```
import pandas as pd
df=pd.read_csv("/content/tmpempmaizr.csv",skipinitialspace = True)
df
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DES
0	232005137	2670	NaN	HARA (HAR/
1	232000017	1831	NaN	SIC
2	232000036	3802	NaN	M/V AC PROPERTY
3	232000034	801	NaN	ASSAULT
4	232000041	801	NaN	ASSAULT
...	...	...	...	
3847	232005597	3115	NaN	INVE
3848	232005598	3115	NaN	INVE
3849	232005595	3831	NaN	M/V - SCENE - PI

```
import warnings
warnings.filterwarnings("ignore")
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3852 entries, 0 to 3851
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   INCIDENT_NUMBER      3852 non-null   int64  
 1   OFFENSE_CODE         3852 non-null   int64  
 2   OFFENSE_CODE_GROUP   0 non-null      float64
 3   OFFENSE_DESCRIPTION  3852 non-null   object 
 4   DISTRICT             3839 non-null   object 
 5   REPORTING_AREA       1779 non-null   float64
 6   SHOOTING             3852 non-null   int64  
 7   OCCURRED_ON_DATE     3852 non-null   object 
 8   YEAR                 3852 non-null   int64  
 9   MONTH                3852 non-null   int64  
 10  DAY_OF_WEEK          3852 non-null   object 
 11  HOUR                 3852 non-null   int64  
 12  UCR_PART             0 non-null      float64
 13  STREET               3852 non-null   object 
 14  Lat                  3561 non-null   float64
 15  Long                 3561 non-null   float64
 16  Location             3561 non-null   object 
dtypes: float64(5), int64(6), object(6)
memory usage: 511.7+ KB
```

```
df.describe()
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	REPORTING_
count	3.852000e+03	3852.000000	0.0	1779.00
mean	2.320439e+08	2406.555296	NaN	368.20
std	2.083995e+06	1146.258206	NaN	244.26
min	2.300005e+08	111.000000	NaN	2.00
25%	2.320015e+08	1402.000000	NaN	167.00
50%	2.320029e+08	3006.000000	NaN	335.00

```
df.columns
```

```
Index(['INCIDENT_NUMBER', 'OFFENSE_CODE', 'OFFENSE_CODE_GROUP',
       'OFFENSE_DESCRIPTION', 'DISTRICT', 'REPORTING_AREA', 'SHOOTING',
       'OCCURRED_ON_DATE', 'YEAR', 'MONTH', 'DAY_OF_WEEK', 'HOUR', 'UCR_PART',
       'STREET', 'Lat', 'Long', 'Location'],
      dtype='object')
```

```
df.isna().sum()
```

```
INCIDENT_NUMBER          0
OFFENSE_CODE             0
OFFENSE_CODE_GROUP    3852
OFFENSE_DESCRIPTION      0
DISTRICT                13
REPORTING_AREA        2073
SHOOTING                 0
OCCURRED_ON_DATE         0
YEAR                     0
MONTH                    0
DAY_OF_WEEK              0
HOUR                     0
UCR_PART              3852
STREET                   0
Lat                    291
Long                   291
```

```
Location               291
dtype: int64
```

```
df.dtypes
```

```
INCIDENT_NUMBER          int64
OFFENSE_CODE             int64
OFFENSE_CODE_GROUP     float64
OFFENSE_DESCRIPTION     object
DISTRICT                object
REPORTING_AREA         float64
SHOOTING                 int64
OCCURRED_ON_DATE        object
YEAR                     int64
MONTH                    int64
DAY_OF_WEEK             object
HOUR                     int64
UCR_PART               float64
STREET                  object
Lat                    float64
Long                   float64
Location                object
dtype: object
```

MISSING VALUE HANDLING

```
df["OFFENSE_CODE_GROUP"].value_counts()
```

```
Series([], Name: OFFENSE_CODE_GROUP, dtype: int64)
```

```
df["OFFENSE_CODE_GROUP"].unique()
```

```
array([nan])
```

```
df["UCR_PART"].value_counts()
```

```
Series([], Name: UCR_PART, dtype: int64)
```

```
df["UCR_PART"].unique()
```

```
array([nan])
```

```
df["DISTRICT"].value_counts()
```

```
B2          570
D4          482
A1          467
C11         457
B3          390
C6          295
D14         284
A7          232
E18         229
E13         197
E5          170
A15          63
External      3
Name: DISTRICT, dtype: int64
```

```
df["Lat"].value_counts()
```

```
42.297555    98
42.284826    91
42.349056    69
42.328663    65
42.339542    60
             ..
42.319285     1
42.331748     1
42.350728     1
42.276070     1
42.377168     1
Name: Lat, Length: 2087, dtype: int64
```

```
df["Long"].value_counts()
```

```
-71.059709    98
-71.091374    91
-71.150498    69
-71.085634    65
-71.069409    60
              ..
-71.050272     1
-71.083657     1
-71.061476     1
-71.089636     1
-71.029257     1
Name: Long, Length: 2087, dtype: int64
```

```
dfm=df.drop(["OFFENSE_CODE_GROUP","UCR_PART"],axis=1)
```

```
dfm["DISTRICT"] = df["DISTRICT"].fillna(method='ffill')
dfm["Lat"]=df["Lat"].fillna(dfm["Lat"].mode()[0])
dfm["Long"]=df["Long"].fillna(dfm["Long"].mode()[0])
dfm["REPORTING_AREA"]=dfm["REPORTING_AREA"].fillna(dfm["REPORTING_AREA"].mode()[0])
```

```
dfm.isna().sum()
```

```
INCIDENT_NUMBER         0
OFFENSE_CODE            0
OFFENSE_DESCRIPTION     0
DISTRICT                0
REPORTING_AREA          0
SHOOTING                0
OCCURRED_ON_DATE        0
YEAR                    0
MONTH                   0
DAY_OF_WEEK             0
HOUR                    0
STREET                  0
Lat                     0
Long                    0
```

```
Location              291
dtype: int64
```

LABEL ENCODING

```
df["DAY_OF_WEEK"].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0       Sunday
1       Sunday
2       Sunday
3       Sunday
4       Sunday
         ...
3847    Sunday
3848    Sunday
3849    Sunday
3850    Sunday
3851    Sunday
Name: DAY_OF_WEEK, Length: 3852, dtype: object>
```

```
df["OFFENSE_DESCRIPTION"].value_counts()
```

```
INVESTIGATE PERSON                         486
SICK ASSIST                                343
M/V - LEAVING SCENE - PROPERTY DAMAGE      201
VANDALISM                                  185
INVESTIGATE PROPERTY                       163
                                          ...
FRAUD - WELFARE                              1
DRUGS - POSSESSION OF DRUG PARAPHANALIA      1
DRUNKENNESS                                  1
MURDER, NON-NEGLIGENT MANSLAUGHTER           1
TRUANCY / RUNAWAY                            1
Name: OFFENSE_DESCRIPTION, Length: 94, dtype: int64
```

```
df["STREET"].value_counts()
```

```
WASHINGTON ST                                  311
BLUE HILL AVE                                  142
GIBSON ST                                       99
CENTRE ST                                       93
HARRISON AVE                                    83
                                              ...
SHAWMUT AVE & LENOX ST\nROXBURY  MA 02118\nUNITED ST      1
EDGE HILL ROAD                                  1
BEVERLY ST & ANTHONY RIP VALENTI WAY\nBOSTON  MA 02       1
MAYWOOD ST                                      1
SUMNER STREET                                   1
Name: STREET, Length: 1393, dtype: int64
```

```
from sklearn.preprocessing import LabelEncoder
le=LabelEncoder()
dfm["DAY_OF_WEEK"]=le.fit_transform(dfm["DAY_OF_WEEK"])
dfm["DISTRICT"]=le.fit_transform(dfm["DISTRICT"])
dfm["OFFENSE_DESCRIPTION"]=le.fit_transform(dfm["OFFENSE_DESCRIPTION"])
dfm["STREET"]=le.fit_transform(dfm["STREET"])
```

SPECIAL CHARACTERS REMOVAL

```
dfm["OCCURRED_ON_DATE"] = dfm["OCCURRED_ON_DATE"].str.replace('[-,+.:]','')
dfm['OCCURRED_ON_DATE']=dfm['OCCURRED_ON_DATE'].str.replace(' ','')
```

```
dfm
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_DESCRIPTION	DISTRICT
0	232005137	2670	31	7
1	232000017	1831	74	0
2	232000036	3802	55	4
3	232000034	801	2	0
4	232000041	801	2	0
...	...	...	...	...
3847	232005597	3115	33	11

```
dfm.isna().sum()
```

```
INCIDENT_NUMBER          0
OFFENSE_CODE             0
OFFENSE_DESCRIPTION      0
DISTRICT                 0
REPORTING_AREA           0
SHOOTING                 0
OCCURRED_ON_DATE         0
YEAR                     0
MONTH                    0
DAY_OF_WEEK              0
HOUR                     0
STREET                   0
Lat                      0
Long                     0
Location               291
dtype: int64
```

```
dfm.drop('Location', axis=1, inplace=True)
```

```
dfm.isna().sum()
```

```
INCIDENT_NUMBER        0
OFFENSE_CODE           0
```

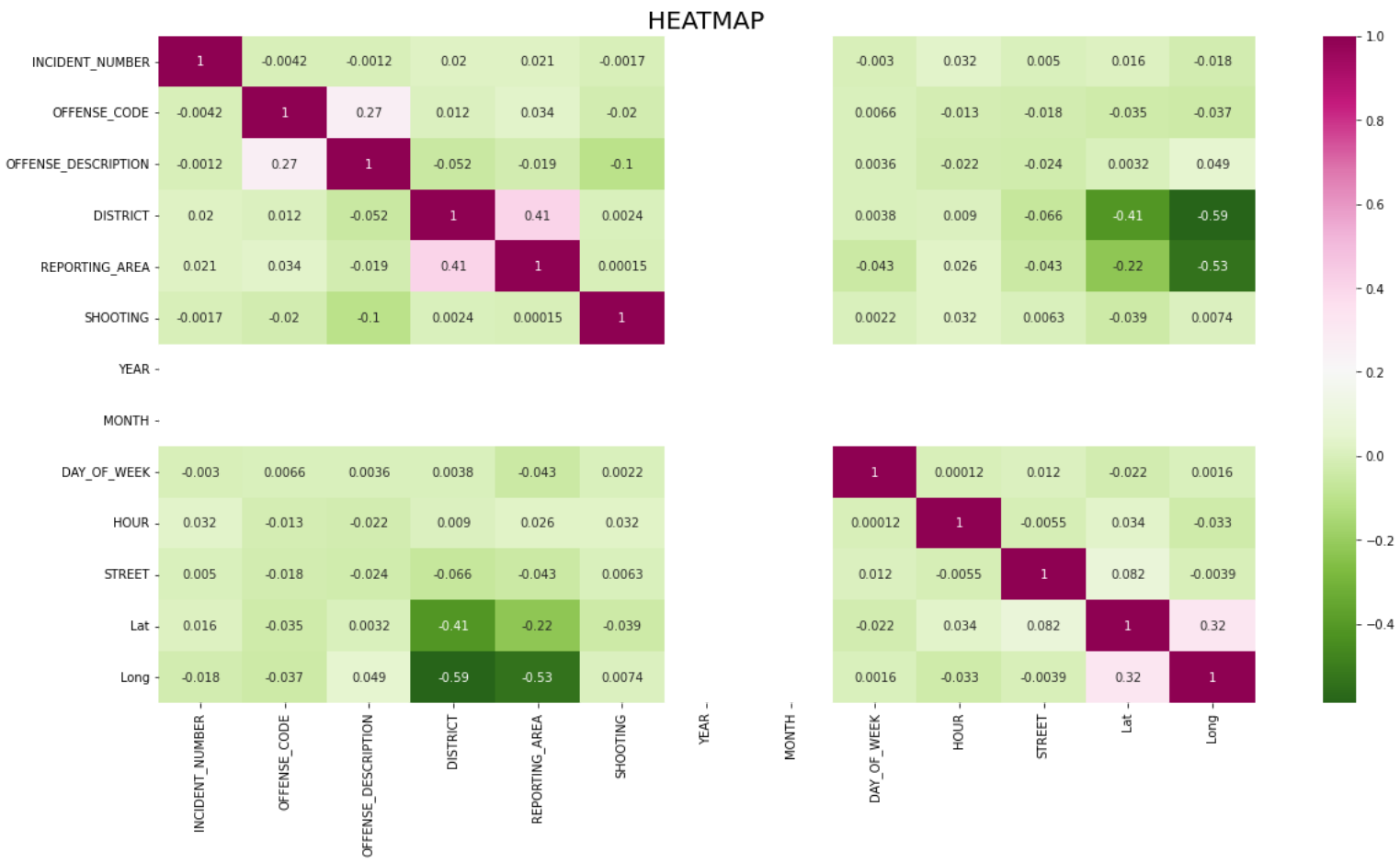
```
OFFENSE_DESCRIPTION    0
DISTRICT               0
REPORTING_AREA         0
SHOOTING               0
OCCURRED_ON_DATE       0
YEAR                   0
MONTH                  0
DAY_OF_WEEK            0
HOUR                   0
STREET                 0
Lat                    0
Long                   0
dtype: int64
```

```
dfm.dtypes
```

```
INCIDENT_NUMBER          int64
OFFENSE_CODE             int64
OFFENSE_DESCRIPTION      int64
DISTRICT                 int64
REPORTING_AREA         float64
SHOOTING                 int64
OCCURRED_ON_DATE        object
YEAR                     int64
MONTH                    int64
DAY_OF_WEEK              int64
HOUR                     int64
STREET                   int64
Lat                    float64
Long                   float64
dtype: object
```

HEAT MAP

```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize=(20,10))
sns.heatmap(dfm.corr(),cmap="PiYG_r",annot=True)
plt.title('HEATMAP',fontsize=20)
plt.show()
```




```
import seaborn as sns
import matplotlib.pyplot as plt
plt.figure(figsize = (20,10))
ax=sns.countplot(x = "OFFENSE_DESCRIPTION",data=dfm)
ax.set_title("COUNT PLOT OF OFFENSES OCCURED",fontsize=30)
plt.xlabel("OFFENSE_DESCRIPTION",fontsize=17)
plt.ylabel("COUNT", fontsize=17)
```

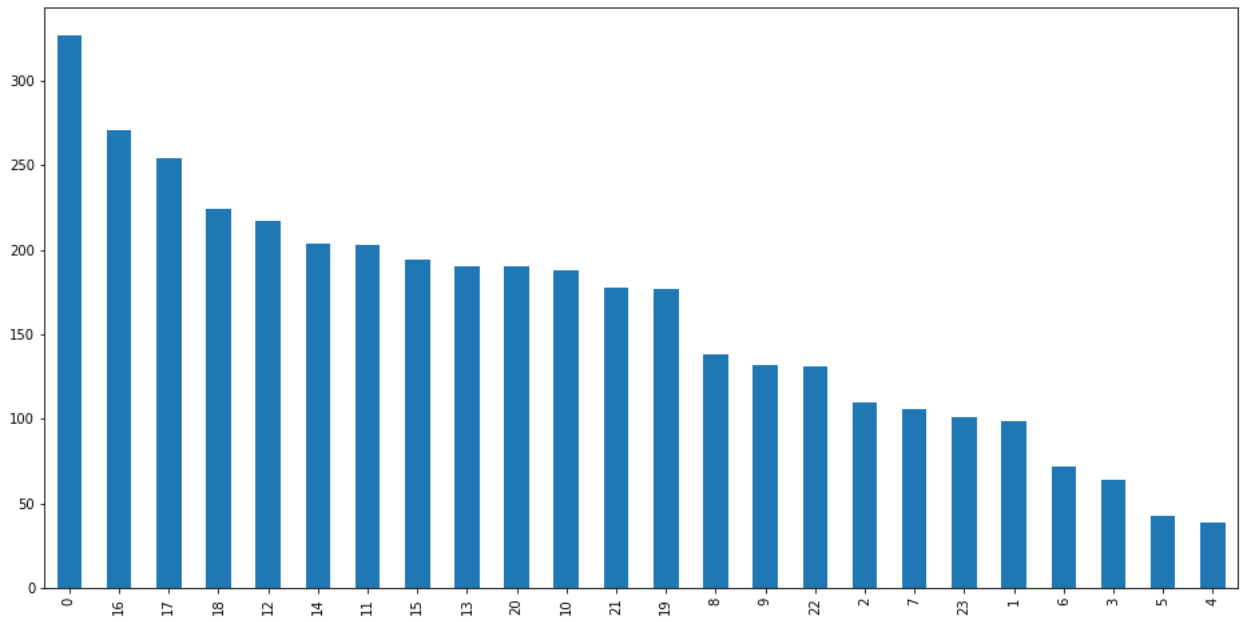
```
Text(0, 0.5, 'COUNT')
```

BAR PLOT REPRESENTATION OF THE DISTRICTS WHERE THE OFFENSE OCCURED

```
plt.figure(figsize=(16,8))
dfm['DISTRICT'].value_counts().plot.bar()
plt.show()
```


BAR PLOT REPRESENTATION OF TIME WHEN THE OFFENSE OCCURED

```
plt.figure(figsize=(16,8))
dfm['HOUR'].value_counts().plot.bar()
plt.show()
```



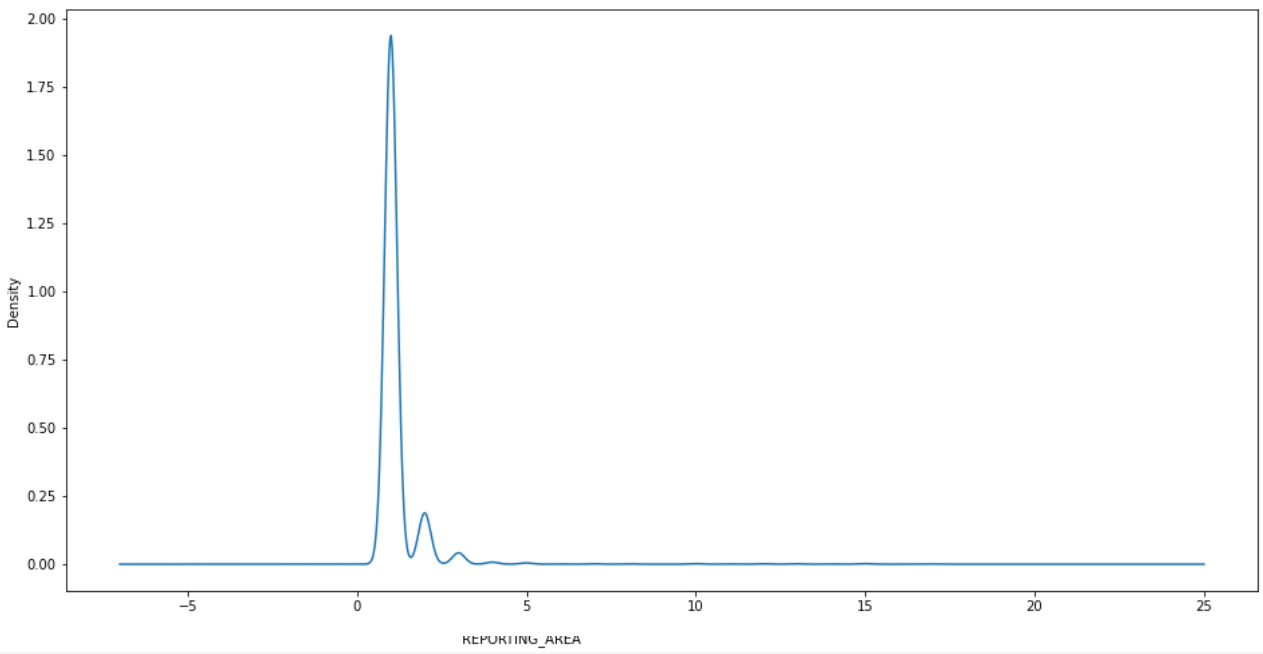
KDE PLOT REPRESENTATION OF REPORTING_AREA OF OFFENSE

```
plt.figure(figsize=(12,12))
sns.kdeplot(data=dfm, x="REPORTING_AREA", multiple="stack")
```

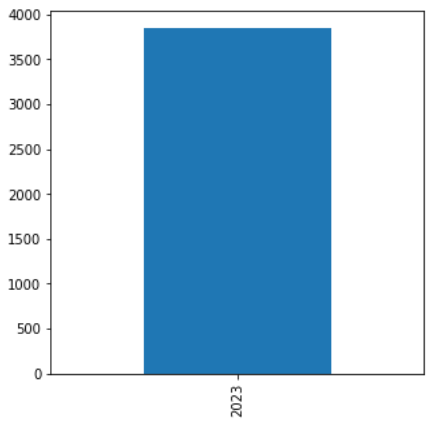
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fc809d43550>
```

BAR PLOT PRESENTATION OF TIME,DATE AND STREET

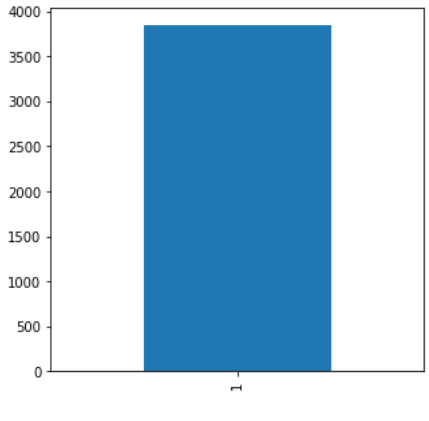
```
plt.figure(figsize=(16,8))
dfm['OCCURRED_ON_DATE'].value_counts().plot.kde()
plt.show()
```




```
plt.figure(figsize=(5,5))
dfm['YEAR'].value_counts().plot.bar()
plt.show()
```

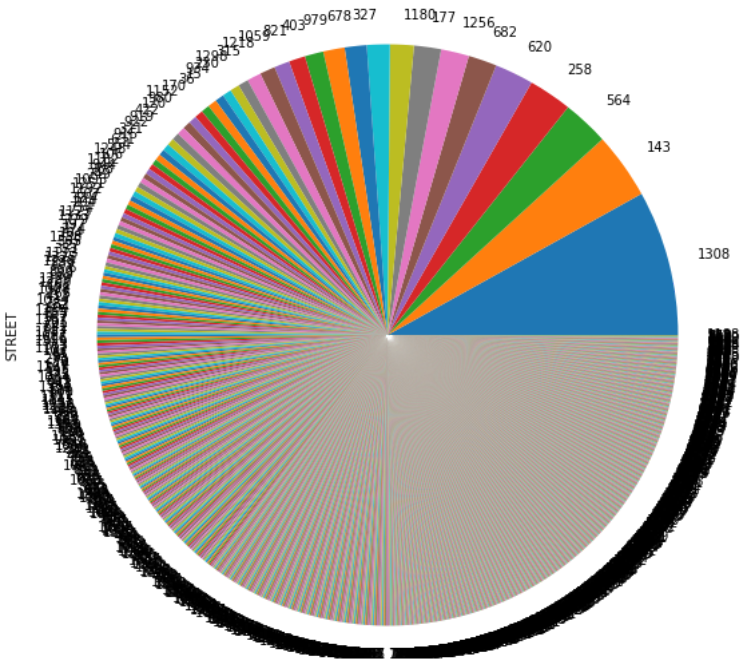



```
plt.figure(figsize=(5,5))
dfm['MONTH'].value_counts().plot.bar()
plt.show()
```



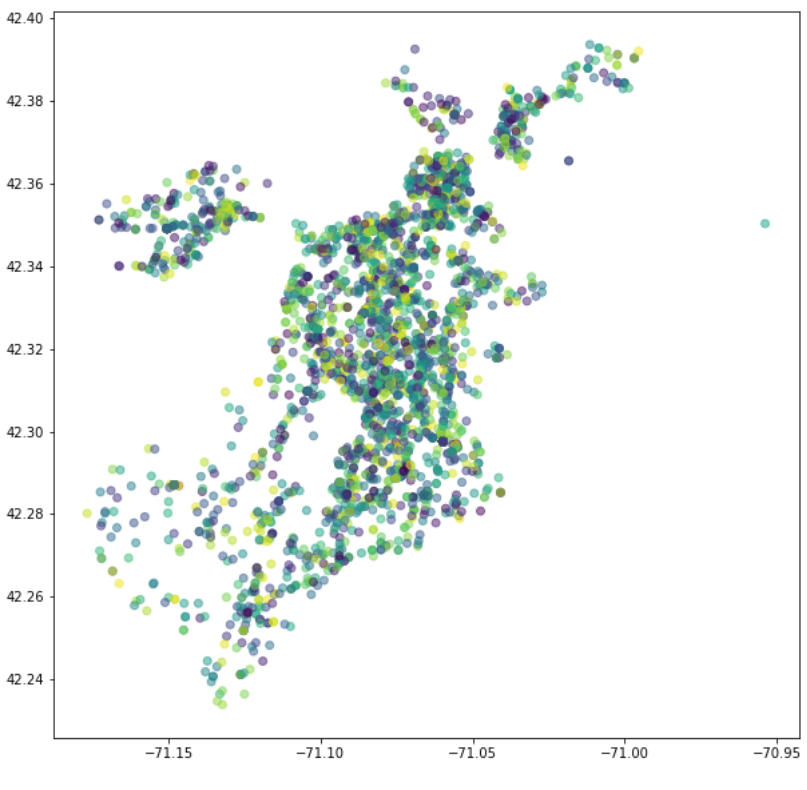

```
plt.figure(figsize=(10,10))
dfm['DAY_OF_WEEK'].value_counts().plot.bar()
plt.show()
```

```
plt.figure(figsize=(10,10))
dfm['STREET'].value_counts().plot.pie()
plt.show()
```



SCATTER PLOT PRESENTATION OF LOCATION

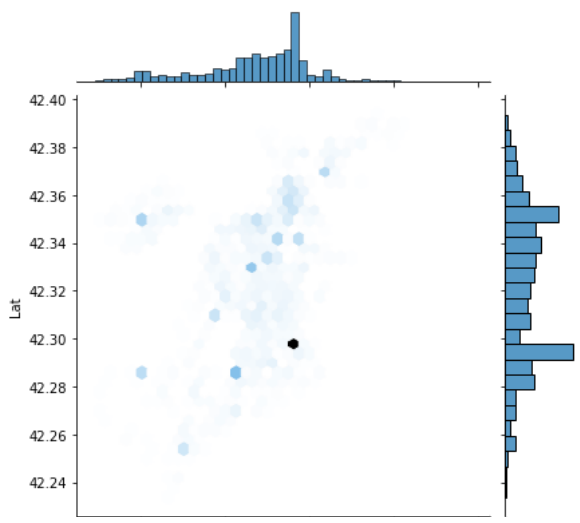
```
import numpy as np
location = dfm[['Lat','Long']]
location = location.loc[(location['Lat']>40) & (location['Long']<-60)]
x = location['Long']
y = location['Lat']
colors = np.random.rand(len(x))
plt.figure(figsize=(10,10))
plt.scatter(x, y,c=colors, alpha=0.5)
plt.show()
```



JOINT PLOT OF LOCATION

```
x = location['Long']
y = location['Lat']
sns.jointplot(x, y, kind='hex')
sns.jointplot(x, y, kind='kde')
```

```
<seaborn.axisgrid.JointGrid at 0x7fc8075fa670>
```



HISTOGRAM OF DATASET

```
dfm.hist(figsize=(20,20),bins=30)
plt.title("HISTOGRAM REPRESENTATION OF OFFENCE")
plt.show()
```

AGGLOMERATIVE CLUSTERING

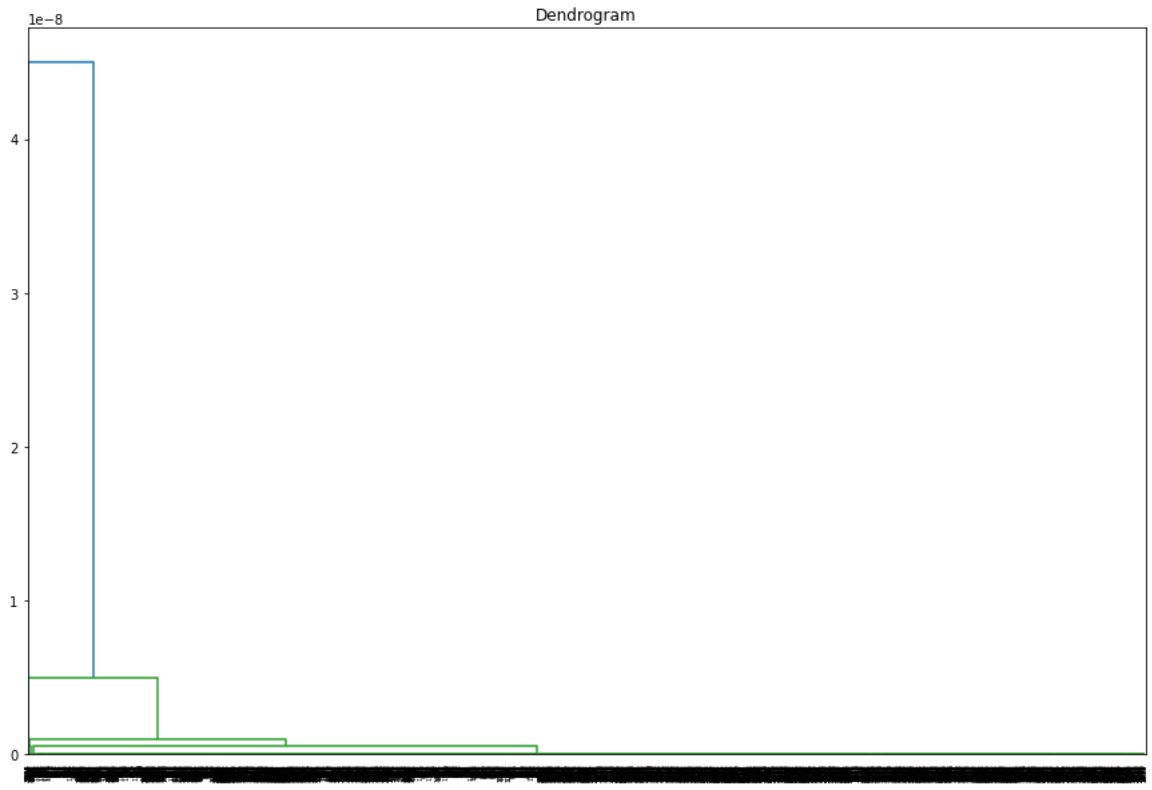
```
from sklearn.preprocessing import normalize
data=normalize(dfm)
data
```

```
array([[ 1.14683133e-07,  1.31981546e-12,  1.53237001e-14, ...,
         6.46561280e-13,  2.09336849e-14, -3.51706096e-14],
       [ 1.14680602e-07,  9.05086929e-13,  3.65791550e-14, ...,
         5.83289228e-13,  2.09082275e-14, -3.51257312e-14],
       [ 1.14680612e-07,  1.87937766e-12,  2.71872098e-14, ...,
         4.62676879e-13,  2.08984675e-14, -3.51412200e-14],
       ...,
       [ 1.14683241e-07,  1.89371077e-12,  2.27383700e-14, ...,
         4.83931832e-13,  2.09444204e-14, -3.51156017e-14],
       [ 1.14683245e-07,  1.43597750e-12,  4.30051781e-14, ...,
         9.58966040e-14,  2.09475592e-14, -3.51106421e-14],
       [ 1.14683246e-07,  9.05085989e-13,  3.65791170e-14, ...,
         5.87243121e-13,  2.09082057e-14, -3.51256948e-14]])
```

```
X=pd.DataFrame(data,columns=dfm.columns)
X
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_DESCRIPTION	DISTRICT	REPORTING_AREA	SHOOTING	OCCURRED_ON_DATE	YEAR	MONTH	DAY_OF_WEEK	HOUR	STREET	Lat	Lon
0	1.146831e-07	1.319815e-12	1.532370e-14	3.460190e-15	3.845754e-13	0.0	1.0	9.999950e-13	4.943129e-16	1.482939e-15	0.000000e+00	6.465613e-13	2.093368e-14	-3.517061e
1	1.146806e-07	9.050869e-13	3.657915e-14	0.000000e+00	3.806209e-14	0.0	1.0	9.999950e-13	4.943129e-16	1.482939e-15	4.943129e-16	5.832892e-13	2.090823e-14	-3.512573e
2	1.146806e-07	1.879378e-12	2.718721e-14	1.977252e-15	1.754811e-13	0.0	1.0	9.999950e-13	4.943129e-16	1.482939e-15	9.886258e-16	4.626769e-13	2.089847e-14	-3.514122e
3	1.146806e-07	3.959446e-13	9.886258e-16	0.000000e+00	5.981186e-14	0.0	1.0	9.999950e-13	4.943129e-16	1.482939e-15	9.886258e-16	5.818063e-13	2.093468e-14	-3.512859e
4	1.146806e-07	3.959446e-13	9.886258e-16	0.000000e+00	1.754811e-13	0.0	1.0	9.999950e-13	4.943129e-16	1.482939e-15	9.886258e-16	5.818063e-13	2.093466e-14	-3.512772e
...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
3847	1.146832e-07	1.539783e-12	1.631231e-14	5.437436e-15	1.754809e-13	0.0	1.0	9.999940e-13	4.943124e-16	1.482937e-15	0.000000e+00	2.758263e-13	2.088936e-14	-3.516933e
								9.999940e-	4.943124e-	1.482937e-		2.194747e-	2.092497e-	-3.511375e

```
import scipy.cluster.hierarchy as shc
import matplotlib.pyplot as plt
plt.figure(figsize=(15,10))
plt.title("Dendrogram")
den=shc.dendrogram(shc.linkage(X,method="average"))
```

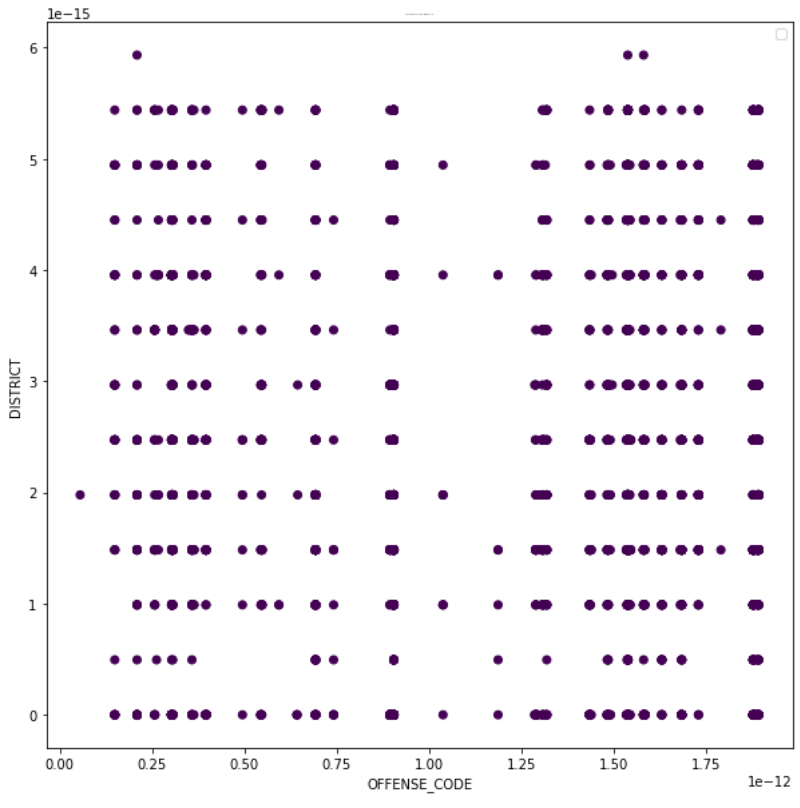



```
from sklearn.cluster import AgglomerativeClustering
ac=AgglomerativeClustering(n_clusters=3,affinity="euclidean",linkage="complete") #linkage = distance between 2 sets of observation
Y=ac.fit_predict(X)
Y
```

```
array([0, 0, 0, ..., 0, 0, 0])
```

```
import matplotlib.pyplot as plt
plt.figure(figsize=(10,10))
plt.scatter(X["OFFENSE_CODE"],X["DISTRICT"],c=ac.labels_)
plt.xlabel("OFFENSE_CODE",fontsize=10)
plt.ylabel("DISTRICT",fontsize=10)
plt.title("SCATTER PLOT OF CRIME AND DISTRICT",fontsize=0)
plt.legend()
plt.show()
```

```
WARNING:matplotlib.legend:No handles with labels found to put in legend.
```



K-MEANS CLUSTERING

```
AX=dfm.iloc[:,1:]
```

```
from sklearn.cluster import KMeans
km=KMeans(n_clusters=3,init="k-means++",random_state=1)
y_kmeans=km.fit_predict(AX)
print(y_kmeans)
```

```
[0 0 0 ... 1 1 1]
```

```
AX["cluster"]=y_kmeans
```

```
AX
```

	OFFENSE_CODE	OFFENSE_DESCRIPTION	DISTRICT	REPORTING_AREA	SHOOTING	OCCURRED_ON_DATE	YEAR	MONTH	DAY_OF_WEEK	HOUR	STREET	Lat	Long	cluster
0	2670	31	7	778.0	0	2023010100000000	2023	1	3	0	1308	42.349056	-71.150498	0
1	1831	74	0	77.0	0	2023010101140000	2023	1	3	1	1180	42.297555	-71.059709	0
2	3802	55	4	355.0	0	2023010102380000	2023	1	3	2	936	42.277811	-71.091043	0
3	801	2	0	121.0	0	2023010102450000	2023	1	3	2	1177	42.351066	-71.065496	0
4	801	2	0	355.0	0	2023010102540000	2023	1	3	2	1177	42.351031	-71.063733	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
3847	3115	33	11	355.0	0	2023012200570000	2023	1	3	0	558	42.259435	-71.147981	1
3848	3115	33	6	355.0	0	2023012201090000	2023	1	3	1	444	42.331476	-71.035544	1
3849	3831	46	2	355.0	0	2023012201170000	2023	1	3	1	979	42.370818	-71.039291	1
3850	2905	87	2	355.0	0	2023012202380000	2023	1	3	2	194	42.377168	-71.029257	1
3851	1831	74	2	29.0	0	2023012202390000	2023	1	3	2	1188	42.297555	-71.059709	1

3852 rows × 14 columns

```
AX["cluster"].value_counts()
```

```
2    1330
0    1294
1    1228
Name: cluster, dtype: int64
```

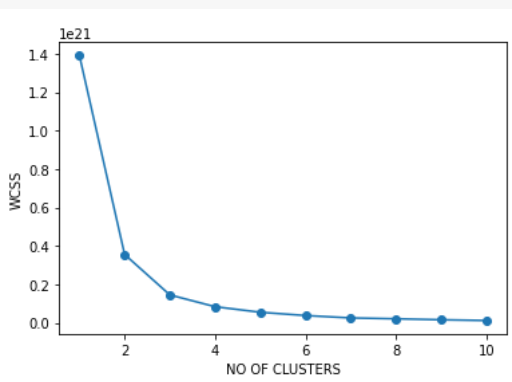
ELBOW METHOD

```
wcss=[]
for i in range(1,11):
  kms=KMeans(n_clusters=i,init="k-means++",n_init=10,max_iter=300,random_state=1)
  kms.fit(AX)
  wcss.append(kms.inertia_)
```

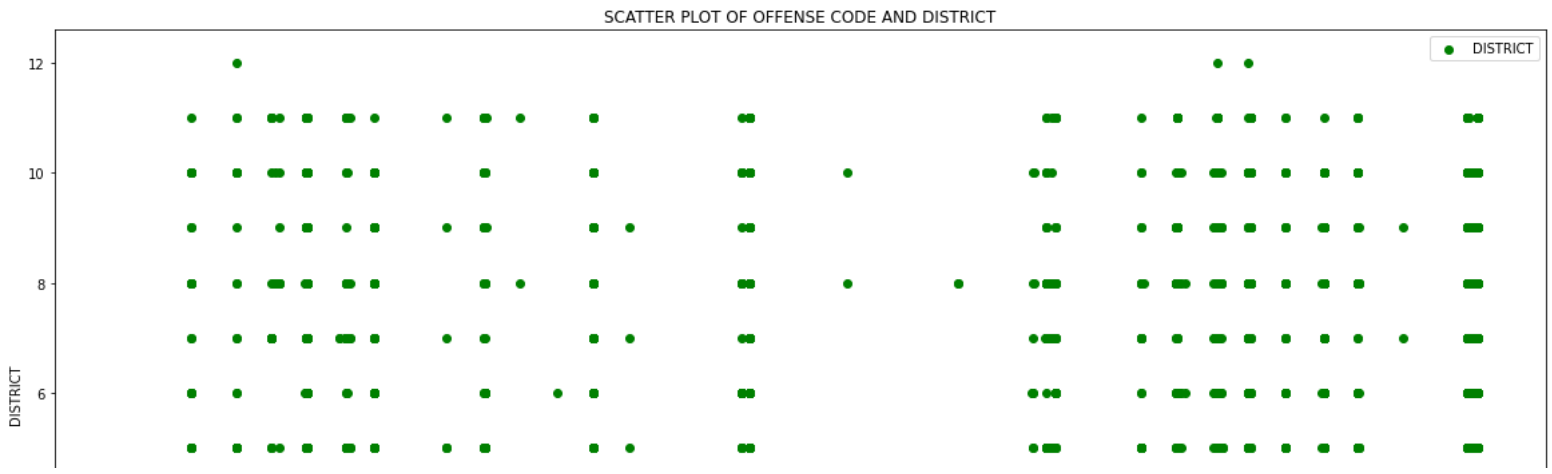
```
wcss
```

```
[1.3901645645417687e+21,
 3.554429175960222e+20,
 1.4523201420140675e+20,
 8.38559664761746e+19,
 5.481178873155117e+19,
 3.791647149620147e+19,
 2.539864335232718e+19,
 2.0946548720287896e+19,
 1.5861870921205334e+19,
 1.1856591938654425e+19]
```

```
plt.plot(range(1,11),wcss,marker="o")
plt.xlabel("NO OF CLUSTERS")
plt.ylabel("WCSS")
plt.show()
```

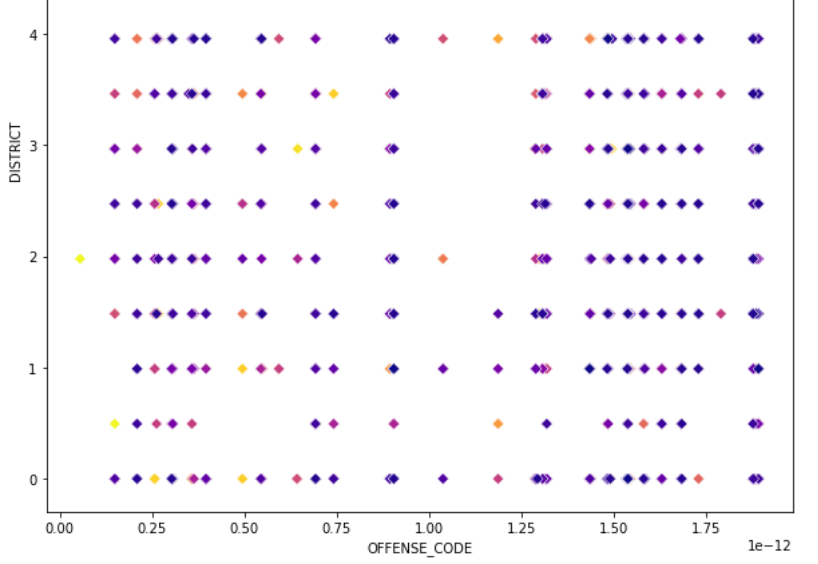



```
plt.figure(figsize=(20,10))
plt.scatter(data=AX,x="OFFENSE_CODE",y="DISTRICT",c="green")
plt.title("SCATTER PLOT OF OFFENSE CODE AND DISTRICT")
plt.xlabel("OFFENSE CODE")
plt.ylabel("DISTRICT")
plt.legend()
plt.show()
```



COMPARISON BETWEEN K-MEANS AND AGGLOMERATIVE CLUSTERING PLOTS

```
plt.figure(figsize=(10,10))
plt.title('K-MeansClustering Plot')
sns.scatterplot(x="OFFENSE_CODE", y="DISTRICT",data=AX,palette='plasma',hue="OFFENSE_CODE",marker="s")
plt.legend(loc="upper right")
plt.figure(figsize=(10,10))
plt.title('Agglomerative Clustering Plot')
sns.scatterplot(x="OFFENSE_CODE", y="DISTRICT",data=X,palette='plasma',hue="YEAR",marker="D")
plt.legend(loc="upper right")
plt.show()
```



✓ 1s completed at 12:51 AM

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.