Cyclistic bike-share case study

In this case study, I analyze historical data from a Chicago based bike-share company in order to identify trends in how their customers use bikes differently. The main tools I used are Excel, Microsoft Sequel Server Management Studio, PowerBi.

A more in-depth breakdown of the case study scenario is included below, followed by my full report.

Scenario

Cyclistic is a bike-share company based in Chicago with two types of customers. Customers who purchase single-ride or full-day passes are known as **casual riders,** while those who purchase annual memberships are known as **members**.

The company has grown in recent years to own about 5,824 bicycles that are geo-tracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any station in the system anytime.

The director of marketing believes the company's future success depends on maximizing the number of annual memberships, and has tasked us with discovering insights into how to convert casual riders into Cyclistic Members.

Business task

Analyze historical data to identify trends in how casual riders and annual members use the service differently, and recommendations on how to convert casual riders into annual members.

Data Source

We will be using Cyclistic's historical bike-trip data from the last 12 months, which can be found with the link http://divvy-tripdata.s3.amazonaws.com/index.html. Before the use of this data it was checked for integrity and bias and the license for this data was provided under this link http://www.divvybikes.com/data-license-agreement. The data is stored in spreadsheets. There are 12 csv files. The data for this fictional company has been made available for use by Motivate International Inc.

Variables

The data includes 13 different variables listed below, each with its descriptions:

- ride_id : Unique ID
- rideable_type : bike type (Classic, Docked, Electric)
- started_at : trip start (day and time)
- ended_at : trip ends (day and time)
- start_station_name : trip start station
- start_station_id : trip start station ID
- end_station_name : trip end station
- end_station_id : trip end station ID
- start_lat : trip start latitude
- start_lng : trip start longitude
- end_lat: trip end latitude
- end_lng : trip end longitude
- member_casual: rider type (Casual or Cyclistic Member)

Data Limitations

As riders personally Identifiable information is prohibited, no pass purchases will be able to connect to credit card numbers to determine if casual riders live in the Cyclistic service area, or if they have purchased multiple single-ride passes.

Data Cleaning and Processing

I familiarize myself with the data on Excel. By sorting and filtering the data in each file, I explored their formatting and noted any missing or duplicate data points.

I imported all my data into Microsoft SQL Server Management Studio to continue with the cleaning process. I created a Database named "Cyclistic_21" and imported the data for each month into the database. Once all the data was successfully imported, I combined all the data into one table under "Cyclistic_year" using the UNION ALL command and replaced all blank cells with NULL to make it easily identifiable and to clean NULL values later on.

The following sql query was ran to merge all data into one.

```
USE cyclistic_21;

SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,

end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual

INTO cyclistic_year

FROM

(

SELECT  ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,

end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual

FROM jan_2021

UNION ALL

SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,

end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
```

```
FROM feb_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM mar_2012
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM apl_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM may_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM jun_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM jul_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM aug_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM sept_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM oct_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM nov_2021
UNION ALL
SELECT ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
FROM dec_2021
) cyclisticyear
```

To begin the analysis of the process, the dataset was checked and was confirmed to have about 3 million rows of data. A back up of this table was created for safety reasons.

The data cleaning process began with checking for significant numbers of NULL values in the relevant columns for our analysis.

Started_at and ended_at (Time) and member_casual columns did not have NULL values which is good news since they are important values for this analysis. There were 4000+ NULL values for latitude and longitude columns.

```
SELECT COUNT (*)
FROM cyclistic_year
WHERE started_at  = NULL
```

```
OR ended_at  = NULL

--- IN TOTAL 0

SELECT COUNT (*)

FROM cyclistic_year

WHERE start_lat  IS NULL OR start_lng  IS NULL

OR end_lat  IS NULL OR end_lng IS NULL

--- IN TOTAL 4771


SELECT COUNT (*)

FROM cyclistic_year

WHERE member_casual IS NULL

--- IN TOTAL 0
```

To continue the cleaning of the data, I checked for duplicate ride id, as that was meant to be unique. The duplicate ride id were deleted once found.

```
--- CHECKING DUPLICATE RIDE_ID

COUNT (ride_id)

FROM cyclistic_year

GROUP BY ride_id

HAVING COUNT (ride_id)    > 1


---DELETING DUPLICATE RIDE_ID FOUND

WITH cte AS

(

SELECT  *, ROW_NUMBER() OVER (PARTITION BY ride_id

ORDER BY ride_id )  AS r_id
FROM cyclistic_year
)
DELETE FROM cte WHERE r_id  <> 1
```

The cleaning process continues by creating a new table with columns needed for the analysis and an index was created. This will result will enable the queries load faster.

```
--- IMPORT NEW COLUMNS FROM PREVIOUS TABLE TO NEW TABLE CALLED FINAL_YEAR
SELECT  ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
INTO final_year
FROM cyclistic_year

--- CREATED AN INDEX TO MAKE THE QUERIES LOAD FASTER
CREATE INDEX table_index
ON final_year
(
ride_id, rideable_type, started_at, ended_at, start_station_name, start_station_id,
end_station_name, end_station_id, start_lat, start_lng, end_lat, end_lng, member_casual
)
```

New columns will be needed for the continuation of the cleaning process. Also the new columns will be populated with data that is necessary for the descriptive analysis of the data. The following sql queries was used to add columns to the table.

```
--- CREATED NEW COLUMNS

ALTER TABLE final_year
ADD duration INT ,
[year] INT ,
[date] DATE ,
day_of_the_week   NCHAR (10) ,
[month] NCHAR (20)

--- FILLED IN VALUES INTO THE NEW COLUMNS
UPDATE final_year
SET duration  =  DATEDIFF  ( minute  , started_at  , ended_at  ) ,
[year]  =  DATEPART  ( year  , started_at  ) ,
[date]  =   CAST ( started_at  AS date ) ,
Day_of_the_week  = DATENAME  ( WEEKDAY  ,  started_at  ) ,
[month]  = DATENAME  ( MONTH , started_at  )
```

The next query was ran to delete cells that contained wrong or unwanted data so the table can be ready for analysis

```
---DELETED ROWS WHERE THE START_STATION_NAME , START_STATION_ID,
END_STATION_NAME, END_STATION_ID  ARE EMPTY
DELETE
FROM final_year
WHERE start_station_name  IS NULL AND start_station_id  IS NULL
AND end_station_name IS NULL AND end_station_id  IS NULL

--- DELETED ROWS CONTAINING NULL OR NEGATIVE VALUES IN THE DURATION COLUMN
```

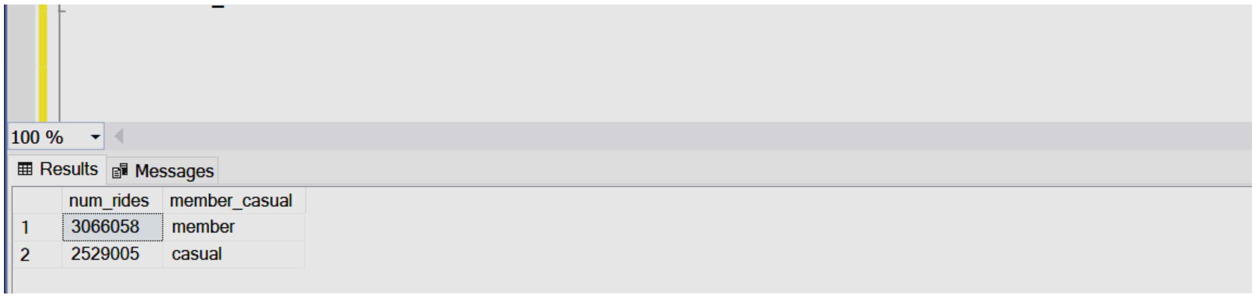
```
DELETE
FROM final_year
WHERE duration IS NULL
OR duration  <= 0
```

Analysis

Now that I have some clean data, I will be organizing and performing calculations on it to Identify key trends and relationships. As the main aim of the analysis is to find the differences between casual and annual riders, most of the analysis was done with filtering between the two parameters.

To begin the analysis, I will find the total number of members and casual riders.

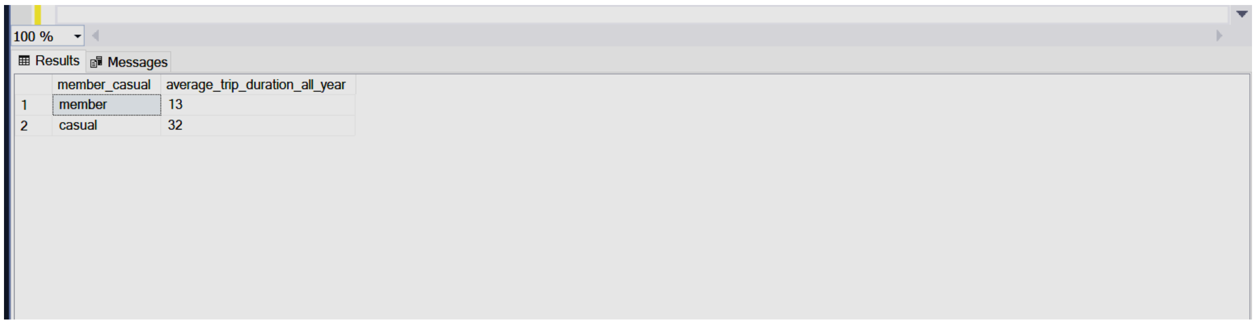
```
SELECT COUNT (*)
 AS num_rides ,  member_casual
FROM final_year
GROUP BY member_casual
ORDER BY num_rides DESC
```



	num_rides	member_casual
1	3066058	member
2	2529005	casual

I then tried to understand how casual riders and annual members use cyclistic bikes differently. I started by calculating the average ride duration for both annual members and casual riders to see how it differs.

```
SELECT member_casual  , CASE

WHEN member_casual  = ‘member’  THEN  ( SELECT AVG (duration))

WHEN member_casual  = ‘casual’   THEN ( SELECT AVG (duration))

END AS avg_duration_final_year

FROM final_year

GROUP BY member_casual
```

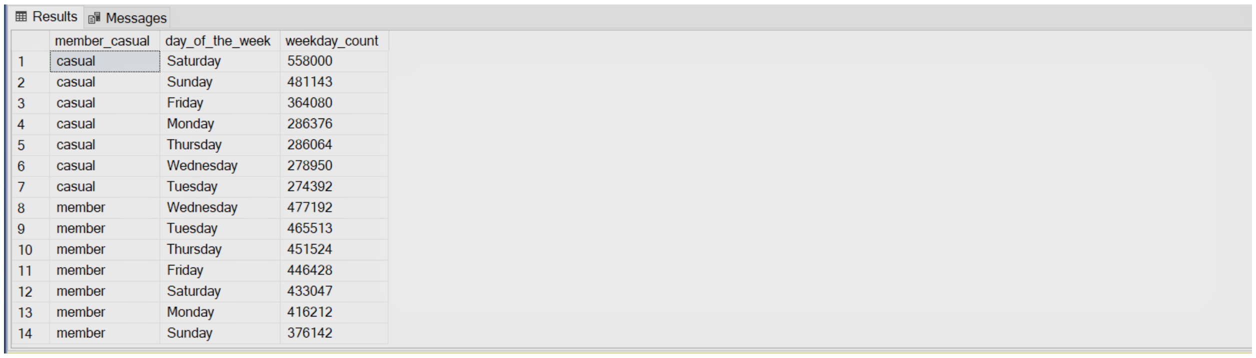


```
100 %
Results   Messages

   member_casual  average_trip_duration_all_year
1  member         13
2  casual         32
```

The average trip duration for casual riders was significantly higher than for annual members.

Then I wanted to find the day of the week that had the highest volume of trips for both casual and annual members, and see if there is a trend.

```
SELECT member_casual , day_of_the_week , CASE

WHEN member_casual = ‘member’ THEN COUNT (*)

WHEN member_casual = ‘casual’ THEN COUNT (*)

END AS weekday_count

FROM final_year

GROUP BY member_casual , day_of_the_week

ORDER BY member_casual , weekday_count DESC
```



	member_casual	day_of_the_week	weekday_count
1	casual	Saturday	558000
2	casual	Sunday	481143
3	casual	Friday	364080
4	casual	Monday	286376
5	casual	Thursday	286064
6	casual	Wednesday	278950
7	casual	Tuesday	274392
8	member	Wednesday	477192
9	member	Tuesday	465513
10	member	Thursday	451524
11	member	Friday	446428
12	member	Saturday	433047
13	member	Monday	416212
14	member	Sunday	376142

For casual rides the highest volume was on Saturday and then on Sunday. The volume for casual riders was at its highest on the weekend and dropped during weekdays. For members, the highest volume happened on Wednesday followed by Thursday and then Friday and the volume was at its lowest on Sunday.

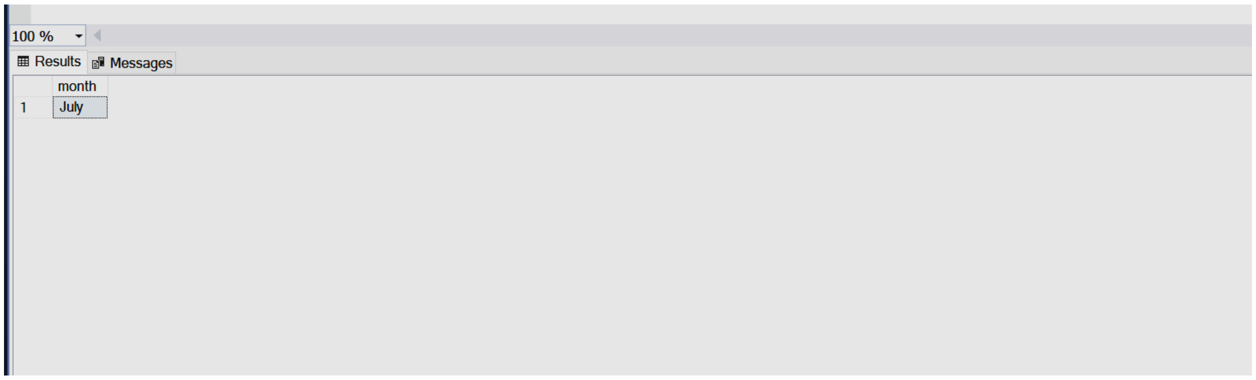
The next query was to find the most common month where the bikes are ridden the most.

SELECT top 1 [month] AS month

FROM final_year

GROUP BY month

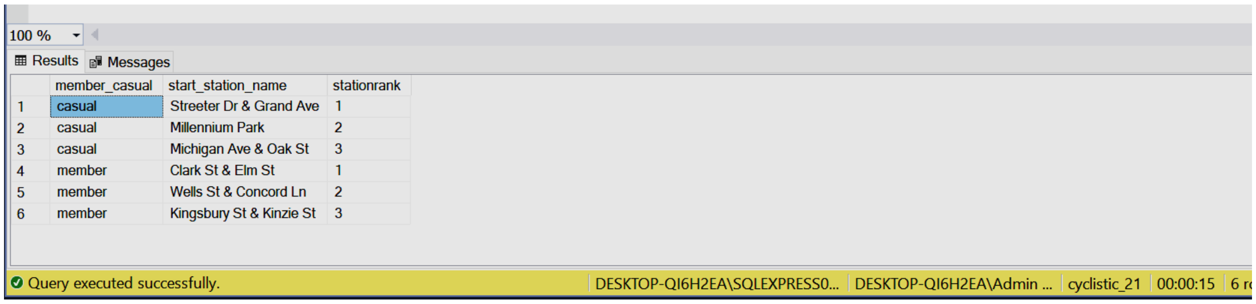
ORDER BY COUNT (*) DESC



	month
1	July

The next query was done to check for the most common stations each type of user uses the most.

```
WITH Stationresult AS

(

SELECT member_casual , start_station_name,

CASE WHEN member_casual = 'casual' THEN

DENSE_RANK() OVER (PARTITION BY member_casual ORDER BY COUNT (start_station_name) DESC)

WHEN member_casual = 'member' THEN

DENSE_RANK() OVER (PARTITION BY member_casual ORDER BY COUNT (start_station_name )DESC)

END AS stationrank

FROM final_year

WHERE start_station_name IS NOT NULL

GROUP BY start_station_name , member_casual


)

SELECT *

FROM Stationresult

WHERE stationrank <=3
```



	member_casual	start_station_name	stationrank
1	casual	Streeter Dr & Grand Ave	1
2	casual	Millennium Park	2
3	casual	Michigan Ave & Oak St	3
4	member	Clark St & Elm St	1
5	member	Wells St & Concord Ln	2
6	member	Kingsbury St & Kinzie St	3

```
Query executed successfully.    DESKTOP-QI6H2EA\SQLEXPRESS0...    DESKTOP-QI6H2EA\Admin ...    cyclistic_21    00:00:15    6 r
```

Top three start_stations for casual riders were:

1- Streeter Dr & Grand Ave

2- Millennium Park

3- Michigan Ave & Oak St

For members:

1- Clark St & Elm St

2- Wells St & Concord Ln

3- Kingsbury St & Kinzie St

The same query was applied for end_station:

For casual riders:

1- Streeter Dr & Grand Ave

2- Millennium Park

3- Michigan Ave & Oak St

For members:

1- Clark St & Elm St

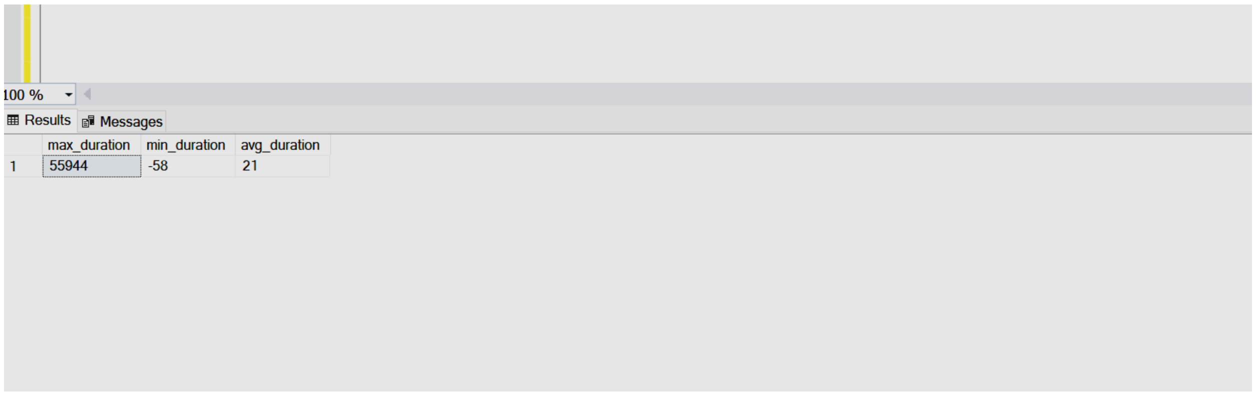
2- Wells St & Concord Ln

3- Kingsbury St & Kinzie St

From the previous result we can see that Streeter Dr & Grand Ave , Millenium Park and Michigan Ave & Oak St were the top three stations with the highest traffic for casual riders to pick up and drop off the bikes.

Another important analysis was done to find the max, min and average duration for the total rides.

```
SELECT MAX (duration ) AS max_duration,
            MIN (duration) AS  min_duration,
            AVG (duration) AS avg_duration
FROM final_year
```



```
100 %
Results  Messages
   max_duration  min_duration  avg_duration
1  55944         -58           21
```

From the results of the query above, it was found out that the max duration time is 55944 minutes, the minimum time is -58 minutes and the average time is 21 minutes.

One last thing I looked at before moving to the visualizations was the rideable_type column. As mentioned before, there are three types of bikes: docked_bike, electric_bike, and classic_bike.

I wrote a query to check for the percentage of rides that were done with each rideable_type from the total rides.

```
WITH cte AS
(
SELECT CAST (COUNT (ride_id) AS FLOAT ) AS total_no
FROM final_year
)
SELECT rideable_type  , CASE
```

WHEN rideable_type = ‘electric_bike ‘ THEN

ROUND (CAST ((COUNT (*) / total_no) * 100 AS NUMERIC) , 2)

WHEN rideable_type = ‘docked_bike’ THEN

ROUND (CAST ((COUNT (*) / total_no) * 100 AS NUMERIC) , 2)

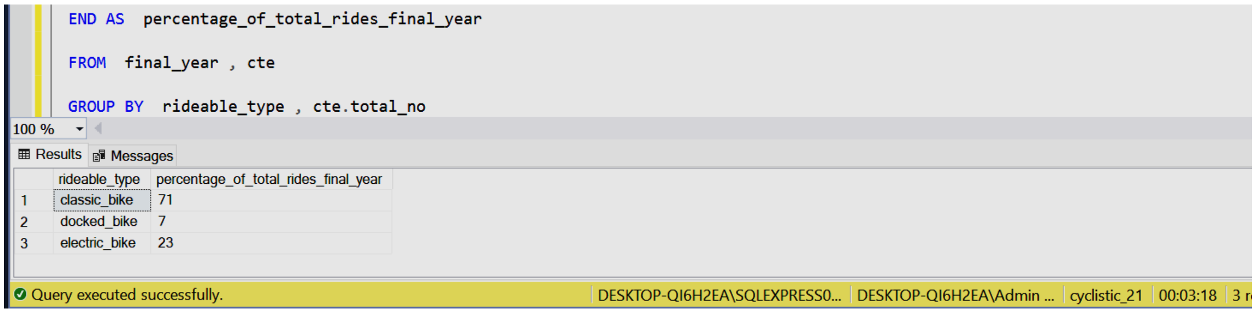
WHEN rideable_type = ‘classic_bike’ THEN

ROUND (CAST ((COUNT(*) / total_no) * 100 AS NUMBER) , 2)

END AS percentage_of_total_rides_final_year

FROM final_year , cte

GROUP BY rideable_type , cte.total_no



```
END AS  percentage_of_total_rides_final_year

FROM  final_year , cte

GROUP BY  rideable_type , cte.total_no
```

	rideable_type	percentage_of_total_rides_final_year
1	classic_bike	71
2	docked_bike	7
3	electric_bike	23

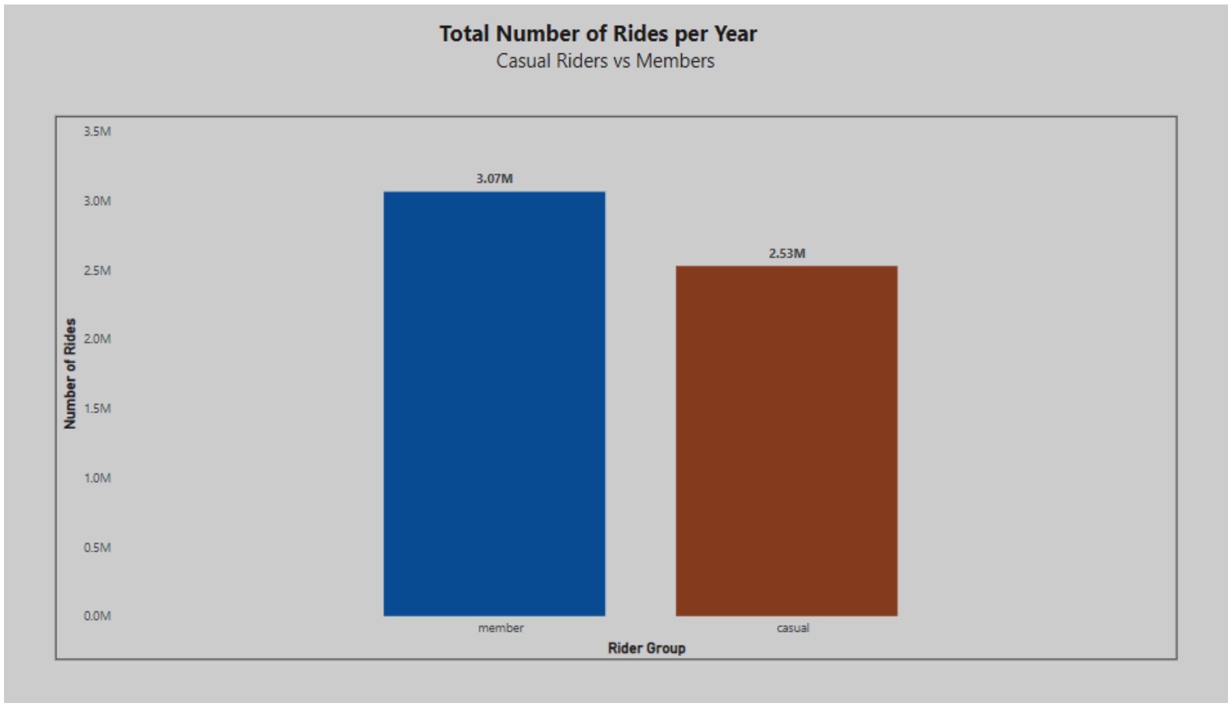
```
Query executed successfully.    DESKTOP-QI6H2EA\SQLEXPRESS0...  DESKTOP-QI6H2EA\Admin ...  cyclistic_21  00:03:18  3 r
```

The majority of the rides were done using classic_bike, but we can also see that electric_bike took a good part of the total rides.

Supporting Visualizations

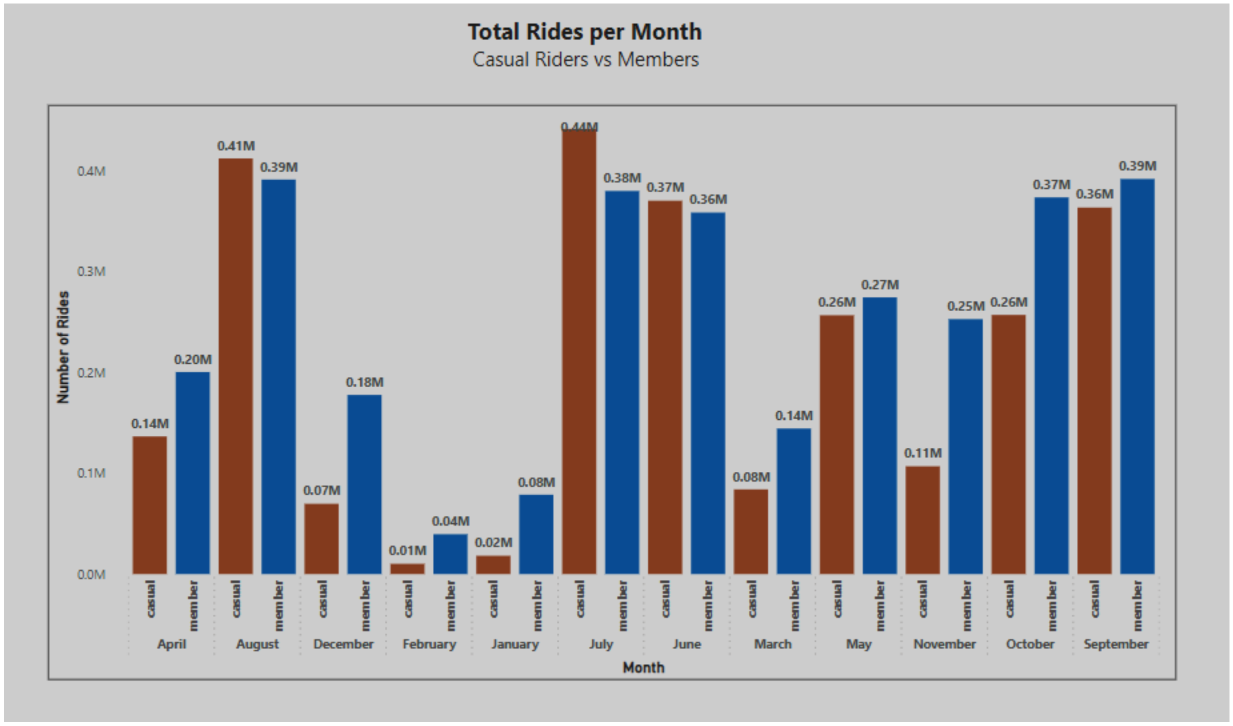
I used PowerBi to run some further analysis and generate visualizations that support the key findings in the analysis.

We will begin with a simple comparison of casual rides and member rides.



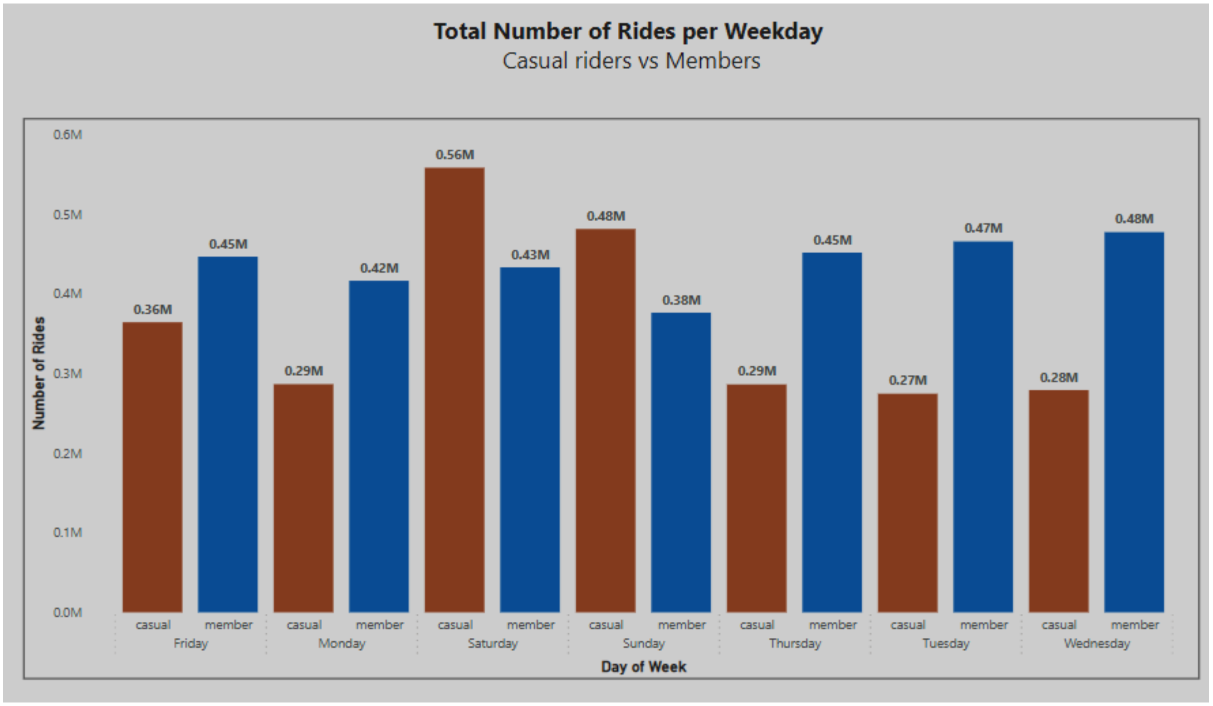
A simple comparison of total rides between the two rider group shows that the majority of cyclistic's customers are members.

The next step in our analysis is to look at the rides per month.



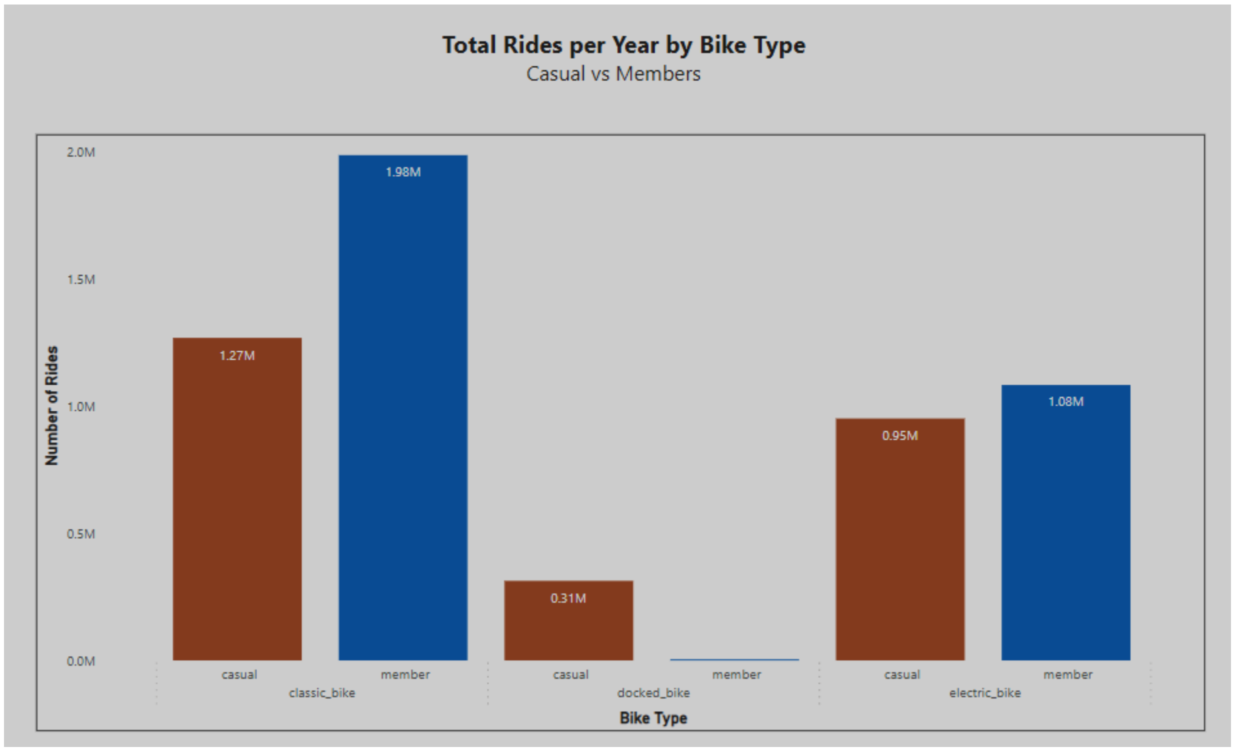
Here we can see and gather a better understanding of our customer's riding habit. Cyclistic members recorded their highest activity in august and the lowest in February. Casual riders recorded their highest activity in July and the lowest in February. From May to August cyclistic members recorded a higher user activity compared to casual riders who had a lesser activity.

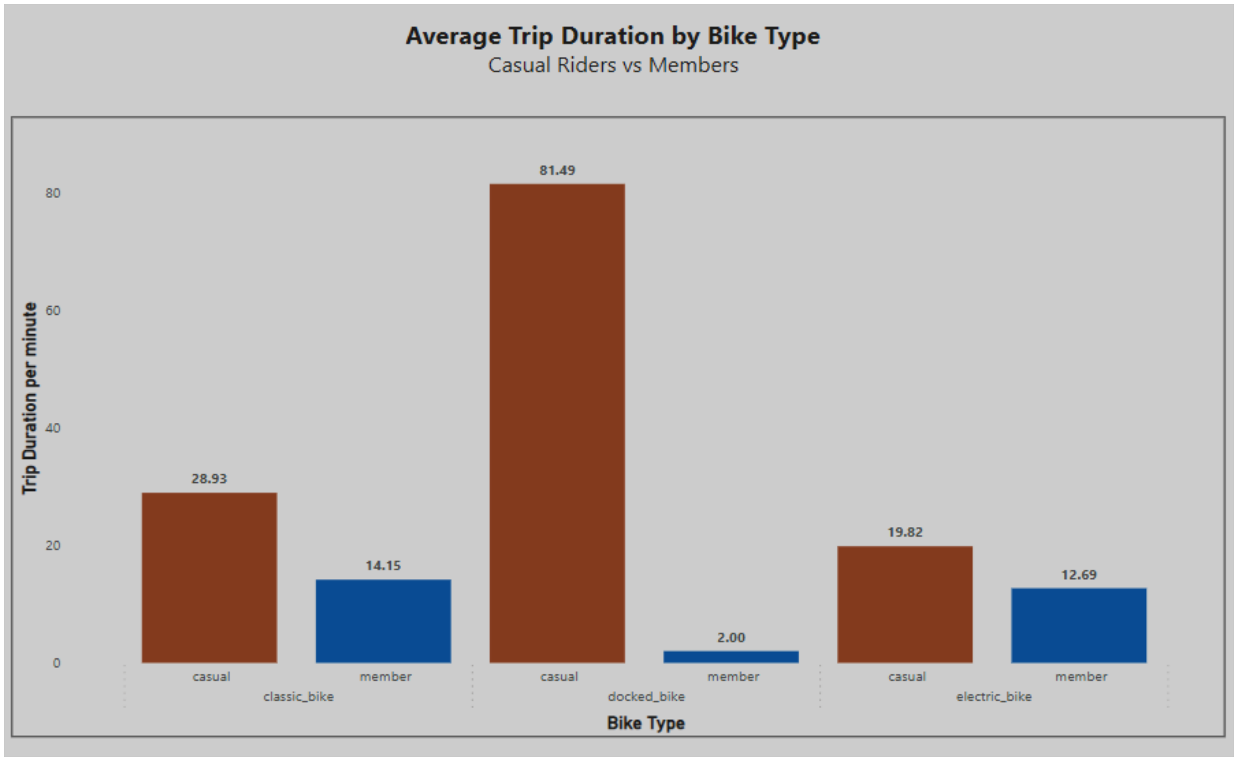
To take this analysis a step further, we will compare the ride patterns on a daily level.



From here we can see that members ride relatively consistently throughout the week, with a slight dip on weekends. Casual riders have a significant increase in bike use during the weekends. These trends indicate that members are using the bikes for consistent purposes (commute, work), while casual riders use the bike for more leisurely purposes.

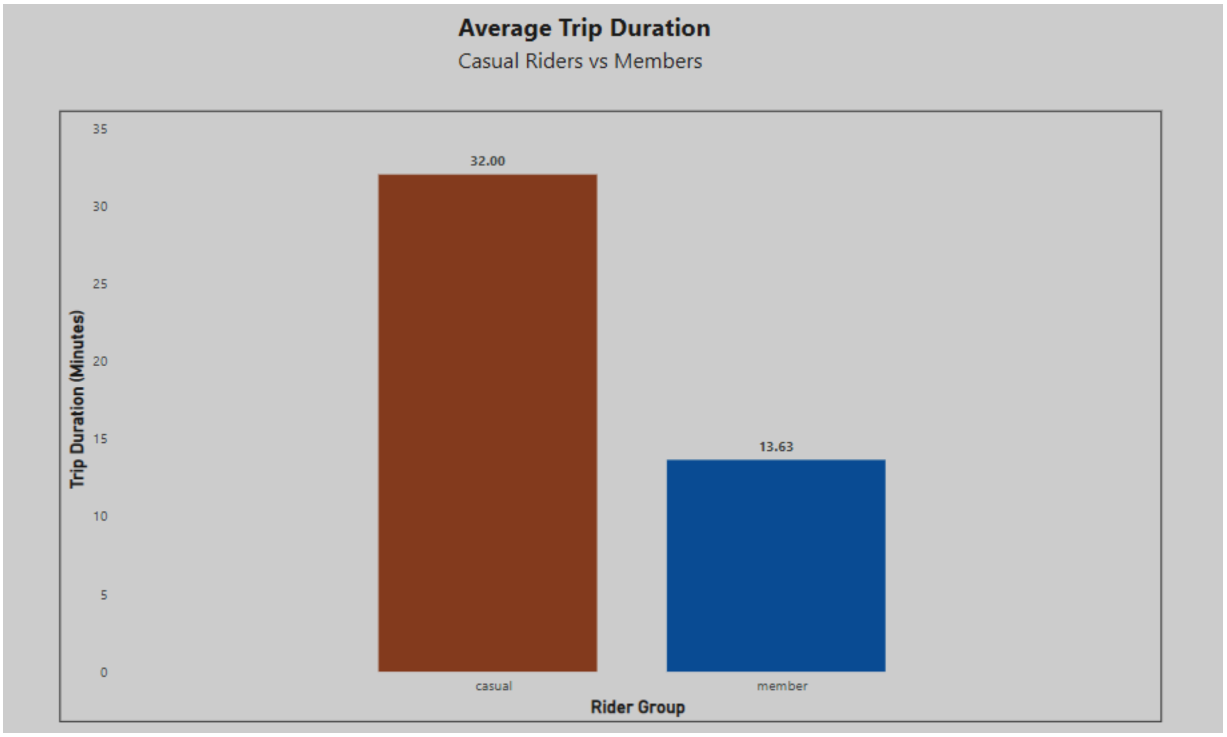
Another metric to compare is the type of bike preferred by the two rider groups.


Total Rides per Year by Bike Type
Casual vs Members
Number of Rides
2.0M
1.5M
1.0M
0.5M
0.0M
1.27M
1.98M
0.31M
0.95M
1.08M
casual
member
classic_bike
casual
member
docked_bike
casual
member
electric_bike
Bike Type


Average Trip Duration by Bike Type
Casual Riders vs Members
Trip Duration per minute
80
60
40
20
0
28.93
14.15
81.49
2.00
19.82
12.69
casual
member
classic_bike
casual
member
docked_bike
casual
member
electric_bike
Bike Type

Casual riders and members both appear to have a very significant preference towards classic bikes over electric, with casual riders choosing them at nearly 3:1 ratio and members at 4:1. One interesting note is the huge difference in preference of docked bikes by casual riders over members. Ride duration does not appear to be significantly impacted by the type of bike, with classic bike being ridden for slightly longer durations on average.

The final metric to compare is the ride duration. The plot below shows that on average, casual riders take rides over twice the duration of the annual members. This trend can be seen at a monthly and daily level, indicating that members are taking more consistent, direct rides to and from work compared to the casual riders.



Conclusion / Recommendations

Based on our analysis, here are some recommendations below:

- As observed, casual riders tend to ride more on weekends, so special subscription offers can be given to casual riders for trying out membership for a month and then discounts can be given if they convert to annual members.
- Saturday and Sunday should be prioritized when it comes to scheduling ads for the digital online campaign.
- Having existing members share their stories about how Cyclistic's bike-share system has facilitated their day-to-day routine. It will help encourage new riders to join the program. Moreover, gathering user feedback on the purpose of their trip would offer insight into any hesitancy in committing to a membership.
- Providing an indoor ride experience, using stationary bikes would allow users who use Cyclistic for exercise still gain value from the service during the months where temperatures are low.
- Offer priority access during busy hours and discounted pricing during non-busy hours for Cyclistic members only. This would encourage casual riders to apply for an annual membership.
- Put up banners or discount advertisements at prominent locations targeting casual riders, which might influence them to become members. It can be a print media on the bike, a booth or social media post.

Thank you for taking the time to read this and I'm looking forward to any questions/suggestions/comments you may have.