Creating Interactive Visualizations with Plotly

Plotly is a powerful library that allows you to create interactive plots for web applications, reports, and data analysis. It offers a variety of chart types, such as scatter plots, line plots, bar charts, pie charts, and more. These visualizations can be embedded in dashboards or web pages for a rich, interactive experience.

---

1. Install Plotly

First, you need to install Plotly:

bash
Copy code
```
pip install plotly
```

---

2. Importing the Plotly Library

python
Copy code
```
import plotly.express as px
import plotly.graph_objects as go
```

- **Plotly Express** (`px`) is a high-level API that makes it easier to create complex plots with minimal code.
- **Plotly Graph Objects** (`go`) provides a low-level API for creating custom and flexible plots.

---

3. Basic Interactive Plots with Plotly Express

Scatter Plot

python
Copy code
```
import plotly.express as px

# Create a sample DataFrame
df = px.data.iris()  # Built-in Iris dataset

# Scatter plot with hover interaction
```

```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species', title='Sepal Width vs Sepal Length')
fig.show()
```

- **Hover effects**: Hovering over data points reveals additional information like the values of the columns and the name of the species in this case.

Line Plot

python
Copy code

```
# Line plot to visualize trends
fig = px.line(df, x='sepal_width', y='sepal_length', title='Sepal
Width vs Sepal Length (Line Plot)')
fig.show()
```

Bar Chart

python
Copy code

```
# Bar chart showing the count of each species
fig = px.bar(df, x='species', title='Species Count')
fig.show()
```

Histogram

python
Copy code

```
# Histogram showing the distribution of sepal length
fig = px.histogram(df, x='sepal_length', nbins=20,
title='Distribution of Sepal Length')
fig.show()
```

Pie Chart

python
Copy code

```
# Pie chart to visualize the proportion of species
fig = px.pie(df, names='species', title='Species Distribution')
fig.show()
```

---

4. Customizing the Plot

You can customize the layout, colors, axis labels, titles, and more using Plotly's extensive options.

Add Titles and Customize Axes

python
Copy code

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

# Customize the layout
fig.update_layout(
    title='Sepal Width vs Sepal Length',
    xaxis_title='Sepal Width (cm)',
    yaxis_title='Sepal Length (cm)',
    template='plotly_dark'  # Use a dark theme
)

fig.show()
```

Change Marker Styles

python
Copy code

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

# Customize marker size and opacity
fig.update_traces(marker=dict(size=12, opacity=0.8,
line=dict(width=2, color='DarkSlateGrey')))
fig.show()
```

---

5. 3D Plotting

Plotly also allows you to create 3D visualizations, which can be useful for multivariate data.

3D Scatter Plot

python
Copy code

```
# 3D Scatter plot
fig = px.scatter_3d(df, x='sepal_width', y='sepal_length',
z='petal_length', color='species', title='3D Scatter Plot')
fig.show()
```

3D Surface Plot

python
Copy code

```
# Create a 3D surface plot (example with a surface grid)
import numpy as np

x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))

fig = go.Figure(data=[go.Surface(z=z, x=x, y=y)])
fig.update_layout(title='3D Surface Plot', scene=dict(
    xaxis_title='X',
    yaxis_title='Y',
    zaxis_title='Z'))
fig.show()
```

---

6. Subplots

You can also create subplots (multiple plots in a single figure) using `make_subplots`.

Multiple Subplots

python
Copy code

```
from plotly.subplots import make_subplots
import plotly.graph_objects as go

# Create subplots
fig = make_subplots(rows=1, cols=2)

# Add scatter plot to the first subplot
fig.add_trace(go.Scatter(x=df['sepal_width'], y=df['sepal_length'],
mode='markers', name='Sepal'), row=1, col=1)

# Add histogram to the second subplot
fig.add_trace(go.Histogram(x=df['sepal_length'], nbinsx=20,
name='Sepal Length'), row=1, col=2)
```

```
fig.update_layout(title='Multiple Subplots')
fig.show()
```

---

7. Interactivity (Hover, Click, Zoom)

Plotly plots are highly interactive by default. You can further customize the interactivity by using event handling and callback functions.

Interactive Hover

Plotly provides detailed hover information, including additional data about the point or region being hovered over.

python
Copy code
```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species')

# Customize hover data
fig.update_traces(hovertemplate='<b>Species:</b> %{text}<br><b>Sepal
Length:</b> %{y}<br><b>Sepal Width:</b> %{x}', text=df['species'])
fig.show()
```

Zooming and Panning

Zooming and panning are built into Plotly. You can interactively zoom in or out and pan across the plot with your mouse. These actions can also be controlled programmatically.

python
Copy code
```
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species')

# Disable zoom
fig.update_layout(dragmode='pan')  # Allows panning but disables
zooming

fig.show()
```

---

8. Creating Dashboards with Plotly and Dash

Plotly can be used with the Dash framework to create interactive dashboards. Dash allows you to build web-based applications using Python.

Install Dash

bash
Copy code

```
pip install dash
```

Basic Dash App Example

python
Copy code

```
import dash
import dash_core_components as dcc
import dash_html_components as html
import plotly.express as px

# Sample data and plot
df = px.data.iris()
fig = px.scatter(df, x='sepal_width', y='sepal_length',
color='species')

# Create a Dash app
app = dash.Dash(__name__)

# Layout of the app
app.layout = html.Div(children=[
    html.H1('Interactive Plotly Visualization'),
    dcc.Graph(
        id='scatter-plot',
        figure=fig
    )
])

# Run the app
if __name__ == '__main__':
    app.run_server(debug=True)
```

This will create a simple web dashboard with an interactive Plotly chart.

---

9. Exporting Plotly Plots

Plotly charts can be exported as static images, HTML files, or shared directly on the web.

Export Plotly Chart as an Image

python
Copy code

```
fig.write_image("plot.png")
```

Export Plotly Chart as HTML

python
Copy code

```
fig.write_html("plot.html")
```

You can share the HTML file, and anyone can view the interactive chart in a web browser.

Summary of Plotly Features

- **Interactive Plots**: Hover, zoom, pan, and click interactions.
- **Multiple Plot Types**: Scatter plots, line plots, bar charts, histograms, pie charts, and more.
- **3D Plots**: 3D scatter plots, surface plots, etc.
- **Customization**: Customize axis labels, titles, colors, themes, and more.
- **Subplots**: Combine multiple plots into a single figure.
- **Dash Integration**: Build interactive dashboards using Plotly and Dash.
- **Exporting**: Save plots as static images or interactive HTML files.

Plotly is an excellent choice for creating beautiful, interactive visualizations that can be easily shared or embedded in web applications.