```
#link='https://drive.google.com/file/d/1fkzGFx2nK0YDXBmJbcz51eugP08F88s8/view?usp=share_link'
```

```
!apt install proj-bin libproj-dev libgeos-dev
!pip install https://github.com/matplotlib/basemap/archive/v1.1.0.tar.gz
!pip install git+https://github.com/python-visualization/folium
```

```
    Preparing metadata (setup.py) ... done
  Requirement already satisfied: numpy>=1.2.1 in /usr/local/lib/python3.10/dist-packages
  Requirement already satisfied: matplotlib>=1.0.0 in /usr/local/lib/python3.10/dist-pack
  Requirement already satisfied: pyproj>=1.9.3 in /usr/local/lib/python3.10/dist-packages
  Requirement already satisfied: pyshp>=1.2.0 in /usr/local/lib/python3.10/dist-packages
  Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packa
  Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages
  Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-pack
  Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-pack
  Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packag
  Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages
  Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packa
  Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-p
  Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from
  Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (fro
  Building wheels for collected packages: basemap
    error: subprocess-exited-with-error

    × python setup.py bdist_wheel did not run successfully.
    │ exit code: 1
    ╰─> See above for output.

    note: This error originates from a subprocess, and is likely not a problem with pip.
    Building wheel for basemap (setup.py) ... error
    ERROR: Failed building wheel for basemap
    Running setup.py clean for basemap
  Failed to build basemap
  ERROR: Could not build wheels for basemap, which is required to install pyproject.toml-
  Collecting git+https://github.com/python-visualization/folium
    Cloning https://github.com/python-visualization/folium to /tmp/pip-req-build-hera69zh
    Running command git clone --filter=blob:none --quiet https://github.com/python-visual
```

```
Successfully installed folium-0.16.1.dev38+gae827c1
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime
import seaborn as sns
%matplotlib inline
import scipy.stats as stats
import statsmodels.api as sm
from statsmodels.tsa.arima_model import ARIMA
from statsmodels.tsa.arima_process import arma_generate_sample, ArmaProcess
sns.set_palette([ "#30a2da", "#fc4f30", "#e5ae38", "#6d904f", "#8b8b8b"])

import folium
from sklearn.cluster import KMeans
```

```
file_id='166E_XsEnUrciks_XpuKmmdf2lNk5mFh9'
link='https://drive.google.com/file/d/1fkzGFx2nK0YDXBmJbcz51eugP08F88s8/view?usp=share_link'
csv_url=link.format(FILE_ID=file_id)
data = pd.read_csv("/content/economic-indicators.csv")
crimes=pd.read_csv("/content/economic-indicators.csv")
```

```
data.head()
```

	Year	Month	logan_passengers	logan_intl_flights	hotel_occup_rate	hotel_avg
0	2013	1	2019662	2986	0.572	
1	2013	2	1878731	2587	0.645	
2	2013	3	2469155	3250	0.819	
3	2013	4	2551246	3408	0.855	
4	2013	5	2676291	3240	0.858	

Next steps: View recommended plots

```
data.shape
```

```
(84, 19)
```

```
data.columns
```

```
Index(['Year', 'Month', 'logan_passengers', 'logan_intl_flights',
       'hotel_occup_rate', 'hotel_avg_daily_rate', 'total_jobs', 'unemp_rate',
       'labor_force_part_rate', 'pipeline_unit', 'pipeline_total_dev_cost',
       'pipeline_sqft', 'pipeline_const_jobs', 'foreclosure_pet',
       'foreclosure_deeds', 'med_housing_price', 'housing_sales_vol',
       'new_housing_const_permits', 'new-affordable_housing_permits'],
      dtype='object')
```

```
data.describe()
```

	Year	Month	logan_passengers	logan_intl_flights	hotel_occup_ra
count	84.000000	84.000000	8.400000e+01	84.000000	84.0000
mean	2016.000000	6.500000	3.015647e+06	3940.511905	0.8177
std	2.012012	3.472786	5.492766e+05	694.479496	0.108
min	2013.000000	1.000000	1.878731e+06	2587.000000	0.5720
25%	2014.000000	3.750000	2.604905e+06	3408.000000	0.7685
50%	2016.000000	6.500000	3.018654e+06	3960.500000	0.8775
75%	2018.000000	9.250000	3.413058e+06	4516.250000	0.9012
max	2019.000000	12.000000	4.120937e+06	5260.000000	0.9310

```
data.isnull().sum()
```

```
Year                               0
Month                              0
logan_passengers                   0
logan_intl_flights                 0
hotel_occup_rate                   0
hotel_avg_daily_rate               0
total_jobs                         0
unemp_rate                         0
labor_force_part_rate              0
pipeline_unit                      0
pipeline_total_dev_cost            0
pipeline_sqft                      0
pipeline_const_jobs                0
foreclosure_pet                    0
foreclosure_deeds                  0
med_housing_price                  0
housing_sales_vol                  0
new_housing_const_permits          0
new-affordable_housing_permits     0
dtype: int64
```

```
data = pd.read_csv("/content/script_113631134_20210423193017_combine.csv")
crimes=pd.read_csv("/content/script_113631134_20210423193017_combine.csv")
```

```
<ipython-input-12-88d18b010942>:1: DtypeWarning: Columns (0) have mixed types. Specify dtyp
  data = pd.read_csv("/content/script_113631134_20210423193017_combine.csv")
<ipython-input-12-88d18b010942>:2: DtypeWarning: Columns (0) have mixed types. Specify dtyp
  crimes=pd.read_csv("/content/script_113631134_20210423193017_combine.csv")
```

```
data.head()
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPTION	DISTRI
0	854652314	3115	NaN	INVESTIGATE PERSON	
1	457856954	3115	NaN	INVESTIGATE PERSON	
2	302030654	1106	NaN	FRAUD – CREDIT CARD / ATM FRAUD	
3	212025777	1108	NaN	FRAUD – WELFARE	
4	212025557	2670	NaN	HARASSMENT/ CRIMINAL HARASSMENT	

Next steps: View recommended plots

```
data.shape
```

```
(70894, 17)
```

```
data.shape
```

```
(70894, 17)
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 70894 entries, 0 to 70893
Data columns (total 17 columns):
 #   Column               Non-Null Count  Dtype  
---  ------               --------------  -----  
 0   INCIDENT_NUMBER      70894 non-null  object 
 1   OFFENSE_CODE         70894 non-null  int64  
 2   OFFENSE_CODE_GROUP   0 non-null      float64
 3   OFFENSE_DESCRIPTION  70894 non-null  object 
 4   DISTRICT             70616 non-null  object 
 5   REPORTING_AREA       70894 non-null  object 
 6   SHOOTING             70894 non-null  int64  
 7   OCCURRED_ON_DATE     70894 non-null  object 
 8   YEAR                 70894 non-null  int64  
 9   MONTH                70894 non-null  int64  
 10  DAY_OF_WEEK          70894 non-null  object 
 11  HOUR                 70894 non-null  int64  
 12  UCR_PART             0 non-null      float64
 13  STREET               70893 non-null  object 
 14  Lat                  70894 non-null  float64
 15  Long                 70894 non-null  float64
 16  Location             70894 non-null  object 
dtypes: float64(4), int64(5), object(8)
memory usage: 9.2+ MB
```

```
data.describe()
```

	OFFENSE_CODE	OFFENSE_CODE_GROUP	SHOOTING	YEAR	MONTH	
count	70894.000000	0.0	70894.000000	70894.0	70894.000000	70894.
mean	2292.077200	NaN	0.015826	2020.0	6.595678	12
std	1257.571021	NaN	0.124805	0.0	3.452960	6.
min	111.000000	NaN	0.000000	2020.0	1.000000	0.
25%	801.000000	NaN	0.000000	2020.0	3.000000	9.
50%	3005.000000	NaN	0.000000	2020.0	7.000000	14.
75%	3125.000000	NaN	0.000000	2020.0	10.000000	18.
max	99999.000000	NaN	1.000000	2020.0	12.000000	23.

```
data.isnull().sum()
```

```
INCIDENT_NUMBER            0
OFFENSE_CODE               0
OFFENSE_CODE_GROUP     70894
OFFENSE_DESCRIPTION        0
DISTRICT                 278
REPORTING_AREA             0
SHOOTING                   0
OCCURRED_ON_DATE           0
YEAR                       0
MONTH                      0
DAY_OF_WEEK                0
HOUR                       0
UCR_PART               70894
STREET                     1
Lat                        0
Long                       0
Location                   0
dtype: int64
```

```
crime_count=pd.DataFrame(data.groupby("OFFENSE_CODE_GROUP").size().sort_values(ascending=False)
crime_count.head(20)
```

OFFENSE_CODE_GROUP	counts

```
street_count=pd.DataFrame(data.groupby("STREET").size().sort_values(ascending=False).rename("co
street_count.head(20)
```

	STREET	counts
0	WASHINGTON ST	3276
1	BLUE HILL AVE	1277
2	TREMONT ST	1076
3	MASSACHUSETTS AVE	1056
4	BOYLSTON ST	994
5	DORCHESTER AVE	983
6	CENTRE ST	940
7	COMMONWEALTH AVE	777
8	HARRISON AVE	730
9	HYDE PARK AVE	707
10	RIVER ST	702
11	HUNTINGTON AVE	545
12	COLUMBIA RD	534
13	NEWBURY ST	440
14	DUDLEY ST	384
15	AMERICAN LEGION HWY	359
16	WARREN ST	351
17	COLUMBUS AVE	348
18	CAMBRIDGE ST	338
19	ADAMS ST	329

Next steps: View recommended plots

```
district_count=pd.DataFrame(data.groupby("DISTRICT").size().sort_values(ascending=False).rename
district_count.head(20)
```

	DISTRICT	counts
0	B2	10442
1	D4	9283
2	C11	8992
3	B3	7828
4	A1	7013
5	C6	5440
6	D14	4756
7	E18	4177
8	E13	4103
9	E5	3595
10	A7	3141
11	A15	1600
12	External	246

Next steps: View recommended plots

```
shoot_data=data.dropna(axis = 0, subset = ['SHOOTING'] )
```

```
shoot_data.head(5)
```

	INCIDENT_NUMBER	OFFENSE_CODE	OFFENSE_CODE_GROUP	OFFENSE_DESCRIPTION	DISTRI
0	854652314	3115	NaN	INVESTIGATE PERSON	
1	457856954	3115	NaN	INVESTIGATE PERSON	
2	302030654	1106	NaN	FRAUD – CREDIT CARD / ATM FRAUD	
3	212025777	1108	NaN	FRAUD – WELFARE	
4	212025557	2670	NaN	HARASSMENT/ CRIMINAL HARASSMENT	

Next steps: View recommended plots

```
data['Lat'].describe()
```

```
count    70894.000000
mean        41.278263
std          6.563217
```

```
min          0.000000
25%         42.295129
50%         42.325303
75%         42.346735
max         42.395042
Name: Lat, dtype: float64
```

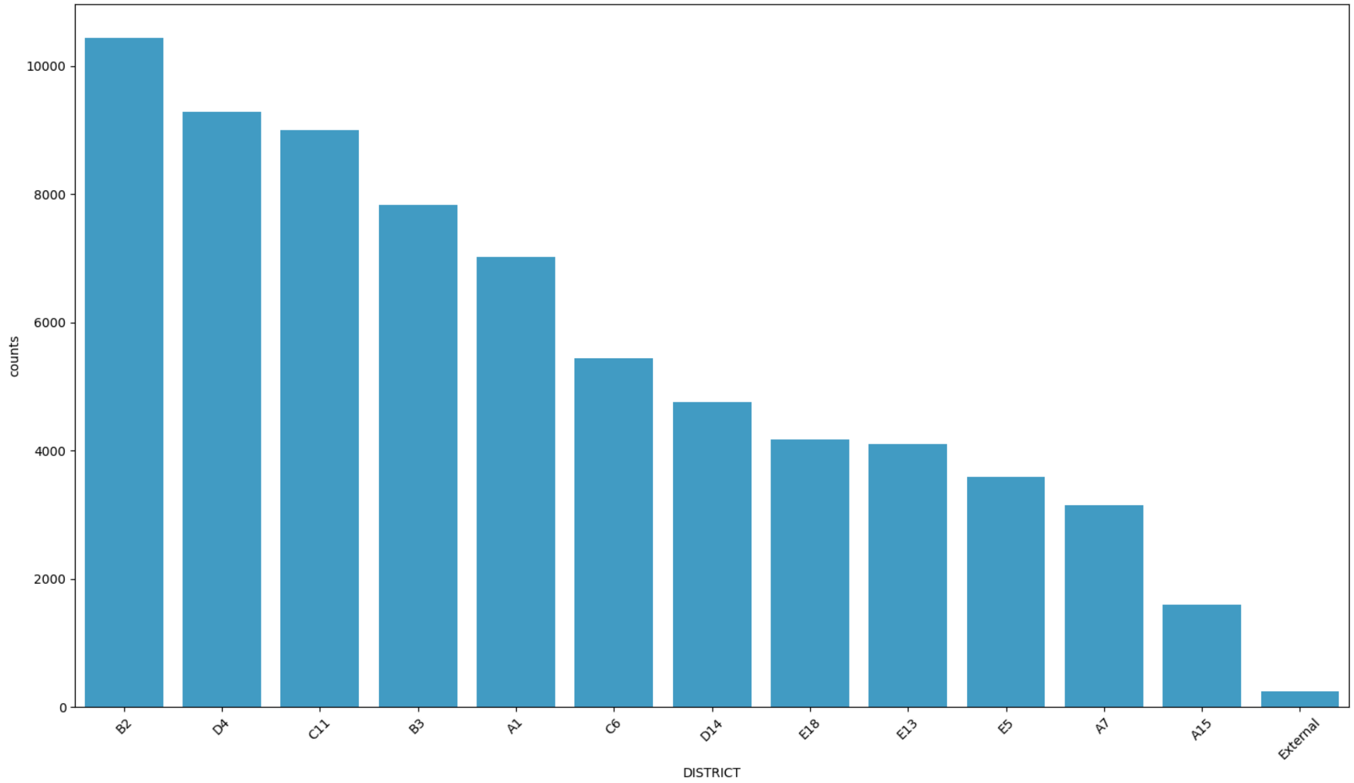
```
data.Lat = data["Lat"].replace(-1,np.nan)
data.Long = data["Long"].replace(-1,np.nan)
shoot_data.Lat = data["Lat"].replace(-1,np.nan)
shoot_data.Long = data["Long"].replace(-1,np.nan)
```

```
data['Lat'].describe()
```

```
count    70894.000000
mean        41.278263
std          6.563217
min          0.000000
25%         42.295129
50%         42.325303
75%         42.346735
max         42.395042
Name: Lat, dtype: float64
```

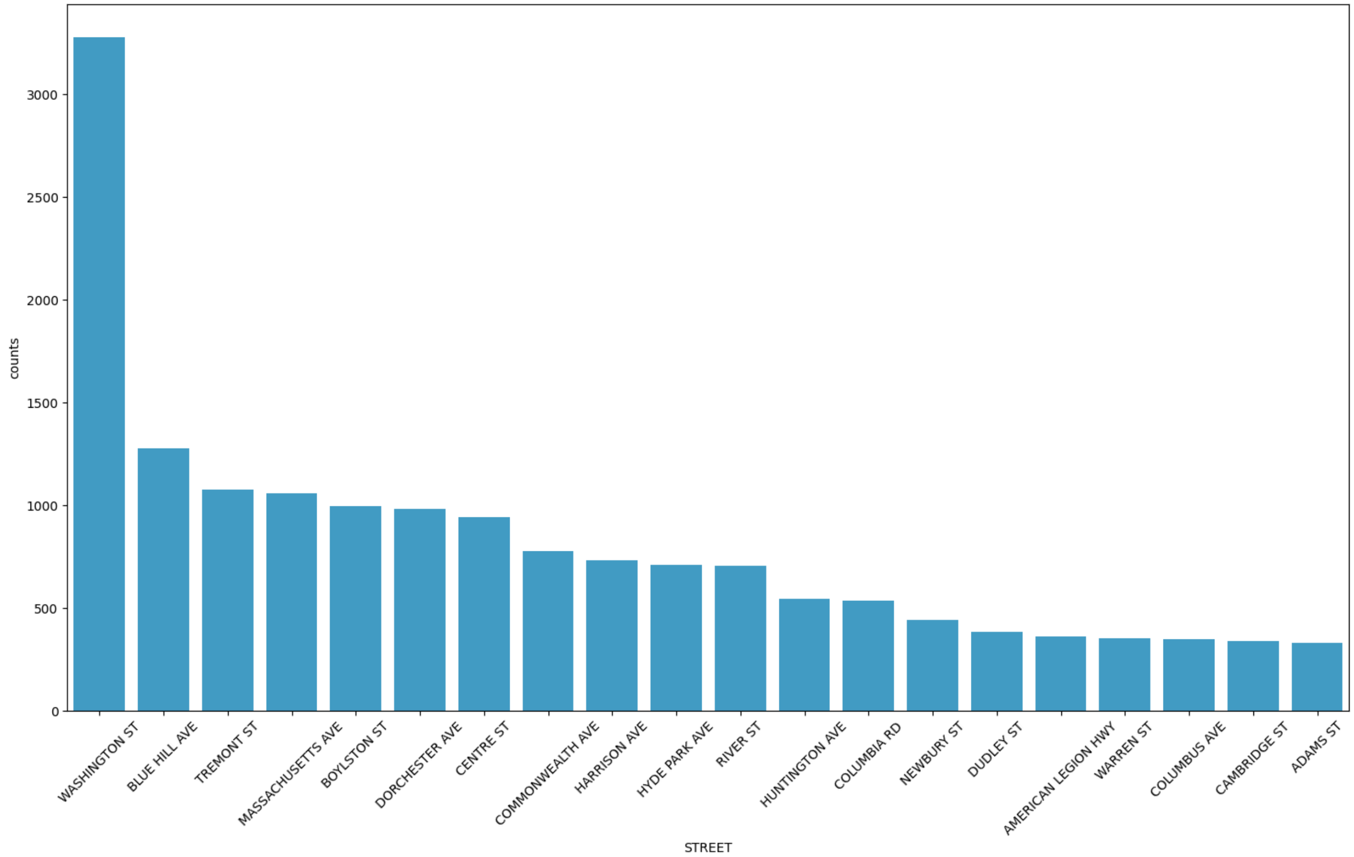
```
district_count = pd.concat([district_count["counts"],district_count["DISTRICT"]],axis=1)
f,ax = plt.subplots(figsize=(18,10))
fig= sns.barplot(x = "DISTRICT",y="counts",data=district_count.head(20))
fig.axis(ymin=0)
plt.xticks(rotation=45)
```

```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12],
 [Text(0, 0, 'B2'),
  Text(1, 0, 'D4'),
  Text(2, 0, 'C11'),
  Text(3, 0, 'B3'),
  Text(4, 0, 'A1'),
  Text(5, 0, 'C6'),
  Text(6, 0, 'D14'),
  Text(7, 0, 'E18'),
  Text(8, 0, 'E13'),
  Text(9, 0, 'E5'),
  Text(10, 0, 'A7'),
  Text(11, 0, 'A15'),
  Text(12, 0, 'External')])
```



```
street_count = pd.concat([street_count["counts"],street_count["STREET"]],axis=1)
f,ax = plt.subplots(figsize=(18,10))
fig= sns.barplot(x = "STREET",y="counts",data=street_count.head(20))
fig.axis(ymin=0)
plt.xticks(rotation=45)
```

```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19],
 [Text(0, 0, 'WASHINGTON ST'),
  Text(1, 0, 'BLUE HILL AVE'),
  Text(2, 0, 'TREMONT ST'),
  Text(3, 0, 'MASSACHUSETTS AVE'),
  Text(4, 0, 'BOYLSTON ST'),
  Text(5, 0, 'DORCHESTER AVE'),
  Text(6, 0, 'CENTRE ST'),
  Text(7, 0, 'COMMONWEALTH AVE'),
  Text(8, 0, 'HARRISON AVE'),
  Text(9, 0, 'HYDE PARK AVE'),
  Text(10, 0, 'RIVER ST'),
  Text(11, 0, 'HUNTINGTON AVE'),
  Text(12, 0, 'COLUMBIA RD'),
  Text(13, 0, 'NEWBURY ST'),
  Text(14, 0, 'DUDLEY ST'),
  Text(15, 0, 'AMERICAN LEGION HWY'),
  Text(16, 0, 'WARREN ST'),
  Text(17, 0, 'COLUMBUS AVE'),
  Text(18, 0, 'CAMBRIDGE ST'),
  Text(19, 0, 'ADAMS ST')])
```



```
df = pd.to_datetime(data.OCCURRED_ON_DATE)
data.OCCURRED_ON_DATE=df
data["DAY"]=df.dt.day
data.index = pd.DatetimeIndex(data.OCCURRED_ON_DATE)
```

```
fig,(ax1,ax2,ax3,ax4,ax5)=plt.subplots(5,1,figsize=(20,35))

sns.countplot(x="YEAR",data=data,ax=ax1)
ax1.set_ylabel("number of crimes")
ax1.set_title("number of crimes by year")

sns.countplot(x="MONTH",data=data,ax=ax2)
ax2.set_ylabel("number of crimes")
ax2.set_title("number of crimes by month")

sns.countplot(x="DAY",data=data,ax=ax3)
ax3.set_ylabel("number of crimes")
ax3.set_title("number of crimes by day")

sns.countplot(x="DAY_OF_WEEK",data=data,ax=ax4)
ax4.set_ylabel("number of crimes")
ax4.set_title("number of crimes by day of week")

sns.countplot(x="HOUR",data=data,ax=ax5)
ax5.set_ylabel("number of crimes")
ax5.set_title("number of crimes by hour")
```