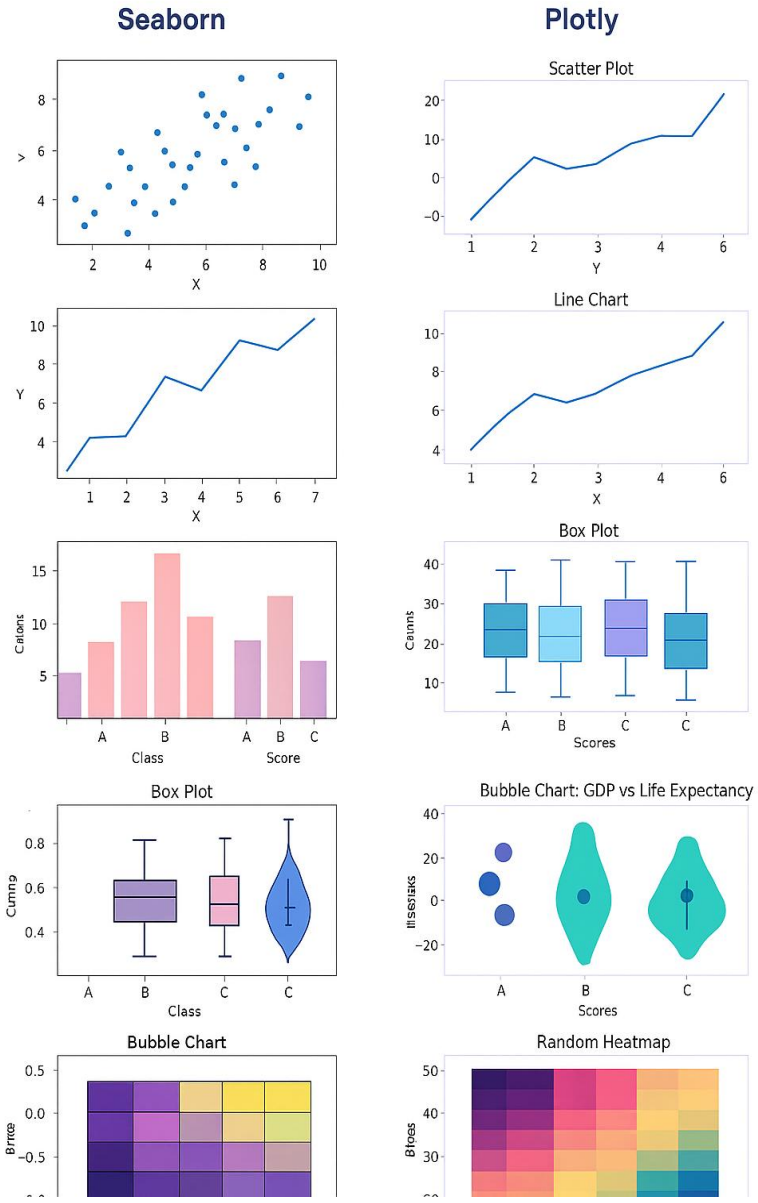
Data Visualization Python



Seaborn vs Plotly

Quick Data Visualizations Made Esay

DR SUBRAMANI

Mentor

Ramisha Rani K

Ramya Dinesh

Seaborn vs Plotly clearly so you can see when to use each, especially in your workflow of quick data visualizations - Python

Python - Choosing the Right Tool for Quick Visualizations

Plotly

- ✅ Best **for interactive, shareable** charts with zoom, hover, and tooltips.
- ✅ Ideal for dashboards, presentations, and web apps.
- ✅ Easy **to add multiple layers** (histograms + KDE, scatter + bubbles, etc.).
- ✅ Supports **animation & interactivity** effortlessly.
- ❌ Slightly **more setup** for complex statistical plots compared to Seaborn

- Workflow Tip:
- Use **Plotly.** express when you need **interactive visualization**, dashboards, or shareable results for stakeholders.

Seaborn

- ✅ Best for **static, publication-quality charts** in Python.
- ✅ Quick to create **histograms, KDEs, scatterplots, boxplots, and heatmaps**.
- ✅ Integrates seamlessly with **Pandas & Matplotlib**.
- ✅ Great for **data exploration & statistical analysis**.
- ❌ Limited **interactive features**.

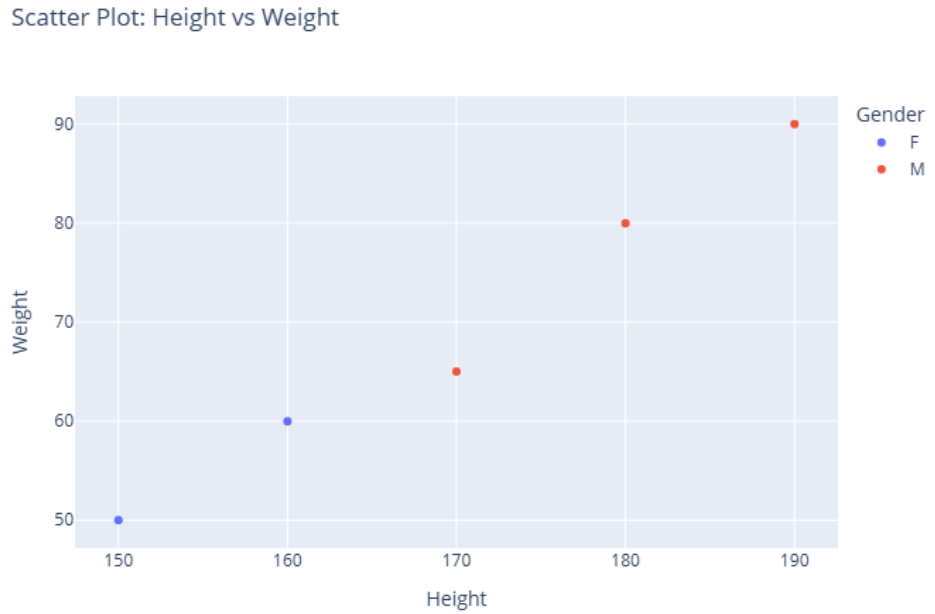
- Workflow Tip:
- Use **Seaborn** for **fast exploratory analysis** and detailed static plots in notebooks.

DR. SUBRAMANI
AI Computational Biologist

1 Scatter Plot

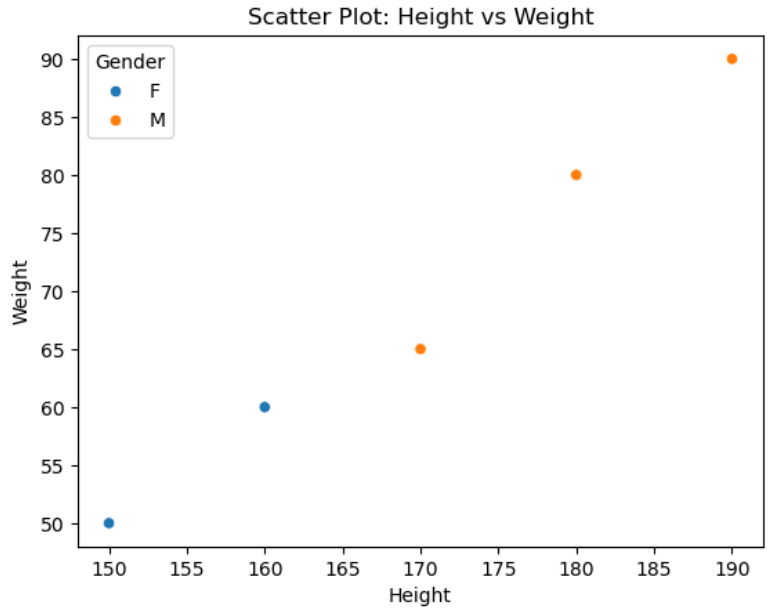
Plotly

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_scatter = pd.DataFrame({
    'Height': [150, 160, 170, 180, 190],
    'Weight': [50, 60, 65, 80, 90],
    'Gender': ['F', 'F', 'M', 'M', 'M']
})
fig = px.scatter(df_scatter, x='Height', y='Weight', color='Gender',
                 title='Scatter Plot: Height vs Weight')
fig.show()
```



Seaborn

```python
import pandas as pd
import seaborn as sns
# Sample dataset
df_scatter = pd.DataFrame({
    'Height': [150, 160, 170, 180, 190],
    'Weight': [50, 60, 65, 80, 90],
    'Gender': ['F', 'F', 'M', 'M', 'M']
})
# Simple scatter plot
sns.scatterplot(data=df_scatter, x='Height', y='Weight',
                hue='Gender').set_title('Scatter Plot: Height vs Weight')
```



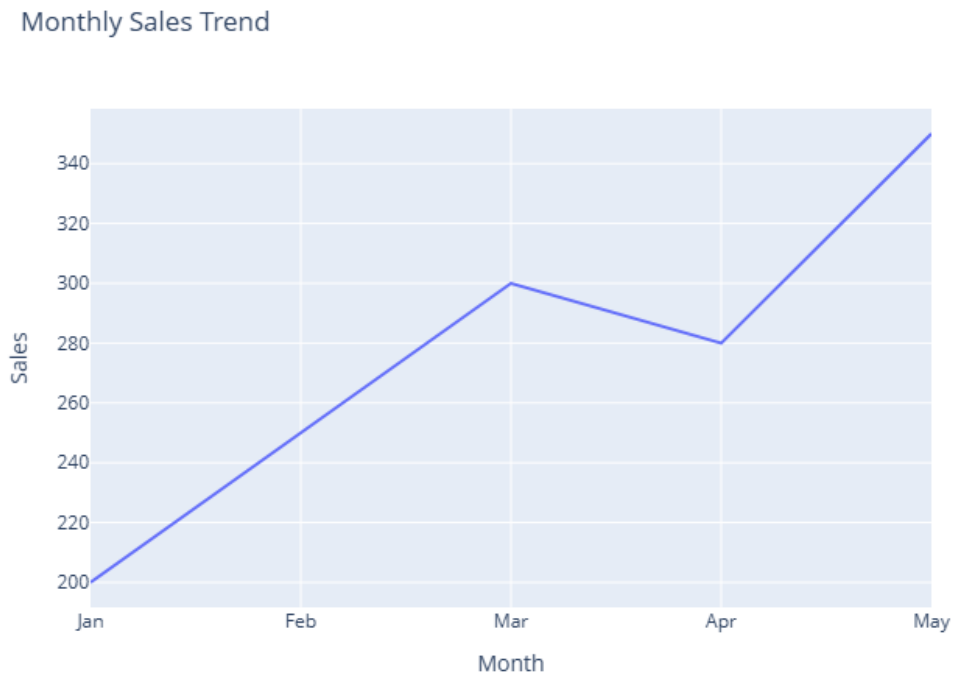
DR. SUBRAMANI
AI Computational Biologist

Plotly

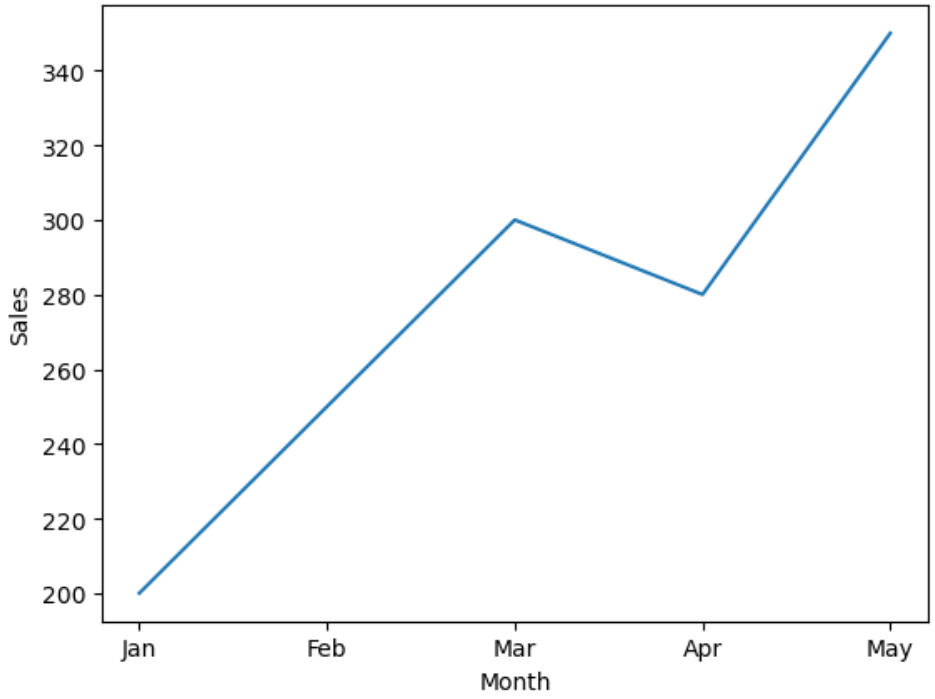
2 Line Chart

Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_line = pd.DataFrame({
    'Month': ['Jan','Feb','Mar','Apr','May'],
    'Sales': [200, 250, 300, 280, 350]
})
fig = px.line(df_line, x='Month', y='Sales', title='Monthly Sales Trend')
fig.show()
```




```python
import pandas as pd
import seaborn as sns
# Sample dataset
df_line = pd.DataFrame({
    'Month': ['Jan', 'Feb', 'Mar', 'Apr', 'May'],
    'Sales': [200, 250, 300, 280, 350]
})
# Simple line chart
sns.lineplot(data=df_line, x='Month', y='Sales')
```



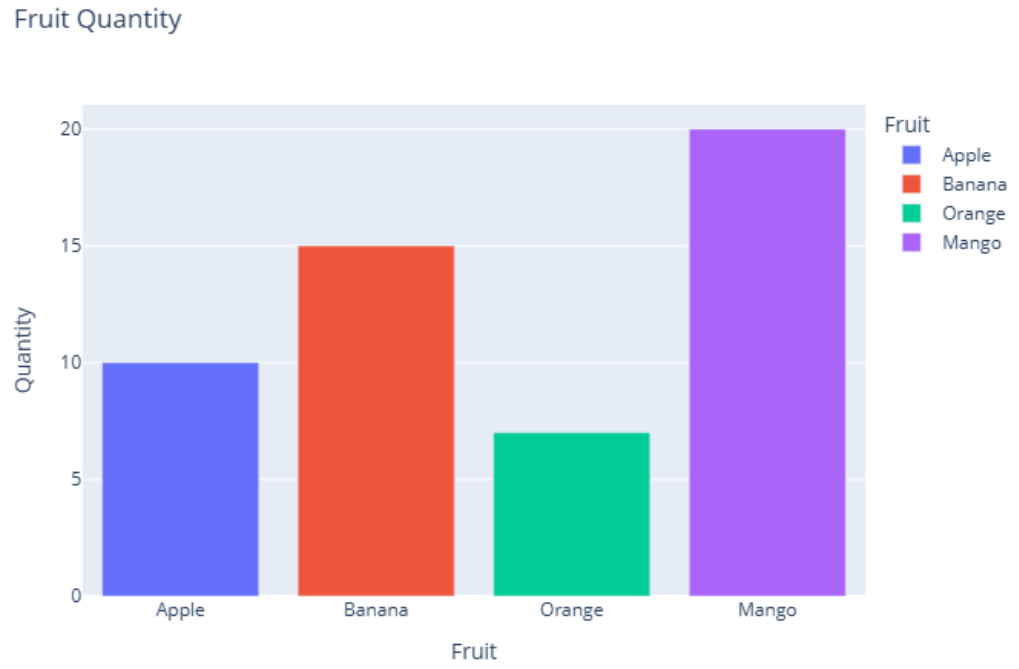
DR. SUBRAMANI
AI Computational Biologist

Plotly

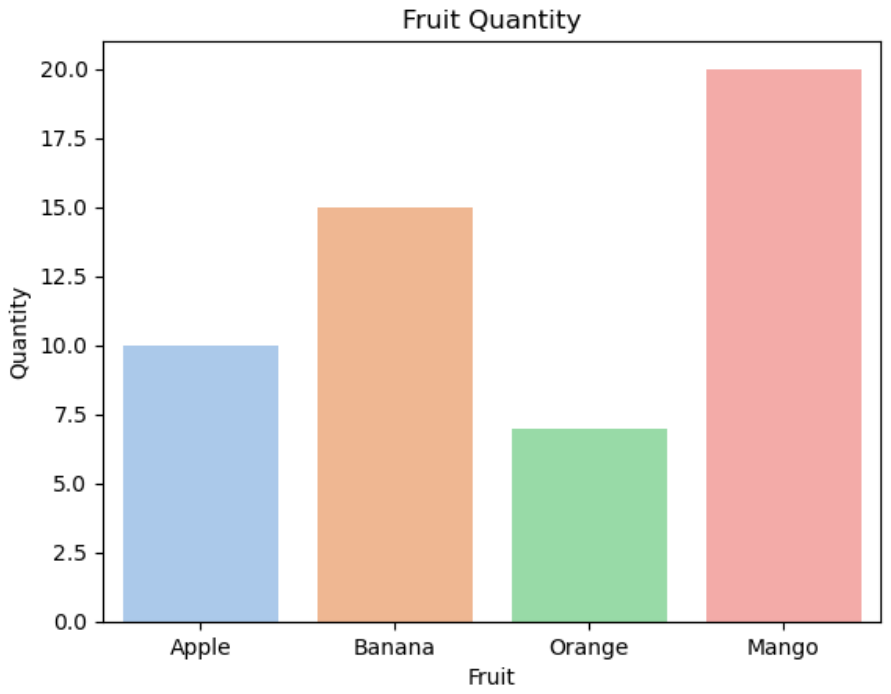
3 Bar Chart

Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_bar = pd.DataFrame({
    'Fruit': ['Apple', 'Banana', 'Orange', 'Mango'],
    'Quantity': [10, 15, 7, 20]
})
fig = px.bar(df_bar, x='Fruit', y='Quantity', color='Fruit', title='Fruit Quantity')
fig.show()
```




```python
import pandas as pd
import seaborn as sns
# Sample dataset
df_bar = pd.DataFrame({
    'Fruit': ['Apple', 'Banana', 'Orange', 'Mango'],
    'Quantity': [10, 15, 7, 20]
})
# Simple bar chart
sns.barplot(data=df_bar, x='Fruit', y='Quantity', palette='pastel').set_title('Fruit Quantity')
```



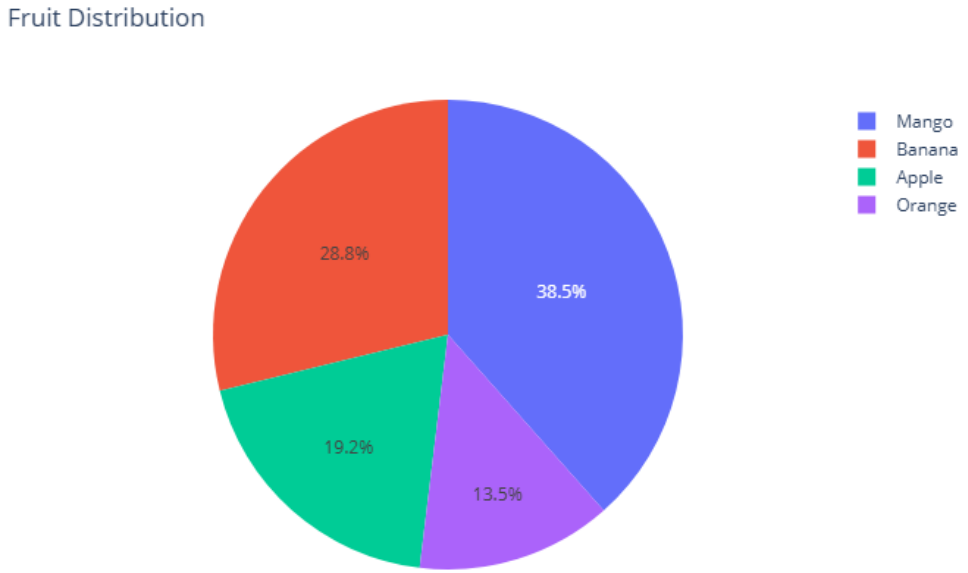
DR. SUBRAMANI
AI Computational Biologist

Plotly

4 Pie Chart

Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_bar = pd.DataFrame({
    'Fruit': ['Apple', 'Banana', 'Orange', 'Mango'],
    'Quantity': [10, 15, 7, 20]
})
fig = px.pie(df_bar, names='Fruit', values='Quantity', title='Fruit Distribution')
fig.show()
```



Seaborn does *not* have a built-in function.

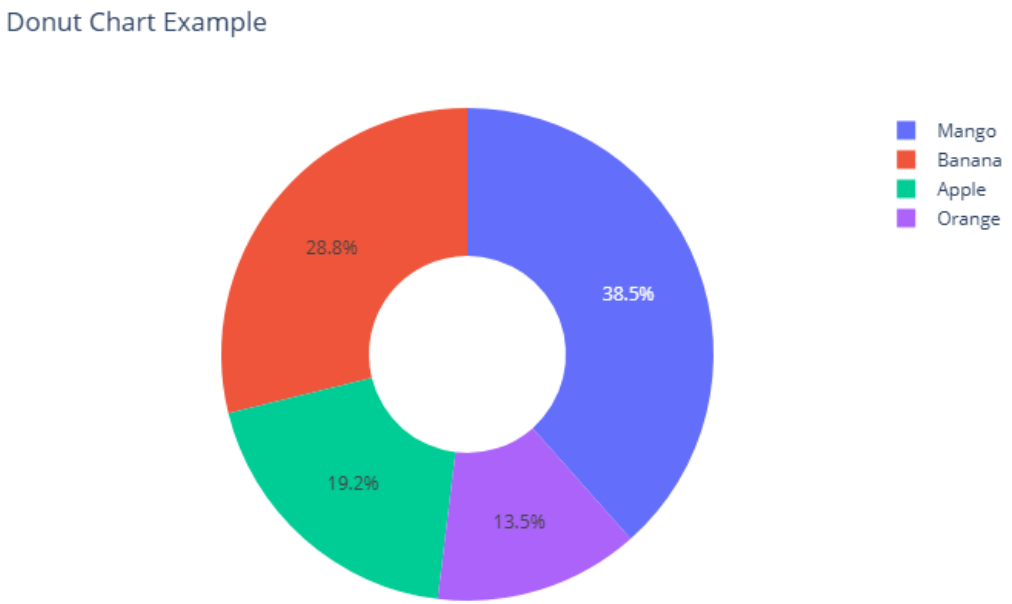
DR. SUBRAMANI
AI Computational Biologist

Plotly

5 Donut Chart

Seaborn

```
import plotly.express as px
import pandas as pd
# Sample dataset
df_bar = pd.DataFrame({
   'Fruit': ['Apple', 'Banana', 'Orange', 'Mango'],
   'Quantity': [10, 15, 7, 20]
})
fig = px.pie(df_bar, names='Fruit', values='Quantity', hole=0.4, title='Donut Chart Example')
fig.show()
```



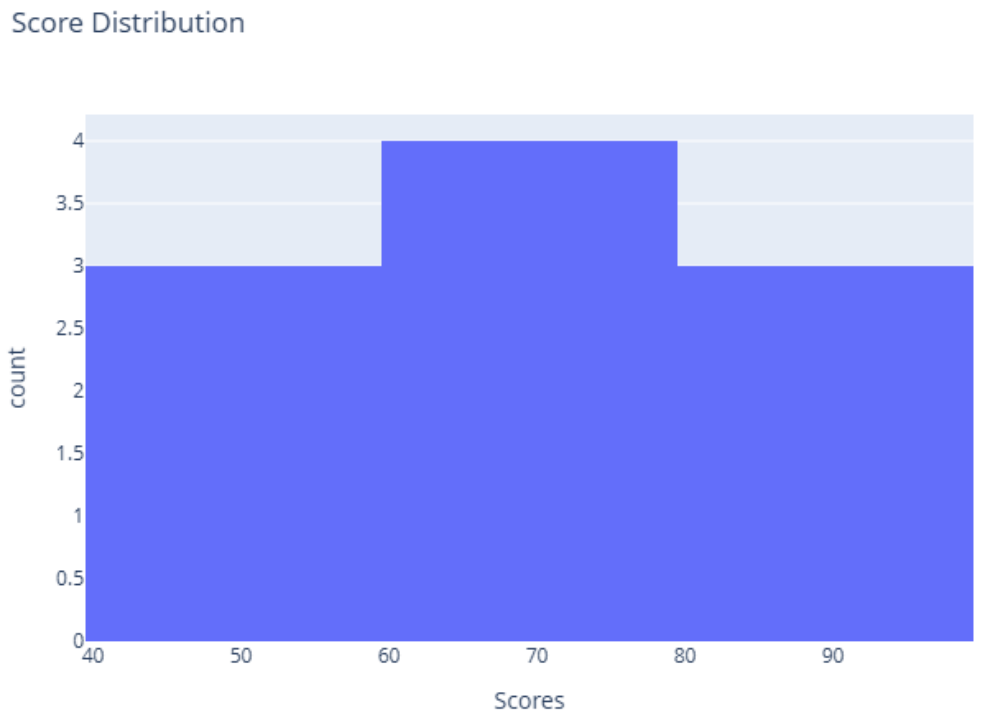
Seaborn does *not* have a built-in function.

DR. SUBRAMANI
AI Computational Biologist

6 Histogram

Plotly

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_hist = pd.DataFrame({'Scores': [45, 55, 65, 75, 85, 95, 55, 65, 75, 85]})
fig = px.histogram(df_hist, x='Scores', nbins=5, title='Score Distribution')
fig.show()
```

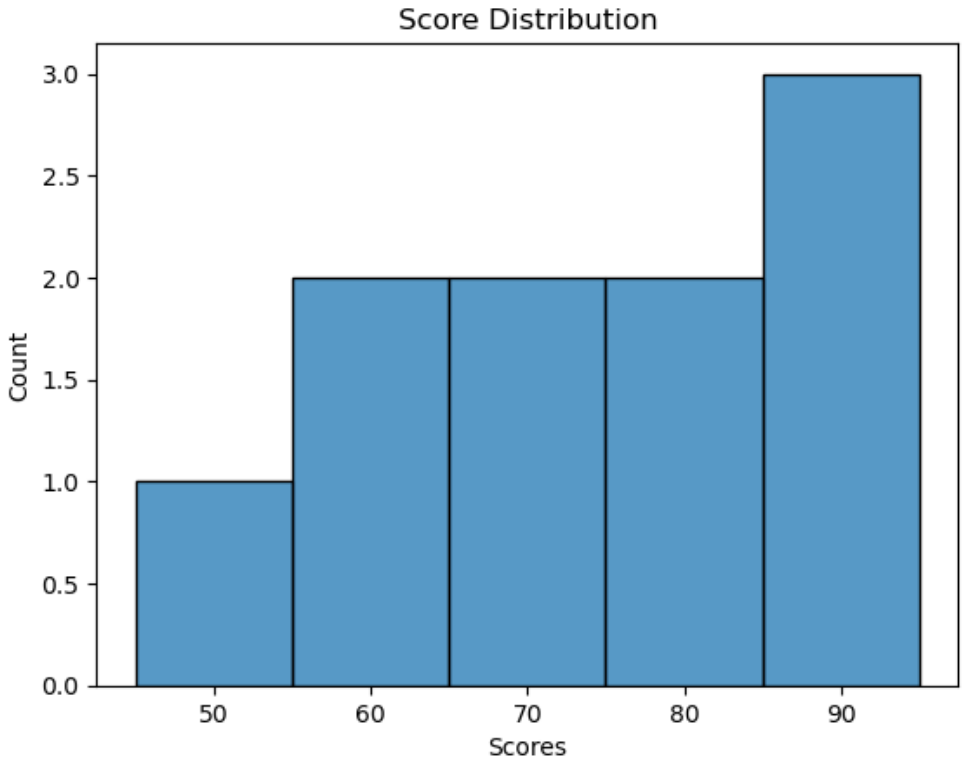


Seaborn

```python
import pandas as pd
import seaborn as sns

# Sample dataset
df_hist = pd.DataFrame({'Scores': [45, 55, 65, 75, 85, 95, 55, 65, 75, 85]})

# Histogram
sns.histplot(data=df_hist, x='Scores', bins=5, kde=False).set_title('Score Distribution')
```

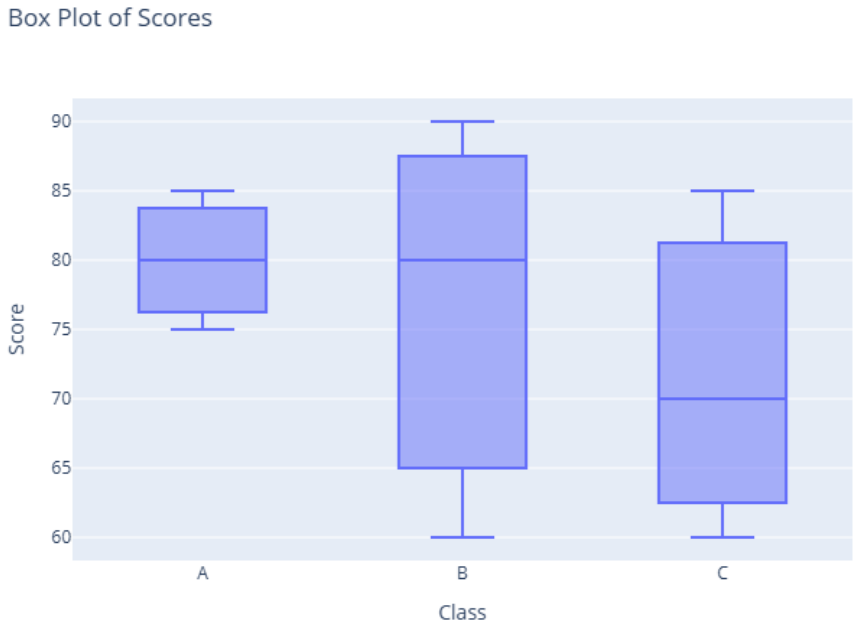


DR. SUBRAMANI
AI Computational Biologist

7 Box Plot

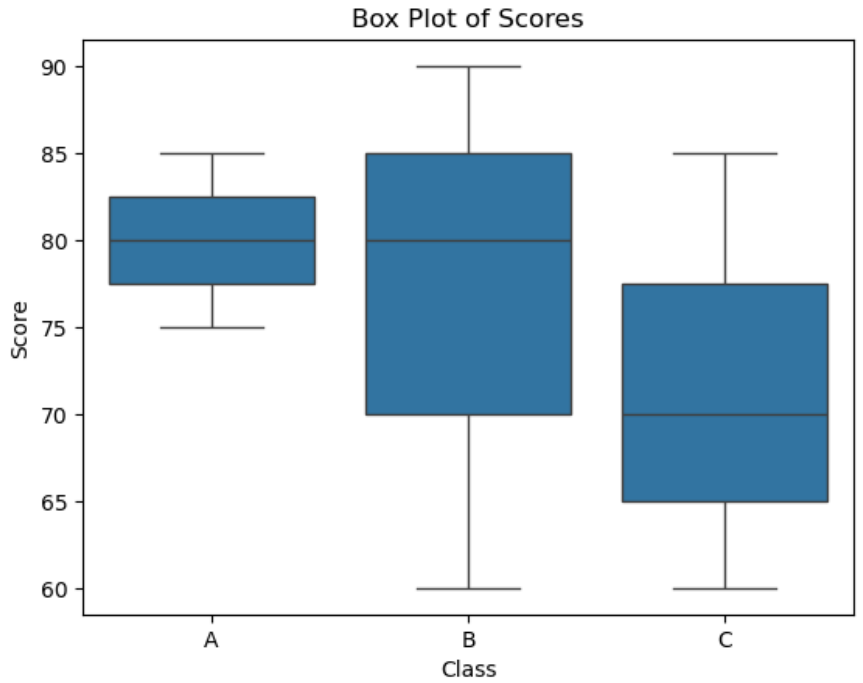
Plotly

```python
import plotly.express as px
import pandas as pd
# Simple dataset
df_simple = pd.DataFrame({
    'Class': ['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C'],
    'Score': [75, 85, 80, 90, 60, 80, 60, 70, 85]
})
fig = px.box(df_simple, x='Class', y='Score', title='Box Plot of Scores')
fig.show()
```



Seaborn

```python
import pandas as pd
import seaborn as sns
# Sample dataset
df_simple = pd.DataFrame({
    'Class': ['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C'],
    'Score': [75, 85, 80, 90, 60, 80, 60, 70, 85]
})
# Box plot
sns.boxplot(data=df_simple, x='Class', y='Score').set_title('Box Plot of Scores')
```



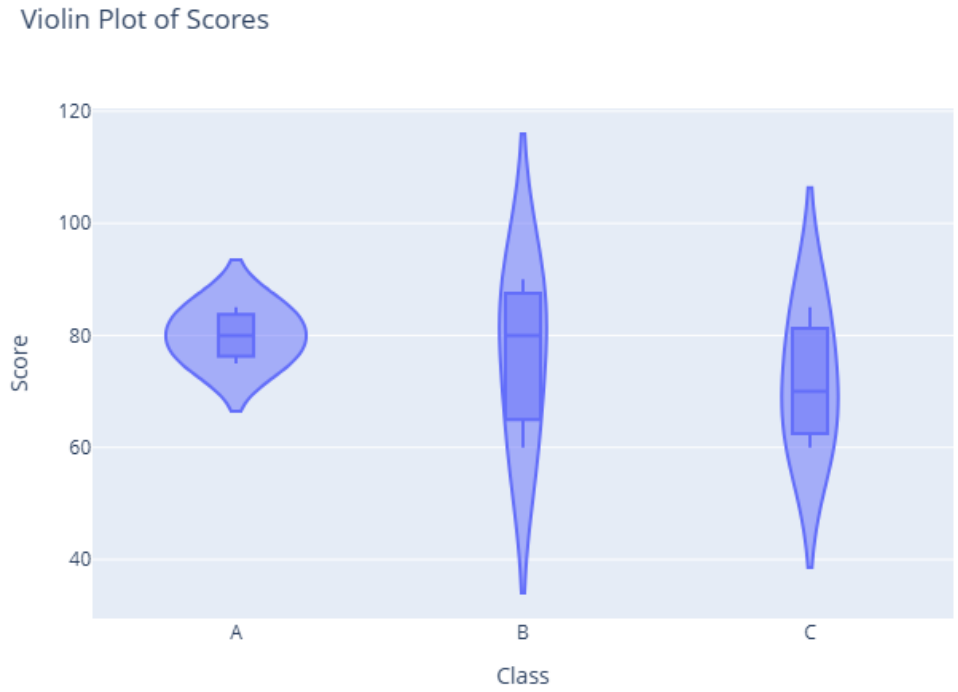
DR. SUBRAMANI
AI Computational Biologist

Plotly

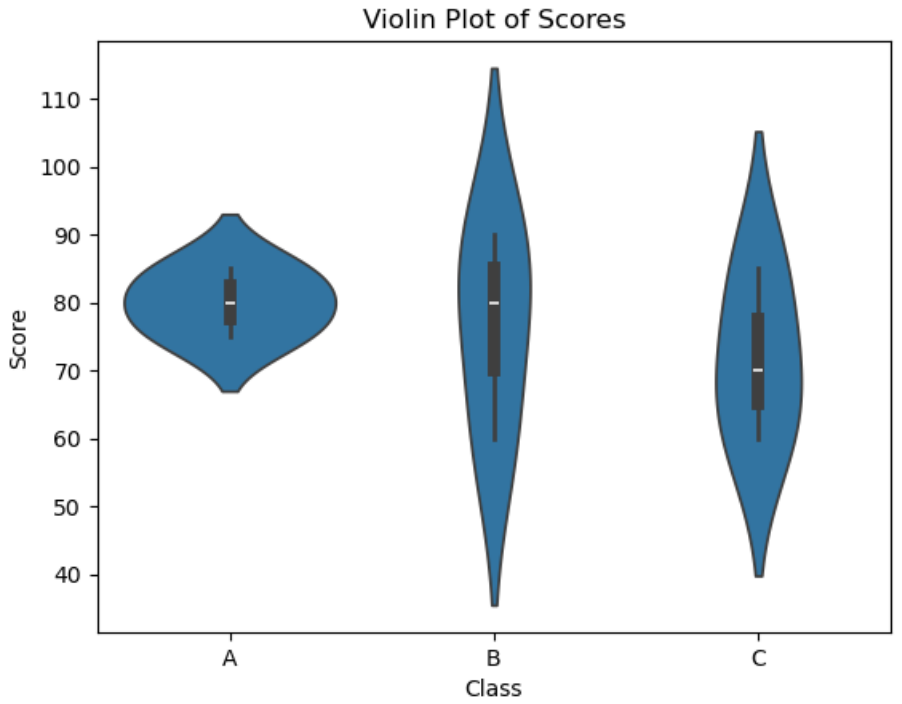
8 Violin Plot

Seaborn

```python
# Simple dataset
df_simple = pd.DataFrame({
    'Class': ['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C'],
    'Score': [75, 85, 80, 90, 60, 80, 60, 70, 85]
})
fig = px.violin(df_simple, x='Class', y='Score', box=True, title='Violin Plot of Scores')
fig.show()
```




```python
import pandas as pd
import seaborn as sns
# Sample dataset
df_simple = pd.DataFrame({
    'Class': ['A', 'A', 'A', 'B', 'B', 'B', 'C', 'C', 'C'],
    'Score': [75, 85, 80, 90, 60, 80, 60, 70, 85]
})
# Violin plot
sns.violinplot(data=df_simple, x='Class', y='Score', inner='box').set_title('Violin Plot of Scores')
```



DR. SUBRAMANI
AI Computational Biologist

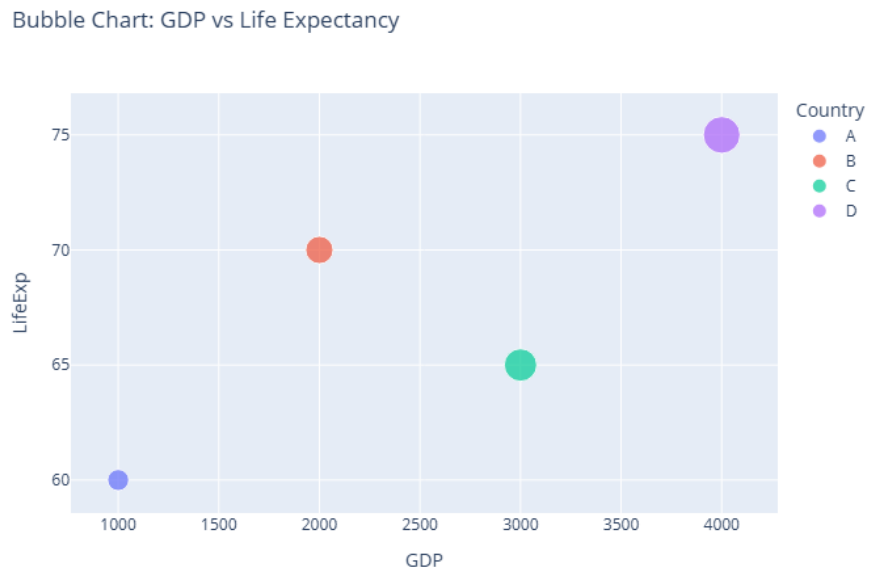
Plotly

9 Bubble Chart

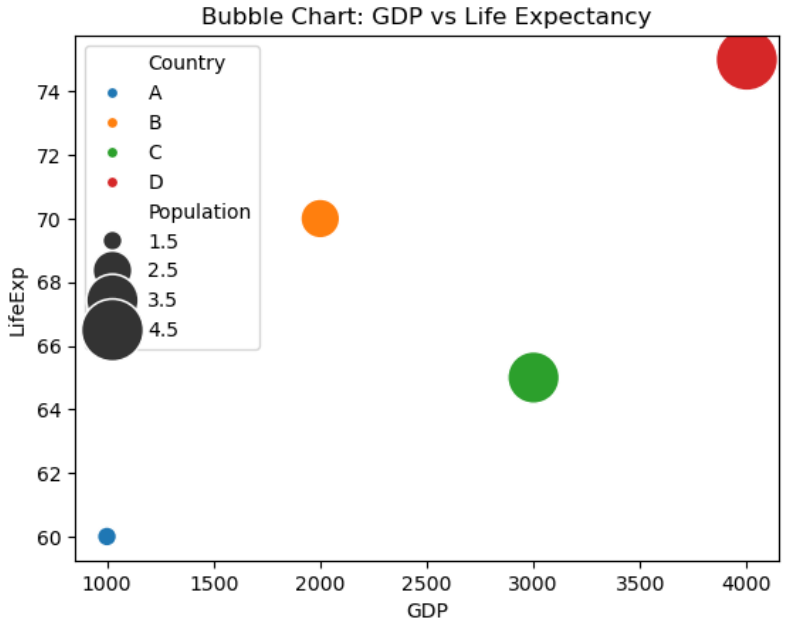
Seaborn

```
import plotly.express as px
import pandas as pd
# Sample dataset
df_bubble = pd.DataFrame({
   'Country': ['A', 'B', 'C', 'D'],
   'GDP': [1000, 2000, 3000, 4000],
   'LifeExp': [60, 70, 65, 75],
   'Population': [1.5, 2.5, 3.5, 4.5]
})

fig = px.scatter(df_bubble, x='GDP', y='LifeExp', size='Population', color='Country',
           hover_name='Country', title='Bubble Chart: GDP vs Life Expectancy')
fig.show()
```




```
import pandas as pd
import seaborn as sns
# Sample dataset
df_bubble = pd.DataFrame({
   'Country': ['A', 'B', 'C', 'D'],
   'GDP': [1000, 2000, 3000, 4000],
   'LifeExp': [60, 70, 65, 75],
   'Population': [1.5, 2.5, 3.5, 4.5]
})
# Bubble chart
sns.scatterplot(
   data=df_bubble, x='GDP', y='LifeExp', size='Population', hue='Country',
 sizes=(100, 1000),  legend='full').set_title('Bubble Chart: GDP vs Life Expectancy')
```



DR. SUBRAMANI
AI Computational Biologist

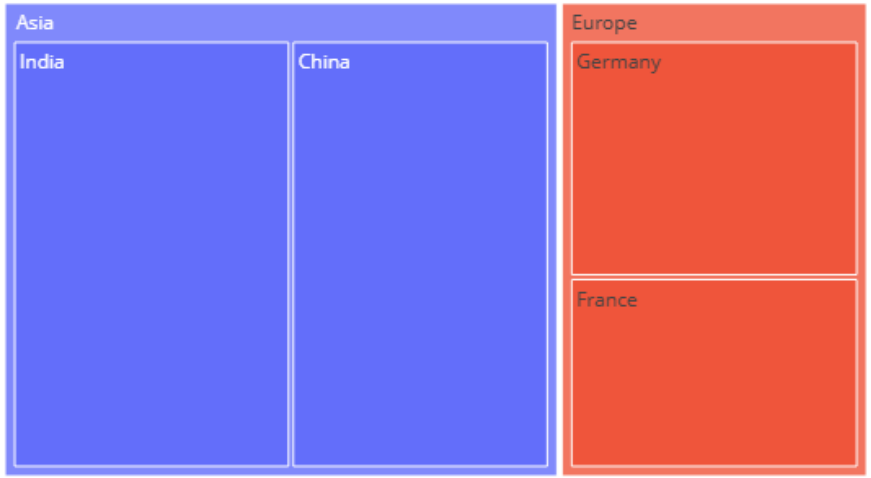
Plotly

10 Treemap

Seaborn

```
import plotly.express as px
import pandas as pd
# Sample dataset
df_tree = pd.DataFrame({
    'Continent': ['Asia','Asia','Europe','Europe'],
    'Country': ['India','China','France','Germany'],
    'Population': [140, 130, 67, 83]
})
fig = px.treemap(df_tree, path=['Continent','Country'], values='Population',
                 title='Treemap of Population by Continent')
fig.show()
```

Treemap of Population by Continent

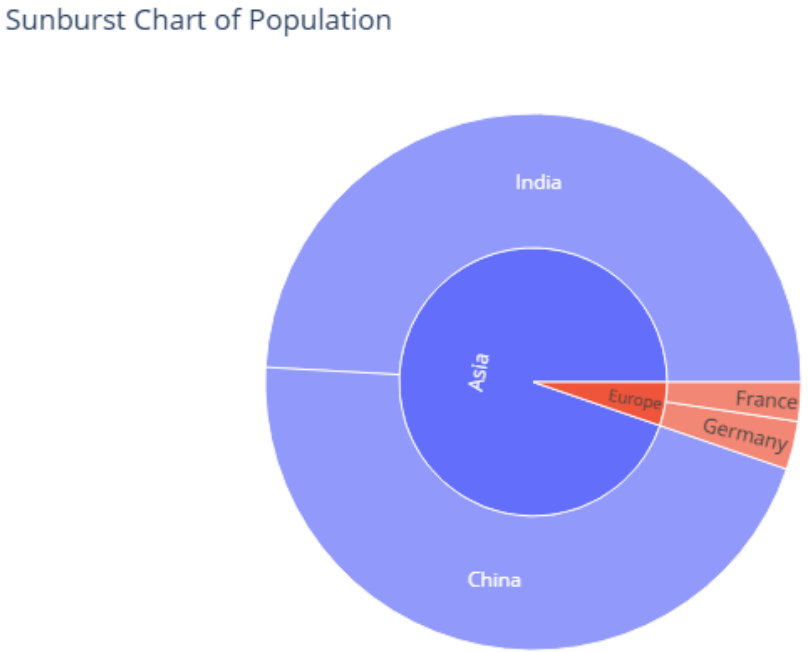


Seaborn does *not* have a built-in function.

DR. SUBRAMANI
AI Computational Biologist

Plotly 11 Sunburst Chart Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_tree = pd.DataFrame({
    'Continent': ['Asia','Asia','Europe','Europe'],
    'Country': ['India','China','France','Germany'],
    'Population': [1400, 1300, 67, 83]
})
fig = px.sunburst(df_tree, path=['Continent','Country'], values='Population',
                  title='Sunburst Chart of Population')
fig.show()
```



Seaborn does *not* have a built-in function.

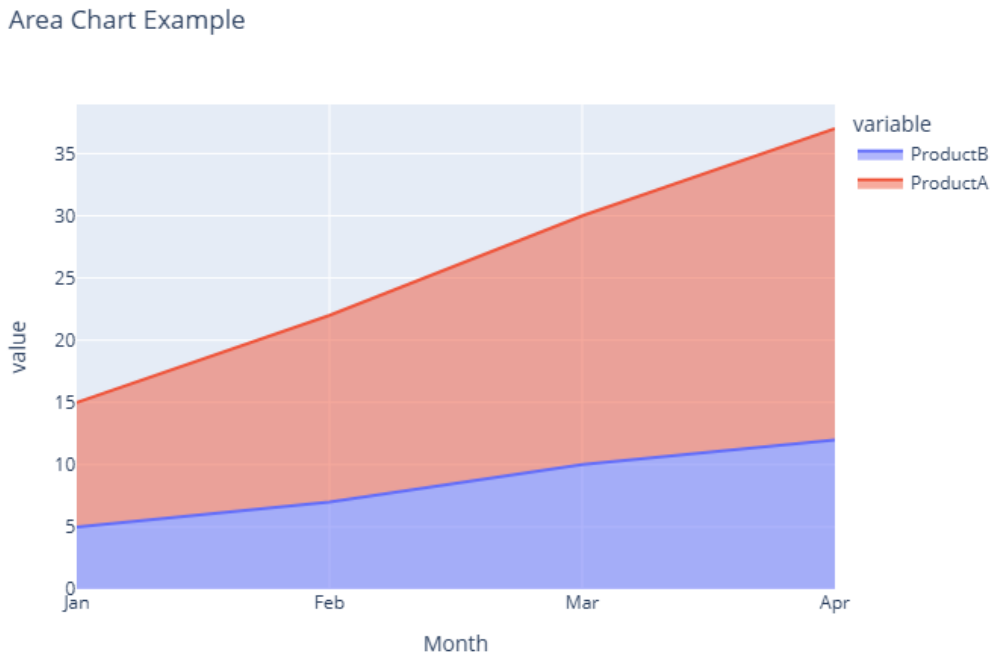
DR. SUBRAMANI
AI Computational Biologist

Plotly

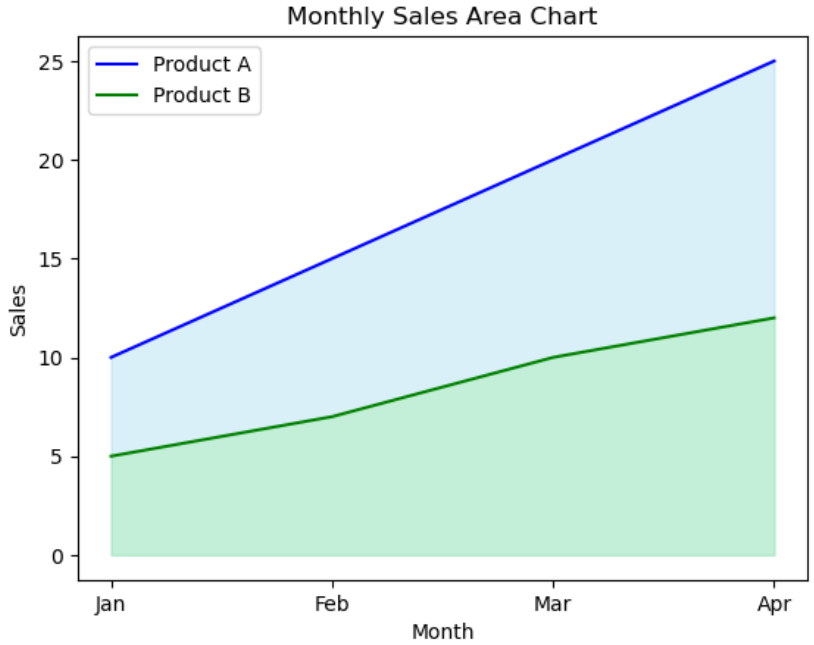
1 2 Area Chart

Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_area = pd.DataFrame({
   'Month': ['Jan','Feb','Mar','Apr'],
   'ProductA': [10, 15, 20, 25],
   'ProductB': [5, 7, 10, 12]
})
fig = px.area(df_area, x='Month', y=['ProductB','ProductA'], title='Area Chart Example')
fig.show()
```




```python
import seaborn as sns
import matplotlib.pyplot as plt
# Sample dataset
df_area = pd.DataFrame({
   'Month': ['Jan','Feb','Mar','Apr'], 'ProductA': [10, 15, 20, 25], 'ProductB': [5, 7, 10, 12]
})
# Plot Product A
sns.lineplot(data=df_area, x='Month', y='ProductA', label='Product A', color='blue')
plt.fill_between(df_area['Month'], df_area['ProductA'], alpha=0.3, color='skyblue')
# Plot Product B
sns.lineplot(data=df_area, x='Month', y='ProductB', label='Product B', color='green')
plt.fill_between(df_area['Month'], df_area['ProductB'], alpha=0.3, color='lightgreen')
plt.title('Monthly Sales Area Chart')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.legend()
plt.show()
```



DR. SUBRAMANI
AI Computational Biologist

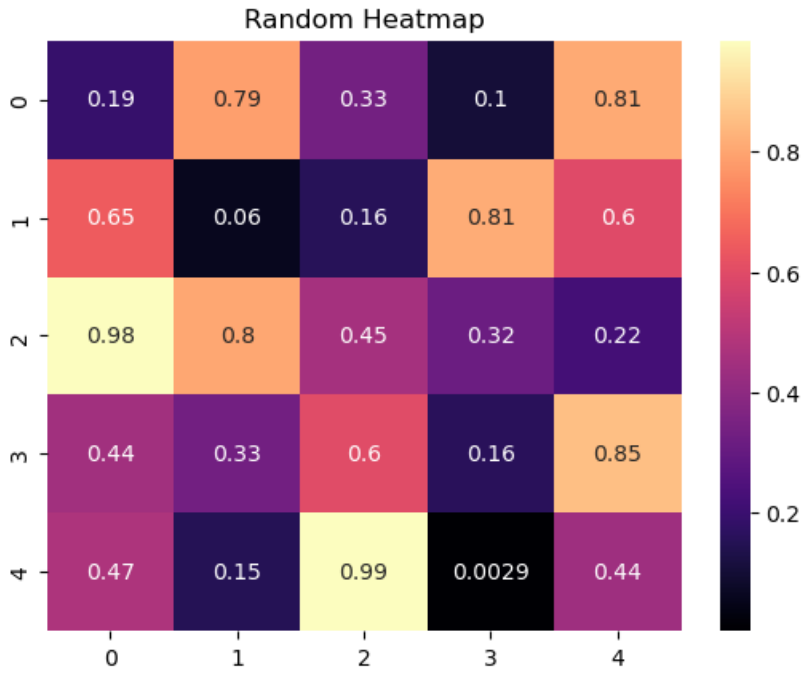
Plotly

13 Heatmap

Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
data = np.random.rand(5,5)
fig = px.imshow(data, text_auto=True, title='Random Heatmap')
fig.show()
```

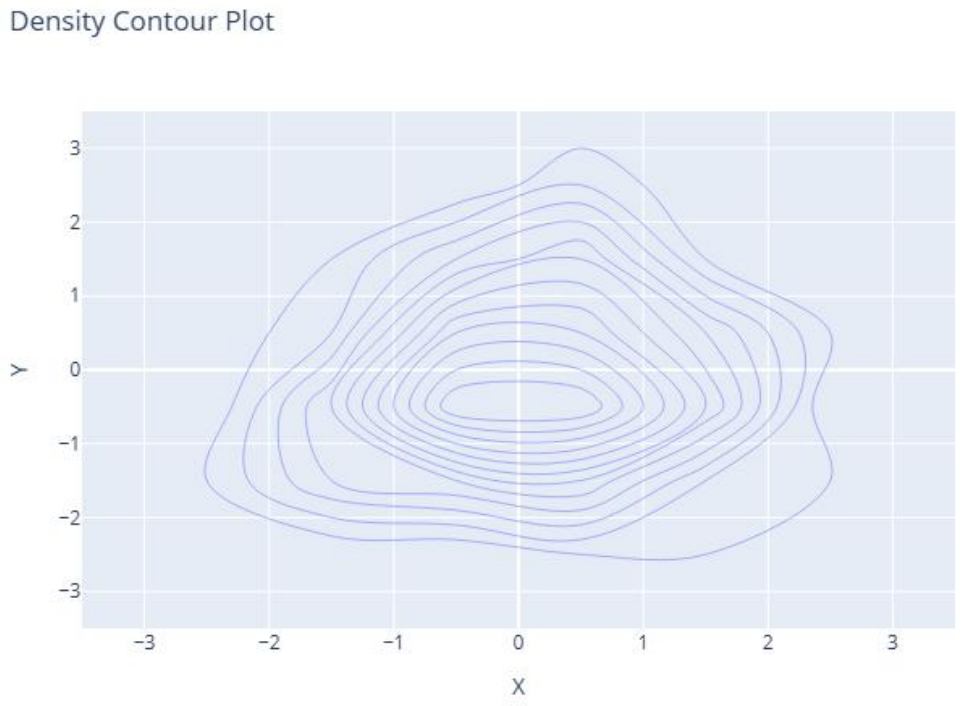

```python
import numpy as np
import seaborn as sns
# Sample data
data = np.random.rand(5,5)
# Heatmap
sns.heatmap(data, annot=True, cmap='magma').set_title('Random Heatmap')
```



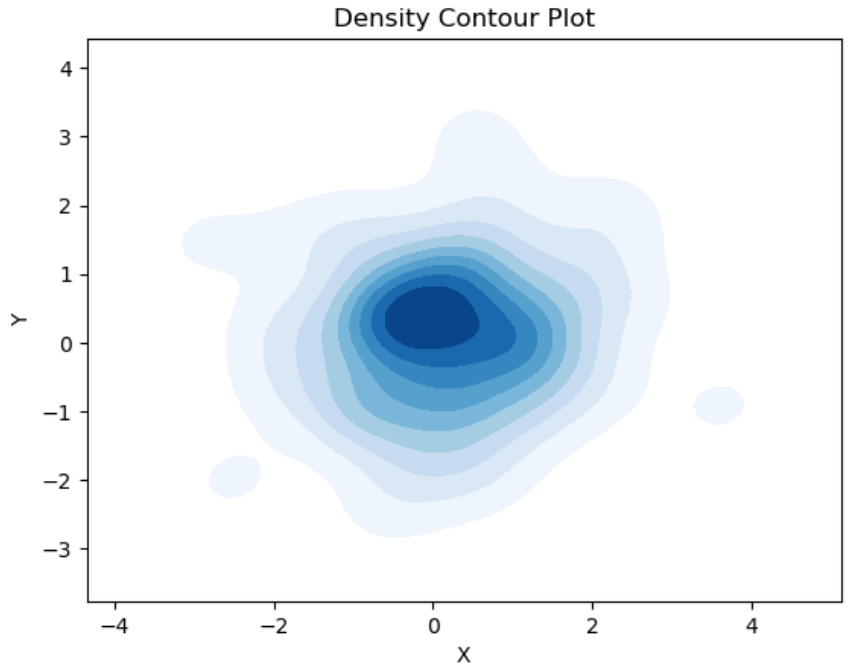
DR. SUBRAMANI
AI Computational Biologist

Plotly 14 Density Contour Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_density = pd.DataFrame({
    'X': np.random.randn(100),
    'Y': np.random.randn(100)
})
fig = px.density_contour(df_density, x='X', y='Y',  title='Density Contour Plot')
fig.show()
```




```python
import pandas as pd
import numpy as np
import seaborn as sns
# Sample dataset
df_density = pd.DataFrame({
    'X': np.random.randn(100),
    'Y': np.random.randn(100)
})
# Density contour plot
sns.kdeplot(data=df_density, x='X', y='Y', fill=True, cmap='Blues').set_title('Density Contour Plot')
```



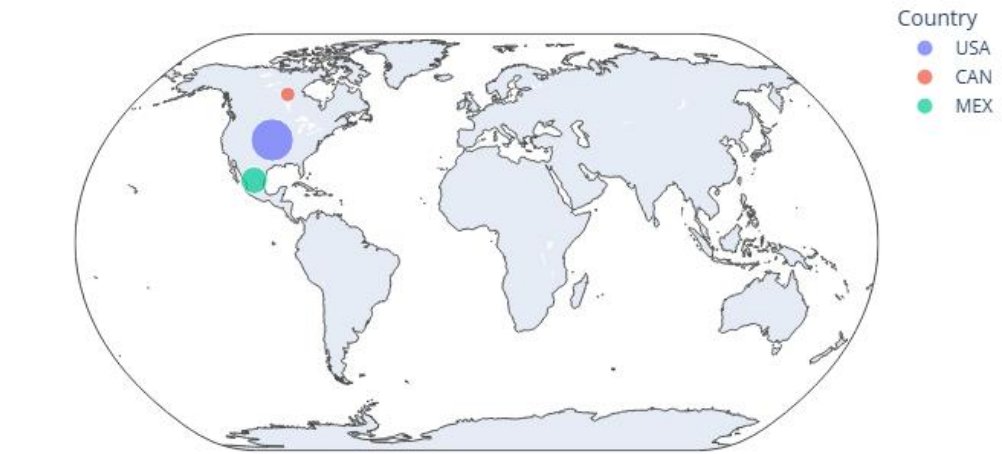
DR. SUBRAMANI
AI Computational Biologist

Plotly

15 Geo Map

Seaborn

```python
import plotly.express as px
import pandas as pd
# Sample dataset
df_map = pd.DataFrame({
   'Country': ['USA','CAN','MEX'],
   'ISO': ['USA','CAN','MEX'],
   'Population': [300, 37, 120]
})
fig = px.scatter_geo(df_map, locations='ISO', color='Country', size='Population',
                     projection='natural earth', title='Sample Map')
fig.show()
```

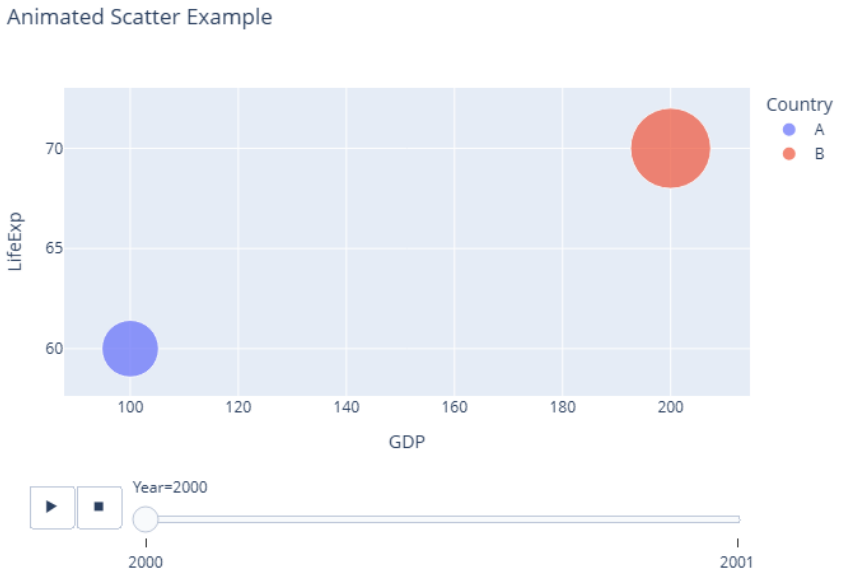


Seaborn does *not* have a built-in function.

DR. SUBRAMANI
AI Computational Biologist

Plotly 16 Animated Scatter Seaborn

```
import plotly.express as px
import pandas as pd
# Sample dataset
df_anim = pd.DataFrame({
   'Year': [2000,2000,2001,2001],
   'Country': ['A','B','A','B'],
   'GDP': [100,200,150,250],
   'LifeExp':[60,70,62,72]
})
fig = px.scatter(df_anim, x='GDP', y='LifeExp', animation_frame='Year', animation_group='Country',
            size='GDP', color='Country', hover_name='Country', log_x=False, size_max=50,
            title='Animated Scatter Example')
fig.show()
```



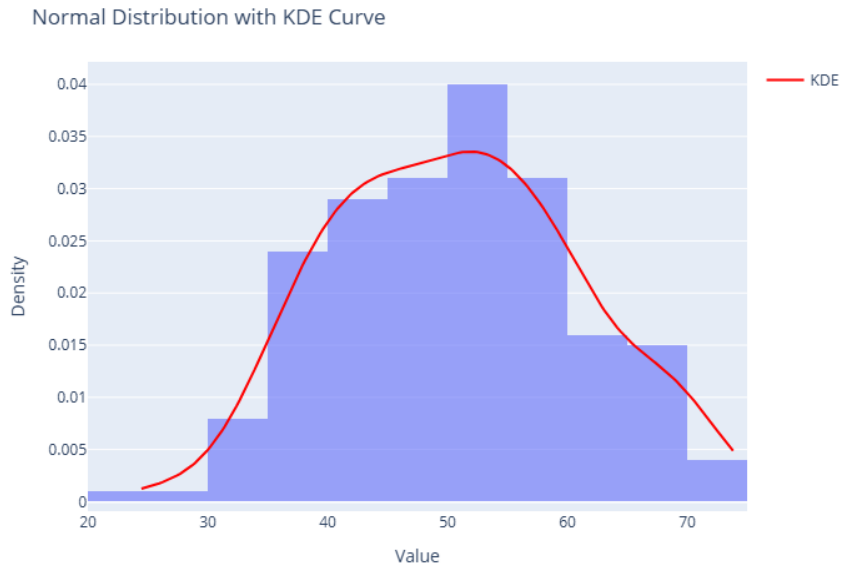
Seaborn does *not* have a built-in function.

DR. SUBRAMANI
AI Computational Biologist

17 Normal Distribution with KDE Curve

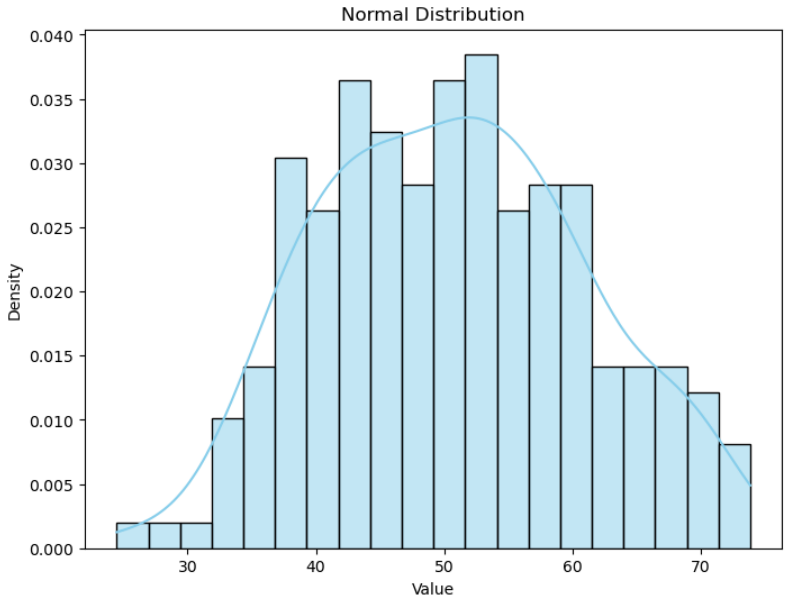
Plotly

```python
import numpy as np
import pandas as pd
import plotly.express as px
from scipy.stats import gaussian_kde
# Sample normal distribution data
np.random.seed(0)
data = np.random.normal(loc=50, scale=10, size=200)
df = pd.DataFrame({'Value': data})
# Histogram
fig = px.histogram(df, x='Value', nbins=20, histnorm='probability density', opacity=0.6)
# Compute KDE
kde = gaussian_kde(data)
x_vals = np.linspace(min(data), max(data), 200)
y_vals = kde(x_vals)
# Add KDE curve
fig.add_scatter(x=x_vals, y=y_vals, mode='lines', name='KDE', line=dict(color='red'))
fig.update_layout(title='Normal Distribution with KDE Curve', xaxis_title='Value', yaxis_title='Density')
fig.show()
```



Seaborn

```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
# Generate sample normal distribution data
np.random.seed(0)
data = np.random.normal(loc=50, scale=10, size=200)  # mean=50, std=10, 200 points
df = pd.DataFrame({'Value': data})
# Plot histogram with KDE
plt.figure(figsize=(8,6))
sns.histplot(df['Value'], bins=20, kde=True, color='skyblue', stat='density')
# stat='density' makes histogram comparable to KDE
plt.title('Normal Distribution')
plt.xlabel('Value')
plt.ylabel('Density')
plt.show()
```



DR. SUBRAMANI
AI Computational Biologist

Supported Charts

Chart Type	Seaborn	Plotly
Scatter	✅	✅ (interactive)
Line	✅	✅ (interactive)
Bar	✅	✅
Box	✅	✅
Violin	✅	✅
Histogram	✅	✅
Heatmap	✅	✅
Density contour	✅	✅
Pie / Donut	❌	✅
Treemap	❌	✅
Sunburst	❌	✅
Bubble	✅ (static)	✅ (interactive)
Geo Map	❌	✅
Animated scatter	❌	✅
Area chart	✅ (static)	✅ (interactive)

🖼️ Plotly Renderers — Quick Reference

```
import plotly.express as px
import plotly.io  as  pio

pio.renderers.default = "svg"
```

Renderer Name	Environment / Use Case	Description
"notebook"	Jupyter Notebook	Displays **interactive** Plotly charts inline (default in Jupyter).
"browser"	Any Python environment	Opens the chart in a **new browser tab** for full interactivity.
"svg"	Static export	Renders a **static vector image** — useful for documents, reports, or PPTs.
"png"	Static export	Shows a **static raster image** (no interactivity, good for quick snapshots).
"iframe"	Embedded HTML	Embeds the chart in an **interactive HTML frame** inside notebooks or web apps.
"colab"	Google Colab	Optimized renderer for **Colab notebooks**.

When to Use Each

Seaborn:
•Quick EDA (Exploratory Data Analysis)
•Static reports, papers, or slides
•Needs clean, simple charts

Plotly:
•Dashboards & web apps
•Interactive presentations
•Animations or geographic visualizations

💡 **Summary:**
•Use **Seaborn** for **speed + simplicity + static analysis**.
•Use **Plotly** for **interaction + presentation + animation**

DR. SUBRAMANI
AI Computational Biologist

Thank You

DR SUBRAMANI