Digital Assessment I

Name: AMAN KUMAR

Roll No: 21BDS0241

GitHub Link: https://github.com/Aman88097/EDA_21BD0241

Dataset Name: countymurders.csv

Dataset Link:
https://raw.githubusercontent.com/salemprakash/EDA/main/Data/countymurders.csv

Importing Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
```

Step 1: Load the Dataset

```
# Load the dataset
url =
"https://raw.githubusercontent.com/salemprakash/EDA/main/Data/countymu
rders.csv"
df = pd.read_csv(url)

# Display structure
df.head()
df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37349 entries, 0 to 37348
Data columns (total 21 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   rownames     37349 non-null  int64  
 1   arrests      36845 non-null  float64
```

```
 2   countyid     37349 non-null  int64
 3   density      37349 non-null  float64
 4   popul        37349 non-null  int64
 5   perc1019     37349 non-null  float64
 6   perc2029     37349 non-null  float64
 7   percblack    37349 non-null  float64
 8   percmale     37349 non-null  float64
 9   rpcincmaint  37346 non-null  float64
 10  rpcpersinc   37346 non-null  float64
 11  rpcunemins   37346 non-null  float64
 12  year         37349 non-null  int64
 13  murders      37349 non-null  int64
 14  murdrate     37349 non-null  float64
 15  arrestrate   36845 non-null  float64
 16  statefips    37349 non-null  int64
 17  countyfips   37349 non-null  int64
 18  execs        37349 non-null  int64
 19  lpopul       37349 non-null  float64
 20  execrate     37349 non-null  float64
dtypes: float64(13), int64(8)
memory usage: 6.0 MB
```

Step 2: Data Cleaning

```
# Check missing values
print("Missing Values:\n", df.isnull().sum())

# Handle missing values
df.fillna(df.mean(numeric_only=True), inplace=True)

# Drop duplicates
df.drop_duplicates(inplace=True)

# Convert categorical columns if any
df = pd.get_dummies(df, drop_first=True)

# Confirm cleaning
df.info()
```

```
Missing Values:
 rownames         0
arrests        504
countyid         0
density          0
popul            0
perc1019         0
perc2029         0
percblack        0
```

```
percmale         0
rpcincmaint      3
rpcpersinc       3
rpcunemins       3
year             0
murders          0
murdrate         0
arrestrate     504
statefips        0
countyfips       0
execs            0
lpopul           0
execrate         0
dtype: int64
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37349 entries, 0 to 37348
Data columns (total 21 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   rownames     37349 non-null  int64  
 1   arrests      37349 non-null  float64
 2   countyid     37349 non-null  int64  
 3   density      37349 non-null  float64
 4   popul        37349 non-null  int64  
 5   perc1019     37349 non-null  float64
 6   perc2029     37349 non-null  float64
 7   percblack    37349 non-null  float64
 8   percmale     37349 non-null  float64
 9   rpcincmaint  37349 non-null  float64
 10  rpcpersinc   37349 non-null  float64
 11  rpcunemins   37349 non-null  float64
 12  year         37349 non-null  int64  
 13  murders      37349 non-null  int64  
 14  murdrate     37349 non-null  float64
 15  arrestrate   37349 non-null  float64
 16  statefips    37349 non-null  int64  
 17  countyfips   37349 non-null  int64  
 18  execs        37349 non-null  int64  
 19  lpopul       37349 non-null  float64
 20  execrate     37349 non-null  float64
dtypes: float64(13), int64(8)
memory usage: 6.0 MB
```

Step 3: Data Handling

```python
# Display basic statistics
print(df.describe())
```

```python
# Check correlation
print(df.corr())

# Detect outliers using Interquartile Range (IQR)
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1

# Remove outliers
df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 *
IQR))).any(axis=1)]
print("Data after removing outliers:", df.shape)
```

```
            rownames       arrests      countyid       density
popul  \
count  37349.000000  37349.000000  37349.000000  37349.000000
3.734900e+04
mean   18675.000000      6.782250  32921.927173    252.241067
8.934354e+04
std    10781.871939     49.789272  15528.352966   1663.768484
2.718545e+05
min        1.000000      0.000000   1001.000000      0.050000
8.500000e+01
25%     9338.000000      0.000000  20105.000000     17.681580
1.314400e+04
50%    18675.000000      1.000000  36065.000000     44.240000
2.879200e+04
75%    28012.000000      3.000000  48049.000000    106.600000
6.648000e+04
max    37349.000000   2391.000000  56045.000000  54058.770000
9.127751e+06

            perc1019      perc2029     percblack      percmale
rpcincmaint  \
count  37349.000000  37349.000000  37349.000000  37349.000000
37349.000000
mean      15.582640     14.584615      7.823194     43.350958
165.450844
std        1.973399      3.696407     13.287067      3.717612
97.485046
min        7.080000      5.617244      0.000000     35.150000
5.490000
25%       14.320000     12.301700      0.200000     40.900000
96.250000
50%       15.420000     14.270000      1.450000     41.810000
145.170000
75%       16.730000     16.200000      8.740000     45.870000
209.880000
max       30.484580     40.520000     86.279340     78.040000
1306.496000
```

```
       ...   rpcunemins          year       murders      murdrate  \
count  ...  37349.000000  37349.000000  37349.000000  37349.000000
mean   ...     70.557953   1988.000000      7.286915      0.508202
std    ...     52.907268      4.899045     47.217586      0.851044
min    ...      0.000000   1980.000000      0.000000      0.000000
25%    ...     35.200000   1984.000000      0.000000      0.000000
50%    ...     57.100000   1988.000000      1.000000      0.241044
75%    ...     89.958000   1992.000000      3.000000      0.735294
max    ...    642.730000   1996.000000   1944.000000     39.840640

         arrestrate     statefips    countyfips         execs
lpopul  \
count  37349.000000  37349.000000  37349.000000  37349.000000
37349.000000
mean       0.511486     32.821575    100.352299      0.006854
10.347289
std        1.224217     15.503684    107.942699      0.112448
1.327145
min        0.000000      1.000000      1.000000      0.000000
4.442651
25%        0.000000     20.000000     33.000000      0.000000
9.483721
50%        0.177154     36.000000     75.000000      0.000000
10.267850
75%        0.691197     48.000000    127.000000      0.000000
11.104660
max      148.658400     56.000000    840.000000      7.000000
16.026830

           execrate
count  37349.000000
mean       0.001042
std        0.029068
min        0.000000
25%        0.000000
50%        0.000000
75%        0.000000
max        2.388916

[8 rows x 21 columns]
             rownames   arrests      countyid   density     popul
perc1019  \
rownames     1.000000 -0.042387  9.874345e-01  0.002649 -0.060608
0.039989
arrests     -0.042387  1.000000 -4.581712e-02  0.311418  0.791257 -
0.054905
countyid     0.987434 -0.045817  1.000000e+00  0.010010 -0.061212
0.036127
density      0.002649  0.311418  1.000971e-02  1.000000  0.380315 -
```

```
0.135204
popul       -0.060608  0.791257 -6.121241e-02  0.380315  1.000000 -
0.119220
perc1019     0.039989 -0.054905  3.612727e-02 -0.135204 -0.119220
1.000000
perc2029    -0.005638  0.097409 -7.710277e-03  0.126121  0.185030
0.256169
percblack   -0.013782  0.118160 -2.857313e-02  0.129959  0.084739
0.122132
percmale     0.001813 -0.020178 -1.443695e-03 -0.020912 -0.013728 -
0.208278
rpcincmaint -0.037361  0.142790 -5.677194e-02  0.164865  0.099991
0.098407
rpcpersinc  -0.051810  0.133192 -4.822744e-02  0.270458  0.331269 -
0.462651
rpcunemins  -0.078840  0.025265 -7.233701e-02  0.034143  0.044673
0.066413
year         0.000454  0.001576  3.461751e-13  0.003468  0.014997 -
0.382645
murders     -0.039936  0.905295 -4.245941e-02  0.433851  0.877110 -
0.073315
murdrate     0.030421  0.170679  1.384874e-02  0.137594  0.118712
0.046229
arrestrate   0.021427  0.154936  5.991958e-03  0.061239  0.064456
0.036582
statefips    0.987121 -0.045881  9.999771e-01  0.009837 -0.060965
0.035966
countyfips   0.270752 -0.001236  2.318303e-01  0.027039 -0.049541
0.031413
execs        0.020697  0.083565  1.865559e-02  0.022459  0.160758 -
0.019806
lpopul      -0.046460  0.295137 -3.736177e-02  0.280124  0.557064 -
0.078025
execrate     0.011430 -0.000748  9.194172e-03 -0.000814 -0.003113 -
0.011402

             perc2029  percblack  percmale  rpcincmaint  ...
rpcunemins  \
rownames    -0.005638  -0.013782  0.001813    -0.037361  ...   -
0.078840
arrests      0.097409   0.118160 -0.020178     0.142790  ...
0.025265
countyid    -0.007710  -0.028573 -0.001444    -0.056772  ...   -
0.072337
density      0.126121   0.129959 -0.020912     0.164865  ...
0.034143
popul        0.185030   0.084739 -0.013728     0.099991  ...
0.044673
perc1019     0.256169   0.122132 -0.208278     0.098407  ...
```

```
0.066413
perc2029      1.000000   0.174507 -0.223615    -0.146472  ...
0.032846
percblack     0.174507   1.000000 -0.104799     0.391585  ...   -
0.129341
percmale     -0.223615  -0.104799  1.000000     0.113600  ...   -
0.088447
rpcincmaint -0.146472   0.391585  0.113600     1.000000  ...
0.189517
rpcpersinc   -0.006004  -0.130672  0.223169    -0.279905  ...   -
0.082969
rpcunemins    0.032846  -0.129341 -0.088447     0.189517  ...
1.000000
year         -0.428285   0.011903  0.660332     0.245385  ...   -
0.202772
murders       0.117278   0.139215 -0.025080     0.157738  ...
0.025989
murdrate      0.074479   0.312546 -0.064556     0.199867  ...   -
0.024076
arrestrate    0.042012   0.225487 -0.037317     0.151704  ...   -
0.030825
statefips    -0.008012  -0.029168 -0.001293    -0.056677  ...   -
0.071220
countyfips    0.041639   0.078872 -0.021957    -0.026593  ...   -
0.177027
execs         0.026597   0.047100  0.028234     0.005941  ...   -
0.018663
lpopul        0.424437   0.148421 -0.082901     0.056214  ...
0.127596
execrate     -0.011871   0.020806  0.029087     0.007829  ...   -
0.014464

                     year   murders  murdrate  arrestrate
statefips  \
rownames      4.543780e-04 -0.039936  0.030421    0.021427  9.871205e-
01
arrests       1.575601e-03  0.905295  0.170679    0.154936 -4.588142e-
02
countyid      3.461751e-13 -0.042459  0.013849    0.005992  9.999771e-
01
density       3.468035e-03  0.433851  0.137594    0.061239  9.837386e-
03
popul         1.499733e-02  0.877110  0.118712    0.064456 -6.096488e-
02
perc1019     -3.826454e-01 -0.073315  0.046229    0.036582  3.596605e-
02
perc2029     -4.282845e-01  0.117278  0.074479    0.042012 -8.012452e-
03
percblack     1.190266e-02  0.139215  0.312546    0.225487 -2.916773e-
```

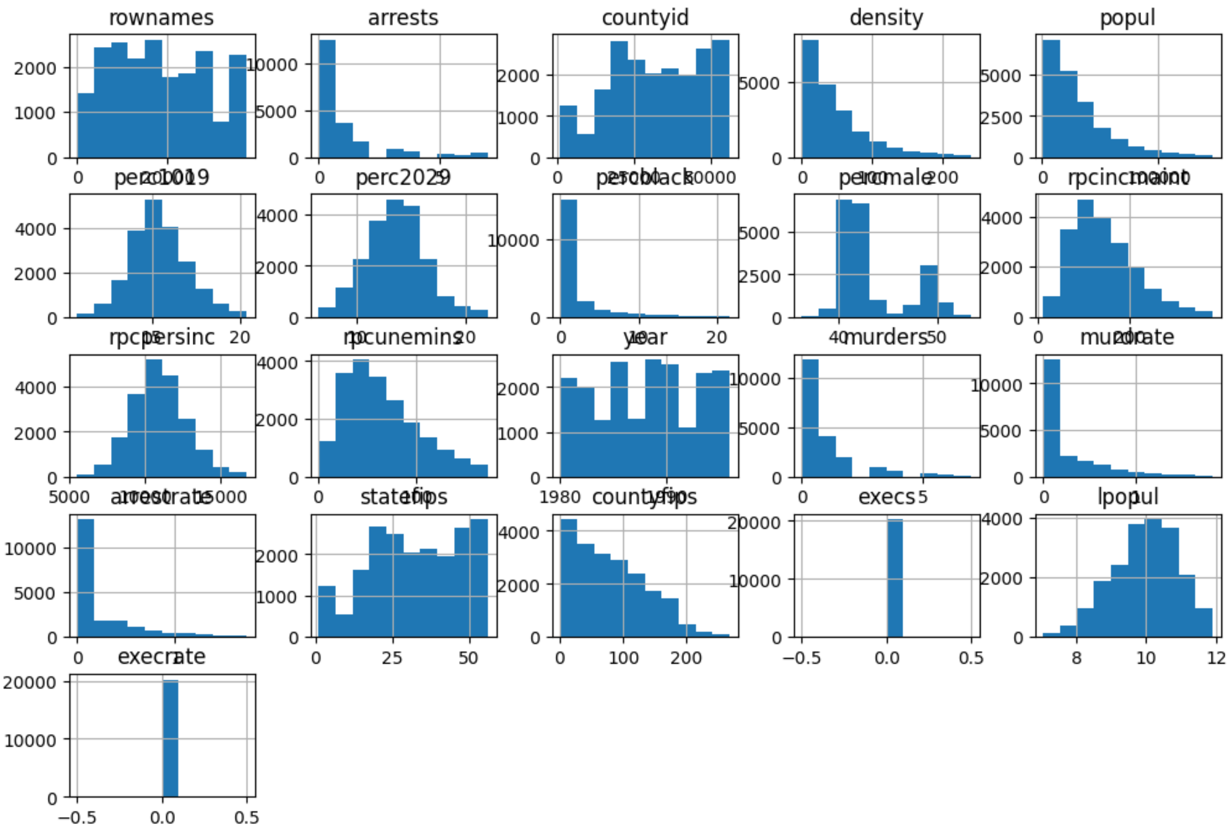
```
02
percmale     6.603317e-01 -0.025080 -0.064556   -0.037317 -1.293116e-
03
rpcincmaint  2.453846e-01  0.157738  0.199867    0.151704 -5.667713e-
02
rpcpersinc   2.927135e-01  0.170730 -0.035015   -0.040336 -4.823532e-
02
rpcunemins  -2.027722e-01  0.025989 -0.024076   -0.030825 -7.121958e-
02
year         1.000000e+00  0.003815 -0.048111   -0.024652  3.456787e-
13
murders      3.814626e-03  1.000000  0.212655    0.119755 -4.247505e-
02
murdrate    -4.811082e-02  0.212655  1.000000    0.448823  1.318676e-
02
arrestrate  -2.465185e-02  0.119755  0.448823    1.000000  5.431574e-
03
statefips    3.456787e-13 -0.042475  0.013187    0.005432
1.000000e+00
countyfips   1.496656e-13 -0.007458  0.098245    0.081857  2.252368e-
01
execs        4.452117e-02  0.155903  0.042837    0.017847  1.844054e-
02
lpopul       1.689796e-02  0.343957  0.068035    0.034184 -3.678215e-
02
execrate     3.038773e-02 -0.000010  0.005167    0.004100  9.045622e-
03

               countyfips     execs    lpopul  execrate
rownames     2.707525e-01  0.020697 -0.046460  0.011430
arrests     -1.235829e-03  0.083565  0.295137 -0.000748
countyid     2.318303e-01  0.018656 -0.037362  0.009194
density      2.703860e-02  0.022459  0.280124 -0.000814
popul       -4.954060e-02  0.160758  0.557064 -0.003113
perc1019     3.141287e-02 -0.019806 -0.078025 -0.011402
perc2029     4.163893e-02  0.026597  0.424437 -0.011871
percblack    7.887198e-02  0.047100  0.148421  0.020806
percmale    -2.195738e-02  0.028234 -0.082901  0.029087
rpcincmaint -2.659270e-02  0.005941  0.056214  0.007829
rpcpersinc  -9.889311e-03  0.050651  0.376050 -0.000379
rpcunemins  -1.770272e-01 -0.018663  0.127596 -0.014464
year         1.496656e-13  0.044521  0.016898  0.030388
murders     -7.457649e-03  0.155903  0.343957 -0.000010
murdrate     9.824459e-02  0.042837  0.068035  0.005167
arrestrate   8.185673e-02  0.017847  0.034184  0.004100
statefips    2.252368e-01  0.018441 -0.036782  0.009046
countyfips   1.000000e+00  0.035151 -0.091789  0.023437
execs        3.515088e-02  1.000000  0.096420  0.350218
lpopul      -9.178859e-02  0.096420  1.000000 -0.009481
```

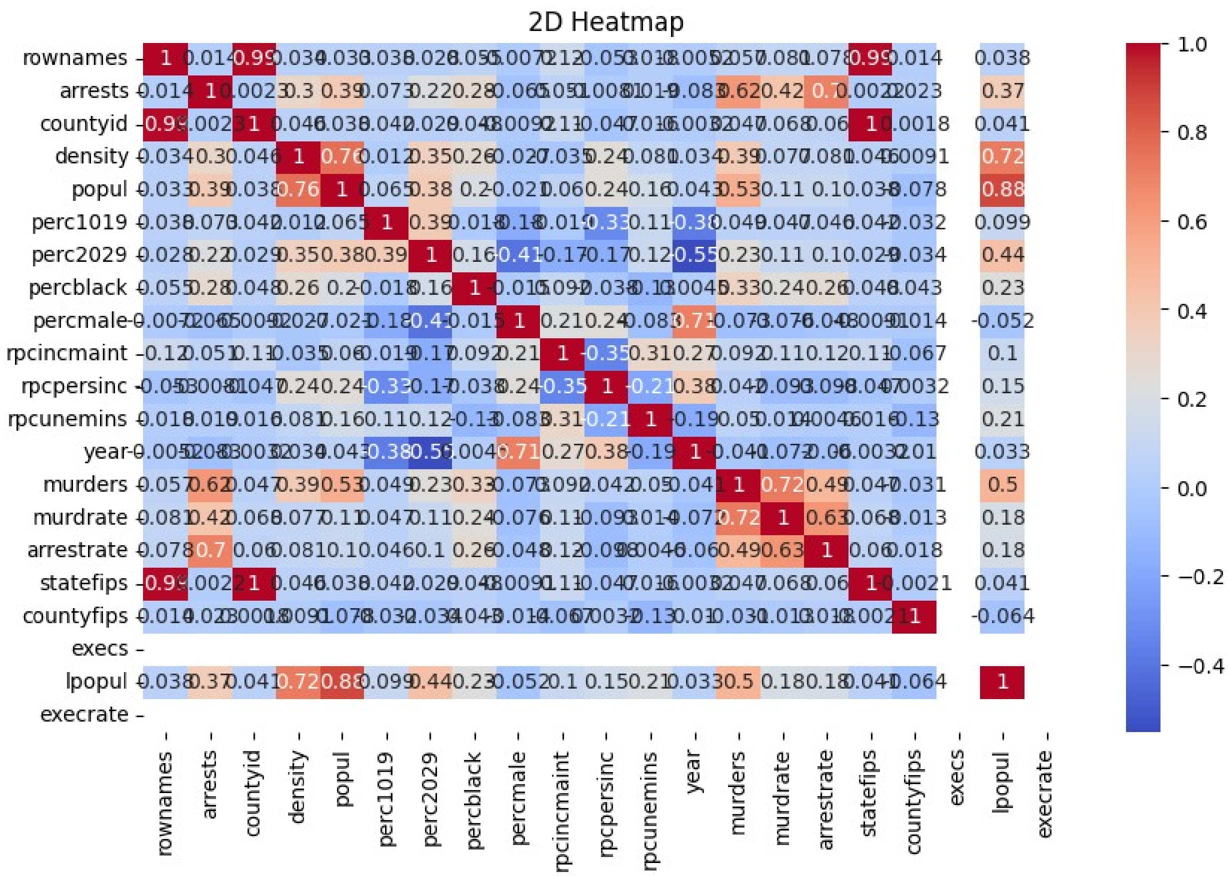
```
execrate     2.343731e-02  0.350218 -0.009481  1.000000

[21 rows x 21 columns]
Data after removing outliers: (20206, 21)
```

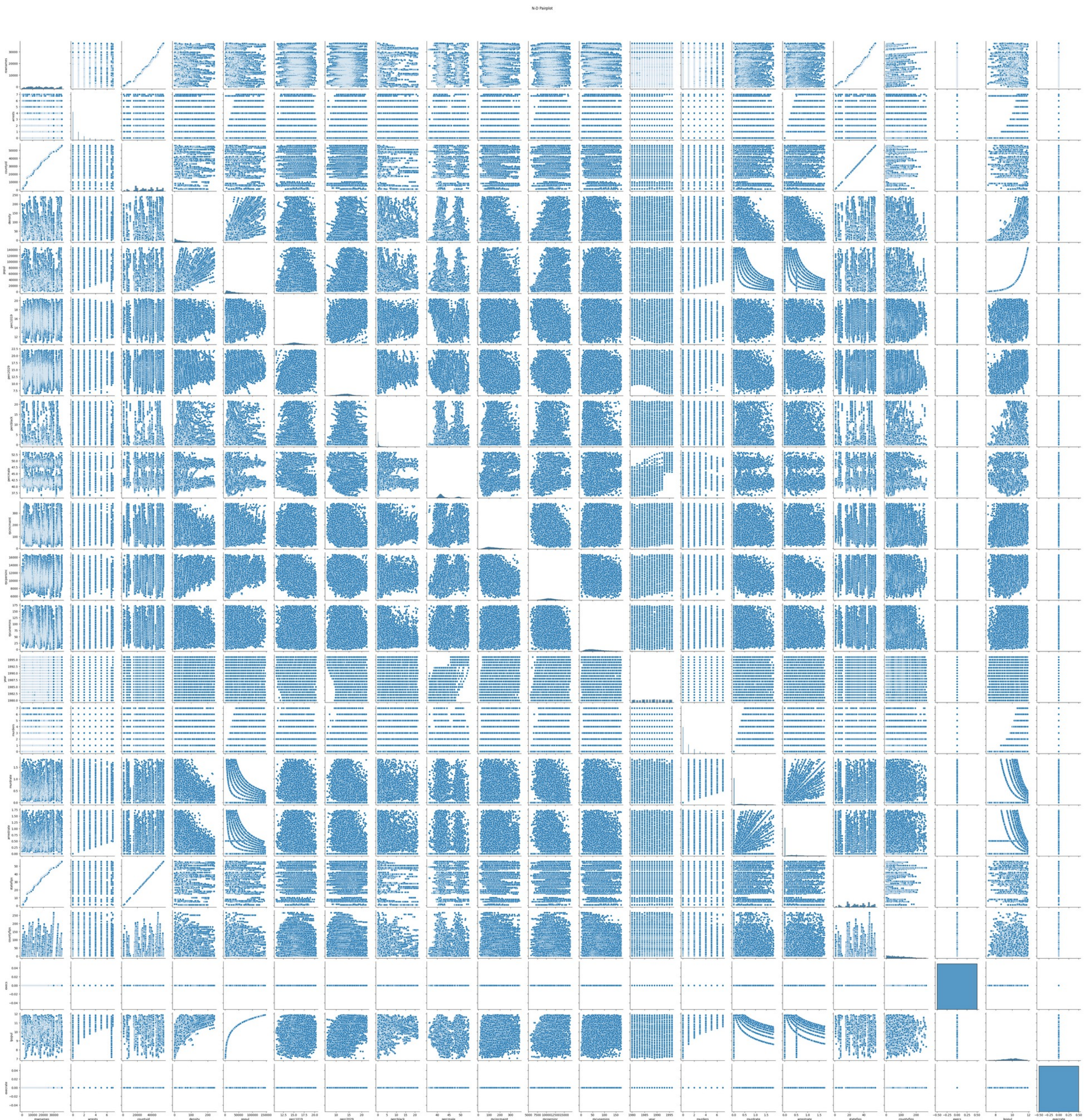
Step 4: 1D, 2D, and N-D Visualization

```python
# 1D: Histogram
df.hist(figsize=(12, 8))
plt.suptitle("1D Histograms")
plt.show()

# 2D: Correlation Heatmap
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("2D Heatmap")
plt.show()

# N-D: Pairplot
sns.pairplot(df)
plt.suptitle("N-D Pairplot", y=1.02)
plt.show()
```

1D Histograms



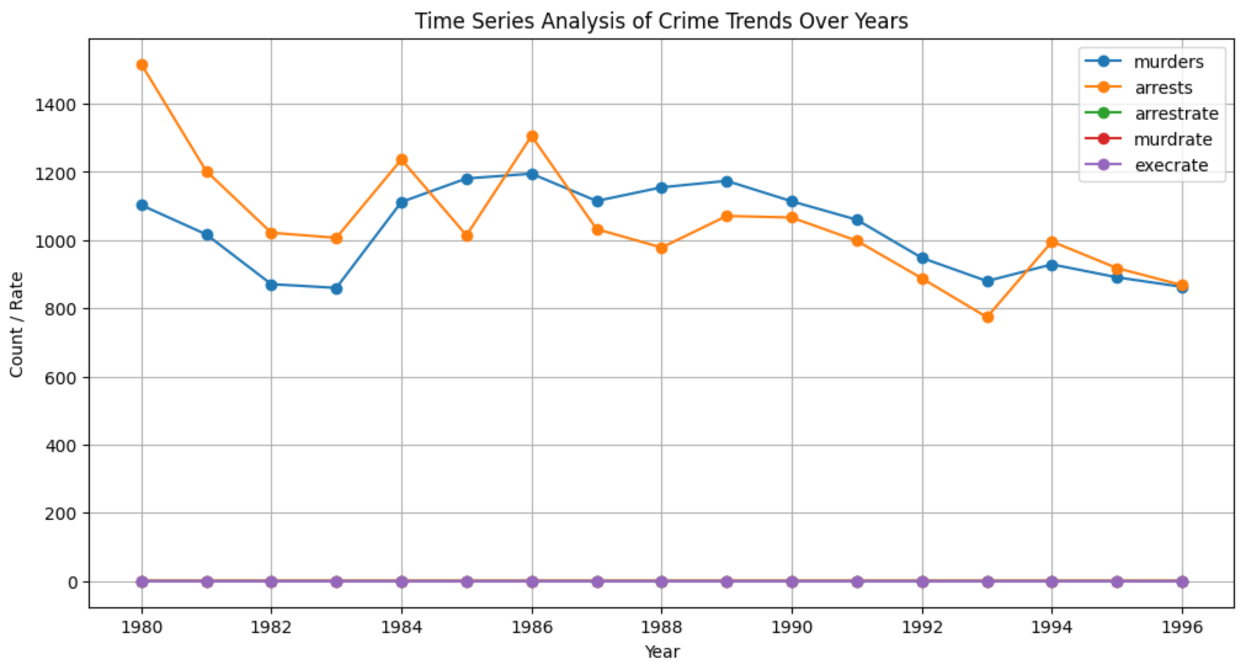

2D Heatmap
rownames
arrests
countyid
density
popul
perc1019
perc2029
percblack
percmale
rpcincmaint
rpcpersinc
rpcunemins
year
murders
murdrate
arrestrate
statefips
countyfips
execs
lpopul
execrate
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4



Step 5: Time-Series Analysis

```
# Group data by 'year' and calculate the sum or mean of relevant
metrics
yearly_data = df.groupby('year').agg({
    'murders': 'sum',
    'arrests': 'sum',
    'arrestrate': 'mean',
    'murdrate': 'mean',
    'execrate': 'mean'
```

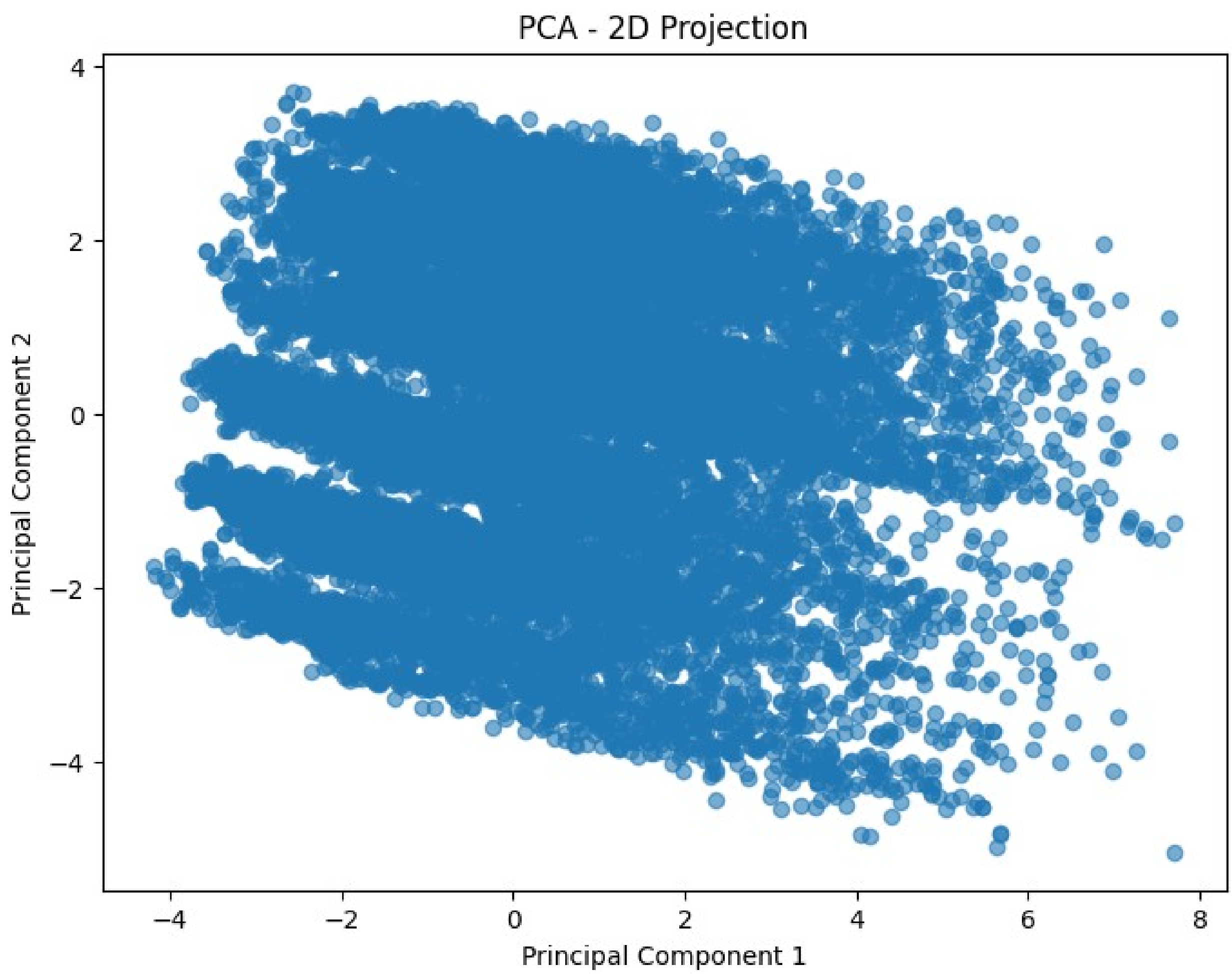
```
})

# Plot time series trends
plt.figure(figsize=(12, 6))
for col in yearly_data.columns:
    plt.plot(yearly_data.index, yearly_data[col], marker='o',
label=col)

plt.title("Time Series Analysis of Crime Trends Over Years")
plt.xlabel("Year")
plt.ylabel("Count / Rate")
plt.legend()
plt.grid(True)
plt.show()
```



Step 6: Dimensionality Reduction (PCA)

```
# Standardize features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df)

# Apply PCA
pca = PCA(n_components=2)
pca_result = pca.fit_transform(X_scaled)

# PCA Visualization
plt.figure(figsize=(8, 6))
```

```
plt.scatter(pca_result[:, 0], pca_result[:, 1], alpha=0.6)
plt.title("PCA - 2D Projection")
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.show()
```



Step 7: Model Building

```
# Create a binary target: high murder rate (above median)
df['high_murder_rate'] = (df['murdrate'] >
df['murdrate'].median()).astype(int)

# Drop columns not useful for modeling
drop_cols = ['rownames', 'murdrate', 'statefips', 'countyfips']  #
also dropping the original murdrate
X = df.drop(columns=drop_cols + ['high_murder_rate'])
y = df['high_murder_rate']
```

```python
# Scale features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Train-test split
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y,
test_size=0.2, random_state=42)

# Train a Random Forest with reduced complexity
model = RandomForestClassifier(n_estimators=50, max_depth=10,
random_state=42)
model.fit(X_train, y_train)

# Predict
y_pred = model.predict(X_test)
```

Step 8: Model Evaluation

```python
# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test,
y_pred))
```

```
Accuracy: 1.0
Classification Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00      2354
           1       1.00      1.00      1.00      1688

    accuracy                           1.00      4042
   macro avg       1.00      1.00      1.00      4042
weighted avg       1.00      1.00      1.00      4042
```