Ford GoBike System Data - 2017

by (Ahmed Al-dayel)

Preliminary Wrangling

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

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sb
%matplotlib inline
```

In [2]:
```
df = pd.read_csv('2017-fordgobike-tripdata.csv')
```

In [3]:
```
df.head()
```

Out[3]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude
0	80110	2017-12-31 16:57:39.6540	2018-01-01 15:12:50.2450	74	Laguna St at Hayes St	37.776435	-122.426244
1	78800	2017-12-31 15:56:34.8420	2018-01-01 13:49:55.6170	284	Yerba Buena Center for the Arts (Howard St at ...	37.784872	-122.400876
2	45768	2017-12-31 22:45:48.4110	2018-01-01 11:28:36.8830	245	Downtown Berkeley BART	37.870348	-122.267764
3	62172	2017-12-31 17:31:10.6360	2018-01-01 10:47:23.5310	60	8th St at Ringold St	37.774520	-122.409449
4	43603	2017-12-31 14:23:14.0010	2018-01-01 02:29:57.5710	239	Bancroft Way at Telegraph Ave	37.868813	-122.258764

In [4]:
```
df.shape
```

Out[4]:
```
(519700, 13)
```

In [5]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 519700 entries, 0 to 519699
Data columns (total 13 columns):
 #   Column                   Non-Null Count   Dtype  
---  ------                   --------------   -----  
 0   duration_sec             519700 non-null  int64  
 1   start_time               519700 non-null  object 
 2   end_time                 519700 non-null  object 
 3   start_station_id         519700 non-null  int64  
 4   start_station_name       519700 non-null  object 
 5   start_station_latitude   519700 non-null  float64
 6   start_station_longitude  519700 non-null  float64
 7   end_station_id           519700 non-null  int64  
 8   end_station_name         519700 non-null  object 
 9   end_station_latitude     519700 non-null  float64
 10  end_station_longitude    519700 non-null  float64
 11  bike_id                  519700 non-null  int64  
 12  user_type                519700 non-null  object 
dtypes: float64(4), int64(4), object(5)
memory usage: 51.5+ MB
```

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
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
dtype: int64
```

In [7]:
```
df.drop(['start_station_latitude','start_station_longitude', 'end_station_latitude', 'end_st
ation_longitude'], axis=1, inplace=True)
```

In [8]:
```
df.head(1)
```

Out[8]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	end_station_id	end_station_name	bike_id	u
0	80110	2017-12-31 16:57:39.6540	2018-01-01 15:12:50.2450	74	Laguna St at Hayes St	43	San Francisco Public Library (Grove St at Hyde...	96	

In [9]:
```
df['start_time'] = pd.to_datetime(df['start_time'])
```

In [10]:
```
df['day'] = df['start_time'].apply(lambda x: x.strftime('%A').lower())
days = ['monday', 'tuesday', 'wednesday', 'thursday', 'friday', 'saturday','sunday']
df['day'] = pd.Categorical(df['day'],categories= days, ordered = True)
```

In [11]:
```
df.head(1)
```

Out[11]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	end_station_id	end_station_name	bike_id	us
0	80110	2017-12-31 16:57:39.654	2018-01-01 15:12:50.2450	74	Laguna St at Hayes St	43	San Francisco Public Library (Grove St at Hyde...	96	C

In [12]:
```
df.tail(1)
```

Out[12]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	end_station_id	end_station_name	bike_
519699	188	2017-06-28 09:49:46.377	2017-06-28 09:52:55.3380	25	Howard St at 2nd St	48	2nd St at S Park St	1

In [13]:
```
df.describe()
```

Out[13]:

	duration_sec	start_station_id	end_station_id	bike_id
count	519700.000000	519700.000000	519700.000000	519700.000000
mean	1099.009521	95.034245	92.184041	1672.533079
std	3444.146451	86.083078	84.969491	971.356959
min	61.000000	3.000000	3.000000	10.000000
25%	382.000000	24.000000	23.000000	787.000000
50%	596.000000	67.000000	66.000000	1728.500000
75%	938.000000	139.000000	134.000000	2520.000000
max	86369.000000	340.000000	340.000000	3733.000000

In [14]:
```
df['user_type'].value_counts()
```

Out[14]:
```
Subscriber    409230
Customer      110470
Name: user_type, dtype: int64
```

In [15]:
```
df['bike_id'].value_counts()
```

Out[15]:
```
68      457
2178    426
210     408
813     403
602     402
       ... 
302       1
501       1
2609      1
3323      1
3723      1
Name: bike_id, Length: 3673, dtype: int64
```

In [16]:
```
df['day'].value_counts()
```

Out[16]:
```
tuesday      87865
wednesday    87752
thursday     85243
monday       81410
friday       81165
saturday     50874
sunday       45391
Name: day, dtype: int64
```

In [17]:
```
df.shape
```

Out[17]:
```
(519700, 10)
```

What is the structure of your dataset?

(We have 519700 rows and 10 columns)

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

duration time for each trip, user type who use the bike and average or number of trip for each day we have.

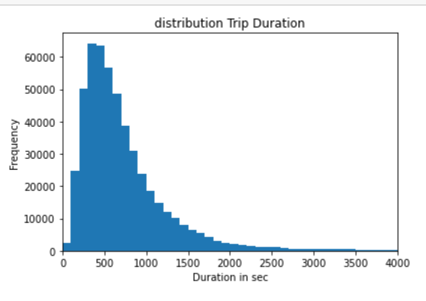
What features in the dataset do you think will help support your investigation into your feature(s) of interest?

- duration_sec column will help me to get the average.
- user_type column will help me to get the average.
- day column will help me to get the average.

Univariate Exploration

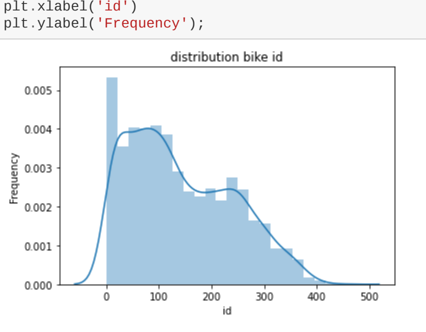
i want to look to the distribution for duration_sec

In [18]:
```
bin_edges = np.arange(0, df['duration_sec'].max()+100, 100)
plt.hist(data = df, x = 'duration_sec', bins = bin_edges)
plt.xlim(0,4000)
plt.title('distribution Trip Duration')
plt.xlabel('Duration in sec')
plt.ylabel('Frequency');
```



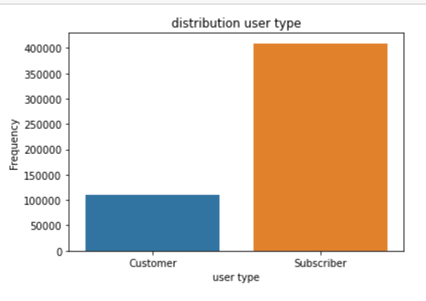
from that we have seen above the often of the duration were on 500 - 600 second, so it's so clear to me now.

In [19]:
```
sb.distplot(df['bike_id'].value_counts())
plt.title('distribution bike id')
plt.xlabel('id')
plt.ylabel('Frequency');
```



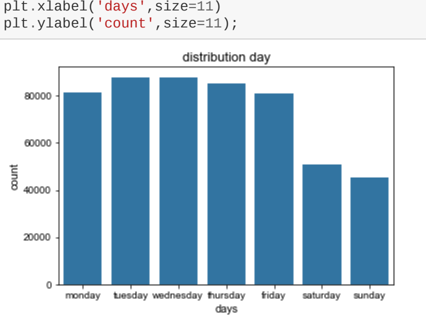
from that we have seen above the highest distribution of start_station_id often on 15 - 75.

In [20]:
```
sb.countplot(data = df, x = 'user_type')
plt.title('distribution user type')
plt.xlabel('user type')
plt.ylabel('Frequency');
```



from that we have seen above we see the Subscribers are higher than Customer.

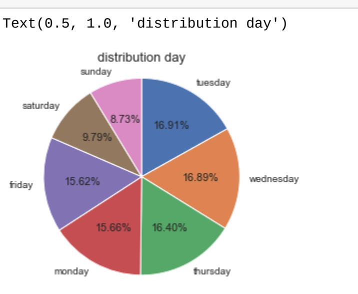
In [21]:
```
base_color = sb.color_palette()[0]
sb.countplot(data = df, x = 'day', color = base_color)
sb.set(font_scale=0.90)
plt.title('distribution day',size=13)
plt.xlabel('days',size=11)
plt.ylabel('count',size=11);
```



from that chart we have seen above we can't see the obvious difference between the tuesday and wednesday clearly, so we should use the pie chart to show the percentage and then we can see the diffrence between days clearly.

In [22]:
```
day_count = df['day'].value_counts()
plt.pie(day_count, labels = day_count.index, startangle = 90,
        counterclock = False,autopct='%.2f%%');
plt.axis('square')
plt.title('distribution day',size=13)
```

Out[22]:
```
Text(0.5, 1.0, 'distribution day')
```



from that pie chart we have seen above we see the tuesday is higher than wednesday in a simple percentage.

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

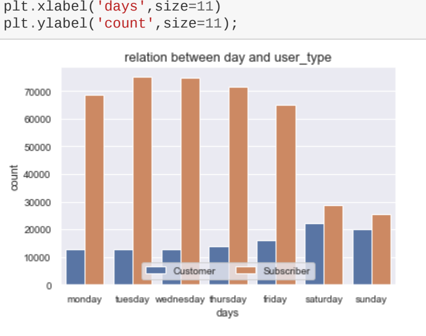
We could see from the charts above that i used a distribuation of trip duration and distribuation of user type and distribuation of day, i don't have any unusual points, i had do the transformation from seaborn countplot to pie chart matplotlib.

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 had do the transformation from seaborn countplot to pie chart matplotlib in day distribution to be showing the percentage for day distribution very clearly.

Bivariate Exploration

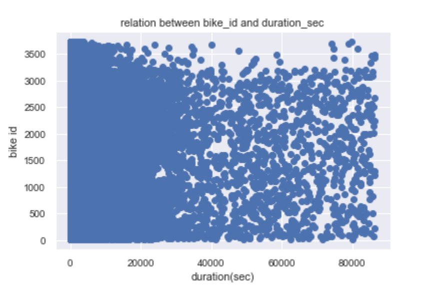
In [23]:
```
ax = sb.countplot(data = df, x = 'day', hue = 'user_type')
ax.legend(loc = 8, ncol = 3, framealpha = 50)
plt.title('relation between day and user_type',size=13)
plt.xlabel('days',size=11)
plt.ylabel('count',size=11);
```



We can see the chart above that tell us the weekend not like Beginning of the week because the people are very close in the weekend not like the Beginning of the week.

In [24]:
```
plt.scatter(data = df, x = 'duration_sec', y = 'bike_id')
plt.title('relation between bike_id and duration_sec')
plt.xlabel('duration(sec)')
plt.ylabel('bike id')
```

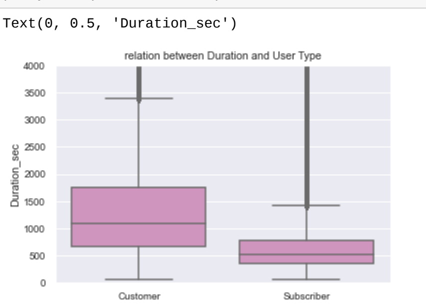
Out[24]:
```
Text(0, 0.5, 'bike id')
```



We can see the chart above that tell us the most of Bike id often work on duration from 0 to 3000.

In [25]:
```
base_color = sb.color_palette()[6]
sb.boxplot(data = df, x = 'user_type', y = 'duration_sec', color = base_color)
plt.ylim([0, 4000])
plt.title('relation between Duration and User Type')
plt.xlabel('User Type')
plt.ylabel('Duration_sec')
```

Out[25]:
```
Text(0, 0.5, 'Duration_sec')
```



We can see the chart above that tell us that customer rides trip have longer duration than subscriber rides trip.

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?

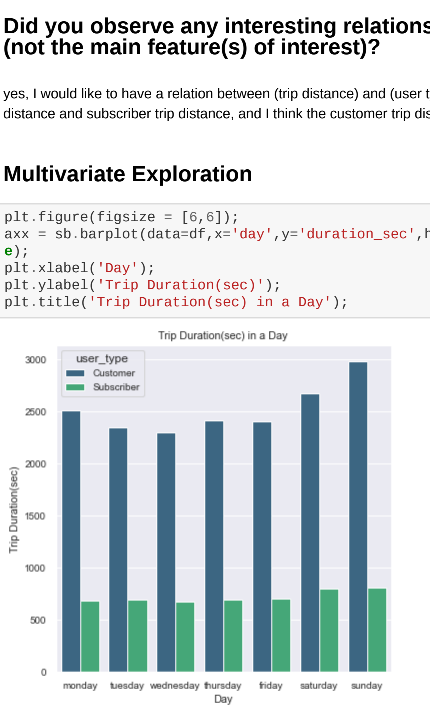
- I have put a relation between Day and User type.
- I have put a relation between Bike id and Duration in second.
- I have put a relation between User Type and Duration in second.

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

yes, I would like to have a relation between (trip distance) and (user type) to know if there is difference between customer trip distance and subscriber trip distance, and I think the customer trip distance is higher than the subscriber trip distance.

Multivariate Exploration

In [26]:
```
plt.figure(figsize = [6,6]);
axx = sb.barplot(data=df,x='day',y='duration_sec',hue='user_type', palette='viridis', ci=Non
e);
plt.xlabel('Day');
plt.ylabel('Trip Duration(sec)');
plt.title('Trip Duration(sec) in a Day');
```



We can see the chart above that tell us the customers are using the bike much longer duration in day than the subscribers.

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

I have made a relationship between day and user_type against duration_sec.