

5a. Write a Python program to explain working with boken line graph using Annotations and Legends

1. Introduction.

- Bokeh is a Python library for creating interactive and visually appealing visualizations in web browsers. It simplifies the development of interactive plots, dashboards, and data applications, supporting various plot types and interactive features like tooltips, zooming, and panning. Bokeh's strength lies in its ease of use and ability to produce interactive visualizations with minimal code. Widely applied in fields like data science, engineering, and finance, Bokeh facilitates dynamic and clear data representation.
- This program offers insights into creating visually informative plots with annotations and legends using the Bokeh library. The program utilizes the Bokeh library to generate a line graph comparing the temperatures of Mysuru and Bengaluru over a five-day period in January. The code employs temperature data for both cities, plotting the information on a graph with days on the x-axis and temperatures in Celsius on the y-axis.
- Annotations are incorporated to highlight specific weather details for each city at a particular day, and a legend is added to distinguish the two lines representing Mysuru and Bengaluru temperatures.

2. Program code.

```
# Import necessary libraries

from bokeh.plotting import figure, show
from bokeh.models import Label, Legend


# Temperature data for Mysuru and Bengaluru in January (in °C)

days = [1, 2, 3, 4, 5]
temperature_bengaluru = [18, 20, 22, 19, 17]
temperature_mysuru = [22, 25, 20, 26, 23]


# Create a Bokeh figure

plot = figure(title="Temperature Comparison", x_axis_label="Days",
y_axis_label="Temperature (°C)")
```

```
# Plot the temperature for Mysuru
line_mysuru = plot.line(days, temperature_mysuru, line_width=2, line_color="blue")

# Plot the temperature for Bengaluru
line_bengaluru = plot.line(days, temperature_bengaluru, line_width=2,
line_color="green")

# Add annotations to the plot
annotation_mysuru = Label(x=4, y=22, text="Weather in Mysuru",
text_font_size="10pt", text_color="blue")

plot.add_layout(annotation_mysuru)

annotation_bengaluru = Label(x=4, y=20, text="Weather in Bengaluru",
text_font_size="10pt", text_color="green")

plot.add_layout(annotation_bengaluru)

# Add legend to the plot
legend = Legend(items=[
    ("Mysuru", [line_mysuru]),
    ("Bengaluru", [line_bengaluru]),
], location="top_left")

plot.add_layout(legend)

# Show the plot
show(plot)
```

3. Explanation of the code

1 from bokeh.plotting import figure, show:

- **figure:** This imports the figure class from Bokeh, which is the core component for creating plots. A figure represents the overall plotting area and is used to configure various aspects of the plot, such as title, axis labels, and visual elements.
- **show:** This function is used to display the Bokeh plot. After configuring the figure and adding desired elements, calling show renders the plot in a web browser or notebook.

2. from bokeh.models import Label, Legend:

- **Label:** This imports the Label class from Bokeh's models module. Labels are used for adding text annotations to the plot at specific coordinates. In this context, labels are employed to annotate points on the graph with information about the weather in Mysuru and Bengaluru.
- **Legend:** This imports the Legend class from Bokeh's models module. Legends are utilized to provide information about the visual elements on the plot, such as lines or markers. In this case, a legend is added to distinguish between the lines representing temperatures in Mysuru and Bengaluru.

3. days = [1, 2, 3, 4, 5]

- This list represents the five days of the observation period. Each element corresponds to a specific day, creating a sequential timeline for the temperature data.

temperature_bengaluru = [18, 20, 22, 19, 17]

- This list contains the recorded temperatures for Bengaluru over the corresponding days. The temperatures are in Celsius and follow the order of the days list.

temperature_mysuru = [22, 25, 20, 26, 23]

- Similarly, this list holds the temperatures recorded for Mysuru over the same five-day period. The temperatures are aligned with the days in the days list.

4. Creating a Bokeh figure object

```
plot=figure(title="TemperatureComparison",x_axis_label="Days",  
y_axis_label="Temperature (°C)")
```

- the line creates a Bokeh figure object, which serves as the foundation for the subsequent plotting of temperature data.

- `plot`: This variable is assigned the instance of the Bokeh figure class, which represents the main plotting area. It acts as a canvas where visual elements, such as lines and annotations, can be added.
- `title="Temperature Comparison"`: This sets the title of the plot to "Temperature Comparison." The title provides a brief description of the content or purpose of the plot.
- `x_axis_label="Days"`: This specifies the label for the x-axis as "Days." The x-axis represents the time variable, with each tick corresponding to a day in the observation period.
- `y_axis_label="Temperature (°C)"`: This designates the label for the y-axis as "Temperature (°C)." The y-axis represents the temperature values, measured in degrees Celsius.

5. Mysuru Temperature Line:

```
line_mysuru = plot.line(days, temperature_mysuru, line_width=2, line_color="blue")
```

- `plot.line(...)`: This method is used to create a line plot on the Bokeh figure (`plot`). It takes the x-axis values (`days`) and y-axis values (`temperature_mysuru`) as arguments.
- `line_width=2`: This parameter sets the width of the line representing Mysuru's temperatures to 2 units.
- `line_color="blue"`: This parameter specifies the color of the line as blue. The line represents the temperature variations in Mysuru over the specified days.
- `line_mysuru`: This variable stores the reference to the line plot for Mysuru. It can be later used for adding annotations or including it in the legend.

Similarly for the Bengaluru Temperature Line

6. Adding Annotations

```
annotation_mysuru = Label(x=4, y=22, text="Weather in Mysuru", text_font_size="10pt",  
text_color="blue"):
```

- `Label`: This is a Bokeh model used to create text annotations on the plot.
- `x=4, y=22`: These parameters set the coordinates where the annotation will be placed. In this case, it's positioned at `x=4` and `y=22` on the plot.

- `text="Weather in Mysuru"`: This sets the text content of the annotation to "Weather in Mysuru."
- `text_font_size="10pt"`: This parameter defines the font size of the text in the annotation. In this case, it's set to 10 points.
- `text_color="blue"`: This specifies the color of the text in the annotation, which is set to blue.

plot.add_layout(annotation_mysuru):

- `plot.add_layout(...)`: This method adds the specified layout (in this case, the `annotation_mysuru` label) to the Bokeh figure (plot).
- By adding this label to the layout, the text "Weather in Mysuru" with the specified styling will appear on the plot at the designated coordinates.

Similarly for Bengaluru, the annotation is generated.

7. Creating and configuring legend

- Creating and configuring a legend to distinguish between the temperature lines representing Mysuru and Bengaluru on the Bokeh plot. Let's break down the code:

```
legend = Legend(items=[("Mysuru", [line_mysuru]), ("Bengaluru", [line_bengaluru])],
location="top_left"):
```

- `Legend`: This is a Bokeh model used to create legends, which provide information about the visual elements on the plot.
- `items`: This parameter is a list of tuples, where each tuple contains a label and a corresponding visual element. In this case, the labels are "Mysuru" and "Bengaluru," and the visual elements are the line plots `line_mysuru` and `line_bengaluru`.
- `location="top_left"`: This sets the location of the legend on the plot. In this case, the legend is positioned at the top-left corner of the plot.

8. Adding the legend to the plot

plot.add_layout(legend):

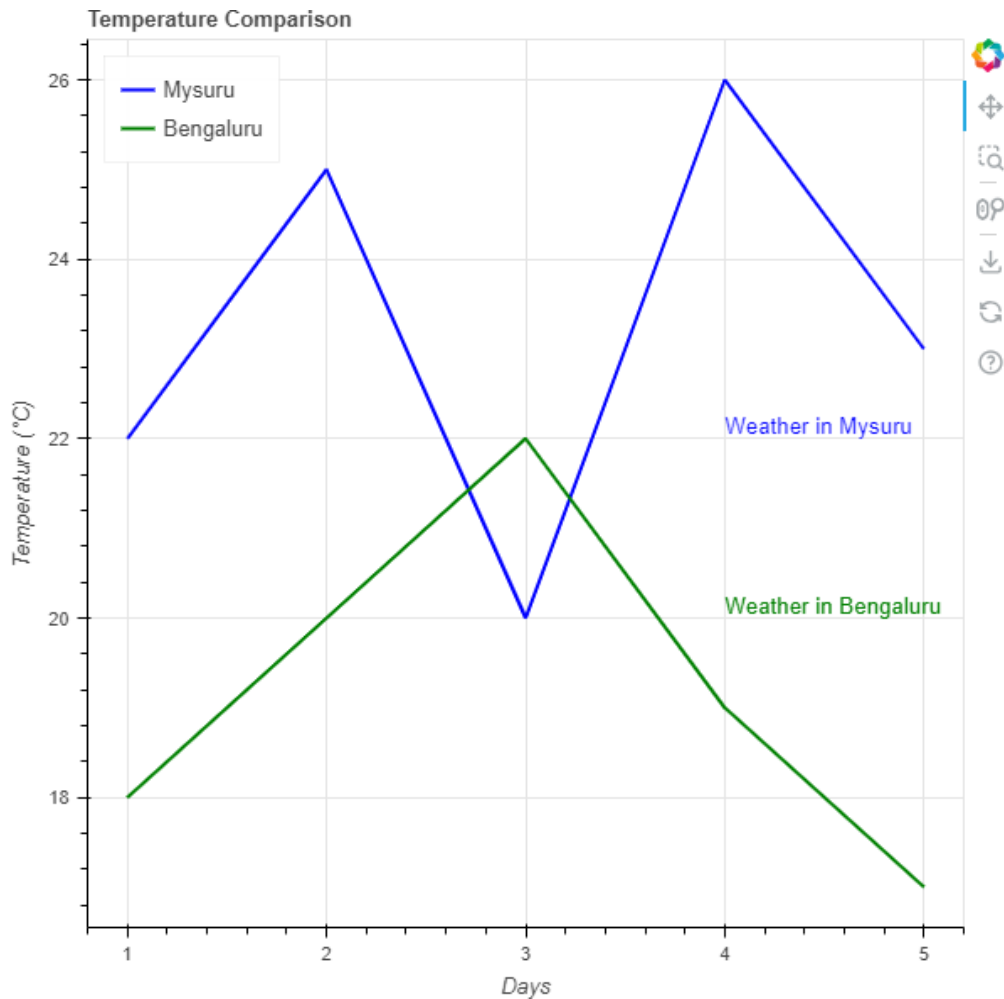
- After defining the legend variable that represents the legend configuration, this line adds the legend to the Bokeh figure (plot). The `add_layout` method is used to include the legend as part of the overall plot layout.

- This step ensures that the legend, which distinguishes between the temperature lines for Mysuru and Bengaluru, is displayed alongside the plot's visual elements.

show(plot):

- This line calls the show function, which renders and displays the Bokeh plot in a web browser or notebook environment.
- The plot variable, representing the Bokeh figure, contains all the configured elements, including the temperature lines, annotations, and legend. When show is invoked, it presents the finalized plot with the added legend to the viewer.

4. Output



Analysis: The plot shows that the temperature in Mysuru fluctuates, reaching a peak on the third day before dropping again, while the temperature in Bengaluru steadily increases until the third day and then sharply decreases. This suggests that the weather in Mysuru is more variable than the weather in Bengaluru, and that both cities experience a drop in temperature towards the end of the week.

Prominent use cases of Bokeh in different industries.

1. Finance: Stock Market Analysis

Bokeh is widely used in the finance industry to create interactive visualizations for stockmarket analysis. Candlestick charts, line graphs, and scatter plots can be employed to represent stock prices, trends, and trading volumes. Traders and analysts benefit from Bokeh's interactivity to zoom into specific time periods and assess historical performance.

2. Healthcare: Medical Data Visualization

Bokeh is utilized in healthcare to visualize complex medical data, such as patient records, disease progression, and treatment outcomes. Interactive plots and dashboards help healthcare professionals explore data trends, monitor patient vital signs, and make informed decisions about patient care.

3. Manufacturing: Quality Control Dashboards

In manufacturing, Bokeh is employed to create quality control dashboards that display real-time data from production processes. Line charts and scatter plots can visualize variations and trends in manufacturing metrics, aiding in process optimization and quality assurance.

4. Renewable Energy: Energy Production Monitoring

Bokeh is applied in the renewable energy sector to monitor and visualize energy production data from sources like solar and wind farms. Interactive plots help analyze production trends, track efficiency metrics, and optimize the performance of renewable energy systems.

5. Marketing and Sales: Customer Engagement Analytics

Bokeh is used in marketing and sales to develop interactive dashboards for customer engagement analytics. Line charts, scatter plots, and other visualizations help marketing teams assess campaign performance, conversion rates, and customer behavior, enabling data-driven decision-making.