DATA Analyzing for Ford GoBike

by. hamed bintalib

Ford GoBike is the Bay Area's bike share system. Bay Area Bike Share was introduced in 2013 as a pilot program for the region, with 700 bikes and 70 stations across San Francisco and San Jose. By the end of 2018, Ford GoBike will grow to 7,000 bikes across San Francisco, the East Bay and San Jose.

In [2]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
%matplotlib inline
```

In [3]:

```
df= pd.read_csv('201904-fordgobike-tripdata.csv')
```

In [4]:

```
df.head()
```

Out[4]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_lat
0	50305	2019-04-30 22:33:55.1550	2019-05-01 12:32:20.4540	368.0	Myrtle St at Polk St	37.78
1	53725	2019-04-30 20:43:41.6320	2019-05-01 11:39:06.9170	246.0	Berkeley Civic Center	37.86
2	78072	2019-04-30 10:32:46.4890	2019-05-01 08:13:58.9750	64.0	5th St at Brannan St	37.77
3	78969	2019-04-30 10:00:51.5500	2019-05-01 07:57:01.2620	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.77
4	1128	2019-04-30 23:59:04.7390	2019-05-01 00:17:53.0910	124.0	19th St at Florida St	37.76

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
duration_sec                0
start_time                  0
end_time                    0
start_station_id           64
start_station_name         64
start_station_latitude      0
start_station_longitude     0
end_station_id             64
end_station_name           64
end_station_latitude        0
end_station_longitude       0
bike_id                     0
user_type                   0
member_birth_year       11199
member_gender           11199
bike_share_for_all_trip     0
dtype: int64
```

In [6]:

```
df[df['start_station_id'].isnull()]
```

Out[6]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start
10983	1131	2019-04-29 15:30:48.3890	2019-04-29 15:49:39.4800	NaN	NaN	37.41	
11568	268	2019-04-29 13:12:08.9280	2019-04-29 13:16:37.8740	NaN	NaN	37.41	
14814	669	2019-04-28 23:02:16.2860	2019-04-28 23:13:25.6670	NaN	NaN	37.40	
16067	94	2019-04-28 16:30:12.3680	2019-04-28 16:31:46.8660	NaN	NaN	37.40	
17179	2389	2019-04-28 12:24:07.6830	2019-04-28 13:03:57.4710	NaN	NaN	37.40	
17185	185	2019-04-28 12:59:44.6410	2019-04-28 13:02:49.7770	NaN	NaN	37.42	
18116	719	2019-04-28	2019-04-28	NaN	NaN	37.41	

In [7]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 239111 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               239111 non-null int64
start_time                 239111 non-null object
end_time                   239111 non-null object
start_station_id           239047 non-null float64
start_station_name         239047 non-null object
start_station_latitude     239111 non-null float64
start_station_longitude    239111 non-null float64
end_station_id             239047 non-null float64
end_station_name           239047 non-null object
end_station_latitude       239111 non-null float64
end_station_longitude      239111 non-null float64
bike_id                    239111 non-null int64
user_type                  239111 non-null object
member_birth_year          227912 non-null float64
member_gender              227912 non-null object
bike_share_for_all_trip    239111 non-null object
```

In [8]:

```
1 df.describe()
```

Out[8]:

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id
count	239111.000000	239047.000000	239111.000000	239111.000000	239047.000000
mean	802.671199	141.836538	37.769536	-122.352503	140.838095
std	1990.006091	116.289776	0.127698	0.277088	116.386168
min	61.000000	3.000000	0.000000	-122.453704	3.000000
25%	349.000000	47.000000	37.770083	-122.413004	44.000000
50%	558.000000	104.000000	37.780760	-122.398285	102.000000
75%	876.000000	240.000000	37.797280	-122.291209	239.000000
max	86114.000000	420.000000	37.880222	0.000000	420.000000

In [9]:

```
1 #removing missing values in gender
2 df_clean=df[df['member_gender'].isnull()==False]
```

In [10]:

```
1 df_clean.head()
```

Out[10]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_lat
0	50305	2019-04-30 22:33:55.1550	2019-05-01 12:32:20.4540	368.0	Myrtle St at Polk St	37.78
2	78072	2019-04-30 10:32:46.4890	2019-05-01 08:13:58.9750	64.0	5th St at Brannan St	37.77
3	78969	2019-04-30 10:00:51.5500	2019-05-01 07:57:01.2620	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.77
4	1128	2019-04-30 23:59:04.7390	2019-05-01 00:17:53.0910	124.0	19th St at Florida St	37.76
5	1388	2019-04-30 23:53:05.9820	2019-05-01 00:16:14.3130	243.0	Bancroft Way at College Ave	37.86

In [11]:

```
1 df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227912 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               227912 non-null int64
start_time                 227912 non-null object
end_time                   227912 non-null object
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227912 non-null float64
start_station_longitude    227912 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227912 non-null float64
end_station_longitude      227912 non-null float64
bike_id                    227912 non-null int64
user_type                  227912 non-null object
member_birth_year          227912 non-null float64
member_gender              227912 non-null object
bike_share_for_all_trip    227912 non-null object
dtypes: float64(7), int64(2), object(7)
memory usage: 29.6+ MB
```

In [12]:

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 239111 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               239111 non-null int64
start_time                 239111 non-null object
end_time                   239111 non-null object
start_station_id           239047 non-null float64
start_station_name         239047 non-null object
start_station_latitude     239111 non-null float64
start_station_longitude    239111 non-null float64
end_station_id             239047 non-null float64
end_station_name           239047 non-null object
end_station_latitude       239111 non-null float64
end_station_longitude      239111 non-null float64
bike_id                    239111 non-null int64
user_type                  239111 non-null object
member_birth_year          227912 non-null float64
member_gender              227912 non-null object
bike_share_for_all_trip    239111 non-null object
dtypes: float64(7), int64(2), object(7)
memory usage: 29.2+ MB
```

In [13]:

```
1 df_clean['member_gender'].value_counts()
```

Out[13]:

```
Male      168140
Female     55498
Other       4274
Name: member_gender, dtype: int64
```

In [14]:

```
1 df_clean['user_type'].value_counts()
```

Out[14]:

```
Subscriber    198510
Customer       29402
Name: user_type, dtype: int64
```

In [15]:

```
1 df_clean['start_station_id'].isnull().sum()
```

Out[15]:

```
64
```

In [16]:

```
# Removing missing values
df_clean=df_clean[df_clean['start_station_id'].isnull()==False]
```

In [17]:

```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null object
end_time                   227848 non-null object
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
dtypes: float64(7), int64(2), object(7)
memory usage: 29.6+ MB
```

In [18]:

```
1 df_clean.isnull().sum()
```

Out[18]:

```
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
```

In [19]:

```
1 df_clean['start_station_id'].isnull().sum()
```

Out[19]:

```
0
```

In [20]:

```
1 df_clean.duplicated().sum()
```

Out[20]:

```
0
```

In [21]:

```
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null object
end_time                   227848 non-null object
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
dtypes: float64(7), int64(2), object(7)
memory usage: 29.6+ MB
```

In [22]:

```
# Changing start_time and end_time to datetime format
df_clean.start_time = pd.to_datetime(df_clean.start_time)
df_clean.end_time = pd.to_datetime(df_clean.end_time)
```

In [23]:

```
df_clean.head()
```

Out[23]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

In [24]:

```
#Extract dayofweek, hours information from the start_time
df_clean['start_time_dayofweek']= df_clean['start_time'].dt.strftime('%a')
df_clean['start_time_hour']= df_clean['start_time'].dt.hour
```

In [25]:

```
df_clean['start_time_dayofweek'].value_counts()

```

Out[25]:

```
Tue    41856
Mon    39296
Thu    37234
Wed    37222
Fri    33733
Sat    21475
Sun    17032
Name: start_time_dayofweek, dtype: int64
```

In [26]:

```
1 df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 18 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
start_time_dayofweek       227848 non-null object
start_time_hour            227848 non-null int64
dtypes: datetime64[ns](2), float64(7), int64(3), object(6)
memory usage: 33.0+ MB
```

In [27]:

```
1 df_clean.head()
```

Out[27]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

In [28]:

```
df_clean['start_time_dayofweek'].value_counts()

```

Out[28]:

```
Tue    41856
Mon    39296
Thu    37234
Wed    37222
Fri    33733
Sat    21475
Sun    17032
Name: start_time_dayofweek, dtype: int64
```

In [29]:

```
# 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 = we
df_clean['start_time_dayofweek'] = df_clean['start_time_dayofweek'].astype(orde
```

In [30]:

```
df_clean.info()

```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 18 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
start_time_dayofweek       227848 non-null category
start_time_hour            227848 non-null int64
dtypes: category(1), datetime64[ns](2), float64(7), int64(3), object(5
)
memory usage: 31.5+ MB
```

In [31]:

```
# identify the age of the customer from member_birth_year column.
df_clean['member_age'] = 2019 - df_clean['member_birth_year']
```

In [32]:

```
df_clean['member_age'] = df_clean['member_age'].astype(int)
df_clean['member_birth_year'] = df_clean['member_birth_year'].astype(int)
```

In [33]:

```
df_clean.info()

```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 19 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null int64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
start_time_dayofweek       227848 non-null category
start_time_hour            227848 non-null int64
member_age                 227848 non-null int64
dtypes: category(1), datetime64[ns](2), float64(6), int64(5), object(5
)
memory usage: 33.2+ MB
```

In [34]:

```
1 df_clean.head()
2
```

Out[34]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

What is the structure of your dataset?

This data set includes information about individual rides made in a bike-sharing system covering the greater San Francisco that happened in 2019:

```
* Trip Duration (seconds)
* Start Time and Date
* End Time and Date
* 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 (Subsc
riber or Customer — “Subscriber” =   Memberor “Customer” = Casual)
```

Member Year of Birth Member Gender

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

- I'll be investigating in udration of biking time
- I'll extract dayofweek, hours for further investigation in start_time
- member_gender
- user_type

In [35]:

```
#. converting duration to minutes instead of second
df_clean['duration_min'] = df_clean['duration_sec']/60
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 20 columns):
duration_sec                227848 non-null int64
start_time                  227848 non-null datetime64[ns]
end_time                    227848 non-null datetime64[ns]
start_station_id            227848 non-null float64
start_station_name          227848 non-null object
start_station_latitude      227848 non-null float64
start_station_longitude     227848 non-null float64
end_station_id              227848 non-null float64
end_station_name            227848 non-null object
end_station_latitude        227848 non-null float64
end_station_longitude       227848 non-null float64
bike_id                     227848 non-null int64
user_type                   227848 non-null object
member_birth_year           227848 non-null int64
member_gender               227848 non-null object
bike_share_for_all_trip     227848 non-null object
start_time_dayofweek        227848 non-null category
start_time_hour             227848 non-null int64
member_age                  227848 non-null int64
duration_min                227848 non-null float64
dtypes: category(1), datetime64[ns](2), float64(7), int64(5), object(5
)
memory usage: 35.0+ MB
```

In [36]:

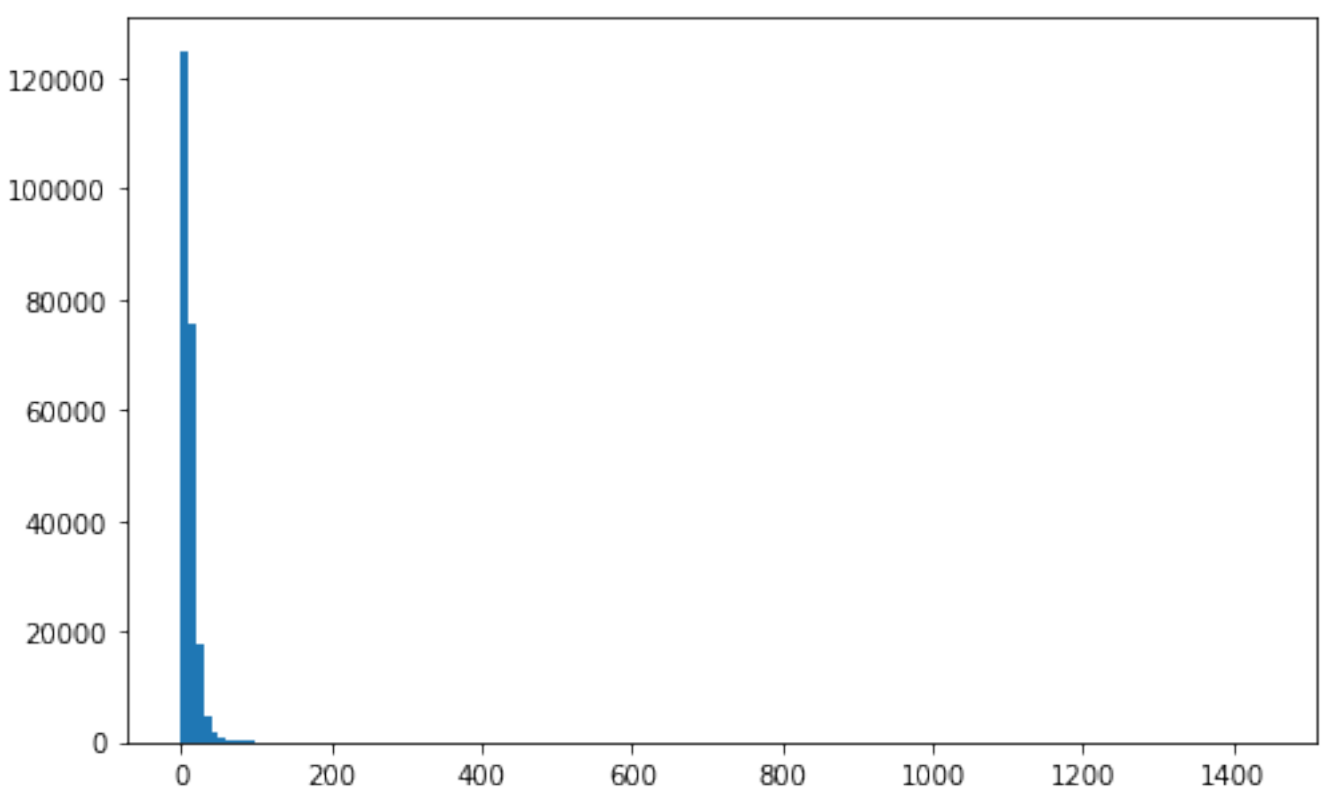
```
df_clean.head()
```

Out[36]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

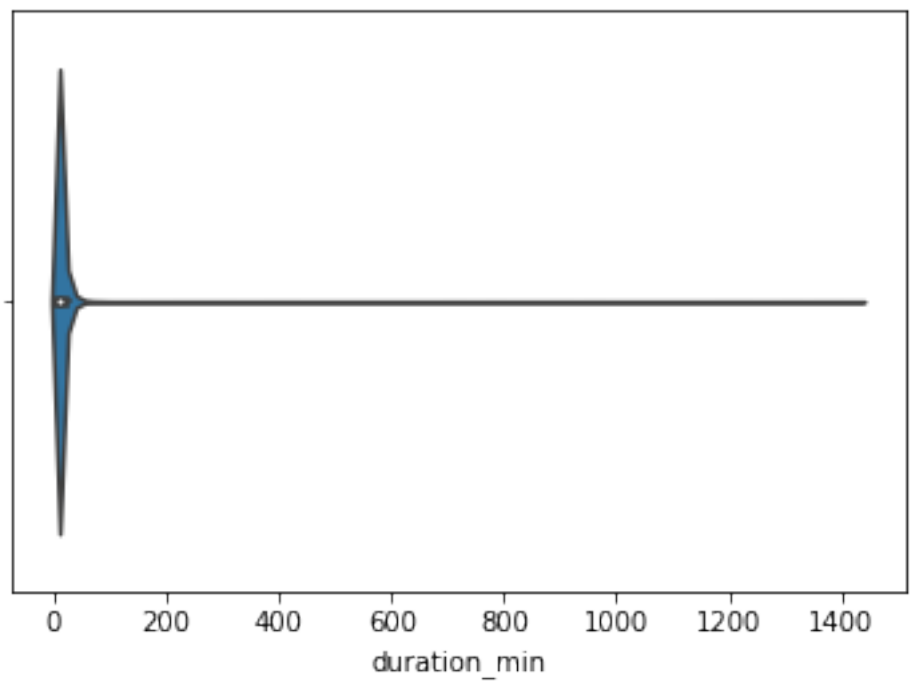
In [37]:

```
binsize = 10
bins = np.arange(0, df_clean['duration_min'].max()+binsize, binsize)

plt.figure(figsize=[8, 5])

plt.hist(data = df_clean, x = 'duration_min', bins=bins);
```



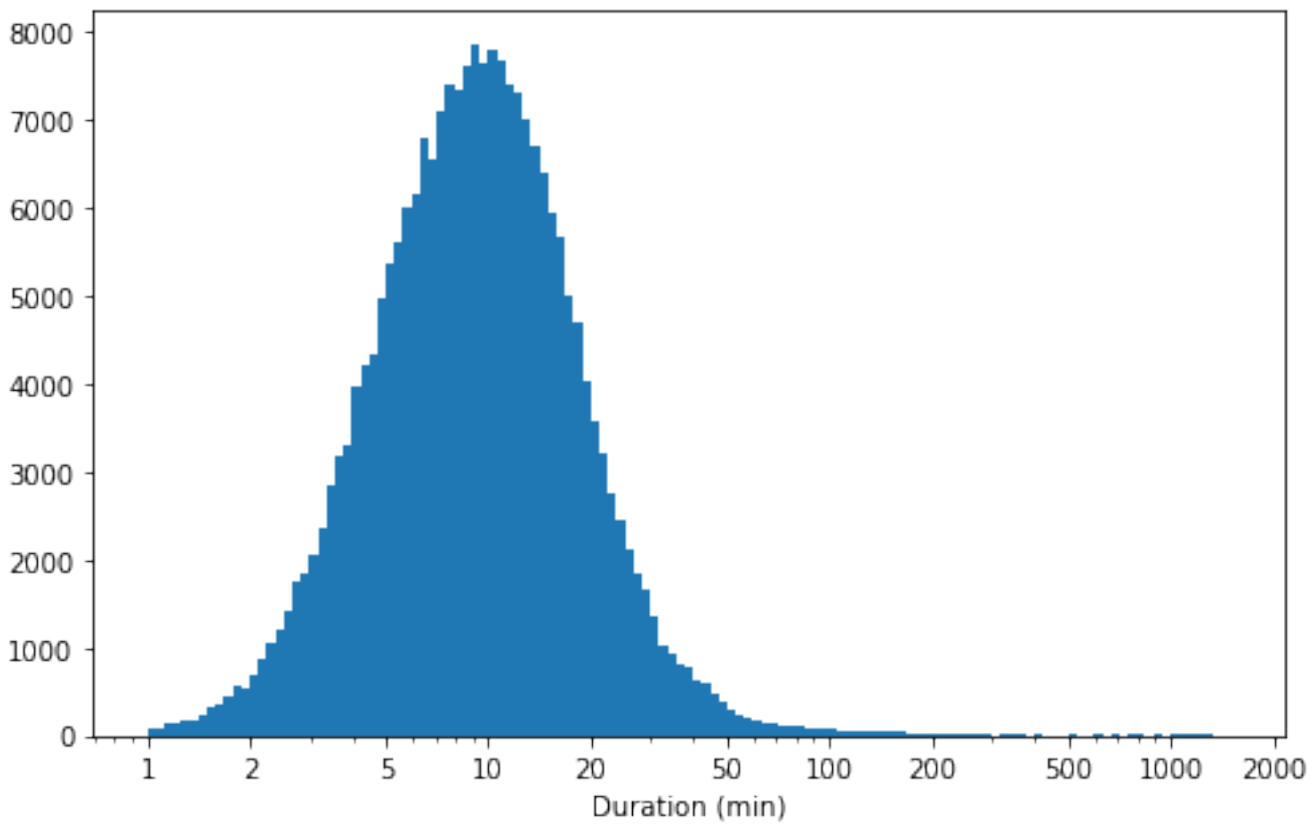
In [38]:

```
sb.violinplot(data = df_clean, x = 'duration_min');
```



In [39]:

```

# there's a long tail in the distribution, so let's put it on a log scale inste
log_binsize = 0.025
bins = 10 ** np.arange(0, np.log10(df_clean['duration_min'].max())+log_binsize,

plt.figure(figsize=[8, 5]);
plt.hist(data = df_clean, x = 'duration_min', bins = bins);
plt.xscale('log');
plt.xticks([1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, 2000], [1, 2, 5, 10, 20, 5
plt.xlabel('Duration (min)');
```



In [40]:

```
# Leave record with duration_min < 100 min
df_clean = df_clean[df_clean['duration_min'] <= 100]
df_clean.info()

```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 226804 entries, 4 to 239110
Data columns (total 20 columns):
duration_sec                226804 non-null int64
start_time                  226804 non-null datetime64[ns]
end_time                    226804 non-null datetime64[ns]
start_station_id            226804 non-null float64
start_station_name          226804 non-null object
start_station_latitude      226804 non-null float64
start_station_longitude     226804 non-null float64
end_station_id              226804 non-null float64
end_station_name            226804 non-null object
end_station_latitude        226804 non-null float64
end_station_longitude       226804 non-null float64
bike_id                     226804 non-null int64
user_type                   226804 non-null object
member_birth_year           226804 non-null int64
member_gender               226804 non-null object
bike_share_for_all_trip     226804 non-null object
start_time_dayofweek        226804 non-null category
start_time_hour             226804 non-null int64
member_age                  226804 non-null int64
duration_min                226804 non-null float64
dtypes: category(1), datetime64[ns](2), float64(7), int64(5), object(5
)
memory usage: 34.8+ MB
```

In [41]:

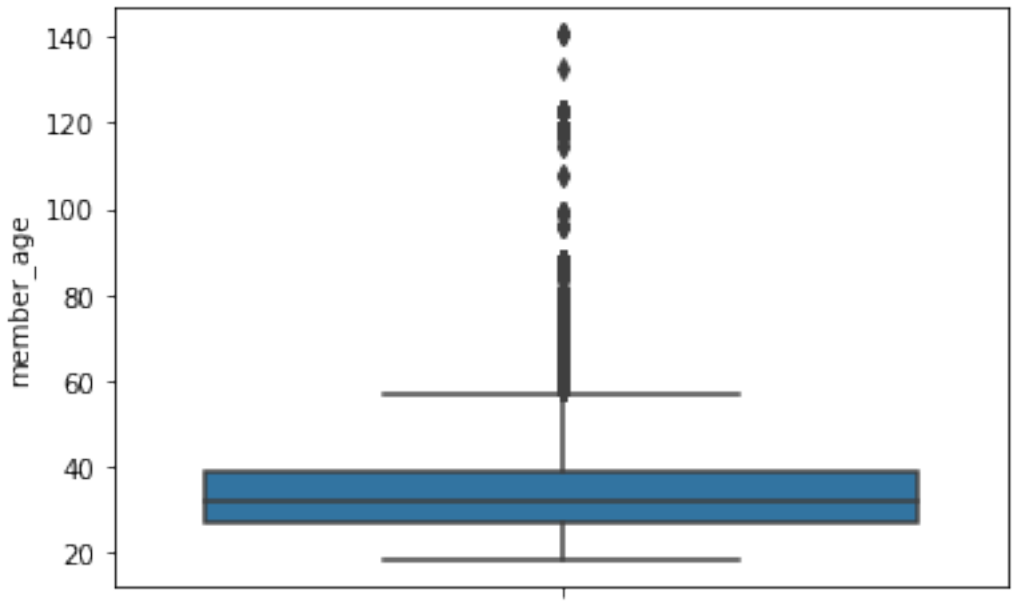
```
df_clean['member_age'].describe()

```

Out[41]:

```
count    226804.000000
mean         34.137008
std           9.954496
min          18.000000
25%          27.000000
50%          32.000000
75%          39.000000
max         141.000000
Name: member_age, dtype: float64
```

In [42]:

```
sb.boxplot(data = df_clean, y = 'member_age');

```



In [43]:

```
# removing outliners.
df_clean = df_clean[df_clean['member_age'] <= 60]

```

In [44]:

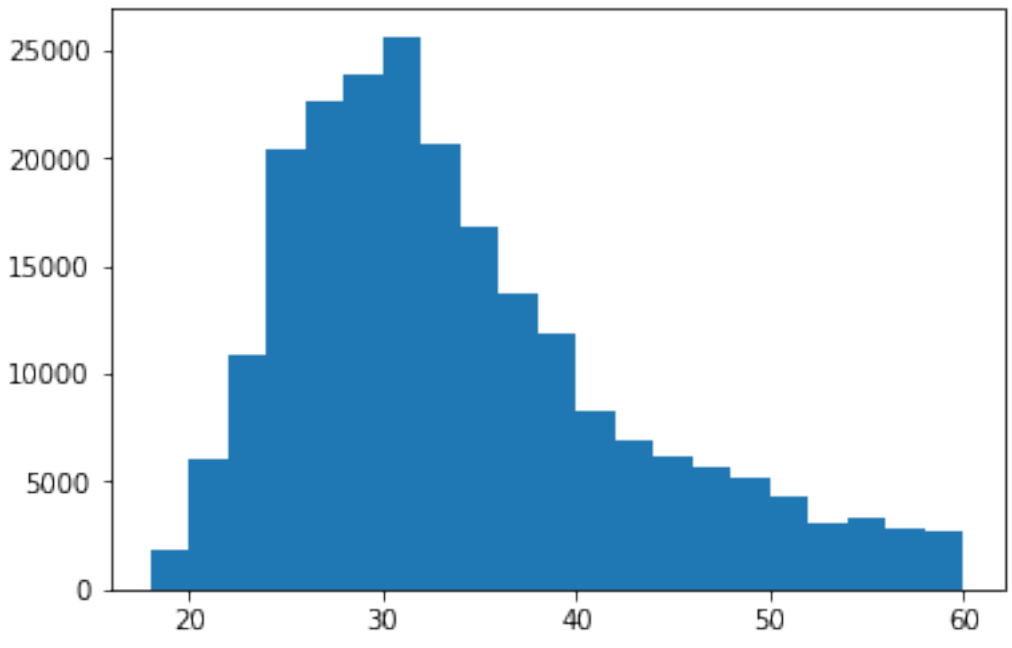
```
df_clean.head()

```

Out[44]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

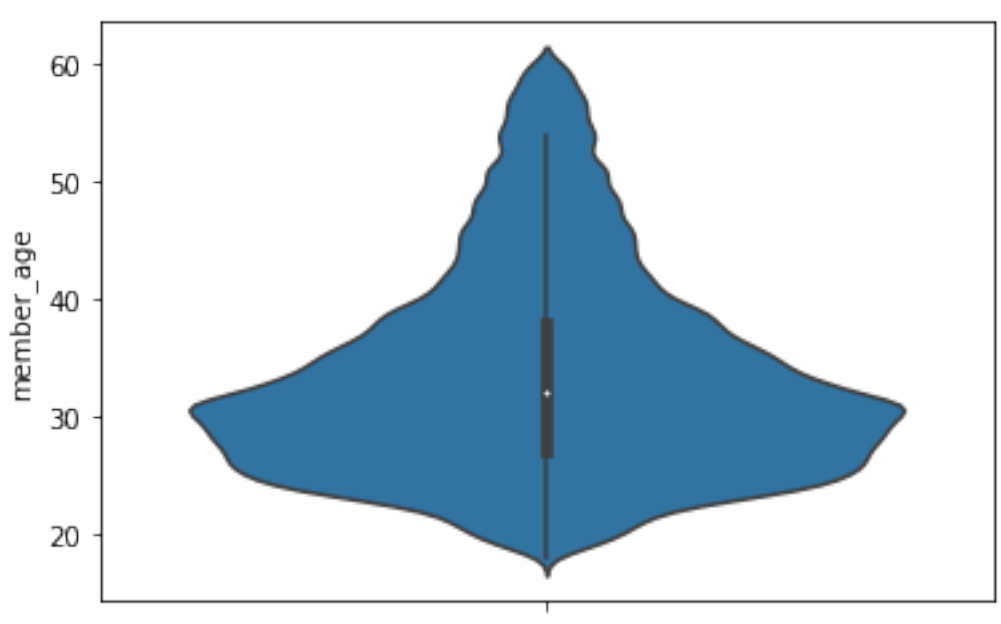
In [45]:

```
binsize = 2
bins = np.arange(18, df_clean['member_age'].max()+binsize, binsize)
plt.hist(data = df_clean, x = 'member_age', bins = bins);
```



In [46]:

```
sb.violinplot(data = df_clean, y = 'member_age');

```



In [47]:

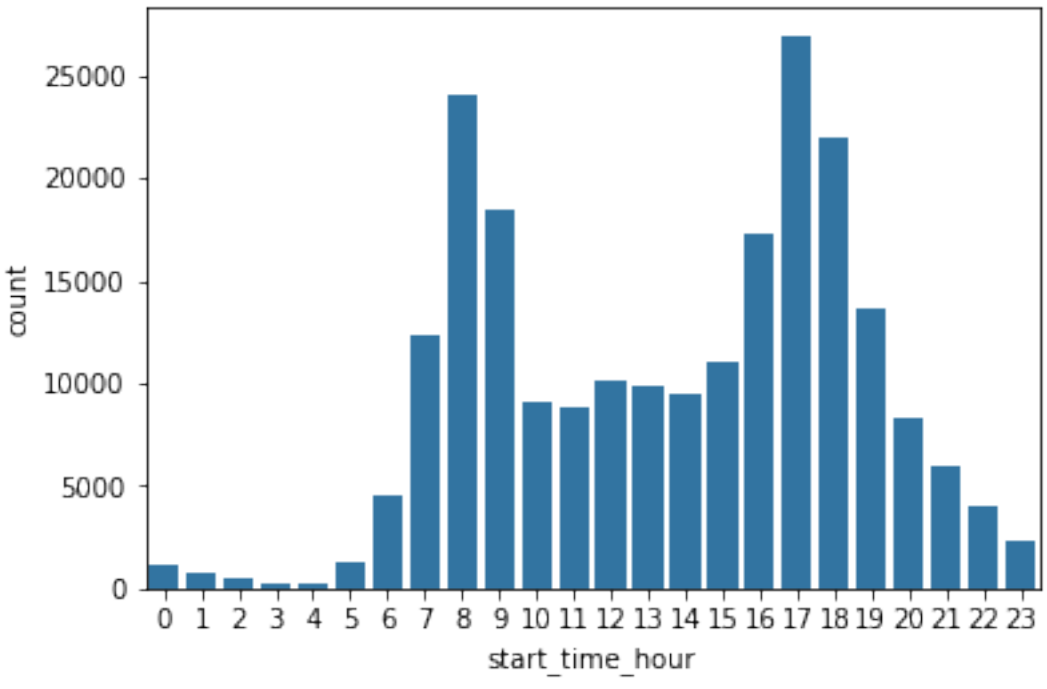
```
# save cleaned data
df_clean.to_csv('clean_master_file.csv', index=False)
```

In [48]:

```
default_color = sb.color_palette()[0]

```

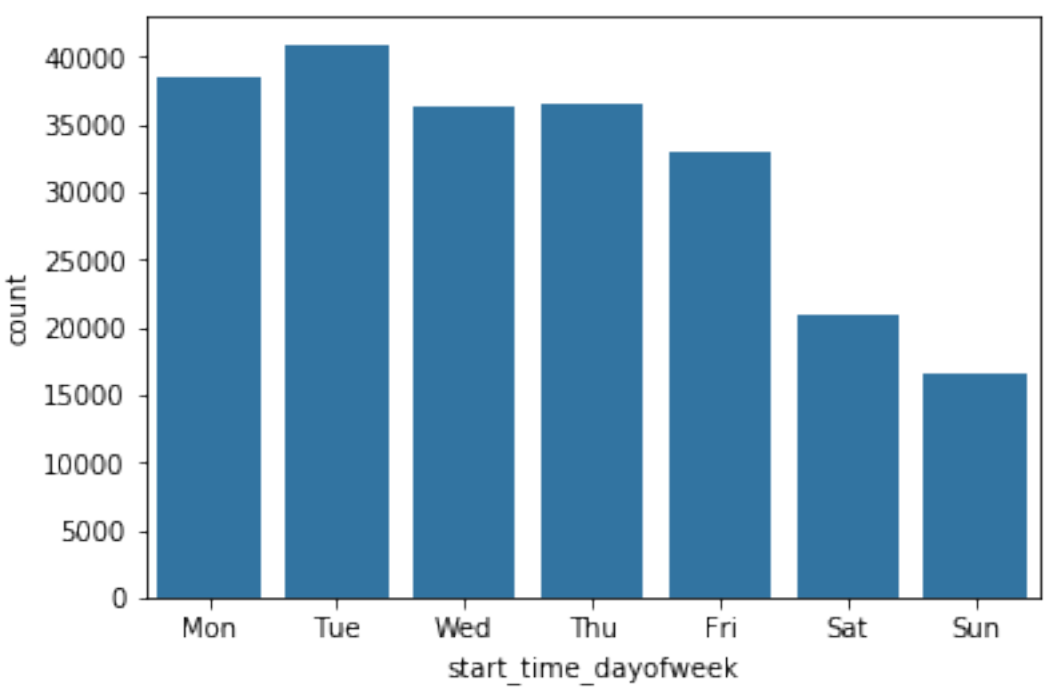
In [49]:

```
sb.countplot(data = df_clean, x = 'start_time_hour', color = default_color);

```



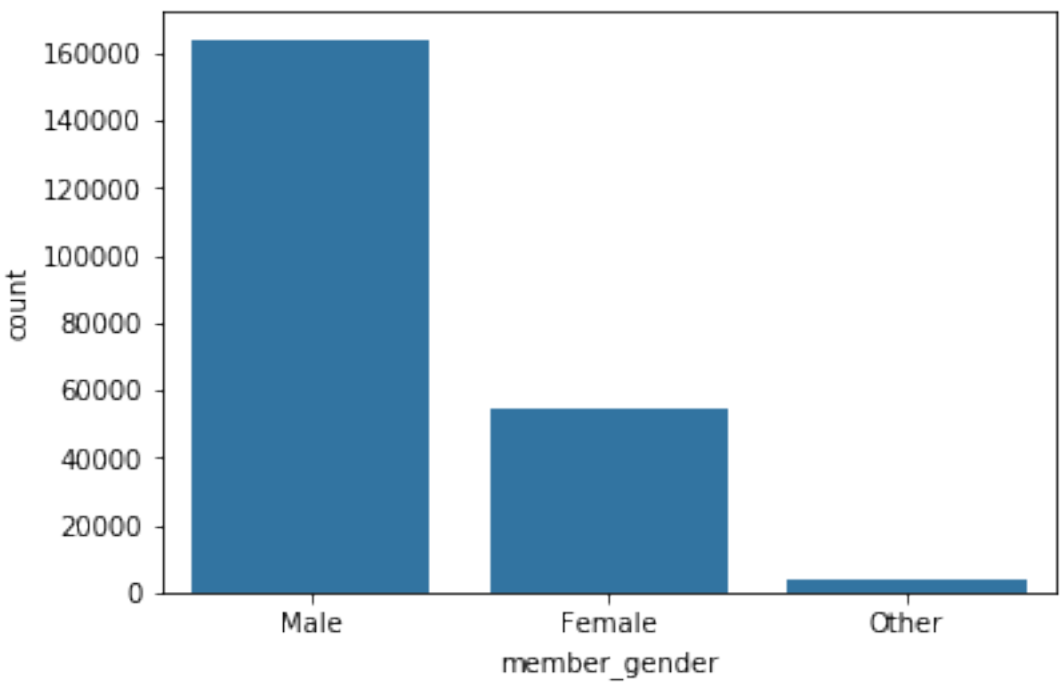
In [50]:

```
sb.countplot(data = df_clean, x = 'start_time_dayofweek', color = default_color

```



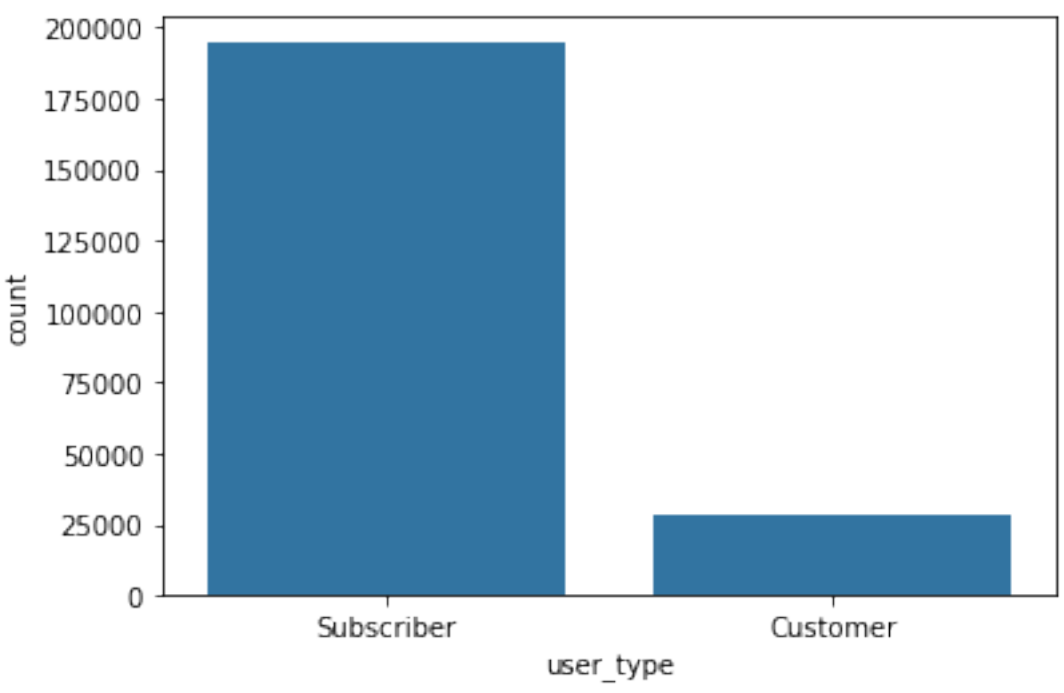
In [51]:

```
sb.countplot(data = df_clean, x = 'member_gender', color = default_color);

```



In [52]:

```
sb.countplot(data = df_clean, x = 'user_type', color = default_color);

```

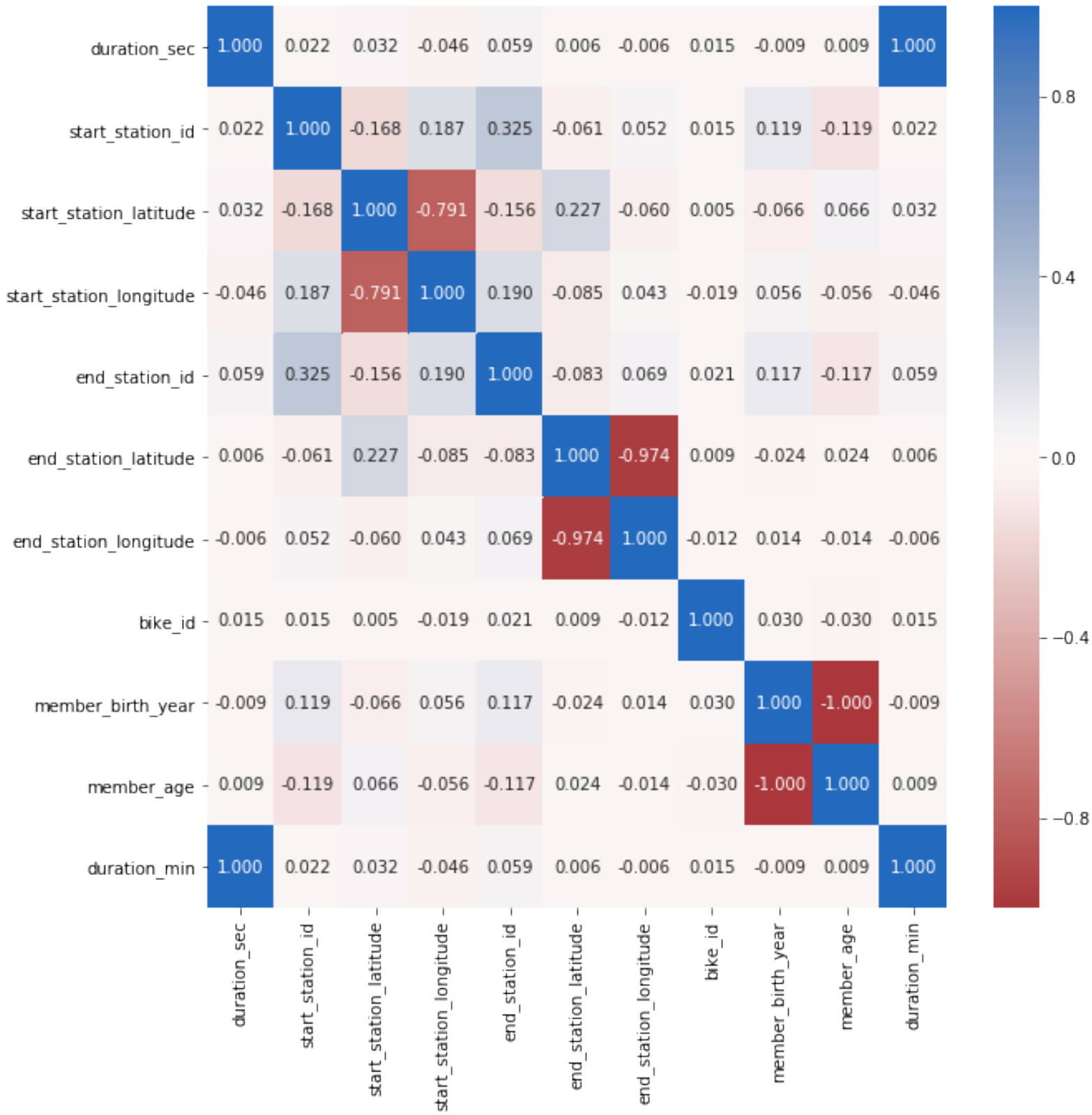


In [53]:

```
all_numeric_vars = ['duration_sec', 'start_time', 'end_time', 'start_station_id
'end_station_longitude', 'bike_id', 'member_birth_year', 'member_age', 'duratio
numeric_vars = ['duration_min', 'member_age']
categoric_vars = ['start_time_dayofweek', 'start_time_hour', 'member_gender', '
```

In [54]:

```
#correlation plot
plt.figure(figsize = [10, 10])
sb.heatmap(df_clean[all_numeric_vars].corr(), annot = True, fmt = '.3f', cmap =
```



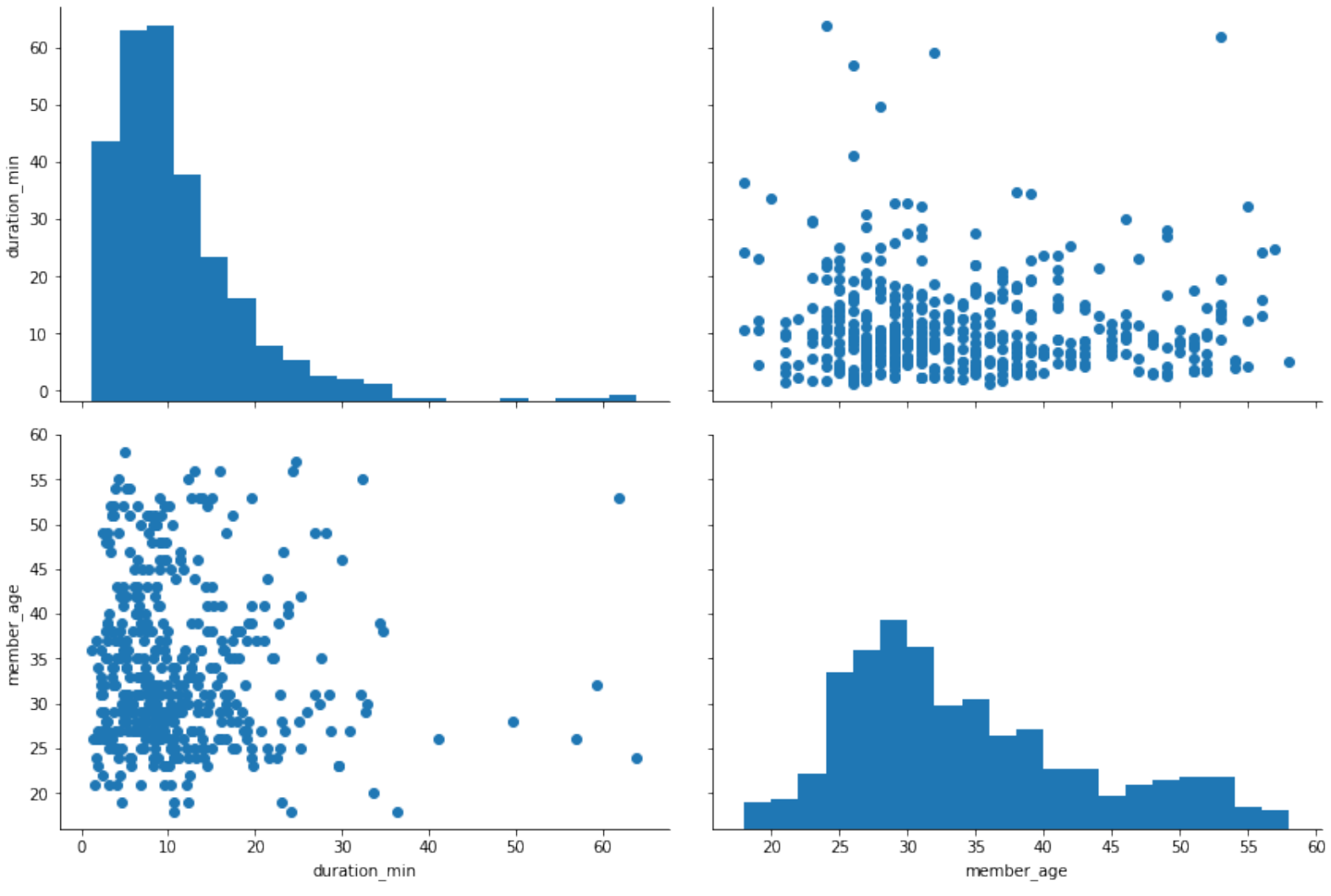
From the above plot we can see that there are few variables which are highly correlated to some other variables either having positive correlation or negative correlation.

In [55]:

```
# plot matrix: sample 500 records so that plots are clearer and
# they render faster

samples = np.random.choice(df_clean.shape[0], 500, replace = False)
samp = df_clean.loc[samples,:]
```

```
6
7  g = sb.PairGrid(data = samp, vars = numeric_vars, height = 4, aspect = 1.5)
8  g = g.map_diag(plt.hist, bins = 20);
9  g.map_offdiag(plt.scatter);
```

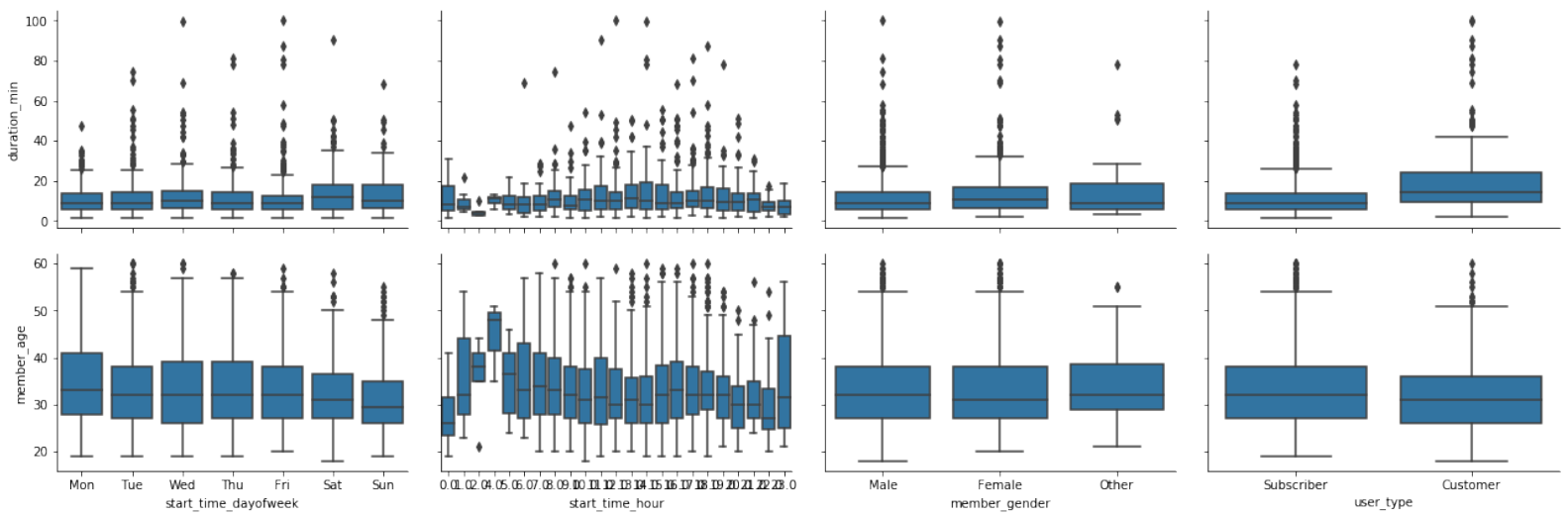
```
/Users/hamedbintalib/anaconda3/lib/python3.7/site-packages/pandas/core
/indexing.py:1494: FutureWarning:
Passing list-likes to .loc or [] with any missing label will raise
KeyError in the future, you can use .reindex() as an alternative.

See the documentation here:
https://pandas.pydata.org/pandas-docs/stable/indexing.html#deprecate-l
oc-reindex-listlike (https://pandas.pydata.org/pandas-docs/stable/inde
xing.html#deprecate-loc-reindex-listlike)
  return self._getitem_tuple(key)
/Users/hamedbintalib/anaconda3/lib/python3.7/site-packages/numpy/lib/h
istograms.py:824: RuntimeWarning: invalid value encountered in greater
_equal
  keep = (tmp_a >= first_edge)
/Users/hamedbintalib/anaconda3/lib/python3.7/site-packages/numpy/lib/h
istograms.py:825: RuntimeWarning: invalid value encountered in less_eq
ual
  keep &= (tmp_a <= last_edge)
```



In [56]:

```
# plot matrix of numeric features against categorical features.
# can use a larger sample since there are fewer plots and they're simpler in na

samples = np.random.choice(df_clean.shape[0], 2000, replace = False);
samp = df_clean.loc[samples, :];

def boxgrid(x, y, **kwargs):
    """ Quick hack for creating box plots with seaborn's PairGrid. """
    default_color = sb.color_palette()[0];
    sb.boxplot(x, y, color = default_color);

plt.figure(figsize = [15, 15]);
g = sb.PairGrid(data = samp, y_vars = ['duration_min', 'member_age'], x_vars = c
g.map(boxgrid);
```

```
<Figure size 1080x1080 with 0 Axes>
```



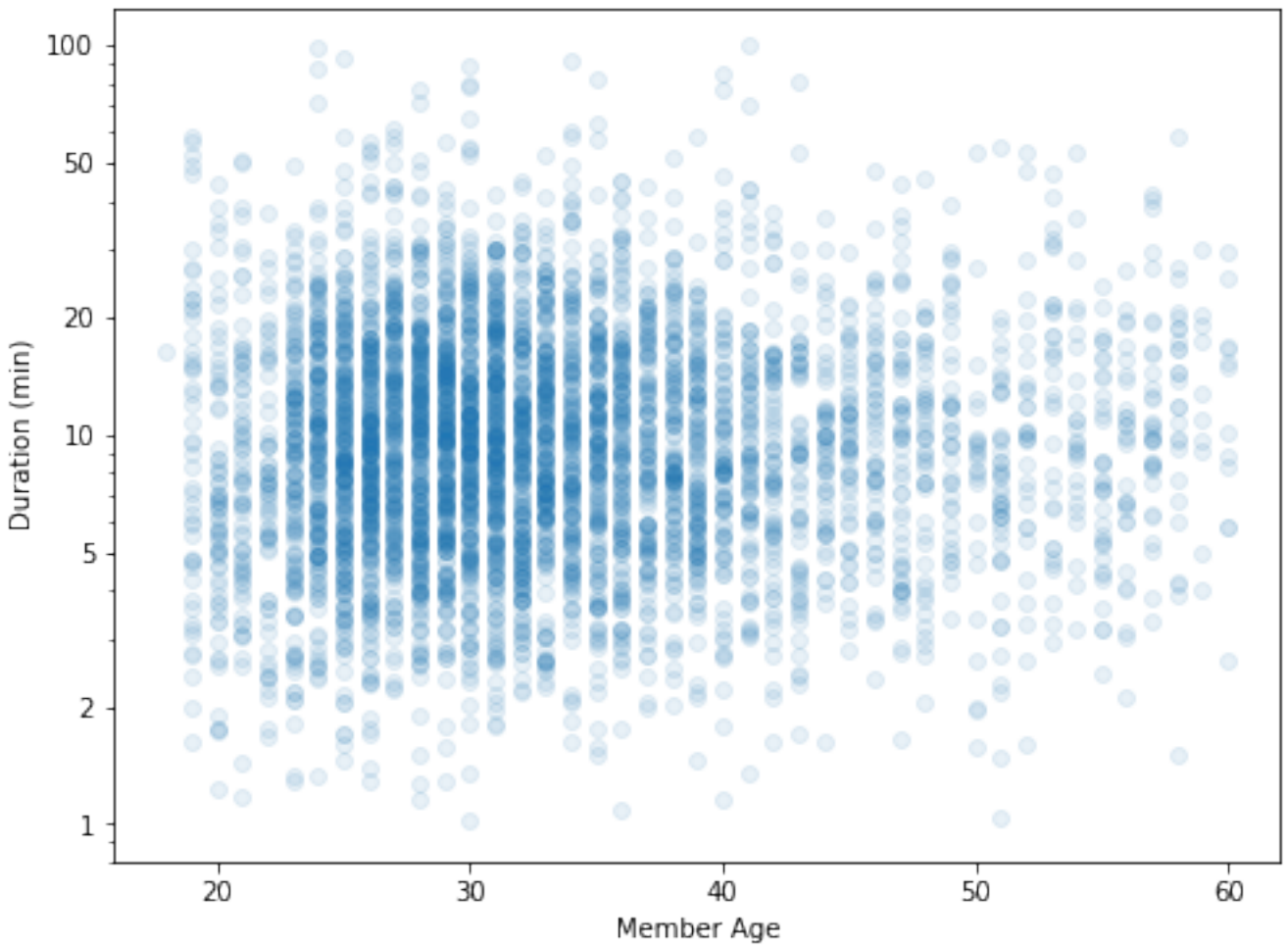
In [57]:

```
plt.figure(figsize = [20, 20]);


plt.subplot(6, 1, 1);
sb.countplot(data = df_clean, x = 'start_time_hour', hue = 'start_time_dayofweel


ax = plt.subplot(6, 1, 2);
sb.countplot(data = df_clean, x = 'start_time_hour', hue = 'member_gender', pale
ax.legend(ncol = 2); # re-arrange legend to reduce overlapping


ax = plt.subplot(6, 1, 3);
sb.countplot(data = df_clean, x = 'start_time_hour', hue = 'user_type', palette
ax.legend(loc = 1, ncol = 2); # re-arrange legend to remove overlapping

ax = plt.subplot(6, 1, 4);
sb.countplot(data = df_clean, x = 'start_time_dayofweek', hue = 'member_gender',

ax = plt.subplot(6, 1, 5);
```

```
sb.countplot(data = df_clean, x = 'start_time_dayofweek', hue = 'user_type', pa
ax = plt.subplot(6, 1, 6);
sb.countplot(data = df_clean, x = 'member_gender', hue = 'user_type', palette =
```

In [58]:

```
# scatter plot of duration_min vs. member_age, with log transform on duration_m
samples = np.random.choice(df_clean.shape[0], 5000, replace = False)
samp = df_clean.loc[samples,:]

plt.figure(figsize = [8, 6]);
plt.scatter(data = samp, x = 'member_age', y = 'duration_min', alpha = 1/10);
#plt.xlim([0, 3.5])
plt.xlabel('Member Age');
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
plt.ylabel('Duration (min)');
```

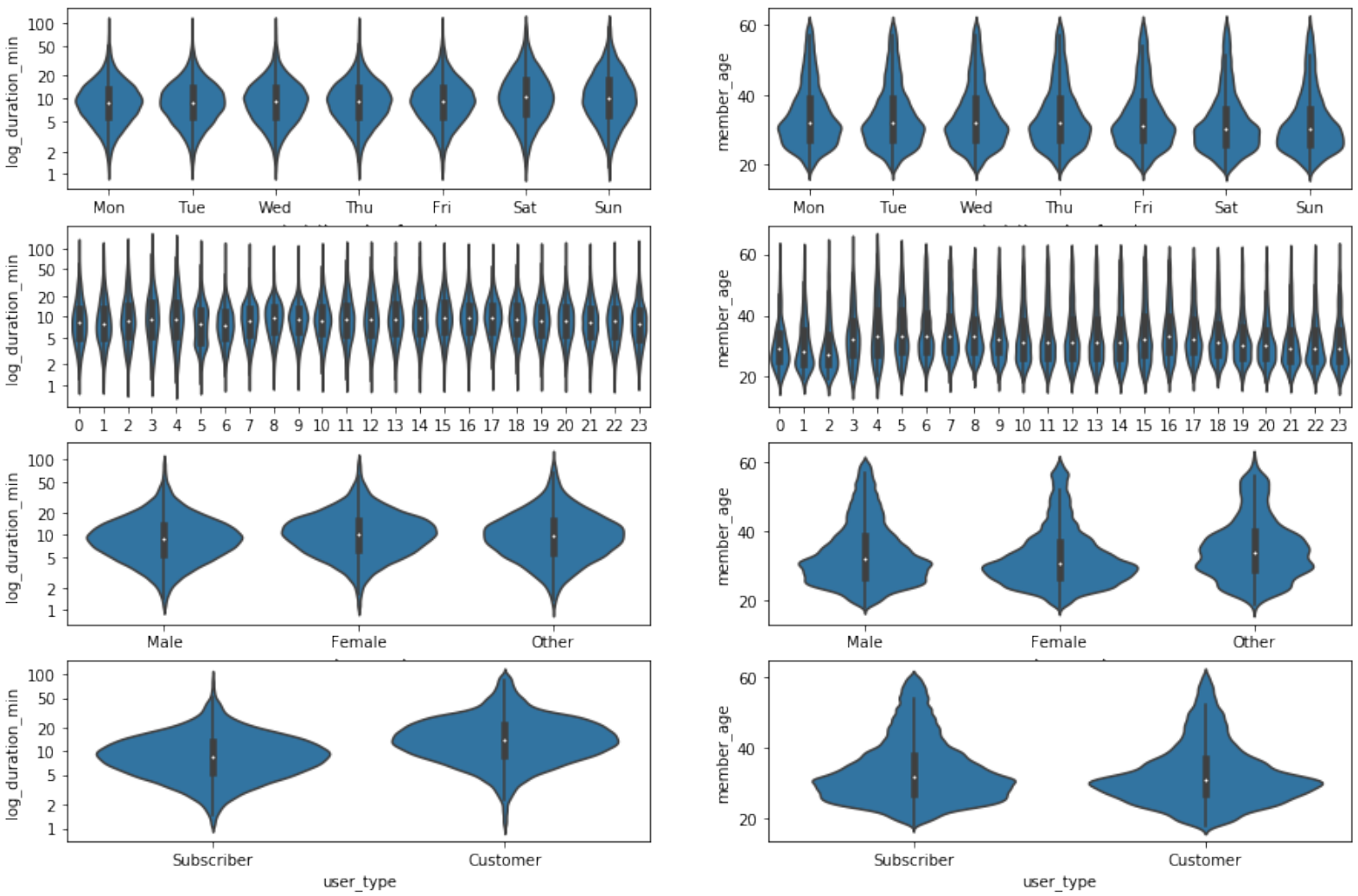


In [59]:

```
# compute the logarithm of price to make multivariate plotting easier
def log_trans(x, inverse = False):
    """ quick function for computing log and power operations """
    if not inverse:
        return np.log10(x)
    else:
        return np.power(10, x)

df_clean['log_duration_min'] = df_clean['duration_min'].apply(log_trans)

```

In [60]:

```
# plot the categorical variables against duration_min and member_age again, this
# with full data and variable transforms
fig, ax = plt.subplots(ncols = 2, nrows = 4 , figsize = [15,10])

for i in range(len(categoric_vars)):
    var = categoric_vars[i]
    sb.violinplot(data = df_clean, x = var, y = 'log_duration_min', ax = ax[i,0
    ax[i,0].set_yticks(log_trans(np.array([1, 2, 5, 10, 20, 50, 100])));
    ax[i,0].set_yticklabels([1, 2, 5, 10, 20, 50, 100]);
    sb.violinplot(data = df_clean, x = var, y = 'member_age', ax = ax[i,1], colo
```

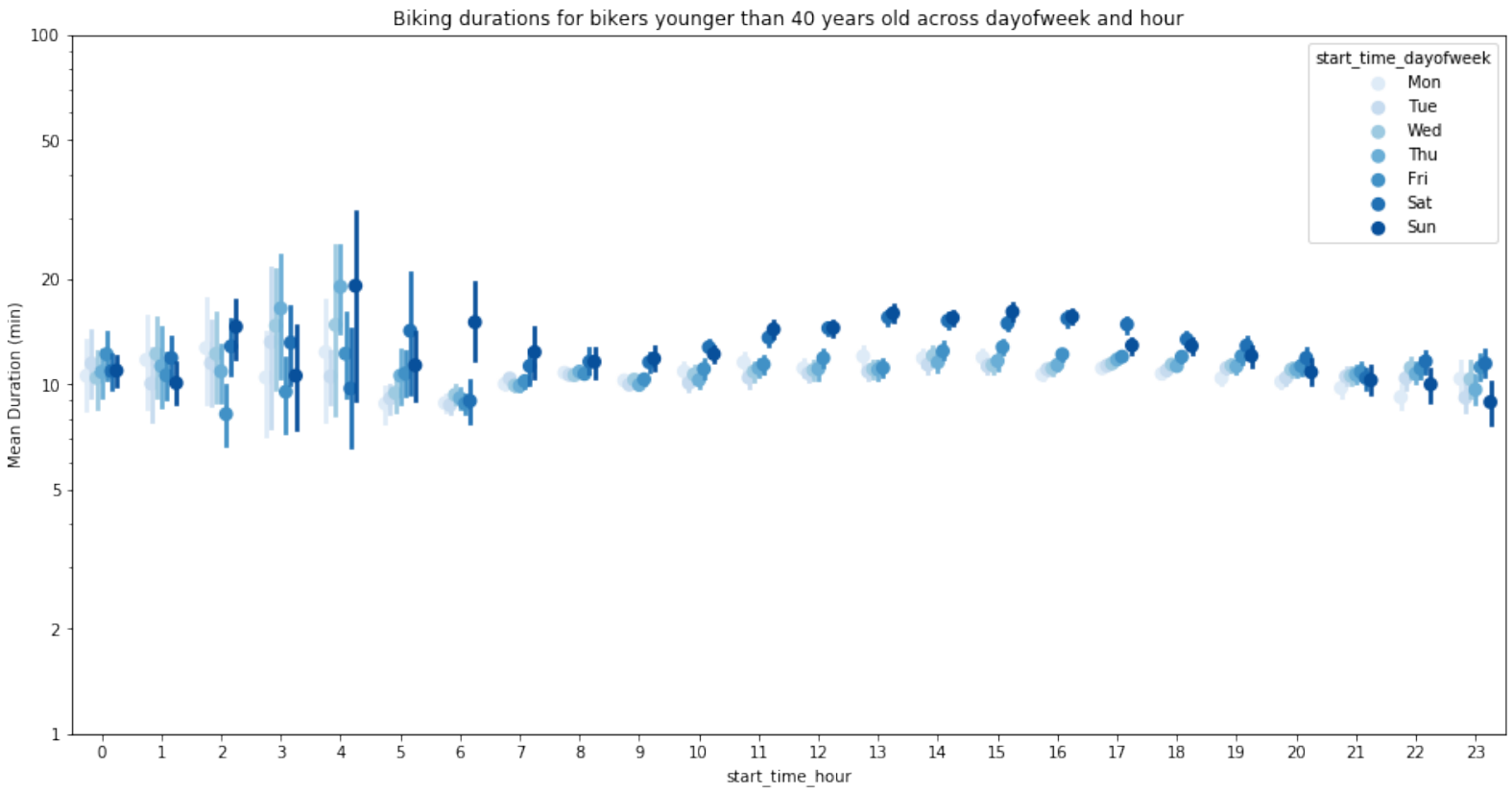


In [63]:

```
age_flag1 = (df_clean['member_age'] < 40)
age_below_forty = df_clean.loc[age_flag1,:]

age_flag2 = (df_clean['member_age'] >= 40)
age_above_forty = df_clean.loc[age_flag2,:]
```

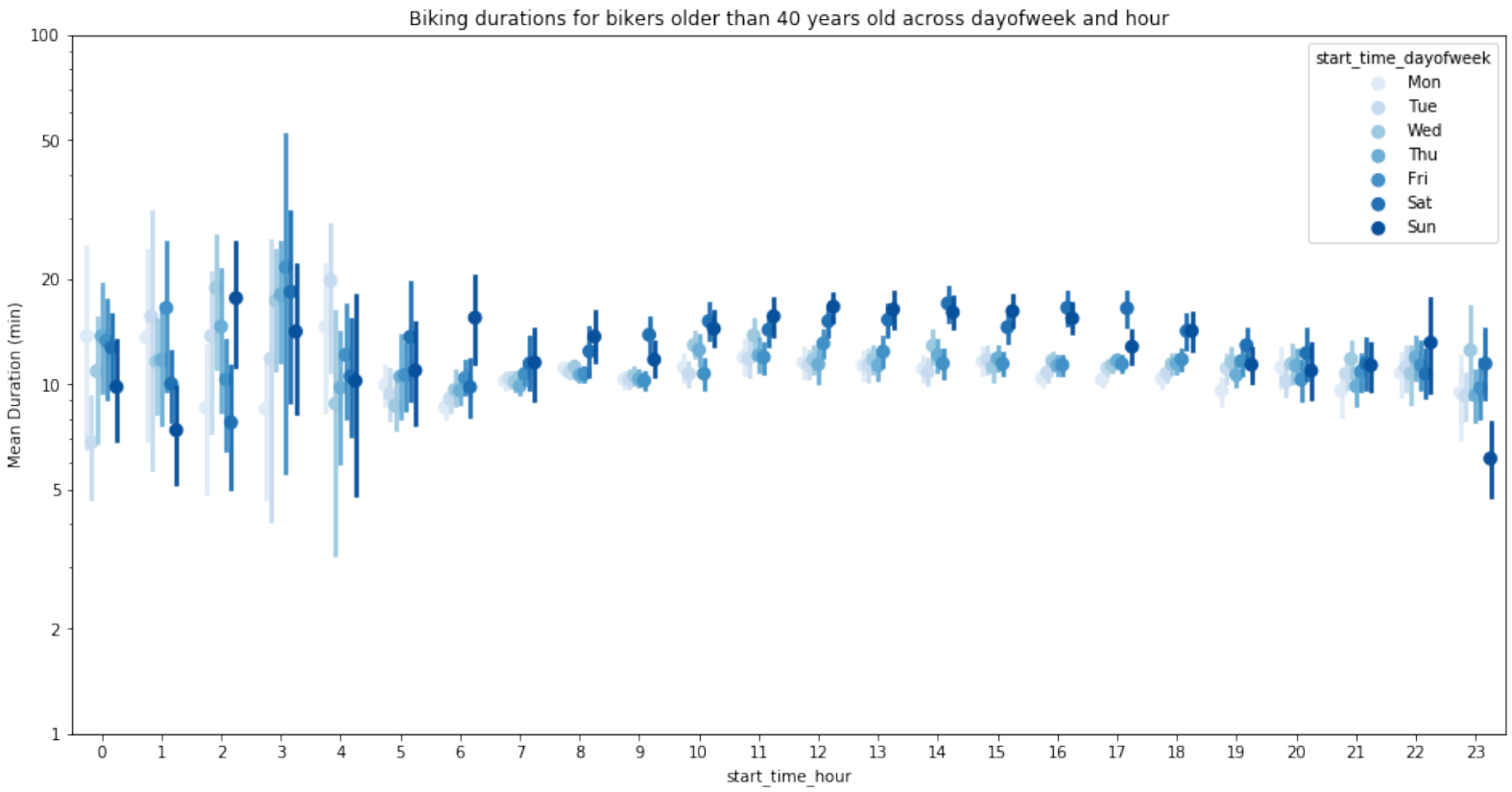
In [64]:

```
fig = plt.figure(figsize = [16,8])

ax = sb.pointplot(data = age_below_forty, x = 'start_time_hour', y = 'duration_r
          palette = 'Blues', linestyles = '', dodge = 0.5);

plt.title('Biking durations for bikers younger than 40 years old across dayofwee
plt.ylabel('Mean Duration (min)');
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
ax.set_yticklabels([],minor = True);
```



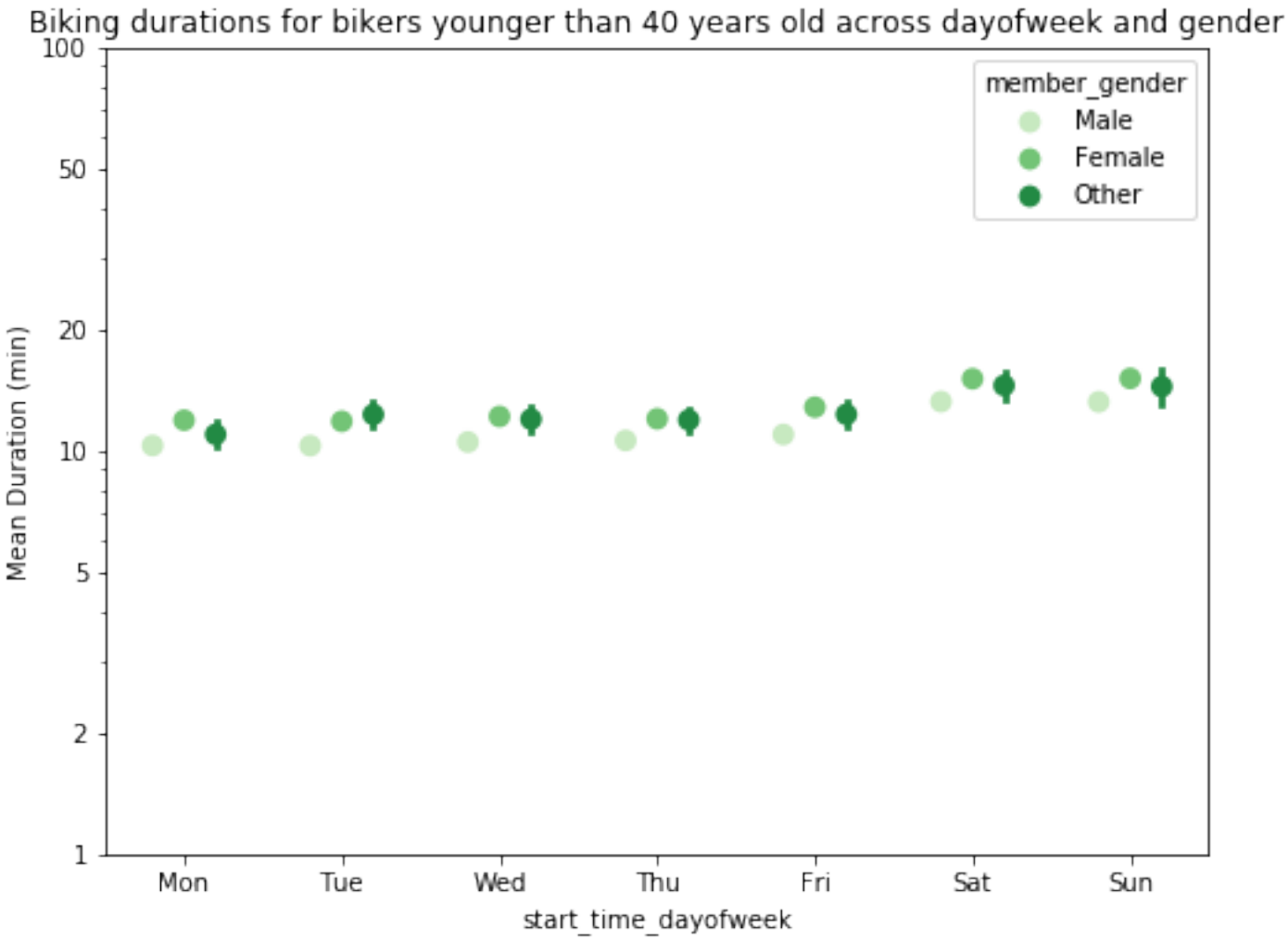
In [65]:

```
fig = plt.figure(figsize = [16,8])

ax = sb.pointplot(data = age_above_forty, x = 'start_time_hour', y = 'duration_r
           palette = 'Blues', linestyles = '', dodge = 0.5);

plt.title('Biking durations for bikers older than 40 years old across dayofweek
plt.ylabel('Mean Duration (min)');
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
ax.set_yticklabels([],minor = True);
```



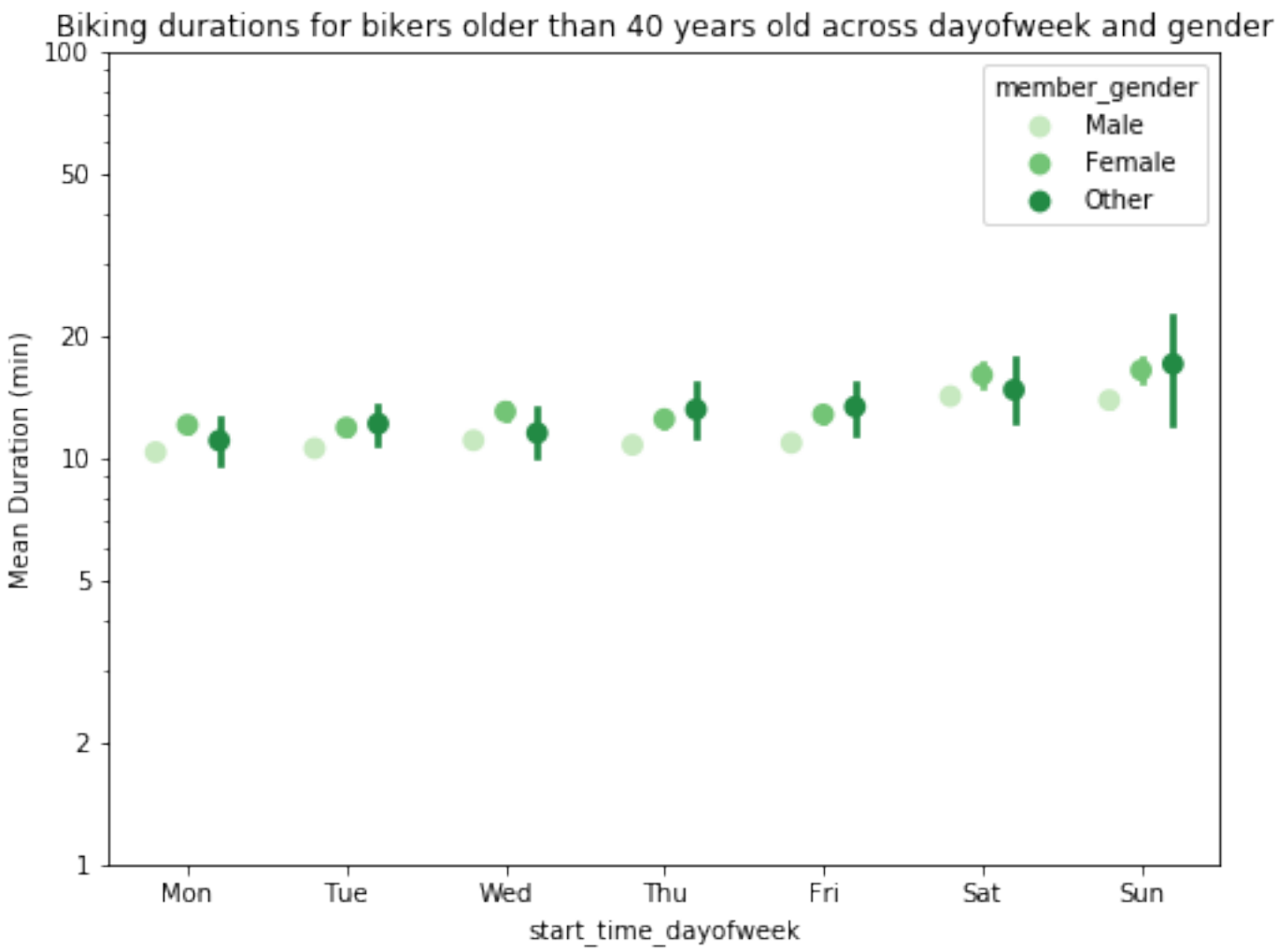
In [66]:

```
fig = plt.figure(figsize = [8,6]);

ax = sb.pointplot(data = age_below_forty, x = 'start_time_dayofweek', y = 'durat
                  palette = 'Greens', linestyles = '', dodge = 0.4);

plt.title('Biking durations for bikers younger than 40 years old across dayofwee
plt.ylabel('Mean Duration (min)');
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
ax.set_yticklabels([],minor = True);
```



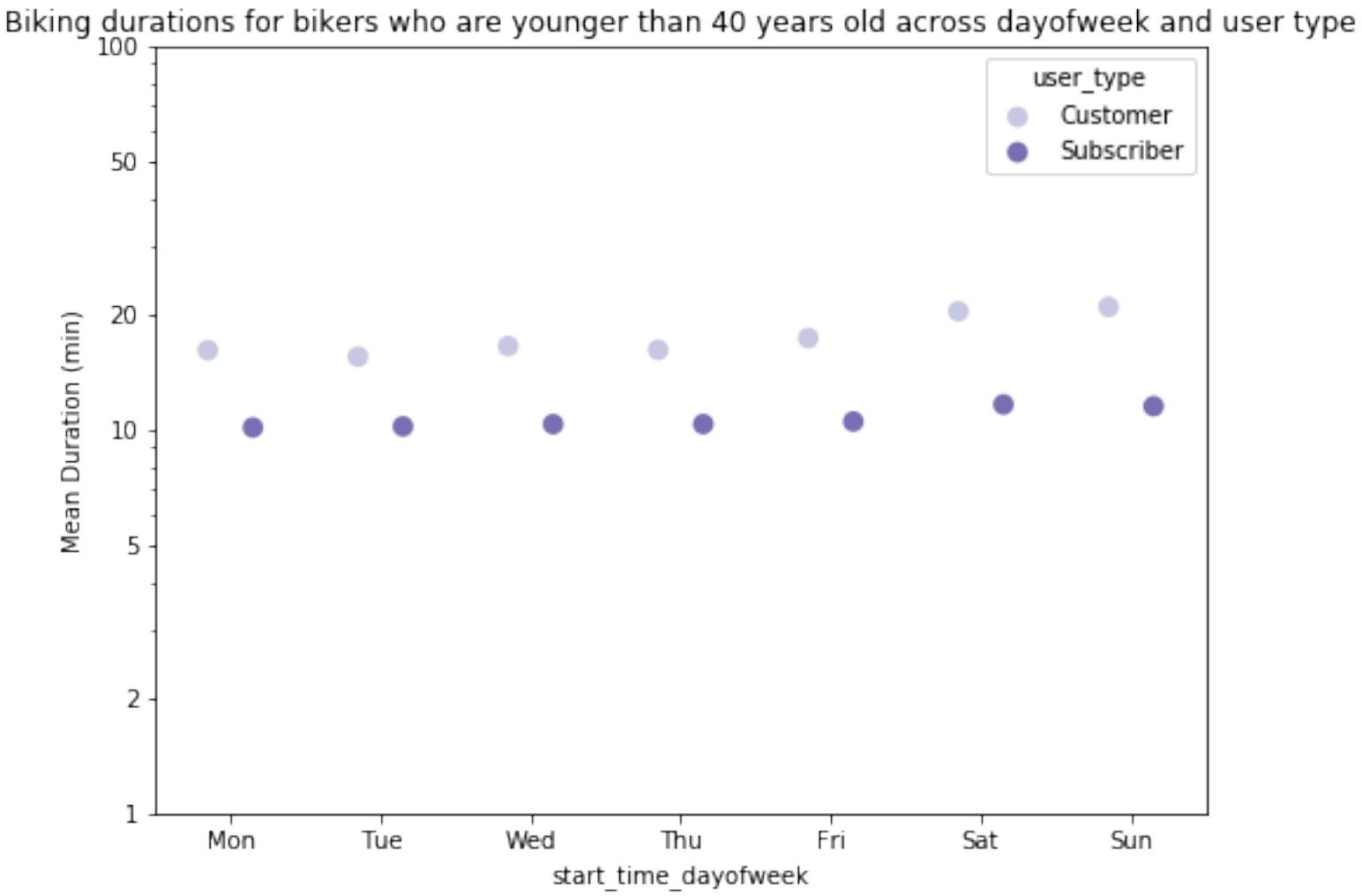
In [67]:

```
fig = plt.figure(figsize = [8,6]);

ax = sb.pointplot(data = age_above_forty, x = 'start_time_dayofweek', y = 'durat
                  palette = 'Greens', linestyles = '', dodge = 0.4);

plt.title('Biking durations for bikers older than 40 years old across dayofweek
plt.ylabel('Mean Duration (min)');
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
ax.set_yticklabels([],minor = True);
```



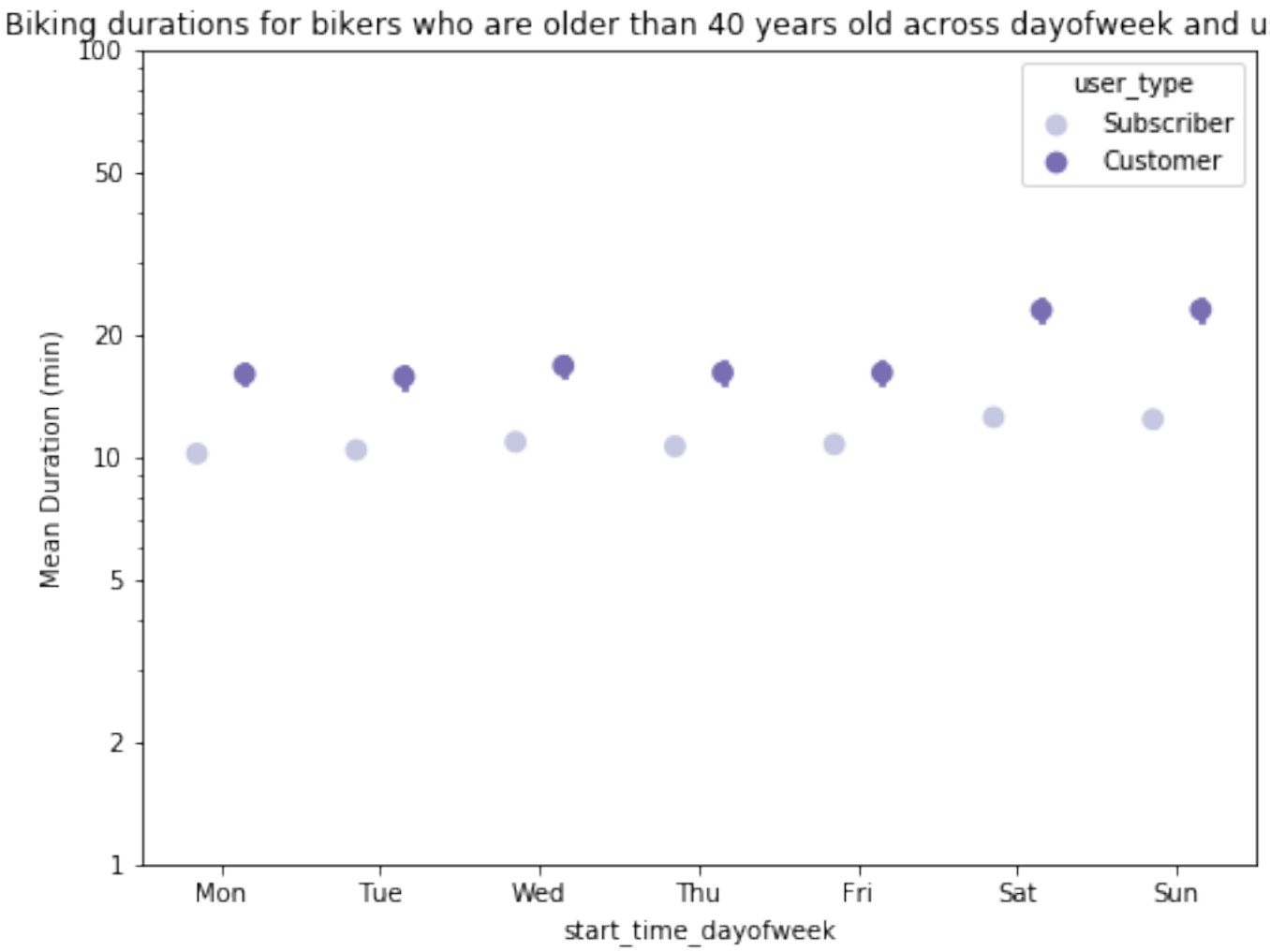
In [68]:

```
fig = plt.figure(figsize = [8,6])
ax = sb.pointplot(data = age_below_forty, x = 'start_time_dayofweek', y = 'durat
           palette = 'Purples', linestyles = '', dodge = 0.3);
plt.title('Biking durations for bikers who are younger than 40 years old across
plt.ylabel('Mean Duration (min)')
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
ax.set_yticklabels([],minor = True);
```



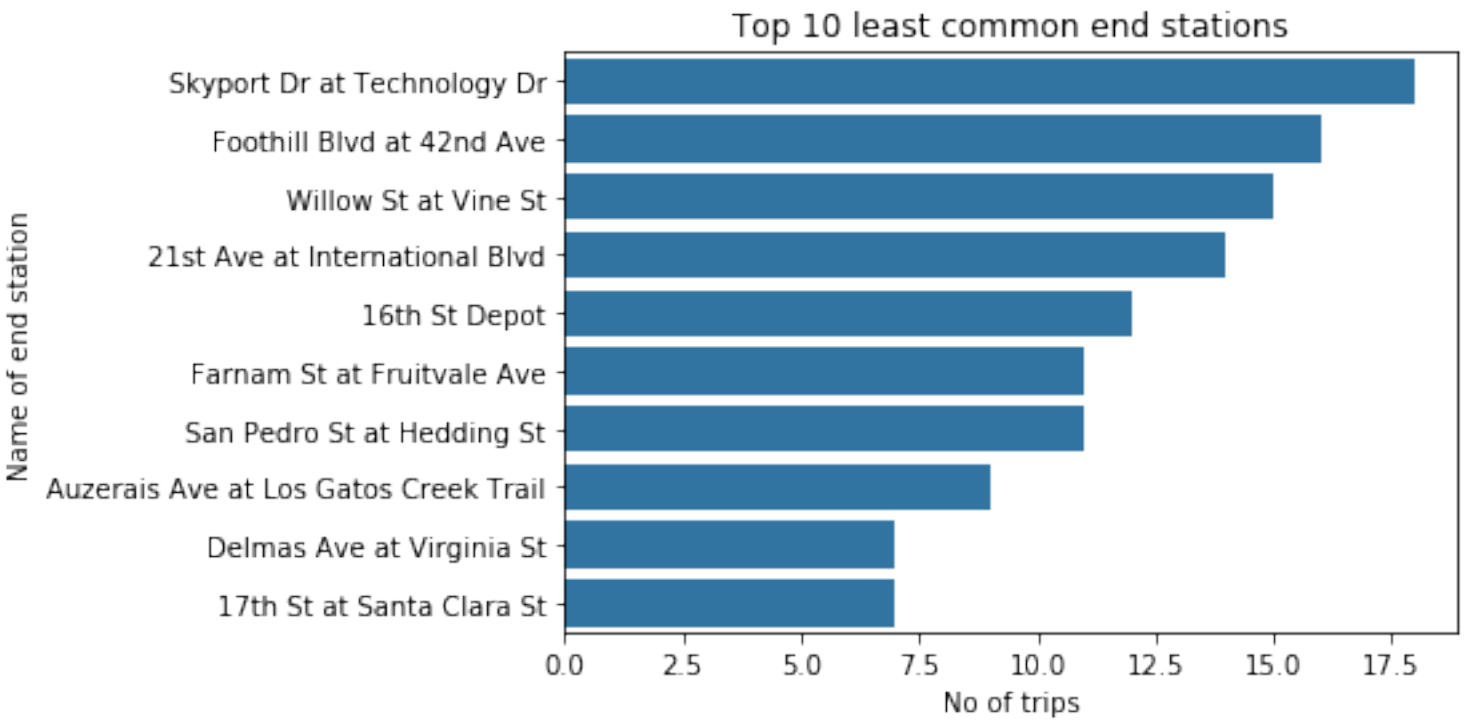
In [69]:

```
fig = plt.figure(figsize = [8,6]);
ax = sb.pointplot(data = age_above_forty, x = 'start_time_dayofweek', y = 'durat
           palette = 'Purples', linestyles = '', dodge = 0.3);
plt.title('Biking durations for bikers who are older than 40 years old across da
plt.ylabel('Mean Duration (min)');
plt.yscale('log');
plt.yticks([1, 2, 5, 10, 20, 50, 100], [1, 2, 5, 10, 20, 50, 100]);
ax.set_yticklabels([],minor = True);
```



In [70]:

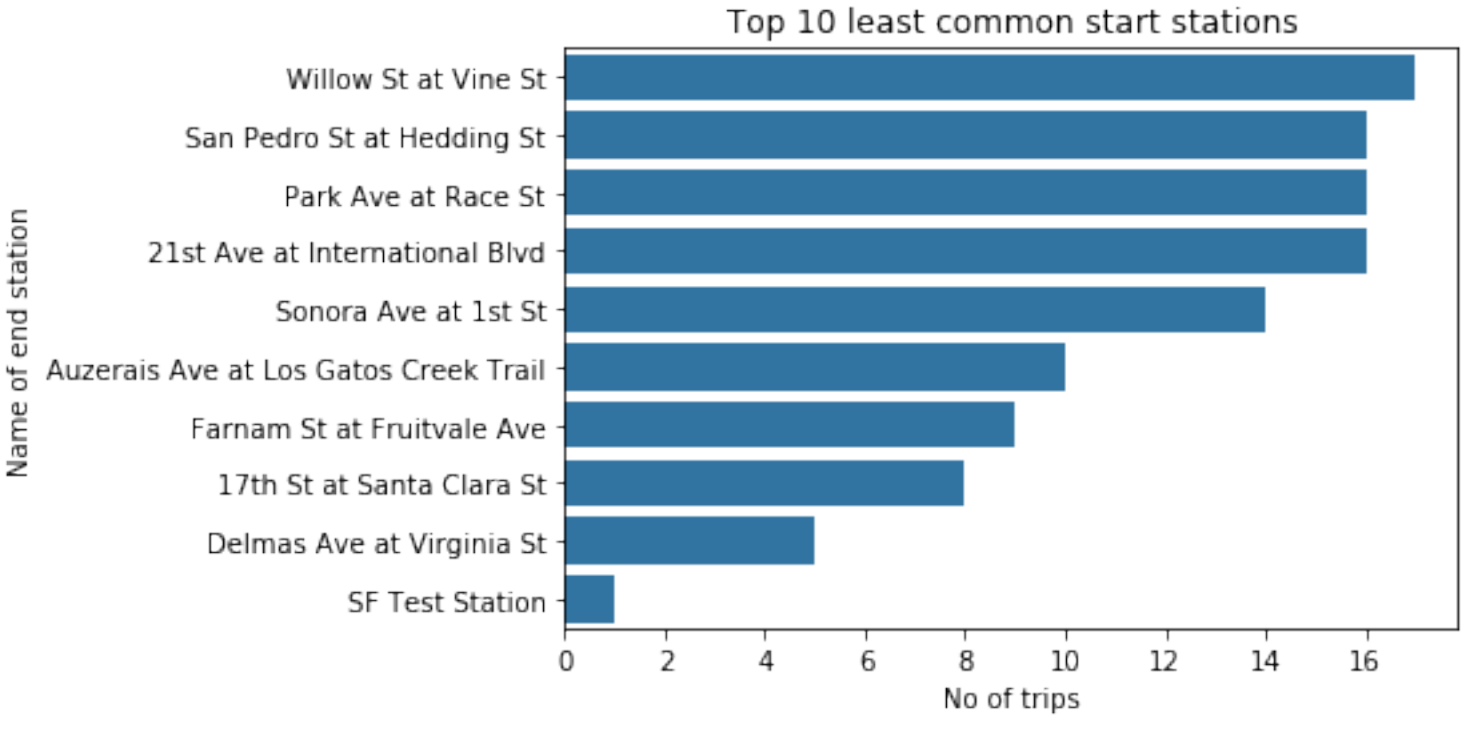
```
1
2  neighbourhood_counts = df['end_station_name'].value_counts()
3  neighbourhood_order = neighbourhood_counts.index
4  sb.countplot(data = df, y = 'end_station_name', order = neighbourhood_order[-10
5  plt.xlabel('No of trips')
6  plt.ylabel('Name of end station')
7  plt.title('Top 10 least common end stations');
```



based on the graph above we can see the top 10 names of the end stationss where most bike stop

In [71]:

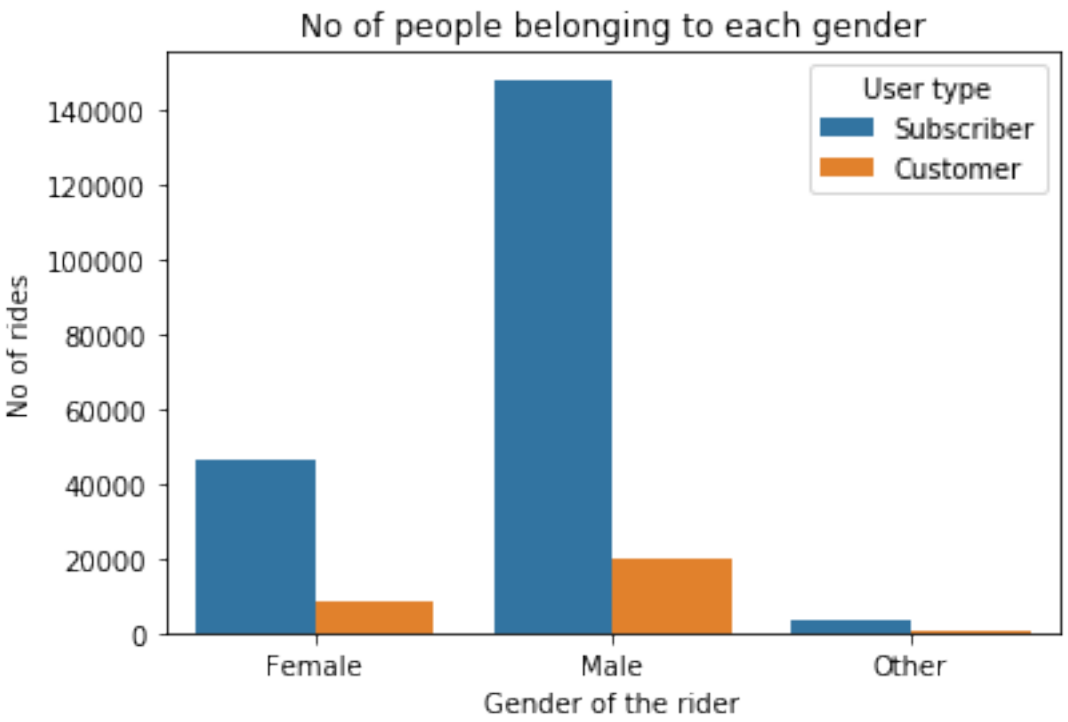
```
neighbourhood_counts = df['start_station_name'].value_counts()
neighbourhood_order = neighbourhood_counts.index
sb.countplot(data = df, y = 'start_station_name', order = neighbourhood_order[-1
plt.xlabel('No of trips')
plt.ylabel('Name of end station')
plt.title('Top 10 least common start stations');
```



based on the graph above we can see the top 10 names of the start stationss where most bike start

In [72]:

```
sb.countplot(data=df, x='member_gender', hue='user_type');
plt.xlabel('Gender of the rider')
plt.ylabel('No of rides')
plt.title('No of people belonging to each gender')
plt.legend(title='User type');
```



We can see that most of the rides are. done by males

In [73]:

```
df_clean.head()
```

Out[73]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

5 rows × 21 columns

In [74]:

```
df_clean.head()
```

Out[74]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

5 rows × 21 columns

In [75]:

```
# Adding Month, Day and Hour of tweets and retweets

df_clean['start_month'] = df_clean['start_time'].dt.month_name()
df_clean['start_day'] = df_clean['start_time'].dt.day_name()
df_clean['start_hour'] = df_clean['start_time'].dt.hour


df_clean['end_month'] = df_clean['end_time'].dt.month_name()
df_clean['end_day'] = df_clean['end_time'].dt.day_name()
df_clean['end_hour'] = df_clean['end_time'].dt.hour


```

In [76]:

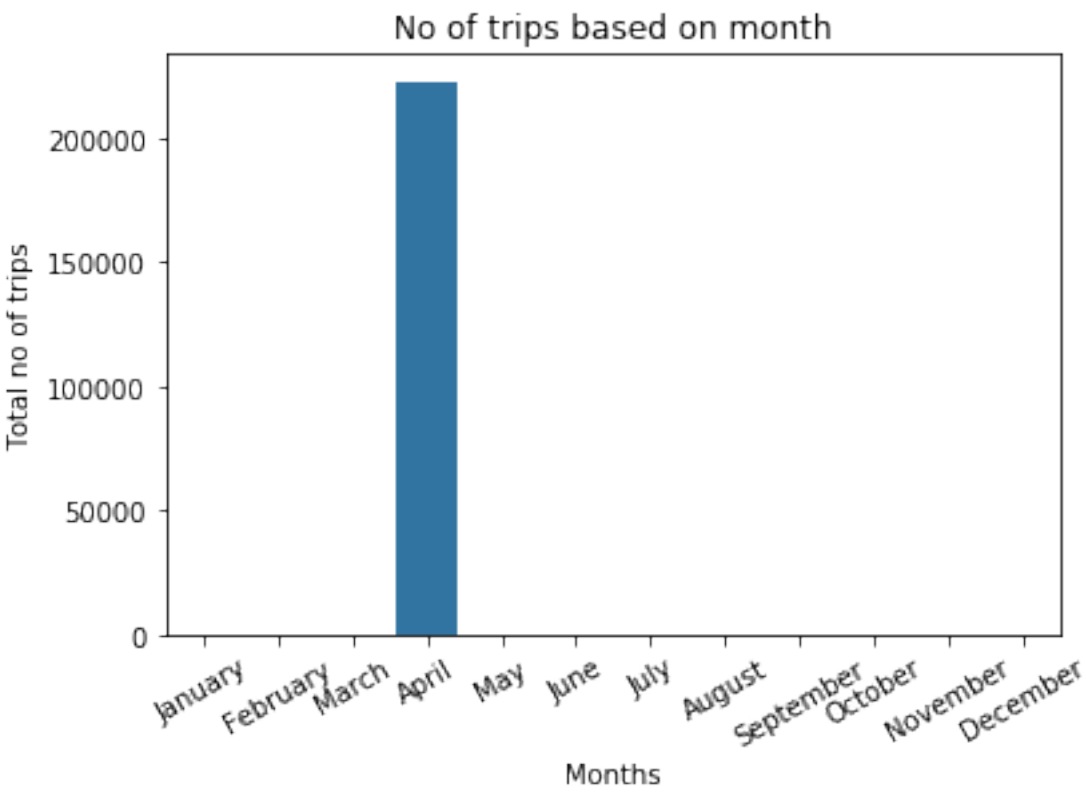
```
df_clean
```

Out[76]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_s
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.760447	
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.869360	
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.800754	
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.781074	
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.789625	

In [77]:

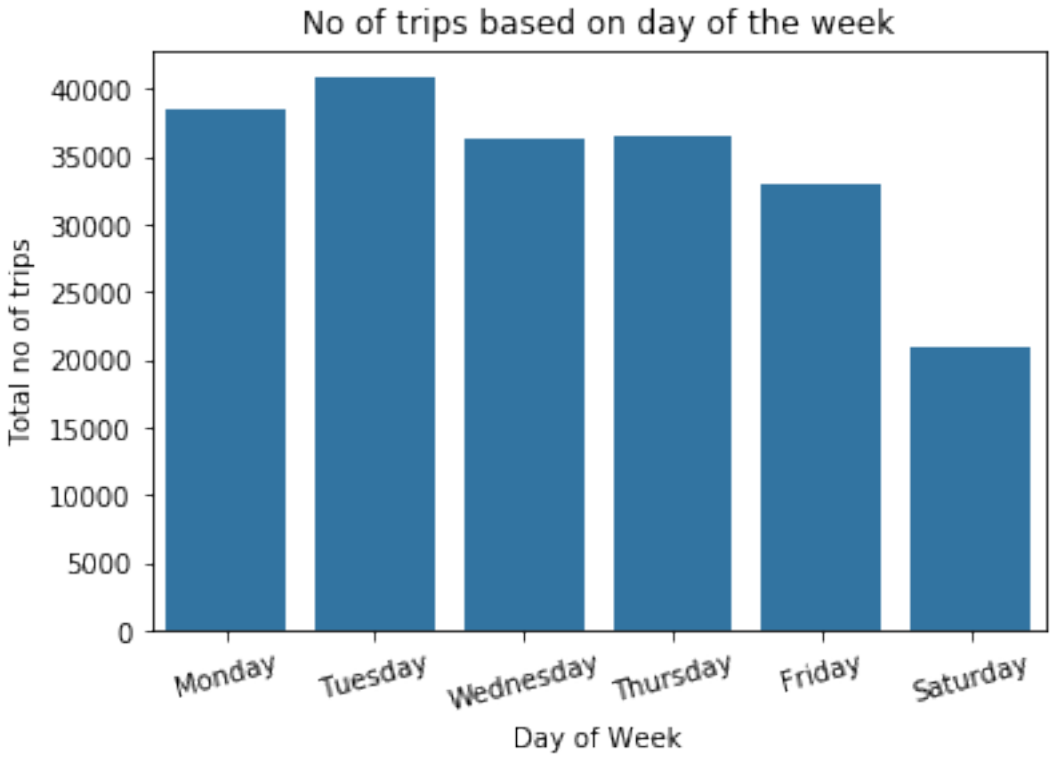
```
x_marker = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'Aug
            'November', 'December']
sb.countplot(data=df_clean, x='start_month', order=x_marker, color=default_colo
plt.xticks(rotation=30)
plt.title('No of trips based on month')
plt.xlabel('Months')
plt.ylabel('Total no of trips');
```



The above graph shows that the given dataset contains the. data only for the moth of april

In [78]:

```
x_marker = ['Monday', 'Tuesday', 'Wednesday', 'Thursday','Friday', 'Saturday']
sb.countplot(data=df_clean, x='start_day', order=x_marker, color=default_color);
plt.title('No of trips based on day of the week')
plt.xlabel('Day of Week')
plt.xticks(rotation=15)
plt.ylabel('Total no of trips');
```



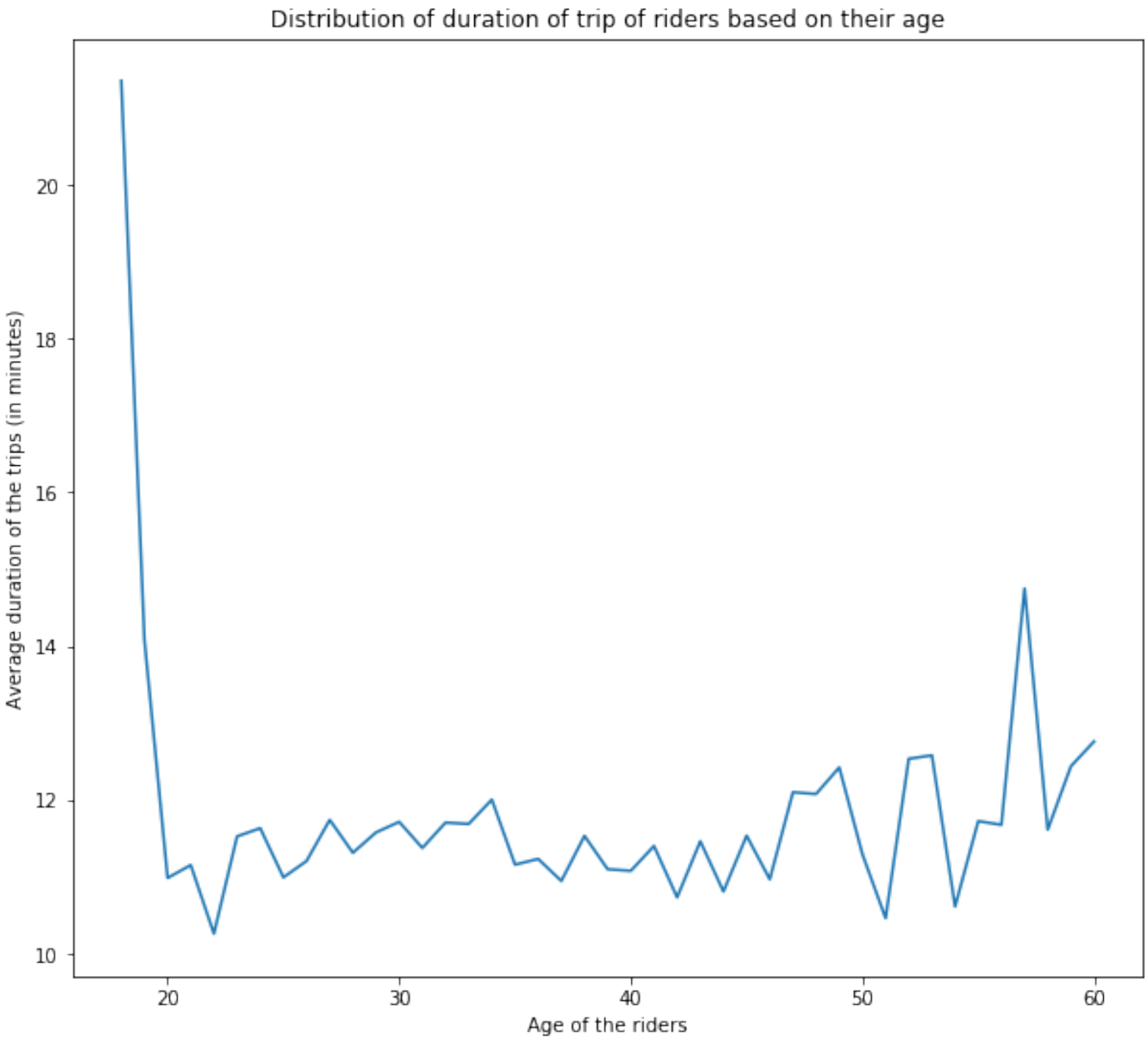
The above graph shows that most of the rides were started on Tuesday and the least on Saturday

In [79]:

```
df_clean.to_csv('clean_master_file.csv', index=False)
```

In [80]:

```
plt.figure(figsize=(10,9))
sb.lineplot(data=df_clean[df_clean['member_age']<100], x='member_age', y='durati
plt.xlabel('Age of the riders')
plt.ylabel('Average duration of the trips (in minutes)')
plt.title('Distribution of duration of trip of riders based on their age');
```



We can see that the highest duration of trips is for the age groups around 20, aslo it show. there is no one older than 60 is riding bike

In [81]:

```
df_clean[df_clean['member_age']>60]
```

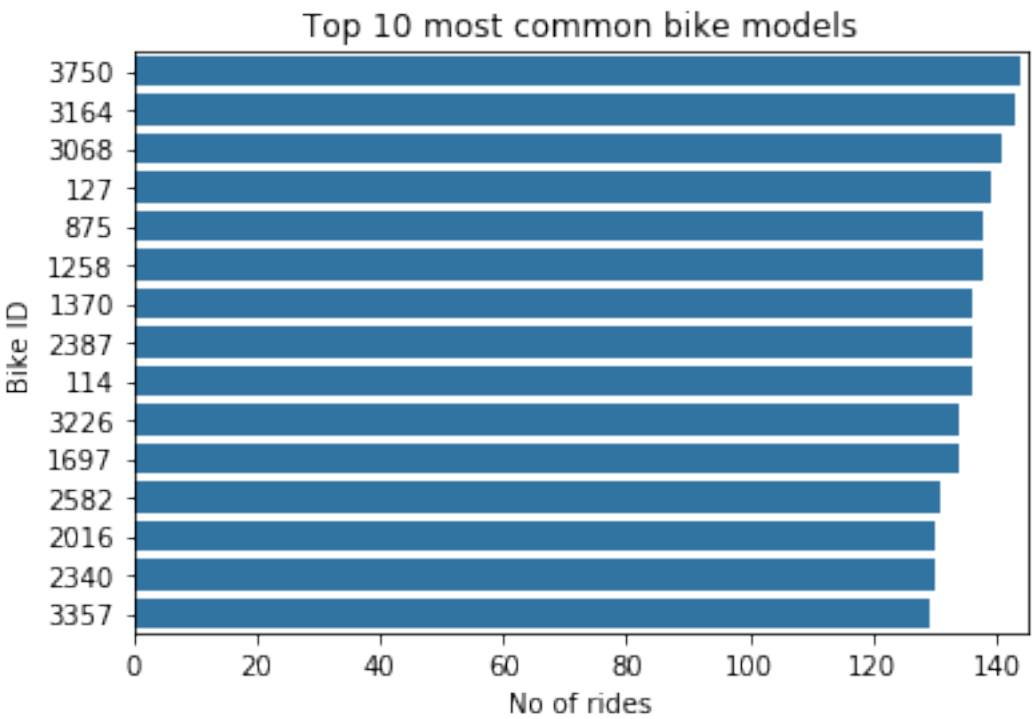
Out[81]:

duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	st

0 rows × 27 columns

In [82]:

```

neighbourhood_counts = df['bike_id'].value_counts()
neighbourhood_order = neighbourhood_counts.index
plt.xlim(0,df['bike_id'].value_counts().max() + 1)
sb.countplot(data = df, y = 'bike_id', order = neighbourhood_order[0:15], color=
plt.xlabel('No of rides')
plt.ylabel('Bike ID')
plt.title('Top 10 most common bike models');
```



based on the graph. above we can see that bike model 3750 is the most used

In [83]:

```
1 df_clean.head()
```

Out[83]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

5 rows × 27 columns

In [ ]:

```
1
```

DATA Analyzing for Ford GoBike

by. hamed bintalib

Ford GoBike is the Bay Area's bike share system. Bay Area Bike Share was introduced in 2013 as a pilot program for the region, with 700 bikes and 70 stations across San Francisco and San Jose. By the end of 2018, Ford GoBike will grow to 7,000 bikes across San Francisco, the East Bay and San Jose.

In [2]:

In [3]:

In [4]:

Out[4]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_lat
0	50305	2019-04-30 22:33:55.1550	2019-05-01 12:32:20.4540	368.0	Myrtle St at Polk St	37.78
1	53725	2019-04-30 20:43:41.6320	2019-05-01 11:39:06.9170	246.0	Berkeley Civic Center	37.86
2	78072	2019-04-30 10:32:46.4890	2019-05-01 08:13:58.9750	64.0	5th St at Brannan St	37.77
3	78969	2019-04-30 10:00:51.5500	2019-05-01 07:57:01.2620	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.77
4	1128	2019-04-30 23:59:04.7390	2019-05-01 00:17:53.0910	124.0	19th St at Florida St	37.76

In [5]:

Out[5]:

```
duration_sec                0
start_time                  0
end_time                    0
start_station_id           64
start_station_name         64
start_station_latitude      0
start_station_longitude     0
end_station_id             64
end_station_name           64
end_station_latitude        0
end_station_longitude       0
bike_id                     0
user_type                   0
member_birth_year       11199
member_gender           11199
bike_share_for_all_trip     0
dtype: int64
```

In [6]:

Out[6]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start
10983	1131	2019-04-29 15:30:48.3890	2019-04-29 15:49:39.4800	NaN	NaN	37.41	
11568	268	2019-04-29 13:12:08.9280	2019-04-29 13:16:37.8740	NaN	NaN	37.41	
14814	669	2019-04-28 23:02:16.2860	2019-04-28 23:13:25.6670	NaN	NaN	37.40	
16067	94	2019-04-28 16:30:12.3680	2019-04-28 16:31:46.8660	NaN	NaN	37.40	
17179	2389	2019-04-28 12:24:07.6830	2019-04-28 13:03:57.4710	NaN	NaN	37.40	
17185	185	2019-04-28 12:59:44.6410	2019-04-28 13:02:49.7770	NaN	NaN	37.42	
18116	719	2019-04-28	2019-04-28	NaN	NaN	37.41	

In [7]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 239111 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               239111 non-null int64
start_time                 239111 non-null object
end_time                   239111 non-null object
start_station_id           239047 non-null float64
start_station_name         239047 non-null object
start_station_latitude     239111 non-null float64
start_station_longitude    239111 non-null float64
end_station_id             239047 non-null float64
end_station_name           239047 non-null object
end_station_latitude       239111 non-null float64
end_station_longitude      239111 non-null float64
bike_id                    239111 non-null int64
user_type                  239111 non-null object
member_birth_year          227912 non-null float64
member_gender              227912 non-null object
bike_share_for_all_trip    239111 non-null object
```

In [8]:

Out[8]:

	duration_sec	start_station_id	start_station_latitude	start_station_longitude	end_station_id
count	239111.000000	239047.000000	239111.000000	239111.000000	239047.000000
mean	802.671199	141.836538	37.769536	-122.352503	140.838095
std	1990.006091	116.289776	0.127698	0.277088	116.386168
min	61.000000	3.000000	0.000000	-122.453704	3.000000
25%	349.000000	47.000000	37.770083	-122.413004	44.000000
50%	558.000000	104.000000	37.780760	-122.398285	102.000000
75%	876.000000	240.000000	37.797280	-122.291209	239.000000
max	86114.000000	420.000000	37.880222	0.000000	420.000000

In [9]:

In [10]:

Out[10]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_lat
0	50305	2019-04-30 22:33:55.1550	2019-05-01 12:32:20.4540	368.0	Myrtle St at Polk St	37.78
2	78072	2019-04-30 10:32:46.4890	2019-05-01 08:13:58.9750	64.0	5th St at Brannan St	37.77
3	78969	2019-04-30 10:00:51.5500	2019-05-01 07:57:01.2620	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.77
4	1128	2019-04-30 23:59:04.7390	2019-05-01 00:17:53.0910	124.0	19th St at Florida St	37.76
5	1388	2019-04-30 23:53:05.9820	2019-05-01 00:16:14.3130	243.0	Bancroft Way at College Ave	37.86

In [11]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227912 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               227912 non-null int64
start_time                 227912 non-null object
end_time                   227912 non-null object
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227912 non-null float64
start_station_longitude    227912 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227912 non-null float64
end_station_longitude      227912 non-null float64
bike_id                    227912 non-null int64
user_type                  227912 non-null object
member_birth_year          227912 non-null float64
member_gender              227912 non-null object
bike_share_for_all_trip    227912 non-null object
```

In [12]:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 239111 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               239111 non-null int64
start_time                 239111 non-null object
end_time                   239111 non-null object
start_station_id           239047 non-null float64
start_station_name         239047 non-null object
start_station_latitude     239111 non-null float64
start_station_longitude    239111 non-null float64
end_station_id             239047 non-null float64
end_station_name           239047 non-null object
end_station_latitude       239111 non-null float64
end_station_longitude      239111 non-null float64
bike_id                    239111 non-null int64
user_type                  239111 non-null object
member_birth_year          227912 non-null float64
member_gender              227912 non-null object
bike_share_for_all_trip    239111 non-null object
```

In [13]:

Out[13]:

```
Male      168140
Female     55498
Other       4274
Name: member_gender, dtype: int64
```

In [14]:

Out[14]:

```
Subscriber    198510
Customer       29402
Name: user_type, dtype: int64
```

In [15]:

Out[15]:

```
64
```

In [16]:

In [17]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null object
end_time                   227848 non-null object
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
```

In [18]:

Out[18]:

```
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
```

In [19]:

Out[19]:

0

In [20]:

Out[20]:

0

In [21]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 16 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null object
end_time                   227848 non-null object
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
```

In [22]:

In [23]:

Out[23]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

In [24]:

In [25]:

Out[25]:

```
Tue    41856
Mon    39296
Thu    37234
Wed    37222
Fri    33733
Sat    21475
Sun    17032
Name: start_time_dayofweek, dtype: int64
```

In [26]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 18 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
```

In [27]:

Out[27]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

In [28]:

Out[28]:

```
Tue    41856
Mon    39296
Thu    37234
Wed    37222
Fri    33733
Sat    21475
Sun    17032
Name: start_time_dayofweek, dtype: int64
```

In [29]:

In [30]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 18 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null float64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
```

In [31]:

In [32]:

In [33]:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 19 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null int64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
```

In [34]:

Out[34]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

What is the structure of your dataset?

This data set includes information about individual rides made in a bike-sharing system covering the greater San Francisco that happened in 2019:

```
* Trip Duration (seconds)
* Start Time and Date
* End Time and Date
* 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 (Subsc
riber or Customer — “Subscriber” =   Memberor “Customer” = Casual)
```

Member Year of Birth Member Gender

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

- I'll be investigating in udration of biking time
- I'll extract dayofweek, hours for further investigation in start_time
- member_gender
- user_type

In [35]:

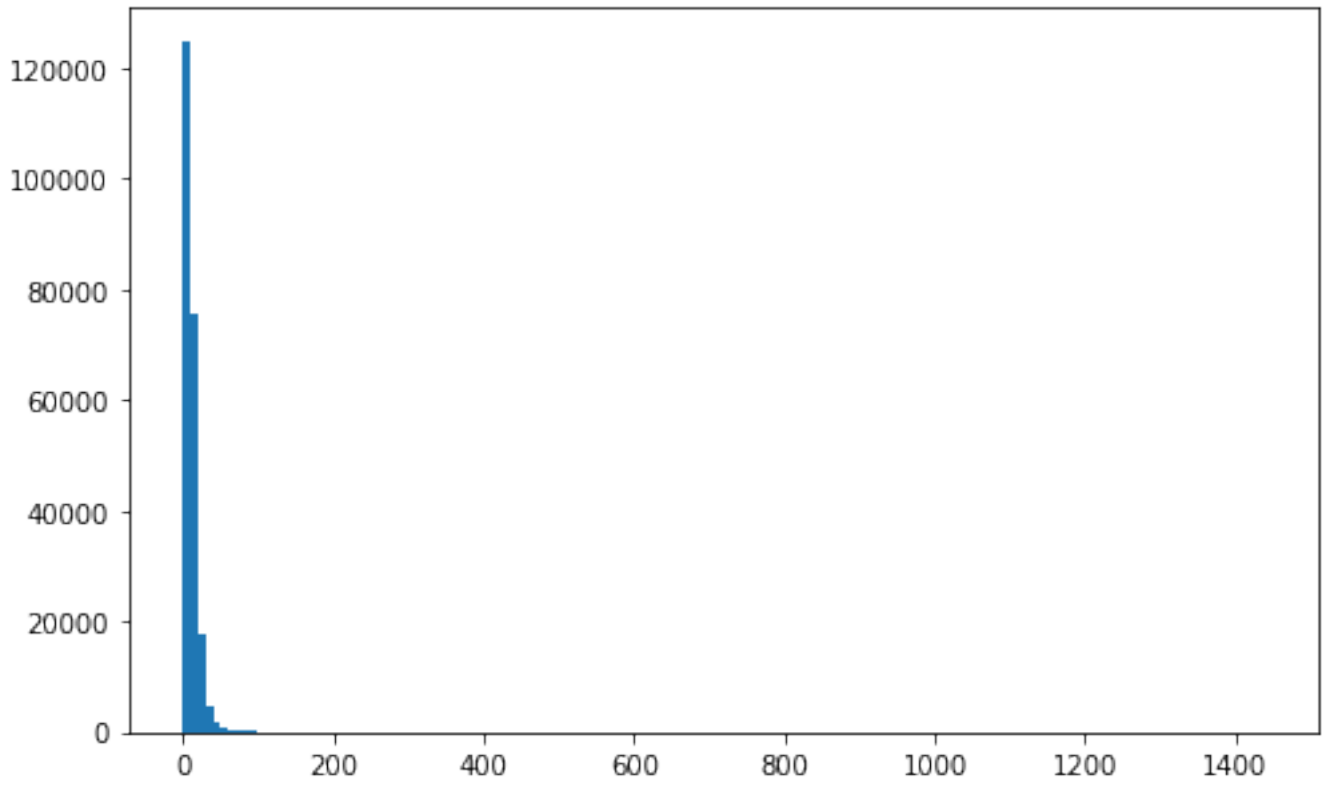
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 227848 entries, 0 to 239110
Data columns (total 20 columns):
duration_sec               227848 non-null int64
start_time                 227848 non-null datetime64[ns]
end_time                   227848 non-null datetime64[ns]
start_station_id           227848 non-null float64
start_station_name         227848 non-null object
start_station_latitude     227848 non-null float64
start_station_longitude    227848 non-null float64
end_station_id             227848 non-null float64
end_station_name           227848 non-null object
end_station_latitude       227848 non-null float64
end_station_longitude      227848 non-null float64
bike_id                    227848 non-null int64
user_type                  227848 non-null object
member_birth_year          227848 non-null int64
member_gender              227848 non-null object
bike_share_for_all_trip    227848 non-null object
```

In [36]:

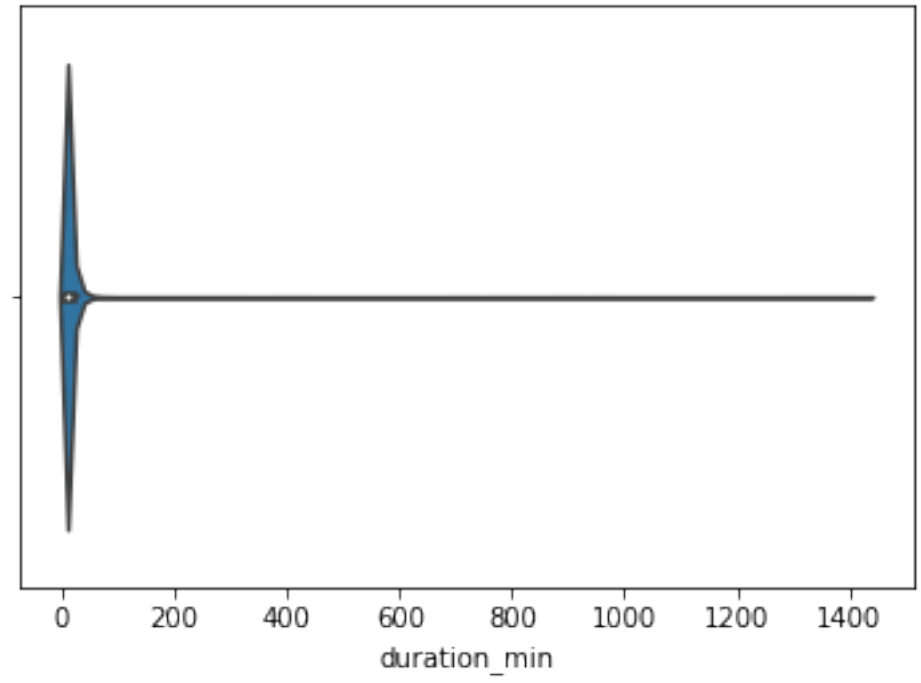
Out[36]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
0	50305	2019-04-30 22:33:55.155	2019-05-01 12:32:20.454	368.0	Myrtle St at Polk St	37.7854
2	78072	2019-04-30 10:32:46.489	2019-05-01 08:13:58.975	64.0	5th St at Brannan St	37.7767
3	78969	2019-04-30 10:00:51.550	2019-05-01 07:57:01.262	67.0	San Francisco Caltrain Station 2 (Townsend St...	37.7766
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693

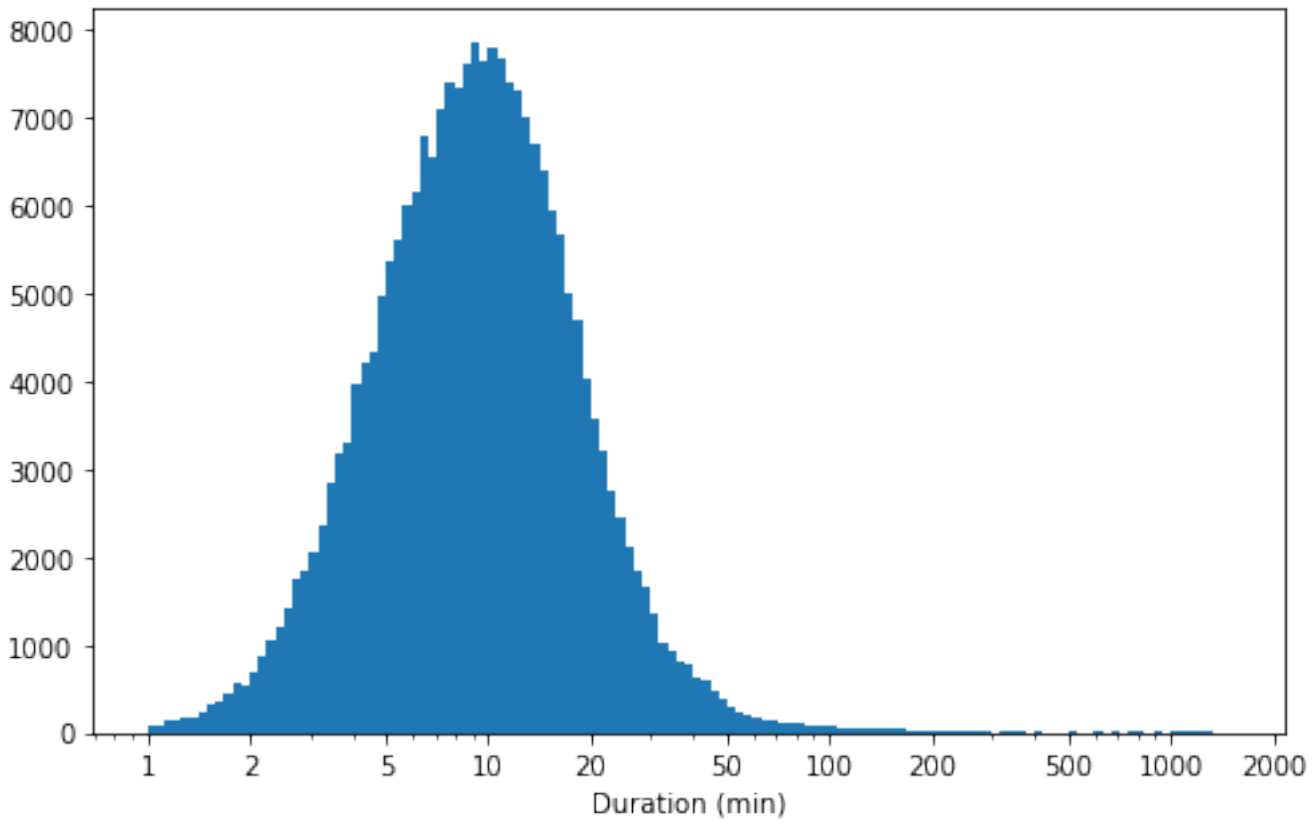
In [37]:


120000
100000
80000
60000
40000
20000
0
0
200
400
600
800
1000
1200
1400

In [38]:


0
200
400
600
800
1000
1200
1400
duration_min

In [39]:



In [40]:

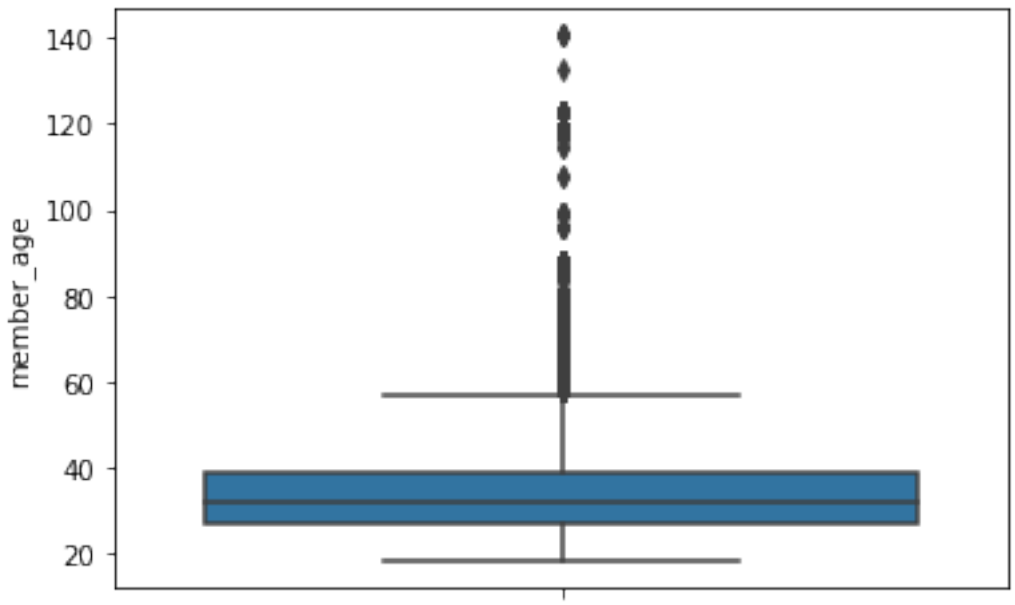
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 226804 entries, 4 to 239110
Data columns (total 20 columns):
duration_sec               226804 non-null int64
start_time                 226804 non-null datetime64[ns]
end_time                   226804 non-null datetime64[ns]
start_station_id           226804 non-null float64
start_station_name         226804 non-null object
start_station_latitude     226804 non-null float64
start_station_longitude    226804 non-null float64
end_station_id             226804 non-null float64
end_station_name           226804 non-null object
end_station_latitude       226804 non-null float64
end_station_longitude      226804 non-null float64
bike_id                    226804 non-null int64
user_type                  226804 non-null object
member_birth_year          226804 non-null int64
member_gender              226804 non-null object
bike_share_for_all_trip    226804 non-null object
```

In [41]:

Out[41]:

```
count    226804.000000
mean         34.137008
std           9.954496
min          18.000000
25%          27.000000
50%          32.000000
75%          39.000000
max         141.000000
Name: member_age, dtype: float64
```

In [42]:



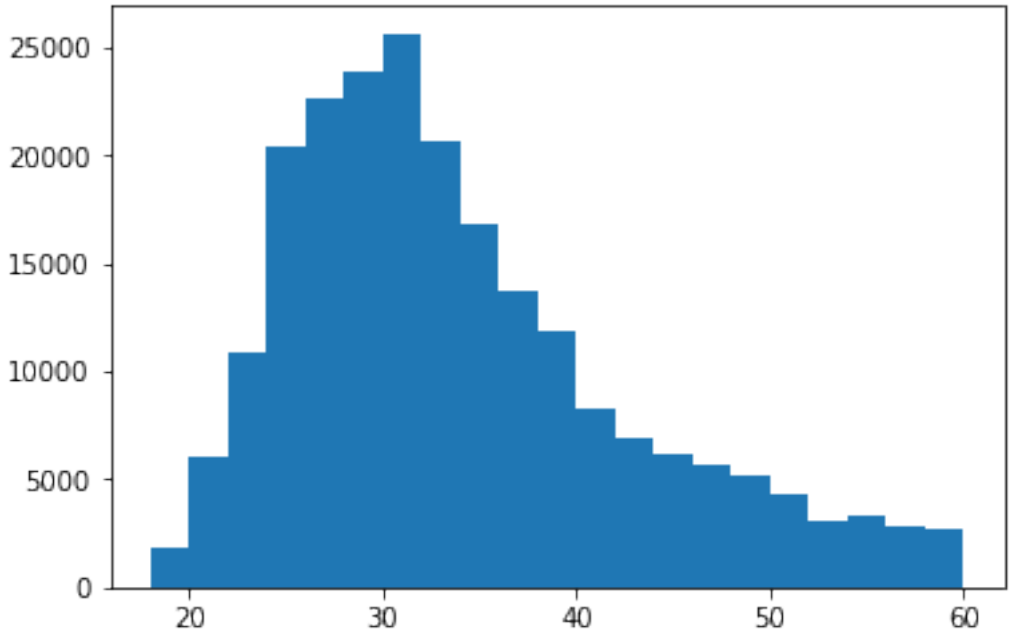
In [43]:

In [44]:

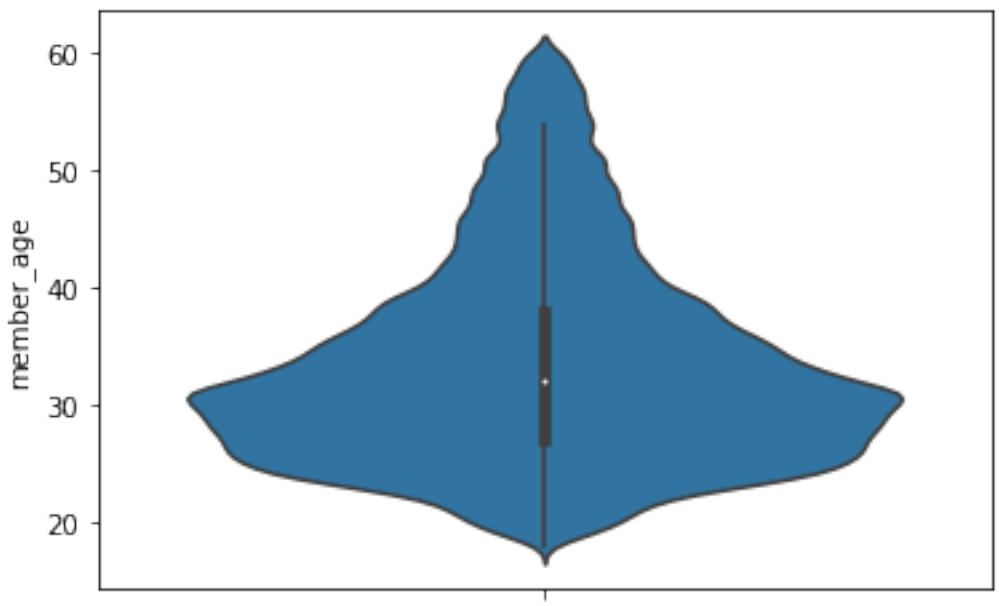
Out[44]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

In [45]:



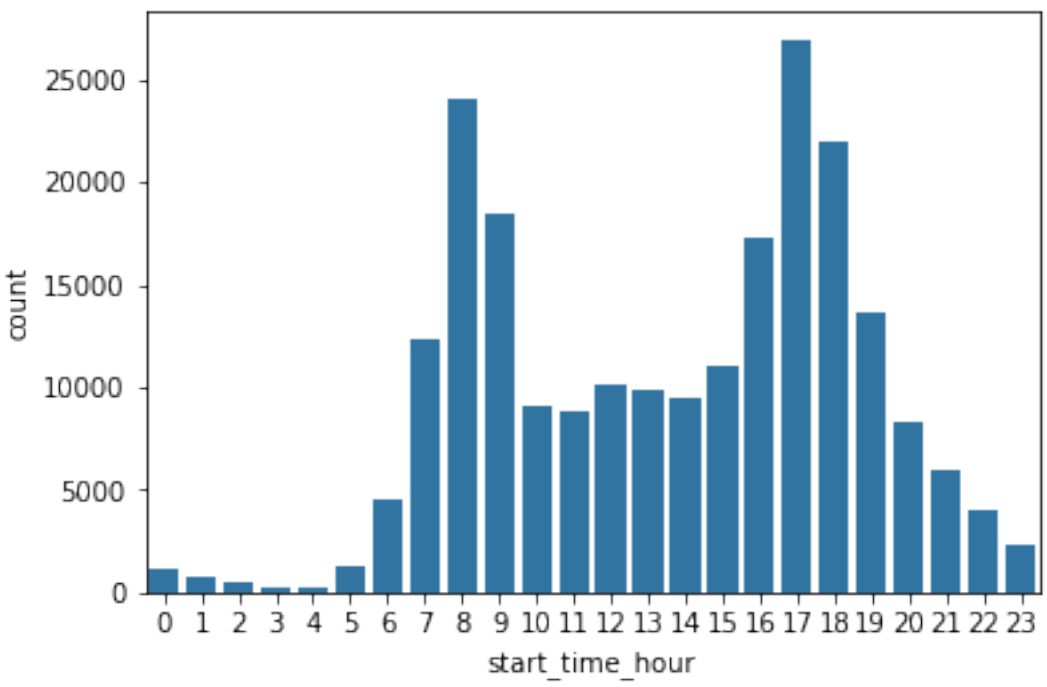
In [46]:



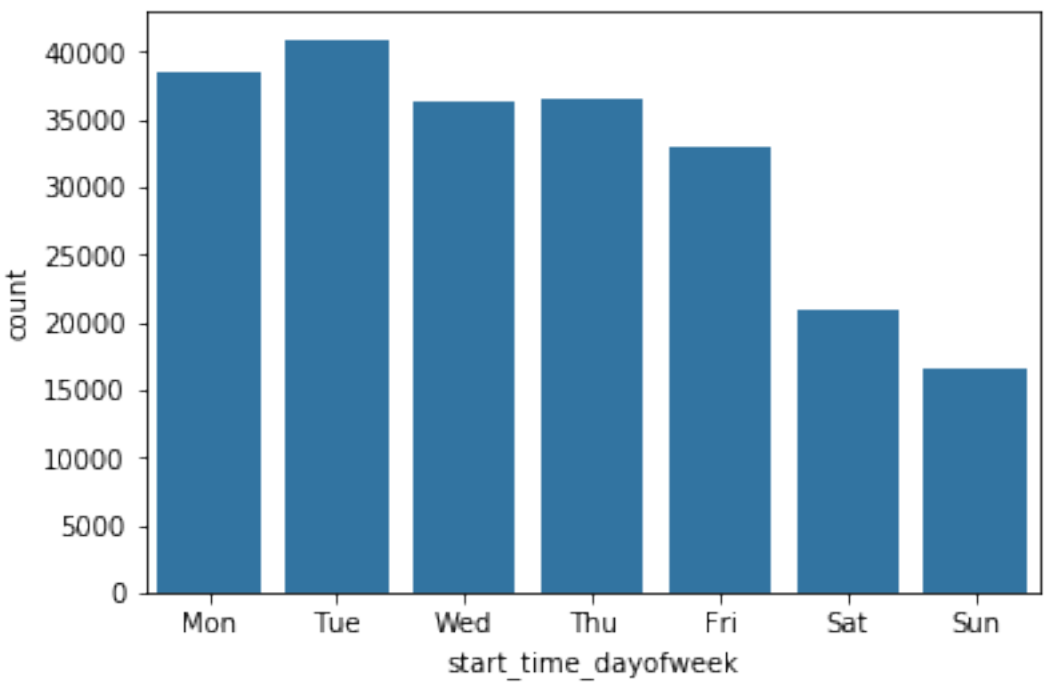
In [47]:

In [48]:

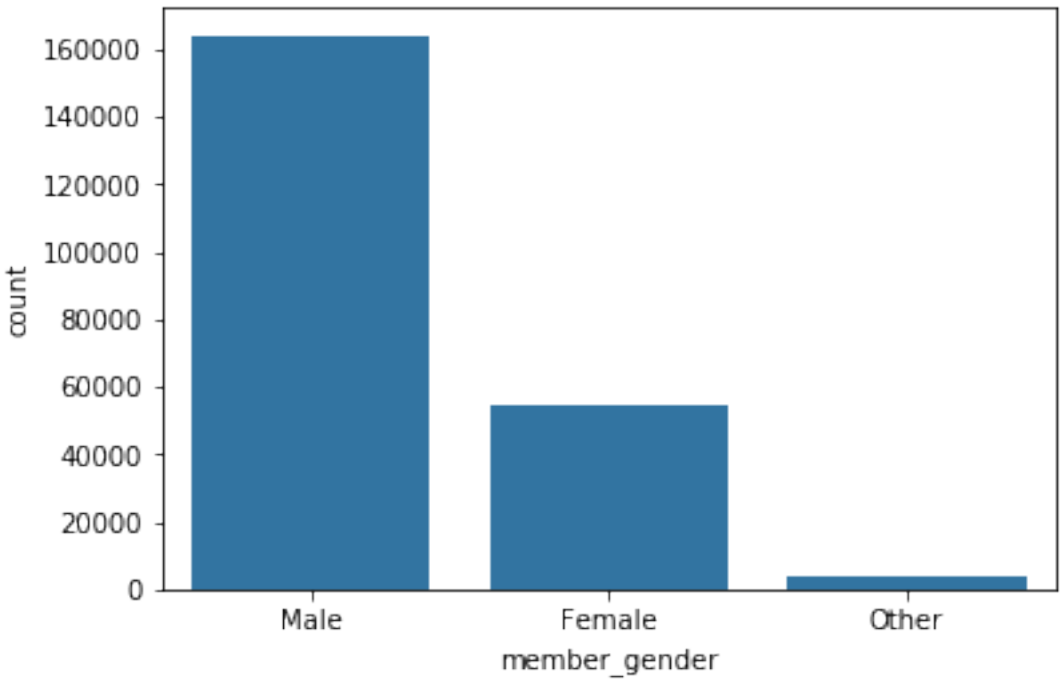
In [49]:



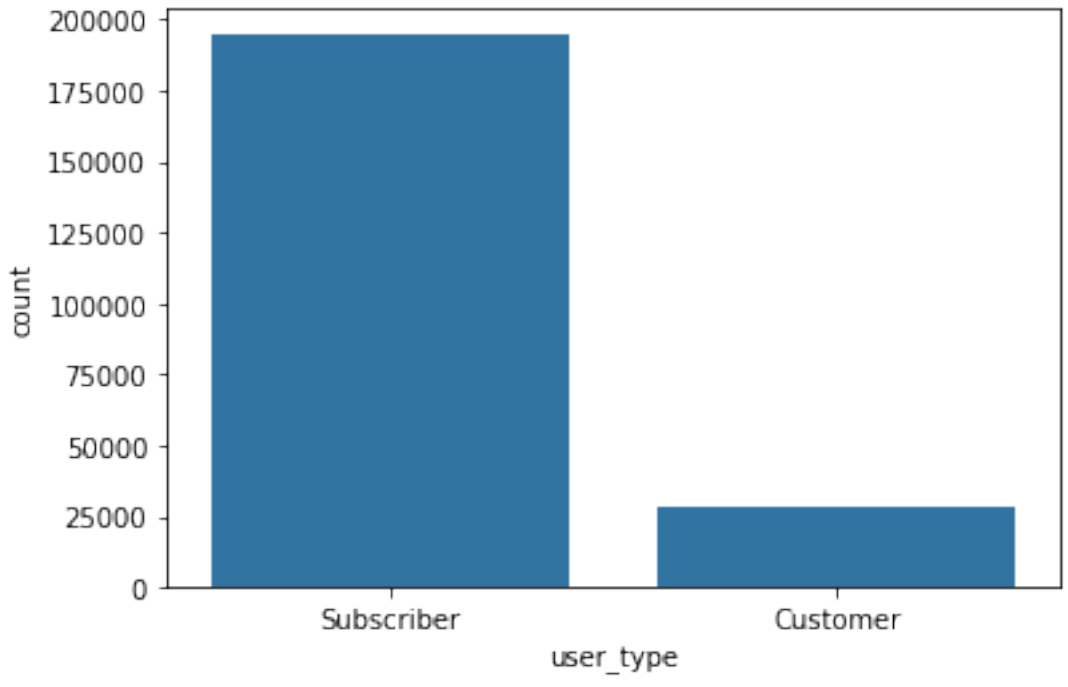
In [50]:


40000
35000
30000
25000
20000
15000
10000
5000
0
count
Mon
Tue
Wed
Thu
Fri
Sat
Sun
start_time_dayofweek

In [51]:

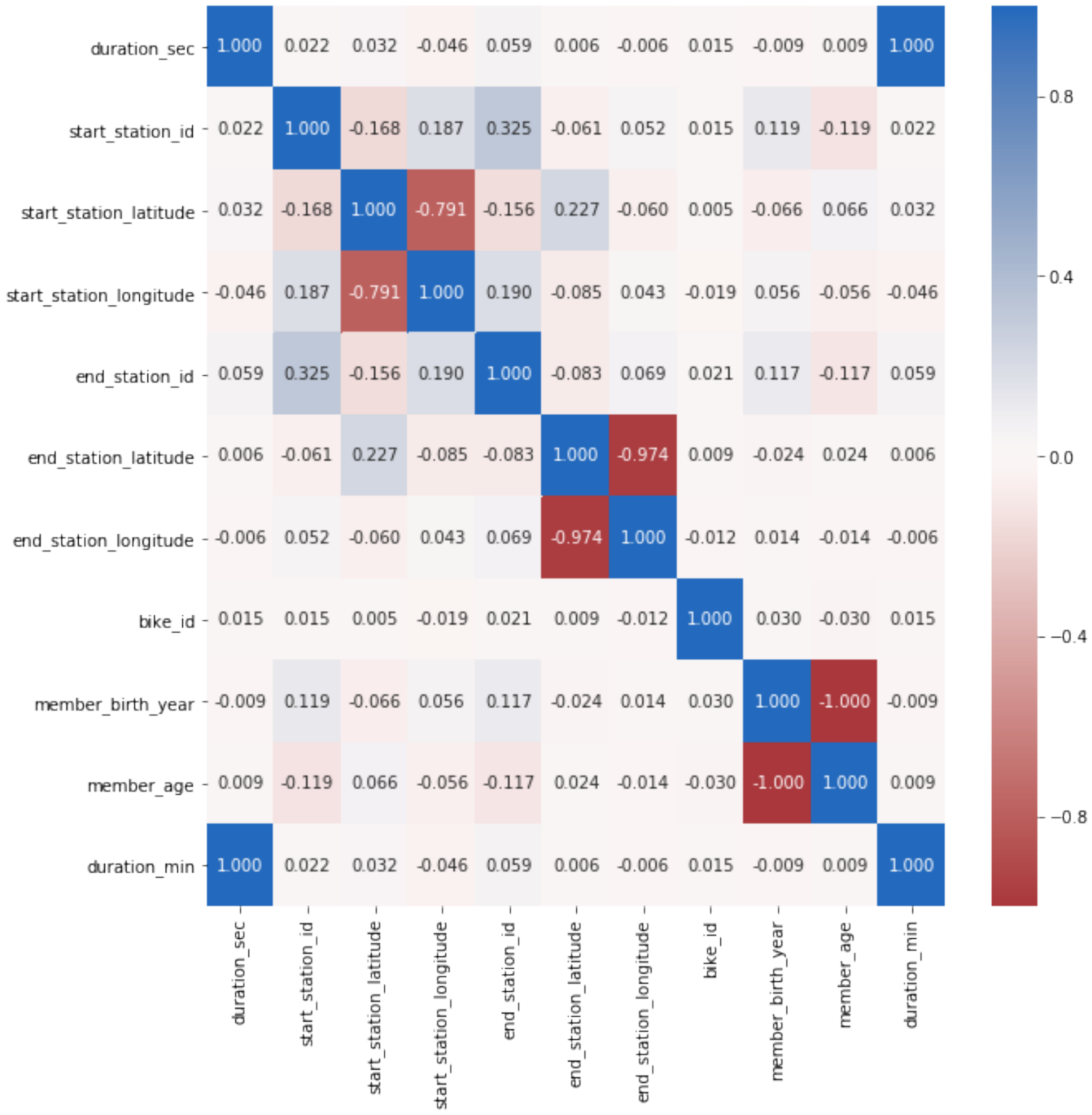

160000
140000
120000
100000
80000
60000
40000
20000
0
count
Male
Female
Other
member_gender

In [52]:


200000
175000
150000
125000
100000
75000
50000
25000
0
count
Subscriber
Customer
user_type

In [53]:

In [54]:



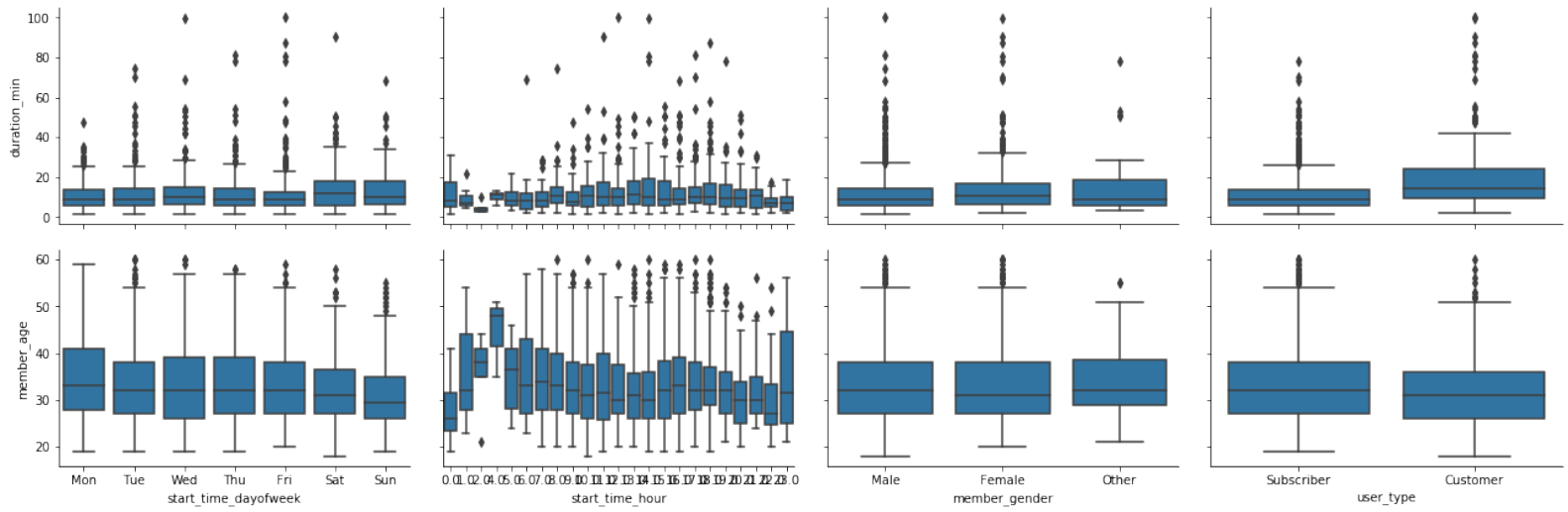
From the above plot we can see that there are few variables which are highly correlated to some other variables either having positive correlation or negative correlation.

In [55]:

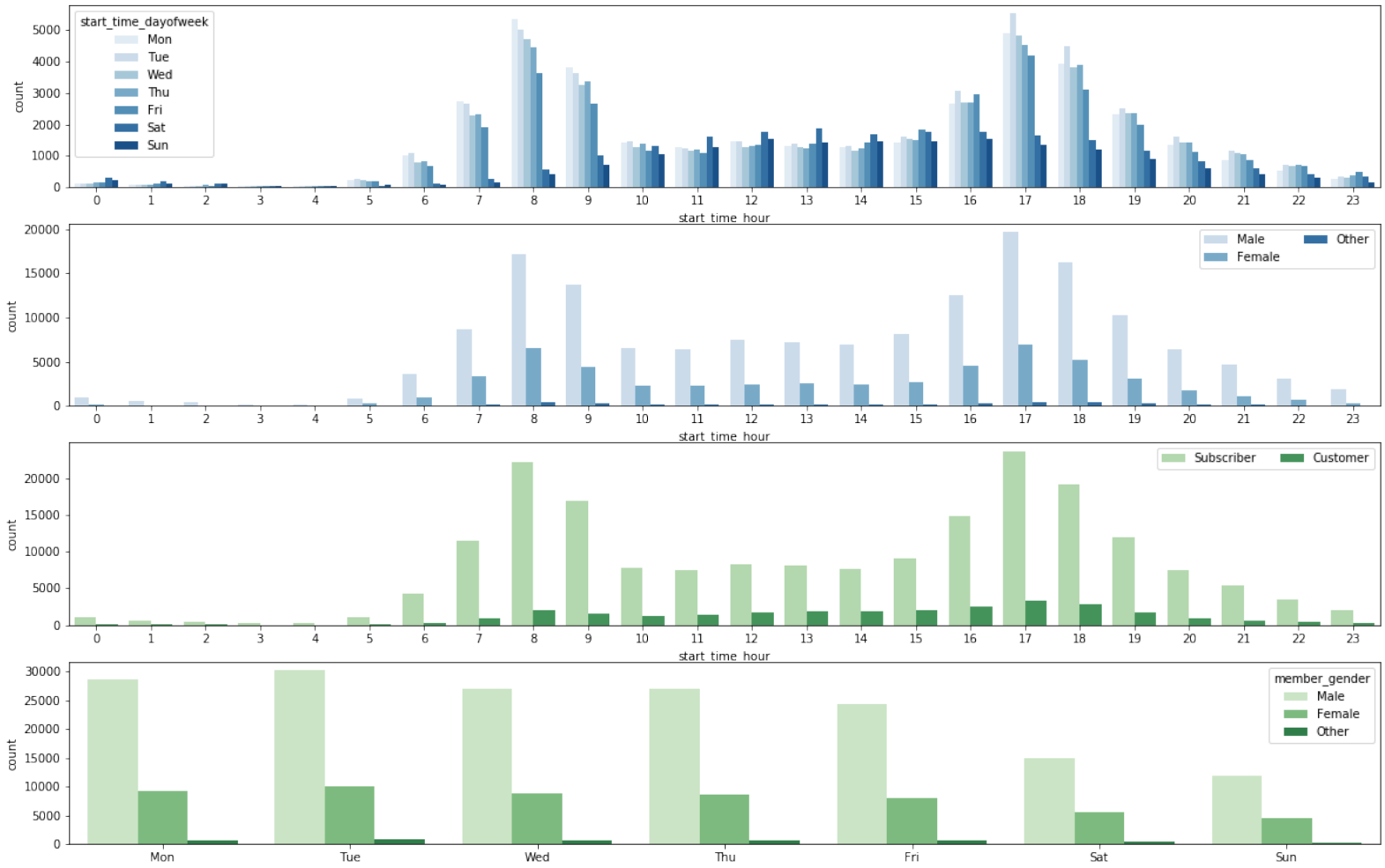
```
/Users/hamedbintalib/anaconda3/lib/python3.7/site-packages/pandas/core
/indexing.py:1494: FutureWarning:
Passing list-likes to .loc or [] with any missing label will raise
KeyError in the future, you can use .reindex() as an alternative.

See the documentation here:
https://pandas.pydata.org/pandas-docs/stable/indexing.html#deprecate-l
oc-reindex-listlike (https://pandas.pydata.org/pandas-docs/stable/inde
xing.html#deprecate-loc-reindex-listlike)
  return self._getitem_tuple(key)
/Users/hamedbintalib/anaconda3/lib/python3.7/site-packages/numpy/lib/h
istograms.py:824: RuntimeWarning: invalid value encountered in greater
_equal
  keep = (tmp_a >= first_edge)
/Users/hamedbintalib/anaconda3/lib/python3.7/site-packages/numpy/lib/h
istograms.py:825: RuntimeWarning: invalid value encountered in less_eq
ual
  keep &= (tmp_a <= last_edge)
```

In [56]:

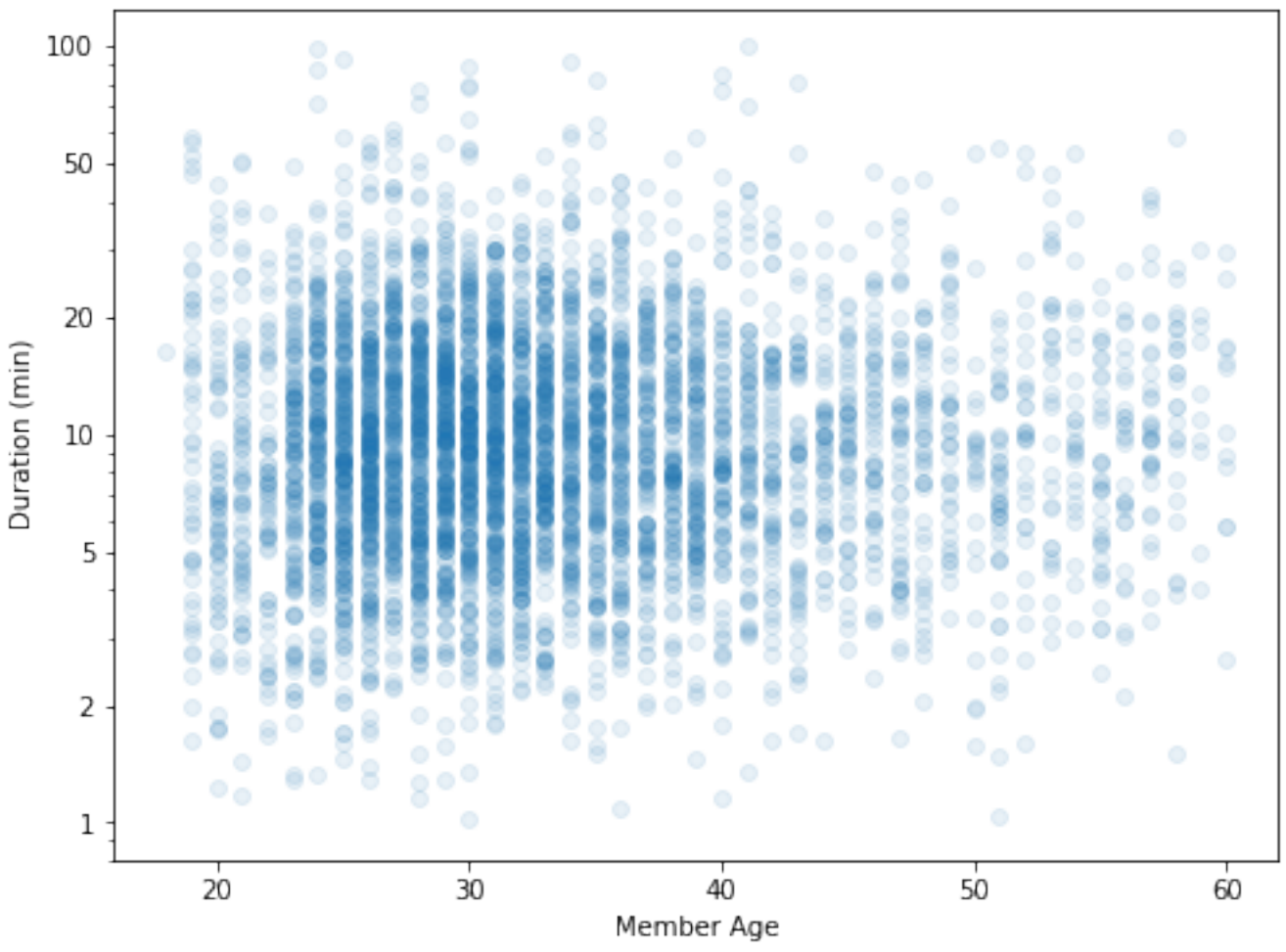
```
<Figure size 1080x1080 with 0 Axes>
```



In [57]:

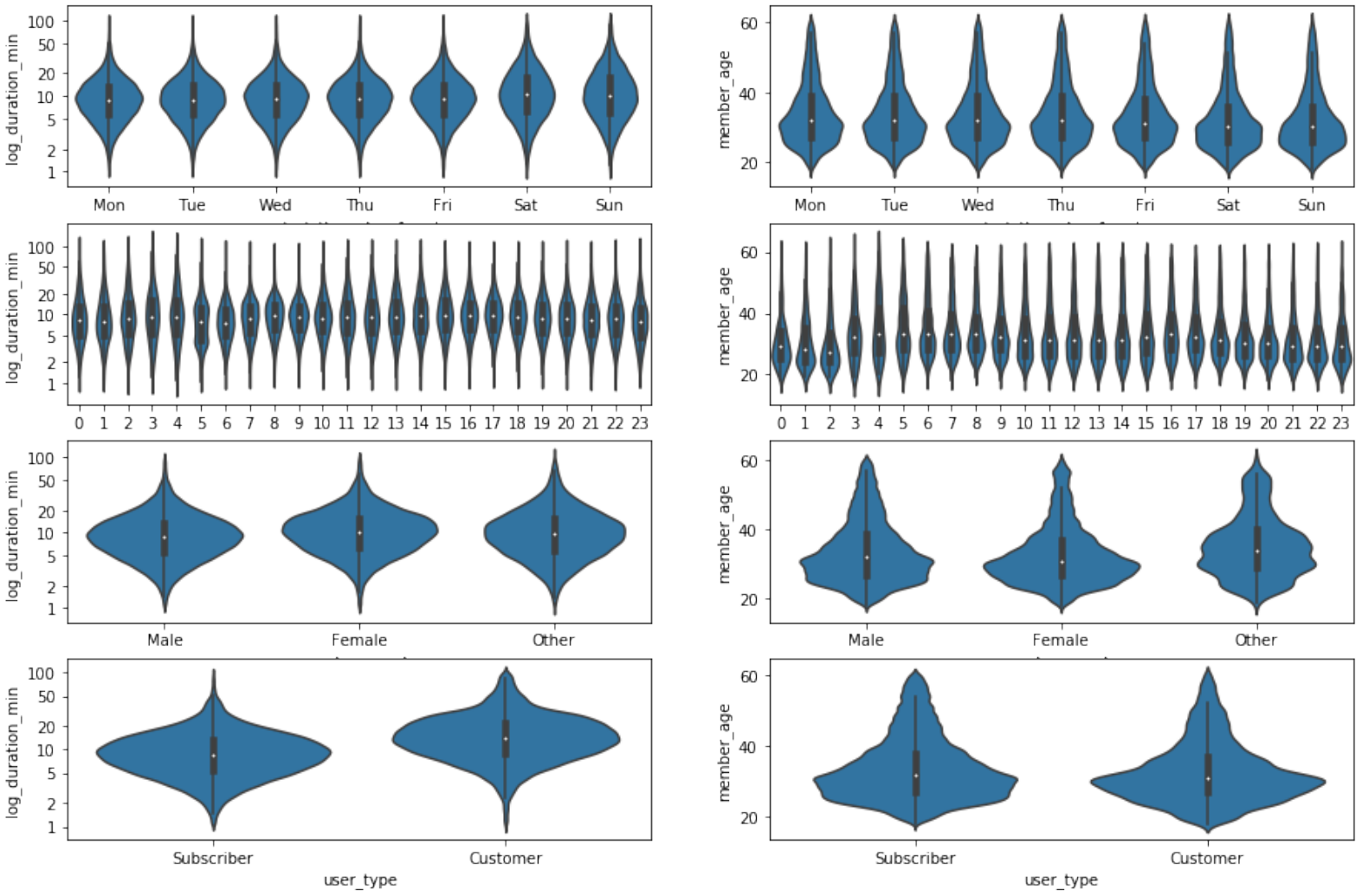

start_time_dayofweek
Mon
Tue
Wed
Thu
Fri
Sat
Sun
count
start_time_hour
Male
Female
Other
Subscriber
Customer
member_gender
Mon
Tue
Wed
Thu
Fri
Sat
Sun
user_type
Subscriber
Customer
start_time_dayofweek
Male
Female
Other
member_gender

In [58]:



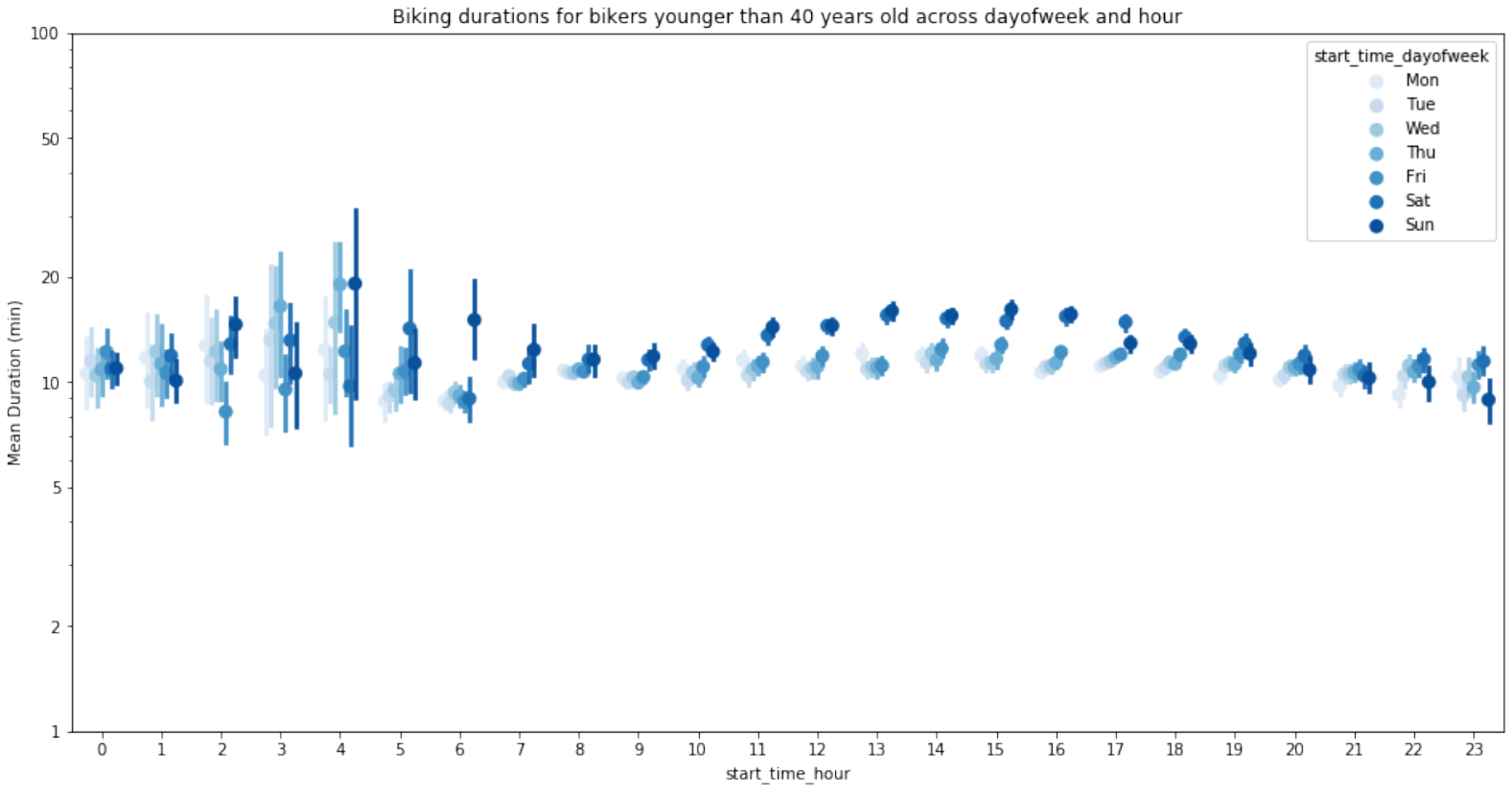
In [59]:

In [60]:

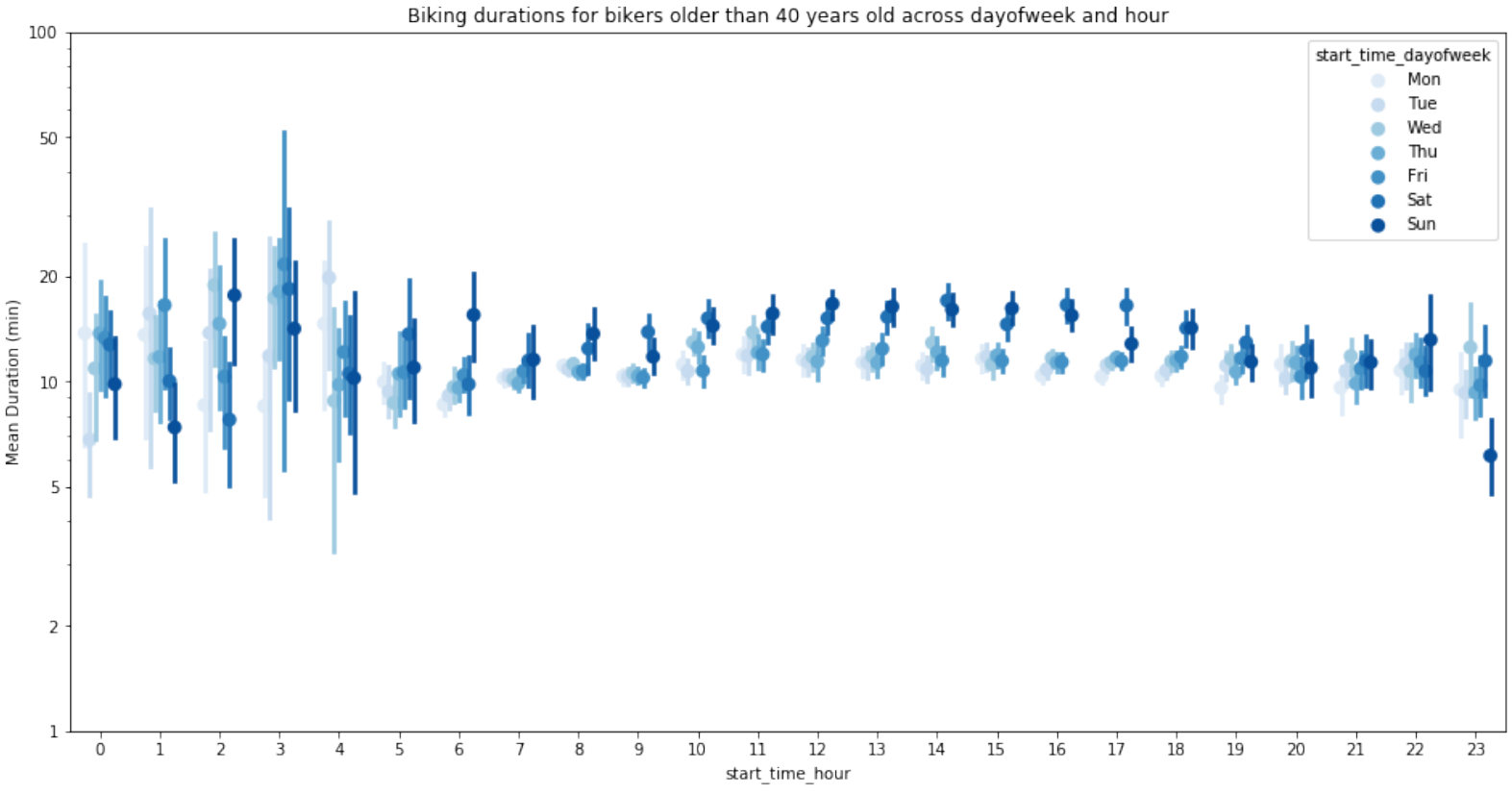


In [63]:

In [64]:



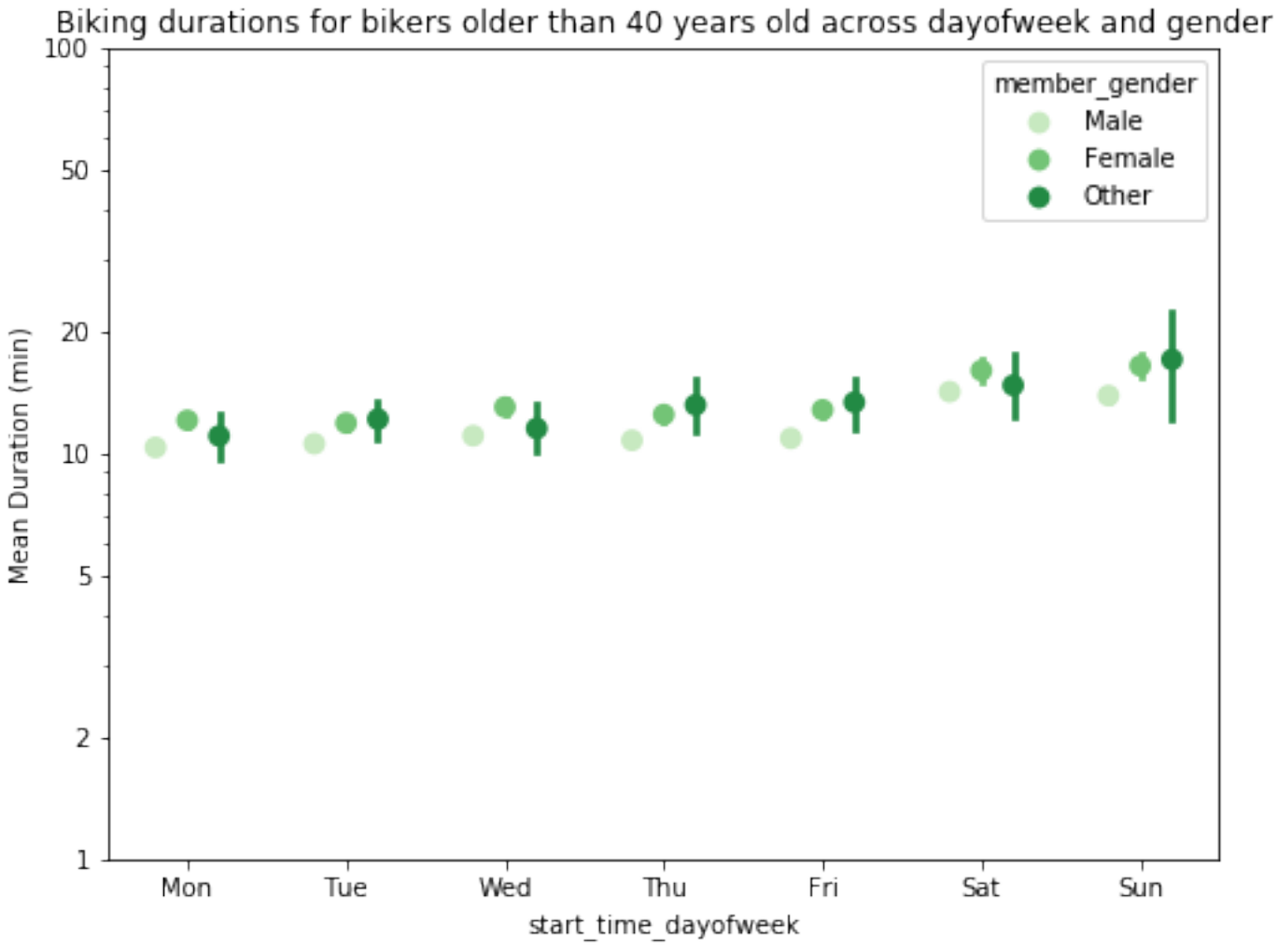
In [65]:


Biking durations for bikers older than 40 years old across dayofweek and hour
Mean Duration (min)
start_time_hour
start_time_dayofweek
Mon
Tue
Wed
Thu
Fri
Sat
Sun

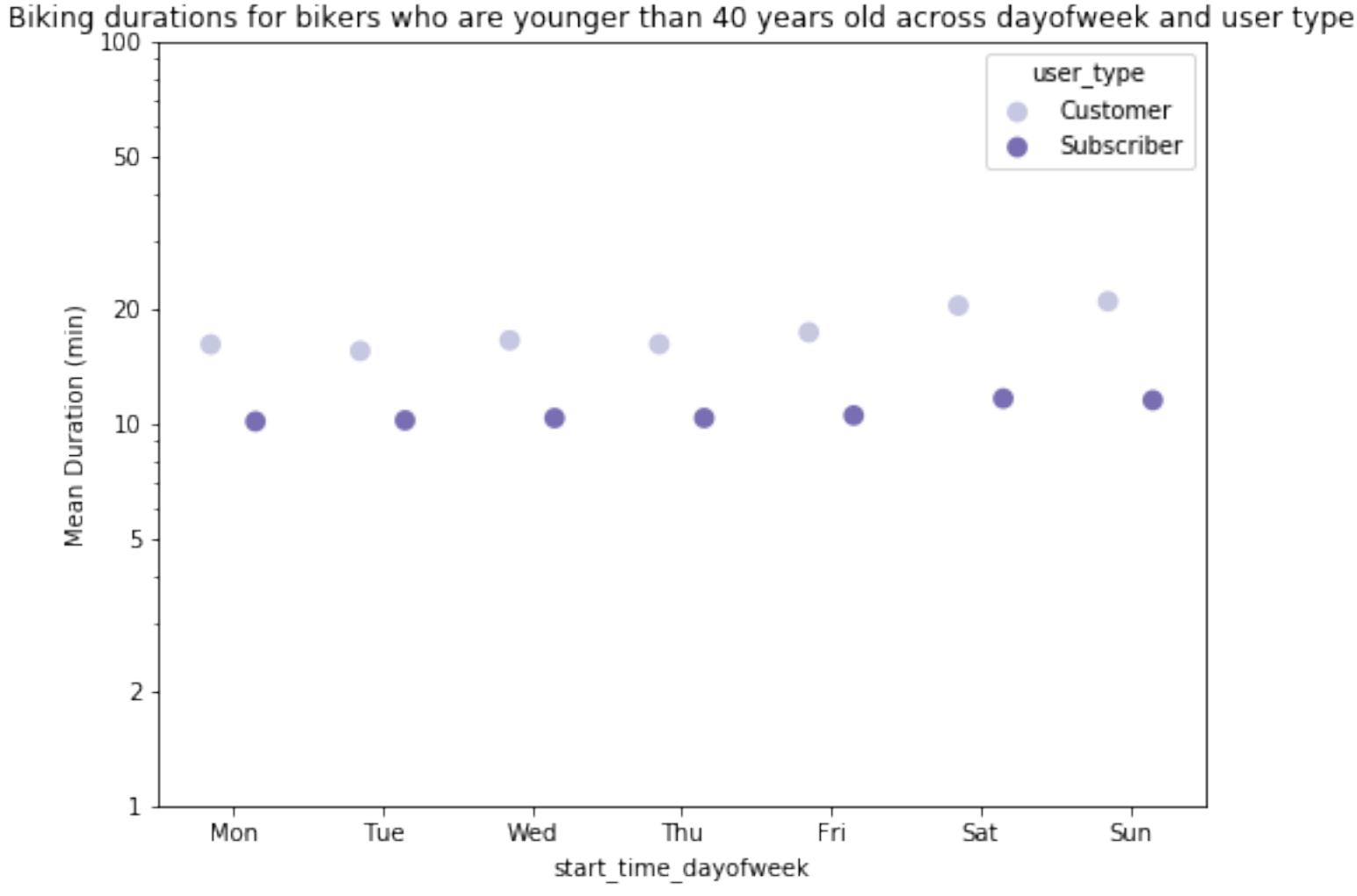
In [66]:

Biking durations for bikers younger than 40 years old across dayofweek and gender
Mean Duration (min)
start_time_dayofweek
member_gender
Male
Female
Other
Mon
Tue
Wed
Thu
Fri
Sat
Sun

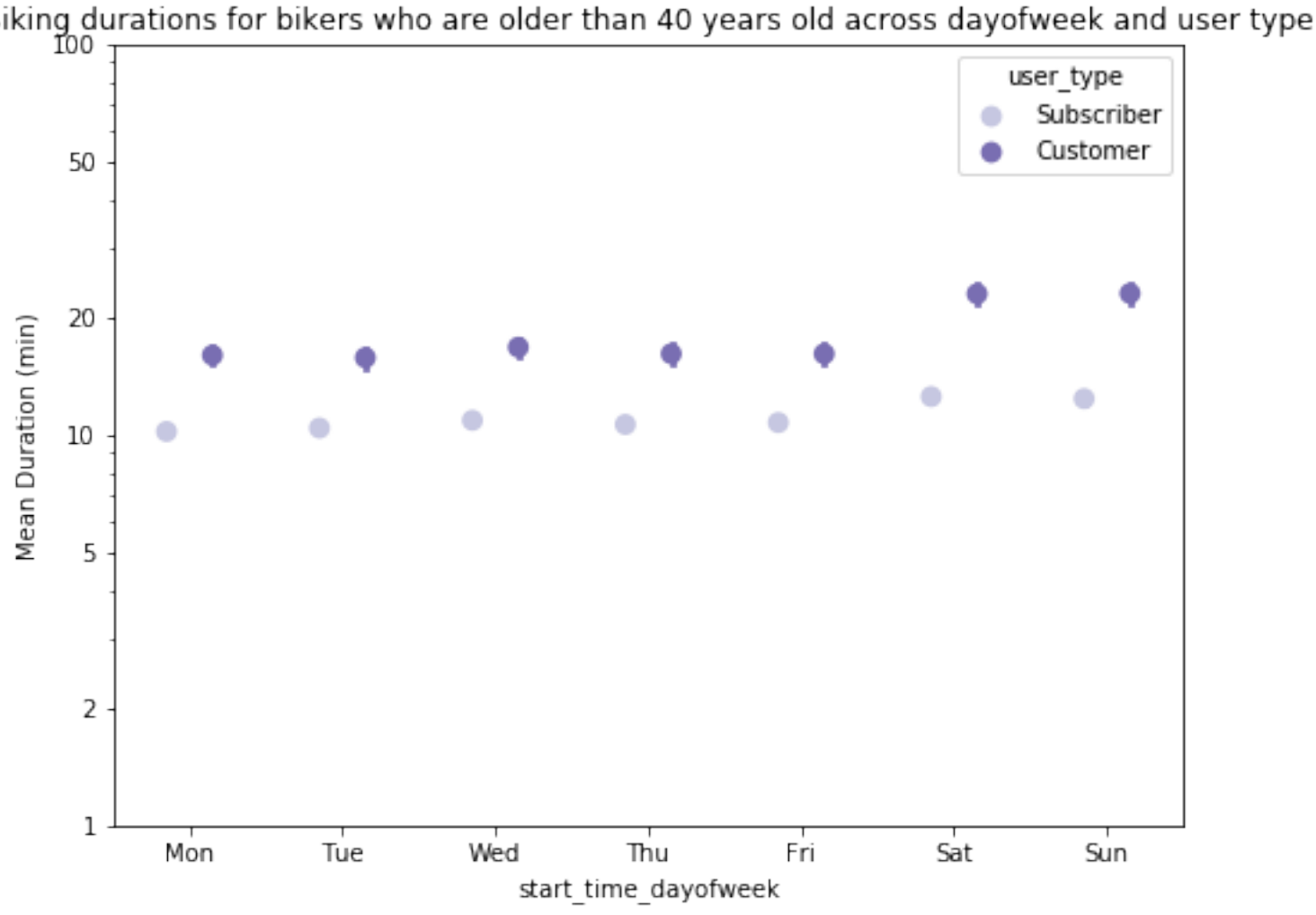
In [67]:


Biking durations for bikers older than 40 years old across dayofweek and gender
member_gender
Male
Female
Other
Mean Duration (min)
100
50
20
10
5
2
1
Mon
Tue
Wed
Thu
Fri
Sat
Sun
start_time_dayofweek

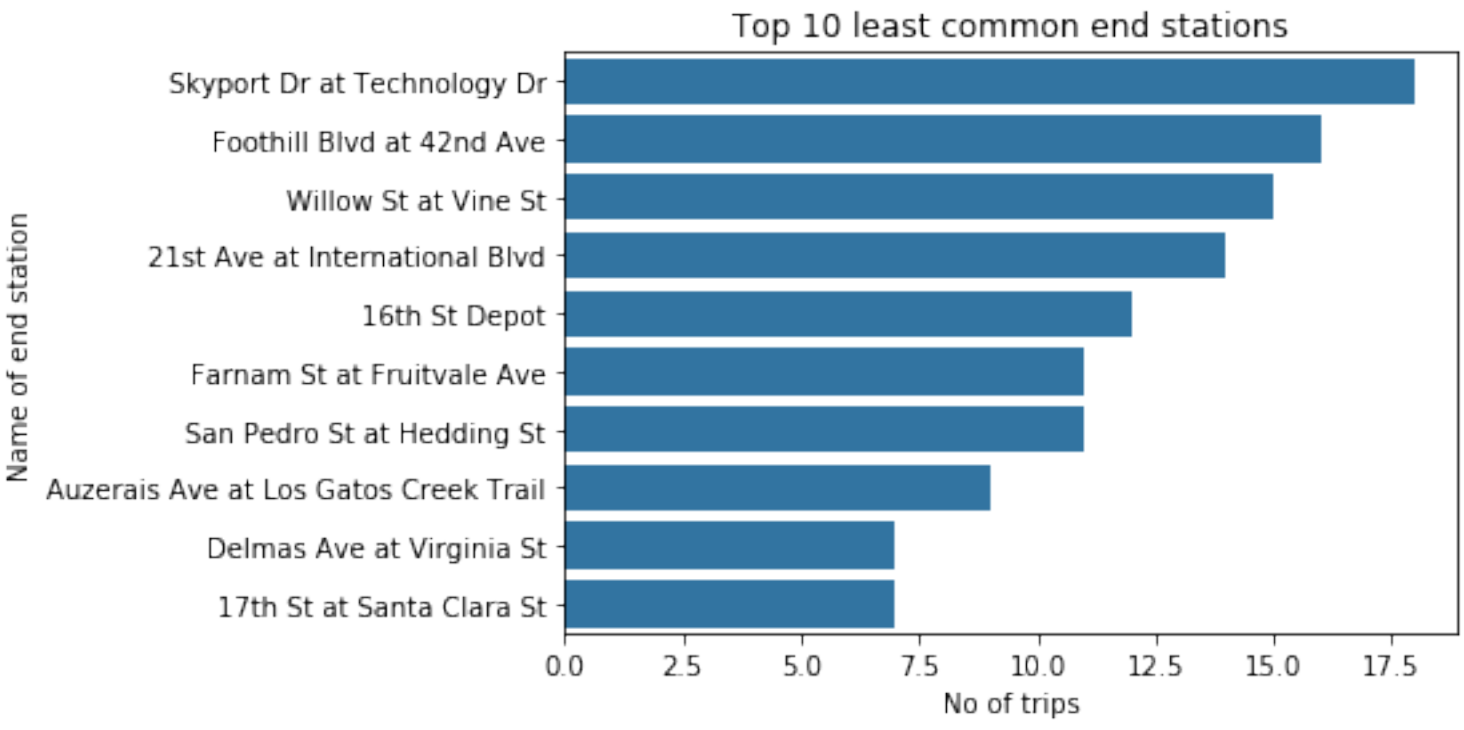
In [68]:


Biking durations for bikers who are younger than 40 years old across dayofweek and user type
user_type
Customer
Subscriber
Mean Duration (min)
100
50
20
10
5
2
1
Mon
Tue
Wed
Thu
Fri
Sat
Sun
start_time_dayofweek

In [69]:

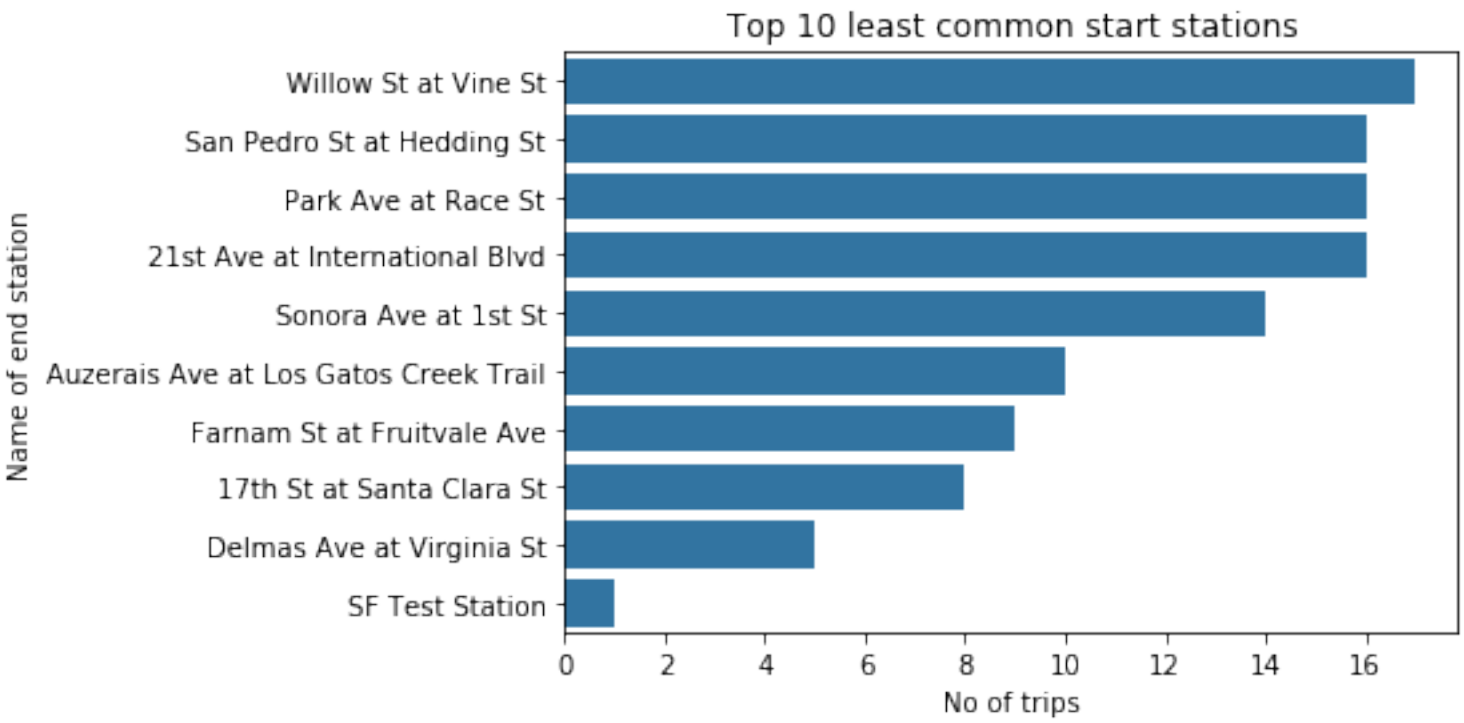


In [70]:



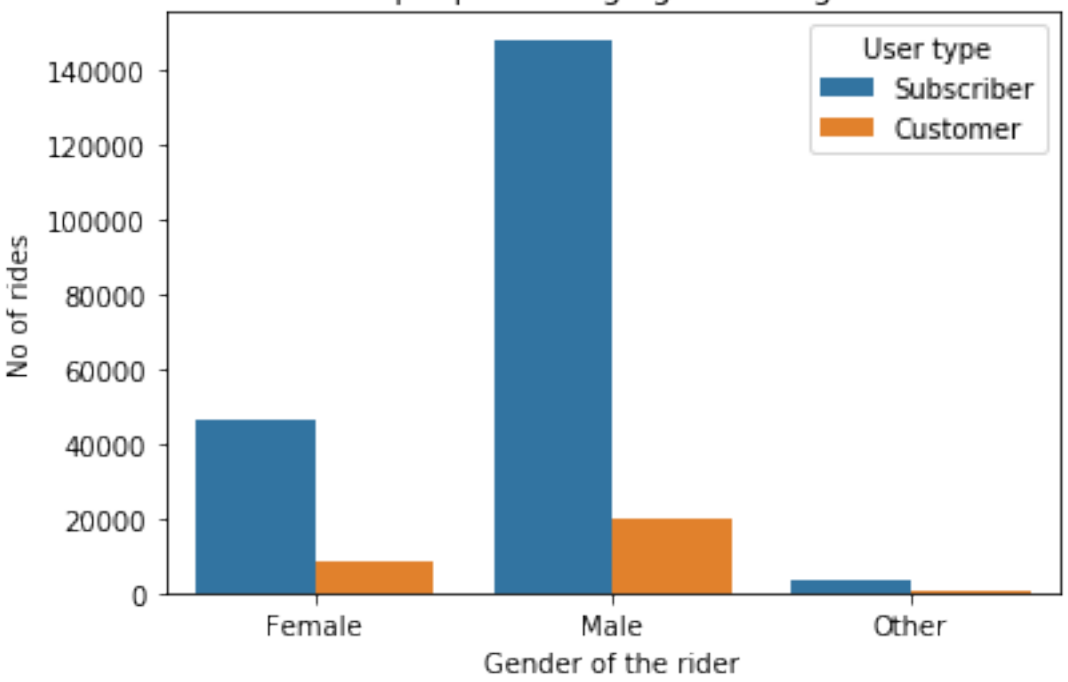
based on the graph above we can see the top 10 names of the end stationss where most bike stop

In [71]:



based on the graph above we can see the top 10 names of the start stationss where most bike start

In [72]:



We can see that most of the rides are. done by males

In [73]:

Out[73]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

5 rows × 21 columns

In [74]:

Out[74]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.7604
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.8693
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.8007
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.7810
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.7896

5 rows × 21 columns

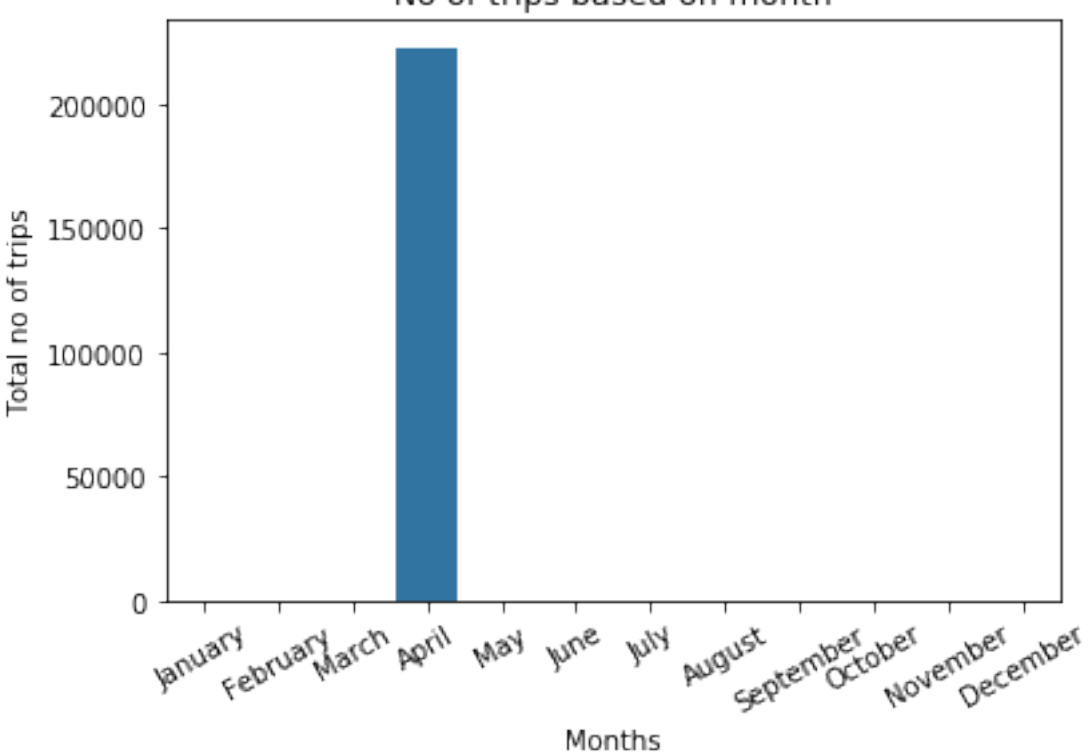
In [75]:

In [76]:

Out[76]:

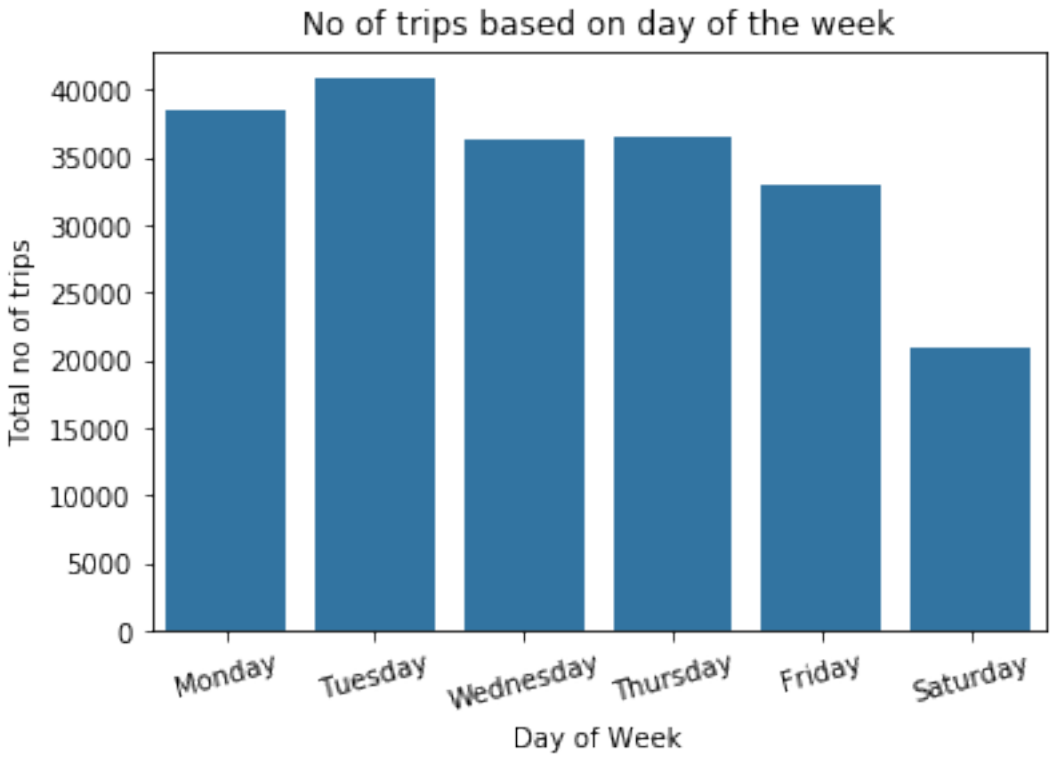
	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_s
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.760447	
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.869360	
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.800754	
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.781074	
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.789625	

In [77]:



The above graph shows that the given dataset contains the. data only for the moth of april

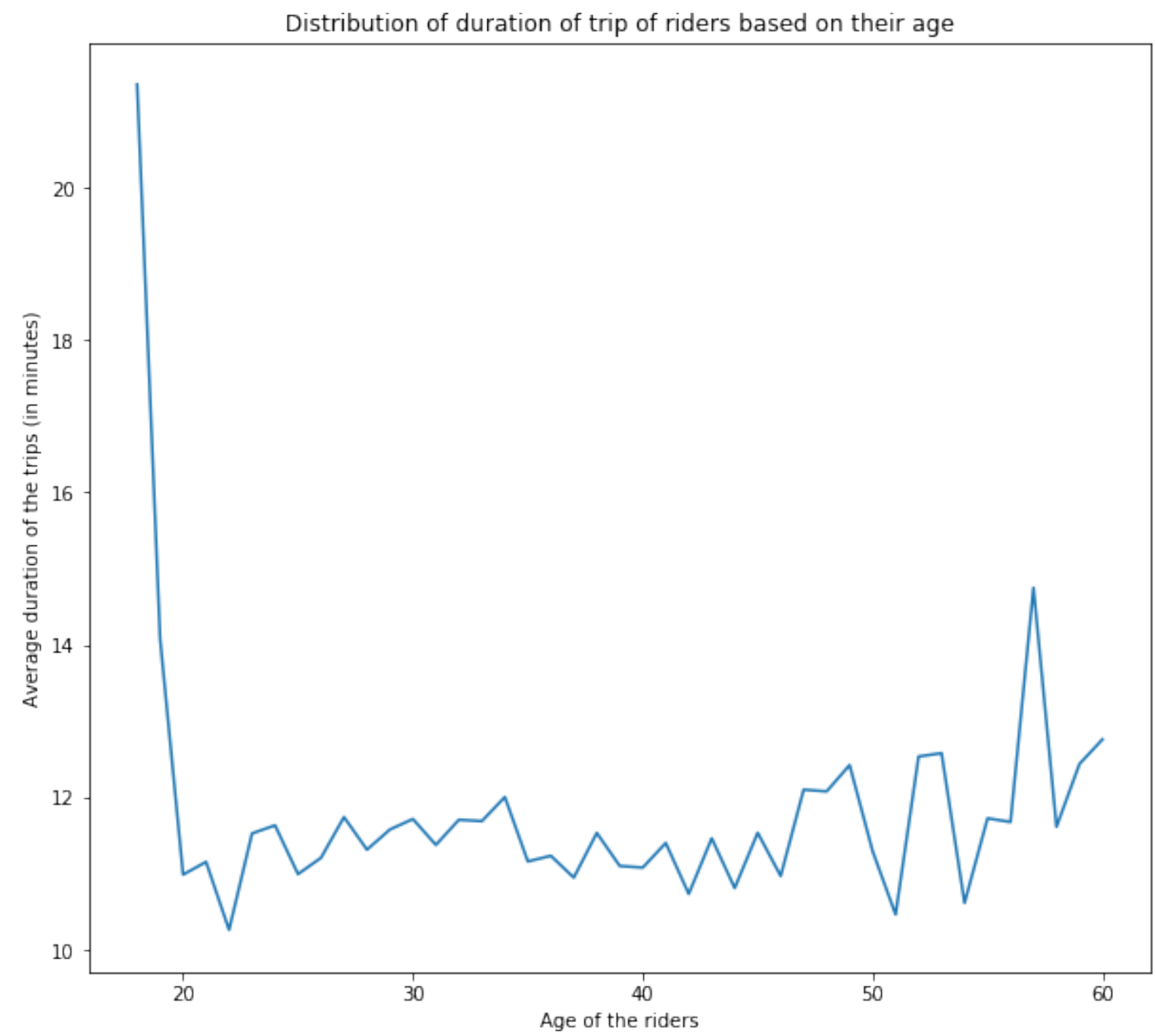
In [78]:



The above graph shows that most of the rides were started on Tuesday and the least on Saturday

In [79]:

In [80]:



We can see that the highest duration of trips is for the age groups around 20, aslo it show. there is no one older than 60 is riding bike

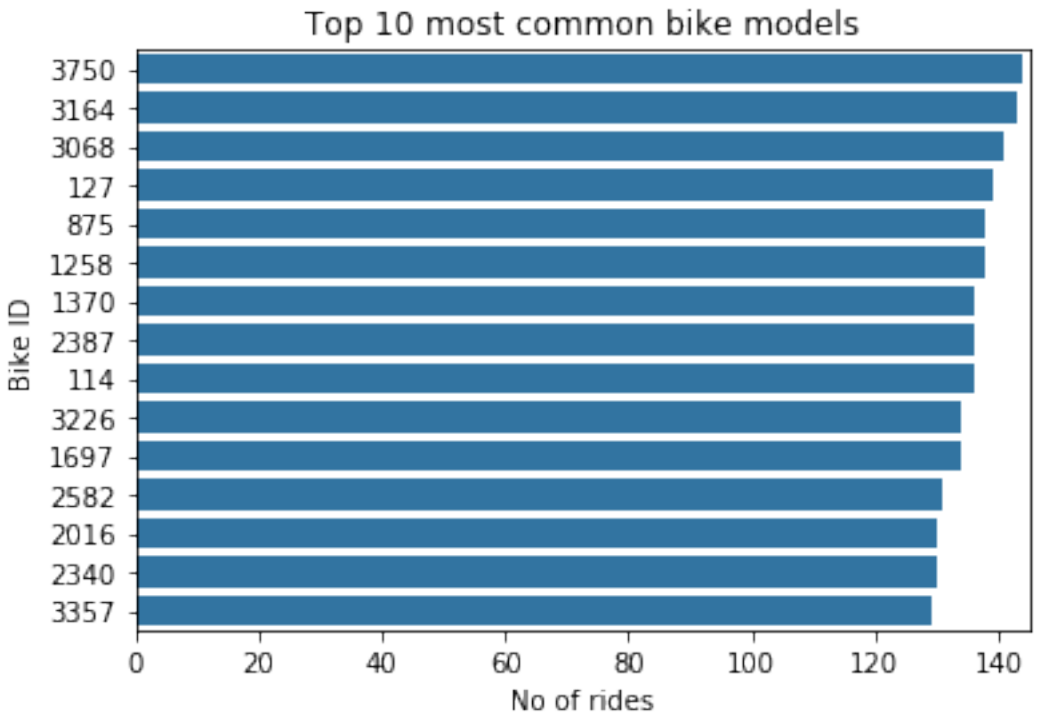
In [81]:

Out[81]:

duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	st

0 rows × 27 columns

In [82]:



based on the graph. above we can see that bike model 3750 is the most used

In [83]:

Out[83]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitu
4	1128	2019-04-30 23:59:04.739	2019-05-01 00:17:53.091	124.0	19th St at Florida St	37.760
5	1388	2019-04-30 23:53:05.982	2019-05-01 00:16:14.313	243.0	Bancroft Way at College Ave	37.869
6	920	2019-04-30 23:57:56.340	2019-05-01 00:13:16.454	202.0	Washington St at 8th St	37.800
7	725	2019-04-30 23:56:11.219	2019-05-01 00:08:16.915	44.0	Civic Center/UN Plaza BART Station (Market St ...	37.781
8	488	2019-04-30 23:59:00.660	2019-05-01 00:07:08.975	21.0	Montgomery St BART Station (Market St at 2nd St)	37.789

5 rows × 27 columns

In [ ]: