

**Department of Engineering Sciences and  
Technology,  
Second Year Btech in Computer Science  
Project Based Learning-Python  
Assignment - 19**

Name - Paritosh kolwadkar

SRN – 31231313

Roll no – 39

Batch – D2

Problem statement : **Write a program to create a box and whiskers plot and an area plot using Matplotlib. Include multiple subplots in a single figure to display different plots side by side.**

**Prerequisite:**

Install **Matplotlib** if not already installed:

```
pip install matplotlib
```

Understanding of box plots, area plots, and subplots in Matplotlib.

Code:

```
import matplotlib.pyplot as plt
import numpy as np

# Data for the plots
data_box = np.random.randn(100) # Random data for the box plot
```

```

x_area = np.linspace(0, 10, 100) # X-axis data for the area plot
y_area = np.sin(x_area) # Y-axis data for the area plot

# Create subplots
fig, axes = plt.subplots(1, 2, figsize=(14, 6)) # 1 row, 2 columns

# 1. Box Plot (Box and Whiskers Plot)
axes[0].boxplot(data_box, vert=True, patch_artist=True, notch=True,
widths=0.5,
                boxprops=dict(facecolor='skyblue', color='blue'),
                whiskerprops=dict(color='blue'),
                flierprops=dict(marker='o', markerfacecolor='red',
markersize=8, markeredgewidth=2))
axes[0].set_title('Box and Whiskers Plot')
axes[0].set_ylabel('Values')
axes[0].grid(True)

# 2. Area Plot
axes[1].fill_between(x_area, y_area, color='skyblue', alpha=0.5) # Create
the area plot
axes[1].plot(x_area, y_area, color='blue', label='Sine Wave', linewidth=2)
# Plot the line over the area
axes[1].set_title('Area Plot')
axes[1].set_xlabel('X-axis')
axes[1].set_ylabel('Y-axis')
axes[1].legend()
axes[1].grid(True)

# Adjust layout to prevent overlapping
plt.tight_layout()

```

```
# Display the plots  
plt.show()
```

Explanation :

1. Box Plot (Box and Whiskers Plot):

- Data: `data_box` is generated using `np.random.randn(100)` to create 100 random data points following a standard normal distribution.
- Plot: The `boxplot()` function is used to create the box plot. It includes:
  - Notch: Provides a visual representation of the confidence interval around the median.
  - Whiskers: Extend to the minimum and maximum values within a set range (often 1.5 times the interquartile range).
  - Outliers (Fliers): Represented by red dots for values outside the whiskers.
  - Customization: Colors are added to the box, whiskers, and fliers for better visualization.

2. Area Plot:

- Data: `x_area` is created using `np.linspace(0, 10, 100)` to generate 100 points between 0 and 10. `y_area` is calculated as the sine of `x_area` using `np.sin(x_area)`.
- Plot: The `fill_between()` function is used to fill the area under the sine wave.
  - The sine wave itself is plotted over the filled area with a blue line using `plot()`.
  - The plot is customized with grid lines and labeled axes.

3. Subplots:

- `fig, axes = plt.subplots(1, 2, figsize=(14, 6))`: Creates a figure with 1 row and 2 columns to display the plots side by side.
- The `axes[0]` and `axes[1]` are used to access the individual plots in the subplots.

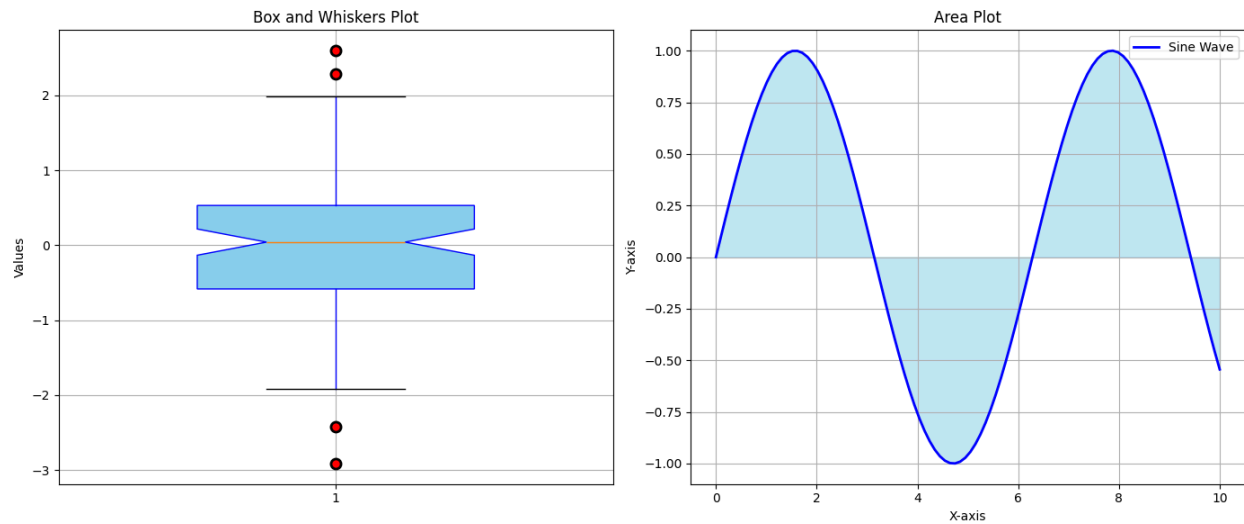
4. Layout:

- `plt.tight_layout()`: Adjusts the subplots to ensure that the labels and titles do not overlap.

5. Display:

- `plt.show()`: Displays the figure with both the box plot and area plot.

## Output:



## Output Explained:

- **Box Plot:** Custom colors are used for the box (`skyblue`), whiskers, and outliers (red).
- **Area Plot:** The sine wave is filled with a semi-transparent blue (`alpha=0.5`), and the line is plotted in blue for better contrast.

By using subplots, both plots are displayed side by side in a single figure, offering a compact and organized view of both visualizations.