DOUGLAS COLLEGE

FUNDAMENTALS OF MACHINE LEARNING IN DATA SCIENCE

CSIS 3290

VISUALIZATION (MATPLOTLIB AND SEABORN)

FATEMEH AHMADI

Matplotlib and Seaborn

- Data Visualization is the graphic representation of data. It converts a huge dataset into small graphs, thus aiding in data analysis and predictions. It is an indispensable element of data science that makes complex data more understandable and accessible. Matplotlib and Seaborn act as the backbone of data visualization through Python.
- **Matplotlib:** It is a Python library used for plotting graphs with the help of other libraries like **NumPy** and **Pandas**. It is a powerful tool for visualizing data in Python. It is used for creating statistical inferences and plotting 2D graphs of arrays. It was first introduced by John D. Hunter in 2002. It uses **Pyplot** for providing MATLAB like interface free and open-source. It is capable of dealing with various operating systems and their graphical backends.
- **Seaborn:** It is also a Python library used for plotting graphs with the help of **Matplotlib**, **Pandas**, and **NumPy.** It is built on the roof of Matplotlib and is considered as a superset of the Matplotlib library. It helps in visualizing univariate and bivariate data. It uses beautiful themes for decorating Matplotlib graphics. It acts as an important tool in picturing Linear Regression Models. It serves in making graphs of statistical Time-Series data. It eliminates the overlapping of graphs and also aids in their beautification.

https://www.geeksforgeeks.org/difference-between-matplotlib-vs-seaborn/

Matplotlib and Seaborn

Features	Matplotlib	Seaborn
Functionality	It is utilized for **making basic graphs**. Datasets are visualized with the help of **barographs**, **histograms**, **pie charts**, **scatter plots**, **lines** and so on.	Seaborn contains a number of patterns and plots for data visualization. It uses **fascinating themes**. It helps in compiling whole data into a single plot. It also provides distribution of data.
Syntax	It uses **comparatively complex and lengthy syntax**. Example: Syntax for bargraph- matplotlib.pyplot.bar(x_axis, y_axis).	It uses **comparatively simple syntax** which is easier to learn and understand. Example: Syntax for bargraph- seaborn.barplot(x_axis, y_axis).
Dealing Multiple Figures	We can open and use multiple figures simultaneously. However they are closed distinctly. Syntax to close one figure at a time: matplotlib.pyplot.close(). Syntax to close all the figures: matplotlib.pyplot.close(“all”)	Seaborn sets time for the creation of each figure. However, it may lead to (OOM) out of memory issues.

https://www.geeksforgeeks.org/difference-between-matplotlib-vs-seaborn/

Matplotlib and Seaborn

Features	Matplotlib	Seaborn
Visualization	**Matplotlib is well connected with NumPy and Pandas and acts as a graphics package for data visualization in python**. PyPlot provides similar features and syntax as in **MATLAB**. Therefore, MATLAB users can easily study it.	Seaborn is more comfortable in handling Pandas data frames. It uses basic sets of methods to provide beautiful graphics in python.
Pliability	Matplotlib is a highly customized and robust.	Seaborn avoids overlapping of plots with the help of its default themes
Data Frames and Arrays	**Matplotlib works efficiently with data frames and arrays.**It treats figures and axes as objects. It contains various stateful APIs for plotting. Therefore plot() like methods can work without parameters.	**Seaborn is much more functional and organized than Matplotlib and treats the whole dataset as a single unit**. Seaborn is not so stateful and therefore, parameters are required while calling methods like plot().
Use Cases	Matplotlib plots various graphs using Pandas and NumPy.	Seaborn is **the extended version of Matplotlib** which uses Matplotlib along with NumPy and Pandas for plotting graphs

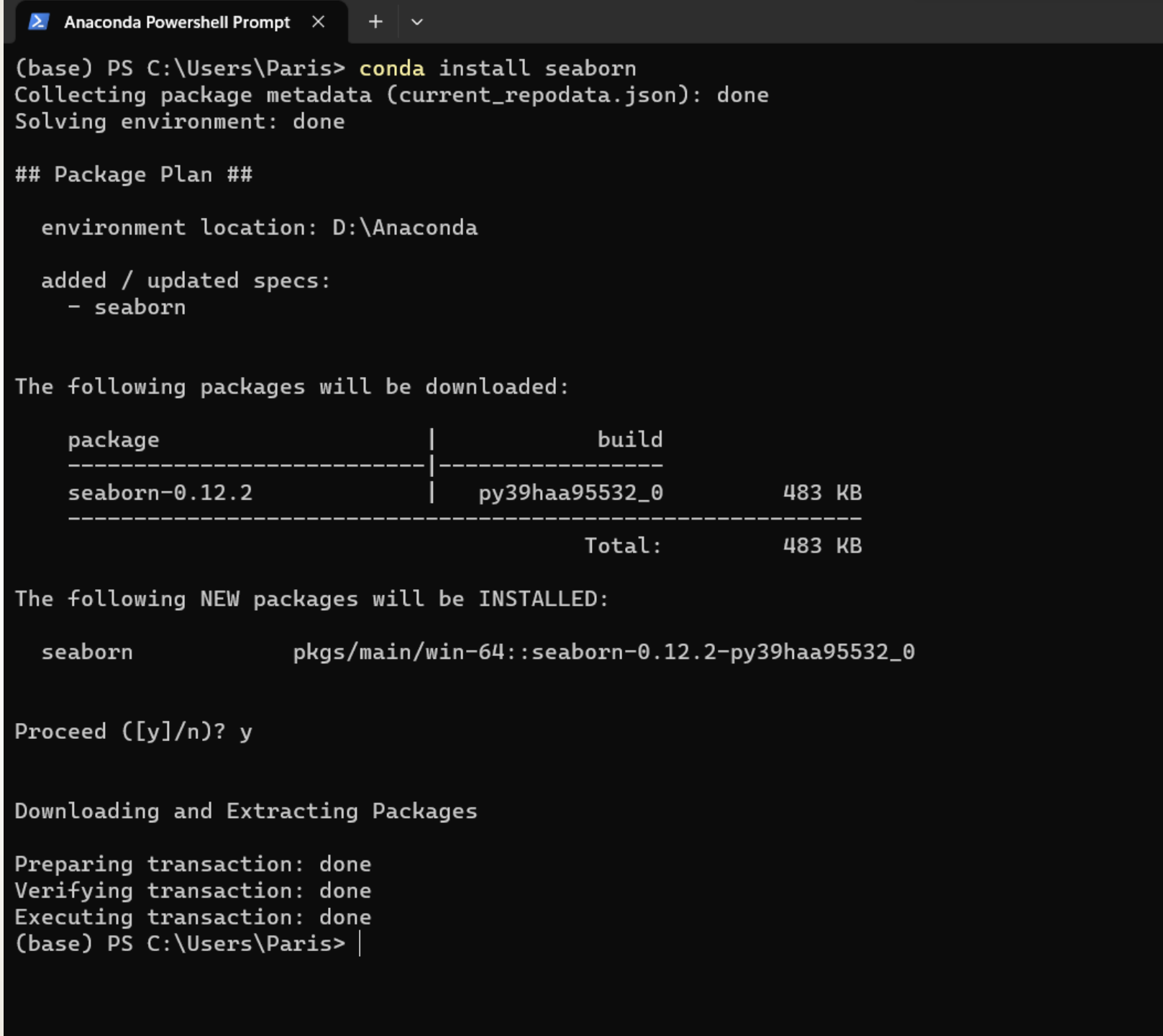
https://www.geeksforgeeks.org/difference-between-matplotlib-vs-seaborn/

Installation

In Anaconda powershell Prompt:

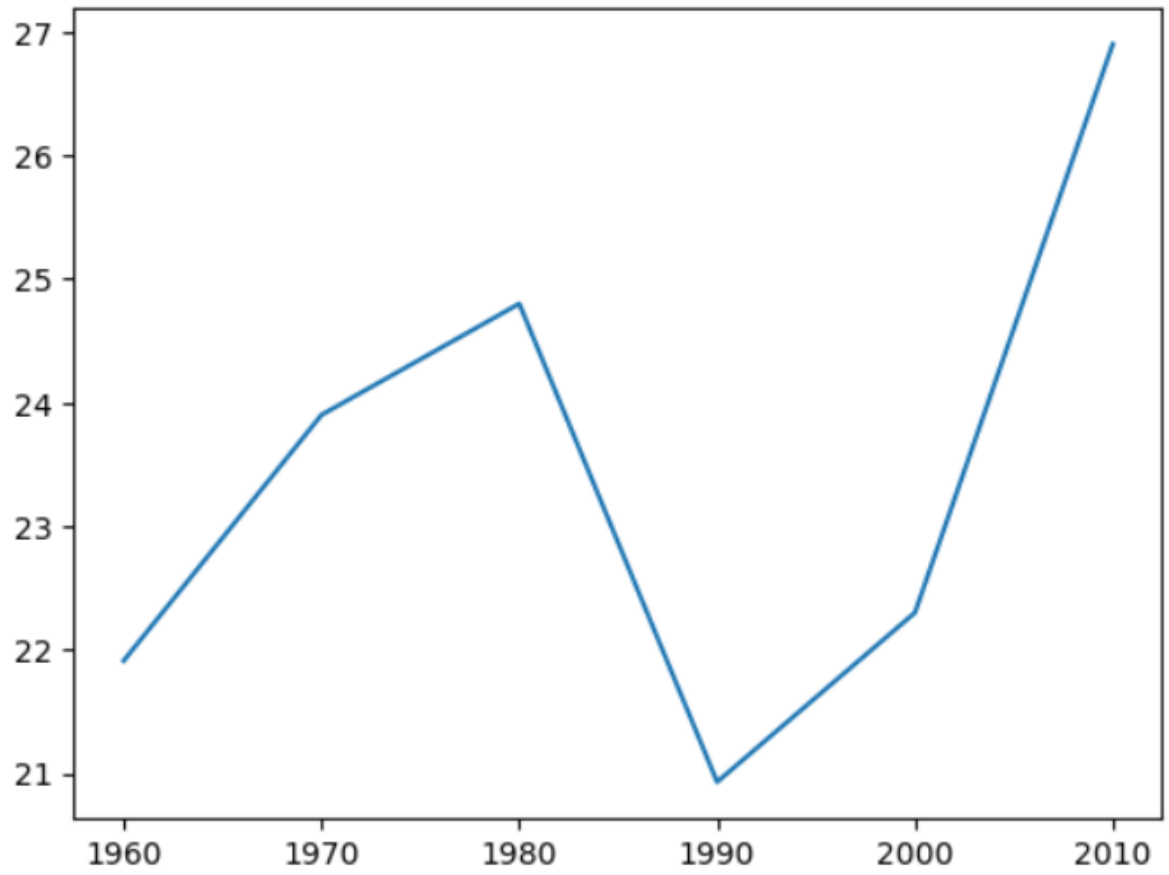
conda install matplotlib

conda install seaborn



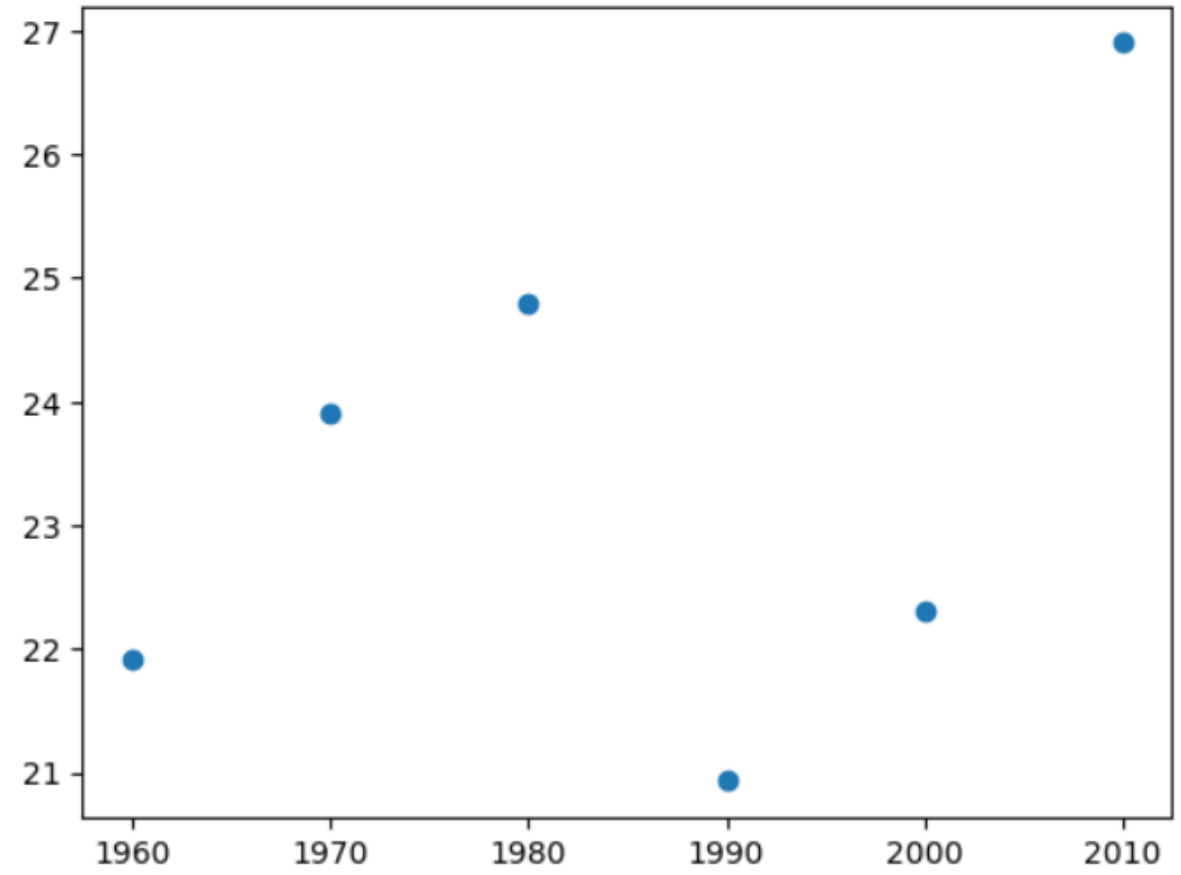
Line Plot

```
In [5]: import numpy as np
        import matplotlib.pyplot as plt

In [6]: years=[1960,1970,1980,1990,2000,2010]
        Population=[21.91,23.90,24.80,20.93,22.30,26.90]

In [7]: #line plot
        plt.plot(years,Population)
        plt.show()
```



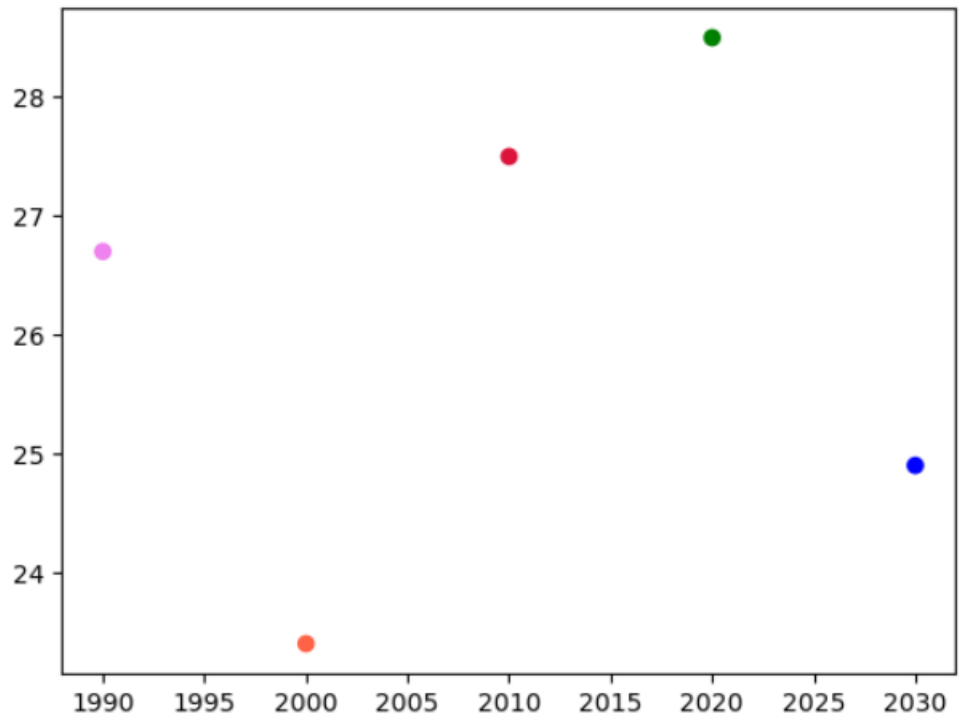
Scatter Plot

```
In [8]: #scatter plot
        plt.scatter(years,Population)
        plt.show()
```



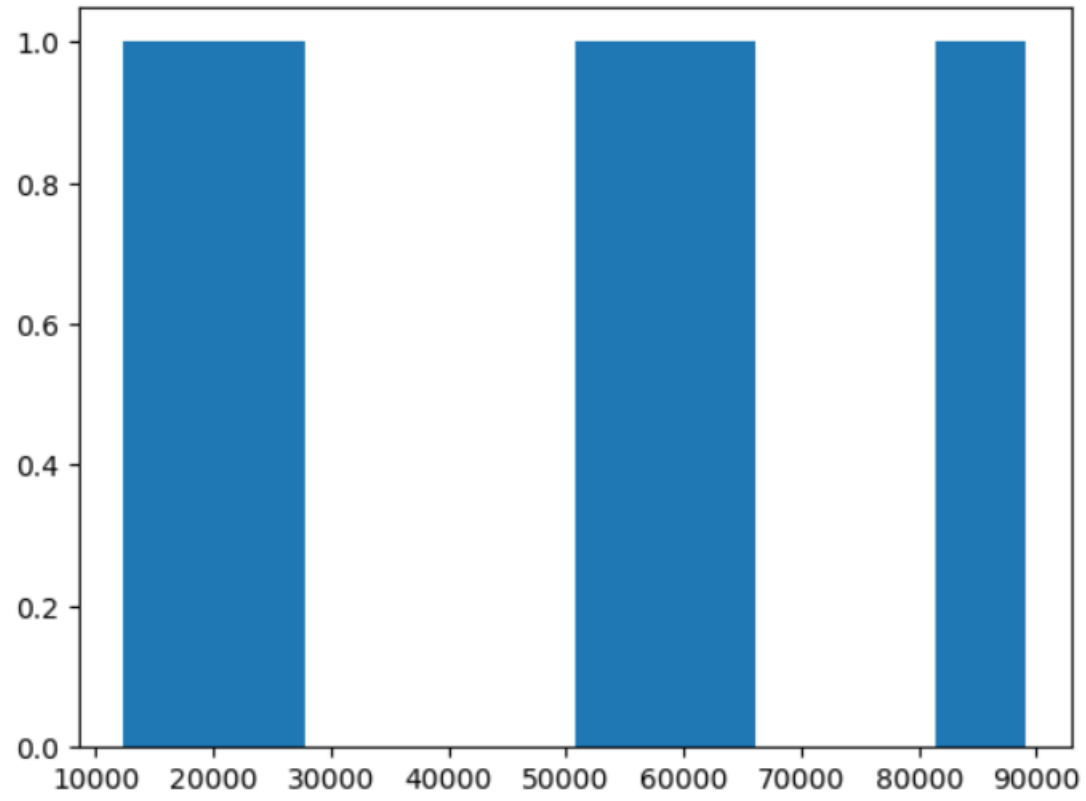
Scatter Plot

```
In [38]: colors=['violet','tomato','crimson','green','blue']
         plt.scatter(Years,Population,c=colors)
         plt.show()
```



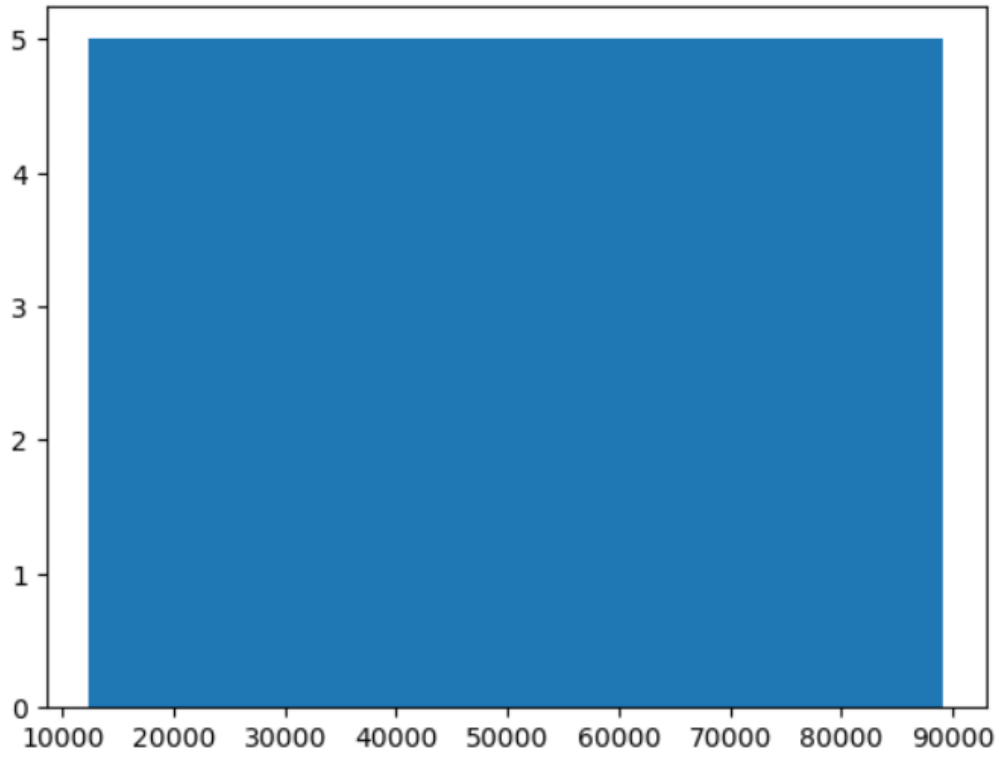
Histogram

```
In [10]: city_Name=['London','Paris','Tokyu','Beijing','Rome']
         city_Pop=[65342,89123,54239,23098,12367]

In [11]: plt.hist(city_Pop)
         plt.show()
```

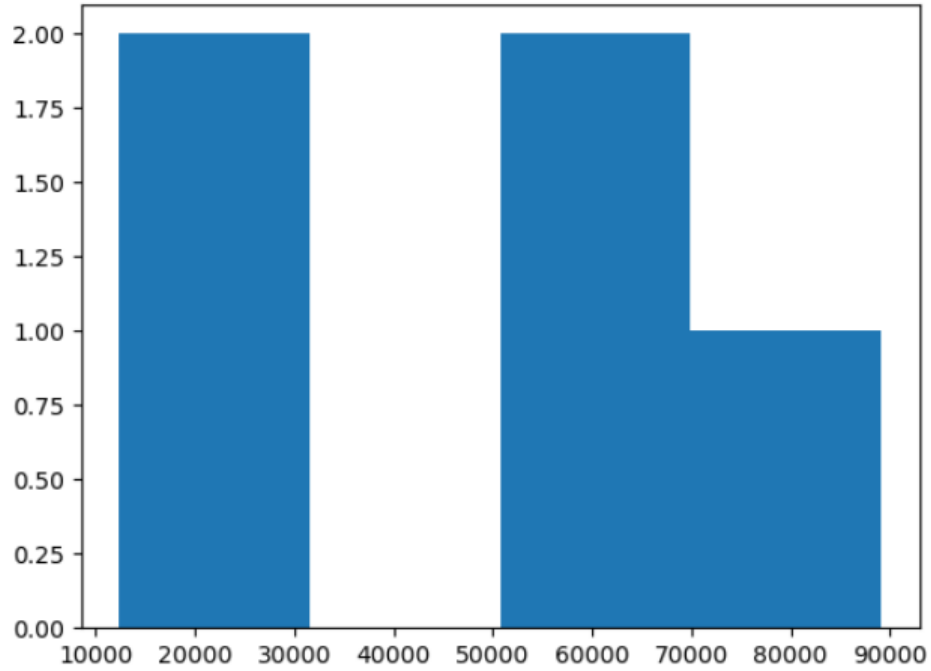


Histogram

```
In [13]: plt.hist(city_Pop, bins=1)
         plt.show()
```

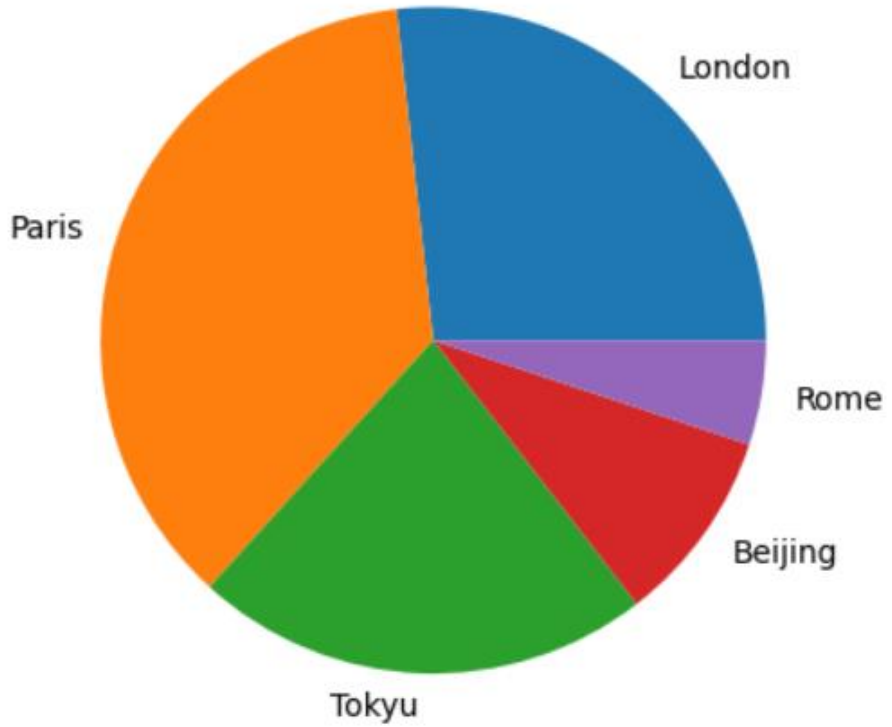



```
In [12]: plt.hist(city_Pop, bins='auto')
         plt.show()
```



Pie Chart

```
In [14]: plt.pie(city_Pop,labels=city_Name)
         plt.show()
```



Plot Customization

```
In [17]: plt.figure(figsize=(9,7), dpi=80)
```

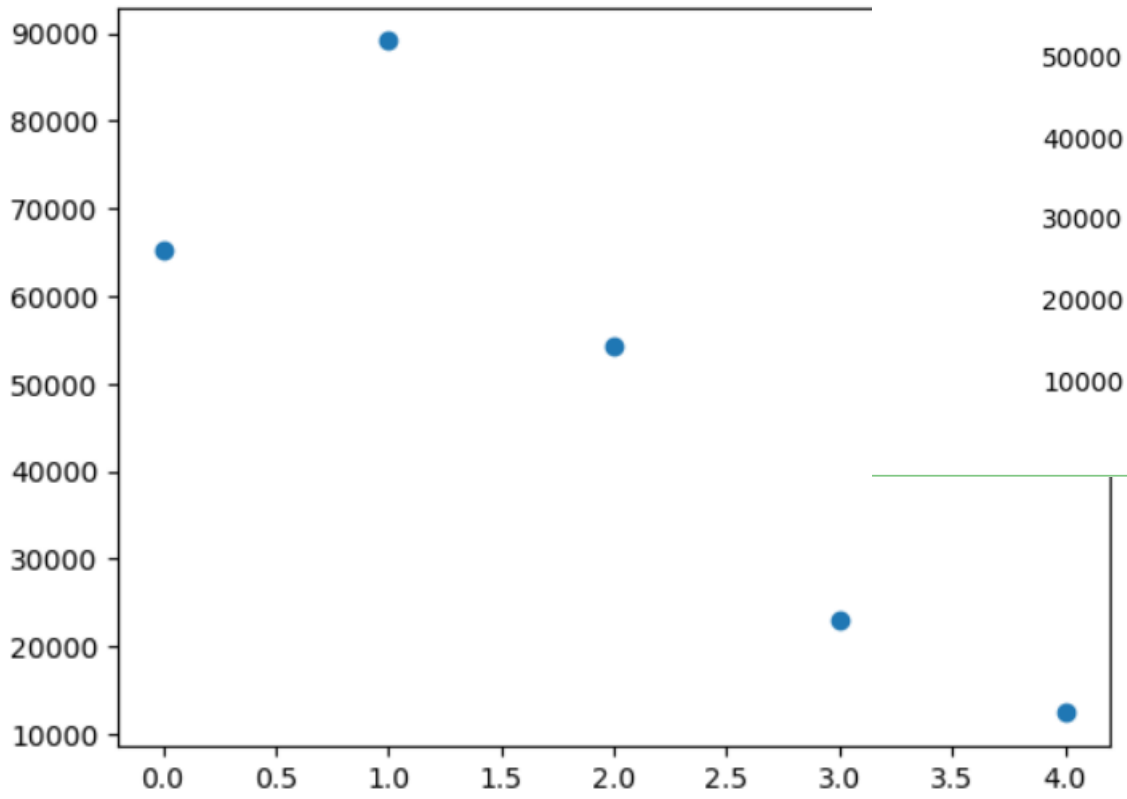
```
Out[17]: <Figure size 720x560 with 0 Axes>

         <Figure size 720x560 with 0 Axes>
```

```
In [18]: plt.plot(years,Population)
         plt.xlabel('Years')
         plt.ylabel('Popuation')
         plt.xticks([1960,1970,1980,1990,2000,2010])
         plt.yticks([21.91,23.90,24.80,20.93,22.30,26.90],['21M','23M','24M','20M','22M','26M'])
```

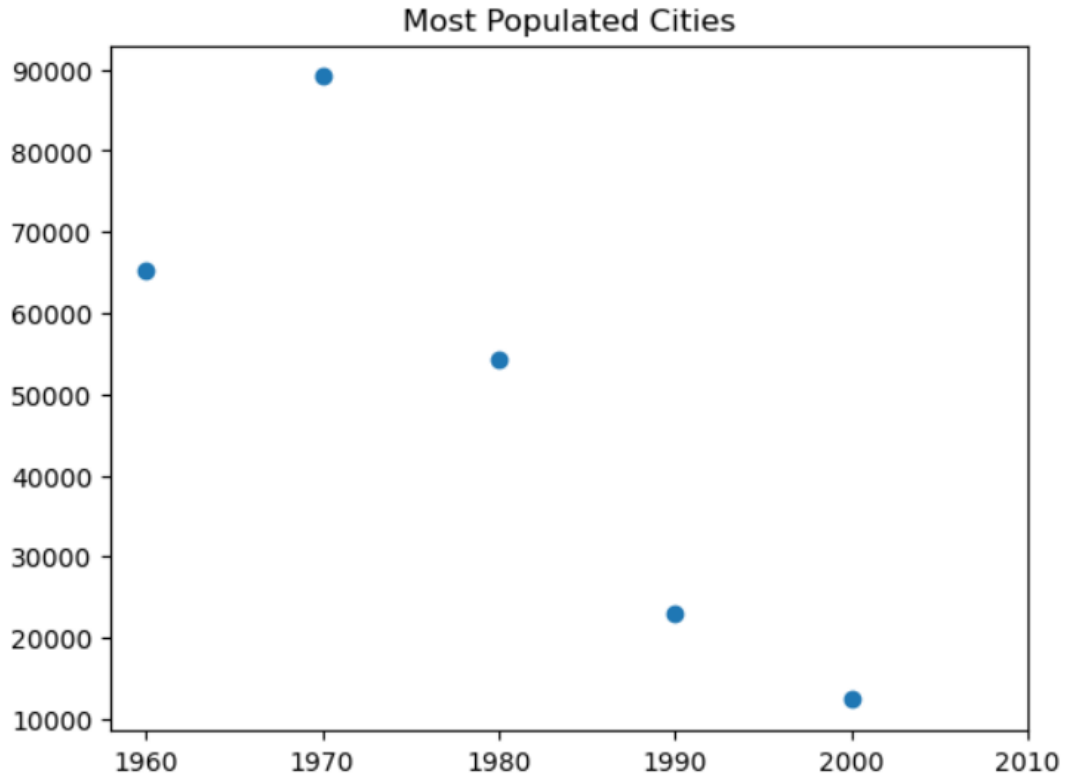
```
Out[18]: ([<matplotlib.axis.YTick at 0x253801c9520>,
           <matplotlib.axis.YTick at 0x253801a90d0>,
           <matplotlib.axis.YTick at 0x253801d1070>,
           <matplotlib.axis.YTick at 0x2538019e9a0>,
           <matplotlib.axis.YTick at 0x25380168490>,
           <matplotlib.axis.YTick at 0x2538014ad60>],
          [Text(0, 21.91, '21M'),
           Text(0, 23.9, '23M'),
           Text(0, 24.8, '24M'),
           Text(0, 20.93, '20M'),
           Text(0, 22.3, '22M'),
           Text(0, 26.9, '26M')])
```

Plot Customization

```
In [20]: plt.scatter(np.arange(5),city_Pop)
         plt.show()
```



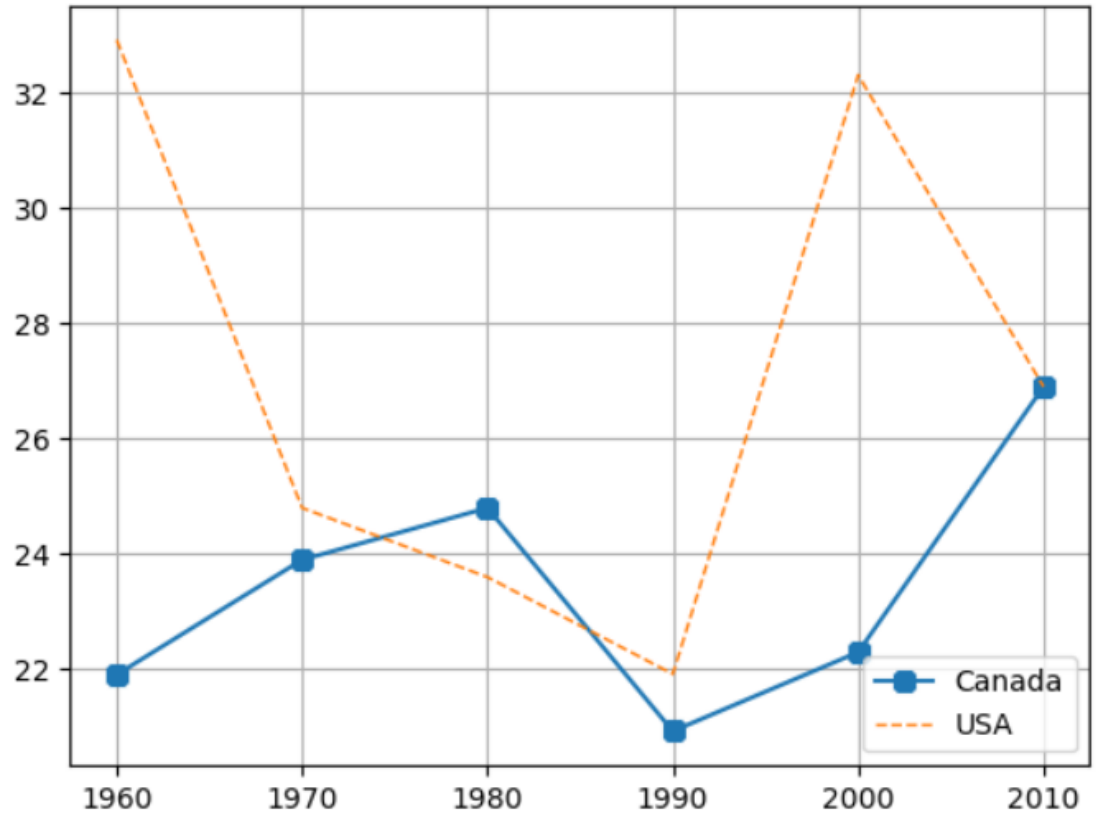

```
In [22]: plt.scatter(np.arange(5),city_Pop)
         plt.xticks([0,1,2,3,4,5],['1960','1970','1980','1990','2000','2010'])
         plt.title('Most Populated Cities')
         plt.show()
```



Multiple Plots

loc: Location
lw: Lineweight
ls: Line_style
mew: Marker size

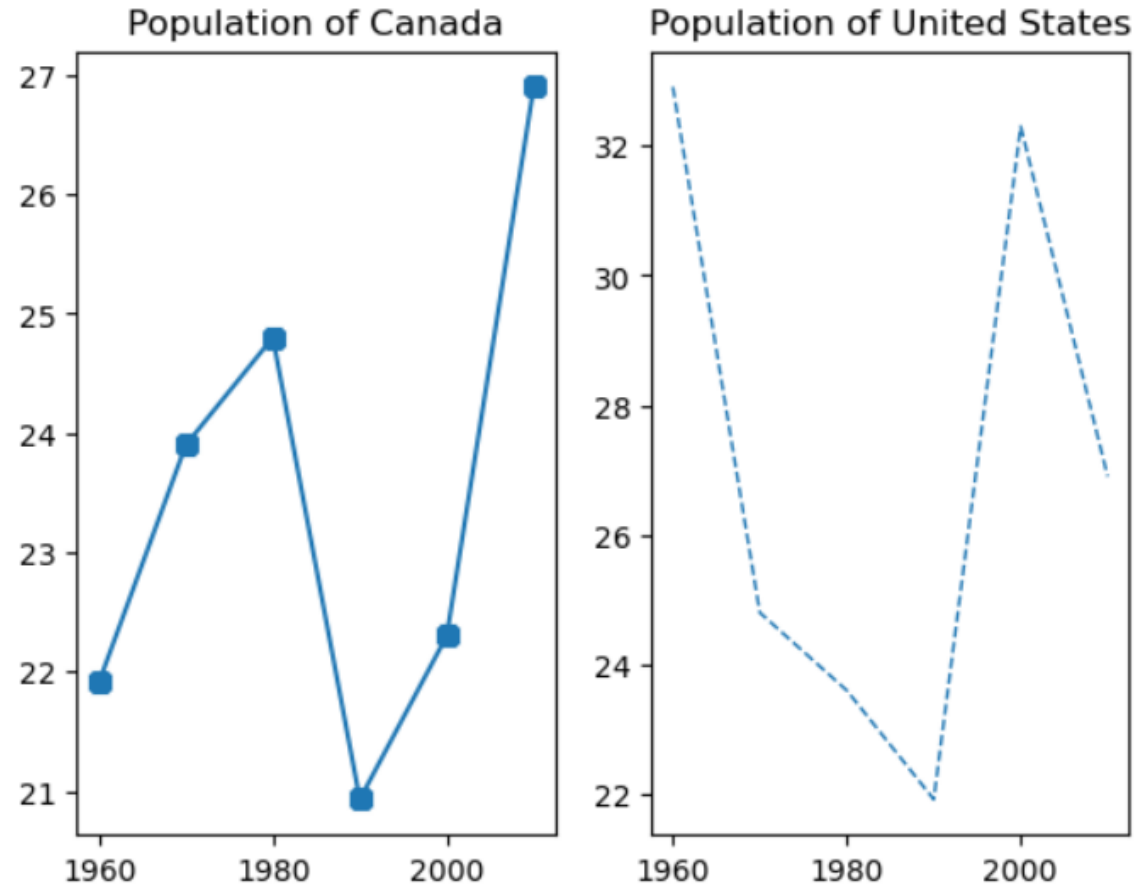
```
In [12]: years=[1960,1970,1980,1990,2000,2010]
         Canada_Pop=[21.91,23.90,24.80,20.93,22.30,26.90]
         USA_Pop=[32.91,24.80,23.60,21.92,32.30,26.90]
         plt.plot(years,Canada_Pop,marker='+',mew=8,ls='-')
         plt.plot(years,USA_Pop,ls='--',lw=1)
         plt.legend(['Canada','USA'],loc='best')
         plt.grid()
         plt.show()
```



Multiple Plots

plt.subplot(1,2,1): one row two columns the first plot
plt.subplot(1,2,2): one row two columns the second plot

```
In [13]: plt.subplot(1,2,1)
         plt.plot(years,Canada_Pop,marker='+',mew=8,ls='-')
         plt.title('Population of Canada')
         plt.subplot(1,2,2)
         plt.plot(years,USA_Pop,ls='--',lw=1)
         plt.title('Population of United States')
         plt.show()
```



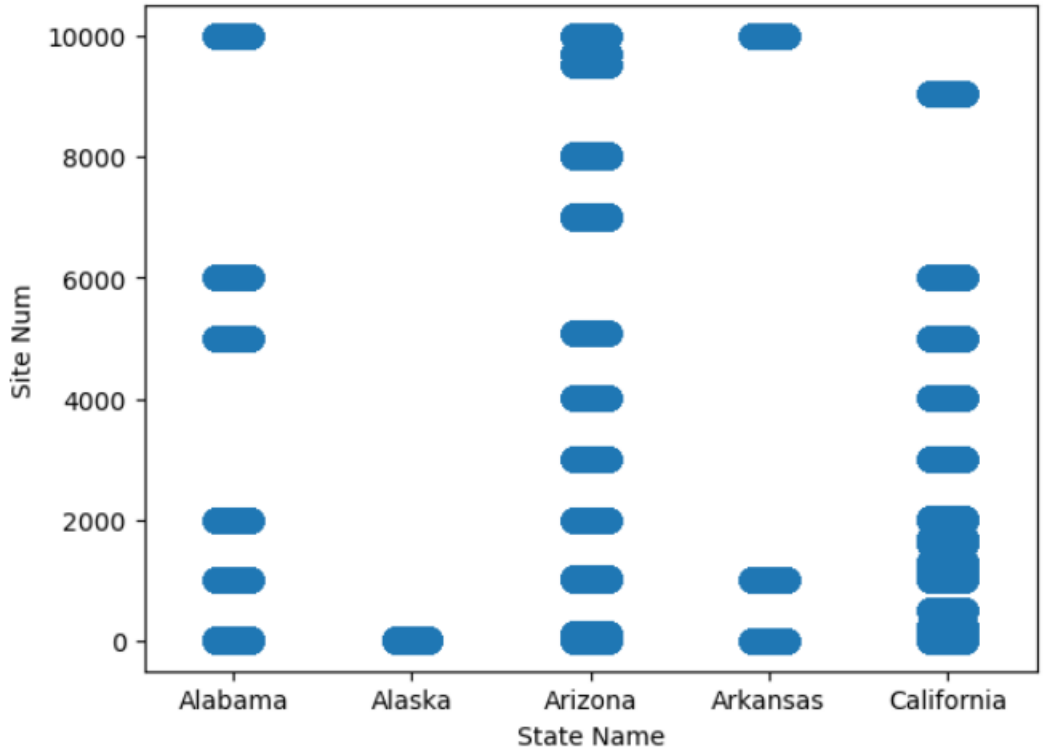
Strip Plot

In [17]:
```
import pandas as pd
import seaborn as sb
```

In [18]:
```
#Strip plot
data1=pd.read_csv('F:/00-Douglas College/1- Semester 1/3- Machine Learning in Data Science(3290)/Slides/ozone1.csv')
```

```
C:\Users\Paris\AppData\Local\Temp\ipykernel_22972\3359739865.py:2: DtypeWarning: Columns (17) have mixed types. Specify dtype o
ption on import or set low_memory=False.
  data1=pd.read_csv('F:/00-Douglas College/1- Semester 1/3- Machine Learning in Data Science(3290)/Slides/ozone1.csv')
```

In [19]:
```
sb.stripplot(x='State Name',y='Site Num',data=data1,size=10,jitter=True)
plt.show()
```



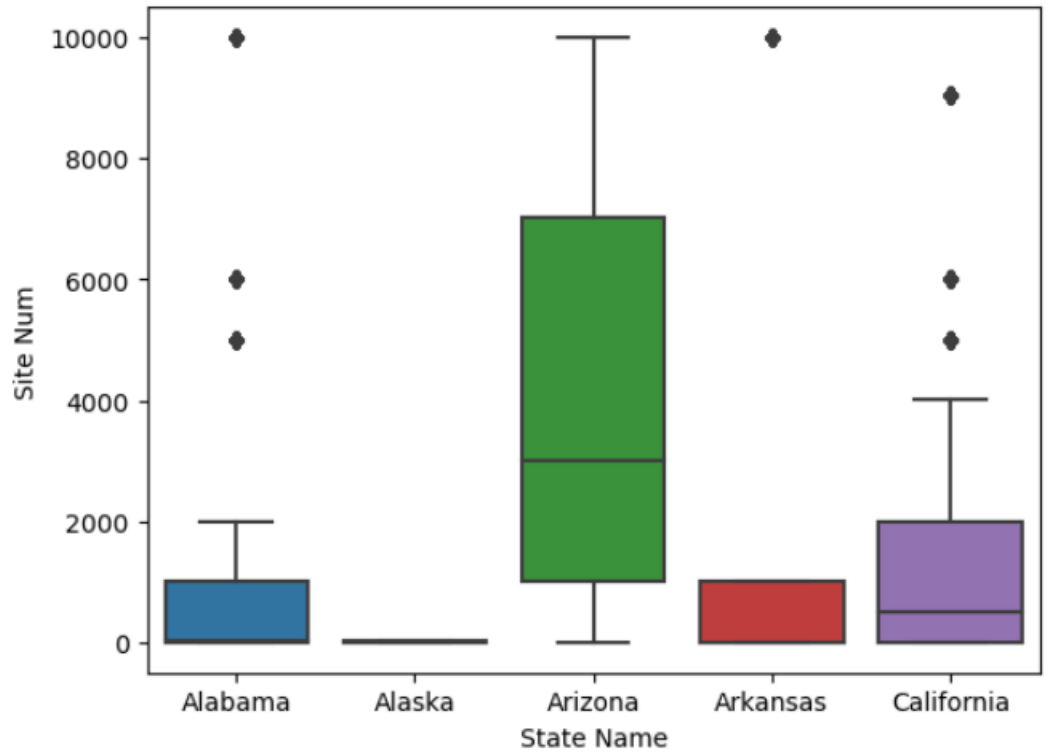
Box Plot

In [3]:
```
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
```

In [2]:
```
data1=pd.read_csv('F:/00-Douglas College/1- Semester 1/3- Machine Learning in Data Science(3290)/Slides/ozone1.csv')
```

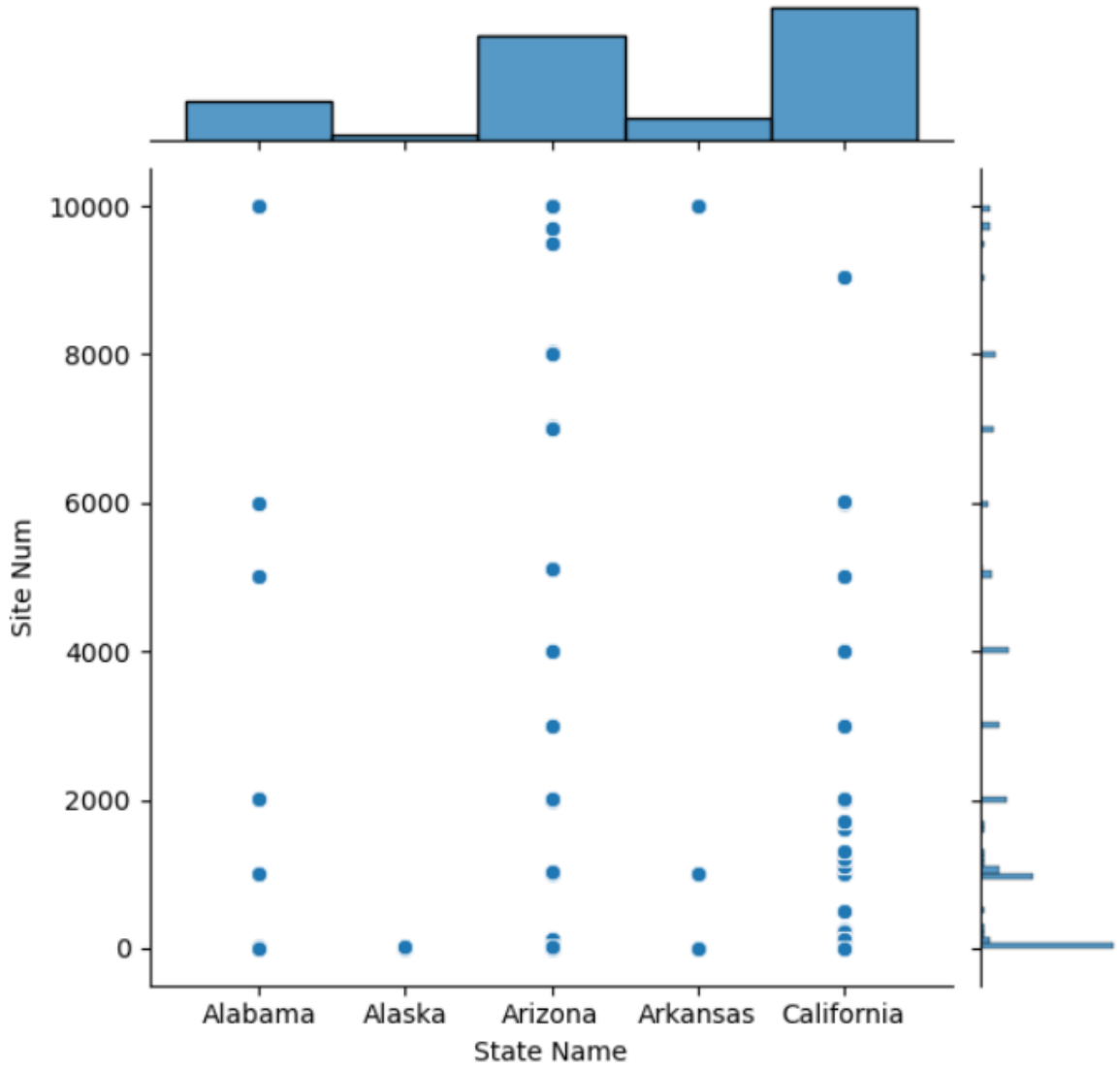
```
C:\Users\Paris\AppData\Local\Temp\ipykernel_22048\1179738982.py:1: DtypeWarning: Columns (17) have mixed types. Specify dtype o
ption on import or set low_memory=False.
  data1=pd.read_csv('F:/00-Douglas College/1- Semester 1/3- Machine Learning in Data Science(3290)/Slides/ozone1.csv')
```

In [4]:
```
sb.boxplot(x='State Name', y='Site Num', data=data1)
plt.show()
```



Joint Plot

```
In [4]: sb.jointplot(x='State Name', y='Site Num', data=data1, kind='scatter')
        plt.show()
```



More Plots in Seaborn

- **Swarm Plot**
- **Pair Plot**
- **…**