

4. Write a Python Program which explains uses of customizing seaborn plots with Aesthetic functions.

1. Introduction.

- This program utilizes the seaborn and matplotlib libraries in Python to create a visually appealing bar plot representing the popularity of different programming languages among students. It customizes the aesthetics for better visual appeal, including a clean background, labeled axes, and annotations on top of the bars.
- Seaborn is a data visualization library based on Matplotlib that provides a high-level interface for creating informative and attractive statistical graphics. Seaborn allows users to easily plot and to customize the appearance and elements of the plot, such as the style, theme, color, palette, labels, ticks, and data labels. Seaborn also integrates well with pandas and matplotlib, which are the other popular libraries for data manipulation and visualization in Python.
- In this program, seaborn is employed to customize the aesthetics of the bar plot, making it more visually appealing.

2. Program code.

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

# Create a sample dataset
data = {
    'Programming Language': ['Python', 'Java', 'C++', 'JavaScript',
                             'R'], 'Number of Users': [55, 40, 35, 25, 15]
}
df = pd.DataFrame(data)

# Set up a more visually appealing plot style
sns.set_theme(style="whitegrid")
```

```
# Create a bar plot with improved aesthetics

plt.figure(figsize=(10, 6))
ax = sns.barplot(x="Programming Language", y="Number of Users", data=df,
palette="deep")
plt.xticks(rotation=0, ha="left", fontsize=12)
plt.yticks(fontsize=12)
plt.title("Programming Language Popularity Among Students",
fontsize=16,color='blue')
plt.xlabel("Programming Language", fontsize=16)
plt.ylabel("Number of Users", fontsize=14)
plt.show()
```

3. Explanation of the code

1. import seaborn as sns:

This line imports the seaborn library and assigns it the alias sns. Seaborn is a statistical data visualization library based on Matplotlib. It provides a high-level interface for creating informative and attractive statistical graphics.

2. import matplotlib.pyplot as plt:

This line imports the pyplot module from the matplotlib library and assigns it the alias plt. matplotlib.pyplot is a collection of functions that make Matplotlib work like MATLAB. It is used for creating various types of plots and visualizations.

3. import pandas as pd:

This line imports the pandas library and assigns it the alias pd. Pandas is a powerful data manipulation and analysis library for Python. It provides data structures like DataFrame, which is particularly useful for handling and analyzing structured data.

4. # Create a sample dataset

```
data = {  
    'Programming Language': ['Python', 'Java', 'C++',  
    'JavaScript', 'R'],  
    'Number of Users': [55, 40, 35, 25, 15]  
}
```

- Here, a Python dictionary named data is defined. It has two key-value pairs. The keys are 'Programming Language' and 'Number of Users'. The values associated with these keys are lists, where the first list corresponds to different programming languages, and the second list contains the number of users for each language.

5. df = pd.DataFrame(data)

- The pd.DataFrame(data) creates a pandas DataFrame from the dictionary data. In this case, it creates a two-column DataFrame where the columns are labeled 'Programming Language' and 'Number of Users', and the data is filled with the corresponding values from the dictionary.
- The resulting DataFrame, 'df,' looks like:

	Programming Language	Number of Users
0	Python	55
1	Java	40
2	C++	35
3	JavaScript	25
4	R	15

This DataFrame serves as the basis for creating visualizations in the subsequent code.

`sns.set_theme(style="whitegrid")`

- The `sns.set_theme()` function is used to set the default seaborn theme for the plots. In this case, the theme is set to "whitegrid". Seaborn provides several built-in themes that affect the overall appearance of the plots. The "whitegrid" theme, as the name suggests, creates a plot with a white background and adds gridlines for better readability. This choice is often preferred for clean and visually appealing visualizations.

`plt.figure(figsize=(10, 6))`

- The `plt.figure(figsize=(10, 6))` line creates a new Matplotlib figure with a specified size. The `figsize` parameter is set to (10, 6), meaning the width of the figure is 10 inches, and the height is 6 inches. This helps control the dimensions of the upcoming bar plot. Specifying the figure size is crucial for ensuring that the plot is visually balanced and has the desired aspect ratio.

`ax=sns.barplot(x="Programming Language", y="Number of Users", data=df, palette="deep")`

- This line of code creates a bar plot using seaborn, visualizing the number of users for each programming language based on the provided DataFrame (df). The aesthetics of the plot, including the color palette, are customized for better visual appeal.
- **ax:** This variable is assigned the result of the `sns.barplot()` function. It stands for "axes" and represents the set of axes on which the bar plot is drawn. The `ax` variable can be used to further customize or annotate the plot.
- **sns.barplot():** This function is part of the seaborn library and is used to create a bar plot. It takes several parameters:
- **x:** Specifies the variable to be plotted on the x-axis. In this case, it's "Programming Language," which is a column in the DataFrame df.
- **y:** Specifies the variable to be plotted on the y-axis. Here, it's "Number of Users," another column in the DataFrame df.
- **data:** The DataFrame containing the actual data to be plotted. In this case, it's the DataFrame df created earlier.
- **palette:** This parameter sets the color palette for the bars. In this case, "deep" is chosen as the palette, which provides a set of distinct, deep colors for the bars.

plt.xticks(rotation=0, ha='left', fontsize=12)

- plt.xticks(): This function is used to customize the properties of the x-axis ticks. In this case:
- rotation=0: Sets the rotation angle of the x-axis tick labels to 0 degrees, meaning the labels will be horizontal.
- ha="left": Aligns the x-axis tick labels to the left.
- fontsize=12: Sets the font size of the x-axis tick labels to 12 points.

plt.yticks(fontsize=12)

- plt.yticks(): Similar to the plt.xticks() function, this line customizes the properties of the y-axis ticks. It sets the font size of the y-axis tick labels to 12 points.

plt.title("Programming Language Popularity Among Students", fontsize=16, color='blue')

- plt.xlabel(): Adds a label to the x-axis. The parameter is:
- The x-axis label text: "Programming Language."
- fontsize=16: Sets the font size of the x-axis label to 16 points.

plt.ylabel("Number of Users", fontsize=14)

- plt.ylabel(): Adds a label to the y-axis. The parameter is:
- The y-axis label text: "Number of Users."
- fontsize=14: Sets the font size of the y-axis label to 14 points.

4. Output

- The output is a bar plot that shows the popularity of five programming languages among students, based on the number of users. Python is the most popular, followed by Java, C++, JavaScript, and R.
- The output of the program demonstrates the use and benefits of seaborn for creating attractive and informative bar plots with customized aesthetics.

.....

Programming Language Popularity Among Students

