Part_I_exploration_template

February 13, 2023

1 Part I - Ford GoBike System Data

1.1 by Waad AlDoaij

1.2 Introduction

This data set which contains details about each trip taken in a bike-sharing program serving the greater San Francisco Bay area, will be used in this study, these dataframe following columns are: (duration_sec, start_time, starting_Day, end_time, ending_Day, start_station_id, start_station_name, start_station_latitude,start_station_longitude, end_station_id, end_station_name,end_station_latitude, end_station_longitude, bike_id, user_type,member_birth_year, member_gender, bike_share_for_all_trip,Hours)

In order to better understand the data, I thought about the provided questions before I began the analysis process. - When in terms of the time of day and day of the week, are most trips taken? - How long does an average trip ? - Does it matter whether a user is a subscriber or a customer?

The step journey in this project began with importing all of the packages, then uploading the dataset, and then beginning wrangling. After that, I made some observations and began cleaning up the issues that I faced, and finally, I began to visualize all of the charts.

```
In [67]: # Import all packages
         import numpy as np
         import pandas as pd
         import matplotlib.pyplot as plt
         import seaborn as sb

         %matplotlib inline
```

1.3 Uploading the Dataset

```
In [68]: Bike_df = pd.read_csv('Fordgobike-tripdata.csv')

         Bike_df.head()
```

```
Out[68]:    duration_sec                start_time                  end_time  \
         0         52185  2019-02-28 17:32:10.1450  2019-03-01 08:01:55.9750
         1         42521  2019-02-28 18:53:21.7890  2019-03-01 06:42:03.0560
         2         61854  2019-02-28 12:13:13.2180  2019-03-01 05:24:08.1460
         3         36490  2019-02-28 17:54:26.0100  2019-03-01 04:02:36.8420
```

```
4          1585  2019-02-28 23:54:18.5490  2019-03-01 00:20:44.0740

   start_station_id                                start_station_name  \
0              21.0  Montgomery St BART Station (Market St at 2nd St)
1              23.0                      The Embarcadero at Steuart St
2              86.0                            Market St at Dolores St
3             375.0                            Grove St at Masonic Ave
4               7.0                                Frank H Ogawa Plaza

   start_station_latitude  start_station_longitude  end_station_id  \
0               37.789625              -122.400811            13.0
1               37.791464              -122.391034            81.0
2               37.769305              -122.426826             3.0
3               37.774836              -122.446546            70.0
4               37.804562              -122.271738           222.0

                             end_station_name  end_station_latitude  \
0               Commercial St at Montgomery St             37.794231
1                           Berry St at 4th St             37.775880
2  Powell St BART Station (Market St at 4th St)             37.786375
3                        Central Ave at Fell St             37.773311
4                         10th Ave at E 15th St             37.792714

   end_station_longitude  bike_id   user_type  member_birth_year  \
0            -122.402923     4902    Customer             1984.0
1            -122.393170     2535    Customer                NaN
2            -122.404904     5905    Customer             1972.0
3            -122.444293     6638  Subscriber             1989.0
4            -122.248780     4898  Subscriber             1974.0

  member_gender bike_share_for_all_trip
0          Male                     No
1           NaN                     No
2          Male                     No
3         Other                     No
4          Male                    Yes
```

1.4 Preliminary Wrangling

```
In [69]: Bike_df.shape
```

```
Out[69]: (183412, 16)
```

```
In [70]: Bike_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
```

```
In [71]: #checking for null values
         Bike_df.isnull().sum()
```

```
Out[71]: 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 [72]: Bike_df.user_type.value_counts()
```

```
Out[72]: Subscriber    163544
         Customer       19868
         Name: user_type, dtype: int64
```

```
In [73]: Bike_df.member_gender.value_counts()
```

```
Out[73]: Male      130651
         Female     40844
```

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

```
In [74]: Bike_df.bike_share_for_all_trip.value_counts()
```

```
Out[74]: No     166053
         Yes     17359
         Name: bike_share_for_all_trip, dtype: int64
```

1.4.1 What is the structure of your dataset?

The dataset has 183412 rows with the following 16 columns (duration sec, start time, end time, start station id, start station name, start station latitude, end station longitude, bike id, user type, member birth year, member gender, and bike share for all trip)

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

I'm especially curious to know whether males or women are the more common gender riders, also if there is a difference in user type, whether subscriber or customer..

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

- member_gender
- user_type

Observation Summary:
`Quality issues` : - Some missing values - wrong datatype for both (start_time,end_time)

1.5 Cleaning

```
In [10]: #Makeing copy of the dataframe

         Bike_df_clean = Bike_df.copy()
```

```
In [11]: #Coverting the datatype to datetime

         Bike_df_clean ['end_time']= pd.to_datetime(Bike_df_clean['end_time'])
         Bike_df_clean ['start_time']= pd.to_datetime(Bike_df_clean['start_time'])

         #Test
         Bike_df_clean.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 datetime64[ns]
```

```
end_time                   183412 non-null datetime64[ns]
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: datetime64[ns](2), float64(7), int64(2), object(5)
memory usage: 22.4+ MB
```

```
In [12]: #Drop the missing values
         Bike_df_clean = Bike_df_clean.dropna()
         #Test
         Bike_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 16 columns):
duration_sec               174952 non-null int64
start_time                 174952 non-null datetime64[ns]
end_time                   174952 non-null datetime64[ns]
start_station_id           174952 non-null float64
start_station_name         174952 non-null object
start_station_latitude     174952 non-null float64
start_station_longitude    174952 non-null float64
end_station_id             174952 non-null float64
end_station_name           174952 non-null object
end_station_latitude       174952 non-null float64
end_station_longitude      174952 non-null float64
bike_id                    174952 non-null int64
user_type                  174952 non-null object
member_birth_year          174952 non-null float64
member_gender              174952 non-null object
bike_share_for_all_trip    174952 non-null object
dtypes: datetime64[ns](2), float64(7), int64(2), object(5)
memory usage: 22.7+ MB
```

```
In [13]: #Coverting the datatype to int
         Bike_df_clean['member_birth_year'] = Bike_df_clean['member_birth_year'].astype('int')
```

```
         #Test
         Bike_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 16 columns):
duration_sec               174952 non-null int64
start_time                 174952 non-null datetime64[ns]
end_time                   174952 non-null datetime64[ns]
start_station_id           174952 non-null float64
start_station_name         174952 non-null object
start_station_latitude     174952 non-null float64
start_station_longitude    174952 non-null float64
end_station_id             174952 non-null float64
end_station_name           174952 non-null object
end_station_latitude       174952 non-null float64
end_station_longitude      174952 non-null float64
bike_id                    174952 non-null int64
user_type                  174952 non-null object
member_birth_year          174952 non-null int64
member_gender              174952 non-null object
bike_share_for_all_trip    174952 non-null object
dtypes: datetime64[ns](2), float64(6), int64(3), object(5)
memory usage: 22.7+ MB
```

```
In [14]: #Creating new two column for returning the end and start day from the start_time.
         Bike_df_clean.insert(2,'starting_Day',Bike_df_clean['start_time'].dt.day_name())

         Bike_df_clean.insert(4,'Ending_Day',Bike_df_clean['end_time'].dt.day_name())
```

```
In [16]: #Creating new column for returning hours
         Bike_df_clean['Hours'] = Bike_df_clean['start_time'].dt.hour
```

```
In [15]: #Test
         Bike_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 18 columns):
duration_sec               174952 non-null int64
start_time                 174952 non-null datetime64[ns]
starting_Day               174952 non-null object
end_time                   174952 non-null datetime64[ns]
Ending_Day                 174952 non-null object
start_station_id           174952 non-null float64
start_station_name         174952 non-null object
start_station_latitude     174952 non-null float64
start_station_longitude    174952 non-null float64
```

```
end_station_id             174952 non-null float64
end_station_name           174952 non-null object
end_station_latitude       174952 non-null float64
end_station_longitude      174952 non-null float64
bike_id                    174952 non-null int64
user_type                  174952 non-null object
member_birth_year          174952 non-null int64
member_gender              174952 non-null object
bike_share_for_all_trip    174952 non-null object
dtypes: datetime64[ns](2), float64(6), int64(3), object(7)
memory usage: 25.4+ MB
```

Functions

```
In [17]: #Returning the ages
         ages = 2019 - Bike_df_clean.member_birth_year

In [18]: #Craeting function for chart labals
         def functionN(x,y,title):
              plt.xlabel(x , fontsize=14)
              plt.ylabel(y , fontsize=14)
              plt.title(title,fontsize=18)

In [75]: #Craeting function for chart size
         def figz(x , y):
             plt.figure(figsize=(x,y))
```

1.6 Univariate Exploration

In this section, investigate distributions of individual variables. If you see unusual points or outliers, take a deeper look to clean things up and prepare yourself to look at relationships between variables.

```
In [19]: Bike_df_clean.shape

Out[19]: (174952, 19)

In [20]: Bike_df_clean.columns

Out[20]: Index(['duration_sec', 'start_time', 'starting_Day', 'end_time', 'Ending_Day',
                'start_station_id', 'start_station_name', 'start_station_latitude',
                'start_station_longitude', 'end_station_id', 'end_station_name',
                'end_station_latitude', 'end_station_longitude', 'bike_id', 'user_type',
                'member_birth_year', 'member_gender', 'bike_share_for_all_trip',
                'Hours'],
               dtype='object')

In [21]: Bike_df_clean.describe()
```

Out[21]:

	duration_sec	start_station_id	start_station_latitude
count	174952.000000	174952.000000	174952.000000
mean	704.002744	139.002126	37.771220
std	1642.204905	111.648819	0.100391
min	61.000000	3.000000	37.317298
25%	323.000000	47.000000	37.770407
50%	510.000000	104.000000	37.780760
75%	789.000000	239.000000	37.797320
max	84548.000000	398.000000	37.880222

	start_station_longitude	end_station_id	end_station_latitude
count	174952.000000	174952.000000	174952.000000
mean	-122.351760	136.604486	37.771414
std	0.117732	111.335635	0.100295
min	-122.453704	3.000000	37.317298
25%	-122.411901	44.000000	37.770407
50%	-122.398279	101.000000	37.781010
75%	-122.283093	238.000000	37.797673
max	-121.874119	398.000000	37.880222

	end_station_longitude	bike_id	member_birth_year	Hours
count	174952.000000	174952.000000	174952.000000	174952.000000
mean	-122.351335	4482.587555	1984.803135	13.456165
std	0.117294	1659.195937	10.118731	4.734282
min	-122.453704	11.000000	1878.000000	0.000000
25%	-122.411647	3799.000000	1980.000000	9.000000
50%	-122.397437	4960.000000	1987.000000	14.000000
75%	-122.286533	5505.000000	1992.000000	17.000000
max	-121.874119	6645.000000	2001.000000	23.000000

In [22]:
```
Bike_df_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 174952 entries, 0 to 183411
Data columns (total 19 columns):
duration_sec               174952 non-null int64
start_time                 174952 non-null datetime64[ns]
starting_Day               174952 non-null object
end_time                   174952 non-null datetime64[ns]
Ending_Day                 174952 non-null object
start_station_id           174952 non-null float64
start_station_name         174952 non-null object
start_station_latitude     174952 non-null float64
start_station_longitude    174952 non-null float64
end_station_id             174952 non-null float64
end_station_name           174952 non-null object
end_station_latitude       174952 non-null float64
end_station_longitude      174952 non-null float64
```

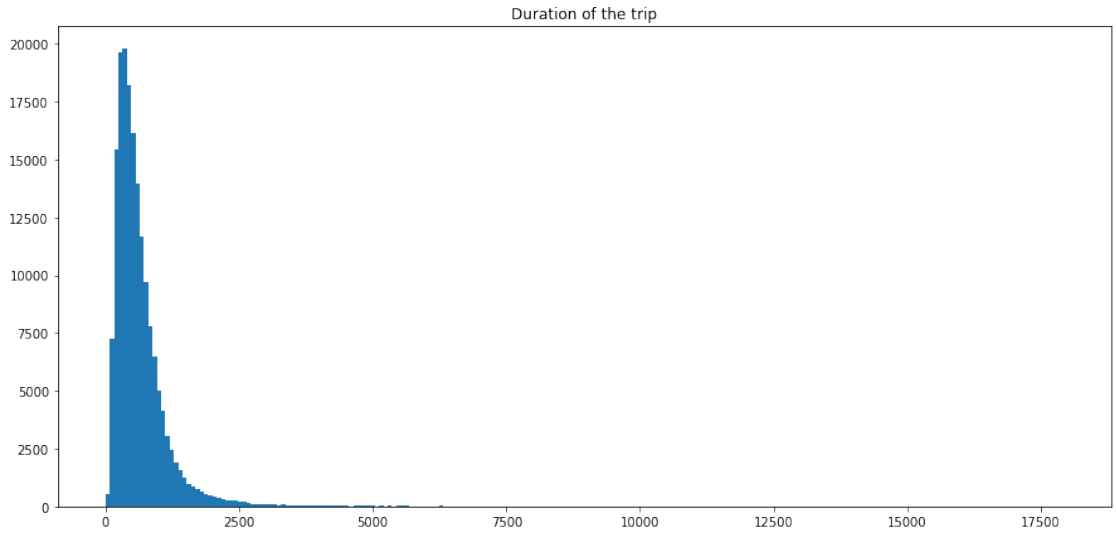
```
bike_id                  174952 non-null int64
user_type                174952 non-null object
member_birth_year        174952 non-null int64
member_gender            174952 non-null object
bike_share_for_all_trip  174952 non-null object
Hours                    174952 non-null int64
dtypes: datetime64[ns](2), float64(6), int64(4), object(7)
memory usage: 26.7+ MB
```

What is the average duration of the trips?

```
In [23]: Bike_df_clean.duration_sec.describe()
```

```
Out[23]: count    174952.000000
         mean        704.002744
         std        1642.204905
         min          61.000000
         25%         323.000000
         50%         510.000000
         75%         789.000000
         max       84548.000000
         Name: duration_sec, dtype: float64
```

```
In [81]: figz(15,7)
         b = np.arange(0, 18000 , 80)

         plt.hist(data= Bike_df_clean, x='duration_sec', bins=b)
         plt.title('Duration of the trip');
```

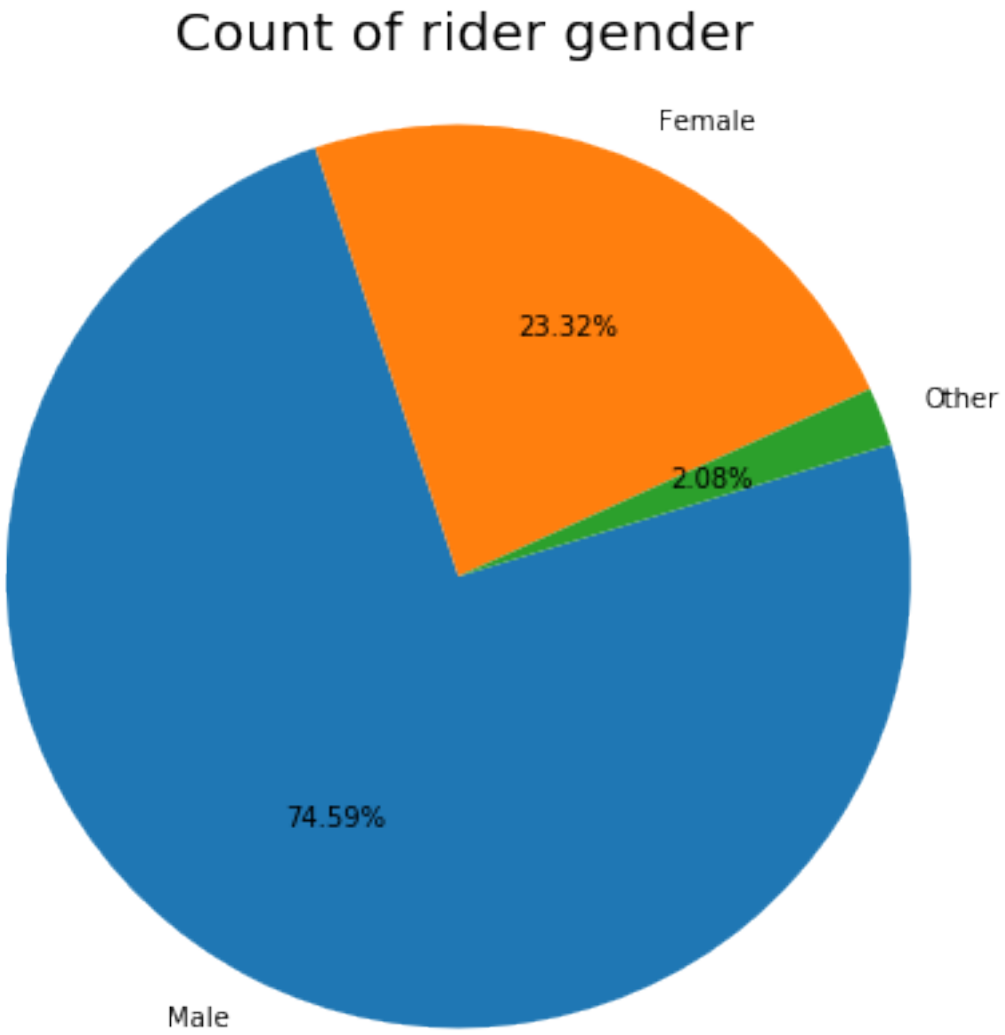


the avrage show to be 704s which is 11m, the max is 84548 lead to be 1409m around 23h, min 61s about 1m,

the chart display the distribuation shown to be right-skewed.

what is the common gender riders ?

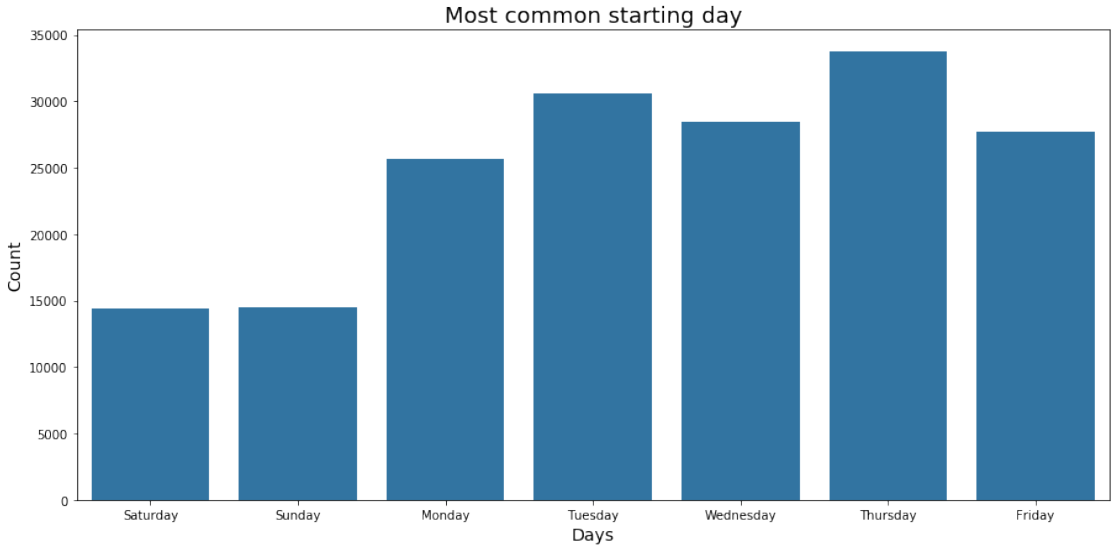
```
In [112]: figz(19,7)
          member_count = Bike_df_clean['member_gender'].value_counts()

          plt.pie(member_count  , labels = member_count.index ,startangle = 17, counterclock = F
          plt.axis('square')
          plt.title('Count of rider gender', fontsize= 20);
```



The above chart shows that male have the higest score of all riders with 74.59%.

What is the most common started day ?

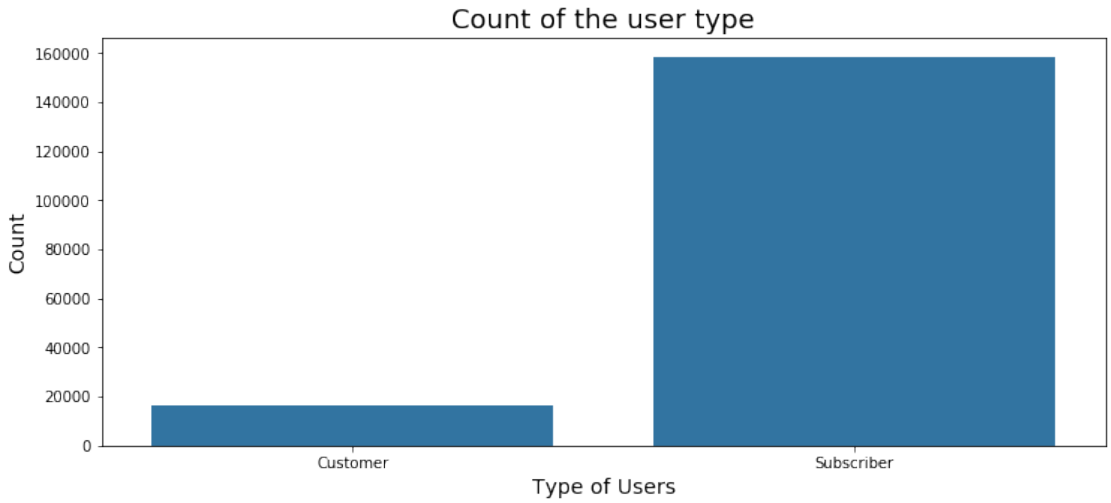
```
In [26]: Bike_df_clean.starting_Day.value_counts()

Out[26]: Thursday     33712
         Tuesday      30584
         Wednesday    28426
         Friday       27663
         Monday       25641
         Sunday       14512
         Saturday     14414
         Name: starting_Day, dtype: int64

In [111]: figz(15,7)
          days = ['Saturday', 'Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday','Friday']
          blue = sb.color_palette()[0]

          sb.countplot(data=Bike_df_clean, x='starting_Day', order=days , color = blue);

          functionN('Days', 'Count','Most common starting day')
```



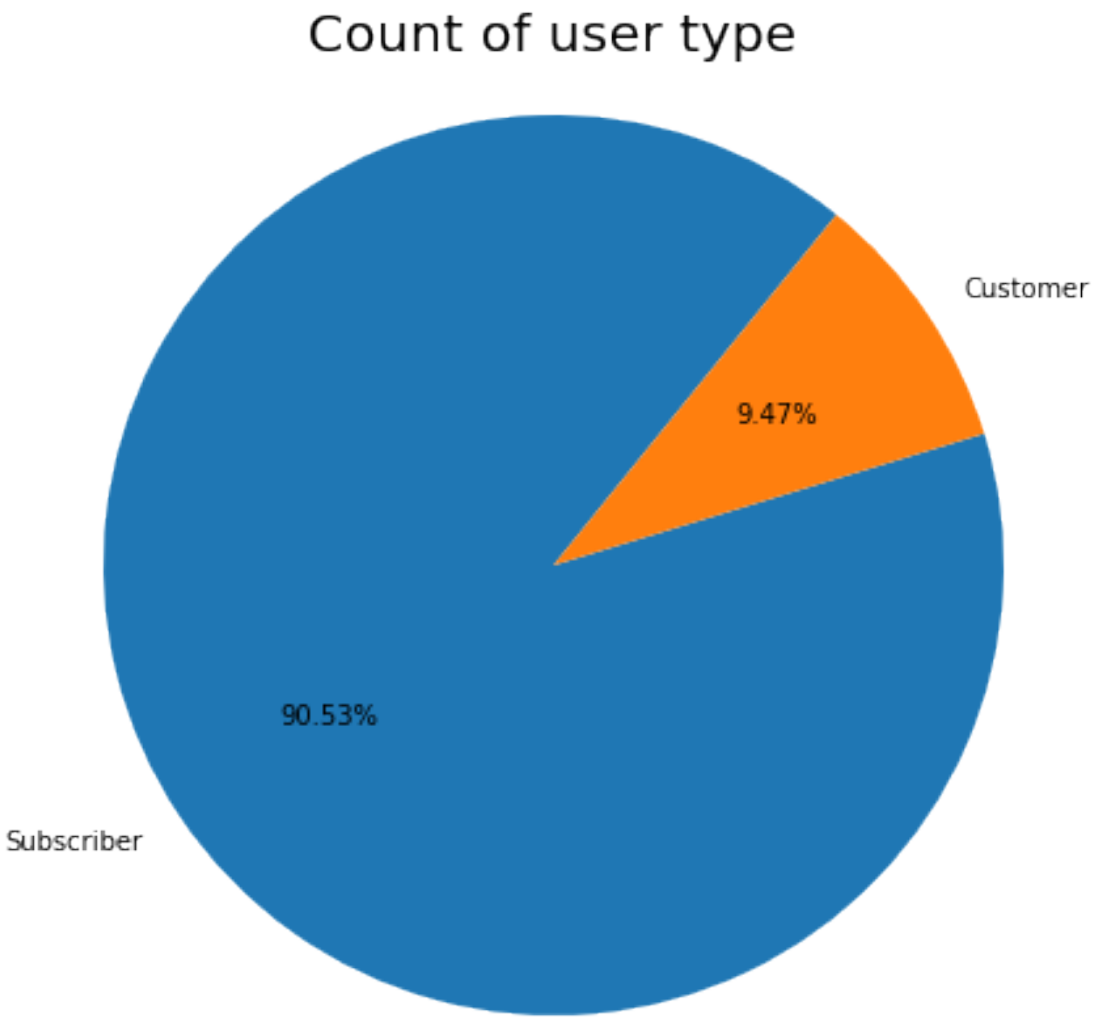
As shown it end to be the most common starting day was Thursday and Tuesday where sunday and saturday was the lowest.

Who is the most recent user customer or subscriber ?

```
In [113]: figz(12,5)
          sb.countplot(data=Bike_df_clean, x ='user_type', color=blue)
          functionN('Type of Users', 'Count','Count of the user type')
```

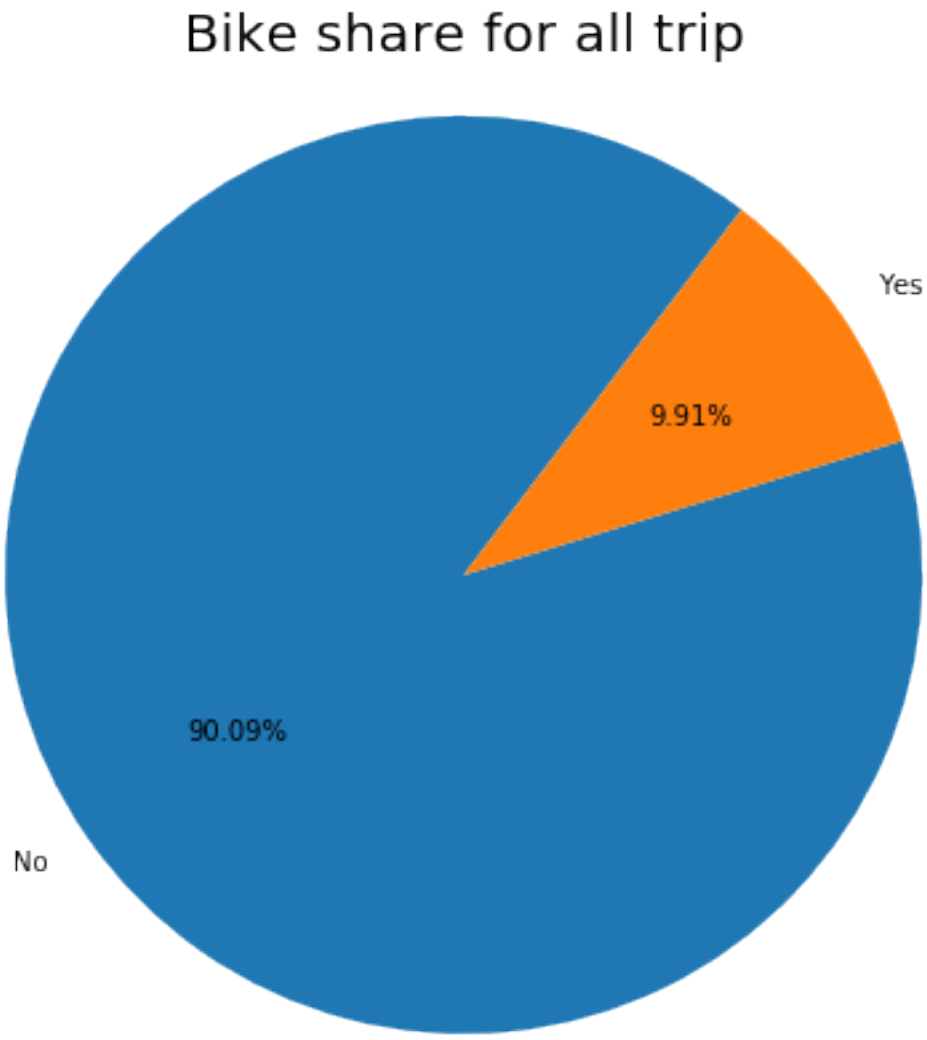



```
In [114]: figz(15,7)
          count_user = Bike_df_clean['user_type'].value_counts()

          plt.pie(count_user , labels = count_user.index ,startangle = 17, counterclock = False,
          plt.axis('square')
          plt.title('Count of user type', fontsize= 20);
```



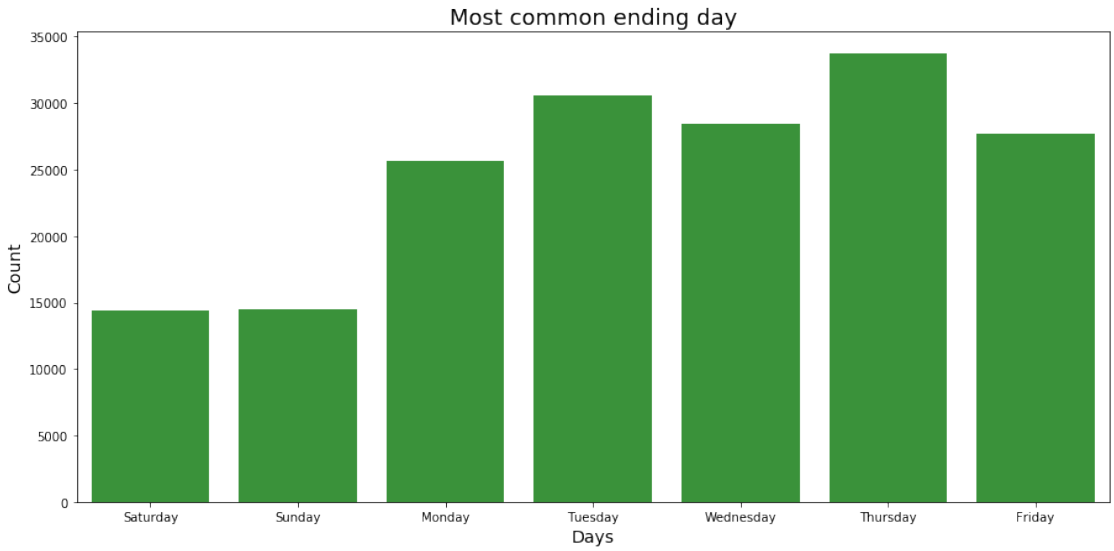
This cahrta displays that most customer were subscriber with about 90%.
Do bike share for all trip?

```
In [115]: figz(15,7)
          a = Bike_df_clean['bike_share_for_all_trip'].value_counts()

          plt.pie(a , labels = a.index ,startangle = 17, counterclock = False, autopct='%1.2f%%'
          plt.axis('square')
          plt.title('Bike share for all trip', fontsize= 20);
```



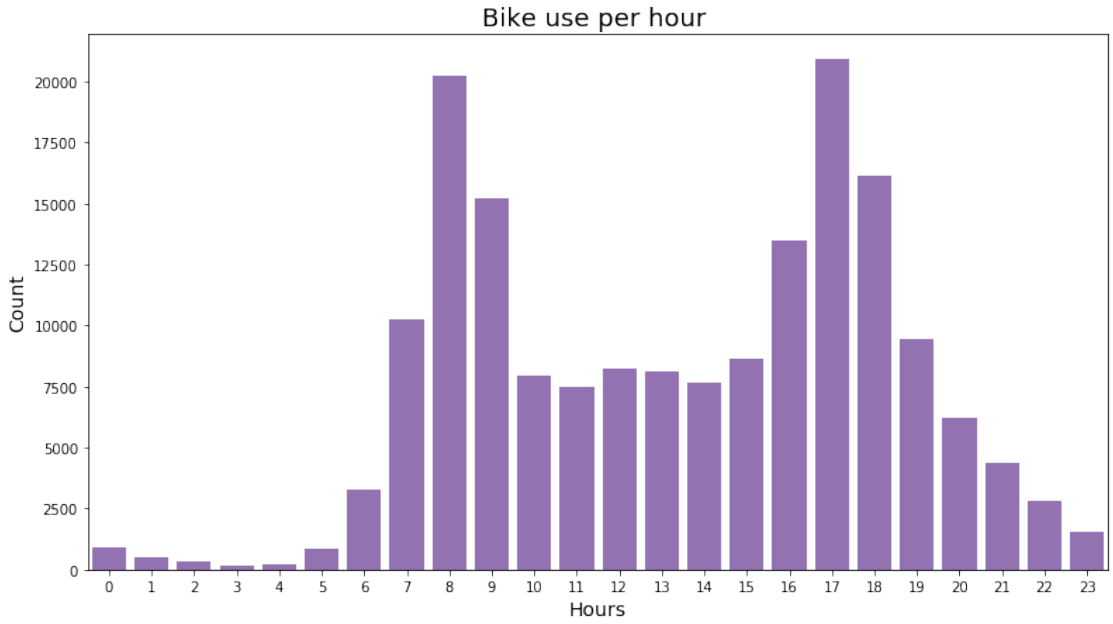
It shows that 90.09% which means the majorty of the users doesnt share.
What is the most common ending day ?

```
In [116]: figz(15,7)
          Green= sb.color_palette()[2]

          sb.countplot(data=Bike_df_clean, x='Ending_Day', order=days , color = Green);

          functionN('Days', 'Count','Most common ending day')
```



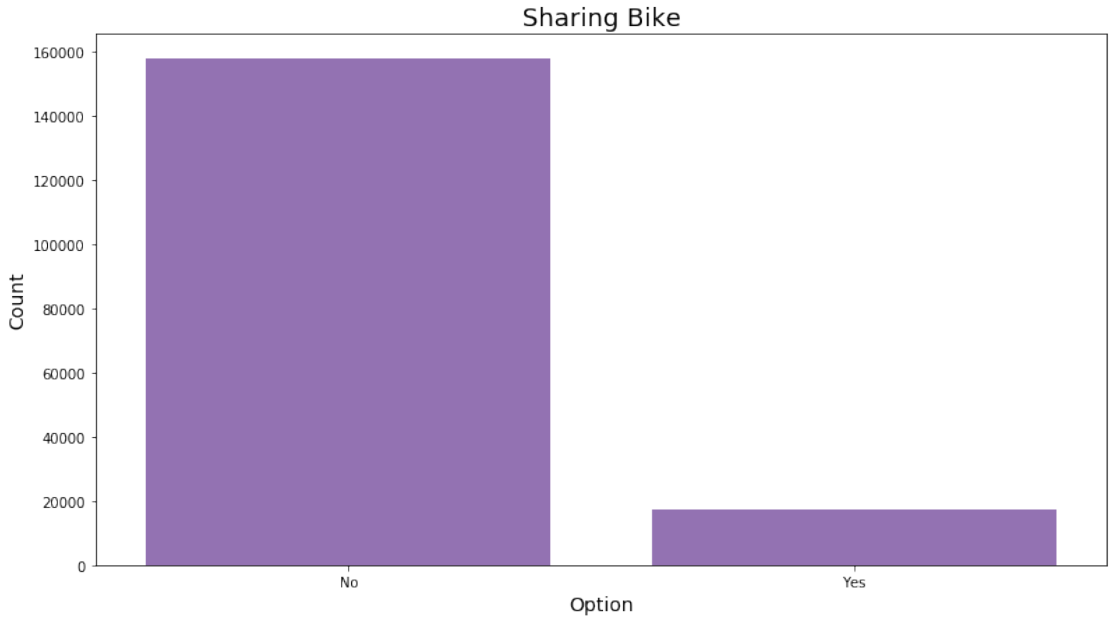
As shown it lead to be that Thursday is the most day to be ended.
Highest use per hour?

```
In [88]: figz(13,7)
         p = sb.color_palette()[4]
         sb.countplot(data = Bike_df_clean, x = 'Hours', color = p )

         functionN('Hours', 'Count','Bike use per hour ')
```



The up chart indicates that there will be an increase in use between 7 and 9 in the morning and 4 to 6 in the evening. I believe it is because of their daily journey to go and back from work .

```
In [118]: figz(13,7)
          sb.countplot(data = Bike_df_clean, x = 'bike_share_for_all_trip', color = p);

          functionN('Option', 'Count','Sharing Bike')
```



Most of them didnt share it.
What is the avrage age of the users ?

```
In [91]: figz(12,7)
         plt.hist(ages)

         functionN('Ages', 'Count','Distribution of the user ages')
```