



# Lecture Notes: Basic Visualization & Statistical Analysis in Python

```
### 🛠️ Setup Section
# This section loads required libraries and sets up synthetic data for use in visualizati
# Numpy, pandas, matplotlib, seaborn, and scipy are essential tools for data analysis in

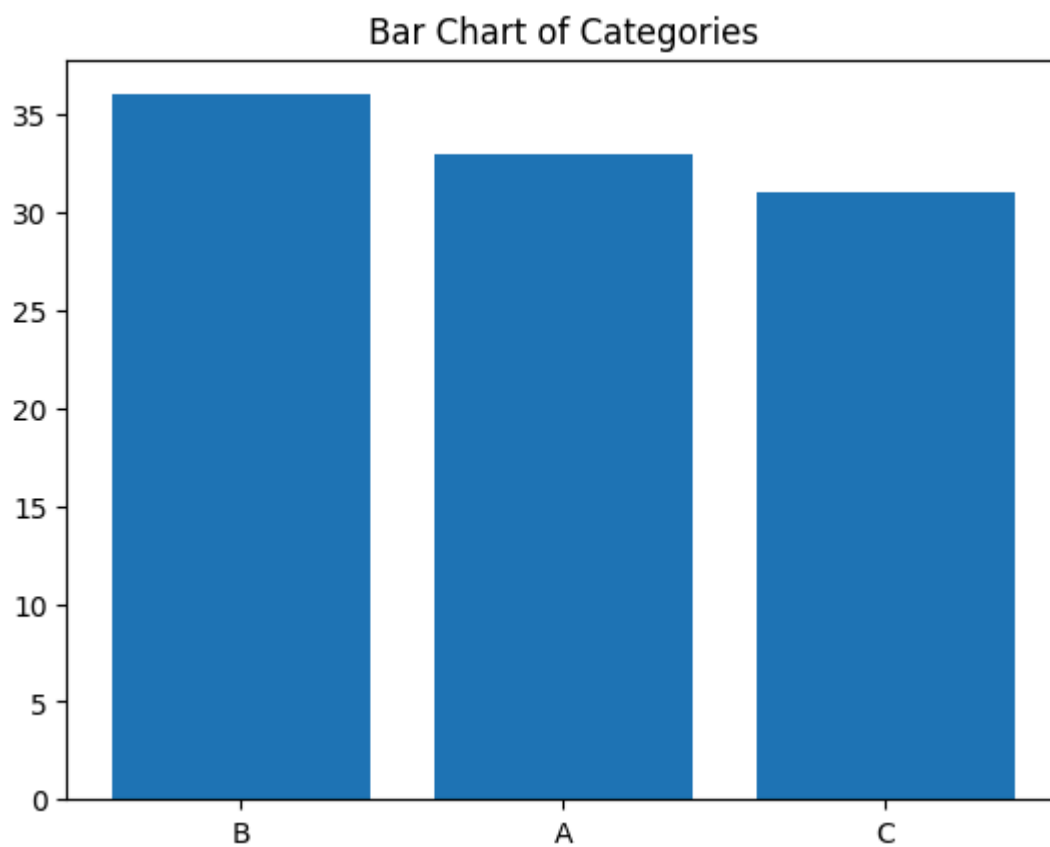
# --- Setup
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy import stats

%matplotlib inline
```

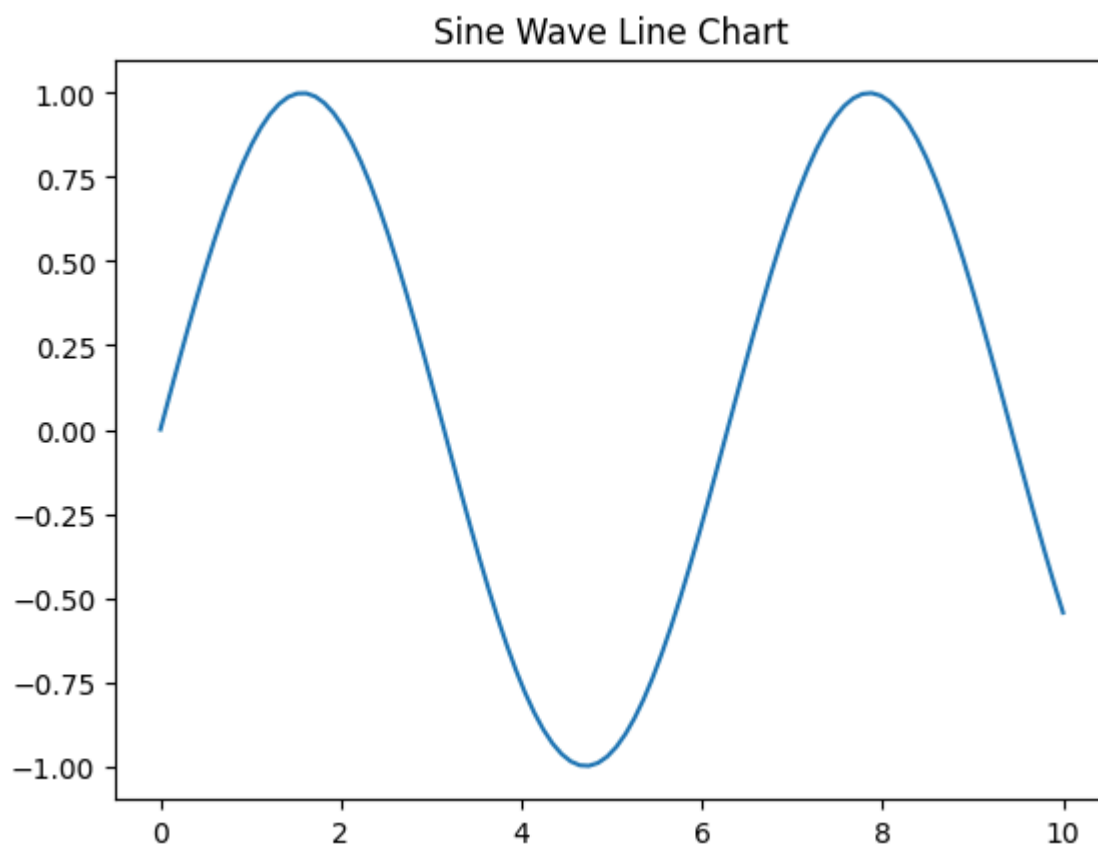
```
# Generate synthetic data for demo
np.random.seed(42)
data = pd.DataFrame({
    'Category': np.random.choice(['A', 'B', 'C'], 100),
    'Value1': np.random.normal(loc=5, scale=2, size=100),
    'Value2': np.random.normal(loc=10, scale=3, size=100),
    'Value3': np.random.randint(1, 100, 100)
})
```

```
### 📊 Bar Chart
# Used to compare frequencies or amounts across different categories.
# Here, we count the occurrences of each category in the 'Category' column.

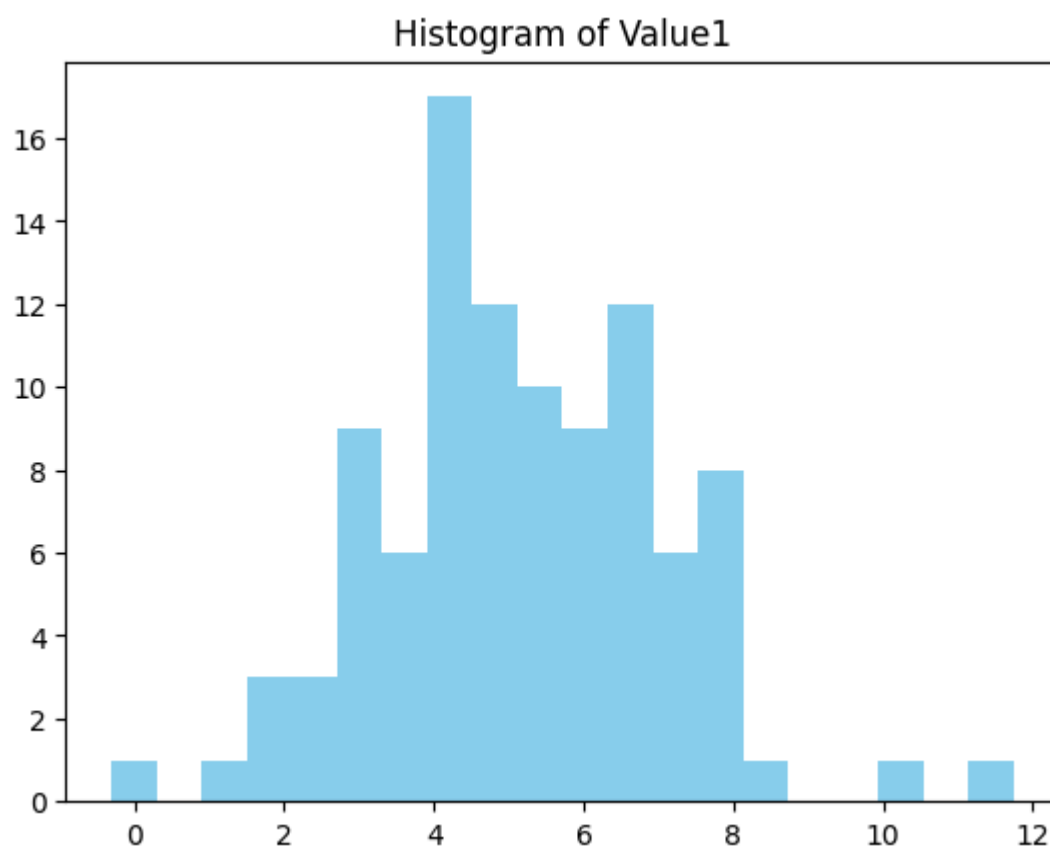
# --- 1. Bar Chart
bar_counts = data['Category'].value_counts()
plt.bar(bar_counts.index, bar_counts.values)
plt.title('Bar Chart of Categories')
plt.show()
```



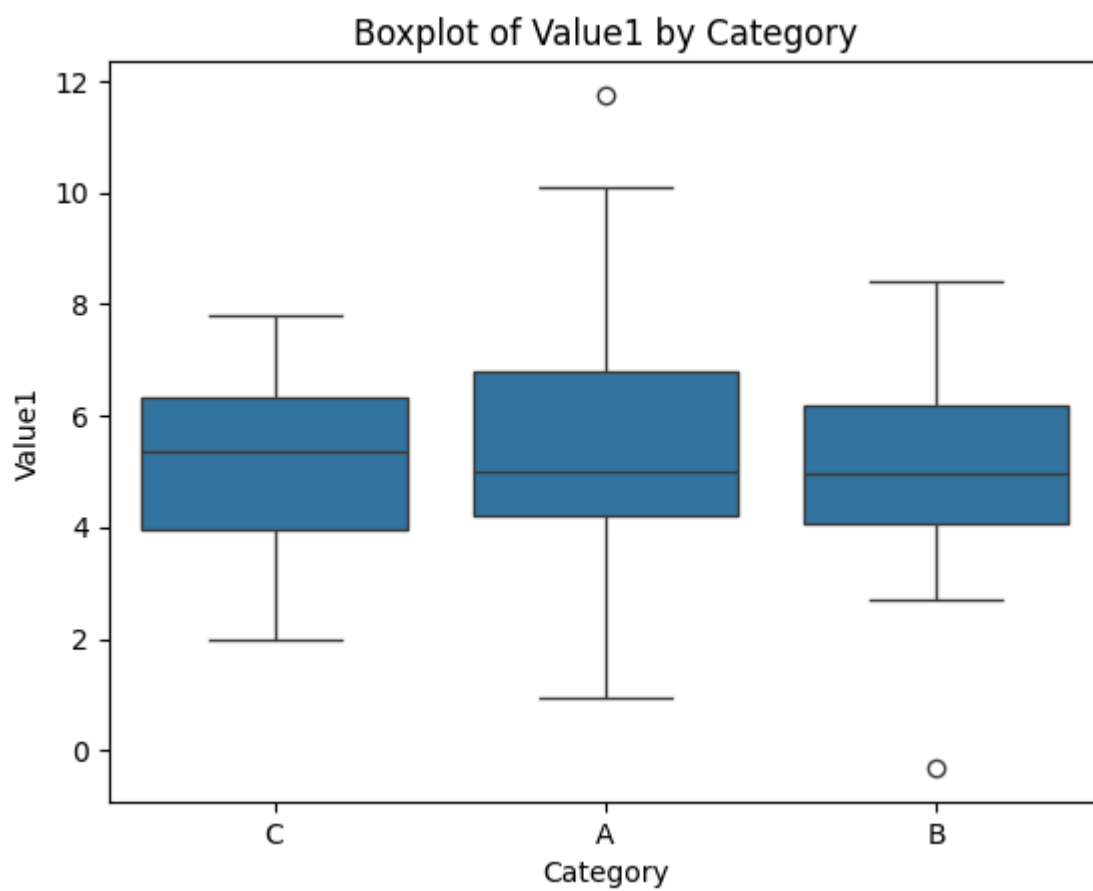
```
# --- 2. Line Chart
x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
plt.title('Sine Wave Line Chart')
plt.show()
```



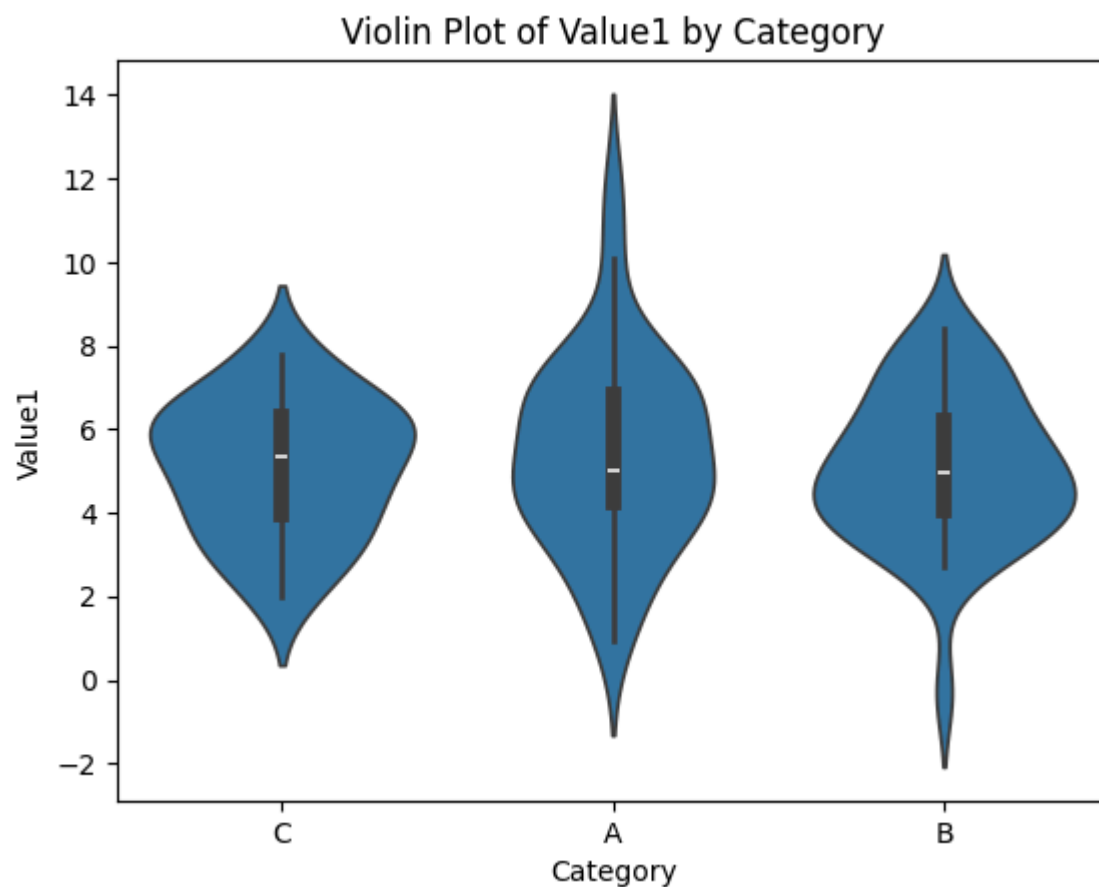
```
# --- 3. Histogram  
plt.hist(data['Value1'], bins=20, color='skyblue')  
plt.title('Histogram of Value1')  
plt.show()
```



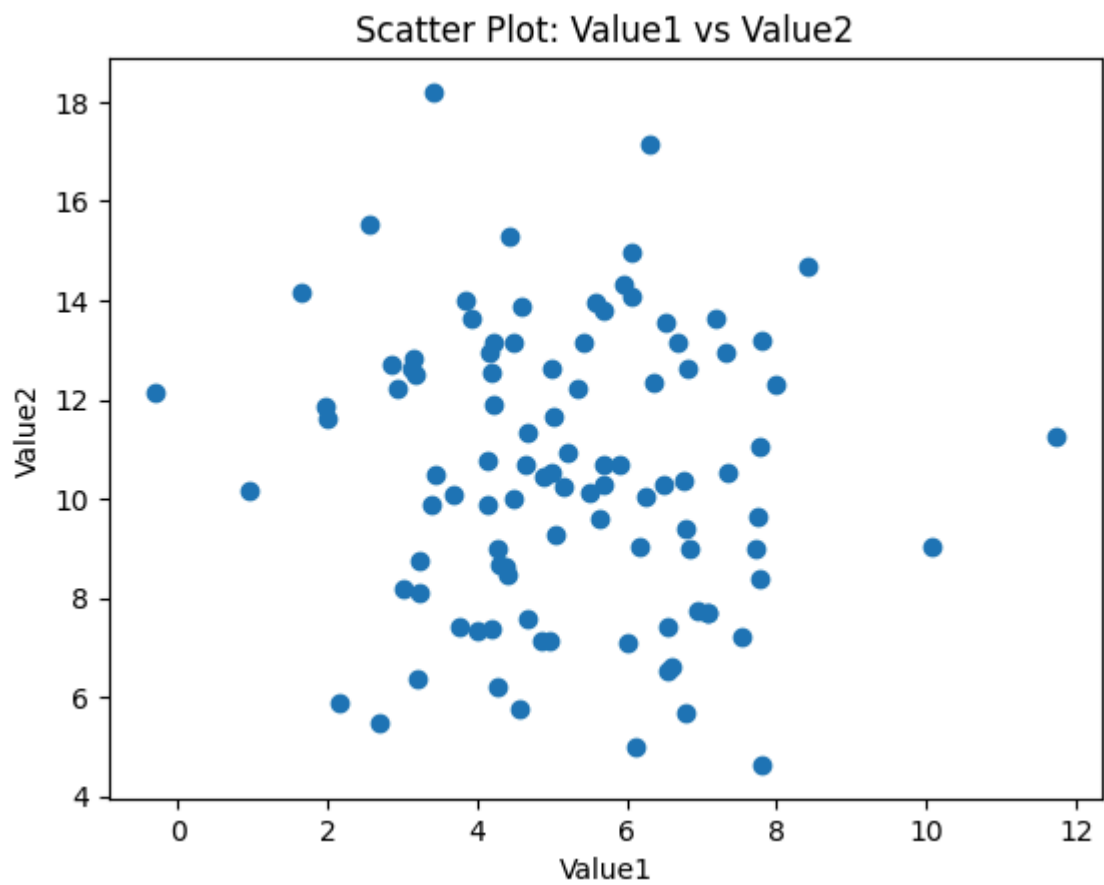
```
# --- 4. Boxplot
sns.boxplot(x='Category', y='Value1', data=data)
plt.title('Boxplot of Value1 by Category')
plt.show()
```



```
# --- 5. Violin Plot
sns.violinplot(x='Category', y='Value1', data=data)
plt.title('Violin Plot of Value1 by Category')
plt.show()
```



```
# --- 6. Scatter Plot
plt.scatter(data['Value1'], data['Value2'])
plt.xlabel('Value1')
plt.ylabel('Value2')
plt.title('Scatter Plot: Value1 vs Value2')
plt.show()
```

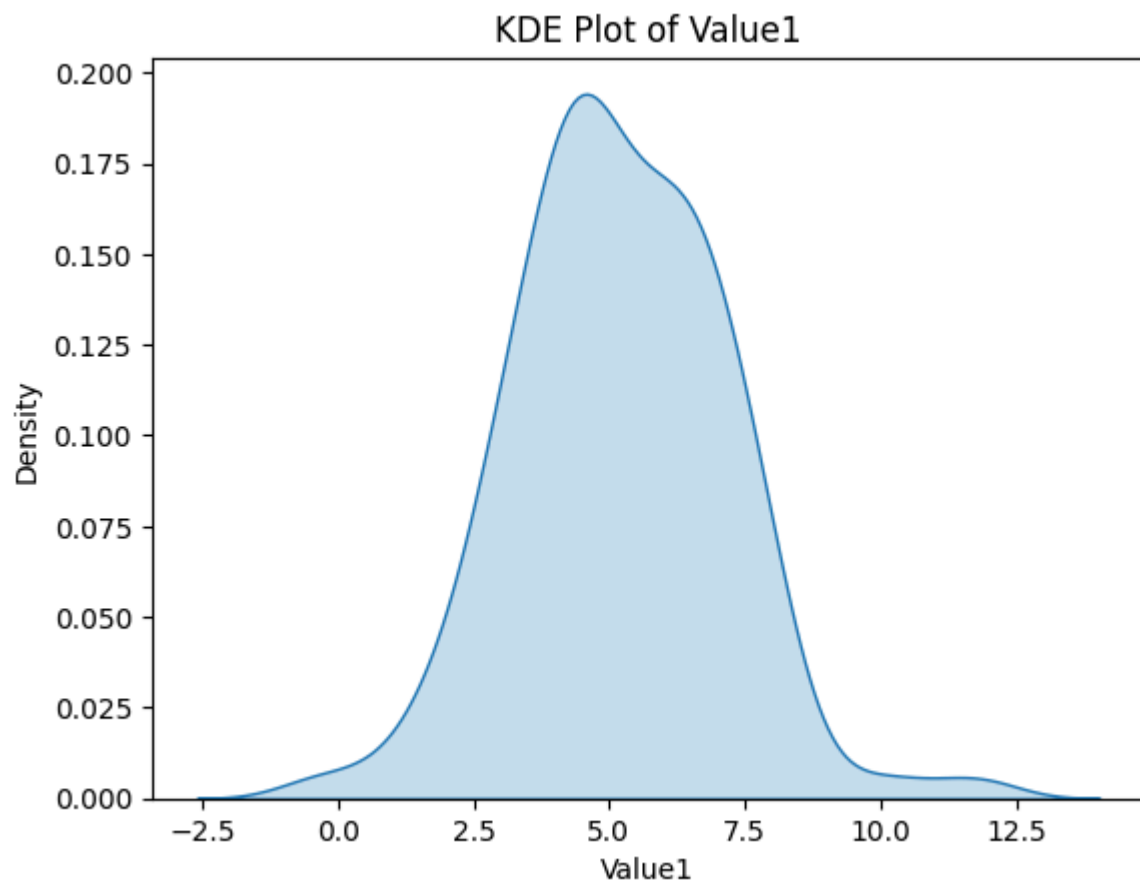


```
# --- 7. KDE Plot
sns.kdeplot(data['Value1'], shade=True)
plt.title('KDE Plot of Value1')
plt.show()
```

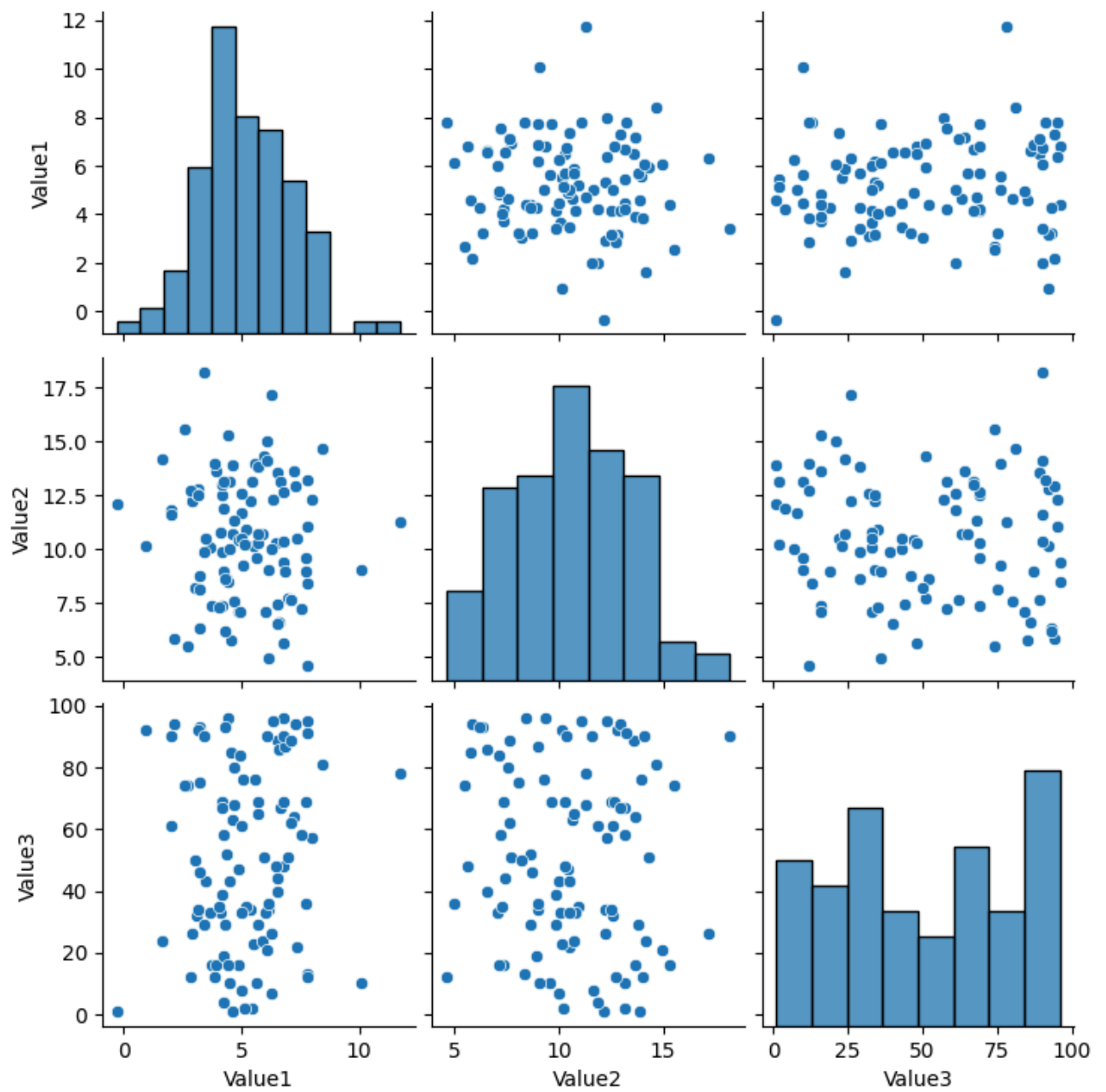
 /tmp/ipython-input-9-3442912687.py:2: FutureWarning:

``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['Value1'], shade=True)
```

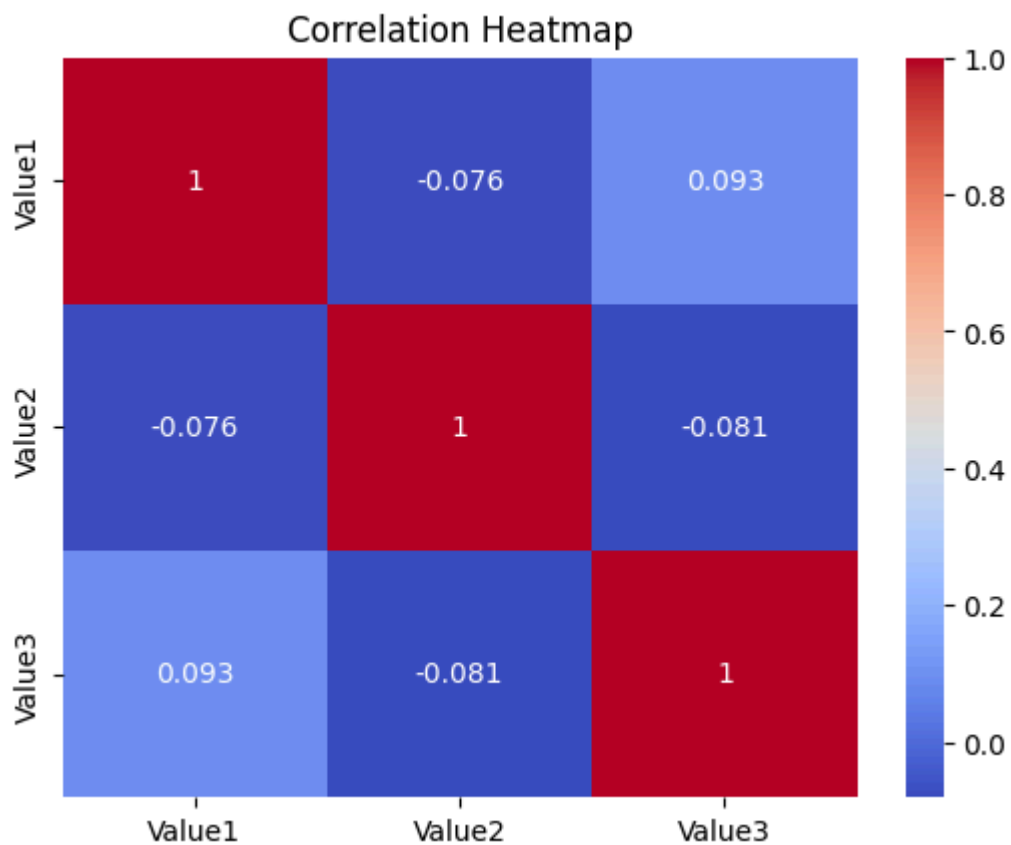


```
# --- 8. Pairplot  
sns.pairplot(data[['Value1', 'Value2', 'Value3']])  
plt.show()
```

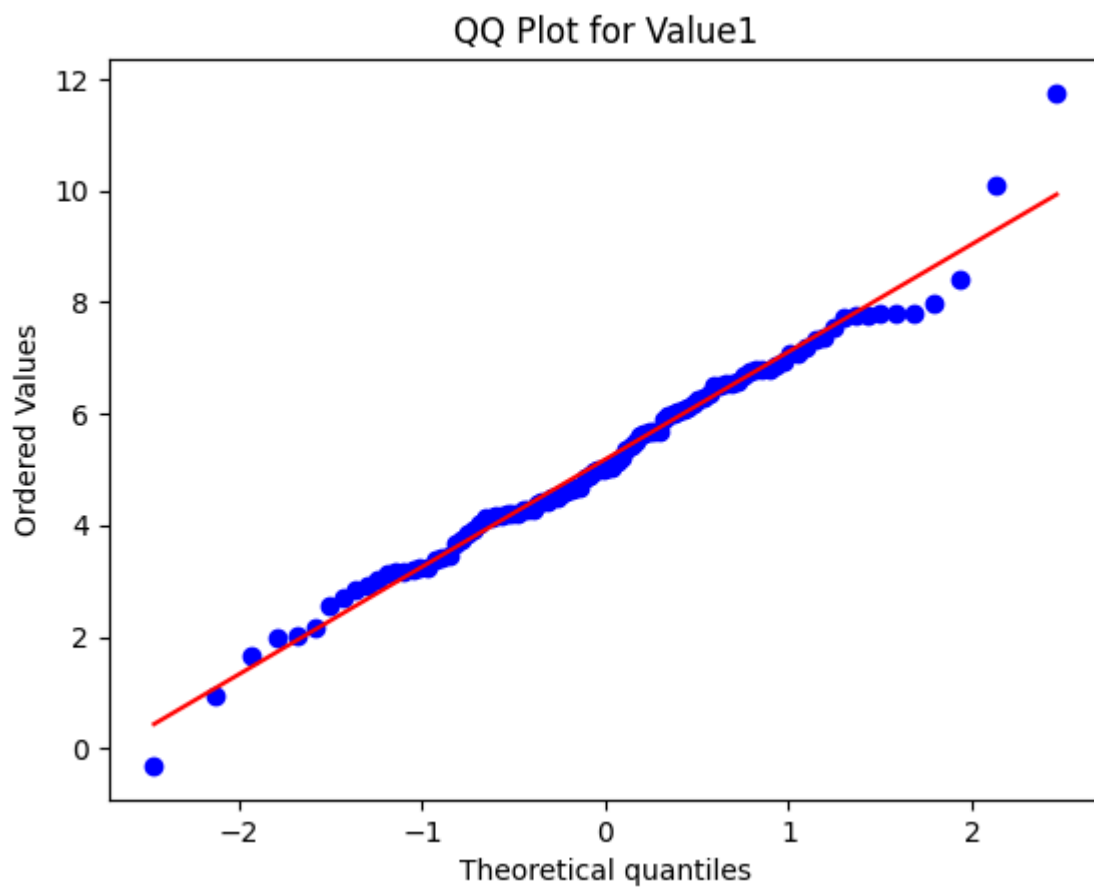


```
# --- 9. Correlation Heatmap
corr = data[['Value1', 'Value2', 'Value3']].corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```





```
# --- 10. QQ Plot  
stats.probplot(data['Value1'], dist="norm", plot=plt)  
plt.title('QQ Plot for Value1')  
plt.show()
```



### 📊 Matrix of Charts using Subplots

# This approach displays multiple chart types in a single view using a 2x2 grid layout.  
# Helps in comparing different visualizations side-by-side.

# --- 13. Chart Grid View Using Subplots

```
fig, axs = plt.subplots(2, 2, figsize=(12, 8))
axs[0, 0].hist(data['Value1'], bins=15, color='orange')
axs[0, 0].set_title('Histogram')
axs[0, 1].scatter(data['Value1'], data['Value2'])
axs[0, 1].set_title('Scatter Plot')
axs[1, 0].boxplot(data['Value2'])
axs[1, 0].set_title('Boxplot')
sns.kdeplot(data['Value3'], ax=axs[1, 1], shade=True)
axs[1, 1].set_title('KDE Plot')
plt.tight_layout()
plt.suptitle('Charts Displayed in Matrix Form', fontsize=16, y=1.05)
plt.show()
```

# --- 14. Seaborn FacetGrid View

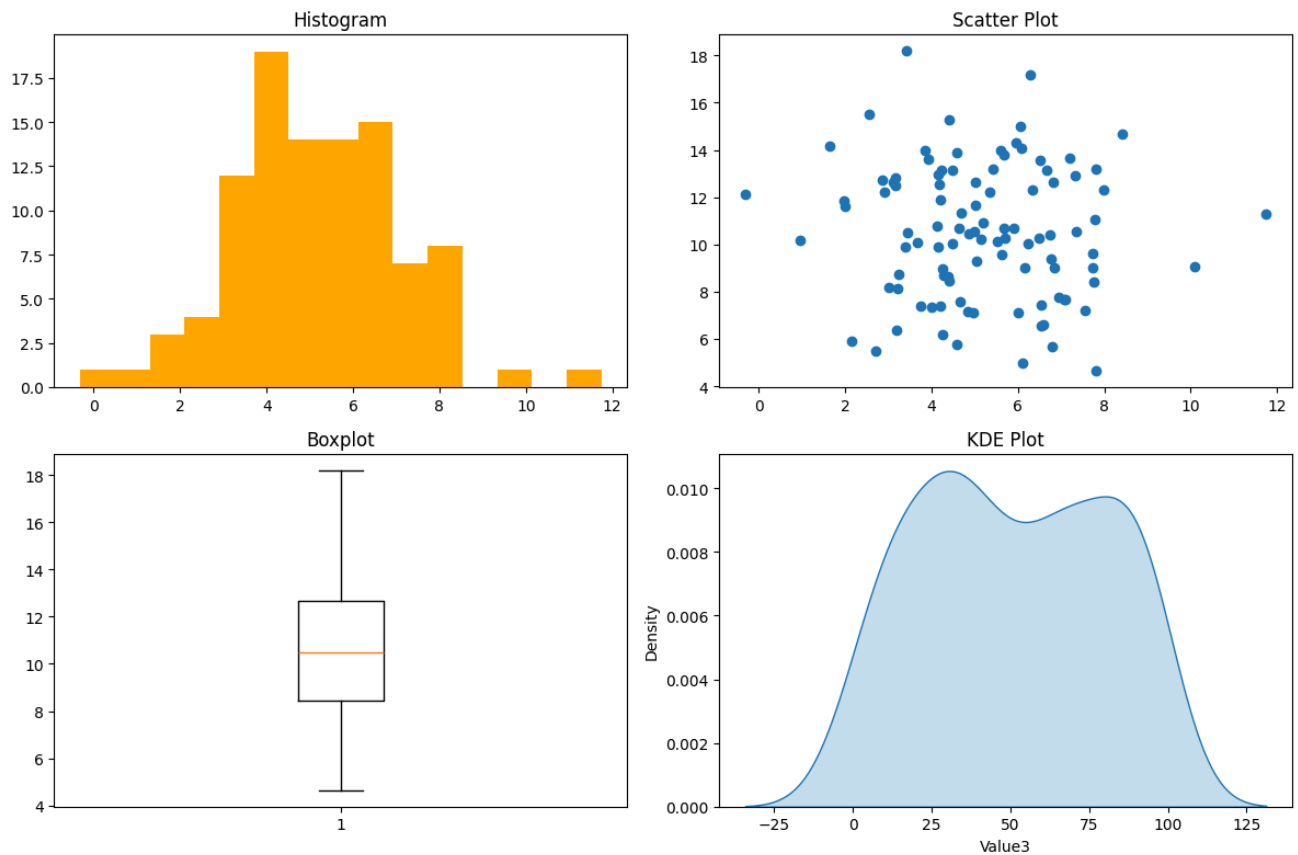
```
facet = sns.FacetGrid(data, col='Category')
facet.map(plt.hist, 'Value1', bins=10, color='teal')
facet.fig.suptitle('Histogram of Value1 by Category (FacetGrid)', y=1.1)
plt.show()
```

🔄 /tmp/ipython-input-13-427440123.py:13: FutureWarning:

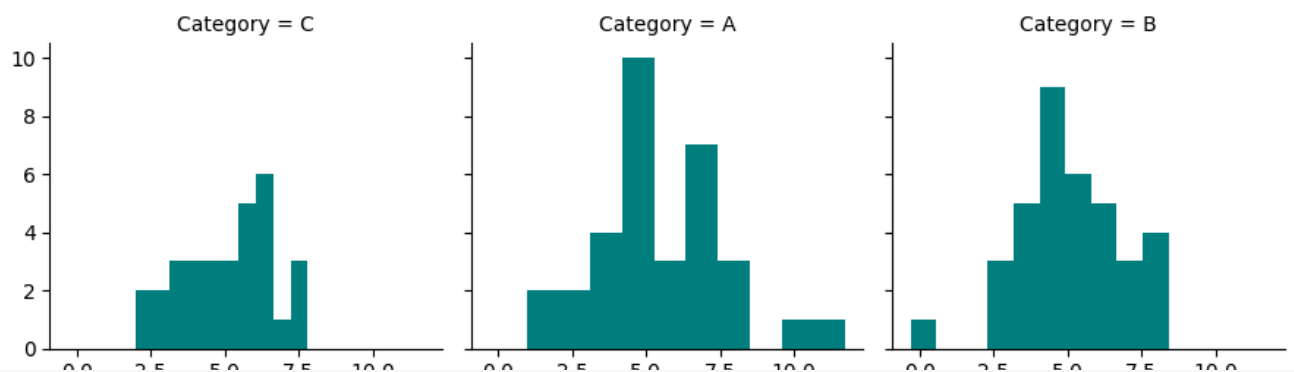
`shade` is now deprecated in favor of `fill`; setting `fill=True`.  
This will become an error in seaborn v0.14.0; please update your code.

```
sns.kdeplot(data['Value3'], ax=axes[1, 1], shade=True)
```

Charts Displayed in Matrix Form



Histogram of Value1 by Category (FacetGrid)



```
### 🧐 Exploratory Data Analysis (EDA)
# Involves summarizing the main characteristics of the data.
# This section includes statistical summaries, pairplots, and correlation heatmaps.

# --- 🧠 EDA Workflow
print("\n--- Exploratory Data Analysis (EDA) Summary ---")
print(data.describe())
print("\nMissing values check:")
print(data.isnull().sum())

sns.pairplot(data, hue='Category')
```