Google Data Analytics Capstone Project: How Does a Bike-Share Navigate Speedy Success?

I. Ask Phase

Identifying business tasks:

These are three questions that need to be answered:

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?

How to approach?

Finding out the trends and correlation between variables in the data with the help of visualizations and statistical analysis then drawing a conclusion of recommendations and suggestions for Cyclistic company to attract more annual memberships.

II. Prepare Phase

The data was provided by Cyclistic, for the business objective, it should be over the past 12 months from January 2021 to December 2021.

On Cyclistic server, data was separated by months and years and saved as .csv files. I stored all needed data then merged them into one file only called **bikeshare.csv.**

The data included following fields:

ride_id: unique ID per ride

rideable_type: type of bicycle customer used

started_at: the time bike was checked out

ended_at: the time bike was checked in

start_station_name: name of departure station

start_station_id: unique id of start station

end_station_name: name of station at the end of the ride

end_station_id: unique id of end station

start_lat: latitude of start station

start_lng: longitude of start station

end_lat: latitude of end station

end_lng: longitude of end station

member_casual: type of customer taking the ride, member or casual

In the analysis, I added 3 fields to get more insights from data:

ride_length: the total time of the trip caculated as ended_at - started_at

week_day: day of week the ride started

month: month of year the ride started

III. Process Phase

I used **Python** for the whole project.

To begin with, I imported some main libraries required for the analysis.

```
# Importing libararies
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
```

For the separated data, I already merged them before into one file named **bikeshare.csv.**

Check the data types:

```
case_1=df
case_1.info()
```
✓ 0.8s

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5823559 entries, 0 to 5823558
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: 577.6+ MB
```

Then I converted time variables into the right type then have a peek at the data:

```
#convert to datetime
case_1['started_at']= pd.to_datetime(case_1['started_at'])
case_1['ended_at']= pd.to_datetime(case_1['ended_at'])
case_1.head()
```
✓ 8.2s Python

	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_ln
0	A5AF35CB4C40C386	electric_bike	2021-08-06 16:43:01	2021-08-06 16:51:21	Bernard St & Elston Ave	18016	Kildare Ave & Montrose Ave	KA1706005015	41.949953	-87.713994	41.961018	-87.73565
1	06A7BEB1B7E25E45	classic_bike	2021-08-20 22:15:26	2021-08-20 22:37:28	Glenwood Ave & Touhy Ave	525	Broadway & Ridge Ave	15578	42.012701	-87.666058	41.984045	-87.66027
2	D06866592603F067	classic_bike	2021-08-17 19:30:18	2021-08-17 19:43:55	Glenwood Ave & Touhy Ave	525	Western Ave & Howard St	527	42.012701	-87.666058	42.018901	-87.69004
3	95B45E9C0877BD63	classic_bike	2021-08-08 12:25:38	2021-08-08 12:44:17	Glenwood Ave & Touhy Ave	525	Broadway & Berwyn Ave	13109	42.012701	-87.666058	41.978353	-87.65975
4	C36DA61E9C306925	classic_bike	2021-08-28 17:12:28	2021-08-28 17:29:19	Glenwood Ave & Touhy Ave	525	Broadway & Ridge Ave	15578	42.012701	-87.666058	41.984045	-87.66027

Adding **week_day, ride_length, month** columns:

```
#finding the weekday, ride_length, month
case_1['week_day'] = case_1['started_at'].apply(lambda x:x.weekday())
case_1['ride_length']= (case_1['ended_at'] - case_1['started_at']) / datetime.timedelta(minutes=1)
case_1['month'] = pd.DatetimeIndex(case_1['started_at']).month
```

Sorting data in an ascending order by start date:

```
#sort values by start time
case_1.sort_values(by=['started_at'], inplace = True, ascending=True)
```

Drop duplicates then take a quick look again at the data:

```
case_1.drop_duplicates()
case_1.head()
```

✓ 31.9s Python

	ride_id	rideable_type	started_at	ended_at	start_station_n
3100833	A3F8D895163BBB49	electric_bike	2021-01-01 00:02:05	2021-01-01 00:12:39	I
3317955	0D139A3203274B87	classic_bike	2021-01-01 00:02:24	2021-01-01 00:08:39	State St & 33r
1635171	C7AE8E9CDB197A8E	classic_bike	2021-01-01 00:06:55	2021-01-01 00:26:36	Lakeview Av Fullerton P
1377821	2633EB2B8A99F5CB	electric_bike	2021-01-01 00:12:13	2021-01-01 00:20:06	Kedzie Av Milwaukee
1377691	3097EF26414C7016	classic_bike	2021-01-01 00:12:21	2021-01-01 00:12:33	Montrose Ha

Note: I didn’t drop null values in this project to get a better overview at this dataset.

IV. Analysis Phase

Statistical analysis

I did some statistical analysis to get an overview of the data

```
#Statistical analysis
from statistics import mode
mode(case_1['week_day'])
```
✓ 1.3s

```
'Friday'
```

```
case_1['ride_length'].head()
```
✓ 0.1s

```
3100833    10.566667
3317955     6.250000
1635171    19.683333
1377821     7.883333
1377691     0.200000
Name: ride_length, dtype: float64
```

```
mean_ride_length = case_1['ride_length'].mean()
mean_ride_length
```
✓ 0.1s

```
21.971850987800902
```

```
max_ride_length = case_1['ride_length'].max()
max_ride_length
```
✓ 0.1s

```
55944.15
```

```
mean_ride_length_member = case_1[case_1['member_casual']=='member']
mean_ride_length_member = mean_ride_length_member['ride_length'].mean()
mean_ride_length_member
```
✓ 1.7s Python

```
13.648009923207674
```

```
mean_ride_length_casual = case_1[case_1['member_casual']=='casual']
mean_ride_length_casual = mean_ride_length_casual['ride_length'].mean()
mean_ride_length_casual
```

✓ 1.4s Python

```
32.19897609602308
```

```
max_ride_length_member = case_1[case_1['member_casual']=='member']
max_ride_length_member = max_ride_length_member['ride_length'].max()
max_ride_length_member
```

✓ 1.6s Python

```
1559.9333333333334
```

```
max_ride_length_casual = case_1[case_1['member_casual']=='casual']
max_ride_length_casual = max_ride_length_casual['ride_length'].max()
max_ride_length_casual
```

✓ 1.7s Python

```
55944.15
```

```
mode_week_day_member = case_1[case_1['member_casual']=='member']
mode_week_day_member = mode_week_day_member['week_day'].mode()
mode_week_day_member
```

✓ 2.3s

```
0    Tuesday
Name: week_day, dtype: object
```

```
mode_week_day_casual = case_1[case_1['member_casual']=='casual']
mode_week_day_casual = mode_week_day_casual['week_day'].mode()
mode_week_day_casual
```

✓ 1.9s

```
0    Friday
Name: week_day, dtype: object
```

```
count_users_week_day = case_1.groupby('week_day')['ride_id'].count()
count_users_week_day = count_users_week_day.sort_index()
count_users_week_day
```

✓ 1.9s

```
week_day
Friday       1036299
Monday        773905
Saturday      892912
Sunday        737413
Thursday      836165
Tuesday       787811
Wednesday     759054
Name: ride_id, dtype: int64
```

```
count_member_week_day = case_1[case_1['member_casual']=='member'].groupby('week_day')['ride_id'].count()
count_member_week_day
```

✓ 2.9s

```
week_day
Friday       456209
Monday       489050
Saturday     394427
Sunday       439047
Thursday     464308
Tuesday      499978
Wednesday    467502
Name: ride_id, dtype: int64
```

V. Share Phase

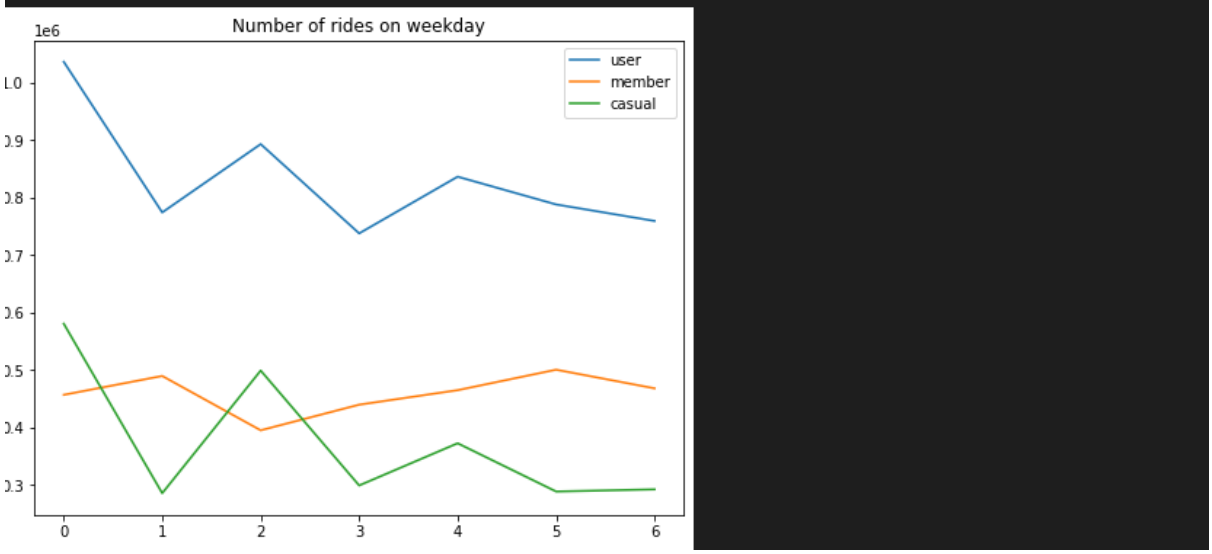
Visualization

Then I visualized data:

Number of ride:

```
plt.figure(figsize= (8,6))
plt.plot(case_1.groupby('week_day')['ride_id'].count().values)
plt.plot( case_1[case_1['member_casual']=='member'].groupby('week_day')['ride_id'].count().values)
plt.plot(case_1[case_1['member_casual']=='casual'].groupby('week_day')['ride_id'].count().values )
plt.title('Number of rides on weekday')
plt.legend(['user', 'member', 'casual'])
labels = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']
# plt.xticks(case_1[case_1['member_casual']=='casual'].groupby('week_day')['ride_id'].count().index, labe
plt.show()
```

✓ 11.9s



It can be seen that users usually rode at the weekend.

```
member_type = pd.pivot_table(case_1, index='member_casual', values='ride_id',aggfunc='count')
member_type
```

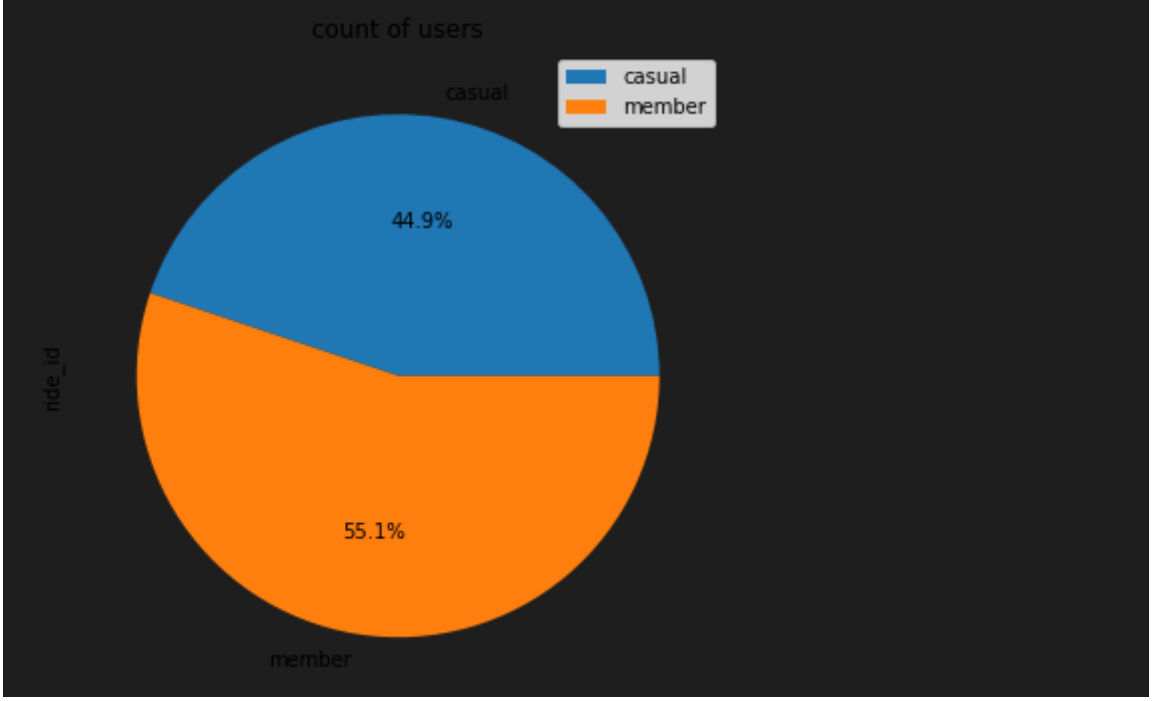
✓ 3.6s

	ride_id
member_casual	
casual	2613038
member	3210521

```
member_type.plot.pie(y='ride_id',autopct='%1.1f%%', figsize=(8,6))
plt.title('count of users')
plt.legend()
```

✓ 0.2s

```
<matplotlib.legend.Legend at 0x240daa6f2b0>
```

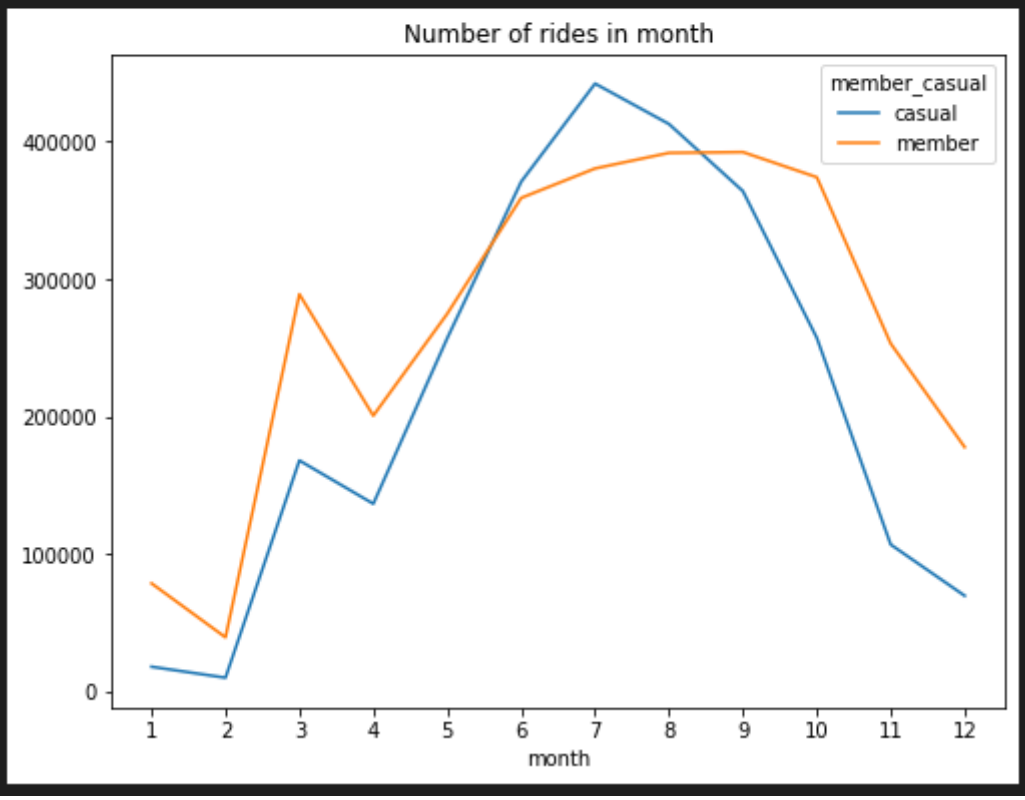


The number of ride of 2 types of users was pretty even.

```
member_type_month.plot(figsize =(8,6))
plt.xticks(member_type_month.index)
plt.title('Number of rides in month')
```

✓ 0.2s

```
Text(0.5, 1.0, 'Number of rides in month')
```



There was a significant increase in the number of ride in the period from May to September.

Types of bike

```
# usage of bike type per month by member type
ride_type = pd.pivot_table(case_1, index= ['member_casual', 'month', 'rideable_type'], aggfunc={'rideable_type':np.size})
ride_type= ride_type.rename(columns={'rideable_type':'total_bike'})
ride_type = ride_type.reset_index()
ride_type
```

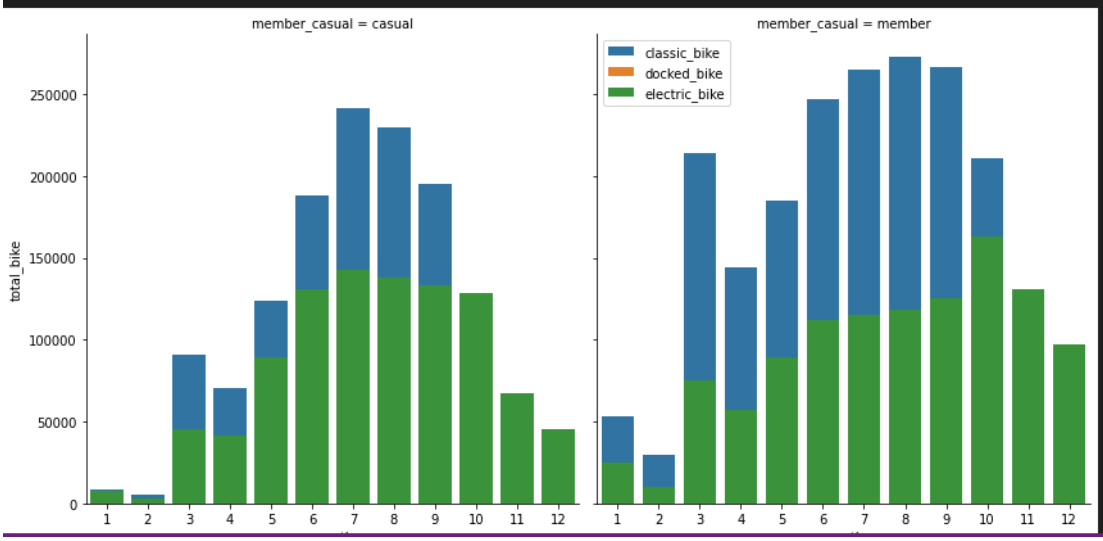
✓ 2.3s

	member_casual	month	rideable_type	total_bike
0	casual	1	classic_bike	8259
1	casual	1	docked_bike	2105
2	casual	1	electric_bike	7753
3	casual	2	classic_bike	5695
4	casual	2	docked_bike	1271
...	...	...	...	...
56	member	10	electric_bike	163434
57	member	11	classic_bike	122173
58	member	11	electric_bike	130876
59	member	12	classic_bike	80829
60	member	12	electric_bike	96973

```
g.map(sns.barplot, 'month','total_bike')
plt.legend()
plt.show()
```

✓ 1.2s

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\axisgrid.py:670: UserWarning: Using the barpl
is likely to produce an incorrect plot.
  warnings.warn(warning)
```



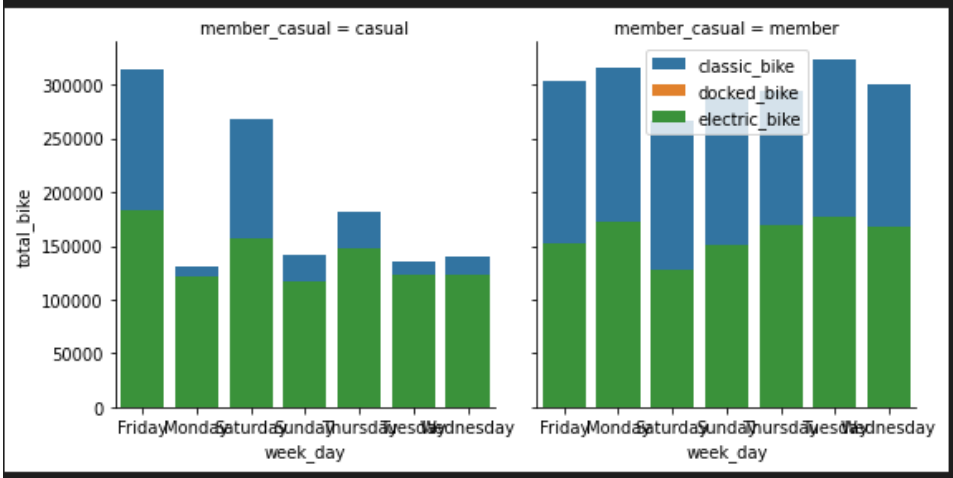
For the type of bikes, it is obvious that classic and electric bikes were the most ones used by both type of users.

```
g= sns.FacetGrid(ride_type_day, col='member_casual', hue='rideable_type', height=4)
g.map(sns.barplot, 'week_day','total_bike')
plt.legend()
plt.show()
```

✓ 0.4s

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\axisgrid.py:670:

Using the barplot function without specifying `order` is likely to produce an incorrect plot.
```



Data by weekday showed the same trends, but it can be seen that casual riders preferred electric bike to classic bike while member rider tended to be opposite.

Trip length

Average trip length on weekdays:

```
weekly_usage = pd.pivot_table(case_1, index=['member_casual', 'week_day'],
                              aggfunc={'member_casual':np.size, 'ride_length':np.mean}
                              ).rename(columns={'member_casual':'total_trips','ride_length':'avg_trip_length'}).reset_index()
weekly_usage
```

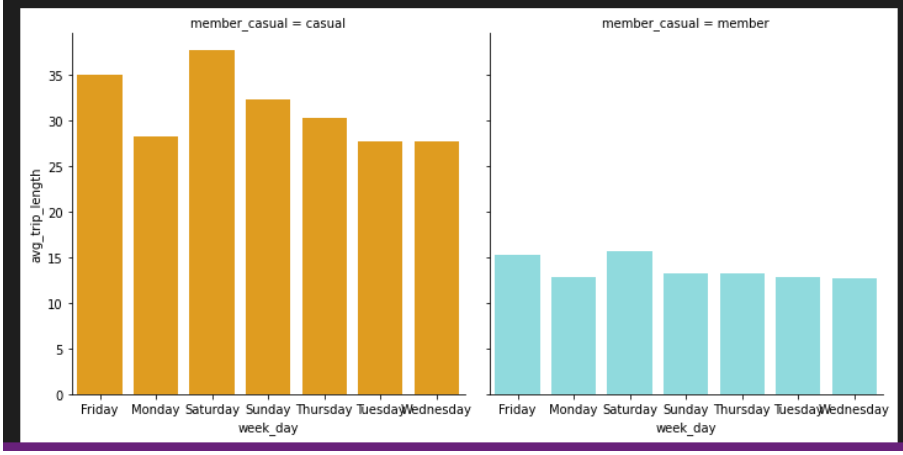
✓ 2.1s

	member_casual	week_day	total_trips	avg_trip_length
0	casual	Friday	580090	34.995022
1	casual	Monday	284855	28.267254
2	casual	Saturday	498485	37.693197
3	casual	Sunday	298366	32.367896
4	casual	Thursday	371857	30.324082
5	casual	Tuesday	287833	27.690464
6	casual	Wednesday	291552	27.752829
7	member	Friday	456209	15.295315
8	member	Monday	489050	12.821334
9	member	Saturday	394427	15.677598
10	member	Sunday	439047	13.293092
11	member	Thursday	464308	13.302639
12	member	Tuesday	499978	12.822820
13	member	Wednesday	467502	12.751772

```
g = sns.FacetGrid(weekly_usage, col='member_casual', hue='member_casual', palette=['orange', '#83e6ea'], height=5)
#g.set_xticklabels(rotation=45)
g.map(sns.barplot, 'week_day', 'avg_trip_length')
plt.show()
```

✓ 0.6s

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\axisgrid.py:670: UserWarning: Using the barplot
is likely to produce an incorrect plot.
  warnings.warn(warning)
```



Average trip length in month:

```
monthly_usage = pd.pivot_table(case_1, index=['member_casual', 'month'],
                               aggfunc={'member_casual':np.size, 'ride_length':np.mean}
                               ).rename(columns={'member_casual':'total_trips','ride_length':'avg_trip_length'}).reset_index()
monthly_usage.head()
```

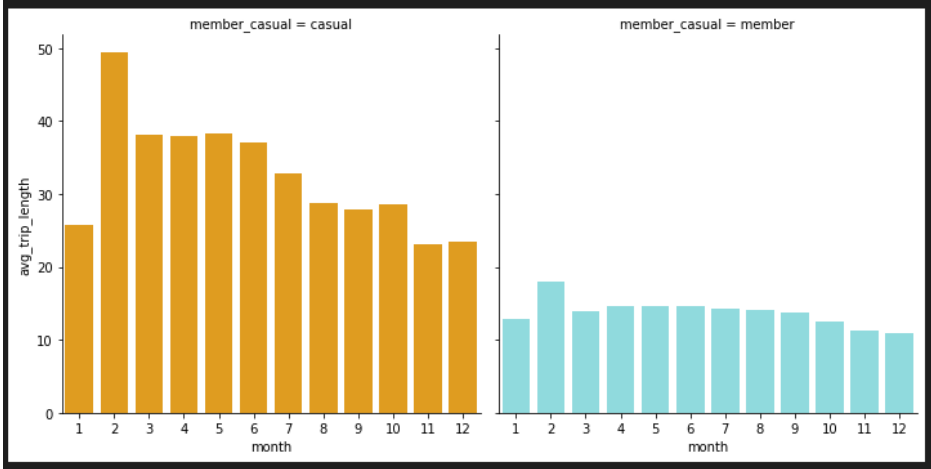
✓ 1.7s

	member_casual	month	total_trips	avg_trip_length
0	casual	1	18117	25.684590
1	casual	2	10131	49.373229
2	casual	3	168066	38.158731
3	casual	4	136601	38.022990
4	casual	5	256916	38.230966

```
g2 = sns.FacetGrid(monthly_usage, col='member_casual', hue='member_casual', palette=['orange', '#83e6ea'], height=5)
#g.set_xticklabels(rotation=45)
g2.map(sns.barplot, 'month', 'avg_trip_length')
plt.show()
```

✓ 0.7s

```
C:\Users\HP\AppData\Local\Programs\Python\Python310\lib\site-packages\seaborn\axisgrid.py:670: UserWarning: Using the bar
is likely to produce an incorrect plot.
  warnings.warn(warning)
```



The charts shows that, casual riders rode much more than member riders.

Geolocation

Geolocation the start station the see where users tended to choose the starting location.

All users:

```
start_station = pd.pivot_table(case_1, index=['start_station_name', 'start_lat', 'start_lng',], aggfunc={'start_station_name':np.size})
start_station= start_station.rename(columns={'start_station_name':'num_trip'})
start_station = start_station.reset_index()
start_station
```

✓ 46.8s

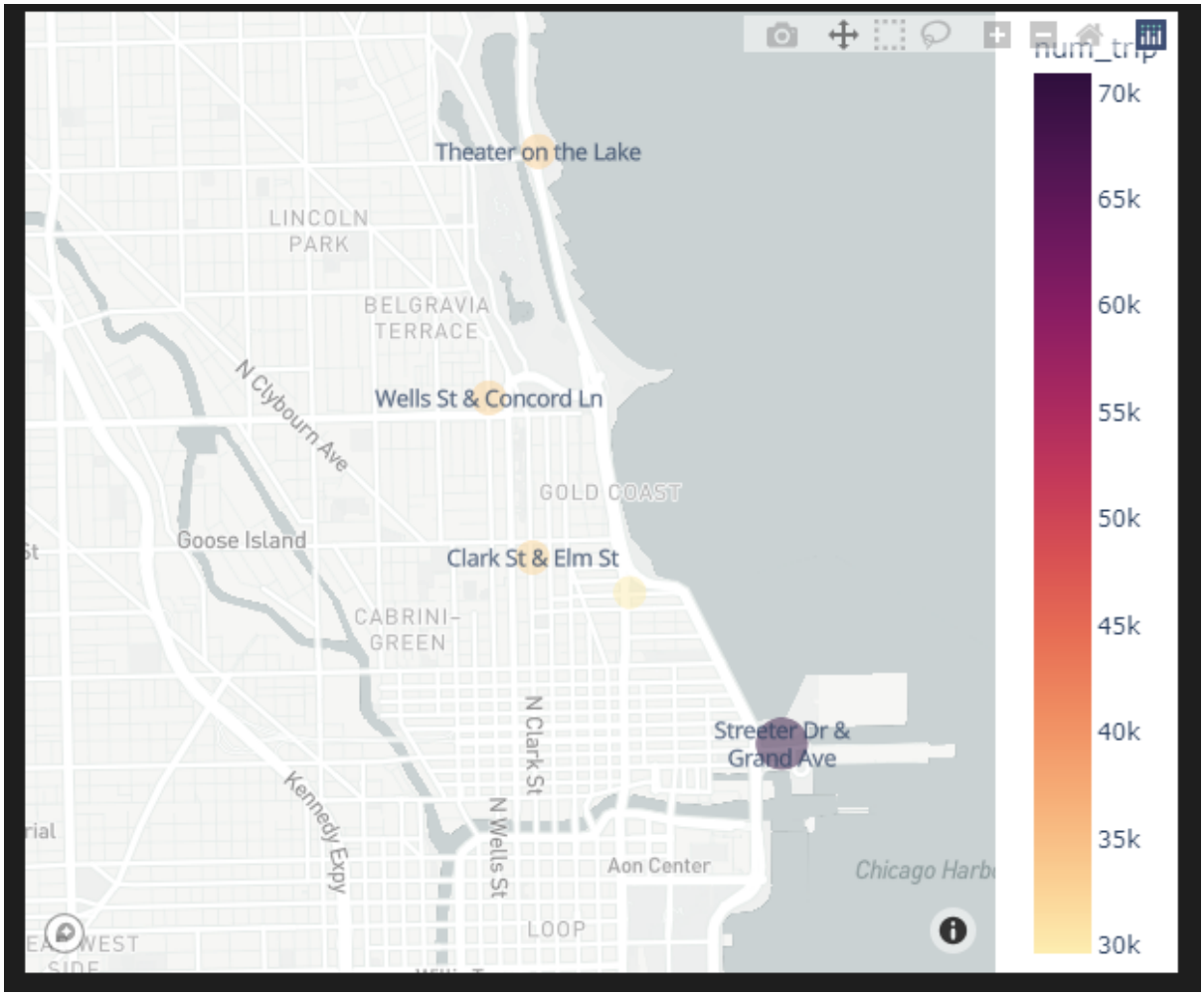
	start_station_name	start_lat	start_lng	num_trip
0	2112 W Peterson Ave	41.991062	-87.683560	2
1	2112 W Peterson Ave	41.991067	-87.683616	1
2	2112 W Peterson Ave	41.991082	-87.683676	1
3	2112 W Peterson Ave	41.991082	-87.683605	1
4	2112 W Peterson Ave	41.991095	-87.683559	1
...	...	...	...	...
1321380	Yates Blvd & 93rd St	41.726189	-87.566373	1
1321381	Yates Blvd & 93rd St	41.726202	-87.566294	1
1321382	Yates Blvd & 93rd St	41.726203	-87.566301	1
1321383	Yates Blvd & 93rd St	41.726206	-87.566314	1
1321384	Yates Blvd & 93rd St	41.726206	-87.566375	1

```
#Top 5 start station used by users
start_station_top5 = start_station.sort_values(by=['num_trip'], ascending=False).head(5)
start_station_top5
```

✓ 0.3s

	start_station_name	start_lat	start_lng	num_trip
1148006	Streeter Dr & Grand Ave	41.892278	-87.612043	70929
1158533	Theater on the Lake	41.926277	-87.630834	32926
1217855	Wells St & Concord Ln	41.912133	-87.634656	32799
228979	Clark St & Elm St	41.902973	-87.631280	32183
813038	Michigan Ave & Oak St	41.900960	-87.623777	29640

```
#Geolocation Top 5 start station used by users
import plotly_express as px
mapbox_access_token = 'pk.eyJ1IjoiaGFtaWJvIiwiYSI6ImNrN2N2Ym5uYTAybzEzb256cmk2NGtjeTUifQ.3qaqDjFrZdS3sqeoPeJG-w'
px.set_mapbox_access_token(mapbox_access_token)
fig = px.scatter_mapbox(start_station_top5, color='num_trip', color_continuous_scale=px.colors.sequential.matter,
    lat ='start_lat', lon= 'start_lng', size ='num_trip', zoom=11, opacity=0.5, text='start_station_name',
    hover_name="start_station_name", hover_data=["start_station_name", "num_trip"])
fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0}, height=500, width=600)
fig.show()
```



Member riders:

```
#Top 5 start station used by members
mem_start_station = pd.pivot_table(case_1[case_1.member_casual=='member'], index=['
mem_start_station= mem_start_station.rename(columns={'start_station_name':'num_trip
mem_start_station = mem_start_station.reset_index()
mem_start_station
```

✓ 32.1s

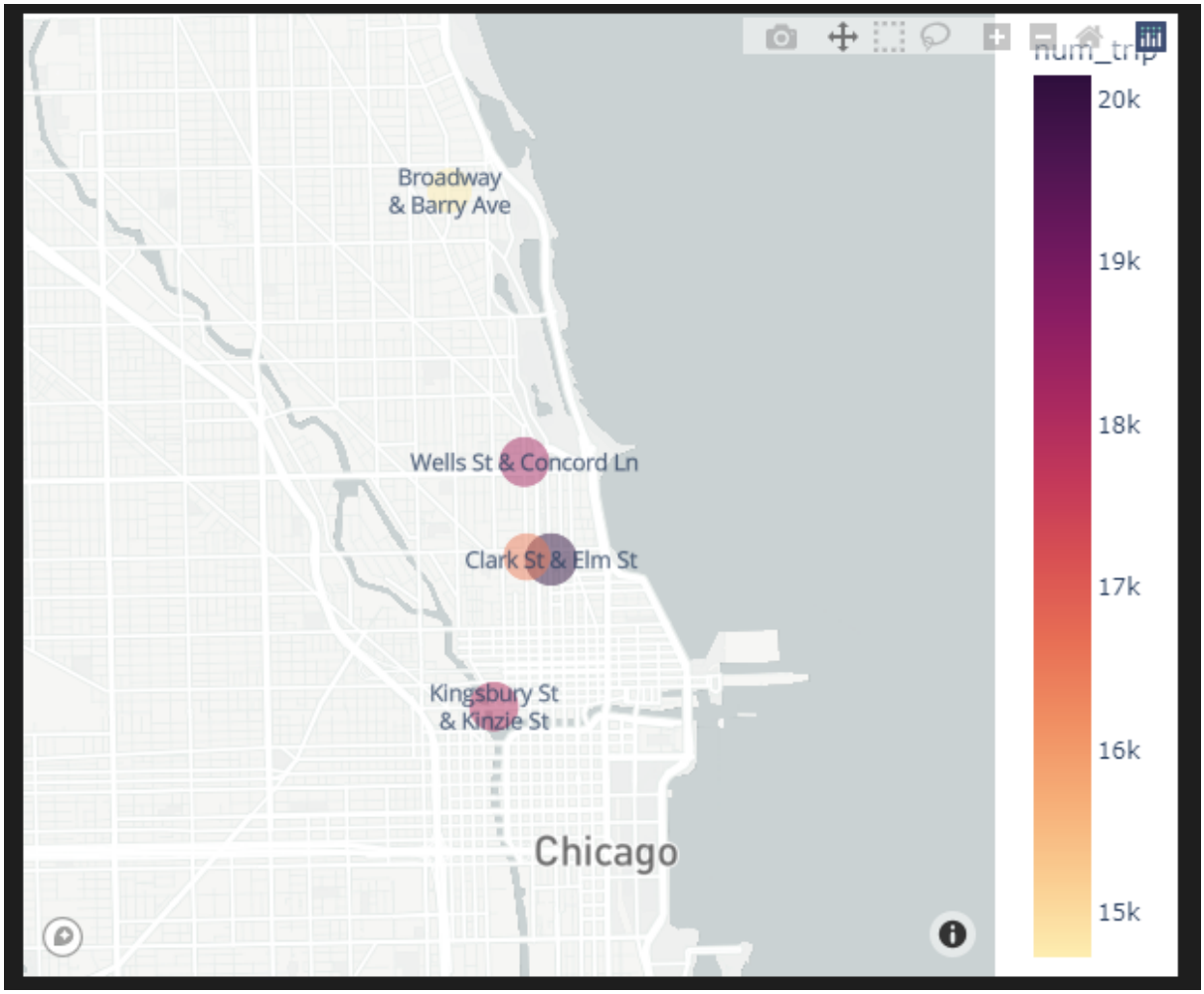
	start_station_name	start_lat	start_lng	member_casual	num_trip
0	2112 W Peterson Ave	41.991062	-87.683560	member	2
1	2112 W Peterson Ave	41.991082	-87.683676	member	1
2	2112 W Peterson Ave	41.991082	-87.683605	member	1
3	2112 W Peterson Ave	41.991097	-87.683590	member	1
4	2112 W Peterson Ave	41.991105	-87.683563	member	1
...	...	...	...	...	...
701194	Yates Blvd & 93rd St	41.726187	-87.566353	member	1
701195	Yates Blvd & 93rd St	41.726189	-87.566373	member	1
701196	Yates Blvd & 93rd St	41.726202	-87.566294	member	1
701197	Yates Blvd & 93rd St	41.726203	-87.566301	member	1
701198	Yates Blvd & 93rd St	41.726206	-87.566314	member	1

```
mem_start_station_top5 = mem_start_station.sort_values(by=['num_trip'], ascending=False).head(5)
mem_start_station_top5
```

✓ 0.2s

	start_station_name	start_lat	start_lng	member_casual	num_trip
122297	Clark St & Elm St	41.902973	-87.631280	member	20151
643922	Wells St & Concord Ln	41.912133	-87.634656	member	18230
344270	Kingsbury St & Kinzie St	41.889177	-87.638506	member	17988
648916	Wells St & Elm St	41.903222	-87.634324	member	16467
37195	Broadway & Barry Ave	41.937582	-87.644098	member	14730

```
#Geolocation Top 5 start station used by members
mapbox_access_token = 'pk.eyJ1IjoiaGFtaWJvIiwiYSI6ImNrN2N2Ym5uYTAybzEzb256cmk2NGtjeTUifQ.3qaqDjFrZdS3sqeoPeJG-w'
px.set_mapbox_access_token(mapbox_access_token)
fig1= px.scatter_mapbox(mem_start_station_top5, color='num_trip',
                        color_continuous_scale=px.colors.sequential.matter,
                        lat ='start_lat', lon= 'start_lng', size ='num_trip',
                        zoom=11.5, opacity=0.5, text='start_station_name')
fig1.update_layout(margin={"r":0,"t":0,"l":0,"b":0}, height=500, width=600)
fig1.show()
```



Casual riders:

```
#Top 5 start station used by casual users
cas_start_station = pd.pivot_table(case_1[case_1.member_casual=='casual'], index=['star
cas_start_station= cas_start_station.rename(columns={'start_station_name':'num_trip'})
cas_start_station = cas_start_station.reset_index()
cas_start_station
```

✓ 25.1s

	start_station_name	start_lat	start_lng	member_casual	num_trip
0	2112 W Peterson Ave	41.991067	-87.683616	casual	1
1	2112 W Peterson Ave	41.991095	-87.683559	casual	1
2	2112 W Peterson Ave	41.991099	-87.683629	casual	1
3	2112 W Peterson Ave	41.991102	-87.683617	casual	1
4	2112 W Peterson Ave	41.991104	-87.683552	casual	1
...	...	...	...	...	...
621859	Yates Blvd & 93rd St	41.726177	-87.566430	casual	1
621860	Yates Blvd & 93rd St	41.726179	-87.566377	casual	1
621861	Yates Blvd & 93rd St	41.726184	-87.566331	casual	1
621862	Yates Blvd & 93rd St	41.726188	-87.566380	casual	1
621863	Yates Blvd & 93rd St	41.726206	-87.566375	casual	1

```
cas_start_station_top5 = cas_start_station.sort_values(by=['num_trip'], ascending=False).head(5)
cas_start_station_top5
```

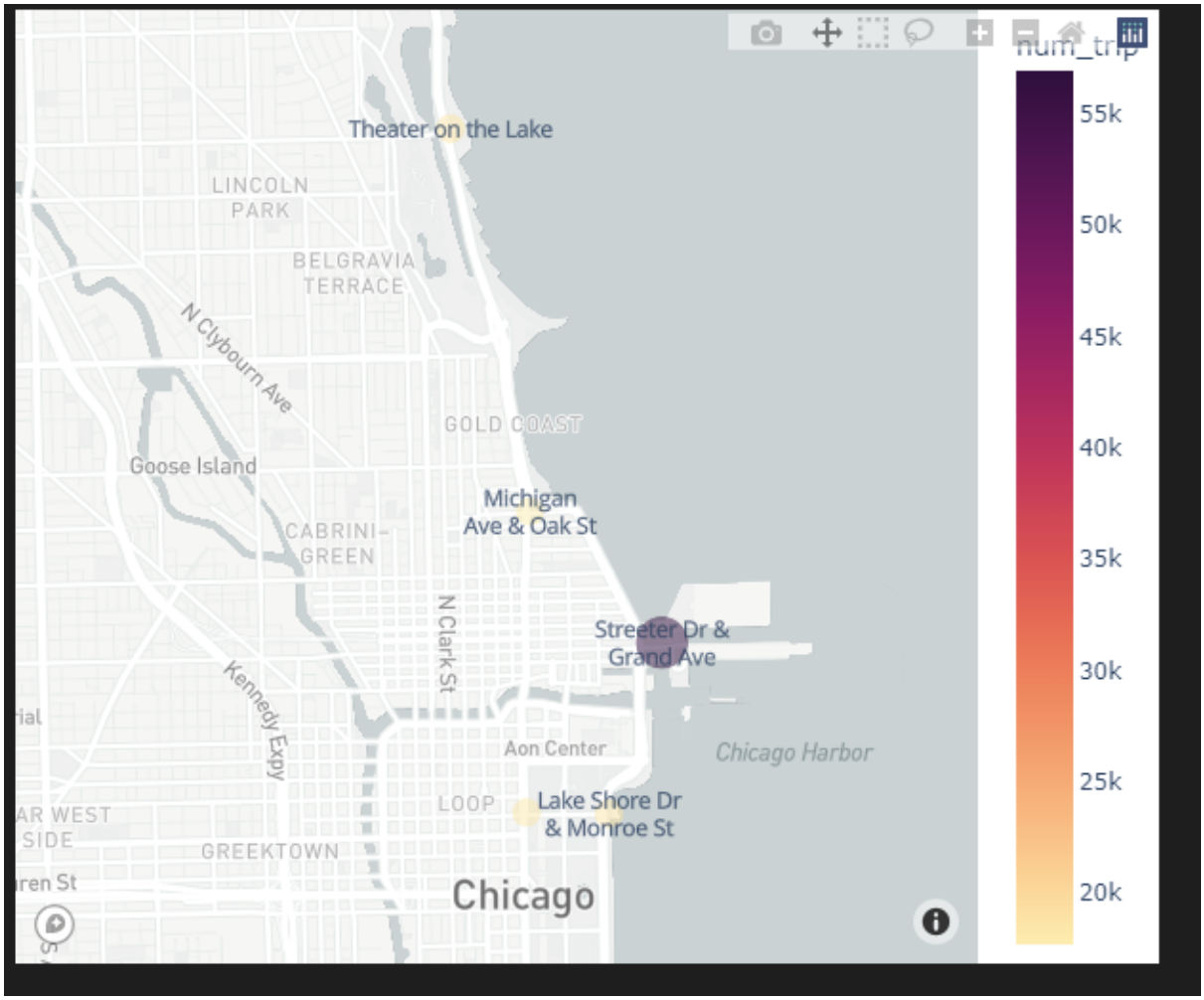
46] ✓ 0.1s

	start_station_name	start_lat	start_lng	member_casual	num_trip
541156	Streeter Dr & Grand Ave	41.892278	-87.612043	casual	56933
548318	Theater on the Lake	41.926277	-87.630834	casual	18526
309103	Lake Shore Dr & Monroe St	41.880958	-87.616743	casual	18401
390842	Millennium Park	41.881032	-87.624084	casual	18205
378095	Michigan Ave & Oak St	41.900960	-87.623777	casual	17700

```
#Geolocation Top 5 start station used by casual users
mapbox_access_token = 'pk.eyJ1IjoiaGFtaWJvIiwiYSI6ImNrN2N2Ym5uYTAybzEzb256cmk2NGtjeTUifQ.3qaqDjFrZdS3sqeoPeJG-w'
px.set_mapbox_access_token(mapbox_access_token)
fig2= (px.scatter_mapbox(cas_start_station_top5, color='num_trip',
                         color_continuous_scale=px.colors.sequential.matter,
                         lat ='start_lat', lon= 'start_lng', size ='num_trip',
                         zoom=12, opacity=0.5, text='start_station_name'))

fig2.update_layout(margin={"r":0,"t":0,"l":0,"b":0},height=500, width=600)
fig2.show()
```

✓ 0.1s



As we can see all types of riders is likely to choose start stations which is nearby the river or riverside, therefore we can conclude that the riders prefer to ride along the river.

VI. Act Phase

1. Casual riders tend to use the service more during the weekends, therefore maybe a promotion through Facebook, Google, Tiktok,... to attract them to use service more on weekdays would be reasonable.
2. Casual riders are likely to ride more in summer and autumn time so the promotion during this time would be more effective.
3. Top 5 start stations used by casual riders are all riverside, advertisements along river would appeal the riders.
4. Docked and electric bikes are the most common bike types to casual riders. It is more productive to focus on these types of transport.