Student Name: Fahad Ahmad **Date: Oct 19/2020**
Instructor: Nasir Mahmood
Course: INFO40041-Predictive Analytics and Machine Learning

- *Note that complete code and output can be viewed from the attached .ipynb file. Thank you*

Q1. Data was imported successfully into a pandas DataFrame. I used Google Colaboratory, so I loaded the dataset linked to my Github account which stored the data set.

Q2. Crimedata.shape
(50, 7)

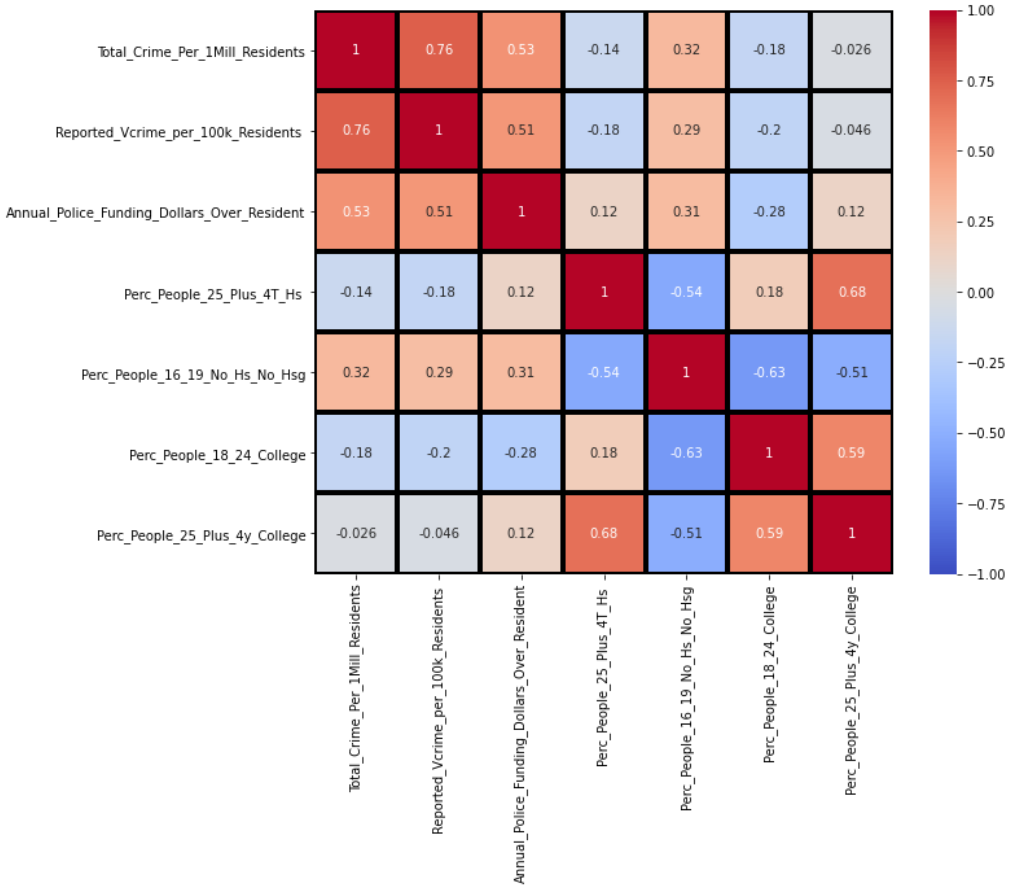
	X1	X2	X3	X4	X5	X6	X7	
Count		50.0	50.0	50.0	50.0	50.0	50.0	50.0
Mean		718.0	616.0	38.0	59.0	15.0	30.0	14.0
Std		294.0	574.0	14.0	10.0	6.0	15.0	5.0
Min		341.0	29.0	16.0	42.0	4.0	7.0	8.0
25%		497.0	231.0	30.0	49.0	11.0	21.0	11.0
50%		654.0	454.0	34.0	59.0	14.0	25.0	12.0
75%		820.0	822.0	42.0	67.0	19.0	34.0	16.0
Max		1740.0	3545.0	86.0	81.0	34.0	81.0	36.0

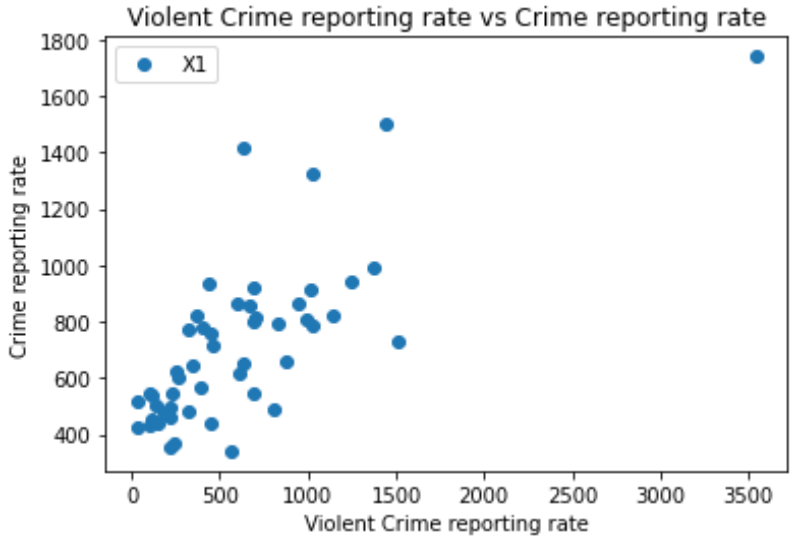
```
Crimedata.isnull().sum()

X1    0
X2    0
X3    0
X4    0
X5    0
X6    0
X7    0
      dtype: int64
```

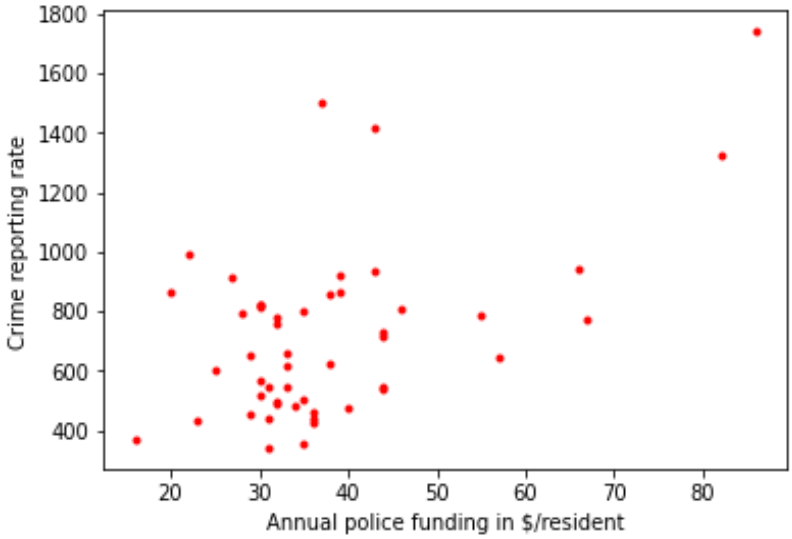
There are no NaN values in the dataset.

Q3. Data was plotted with more then two chart types, show below are 2 plots; Correlation plot and Scatterplots. From correlation plot, we can see that reported violent crime rate per 100,000 residents has the highest correlation with Total overall reported crime rate per 1 million residents. Annual police funding in dollars per resident was also correlated positively with total

reported crime rate. Scatterplots show a positive relationship between both independent variables and total overall reported total crime rate per 1 million residents.







Q4. Output of training and testing set
Q5. Train a linear regression model using the SKLearn library, with training data

I chose to execute two different training models. One training model with the independent variable being reported violent crime rate per 1 million residents and the second training model with independent variable being annual Police funding in dollars/resident. I wanted to compare both training models and I expected the first training model to be more accurate as there was a higher correlation between reported violent crime rate and total reported crime rate compared to annual police funding.

```
#Used ScikitLearn library and the train_test_split method, 25% of data is
assigned to test set; 75% to the training set
#created the training and testing variables
#from SKlearn, sub-library linear_model, imported LinearRegression
#from Sklearn, sublibrary model_selection, imported train_test_split, so c
an split to training and test sets
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
lm = LinearRegression()
print(lm)

# Split x and y; data is split into training data and test data
#Here x represents data from the independent variable "Police funding"
#y represents data from the dependent variable ""
#Training set contains a known output and model learns on the training dat
a, in orderthe model developed from training data be generalized to other
data such as testing data for predictions;Test data set or in this case, t
he subset is to test our model's prediction
#Use ScikitLearn library and the train_test_split method, 25% of data is a
ssigned to test set; 75% to the training set
```

```
#create the training and testing variables
cr_x_train, cr_x_test, cr_y_train, cr_y_test = train_test_split(cr_x3, cr_
y, test_size=0.25, random_state=1)
print (cr_x_train.shape, cr_y_train.shape)
print (cr_x_test.shape, cr_y_test.shape)

# Training the Algorithm
# Important step:Fit the model on the training data
lm.fit(cr_x_train, cr_y_train)

model = lm.fit(cr_x_train, cr_y_train)
```

When reported violent crime rate was taken as the independent variable, the output was:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
normalize=False)
(37, 1) (37,)
(13, 1) (13,)
 y = 443.9255560919859 + x * 0.5056734703085404
```

When annual police funding was taken as the independent variable, the output was:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None,
normalize=False)
(37, 1) (37,)
(13, 1) (13,)
 y = 431.47208579594326 + x * 7.257484946863121
```

Q6. Predictions from testing data set

When reported violent crime rate was taken as the independent variable, the output for predicted values based on the training model was:

```
array([ 920.26996512, 1211.03221055, 2236.53800834, 1139.22657777,
        619.39425029,  729.63106682,  960.72384275,  553.15102568,
        675.01833202,  944.03661823,  762.49984239, 1019.88763877,
        554.16237262])
```

When annual police funding was taken as the independent variable, the output for predicted values based on the training model was:

```
array([ 714.51399872,  750.80142346, 1055.61579123,  591.13675463,
        845.14872777,  656.45411915, 1026.58585144,  692.74154388,
        750.80142346,  765.31639335,  641.93914925,  649.1966342 ,
        685.48405894])
```

	Actual	Predicted
0	867	714.513999
1	732	750.801423

Actual	Predicted	
2	1740	1055.615791
3	989	591.136755
4	643	845.148728
5	341	656.454119
6	1324	1026.585851
7	462	692.741544
8	715	750.801423
9	805	765.316393
10	652	641.939149
11	821	649.196634
12	357	685.484059

Q7. Performance of the model with Mean Squared Error

When reported violent crime rate was taken as the independent variable, mean squared error was 6905 and accuracy of the training model was 49%.

```
Mean Absolute Error: 210.09991547818552
Mean Squared Error: 69605.0926097043
Root Mean Squared Error: 263.82777073254493

train_score 0.48357601836040126
test_score 0.4911481442420067
0.4911481442420067
```

When annual police funding was taken as the independent variable, mean squared error was 82517 and accuracy of the training model was 40%.

```
Mean Absolute Error: 221.93282490976657
Mean Squared Error: 82517.73472011375
Root Mean Squared Error: 287.2590028530242

train_score 0.08821677750141366
test_score 0.39674956427798014
0.39674956427798014
```

Accuracy of the training model when reported violent crime rate was taken as the independent variable compared to annual police funding, because of the higher correlation of violent crime rate with reported crime rate.