Part I - Communicate data findings: Ford GoBike System Data

by Mame Diarra NIANE

Introduction

> Bike-sharing systems have become increasingly popular in recent years as a sustainable and affordable means of transportation in urban areas. The Ford GoBike bike-sharing system in San Francisco's Bay Area is one such example, providing a convenient and efficient way to travel around the city. In this project, we will perform exploratory data analysis (EDA) on the usage data of the Ford GoBike system for February 2019.

> The data set includes information that covers over 180K entries of individual rides made in a bike-sharing system covering the greater San Francisco Bay area at the period of February 2019.

Preliminary Wrangling

In [1]:
```
# import all packages and set plots to be embedded inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import datetime as dt
import pandas_profiling as pp
```

```
%matplotlib inline
```

In [2]:
```
#Load the dataset
df_bike=pd.read_csv('201902-fordgobike-tripdata.csv')
df_bike.head()
```

Out[2]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	52185	2019-02-28 17:32:10.1450	2019-03-01 08:01:55.9750	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896
1	42521	2019-02-28 18:53:21.7890	2019-03-01 06:42:03.0560	23.0	The Embarcadero at Steuart St	37.7914
2	61854	2019-02-28 12:13:13.2180	2019-03-01 05:24:08.1460	86.0	Market St at Dolores St	37.7693
3	36490	2019-02-28 17:54:26.0100	2019-03-01 04:02:36.8420	375.0	Grove St at Masonic Ave	37.7748
4	1585	2019-02-28 23:54:18.5490	2019-03-01 00:20:44.0740	7.0	Frank H Ogawa Plaza	37.8045

In [3]:
```
df_bike.shape
```

Out[3]:
```
(183412, 16)
```

In [4]:
```
df_bike.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183412 entries, 0 to 183411
Data columns (total 16 columns):
 #   Column                   Non-Null Count   Dtype  
---  ------                   --------------   -----  
 0   duration_sec             183412 non-null  int64  
 1   start_time               183412 non-null  object 
 2   end_time                 183412 non-null  object 
 3   start_station_id         183215 non-null  float64
 4   start_station_name       183215 non-null  object 
 5   start_station_latitude   183412 non-null  float64
 6   start_station_longitude  183412 non-null  float64
 7   end_station_id           183215 non-null  float64
 8   end_station_name         183215 non-null  object 
 9   end_station_latitude     183412 non-null  float64
 10  end_station_longitude    183412 non-null  float64
 11  bike_id                  183412 non-null  int64  
 12  user_type                183412 non-null  object 
 13  member_birth_year        175147 non-null  float64
 14  member_gender            175147 non-null  object 
 15  bike_share_for_all_trip  183412 non-null  object 
dtypes: float64(7), int64(2), object(7)
memory usage: 22.4+ MB
```

In [5]:
```
df_bike.describe()
```

Out[5]:

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id
count	183412.000000	183215.000000	183412.000000	183412.000000	183215.000000
mean	726.078435	138.590427	37.771223	-122.352664	136.249123
std	1794.389780	111.778864	0.099581	0.117097	111.515131
min	61.000000	3.000000	37.317298	-122.453704	3.000000
25%	325.000000	47.000000	37.770083	-122.412408	44.000000
50%	514.000000	104.000000	37.780760	-122.398285	100.000000
75%	796.000000	239.000000	37.797280	-122.286533	235.000000
max	85444.000000	398.000000	37.880222	-121.874119	398.000000

We can see that he maximum "duration_sec" is very high . we are going to check if there is more longest trip like that in the dataset

In [6]:
```
df_bike.nlargest(20,'duration_sec')
```

Out[6]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station
101361	85444	2019-02-13 17:59:55.1240	2019-02-14 17:43:59.9540	5.0	Powell St BART Station (Market St at 5th St)	3
85465	84548	2019-02-16 15:48:25.0290	2019-02-17 15:17:33.0800	3.0	Powell St BART Station (Market St at 4th St)	3
153705	83772	2019-02-05 12:44:54.2860	2019-02-06 12:01:06.6310	78.0	Folsom St at 9th St	
127999	83519	2019-02-09 15:16:17.5370	2019-02-10 14:28:17.2700	72.0	Page St at Scott St	3
112435	83407	2019-02-11 16:25:33.0690	2019-02-12 15:35:40.9560	77.0	11th St at Natoma St	3
5203	83195	2019-02-27 14:47:23.1810	2019-02-28 13:53:58.4330	243.0	Bancroft Way at College Ave	3
95750	82512	2019-02-14 13:56:21.7280	2019-02-15 12:51:34.3150	368.0	Myrtle St at Polk St	3
173365	82385	2019-02-02 13:07:43.0360	2019-02-03 12:00:48.1750	377.0	Fell St at Stanyan St	
8631	81549	2019-02-27 09:41:38.5520	2019-02-28 08:20:48.3860	138.0	Jersey St at Church St	3
176987	80891	2019-02-01 11:05:18.9760	2019-02-02 09:33:30.1690	44.0	Civic Center/UN Plaza BART Station (Market St ...	
107581	79548	2019-02-12 17:45:50.5360	2019-02-13 15:51:38.8590	79.0	7th St at Brannan St	3
94581	75262	2019-02-14 19:25:02.3820	2019-02-15 16:19:24.9570	200.0	2nd Ave at E 18th St	3
90195	74408	2019-02-15 16:54:01.0600	2019-02-16 13:34:09.3670	3.0	Powell St BART Station (Market St at 4th St)	3
86454	74097	2019-02-16 16:20:41.4650	2019-02-17 12:55:38.4670	99.0	Folsom St at 15th St	3
123383	73930	2019-02-10 13:03:36.4040	2019-02-11 09:35:46.4460	270.0	Ninth St at Heinz Ave	3
73587	72824	2019-02-18 17:10:52.9390	2019-02-19 13:24:37.5570	14.0	Clay St at Battery St	3
129176	72627	2019-02-09 15:15:59.2380	2019-02-10 11:26:26.3300	72.0	Page St at Scott St	3
116671	72590	2019-02-11 11:26:36.9850	2019-02-12 07:36:27.3610	39.0	Scott St at Golden Gate Ave	3
129177	72576	2019-02-09 15:16:26.2830	2019-02-10 11:26:02.5440	72.0	Page St at Scott St	3
136599	72361	2019-02-07 13:37:01.7560	2019-02-08 09:43:03.2850	66.0	3rd St at Townsend St	3

`there's indeed many bikes trip with long ride`

In [7]:
```
df_bike['start_station_name'].nunique()
```

Out[7]: 329

In [8]:
```
#check the missing values in the dataset
df_bike.isnull().sum()
```

Out[8]:
```
duration_sec                  0
start_time                    0
end_time                      0
start_station_id            197
start_station_name          197
start_station_latitude        0
start_station_longitude       0
end_station_id              197
end_station_name            197
end_station_latitude          0
end_station_longitude         0
bike_id                       0
user_type                     0
member_birth_year          8265
member_gender              8265
bike_share_for_all_trip       0
dtype: int64
```

In [9]:
```
df_bike.columns
```

Out[9]:
```
Index(['duration_sec', 'start_time', 'end_time', 'start_station_id',
       'start_station_name', 'start_station_latitude',
       'start_station_longitude', 'end_station_id', 'end_station_name',
       'end_station_latitude', 'end_station_longitude', 'bike_id', 'user_type',
       'member_birth_year', 'member_gender', 'bike_share_for_all_trip'],
      dtype='object')
```

In [10]:
```
#Check for duplicated rows
df_bike.duplicated().sum()
```

Out[10]:
```
0
```

In [11]:
```
#check the differents usertypes and their numbers
df_bike.user_type.value_counts()
```

Out[11]:
```
Subscriber    163544
Customer       19868
Name: user_type, dtype: int64
```

Observations:

> - the Data consists of 183412 rows and 16 columns
> - start time and end time are not datetime
> - Missing values in the **start_station_id , start_station_name ,end_station_id ,end_station_name,member_birth_year and member_gender** columns
> - There's no duplicates in the dataset
> - The minimum member_birth_year is out of range(humanly impossible)

Data cleaning issues

1. Erroneous data type in the **start_time ,end_time (datetime instead of object)** and the **user_type** must be a categorical type
2. Missing values in the **start_station_id , start_station_name ,end_station_id ,end_station_name,member_birth_year and member_gender** columns

3. integrate **start_day,start_day_name,start_hour and end_day,end_day_name,end_hour** column in the dataset
4. Drop columns that i didn't need for analysis
5. Create a **member_age** column and drop the **member_birth** year column
6. Change the trip duration unit from seconds to minute for convenience

Data Cleaning

In [12]:
```
#Make a copy of the original data before cleaning
df_bike_clean=df_bike.copy()
```

> 1. Erroneous data type in the **start_time ,end_time (datetime instead of object)** and the **user_type** must be a categorical type , **bike_id** should be string

In [13]:
```
#Convert start_time and end_time  into datetime type
for column in ['start_time','end_time']:
    df_bike_clean[column]=df_bike_clean[column].astype('datetime64[ns]')
df_bike_clean.dtypes
```

Out[13]:
```
duration_sec                        int64
start_time                 datetime64[ns]
end_time                   datetime64[ns]
start_station_id                  float64
start_station_name                 object
start_station_latitude            float64
start_station_longitude           float64
end_station_id                    float64
end_station_name                   object
end_station_latitude              float64
end_station_longitude             float64
bike_id                             int64
user_type                          object
member_birth_year                 float64
member_gender                      object
bike_share_for_all_trip            object
dtype: object
```

In [14]:
```
#convert user_type to category type
df_bike_clean['user_type']=pd.Categorical(df_bike_clean.user_type)
```

In [15]:
```
#convert bike_id to object type
df_bike_clean['bike_id']=df_bike_clean['bike_id'].astype('str')
```

> 2. Remove all the rows with missing values

In [16]:
```
#Remove missing values in the dataset
df_bike_clean=df_bike_clean.dropna()
df_bike_clean.isnull().sum()
```

Out[16]:
```
duration_sec               0
start_time                 0
end_time                   0
start_station_id           0
start_station_name         0
start_station_latitude     0
start_station_longitude    0
end_station_id             0
end_station_name           0
end_station_latitude       0
end_station_longitude      0
bike_id                    0
user_type                  0
member_birth_year          0
member_gender              0
bike_share_for_all_trip    0
dtype: int64
```

> **member_birth_year** should be an int

In [17]:
```
#convert member_birth_year to int type
df_bike_clean['member_birth_year']=df_bike_clean['member_birth_year'].astype('int')
```

In [18]:
```
df_bike_clean.dtypes
```

Out[18]:
```
duration_sec                    int64
start_time                 datetime64[ns]
end_time                   datetime64[ns]
start_station_id                  float64
start_station_name                 object
start_station_latitude            float64
start_station_longitude           float64
end_station_id                    float64
end_station_name                   object
end_station_latitude              float64
end_station_longitude             float64
bike_id                            object
user_type                        category
member_birth_year                   int64
member_gender                      object
bike_share_for_all_trip            object
dtype: object
```

In [19]:
```
df_bike_clean.describe()
```

Out[19]:

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id
count	174952.000000	174952.000000	174952.000000	174952.000000	174952.000000
mean	704.002744	139.002126	37.771220	-122.351760	136.604486
std	1642.204905	111.648819	0.100391	0.117732	111.335635
min	61.000000	3.000000	37.317298	-122.453704	3.000000
25%	323.000000	47.000000	37.770407	-122.411901	44.000000
50%	510.000000	104.000000	37.780760	-122.398279	101.000000
75%	789.000000	239.000000	37.797320	-122.283093	238.000000
max	84548.000000	398.000000	37.880222	-121.874119	398.000000

> Remove the columns not needed in the analysis(**start_station_id ,start_station_latitude**

,start_station_longitude,end_station_latitude,end_station_longitude,bike_share_for_all_
)

In [20]:
```
no_needed_columns=['start_station_id' ,'start_station_latitude' ,'start_station_longi
df_bike_clean.drop(no_needed_columns,axis=1,inplace=True)
```

In [21]:
```
df_bike_clean.columns
```

Out[21]:
```
Index(['duration_sec', 'start_time', 'end_time', 'start_station_name',
       'end_station_name', 'bike_id', 'user_type', 'member_birth_year',
       'member_gender'],
      dtype='object')
```

Create the columns **start_day,start_day_name,start_hour and end_day,end_day_name,end_hour** in the dataset

In [22]:
```
df_bike_clean['start_day'] = df_bike_clean['start_time'].dt.day
df_bike_clean['start_day_name'] = df_bike_clean['start_time'].dt.day_name()
df_bike_clean['start_hour'] = df_bike_clean['start_time'].dt.hour
#df_bike_clean['start_time']=df_bike_clean['start_time'].dt.time

df_bike_clean['end_day'] = df_bike_clean['end_time'].dt.day
df_bike_clean['end_day_name'] = df_bike_clean['end_time'].dt.day_name()
df_bike_clean['end_hour'] = df_bike_clean['end_time'].dt.hour
```

In [23]:
```
df_bike_clean.dtypes
```

Out[23]:
```
duration_sec                int64
start_time         datetime64[ns]
end_time           datetime64[ns]
start_station_name         object
end_station_name           object
bike_id                    object
user_type                category
member_birth_year           int64
member_gender              object
start_day                   int64
start_day_name             object
start_hour                  int64
end_day                     int64
end_day_name               object
end_hour                    int64
dtype: object
```

5. Create a **member_age** column and drop the **member_birth_year** column

In [24]:
```
df_bike_clean['member_age'] = 2019-df_bike_clean.member_birth_year
df_bike_clean=df_bike_clean.drop(['member_birth_year'],axis=1)
```

In [25]:
```
df_bike_clean.sample(10)
```

Out[25]:

	duration_sec	start_time	end_time	start_station_name	end_station_name	bike_id	use
80489	509	2019-02-18 15:29:44.355	2019-02-18 15:38:13.615	Howard St at Mary St	Valencia St at Clinton Park	5164	Sub
14870	536	2019-02-27 10:44:27.439	2019-02-27 10:53:23.680	Market St at 10th St	Irwin St at 8th St	5801	Sub
40327	311	2019-02-22 21:21:08.239	2019-02-22 21:26:19.905	MacArthur BART Station	Shattuck Ave at 55th St	5176	Sub
10981	1805	2019-02-27 18:55:55.858	2019-02-27 19:26:01.150	MLK Jr Way at University Ave	Genoa St at 55th St	6452	Sub
174510	469	2019-02-02 17:51:05.435	2019-02-02 17:58:54.947	Hearst Ave at Euclid Ave	Parker St at Fulton St	4909	Cu
123834	427	2019-02-11 09:01:32.318	2019-02-11 09:08:39.883	San Francisco Caltrain (Townsend St at 4th St)	8th St at Ringold St	3729	Cu
33710	336	2019-02-24 10:53:38.168	2019-02-24 10:59:14.328	Webster St at 2nd St	Frank H Ogawa Plaza	4853	Sub
67876	1495	2019-02-20 07:02:05.463	2019-02-20 07:27:00.720	Scott St at Golden Gate Ave	Montgomery St BART Station (Market St at 2nd St)	4725	Cu
45136	139	2019-02-22 13:01:15.121	2019-02-22 13:03:34.639	Bancroft Way at College Ave	Haste St at Telegraph Ave	5258	Sub
569	889	2019-02-28 20:54:11.919	2019-02-28 21:09:01.301	San Francisco Ferry Building (Harry Bridges Pl...	Mississippi St at 17th St	4762	Sub

> 6. Change the trip duration unit from seconds to minute for convenience

In [26]:
```
df_bike_clean['duration_min']=df_bike_clean['duration_sec']/60
df_bike_clean=df_bike_clean.drop(labels='duration_sec',axis=1)
```

In [65]:
```
#Save the Clean file
df_bike_clean.to_csv('201902_fordgobike_clean.csv',index=False)
```

In [66]:
```
df=pd.read_csv('201902_fordgobike_clean.csv')
df.head()
```

Out[66]:

	start_time	end_time	start_station_name	end_station_name	bike_id	user_type	member_gen
0	2019-02-28 17:32:10.145	2019-03-01 08:01:55.975	Montgomery St BART Station (Market St at 2nd St)	Commercial St at Montgomery St	4902	Customer	M
1	2019-02-28 12:13:13.218	2019-03-01 05:24:08.146	Market St at Dolores St	Powell St BART Station (Market St at 4th St)	5905	Customer	M
2	2019-02-28 17:54:26.010	2019-03-01 04:02:36.842	Grove St at Masonic Ave	Central Ave at Fell St	6638	Subscriber	O
3	2019-02-28 23:54:18.549	2019-03-01 00:20:44.074	Frank H Ogawa Plaza	10th Ave at E 15th St	4898	Subscriber	M
4	2019-02-28 23:49:58.632	2019-03-01 00:19:51.760	4th St at Mission Bay Blvd S	Broadway at Kearny	5200	Subscriber	M

In [67]:
```
df_bike_clean.describe()
```

Out[67]:

	start_day	start_hour	end_day	end_hour	member_age	duration_min
count	174875.000000	174875.000000	174875.000000	174875.000000	174875.000000	174875.000000
mean	15.312149	13.456172	15.311525	13.609550	34.160274	11.733800
std	8.033935	4.734320	8.034136	4.748068	9.968641	27.375632
min	1.000000	0.000000	1.000000	0.000000	18.000000	1.016667
25%	8.000000	9.000000	8.000000	9.000000	27.000000	5.383333
50%	15.000000	14.000000	15.000000	14.000000	32.000000	8.500000
75%	22.000000	17.000000	22.000000	18.000000	39.000000	13.150000
max	28.000000	23.000000	28.000000	23.000000	89.000000	1409.133333

What is the structure of your dataset?

> The datasset includes 183412 individuals bikeride in february 2019 in the greater San Francisco Bay area with 16 features among which there is :
>
> - **duration_sec:** the duartion of the trip in seconds
> - **start_time:** the time the bikeride started
> - **end_time:** the time the bikeride ended
> - **start_station_name:** the name of the station departure
> - **end_station_name:** the name of the arrival station
> - **user_type:** if the user is a subscriber or a customer
> - **member_gender:** the gender of the user

What is/are the main feature(s) of interest in your dataset?

Features of interest :

- duration_min
- member_age
- member_gender
- user_type
- time features

What features in the dataset do you think will help support your investigation into your feature(s) of interest?

> I expect that the age and the time will have the strongest effect in the duration of trip. Like the Younger the user the highest the trip duration. However the gender may have another effect on the duration but we will see all of this at the end of our analysis

> - How is distributed the trip duration ?
> - Which days are the busiest in this San Fracisco bike sharing?
> - The busiest hours?
> - How the others features affects the trip duration ?

Univariate Exploration

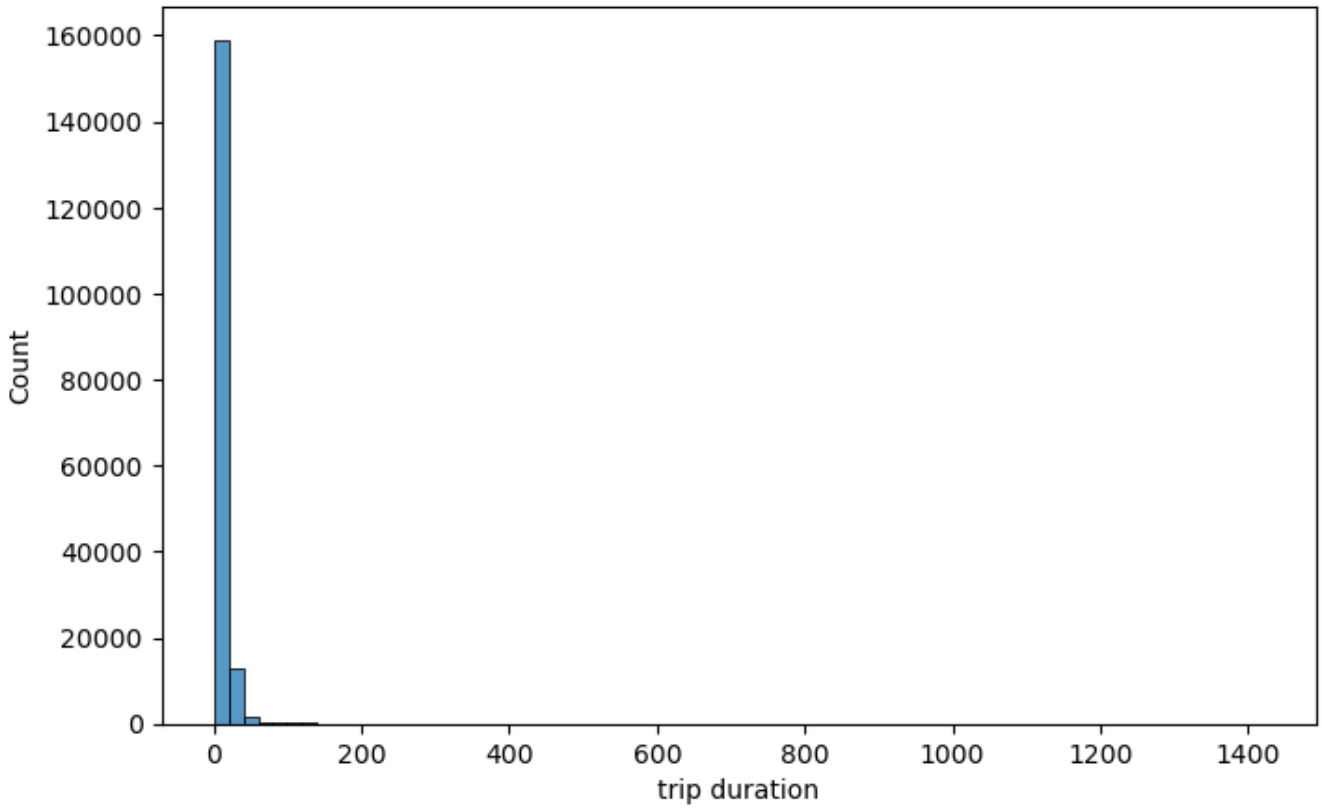
I'll start by looking at the distribution of the main variable of interest: **duration_sec.**

Question

`What is the distribution of the duration ?`

Visualization

In [30]:
```
binsize=20
default_color = sns.color_palette()[0]
bins=np.arange(0,df_bike_clean['duration_min'].max()+binsize,binsize)
plt.figure(figsize=[8,5])
sns.histplot(data=df_bike_clean,x ='duration_min',bins=bins,color=default_color)
plt.xlabel('trip duration')
plt.show()
```

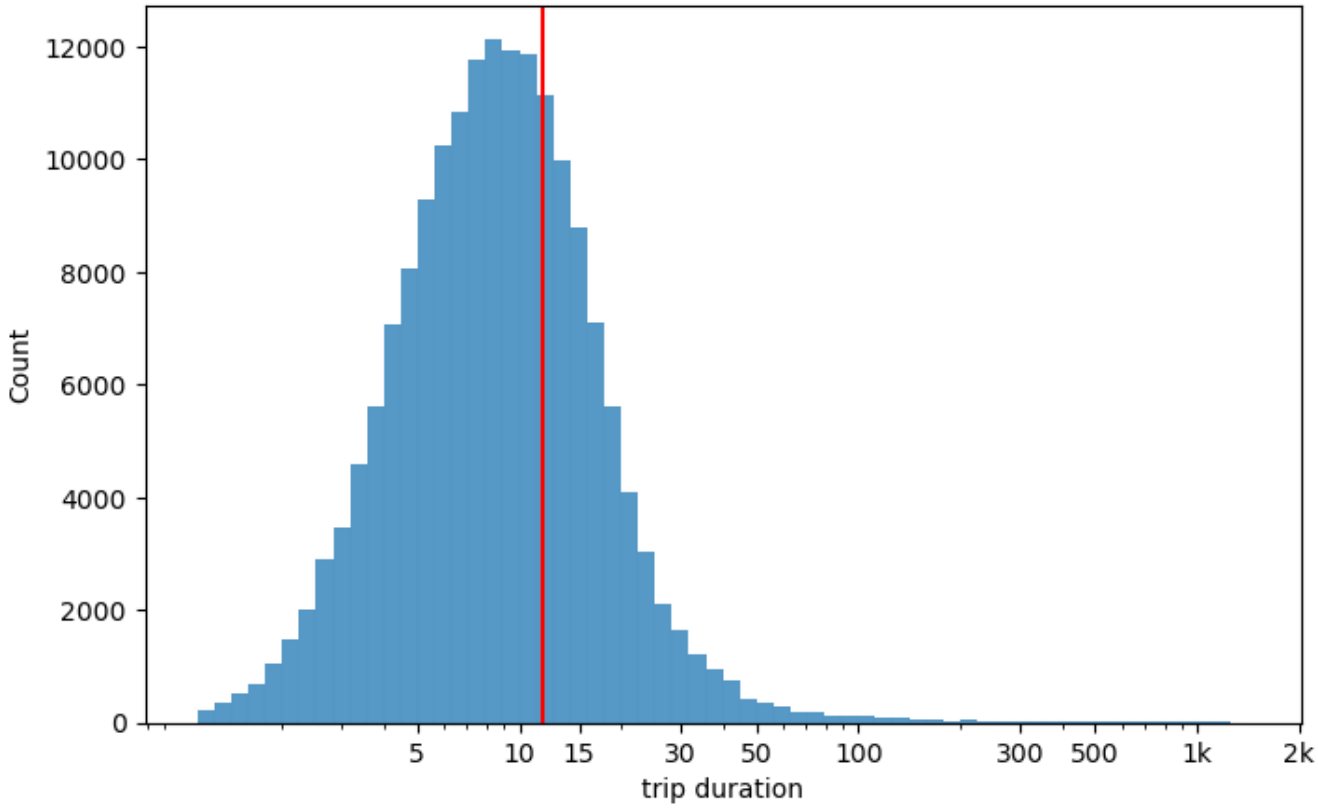


the data would be better viewed in a logarythmic scale

In [31]:
```
log_binsize = 0.05
bins = 10 ** np.arange(0.05, np.log10(df_bike_clean['duration_min'].max()) + log_bins
plt.figure(figsize=[8,5])
#plt.hist(data = df_bike_clean,x ='duration_min',bins = bins)
sns.histplot(data=df_bike_clean,x ='duration_min',bins=bins,color=default_color)


plt.xscale('log')
plt.xticks([5,10,15,30 ,50, 100,300,500, 1e3, 2e3], [5,10,15,30,50,100,300,500, '1k',
plt.xlabel('trip duration')
plt.axvline(x=df_bike_clean.duration_min.mean(),color='red')

plt.show()
```



In [32]:
```
df_bike_clean.duration_min.describe()
```

Out[32]:
```
count    174952.000000
mean         11.733379
std          27.370082
min           1.016667
25%           5.383333
50%           8.500000
75%          13.150000
max        1409.133333
Name: duration_min, dtype: float64
```

Observations

The histograms shows that most of the bikerides took between 5 mins and 15mins and a fews rides was more then way more than 300mins . it might because the users didnt log off after their rides for differentes reasons

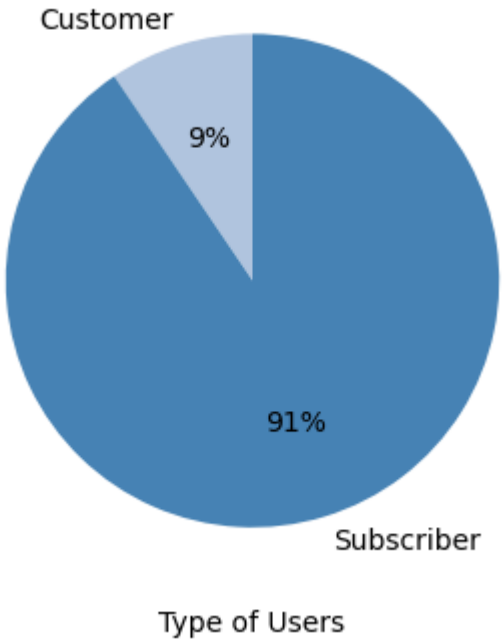
The average duration is 11 minutes

Question

How the Gobike fleet is distributed between customers and subscribers?

Visualization

In [33]:
```
#Plot a pie chart for the user type distribution
plt.figure(figsize=[6,4])
default_color = sns.color_palette()[0]
df_bike_clean.groupby('user_type')\
.size().plot(kind='pie',startangle=90,autopct='%1.0f%%',textprops= {'fontsize':10},co
plt.ylabel('',size=5)
plt.xlabel('Type of Users')
plt.show()
```



Observations

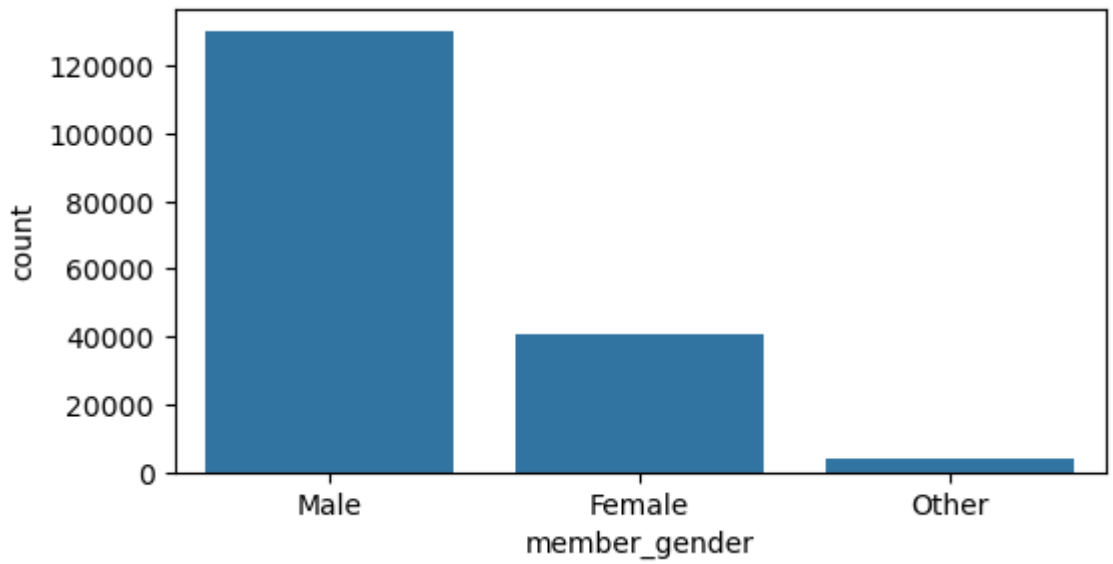
We can say that the majority of the bikerides in february 2019 were done by subscribers . Only 9% of them were made by customers.

Question

How the usage of the Gobike Sharing is distributed gender wise?

Visualization

In [34]:
```
default_color=sns.color_palette()[0]
plt.figure(figsize=[6,3])
sns.countplot(data=df_bike_clean,x='member_gender',color=default_color,order=df_bike_
plt.show()
```



Observations

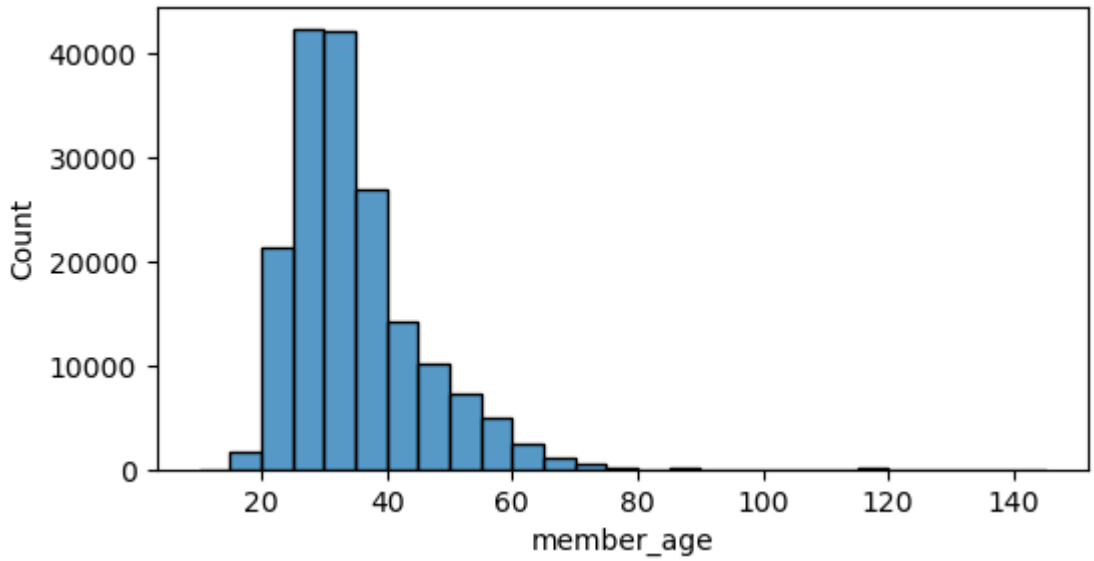
There's is clearly more male bike users in this month of february

Question

Let's looks for outlier's in the users age

Visualization

In [35]:
```
default_color=sns.color_palette()[0]
plt.figure(figsize=[6,3])
sns.histplot(data=df_bike_clean,x='member_age',bins=range(10,df_bike_clean.member_age
plt.show()
```

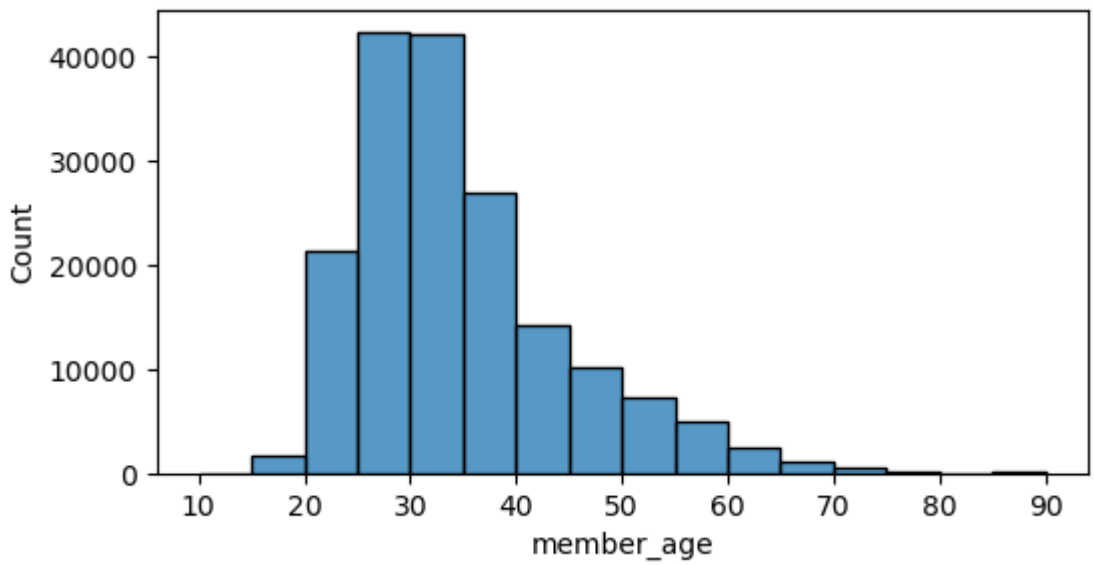


Observations

The distribution is right skewed. Most of the users have between 25 and 35 .There's clearly users with age that is not usual for bikeride so i decide that i will remove the users with an age >90.

In [36]:
```
#Remove members with age >90 in the member_age column
df_bike_clean=df_bike_clean.drop(df_bike_clean[df_bike_clean.member_age>90].index)
```

In [37]:
```
#Let's look at the new age distribution
default_color=sns.color_palette()[0]
plt.figure(figsize=[6,3])
sns.histplot(data=df_bike_clean,x='member_age',bins=range(10,df_bike_clean.member_age
plt.show()
```



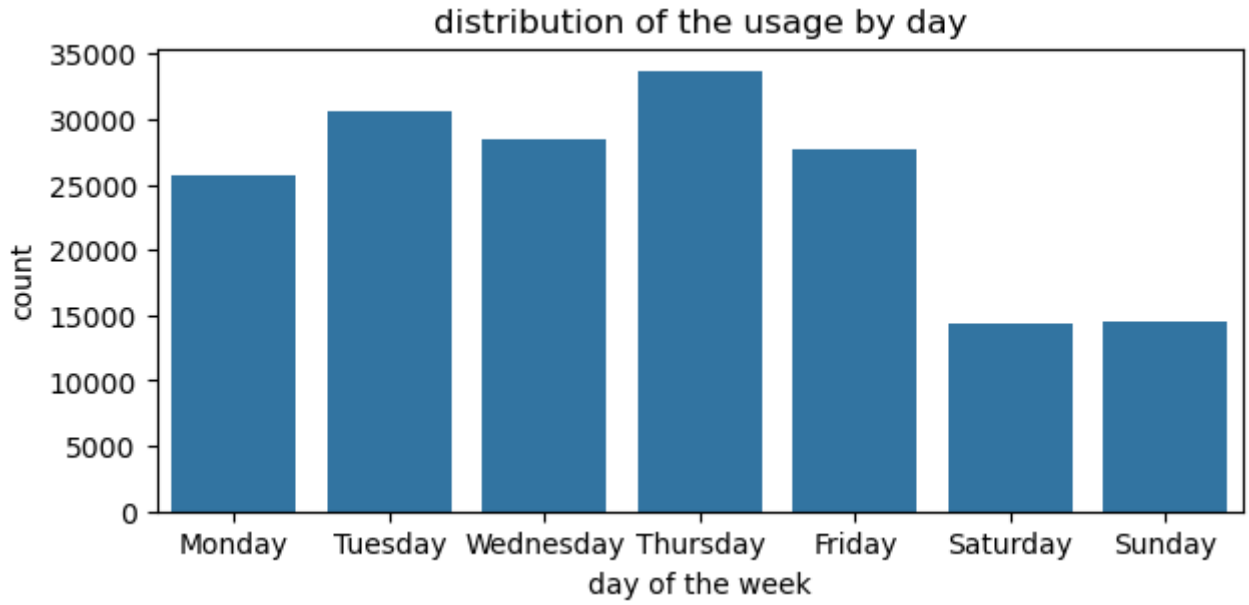
Question

what day has the most bike users?

Visualization

In [64]:
```
#convert the start_day_name into a ordered categorical type
days=['Monday','Tuesday','Wednesday','Thursday','Friday','Saturday','Sunday']
ordered_var=pd.api.types.CategoricalDtype(ordered=True,categories=days)
df_bike_clean['start_day_name']=df_bike_clean['start_day_name'].astype(ordered_var)
df_bike_clean['end_day_name']=df_bike_clean['end_day_name'].astype(ordered_var)
```

In [39]:
```
default_color=sns.color_palette()[0]
plt.figure(figsize=[7,3])
sns.countplot(data=df_bike_clean,x='start_day_name',color=default_color)
plt.xlabel('day of the week')
plt.title('distribution of the usage by day')
plt.show()
```



Observations

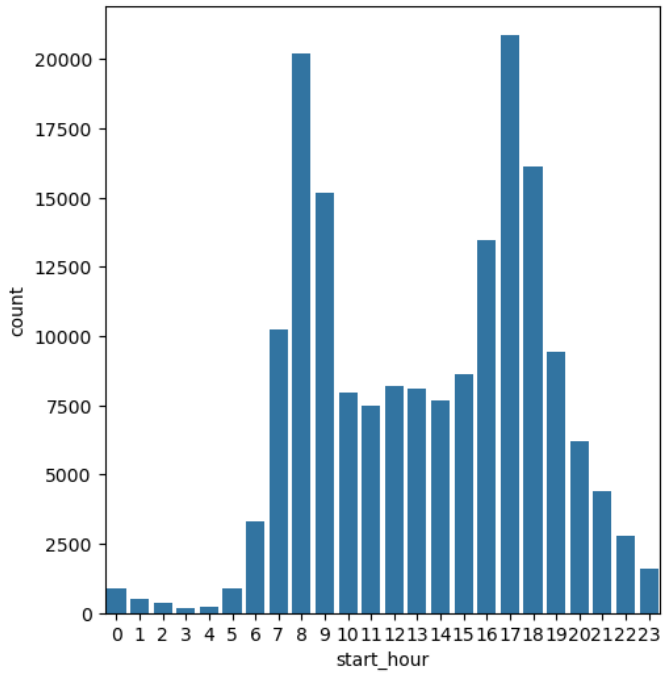
Bike are used the most on Thursday and Tuesday.

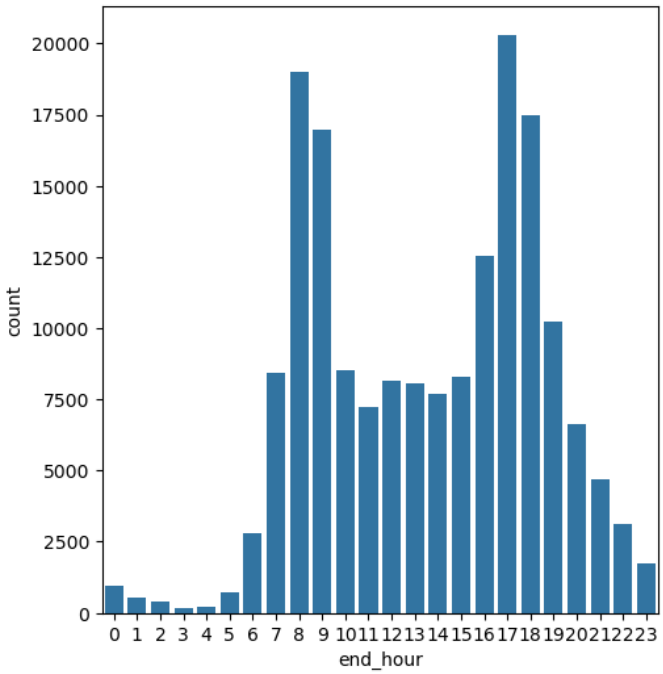
Question

what time of the day there's the most usage of the bikes?

Visualization

In [40]:
```
fig,ax=plt.subplots(1,2,figsize=[12,6])
default_color=sns.color_palette()[0]
sns.countplot(data=df_bike_clean,x='start_hour',color=default_color,ax=ax[0])
sns.countplot(data=df_bike_clean,x='end_hour',color=default_color,ax=ax[1])
plt.show()
```





Observations

Bike are used the most during rush hours. 8:00 in the morning and 17:00 to 18:00 .

Question

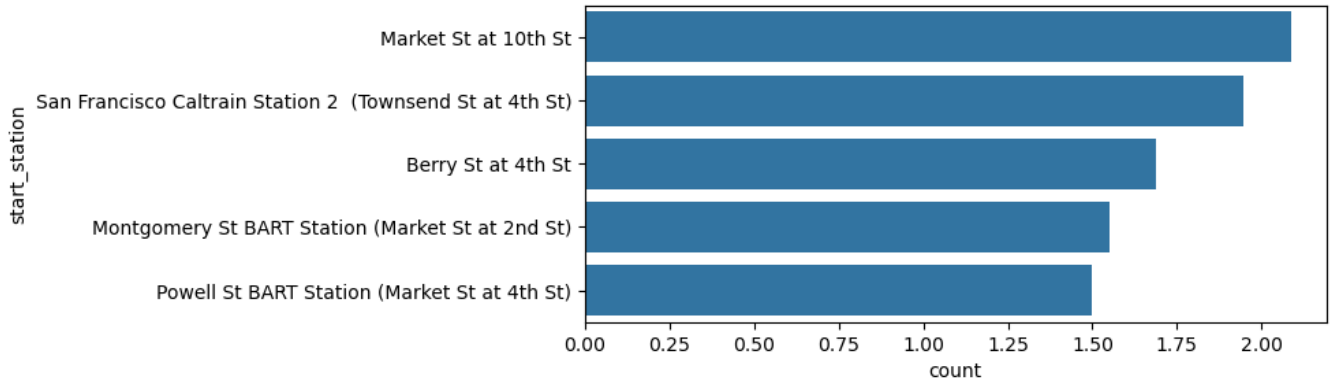
What is the most used stations?

Visualization

In [41]:
```
s=(df_bike_clean.start_station_name.value_counts(normalize=True)*100).nlargest(5)
s
```

Out[41]:
```
Market St at 10th St                                         2.086633
San Francisco Caltrain Station 2  (Townsend St at 4th St)    1.947677
Berry St at 4th St                                           1.687491
Montgomery St BART Station (Market St at 2nd St)             1.549107
Powell St BART Station (Market St at 4th St)                 1.498213
Name: start_station_name, dtype: float64
```

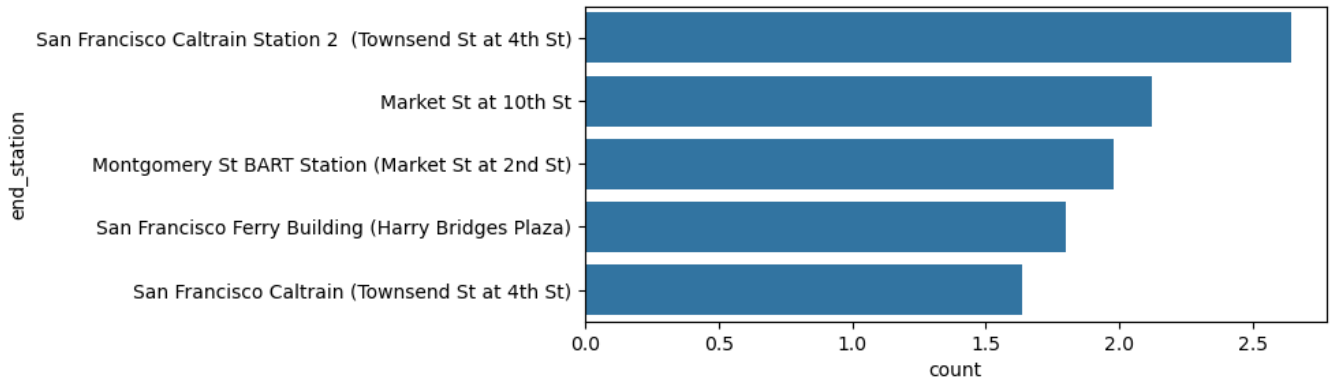
In [42]:
```
plt.figure(figsize=[7,3])
default_color=sns.color_palette()[0]
s=(df_bike_clean.start_station_name.value_counts(normalize=True)*100).nlargest(5)
s=s.rename_axis("start_station").reset_index(name="count")
sns.barplot(data=s,x="count",y="start_station",color=default_color)
plt.show()
```



In [43]:
```
m=(df_bike_clean.end_station_name.value_counts(normalize=True)*100).nlargest(5)
m
```

Out[43]:
```
San Francisco Caltrain Station 2  (Townsend St at 4th St)    2.642459
Market St at 10th St                                         2.120944
Montgomery St BART Station (Market St at 2nd St)             1.978556
San Francisco Ferry Building (Harry Bridges Plaza)           1.801858
San Francisco Caltrain (Townsend St at 4th St)               1.635454
Name: end_station_name, dtype: float64
```

In [44]:
```
plt.figure(figsize=[7,3])
default_color=sns.color_palette()[0]
m=(df_bike_clean.end_station_name.value_counts(normalize=True)*100).nlargest(5)
m=m.rename_axis("end_station").reset_index(name="count")
sns.barplot(data=m,x="count",y="end_station",color=default_color)
plt.show()
```



Observations

`Market St at 10th St is the busiest bike start station . San Francisco Caltrain Station 2 (Townsend St at 4th St) is the busiest end station`

Discuss the distribution(s) of your variable(s) of interest. Were there any unusual points? Did you need to perform any transformations?

> Most of the bikerides took between 5min and 15min and there were unsual point in the duration However i decided to not perform any transformation in the duration since there are maybe some users that can rent a bike for long hours

Of the features you investigated, were there any unusual distributions? Did you perform any operations on the data to tidy, adjust, or change the form of the data? If so, why did you do this?

> Nothing seems unsual except the age in which i did transformation by removing users where the age was > 75 because after this age it seems unsual to see people with a bike . There can be exception as always.

> we also see that Mens use the most the bike rental in sanFrancisco

> i've added more columns (**start_day ,start_hour,start_day_name**)from the start_time and end_time columns (**end_day ,end_hour,end_day_name**) in order look in more details to the data

> the usage is more frequent on thursday at the rush hour (8:00 am and 5:00 pm) and the busiest start station is Market St at 10th St while the busiest end station is San Francisco Caltrain Station 2 (Townsend St at 4th St)

Bivariate Exploration

Here's the categorical variables in our dataset

start_station_name,end_station_name,user_type,member_gender,start_day_name,end_day_na

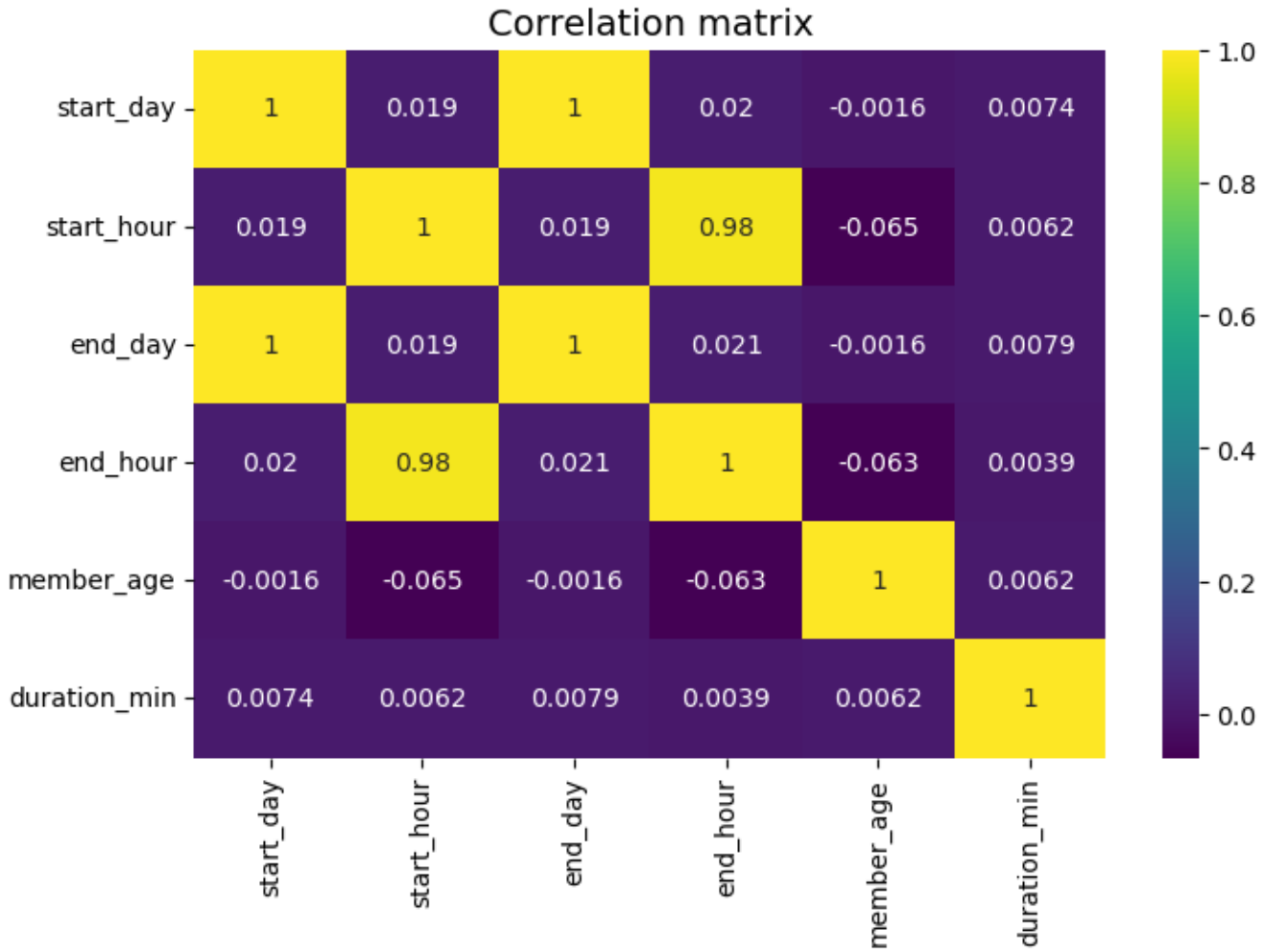
Here's the numeric variables in our dataset

duration_sec,bike_id,start_day,start_hour,end_day,end_hour,member_age

In [45]:
```
numeric_var=['duration_min','bike_id','start_day','start_hour','end_day','end_hour','
needed_numeric_var=['duration_min','member_age']
categorical_var=['start_station_name','end_station_name','user_type','member_gender',
```

In [46]:
```
plt.figure(figsize = (8,5), dpi = 100)

sns.heatmap(df_bike_clean.corr(),cmap="viridis",annot=True)
plt.title("Correlation matrix",fontsize=14)
plt.show()
```

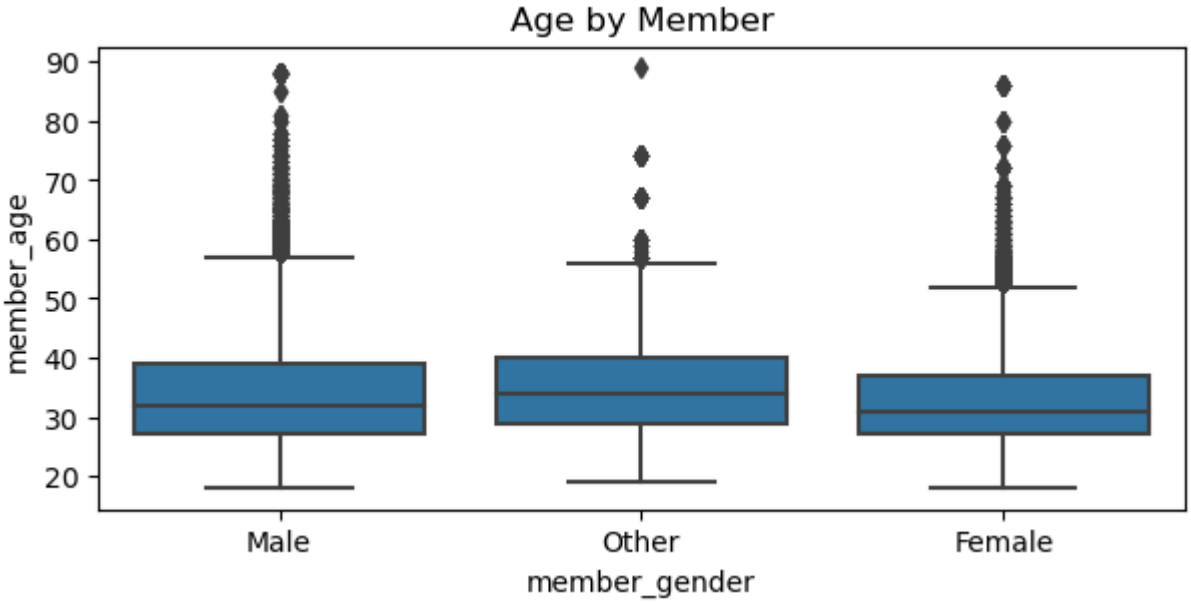


Question

`Relation between member_age and member_genre in the gobike rental usage`

Visualization

In [47]:
```
plt.figure(figsize=[7,3])
default_color=sns.color_palette()[0]
sns.boxplot(data=df_bike_clean,x='member_gender',y='member_age',color=default_color).
plt.show()
```



Observations

- With this boxplot that show the distribution of the member age by gender ,we can see the median age of the 'other' is about 35 and the median age of male users is about 33 and female is 30

- We can see the central 50% age range of male is between 28 and 39 while the 50% central range of female is between 28 and 37.

- we can see that the youngest female is 18 and the youngest male is 18 too .

Question

Relation between the trip duration ,the member age and the genre and user_type

Visualization

In [48]:
```
round((df_bike_clean.shape[0]*1)/100)
```

Out[48]: 1749

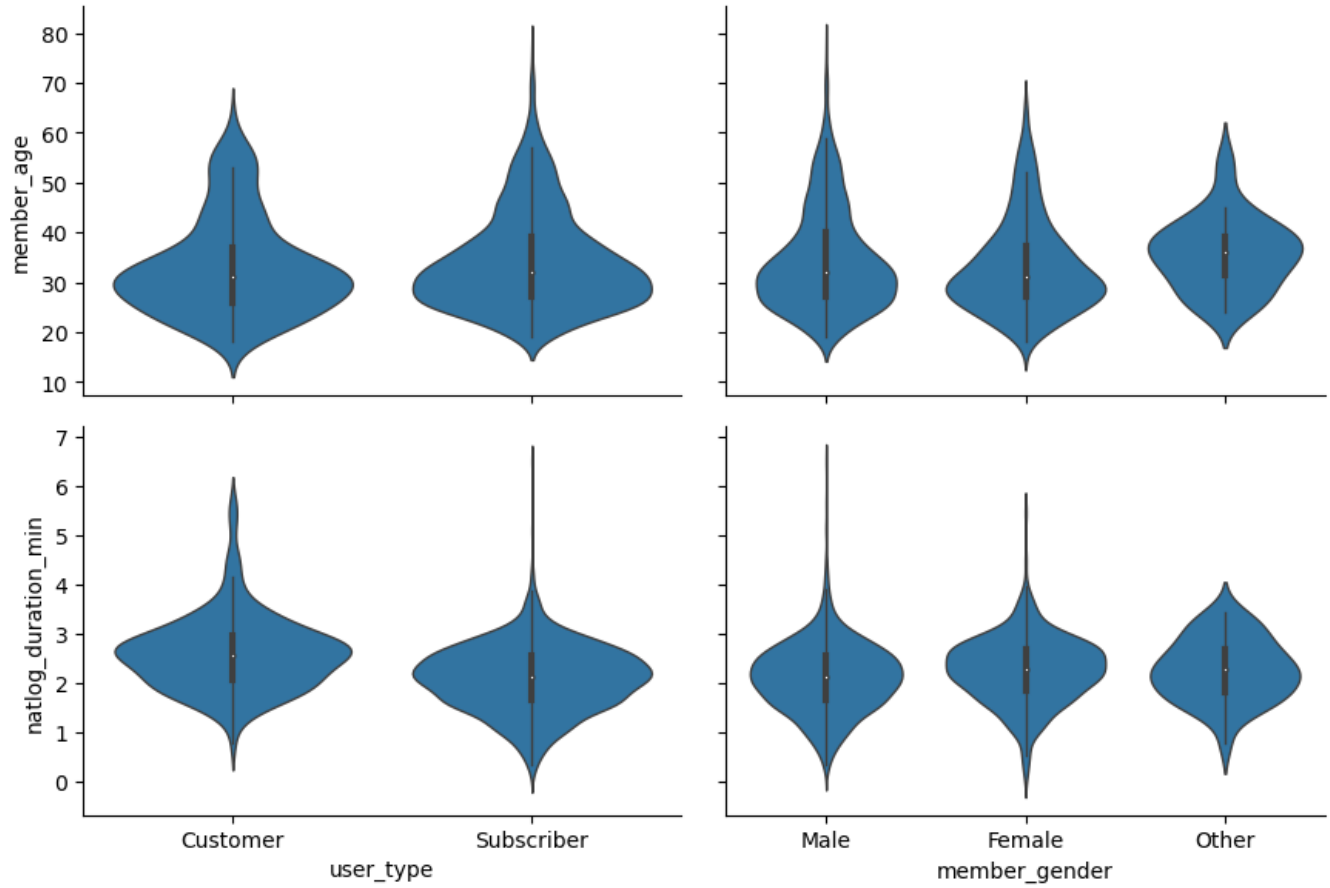
In [49]:
```
plt.figure(figsize=[7,5])
default_color=sns.color_palette()[0]
bike_sample= df_bike_clean.sample(n=round((df_bike_clean.shape[0]*1)/100), replace =
bike_sample['natlog_duration_min']=np.log(bike_sample['duration_min'])

bike_sample = bike_sample.reset_index()


g = sns.PairGrid(bike_sample , x_vars = ['user_type','member_gender'],\
                 y_vars=['member_age','natlog_duration_min'],height=3,aspect=1.5)

g.map(sns.violinplot,color=default_color,linewidth=1)
plt.show()
```

```
<Figure size 700x500 with 0 Axes>
```



Observations

- Customers bikers have in average longer trip than subscriber bikers
- Female users bike a little bit longer in average than male users .
- the median age of customers is higher than the median age of subscribers
- the median age of male is slightly higher than the median age of female

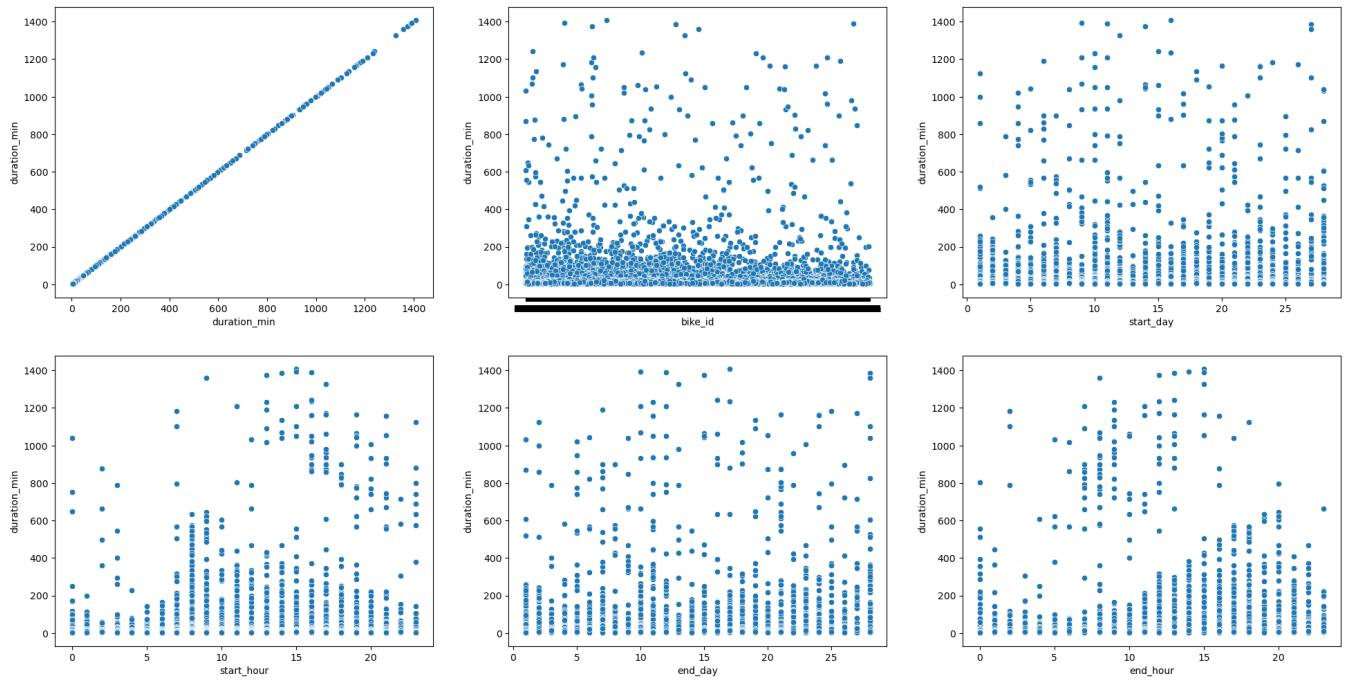
Question

Relation between the others quantitatives variables column() and the duration

Visualization

In [50]:
```
color=sns.color_palette()[0]
nrows,ncols=2,3
var=['start_day','start_hour','end_day','end_hour','member_age']
fig,ax=plt.subplots(2,3,figsize=[24,12])
fig.suptitle("Relation between duration and the others numeric value of the dataset",
for i in range (2):
    for j in range(3):
        sns.scatterplot(data=df_bike_clean,x=df_bike_clean[numeric_var[i*3+j]],\
                        y=df_bike_clean['duration_min'],color=color,ax=ax[i,j])
plt.show()
```

Relation between duration and the others numeric value of the dataset

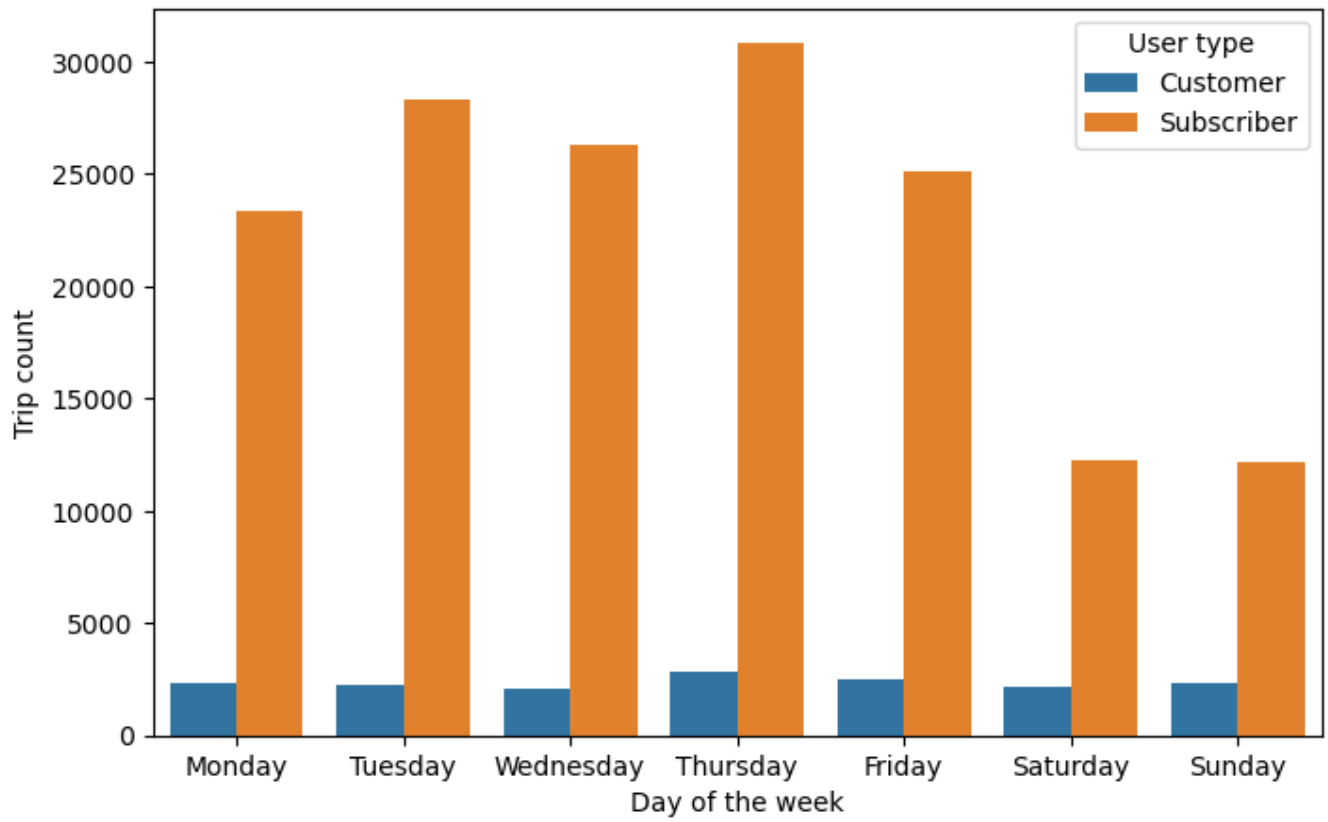


Question

`Relation between the user_type and the days of the week`

Visualization

In [51]:
```
plt.figure(figsize = (8,5), dpi = 100)
sns.color_palette
sns.countplot(x='start_day_name', hue='user_type', data=df_bike_clean)
plt.xlabel('Day of the week')
plt.ylabel('Trip count')
plt.legend(title='User type')
plt.show()
```



Observations

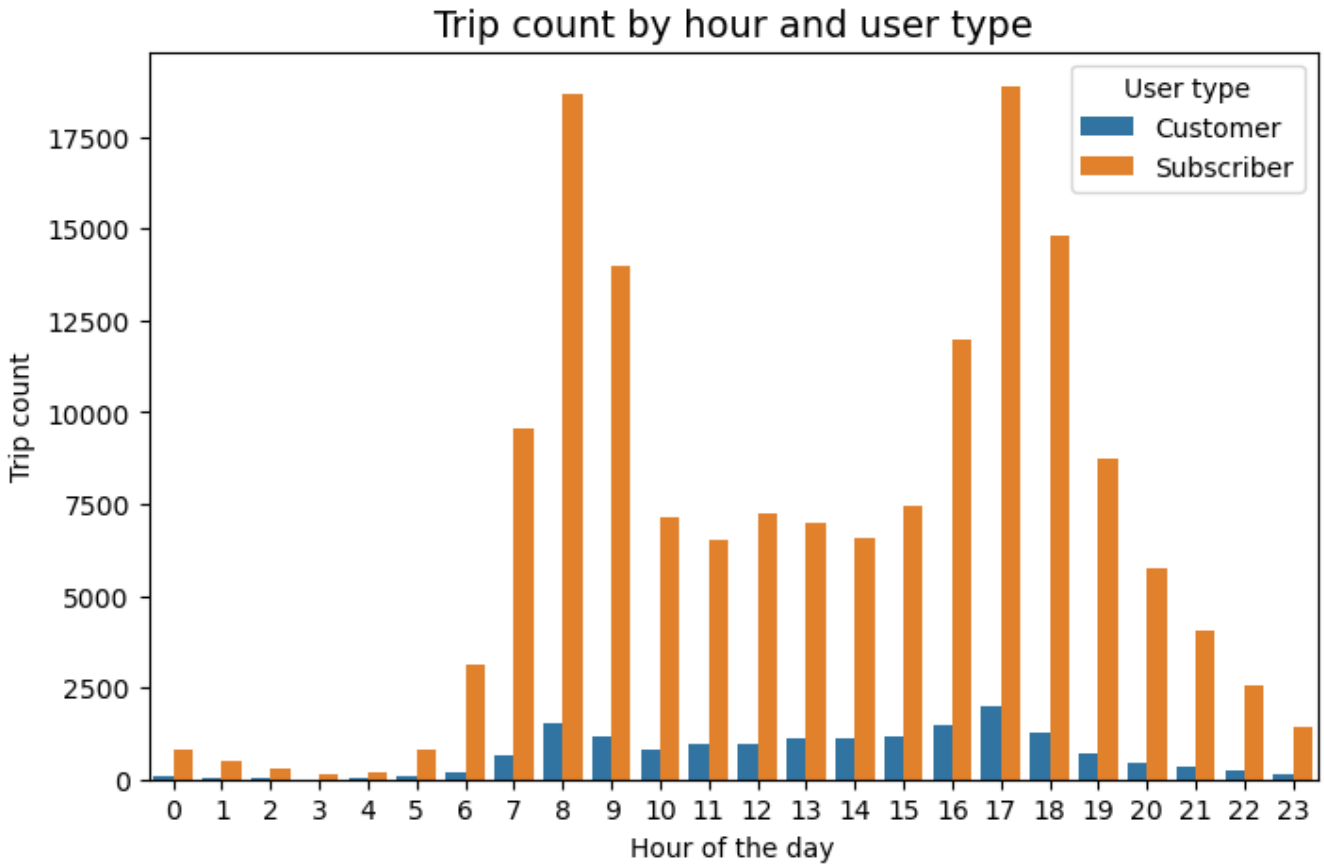
`Thurday is the day of the week with the highest usage frequency in both user type .`

Question

`Relation between the hour of the day and the user type`

Visualization

In [52]:
```
plt.figure(figsize = (8,5), dpi = 100)
sns.countplot(x='start_hour', hue='user_type', data=df_bike_clean)
plt.xlabel('Hour of the day')
plt.ylabel('Trip count')
plt.legend(title='User type')
plt.title('Trip count by hour and user type',fontsize=14)
plt.show()
```



Observations

`The same observation is made in this plot the hours with most usage of the bikes are at 8 in the morning and 17 in the afternoon . but we can also see that the subscribers tend to not uses the bike in weekend as frequent as they use it in weekdays.`

Question

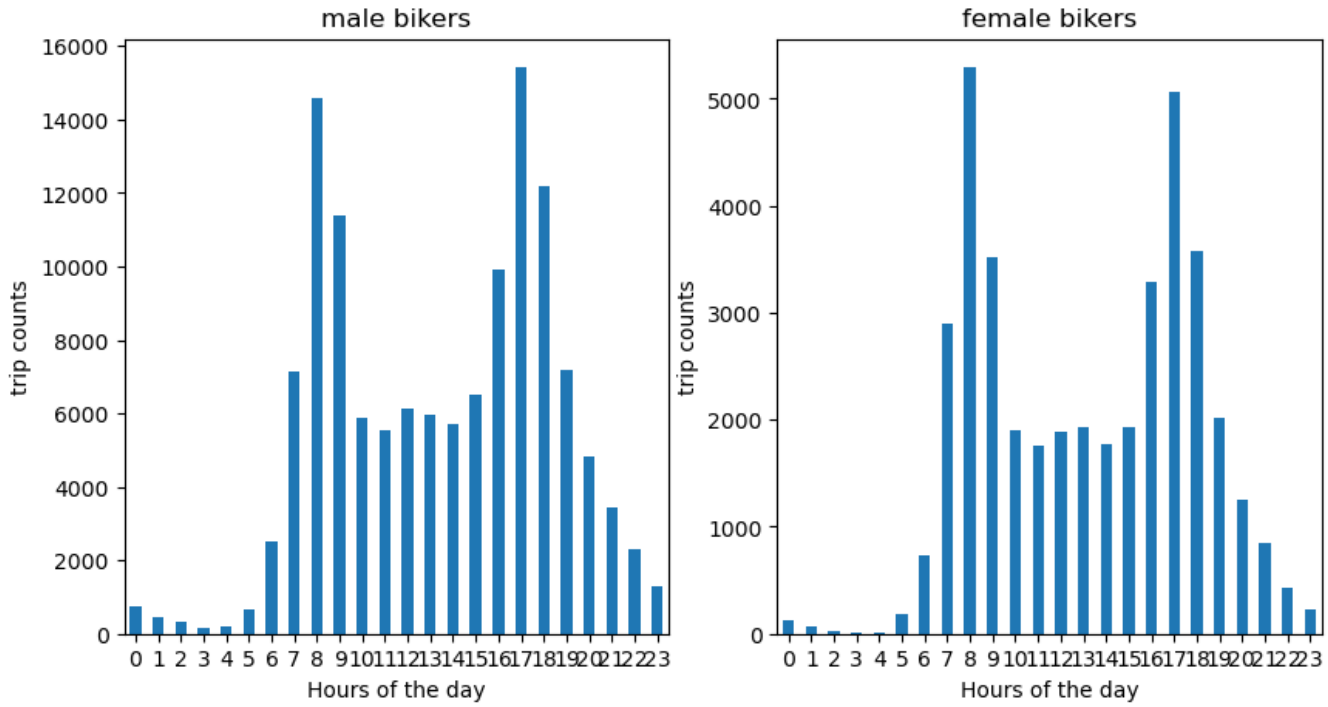
`Relation between the hour of the day and the genre`

Visualization

In [53]:
```
gender_trip = df_bike_clean.pivot_table(values="duration_min",\
                                        index='start_hour',columns='member_gender',ag
fig,axes =plt.subplots(1,2,figsize=(10,5))
gender_trip['Male'].plot(kind='bar',ax=axes[0],title="male bikers")
gender_trip['Female'].plot(kind='bar',ax=axes[1],title='female bikers')

for ax in axes:
    ax.set_xlabel('Hours of the day')
    ax.set_ylabel('trip counts')
    ax.set_xticklabels(ax.get_xticklabels(),rotation=0)


plt.show()
```



Observations

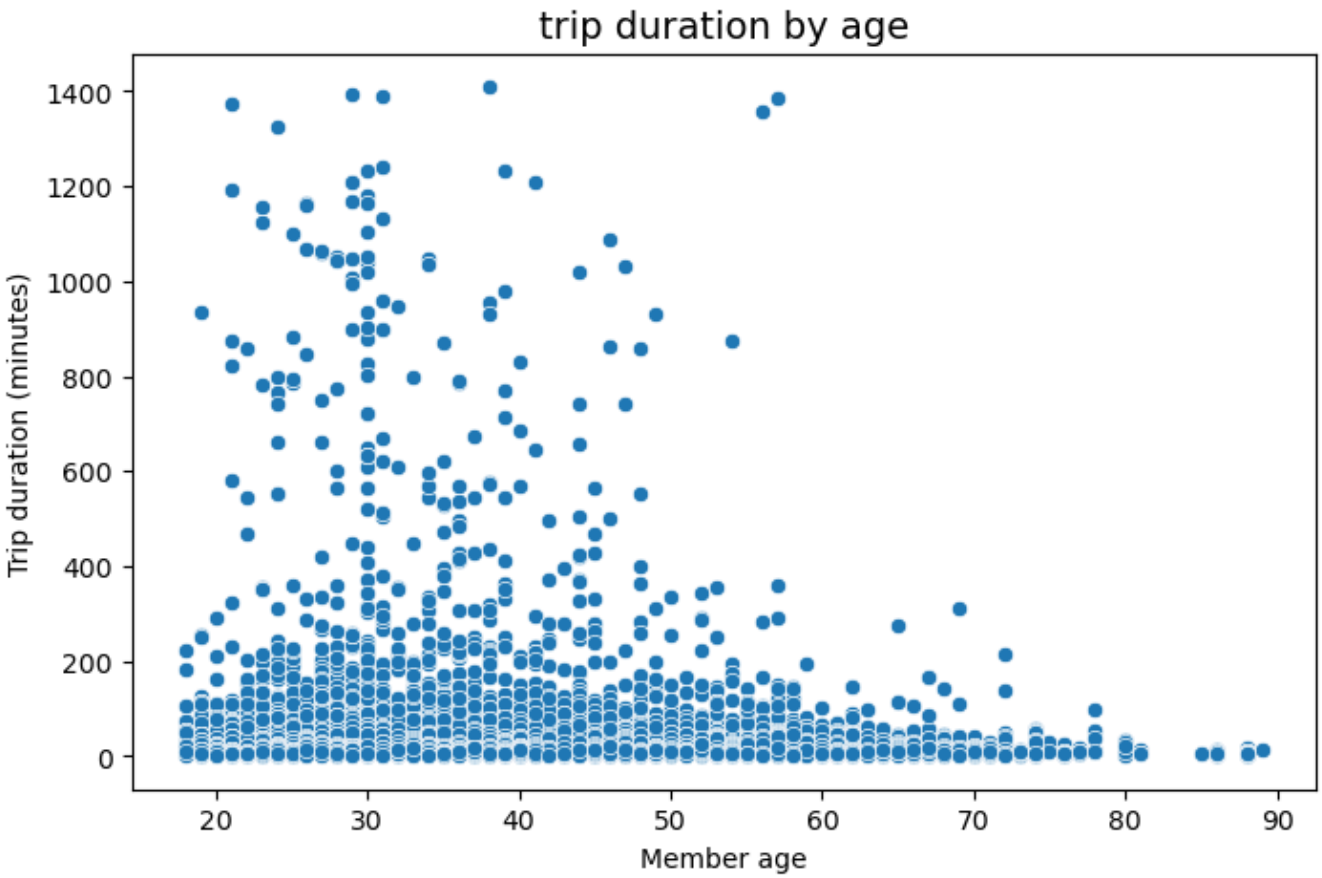
Even if the usage is less frequent between midnight and 5:00 am for both gender ,we can see that female bikers tends to not use the service between 1:00 AM and 5:00 AM this can be for safety reasons

Question

Relation between the trip duration and the age of the biker

Visualization

In [54]:
```
plt.figure(figsize = (8,5), dpi = 100)
sns.scatterplot(df_bike_clean,x='member_age',y='duration_min')
plt.xlabel('Member age')
plt.ylabel('Trip duration (minutes)')
plt.title('trip duration by age',fontsize=14)
plt.show()
```



Observations

`The bikers who covers the longest ride are the one between 20 and 50 even if they are not many compared to the total number of bikers`

CONCLUSION :

> `There is higher amount of biker the thurdays at 8am et 5pm and the usage of bike at his lowest between midnight and 5am everyday .`

> `Female bikers tends to not use the bikes between midnight and 5AM and it can be for safety reasons`

> `In the male as well as in the female there is more subscriber than customers`

Talk about some of the relationships you observed in this part of the investigation. How did the feature(s) of interest vary with other features in the dataset?

> **Variable of interest : duration**

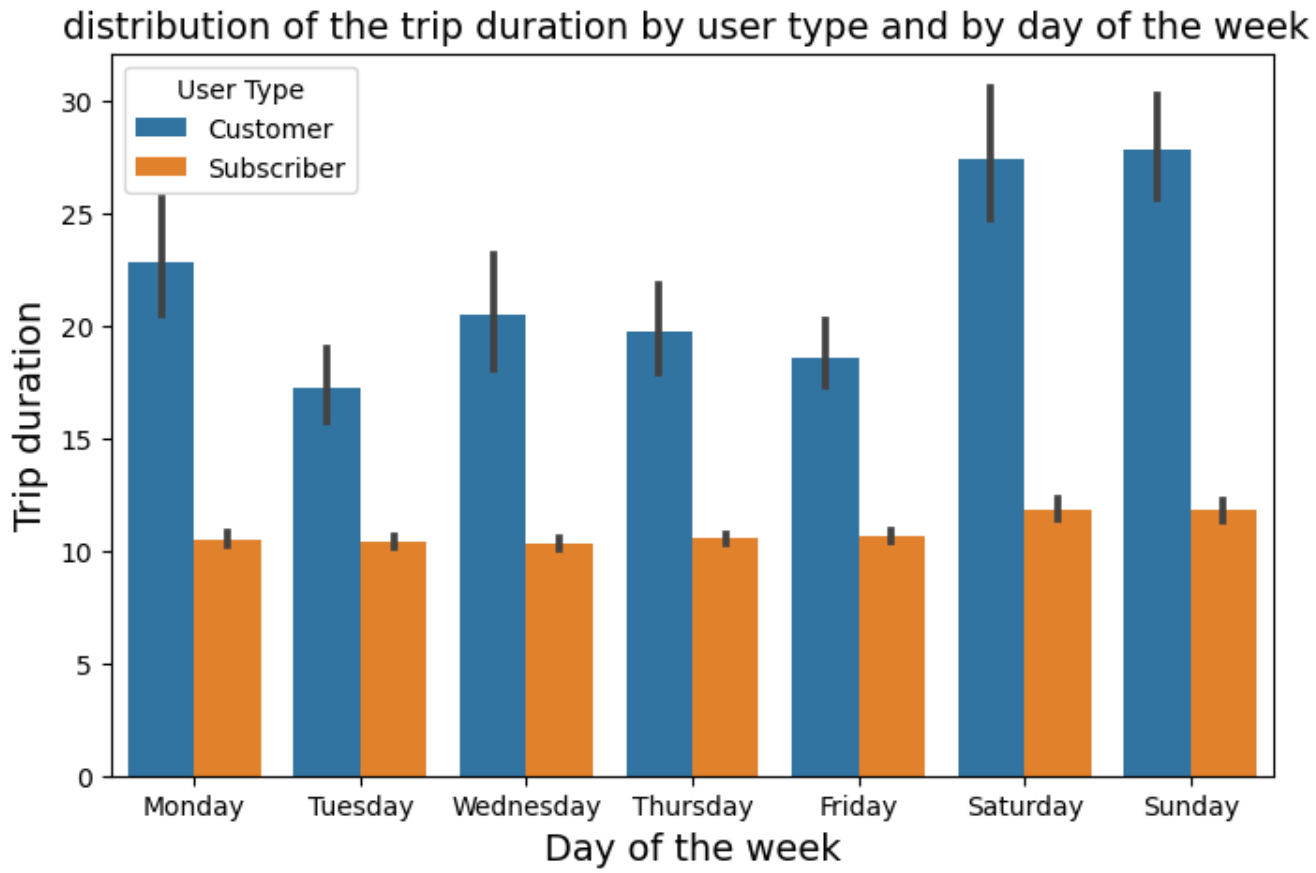
- **Subscribers** that have the possibility to ride regularly use it for short ride and the **customers** tends to use it for longer time.
- there is no clear correlation between the **age** and the duration of the rides
- the **genre** did not impact the trip duration
- There is a peak usage at 8am and 17pm which shows commute for work.

Did you observe any interesting relationships between the other features (not the main feature(s) of interest)?

There is no significant relationship between others features in my analysis

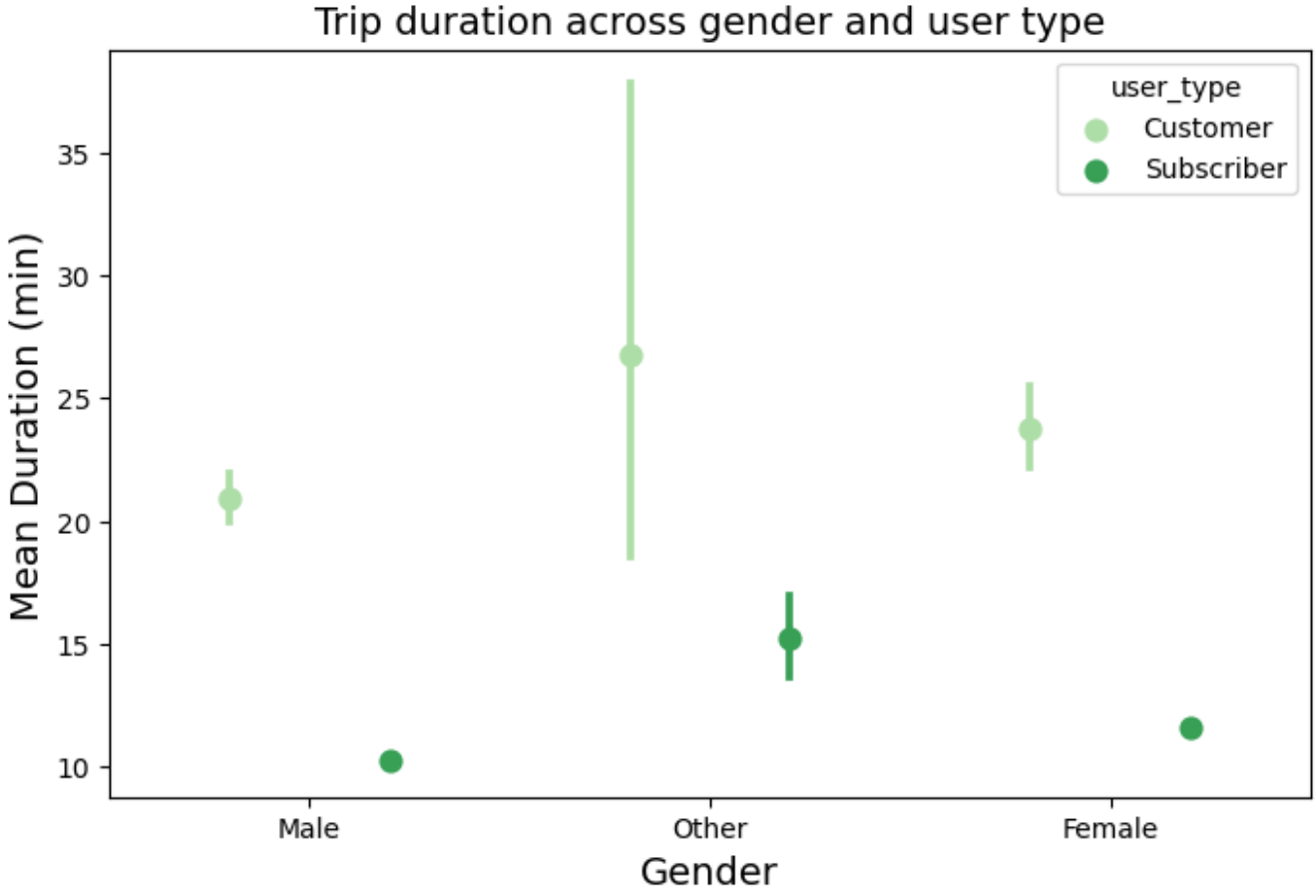
Multivariate Exploration

In [55]:
```
#Distibution of the trip duration by user type and by day of the week
plt.figure(figsize = (8,5), dpi = 100)
sns.barplot(data = df_bike_clean, x = "start_day_name", y = "duration_min", hue='user
plt.xlabel("Day of the week", fontsize=14)
plt.ylabel("Trip duration", fontsize=14)
plt.title("distribution of the trip duration by user type and by day of the week ",fo
plt.legend(title="User Type")
plt.show()
```



The customers have trips that are longer than the subscribers and bike ride on weekend are longer than in the one in weekdays

In [56]:
```
plt.figure(figsize = (8,5), dpi = 100)
ax = sns.pointplot(data = df_bike_clean, x = 'member_gender', y = 'duration_min', hue
           palette = 'Greens', linestyles = '', dodge = 0.4)
plt.title('Trip duration across gender and user type',fontsize=14)
plt.xlabel("Gender", fontsize=14)
plt.ylabel('Mean Duration (min)', fontsize=14)
ax.set_yticklabels([],minor = True)
plt.show();
```



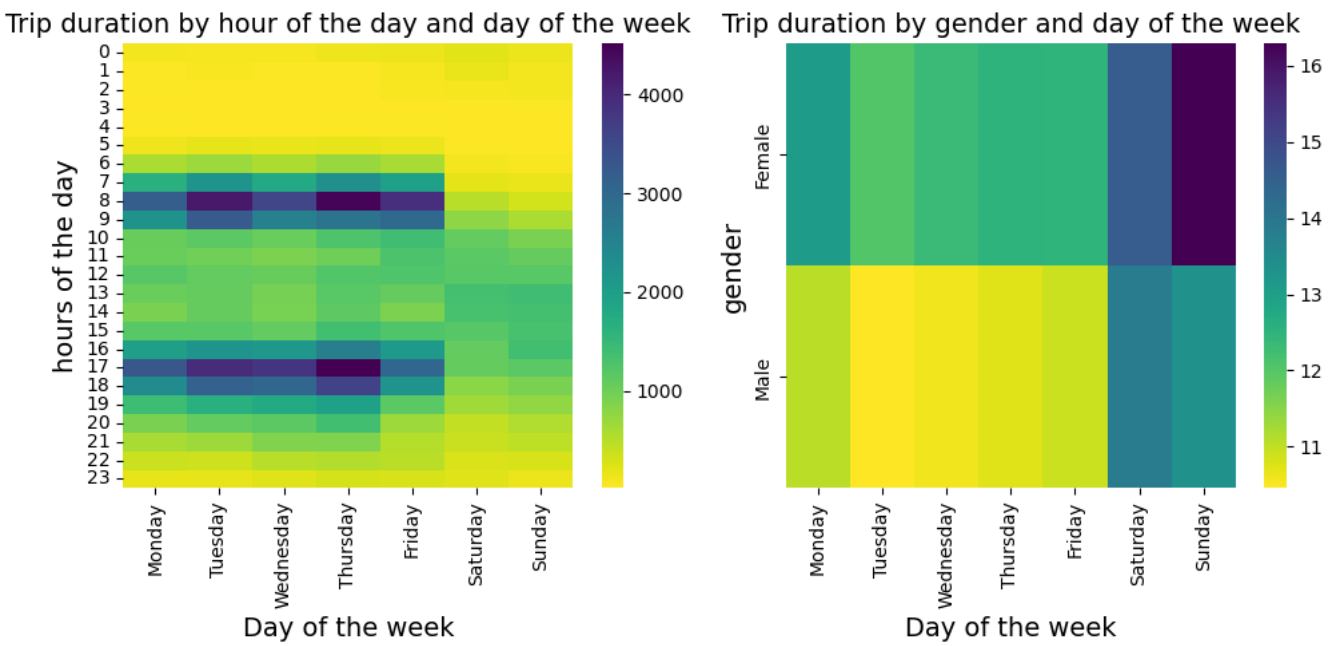
Among the type of users ,we can see that female bike longer trip than male and the other gender and that the customers in any genre have longer trip than the subscriber

In [69]:
```
hour_day_trip = df_bike_clean.pivot_table(values='duration_min', index='start_hour',
                                          columns='start_day_name', aggfunc='count')
gender_day_trip = df_bike_clean.pivot_table(values='duration_min', index='member_gend
                                            columns='start_day_name', aggfunc='mean')
gender_day_trip=gender_day_trip.drop(index='Other')
fig,axes =plt.subplots(1,2,figsize=(10,5))

sns.heatmap(hour_day_trip, cmap='viridis_r',ax=axes[0])
axes[0].set_title('Trip duration by hour of the day and day of the week',fontsize=14)

sns.heatmap(gender_day_trip, cmap='viridis_r',ax=axes[1])
axes[1].set_title('Trip duration by gender and day of the week',fontsize=14)

axes[0].set_ylabel('hours of the day',fontsize=14)
axes[1].set_ylabel('gender',fontsize=14)
for ax in axes:
    ax.set_xlabel('Day of the week',fontsize=14)
plt.tight_layout()
plt.show()
```



`Male have shorter trip than female on average. and the there is much less trip on weekend. between midnight and 5 there is clearly less rides.`

Talk about some of the relationships you observed in this part of the investigation. Were there features that strengthened each other in terms of looking at your feature(s) of interest?

> In the Multivariate analysis ,we can confirm our conclusion in the bivariate analysis. The relationship were between the trip duration and the gender,the trip duration and the hour of the day and the day of the week and also the trip duration and the type of users

Were there any interesting or surprising interactions between features?

> There was not really strong correlation between the trip duration and neither the gender or the age of the bike rider

Conclusions

The analysis shows that there are gender and user-type differences in bike usage patterns. Female users tend to ride longer but avoid biking at night, and customers have longer trips than subscribers. The higher frequency of bike usage on Thursday and during rush hours suggests that the bike-sharing system is primarily used for commuting purposes. Furthermore, the high number of subscribers indicates that the system is more frequently used by regular users rather than occasional ones. These findings can be used to optimize the bike-sharing system to better cater to the needs of different user groups and encourage more sustainable transportation.