Project_5

August 8, 2020

BayWheels Lyft Analysis

0.0.1 Data is downloaded from the official lyft website as a ZIP file

0.0.2 Importing libraries

[2]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
import folium
%matplotlib inline
```

[3]:
```
df=pd.read_csv("2017-fordgobike-tripdata.csv")
df.head()
```

[3]:

	duration_sec	start_time	end_time	start_station_id	start_station_name	start_station_latitude	start_station_longitude	end_station_id
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	43
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	96
2	45768	2017-12-31 22:45:48.4110	2018-01-01 11:28:36.8830	245	Downtown Berkeley BART	37.870348	-122.267764	245
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	5
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	247

```
                                 end_station_name  end_station_latitude  \
0  San Francisco Public Library (Grove St at Hyde…             37.778768
1                            Dolores St at 15th St             37.766210
2                           Downtown Berkeley BART             37.870348
3      Powell St BART Station (Market St at 5th St)             37.783899
4                          Fulton St at Bancroft Way             37.867789

   end_station_longitude  bike_id   user_type
0            -122.415929       96    Customer
1            -122.426614       88    Customer
2            -122.267764     1094    Customer
3            -122.408445     2831    Customer
4            -122.265896     3167  Subscriber
```

1 Assessing Data and Cleaning Data

[4]:
```
df.head()
```

[4]:
```
   duration_sec                start_time                  end_time  \
0         80110  2017-12-31 16:57:39.6540  2018-01-01 15:12:50.2450
1         78800  2017-12-31 15:56:34.8420  2018-01-01 13:49:55.6170
2         45768  2017-12-31 22:45:48.4110  2018-01-01 11:28:36.8830
3         62172  2017-12-31 17:31:10.6360  2018-01-01 10:47:23.5310
4         43603  2017-12-31 14:23:14.0010  2018-01-01 02:29:57.5710

   start_station_id                                start_station_name  \
0                74                             Laguna St at Hayes St
1               284  Yerba Buena Center for the Arts (Howard St at …
2               245                           Downtown Berkeley BART
3                60                             8th St at Ringold St
4               239                      Bancroft Way at Telegraph Ave

   start_station_latitude  start_station_longitude  end_station_id  \
0               37.776435              -122.426244              43
1               37.784872              -122.400876              96
2               37.870348              -122.267764             245
3               37.774520              -122.409449               5
4               37.868813              -122.258764             247

                                 end_station_name  end_station_latitude  \
0  San Francisco Public Library (Grove St at Hyde…             37.778768
1                            Dolores St at 15th St             37.766210
2                           Downtown Berkeley BART             37.870348
3      Powell St BART Station (Market St at 5th St)             37.783899
4                          Fulton St at Bancroft Way             37.867789
```

```
   end_station_longitude  bike_id   user_type
0            -122.415929       96    Customer
1            -122.426614       88    Customer
2            -122.267764     1094    Customer
3            -122.408445     2831    Customer
4            -122.265896     3167  Subscriber
```

[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
```

The `start_time` and `end_time` need to be casted to date and time instead of string to ease the use of them in the analysis

[6]:
```
df['end_time']=pd.to_datetime(df['end_time'])
df['start_time']=pd.to_datetime(df['start_time'])
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  datetime64[ns]
 2   end_time                 519700 non-null  datetime64[ns]
 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: datetime64[ns](2), float64(4), int64(4), object(3)
memory usage: 51.5+ MB
```

As we can see above there's no missing data in the columns , so we can start the analysis

2 Data Analysis

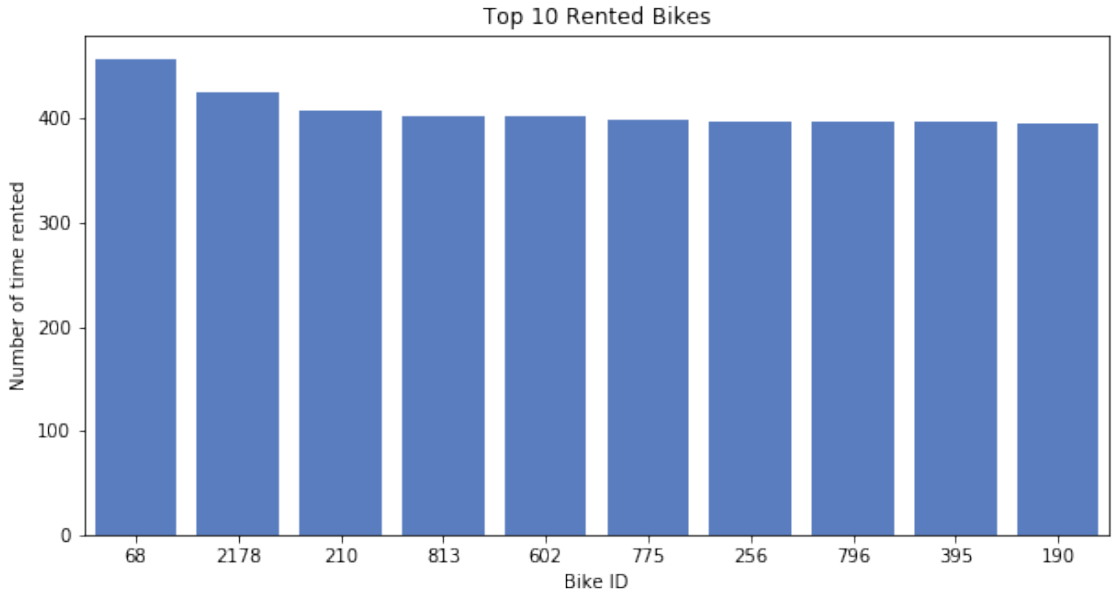
Since the bike IDs are unique to each bike, let's see how many bike does BayWheels has

[7]:
```
df.bike_id.nunique()
```

[7]:
```
3673
```

Let's see the top 10 rented bikes

[8]:
```
top_10_bikes=df.bike_id.value_counts()[0:10]
plt.figure(figsize=(10,5))
sns.barplot(x=top_10_bikes.index.values,y=top_10_bikes.values,color=sns.
 ↪color_palette('muted')[0],order=top_10_bikes.index.values);
plt.xlabel("Bike ID");
plt.ylabel("Number of time rented");
plt.title("Top 10 Rented Bikes");
```



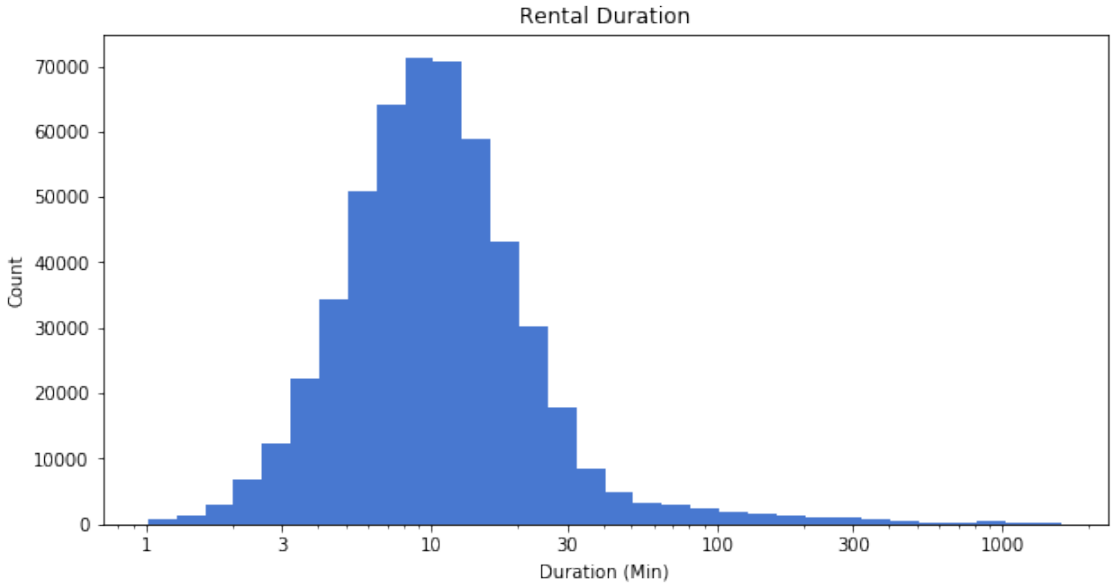
Let's check the distribution of the renteal duration in minutes

[9]:
```
duration_minutes=df.duration_sec/60
```

[ ]:
```

```

[10]:
```
plt.figure(figsize=(10,5))
bin_edges = 10 ** np.arange(np.log10(duration_minutes.min()), np.
 ↪log10(duration_minutes.max())+0.1, 0.1)
plt.hist(duration_minutes, bins = bin_edges,color=sns.
 ↪color_palette('muted')[0]);
plt.xscale('log');
tick_locs = [1,3,10, 30, 100, 300, 1000];
plt.xticks(tick_locs, tick_locs);
plt.xlabel("Duration (Min)");
plt.ylabel("Count");
plt.title("Rental Duration");
```

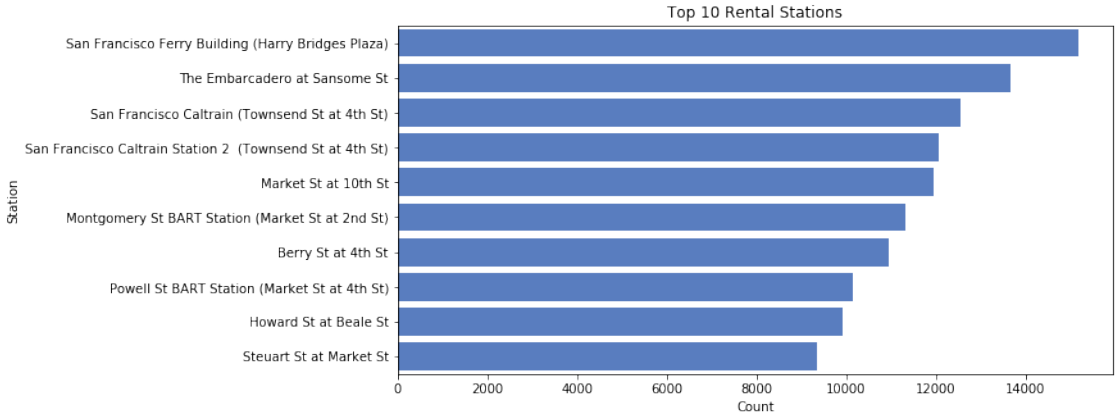


We can see from the above plot that the rental duration median is around 10 minutes and it's normaly distributed with many outliers

Lets check Top 10 stations which have the most rentals

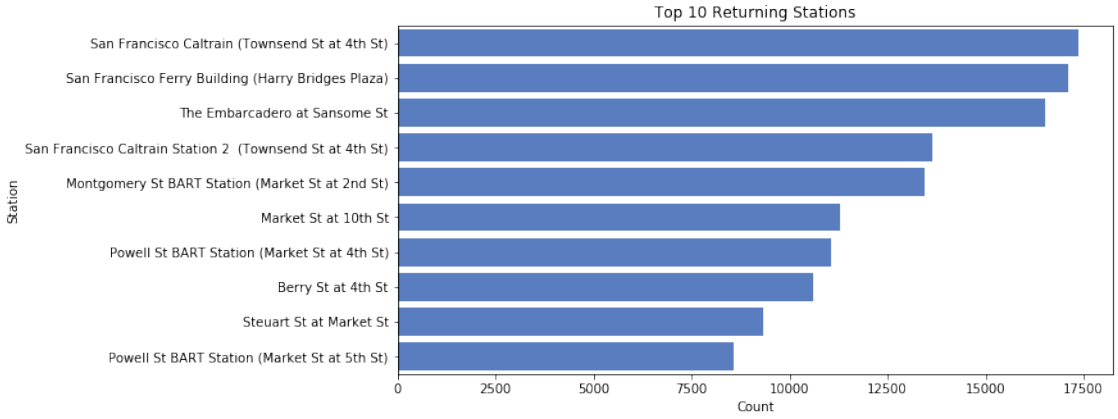
[11]:
```
top_10_stations=df[df.start_station_name.isin(df.start_station_name.
 ↪value_counts()[0:10].index)].start_station_name
plt.figure(figsize=(10,5))
```

```
sns.countplot(y=top_10_stations,color=sns.
 ↪color_palette('muted')[0],order=top_10_stations.value_counts().index.values);
plt.ylabel("Station");
plt.xlabel("Count");
plt.title("Top 10 Rental Stations");
```



Lets check Top 10 returning stations

[12]:
```
top_10_stations_return=df[df.end_station_name.isin(df.end_station_name.
 ↪value_counts()[0:10].index)].end_station_name
plt.figure(figsize=(10,5))
sns.countplot(y=top_10_stations_return,color=sns.
 ↪color_palette('muted')[0],order=top_10_stations_return.value_counts().index.
 ↪values);
plt.ylabel("Station");
plt.xlabel("Count");
plt.title("Top 10 Returning Stations");
```



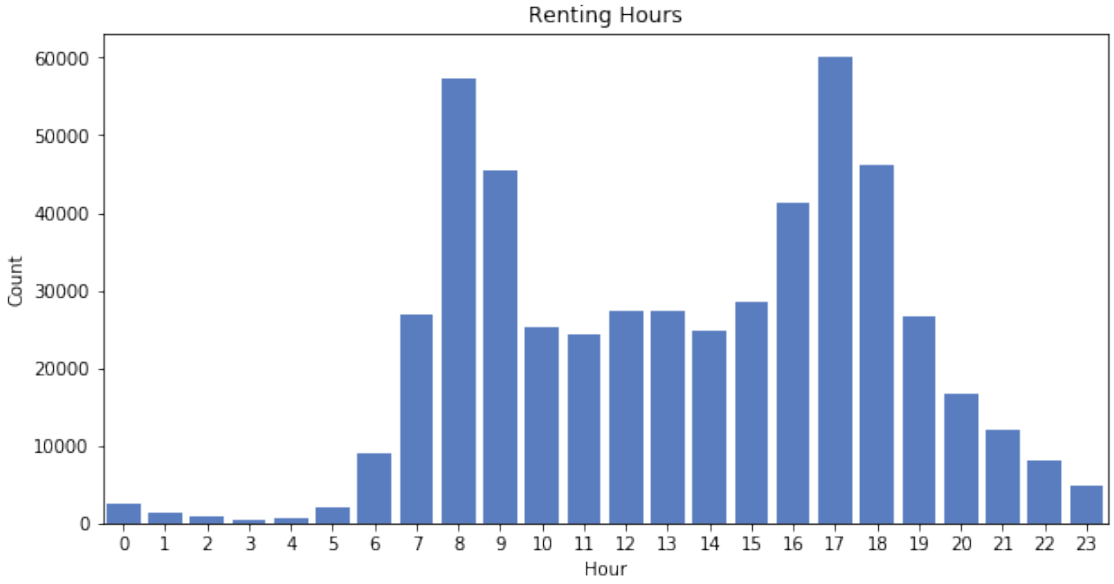
Let's see the renting hours

[13]:
```
df['start_hour']=df.start_time.apply(lambda x:x.hour)
```

[14]:
```
labels=["Night",'Morning',"After Noon","Evening"]
bins=[0,6,12,17,23]
```

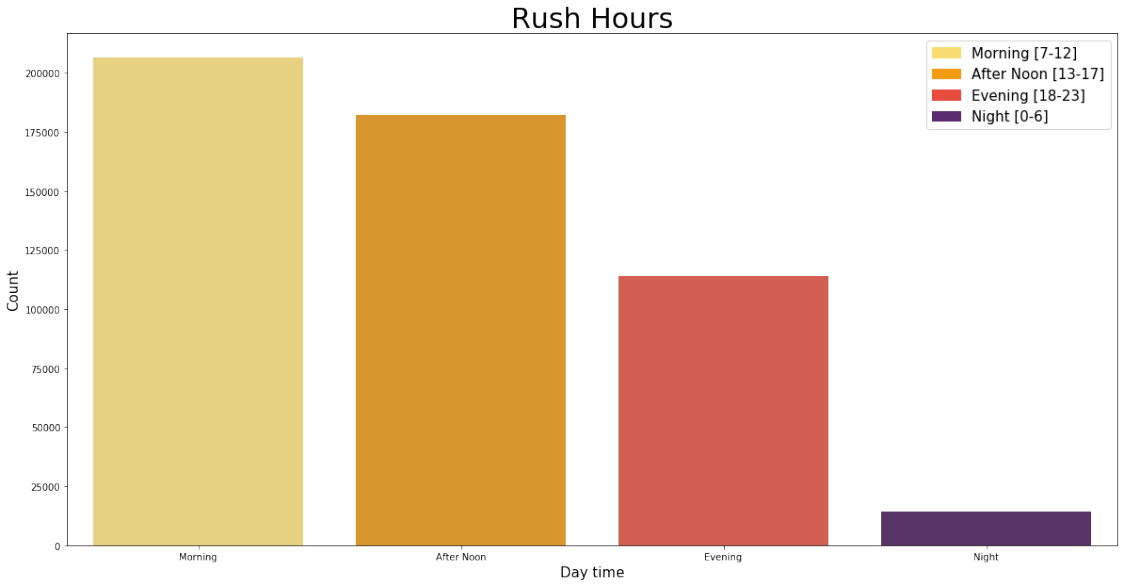
[15]:
```
df['start_hour_labeld']=pd.cut(df.start_hour,bins=bins,labels=labels)
```

[16]:
```
plt.figure(figsize=(10,5))
sns.countplot(df.start_hour,color=sns.color_palette('muted')[0],);
plt.ylabel("Count");
plt.xlabel("Hour");
plt.title("Renting Hours");
```



[17]:
```
colors=["#f7dc6f","#F39C12","#E74C3C","#5B2C6F"]
customPalette = sns.set_palette(sns.color_palette(colors))
plt.figure(figsize=(20,10))
ax=sns.countplot(df.
 ↪start_hour_labeld,palette=customPalette,order=['Morning',"After␣
 ↪Noon","Evening","Night"]);
plt.ylabel("Count",fontsize=15);
plt.xlabel("Day time",fontsize=15);
plt.title("Rush Hours",fontsize=30);
Night = mpatches.Patch(color=sns.color_palette(colors)[3], label='Night [0-6]')
Morning = mpatches.Patch(color=sns.color_palette(colors)[0], label='Morning␣
 ↪[7-12]')
```

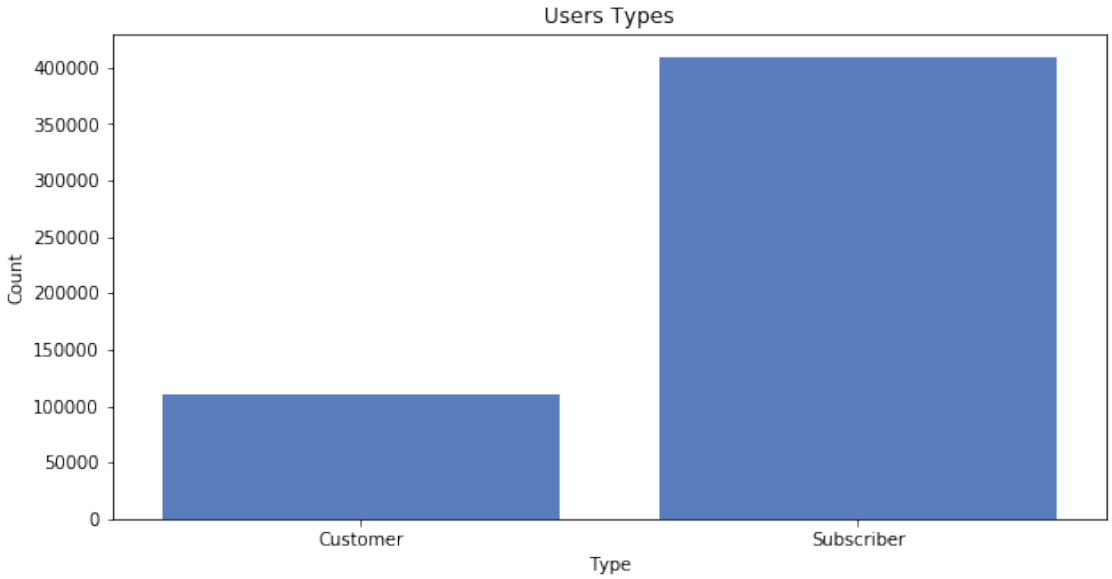
```
After_Noon = mpatches.Patch(color=sns.color_palette(colors)[1], label='After␣
 ↪Noon [13-17]')
Evening = mpatches.Patch(color=sns.color_palette(colors)[2], label='Evening␣
 ↪[18-23]')
plt.legend(handles=[Morning,After_Noon,Evening,Night],fontsize=15);
```



As we can see in the above plot that the rush hour for renting is 8:00 AM and

Let's see the different types of users

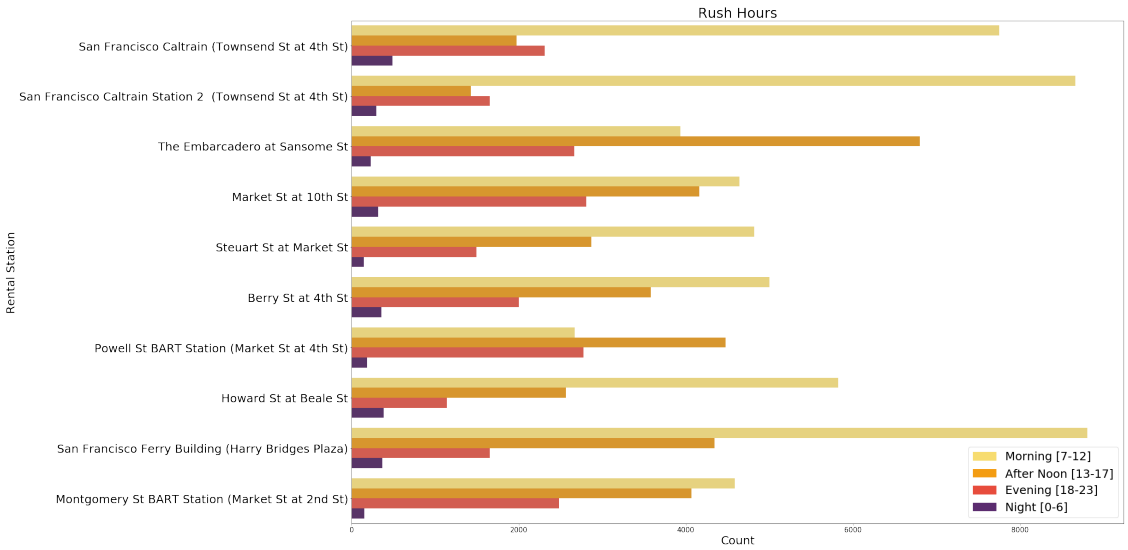
[18]:
```
plt.figure(figsize=(10,5))
sns.countplot(df.user_type,color=sns.color_palette('muted')[0])
plt.ylabel("Count");
plt.xlabel("Type");
plt.title("Users Types");
```



2.1 Do the top 10 renting stations have different rush hours ?

[19]:
```
top_10_stations_df=df[df.start_station_name.isin(df.start_station_name.
↪value_counts()[0:10].index)]
```

[20]:
```
colors=["#f7dc6f","#F39C12","#E74C3C","#5B2C6F"]
customPalette = sns.set_palette(sns.color_palette(colors))
plt.figure(figsize=(30,20))
ax=sns.countplot(y=top_10_stations_df.start_station_name,palette=customPalette,
                 hue_order=['Morning',"After Noon","Evening","Night"],
                 hue=top_10_stations_df.start_hour_labeld);
plt.ylabel("Rental Station",fontsize=25);
plt.xlabel("Count",fontsize=25);
plt.title("Rush Hours",fontsize=30);
plt.xticks(fontsize=15)
plt.yticks(fontsize=25)
Night = mpatches.Patch(color=sns.color_palette(colors)[3], label='Night [0-6]')
Morning = mpatches.Patch(color=sns.color_palette(colors)[0], label='Morning␣
↪[7-12]')
After_Noon = mpatches.Patch(color=sns.color_palette(colors)[1], label='After␣
↪Noon [13-17]')
Evening = mpatches.Patch(color=sns.color_palette(colors)[2], label='Evening␣
↪[18-23]')
plt.legend(handles=[Morning,After_Noon,Evening,Night],fontsize=25);
```

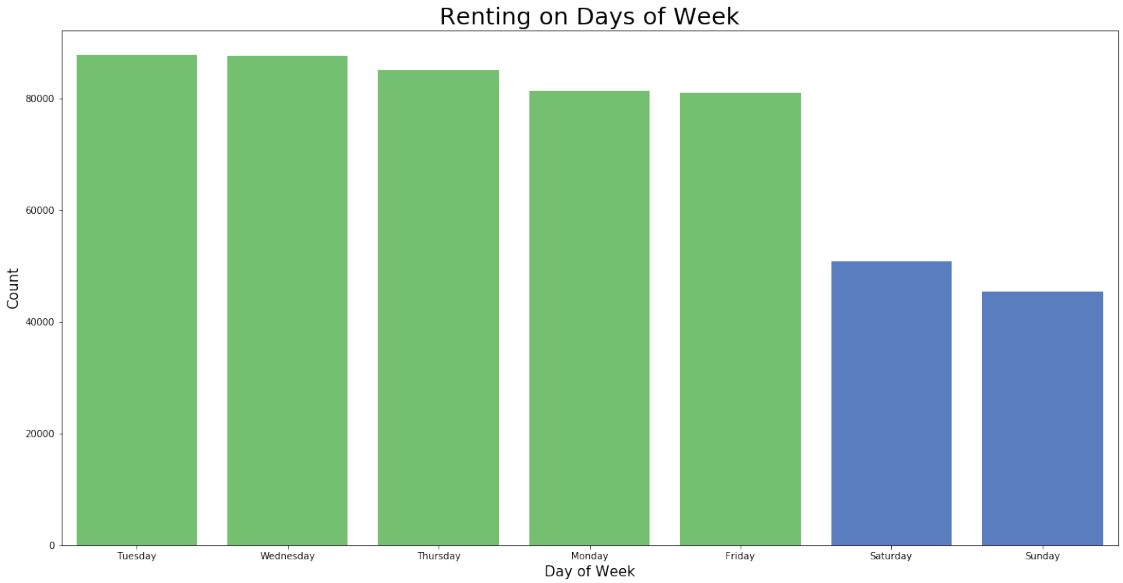


As we can see above both `Powel St BART Station` and `The Embarcadero at Sansome St` have rush hours at the after noon rather than the morning

2.2 Which days has more renting ?

[21]:
```
pallet=[]
for i in range(7):
    if i < 2:
        pallet.append(sns.color_palette('muted')[0])
    else:
        pallet.append(sns.color_palette('muted')[2])
pallet.reverse()
```

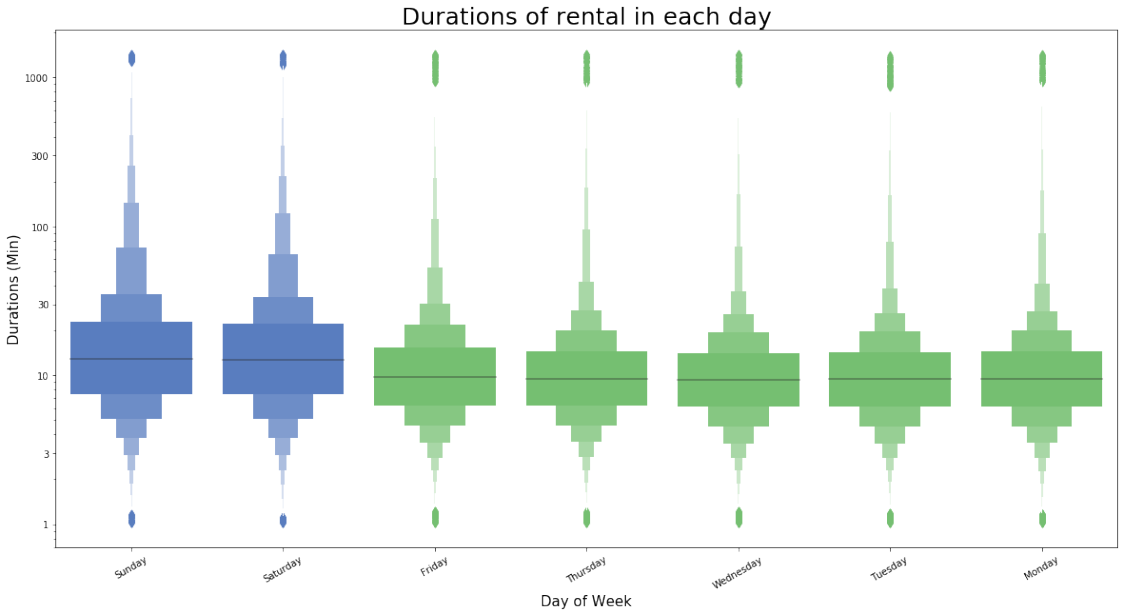
[22]:
```
plt.figure(figsize=(20,10))
sns.countplot(df.start_time.dt.day_name(),
              order=df.start_time.dt.day_name().value_counts().index,
              palette=pallet);
plt.xlabel("Day of Week",fontsize=15);
plt.ylabel("Count",fontsize=15);
plt.title("Renting on Days of Week",fontsize=25);
```



We can see the huge difference in Saturdays and Sundays, they're much lower than the rest of the week

2.3 Do people rent more time on specific days ?

[23]:
```
pallet.reverse()
plt.figure(figsize=(20,10));
ax=sns.boxenplot(x=df.start_time.dt.day_name(),
                 y=duration_minutes,
                palette=pallet);
plt.xticks(rotation=30);
plt.yscale('log');
plt.yticks(tick_locs,tick_locs);
plt.ylabel("Durations (Min)",fontsize=15);
plt.xlabel("Day of Week",fontsize=15);
plt.title("Durations of rental in each day",fontsize=25);
```

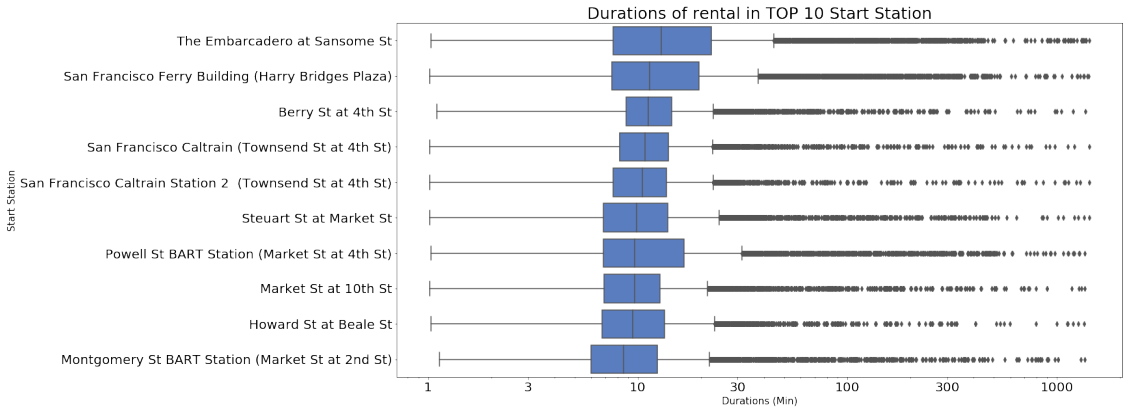


As we can see on Sundays and Saturday the median is more than the rest of the week , which means that people rents in more duration on those days

We can conclude from the previous two plots , that people tend to rent less on weekends , but when they do , they rent for a long time

2.4 Does Start Station affects the renting duration?

[24]:
```
top_10_stations_df=top_10_stations_df.copy()
top_10_stations_df['duration_min']=top_10_stations_df.duration_sec/60
my_order=top_10_stations_df.groupby(by=["start_station_name"])["duration_min"].
↪median().sort_values(ascending=False).index
plt.figure(figsize=(20,10))
ax=sns.boxplot(y=top_10_stations_df.start_station_name,
               x=top_10_stations_df.duration_min,
               order=my_order,
               color=sns.color_palette('muted')[0]);
plt.xscale('log');
plt.xticks(tick_locs,tick_locs,fontsize=20);
plt.xlabel("Durations (Min)",fontsize=15);
plt.ylabel("Start Station",fontsize=15);
plt.yticks(fontsize=20)
plt.title("Durations of rental in TOP 10 Start Station",fontsize=25);
```

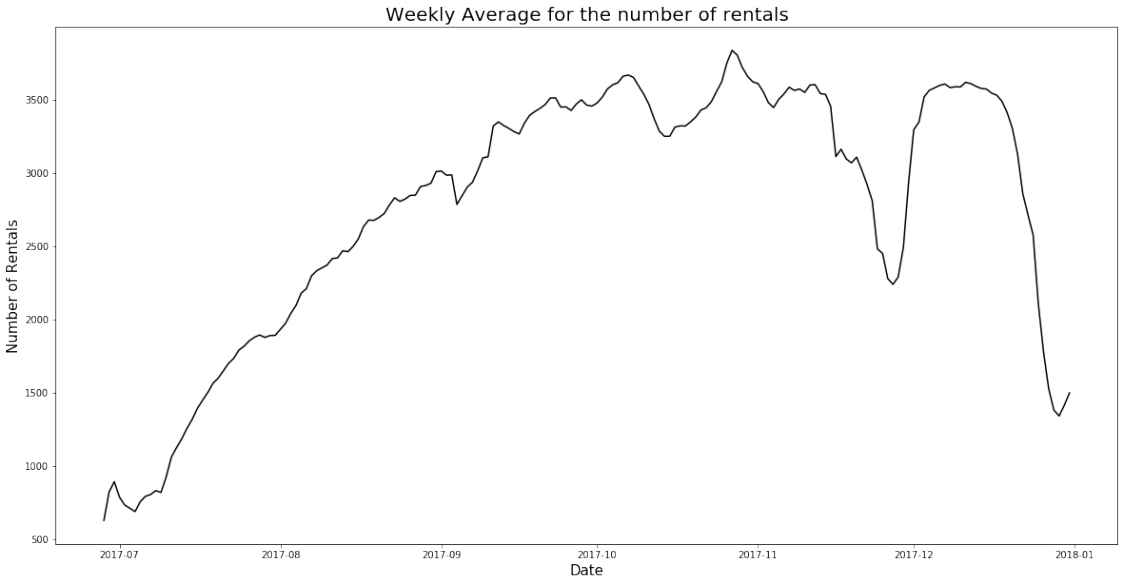


As we can see that "The Embarcadero at Sansome St" has a the biggest median and interquantile range , which means it has more rental duration than the other stations

2.5 Let's Check the weekly sales(Rentals)

[25]:
```
rentals_by_date=df.groupby(df.start_time.dt.date)['duration_sec'].count().
 ↪to_frame()
rentals_by_date_rolled=rentals_by_date.rolling(7,min_periods=1).mean()
rentals_by_date.rename(columns={'start_time':'date','duration_sec':
 ↪'rentals'},inplace=True)
rentals_by_date_rolled.reset_index(inplace=True)
rentals_by_date_rolled.rename(columns={'start_time':'date','duration_sec':
 ↪'rentals'},inplace=True)
```

[26]:
```
plt.figure(figsize=(20,10))
sns.lineplot(x='date',y='rentals',data=rentals_by_date_rolled,color='black');
plt.xlabel("Date",fontsize=15);
plt.ylabel("Number of Rentals",fontsize=15);
plt.title("Weekly Average for the number of rentals",fontsize=20);
```



We can see a huge decline at December, the reason is the holidays , Because people prefer staying with families rather than cycling

2.6 Let's see the distribution of the rentals on the map

[27]:
```
df.shape[0]
```

[27]: 519700

Since the data is barely half a milion , we should only use sample the first 100,000 and use them to minimize the computational cost

[28]:
```
df_sampled=df.sample(100000)
```

[29]:
```
df_sampled.shape[0]
```

[29]: 100000

[30]:
```
latitude = 37.77
longitude = -122.42
```

[31]:
```
from folium import plugins

sanfran_map = folium.Map(location = [latitude, longitude], zoom_start = 10)

incidents = plugins.MarkerCluster().add_to(sanfran_map)
```

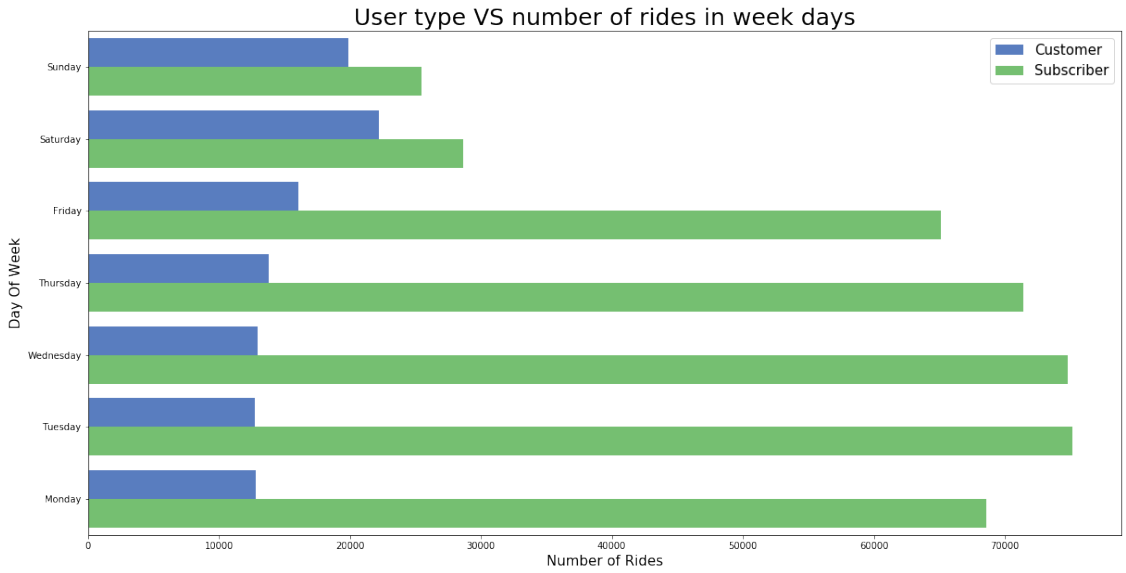
```
for lat, lng, label, in zip(df_sampled.start_station_latitude, df_sampled.
 ↪start_station_longitude, df_sampled.start_station_name):
    folium.Marker(
        location=[lat, lng],
        icon=None,
        popup=label,
    ).add_to(incidents)

sanfran_map
```

[31]: <folium.folium.Map at 0x7f32dc284050>

2.7 Let's see relationship between the user type and the number of rides and the week of day

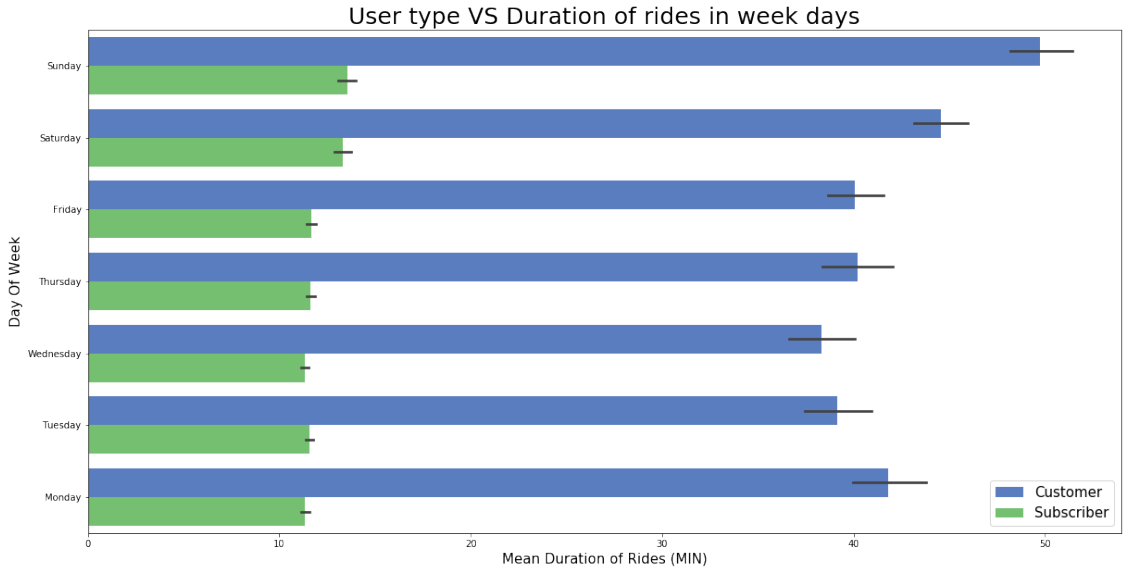
[32]:
```
plt.figure(figsize=(20,10))
sns.countplot(y=df.start_time.dt.day_name(),hue=df.user_type,palette=sns.
 ↪color_palette('muted')[0:3:2]);
plt.xlabel("Number of Rides",fontsize=15)
plt.ylabel("Day Of Week",fontsize=15)
plt.legend(fontsize=15)
plt.title("User type VS number of rides in week days",fontsize=25);
```



We can see from the above plots that the number of customers increase on Sundays and Saturdays

2.8 Is there any special days the cutomers ride more than subscribers?

[33]:
```
duration_minutes=df.duration_sec/60
plt.figure(figsize=(20,10))
sns.barplot(x=duration_minutes,y=df.start_time.dt.day_name(),hue=df.
 ↪user_type,palette=sns.color_palette('muted')[0:3:2]);
plt.xlabel("Mean Duration of Rides (MIN)",fontsize=15)
plt.ylabel("Day Of Week",fontsize=15)
plt.legend(fontsize=15)
plt.title("User type VS Duration of rides in week days",fontsize=25);
```



We can conclude from the above two plots that subscribers make a huge number of rides but with a small duration compared to the Customers We can say that the subscribers renting the bicycle for a certain reason that's why thier mean is approximatly the same, in contrast , the costumers are just having rides..