELECT TIMESTAMPDIFF(minute,STARTED_AT,ENDED_AT) AS DURATION_MINUTE
FROM TRIP_2021;
```

**Output:**

Results Data Preview Open History

Query ID SQL 802ms 5,595,063 rows

Filter result... Copy Columns

| Row | DURATION_MINUTE |
|---|---|
| 1 | 14 |
| 2 | 10 |
| 3 | 7 |
| 4 | 17 |
| 5 | 21 |
| 6 | 5 |
| 7 | 41 |
| 8 | 15 |
| 9 | 4 |
| 10 | 2 |
| 11 | 2 |
| 12 | 31 |

---------- RETRIEVE TRIP DURATION RIDER TYPE WISE

```
SELECT MEMBER_CASUAL AS
RIDER_TYPE,SUM(TIMESTAMPDIFF(minute,STARTED_AT,ENDED_AT))/60 AS
DURATION_HOUR

FROM TRIP_2021 GROUP BY MEMBER_CASUAL

ORDER BY SUM(TIMESTAMPDIFF(minute,STARTED_AT,ENDED_AT))/60 DESC;
```

**Output:**

| Row | RIDER_TYPE | DURATION_HOUR |
|---|---|---|
| 1 | casual | 1348854.383333 |
| 2 | member | 696673.783333 |

---------- RETRIEVE TRIP DURATION BY RIDER TYPE AND BIKE TYPE

```
SELECT MEMBER_CASUAL AS RIDER_TYPE, RIDEABLE_TYPE AS BIKE_TYPE,
SUM(TIMESTAMPDIFF(minute,STARTED_AT,ENDED_AT))/60 AS DURATION_HOUR

FROM TRIP_2021

GROUP BY MEMBER_CASUAL, RIDEABLE_TYPE

ORDER BY MEMBER_CASUAL,
SUM(TIMESTAMPDIFF(minute,STARTED_AT,ENDED_AT))/60 DESC;
```

**Output:**

Results Data Preview — Open History

Query ID SQL 1.78s 6 rows

Filter result... Copy Columns

| Row | RIDER_TYPE | BIKE_TYPE | DURATION_HOUR |
|---|---|---|---|
| 1 | casual | classic_bike | 610730.366667 |
| 2 | casual | docked_bike | 424233.333333 |
| 3 | casual | electric_bike | 313890.683333 |
| 4 | member | classic_bike | 467873.183333 |
| 5 | member | electric_bike | 228800.566667 |
| 6 | member | docked_bike | 0.033333 |

---------- COVERED DISTANCE BY EACH RIDE

```
SELECT

RIDE_ID,

SQRT(POWER(69.1 * ( END_LAT - START_LAT),  2) + POWER(69.1 * ( START_LNG  -
END_LNG )  * COS(END_LAT / 57.3), 2)) AS DISTANCE_MILES

FROM TRIP_2021

WHERE DISTANCE_MILES IS NOT NULL

ORDER BY 1 DESC;
```

**Output:**

Results Data Preview Open History

Query ID SQL 2.69s 5,590,292 rows (5,002,436 shown)

Filter result... Copy Columns

| Row | RIDE_ID | DISTANCE_MILES |
|---|---|---|
| 1 | FFFFFF1C07792282 | 0.7671322822 |
| 2 | FFFFFF0C829D3E7A | 7.048935555 |
| 3 | FFFFFC1045B11550 | 4.053734006 |
| 4 | FFFFF59D0A499AFE | 1.275468844 |
| 5 | FFFFF07D3B1BAF3F | 1.005722892 |
| 6 | FFFFEE4EEC2AFE0A | 5.252927867 |
| 7 | FFFFEDAAB98A2B5B | 1.061720687 |
| 8 | FFFFEC4F43C8C2FF | 0.9048796641 |
| 9 | FFFFEB6E60688D1D | 1.474890554 |
| 10 | FFFFE2898F5477B5 | 2.212842529 |
| 11 | FFFFE202FEF67F92 | 0.7146229877 |
| 12 | FFFFE1F4CF59EB3F | 2.53485606 |

---------- LOOKING FOR STATION NAMES FROM WHERE MAXIMUM RIDES START

```
SELECT
START_STATION_NAME,
START_STATION_ID,
COUNT(*) AS TOTAL_RIDE
FROM TRIP_2021
GROUP BY START_STATION_ID,START_STATION_NAME
ORDER BY 3 DESC;
```

**Output:**

Query ID SQL 342ms 852 rows

Filter result... Copy Columns

| Row | START_STATION_NAME | START_STATION_ID | TOTAL_RIDE |
|---|---|---|---|
| 1 | NULL | NULL | 690806 |
| 2 | Streeter Dr & Grand Ave | 13022 | 82715 |
| 3 | Michigan Ave & Oak St | 13042 | 44348 |
| 4 | Wells St & Concord Ln | TA1308000050 | 43610 |
| 5 | Millennium Park | 13008 | 42223 |
| 6 | Clark St & Elm St | TA1307000039 | 41218 |
| 7 | Wells St & Elm St | KA1504000135 | 37690 |
| 8 | Theater on the Lake | TA1308000001 | 36840 |
| 9 | Kingsbury St & Kinzie St | KA1503000043 | 33581 |
| 10 | Clark St & Lincoln Ave | 13179 | 33382 |
| 11 | Clark St & Armitage Ave | 13146 | 32904 |
| 12 | Wabash Ave & Grand Ave | TA1307000117 | 32783 |
| 13 | Dearborn St & Erie St | 13045 | 32156 |
| 14 | Wells St & Huron St | TA1306000012 | 31415 |

Results shows null station as some station name not given. So we only consider station name which is given.

---------- LOOKING FOR STATION NAMES FROM WHERE MAXIMUM RIDES END

```
SELECT
END_STATION_NAME,
END_STATION_ID,
COUNT(*) AS TOTAL_RIDE
FROM TRIP_2021
GROUP BY END_STATION_ID,END_STATION_NAME
ORDER BY 3 DESC;
```

**Output:**

Results  Data Preview  Open History

Query ID  SQL  316ms  846 rows

Filter result...  Copy  Columns

| Row | END_STATION_NAME | END_STATION_ID | TOTAL_RIDE |
|---|---|---|---|
| 1 | NULL | NULL | 739170 |
| 2 | Streeter Dr & Grand Ave | 13022 | 83390 |
| 3 | Michigan Ave & Oak St | 13042 | 44834 |
| 4 | Wells St & Concord Ln | TA1308000050 | 43850 |
| 5 | Millennium Park | 13008 | 42933 |
| 6 | Clark St & Elm St | TA1307000039 | 40531 |
| 7 | Wells St & Elm St | KA1504000135 | 37348 |
| 8 | Theater on the Lake | TA1308000001 | 37046 |
| 9 | Clark St & Lincoln Ave | 13179 | 33295 |
| 10 | Wabash Ave & Grand Ave | TA1307000117 | 33132 |
| 11 | Kingsbury St & Kinzie St | KA1503000043 | 32951 |
| 12 | Dearborn St & Erie St | 13045 | 32534 |
| 13 | Clark St & Armitage Ave | 13146 | 32328 |

Results shows null station as some station name not given. So we only consider station name which is given.

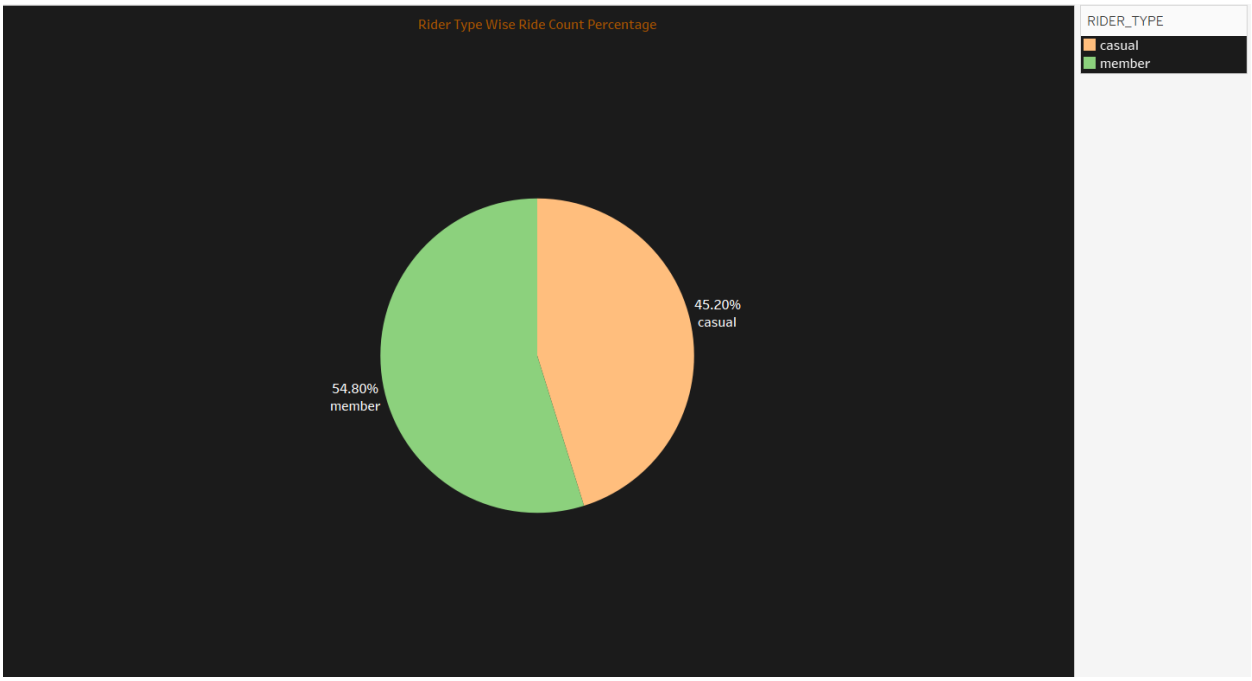
# PHASE 5: Visualizations

I used Tableau Public to run some further analysis and generate visualizations that support the key findings in the analysis.

**For full visualization follow the link:**

https://public.tableau.com/app/profile/md.sahil/viz/CYCLISTIC_16648956615920/Dashboard1

## VIZ-1

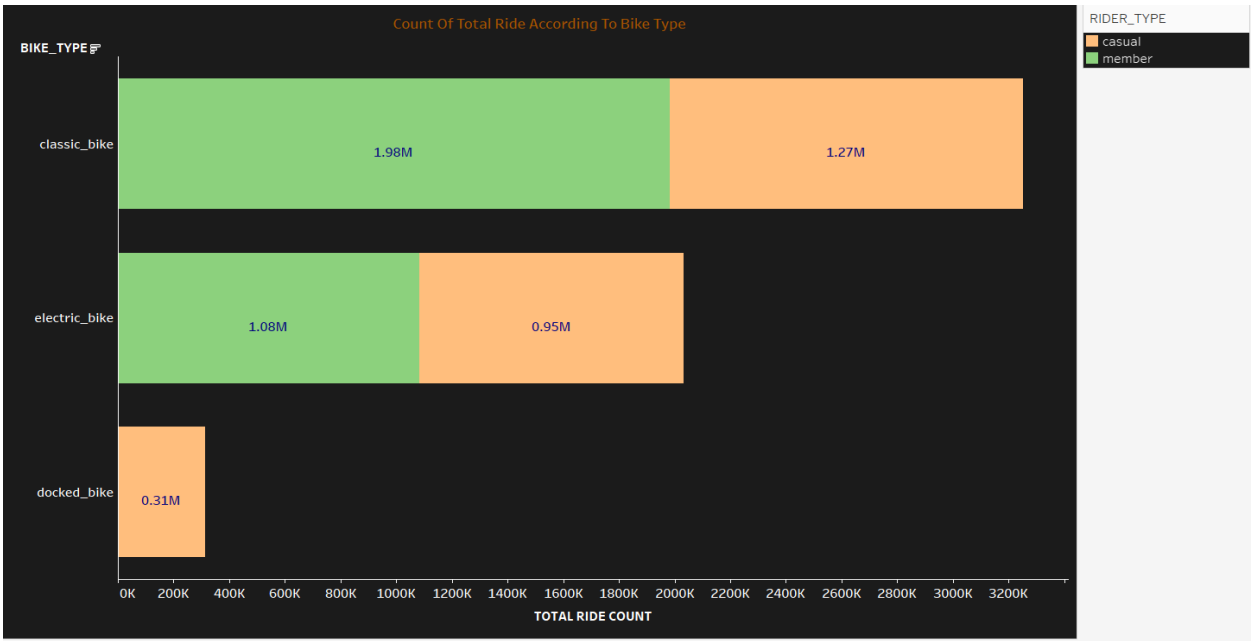



It's clearly shows that

Member among rider: 54.80% whereas

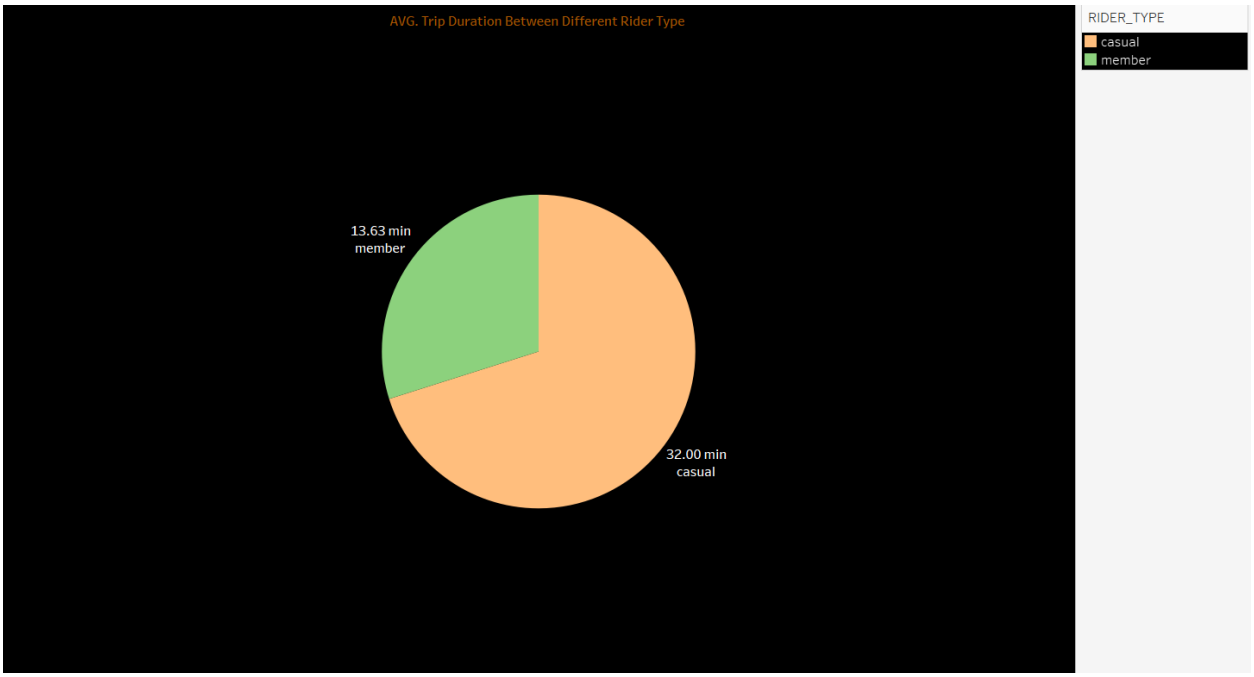
Casual among rider: 42.50%

## VIZ-2



From the visualization it is clear that docked_bike is least used bike and only used by casual member. Whereas classic_bike is most used bike and after that electric_bike is next most used bike. In both cases number of member is higher than casual rider.

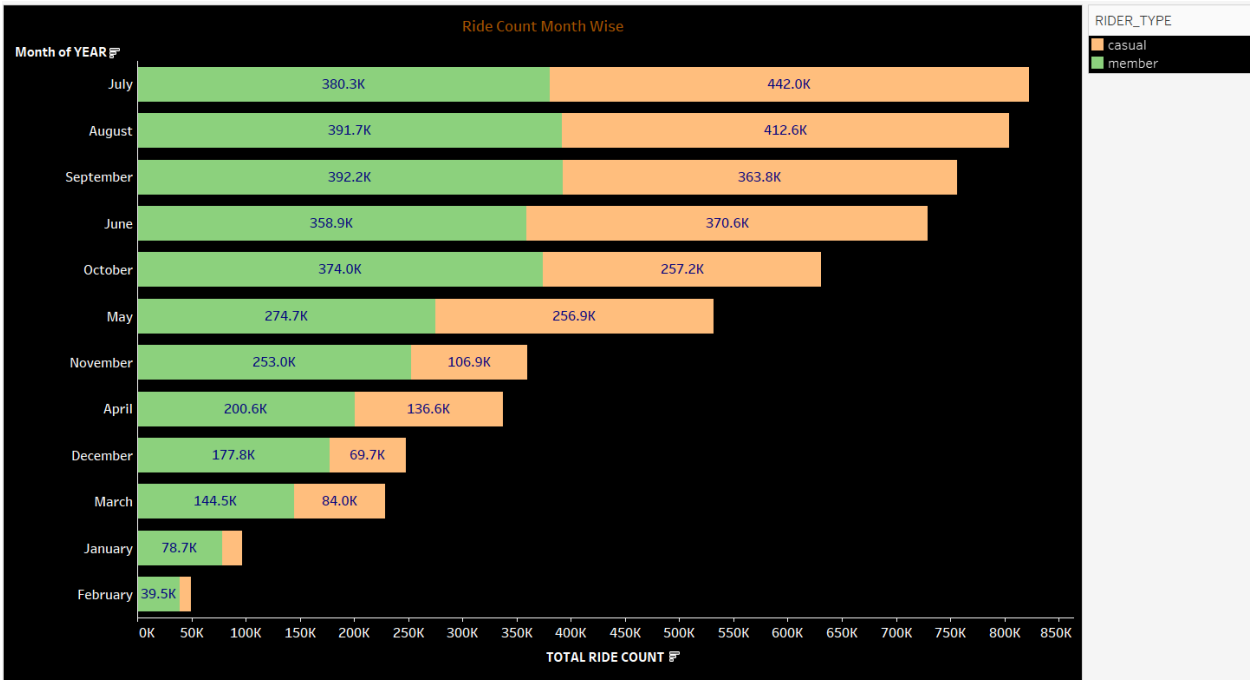
## VIZ-3




From above visualization it is clear that

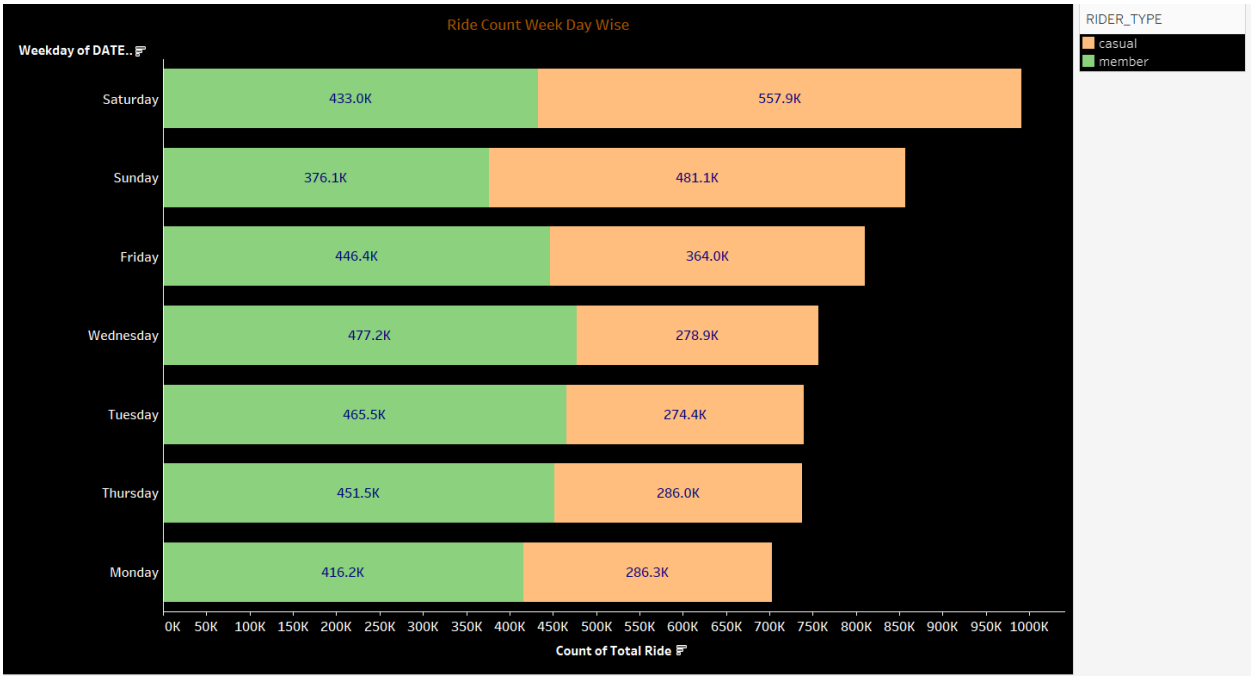
Average member trip duration (13.63 min) < Average casual trip duration (32 min)

## VIZ-4



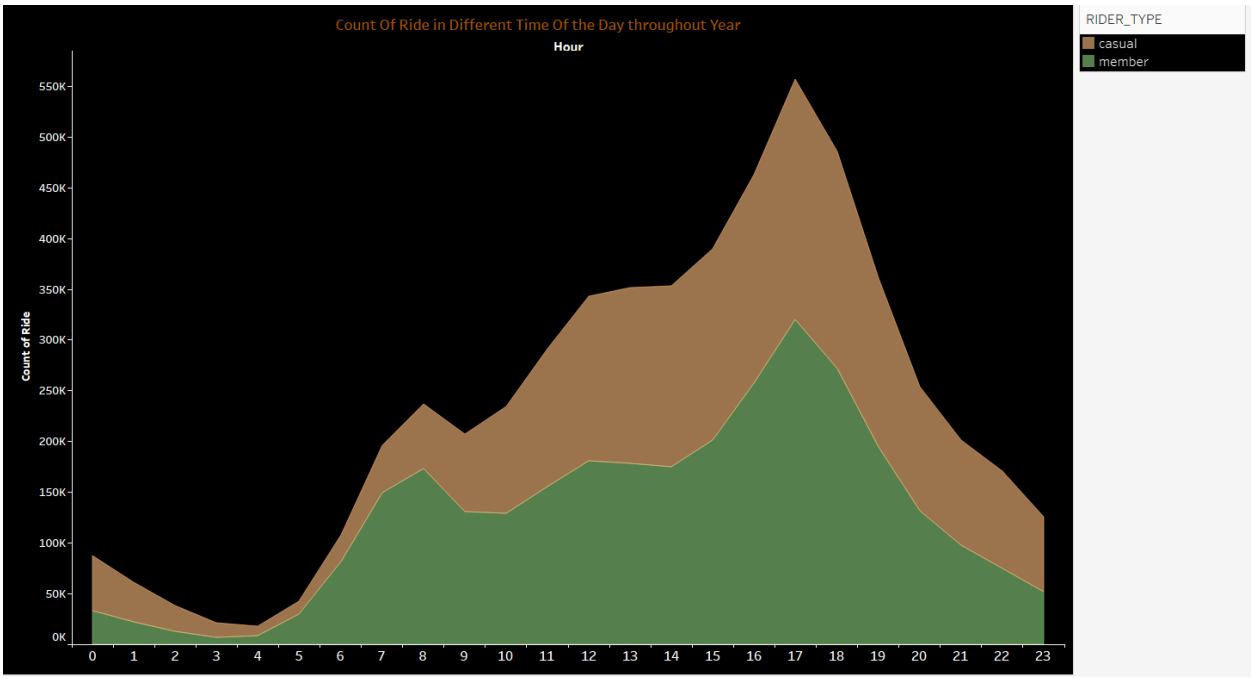

From the above visualization it is clear that highest rides counts months are July, august, September, June. And least rides count months are February and January.

## VIZ-5



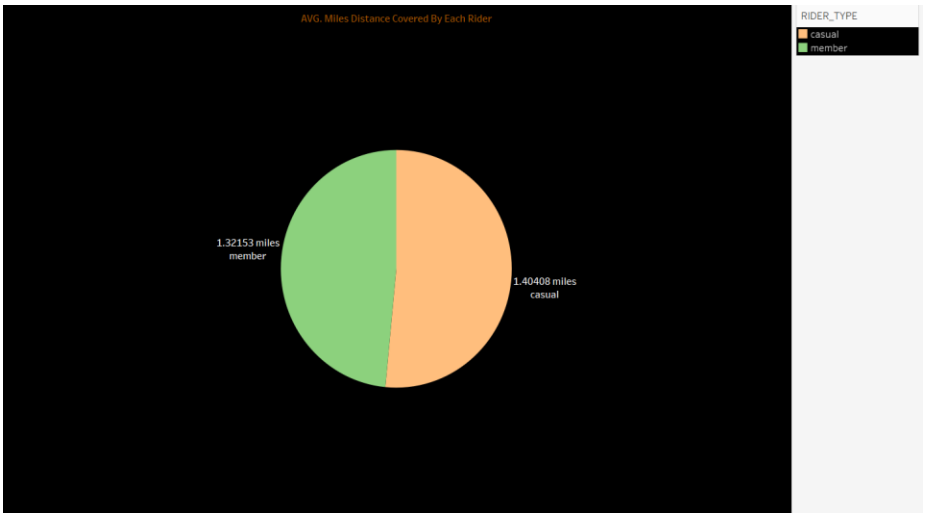
From the above visualization it is clear that highest rides counts days are Sunday and Saturday. And least rides count days are Monday and Tuesday.

## VIZ-6



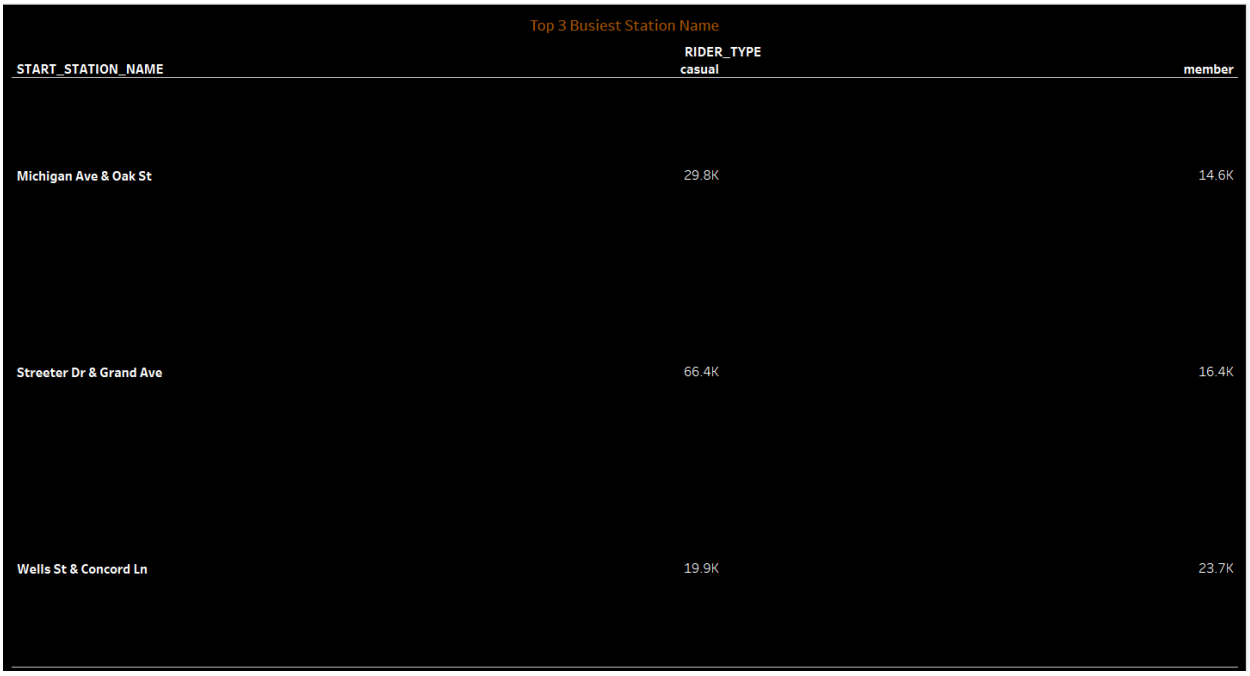

From the above visualization it is clear that highest rides counts hours are 16,17,18,19. And least rides count hours are 3,4.

## VIZ-7



Average distance covered by member (1.31 miles) < Average distance covered by member (1.40miles)

## VIZ-8




Top 3 busiest stations are Michigan Ave & Oak St, Streeter Dr & Grand Ave and Wells St & Concord Ln.

# PHASE 6: Act

The number of trips by casual riders is highest during the months of July(maximum), August, September and June where the maximum trips happen on Weekends and mostly in the Afternoon. Maximum casual trips occur at 5 PM. Average casual trip duration are more than double of average member trip duration.

My top 4 recommendations are:

1. **Marketing campaign shall be done during the months of June, July, August and September.**
2. **Weekends should be prioritized when it comes to scheduling ads. The best time for ads is the afternoon and evening time ie. 4 PM - 7 PM.**
3. **Marketing strategies should emphasize on "Casual riders with more number of rides with shorter ride length".**
4. **Top 3 station for ad campaign are Michigan Ave & Oak St, Streeter Dr & Grand Ave and Wells St & Concord Ln.**