Part I - Ford GoBike Data Set Exploration

by Shanshan Chu

Introduction

> This data set includes information about individual rides made in a bike-sharing system covering the greater San Francisco Bay area. The data includes information about each trip for Ford GoBike (e.g. the start and end time, the start and ending station) and user information (e.g. user's subscription status and ages). The data exploration of this data set will give us some insights about the bike-sharing market in the greater San Francisco Bay area.

Preliminary Wrangling

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

%matplotlib inline
```

Gather Data

In [2]:
```
# Load data
df = pd.read_csv('201902-fordgobike-tripdata.csv')
# Check data
df.head()
```

Out[2]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_s
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)	
1	42521	2019-02-28 18:53:21.7890	2019-03-01 06:42:03.0560	23.0	The Embarcadero at Steuart St	
2	61854	2019-02-28 12:13:13.2180	2019-03-01 05:24:08.1460	86.0	Market St at Dolores St	
3	36490	2019-02-28 17:54:26.0100	2019-03-01 04:02:36.8420	375.0	Grove St at Masonic Ave	
4	1585	2019-02-28 23:54:18.5490	2019-03-01 00:20:44.0740	7.0	Frank H Ogawa Plaza	

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 183412 entries, 0 to 183411
Data columns (total 16 columns):
duration_sec               183412 non-null int64
start_time                 183412 non-null object
end_time                   183412 non-null object
start_station_id           183215 non-null float64
start_station_name         183215 non-null object
start_station_latitude     183412 non-null float64
start_station_longitude    183412 non-null float64
end_station_id             183215 non-null float64
end_station_name           183215 non-null object
end_station_latitude       183412 non-null float64
end_station_longitude      183412 non-null float64
bike_id                    183412 non-null int64
user_type                  183412 non-null object
member_birth_year          175147 non-null float64
member_gender              175147 non-null object
bike_share_for_all_trip    183412 non-null object
dtypes: float64(7), int64(2), object(7)
memory usage: 22.4+ MB
```

What is the structure of your dataset?

> This is a dataset contains of 182312 trips for the shared-bikes. There are 16 features in the dataset. Some of the values are missing in this dataset (e.g. the birth year and the gender of the members).

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

> I would like to know more time and location related information for the collected trips, as well as the user information. I plan to answer the following questions which are related to user information, time and location related information.

Univariate Questions:

> (Time related questions)
>
> 1. How is the bike usage distributed in a day (24 hours)?
>
> 2. How is the bike usage distributed in a week (7 days)?
>
> 3. How long was the longest-time trip?
>
> (Location related questions)
>
> 4. Which stations are the top 3 stations as the start station?
>
> 5. Which stations are the top 3 stations as the end station?
>
> (User related questions)
>
> 6. What is the user's age distribution?
>
> 7. Do most of the users subsribe as members?
>
> 8. Does users' gender have influence on the bike usage? (bar plot)
>
> 9. Does users' gender have influence on the bike usage? (pie plot, to show the percentage)

Bivariate Questions:

> (User and time related questions)
>
> 10. For different user types (substribed or not), what are their duration distribution?
>
> 11. For different user genders, what are their duration distribution?
>
> 12. For different user ages, what is the user type (subsribed or not)?
>
> 13. For different user ages, how is the bike usage distributed in a week (7 days)?

Multivariate Questions:

> (User and time related questions)
>
> 14. For different user types, how is the hourly trip during a week (7 days) look like?
>
> 15. For different user types and genders, what does the trip duration in minutes look like?

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

> Based on my proposed questions, the features that I think would be helpful are time and location related information for the collected trips, and the user information.

Access data

In [4]:
```
# Check duplicated data
df.duplicated().sum()
```

Out[4]:
```
0
```

In [5]:
```
# Check missing data
df.isna().sum()
```

Out[5]:
```
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 [6]:
```
# Check data types
df.dtypes
```

Out[6]:
```
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
```

Clean Data

In this section, I will clean the data using the following definition.

Definition

Drop irrelavant rows and columns:

> 1. Drop rows and columns that are not relavant to my proposed questions (e.g. station latitude and longtitude)

> 2. Drop rows that contains missing values

Change data type:

> 3. Change station names to string type

> 4. Change genders to string type

> 5. Change user types to string type

> 6. Change duration time (start and end time) to datatime format

> 7. Change birth year to int type

Add new rows and columns:

> 8. Add new columns (e.g. age, days of week, hours, and duration in minutes) and ensure that they are in good data types

Code

In [7]:
```
# Make a cope of the original data set at first
df_clean=df.copy()
```

In [8]:
```
# 1. Drop rows and columns that are not relavant to my proposed questions (e
df_clean.drop(['start_station_latitude', 'start_station_longitude', 'start_s
```

In [9]:
```
# 2. Drop rows that contains missing values
df_clean.dropna(inplace=True)
```

In [10]:
```
# 3. Change station names to string type
df_clean['start_station_name'] = df_clean['start_station_name'].astype(str)
df_clean['end_station_name'] = df_clean['end_station_name'].astype(str)
```

In [11]:
```
# 4. Change genders to string type
df_clean['member_gender'] = df_clean['member_gender'].astype('category')
```

In [12]:
```
# 5. Change user types to string type
df_clean['user_type'] = df_clean['user_type'].astype('category')
```

In [13]:
```
# 6. Change duration time (start and end time) to datatime format
df_clean['start_time'] = pd.to_datetime(df_clean['start_time'])
df_clean['end_time'] = pd.to_datetime(df_clean['end_time'])
```

In [14]:
```
# 7. Change birth year to int type
df_clean['member_birth_year'] = df_clean['member_birth_year'].astype(int)
```

In [15]:
```
# 8.Add new columns (e.g. age, days of week, hours, and duration in minutes)
```

In [16]:
```
# Age
df_clean['member_age'] = 2019 - df_clean['member_birth_year']
```

In [17]:
```
# Days of week
df_clean['weekday'] = df_clean[['start_time']].apply(lambda x: dt.datetime.s
```

In [18]:
```
# Hours
df_clean['start_time_hour'] = df_clean['start_time'].dt.hour
```

In [19]:
```
# Duration in minutes
df_clean['duration_minute'] = df_clean['duration_sec']/60
# Drop the duration in seconds
df_clean.drop(['duration_sec'], axis=1, inplace=True)
```

In [20]:
```
# And special treatmeant for members' ages for later visulization
df_clean['member_age'].describe()
```

Out[20]:
```
count    174952.000000
mean         34.196865
std          10.118731
min          18.000000
25%          27.000000
50%          32.000000
75%          39.000000
max         141.000000
Name: member_age, dtype: float64
```

> Notice there is one extreme who is 141 years old. My guess is that the 141 is an inccorrect value, and we can only drop this value. Moreover, more than 75% of the members are younger than 40, so I think we can drop more. Here I picked 60 years old as the threshold.

In [21]:
```
df_clean = df_clean[df_clean['member_age'] <=60]
df_clean['age_bins'] = pd.cut(x=df_clean['member_age'], bins=[10, 20, 30, 40
```

In [22]:
```
bins = [10,20,30,40,50,60]
labels=['kids','young adult','middle-aged adult','old-aged adults','senior']
df_clean['bins'] = pd.cut(df_clean['member_age'], bins=bins, labels=labels)
```

In [23]:
```
# 8. Ensure the new columns are in good data types
```

In [24]:
```
# Age
df_clean['member_age'] = df_clean['member_age'].astype(int)
```

In [25]:
```
# Weekdays
df_clean['weekday'] = df_clean['weekday'].astype(str)
```

In [26]:
```
# start and end time
df_clean['start_time_hour'] = df_clean['start_time'].dt.hour
df_clean['start_time_hour'] = df_clean['start_time_hour'].astype(int)
df_clean['end_time_hour'] = df_clean['end_time'].dt.hour
df_clean['end_time_hour'] = df_clean['end_time_hour'].astype(int)
```

Test

In [27]:
```
df_clean.head()
```

Out[27]:

	start_time	end_time	start_station_name	end_station_name	bike_id	user_type	r
0	2019-02-28 17:32:10.145	2019-03-01 08:01:55.975	Montgomery St BART Station (Market St at 2nd St)	Commercial St at Montgomery St	4902	Customer	
2	2019-02-28 12:13:13.218	2019-03-01 05:24:08.146	Market St at Dolores St	Powell St BART Station (Market St at 4th St)	5905	Customer	
3	2019-02-28 17:54:26.010	2019-03-01 04:02:36.842	Grove St at Masonic Ave	Central Ave at Fell St	6638	Subscriber	
4	2019-02-28 23:54:18.549	2019-03-01 00:20:44.074	Frank H Ogawa Plaza	10th Ave at E 15th St	4898	Subscriber	
5	2019-02-28 23:49:58.632	2019-03-01 00:19:51.760	4th St at Mission Bay Blvd S	Broadway at Kearny	5200	Subscriber	

In [28]:

```
# Check duplicated data
df_clean.duplicated().sum()
```

Out[28]:

```
0
```

In [29]:

```
# Check missing data
df.isna().sum()
```

Out[29]:

```
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 [31]:

```
# Check data information
df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 171422 entries, 0 to 183411
Data columns (total 16 columns):
start_time                 171422 non-null datetime64[ns]
end_time                   171422 non-null datetime64[ns]
start_station_name         171422 non-null object
end_station_name           171422 non-null object
bike_id                    171422 non-null int64
user_type                  171422 non-null category
member_birth_year          171422 non-null int64
member_gender              171422 non-null category
bike_share_for_all_trip    171422 non-null object
member_age                 171422 non-null int64
weekday                    171422 non-null object
start_time_hour            171422 non-null int64
duration_minute            171422 non-null float64
age_bins                   171422 non-null category
bins                       171422 non-null category
end_time_hour              171422 non-null int64
dtypes: category(4), datetime64[ns](2), float64(1), int64(5), object(4)
memory usage: 17.7+ MB
```

In [32]:
```
# Save the cleaned dataset as CSV file
df_clean.to_csv('GoBikeDataClean.csv')
```

Univariate Exploration

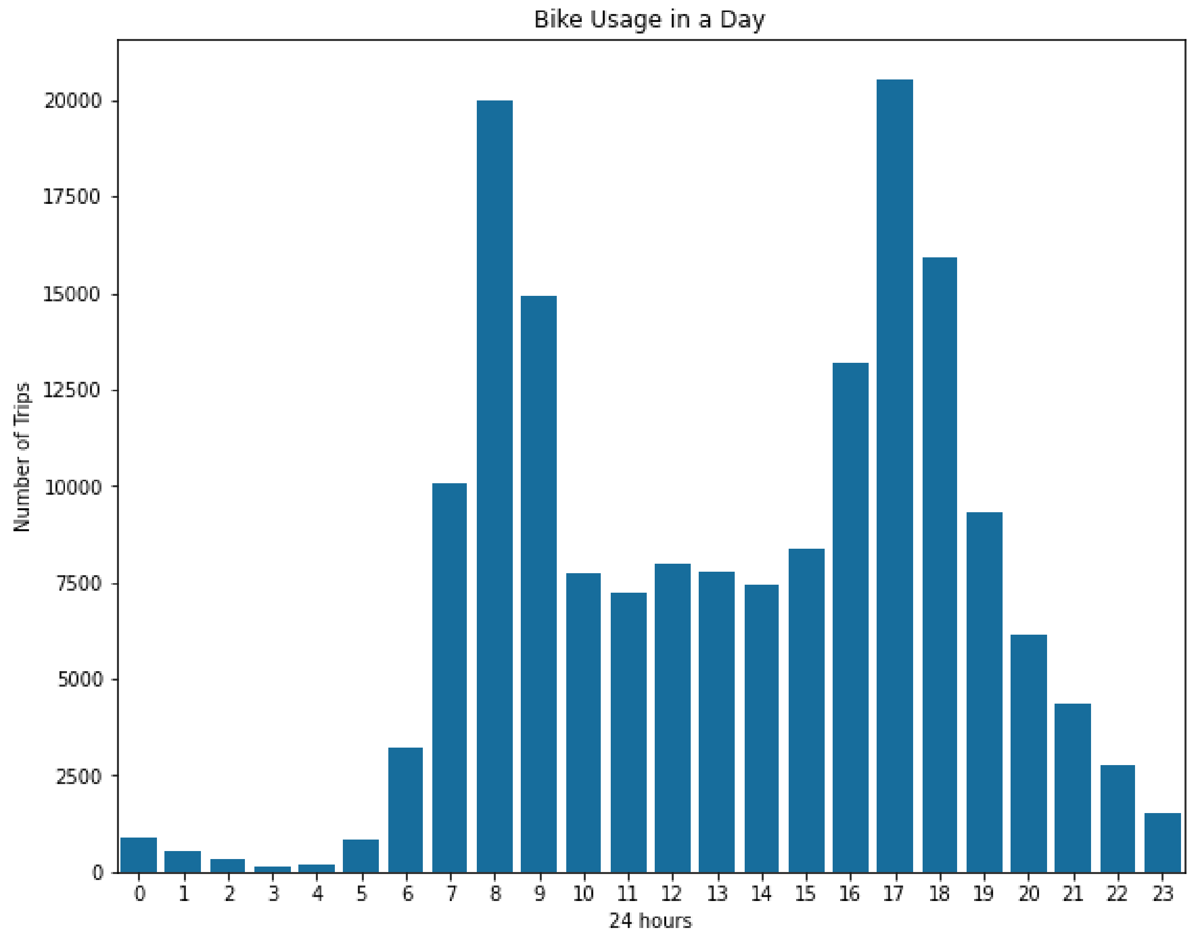
> In this section, I will investigate distributions of individual variables. I would like to know more time and location related information for the collected trips, as well as the user information.

Time related questions

1. How is the bike usage distributed in a day (24 hours)?

In [33]:
```
# Set color
color = sb.color_palette('colorblind')[0]
```

In [34]:
```
plt.figure(figsize=(10,8))
sb.countplot(data = df_clean, x='start_time_hour', color=color)
plt.title("Bike Usage in a Day")
plt.xlabel("24 hours")
plt.ylabel("Number of Trips")
plt.show();
```

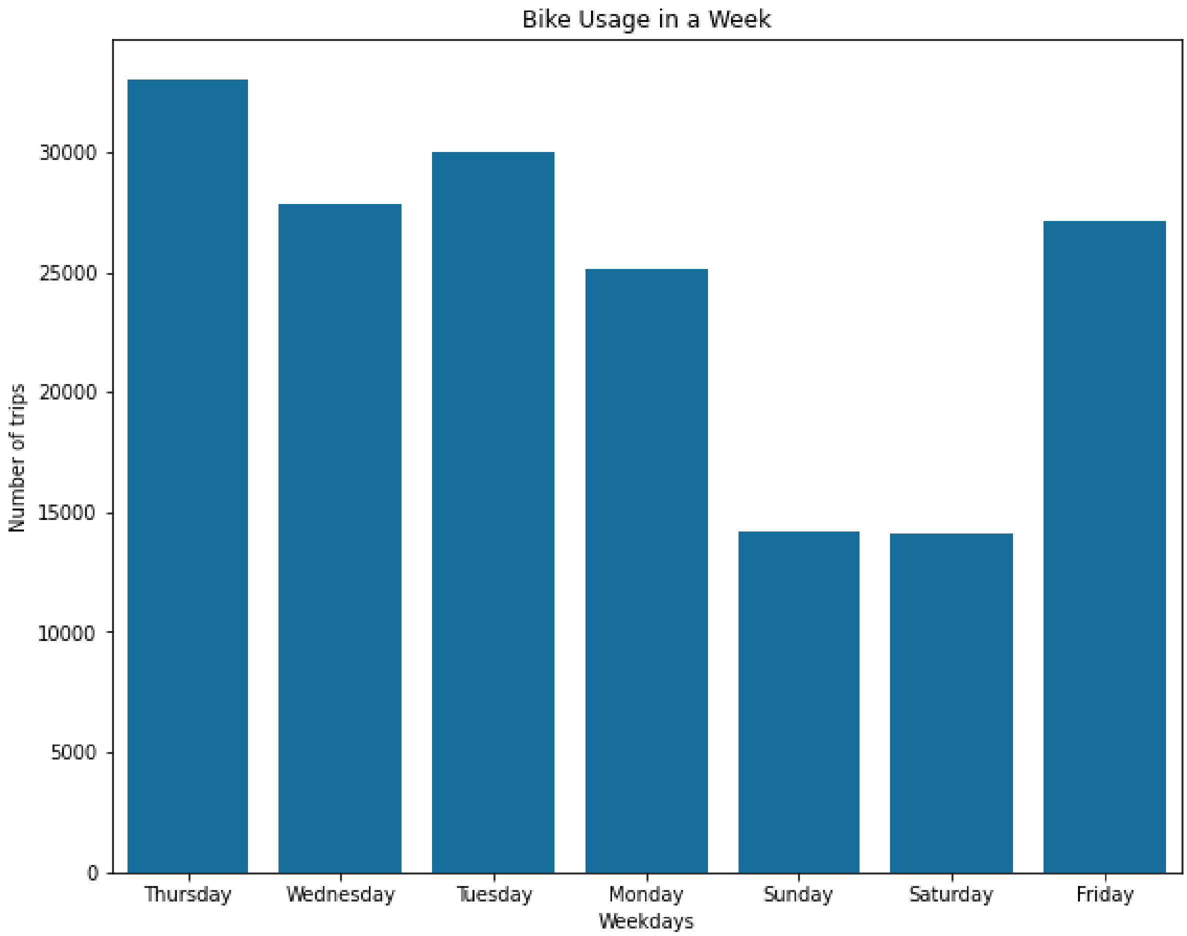


> Comment: See from the result, the time that bikes are mostly used are duing the rush hour for going to and go back from work/school.

2. How is the bike usage distributed in a week (7 days)?

In [35]:

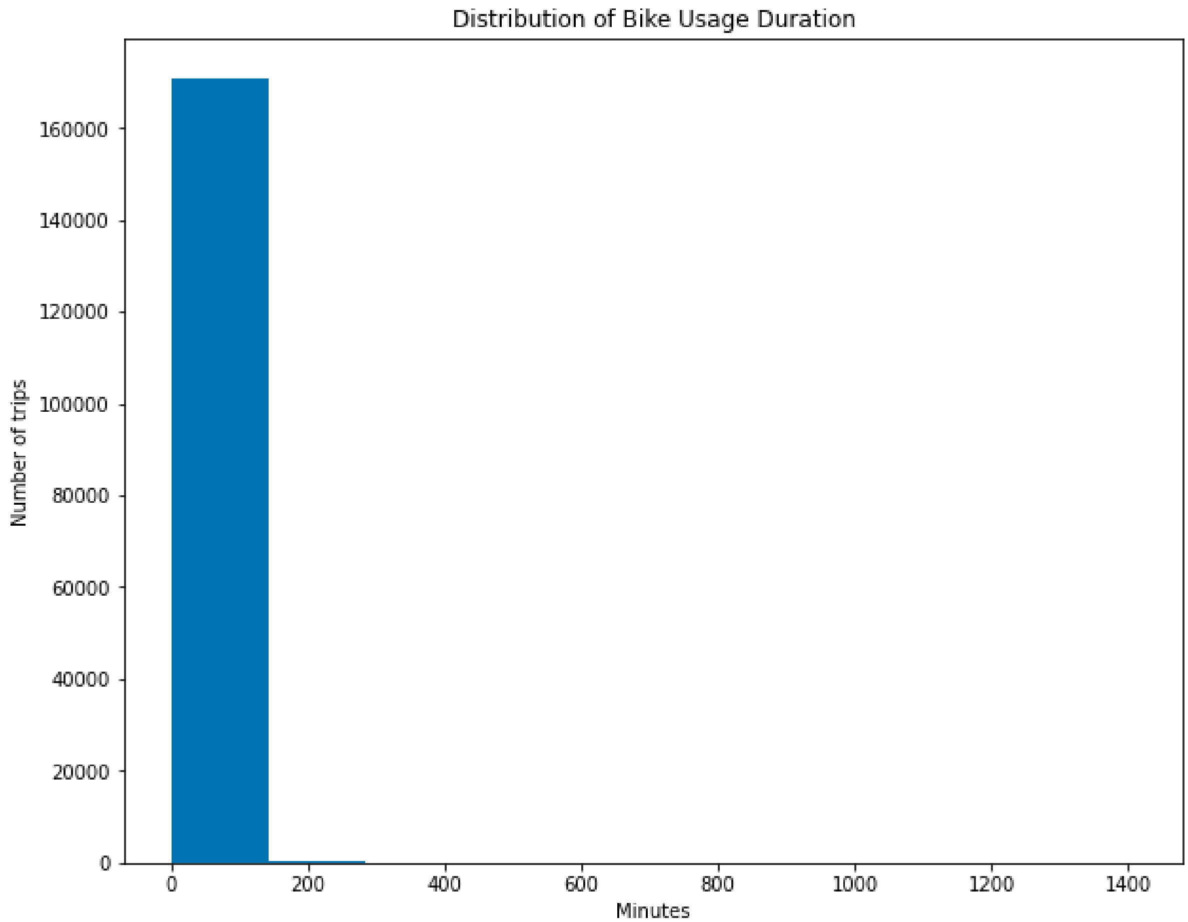
```
plt.figure(figsize=(10,8))
sb.countplot(x = 'weekday', data = df_clean, color=color)
plt.title('Bike Usage in a Week')
plt.xlabel('Weekdays')
plt.ylabel('Number of trips');
```



> Comment: See from the result, the usage on workdays is obviously much more than on weekend.

3. How long was the longest-time trip?

In [36]:
```
plt.figure(figsize=(10,8))
plt.hist(x = 'duration_minute', data = df_clean, color=color)
plt.title('Distribution of Bike Usage Duration')
plt.xlabel('Minutes')
plt.ylabel('Number of trips');
```

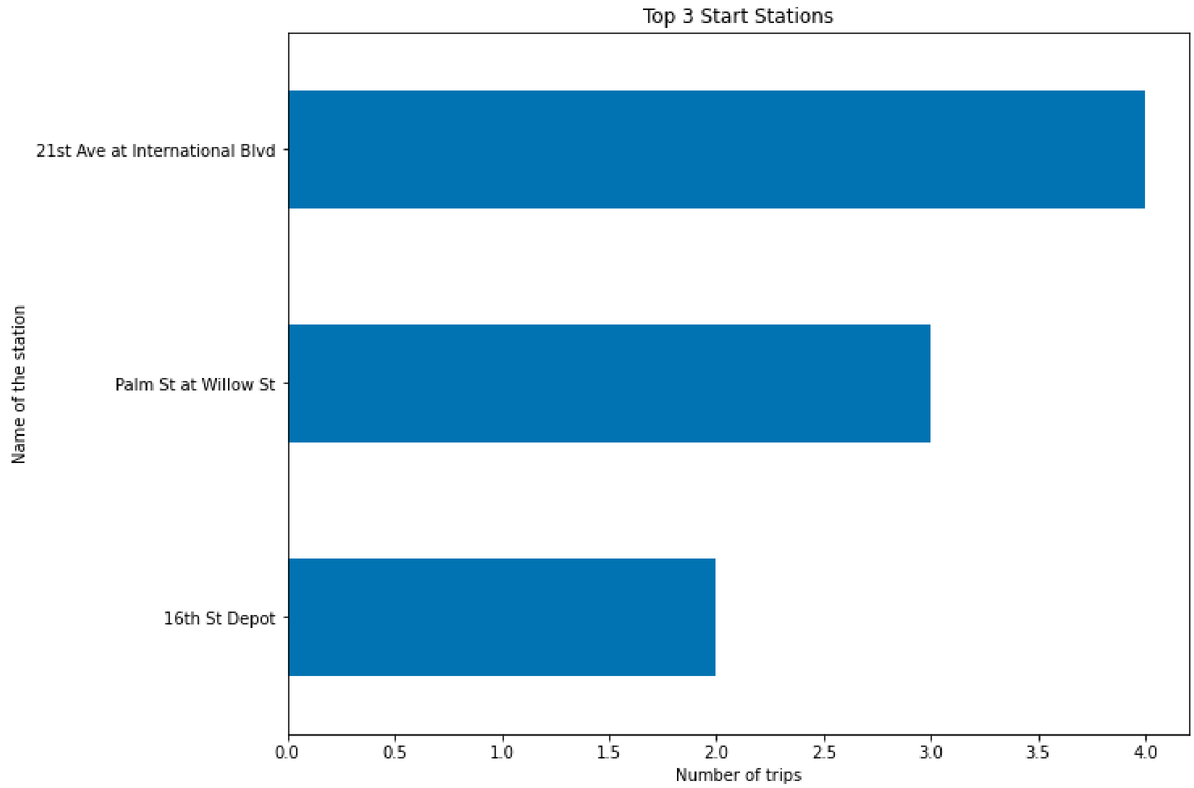


> Comment: See from the result, most duration time is less than 200 minutes. The longest time is more than 1400 minutes which is nearly a day. In this case, I think the 1400 might be incorrect.

Location related questions

4. Which stations are the top 3 stations as the start station?

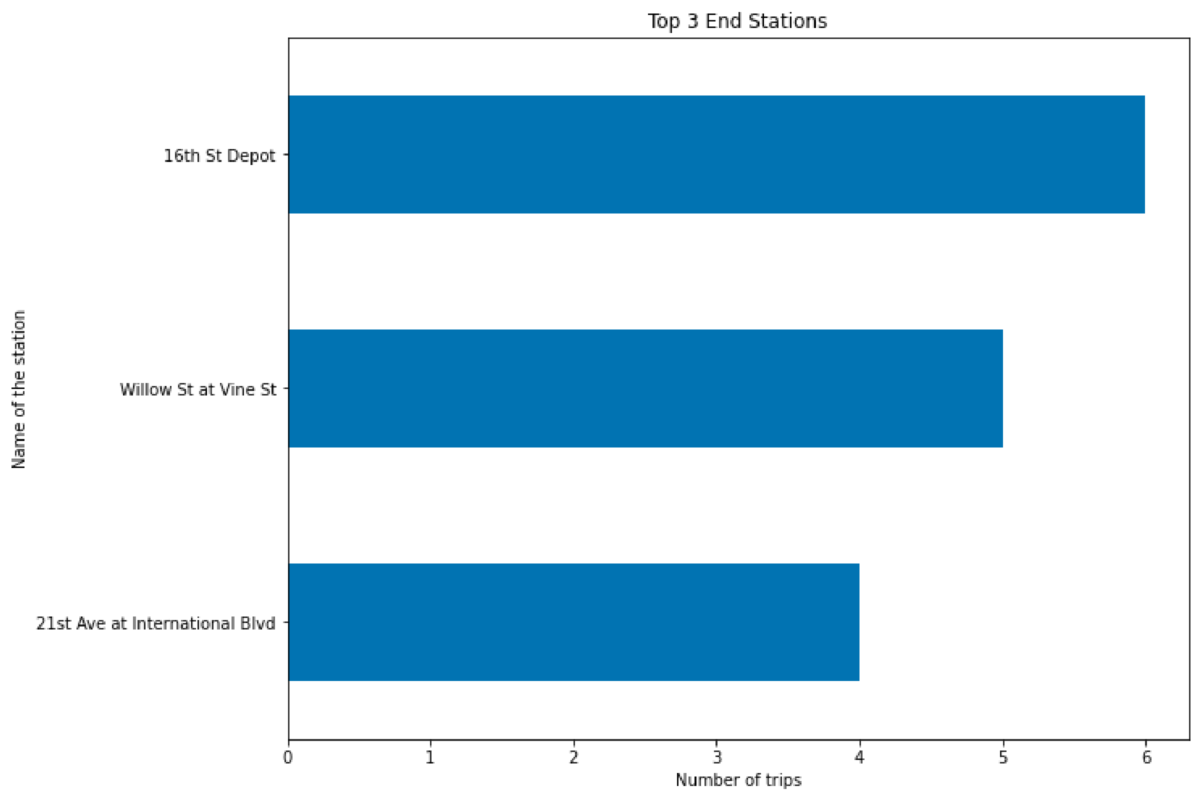
In [37]:
```
plt.figure(figsize=(10,8))
df_clean.start_station_name.value_counts(sort=True, ascending=True)[:3].plot
plt.title('Top 3 Start Stations')
plt.xlabel('Number of trips')
plt.ylabel('Name of the station');
```



> Comment: See from the result, the top 3 stating stations that has the highest frequency is 21st Ave at International Blvd, Palm St at Willow St, and 16th St Depot.

5. Which stations are the top 3 stations as the end station?

In [45]:
```
plt.figure(figsize=(10,8))
df_clean.end_station_name.value_counts(sort=True, ascending=True)[:3].plot(k
plt.title('Top 3 End Stations')
plt.xlabel('Number of trips')
plt.ylabel('Name of the station');
```

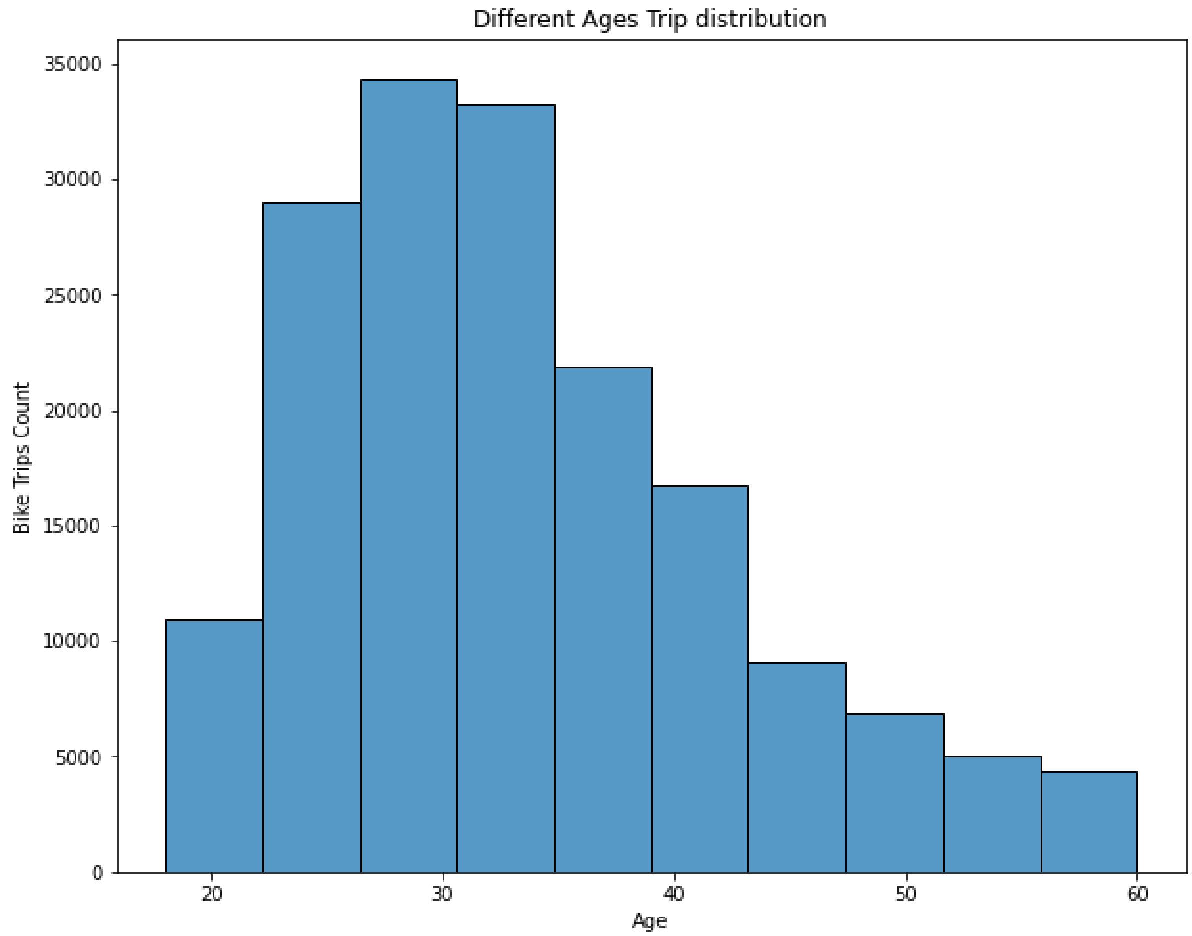


> Comment: See from the result, the top 3 stating stations that has the highest frequency is 16th St Depot, Willow st at Vine St and 21st Ave at International Blvd. There are 2 stations are the same as the start station, my guess is that these stations locate in busy areas.

User related questions

6. What is the user's age distribution?

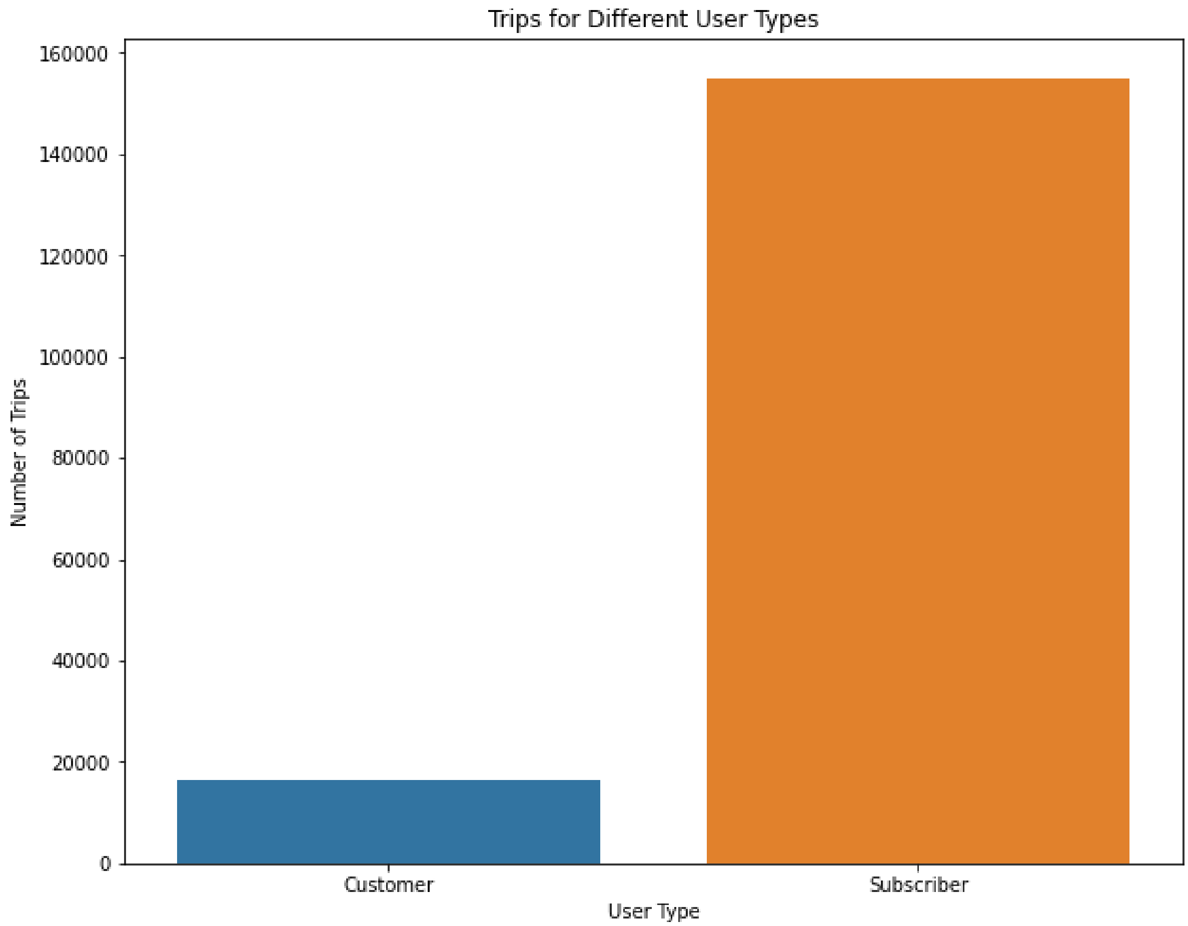
In [43]:
```
plt.figure(figsize=(10,8))
sb.histplot(x = 'member_age', data = df_clean,bins=10)
plt.title('Different Ages Trip distribution')
plt.xlabel('Age')
plt.ylabel('Bike Trips Count');
```



> Comment: See from the result, it is a right skewed distribution, with most of people are below 40 years, and the most of users are around 30. This result makes sense becuase people who just start to work and do not have too much saving for cars would choose to use sharing bikes

7. Do most of the users subsribe as members?

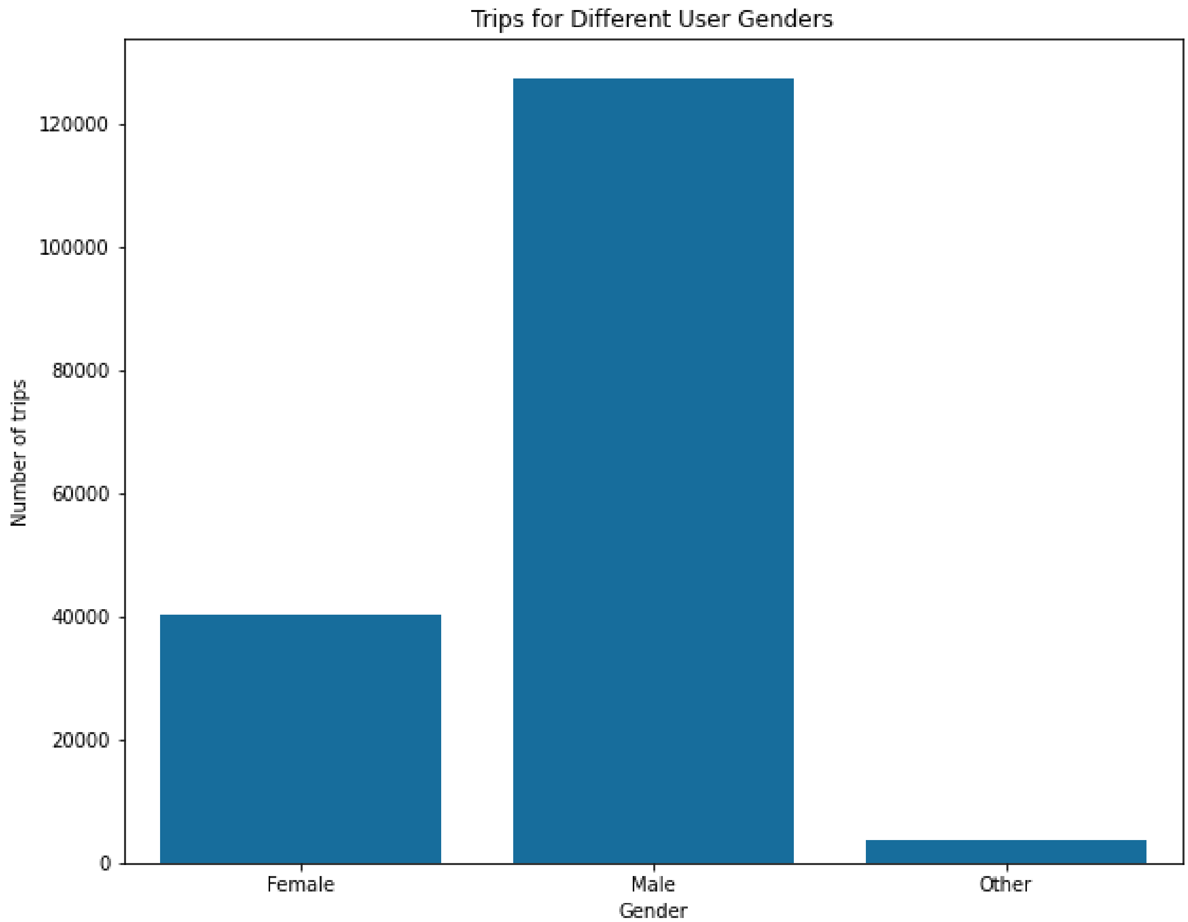
In [47]:
```
plt.figure(figsize=(10,8))
sb.countplot(data=df_clean, x='user_type')
plt.title('Trips for Different User Types')
plt.ylabel('Number of Trips')
plt.xlabel('User Type');
```



> Comment: See from the result, most users are subscribers.

8. Does users' gender have influence on the bike usage? (bar plot)

In [48]:
```
plt.figure(figsize = [10, 8])
sb.countplot(data=df_clean, x='member_gender', color=color);
plt.title('Trips for Different User Genders')
plt.xlabel('Gender');
plt.ylabel('Number of trips');
```

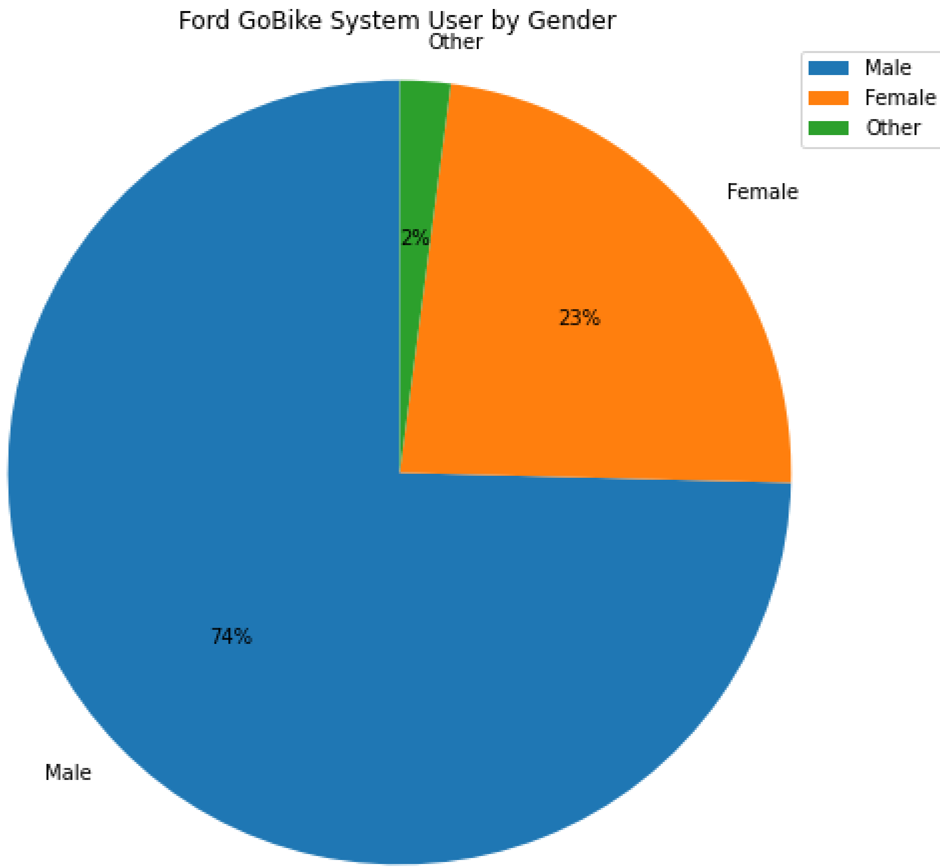


> Comment: See from the result, most users are male.

9. Does users' gender have influence on the bike usage? (pie plot, to show the percentage)

In [56]:
```
fig1, ax1 = plt.subplots(figsize=(10,8))
gender = df.member_gender.value_counts()
ax1.pie(gender, labels = gender.index, autopct='%1i%%', shadow=False, starta
ax1.axis('equal')
plt.legend(labels =gender.index)
plt.title("Ford GoBike System User by Gender")
```

Out[56]:
```
Text(0.5, 1.0, 'Ford GoBike System User by Gender')
```



> Comment: See from the result, most users are male, and the number of male users is around 74% of the total number of users.

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

> In this section about univariate exploration, I found the age and duration has unusual points. For both of the cases, I think they are incorrect data and I just discard the incorrect data. After my correction, for both age and duration, they have right skewed distributions. For age, most user are around 30 years old. For duration, most people would use the bike for less than 200 minutes.

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?

> I mainly investigate informations about time, location and user background. I discard irrelavant columns and rows, changed data type and add more

> columns in the dataset. The reason why I add more columns like age and duration in minutes are for the convenience of later analysis.

Bivariate Exploration

> In this section, investigate relationships between pairs of variables in your

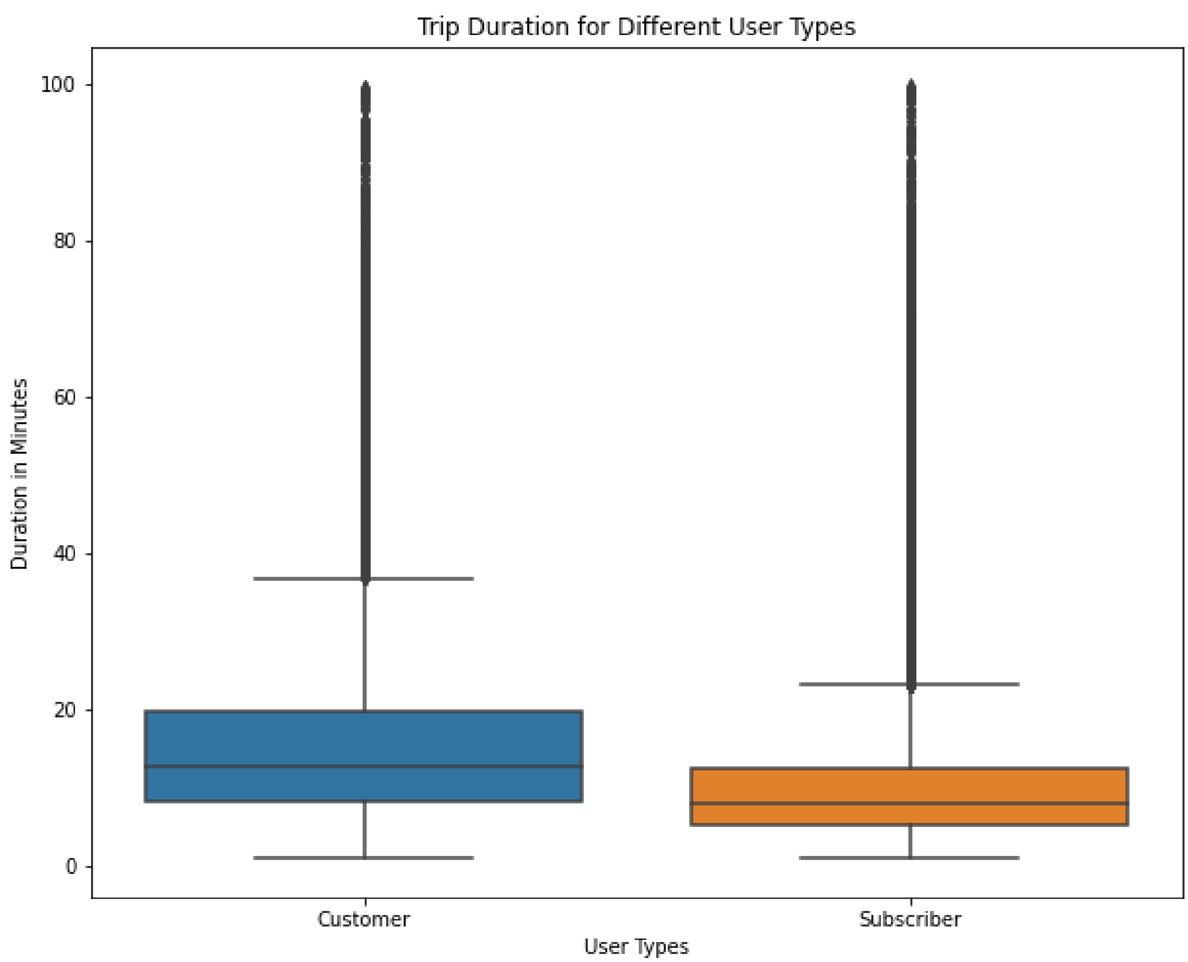
data. I would like to know more about the time and the users relation.

10. For different user types (substribed or not), what are their duration distribution?

In [70]:
```
plt.figure(figsize=[10,8])

tempdf = df_clean[df_clean['duration_minute'] <=100]
sb.boxplot(data = tempdf, x = 'user_type', y = 'duration_minute')

plt.title('Trip Duration for Different User Types')
plt.xlabel('User Types')
plt.ylabel('Duration in Minutes')
plt.show()
```

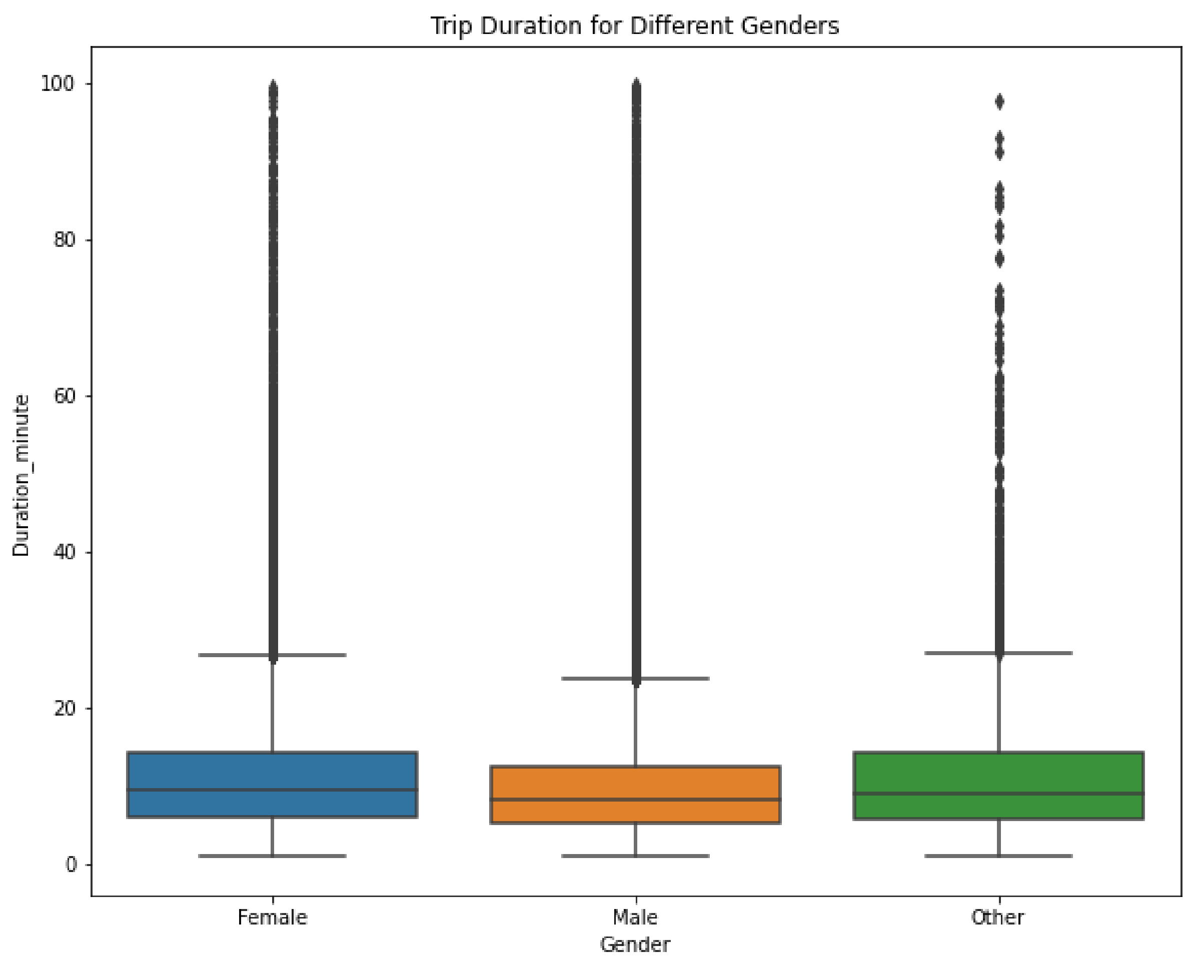


> Comment: See from the result, customers would typically ride slightly longer than subsribers. Both customer and substriber will not spend too much time riding the bike.

11. For different user genders, what are their duration distribution?

In [76]:
```
plt.figure(figsize = [10, 8])

sb.boxplot(data = tempdf, x = 'member_gender', y = 'duration_minute')
plt.title('Trip Duration for Different Genders')
plt.xlabel('Gender')
plt.ylabel('Duration_minute')
plt.show()
```

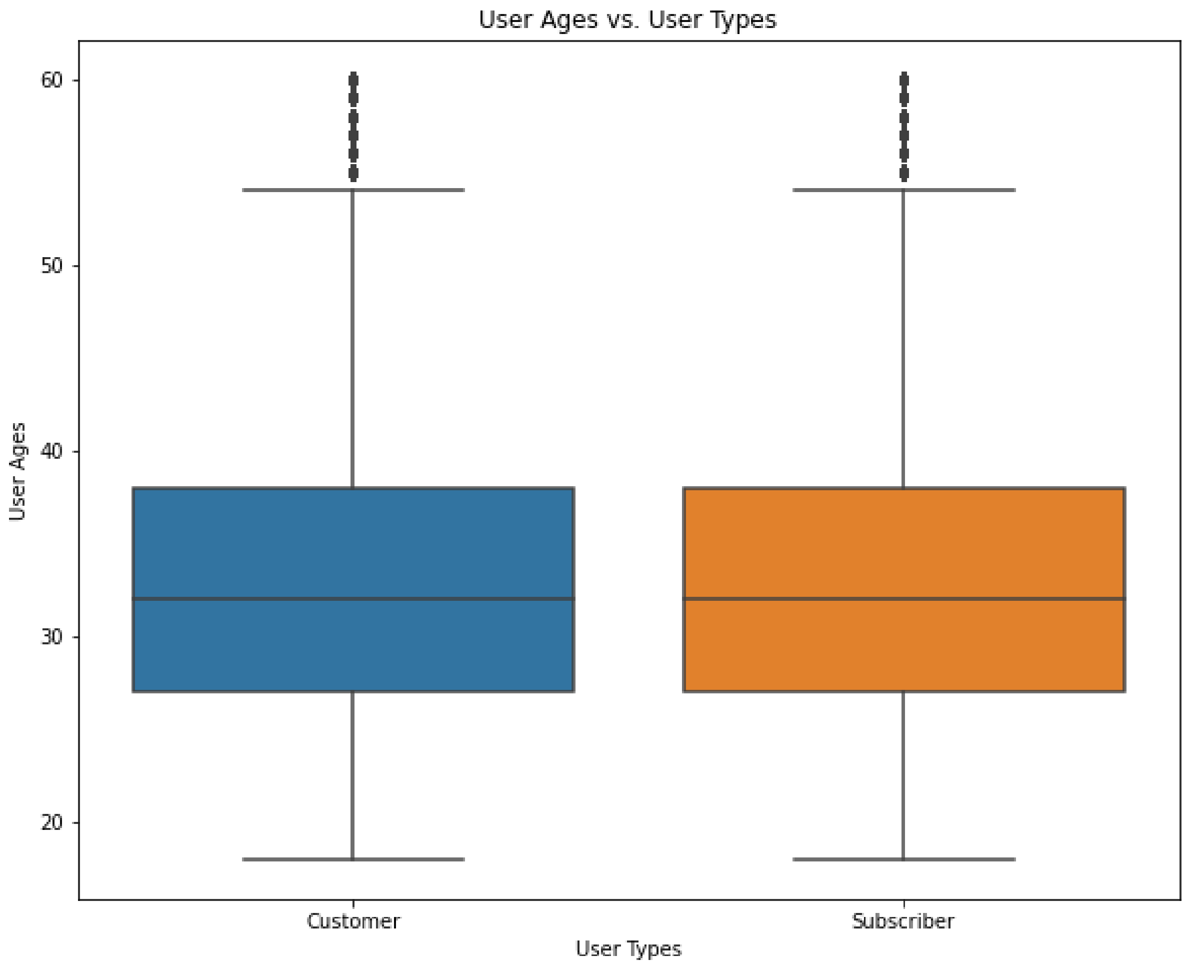


> Comment: See from the result, female users would typically ride slightly longer than other genders. All genders will not spend too much time riding the bike.

12. For different user ages, what is the user type (subsribed or not)?

In [80]:
```
plt.figure(figsize=(10,8))
sb.boxplot(data=df_clean, x='user_type', y='member_age');
```

```
plt.title('User Ages vs. User Types')
plt.xlabel('User Types');
plt.ylabel('User Ages');
```



> Comment: see from the result, the user age are not influenced too much by user types.

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?

> In this section about bivariate exploration, I mainly explored the relation between user information with trip duration. I found that the user age are not influenced too much by user types. No matter what genders or user types, most users would not ride the bikes for more than 40 munutes.

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

> I found that duration would be too long for most of the users.

Multivariate Exploration

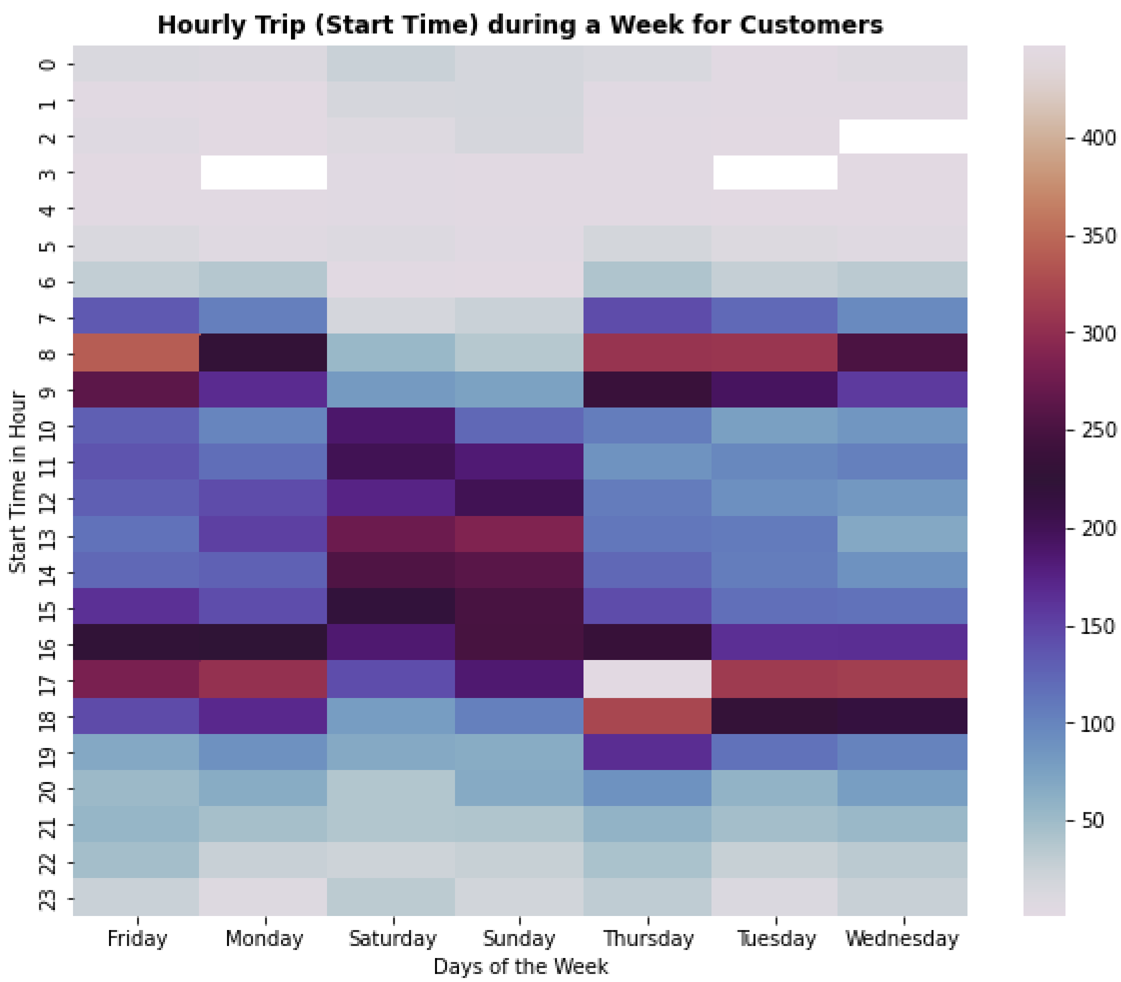
> In this section, I will create plots of three or more variables to investigate the data even

further. I am still interested to know more about the time and the users relation.

14. For different user types, how is the hourly trip during a week (7 days) look like?

In [109…

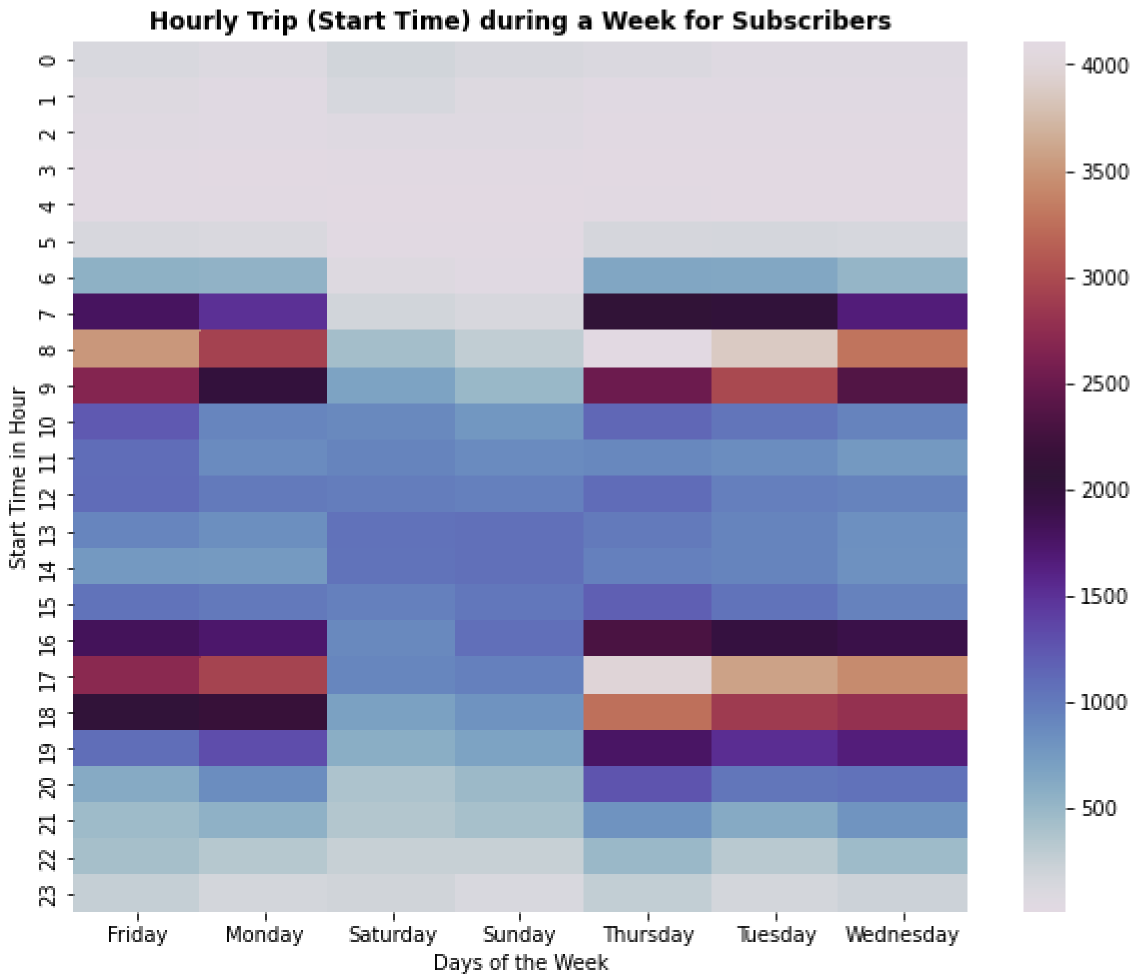
```
# For customers, their hourly trip during a week
plt.figure(figsize=(10,8))
customers = df_clean.query('user_type == "Customer"').groupby(['start_time_h
customers = customers.pivot('start_time_hour', 'weekday', 'bike_id')
heat_map = sb.heatmap(customers, cmap = 'twilight')
plt.title('Hourly Trip (Start Time) during a Week for Customers', fontweight
plt.xlabel('Days of the Week')
plt.ylabel('Start Time in Hour')
plt.show()
```



In [111…

```
# For subscribers, their hourly trip during a week
plt.figure(figsize=(10,8))
subscribers = df_clean.query('user_type == "Subscriber"').groupby(['start_ti
subscribers = subscribers.pivot('start_time_hour', 'weekday', 'bike_id')
```

```
heat_map = sb.heatmap(subscribers, cmap = 'twilight')
plt.title('Hourly Trip (Start Time) during a Week for Subscribers', fontweig
plt.xlabel('Days of the Week')
plt.ylabel('Start Time in Hour')
plt.show()
```



> Comment: see from the result, subscribers share the similar heatmap with customers on weekdays. However, subscribers would not use the bikes as mush as customers on weekends.

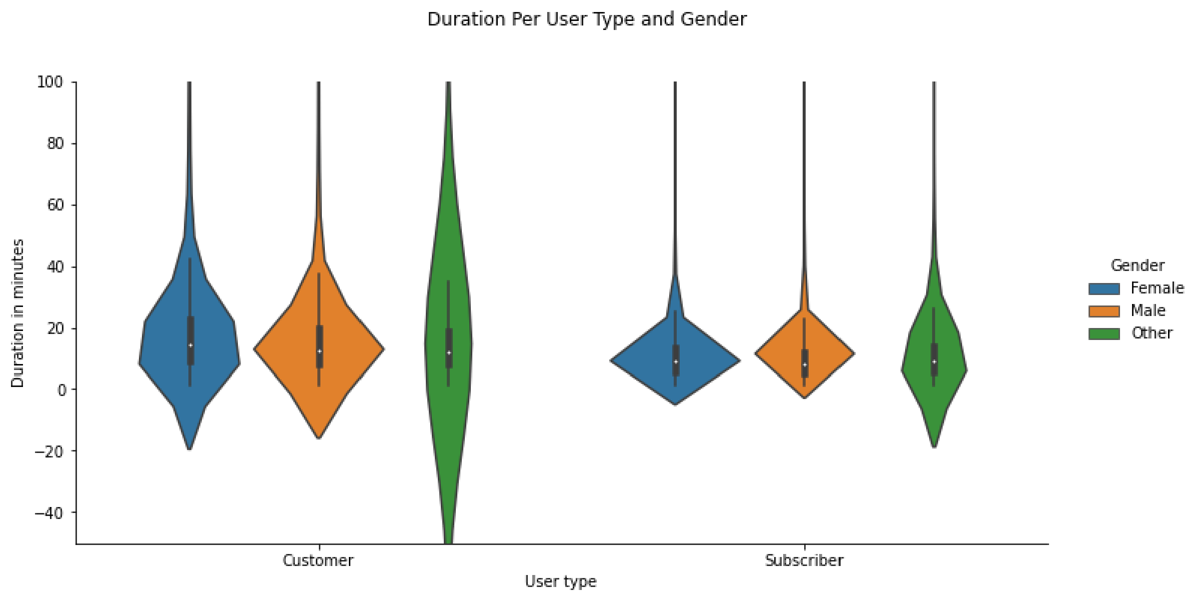
15. For different user types and genders, what does the trip duration in minutes look like?

In [102…

```
plt.figure(figsize=(10,8))
graph = sb.catplot(data=df_clean, x='user_type', y="duration_minute", hue="m

plt.ylim([-50, 100])
graph.set_axis_labels("User type", "Duration in minutes")
graph._legend.set_title('Gender')
graph.fig.suptitle('Duration Per User Type and Gender', y=1.1);
```

```
<Figure size 720x576 with 0 Axes>
```



> Comment: see from the result, male users has the least or similar variation as female users.

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?

> From the plot for question 14, I found that subscribers are more likely to use the bike for work. These features are strengthended each other by using univariate, bivariate and mutivariate analysis.

Were there any interesting or surprising interactions between features?

> Although the male has the largest number of usage, their usage duration is concentrated comparing to other genders.

Conclusions

> Since I have already commented under each plot, here I only write down the key insights.

> 1. User background: The most users for Ford GoBike are around 20 to 40 years old, male. Most of the users are subscribers.

> 2. Using time: The most bike using time is on weekdays during rush hours for work.

> 3. Usage duration: despite those extremely long and short usage, most users use the bike less than 40 minutes. Gender and user type does not have a large impact on the duration time.