exploration_template

September 3, 2021

1 Project " Analyzing FordGoBike Dataset (2021) "

1.1 by (Shahad Alotaibi)

1.2 Table of Content

Introduction

Gathering Data

Assesing Data

Cleaning Data

Data Visualization in Data analysis

1.2.1 Introduction

In this project, we aim to analyze data Ford GoBike is a regional public bicycle sharing system in the San Francisco Bay Area, California. Beginning operation in August 2013 as Bay Area Bike Share, the Ford GoBike system currently has over 2,600 bicycles in 262 stations across San Francisco, East Bay and San Jose. On June 28, 2017, the system officially launched as Ford GoBike in a partnership with Ford Motor Company. Ford GoBike It consists like other bicycle systems and is a system throughout the city, allowing users to move around and be able to use different stations around the city by opening the use of the bicycle from one station and the ability to retrieve it from another station, in order to reach the highest category from users.

[ ]:

[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 sb

%matplotlib inline
```

1.2.2 Gathering and Assesing Data

[2]:
```
Data = pd.read_csv('201902-fordgobike-tripdata.csv')
Data.head()
```

[2]:

	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
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)	37.789625	-122.400811	13.0	Commercial St at Montgomery St	37.794231	-122.402923	4902	Customer	1984.0	Male	No
1	42521	2019-02-28 18:53:21.7890	2019-03-01 06:42:03.0560	23.0	The Embarcadero at Steuart St	37.791464	-122.391034	81.0	Berry St at 4th St	37.775880	-122.393170	2535	Customer	NaN	NaN	No
2	61854	2019-02-28 12:13:13.2180	2019-03-01 05:24:08.1460	86.0	Market St at Dolores St	37.769305	-122.426826	3.0	Powell St BART Station (Market St at 4th St)	37.786375	-122.404904	5905	Customer	1972.0	Male	No
3	36490	2019-02-28 17:54:26.0100	2019-03-01 04:02:36.8420	375.0	Grove St at Masonic Ave	37.774836	-122.446546	70.0	Central Ave at Fell St	37.773311	-122.444293	6638	Subscriber	1989.0	Other	No
4	1585	2019-02-28 23:54:18.5490	2019-03-01 00:20:44.0740	7.0	Frank H Ogawa Plaza	37.804562	-122.271738	222.0	10th Ave at E 15th St	37.792714	-122.248780	4898	Subscriber	1974.0	Male	Yes

[3]:
```
Data.shape
```

[3]: (183412, 16)

[4]:
```
## show data informaation
Data.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
```

[5]:
```
Data.describe()
```

[5]:

	duration_sec	start_station_id	start_station_latitude	\
count	183412.000000	183215.000000	183412.000000	
mean	726.078435	138.590427	37.771223	
std	1794.389780	111.778864	0.099581	
min	61.000000	3.000000	37.317298	
25%	325.000000	47.000000	37.770083	
50%	514.000000	104.000000	37.780760	
75%	796.000000	239.000000	37.797280	
max	85444.000000	398.000000	37.880222	

	start_station_longitude	end_station_id	end_station_latitude	\
count	183412.000000	183215.000000	183412.000000	
mean	-122.352664	136.249123	37.771427	
std	0.117097	111.515131	0.099490	

```
min               -122.453704        3.000000             37.317298
25%               -122.412408       44.000000             37.770407
50%               -122.398285      100.000000             37.781010
75%               -122.286533      235.000000             37.797320
max               -121.874119      398.000000             37.880222

       end_station_longitude        bike_id  member_birth_year
count          183412.000000  183412.000000      175147.000000
mean             -122.352250    4472.906375        1984.806437
std                 0.116673    1664.383394          10.116689
min              -122.453704      11.000000        1878.000000
25%              -122.411726    3777.000000        1980.000000
50%              -122.398279    4958.000000        1987.000000
75%              -122.288045    5502.000000        1992.000000
max              -121.874119    6645.000000        2001.000000
```

[6]:
```
# will show the null information
Data.isnull().sum()
```

[6]:
```
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
```

[7]:
```
# will see the duplicated information
Data.duplicated().sum()
```

[7]:
```
0
```

[8]:
```
# will see the count value of colum member_gender
Data.member_gender.value_counts()
```

[8]:
```
Male      130651
Female     40844
```

```
Other       3652
Name: member_gender, dtype: int64
```

[9]:
```
# will see the count value of colum member_birth_year
Data.member_birth_year.value_counts()
```

[9]:
```
1988.0    10236
1993.0     9325
1989.0     8972
1990.0     8658
1991.0     8498
          …
1928.0        1
1878.0        1
1930.0        1
1910.0        1
1927.0        1
Name: member_birth_year, Length: 75, dtype: int64
```

[10]:
```
Data.user_type.value_counts()
```

[10]:
```
Subscriber    163544
Customer       19868
Name: user_type, dtype: int64
```

[11]:
```
Data.bike_share_for_all_trip.value_counts()
```

[11]:
```
No     166053
Yes     17359
Name: bike_share_for_all_trip, dtype: int64
```

1.2.3 Data Cleaning

[12]:
```
# Making a copy of dataframes before data cleaning keep orginals data

data_clean = Data.copy()
```

[13]:
```
# Remove rows that does not have gender value
data_clean = Data[Data['member_gender'].isnull() == False]
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 175147 entries, 0 to 183411
Data columns (total 16 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
```

```
 0   duration_sec             175147 non-null  int64
 1   start_time               175147 non-null  object
 2   end_time                 175147 non-null  object
 3   start_station_id         174952 non-null  float64
 4   start_station_name       174952 non-null  object
 5   start_station_latitude   175147 non-null  float64
 6   start_station_longitude  175147 non-null  float64
 7   end_station_id           174952 non-null  float64
 8   end_station_name         174952 non-null  object
 9   end_station_latitude     175147 non-null  float64
 10  end_station_longitude    175147 non-null  float64
 11  bike_id                  175147 non-null  int64
 12  user_type                175147 non-null  object
 13  member_birth_year        175147 non-null  float64
 14  member_gender            175147 non-null  object
 15  bike_share_for_all_trip  175147 non-null  object
dtypes: float64(7), int64(2), object(7)
memory usage: 22.7+ MB
```

[14]:
```
# Remove reows does not have station id
data_clean = data_clean[data_clean['start_station_id'].isnull() == False]
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 16 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   duration_sec             174952 non-null  int64
 1   start_time               174952 non-null  object
 2   end_time                 174952 non-null  object
 3   start_station_id         174952 non-null  float64
 4   start_station_name       174952 non-null  object
 5   start_station_latitude   174952 non-null  float64
 6   start_station_longitude  174952 non-null  float64
 7   end_station_id           174952 non-null  float64
 8   end_station_name         174952 non-null  object
 9   end_station_latitude     174952 non-null  float64
 10  end_station_longitude    174952 non-null  float64
 11  bike_id                  174952 non-null  int64
 12  user_type                174952 non-null  object
 13  member_birth_year        174952 non-null  float64
 14  member_gender            174952 non-null  object
 15  bike_share_for_all_trip  174952 non-null  object
dtypes: float64(7), int64(2), object(7)
memory usage: 22.7+ MB
```

[16]:
```
# Change start time and end time to datetime format
data_clean.start_time = pd.to_datetime(data_clean.start_time)
data_clean.end_time = pd.to_datetime(data_clean.end_time)
```

[17]:
```
data_clean['start_time_dayofweek']= data_clean['start_time'].dt.strftime('%a')
data_clean['start_time_hour']= data_clean['start_time'].dt.hour
```

[18]:
```
print(data_clean.start_time.dtype)
print(data_clean.end_time.dtype)
```

```
datetime64[ns]
datetime64[ns]
```

[19]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 18 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   duration_sec             174952 non-null  int64
 1   start_time               174952 non-null  datetime64[ns]
 2   end_time                 174952 non-null  datetime64[ns]
 3   start_station_id         174952 non-null  float64
 4   start_station_name       174952 non-null  object
 5   start_station_latitude   174952 non-null  float64
 6   start_station_longitude  174952 non-null  float64
 7   end_station_id           174952 non-null  float64
 8   end_station_name         174952 non-null  object
 9   end_station_latitude     174952 non-null  float64
 10  end_station_longitude    174952 non-null  float64
 11  bike_id                  174952 non-null  int64
 12  user_type                174952 non-null  object
 13  member_birth_year        174952 non-null  float64
 14  member_gender            174952 non-null  object
 15  bike_share_for_all_trip  174952 non-null  object
 16  start_time_dayofweek     174952 non-null  object
 17  start_time_hour          174952 non-null  int64
dtypes: datetime64[ns](2), float64(7), int64(3), object(6)
memory usage: 25.4+ MB
```

[20]:
```
# Convert the start_time_dayofweek to ordinal variables.

weekdays = ['Mon','Tue','Wed','Thu','Fri', 'Sat', 'Sun']
ordered_weekdays = pd.api.types.CategoricalDtype(ordered = True, categories =␣
↪weekdays)
```

```
data_clean['start_time_dayofweek'] = data_clean['start_time_dayofweek'].
 ↪astype(ordered_weekdays)
```

[21]:
```
data_clean['start_time_dayofweek'].value_counts()
```

[21]:
```
Thu    33712
Tue    30584
Wed    28426
Fri    27663
Mon    25641
Sun    14512
Sat    14414
Name: start_time_dayofweek, dtype: int64
```

[22]:
```
# Drop other Gender
drop_other = data_clean[data_clean['member_gender']=='Other'].index
data_clean = data_clean.drop(index=drop_other)
```

[23]:
```
#Test
data_clean['member_gender'].value_counts()
```

[23]:
```
Male      130500
Female     40805
Name: member_gender, dtype: int64
```

[24]:
```
data_clean.sample(10)
```

[24]:

	duration_sec	start_time	end_time	start_station_id	start_station_name
115508	335	2019-02-12 08:40:02.656	2019-02-12 08:45:38.116	16.0	Steuart St at Market St
16371	1103	2019-02-27 08:24:13.021	2019-02-27 08:42:36.386	85.0	Church St at Duboce Ave
150082	673	2019-02-06 18:26:51.691	2019-02-06 18:38:04.953	59.0	S Van Ness Ave at Market St
170234	1123	2019-02-04 08:13:52.381	2019-02-04 08:32:35.508	132.0	24th St at Chattanooga St
140834	443	2019-02-07 18:36:32.739	2019-02-07 18:43:56.213	30.0	San Francisco Caltrain (Townsend St at 4th St)
113216	420	2019-02-12 12:50:13.842	2019-02-12 12:57:14.784	3.0	Powell St BART Station (Market St at 4th St)
123318	646	2019-02-11 09:28:37.400	2019-02-11 09:39:24.298	5.0	Powell St BART Station (Market St at 5th St)
24576	1435	2019-02-25 17:47:37.285	2019-02-25 18:11:32.533		
161793	479	2019-02-05 13:43:16.820	2019-02-05 13:51:15.968		
18984	296	2019-02-26 17:44:33.281	2019-02-26 17:49:30.175		

```
24576             155.0                          Emeryville Public Market
161793             15.0  San Francisco Ferry Building (Harry Bridges Pl…
18984             253.0                           Haste St at College Ave

        start_station_latitude  start_station_longitude  end_station_id  \
115508                37.794130              -122.394430             6.0
16371                 37.770083              -122.429156            23.0
150082                37.774814              -122.418954            93.0
170234                37.751819              -122.426614            90.0
140834                37.776598              -122.395282            27.0
113216                37.786375              -122.404904            10.0
123318                37.783899              -122.408445           102.0
24576                 37.840521              -122.293528           243.0
161793                37.795392              -122.394203           343.0
18984                 37.866418              -122.253799           256.0

                    end_station_name  end_station_latitude  \
115508  The Embarcadero at Sansome St             37.804770
16371   The Embarcadero at Steuart St             37.791464
150082   4th St at Mission Bay Blvd S             37.770407
170234          Townsend St at 7th St             37.771058
140834        Beale St at Harrison St             37.788059
113216     Washington St at Kearny St             37.795393
123318              Irwin St at 8th St             37.766883
24576     Bancroft Way at College Ave             37.869360
161793             Bryant St at 2nd St             37.783172
18984         Hearst Ave at Euclid Ave             37.875112

        end_station_longitude  bike_id   user_type  member_birth_year  \
115508            -122.403234     4737  Subscriber             1990.0
16371             -122.391034     5043  Subscriber             1990.0
150082            -122.391198     4761  Subscriber             1988.0
170234            -122.402717     5218  Subscriber             1995.0
140834            -122.391865      746  Subscriber             1988.0
113216            -122.404770     5253  Subscriber             1999.0
123318            -122.399579     1714    Customer             1990.0
24576             -122.254337     4498  Subscriber             1992.0
161793            -122.393572     4902  Subscriber             1968.0
18984             -122.260553     5658  Subscriber             1997.0

       member_gender bike_share_for_all_trip start_time_dayofweek  \
115508        Female                     No                  Tue
16371         Female                     No                  Wed
150082          Male                     No                  Wed
170234        Female                     No                  Mon
140834          Male                     No                  Thu
113216          Male                     No                  Tue
```

```
123318          Male                    No                  Mon
24576           Male                    No                  Mon
161793        Female                    No                  Tue
18984           Male                    No                  Tue

        start_time_hour
115508                8
16371                 8
150082               18
170234                8
140834               18
113216               12
123318                9
24576                17
161793               13
18984                17
```

[25]:
```
# Calculate member age from member_birth_year to get the age.
data_clean['member_age'] = 2021 - data_clean['member_birth_year']
```

[26]:
```
data_clean['member_age'] = data_clean['member_age'].astype(int)
data_clean['member_birth_year'] = data_clean['member_birth_year'].astype(int)
```

[27]:
```
data_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 171305 entries, 0 to 183411
Data columns (total 19 columns):
 #   Column                   Non-Null Count   Dtype
---  ------                   --------------   -----
 0   duration_sec             171305 non-null  int64
 1   start_time               171305 non-null  datetime64[ns]
 2   end_time                 171305 non-null  datetime64[ns]
 3   start_station_id         171305 non-null  float64
 4   start_station_name       171305 non-null  object
 5   start_station_latitude   171305 non-null  float64
 6   start_station_longitude  171305 non-null  float64
 7   end_station_id           171305 non-null  float64
 8   end_station_name         171305 non-null  object
 9   end_station_latitude     171305 non-null  float64
 10  end_station_longitude    171305 non-null  float64
 11  bike_id                  171305 non-null  int64
 12  user_type                171305 non-null  object
 13  member_birth_year        171305 non-null  int64
 14  member_gender            171305 non-null  object
 15  bike_share_for_all_trip  171305 non-null  object
 16  start_time_dayofweek     171305 non-null  category
```

```
 17  start_time_hour          171305 non-null  int64
 18  member_age               171305 non-null  int64
dtypes: category(1), datetime64[ns](2), float64(6), int64(5), object(5)
memory usage: 25.0+ MB
```

[28]:
```
## create duration min colum
data_clean['duration_min'] = data_clean['duration_sec']/60
```

[29]:
```
data_clean.describe()
```

[29]:

	duration_sec	start_station_id	start_station_latitude
count	171305.000000	171305.00000	171305.000000
mean	697.757981	138.70695	37.770629
std	1577.253741	111.71479	0.101225
min	61.000000	3.00000	37.317298
25%	322.000000	47.00000	37.770083
50%	509.000000	104.00000	37.780760
75%	787.000000	239.00000	37.797280
max	84548.000000	398.00000	37.880222

	start_station_longitude	end_station_id	end_station_latitude
count	171305.000000	171305.000000	171305.000000
mean	-122.351657	136.304889	37.770831
std	0.118522	111.421147	0.101130
min	-122.453704	3.000000	37.317298
25%	-122.411901	44.000000	37.770407
50%	-122.398279	100.000000	37.781010
75%	-122.283127	237.000000	37.797320
max	-121.874119	398.000000	37.880222

	end_station_longitude	bike_id	member_birth_year
count	171305.000000	171305.000000	171305.000000
mean	-122.351225	4481.294136	1984.839351
std	0.118088	1659.524197	10.116083
min	-122.453704	11.000000	1878.000000
25%	-122.411647	3796.000000	1980.000000
50%	-122.397437	4960.000000	1987.000000
75%	-122.287610	5505.000000	1992.000000
max	-121.874119	6645.000000	2001.000000

	start_time_hour	member_age	duration_min
count	171305.000000	171305.000000	171305.000000
mean	13.451545	36.160649	11.629300
std	4.733722	10.116083	26.287562
min	0.000000	20.000000	1.016667
25%	9.000000	29.000000	5.366667
50%	14.000000	34.000000	8.483333

```
75%          17.000000      41.000000      13.116667
max          23.000000     143.000000    1409.133333
```

[ ]:

1.2.4 What is the structure of your dataset?

The data contains about 16 columns and 183412 of information, which is the date of the start of the trip, the date of the end of the trip, the type of users of subscribers and customers, the type of user of men and women, and there is also a column on the days of the week to help obtain information in more number of days of use

[31]:
```
data_clean.head(20)
```

[31]:
```
    duration_sec              start_time                 end_time  \
0          52185 2019-02-28 17:32:10.145 2019-03-01 08:01:55.975
2          61854 2019-02-28 12:13:13.218 2019-03-01 05:24:08.146
4           1585 2019-02-28 23:54:18.549 2019-03-01 00:20:44.074
5           1793 2019-02-28 23:49:58.632 2019-03-01 00:19:51.760
6           1147 2019-02-28 23:55:35.104 2019-03-01 00:14:42.588
7           1615 2019-02-28 23:41:06.766 2019-03-01 00:08:02.756
9           1049 2019-02-28 23:49:47.699 2019-03-01 00:07:17.025
10           458 2019-02-28 23:57:57.211 2019-03-01 00:05:35.435
11           506 2019-02-28 23:56:55.540 2019-03-01 00:05:21.733
12          1176 2019-02-28 23:45:12.651 2019-03-01 00:04:49.184
14           395 2019-02-28 23:56:26.848 2019-03-01 00:03:01.947
15           208 2019-02-28 23:59:18.548 2019-03-01 00:02:47.228
16           548 2019-02-28 23:50:41.607 2019-02-28 23:59:49.953
17           674 2019-02-28 23:48:25.095 2019-02-28 23:59:40.092
18           557 2019-02-28 23:49:01.851 2019-02-28 23:58:19.809
19           874 2019-02-28 23:43:05.183 2019-02-28 23:57:39.796
20           417 2019-02-28 23:50:38.239 2019-02-28 23:57:35.852
21           414 2019-02-28 23:50:26.879 2019-02-28 23:57:21.130
22           743 2019-02-28 23:44:56.439 2019-02-28 23:57:20.212
23           367 2019-02-28 23:51:06.014 2019-02-28 23:57:13.312

    start_station_id                                start_station_name  \
0               21.0   Montgomery St BART Station (Market St at 2nd St)
2               86.0                            Market St at Dolores St
4                7.0                                Frank H Ogawa Plaza
5               93.0                      4th St at Mission Bay Blvd S
6              300.0                             Palm St at Willow St
7               10.0                       Washington St at Kearny St
9               19.0                             Post St at Kearny St
10             370.0                              Jones St at Post St
11              44.0  Civic Center/UN Plaza BART Station (Market St …
```

```
12             127.0                          Valencia St at 21st St
14             243.0                    Bancroft Way at College Ave
15             349.0                           Howard St at Mary St
16             131.0                          22nd St at Dolores St
17              74.0                          Laguna St at Hayes St
18             321.0                               5th St at Folsom
19             180.0                     Telegraph Ave at 23rd St
20              72.0                           Page St at Scott St
21             163.0                    Lake Merritt BART Station
22             370.0                          Jones St at Post St
23             243.0                    Bancroft Way at College Ave

    start_station_latitude  start_station_longitude  end_station_id  \
0                37.789625              -122.400811            13.0
2                37.769305              -122.426826             3.0
4                37.804562              -122.271738           222.0
5                37.770407              -122.391198           323.0
6                37.317298              -121.884995           312.0
7                37.795393              -122.404770           127.0
9                37.788975              -122.403452           121.0
10               37.787327              -122.413278            43.0
11               37.781074              -122.411738           343.0
12               37.756708              -122.421025           323.0
14               37.869360              -122.254337           252.0
15               37.781010              -122.405666            60.0
16               37.755000              -122.425728            71.0
17               37.776435              -122.426244           336.0
18               37.780146              -122.403071            75.0
19               37.812678              -122.268773           180.0
20               37.772406              -122.435650           107.0
21               37.797320              -122.265320           221.0
22               37.787327              -122.413278            52.0
23               37.869360              -122.254337           269.0

                                    end_station_name  end_station_latitude  \
0                     Commercial St at Montgomery St             37.794231
2        Powell St BART Station (Market St at 4th St)             37.786375
4                              10th Ave at E 15th St             37.792714
5                                 Broadway at Kearny             37.798014
6                            San Jose Diridon Station             37.329732
7                              Valencia St at 21st St             37.756708
9                                 Mission Playground             37.759210
10  San Francisco Public Library (Grove St at Hyde…             37.778768
11                               Bryant St at 2nd St             37.783172
12                                Broadway at Kearny             37.798014
14                     Channing Way at Shattuck Ave             37.865847
15                             8th St at Ringold St             37.774520
```

```
16                            Broderick St at Oak St             37.773063
17                       Potrero Ave and Mariposa St             37.763281
18                          Market St at Franklin St             37.773793
19                          Telegraph Ave at 23rd St             37.812678
20                             17th St at Dolores St             37.763015
21          6th Ave at E 12th St (Temporary Location)             37.794396
22                           McAllister St at Baker St             37.777416
23                        Telegraph Ave at Carleton St             37.862320

    end_station_longitude  bike_id   user_type  member_birth_year  \
0              -122.402923     4902    Customer               1984
2              -122.404904     5905    Customer               1972
4              -122.248780     4898  Subscriber               1974
5              -122.405950     5200  Subscriber               1959
6              -121.901782     3803  Subscriber               1983
7              -122.421025     6329  Subscriber               1989
9              -122.421339     6488  Subscriber               1992
10             -122.415929     5318  Subscriber               1996
11             -122.393572     5848  Subscriber               1993
12             -122.405950     5328    Customer               1990
14             -122.267443     4786  Subscriber               1988
15             -122.409449     6361  Subscriber               1993
16             -122.439078     6572  Subscriber               1981
17             -122.407377     5343  Subscriber               1975
18             -122.421239     5854  Subscriber               1990
19             -122.268773     5629    Customer               1978
20             -122.426497     4999  Subscriber               1983
21             -122.253842     6007  Subscriber               1984
22             -122.441838     5479  Subscriber               1991
23             -122.258801     1804  Subscriber               1997

   member_gender bike_share_for_all_trip start_time_dayofweek  \
0           Male                     No                  Thu
2           Male                     No                  Thu
4           Male                    Yes                  Thu
5           Male                     No                  Thu
6         Female                     No                  Thu
7           Male                     No                  Thu
9           Male                     No                  Thu
10        Female                    Yes                  Thu
11          Male                     No                  Thu
12          Male                     No                  Thu
14          Male                     No                  Thu
15          Male                    Yes                  Thu
16          Male                     No                  Thu
17          Male                     No                  Thu
18          Male                     No                  Thu
```

```
19          Male                    No                 Thu
20          Male                    No                 Thu
21          Male                   Yes                 Thu
22        Female                    No                 Thu
23        Female                    No                 Thu

    start_time_hour  member_age  duration_min
0                17          37    869.750000
2                12          49   1030.900000
4                23          47     26.416667
5                23          62     29.883333
6                23          38     19.116667
7                23          32     26.916667
9                23          29     17.483333
10               23          25      7.633333
11               23          28      8.433333
12               23          31     19.600000
14               23          33      6.583333
15               23          28      3.466667
16               23          40      9.133333
17               23          46     11.233333
18               23          31      9.283333
19               23          43     14.566667
20               23          38      6.950000
21               23          37      6.900000
22               23          30     12.383333
23               23          24      6.116667
```

[32]:
```
# descriptive statistics for numeric variables
data_clean.describe()
```

[32]:
```
        duration_sec  start_station_id  start_station_latitude  \
count  171305.000000      171305.00000           171305.000000
mean      697.757981         138.70695               37.770629
std      1577.253741         111.71479                0.101225
min        61.000000           3.00000               37.317298
25%       322.000000          47.00000               37.770083
50%       509.000000         104.00000               37.780760
75%       787.000000         239.00000               37.797280
max     84548.000000         398.00000               37.880222

       start_station_longitude  end_station_id  end_station_latitude  \
count            171305.000000   171305.000000         171305.000000
mean               -122.351657      136.304889             37.770831
std                   0.118522      111.421147              0.101130
min                -122.453704        3.000000             37.317298
25%                -122.411901       44.000000             37.770407
```

```
50%                -122.398279      100.000000             37.781010
75%                -122.283127      237.000000             37.797320
max                -121.874119      398.000000             37.880222

       end_station_longitude        bike_id  member_birth_year  \
count          171305.000000  171305.000000      171305.000000
mean             -122.351225    4481.294136        1984.839351
std                 0.118088    1659.524197          10.116083
min              -122.453704      11.000000        1878.000000
25%              -122.411647    3796.000000        1980.000000
50%              -122.397437    4960.000000        1987.000000
75%              -122.287610    5505.000000        1992.000000
max              -121.874119    6645.000000        2001.000000

       start_time_hour     member_age   duration_min
count    171305.000000  171305.000000  171305.000000
mean         13.451545      36.160649      11.629300
std           4.733722      10.116083      26.287562
min           0.000000      20.000000       1.016667
25%           9.000000      29.000000       5.366667
50%          14.000000      34.000000       8.483333
75%          17.000000      41.000000      13.116667
max          23.000000     143.000000    1409.133333
```

1.2.5 What does this data contain and what are its advantages?

This data contains a set of data that talks about the trips, the beginning of the trip, the end of the trip, the trips that take place on weekdays, the trips that are dealt with on weekends, and also contains the types of users' gender, and there are types of users between the subscriber and between customer We processed and classified this data to be used in the analyzes and to show its results in recent presentations and to be better read and better visualize the data in a graphical form instead of making it readable data

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

I think the following features may help to support my investigation into the feature (duration). start_time member_birth_year - we may derive member age and investigate the relationship between member age and bike durations. member_gender user_type bike_id

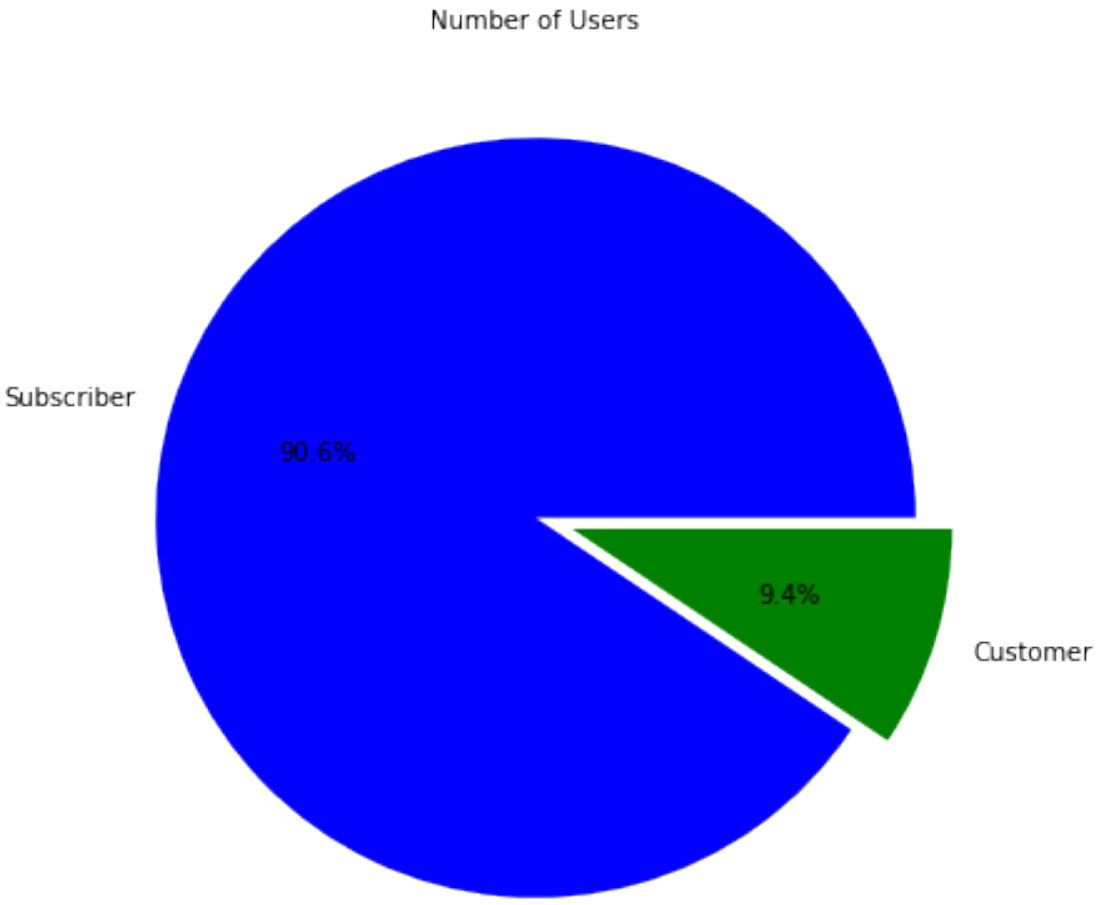
1.3 Visualization and Analysis

In this section, we will analyze the data that we have obtained after the cleaning process and the results that we touched on to help obtain information that helps in developing the tour system,

and the offers are exploited in providing them. There are many suggestions that we touched on in the Readme file

1.3.1 How many the ratio of the use of subscribers and customer "user type" ?

[33]:
```
labels=data_clean['user_type'].value_counts().index
colors=['blue','green']
explode=[0,0.1]
values=data_clean['user_type'].value_counts().values
plt.figure(figsize=(7,7))
plt.pie(values,explode=explode,labels=labels,colors=colors,autopct='%1.1f%%')
plt.title('Number of Users',fontsize=10)
plt.show()
```

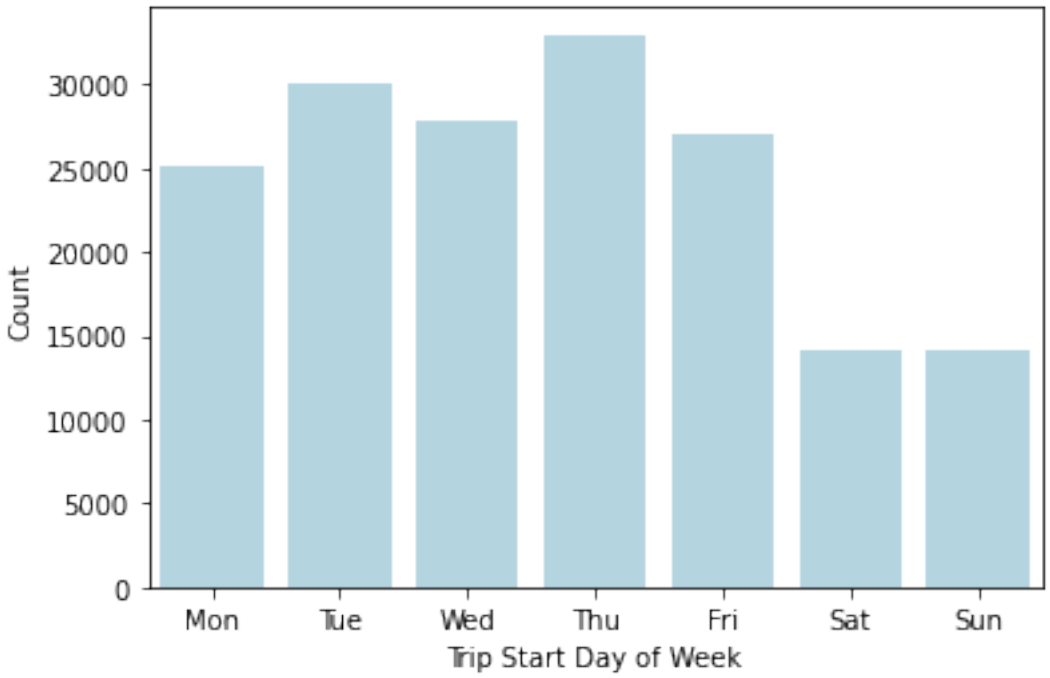


We note here that the percentage of subscribers exceeds 90% of the percentage of customers, with

a difference of about 9.2%, the difference between the percentage of subscribers and the percentage of customers, and this gives a strong indication that the percentage of subscribers is higher and this indicates the existence of a large category and segment that has been benefited from

1.3.2 How many bikes are used during the week?

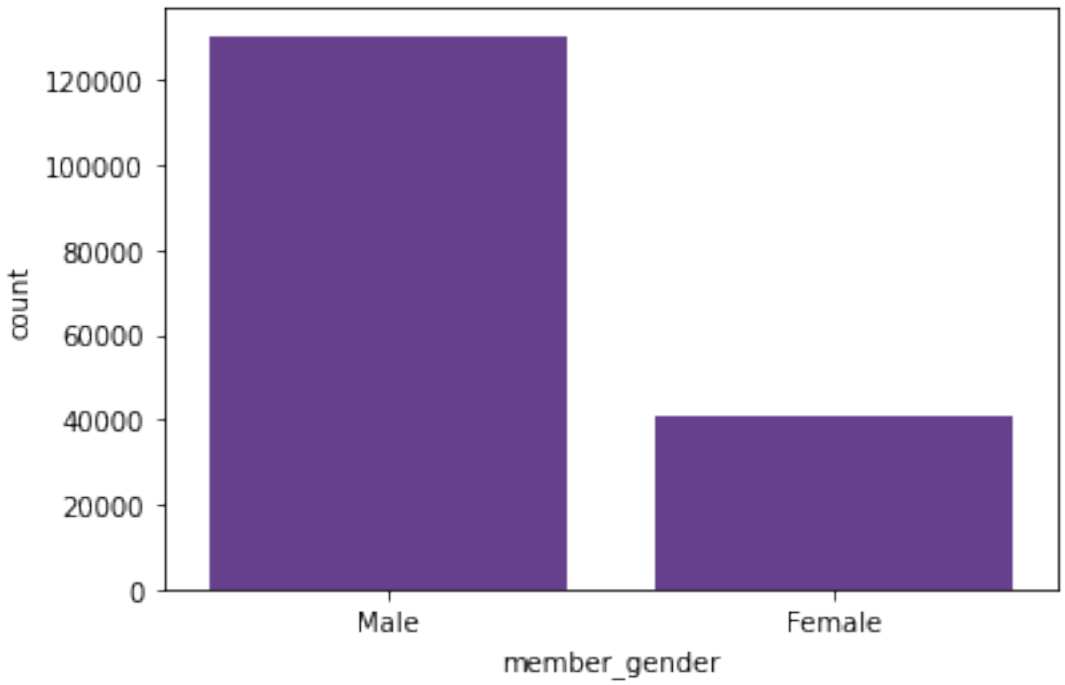
[34]:
```
weekday = ['Mon','Tue','Wed','Thu','Fri', 'Sat', 'Sun']
weekdaycat = pd.api.types.CategoricalDtype(ordered=True, categories=weekday)
data_clean['start_time_dayofweek'] = data_clean['start_time_dayofweek'].
 ↪astype(weekdaycat)

sb.countplot(data=data_clean, x='start_time_dayofweek', color='lightblue');
plt.xlabel('Trip Start Day of Week');
plt.ylabel('Count');
```



We note here that Thursday is the highest day of the week in the use of bikes, and that is considered the weekend, while there is a very close ratio between Friday and Tuesday in the rate of use

1.3.3 What is the percentage of bikes use between the of Male and Female ?

[35]:
```
sb.countplot(data = data_clean, x = 'member_gender', color = 'rebeccapurple');
```



1.3.4 Beginning hours of using the bike

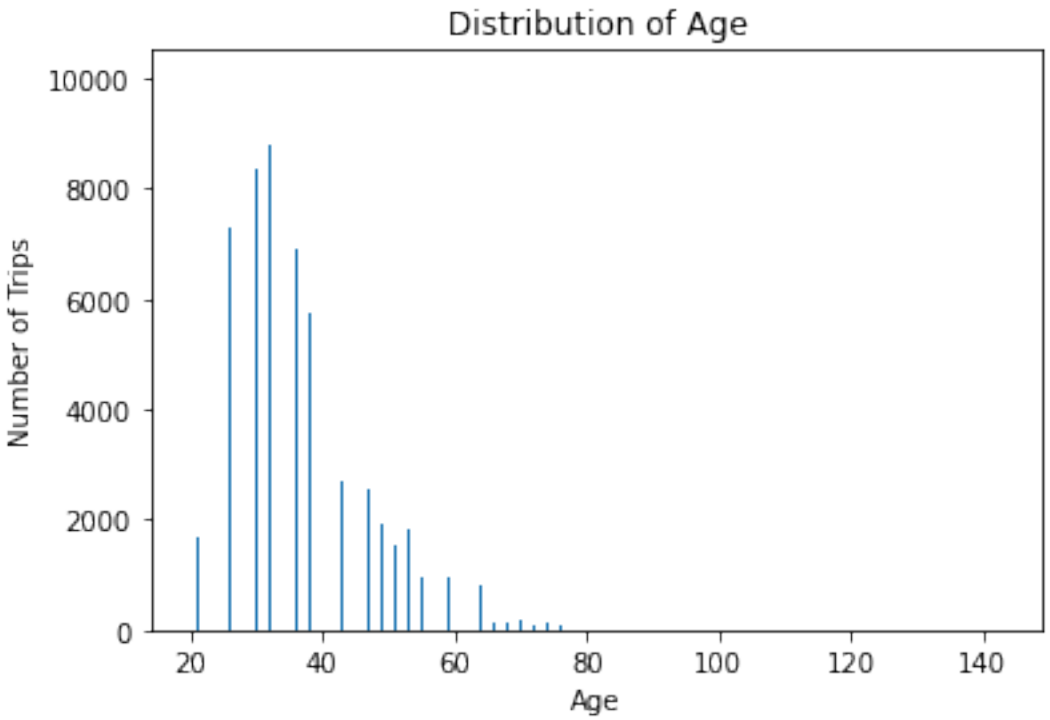
[36]:
```
data_clean ['start_time_hour'].value_counts()
```

[36]:
```
17    20497
8     19881
18    15785
9     14907
16    13196
7     10055
19     9210
15     8468
12     8020
13     7906
10     7759
14     7494
11     7298
20     6085
21     4291
6      3258
22     2720
23     1530
0       869
```

```
5       860
1       509
2       340
4       215
3       152
Name: start_time_hour, dtype: int64
```

1.3.5 What are the ages for bike the most?

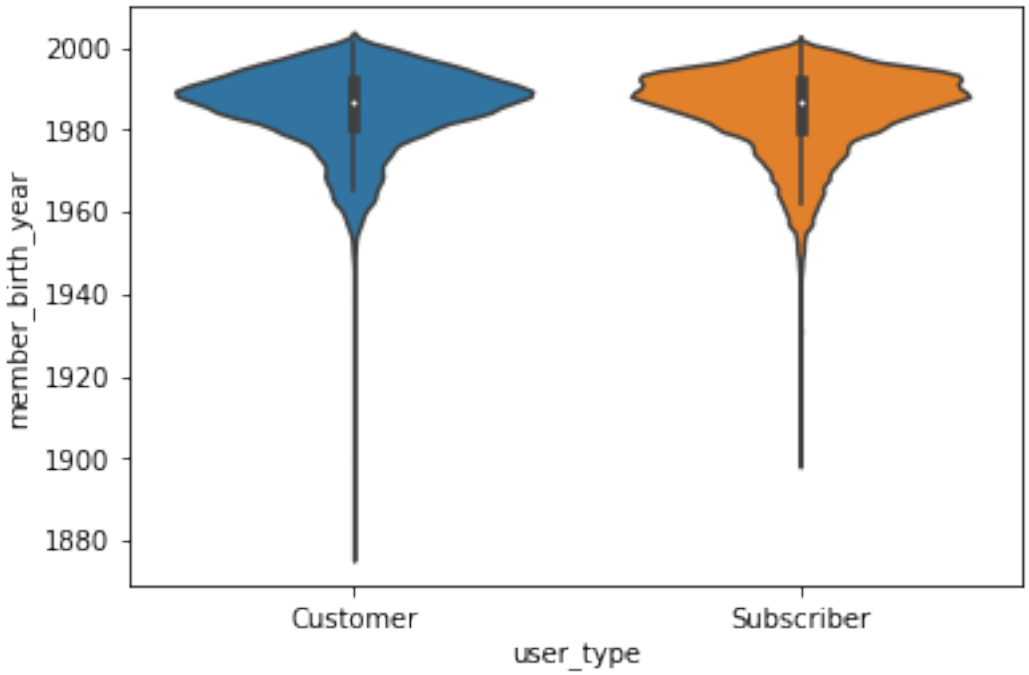
[37]:
```
plt.hist(data = data_clean, x = 'member_age',bins=1000)
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Number of Trips')
plt.figure(figsize=[8,5])
plt.show()
```




```
<Figure size 576x360 with 0 Axes>
```

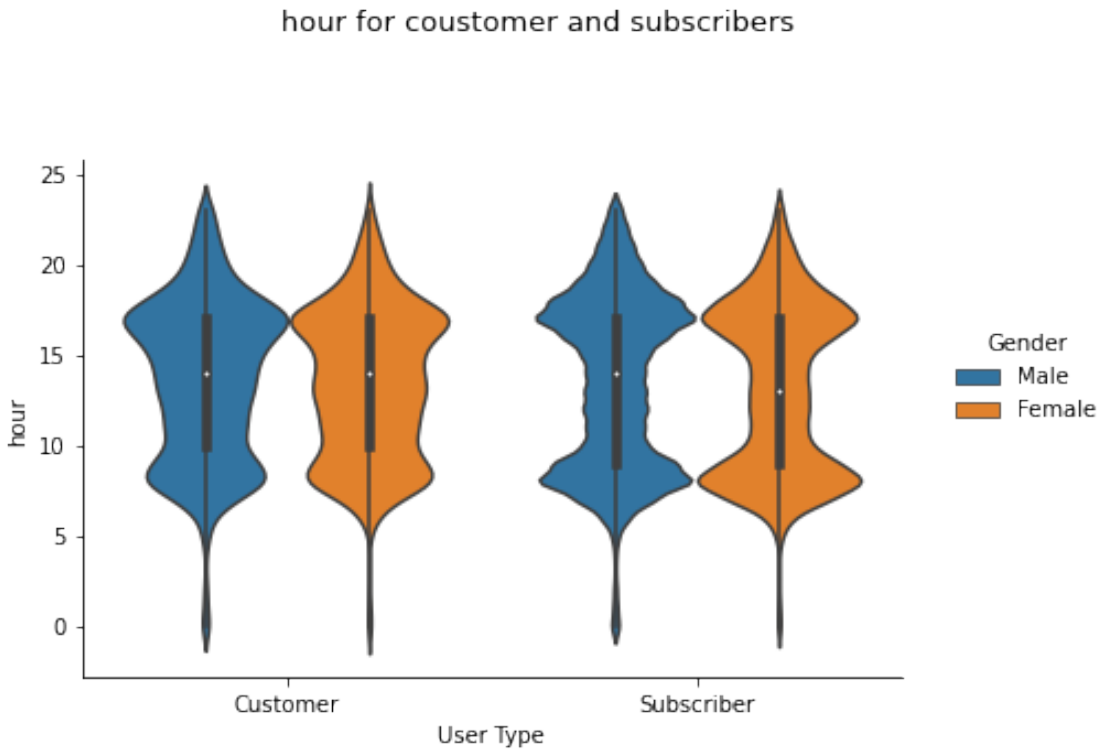
We note here that the average age of use is between 20-40

[38]:
```
sb.violinplot(data = data_clean, x='user_type', y='member_birth_year');
```



1.3.6 User Type and Gender Member

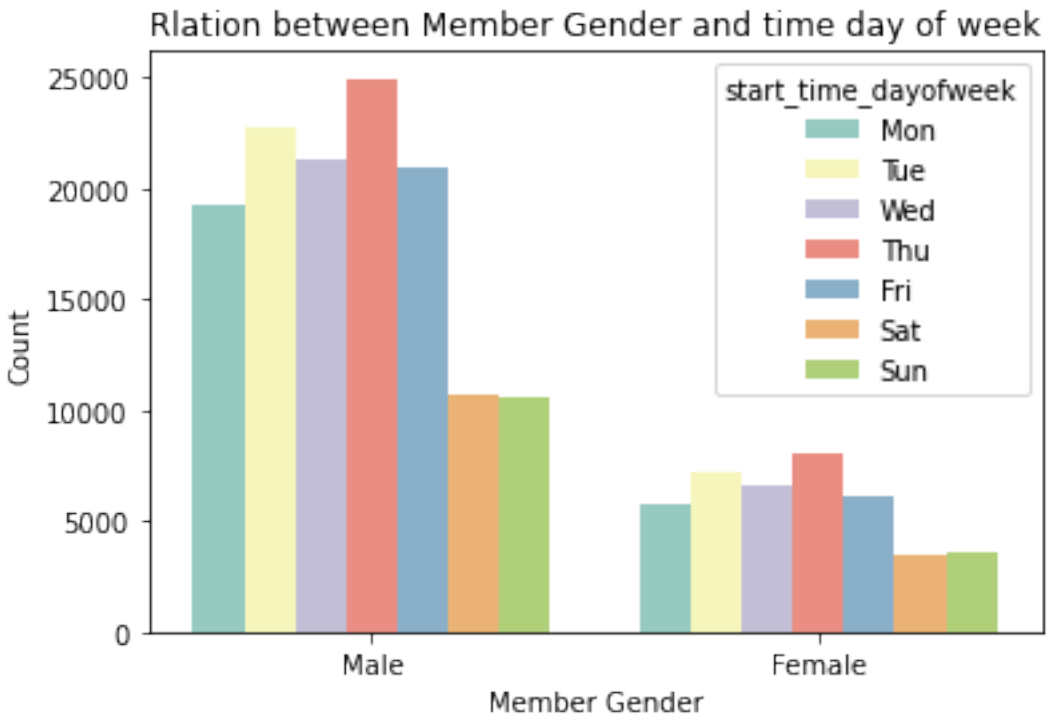
[39]:
```
z = sb.catplot(data=data_clean,height=4, aspect=1.5, x='user_type',␣
↪y="start_time_hour", hue="member_gender", kind="violin");
z.set_axis_labels("User Type", "hour")
z._legend.set_title('Gender')
z.fig.suptitle('hour for coustomer and subscribers', y=1.2, fontsize=13);
```



1.3.7 How likely is the ratio of Female to Male using during the days of the week?

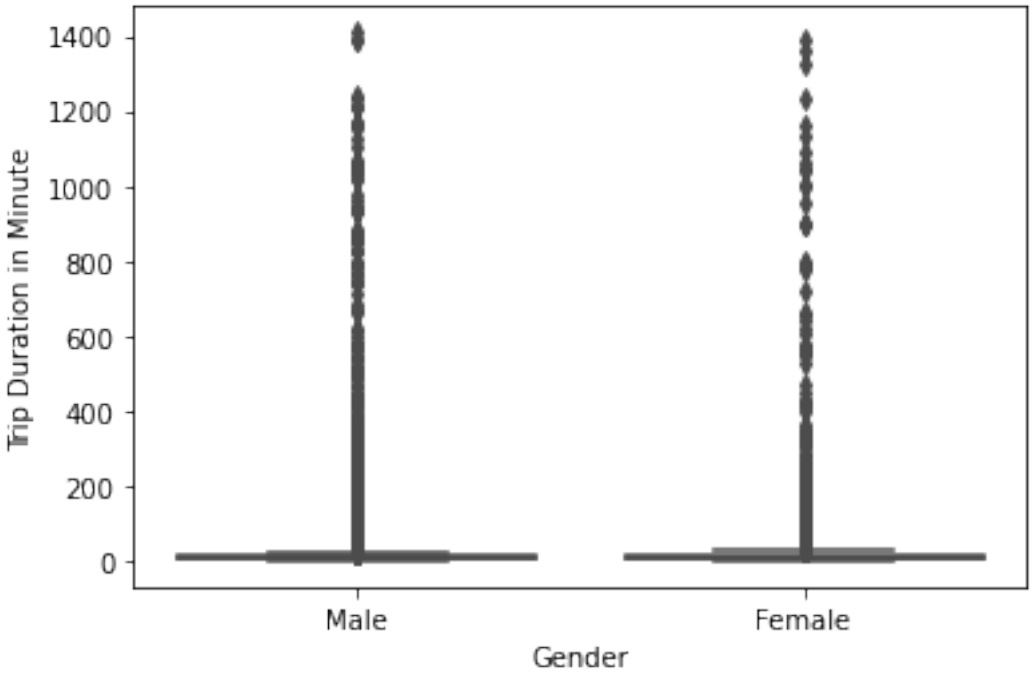
[45]:
```
sb.countplot(data = data_clean, x = 'member_gender', hue =␣
↪'start_time_dayofweek',palette="Set3");
plt.title('Rlation between Member Gender and time day of week');
plt.xlabel('Member Gender')
plt.ylabel('Count')
```

[45]:
```
Text(0, 0.5, 'Count')
```



1.3.8 What is the percentage of the use of gender types of users?

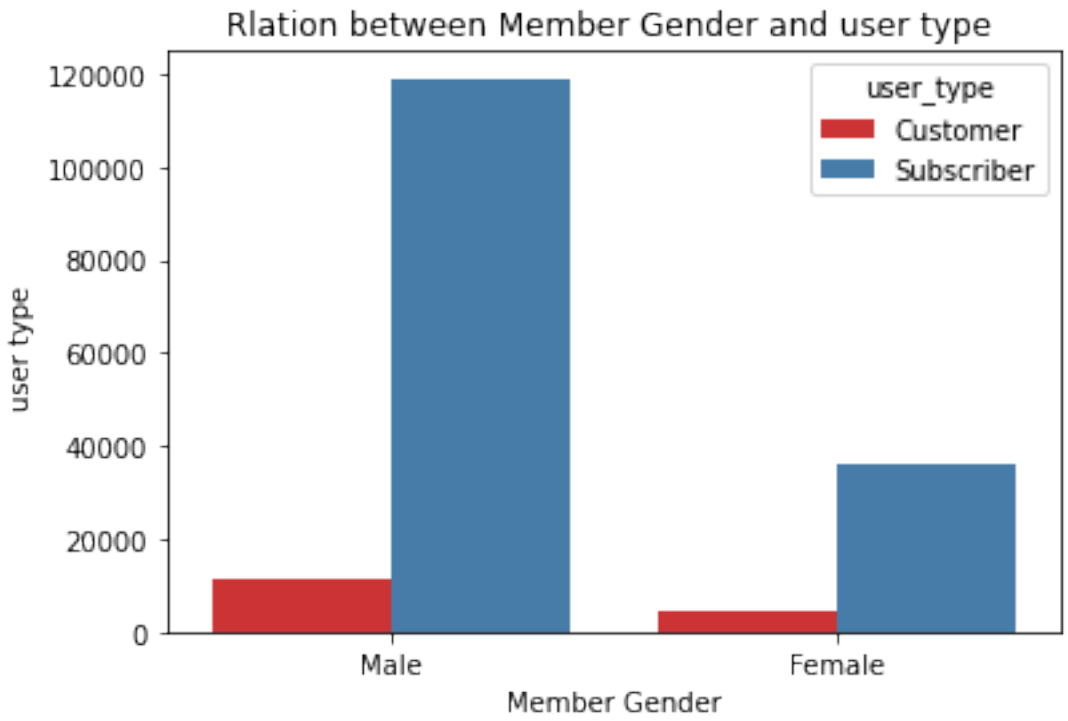
[53]:
```
sb.boxplot(data=data_clean, x='member_gender', y='duration_min', color='blue');
plt.xlabel('Gender');
plt.ylabel('Trip Duration in Minute');
```



1.3.9 What is the ratio of the type of user between Male and Female?

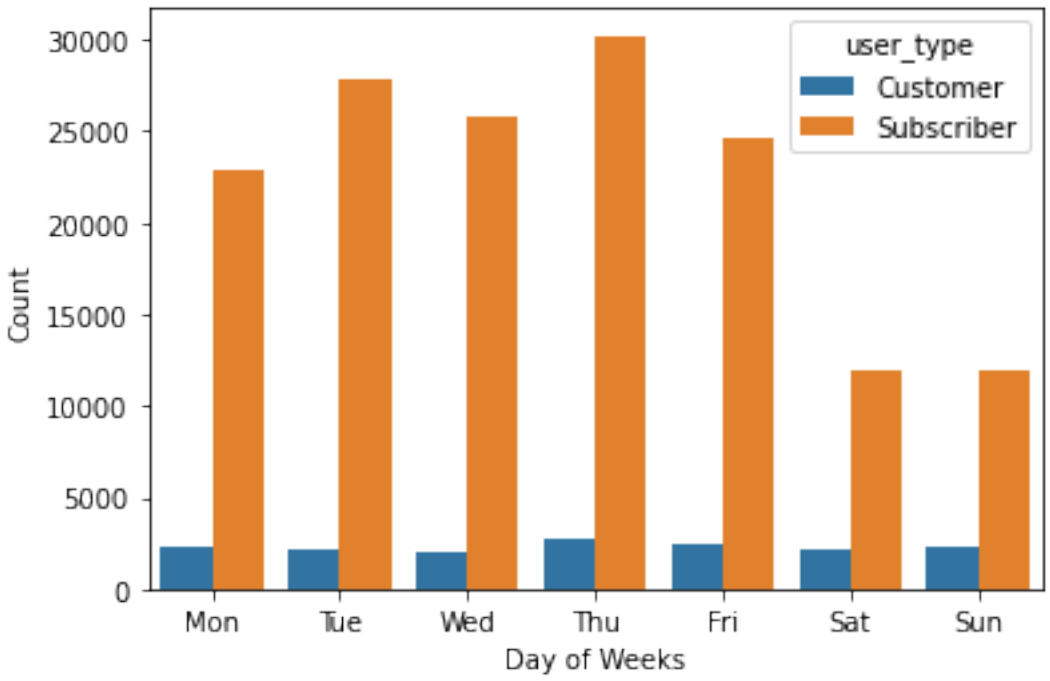
[54]:
```
sb.countplot(data = data_clean, x = 'member_gender', hue =␣
↪'user_type',palette="Set1");
plt.title('Rlation between Member Gender and user type ');
plt.xlabel('Member Gender')
plt.ylabel('user type ')
```

[54]:
```
Text(0, 0.5, 'user type ')
```



1.3.10 Percentage of user type usage with days of the week

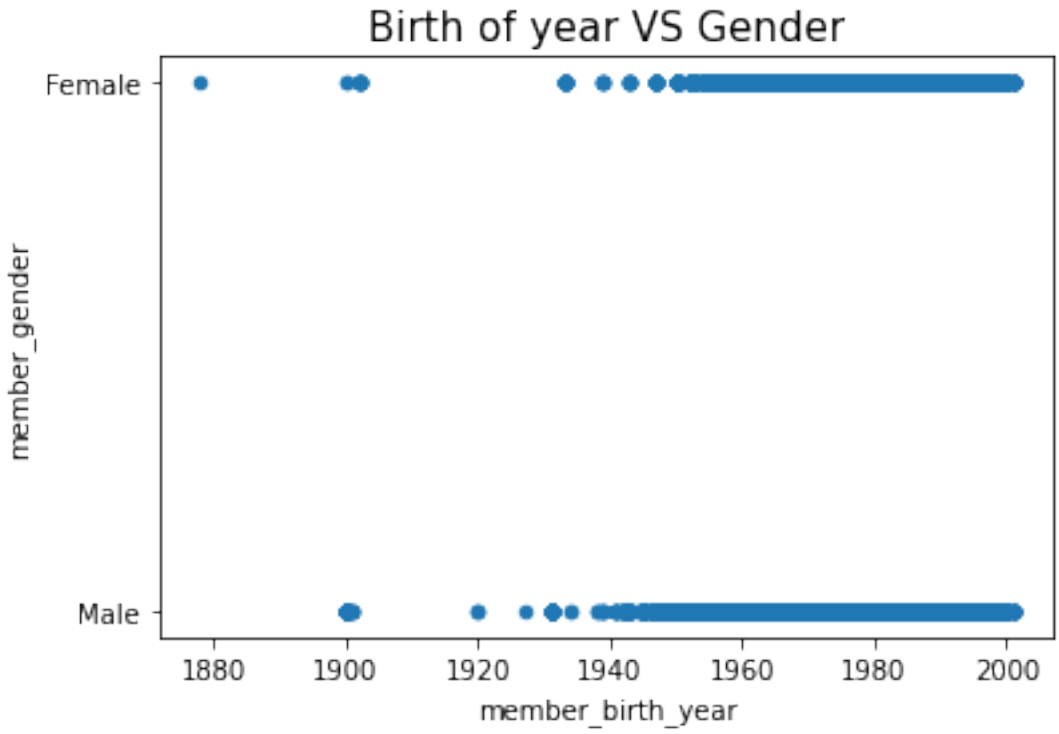
[57]:
```
sb.countplot(data=data_clean, x='start_time_dayofweek', hue='user_type');
plt.xlabel('Day of Weeks');
plt.ylabel('Count');
```



1.3.11 Comparing the percentage of years and type of users between men and women

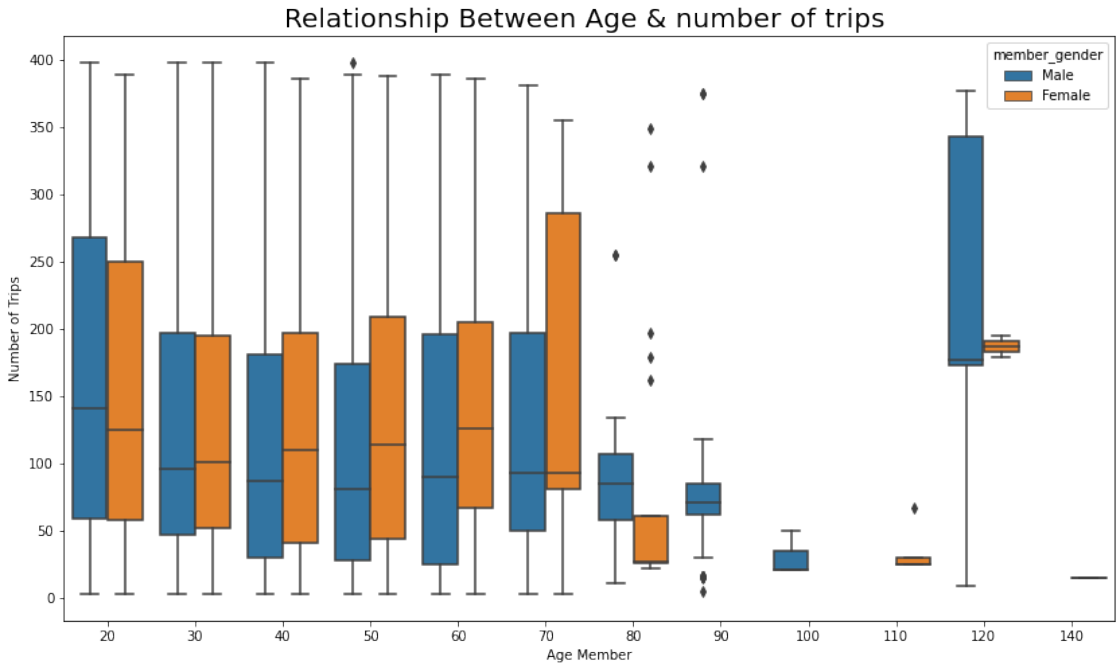
[59]:
```
data_clean.plot(x='member_birth_year', y='member_gender', kind='scatter').
↪set_title("Birth of year VS Gender",size=15)
```

[59]:
```
Text(0.5, 1.0, 'Birth of year VS Gender')
```



1.3.12 The percentage of the number of trips compared to the ages that use this type of trips

[60]:
```
data_clean['member_age_int']=data_clean['member_age'].apply(lambda x:(x//
 ↪10)*10).astype(int)
f, g =plt.subplots(figsize=(14,8))
g= sb.boxplot(data=data_clean,␣
 ↪y='start_station_id',x='member_age_int',hue='member_gender')
plt.title('Relationship Between Age & number of trips ', fontsize=20)
plt.xlabel('Age Member', fontsize=10);
plt.ylabel('Number of Trips', fontsize=10);
```



We note here that the percentage of Male and Female in the age of 30 is very close in using bike trips, and from 40-50 the percentage of Female is considered to be slightly higher than Male, and the percentage of ages between 20-70 is the most used age for these bike trips

1.4 Findings exploratory visualizations

At first, we note that there is a great demand for the use of bicycles and that they are very popular, and that they have provided many services to many users of both sexes, and this is evidence that they provide great service and benefit. We also noted that the percentage of subscribers is 90.6% and the percentage of customers is 9. 4%, and the difference between them does not exceed 9%. We also noticed that the most days on which bicycles are used, which is Thursday, was the most used day, and Friday and Tuesday are close to it. These days are widely used and in close proportions. We discussed the consideration between the two sexes and which of them had a significant impact on the use of bicycles, and the percentage of men was higher than women in using this type of bicycle, a "recommendation" and women must be motivated to use this type, and motivational plans must be developed for women by presenting offers On Mother's Day and other holidays that may be a reason for women to use this kind of grades. One of the important observations that we addressed through the analysis is the percentage of ages that are dealt with and are used more, and the percentage of ages ranged between 20-60 We recommend distributing these ages and motivating them to continue in them, especially the ages between 40-60, by placing advertisements for this age group to motivate them to be active and maintain health in using this type of bicycle.

[ ]: