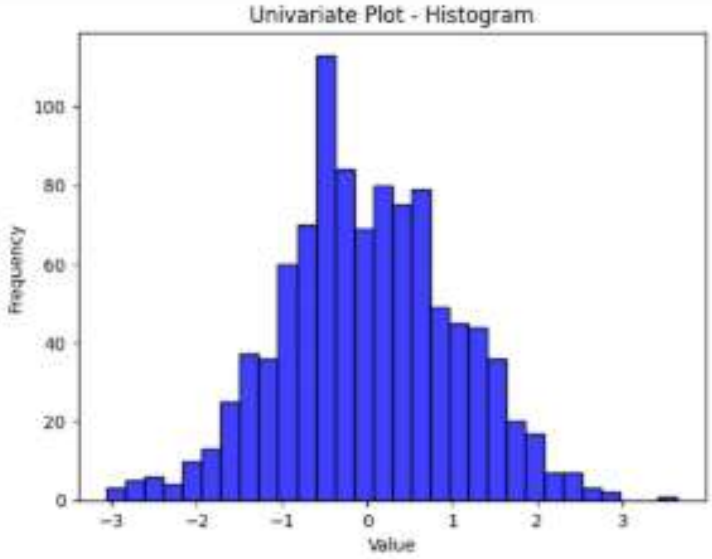
Seaborn:
Seaborn is a powerful and user-friendly Python library built on top of Matplotlib. It simplifies the process of creating attractive and informative statistical graphics, making it an excellent choice for data visualization.

Key Features:
1. Ease of use: Simple syntax to create complex visualizations.
2. Built-in themes: Customizable themes for styling plots.
3. Statistical plotting: Built-in methods for regression, box plots, violin plots, etc.
4. Integration: Works seamlessly with pandas DataFrames and NumPy arrays.

Univariate Plots
1. Histogram

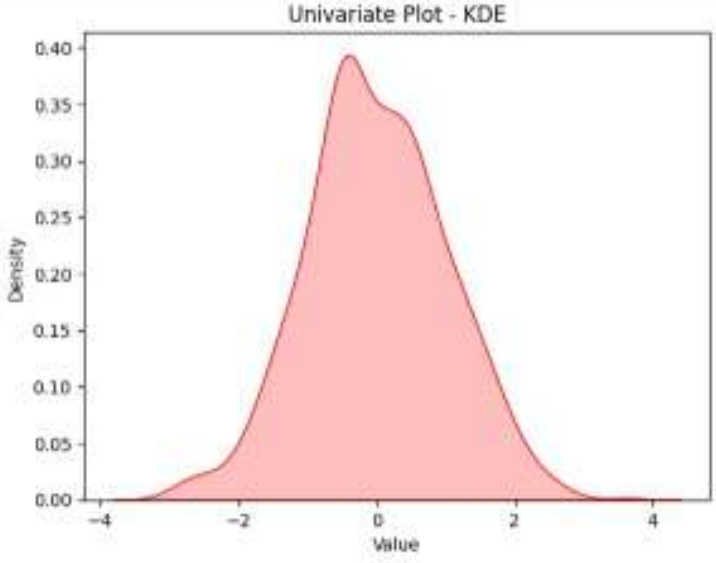
[1]:
```
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
# Generate random data
data = np.random.randn(1000)
# Create a histogram
sns.histplot(data, bins=30, color="blue")
plt.title("Univariate Plot - Histogram")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```



2. KDE Plot (Kernel Density Estimate)

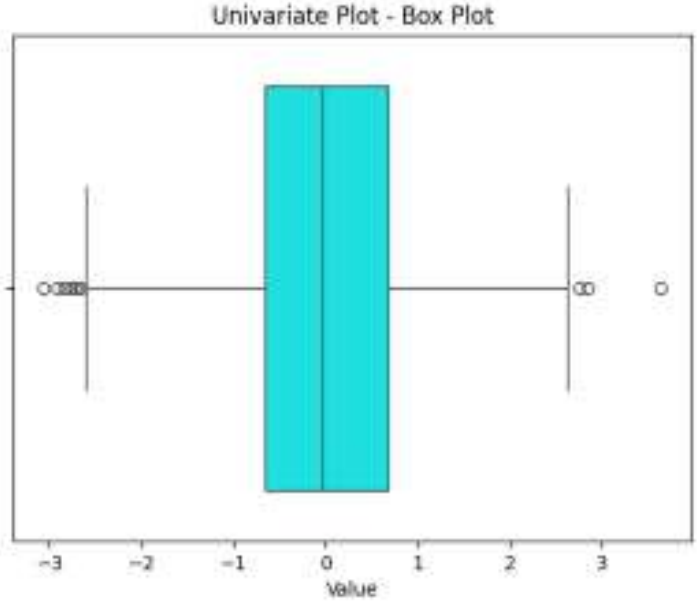
[2]:
```
sns.kdeplot(data, shade=True, color="red")
plt.title("Univariate Plot - KDE")
plt.xlabel("Value")
plt.ylabel("Density")
plt.show()
```

```
C:\Users\jjsrl\AppData\Local\Temp\ipykernel_5356\1762614588.py:1: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(data, shade=True, color="red")
```



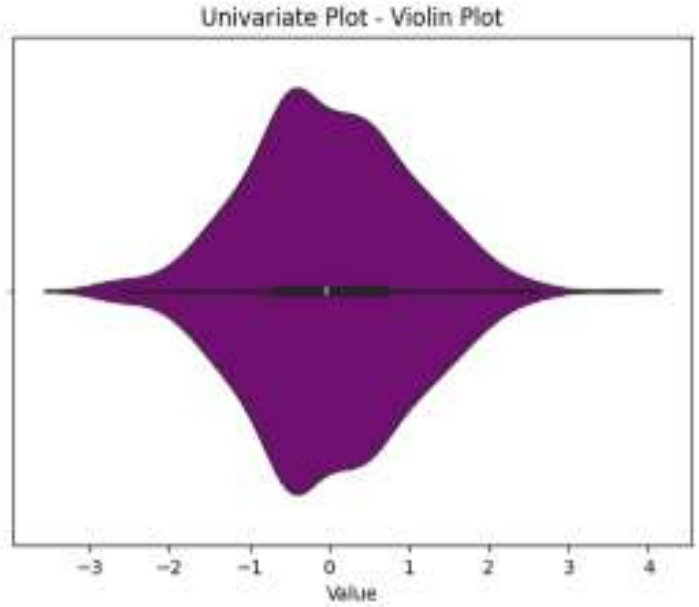
3. Box Plot

[5]
```
sns.boxplot(x=data, color="cyan")
plt.title("Univariate Plot - Box Plot")
plt.xlabel("Value")
plt.show()
```



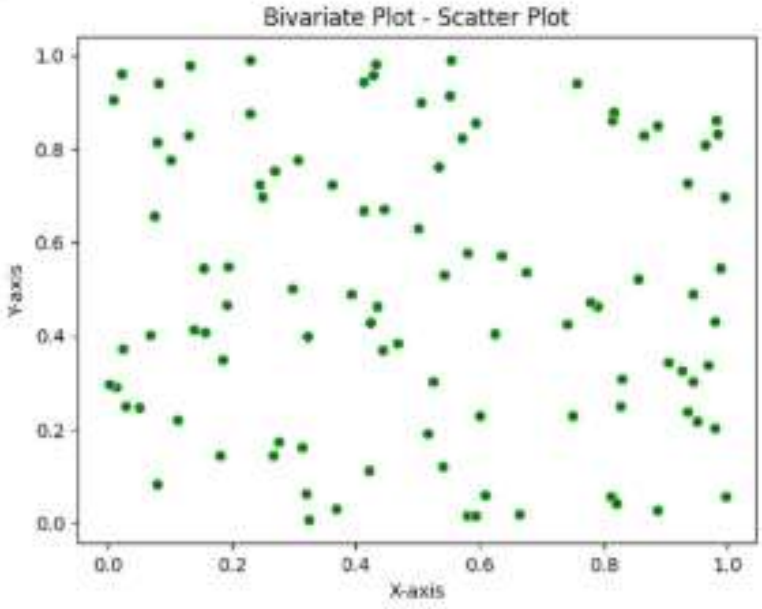
4. Violin Plot

[6]
```
sns.violinplot(x=data, color="purple")
plt.title("Univariate Plot - Violin Plot")
plt.xlabel("Value")
plt.show()
```



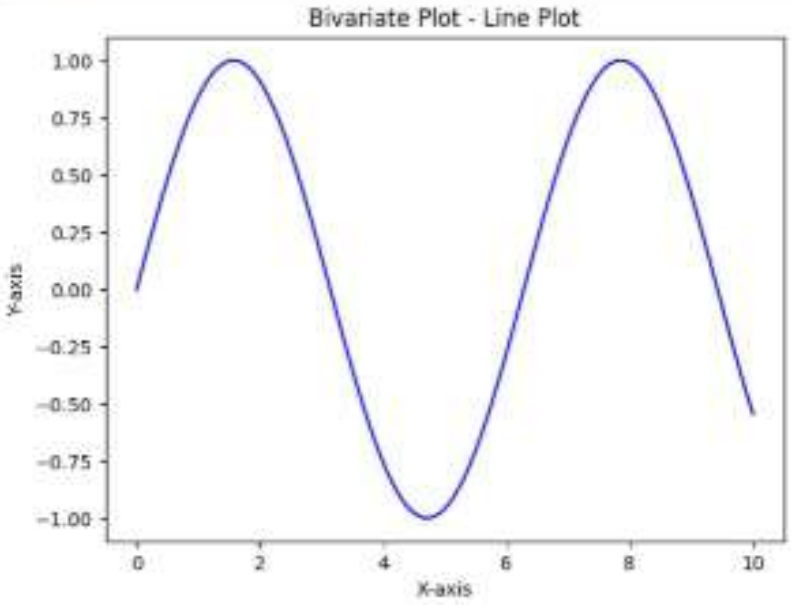
```
Bivariate Plots
1. Scatter Plot
```

[7]:
```
x = np.random.rand(100)
y = np.random.rand(100)
sns.scatterplot(x=x, y=y, color="green")
plt.title("Bivariate Plot - Scatter Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



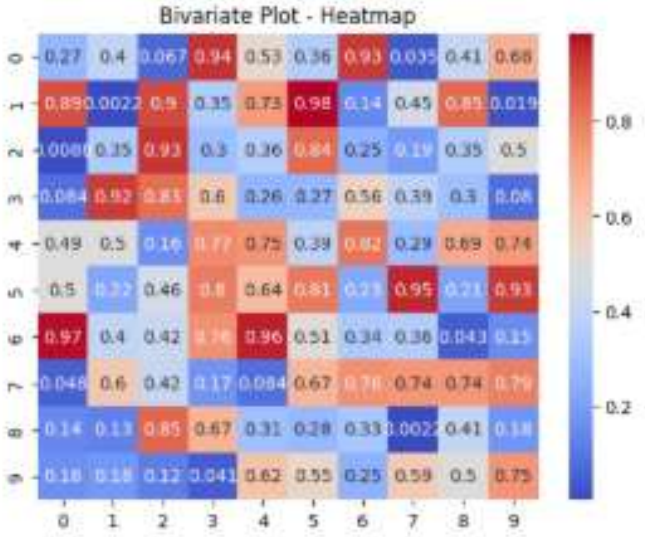

```
2. Line Plot
```

[8]:
```
x = np.linspace(0, 10, 100)
y = np.sin(x)
sns.lineplot(x=x, y=y, color="blue")
plt.title("Bivariate Plot - Line Plot")
plt.xlabel("X-axis")
plt.ylabel("Y-axis")
plt.show()
```



3. Heatmap

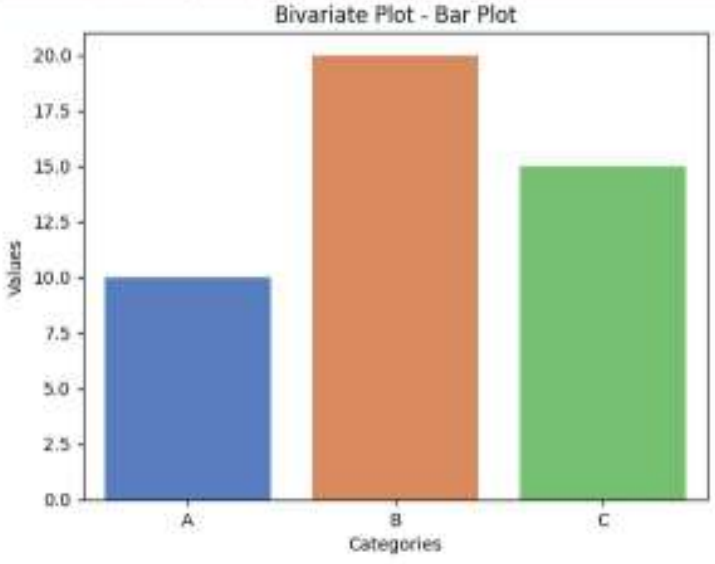
[10]:
```
data = np.random.rand(10, 10)
sns.heatmap(data, annot=True, cmap="coolwarm")
plt.title("Bivariate Plot - Heatmap")
plt.show()
```



4. Bar Plot

[11]:
```
categories = ["A", "B", "C"]
values = [10, 20, 15]
sns.barplot(x=categories, y=values, palette="muted")
plt.title("Bivariate Plot - Bar Plot")
plt.xlabel("Categories")
plt.ylabel("Values")
plt.show()
```

```
C:\Users\Jerl\AppData\Local\Temp\ipykernel_5396\1698695.py:3: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  sns.barplot(x=categories, y=values, palette="muted")
```



5. Pair Plot:(Special Bivariate Plot)

[12]
```
import pandas as pd
data = pd.DataFrame({
"X": np.random.rand(50),
"Y": np.random.rand(50),
"Z": np.random.rand(50)
})
sns.pairplot(data)
plt.suptitle("Bivariate Plot - Pair Plot", y=1.02)
plt.show()
```