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

Scenario:

The director of marketing believes the company's future success depends on maximizing the number of annual memberships. Therefore, the team wants to understand how casual riders and annual members use Cyclistic bikes differently. From these insights, the team will design a new marketing strategy to convert casual riders into annual members.

Characters and teams

● **Cyclistic:** A bike-share program that features more than 5,800 bicycles and 600 docking stations.

● **Lily Moreno:** The director of marketing and the manager. Moreno is responsible for the development of campaigns and initiatives to promote the bike-share program.

● **Cyclistic marketing analytics team:** A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide Cyclistic marketing strategy.

● **Cyclistic executive team:** The notoriously detail-oriented executive team will decide whether to approve the recommended marketing program.

Observations:

To understand how casual riders and annual members use bikes differently. The end goal is to convert the casual riders to memberships.

Pricing plans:

- Single-ride passes (Casual riders)
- Full-day passes (Casual riders)
- Annual memberships (Cyclistic members)

Membership riders are bringing profits to the company compared to casual riders. And membership will be the key factor for the company's future growth. Now the company targets the new customers.

Three questions will guide the future marketing program:

- How do annual members and casual riders use Cyclistic bikes differently?
- Why would casual riders buy Cyclistic annual memberships?
- How can Cyclistic use digital media to influence casual riders to become members?

Ask

What is the problem you are trying to solve?

We are trying to solve a business problem to find how to make the monthly and casual riders to annual membership and what will make them to do it. Then I should make sure that what type of information is available, how I should organize it like total number of customers, subscriptions types,

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

user's daily usage of bikes, monthly usages etc. And by this information to find the possible pattern and unusual things. At last, to give the right decision to make.

The first question we are trying to answer is: How do annual members and casual riders use Cyclistic bikes differently? Let's see about it elaborately in this article.

Prepare

From here the Datasets can be downloaded. For this case study, the data from period January 2023 to December 2023 have been taken. Here are some of the questions to proceed and to check the credibility of the data.

- How is the data organized?

  Each file is in CSV file format. The file has all the data which related to rides from January 1st to December 31st. Each file has 13 columns which starts with ride_id which is unique, rideable_type which says which type of bike have been used by the rider, started_at which denotes the started time similary_ended_at, start_station_name, member or casual rider etc. and each file has about 100K to 500K rows.

- Are there issues with bias or credibility in this data?

  Since, the data is from the company, it is trustable and it satisfies the ROCCC (Reliable Original Comprehensive Current Cited) method. There is no problem with the privacy and security. There is no duplicate value found and the data integrity is achieved.

- How does it help you answer your question?

  The data has different types of data starting from the type of vehicle the people use, how much hours they ride and whether it is weekly or daily etc. which helps in making the right decision and prediction.

Business tasks:

The data is from trustable party and it is stored in both local and online storage medium. From the data, it found that some data is not in the proper form. Some of the data contains mixed data types such as integer and character. It has to be consult with the stakeholders and should take an action accordingly.

Process

- What tools are you choosing and why?

  I'm choosing the tool SQL, Python and Tableau to work and manipulate the process. Python is easy in merging multiple filed and cleaning the data. SQL is used to manipulate the data. For the visualization, Tableau is used.

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

Merging the data:

```
import pandas as pd
import numpy as np  # numpy for numerical operations
import matplotlib.pyplot as plt  # matplotlib for data visualization
import seaborn as sns  # seaborn for enhanced data visualization
```

```
df = pd.concat(
    map(pd.read_csv, ['202301-divvy-tripdata.csv', '202302-divvy-tripdata.csv' , '202303-divvy-tripdata.csv' ,
                      '202304-divvy-tripdata.csv', '202305-divvy-tripdata.csv', '202306-divvy-tripdata.csv',
                      '202307-divvy-tripdata.csv', '202308-divvy-tripdata.csv', '202309-divvy-tripdata.csv',
                      '202310-divvy-tripdata.csv' , '202311-divvy-tripdata.csv' , '202312-divvy-tripdata.csv']),
    ignore_index=True)
print(df)
```

By using pandas library, the CSV files were merged and assigned to a variable 'df' in a dataframe format.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5719877 entries, 0 to 5719876
Data columns (total 13 columns):
 #   Column              Dtype
---  ------              -----
 0   ride_id             object
 1   rideable_type       object
 2   started_at          object
 3   ended_at            object
 4   start_station_name  object
 5   start_station_id    object
 6   end_station_name    object
 7   end_station_id      object
 8   start_lat           float64
 9   start_lng           float64
 10  end_lat             float64
 11  end_lng             float64
 12  member_casual       object
dtypes: float64(4), object(9)
memory usage: 567.3+ MB
```

The merged file is exported into CSV file format for visualization purpose.

```
df.to_csv('all_data_2023.csv', index = None)
```

- Have you ensured your data's integrity?

  It is always a best practice to make a copy of a dataset. Deep copy creates a completely independent copy of the original object. The df1 stores a copy of the actual dataframe df.

```
import copy
df1 = copy.deepcopy(df)
```

  Next is to find is there any duplicate values in the dataframe. Duplicates may affect various operations and analyses. We can use the **duplicated()** method to identify duplicate rows or **drop_duplicates()** method to remove duplicate rows from the DataFrame.

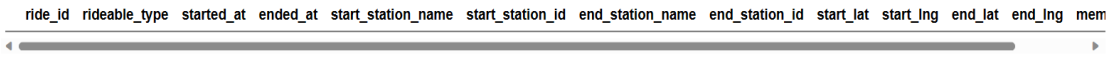
Case Study: How Does a Bike-Share Navigate Speedy Success?

```
df1.duplicated()
```

```
0          False
1          False
2          False
3          False
4          False
           ...
5667712    False
5667713    False
5667714    False
5667715    False
5667716    False
Length: 5667717, dtype: bool
```

Since we have huge data, it is difficult to find for each line. So, by passing it inside a dataframe, we can get the actual duplicate values.

```
df1[df1.duplicated()] == True
```



- What steps have you taken to ensure that your data is clean? How can you verify that your data is clean and ready to analyze?

The data has the record of the month January 2022 to December 2022. The data are organized but inconsistency. The data is not complete and adequate. Start Station name, start Station ID, end station name, end station ID has blank value.

```
df1.isnull().sum() # finding the null values
```

```
ride_id                 0
rideable_type           0
started_at              0
ended_at                0
start_station_name    875716
start_station_id      875848
end_station_name      929202
end_station_id        929343
start_lat               0
start_lng               0
end_lat                6990
end_lng                6990
member_casual           0
dtype: int64
```

In Ride ID, some values are completed and other values are in short notation. The whole data doesn't contain any duplicate values. We can treat the empty values as NULL but since for analyzing purpose (in POSTGRESQL), we can drop the columns which won't needed and export it to the .csv file

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

```
df1.drop(['start_station_name','start_station_id','end_station_name','end_station_id',
          'start_lat','start_lng','end_lat','end_lng'],axis=1, inplace=True)
df1
```

	ride_id	rideable_type	started_at	ended_at	member_casual
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	member
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	member
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	casual
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	member
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	member
...	...	...	...	...	...
5719872	F74DF9549B504A6B	electric_bike	2023-12-07 13:15:24	2023-12-07 13:17:37	casual
5719873	BCDA66E761CC1029	classic_bike	2023-12-08 18:42:21	2023-12-08 18:45:56	casual
5719874	D2CF330F9C266683	classic_bike	2023-12-05 14:09:11	2023-12-05 14:13:01	member
5719875	3829A0D1E00EE970	electric_bike	2023-12-02 21:36:07	2023-12-02 21:53:45	casual
5719876	A373F5B447AEA508	classic_bike	2023-12-11 13:07:46	2023-12-11 13:11:24	member

5719877 rows × 5 columns

```
null_percet = 100*(df1.isnull().sum()/len(df1.index))
null_percet
```

```
ride_id          0.0
rideable_type    0.0
started_at       0.0
ended_at         0.0
member_casual    0.0
dtype: float64
```

Now, the dataframe is clean and it doesn't have any null values. Hence, we can proceed with the further analysis.

Analyse

We have the three types of bikes available. Casual, electric and docked bikes. Let's calculate the percentage distribution among the two groups using PostgreSQL and for further analysis. The data is exported in CSV file format then it is imported into pgadmin and loaded into the table cyclistic_data.

- Members type

```
-- PERCENTAGE OF MEMBERS TYPE
WITH CTE AS (
        SELECT MEMBER_CASUAL, COUNT(RIDE_ID) AS NUMBER_OF_RIDE
        FROM CYCLISTIC_DATA
        GROUP BY 1
        ORDER BY 2 DESC)
SELECT
    MEMBER_CASUAL, NUMBER_OF_RIDE,
    ROUND((NUMBER_OF_RIDE :: NUMERIC/(SELECT COUNT(RIDE_ID) FROM CYCLISTIC_DATA))*100,2) PERCENT_VALUE
FROM CTE
ORDER BY 2 DESC, 3;
```

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

member_casual character varying (20)	number_of_ride bigint	percent_value numeric
member	3660698	64.00
casual	2059179	36.00

- Most used bike by the riders

```
-- MOST USED BIKE
WITH CTE AS (
    SELECT RIDEABLE_TYPE, COUNT(RIDE_ID) AS NUMBER_OF_RIDE
    FROM CYCLISTIC_DATA
    GROUP BY 1
    ORDER BY 2 DESC)
SELECT
    RIDEABLE_TYPE, NUMBER_OF_RIDE,
    ROUND((NUMBER_OF_RIDE :: NUMERIC/(SELECT COUNT(RIDE_ID) FROM CYCLISTIC_DATA))*100,2) PERCENT_VALUE
FROM CTE
ORDER BY 2 DESC, 3;
```

rideable_type character varying (30)	number_of_ride bigint	percent_value numeric
electric_bike	2945579	51.50
classic_bike	2696011	47.13
docked_bike	78287	1.37

```
WITH CTE AS (
    SELECT RIDEABLE_TYPE, MEMBER_CASUAL, COUNT(RIDE_ID) AS NUMBER_OF_RIDE
    FROM CYCLISTIC_DATA
    GROUP BY 1, 2
    ORDER BY 3 DESC)
SELECT
    RIDEABLE_TYPE, MEMBER_CASUAL, NUMBER_OF_RIDE,
    ROUND((NUMBER_OF_RIDE :: NUMERIC/(SELECT COUNT(RIDE_ID) FROM CYCLISTIC_DATA))*100,2) PERCENT_VALUE
FROM CTE
ORDER BY 3 DESC, 4;
```

rideable_type character varying (30)	member_casual character varying (20)	number_of_ride bigint	percent_value numeric
electric_bike	member	1841568	32.20
classic_bike	member	1819130	31.80
electric_bike	casual	1104011	19.30
classic_bike	casual	876881	15.33
docked_bike	casual	78287	1.37

Most riders uses electric bike over other two but the classic bike usage is quite closer to the electric bike. The docked bike may used occasionally based on the user needs.

- Number of rides per Day

```
-- RIDES PER WEEKDAY
SELECT STARTED_DAY, COUNT(RIDE_ID) AS NUMBER_OF_RIDES
FROM CYCLISTIC_DATA
GROUP BY 1
ORDER BY 2 DESC;
```

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

started_day character	number_of_rides bigint
Saturday	883566
Thursday	860202
Friday	843524
Wednesday	835625
Tuesday	822978
Sunday	744578
Monday	729404

The data shown that the majority of the rides has been happened only during the weekends and with the minimum differences with the other weekdays which are closer to the weekends.

- Number of rides per month and per season

```
-- RIDES PER MONTH
SELECT STARTED_MONTH, COUNT(RIDE_ID) AS NUMBER_OF_RIDES
FROM CYCLISTIC_DATA
GROUP BY 1
ORDER BY 2 DESC;
```

started_month character	number_of_rides bigint
August	771693
July	767650
June	719618
September	666371
May	604827
October	537113
April	426590
November	362518
March	258678
December	224073
February	190445
January	190301

During the month of June, July and August the usage of bikes and rides have increased. This change may cause based upon the seasonal changes too. As we observed by the below code, during the season of summer, more rides have occurred followed by autumn and less in the winter season.

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

```sql
-- RIDES PER SEASON
WITH CTE AS (
    SELECT
        CASE
            WHEN STARTED_MONTH IN ('December','January','February') THEN 'WINTER'
            WHEN STARTED_MONTH IN ('March','April','May') THEN 'SPRING'
            WHEN STARTED_MONTH IN ('June','July','August') THEN 'SUMMER'
            ELSE 'AUTUMN'
        END AS SEASON,
        COUNT(RIDE_ID) AS NUMBER_OF_RIDE
    FROM CYCLISTIC_DATA
    GROUP BY 1
)
SELECT
    SEASON,
    NUMBER_OF_RIDE,
    ROUND((NUMBER_OF_RIDE::float / total_ride_count * 100):: NUMERIC, 2) AS PERCENT_VALUE
FROM
    CTE,
    (SELECT COUNT(RIDE_ID) AS total_ride_count FROM CYCLISTIC_DATA) AS total_rides
ORDER BY
    2 DESC, 3;
```

season text	number_of_ride bigint	percent_value numeric
SUMMER	2258961	39.49
AUTUMN	1566002	27.38
SPRING	1290095	22.55
WINTER	604819	10.57

- Average Travel Duration

```sql
-- AVERAGE TRAVEL DURATION
SELECT member_casual, AVG(TRAVEL_DURATION)
FROM CYCLISTIC_DATA
GROUP BY 1;
```

member_casual character varying (20)	avg interval
casual	00:21:16.572445
member	00:12:07.282299

```sql
-- AVERAGE TRAVEL DURATION
SELECT RIDEABLE_TYPE, member_casual, AVG(TRAVEL_DURATION)
FROM CYCLISTIC_DATA
GROUP BY 1,2
order by 3 desc;
```

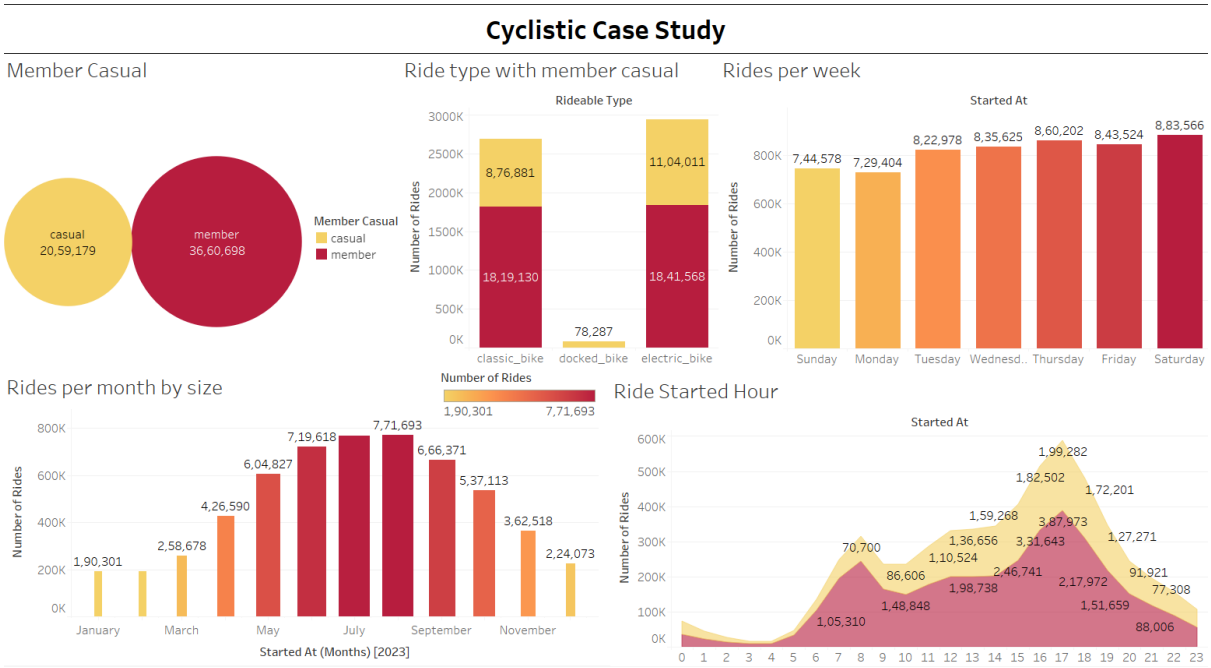
rideable_type character varying (30)	member_casual character varying (20)	avg interval
docked_bike	casual	01:06:51.53553
classic_bike	casual	00:25:51.130523
electric_bike	casual	00:14:24.559506
classic_bike	member	00:13:01.934714
electric_bike	member	00:11:13.295779

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

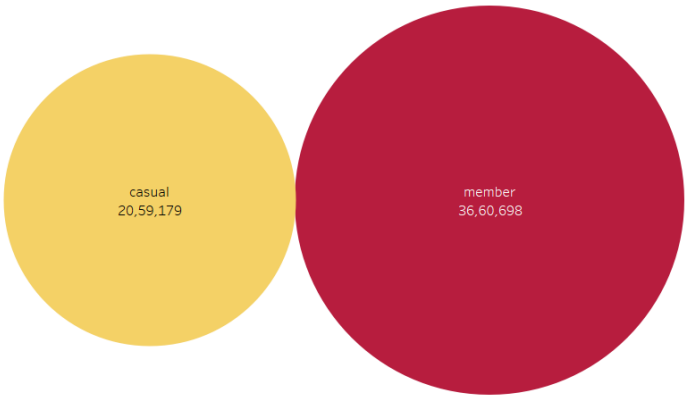
Since there is no distance provided in the dataset, we are trying to measure it with the time duration travelled by the rider. The above calculation shows that casual riders have ridden more than the member riders. The docked bikes are having minimum number of usage and but it is being used by the riders for the long travel.

Share

For visualization, Tableau is used. By visualizing, we can get the clear idea of data and the trends over the period of time. Below is the dashboard of the case study and let's break it down one by one for better understanding.



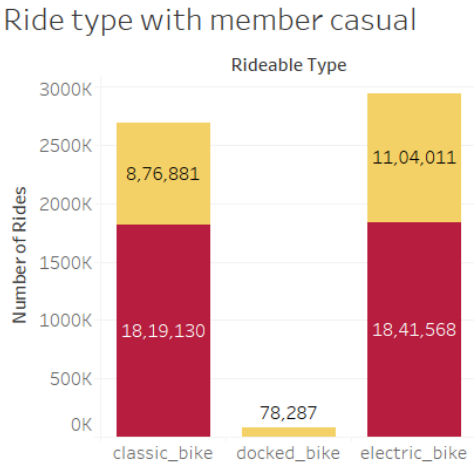
- User type



We have two types of users member and casual. More ride was made by Members when compared to casual riders.

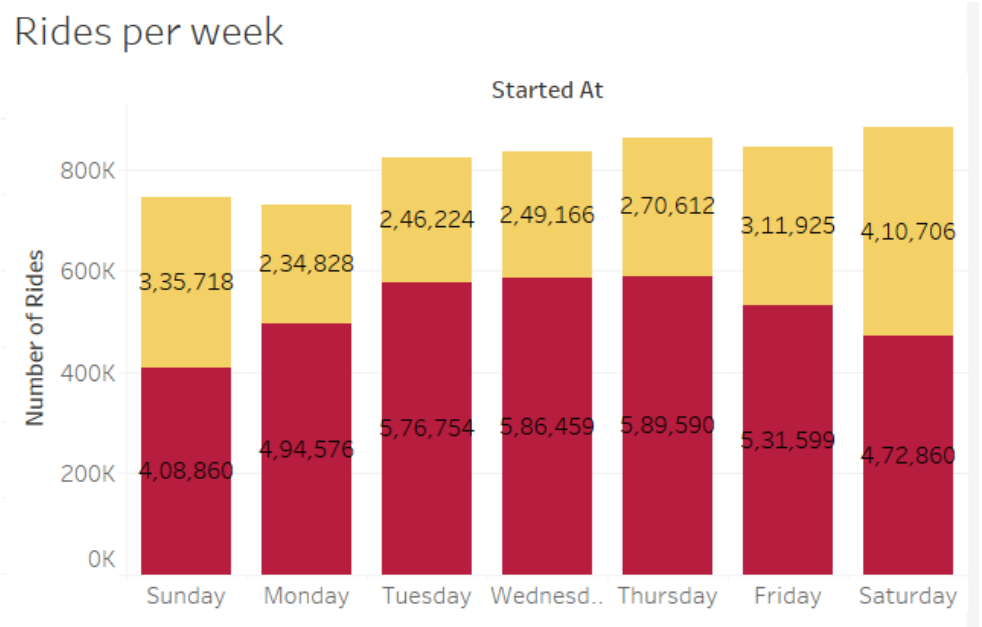
Case Study: How Does a Bike-Share Navigate Speedy Success?

- Rider type and bike type



Both casual riders and member riders have used the classic and electric bike equally. The docked bike may be used occasionally for specific purposes. But more than classic bike, electric bike is being used by both members and casuals in the large scale.

- Rides per week

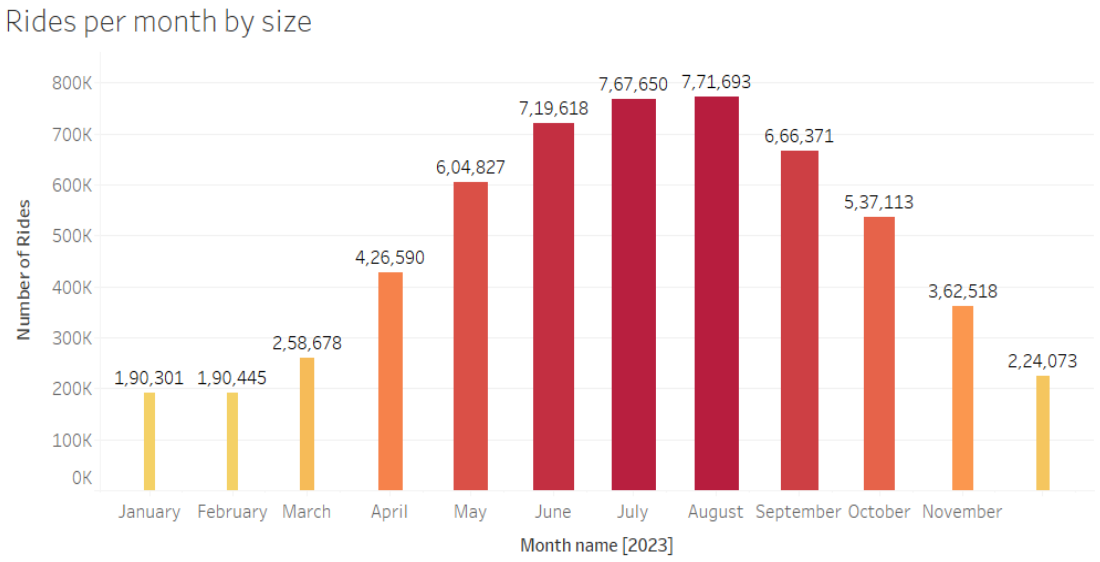


We can clearly observe that demand increases during the weekends. Both casual and member have taken rides during the weekends when compared to weekdays.

- Rides per Month

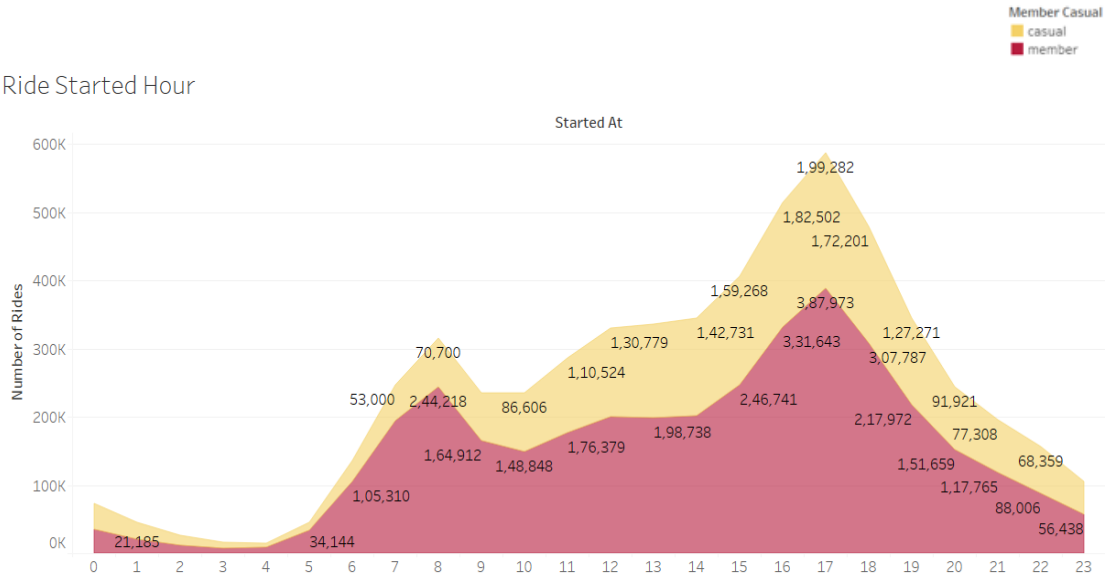
As we discussed in the seasonal rides in the query above, the riders are using the bikes often in the specific season. As we can see below, huge rides have happened during the month of July and August followed by subsequent months June and September.

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



- Ride started Hour

  For both member and casual riders, we can see the similar pattern on the riders that they have started the rides on the same hours in the whole 24hrs. We can see the huge demand and peak during 5PM starting from the 2PM. This shows that majority of the riders may be the working professionals.



Act

- The company needs to begin offering special plans for riders during different times of the year. For example, they could have deals just for summer or winter. This would make riders happy and attract more customers.
- The company should offer coupon codes or ride deals based on hours. We noticed that people often use bikes at certain times, so it's a good idea to give discounts during busy times. This might get more people who don't usually ride bikes to give it a try.
- The company should offer discounts every week because we can observe that lots of people use bikes on weekends. So, it's smart to give discounts during that time.