Visualization with Matplotlib, Seaborn, Plotly

Matplotlib:

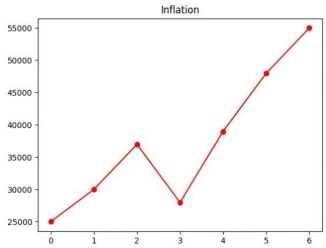
Matplotlib is a popular Python library that helps in making static, animated, and interactive data visualizations. It offers a complete range of tools for creating different kinds of plots and charts, allowing users to represent and analyze data effectively.

Example:

```python
import matplotlib.pyplot as plt
import pandas as pd

# Data
data = {"Salary": [25000, 30000, 37000, 28000, 39000, 48000, 55000]}
df = pd.DataFrame(data)

# Plotting
plt.plot(df["Salary"], color = "red", marker = "o")
plt.title("Inflation")
plt.show()
```



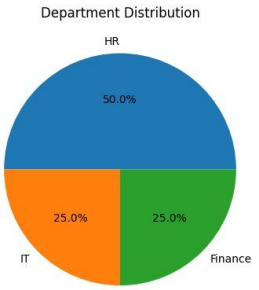

```python
import matplotlib.pyplot as plt
import pandas as pd

# Data
data = {"Salary": [10, 30, 60, 10]}
df = pd.DataFrame(data)

# Departments (exactly 7 values)
df["dept"] = ['HR', "IT", 'Finance', 'HR']

# Count each department
count = df["dept"].value_counts()

# Plot pie chart
plt.pie(count.values, labels=count.index, autopct='%1.1f%%')
plt.title("Department Distribution")
plt.show()
```



Seaborn:

Seaborn is a popular Python library for creating attractive statistical visualizations. Built on Matplotlib and integrated with Pandas, it simplifies complex plots like line charts, heatmaps and violin plots with minimal code.

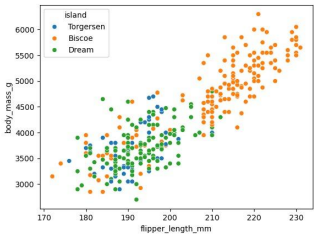
Example:

```python
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load penguins dataset
penguins = sns.load_dataset('penguins')

# Scatter plot
sns.scatterplot(data=penguins, x="flipper_length_mm", y="body_mass_g", hue = "island")

# Show plot
plt.show()
```

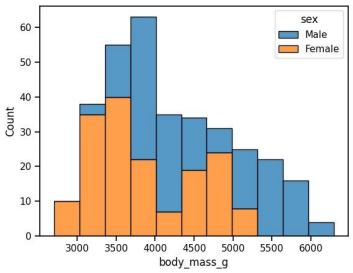


```python
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Load penguins dataset
penguins = sns.load_dataset('penguins')

# histogram:
sns. set_context ("notebook" )
sns .histplot(data = penguins, x = "body_mass_g", hue="sex", multiple="stack")

# Show plot
plt.show()
```



Plotly

Plotly stands out as a versatile and powerful library that transforms static charts into dynamic, interactive visualizations. It helps users to explore data through features like zooming, additional details and clicking for deeper insights.

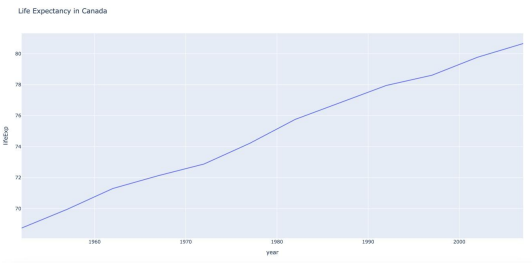
Example:

```python
import plotly.express as px
import webbrowser
import tempfile
import os

# Load data
df = px.data.gapminder().query("country == 'Canada'")
fig = px.line(df, x="year", y="lifeExp", title="Life Expectancy in Canada")

# Create a temporary HTML file
tmp_file = tempfile.NamedTemporaryFile(delete=False, suffix='.html')
fig.write_html(tmp_file.name)

# Open in default browser
webbrowser.open('file://' + os.path.realpath(tmp_file.name))
```




```python
import plotly.express as px
import webbrowser
import tempfile
import os

# Load data
df = px.data.stocks()
fig = px.line(df, x="date", y="GOOG", title="Stock Price Of Google")

# Create a temporary HTML file
tmp_file = tempfile.NamedTemporaryFile(delete=False, suffix='.html')
fig.write_html(tmp_file.name)

# Open in default browser
webbrowser.open('file://' + os.path.realpath(tmp_file.name))
```