Submitted by Muhammed nasim mp

14/04/2023

About the company

In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles thatare geotracked and locked into a network of 692 stations across Chicago. The bikes can be unlocked from one station and returned to any other station in the system anytime. Until now, Cyclistic's marketing strategy relied on building general awareness and appealing to broad consumer segments. One approach that helped make these things possible was the flexibility of its pricing plans: single-ride passes, full-day passes,and annual memberships. Customers who purchase single-ride or full-day passes are referred to as casual riders. Customers who purchase annual memberships are Cyclistic members. Cyclistic's finance analysts have concluded that annual members are much more profitable than casual riders. Although the pricing flexibility helps Cyclistic attract more customers, Moreno believes that maximizing the number of annual members will be key to future growth. Rather than creating a marketing campaign that targets all-new customers, Moreno believes there is a very good chance to convert casual riders into members. She notes that casual riders are already aware of the Cyclistic program and have chosen Cyclistic for their mobility needs. Moreno has set a clear goal: Design marketing strategies aimed at converting casual riders into annual members. In order to do that, however, the marketing analyst team needs to better understand how annual members and casual riders differ, why casual riders would buy a membership, and how digital media could affect their marketing tactics. Moreno and her team are interested in analyzing the Cyclistic historical bike trip data to identify trends.

How Does a Bike-Share Navigate Speedy Success?

In this case study, I will work for a fictional company, Cyclistic, a bike-share company in Chicago. I will design a new marketing strategy to convert casual riders into annual members. But first, Cyclistic executives must approve my recommendations, so they must be backed up with compelling data insights and professional data visualizations.

What is the problem to be solved?

The director of marketing and finance analysts believe that annual members are more profitable to the company than the casual riders. One of the questions assigned to this analysis to be answered is that how do annual members and casual riders use Cyclistic bikes differently? Thus we are intersted in analyzing the Cyclistic historical bike trip data to identify trends and find the solution to the business task. The result of this analysis will be used to design marketing strategies to convert casual riders into annual members.

Datasets

The required data sources, which included 12 files for last 12 months history of bike trips from 06-2021 to 05-2022, This is public data that we can use to explore how different customer types are using Cyclistic bikes. But data-privacy issues prohibit us from using riders personally identifiable information.

The data provided is in the csv format and is organized in rows and columns. The provided data source comes from a trusted data source so we are sure that it is complete, accurate and reliable and not bias in anyway and it reflects the overal population.

We have access to the original data source from a reliable organization and we are not relying to second or third-party inoformation, so we don't need to worry about the origin of the data we are going to base our analysis on. The provided data soure is comprehensive and contains all the necessary information wee need to answer the questions and accomplish the business task. We can also see that Motivate International keeps the data conitously updated and we have access to the most recent data (for the year 2022) as well. This gives us the confidence that it is the most current information available.

Process

For the purpose of this project I chose Jupyter notebook to do most of the data cleaning tasks and to create related reports and visualizations. In order to make sure the data is cleand, I will checked for any null and duplicated values. The column

In [2]:

```
# importing necessary libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import warnings
import seaborn as sns
warnings.filterwarnings('ignore')
import klib as k
import plotly.express as plx
```

In [3]:

```
#importing datasets
data1=pd.read_csv('./202201-divvy-tripdata.csv')
data2=pd.read_csv('./202202-divvy-tripdata.csv')
data3=pd.read_csv('./202203-divvy-tripdata.csv')
data4=pd.read_csv('./202204-divvy-tripdata.csv')
data5=pd.read_csv('./202205-divvy-tripdata.csv')
data6=pd.read_csv('./202106-divvy-tripdata.csv')
data7=pd.read_csv('./202107-divvy-tripdata.csv')
data8=pd.read_csv('./202108-divvy-tripdata.csv')
data9=pd.read_csv('./202109-divvy-tripdata.csv')
data10=pd.read_csv('./202110-divvy-tripdata.csv')
data11=pd.read_csv('./202111-divvy-tripdata.csv')
data12=pd.read_csv('./202112-divvy-tripdata.csv')
```

In [4]:

```
#merging datasets
frames=[data1,data2,data3,data4,data5,data6,data7,data8,data9,data10,data11,data12]
new_data=pd.concat(frames)
```

In [5]:

```
new_data.head()
```

Out[5]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_station_r
0	C2F7DD78E82EC875	electric_bike	2022-01-13 11:59:47	2022-01-13 12:02:44	Glenwood Ave & Touhy Ave	525	Clark St & T
1	A6CF8980A652D272	electric_bike	2022-01-10 08:41:56	2022-01-10 08:46:17	Glenwood Ave & Touhy Ave	525	Clark St & T
2	BD0F91DFF741C66D	classic_bike	2022-01-25 04:53:40	2022-01-25 04:58:01	Sheffield Ave & Fullerton Ave	TA1306000016	Greenview A Fullertor
3	CBB80ED419105406	classic_bike	2022-01-04 00:18:04	2022-01-04 00:33:00	Clark St & Bryn Mawr Ave	KA1504000151	Paulina Montrose
4	DDC963BFDDA51EEA	classic_bike	2022-01-20 01:31:10	2022-01-20 01:37:12	Michigan Ave & Jackson Blvd	TA1309000002	State Randol

In [6]:

```
new_data.tail()
```

Out[6]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_stat
247535	847431F3D5353AB7	electric_bike	2021-12-12 13:36:55	2021-12-12 13:56:08	Canal St & Madison St	13341	
247536	CF407BBC3B9FAD63	electric_bike	2021-12-06 19:37:50	2021-12-06 19:44:51	Canal St & Madison St	13341	King
247537	60BB69EBF5440E92	electric_bike	2021-12-02 08:57:04	2021-12-02 09:05:21	Canal St & Madison St	13341	Dea
247538	C414F654A28635B8	electric_bike	2021-12-13 09:00:26	2021-12-13 09:14:39	Lawndale Ave & 16th St	362.0	
247539	37AC57E34B2E7E97	classic_bike	2021-12-13 08:45:32	2021-12-13 08:49:09	Michigan Ave & Jackson Blvd	TA1309000002	Dea

In [6]:

```
new_data.sample(10)
```

Out[6]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id	end_stat
127200	99D56D19B79F5BE0	classic_bike	2021-12-04 10:34:18	2021-12-04 10:39:38	Clark St & Schiller St	TA1309000024	Lar
696995	04D443BB47372723	electric_bike	2021-09-25 17:23:14	2021-09-25 17:48:04	NaN	NaN	
220943	2BFA2AEE09A7EA2F	electric_bike	2021-08-27 11:18:58	2021-08-27 11:23:09	Canal St & Madison St	13341	Clinton S
684155	A3B3C0E60AE855A0	classic_bike	2021-08-18 11:57:35	2021-08-18 12:09:27	DuSable Lake Shore Dr & Monroe St	13300	St
225015	4909E2A51D3FCBE8	classic_bike	2022-05-10 18:02:27	2022-05-10 18:39:38	Clinton St & Jackson Blvd	638	Bernard :
410277	7FCD3EE76709CFDB	classic_bike	2021-07-20 08:46:32	2021-07-20 08:54:33	Ashland Ave & Division St	13061	Ash
459936	25A9DE1ADBDDFF41	electric_bike	2021-09-02 18:25:10	2021-09-02 18:39:44	Pine Grove Ave & Irving Park Rd	TA1308000022	Montr
278026	96E5DCA384AB0DE4	classic_bike	2021-10-14 14:29:51	2021-10-14 14:39:07	Paulina St & Flournoy St	KA1504000104	Sang Washi
543210	CF9A43A63228A03A	classic_bike	2022-05-17 17:30:54	2022-05-17 17:47:40	Pine Grove Ave & Waveland Ave	TA1307000150	Lar W
296106	C2EE20371B8C8DB5	docked_bike	2021-06-03 15:46:54	2021-06-03 16:00:38	Wood St & Milwaukee Ave	13221	Clarer

Data cleaning

In [7]:

```
#droping unnecessary columns
new_data.drop(columns='ride_id',inplace=True)
new_data.drop(columns='start_station_id',inplace=True)
```

In [8]:

```
new_data.drop(columns='end_station_id',inplace=True)
```

In [9]:

```
new_data.describe().T
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
start_lat	5860776.0	41.900356	0.047040	41.64	41.881032	41.898886	41.929143	45.635034
start_lng	5860776.0	-87.646809	0.030490	-87.84	-87.661198	-87.642985	-87.627834	-73.796477
end_lat	5855740.0	41.900612	0.047134	41.39	41.881109	41.899181	41.929465	42.168116
end_lng	5855740.0	-87.647004	0.030123	-88.97	-87.661406	-87.643118	-87.627844	-87.490000

In [8]:

```
new_data.shape
```

Out[8]:

```
(5860776, 13)
```

In [9]:

```
new_data.isnull().mean()*100
```

Out[9]:

```
ride_id                0.000000
rideable_type          0.000000
started_at             0.000000
ended_at               0.000000
start_station_name    14.045358
start_station_id      14.045307
end_station_name      14.986718
end_station_id        14.986718
start_lat              0.000000
start_lng              0.000000
end_lat                0.085927
end_lng                0.085927
member_casual          0.000000
dtype: float64
```

In [10]:

```
# droping null values
new_data.dropna(inplace=True)
```

In [11]:

```
new_data.duplicated().sum()
#no duplicated values to drop
```

Out[11]:

```
0
```

In [12]:

```
new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4667299 entries, 0 to 247539
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: 498.5+ MB
```

In [15]:

```
#converting column type object to datetime
new_data[["started_at", "ended_at"]] = new_data[["started_at", "ended_at"]].apply(pd.to_datetime)
```

In [16]:

```
converted_new_data=new_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4667299 entries, 0 to 247539
Data columns (total 10 columns):
 #   Column              Dtype         
---  ------              -----         
 0   rideable_type       object        
 1   started_at          datetime64[ns]
 2   ended_at            datetime64[ns]
 3   start_station_name  object        
 4   end_station_name    object        
 5   start_lat           float64       
 6   start_lng           float64       
 7   end_lat             float64       
 8   end_lng             float64       
 9   member_casual       object        
dtypes: datetime64[ns](2), float64(4), object(4)
memory usage: 391.7+ MB
```

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

To answer the business question, I needed to calculate the duration of the trip. To accomplish this, I added two new columns to the existing data and assign them to ride_length and ride_length_seconds columns.

In [17]:

```
new_data['ride_length'] = new_data.ended_at - new_data.started_at
new_data['ride_length_seconds'] = new_data['ride_length'].dt.total_seconds()
```

In [18]:

```
new_data
```

Out[18]:

	rideable_type	started_at	ended_at	start_station_name	end_station_name	start_lat	start_lng	e
0	electric_bike	2022-01-13 11:59:47	2022-01-13 12:02:44	Glenwood Ave & Touhy Ave	Clark St & Touhy Ave	42.012800	-87.665906	42.0
1	electric_bike	2022-01-10 08:41:56	2022-01-10 08:46:17	Glenwood Ave & Touhy Ave	Clark St & Touhy Ave	42.012763	-87.665967	42.0
2	classic_bike	2022-01-25 04:53:40	2022-01-25 04:58:01	Sheffield Ave & Fullerton Ave	Greenview Ave & Fullerton Ave	41.925602	-87.653708	41.9
3	classic_bike	2022-01-04 00:18:04	2022-01-04 00:33:00	Clark St & Bryn Mawr Ave	Paulina St & Montrose Ave	41.983593	-87.669154	41.9
4	classic_bike	2022-01-20 01:31:10	2022-01-20 01:37:12	Michigan Ave & Jackson Blvd	State St & Randolph St	41.877850	-87.624080	41.8
...	...	...	...	...	...	...	...	
247528	electric_bike	2021-12-07 15:55:37	2021-12-07 16:00:17	Canal St & Madison St	Desplaines St & Kinzie St	41.881372	-87.640042	41.8
247532	electric_bike	2021-12-01 16:50:52	2021-12-01 16:55:18	Canal St & Madison St	Desplaines St & Kinzie St	41.881999	-87.639265	41.8
247536	electric_bike	2021-12-06 19:37:50	2021-12-06 19:44:51	Canal St & Madison St	Kingsbury St & Kinzie St	41.882123	-87.640053	41.8
247537	electric_bike	2021-12-02 08:57:04	2021-12-02 09:05:21	Canal St & Madison St	Dearborn St & Monroe St	41.881956	-87.639955	41.8
247539	classic_bike	2021-12-13 08:45:32	2021-12-13 08:49:09	Michigan Ave & Jackson Blvd	Dearborn St & Monroe St	41.877850	-87.624080	41.8

4667299 rows × 12 columns

Further on, I will compute and assign the day of the week and the hour of the day in which the bike was rented and assign it to the corresponding columns.

Day of the week is mapped this way: (Monday=1, Sunday=7)

In [19]:

```
new_data['which_day_of_week'] = new_data['started_at'].dt.dayofweek
new_data['what_hour_of_day'] = new_data['started_at'].dt.hour
new_data['which_month_of_year'] = new_data['started_at'].dt.month
```

In [20]:

```
new_data
```

Out[20]:

station_name	start_lat	start_lng	end_lat	end_lng	member_casual	ride_length	ride_length_seconds	w
ark St & Touhy Ave	42.012800	-87.665906	42.012560	-87.674367	casual	0 days 00:02:57	177.0	
ark St & Touhy Ave	42.012763	-87.665967	42.012560	-87.674367	casual	0 days 00:04:21	261.0	
eenview Ave & Fullerton Ave	41.925602	-87.653708	41.925330	-87.665800	member	0 days 00:04:21	261.0	
Paulina St & Montrose Ave	41.983593	-87.669154	41.961507	-87.671387	casual	0 days 00:14:56	896.0	
State St & Randolph St	41.877850	-87.624080	41.884621	-87.627834	member	0 days 00:06:02	362.0	
...	...	...	...	...	...	...	...	
esplaines St & Kinzie St	41.881372	-87.640042	41.888456	-87.644336	casual	0 days 00:04:40	280.0	
esplaines St & Kinzie St	41.881999	-87.639265	41.888415	-87.644342	casual	0 days 00:04:26	266.0	
Kingsbury St & Kinzie St	41.882123	-87.640053	41.889106	-87.638862	member	0 days 00:07:01	421.0	
Dearborn St & Monroe St	41.881956	-87.639955	41.880254	-87.629603	member	0 days 00:08:17	497.0	
Dearborn St & Monroe St	41.877850	-87.624080	41.881320	-87.629521	member	0 days 00:03:37	217.0	

.

In order to speed up the process I divided the two categories of the users and assign them to casual_users and member_users dataframes.

In [21]:

```
casual_users = new_data[new_data['member_casual'] == 'casual']
member_users = new_data[new_data['member_casual'] == 'member']
```

In [22]:

```
new_data
```

Out[22]:

	rideable_type	started_at	ended_at	start_station_name	end_station_name	start_lat	start_lng	e
0	electric_bike	2022-01-13 11:59:47	2022-01-13 12:02:44	Glenwood Ave & Touhy Ave	Clark St & Touhy Ave	42.012800	-87.665906	42.0
1	electric_bike	2022-01-10 08:41:56	2022-01-10 08:46:17	Glenwood Ave & Touhy Ave	Clark St & Touhy Ave	42.012763	-87.665967	42.0
2	classic_bike	2022-01-25 04:53:40	2022-01-25 04:58:01	Sheffield Ave & Fullerton Ave	Greenview Ave & Fullerton Ave	41.925602	-87.653708	41.9
3	classic_bike	2022-01-04 00:18:04	2022-01-04 00:33:00	Clark St & Bryn Mawr Ave	Paulina St & Montrose Ave	41.983593	-87.669154	41.9
4	classic_bike	2022-01-20 01:31:10	2022-01-20 01:37:12	Michigan Ave & Jackson Blvd	State St & Randolph St	41.877850	-87.624080	41.8
...	...	...	...	...	...	...	...	
247528	electric_bike	2021-12-07 15:55:37	2021-12-07 16:00:17	Canal St & Madison St	Desplaines St & Kinzie St	41.881372	-87.640042	41.8
247532	electric_bike	2021-12-01 16:50:52	2021-12-01 16:55:18	Canal St & Madison St	Desplaines St & Kinzie St	41.881999	-87.639265	41.8
247536	electric_bike	2021-12-06 19:37:50	2021-12-06 19:44:51	Canal St & Madison St	Kingsbury St & Kinzie St	41.882123	-87.640053	41.8
247537	electric_bike	2021-12-02 08:57:04	2021-12-02 09:05:21	Canal St & Madison St	Dearborn St & Monroe St	41.881956	-87.639955	41.8
247539	classic_bike	2021-12-13 08:45:32	2021-12-13 08:49:09	Michigan Ave & Jackson Blvd	Dearborn St & Monroe St	41.877850	-87.624080	41.8

4667299 rows × 15 columns

Now that the data is stored appropriately and has been prepared for analysis, I will start putting it to work. Since the purpose of this analysis is to know how users (casual and members) use Cyclistic differently

In [24]:

```
#droping values in ride_length_column that lower than 0
new_data.drop(new_data[new_data['ride_length_seconds'] < 0].index, inplace = True)
```

In [25]:

```
min_ride_length_casual = casual_users['ride_length'].min()
min_ride_length_member = member_users['ride_length'].min()

mean_ride_length_casual = casual_users['ride_length'].mean()
mean_ride_length_member = member_users['ride_length'].mean()
```

In [26]:

```
print('Minimum ride length of casual :',min_ride_length_casual)
print('Minimum ride length of member :',min_ride_length_member)

print('Mean ride length of casual :',mean_ride_length_casual)
print('Mean ride length of member :',mean_ride_length_member)
```

```
Minimum ride length of casual : -1 days +23:04:06
Minimum ride length of member : -1 days +23:05:55
Mean ride length of casual : 0 days 00:30:06.016790533
Mean ride length of member : 0 days 00:12:40.530999982
```

In this result it is understandable that on average, casual users spend more time cycling than members.

How different users ride accross each day of the week

It is great to have a better understanding of renting behavious of the two groups of users per each day of the week:

In [27]:

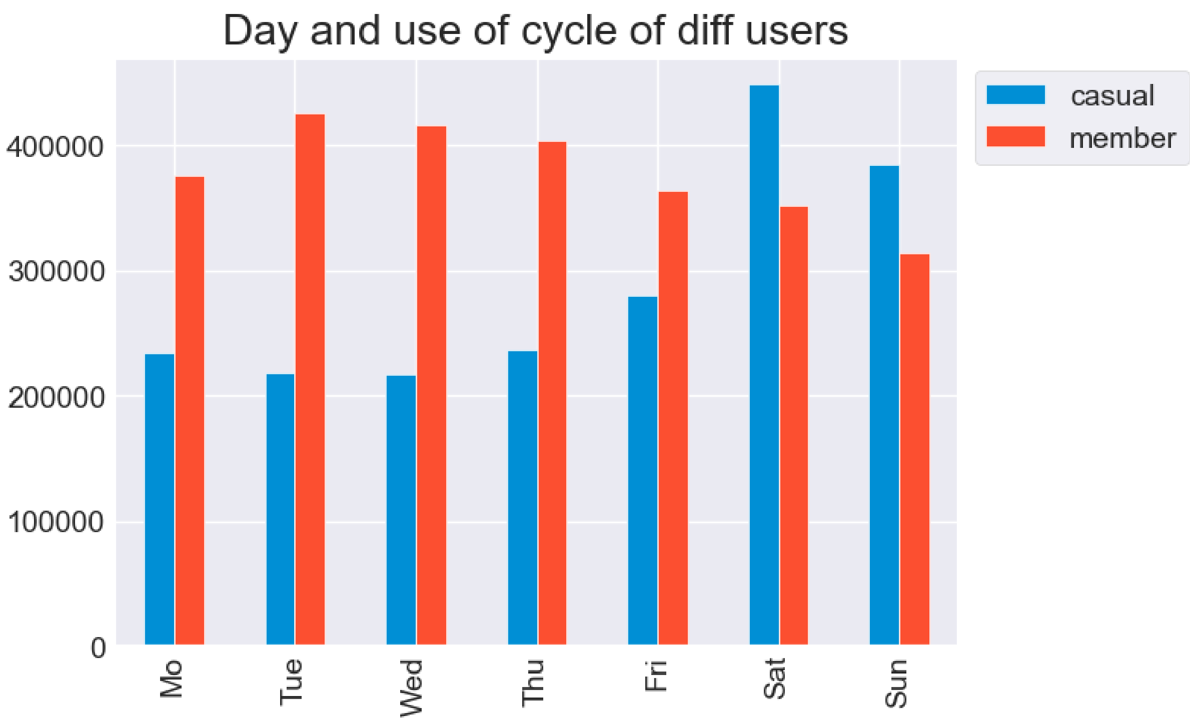
```
pivot_by_day = new_data.pivot_table(index='which_day_of_week',
               columns='member_casual',
               values='rideable_type',
               aggfunc=len,
               fill_value=0)

pivot_by_day.set_axis(['Mo', 'Tue','Wed', 'Thu', 'Fri', 'Sat', 'Sun'], axis=0, inplace=True )
pivot_by_day
```

Out[27]:

member_casual	casual	member
Mo	234476	375390
Tue	218114	424836
Wed	217423	415199
Thu	236528	403001
Fri	279818	364131
Sat	447979	350948
Sun	384549	314058

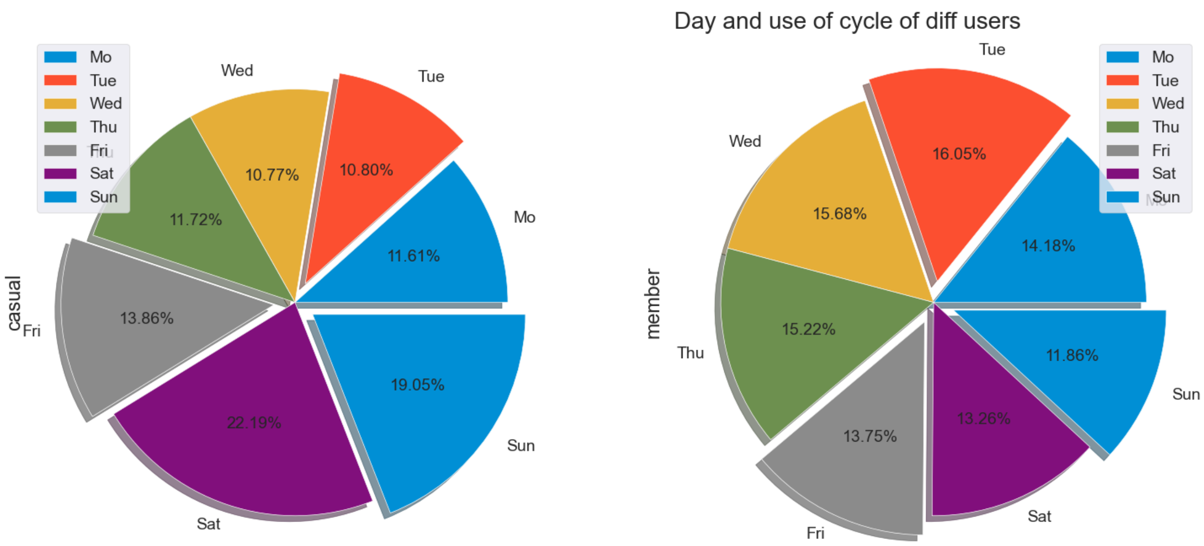
In [115]:

```
pivot_by_day.plot.bar()
plt.title('Day and use of cycle of diff users')
plt.legend(bbox_to_anchor=(1,1))

plt.show()
```



In [114]:

```
myexplode=0,0.1,0,0,0.1,0,0.1
pivot_by_day.plot(kind='pie', subplots=True, figsize=(16,8),autopct='%1.2f%%',explode=myexplode,shadow
plt.title(' Day and use of cycle of diff users', loc="left")

plt.show()
```



From the above graphs, it is clear that casual users mostly rent and ride bikes on weekends (Saturday and Sunday), however this happens on Tuesday for members. On Tuesday and Wednesday less causual users rent a bike from Cyclistic. Annual members, by the way ride less often on Sunday.

What sort of bikes each user group uses mostly¶

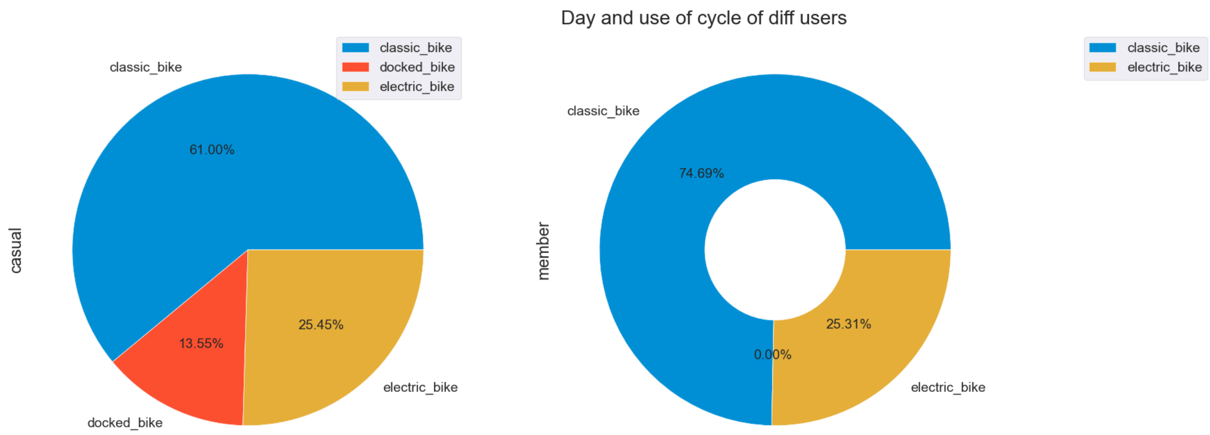
To find this, we need to categorize the rides by the membership type and the bike type they used for the ride:

In [32]:

```
pivot_by_type = new_data.pivot_table(index='rideable_type',
               columns='member_casual',
               values='start_station_name',
               aggfunc=len,
               fill_value=0)
```

In [113]:

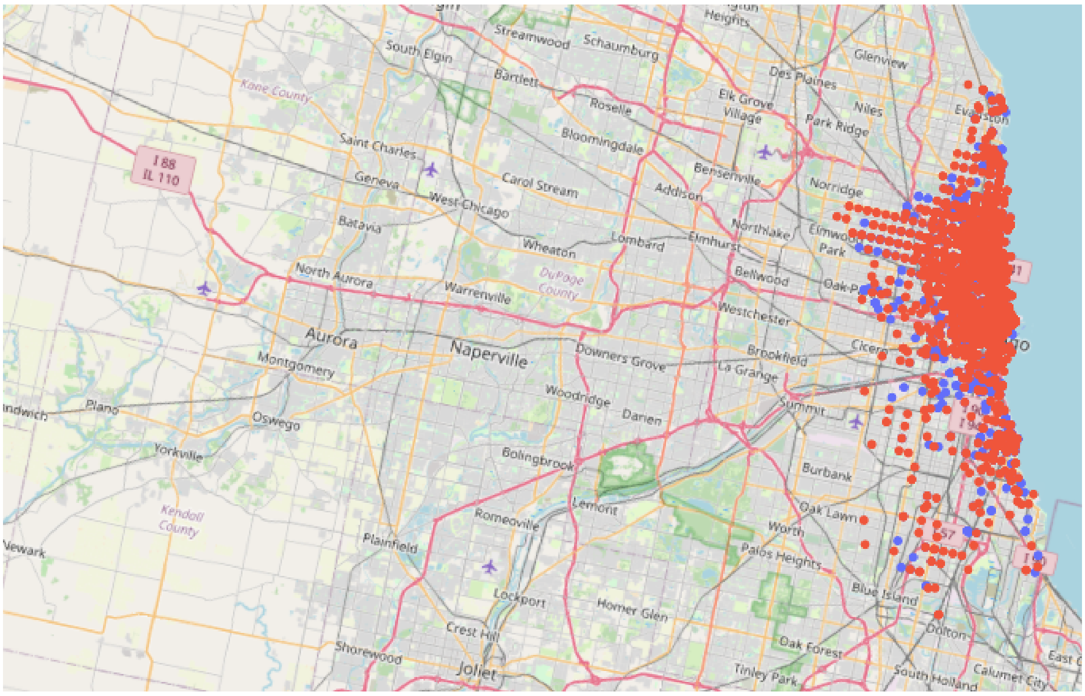
```
pivot_by_type.plot(kind='pie', subplots=True, figsize=(16,8),autopct='%1.2f%%')
plt.title(' Day and use of cycle of diff users', loc="left")
plt.legend(bbox_to_anchor=(1.5,1))
centre_circle = plt.Circle((0,0),0.40,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.figure(figsize=[20,20])
plt.show()
```




```
<Figure size 2000x2000 with 0 Axes>
```

In [7]:

```
sampl_data=new_data.sample(10000)
fig=plx.scatter_mapbox(sampl_data,lat='start_lat',lon='start_lng',color="member_casual")
fig.update_layout(mapbox_style="open-street-map")
fig.show()
```



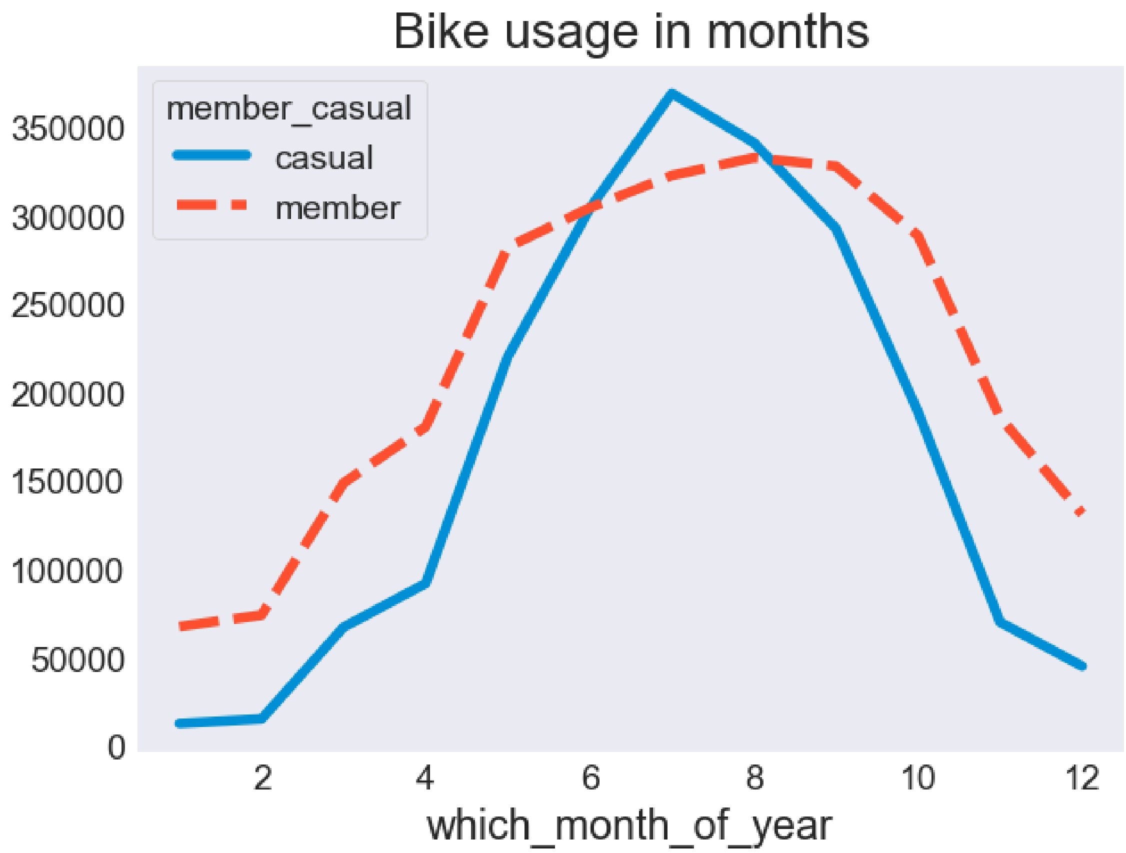
From the above map, we can see that casual riders mostly ride around the city center where the most cultural and recreational places are located. In contrast, bike usage of the annual members happens lessly around the recreational spots and instead, it is a lot common in office buildings areas, which suggest that the most of the Cyclistic members use the bike to commute to work

In [48]:

```
pivot_by_month = new_data.pivot_table(index='which_month_of_year',
               columns='member_casual',
               values='start_station_name',
               aggfunc=len,
               fill_value=0)
```

In [110]:

```
sns.lineplot(data=pivot_by_month)
plt.title('Bike usage in months')
plt.style.use('fivethirtyeight')

sns.set_style('dark')
```



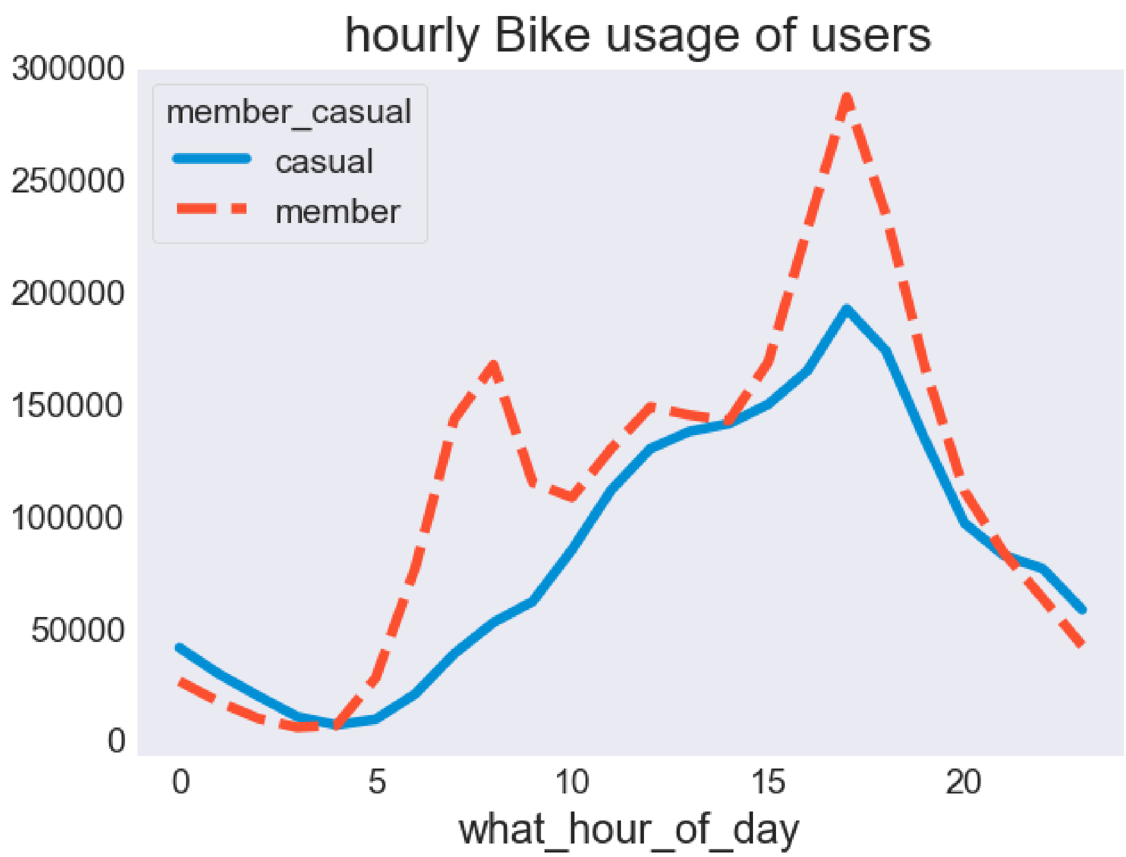
It is clear from the graph that casual and member users are using cycle in summer months,comparitively low use in winter

In [111]:

```
hourly_usage=new_data.pivot_table(index='what_hour_of_day',
                                   columns='member_casual',
                                  values='start_station_name',
                                 aggfunc=len)
```

In [112]:

```
sns.lineplot(data=hourly_usage)
plt.title(' hourly Bike usage of users')
sns.set_style('darkgrid')
xlabel='Hour'
```



In this graph it is well clear that from 5.00 morning casual and member users start to use the bike and the use decrease after 10.00 pm

My findings

any of the users group prefer to use docked type bikes. However the classic bike type is the most favourable for either casual users or annual members.

found that yearly members mostly use Cyclistic bike to commute to and from their work, while casual users borrowed the bikes for recreational purposes.

Heavy traffic takesplace around 17:00 PM for both user categories.

Bike usage peaks during the summer months and the lowest usage happens in January and Februrary.

On average, casual users spend more time cycling than members.

Casual users using bike more in weekend days

member users bike usage is high in work days

what i suggest

when the bike usage peak happens during the summer months, we can offer promotions for summer months to casual users turning them to annual mmbers during.

Casual users ride longer trips than annual members. Based on this insight, we can target those casual riders who rides the longest and offer them discount code as an award for their annual membership.

Classic bikes are more popular among casual users, so in order to turn them into annual users, we can offer them a discount for this type of bike.

we can provide diffrent catogary plans for annual members to attract casual users