Data Visualization in Python

Hariharan Vels

What is Data Visualization?

- Data visualization is the graphical representation of information and data.
- Helps in understanding trends, outliers, and patterns in data.
- Common libraries in Python: Matplotlib, Seaborn, Plotly.

Importance of Data Visualization

- Simplifies complex data.
- Enhances data analysis and decision-making.
- Helps in identifying trends and correlations.

Introduction to Matplotlib

- Matplotlib is a widely used library for creating static, animated, and interactive plots.
- Install using:

Command

```
pip install matplotlib
```

- Import Matplotlib:

Python Code

```
import matplotlib.pyplot as plt
```

Basic Plot in Matplotlib

Example

```
 import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [10, 20, 25, 30, 40]
plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Simple Line Plot')
plt.show()
```

Sine Wave Plot in Matplotlib

Example

```
 import numpy as np
import matplotlib.pyplot as plt
x = np.linspace(0, 10, 100)
y = np.sin(x)
plt.plot(x, y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Sine Wave')
plt.show()
```

Introduction to Seaborn

- Seaborn is built on top of Matplotlib and provides beautiful themes and high-level interface.
- Install using:

Command

```
pip install seaborn
```

- Import Seaborn:

Python Code

```
import seaborn as sns
```

Basic Plot in Seaborn

Example

```
 import seaborn as sns
import matplotlib.pyplot as plt
data = sns.load_dataset("tips")
sns.scatterplot(x='total_bill', y='tip', data=data)
plt.show()
```

Introduction to Plotly

- Plotly is an interactive visualization library.
- Install using:

Command

```
pip install plotly
```

- Import Plotly:

Python Code

```
import plotly.express as px
```

Basic Interactive Plot in Plotly

Example

```
 import plotly.express as px
data = px.data.iris()
fig = px.scatter(data, x='sepal_width', y='sepal_length',
color='species')
fig.show()
```

Single Plot Example

Creating Simple Plot

```
 x = np.linspace(0, 10, 100)
fig, ax = plt.subplots()
ax.plot(x, np.sin(x))
ax.set_title('A single plot', fontsize=14)
plt.show()
```

Importing Libraries

```
1 import matplotlib.pyplot as plt
2 import seaborn as sns
3 import numpy as np
4 import pandas as pd
```

Single Plot Example

```
x = np.linspace(0, 10, 100)
fig, ax = plt.subplots()
ax.plot(x, np.sin(x))
ax.set_title('A single plot', {'fontsize':14})
plt.show()
```

Multiple Subplots Example

```
fig, axs = plt.subplots(2)
axs[0].plot(x, np.sin(x))
axs[1].plot(x, -np.sin(x))
fig.suptitle('Main Figure Title', fontsize=12)
axs[0].set_title('Axes 0 Title', fontsize=10)
axs[1].set_title('Axes 1 Title', fontsize=10)
plt.show()
```

Shared X-axis Subplots

```
fig, axs = plt.subplots(2, sharex=True)
axs[0].plot(x, np.sin(x))
axs[1].plot(x, -np.sin(x))
fig.suptitle('Main Figure Title', fontsize=12)
axs[0].set_title('Axes 0 Title', fontsize=10)
axs[1].set_title('Axes 1 Title', fontsize=10)
plt.show()
```

2D Grid Subplots

```
fig, axs = plt.subplots(2, 2)
fig.suptitle('Main Title', fontsize=14)
axs[0, 0].plot(x, np.sin(x), 'g', linewidth=2)
axs[0, 0].set_title('Axes [0,0] Title')
axs[0, 1].plot(x, np.cos(x), '--r', linewidth=3)
axs[0, 1].set_title('Axes [0,1] Title')
plt.show()
```

Seaborn Styling

```
sns.set_style('ticks')
fig, axs = plt.subplots(2, 2, sharex=True, sharey=True
    )
(ax1, ax2), (ax3, ax4) = axs
fig.suptitle('Main Title', fontsize=14)
ax1.plot(x, np.sin(x), 'g', linewidth=2)
ax2.plot(x, np.cos(x), '--r', linewidth=3)
ax3.plot(x, -np.sin(x), '-.b', linewidth=2)
ax4.plot(x, -np.cos(x), ':y', linewidth=3)
plt.show()
```

Histogram Plot

```
x = np.random.randn(10000)
n_bins = 50
plt.hist(x, n_bins, density=1, facecolor="green",
    alpha=0.5)
plt.xlabel("X values")
plt.ylabel("Frequency")
plt.title("Histogram")
sns.distplot(x, bins=n_bins, color='g')
plt.show()
```

Pie Chart Example

```
product = ['Webcam', 'Keyboard', 'Mouse', 'Mic', 'PC']
pct = [20, 25, 30, 10, 15]
cmap = plt.get_cmap('Set2')
colors = cmap(np.arange(5))
plt.pie(pct, colors=colors, labels=product, autopct='
    %1.1f%%', shadow=True)
plt.title('Pie Chart')
plt.show()
```