Matplotlib - Heatmaps for Matrices

Basic Correlation Heatmap

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

data = np.random.randn(50, 4)
correlation = np.corrcoef(data.T)

plt.figure(figsize=(8, 6))
plt.imshow(correlation, cmap='coolwarm', aspect='auto')
plt.colorbar(label='Correlation')
plt.title('Correlation Matrix')
plt.xticks(range(4), ['Feature 1', 'Feature 2',
                      'Feature 3', 'Feature 4'])
plt.yticks(range(4), ['Feature 1', 'Feature 2',
                      'Feature 3', 'Feature 4'])
plt.show()
```

Heatmap with Annotations

```
# Create sample data
data = np.array([[1, 2, 3, 4],
                 [5, 6, 7, 8],
                 [9, 10, 11, 12],
                 [13, 14, 15, 16]])

plt.figure(figsize=(8, 6))
im = plt.imshow(data, cmap='viridis')

# Add text annotations
for i in range(data.shape[0]):
    for j in range(data.shape[1]):
        text = plt.text(j, i, data[i, j],
                        ha="center", va="center",
                        color="white", fontsize=14)

plt.colorbar(im, label='Value')
plt.title('Annotated Heatmap', fontsize=14)
plt.xlabel('Column Index')
plt.ylabel('Row Index')
plt.xticks(range(4))
plt.yticks(range(4))
plt.show()
```

Multiple Heatmaps Comparison

```
fig, axes = plt.subplots(1, 3, figsize=(15, 5))

# Three different correlation matrices
for idx, (ax, cmap) in enumerate(zip(axes,
                                     ['coolwarm', 'RdYlBu', 'viridis'])):
    data = np.random.randn(50, 5)
    corr = np.corrcoef(data.T)

    im = ax.imshow(corr, cmap=cmap, vmin=-1, vmax=1, aspect='auto')
    ax.set_title(f'Colormap: {cmap}')
    ax.set_xticks(range(5))
    ax.set_yticks(range(5))
```

```
    # Add colorbar
    plt.colorbar(im, ax=ax)

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

Confusion Matrix Heatmap

```
from sklearn.metrics import confusion_matrix

# Simulated predictions and true labels
y_true = np.random.randint(0, 3, 100)
y_pred = np.random.randint(0, 3, 100)

cm = confusion_matrix(y_true, y_pred)
classes = ['Class 0', 'Class 1', 'Class 2']

plt.figure(figsize=(8, 6))
im = plt.imshow(cm, cmap='Blues')

# Annotations
for i in range(len(classes)):
    for j in range(len(classes)):
        text = plt.text(j, i, cm[i, j],
                        ha="center", va="center",
                        color="white" if cm[i, j] > cm.max()/2 else "black",
                        fontsize=16, fontweight='bold')

plt.colorbar(im, label='Count')
plt.title('Confusion Matrix', fontsize=14, fontweight='bold')
plt.xlabel('Predicted Label', fontsize=12)
plt.ylabel('True Label', fontsize=12)
plt.xticks(range(3), classes, rotation=45)
plt.yticks(range(3), classes)
plt.tight_layout()
plt.show()
```

Advanced Heatmap with Custom Scaling

```
# Create data with different scales
data = np.random.exponential(2, (10, 10))

fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15, 5))

# Linear scale
im1 = ax1.imshow(data, cmap='hot')
ax1.set_title('Linear Scale')
plt.colorbar(im1, ax=ax1)

# Log scale
im2 = ax2.imshow(np.log(data + 1), cmap='hot')
ax2.set_title('Log Scale')
plt.colorbar(im2, ax=ax2)

# Normalized
data_norm = (data - data.min()) / (data.max() - data.min())
im3 = ax3.imshow(data_norm, cmap='hot')
ax3.set_title('Normalized [0, 1]')
plt.colorbar(im3, ax=ax3)

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