

Matplotlib - Scatter Plots and Histograms

Basic Scatter Plot and Histogram

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

np.random.seed(42)
x = np.random.randn(100)
y = 2 * x + np.random.randn(100) * 0.5

plt.figure(figsize=(12, 5))

# Scatter plot
plt.subplot(1, 2, 1)
plt.scatter(x, y, alpha=0.6)
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Scatter Plot')
plt.grid(True, alpha=0.3)

# Histogram
plt.subplot(1, 2, 2)
plt.hist(y, bins=20, edgecolor='black', alpha=0.7)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.grid(True, alpha=0.3)

plt.tight_layout()
plt.show()
```

Advanced Scatter Plot

```
# Generate data with categories
n = 200
categories = np.random.choice(['A', 'B', 'C'], n)
x = np.random.randn(n)
y = np.random.randn(n)
colors = {'A': 'red', 'B': 'blue', 'C': 'green'}
sizes = np.random.randint(20, 200, n)

plt.figure(figsize=(10, 8))
for category in ['A', 'B', 'C']:
    mask = categories == category
    plt.scatter(x[mask], y[mask],
                c=colors[category],
                s=sizes[mask],
                alpha=0.6,
                label=f'Category {category}',
                edgecolors='black',
                linewidths=0.5)

plt.xlabel('X Value', fontsize=12)
plt.ylabel('Y Value', fontsize=12)
plt.title('Scatter Plot with Categories', fontsize=14, fontweight='bold')
plt.legend(fontsize=10)
plt.grid(True, alpha=0.3, linestyle='--')
plt.show()
```

Multiple Histograms

```

data1 = np.random.normal(0, 1, 1000)
data2 = np.random.normal(2, 1.5, 1000)
data3 = np.random.normal(-2, 0.8, 1000)

plt.figure(figsize=(12, 6))
plt.hist([data1, data2, data3], bins=30,
        label=[' $\mu=0, \sigma=1$ ', ' $\mu=2, \sigma=1.5$ ', ' $\mu=-2, \sigma=0.8$ '],
        color=['blue', 'red', 'green'],
        alpha=0.5, edgecolor='black')

plt.xlabel('Value', fontsize=12)
plt.ylabel('Frequency', fontsize=12)
plt.title('Overlapping Histograms', fontsize=14)
plt.legend(fontsize=10)
plt.grid(True, alpha=0.3, axis='y')
plt.show()

```

2D Histogram (Hexbin)

```

x = np.random.randn(10000)
y = x + np.random.randn(10000) * 0.5

plt.figure(figsize=(12, 5))

# 2D histogram
plt.subplot(1, 2, 1)
plt.hist2d(x, y, bins=30, cmap='Blues')
plt.colorbar(label='Count')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('2D Histogram')

# Hexbin plot
plt.subplot(1, 2, 2)
plt.hexbin(x, y, gridsize=30, cmap='Reds')
plt.colorbar(label='Count')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Hexbin Plot')

plt.tight_layout()
plt.show()

```

Box Plot and Violin Plot

```

data = [np.random.normal(0, std, 100) for std in range(1, 5)]

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(14, 6))

# Box plot
ax1.boxplot(data, labels=['1', '2', '3', '4'])
ax1.set_xlabel('Standard Deviation', fontsize=12)
ax1.set_ylabel('Value', fontsize=12)
ax1.set_title('Box Plot', fontsize=14)
ax1.grid(True, alpha=0.3, axis='y')

# Violin plot
parts = ax2.violinplot(data, positions=[1, 2, 3, 4],
                       showmeans=True, showmedians=True)
ax2.set_xlabel('Standard Deviation', fontsize=12)
ax2.set_ylabel('Value', fontsize=12)
ax2.set_title('Violin Plot', fontsize=14)
ax2.set_xticks([1, 2, 3, 4])
ax2.set_xticklabels(['1', '2', '3', '4'])
ax2.grid(True, alpha=0.3, axis='y')

plt.tight_layout()
plt.show()

```

