Communicate Data Findings (Ford GoBike System Data)

by (Amal Aljabri)

Preliminary Wrangling

This document explores a dataset that includes information about individual rides made in a bike-sharing system covering the greater San Francisco Bay area.

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
from matplotlib import rc
import seaborn as sns

%matplotlib inline
```

In [2]:
```
from IPython.core.display import display
sns.set(rc={'figure.figsize':(14,6)})
```

In [3]:
```
# load in the dataset into a pandas dataframe
df = pd.read_csv('201902-fordgobike-tripdata.csv')
```

Dataset Overview

In [4]:
```
# high-level overview of data shape and composition
print(df.shape)
print(df.dtypes)
print(df.info())
df.head(10)
```

```
(183412, 16)
duration_sec                 int64
start_time                  object
end_time                    object
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
<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
None
```

Out[4]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station
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	1
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	8
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	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	7
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	22
5	1793	2019-02-28 23:49:58.6320	2019-03-01 00:19:51.7600	93.0	4th St at Mission Bay Blvd S	37.770407	-122.391198	32
6	1147	2019-02-28 23:55:35.1040	2019-03-01 00:14:42.5880	300.0	Palm St at Willow St	37.317298	-121.884995	31
7	1615	2019-02-28 23:41:06.7660	2019-03-01 00:08:02.7560	10.0	Washington St at Kearny St	37.795393	-122.404770	12
8	1570	2019-02-28 23:41:48.7900	2019-03-01 00:07:59.7150	10.0	Washington St at Kearny St	37.795393	-122.404770	12
9	1049	2019-02-28 23:49:47.6990	2019-03-01 00:07:17.0250	19.0	Post St at Kearny St	37.788975	-122.403452	12

In [5]:
```
# descriptive statistics for numeric variables
df.describe()
```

Out[5]:

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id	end_station_latitude	end_station_longit
count	183412.000000	183215.000000	183412.000000	183412.000000	183215.000000	183412.000000	183412.000
mean	726.078435	138.590427	37.771223	-122.352664	136.249123	37.771427	-122.352
std	1794.389780	111.778864	0.099581	0.117097	111.515131	0.099490	0.116
min	61.000000	3.000000	37.317298	-122.453704	3.000000	37.317298	-122.453
25%	325.000000	47.000000	37.770083	-122.412408	44.000000	37.770407	-122.411
50%	514.000000	104.000000	37.780760	-122.398285	100.000000	37.781010	-122.398
75%	796.000000	239.000000	37.797280	-122.286533	235.000000	37.797320	-122.288
max	85444.000000	398.000000	37.880222	-121.874119	398.000000	37.880222	-121.874

In [6]:
```
# Check if duplicates exist
df.duplicated().sum()
```

Out[6]: 0

In [7]:
```
# Check if unique values
df.nunique()
```

Out[7]:
```
duration_sec                 4752
start_time                 183401
end_time                   183397
start_station_id              329
start_station_name            329
start_station_latitude        334
start_station_longitude       335
end_station_id                329
end_station_name              329
end_station_latitude          335
end_station_longitude         335
bike_id                      4646
user_type                       2
member_birth_year              75
member_gender                   3
bike_share_for_all_trip         2
dtype: int64
```

In [8]:
```
# Check if missing values
df.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 = df.astype({'start_time': 'datetime64', 'end_time': 'datetime64'}, copy=False)
df = df[~ (df['start_station_id'].isnull())]
df = df[~ (df['member_birth_year'].isnull() | df['member_gender'].isnull())]
df = df.query('member_birth_year > 1931')
```

In [10]:
```
df['week_day'] = df['start_time'].dt.day_name()

weekday = ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
weekday_classes = pd.api.types.CategoricalDtype(ordered = True, categories = weekday)

df['week_day'] = df['week_day'].astype(weekday_classes)
```

In [11]:
```
df['start_hour'] = df['start_time'].dt.hour
df['duration_minutes'] = df['duration_sec'] / 60
df['member_age'] = 2021 - df['member_birth_year']
```

In [12]:
```
df.isnull().sum()
```

Out[12]:
```
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
week_day                   0
month                      0
start_hour                 0
duration_minutes           0
member_age                 0
dtype: int64
```

In [13]:
```
# high-level overview of data shape and composition
print(df.shape)
print(df.dtypes)
print(df.info())
df.head(10)
```

```
(174785, 21)
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
week_day                         category
month                              object
start_hour                          int64
duration_minutes                  float64
member_age                        float64
dtype: object
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174785 entries, 0 to 183411
Data columns (total 21 columns):
 #   Column                   Non-Null Count   Dtype         
---  ------                   --------------   -----         
 0   duration_sec             174785 non-null  int64         
 1   start_time               174785 non-null  datetime64[ns]
 2   end_time                 174785 non-null  datetime64[ns]
 3   start_station_id         174785 non-null  float64       
 4   start_station_name       174785 non-null  object        
 5   start_station_latitude   174785 non-null  float64       
 6   start_station_longitude  174785 non-null  float64       
 7   end_station_id           174785 non-null  float64       
 8   end_station_name         174785 non-null  object        
 9   end_station_latitude     174785 non-null  float64       
 10  end_station_longitude    174785 non-null  float64       
 11  bike_id                  174785 non-null  int64         
 12  user_type                174785 non-null  object        
 13  member_birth_year        174785 non-null  float64       
 14  member_gender            174785 non-null  object        
 15  bike_share_for_all_trip  174785 non-null  object        
 16  week_day                 174785 non-null  category      
 17  month                    174785 non-null  object        
 18  start_hour               174785 non-null  int64         
 19  duration_minutes         174785 non-null  float64       
 20  member_age               174785 non-null  float64       
dtypes: category(1), datetime64[ns](2), float64(9), int64(3), object(6)
memory usage: 28.2+ MB
None
```

Out[13]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_
0	52185	2019-02-28 17:32:10.145	2019-03-01 08:01:55.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.789625	-122.400811	13
2	61854	2019-02-28 12:13:13.218	2019-03-01 05:24:08.146	86.0	Market St at Dolores St	37.769305	-122.426826	3
3	36490	2019-02-28 17:54:26.010	2019-03-01 04:02:36.842	375.0	Grove St at Masonic Ave	37.774836	-122.446546	70
4	1585	2019-02-28 23:54:18.549	2019-03-01 00:20:44.074	7.0	Frank H Ogawa Plaza	37.804562	-122.271738	222
5	1793	2019-02-28 23:49:58.632	2019-03-01 00:19:51.760	93.0	4th St at Mission Bay Blvd S	37.770407	-122.391198	323
6	1147	2019-02-28 23:55:35.104	2019-03-01 00:14:42.588	300.0	Palm St at Willow St	37.317298	-121.884995	312
7	1615	2019-02-28 23:41:06.766	2019-03-01 00:08:02.756	10.0	Washington St at Kearny St	37.795393	-122.404770	127
8	1570	2019-02-28 23:41:48.790	2019-03-01 00:07:59.715	10.0	Washington St at Kearny St	37.795393	-122.404770	127
9	1049	2019-02-28 23:49:47.699	2019-03-01 00:07:17.025	19.0	Post St at Kearny St	37.788975	-122.403452	121
10	458	2019-02-28 23:57:57.211	2019-03-01 00:05:35.435	370.0	Jones St at Post St	37.787327	-122.413278	43

10 rows × 21 columns

What is the structure of your dataset?

The dataset includes information about individual rides made in a bike-sharing system covering the greater San Francisco Bay area.

The dataset contains these key columns:

- Member Year of Birth
- Member Gender
- User Type (Subscriber or Customer)
- Trip Duration(in seconds)
- Bike ID
- Bike Share for All Trip
- Start Time
- End Time
- Start Station ID
- End Station ID
- Start Station Name
- End Station Name
- End Station Latitude
- End Station Longitude
- Start Station Latitude
- Start Station Longitude

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

- I am interested to find the relationship between days of the week, hours of the day, trip duration, user type, user age, and gender to find out new insights.

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

- Member Year of Birth
- Member Gender
- User Type (Subscriber or Customer)
- Trip Duration(in seconds)
- Start Time

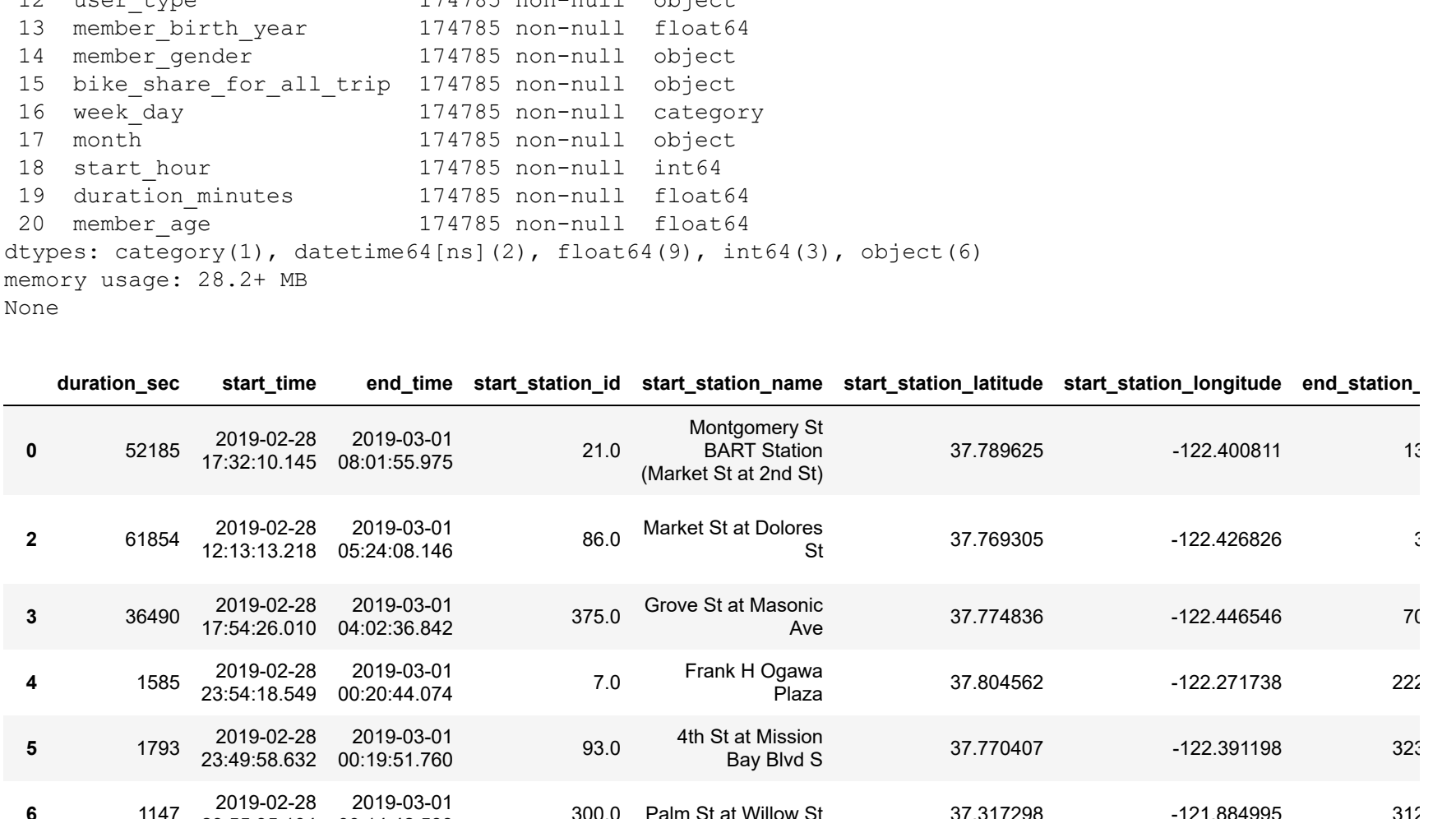
Univariate Exploration

Number of bike trips used per week day

In [21]:
```
num_bike_trips_week = df['week_day'].value_counts()
print('Number of bike trips used per week day:')
print(num_bike_trips_week)
sns.countplot(data = df, x = 'week_day')
plt.title('\nNumber of bike trips used per week day',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xlabel('Week days',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Number of bike trips',
           fontsize = 18,
           labelpad=30)
```

```
Number of bike trips used per week day:
Thursday     33699
Tuesday      30566
Wednesday    28399
Friday       27626
Monday       25593
Sunday       14492
Saturday     14410
Name: week_day, dtype: int64
```

Out[21]: Text(0, 0.5, 'Number of bike trips')



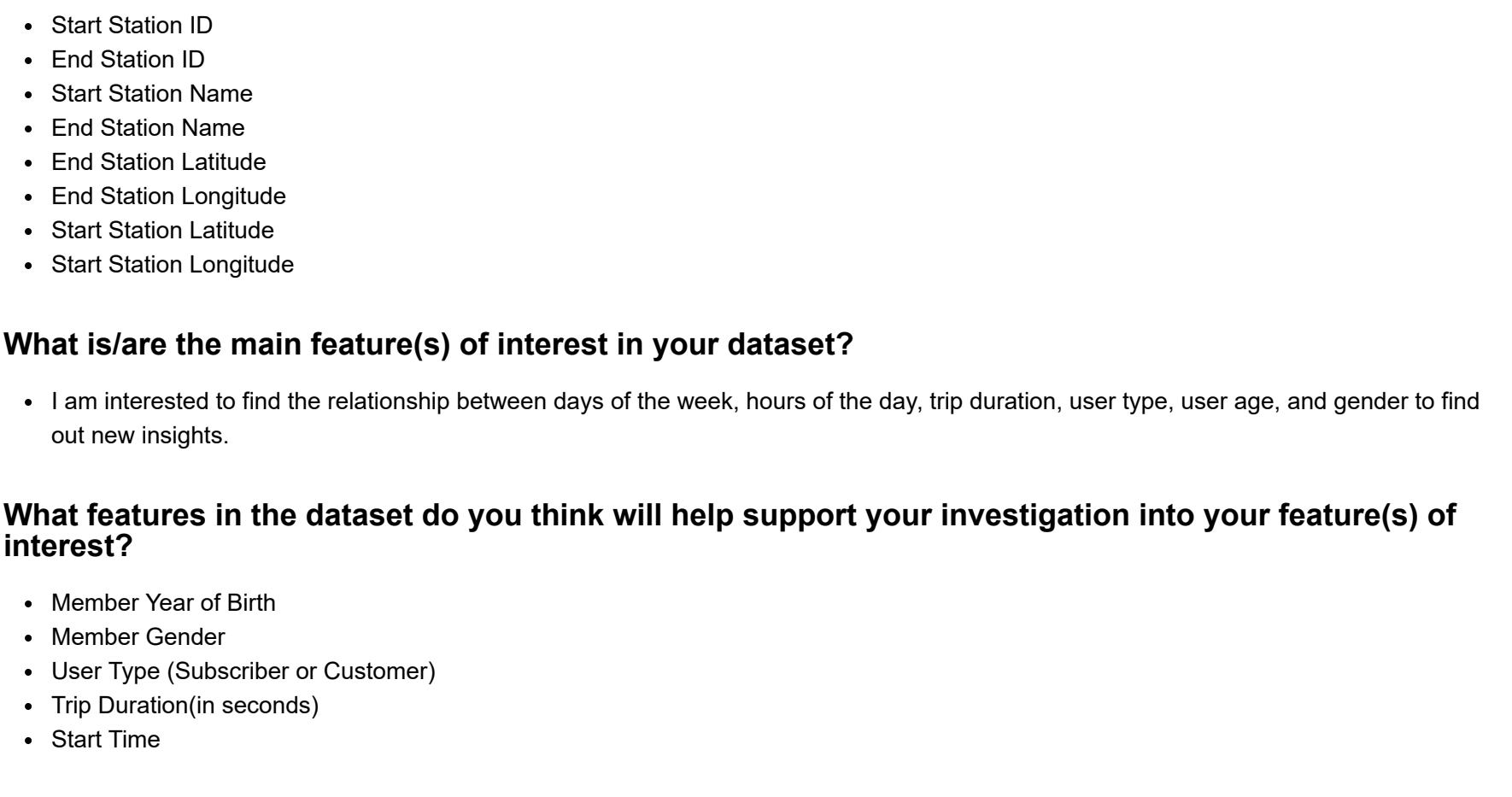
From the previous bar chart, I've concluded that the number of bike trips is higher on work days (Monday, Tuesday, Wednesday, Thursday, Friday) and lower on weekends days (Saturday, Sunday) in San Francisco.

Distribution of bike trips used per day hour

In [15]:
```
num_bike_trips_hour = df['start_hour'].value_counts()
print('Distribution of bike trips used per day hour:')
print(num_bike_trips_hour.sort_values())
plt.hist(data = df, x = 'start_hour', bins = np.arange(0,23+2,1))
plt.title('\nDistribution of bike trips used per day hour',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xticks(np.arange(0,23+2,1))
plt.xlabel('Hour of day',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Number of bike trips',
           fontsize = 18,
           labelpad=30)
```

```
 Distribution of bike trips used per day hour:
3       164
4       227
2       355
1       525
5       866
0       892
23     1572
22     2793
6      3289
21     4399
20     6210
11     7452
14     7657
10     7964
13     8075
12     8213
15     8621
19     9418
7     10226
16    13454
9     15196
18    16104
8     20218
17    20895
Name: start_hour, dtype: int64
```

Out[15]: Text(0, 0.5, 'Number of bike trips')



From the previous histogram, I've concluded that the distribution of bike trips used per day hour is popular between 8-9 AM and 4-6 PM.

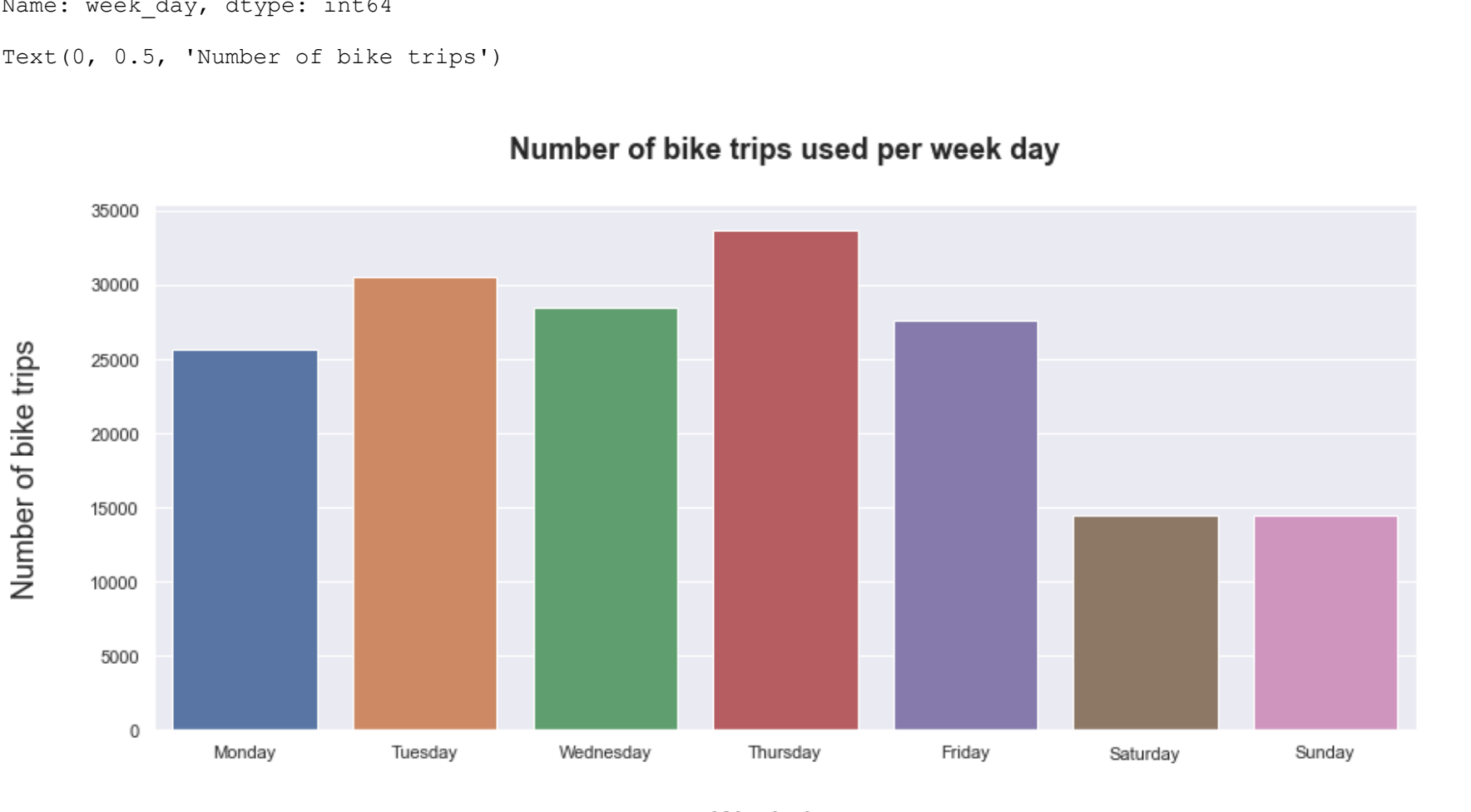
Ages distribution of bike trips

In [16]:
```
ages_distribution = df['member_age'].value_counts()
print('Ages distribution of bike trips:')
print(ages_distribution.sort_values())
plt.hist(data = df, x = 'member_age', bins = np.arange(0, df['member_age'].max() + 5, 5))
plt.title('\nAges distribution of bike trips',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xticks(np.arange(0, df['member_age'].max() + 5, 5))

plt.xlabel('Age',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Number of member',
           fontsize = 18,
           labelpad=30)
```

```
Ages distribution of bike trips:
87.0        2
77.0        2
83.0        3
80.0        9
82.0       11
        ...  
30.0     8484
31.0     8640
32.0     8967
28.0     9323
33.0    10214
Name: member_age, Length: 65, dtype: int64
```

Out[16]: Text(0, 0.5, 'Number of member')



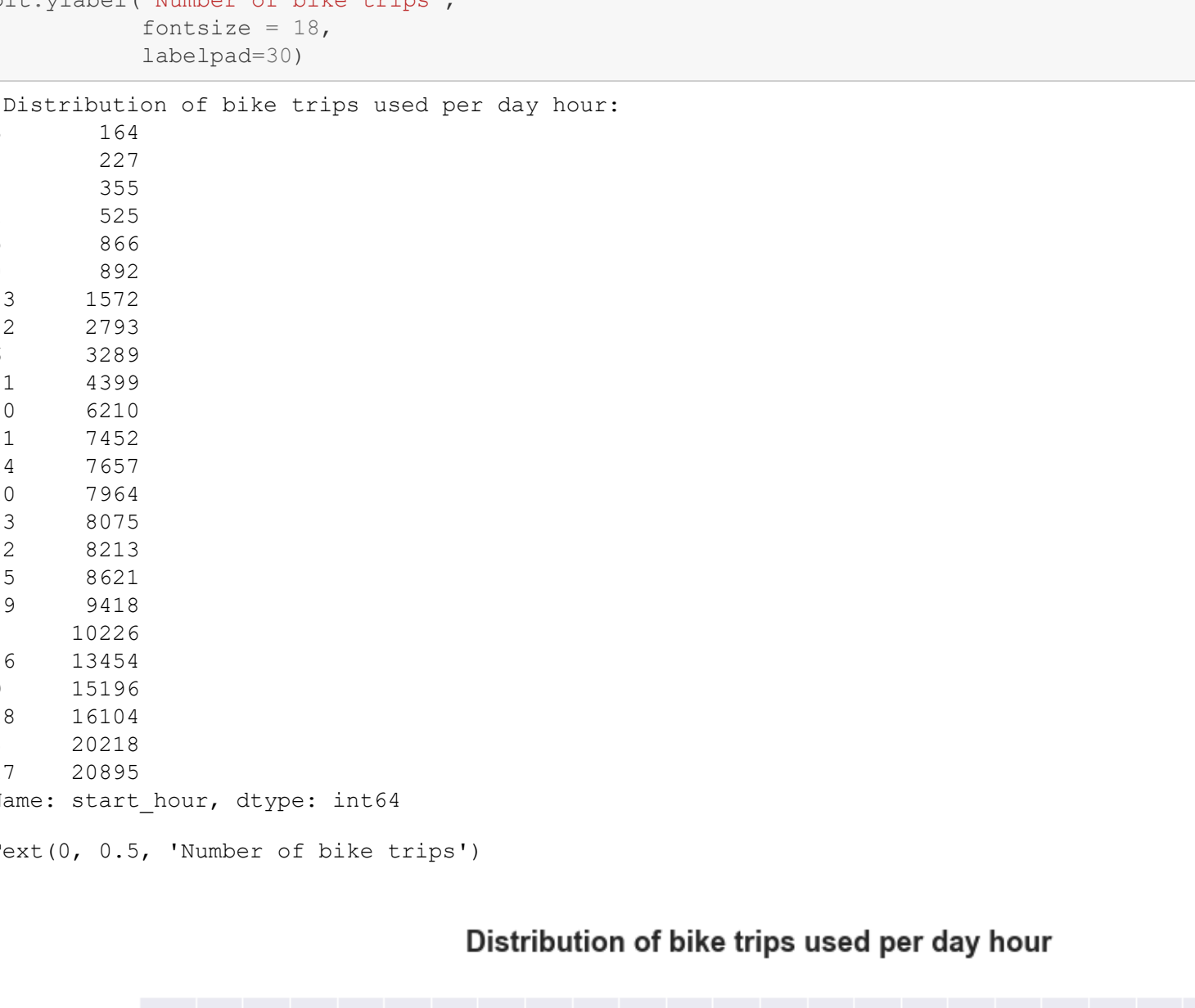
From the previous histogram, I've concluded that the ages distribution of bike trips is most popular with the age group between 25-40 years old.

Gender distribution of bike trips

In [17]:
```
member_gender_counts = df['member_gender'].value_counts()
print('Gender distribution of bike trips:')
print(member_gender_counts.sort_values())
plt.figure(figsize=(10,10))
plt.pie(member_gender_counts,
        labels = member_gender_counts.index,
        counterclock = False,
        autopct = '%1.1f%%',
        wedgeprops = {'width' : 0.2})
plt.legend(loc="center right", bbox_to_anchor=(1.3,0.5) ,fontsize=18)
plt.title('\nGender distribution of bike trips',
          fontsize = 18,
          fontweight = 'bold')
```

```
Gender distribution of bike trips:
Other       3644
Female     40791
Male      130350
Name: member_gender, dtype: int64
```

Out[17]: Text(0.5, 1.0, '\nGender distribution of bike trips')



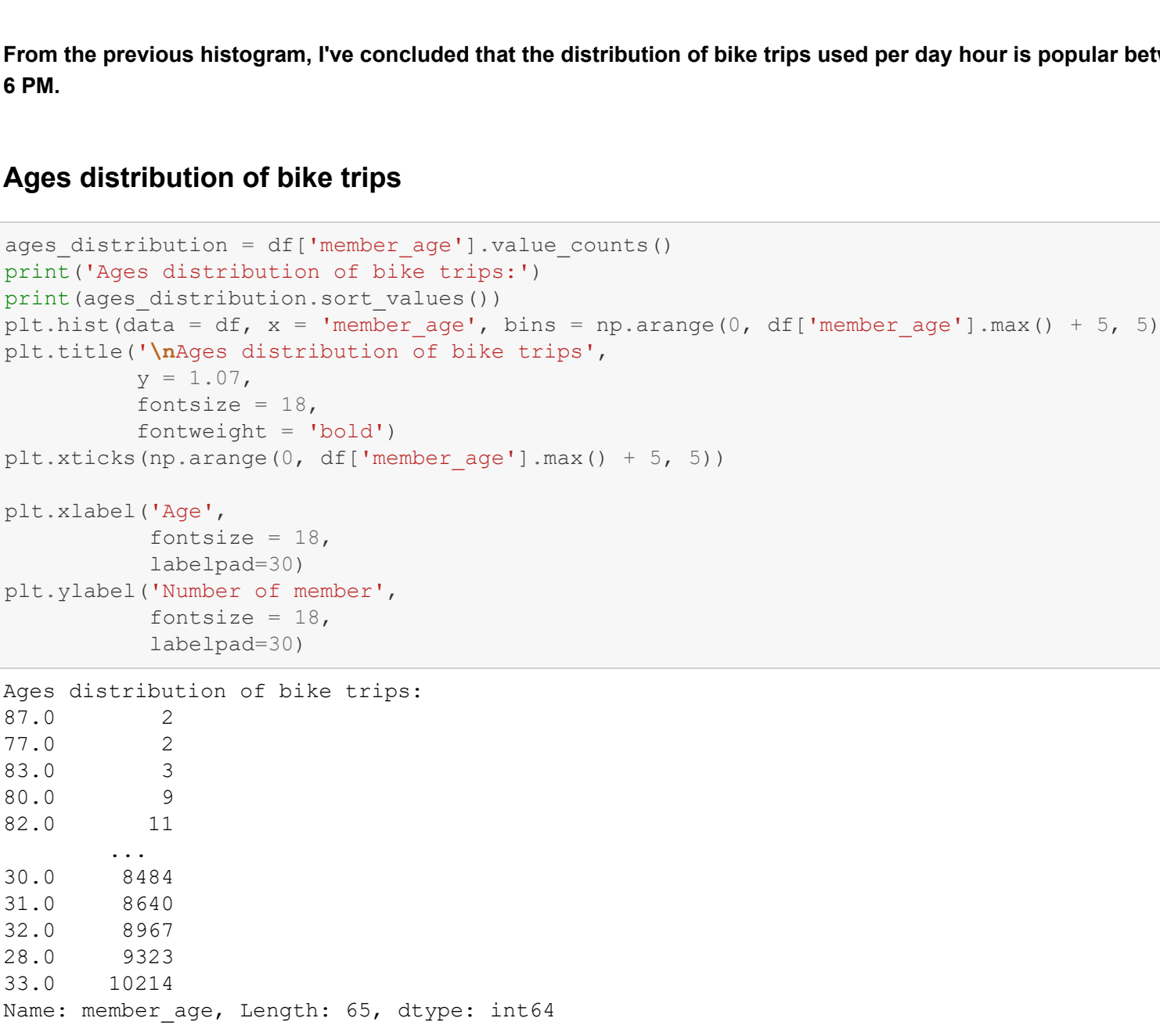
From the previous pie chart, I've concluded that the gender distribution of bike trips was (74.6%) are males, while females are (23.3%).

User types distribution of bike trips

In [18]:
```
user_type_counts = df['user_type'].value_counts()
print('User types distribution of bike trips:')
print(user_type_counts.sort_values())
plt.figure(figsize=(10,10))
plt.pie(user_type_counts,
        labels = user_type_counts.index,
        counterclock = False,
        autopct = '%1.1f%%',
        wedgeprops = {'width' : 0.2})
plt.legend(loc="center right", bbox_to_anchor=(1.3,0.5) ,fontsize=18)
plt.title('\nUser types distribution of bike trips',
          fontsize = 18,
          fontweight = 'bold')
```

```
User types distribution of bike trips:
Customer       16559
Subscriber    158226
Name: user_type, dtype: int64
```

Out[18]: Text(0.5, 1.0, '\nUser types distribution of bike trips')



From the previous pie chart, I've concluded that the user types distribution of bike trips was (90.5%) are subscriber, while customer are (9.5%).

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

- The number of bike trips is higher on work days (Monday, Tuesday, Wednesday, Thursday, Friday) and lower on weekends days (Saturday, Sunday) in San Francisco.
- The distribution of bike trips used per day hour is popular between 8-9 AM and 4-6 PM.
- The ages distribution of bike trips is most popular with the age group between 25-40 years old.
- The gender distribution of bike trips was (74.6%) are males, while females are (23.3%).
- The user types distribution of bike trips was (90.5%) are subscriber, while customer are (9.5%).

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?

- Add week_day from start_time.
- Add member_age by subtracting the current year from member_birth_year.
- Add start_hour from start_time.

Bivariate Exploration

Average bike trip duration in minutes per user type

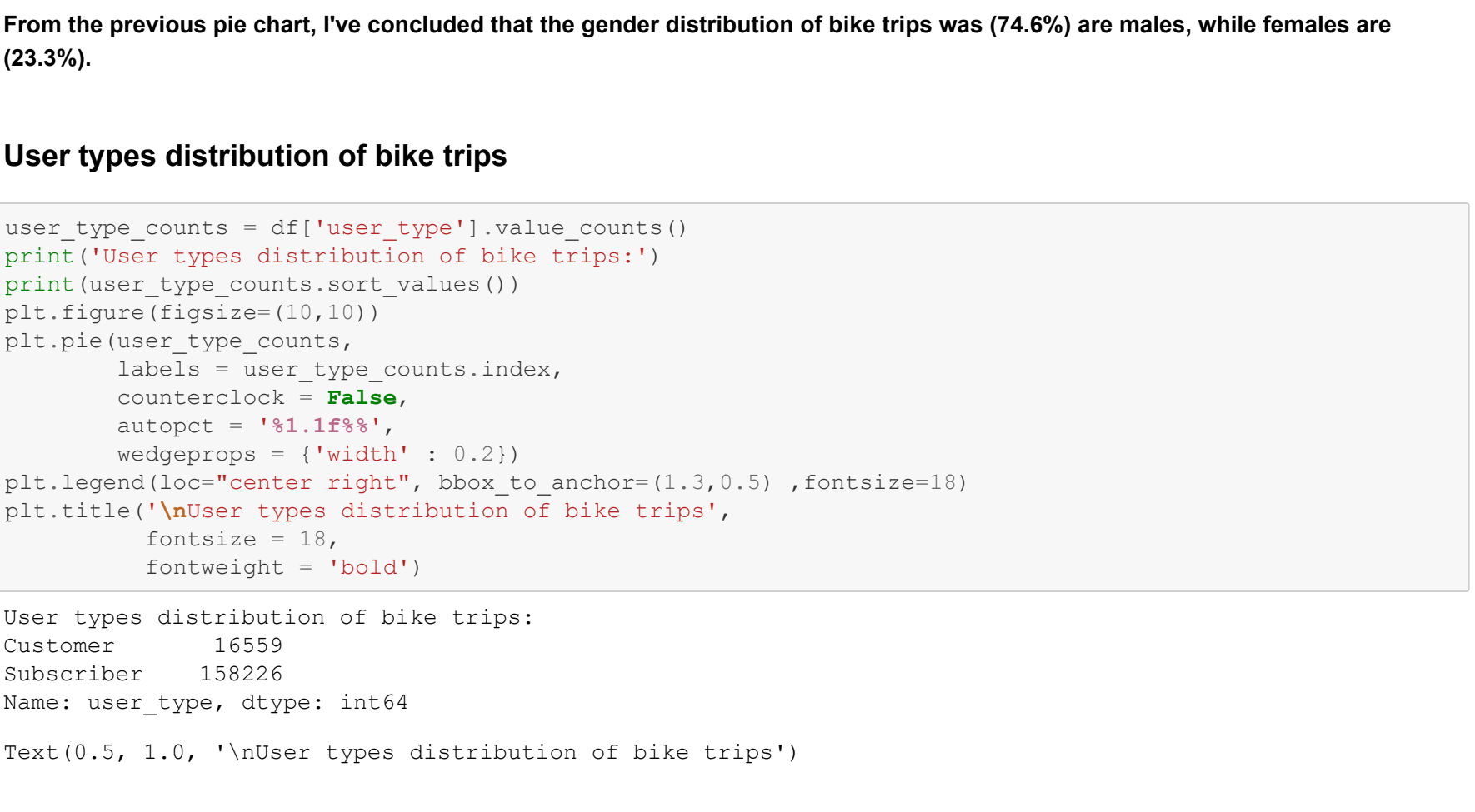
In [113]:
```
user_duration_average = df.groupby('user_type')['duration_minutes'].mean()
print('Average bike trip duration in minutes per user type:')
print(user_duration_average)

user_duration_average.plot(kind = 'barh')

plt.title('\nAverage bike trip duration in minutes per user type',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xlabel('Average duration in minutes',
           fontsize = 18,
           labelpad=30)
plt.ylabel('User type',
           fontsize = 18,
           labelpad=30)
```

```
Average bike trip duration in minutes per user type:
user_type
Customer      21.847536
Subscriber    10.679922
Name: duration_minutes, dtype: float64
```

Out[113]: Text(0, 0.5, 'User type')



From the previous barh chart, I've concluded that the average bike trip duration in minutes per user type was (10.679922) are subscriber, while customer are (21.847536).

Distribution of bike trip duration in minutes per user type

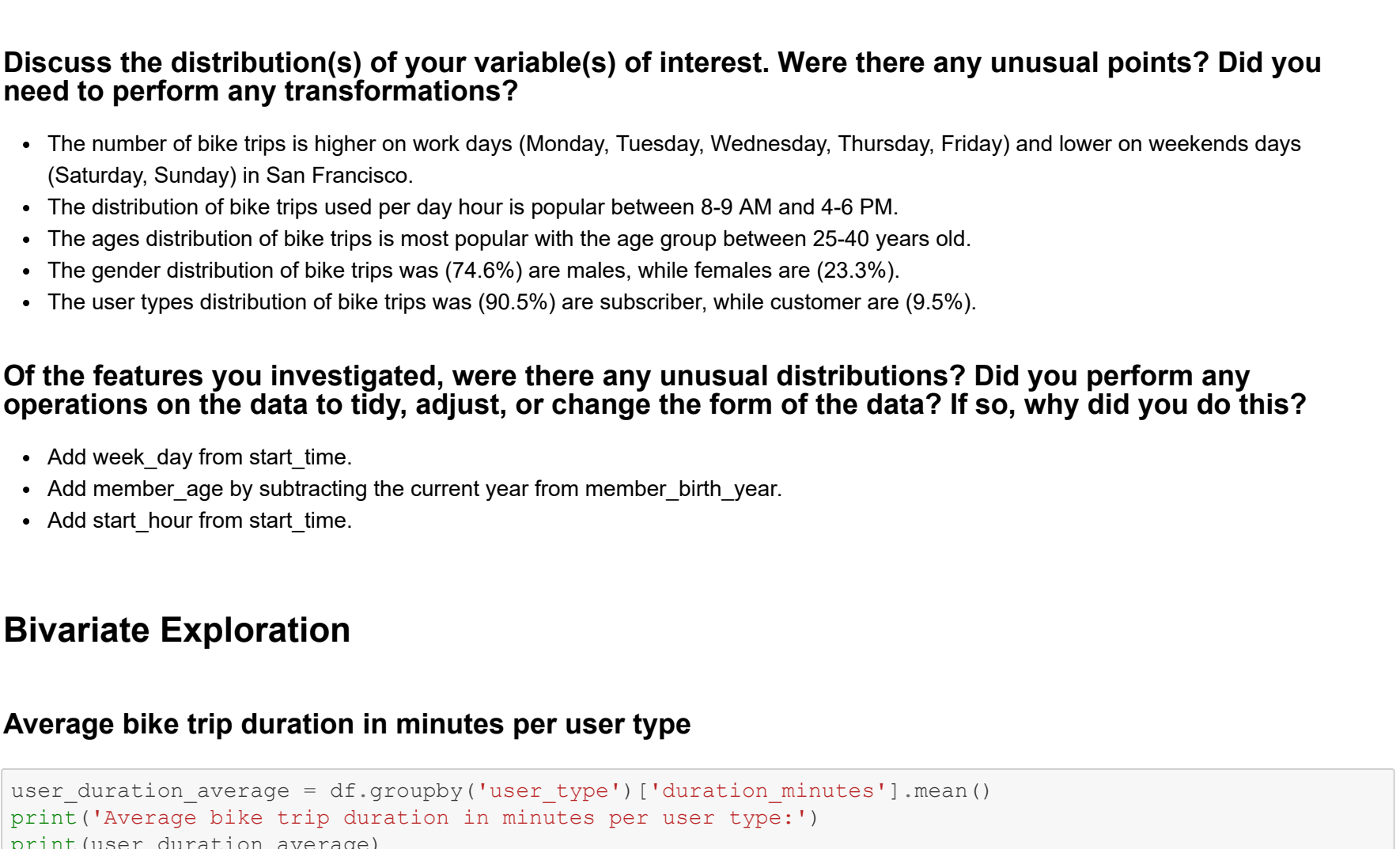
In [112]:
```
sns.boxplot(data= df, x = 'user_type', y = 'duration_minutes')

plt.xticks(rotation=15)

plt.ylim(0,40)

plt.title('\nDistribution of bike trip duration in minutes per user type',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xlabel('User type',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Average duration in minutes',
           fontsize = 18,
           labelpad=30)
```

Out[112]: Text(0, 0.5, 'Average duration in minutes')

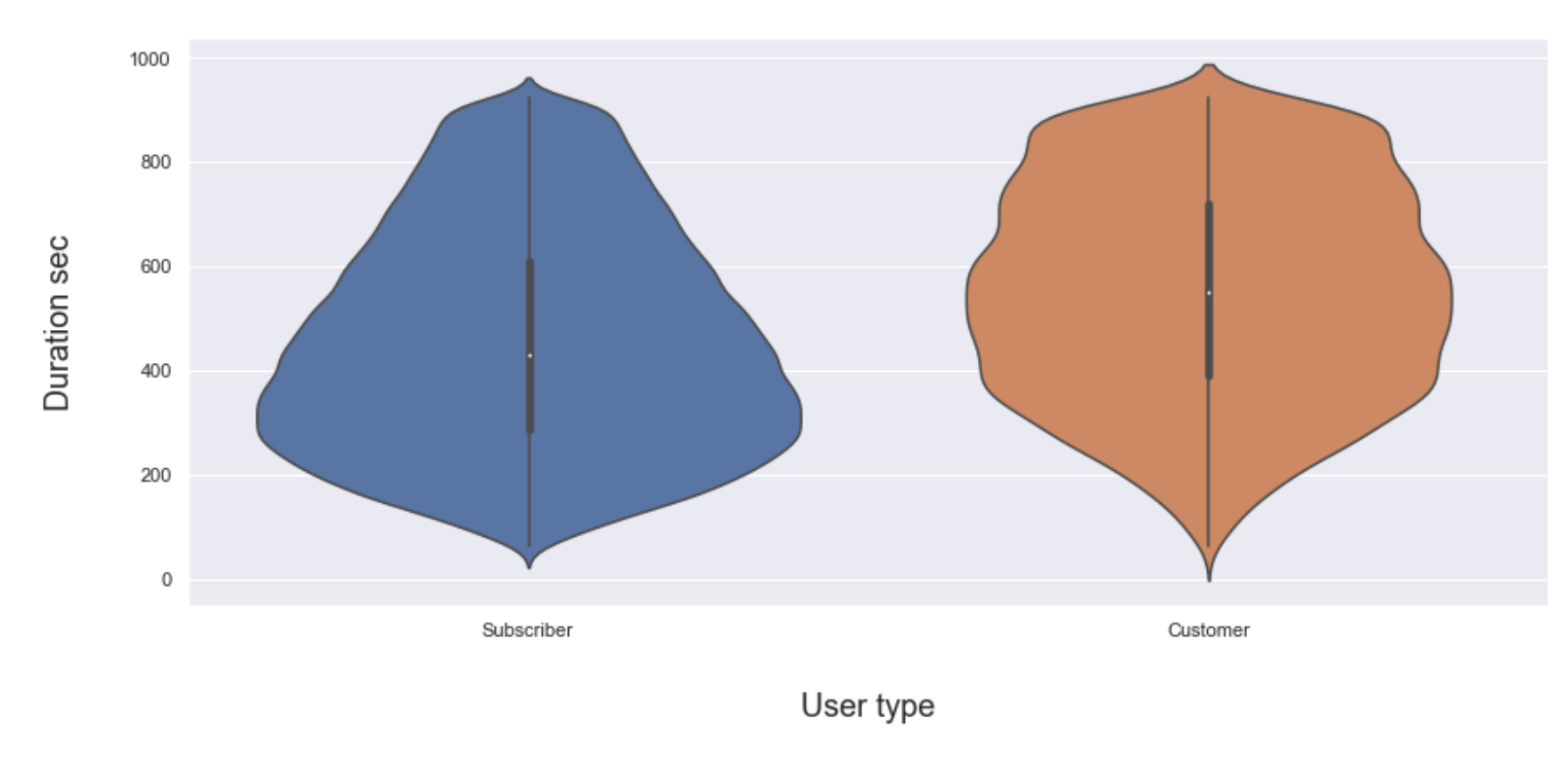


From the previous box plot, I've concluded that the distribution of bike trip duration in minutes per user type was the distribution of customer trip time shows more variability than subscribers.

Distribution of bike trip duration in sec per user type

```python
In [117]: sns.violinplot(data= df[df['duration_sec'] <= 923],x='user_type',y='duration_sec')
plt.xlabel('User type',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Duration sec',
           fontsize = 18,
           labelpad=30)
```

Out[117]: Text(0, 0.5, 'Duration sec')

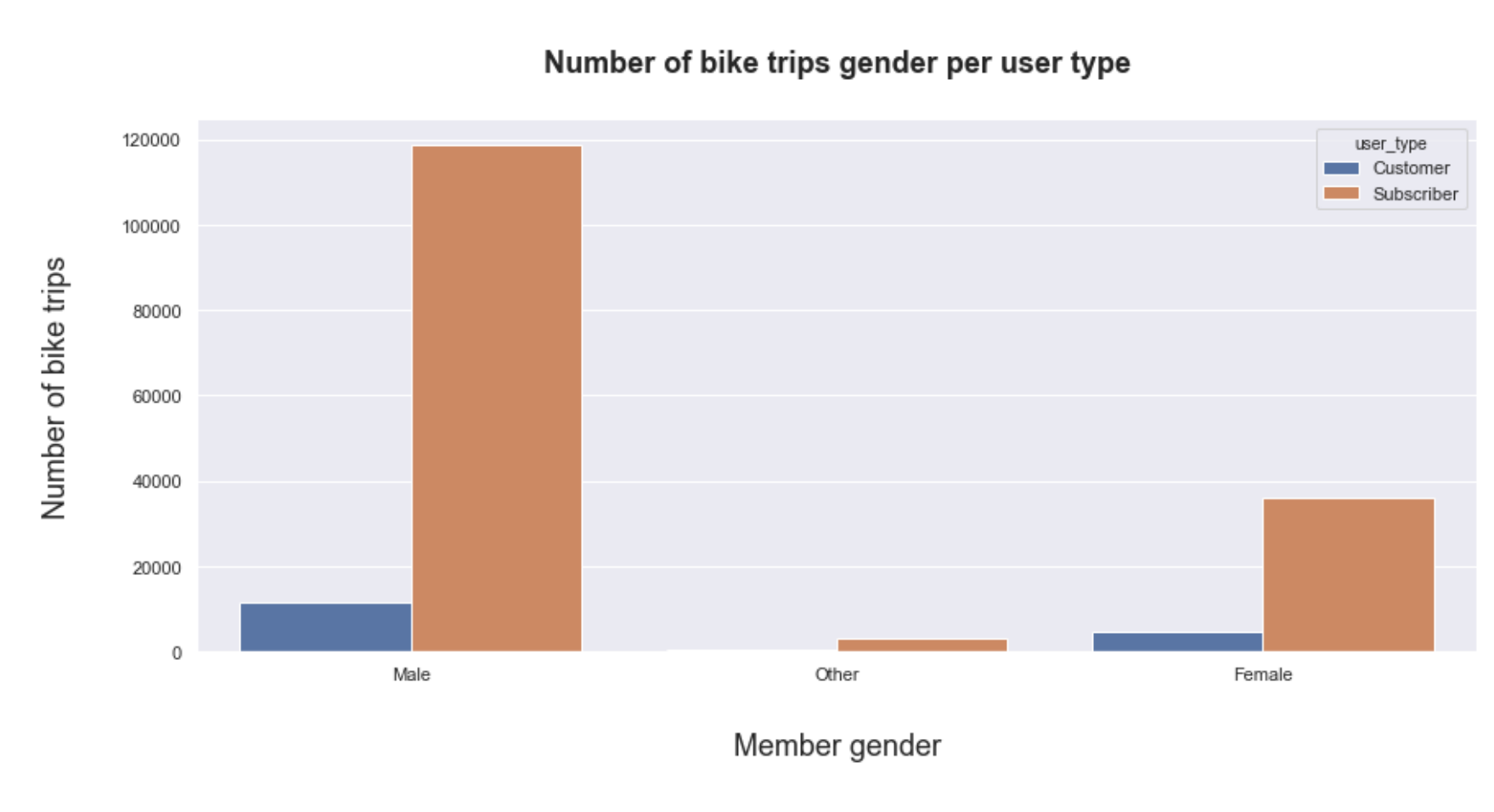


Number of bike trips gender per user type

```python
In [76]: sns.countplot(x = df['member_gender'], hue = df['user_type'])

plt.title('\nNumber of bike trips gender per user type',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xlabel('Member gender',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Number of bike trips',
           fontsize = 18,
           labelpad=30)
```

Out[76]: Text(0, 0.5, 'Number of bike trips')

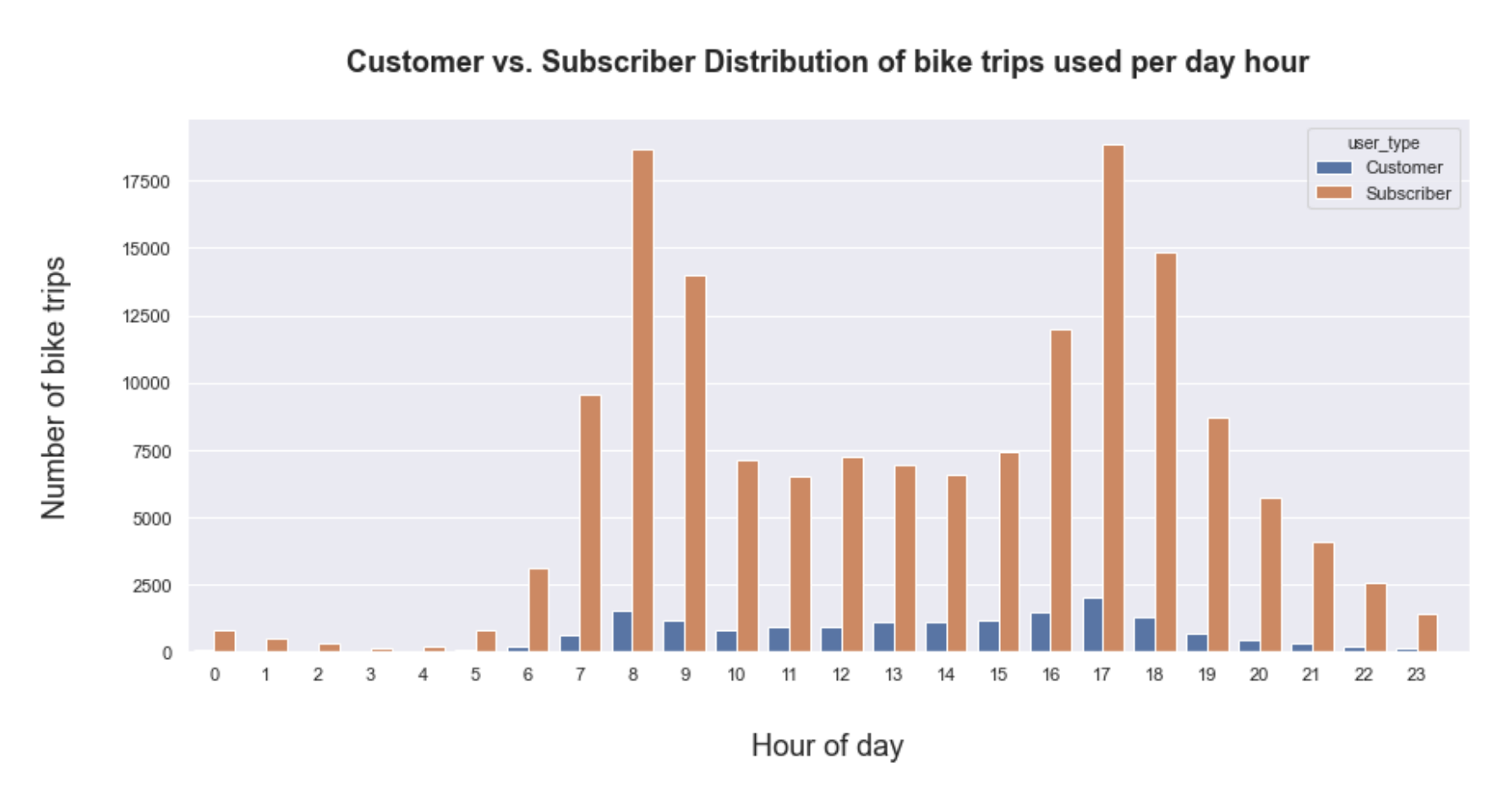


I've concluded that the number of bike trips gender per user type was a subscribers bigger than customers.

Customer vs. Subscriber distribution of bike trips used per day hour

```python
In [83]: sns.countplot(x = df['start_hour'], hue = df['user_type'])
plt.title('\nCustomer vs. Subscriber distribution of bike trips used per day hour',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xticks(np.arange(0,23+2,1))
plt.xlabel('Hour of day',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Number of bike trips',
           fontsize = 18,
           labelpad=30)
```

Out[83]: Text(0, 0.5, 'Number of bike trips')

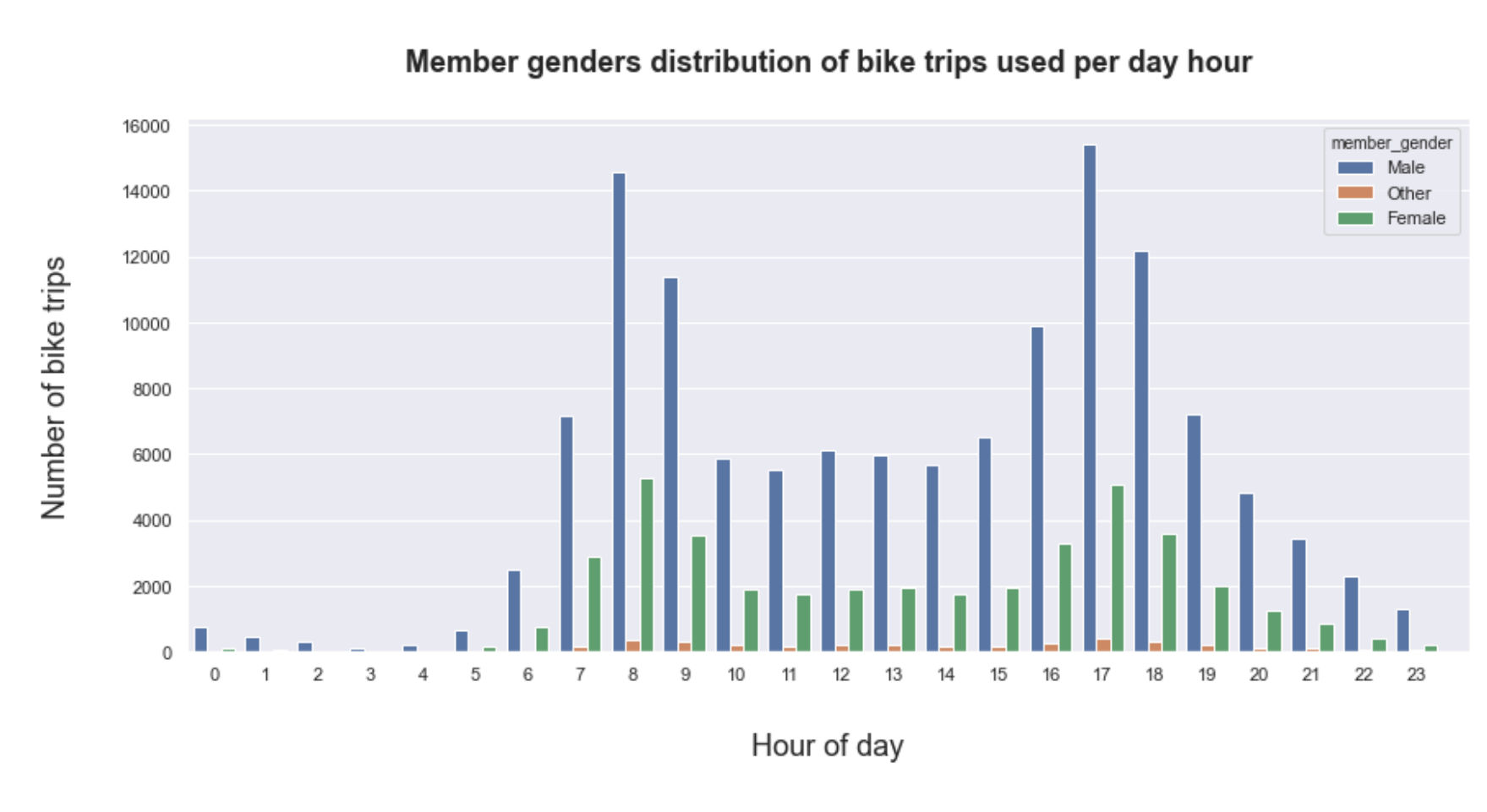


I've concluded that the distribution of bike trips used per day hour was a subscribers bigger than customers.

Member genders distribution of bike trips used per day hour

```python
In [111]: sns.countplot(x = df['start_hour'], hue = df['member_gender'])
plt.title('\nMember genders distribution of bike trips used per day hour',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xticks(np.arange(0,23+2,1))
plt.xlabel('Hour of day',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Number of bike trips',
           fontsize = 18,
           labelpad=30)
```

Out[111]: Text(0, 0.5, 'Number of bike trips')



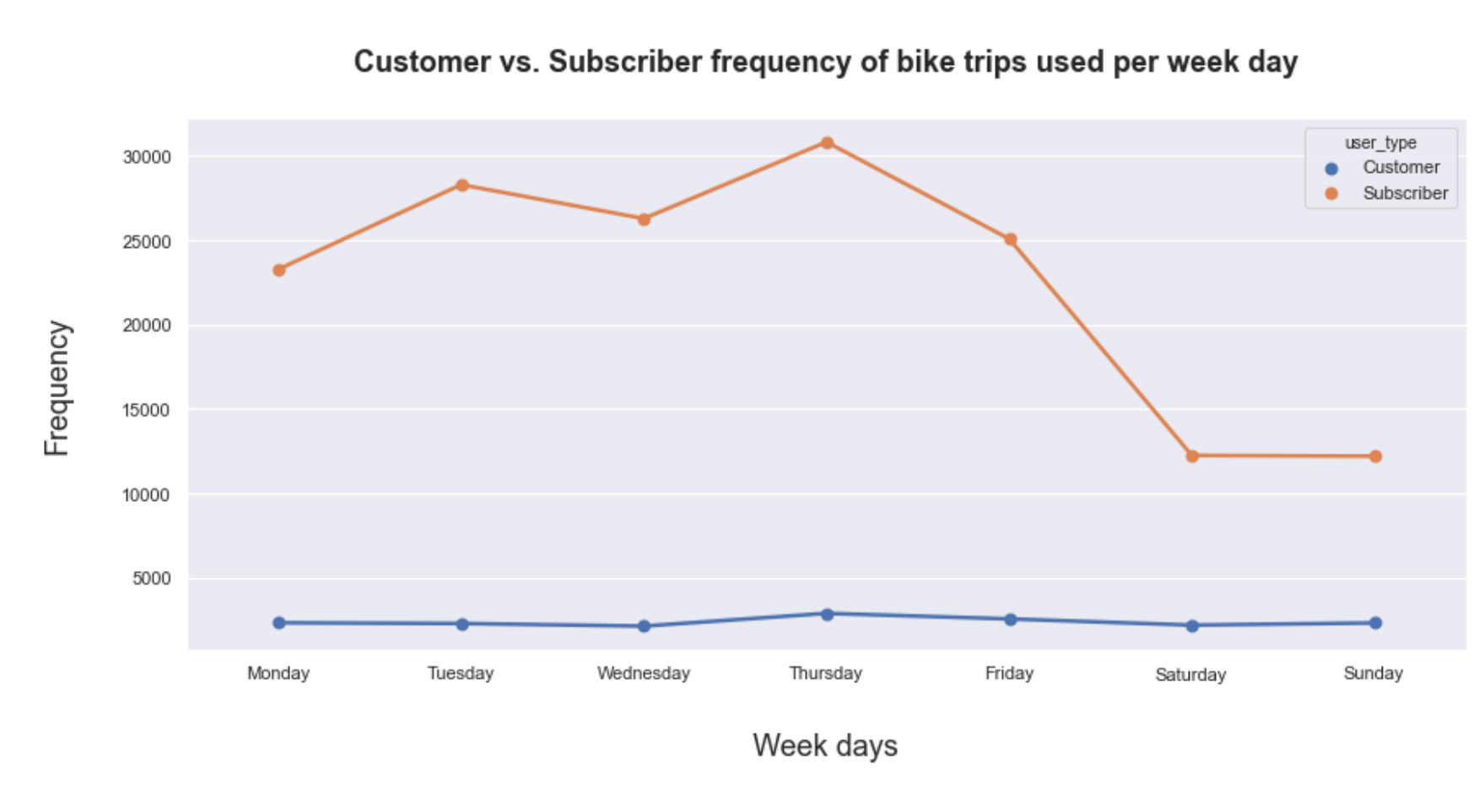
Customer vs. Subscriber frequency of bike trips used per week day

```python
In [106]: frequency1 = df.groupby(['week_day', 'user_type']).size().reset_index()
frequency1.columns = frequency1.columns[:2].append(pd.Index(['frequency']))

sns.pointplot(data=frequency1, x='week_day', y='frequency', hue = 'user_type', scale=.9);

plt.title('\nCustomer vs. Subscriber frequency of bike trips used per week day',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xlabel('Week days',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Frequency',
           fontsize = 18,
           labelpad=30)
```

Out[106]: Text(0, 0.5, 'Frequency')



- **The number of bike trips for subscribers is higher on work days (Monday, Tuesday, Wednesday, Thursday, Friday) and lower on weekends days (Saturday, Sunday) in San Francisco.**
- **The number of bike trips for customers is consistent on all week days in San Francisco.**

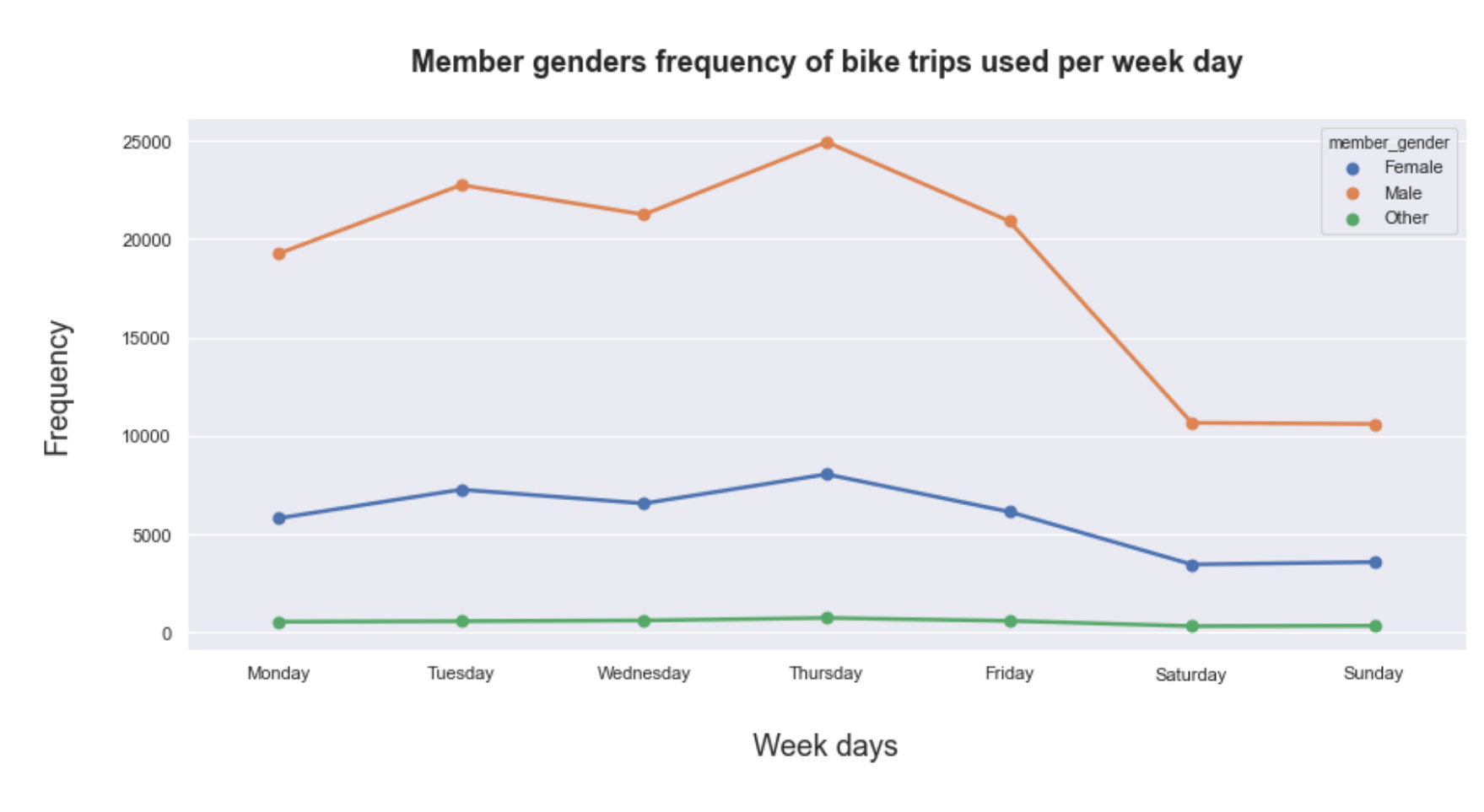
Member genders frequency of bike trips used per week day

```python
In [109]: frequency2 = df.groupby(['week_day', 'member_gender']).size().reset_index()
frequency2.columns = frequency2.columns[:2].append(pd.Index(['frequency']))

sns.pointplot(data=frequency2, x='week_day', y='frequency', hue = 'member_gender', scale=.9)

plt.title('\nMember genders frequency of bike trips used per week day',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')
plt.xlabel('Week days',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Frequency',
           fontsize = 18,
           labelpad=30)
```

Out[109]: Text(0, 0.5, 'Frequency')



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?

- The importance of including user type in data analysis revealed that there are some differences in the behavior of customers and subscribers
- The average bike trip duration in minutes per user type was (10.679922) are subscriber, while customer are (21.847536).
- The distribution of bike trip duration in minutes per user type was the distribution of customer trip time shows more variability than subscribers.
- The number of bike trips gender per user type was a subscribers bigger than customers.
- The distribution of bike trips used per day hour was a subscribers bigger than customers.
- The number of bike trips for subscribers is higher on work days (Monday, Tuesday, Wednesday, Thursday, Friday) and lower on weekends days (Saturday, Sunday) in San Francisco.
- The number of bike trips for customers is consistent on all week days in San Francisco.

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

- I've observed that the number of bike trips for subscribers is less on weekends, as opposed to customers which stays consistent on all week days.

Multivariate Exploration

The relation between the hour of day, bike trip duration in seconds, and member gender

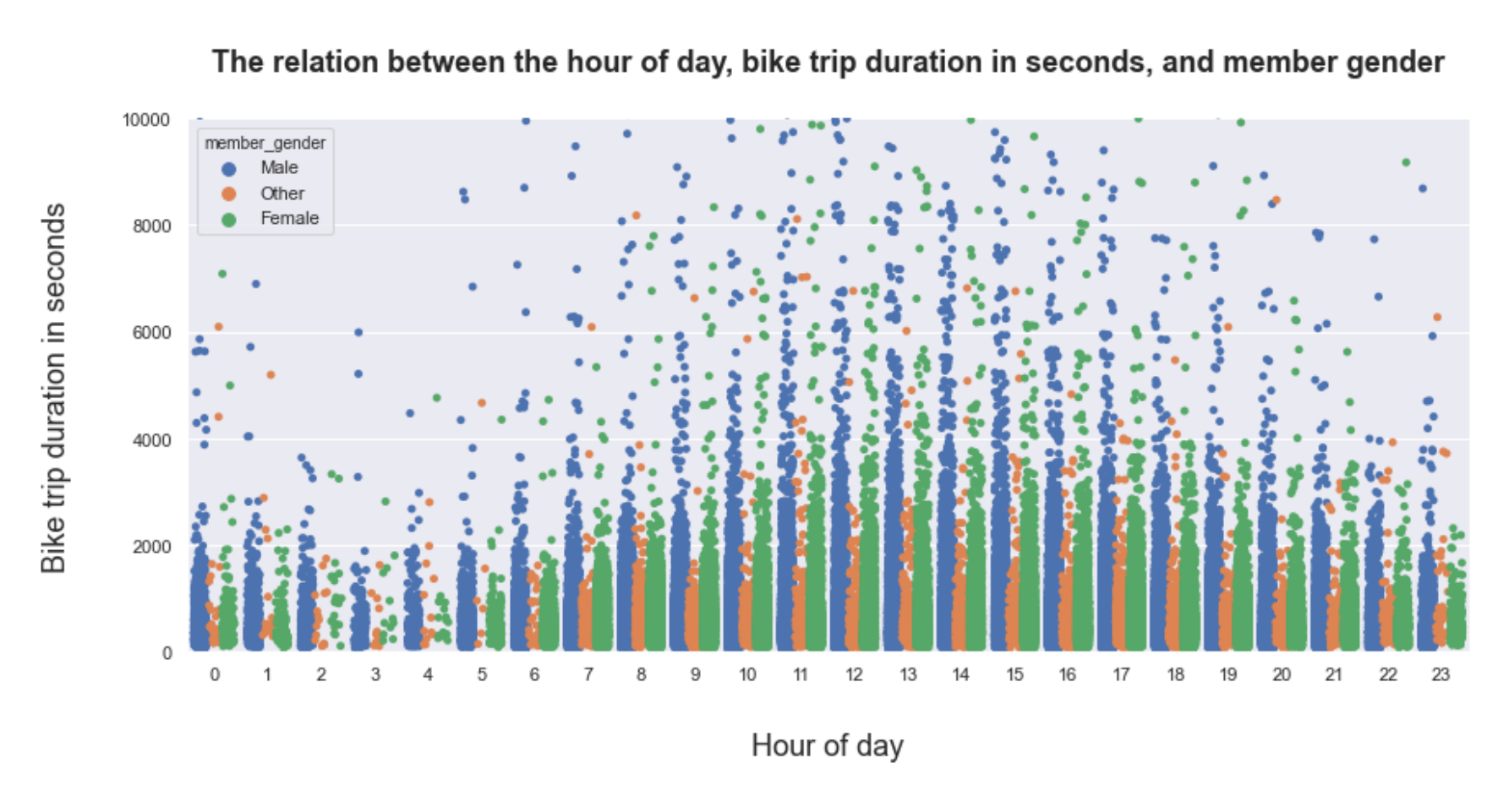
```python
In [121]: sns.stripplot(data = df, x = 'start_hour' , y = 'duration_sec', hue = 'member_gender', jitter = 0.35, d
odge = True)

plt.ylim(0,10000)

plt.title('\nThe relation between the hour of day, bike trip duration in seconds, and member gender',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')

plt.xlabel('Hour of day',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Bike trip duration in seconds',
           fontsize = 18,
           labelpad=30)
```

Out[121]: Text(0, 0.5, 'Bike trip duration in seconds')



The relation between the hour of day, bike trip duration in seconds, and user type

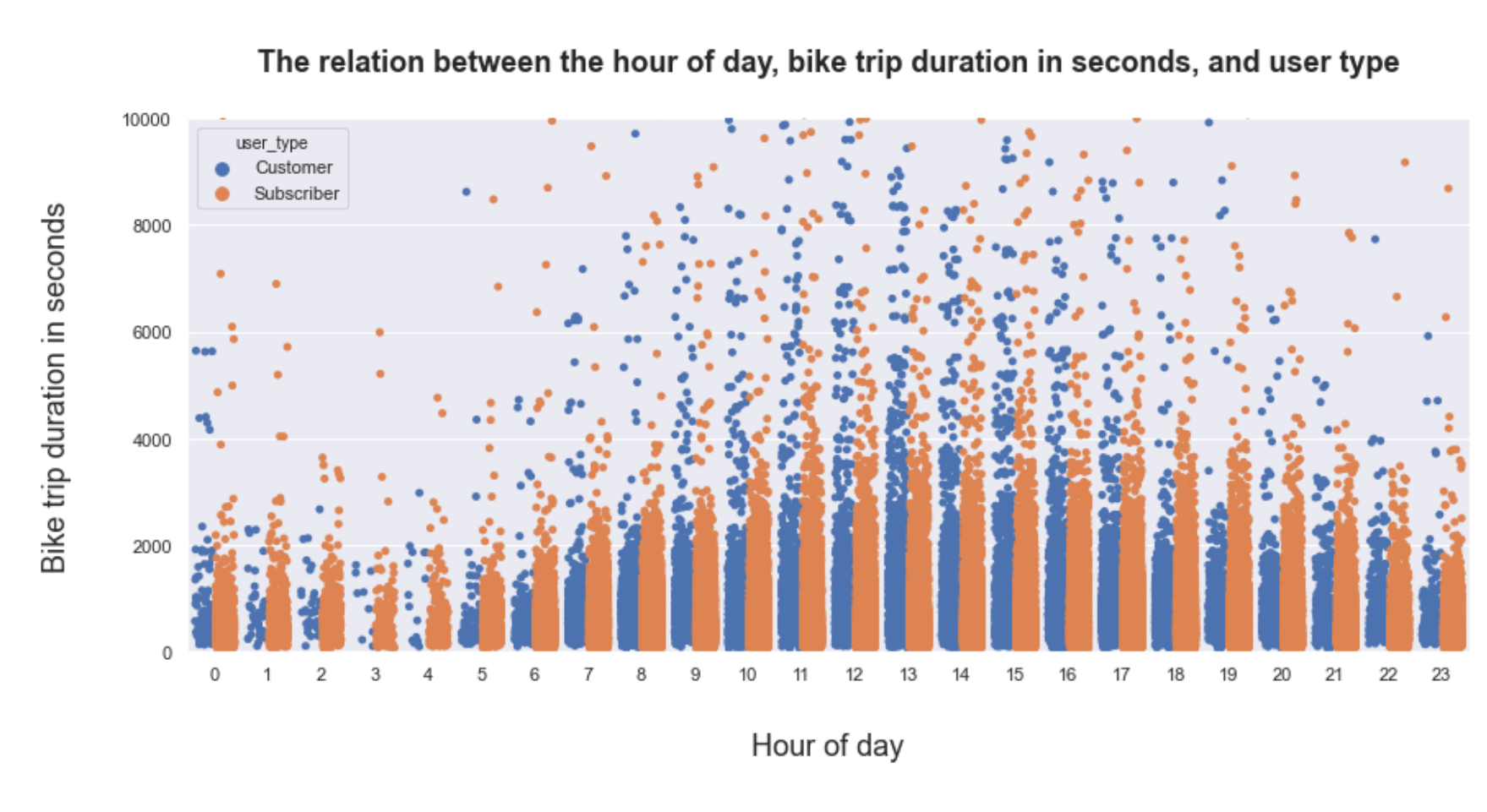
```python
In [122]: sns.stripplot(data = df, x = 'start_hour' , y = 'duration_sec', hue = 'user_type', jitter = 0.35, dodge
 = True)

plt.ylim(0,10000)

plt.title('\nThe relation between the hour of day, bike trip duration in seconds, and user type',
          y = 1.07,
          fontsize = 18,
          fontweight = 'bold')

plt.xlabel('Hour of day',
           fontsize = 18,
           labelpad=30)
plt.ylabel('Bike trip duration in seconds',
           fontsize = 18,
           labelpad=30)
```

Out[122]: Text(0, 0.5, 'Bike trip duration in seconds')



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?

- I've observed that male customers take longer bike trips than female customers.
- I've observed that males take long bike trips from 11 AM to 4 PM.
- I've observed that subscribers take long bike trips from 11 AM to 5 PM.

Were there any interesting or surprising interactions between features?

- The interaction between the hour of the day, bike trip duration in seconds, and user type were surprising.
- The time between 11 AM to 5 PM is significantly different.

```python
In [ ]:
```