Company name : Cyclistic (fictional)

Type of company : Bike-share company

Context : *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.*

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

In [ ]:

In [1]:

```
import os
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

Combine files

Folder have 12 csv files.

These files have data of 2023.

Combine files to analyse 12 months data.

In [2]:

```
# files = [file for file in os.listdir('./csv_files_202301_202312')]

# all_months_data = pd.DataFrame()

# for file in files:
#     current_data = pd.read_csv('./csv_files_202301_202312/'+file)
#     all_months_data = pd.concat([all_months_data, current_data])

# all_months_data.to_csv('2023tripdata.csv')
```

In [3]:

```
all_months_data = pd.read_csv('2023tripdata.csv', index_col=0)
```

In [ ]:

Gather information about data

In [ ]:

In [4]:

```
all_months_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5719877 entries, 0 to 224072
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: 610.9+ MB
```

In [5]:

```
all_months_data.head()
```

Out[5]:

	ride_id	rideable_type	started_at	ended_at	start_station_name	start_sta
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	Lincoln Ave & Fullerton Ave	TA1309
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	Kimbark Ave & 53rd St	TA1309
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	Western Ave & Lunt Ave	
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	Kimbark Ave & 53rd St	TA1309
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	Kimbark Ave & 53rd St	TA1309

In [6]:

```
rows, columns = all_months_data.shape
print(f'Rows: {rows}\nColumns: {columns}')
```

```
Rows: 5719877
Columns: 13
```

In [7]:

```
all_months_data.isnull().sum()
```

Out[7]:

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

In [ ]:

Delete unnecessary columns

Here we are not going to look at the station data.

In [ ]:

In [8]:

```
all_months_data.drop(['start_station_name','start_station_id','end_station_name'
```

In [9]:

```
all_months_data.isnull().sum()
```

Out[9]:

```
ride_id              0
rideable_type        0
started_at           0
ended_at             0
start_lat            0
start_lng            0
end_lat           6990
end_lng           6990
member_casual        0
dtype: int64
```

In [10]:

```
all_months_data = all_months_data.rename(columns={'member_casual': 'user_type'})
```

Changed datatype

Column having date is of type object.

Change column type to datetime.

Create new column for the ride length.

In [ ]:

In [11]:

```
all_months_data['started_at'] = pd.to_datetime(all_months_data['started_at'])
all_months_data['ended_at'] = pd.to_datetime(all_months_data['ended_at'])
all_months_data['ride_len'] = all_months_data['ended_at'] - all_months_data['sta
all_months_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5719877 entries, 0 to 224072
Data columns (total 10 columns):
 #   Column         Dtype
---  ------         -----
 0   ride_id        object
 1   rideable_type  object
 2   started_at     datetime64[ns]
 3   ended_at       datetime64[ns]
 4   start_lat      float64
 5   start_lng      float64
 6   end_lat        float64
 7   end_lng        float64
 8   user_type      object
 9   ride_len       timedelta64[ns]
dtypes: datetime64[ns](2), float64(4), object(3), timedelta64[ns](1)
memory usage: 480.0+ MB
```

Deleting rows

Delete rows having ride length less than 0 min.

In [12]:

```
all_months_data['ride_len_min'] = round(all_months_data['ride_len'].dt.total_sec
all_months_data = all_months_data[all_months_data['ride_len_min'] > 0]
all_months_data.head()
```

Out[12]:

	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng	en
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	41.924074	-87.646278	41.93
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	41.799568	-87.594747	41.80
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	42.008571	-87.690483	42.03
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	41.799568	-87.594747	41.80
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	41.799568	-87.594747	41.80

In [ ]:

Analyse Data

Check ride length data for outliers.

In [ ]:

In [13]:
```
all_months_data['ride_len'].describe()
```

Out[13]:
```
count                        5621879
mean      0 days 00:18:29.871122270
std       0 days 03:01:45.462095704
min                 0 days 00:00:31
25%                 0 days 00:05:36
50%                 0 days 00:09:42
75%                 0 days 00:17:07
max                68 days 09:29:04
Name: ride_len, dtype: object
```

In [14]:
```
all_months_data[(all_months_data['ride_len'] > '0 days 4:00:00') & (all_months_d
```

Out[14]:
```
ride_id          4360
rideable_type    4360
started_at       4360
ended_at         4360
start_lat        4360
start_lng        4360
end_lat          3246
end_lng          3246
user_type        4360
ride_len         4360
ride_len_min     4360
dtype: int64
```

In [15]:
```
all_months_data[(all_months_data['ride_len'] > '0 days 4:00:00') & (all_months_d
```

Out[15]:
```
ride_id          11522
rideable_type    11522
started_at       11522
ended_at         11522
start_lat        11522
start_lng        11522
end_lat           6019
end_lng           6019
user_type        11522
ride_len         11522
ride_len_min     11522
dtype: int64
```

In [16]:
```
all_months_data['rideable_type'].unique()
```

Out[16]:
```
array(['electric_bike', 'classic_bike', 'docked_bike'], dtype=object)
```

In [ ]:

Create Columns

Create columns for DAY, MONTH, HOUR, DAY NAME

These will be used for analysing rider behaviour

In [ ]:

In [17]:

```
all_months_data['started_at_day'] = all_months_data['started_at'].dt.day
all_months_data['started_at_month'] = all_months_data['started_at'].dt.month
all_months_data['started_at_hour'] = all_months_data['started_at'].dt.hour
all_months_data['started_at_dayname'] = all_months_data['started_at'].dt.day_nam
```

In [18]:

```
all_months_data.head()
```

Out[18]:

	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng	en
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	41.924074	-87.646278	41.93
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	41.799568	-87.594747	41.80
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	42.008571	-87.690483	42.03
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	41.799568	-87.594747	41.80
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	41.799568	-87.594747	41.80

In [ ]:

Total number of rides per month

In [ ]:

In [19]:

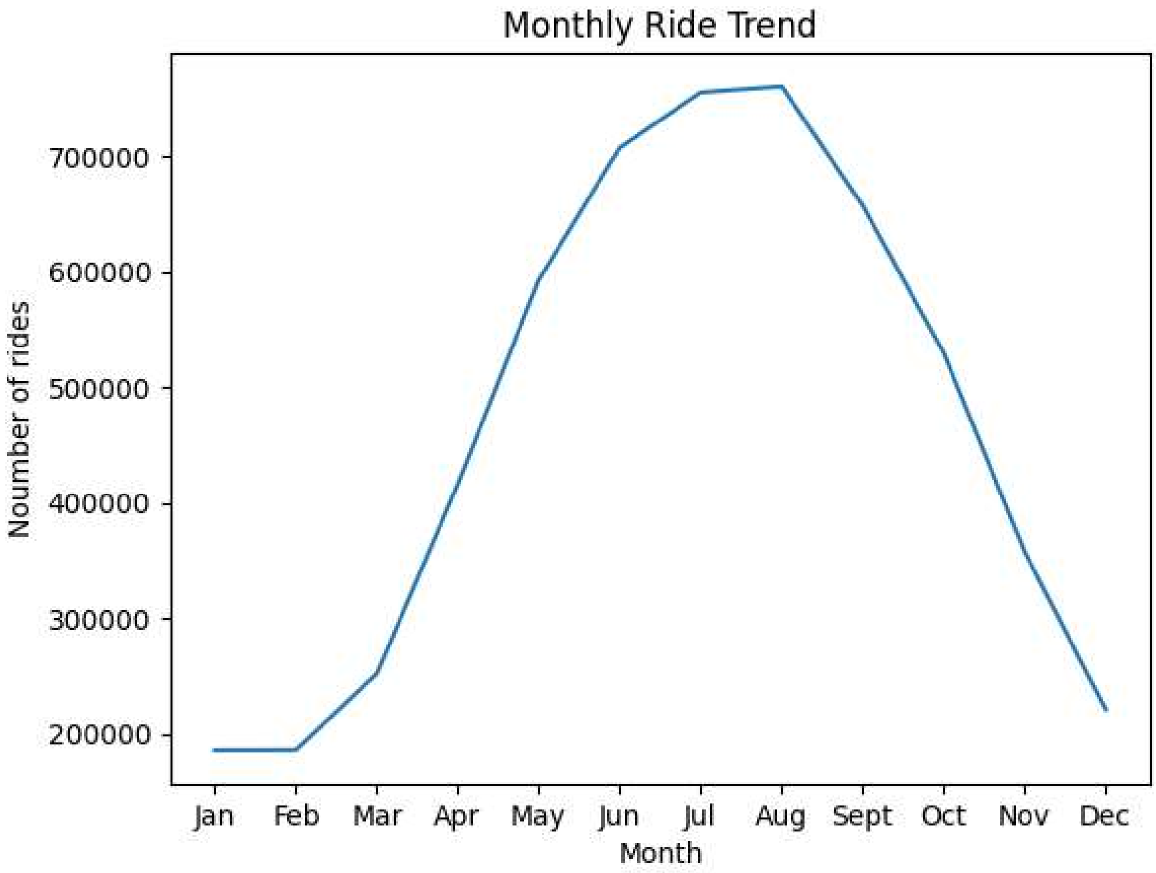
```
rides_per_month = all_months_data.groupby('started_at_month').count()['ride_id']
rides_per_month
```

Out[19]:

```
started_at_month
1     186020
2     186192
3     252247
4     416239
5     592901
6     707260
7     754878
8     760023
9     657215
10    529761
11    357800
12    221343
Name: ride_id, dtype: int64
```

In [20]:

```
plt.plot(rides_per_month)
plt.xlabel('Month')
plt.xticks(np.arange(1,13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'A
plt.ylabel('Noumber of rides')
plt.title('Monthly Ride Trend')
plt.xticks(np.arange(1,13))
plt.show()
```



```
In [ ]:
```

Total number of rides taken by casual riders and members per month

```
In [ ]:
```

```
In [21]: rides_per_month_by_member = all_months_data.groupby(['started_at_month','user_ty
         rides_per_month_by_member
```

```
Out[21]: started_at_month  user_type
         1                 casual        39236
                           member       146784
         2                 casual        42204
                           member       143988
         3                 casual        60887
                           member       191360
         4                 casual       144132
                           member       272107
         5                 casual       229817
                           member       363084
         6                 casual       295977
                           member       411283
         7                 casual       326009
                           member       428869
         8                 casual       306580
                           member       453443
         9                 casual       257933
                           member       399282
         10                casual       174511
                           member       355250
         11                casual        97035
                           member       260765
         12                casual        51004
                           member       170339
         Name: ride_id, dtype: int64
```

```
In [22]: rides_per_month_by_member = rides_per_month_by_member.reset_index()
         rides_per_month_by_member
```

Out[22]:

	started_at_month	user_type	ride_id
0	1	casual	39236
1	1	member	146784
2	2	casual	42204
3	2	member	143988
4	3	casual	60887
5	3	member	191360
6	4	casual	144132
7	4	member	272107
8	5	casual	229817
9	5	member	363084
10	6	casual	295977
11	6	member	411283
12	7	casual	326009
13	7	member	428869
14	8	casual	306580
15	8	member	453443
16	9	casual	257933
17	9	member	399282
18	10	casual	174511
19	10	member	355250
20	11	casual	97035
21	11	member	260765
22	12	casual	51004
23	12	member	170339

In [23]:

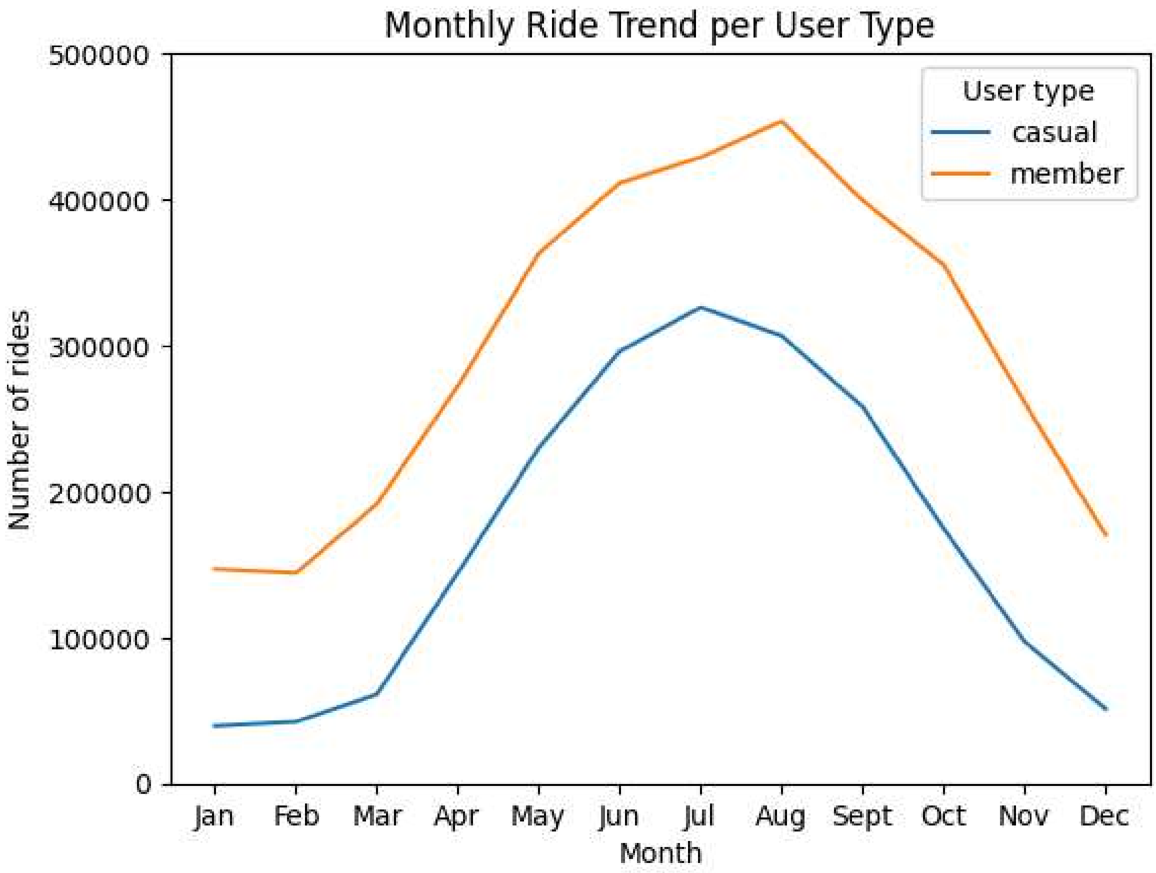
```
pivot_table = rides_per_month_by_member.pivot(index='started_at_month', columns=
pivot_table
```

Out[23]:

user_type	casual	member
started_at_month		
1	39236	146784
2	42204	143988
3	60887	191360
4	144132	272107
5	229817	363084
6	295977	411283
7	326009	428869
8	306580	453443
9	257933	399282
10	174511	355250
11	97035	260765
12	51004	170339

In [24]:

```
pivot_table.plot(kind='line')
plt.xlabel('Month')
plt.ylabel('Number of rides')
plt.title('Monthly Ride Trend per User Type')
plt.yticks([i for i in range(0,600000,100000)])
plt.xticks(np.arange(1,13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'A
plt.legend(title='User type')
plt.show()
```



In [ ]:

Making weekday a categorical data to maintain order.

In [ ]:

In [25]:

```
weekday_order = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturd
all_months_data['started_at_dayname'] = pd.Categorical(all_months_data['started_
all_months_data.head()
```

Out[25]:

	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng	en
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	41.924074	-87.646278	41.93
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	41.799568	-87.594747	41.80
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	42.008571	-87.690483	42.03
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	41.799568	-87.594747	41.80
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	41.799568	-87.594747	41.80

In [ ]:

Total number of rides taken by casual and members per weekday

In [ ]:

In [26]:

```
rides_per_week = all_months_data.groupby(['started_at_dayname','user_type']).cou
rides_per_week = rides_per_week.reset_index()
pivot_table = rides_per_week.pivot(index='started_at_dayname' , columns='user_ty
pivot_table
```

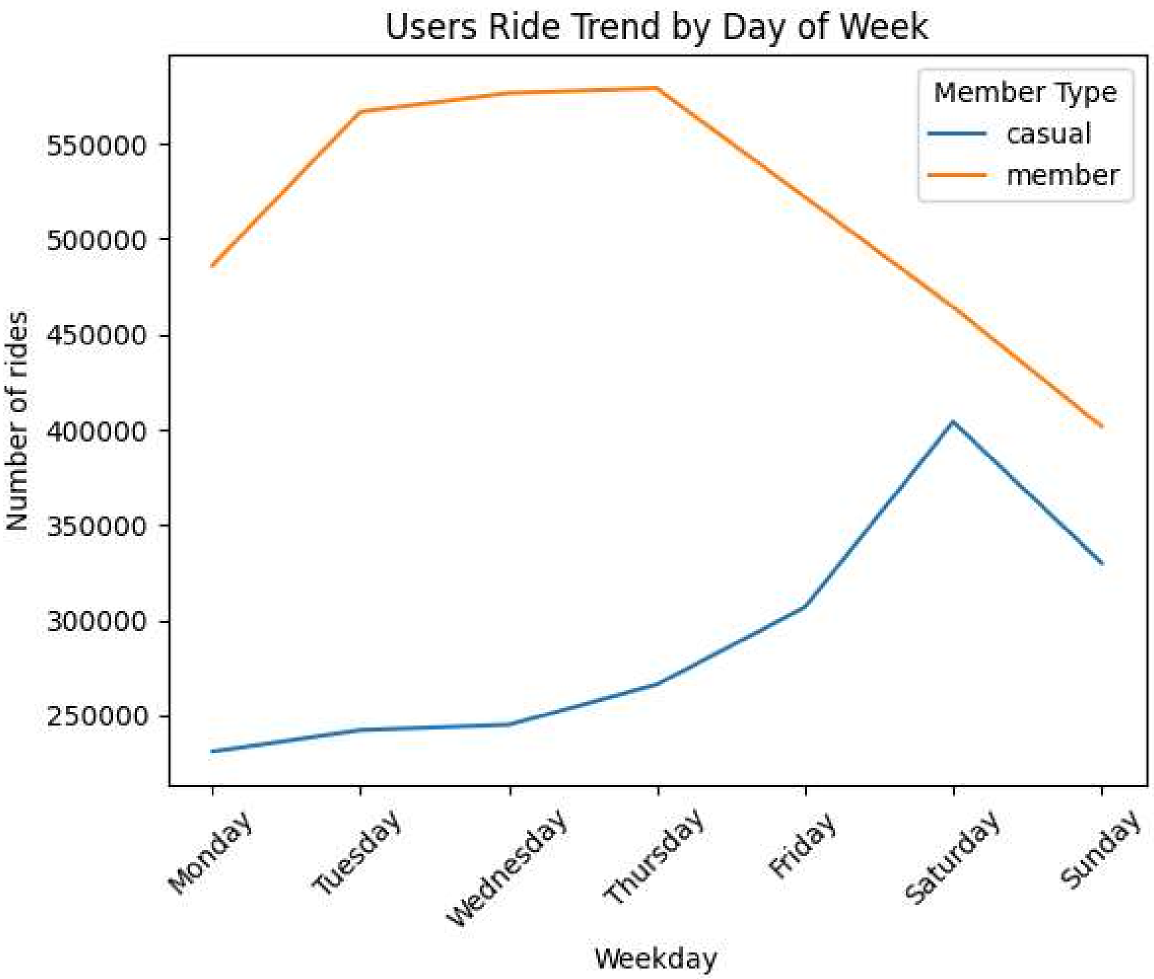
```
C:\Users\prati\AppData\Local\Temp\ipykernel_33116\3086334522.py:1: FutureWarning:
The default of observed=False is deprecated and will be changed to True in a futu
re version of pandas. Pass observed=False to retain current behavior or observed=
True to adopt the future default and silence this warning.
  rides_per_week = all_months_data.groupby(['started_at_dayname','user_type']).co
unt()['ride_id']
```

Out[26]:

user_type	casual	member
started_at_dayname		
Monday	231014	486069
Tuesday	242212	566907
Wednesday	245031	576550
Thursday	266154	579072
Friday	306744	521937
Saturday	404019	464251
Sunday	330151	401768

In [27]:

```
pivot_table.plot(kind='line')
plt.xlabel('Weekday')
plt.ylabel('Number of rides')
plt.title('Users Ride Trend by Day of Week')
plt.legend(title='Member Type')
plt.xticks(rotation=45)
plt.show()
```



In [ ]:

Total number of rides taken by casual and members using different types of cycle

In [ ]:

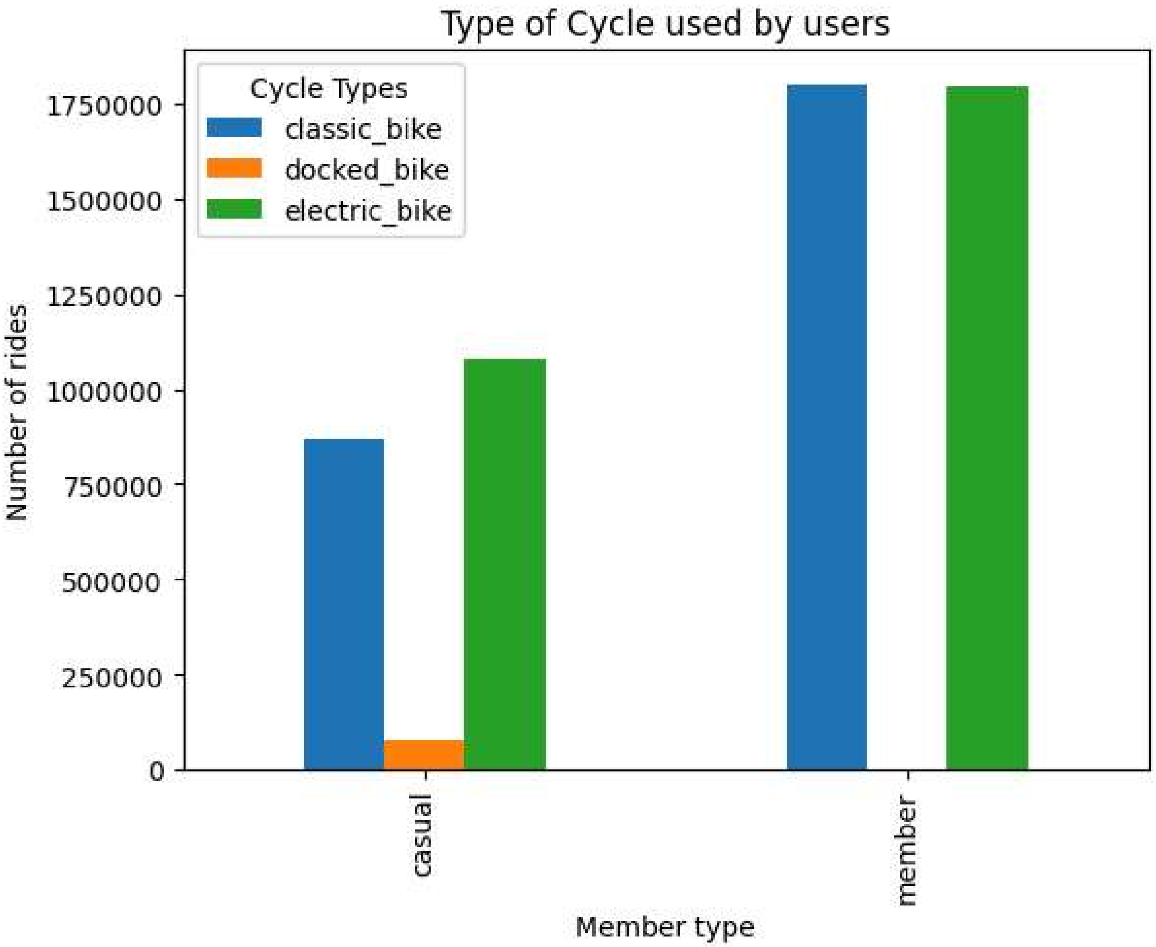
In [28]:

```
bike_vs_member_pivot_table = all_months_data.groupby(['rideable_type', 'user_typ
bike_vs_member_pivot_table
```

Out[28]:

rideable_type	classic_bike	docked_bike	electric_bike
user_type			
casual	869750.0	77826.0	1077749.0
member	1800170.0	0.0	1796384.0

In [29]:
```
bike_vs_member_pivot_table.plot(kind='bar')
plt.title('Type of Cycle used by users')
plt.xlabel('Member type')
plt.ylabel('Number of rides')
plt.legend(title='Cycle Types')
plt.yticks(np.arange(0,2000000, 250000),[0, 250000, 500000, 750000, 1000000, 125
plt.show()
```



In [30]:
```
all_months_data.sample(5)
```

Out[30]:

	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng
149382	2B7334B5C0C8FA85	classic_bike	2023-12-13 17:26:53	2023-12-13 17:33:14	41.883380	-87.641170
41819	6283D8F0A6C15B28	classic_bike	2023-10-27 12:43:07	2023-10-27 13:04:03	41.881320	-87.629521
741873	04A23656E5F581E4	electric_bike	2023-07-07 13:37:44	2023-07-07 13:41:12	41.900000	-87.630000
199840	7EAB2F44014C4FD3	electric_bike	2023-08-12 01:39:39	2023-08-12 01:50:08	41.902345	-87.627863
215292	5E08ED8F6EDC2690	electric_bike	2023-12-08 12:24:56	2023-12-08 12:29:08	41.912595	-87.681428

In [31]:

```
all_months_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5621879 entries, 0 to 224072
Data columns (total 15 columns):
 #   Column              Dtype          
---  ------              -----          
 0   ride_id             object         
 1   rideable_type       object         
 2   started_at          datetime64[ns] 
 3   ended_at            datetime64[ns] 
 4   start_lat           float64        
 5   start_lng           float64        
 6   end_lat             float64        
 7   end_lng             float64        
 8   user_type           object         
 9   ride_len            timedelta64[ns]
 10  ride_len_min        int32          
 11  started_at_day      int32          
 12  started_at_month    int32          
 13  started_at_hour     int32          
 14  started_at_dayname  category       
dtypes: category(1), datetime64[ns](2), float64(4), int32(4), object(3), timedelt
a64[ns](1)
memory usage: 563.0+ MB
```

In [32]:

```
all_months_data['ride_len_min'].describe()
```

Out[32]:

```
count    5.621879e+06
mean     1.850010e+01
std      1.817594e+02
min      1.000000e+00
25%      6.000000e+00
50%      1.000000e+01
75%      1.700000e+01
max      9.848900e+04
Name: ride_len_min, dtype: float64
```

In [ ]:

Total number of rides taken by casual and members per day

In [33]:

```
all_months_data.head()
```

Out[33]:

	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng	en
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	41.924074	-87.646278	41.93
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	41.799568	-87.594747	41.80
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	42.008571	-87.690483	42.03
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	41.799568	-87.594747	41.80
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	41.799568	-87.594747	41.80

In [34]:

```
rides_per_day = all_months_data.groupby(['started_at_day', 'user_type']).count()
rides_per_day
```

Out[34]:

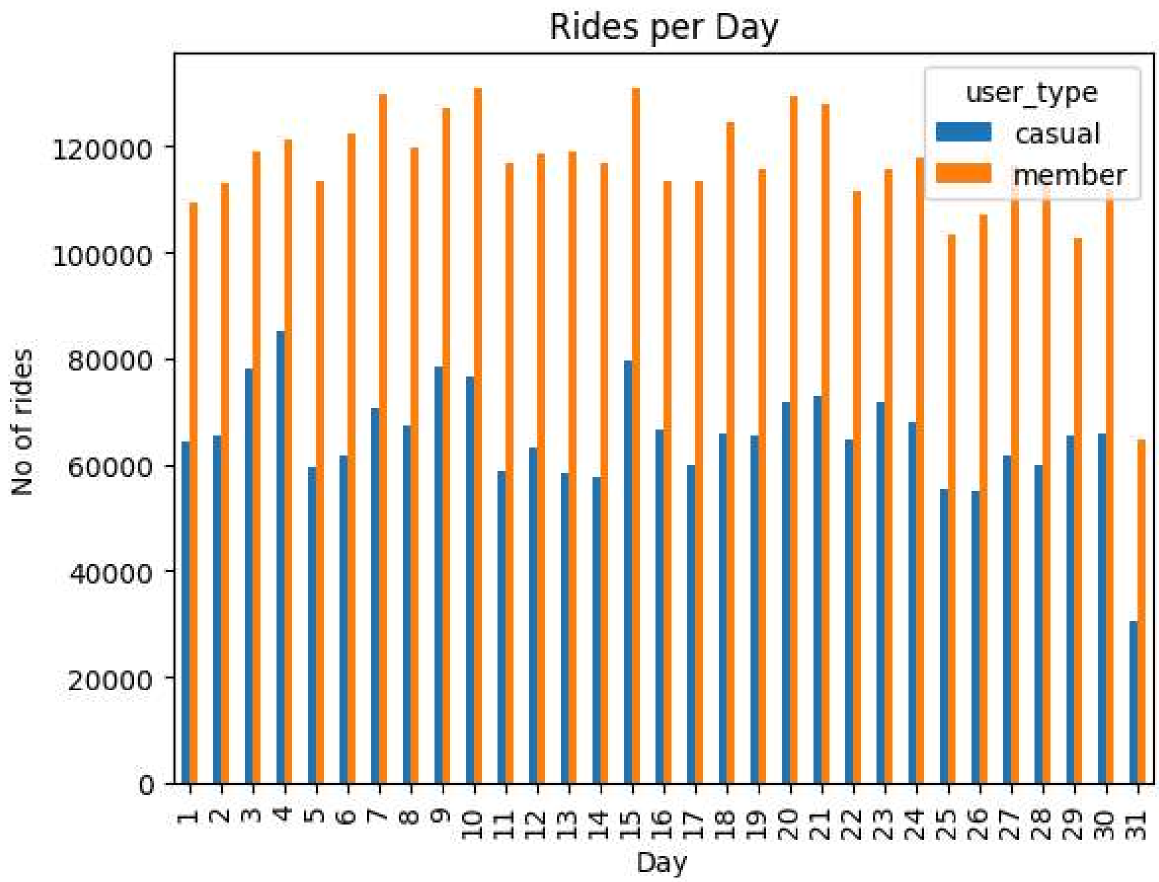
user_type	casual	member
started_at_day		
1	64352	109182
2	65361	112869
3	77944	118995
4	85136	121099
5	59388	113224
6	61704	122418
7	70528	129787
8	67304	119760
9	78282	127272
10	76730	130812
11	58790	116609
12	63318	118524
13	58287	118949
14	57702	116850
15	79469	130898
16	66531	113487
17	59935	113320
18	65862	124485
19	65371	115794
20	71692	129385
21	72886	127813
22	64717	111508
23	71803	115579
24	68010	118015
25	55560	103420
26	55137	106919
27	61797	116442
28	59903	113825
29	65516	102575
30	65825	112017
31	30485	64722

In [35]:

```
rides_per_day.plot(kind='bar')
plt.title('Rides per Day')
plt.xlabel('Day')
plt.ylabel('No of rides')
```

Out[35]:

```
Text(0, 0.5, 'No of rides')
```



In [36]:

```
all_months_data.head()
```

Out[36]:

	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng	en
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	41.924074	-87.646278	41.93
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	41.799568	-87.594747	41.80
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	42.008571	-87.690483	42.03
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	41.799568	-87.594747	41.80
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	41.799568	-87.594747	41.80

In [ ]:

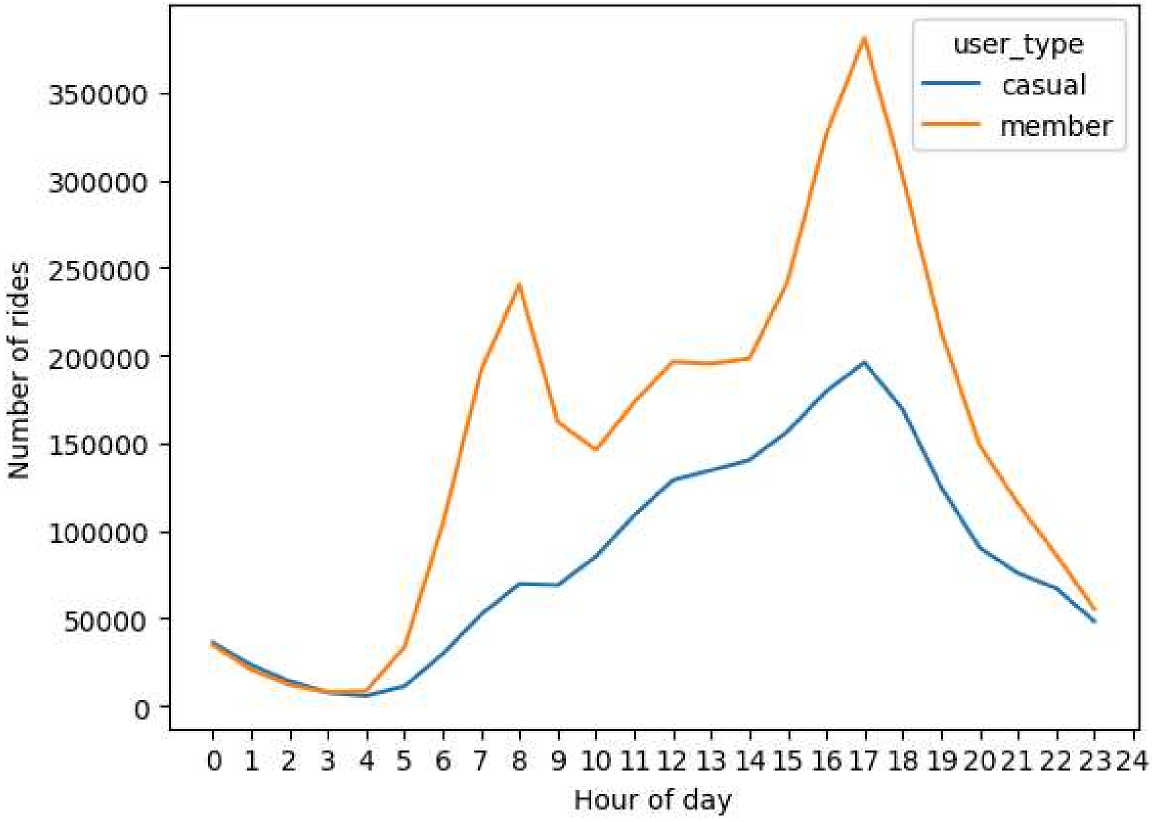
No. of rides taken by casual and members per hour in day.

In [ ]:

In [37]:

```
plt.figure(figsize=(12,6))
ride_per_hour = all_months_data.groupby(['started_at_hour','user_type']).count()
ride_per_hour.plot(kind='line')
plt.xlabel('Hour of day')
plt.xticks(np.arange(0,25))
plt.ylabel('Number of rides')
plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```

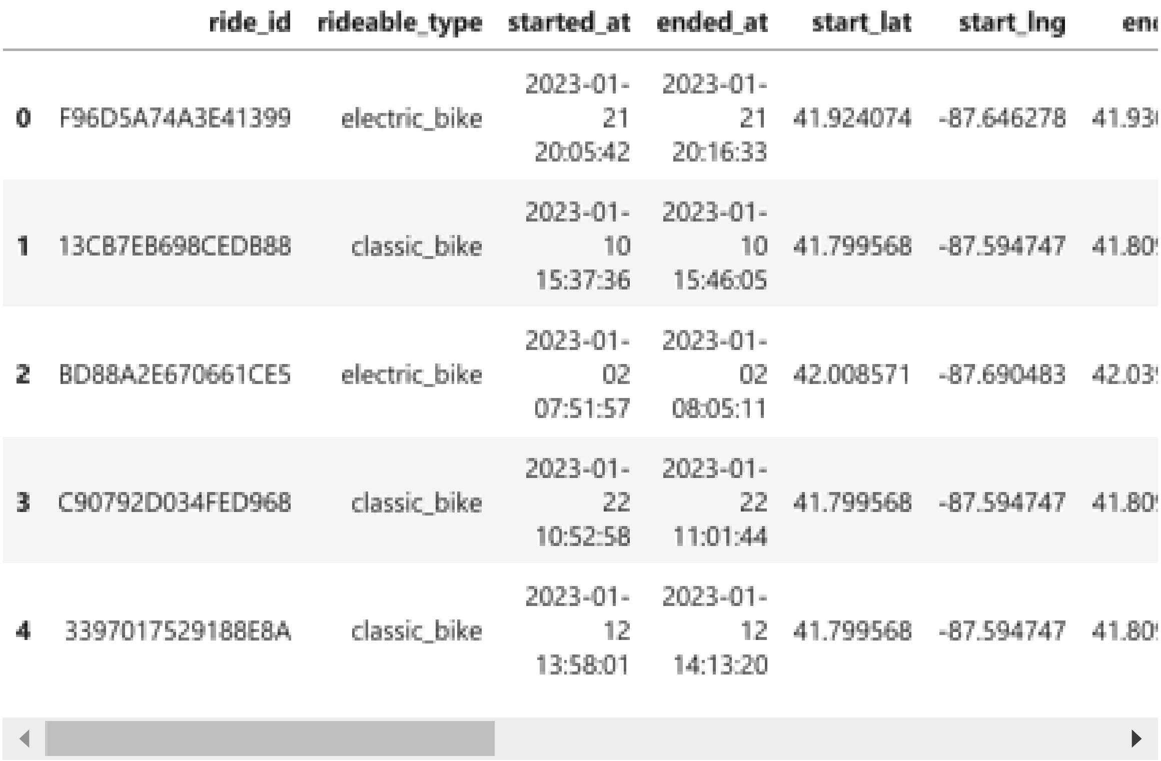


In [ ]:

In [38]:

```
all_months_data.head()
```

Out[38]:



	ride_id	rideable_type	started_at	ended_at	start_lat	start_lng	en
0	F96D5A74A3E41399	electric_bike	2023-01-21 20:05:42	2023-01-21 20:16:33	41.924074	-87.646278	41.93
1	13CB7EB698CEDB88	classic_bike	2023-01-10 15:37:36	2023-01-10 15:46:05	41.799568	-87.594747	41.80
2	BD88A2E670661CE5	electric_bike	2023-01-02 07:51:57	2023-01-02 08:05:11	42.008571	-87.690483	42.03
3	C90792D034FED968	classic_bike	2023-01-22 10:52:58	2023-01-22 11:01:44	41.799568	-87.594747	41.80
4	3397017529188E8A	classic_bike	2023-01-12 13:58:01	2023-01-12 14:13:20	41.799568	-87.594747	41.80

In [47]:

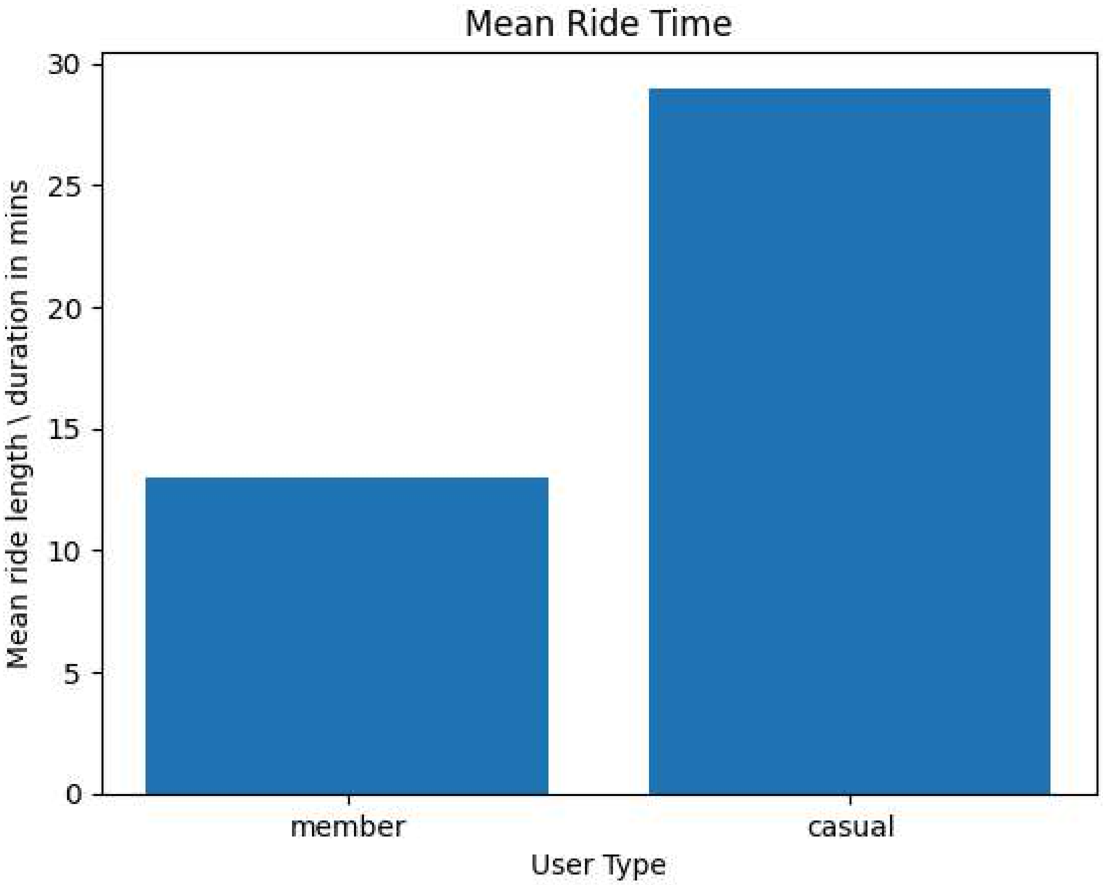
```
new_df = all_months_data[all_months_data['user_type'] == 'member']
m_mean = round(new_df['ride_len_min'].mean())
```

In [48]:

```
new_df_2 = all_months_data[all_months_data['user_type'] == 'casual']
c_mean = round(new_df_2['ride_len_min'].mean())
```

In [51]:

```
plt.bar(['member','casual'], [m_mean, c_mean])
plt.title('Mean Ride Time')
plt.xlabel('User Type')
plt.ylabel('Mean ride length \ duration in mins')
plt.show()
```



In [ ]:

In [52]:
```
# ex_df = all_months_data.sample(100000)
# ex_df.to_csv('sample_d_t.csv')
```

In [ ]:

In [ ]:

In [ ]: