Name: Akash Patel
Roll no: 281029
PRN : 22310745
Batch: A2

Assignment 6

Statement:

In this assignment we have to:
a) Perform data visualization using Seaborn library.
b) Create various types of plots for statistical data visualization.
c) Customize plots using Seaborn's in-built features.
d) Analyze the relationship between variables using visual representation.
e) Explore advanced plotting techniques for better data insights.

Objective:

1. To understand data visualization using Seaborn library in Python.
2. To create different types of plots for statistical and relational data visualization.
3. To learn about customization and styling options available in Seaborn.
4. To improve data analysis using attractive and informative plots.

Resources Used:

Software Used: Jupyter Notebook
Libraries Used: Pandas, NumPy, Matplotlib, Seaborn

Introduction to Seaborn:

Seaborn is a powerful Python data visualization library based on Matplotlib.
It provides a high-level interface for creating attractive and informative statistical graphics.
Key Features of Seaborn:

- Built-in themes for better visualization.
- Automatic estimation and plotting of linear regression models.
- Supports complex visualizations like heatmaps, pair plots, violin plots, etc.
- Better default aesthetics than Matplotlib.

Some Basic Functions Used in Program:

Function	**Purpose**
sns.lineplot()	To create line plots
sns.barplot()	To create bar graphs
sns.histplot()	To create histograms
sns.scatterplot()	To create scatter plots
sns.boxplot()	To create box plots
sns.heatmap()	To create heatmaps
sns.pairplot()	To plot pairwise relationships
sns.countplot()	To count occurrences of categories
sns.distplot()	To plot distribution of data
plt.show()	To display the plot

Methodology:

1. Imported necessary libraries like Pandas, NumPy, Matplotlib, and Seaborn.
2. Loaded the dataset using pd.read_csv() function.
3. Created line plots using sns.lineplot() to show trends.
4. Created bar graphs using sns.barplot() to compare categories.
5. Created histograms using sns.histplot() for frequency distribution.
6. Created scatter plots using sns.scatterplot() to show relationship between variables.
7. Used sns.boxplot() to visualize distribution and detect outliers.
8. Used sns.heatmap() for correlation matrix visualization.
9. Used sns.pairplot() for pairwise data visualization.
10. Customized plots with titles, labels, colors, and themes.
11. Displayed all graphs using plt.show() function.

Advantages:

- Provides attractive and informative statistical plots.
- Built-in themes for improved aesthetics.
- Easily integrates with Pandas DataFrames.
- Simplifies complex visualizations.
- Good for exploring and understanding data patterns.

Disadvantages:

- Less control over individual plot elements compared to Matplotlib.
- May consume more memory for large datasets.

- Dependent on Matplotlib backend for rendering.

Conclusion:

In this assignment, I learned how to use Seaborn for data visualization in Python. I created various types of plots like line plots, bar graphs, histograms, scatter plots, box plots, heatmaps, and pair plots. Seaborn provides better aesthetics and makes complex visualizations easier to create. This assignment helped me to explore the relationship between variables visually and gain better insights from data.