BEGINNER TASK

1. Overview of the Library  
Matplotlib is the core Python charting package that provides a great deal of control over plot components.  
  
Important attributes:  
  
Complete customization for a low-level API.  
  
produces output in a variety of formats (PNG, PDF, SVG).  
  
performs well in notebooks, GUIs, and scripts.  
  
Use cases include static publications, personalized dashboards, and scientific graphs.  
  
The Seaborn  
Description: It provides statistical plotting and streamlines intricate visuals by building upon Matplotlib.  
  
Important attributes:  
  
syntax that is cleaner.  
  
manages aesthetics automatically.  
  
fits quite nicely with Pandas.  
  
pre-existing color schemes and themes.  
  
Use cases include statistical data visualization and exploratory data analysis (EDA).

2. Graph Types   
A line plot illustrates patterns throughout time.  
  
The plt.plot() function in Matplotlib  
  
sns.lineplot(x=x, y=y, data=df) in Seaborn  
  
A scatter plot shows how variables relate to one another.  
  
Plot.scatter(x, y) in Matplotlib  
  
sns.scatterplot(x=x, y=y, data=df) in Seaborn  
  
Categories are compared using a bar chart.  
  
Matplotlib: plt.bar(values, categories)  
  
The sns.barplot function in Seaborn (x=cat, y=val, data=df)  
  
Data distribution is displayed using a histogram.  
  
Plot.hist(data) in Matplotlib  
  
sns.histplot(data, kde=True) in Seaborn  
  
Pie Chart: For proportions, only in Matplotlib.  
  
plt.pie(sizes, labels=labels) in Matplotlib

1. Comparison

In contrast   
Although it takes more code, Matplotlib provides greater control and flexibility. For rapid, visually appealing statistical charts, Seaborn is easier to use and more effective. While Seaborn focuses on lucid, visually appealing visualizations, it is less configurable without Matplotlib. Matplotlib also supports pie charts and is superior at handling massive datasets. Neither has native support for interactivity.