# SHADOWFOX DATA SCIENCE INTERNSHIP

## VISUALIZATION LIBRARY DOCUMENTATION

### BY MANASA M

## 1. Library Overview

### 1.1 Plotly

Plotly is a powerful Python visualization library known for its interactive and web-friendly visualizations. It supports a wide variety of chart types, from basic line and bar charts to advanced 3D and geospatial plots. Plotly is widely used in dashboards and web applications due to its interactive capabilities.

**Key Features:**

- Highly interactive visualizations.
- Supports a wide range of charts, including 3D plots and maps.
- Seamless integration with web applications.
- Works well with large datasets.
- Built-in support for animations and hover tooltip.

### 1.2 Seaborn

Seaborn is a high-level data visualization library built on Matplotlib. It simplifies the process of creating aesthetically pleasing and informative statistical visualizations. Seaborn is particularly useful for exploring and analyzing data distributions, relationships, and categorical comparisons.

**Key Features:**

- Built-in themes for visually appealing plots.
- Easy integration with Pandas DataFrames.
- Advanced statistical visualizations, including regression plots and categorical plots.
- Automatic estimation and visualization of distributions.
- Efficient handling of multi-variable datasets.

## 2. Graph Types

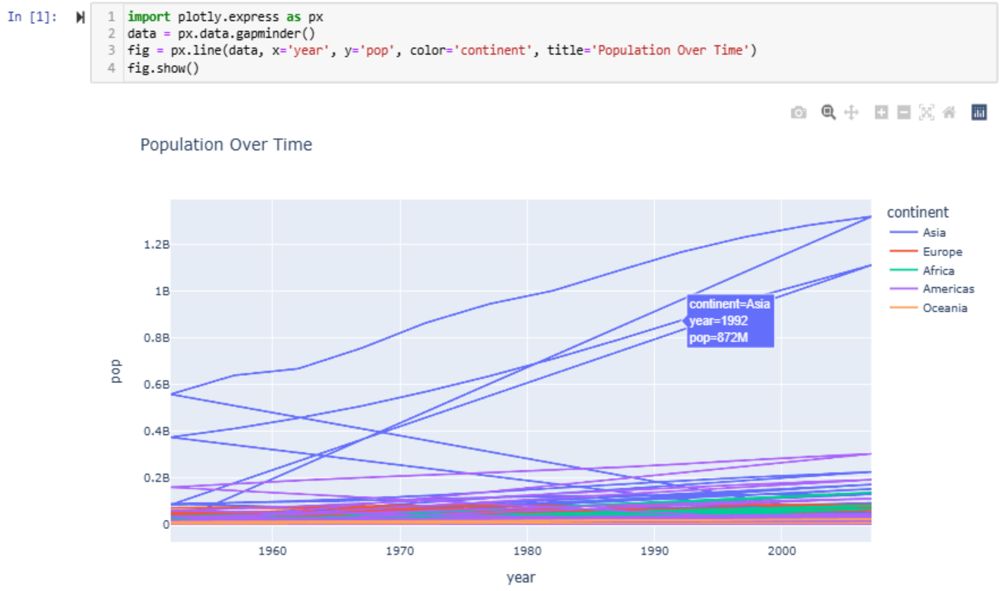
### 2.1 Plotly Graphs

1. **Line Plot**

   **Description:** Similar to Seaborn, used to show trends over time.

   **Use Case:** Analyzing stock market trends, Monitoring real-time sensor data

**Example:**



In [1]:

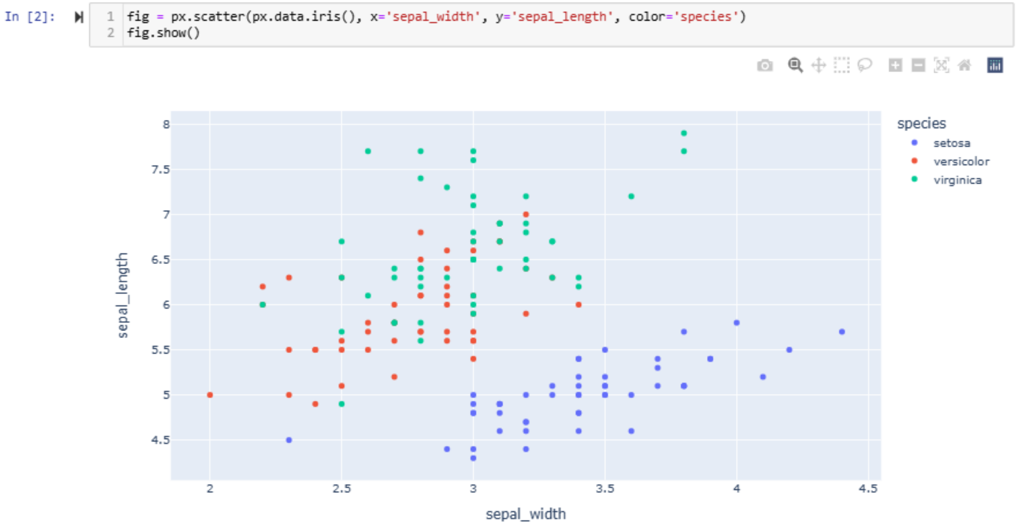
```
import plotly.express as px
data = px.data.gapminder()
fig = px.line(data, x='year', y='pop', color='continent', title='Population Over Time')
fig.show()
```

## 2. Scatter Plot

**Description:** Shows relationships between numerical variables.

**Use Case:** Analyzing correlations in datasets.

**Example:**



In [2]:

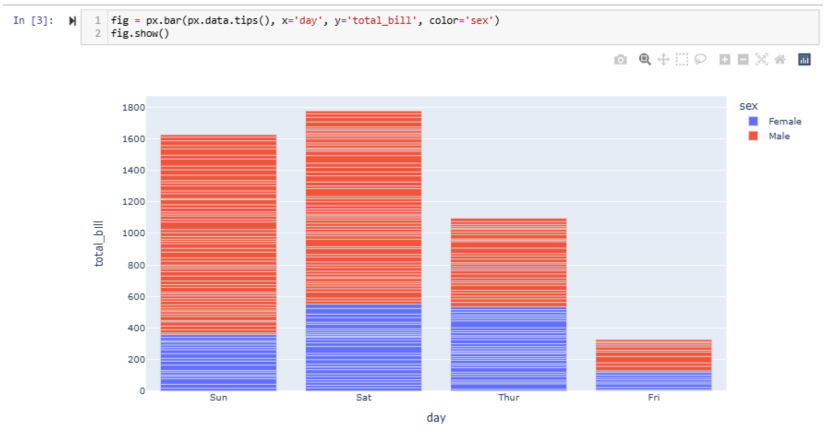
```
fig = px.scatter(px.data.iris(), x='sepal_width', y='sepal_length', color='species')
fig.show()
```

### 3. Bar chart

**Description:** A bar chart displays categorical data as bars.

**Use Case:** Comparing sales data for different products.

**Example:**



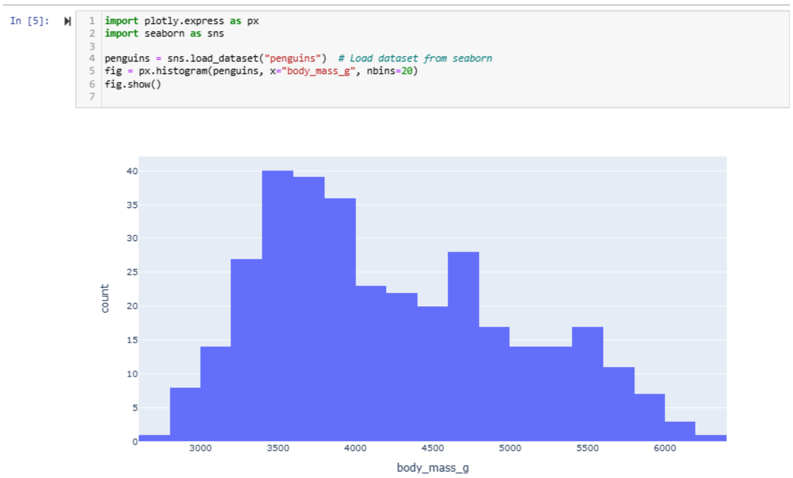
```
fig = px.bar(px.data.tips(), x='day', y='total_bill', color='sex')
fig.show()
```

### 4. Histogram

**Description:** A histogram represents the distribution of numerical data using rectangular bars.

**Use Case:** Understanding customer age distribution in a business.

**Example:**



```
import plotly.express as px
import seaborn as sns

penguins = sns.load_dataset("penguins")  # Load dataset from seaborn
fig = px.histogram(penguins, x="body_mass_g", nbins=20)
fig.show()

```

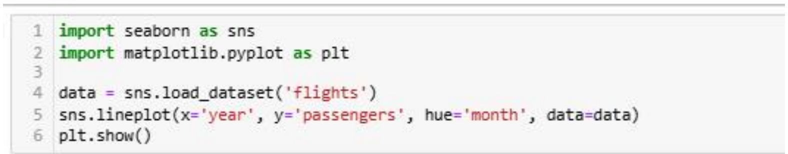
## 2.2 Seaborn Graphs

### 1. Line Plot

**Description:** A line plot is used to show trends and changes over time.

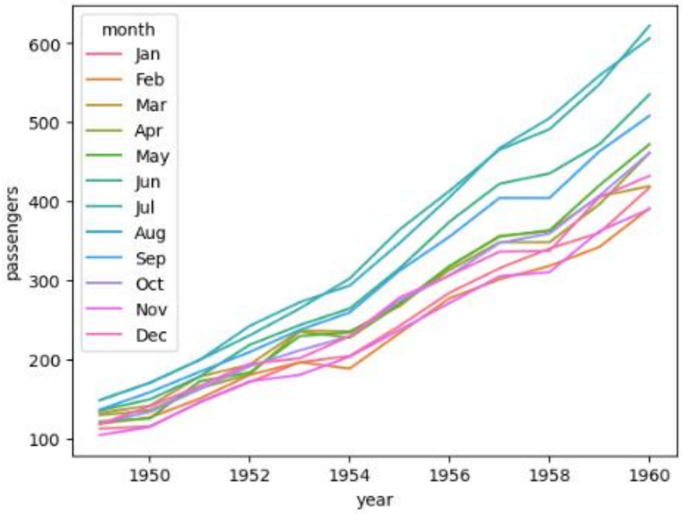
**Use Case:** Analyzing sales growth, weather trends, or website traffic.

**Example:**



```python
import seaborn as sns
import matplotlib.pyplot as plt

data = sns.load_dataset('flights')
sns.lineplot(x='year', y='passengers', hue='month', data=data)
plt.show()
```

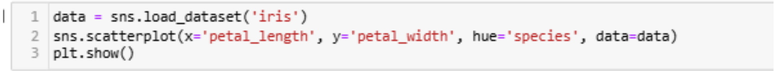


### 2. Scatter Plot

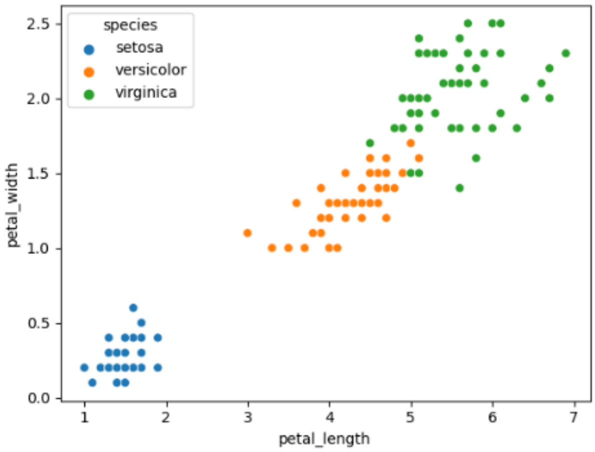
**Description:** Used to observe relationships and trends between two continuous variables.

**Use Case:** Comparing petal length and width in different species of flowers.

**Example**



```python
data = sns.load_dataset('iris')
sns.scatterplot(x='petal_length', y='petal_width', hue='species', data=data)
plt.show()
```

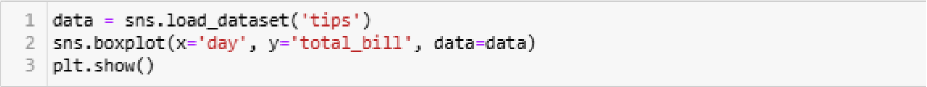


3. **Bar chart**

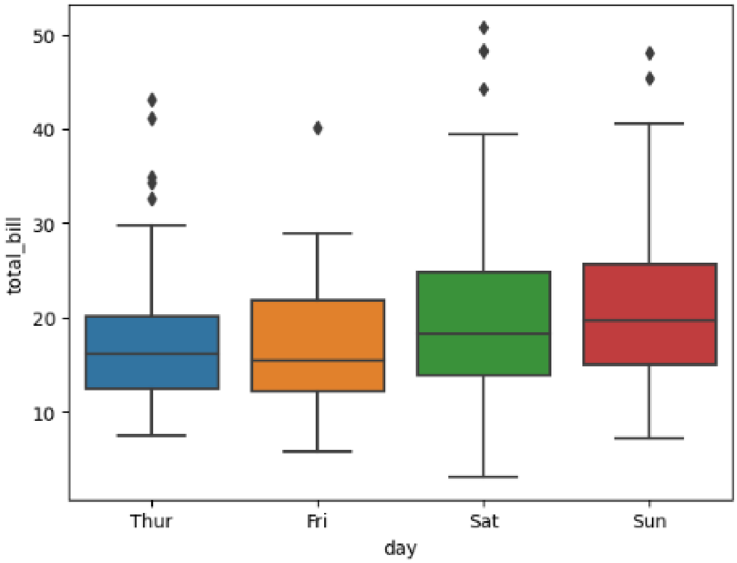
**Description:** A box plot provides insights into data distribution, median, and outliers.

**Use Case:** Comparing total bill amounts across different days in a restaurant dataset.

**Example:**



```
data = sns.load_dataset('tips')
sns.boxplot(x='day', y='total_bill', data=data)
plt.show()
```

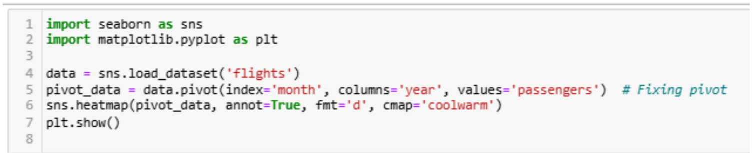


4. **Heatmap**

**Description:** Displays correlations or patterns in matrix data using color variations.

**Use Case:** Identifying correlations between various financial indicators.

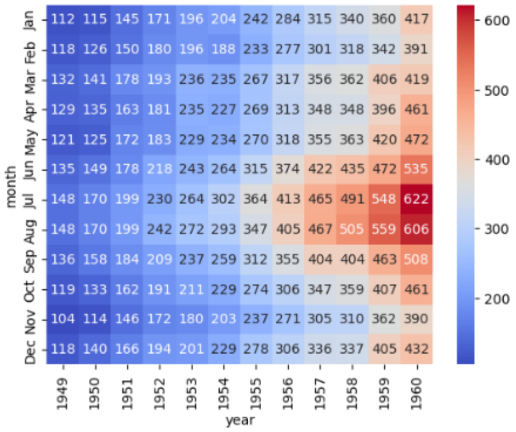
**Example:**



```
import seaborn as sns
import matplotlib.pyplot as plt

data = sns.load_dataset('flights')
pivot_data = data.pivot(index='month', columns='year', values='passengers')  # Fixing pivot
sns.heatmap(pivot_data, annot=True, fmt='d', cmap='coolwarm')
plt.show()

```



| month | 1949 | 1950 | 1951 | 1952 | 1953 | 1954 | 1955 | 1956 | 1957 | 1958 | 1959 | 1960 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Jan | 112 | 115 | 145 | 171 | 196 | 204 | 242 | 284 | 315 | 340 | 360 | 417 |
| Feb | 118 | 126 | 150 | 180 | 196 | 188 | 233 | 277 | 301 | 318 | 342 | 391 |
| Mar | 132 | 141 | 178 | 193 | 236 | 235 | 267 | 317 | 356 | 362 | 406 | 419 |
| Apr | 129 | 135 | 163 | 181 | 235 | 227 | 269 | 313 | 348 | 348 | 396 | 461 |
| May | 121 | 125 | 172 | 183 | 229 | 234 | 270 | 318 | 355 | 363 | 420 | 472 |
| Jun | 135 | 149 | 178 | 218 | 243 | 264 | 315 | 374 | 422 | 435 | 472 | 535 |
| Jul | 148 | 170 | 199 | 230 | 264 | 302 | 364 | 413 | 465 | 491 | 548 | 622 |
| Aug | 148 | 170 | 199 | 242 | 272 | 293 | 347 | 405 | 467 | 505 | 559 | 606 |
| Sep | 136 | 158 | 184 | 209 | 237 | 259 | 312 | 355 | 404 | 404 | 463 | 508 |
| Oct | 119 | 133 | 162 | 191 | 211 | 229 | 274 | 306 | 347 | 359 | 407 | 461 |
| Nov | 104 | 114 | 146 | 172 | 180 | 203 | 237 | 271 | 305 | 310 | 362 | 390 |
| Dec | 118 | 140 | 166 | 194 | 201 | 229 | 278 | 306 | 336 | 337 | 405 | 432 |

## 3. Comparison

| **Feature** | **Seaborn** | **Plotly** |
|---|---|---|
| **Ease of Use** | Simple syntax, works well with Pandas | Slightly more complex but very powerful |
| **Customization** | Limited compared to Plotly | Extensive customization options |
| **Interactivity** | Static images | Fully interactive visualizations |
| **Performance** | Works well with moderate datasets | Handles large datasets efficiently |
| **Web Integration** | Not designed for web use | Generates HTML & JavaScript for web deployment |

## Conclusion

- **Seaborn** is excellent for statistical analysis and quick visualizations, especially when working with Pandas.
- **Plotly** is ideal for creating interactive and web-based visualizations.
- If you need static, publication-ready graphs, use Seaborn.
- If you require interactive dashboards and large dataset support, choose Plotly.