Project Development Phase Sprint 3

|  |  |
| --- | --- |
| Date | 07 November 2022 |
| Team ID | PNT2022TMID22202 |
| Team Lead | Santhosh kumar.V |
| Team Members | Sharan R  Yogesh S  Yuvaganesh GK |
| Project Name | A New hint to transportation – Analysis of the  NYC bike share system. |
| Maximum Marks | 20 Marks |

**Creating a dashboard including all the visualizations created in the cognos platform:**

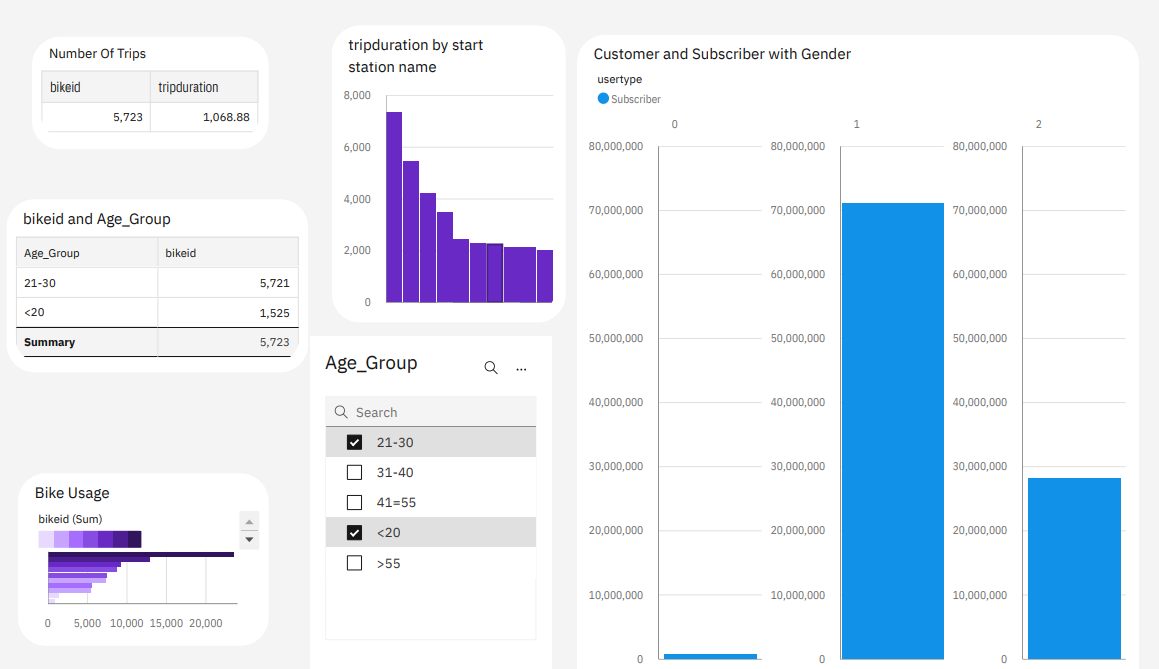
**This dashboard has the charts including**

# Number of trips

1. **Customer and Subscriber percentage with gender**

# Bike Usage

1. **BikeId and Age Group**
2. **Trip duration by start station name**



**Visualization Charts using Python:**

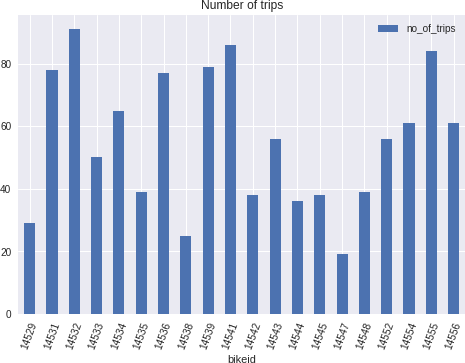
**Finding the number of trips per bike:**

trips = pd.DataFrame() #creating a dataframe

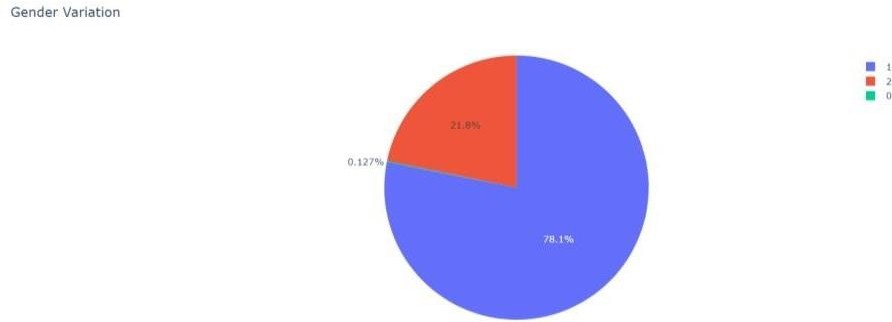
trips['no\_of\_trips'] = df.groupby("bikeid")["bikeid"].count() #finding the number of trips by each bike trips['avg\_duration'] = df.groupby("bikeid")["tripduration"].mean() #avg duration of the trips

trips\_graph=trips.head(20)

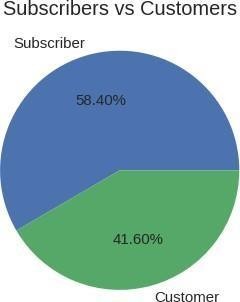
trips\_graph.plot.bar(x="bikeid", y="no\_of\_trips", rot=70, title="Number of trips")



# Gender Variation:

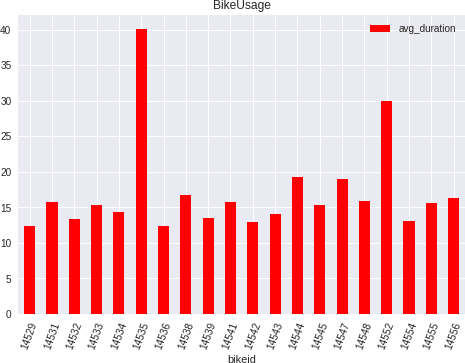
plt.pie(values = df\_bike['Gender'].value\_counts(), names =df\_bike['Gender'].value\_counts().index, title ="Gender Variation")

# Percentage of Subscribers and Customers:



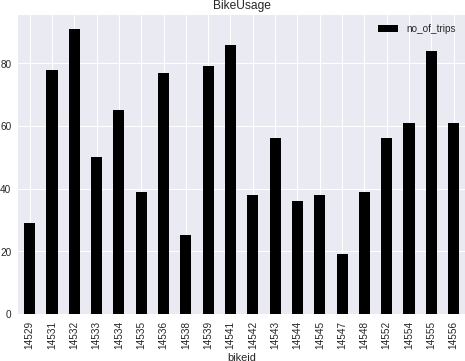
**Bike Usage Based on Average Duration:**

trips\_graph.plot.bar(x="bikeid", y="avg\_duration", rot=70, title="BikeUsage",color="red")

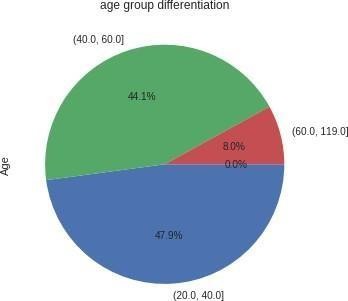


# Bike Usage Based on No of Trips:

trips\_graph.plot.bar(x="bikeid", y="no\_of\_trips", rot=90, title="BikeUsage",color="black")



# Age Group Differentiation:

agegroup = pd.cut(df['Age'], bins=bins).value\_counts() agegroup.plot.pie(autopct="%.1f%%",title='age group differentiation',counterclock=False);

# Top 10 Start Station:

most=pd.DataFrame() most\_graph=pd.DataFrame()

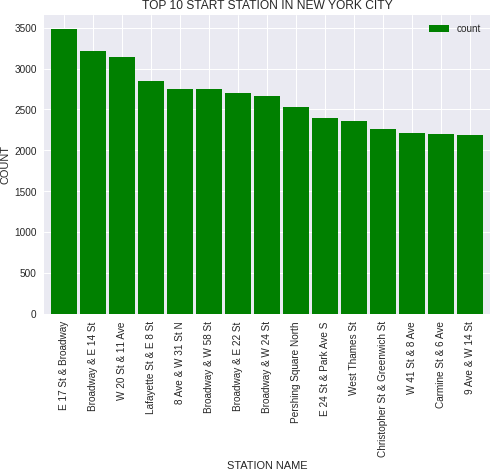
most['name']=df["start station name"].value\_counts().index most['count']=df["start station name"].value\_counts().values most\_graph=most.head(15)

most\_graph.plot.bar(x="name", y="count", width=0.9,rot=90, title="BikeUsage",color="green") plt.xlabel("STATION NAME")

plt.ylabel("COUNT")

plt.title("TOP 10 START STATION IN NEW YORK CITY")

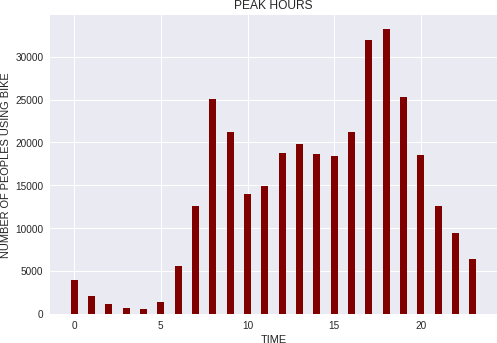
plt.show()



**Finding the Peak Hours of Travel:** ind=peak\_hour["Hour"].value\_counts().index y=peak\_hour["Hour"].value\_counts().values plt.bar(ind, y, color ='maroon', width = 0.4) plt.xlabel("TIME")

plt.ylabel("NUMBER OF PEOPLES USING BIKE")

plt.title("PEAK HOURS") plt.show()



# Bike Trend for the month June:

#converting string to datetime object df['starttime']= pd.to\_datetime(df['starttime'])

#since we are dealing with single month, we grouping by days

#using count aggregation to get number of occurances i.e, total trips per day start\_time\_count = df.set\_index('starttime').groupby(pd.Grouper(freq='D')).count()

#we have data from July month for only one day which is at last row, lets drop it start\_time\_count.drop(start\_time\_count.tail(1).index, axis=0, inplace=True)

#again grouping by day and aggregating with sum to get total trip duration per day #which will used while plotting

trip\_duration\_count = df.set\_index('starttime').groupby(pd.Grouper(freq='D')).sum()

#again dropping the last row for same reason trip\_duration\_count.drop(trip\_duration\_count.tail(1).index, axis=0, inplace=True)

#plotting total rides per day

#using start station id to get the count fig,ax=plt.subplots(figsize=(25,10))

ax.bar(start\_time\_count.index, 'start station id', data=start\_time\_count, label='Total riders') #bbox\_to\_anchor is to position the legend box

ax.legend(loc ="lower left", bbox\_to\_anchor=(0.01, 0.89), fontsize='20') ax.set\_xlabel('Days of the month June 2018', fontsize=30) ax.set\_ylabel('Riders', fontsize=40)

ax.set\_title('Bikers trend for the month June', fontsize=50)

#creating twin x axis to plot line chart is same figure ax2=ax.twinx()

#plotting total trip duration of all user per day

ax2.plot('tripduration', data=trip\_duration\_count, color='y', label='Total trip duration', marker='o', line width=5, markersize=12)

ax2.set\_ylabel('Time duration', fontsize=40)

ax2.legend(loc ="upper left", bbox\_to\_anchor=(0.01, 0.9), fontsize='20')

ax.set\_xticks(trip\_duration\_count.index) ax.set\_xticklabels([i for i in range(1,31)])

#tweeking x and y ticks labels of axes1 ax.tick\_params(labelsize=30, labelcolor='#eb4034') #tweeking x and y ticks labels of axes2 ax2.tick\_params(labelsize=30, labelcolor='#eb4034')

plt.show()



# Least Used End Stations:

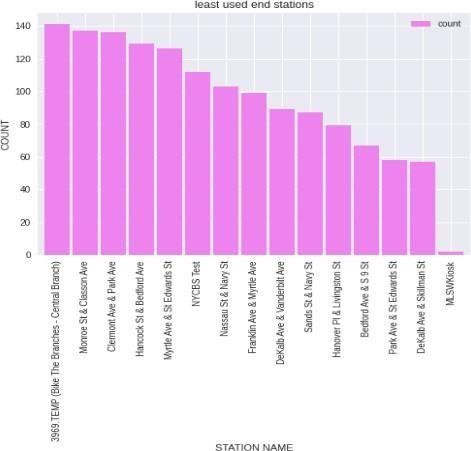
least=pd.DataFrame() least\_graph=pd.DataFrame()

least['name']=df["end station name"].value\_counts().index least['count']=df["end station name"].value\_counts().values least\_graph=most.tail(15)

least\_graph

least\_graph.plot.bar(x="name", y="count", width=0.9,rot=90, title="BikeUsage",color="violet") plt.xlabel("STATION NAME")

plt.ylabel("COUNT") plt.title("least used end stations") plt.show()



# Same start and end location Vs Different start and end location:

#number of trips that started and ended at same station

start\_end\_same = df[df['start station name'] == df['end station name']].shape[0]

#number of trips that started and ended at different station start\_end\_diff = df.shape[0]-start\_end\_same

fig,ax=plt.subplots()

ax.pie([start\_end\_same,start\_end\_diff], labels=['Same', 'Different'], autopct='%1.2f%%', textprops={'f ontsize': 20})

ax.set\_title('Same start and end location vs Different start and end location', fontsize=20)

circle = Circle((0,0), 0.6, facecolor='white') ax.add\_artist(circle)

plt.show()

