**CERTIFICATION PROJECT DATA ANALYSIS AND VISUALIZATION**

**import** pandas **as** pd  
**import** numpy **as** np  
**import** csv  
**import** folium **as** fm  
**import** matplotlib.pyplot **as** plt  
**import** seaborn **as** sns  
*#Q.1Compute --What are the top 10 Zipcodes for 911 & Question 1: Are Zipcodes 19446 and 19090 present ?*df=pd.read\_csv(**'911.csv'**)  
df=df.dropna()  
*#print(df)*df1=df[**'zip'**].value\_counts()  
df1=df1.head(10)  
df1.to\_csv(**'1.csv'**)  
**with** open(**'1.csv'**,newline=**''**) **as** f:  
 r = csv.reader(f)  
 data = [line **for** line **in** r]  
**with** open(**'1.csv'**,**'w'**,newline=**''**) **as** f:  
 w = csv.writer(f)  
 w.writerow([**'zip'**,**'frequency'**])  
 w.writerows(data)  
df2=pd.read\_csv(**'1.csv'**)  
print(df2)  
  
a1=list(df2[**'zip'**])  
a2=list(df2[**'frequency'**])  
x=19090  
y=19446  
*#print(df2.isin([19090,19446]))***if** (x **in** a1) & (y **in** a1):  
 print(**"Zipcodes are present in top 10"**)  
**else**:  
 print(**"Zipcodes are not found"**)  
  
*#Q.2Compute --What are the top 4 townships (twp) for 911 calls  
# & Question 2: Which of the following township are not present? --LOWER POTTSGROVE, NORRISTOWN, HORSHAM, ABINGTON*df3=df[**'twp'**].value\_counts()  
df3=df3.head(4)  
*#print(df3)*df3.to\_csv(**'2.csv'**)  
**with** open(**'2.csv'**,newline=**''**) **as** f1:  
 r1 = csv.reader(f1)  
 data1 = [line1 **for** line1 **in** r1]  
**with** open(**'2.csv'**,**'w'**,newline=**''**) **as** f1:  
 w1 = csv.writer(f1)  
 w1.writerow([**'Town'**,**'frequency'**])  
 w1.writerows(data1)  
df4=pd.read\_csv(**'2.csv'**)  
print(df4)  
  
b1=list(df4[**'Town'**])  
b2=list(df4[**'frequency'**])  
x1=**"LOWER POTTSGROVE"**x2=**"NORRISTOWN"**x3=**"HORSHAM"**x4=**"ABINGTON"  
if**(x1 **in** b1):  
 print(x1,**"is in top four towns"**)  
**else**:  
 print(x1,**"is not in top four towns"**)  
**if**(x2 **in** b1):  
 print(x2,**"in in top four towns"**)  
**else**:  
 print(x2,**"is not in top four towns"**)  
**if**(x3 **in** b1):  
 print(x3,**"is in top four towns"**)  
**else**:  
 print(x3,**"is not in top four towns"**)  
**if**(x4 **in** b1):  
 print(x4,**"is in top four towns"**)  
**else**:  
 print(x4,**"is not in top four towns"**)  
  
*#Q.3Compute --Create new features & Question 3: What is the most common Reason for a 911 call based on Reason Column? Which comes second*df[[**'Reason'**, **'Title'**]] = df.title.str.split(**': '**, expand = **True**)  
df.to\_csv(**'3.csv'**,index=**False**)  
*#print(df)*df5=df[**'Reason'**].value\_counts()  
print(**'The 2 most common reasons for calling 911 are'**,df5.head(2))  
df5.to\_csv(**'4.csv'**)  
**with** open(**'4.csv'**,newline=**''**) **as** f2:  
 r2 = csv.reader(f2)  
 data2 = [line2 **for** line2 **in** r2]  
**with** open(**'4.csv'**,**'w'**,newline=**''**) **as** f2:  
 w2 = csv.writer(f2)  
 w2.writerow([**'Reason'**,**'frequency'**])  
 w2.writerows(data2)  
  
df6=pd.read\_csv(**'4.csv'**,index\_col=**False**)  
print(df6)  
  
  
*#Compute --Plot barchart using matplot for 911 calls by Reason & Question 4: How can you plot the bars horizontally ?*objects=tuple(df6[**'Reason'**])  
y\_pos = np.arange(len(objects))  
frequency1=list(df6[**'frequency'**])  
  
plt.bar(y\_pos, frequency1, align=**'center'**)  
plt.barh(y\_pos,frequency1,align=**'center'**)  
plt.xticks(y\_pos, objects)  
plt.ylabel(**'FRequency'**)  
plt.title(**'Statistics'**)  
plt.show()  
  
*#Do data manipulation & Question 5: Which day got maximum calls for EMS and how many?*df[**'timeStamp'**]=pd.to\_datetime(df[**'timeStamp'**])  
*#df.to\_csv('5.csv')  
#df7=pd.read\_csv('5.csv')  
#df[['Date', 'Time']] = df.timeStamp.str.split(' ', expand = True)*df[**'day\_of\_week'**] = df[**'timeStamp'**].dt.day\_name()  
df.to\_csv(**'5.csv'**)  
df7=df[**'day\_of\_week'**].value\_counts()  
print(**"the maximum call for ems was on"**,df7.head(1))  
  
*#Compute --Create a countplot of the Day of Week column with the hue based of the Reason column & Question 6: On which day traffic calls were lowest ?*sns.set(style=**"darkgrid"**)  
sns.countplot(x=**'day\_of\_week'**, hue=**'Reason'**, data=df)  
plt.legend(bbox\_to\_anchor=(1,1))  
plt.show()  
  
*#Compute --Create a countplot month wise --Question 7: Which month saw highest calls for fire?*df[**'Month'**] = df[**'timeStamp'**].apply(**lambda** time:time.month)  
sns.countplot(x=**'Month'**, hue=**'Reason'**, data=df)  
plt.legend(bbox\_to\_anchor=(1,1))  
plt.show()  
  
*#Compute --Create Web Map for Traffic Calls & Question 8: Why some areas seem tohave lower or almost zero traffic calls? Hint: Zoom the map*SF\_COORDINATES = (40.2978759,-75.5812935)  
mdata = pd.read\_csv(**'911.csv'**)  
map = fm.Map(location=SF\_COORDINATES, zoom\_start=12,tiles=**"OpenStreetMap"**)  
fg = fm.FeatureGroup(name=**"Locations"**)  
**for** data **in** mdata.iterrows():  
 fg.add\_child(fm.Marker(location=[data[1][**'lng'**], data[1][**'lat'**] ],icon=fm.Icon(color=**"red"**)))  
map.add\_child(fg)  
map.save(**"BasicWebMap.html"**)  
print(map)