Cyclistic Bike Sharing

Executive Summary

Cyclistic introduced a bike-share program in 2016, which has experienced remarkable success. The program has expanded to include a fleet of 5,824 bicycles, equipped with geotracking technology and securely docked at 692 stations throughout Chicago. Moreno, the marketing director at Cyclistic, recognizes that the strategic focus on increasing the number of annual members will be pivotal for driving future growth.

The objective of this project was to develop marketing strategies targeting casual riders with the aim of converting them into annual members. Our analysis revealed that casual riders tend to rent a higher quantity of bikes over the weekends, engaging in longer rides primarily for leisure and recreational purposes. On the other hand, members demonstrate a consistent daily utilization pattern, utilizing Cyclistic's bikes for shorter rides indicative of commuting to work or school, as well as running errands throughout the week. To complete this project I will follow the steps of the data analysis process: ask, prepare, process, analyze, share, and act.

Based on a thorough analysis of the data, we recommend several strategic actions for the company. Firstly, there is a need to enhance the marketing campaign by leveraging targeted channels such as email, text messages, and app notifications. Secondly, implementing a rewards points system for members would be beneficial. Lastly, introducing new membership options tailored to regular weekend users and seasonal riders would help attract and accommodate their specific preferences.

Introduction

This Cyclistic Bike Sharing Project is a Capstone Project of [Google Data Analytics Professional Certificate (https://www.coursera.org/professional-certificates/google-data-analytics?)](https://www.coursera.org/professional-certificates/google-data-analytics?) which is available on Coursera. It is about a bike sharing company called Cyclistic which is located in Chicago, USA. In 2016, Cyclistic launched a successful bike-share offering. Since then, the program has grown to a fleet of 5,824 bicycles that are 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.

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

STAKEHOLDERS:

Lily Moreno – Cyclistic’s marketing director.

Cyclistic marketing analytics team – A team of data analysts who are responsible for collecting, analyzing, and reporting data that helps guide the company’s marketing strategy.

Cyclistic executive team – Responsible for deciding whether to approve the recommended marketing program.

Goal:

Design marketing strategies aimed at converting casual riders into annual members.

Business Tasks:

Analyze the Cyclistic historical bike trip data to identify trends.

ASK

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?

PREPARE

The data has been made available by Motivate International Inc. [Dataset Website (https://divvy-tripdata.s3.amazonaws.com/index.html?)](https://divvy-tripdata.s3.amazonaws.com/index.html?). The datasets are named differently because Cyclistic is a fictional company. Divvy, the name you will see on the files, is a real bike-share system in Chicago with over 600 stations and 6,000+ bikes across the city. For this reason, the data is quite appropriate and will help us explore how different customer types are using Cyclistic's bikes.

We will be working with 12 CSV files, comprehending the dataset for the year 2022 where each moth has its own CSV file.

Each file has information regarding the bike trip such as ride id, type of bike they are using, start and end stations name, their membership type and more.

Bikeshare hereby grants to you a non-exclusive, royalty-free, limited, perpetual license to access, reproduce, analyze, copy, modify, distribute in your product or service and use the Data for any lawful purpose [License (https://ride.divvybikes.com/data-license-agreement?)](https://ride.divvybikes.com/data-license-agreement?).

Data Limitation:

There is no information regarding daily-pass price and annual membership prices. So, we would not be able to find out which membership type spend more money.

Also, we don't have any information regarging the gender, age and neighbourhood of the riders.

PROCESS

We will begin this phase by loading the libraries and datasets. Then we will explore the dataset and get the overview of the data. After that we will perform Data Cleaning and Data Transformation.

Loading Libraries

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
import seaborn as sns

from datetime import timedelta

from pandas.api.types import CategoricalDtype
```

Loading Datasets

In [2]:
```
# read csv

df = pd.concat(map( pd.read_csv,
                                 ['202201-divvy-tripdata.csv',
                                  '202202-divvy-tripdata.csv',
                                  '202203-divvy-tripdata.csv',
                                  '202204-divvy-tripdata.csv',
                                  '202205-divvy-tripdata.csv',
                                  '202206-divvy-tripdata.csv',
                                  '202207-divvy-tripdata.csv',
                                  '202208-divvy-tripdata.csv',
                                  '202209-divvy-tripdata.csv',
                                  '202210-divvy-tripdata.csv',
                                  '202211-divvy-tripdata.csv',
                                  '202212-divvy-tripdata.csv'
                                 ]
                  ), ignore_index=True )
```

Data Exploration

Overview of the data

In [3]:
```
df.head()
```

Out[3]:

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

Closer Look at the dataset

In [4]:
```
# what is the datatype of columns in the dataset?

df.info()         # or  df.dtypes
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5667717 entries, 0 to 5667716
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: 562.1+ MB
```

In [5]:
```
# How many rows and columns in the dataset?

df.shape
```

Out[5]:
```
(5667717, 13)
```

In [6]:
```
# How many unique rides are there in total?

df['ride_id'].nunique()

# Takeaway: All 5,667,717 ride ids are unique.
```

Out[6]:
```
5667717
```

In [7]:
```
# How many types of bikes are there?

df['rideable_type'].value_counts()

# Takeaway: So there are there types of bikes in the dataset.
```

Out[7]:
```
electric_bike    2889029
classic_bike     2601214
docked_bike       177474
Name: rideable_type, dtype: int64
```

Data Cleaning

Checking Null or missing values

In [8]:
```
# checking missing or null values in each columns

df.isna().sum()
# df.isnull().sum()
```

Out[8]:
```
ride_id                    0
rideable_type              0
started_at                 0
ended_at                   0
start_station_name    833064
start_station_id      833064
end_station_name      892742
end_station_id        892742
start_lat                  0
start_lng                  0
end_lat                 5858
end_lng                 5858
member_casual              0
dtype: int64
```

In [9]:
```
# Percentage of missing or null values in each column

df.isna().sum()/df.shape[0]

# Takeaway: maximum around 15% null values in end_station_name and end_station
```

Out[9]:
```
ride_id               0.000000
rideable_type         0.000000
started_at            0.000000
ended_at              0.000000
start_station_name    0.146984
start_station_id      0.146984
end_station_name      0.157514
end_station_id        0.157514
start_lat             0.000000
start_lng             0.000000
end_lat               0.001034
end_lng               0.001034
member_casual         0.000000
dtype: float64
```

In [10]:
```
# total number of empty cells in the dataset

df.isnull().sum().sum()
```

Out[10]:
```
3463328
```

In [11]:
```
(df.isnull().sum().sum())/(df.shape[0]*df.shape[1])*100

# around 5% cells are empty in the complete dataset.
```

Out[11]:
```
4.700478978640715
```

Dropping null or missing values

In [12]:
```
df.dropna(axis=0, inplace=True)
```

Verifying that all null or missing values are dropped.

In [13]:
```
df.isna().sum()
```

Out[13]:
```
ride_id               0
rideable_type         0
started_at            0
ended_at              0
start_station_name    0
start_station_id      0
end_station_name      0
end_station_id        0
start_lat             0
start_lng             0
end_lat               0
end_lng               0
member_casual         0
dtype: int64
```

Checking Duplicates

In [14]:
```
# checking duplicate rows in the dataset

df.duplicated().sum()

# Takeaway: There are no duplicate rows in the dataset.
```

Out[14]:
```
0
```

Data Transformation

Fixing Datatypes

In [15]:
```
# changing datatype of started_at and ended_at columns to datetime.

df['started_at'] = pd.to_datetime(df['started_at'], dayfirst=True)
df['ended_at'] = pd.to_datetime(df['ended_at'], dayfirst=True)
```

Making new columns

In [16]:
```
df['year'] = df['started_at'].dt.year
df['month'] = df['started_at'].dt.month             # 2nd way   # df['month']
df['day'] = df.started_at.apply(lambda x: x.day)
df['hour'] = df['started_at'].apply(lambda x: x.hour)


df = df.astype({'year': 'int16',
                'hour': 'int8'
                })


df['Total_ride_time'] = (df['ended_at'] - df['started_at'])/timedelta(minutes=
df['Total_ride_time'] = df['Total_ride_time'].round(decimals=1)


cats1 = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'Augu
df['month_name'] = df['started_at'].dt.month_name()
df['month_name'] = df['month_name'].astype(CategoricalDtype(categories=cats1,


cats2 = ['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Sa
df['day_of_week'] = df['started_at'].dt.day_name()
df['day_of_week'] = df['day_of_week'].astype(CategoricalDtype(categories=cats2


# verfiy that new columns are added to the dataset
df.head(3)
```

Out[16]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_station_id
0	C2F7DD78E82EC875	electric_bike	2022-01-13 11:59:47	2022-01-13 12:02:44	Glenwood Ave & Touhy Ave	525
1	A6CF8980A652D272	electric_bike	2022-01-10 08:41:56	2022-01-10 08:46:17	Glenwood Ave & Touhy Ave	525
2	BD0F91DFF741C66D	classic_bike	2022-01-25 04:53:40	2022-01-25 04:58:01	Sheffield Ave & Fullerton Ave	TA1306000016

Check for Outliers

In [17]:
```
# check rows which are not making sense
# How many rows are there with total ride time less than a minute or negative

df[df['Total_ride_time'] <1].count()
```

Out[17]:
```
ride_id               74266
rideable_type         74266
started_at            74266
ended_at              74266
start_station_name    74266
start_station_id      74266
end_station_name      74266
end_station_id        74266
start_lat             74266
start_lng             74266
end_lat               74266
end_lng               74266
member_casual         74266
year                  74266
month                 74266
day                   74266
hour                  74266
Total_ride_time       74266
month_name            74266
day_of_week           74266
dtype: int64
```

Removing Outliers

In [18]:
```
# deleting rows with total ride time less than a minute

df = df[df['Total_ride_time'] >= 1]
df = df.reset_index(drop=True)      # by default reset_index makes a new colu
                                    # so, by mentioning drop=True, we can dro
                                    # 2nd way, df = df.drop(columns=['index']
```

ANALYZE

Checking main columns of interest

(i.e member_casual column which shows Customer Type)

Total Rides by Customer Type

In [19]:
```
# Checking Target column which is member_casual describing "cutomer type"

df['member_casual'].value_counts()
```

Out[19]:
```
member    2562932
casual    1732162
Name: member_casual, dtype: int64
```

In [20]:
```
df['member_casual'].value_counts().member/df.shape[0]

# Takeaway: 59% are member and rest 41% are casual
```

Out[20]:
```
0.5967115038692983
```

Average Ride Duration by Customer Type

In [21]:
```
df.groupby('member_casual', as_index =True)[['Total_ride_time']].mean()

# Takeaway: On average, casual riders have a ride duration that is twice as lo
```

Out[21]:

	Total_ride_time
member_casual	
casual	24.346798
member	12.678553

Total rides in different bikes by Customer Type

In [22]:
```
# What is the most used ride by Customer Type?

df.groupby(['member_casual', 'rideable_type'], as_index=True)[['ride_id']].cou

# Takeaway: classic_bike is a popular choice in both casual and member custome
```

Out[22]:

		ride_id
member_casual	**rideable_type**	
casual	**classic_bike**	876465
	docked_bike	173398
	electric_bike	682299
member	**classic_bike**	1683675
	electric_bike	879257

Average ride duration of different bikes by customer Type

In [23]:
```
# On Average which ride has higest ride duration by customer type?

df.groupby(['member_casual', 'rideable_type'])[['Total_ride_time']].mean().sor

# Takeaway: Average ride duration for electric bikes are less than classic bik
```

Out[23]:

member_casual	rideable_type	Total_ride_time
casual	docked_bike	51.131608
	classic_bike	24.783909
	electric_bike	16.978262
member	classic_bike	13.424618
	electric_bike	11.249924

Time Based Analysis

Most popular month for Bike Rentals

In [24]:
```
pd.pivot_table(df,
               index=['month_name', 'member_casual'],
               values = ['ride_id'],
               aggfunc = ['count']
              )
```

Out[24]:

		count
		ride_id
month_name	member_casual	
January	casual	12483
	member	66608
February	casual	14977
	member	72733
March	casual	66425
	member	146573
April	casual	90847
	member	177819
May	casual	217051
	member	277312
June	casual	287709
	member	322438
July	casual	306806
	member	324495
August	casual	265922
	member	328766
September	casual	217606
	member	308043
October	casual	148950
	member	257607
November	casual	72389
	member	178846
December	casual	30997
	member	101692

Most popular Day of the week for bike Rentals

In [25]:
```
pd.pivot_table(df,
               index = ['day_of_week', 'member_casual'],
               values = ['ride_id'],
               aggfunc = ['count']
              )
```

Out[25]:

		count
		ride_id
day_of_week	member_casual	
Sunday	casual	296795
	member	291784
Monday	casual	207649
	member	368451
Tuesday	casual	193512
	member	404089
Wednesday	casual	200612
	member	405380
Thursday	casual	226674
	member	408446
Friday	casual	245128
	member	353272
Saturday	casual	361792
	member	331510

Most Popular Hour of the day for Bike Rentals

In [26]:
```
pd.pivot_table(df,
               index = ['hour', 'member_casual'],
               values = ['ride_id'],
               aggfunc = ['count']
              )
```

Out[26]:

hour	member_casual	count ride_id
0	casual	32759
	member	24690
1	casual	21000
	member	15259
2	casual	12489
	member	8399
3	casual	7092
	member	5143
4	casual	4579
	member	6061
5	casual	8304
	member	25692
6	casual	21436
	member	74487
7	casual	37402
	member	140891
8	casual	51549
	member	164229
9	casual	53821
	member	110492
10	casual	71315
	member	102226
11	casual	93285
	member	122623
12	casual	110061
	member	141405
13	casual	114191
	member	138971
14	casual	120800
	member	138413
15	casual	133467
	member	168712

		count
		ride_id
hour	member_casual	
16	**casual**	149790
	member	228217
17	**casual**	167394
	member	276170
18	**casual**	148084
	member	219622
19	**casual**	112381
	member	156585
20	**casual**	81806
	member	108440
21	**casual**	70106
	member	83825
22	**casual**	62933
	member	62587
23	**casual**	46118
	member	39793

Average ride duration by month

In [27]:
```
# Which months have high ride duration?

pd.pivot_table(df,
               index = ['month_name', 'member_casual'],
               values = ['Total_ride_time'],
               aggfunc = ['mean'],
               margins = True,
               margins_name = 'ride duration mean'
              )
```

Out[27]:

		mean
		Total_ride_time
month_name	member_casual	
January	casual	27.618898
	member	10.405459
February	casual	25.095994
	member	10.828147
March	casual	28.744385
	member	11.971796
April	casual	26.241908
	member	11.788907
May	casual	28.106695
	member	13.533155
June	casual	25.391604
	member	13.924662
July	casual	25.484390
	member	13.765817
August	casual	23.642685
	member	13.355447
September	casual	22.128907
	member	12.867177
October	casual	20.787402
	member	11.908533
November	casual	17.513144
	member	11.024843
December	casual	15.077127
	member	10.413488
ride duration mean		17.384222

Average ride duration by Day of the week

In [28]:
```
pd.pivot_table(df,
               index = ['day_of_week', 'member_casual'],
               values = ['Total_ride_time'],
               aggfunc = ['mean']
              )
```

Out[28]:

		mean
		Total_ride_time
day_of_week	member_casual	
Sunday	casual	27.634102
	member	14.124693
Monday	casual	25.199399
	member	12.244926
Tuesday	casual	21.753495
	member	11.992901
Wednesday	casual	21.017983
	member	12.056473
Thursday	casual	21.711755
	member	12.243782
Friday	casual	22.685493
	member	12.454164
Saturday	casual	27.170151
	member	14.258908

Location based Analysis

Most used Stations by Customer Type

In [29]:
```
# What are top 5 most used stations by customer types?

df.groupby(['member_casual', 'start_station_name'])[['ride_id']].count().sort_
```

Out[29]:

		ride_id
member_casual	**start_station_name**	
member	**Kingsbury St & Kinzie St**	23142
	Clark St & Elm St	20198
	Wells St & Concord Ln	19370
	Clinton St & Washington Blvd	18415
	Loomis St & Lexington St	17836
casual	**Streeter Dr & Grand Ave**	54176
	DuSable Lake Shore Dr & Monroe St	29803
	Millennium Park	23586
	Michigan Ave & Oak St	23419
	DuSable Lake Shore Dr & North Blvd	21841

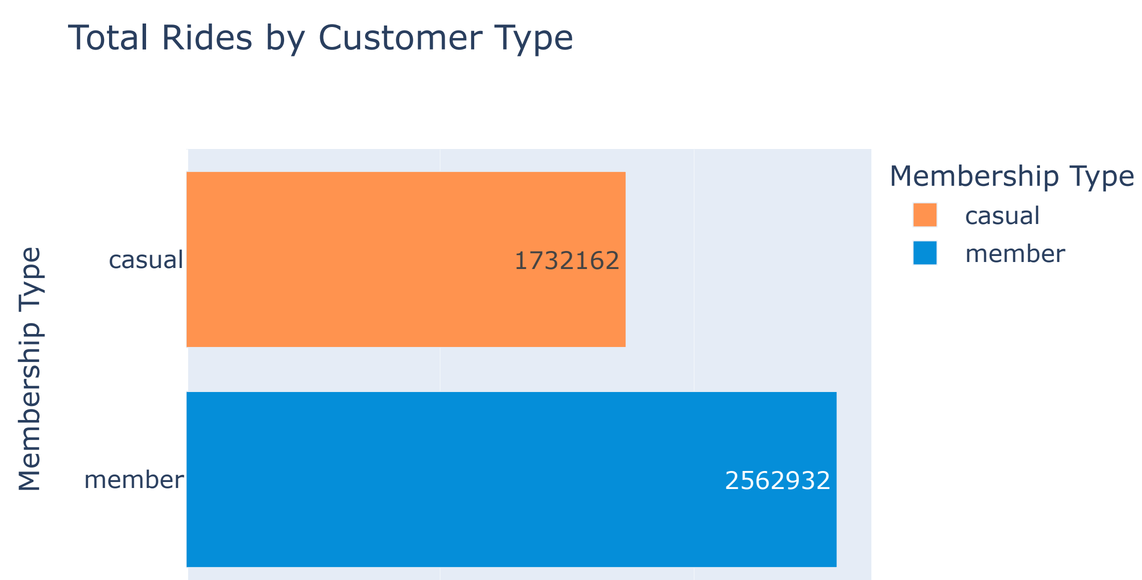
SHARE

Visualizing Total Rides by Customer Type

In [30]:
```
fig_1 = df.groupby('member_casual', as_index =False)[['ride_id']].count()

fig = px.bar(fig_1, x='ride_id', y='member_casual',
             color='member_casual',
             title='Total Rides by Customer Type',
             text= 'ride_id',
             labels={'ride_id':'No. of Rides', 'member_casual': 'Membership Ty
             hover_name='member_casual',
             hover_data={'member_casual':False, 'ride_id':True},
             color_discrete_map = {'casual': '#FF934F', 'member': '#058ED9'},
             height=400,

            )
fig.show()
```



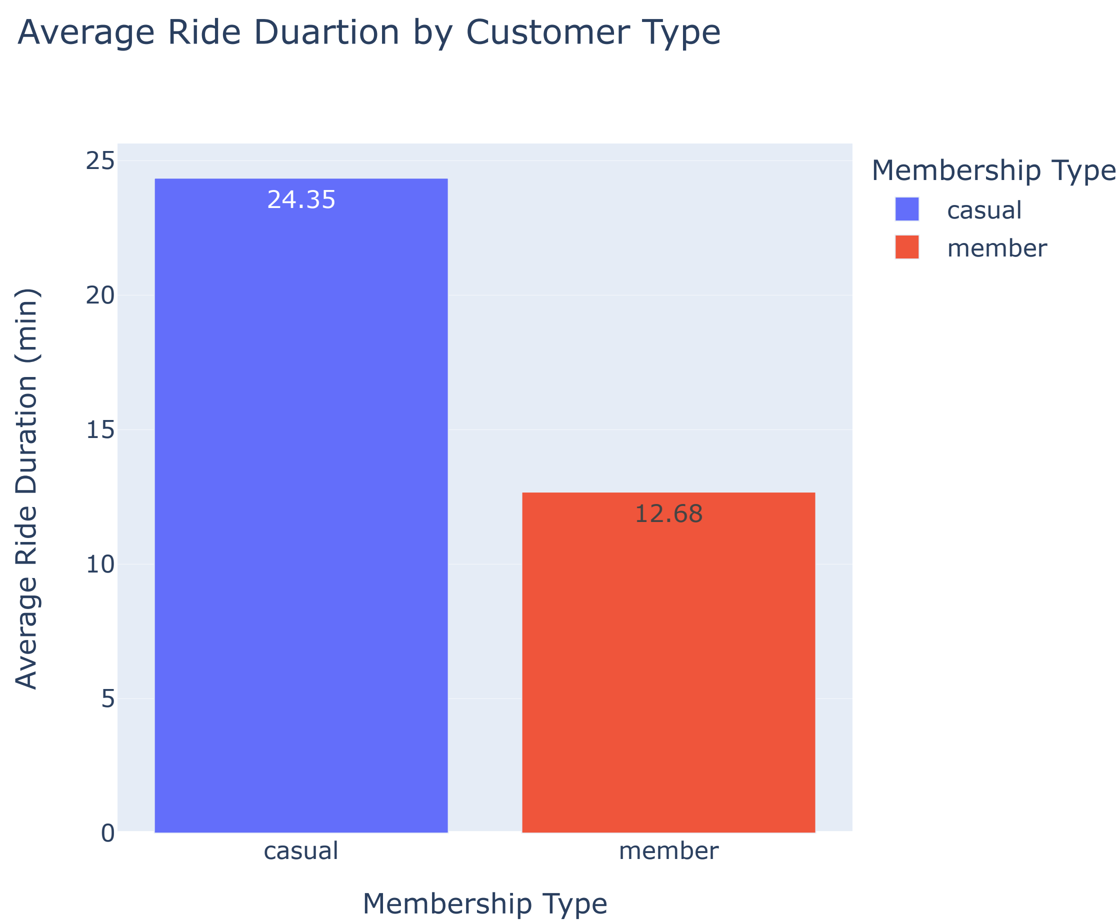
This bar chart shows that around 40% of users are casual riders and 60% are member riders.

Average Ride Duration by Customer Type

In [31]:
```
fig_2 = df.groupby('member_casual', as_index =False)[['Total_ride_time']].mear

fig = px.bar(fig_2, x='member_casual', y='Total_ride_time',
             color='member_casual',
             text='Total_ride_time',
             title='Average Ride Duartion by Customer Type',
             labels={'member_casual':'Membership Type', 'Total_ride_time':'Ave
            )

fig.show()
```

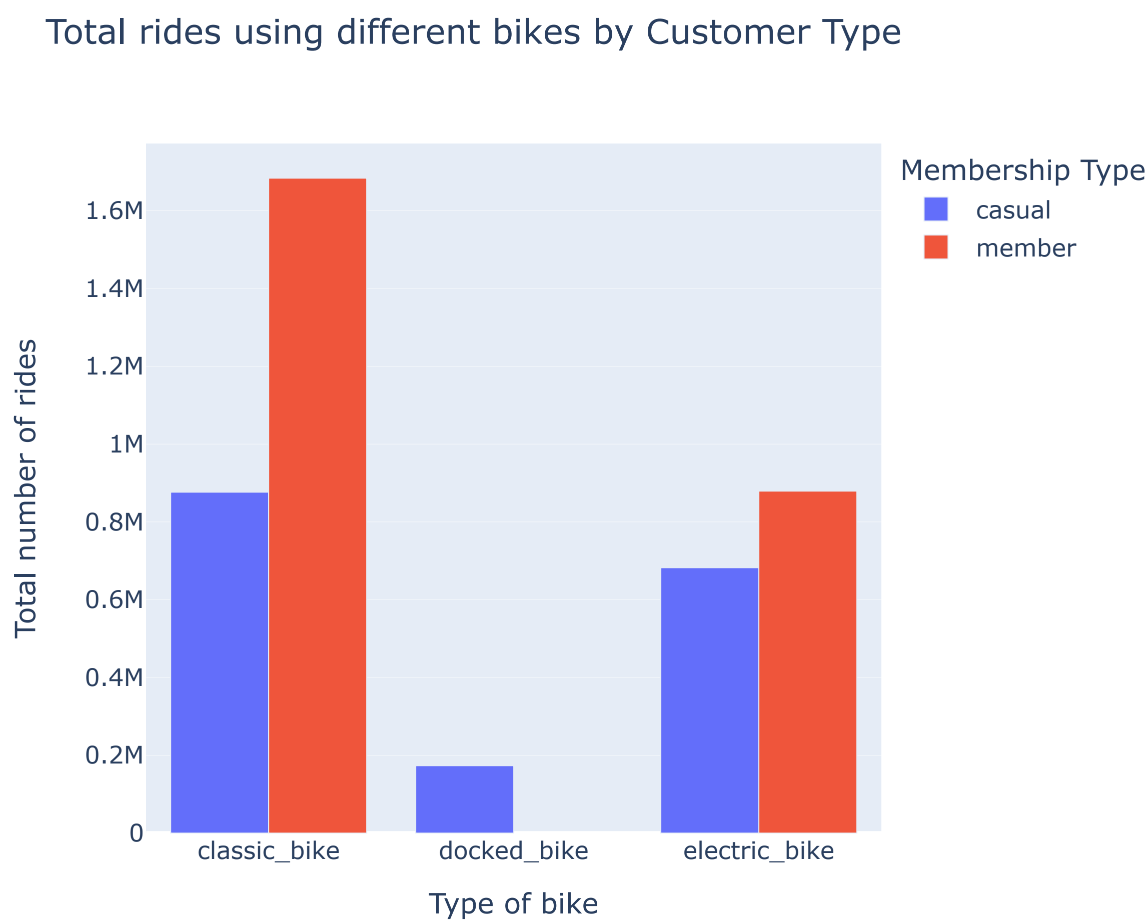


The above column chart shows that casual riders have much greater Average Ride Duration than member riders. On average Casual riders ride for 24 minutes and member riders ride for 13 minutes which is almost half of casual riders.

Total rides in different bikes by Customer Type

In [32]:
```
fig_3 = df.groupby(['member_casual', 'rideable_type'], as_index=False)[['ride_
fig = px.bar(fig_3, x='rideable_type', y='ride_id', color='member_casual', bar
             title='Total rides using different bikes by Customer Type',
             labels={'ride_id':'Total number of rides', 'rideable_type':'Type
            )

fig.show()
```



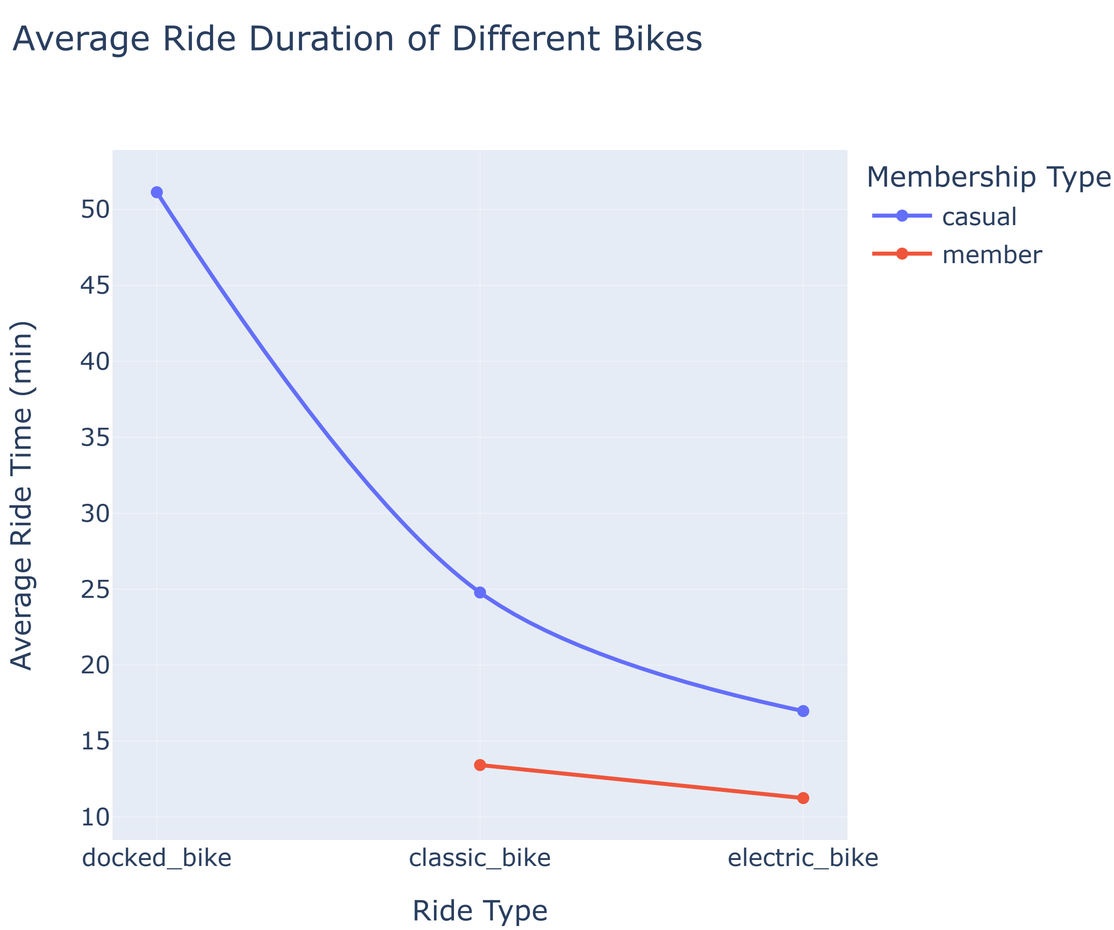
There are there types of bike: classic bike, docked bike, electric bike. Classic bike is the most rented bike followed by electric bike. Docked bike are not very popular choice among riders.

Average Ride Duration of different bikes by Customer Type

In [33]:
```
fig_4 = df.groupby(['member_casual', 'rideable_type'], as_index=False)[['Total

fig = px.line(fig_4, x='rideable_type', y='Total_ride_time',
              color='member_casual',
              line_shape = 'spline',
              markers=True,
              labels={'rideable_type': 'Ride Type', 'Total_ride_time': 'Averag
              title='Average Ride Duration of Different Bikes'

             )
fig.show()
```



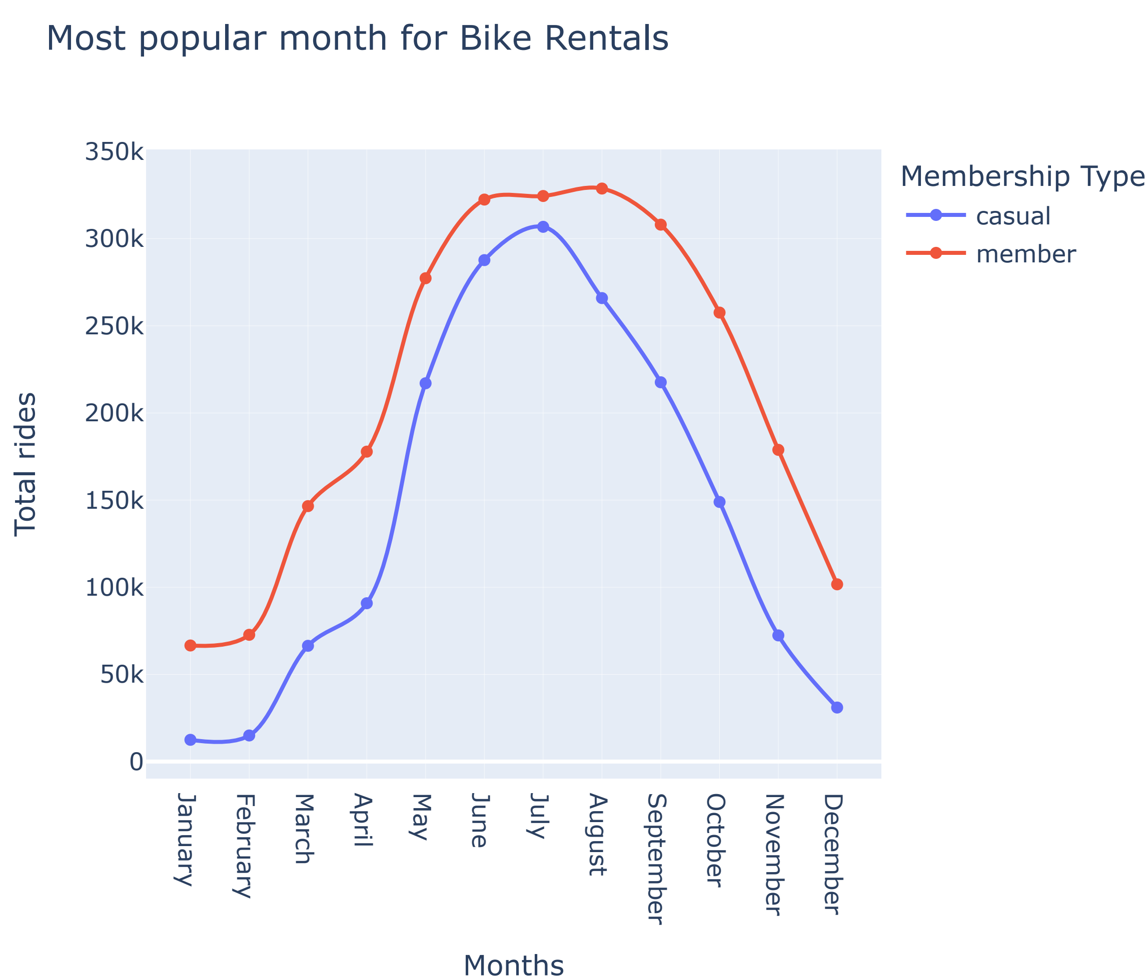
In general casual riders have higher Average Ride Time. When it comes to bike type classic bike have higher Average Ride Time than electric bike.

Most popular month for Bike Rental

In [34]:
```
fig_5 = df.groupby(['month_name', 'member_casual'], as_index=False)[['ride_id'

fig = px.line(fig_5, x='month_name', y='ride_id', color='member_casual',
              title='Most popular month for Bike Rentals',
              line_shape='spline',
              markers=True,
              labels={'month_name':'Months', 'member_casual':'Membership Type'
              )

fig.show()
```



Observation:

The number of rentals increases during summer months and gradually drops during winter months for both Membership Type.

Although, both member and casual riders have same riding pattern over the year. Member riders are always more than casual riders if month wise data is compared.

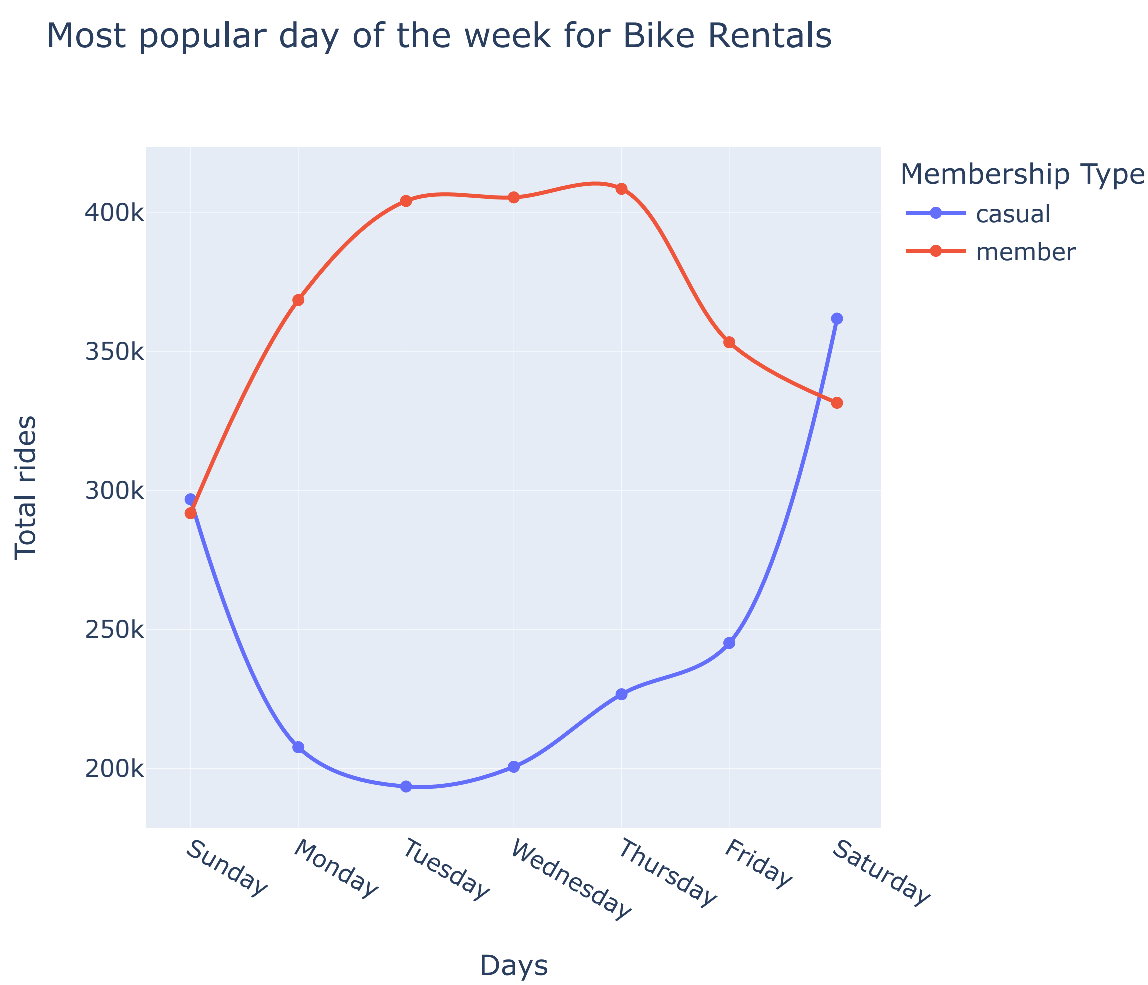
Conclusion: Weather in Summer months is more suitable for bike ride. While in winter it is too cold to ride a bike. Hence Total riders in winters are less as compare to summers.

Most popular Day of the week for bike Rentals

In [35]:
```
fig_6 = df.groupby(['day_of_week', 'member_casual'], as_index=False)[['ride_id

fig = px.line(fig_6, x='day_of_week', y='ride_id', color='member_casual',
              title='Most popular day of the week for Bike Rentals',
              line_shape='spline',
              markers=True,
              labels={'day_of_week':'Days', 'member_casual':'Membership Type',
              )

fig.show()
```



Observation:

Casual rider are more active on weekend and member riders are more active on weekdays. Tuesday is among most popular days for member riders and least popular for

casual riders.

Saturday is the only day when casaul rider are significantly more than member riders. And, on Sunday both riders are equal in number.

Conclusion:

Member riders use bike to go to their work. So their activity increases during the weekdays. Casual riders are using bike for leisure activites. Hence, their activities are more on weekends.

Also although, activities of member riders decreases significantly on weekends. It is still comparable to casual rides. So, member riders might also be going out for some leisure activities or maybe groceries shopping and other activities.

Most Popular Hour of the day for Bike Rentals

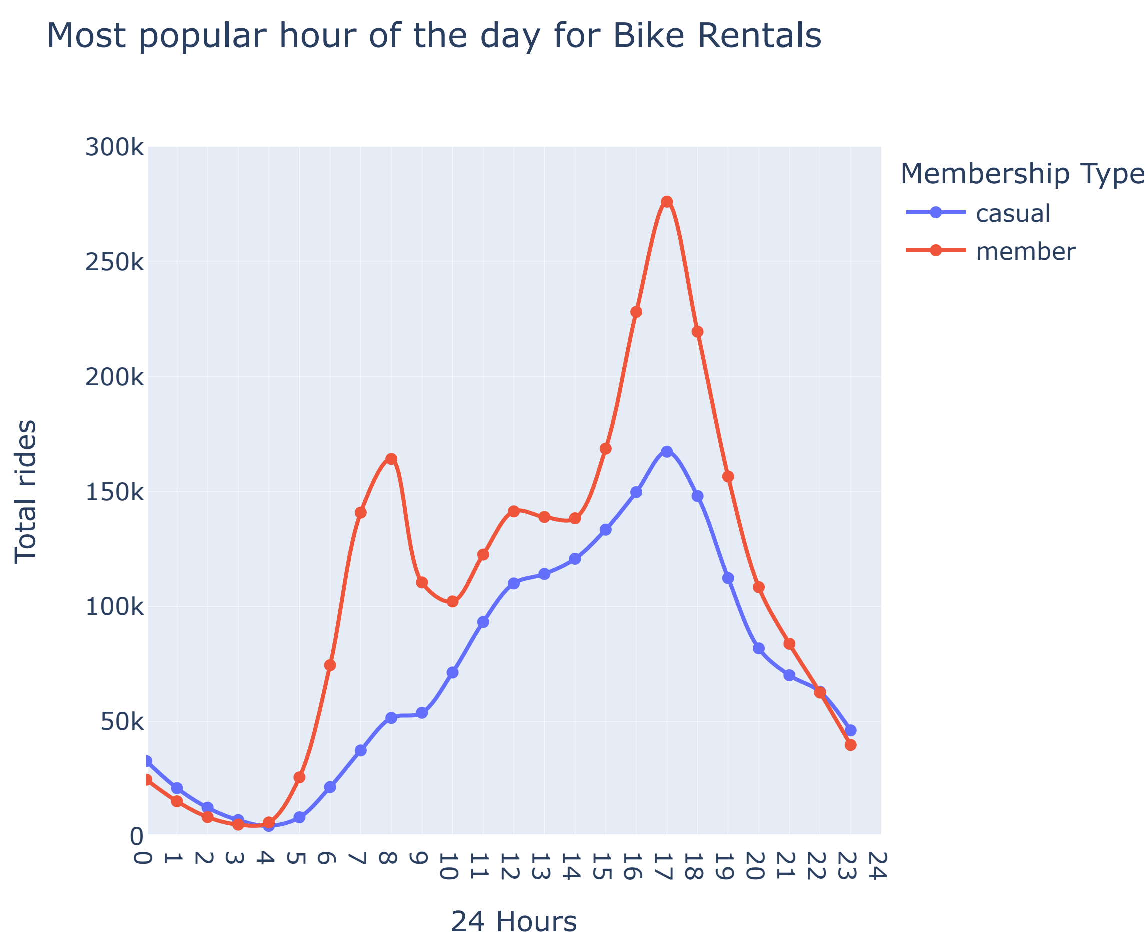
In [36]:
```
fig_7 = df.groupby(['hour', 'member_casual'], as_index=False)[['ride_id']].cou

fig = px.line(fig_7, x='hour', y='ride_id', color='member_casual',
              range_x=[0,24], range_y=[0,300000],

              title='Most popular hour of the day for Bike Rentals',
              line_shape='spline',
              markers=True,
              labels={'hour':'24 Hours', 'member_casual':'Membership Type', 'r
              )

fig.update_xaxes(dtick=1)

fig.show()
```



Observation:

Both casual and member riders are most active around 5 PM in the evening. Also, member rider are very active around 8 AM.

Conclusion:

Around 5 PM in the evening member riders are coming back from office and casual riders might be going out for some leisure activity or to meet their friends after their work hours. Hence, Total riders are very high around this time.

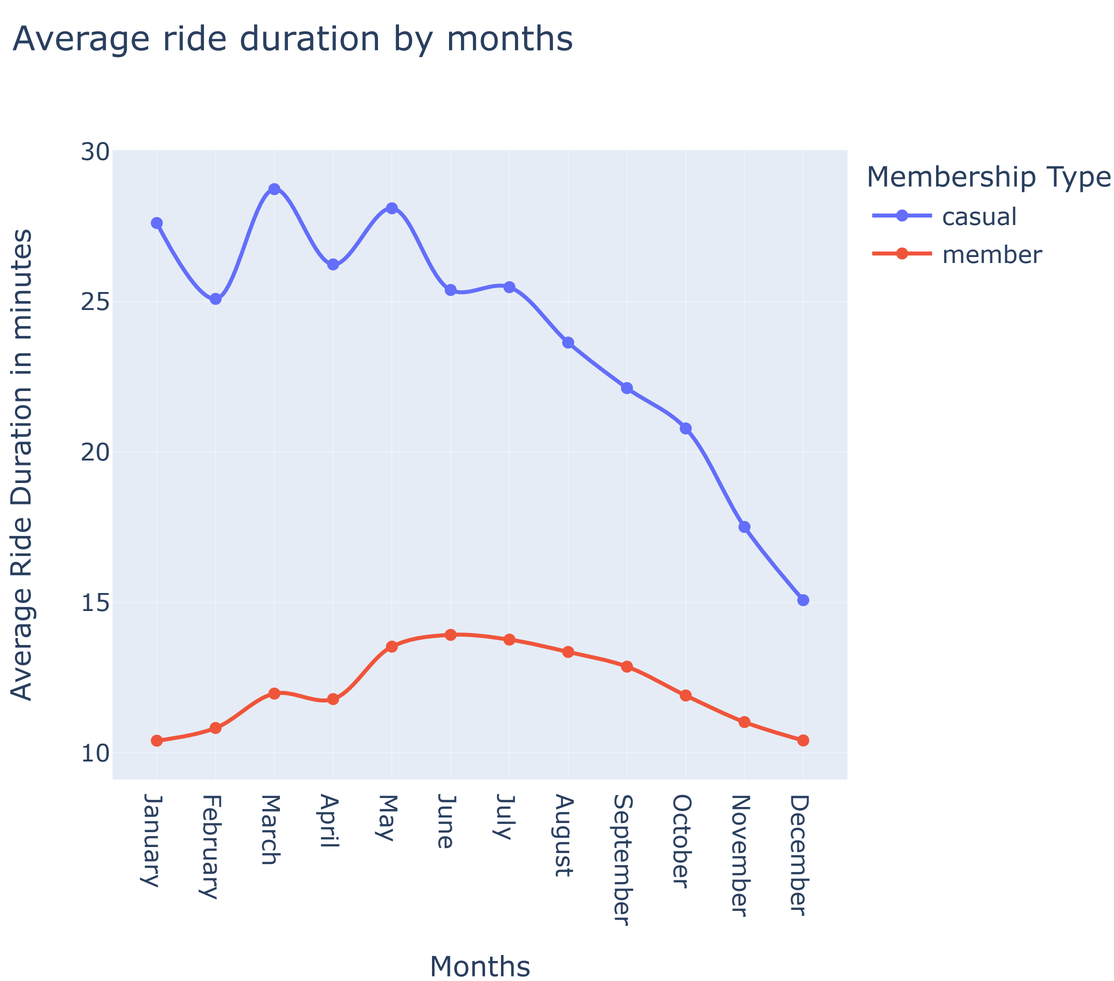
Also arond 8 AM in the morning, it's time to go work or school. Hence, number of total rides around this time is high for member riders who primarily use bikes to go to work.

Average ride duration by month

In [37]:
```
fig_8 = df.groupby(['month_name', 'member_casual'], as_index=False)[['Total_ri

fig = px.line(fig_8, x='month_name', y='Total_ride_time', color='member_casual
              title='Average ride duration by months',
              line_shape='spline',
              markers=True,
              labels={'month_name':'Months', 'member_casual':'Membership Type'
              )

fig.show()
```



Observation:

For casual riders Average Ride Duration is really high for summer months and then gradually started decreasing from August. Also For some winter months it is very high.

But for member riders Average Ride Duration is not changing drastically. However it is slightly higher during summer months as well.

Conclusion:

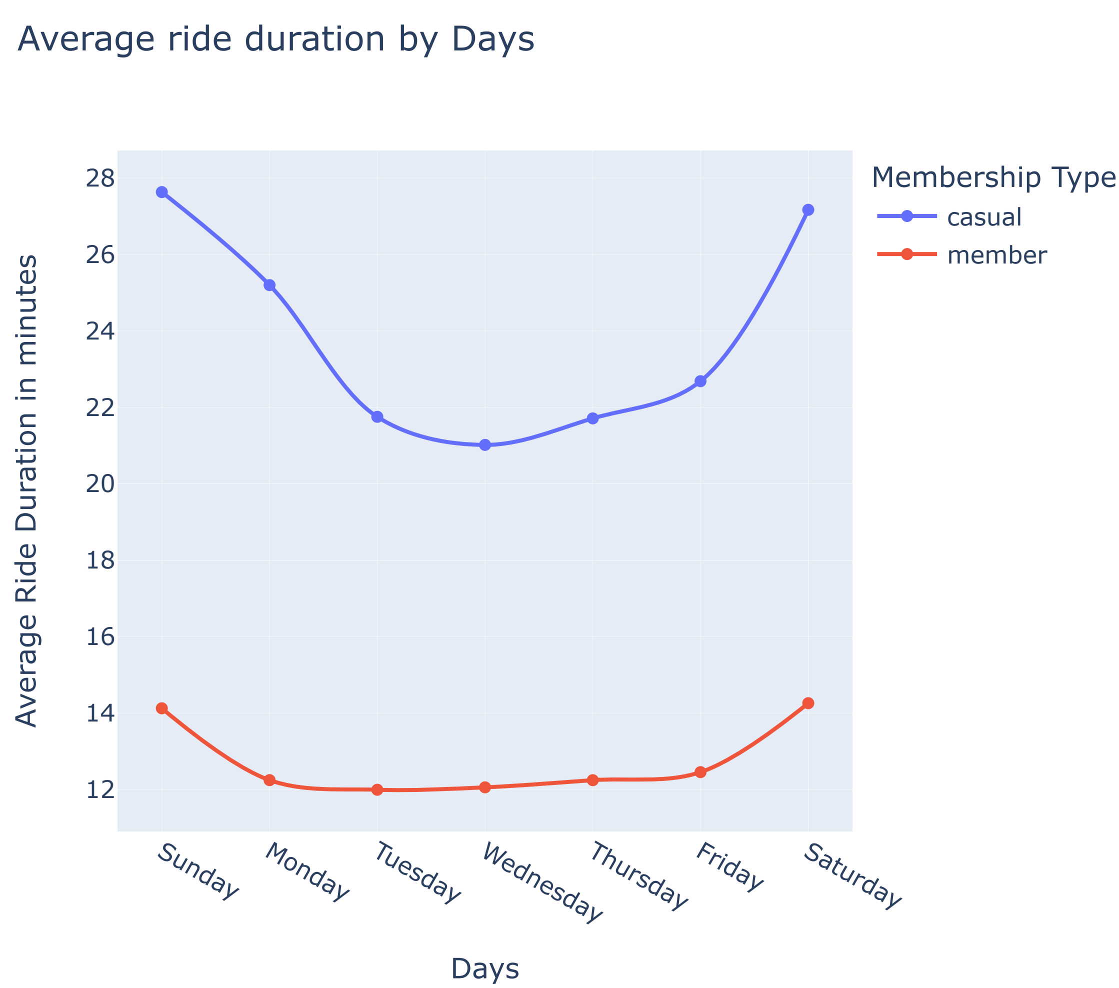
During some winter months, Average Ride Duration for Casual Member is high because there might be too much snow, bad weather and rider was not able to return their bikes for couple of weeks

Average ride duration by Day of the week

In [38]:
```
fig_9 = df.groupby(['day_of_week', 'member_casual'], as_index=False)[['Total_r

fig = px.line(fig_9, x='day_of_week', y='Total_ride_time', color='member_casua
              title='Average ride duration by Days',
              line_shape='spline',
              markers=True,
              labels={'day_of_week':'Days', 'member_casual':'Membership Type',
              )

fig.show()
```



Observation:

Average Ride duration pattern is almost same for both casual and member riders that it is higher during the weekends. Casual riders have very high Average Ride Duration, averageing 24 minutes as compares to Member riders with 12 minutes.

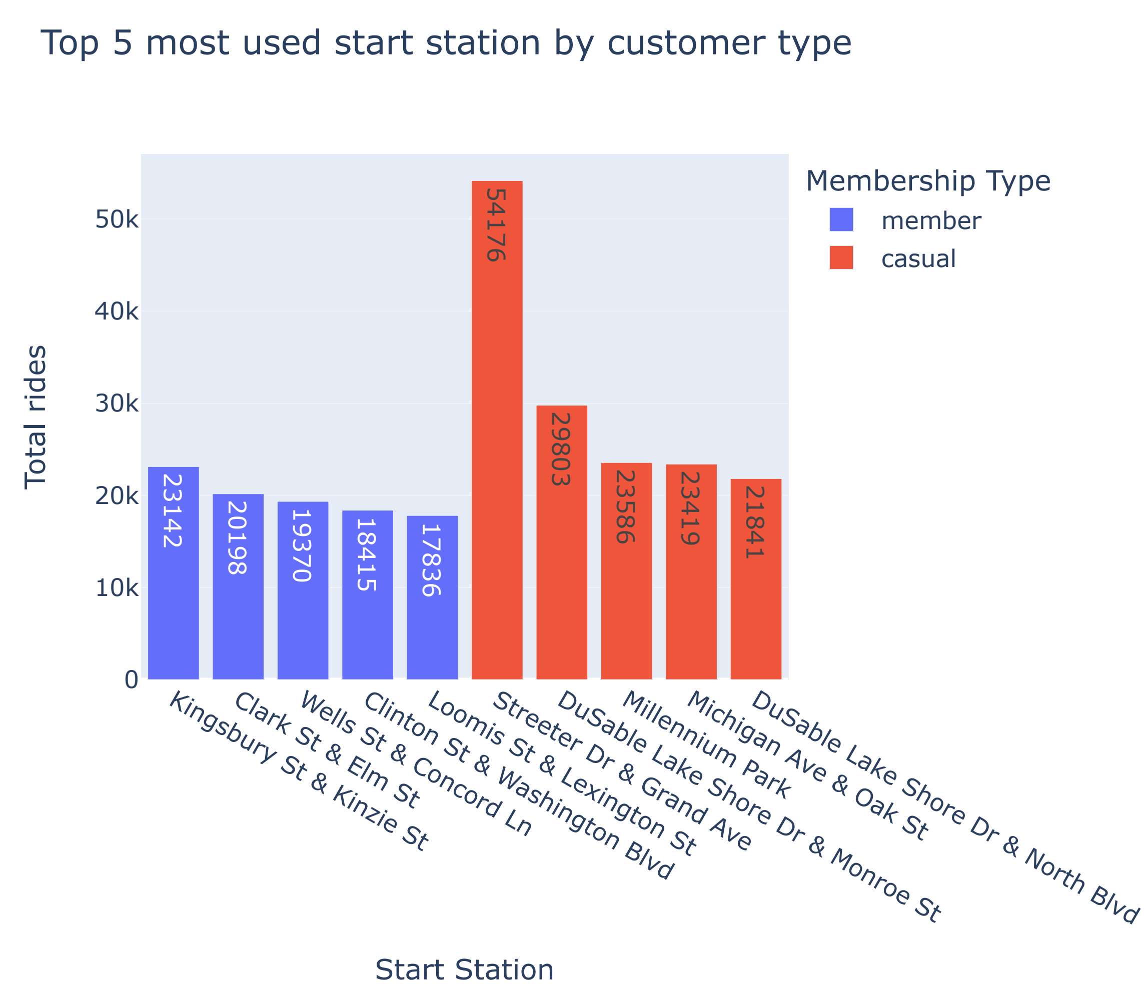
Conclusion: Such difference in Average ride duration is likely because member riders live close to their work or school. Casual riders are using it for leisure or tourism.

Most used station by Customer type

In [39]:
```
fig_9 = df.groupby(['member_casual', 'start_station_name'], as_index=False)[['

fig = px.bar(fig_9, x='start_station_name', y='ride_id', color='member_casual'
             text='ride_id',
             title='Top 5 most used start station by customer type',
             labels={'member_casual':'Membership Type', 'start_station_name':

            )
fig.show()
```



Observation: Above chart show the top 5 start station for members and casual riders. For casual riders 'Kingsbury St and Kinzie St' is the most popular while for member riders 'Streeter Dr & Grand Ave' is the most popular one.

Conclusion: These top stations are very good spots to advertise and market Cyclistic's products.

ACT

Now its time to summerize all the key findings and provide recommendations for cyclistic's marketing team.

Key Findings

Casual riders comprise a substantial portion, accounting for 40% of Cycistic's customer base.

Casual riders enjoy an average ride duration of 24 minutes, whereas members tend to take shorter trips, averaging around 13 minutes.

The majority of Cycistic's members opt for the Classic Bike, which boasts a rider count that surpasses all other categories by more than double. Following closely behind is the Electric Bike category. In terms of casual riders, there is a nearly equal distribution between Classic and Electric bikes, with the Classic Bike slightly edging ahead.

Both member and casual users experience longer average ride durations when utilizing the Classic Bike and Electric Bike options.

Summer months June, July are the most popular month for bike rental.

Casual rider are more active on weekend and member riders are more active on weekdays.

Both casual and member riders are most active around 5 PM in the evening. Also, member rider are very active around 8 AM.

During some winter months, Average Ride Duration for Casual Member is high because there might be too much snow, bad weather and rider was not able to return their bikes for couple of weeks.

Weekends has higher Average Ride duration for both Casual and Member riders while overall Average of Casual is 24 minutes which is double than average of Member riders that is 12 minutes.

How casual riders and members use Cyclistic bikes differently?

Casual riders demonstrate a tendency to rent a larger number of bikes during the weekends. With their extended average ride durations, these factors combined indicate that this particular customer group utilizes the bikes primarily for leisure and recreational purposes.

Recommendations

The following recommendations are made to help guide Cyclistic's marketing Strategy.

To effectively reach out to casual riders during the summer, when bike rentals are in higher demand, it is recommended to intensify the marketing campaign through targeted channels such as email, text messages, and app notifications. The advertisements can include enticing promotions and exclusive discounts for annual memberships, aiming to incentivize casual riders to transition into becoming members.

Cyclistic has the opportunity to explore collaborations with local businesses and create exclusive partnerships that offer attractive deals and discounts at establishments such as fitness centers, movie theaters, beauty salons, and restaurants. By concentrating on the entertainment and wellness sectors, Cyclistic can effectively motivate casual riders to consider enrolling in the annual membership program.

Implementing a rewards points system for members is a viable strategy, whereby each bike rental contributes a designated number of points to the user's account. The points accumulation could be influenced by factors such as rental frequency and trip duration, with higher values awarded for more frequent and longer rides. At the end of each year, users would have the option to redeem their accumulated points for a discounted membership renewal, providing them with an additional incentive to remain loyal members.

To cater to the rental patterns of casual riders, Cyclistic's marketing team should consider introducing two new membership options. Firstly, an annual weekend-only membership could be offered, allowing customers who rent bikes exclusively on weekends to pay accordingly while contributing to the company's growing membership base. Secondly, a seasonal membership tailored for summer riders, billed annually and at a discounted rate compared to single-ride or full-day passes, would cater to those who prefer biking during the summer months without interest in year-round rentals. These membership variations would provide flexible options and attract a wider range of customers while supporting Cyclistic's business growth.

Additional Consideration

Due to the gaps in our data, including missing payment, age, gender, and geographical information, we currently face limitations in providing further recommendations. Demographic data plays a crucial role in developing effective marketing campaigns as it enables a deeper understanding of the target audience. By obtaining this information, businesses can segment customers based on their habits, interests, and other important characteristics. This segmentation allows the marketing team to create personalized ad campaigns that align with Cyclistic's marketing and financial objectives, ultimately maximizing their effectiveness.