SuperBike - Report

In [1]:
```
# Load the libraries needed
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# Load the dataset of Number of Rides per month in 2022
data_num_rides = pd.read_csv('df_num_rides.csv')
data_num_rides
```

Out[2]:

	rider_type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
0	casual	18520	21416	89882	126417	280415	369051	406055	358924	296697	208989
1	member	85250	94193	194160	244832	354443	400153	417433	427008	404642	349696
2	total	103770	115609	284042	371249	634858	769204	823488	785932	701339	558685

Total Number of Rides in 2022

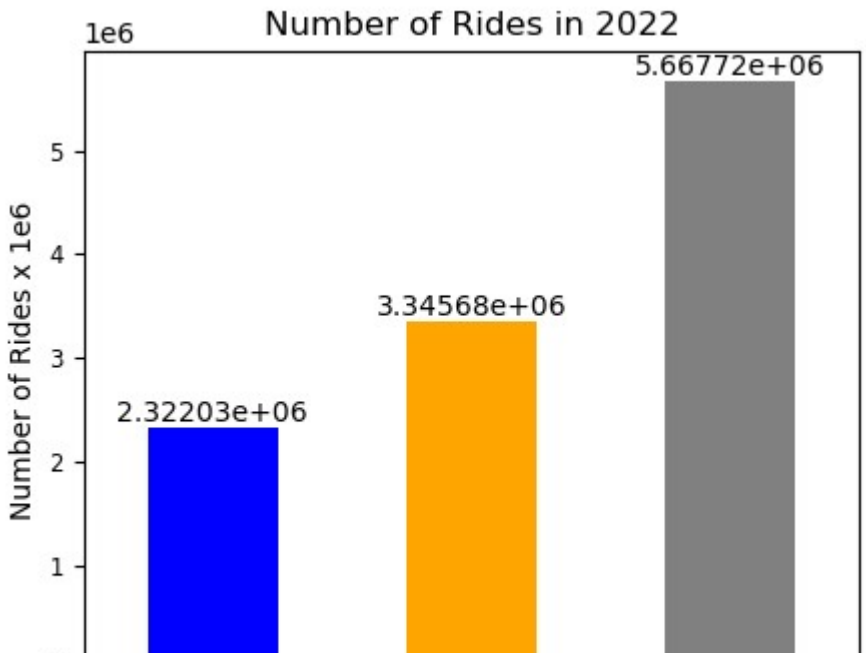
In [3]:
```
# Calculation of total number of rides
import re
casual_ri = data_num_rides.select_dtypes(include='number').iloc[[0]].sum(axi
member_ri = data_num_rides.select_dtypes(include='number').iloc[[1]].sum(axi
total_ri = data_num_rides.select_dtypes(include='number').iloc[[2]].sum(axis
print('Total Number of Rides in 2022: ',re.sub("(\d)(?=(\d{3})+(?!\d))", r"\
print('Number of Casual Rides in 2022: ',re.sub("(\d)(?=(\d{3})+(?!\d))", r"
print('Number of Member Rides in 2022: ',re.sub("(\d)(?=(\d{3})+(?!\d))", r"
```

```
Total Number of Rides in 2022:  5,667,717
Number of Casual Rides in 2022:  2,322,032
Number of Member Rides in 2022:  3,345,685
```

In [4]:
```
from tabulate import tabulate
num_ri = {'Type':['Casual','Member','Total'],
          'Num_Rides':[re.sub("(\d)(?=(\d{3})+(?!\d))", r"\1,", "%d" % casual
                       re.sub("(\d)(?=(\d{3})+(?!\d))", r"\1,", "%d" % member_r
                       re.sub("(\d)(?=(\d{3})+(?!\d))", r"\1,", "%d" % total_ri
num_ri = pd.DataFrame(num_ri)
print('Number of Rides in 2022')
print(tabulate(num_ri, tablefmt='grid'))
```

```
Number of Rides in 2022
+---+--------+-----------+
| 0 | Casual | 2,322,032 |
+---+--------+-----------+
| 1 | Member | 3,345,685 |
+---+--------+-----------+
| 2 | Total  | 5,667,717 |
+---+--------+-----------+
```

In [5]:
```
# Plot of the Number of rides in 2022
num_ri = num_ri = {'Type':['Casual','Member','Total'],
          'Num_Rides':[2322032,3345685,5667717]}
num_ri = pd.DataFrame(num_ri)
ax = num_ri.plot.bar(x='Type', y='Num_Rides', rot=0, color=['blue','orange',
                     xlabel="", ylabel="Number of Rides x 1e6", fontsize='sma
plt.title('Number of Rides in 2022', size=12)
# Annotate the values in the bar
for container in ax.containers:
    ax.bar_label(container, size=10)
```

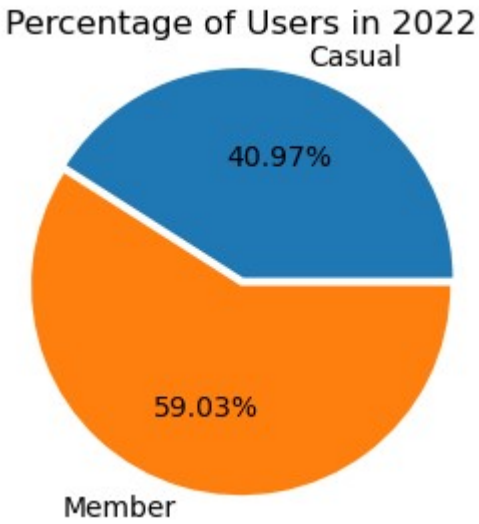


Percentage of type of Riders

In [6]:
```
# Percentage of casual and member rides in 2022
print('Percentage of Casual rides in 2022: ',round(float(casual_ri)*100/floa
print('Percentage of Member rides in 2022: ',round(float(member_ri)*100/floa
```

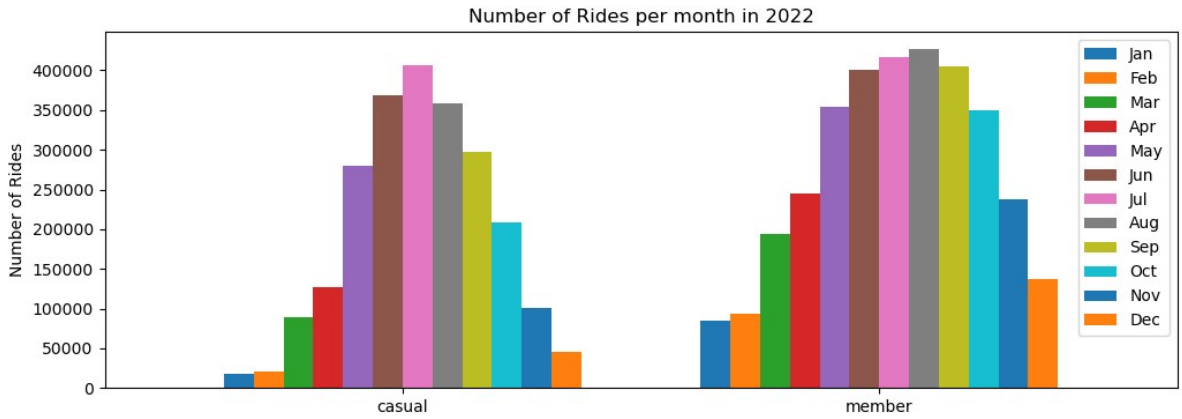
```
Percentage of Casual rides in 2022:  40.97 %
Percentage of Member rides in 2022:  59.03 %
```

In [7]:
```
# Pie Chart of the Percentage of users in 2022.
p1 = [40.97, 59.03]
p2 = ['Casual','Member']
myexplode = [0, 0.02]
plt.figure(figsize = (3,3))
plt.pie(p1,labels=p2, radius=0.5, autopct='%0.2f%%', shadow=False, frame=Fals
              explode = myexplode);
plt.axis('Equal');
#plt.legend(title='Type of Users',loc ="upper left",bbox_to_anchor =(0, 0, 0
plt.title("Percentage of Users in 2022",fontsize=12);
```



Number of Rides per month

In [8]:
```
# Number of Rides per month and type of user in 2022
data_num_rides.iloc[[0,1]].plot.bar(x='rider_type',rot=0,figsize=(12,4),
                                    title='Number of Rides per month in 2022
                                    ylabel='Number of Rides',
                                   align='center', width=0.75);
```



Percentage of type of User per month

In [9]:
```
# Percentage of Type of Rides per month
df_p = pd.read_csv('df_perc_rides.csv')
df_p
```

Out[9]:

	rider_type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
0	casual	17.85	18.52	31.64	34.05	44.17	47.98	49.31	45.67	42.3	37.42	29.84	24.69
1	member	82.15	81.48	68.36	65.95	55.83	52.02	50.69	54.33	57.7	62.59	70.16	75.31

In [10]:
```
# Create a list with the values of the above table
list1 = df_p.values.tolist()
```

In [11]:
```
# Remove some elements of the list
list1[0].remove('casual')
list1[1].remove('member')
```

In [12]:
```
# List with the names of the columns
list2 = df_p.columns.values.tolist()
```

In [13]:
```
list2.remove('rider_type')
```

In [14]:
```
# List of percentages for Casual users
mylist_casual = list1[0]
mylist_casual = [int(round(x,0)) for x in mylist_casual]
print(mylist_casual)
```

```
[18, 19, 32, 34, 44, 48, 49, 46, 42, 37, 30, 25]
```

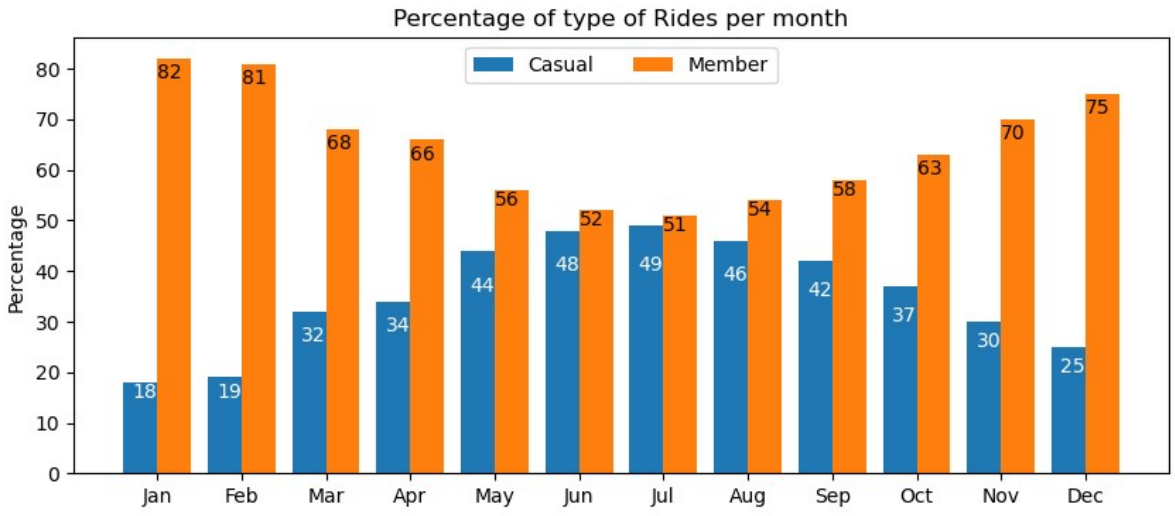
In [15]:
```
# List of percentages for Member users
mylist_member = list1[1]
mylist_member = [int(round(x,0)) for x in mylist_member]
print(mylist_member)
```

```
[82, 81, 68, 66, 56, 52, 51, 54, 58, 63, 70, 75]
```

In [16]:
```
# List of columns
print(list2)
```

```
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov',
'Dec']
```

In [17]:
```
# Plot of the Percentage of type rides per month
import numpy as np
# function to add value labels
def addlabels1(x,y):
    for i in range(len(x)):
        plt.text(i, y[i]//1.2, y[i], ha = 'right', color='white', size=10)
def addlabels2(x,y):
    for i in range(len(x)):
        plt.text(i, y[i]//1.05, y[i], ha = 'left', color='black')

if __name__ == '__main__':

    # creating data on which bar chart will be plot
    x = list2
    ycasual = mylist_casual
    ymember = mylist_member

    X_axis = np.arange(len(x))

# setting figure size by using figure() function
    plt.figure(figsize = (10,4))

    # making the bar chart on the data
    plt.bar(X_axis- 0.2, ycasual,0.4, label = 'Casual');
    plt.bar(X_axis+ 0.2,ymember,0.4, label = 'Member');
    # calling the function to add value labels
    addlabels1(X_axis, ycasual)
    addlabels2(X_axis,ymember)
    # giving title to the plot
    plt.title("Percentage of type of Rides per month")

    # giving X and Y labels
    plt.xlabel("")
    plt.ylabel("Percentage")
    plt.xticks(X_axis, x, fontsize=10)
    plt.legend(loc='upper center',ncol=2, fontsize=10)
    # visualizing the plot
    plt.show()
```



Percentage of Rides per Group

In [18]:
```
# Dataframe of the number of rides per month
data_num_rides
```

Out[18]:

	rider_type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
0	casual	18520	21416	89882	126417	280415	369051	406055	358924	296697	208989
1	member	85250	94193	194160	244832	354443	400153	417433	427008	404642	349696
2	total	103770	115609	284042	371249	634858	769204	823488	785932	701339	558685

In [19]:
```
# Create a list of the dataset
list_1 = data_num_rides.values.tolist()
print(list_1)
```

```
[['casual', 18520, 21416, 89882, 126417, 280415, 369051, 406055, 358924, 2966
97, 208989, 100772, 44894], ['member', 85250, 94193, 194160, 244832, 354443,
400153, 417433, 427008, 404642, 349696, 236963, 136912], ['total', 103770, 11
5609, 284042, 371249, 634858, 769204, 823488, 785932, 701339, 558685, 337735,
181806]]
```

In [20]:
```
# Remove some elements from the list
list_1[0].remove('casual')
list_1[1].remove('member')
```

In [21]:
```
print('Total number of Casual users: ',re.sub("(\d)(?=(\d{3})+(?!\d))", r"\1
print('Total number of Member users: ',re.sub("(\d)(?=(\d{3})+(?!\d))", r"\1
```

```
Total number of Casual users:  2,322,032
Total number of Member users:  3,345,685
```

In [22]:
```
# Percentage of Casual users in the year
list_p_casual=[]
for x in list_1[0]:
    list_p_casual.append(int(round(x*100/sum(list_1[0]),0)))
```

In [23]:
```
list_p_casual
```

Out[23]: [1, 1, 4, 5, 12, 16, 17, 15, 13, 9, 4, 2]

In [24]:
```
# Percentage of Member users in the year
list_p_member=[]
for x in list_1[1]:
    list_p_member.append(int(round(x*100/sum(list_1[1]),0)))
```

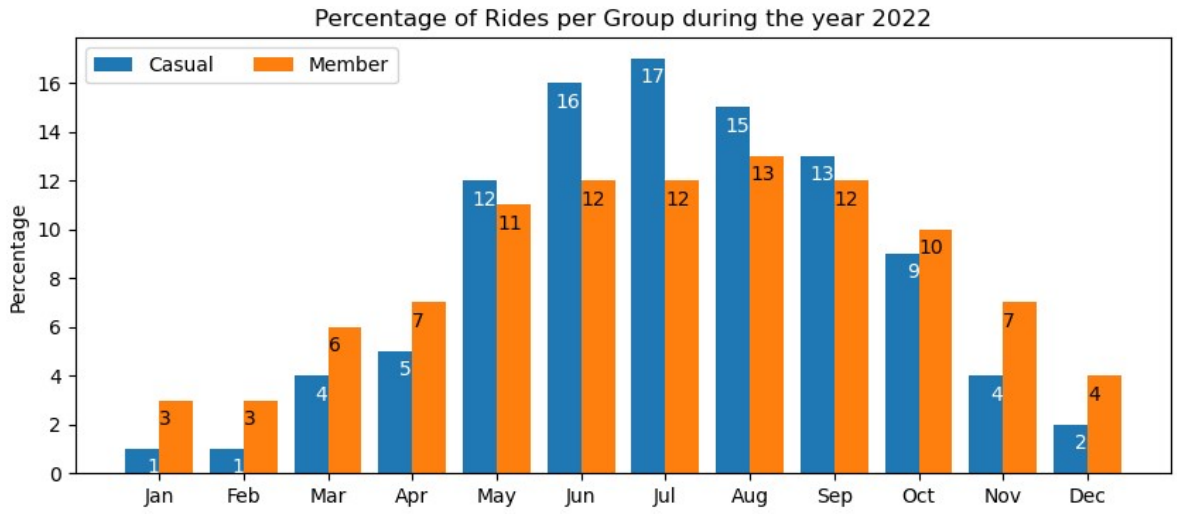
In [25]:
```
list_p_member
```

Out[25]: [3, 3, 6, 7, 11, 12, 12, 13, 12, 10, 7, 4]

In [26]:
```
print(list2)
```

```
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov',
'Dec']
```

In [27]:
```
# function to add value labels
def addlabels1(x,y):
    for i in range(len(x)):
        plt.text(i, y[i]//1.01, y[i], ha = 'right', color='white', size=10)
def addlabels2(x,y):
    for i in range(len(x)):
        plt.text(i, y[i]//1.01, y[i], ha = 'left', color='black')

if __name__ == '__main__':

    # creating data on which bar chart will be plot
    x = list2
    ycasual = list_p_casual
    ymember = list_p_member

    X_axis = np.arange(len(x))

    # setting figure size by using figure() function
    plt.figure(figsize = (10, 4))

    # making the bar chart on the data
    plt.bar(X_axis- 0.2, ycasual,0.4, label = 'Casual');
    plt.bar(X_axis+ 0.2,ymember,0.4, label = 'Member');
    # calling the function to add value labels
    addlabels1(X_axis, ycasual)
    addlabels2(X_axis,ymember)
    # giving title to the plot
    plt.title("Percentage of Rides per Group during the year 2022")

    # giving X and Y labels
    plt.xlabel("")
    plt.ylabel("Percentage")
    plt.xticks(X_axis, x, fontsize=10)
    plt.legend(loc='upper left',ncol=2, fontsize=10)
    # visualizing the plot
    plt.show()
```



Number of Rides per month

In [28]:
```
data_num_rides
```

Out[28]:

	rider_type	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct
0	casual	18520	21416	89882	126417	280415	369051	406055	358924	296697	208989
1	member	85250	94193	194160	244832	354443	400153	417433	427008	404642	349696
2	total	103770	115609	284042	371249	634858	769204	823488	785932	701339	558685

In [29]:
```
# Dictionary of the values of the dataset for Casual riders
d ={'rider_type':['casual', 'casual', 'casual', 'casual', 'casual','casual',
                  'casual','casual','casual','casual'],
   'month':['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
   'num_rides':[18520,21416,89882,126417,280415,369051,406055,358924,296697,
```

In [30]:
```
df = pd.DataFrame(d)
```

In [31]:
```
# Dictionary of the values of the dataset for Member riders
d1 ={'rider_type':['member', 'member', 'member', 'member', 'member','member'
                  'member','member','member','member'],
   'month':['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
   'num_rides':[85250,94193,194160,244832,354443,400153,417433,427008,404642
```

In [32]:
```
df1 = pd.DataFrame(d1)
```

In [33]:
```
# Dictionary of the values of the dataset for the Total riders
d2 = {'rider_type':['total', 'total', 'total', 'total', 'total','total','tota
                  'total','total','total','total'],
   'month':['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
   'num_rides':[103770,115609,284042,371249,634858,769204,823488,785932,7013
                181806]}
```

In [34]:
```
df2 = pd.DataFrame(d2)
```

In [35]:
```
frames = [df,df1,df2]
result = pd.concat(frames)
```

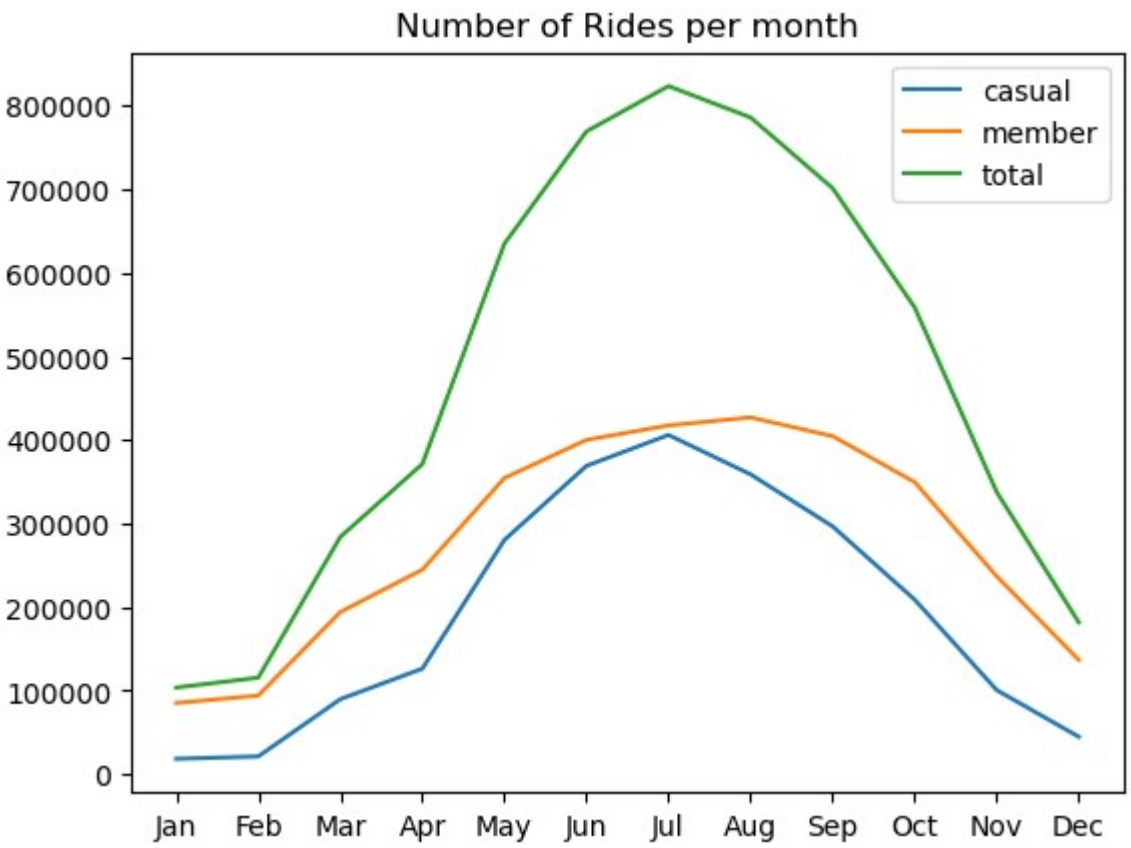
In [36]:
```
result
```

Out[36]:

	rider_type	month	num_rides
0	casual	Jan	18520
1	casual	Feb	21416
2	casual	Mar	89882
3	casual	Apr	126417
4	casual	May	280415
5	casual	Jun	369051
6	casual	Jul	406055
7	casual	Aug	358924
8	casual	Sep	296697
9	casual	Oct	208989
10	casual	Nov	100772
11	casual	Dec	44894
0	member	Jan	85250
1	member	Feb	94193
2	member	Mar	194160
3	member	Apr	244832
4	member	May	354443
5	member	Jun	400153
6	member	Jul	417433
7	member	Aug	427008
8	member	Sep	404642
9	member	Oct	349696
10	member	Nov	236963
11	member	Dec	136912
0	total	Jan	103770
1	total	Feb	115609
2	total	Mar	284042
3	total	Apr	371249
4	total	May	634858
5	total	Jun	769204
6	total	Jul	823488
7	total	Aug	785932
8	total	Sep	701339

	rider_type	month	num_rides
9	total	Oct	558685

In [37]:
```
# Plot of the Number of Users per month
rider_set = set(result['rider_type'])

plt.figure()
for user in rider_set:
     selected_data = result.loc[result['rider_type'] == user]
     plt.plot(selected_data['month'], selected_data['num_rides'], label=user

plt.title('Number of Rides per month')
plt.legend()
plt.show()
```



Number of Rides per Day of the Week and Month

January

In [38]:
```
# Load DataFrame of number of rides in January
df_j = pd.read_csv('df_num_days_jan')
df_j
```

Out[38]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	2515	2429	2394	2389	2543	2459	3791	Jan
1	member	8994	13374	13755	12785	14011	11352	10979	Jan
2	total	11509	15803	16149	15174	16554	13811	14770	Jan

In [39]:
```
# Select only the third row of values
df_jan = df_j[df_j['type']=='total']
df_jan
```

Out[39]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
2	total	11509	15803	16149	15174	16554	13811	14770	Jan

In [40]:
```
# Create a list with the days of the week
list_day = df_jan.columns.values.tolist()
list_day.remove('type')
list_day.remove('month')
print(list_day)
print(len(list_day))
```

```
['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday
']
7
```

In [41]:
```
# Create a list with the number of rides in January
list_j1 = df_jan.select_dtypes(include='number').values.tolist()
print(list_j1)
len(list_j1)
```

```
[[11509, 15803, 16149, 15174, 16554, 13811, 14770]]
```

Out[41]: 1

In [42]:
```
# Convert a nested list into a flat list
list_j1_co = sum(list_j1, [])

print('New list', list_j1_co)
print(len(list_j1_co))
```

```
New list [11509, 15803, 16149, 15174, 16554, 13811, 14770]
7
```

In [43]:
```
# Dictionary of the number of rides in January
dict_jan = {'month':['jan']*7,'num_rides':list_j1_co,'day':list_day}
print(dict_jan)
```

```
{'month': ['jan', 'jan', 'jan', 'jan', 'jan', 'jan', 'jan'], 'num_rides': [11
509, 15803, 16149, 15174, 16554, 13811, 14770], 'day': ['Sunday', 'Monday', '
Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']}
```

In [44]:
```
# DataFrame of the number of rides in January
df_jan1 = pd.DataFrame(dict_jan)
df_jan1
```

Out[44]:

	month	num_rides	day
0	jan	11509	Sunday
1	jan	15803	Monday
2	jan	16149	Tuesday
3	jan	15174	Wednesday
4	jan	16554	Thursday
5	jan	13811	Friday
6	jan	14770	Saturday

February

In [45]:
```
# Load February dataset of number of rides per day of the week
df_f = pd.read_csv('df_num_days2_feb')
df_f
```

Out[45]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	4206	4405	2787	2623	1879	2698	2818	Feb
1	member	11686	18375	16259	14609	11635	11960	9669	Feb
2	total	15892	22780	19046	17232	13514	14658	12487	Feb

In [46]:
```
# Seleted row of the total rides per day of the week
df_feb = df_f[df_f['type']=='total']
df_feb
```

Out[46]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
2	total	15892	22780	19046	17232	13514	14658	12487	Feb

In [47]:
```
# List of number of rides (values) of each day of the week
list_feb= df_feb.select_dtypes(include='number').values.tolist()
print(list_feb)
len(list_feb)
```

```
[[15892, 22780, 19046, 17232, 13514, 14658, 12487]]
```

Out[47]: 1

In [48]:
```
# Convert the previous nested list into a flat list
list_feb_co = sum(list_feb, [])
print(list_feb_co)
print(len(list_feb_co))
```

```
[15892, 22780, 19046, 17232, 13514, 14658, 12487]
7
```

In [49]:
```
# Dictionary of the number of rides per day of the week for February
dict_feb = {'month':['feb']*7,'num_rides':list_feb_co,'day':list_day}
print(dict_feb)
```

```
{'month': ['feb', 'feb', 'feb', 'feb', 'feb', 'feb', 'feb'], 'num_rides': [15
892, 22780, 19046, 17232, 13514, 14658, 12487], 'day': ['Sunday', 'Monday', '
Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday']}
```

In [50]:
```
# DataFrame of the number of rides per day of the week in February
df_feb1 = pd.DataFrame(dict_feb)
df_feb1
```

Out[50]:

	month	num_rides	day
0	feb	15892	Sunday
1	feb	22780	Monday
2	feb	19046	Tuesday
3	feb	17232	Wednesday
4	feb	13514	Thursday
5	feb	14658	Friday
6	feb	12487	Saturday

March

In [51]:
```
# Load March dataset of number of rides per day of the week
df_m = pd.read_csv('df_num_days3_mar')
df_m
```

Out[51]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	16575	14449	10154	14540	12024	7156	14984	Mar
1	member	22068	29449	34406	35957	32141	20492	19647	Mar
2	total	38643	43898	44560	50497	44165	27648	34631	Mar

In [52]:
```
# Function to create a dataframe with the number of rides in March
def selected_row(x):
    global df_m
    df_m = df_m[df_m['type']=='total']
    list_m= df_m.select_dtypes(include='number').values.tolist()
    list_m = sum(list_m, [])
    dict_m = {'month':[x]*7,'num_rides':list_m,'day':list_day}
    df_m = pd.DataFrame(dict_m)
    return df_m

selected_row('mar')
```

Out[52]:

	month	num_rides	day
0	mar	38643	Sunday
1	mar	43898	Monday
2	mar	44560	Tuesday
3	mar	50497	Wednesday
4	mar	44165	Thursday
5	mar	27648	Friday
6	mar	34631	Saturday

April

In [53]:
```
# April dataset of number of rides per day of the week
df_a = pd.read_csv('df_num_days4_apr.csv')
df_a
```

Out[53]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	19388	12063	14550	10457	16779	16850	36330	Apr
1	member	25457	33931	40432	32387	38595	35963	38067	Apr
2	total	44845	45994	54982	42844	55374	52813	74397	Apr

In [54]:
```
# Function to create a dataframe with the number of rides in April
def selected_row1(x):
    global df_a
    df_a = df_a[df_a['type']=='total']
    list_a= df_a.select_dtypes(include='number').values.tolist()
    list_a = sum(list_a, [])
    dict_a = {'month':[x]*7,'num_rides':list_a,'day':list_day}
    df_a = pd.DataFrame(dict_a)
    return df_a

selected_row1('apr')
```

Out[54]:

	month	num_rides	day
0	apr	44845	Sunday
1	apr	45994	Monday
2	apr	54982	Tuesday
3	apr	42844	Wednesday
4	apr	55374	Thursday
5	apr	52813	Friday
6	apr	74397	Saturday

May

In [55]:
```
# May dataset of number of rides per day of the week
df_ma = pd.read_csv('df_num_days_may.csv')
df_ma
```

Out[55]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	55321	47469	35067	24066	33399	32238	52855	May
1	member	48774	62061	59543	45097	51663	42310	44995	May
2	total	104095	109530	94610	69163	85062	74548	97850	May

In [56]:
```
# Function to create a dataframe with the number of rides in May
def selected_row2(x):
    global df_ma
    df_ma = df_ma[df_ma['type']=='total']
    list_ma= df_ma.select_dtypes(include='number').values.tolist()
    list_ma = sum(list_ma, [])
    dict_ma = {'month':[x]*7,'num_rides':list_ma,'day':list_day}
    df_ma = pd.DataFrame(dict_ma)
    return df_ma

selected_row2('may')
```

Out[56]:

	month	num_rides	day
0	may	104095	Sunday
1	may	109530	Monday
2	may	94610	Tuesday
3	may	69163	Wednesday
4	may	85062	Thursday
5	may	74548	Friday
6	may	97850	Saturday

June

In [57]:
```
# June dataset of number of rides per day of the week
df_jun = pd.read_csv('df_num_days6_jun.csv')
df_jun
```

Out[57]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	65851	37005	38825	48387	57978	55868	65137	Jun
1	member	49062	46791	54986	68797	73501	57322	49694	Jun
2	total	114913	83796	93811	117184	131479	113190	114831	Jun

In [58]:
```
# Function to create a dataframe with the number of rides in June
def selected_row3(x):
    global df_jun
    df_jun = df_jun[df_jun['type']=='total']
    list_jun= df_jun.select_dtypes(include='number').values.tolist()
    list_jun = sum(list_jun, [])
    dict_jun = {'month':[x]*7,'num_rides':list_jun,'day':list_day}
    df_jun = pd.DataFrame(dict_jun)
    return df_jun

selected_row3('jun')
```

Out[58]:

	month	num_rides	day
0	jun	114913	Sunday
1	jun	83796	Monday
2	jun	93811	Tuesday
3	jun	117184	Wednesday
4	jun	131479	Thursday
5	jun	113190	Friday
6	jun	114831	Saturday

July

In [59]:
```
# July dataset of number of rides per day of the week
df_jul = pd.read_csv('df_num_days7_jul.csv')
df_jul
```

Out[59]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	78251	43971	41455	42850	47793	56505	95230	Jul
1	member	58780	49850	57524	59611	61155	61644	68869	Jul
2	total	137031	93821	98979	102461	108948	118149	164099	Jul

In [60]:
```
# Function to create a dataframe with the number of rides in July
def selected_row4(x):
    global df_jul
    df_jul = df_jul[df_jul['type']=='total']
    list_jul= df_jul.select_dtypes(include='number').values.tolist()
    list_jul = sum(list_jul, [])
    dict_jul = {'month':[x]*7,'num_rides':list_jul,'day':list_day}
    df_jul = pd.DataFrame(dict_jul)
    return df_jul

selected_row4('jul')
```

Out[60]:

	month	num_rides	day
0	jul	137031	Sunday
1	jul	93821	Monday
2	jul	98979	Tuesday
3	jul	102461	Wednesday
4	jul	108948	Thursday
5	jul	118149	Friday
6	jul	164099	Saturday

August

In [61]:
```
# August dataset of number of rides per day of the week
df_aug = pd.read_csv('df_num_days8_aug.csv')
df_aug
```

Out[61]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	48154	42362	51509	51488	42353	56868	66190	Aug
1	member	42972	62603	76718	76616	57510	58703	51886	Aug
2	total	91126	104965	128227	128104	99863	115571	118076	Aug

In [62]:
```
# Function to create a dataframe with the number of rides in August
def selected_row5(x):
    global df_aug
    df_aug = df_aug[df_aug['type']=='total']
    list_aug= df_aug.select_dtypes(include='number').values.tolist()
    list_aug = sum(list_aug, [])
    dict_aug = {'month':[x]*7,'num_rides':list_aug,'day':list_day}
    df_aug = pd.DataFrame(dict_aug)
    return df_aug

selected_row5('aug')
```

Out[62]:

	month	num_rides	day
0	aug	91126	Sunday
1	aug	104965	Monday
2	aug	128227	Tuesday
3	aug	128104	Wednesday
4	aug	99863	Thursday
5	aug	115571	Friday
6	aug	118076	Saturday

September

In [63]:
```
# September dataset of number of rides per day of the week
df_s = pd.read_csv('df_num_days9_sep.csv')
df_s
```

Out[63]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	36254	31052	29586	33504	45839	56385	64077	Sep
1	member	35740	47422	57033	61507	76591	72330	54019	Sep
2	total	71994	78474	86619	95011	122430	128715	118096	Sep

In [64]:
```
# Function to create a dataframe with the number of rides in September
def selected_row6(x):
    global df_s
    df_s = df_s[df_s['type']=='total']
    list_s= df_s.select_dtypes(include='number').values.tolist()
    list_s = sum(list_s, [])
    dict_s = {'month':[x]*7,'num_rides':list_s,'day':list_day}
    df_s = pd.DataFrame(dict_s)
    return df_s

selected_row6('sep')
```

Out[64]:

	month	num_rides	day
0	sep	71994	Sunday
1	sep	78474	Monday
2	sep	86619	Tuesday
3	sep	95011	Wednesday
4	sep	122430	Thursday
5	sep	128715	Friday
6	sep	118096	Saturday

October

In [65]:
```
# October dataset of number of rides per day of the week
df_o = pd.read_csv('df_num_days10_oct.csv')
df_o
```

Out[65]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	44681	27234	15694	20593	22585	25980	52222	Oct
1	member	50366	58289	39748	48893	49239	45024	58137	Oct
2	total	95047	85523	55442	69486	71824	71004	110359	Oct

In [66]:
```
# Function to create a dataframe with the number of rides in October
def selected_row7(x):
    global df_o
    df_o = df_o[df_o['type']=='total']
    list_o= df_o.select_dtypes(include='number').values.tolist()
    list_o = sum(list_o, [])
    dict_o = {'month':[x]*7,'num_rides':list_o,'day':list_day}
    df_o = pd.DataFrame(dict_o)
    return df_o

selected_row7('oct')
```

Out[66]:

	month	num_rides	day
0	oct	95047	Sunday
1	oct	85523	Monday
2	oct	55442	Tuesday
3	oct	69486	Wednesday
4	oct	71824	Thursday
5	oct	71004	Friday
6	oct	110359	Saturday

November

In [67]:
```
# November dataset of number of rides per day of the week
df_n = pd.read_csv('df_num_days11_nov.csv')
df_n
```

Out[67]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	12497	10300	15808	17779	17981	14538	11869	Nov
1	member	21207	32305	46114	47448	39125	30324	20440	Nov
2	total	33704	42605	61922	65227	57106	44862	32309	Nov

In [68]:
```
# Function to create a dataframe with the number of rides in November
def selected_row8(x):
    global df_n
    df_n = df_n[df_n['type']=='total']
    list_n= df_n.select_dtypes(include='number').values.tolist()
    list_n = sum(list_n, [])
    dict_n = {'month':[x]*7,'num_rides':list_n,'day':list_day}
    df_n = pd.DataFrame(dict_n)
    return df_n

selected_row8('nov')
```

Out[68]:

	month	num_rides	day
0	nov	33704	Sunday
1	nov	42605	Monday
2	nov	61922	Tuesday
3	nov	65227	Wednesday
4	nov	57106	Thursday
5	nov	44862	Friday
6	nov	32309	Saturday

December

In [69]:
```
# December dataset of number of rides per day of the week
df_d = pd.read_csv('df_num_days12_dec.csv')
df_d
```

Out[69]:

	type	Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	month
0	casual	5343	4936	5917	5678	8177	7156	7687	Dec
1	member	12117	18889	22108	20162	27095	19662	16879	Dec
2	total	17460	23825	28025	25840	35272	26818	24566	Dec

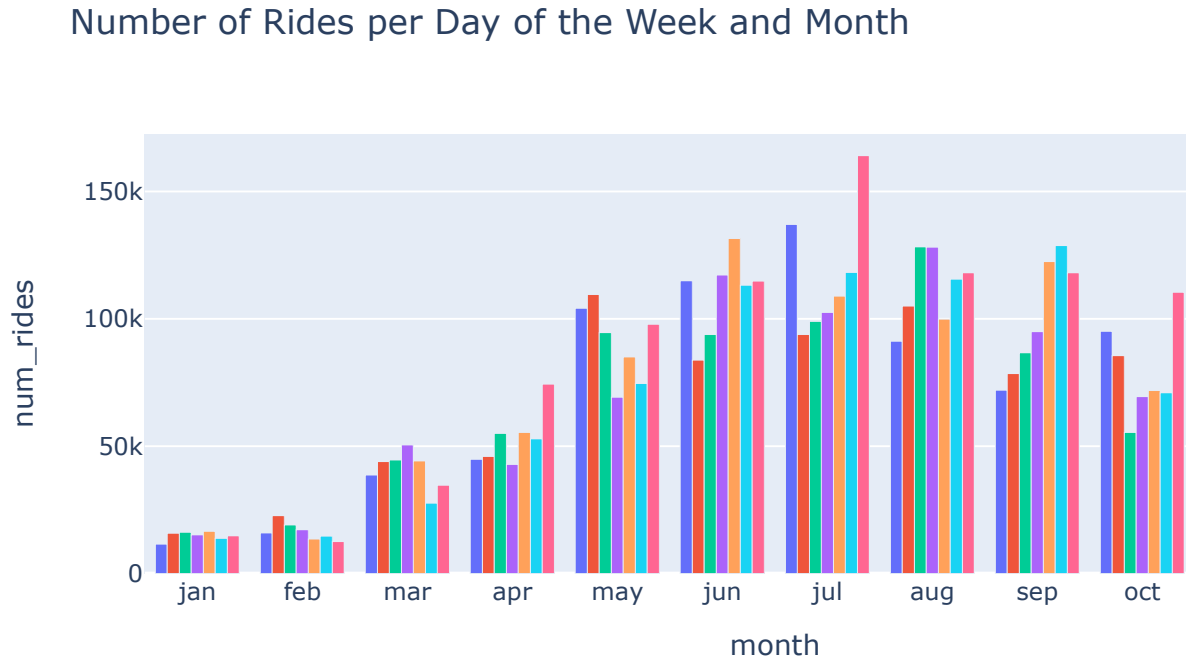
In [70]:
```
# Function to create a dataframe with the number of rides in December
def selected_row9(x):
    global df_d
    df_d = df_d[df_d['type']=='total']
    list_d= df_d.select_dtypes(include='number').values.tolist()
    list_d = sum(list_d, [])
    dict_d = {'month':[x]*7,'num_rides':list_d,'day':list_day}
    df_d = pd.DataFrame(dict_d)
    return df_d

selected_row9('dec')
```

Out[70]:

	month	num_rides	day
0	dec	17460	Sunday
1	dec	23825	Monday
2	dec	28025	Tuesday
3	dec	25840	Wednesday
4	dec	35272	Thursday
5	dec	26818	Friday
6	dec	24566	Saturday

In [71]:
```
# Create a datadframe with the number of rides of every month in 2022
frames = [df_jan1,df_feb1,df_m,df_a,df_ma,df_jun,df_jul,df_aug,df_s,df_o,df_
df = pd.concat(frames)
```

In [72]:
```
# Plot the number of rides per day of the week
import plotly.express as px

fig = px.bar(df, x="month", y="num_rides", color='day',width=850, height=400
             title='Number of Rides per Day of the Week and Month',barmode='g
fig.show()
```



Most Frequent Trip Duration in minutes for Casual/Member users

In [73]:
```
# Load the dataset for January
df_m_jan = pd.read_csv('df_mode_diff_jan')
df_m_jan
```

Out[73]:

	type	mode	month
0	casual	0.17	Jan
1	member	4.33	Jan

In [74]:
```
df_m_feb = pd.read_csv('df_mode_diff2_feb')
df_m_feb
```

Out[74]:

	type	mode	month
0	casual	0.03	Feb
1	member	0.03	Feb

In [75]:
```
df_m_mar = pd.read_csv('df_mode_diff3_mar.csv')
df_m_mar
```

Out[75]:

	type	mode	month
0	casual	7.15	Mar
1	member	4.32	Mar

In [76]:
```
df_m_apr = pd.read_csv('df_mode_diff4_apr.csv')
df_m_apr
```

Out[76]:

	type	mode	month
0	casual	0.03	Apr
1	member	4.30	Apr

In [77]:
```
df_m_may = pd.read_csv('df_mode_diff5_may.csv')
df_m_may
```

Out[77]:

	type	mode	month
0	casual	9.58	May
1	member	4.38	May

In [78]:
```
df_m_jun = pd.read_csv('df_mode_diff6_jun.csv')
df_m_jun
```

Out[78]:

	type	mode	month
0	casual	8.42	Jun
1	member	5.38	Jun

In [79]:
```
df_m_jul = pd.read_csv('df_mode_diff7_jul.csv')
df_m_jul
```

Out[79]:

	type	mode	month
0	casual	7.9	Jul
1	member	5.4	Jul

In [80]:
```
df_m_aug = pd.read_csv('df_mode_diff8_aug.csv')
df_m_aug
```

Out[80]:

	type	mode	month
0	casual	8.07	Aug
1	member	6.63	Aug

In [81]:
```
df_m_sep = pd.read_csv('df_mode_diff9_sep.csv')
df_m_sep
```

Out[81]:

	type	mode	month
0	casual	7.63	Sep
1	member	4.72	Sep

In [82]:
```
df_m_oct = pd.read_csv('df_mode_diff10_oct.csv')
df_m_oct
```

Out[82]:

	type	mode	month
0	casual	6.03	Oct
1	member	3.90	Oct

In [83]:
```
df_m_nov = pd.read_csv('df_mode_diff11_nov.csv')
df_m_nov
```

Out[83]:

	type	mode	month
0	casual	5.40	Nov
1	member	4.57	Nov

In [84]:
```
df_m_dec = pd.read_csv('df_mode_diff12_dec.csv')
df_m_dec
```

Out[84]:

	type	mode	month
0	casual	4.71	Dec
1	member	4.48	Dec

In [85]:
```
# Create a dataframe with the values for each month
frames1 = [df_m_jan,df_m_feb,df_m_mar,df_m_apr,df_m_may,df_m_jun,df_m_jul,df_
           df_m_nov,df_m_dec]
df1 = pd.concat(frames1)
df1.head(3)
```

Out[85]:

	type	mode	month
0	casual	0.17	Jan
1	member	4.33	Jan
0	casual	0.03	Feb

In [86]:
```
# Plot the most frequent trip duration per users in minutes
fig1 = px.bar(df1, x="month", y="mode",color='type',width=800, height=400, ba
             title='Most Frequent Trip Duration in minutes')
fig1.show()
```