DOCUMENTED BY RAKIB AHMED

Statistics: Statistics is a branch of mathematics concerned with collecting, analyzing, interpreting, presenting, and organizing data.

Types of statistics include Descriptive statistics, which summarize and describe features of a dataset, and Inferential statistics, which involve making inferences or predictions about a population based on a sample.

Here we'll learn some descriptive statistics term and how to use them by using Pyhton.

Some Important Libraries:

1. `pandas` : Data manipulation and analysis, e.g., loading datasets, performing data transformations.
2. `scipy.stats.trim_mean` : Calculating the trimmed mean, which removes a specified proportion of outliers from a dataset.
3. `numpy` : Mathematical operations on arrays and matrices, commonly used for numerical computing.
4. `matplotlib.pyplot` : Data visualization, particularly plotting graphs and charts.
5. `seaborn` : Statistical data visualization, providing a high-level interface for drawing attractive and informative statistical graphics.

In [23]:

```
import pandas as pd
from scipy.stats import trim_mean
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

We Create a Example Dataset for our Analysis.

In [24]:

```
#Create Dataframe
State = {"State":["Alabama", "Alaska", "Arizona","Arkansas","California","Colorado","Con
"Population":[4779736,710231,6392017,2915918,37253956,5029196,3574097,897934],
"Murder rate": [5.7,5.6,4.7,5.6,4.4,2.8,2.4,5.8],
"Abbreviation":["AL","AK","AZ","AR","CA","CO","CT","DE"]
}
```

In [25]:

```
state = pd.DataFrame(State)
print(state)
```

```
         State  Population  Murder rate Abbreviation
0      Alabama     4779736          5.7           AL
1       Alaska      710231          5.6           AK
2      Arizona     6392017          4.7           AZ
3     Arkansas     2915918          5.6           AR
4   California    37253956          4.4           CA
5     Colorado     5029196          2.8           CO
6  Connecticut     3574097          2.4           CT
7     Delaware      897934          5.8           DE
```

The mean is the average of a set of numbers, used to summarize central tendency or the typical value within a dataset.

In [26]:

```
#Mean
print("Mean:",state['Population'].mean())
```

```
Mean: 7694135.625
```

Trimmed mean is a statistical measure that calculates the mean after removing a specified proportion of extreme values, useful for reducing the influence of outliers on the average.

In [27]:

```python
#Trimmed_mean
print("Trimmed_mean:", trim_mean(state['Population'],0.1))
```

```
Trimmed_mean: 7694135.625
```

The median is the middle value of a dataset, used to describe the central tendency and is less sensitive to outliers than the mean.

In [28]:

```python
#median
print("Median:", state['Population'].median())
```

```
Median: 4176916.5
```

Mode is the value that appears most frequently in a dataset, used to identify the most common observation or category.

In [29]:

```python
#mode
print("Mode:",state['Population'].mode())
```

```
Mode: 0      710231
1       897934
2      2915918
3      3574097
4      4779736
5      5029196
6      6392017
7     37253956
Name: Population, dtype: int64
```

A quantile is a value that divides a dataset into equal-sized parts, used to understand the distribution and variability of the data.

In [30]:

```python
#quantile
print("Quantile:", state['Population'].quantile([0.25,0.5,0.75]))
```

```
Quantile: 0.25    2411422.00
0.50    4176916.50
0.75    5369901.25
Name: Population, dtype: float64
```

Range is the difference between the maximum and minimum values in a dataset, providing a measure of the spread or dispersion of the data.

In [31]:

```python
#range
print("Range:",state['Population'].max()-state['Population'].min())
```

```
Range: 36543725
```

Standard deviation measures the dispersion or spread of data points from the mean; it's used to understand the variability within a dataset.

In [32]:

```python
#standard deviation
print("Standard Deviation:", state['Population'].std())
```

```
Standard Deviation: 12105745.29585633
```

Variance measures the dispersion of data points around the mean, providing insight into the spread or variability of a dataset.

In [33]:

```
#variance
print("Variance:", state['Population'].var())
```

```
Variance: 146549069168147.7
```

Skewness measures the asymmetry of a probability distribution; it's used to understand the shape and symmetry of data distribution. WE can visualize skewness using histograms, density plots, or box plots, which show the distribution's asymmetry in terms of its tail direction and magnitude.

In [34]:

```
#skewness
print("Skewness:", state['Population'].skew())
```

```
Skewness: 2.678713709827244
```

Kurtosis measures the tailedness or peakedness of a probability distribution; it's used to understand the shape of a distribution's tails relative to the normal distribution.

We can visualize kurtosis using a histogram or a kernel density plot to observe the distribution's shape, or using a boxplot to identify outliers and assess tail behavior

In [35]:

```
#kurtosis
print("Kurtosis:", state['Population'].kurtosis())
```

```
Kurtosis: 7.3755721047735445
```

A percentile is a measure indicating the value below which a given percentage of observations in a group of observations falls, commonly used to understand the relative standing of an individual within a dataset or population.

In [36]:

```
#percentile
print("Percentile:", state['Population'].quantile([0.25,0.5,0.75]))
```

```
Percentile: 0.25    2411422.00
0.50    4176916.50
0.75    5369901.25
Name: Population, dtype: float64
```

The interquartile range (IQR) is a measure of statistical dispersion, representing the range between the 25th and 75th percentiles of a dataset, often used to identify variability and outliers in data.

In [37]:

```
#Interquantile
print("Interquantile:", state['Population'].quantile(0.75)- state['Population'].quantile
```

```
Interquantile: 2958479.25
```

Correlation measures the strength and direction of the linear relationship between two variables, useful for understanding how changes in one variable relate to changes in another.

In [38]:

```
#correlation
print("Correlation:", state['Population'].corr(state['Murder rate']))
```

```
Correlation: -0.13369066206896454
```

Covariance measures the degree to which two variables change together, indicating the direction of their linear relationship in a dataset. We use it to understand how changes in one variable correspond to changes in another variable.

In [39]:

```
#covariance
```

```
print("Covariance:", state['Population'].cov(state['Murder rate']))
```

```
Covariance: -2186371.9178571403
```

Creating a frequency table with binned population data allows us to understand the distribution of population across different intervals or ranges, providing insights into population density and patterns within the dataset.

In [40]:

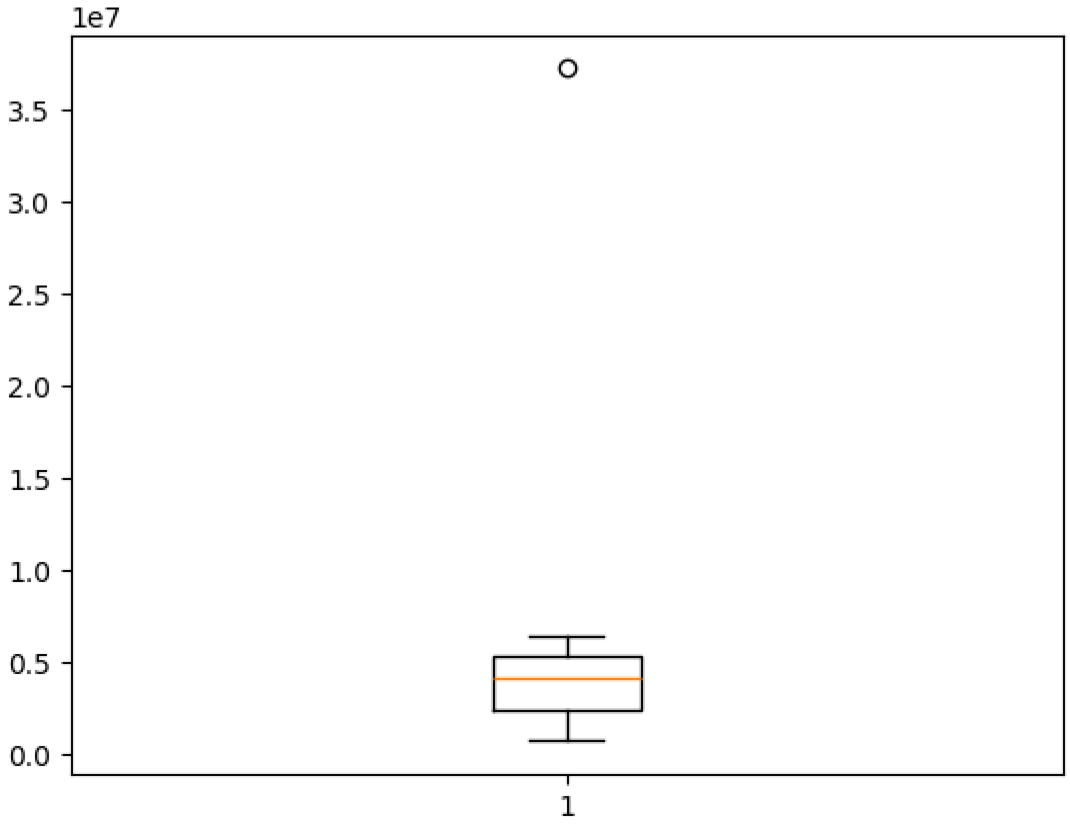
```
#Frequency table
binnedpopulation = pd.cut(state['Population'], 10)
print(binnedpopulation)
print(binnedpopulation.value_counts())
```

```
0       (4364603.5, 8018976.0]
1     (673687.275, 4364603.5]
2       (4364603.5, 8018976.0]
3     (673687.275, 4364603.5]
4    (33599583.5, 37253956.0]
5       (4364603.5, 8018976.0]
6     (673687.275, 4364603.5]
7     (673687.275, 4364603.5]
Name: Population, dtype: category
Categories (10, interval[float64, right]): [(673687.275, 4364603.5] < (4364603.5, 801897
6.0] < (8018976.0, 11673348.5] < (11673348.5, 15327721.0] ... (22636466.0, 26290838.5] <
(26290838.5, 29945211.0] < (29945211.0, 33599583.5] < (33599583.5, 37253956.0]]
Population
(673687.275, 4364603.5]     4
(4364603.5, 8018976.0]      3
(33599583.5, 37253956.0]    1
(8018976.0, 11673348.5]     0
(11673348.5, 15327721.0]    0
(15327721.0, 18982093.5]    0
(18982093.5, 22636466.0]    0
(22636466.0, 26290838.5]    0
(26290838.5, 29945211.0]    0
(29945211.0, 33599583.5]    0
Name: count, dtype: int64
```

A boxplot is a visual representation of the distribution of data, showing the median, quartiles, and outliers. It's useful for comparing distributions and identifying outliers. Appropriate for visualizing continuous or ordinal data.

In [41]:

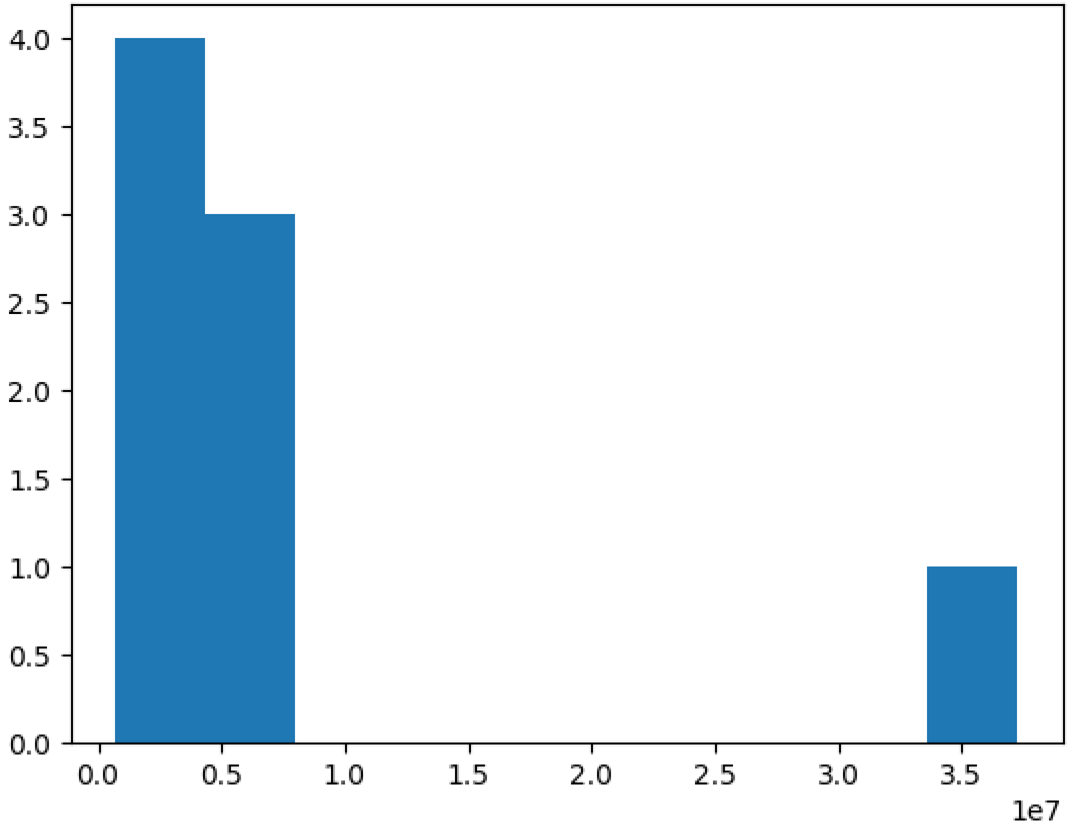
```
#boxplot
plt.boxplot(state['Population'])
plt.show()
```



A histogram is a graphical representation of the distribution of numerical data. It's used to visualize the frequency or probability distribution of a dataset. It's appropriate for continuous or discrete data, especially when exploring the distribution and frequency of values within a dataset.

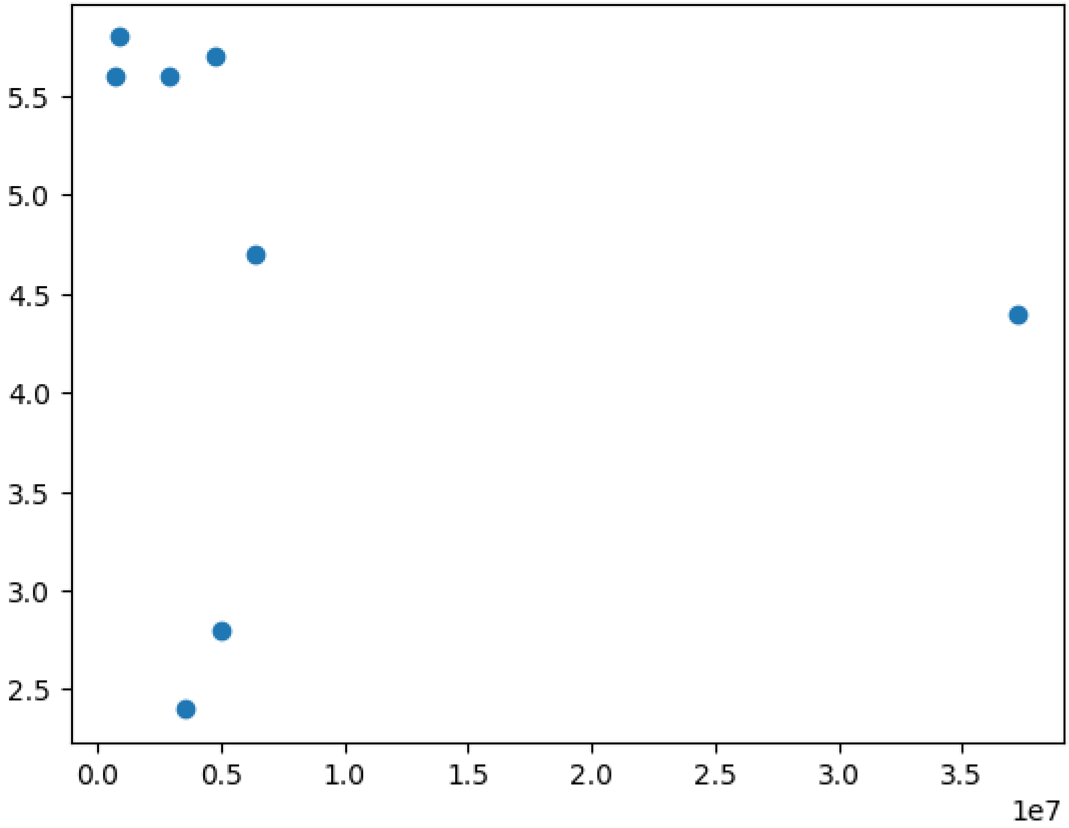
In [42]:

```
#histogram
plt.hist(state['Population'])
plt.show()
```



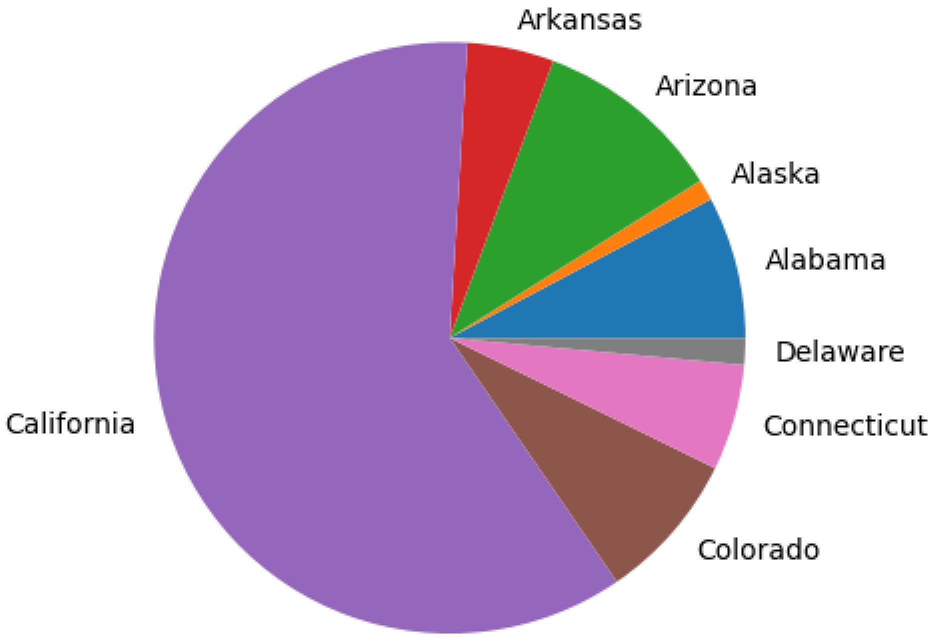
A scatterplot is a type of data visualization that displays the relationship between two numerical variables, showing their individual data points on a Cartesian plane. It's used to identify patterns or correlations between variables. It's appropriate for continuous data and is particularly useful for detecting trends, clusters, or outliers within the data.

In [43]:
```
#scatter plot
plt.scatter(state['Population'],state['Murder rate'])
plt.show()
```



A pie plot is a circular statistical graphic that displays the proportional composition of a dataset, useful for showing relative sizes of categorical variables; it's appropriate for displaying data with distinct categories and their respective proportions.

In [44]:
```
#pie chart
plt.pie(state['Population'],labels=state['State'])
plt.show()
```


Arkansas
Arizona
Alaska
Alabama
Delaware
California
Connecticut
Colorado