

## ✓ Experiment No 5

### Implement Area plot on Data

```
from google.colab import drive
drive.mount('/content/drive', force_remount=True)
```

```
Mounted at /content/drive
```

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

```
file_path = '/content/drive/MyDrive/Colab Notebooks/LY DV Practicles/Salary_Data.csv'
```

```
# Read data from CSV file
data = pd.read_csv(file_path)
```

```
# Extract date/month for x-axis
x = data['YearsExperience']
```

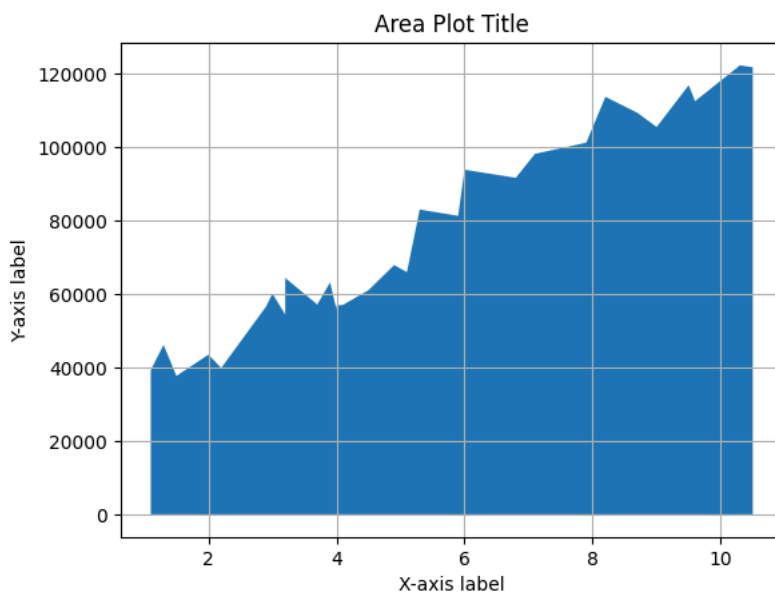
```
# Extract values for y-axis
y = data['Salary']
```

```
# Create the area plot
plt.fill_between(x, y)
```

```
# Customize the plot
plt.xlabel("X-axis label") # Replace with your desired label
plt.ylabel("Y-axis label") # Replace with your desired label
plt.title("Area Plot Title") # Replace with your desired title
```

```
# Optional: Set gridlines
plt.grid(True)
```

```
# Show the plot
plt.show()
```



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

```
file_path1 = '/content/drive/MyDrive/Colab Notebooks/LY DV Practicles/Car_Info.csv'
```

```
# Sample data
data1 = pd.read_csv(file_path1)
```

```
# Extract date/month for x-axis
x = data1['Year']
```

```
# Extract values for y-axis
y1 = data1['Engine HP']
y2 = data1['Popularity']
```

```

yz = data['popularity']

# Create a basic line chart
fig = px.line(x=x, y=y1, line_shape='linear',
              labels={'y': 'Y-axis'})

# Update the trace to fill the area between the line and the axis
fig.update_traces(fill='tozeroy', line_color='rgba(0,100,80,0.2)')

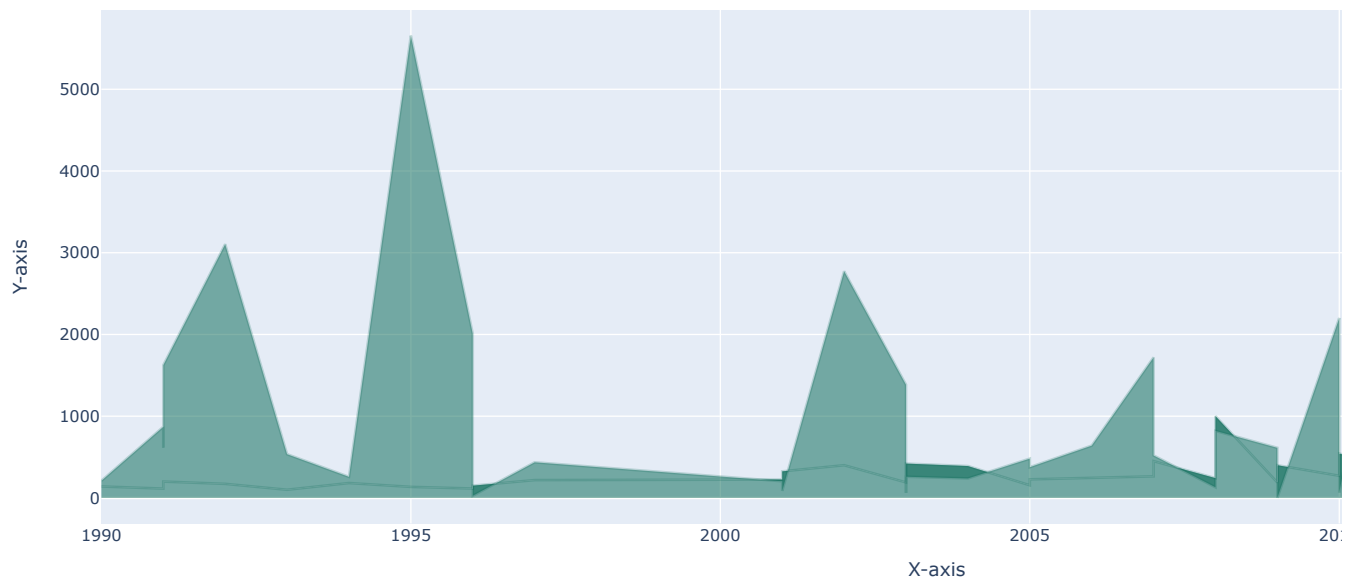
# Add another line trace
fig.add_trace(px.line(x=x, y=y2, line_shape='linear').update_traces(fill='tonexty', line_color='rgba(0,100,80,0.2)').data[0])

# Update layout
fig.update_layout(title='Customized Filled Area Chart', xaxis_title='X-axis')

# Show the plot
fig.show()

```

Customized Filled Area Chart



```

import plotly.graph_objects as go

# Sample data
x = [1, 2, 3, 4, 5]
y1 = [10, 15, 13, 17, 18]
y2 = [16, 5, 11, 9, 7]

# Create traces
trace1 = go.Scatter(x=x, y=y1, mode='lines', name='Trace 1', fill='tozeroy')
trace2 = go.Scatter(x=x, y=y2, mode='lines', name='Trace 2', fill='tonexty')

# Create figure
fig = go.Figure([trace1, trace2])

# Update layout with legend properties
fig.update_layout(
    legend=dict(
        title='Legend Title',
        x=0.85,
        y=0.95,
        traceorder='normal',
        font=dict(
            family='Arial, sans-serif',
            size=12,
            color='black'
        ),
        bgcolor='LightSteelBlue',
        bordercolor='Black',
        borderwidth=2
    )
)

# Show figure
fig.show()

```

