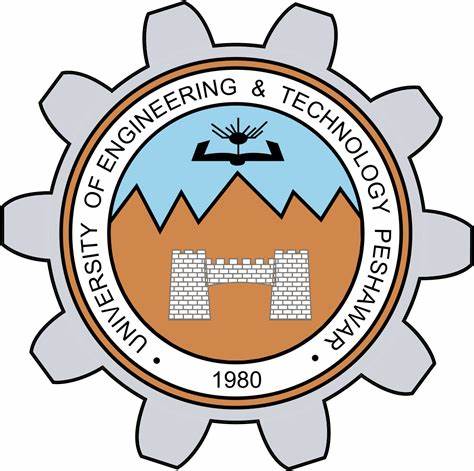
Lab report 03

**Bokeh**

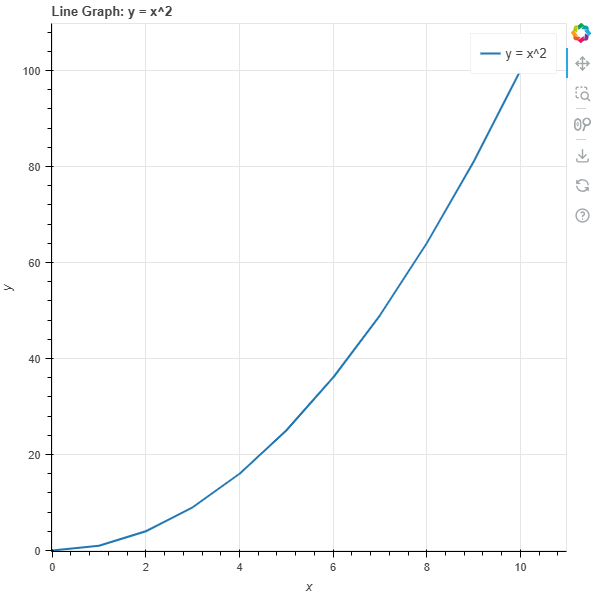
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**



**CSE-422L-Data Analytics Lab**

**Department of Computer System Engineering**

**University of Engineering and Technology Peshawar**

**Lab 3: Tasks (Bokeh)**

**Task 1: Plot a simple line graph with x-values ranging from 0 to 10 and y-values as the squares of x.**

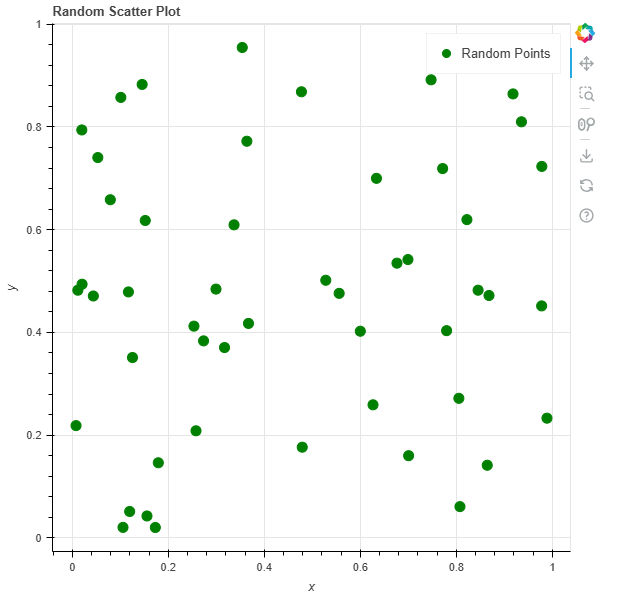
**from** **bokeh.plotting** **import** figure, show

x = [i **for** i **in** range(11)]y = [i\*\*2 **for** i **in** x]

p = figure(title="Line Graph: y = x^2", x\_axis\_label="x", y\_axis\_label="y")p.line(x, y, legend\_label="y = x^2", line\_width=2)

show(p)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**Task 2: Create a scatter plot of random points with custom marker shapes.**

**import** **numpy** **as** **np**

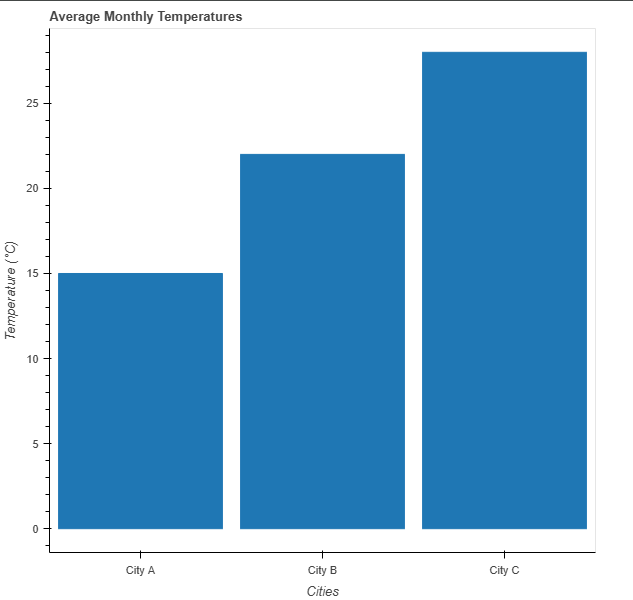
x = np.random.rand(50)y = np.random.rand(50)

p = figure(title="Random Scatter Plot", x\_axis\_label="x", y\_axis\_label="y")p.scatter(x, y, size=10, color="green", marker="circle", legend\_label="Random Points")

show(p)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 3: Plot a bar chart showing the average monthly temperatures for three cities.**



cities = ["City A", "City B", "City C"]temperatures = [15, 22, 28]

p = figure(x\_range=cities, title="Average Monthly Temperatures", toolbar\_location=**None**, tools="")

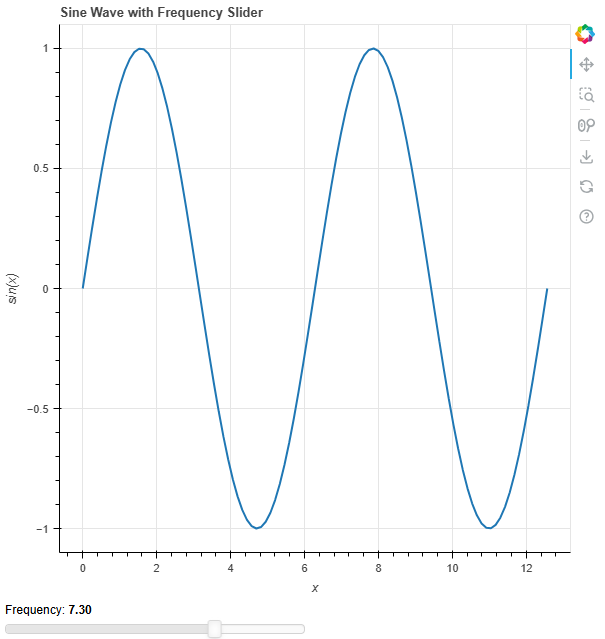
p.vbar(x=cities, top=temperatures, width=0.9)

p.xaxis.axis\_label = "Cities"p.yaxis.axis\_label = "Temperature (°C)"p.grid.grid\_line\_color = **None**

show(p)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 4: Create a sine wave that updates based on a frequency slider.**



**from** **bokeh.layouts** **import** columnfrom bokeh.models **import** **Slider**

x = np.linspace(0, 4 \* np.pi, 100)y = np.sin(x)

p = figure(title="Sine Wave with Frequency Slider", x\_axis\_label="x", y\_axis\_label="sin(x)")line = p.line(x, y, line\_width=2)

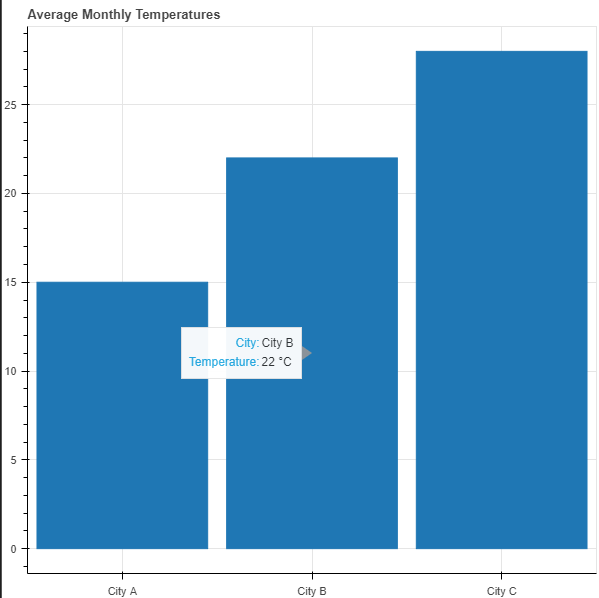
slider = Slider(start=1, end=10, value=1, step=0.1, title="Frequency")

**def** update(attr, old, new):    freq = slider.value    line.data\_source.data['y'] = np.sin(freq \* x)

slider.on\_change('value', update)

show(column(p, slider))

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 5: Create a bar chart with hover tooltips showing the value of each bar.**

**from** **bokeh.models** **import** HoverTool

cities = ["City A", "City B", "City C"]temperatures = [15, 22, 28]

p = figure(x\_range=cities, title="Average Monthly Temperatures",           toolbar\_location=**None**, tools="")

p.vbar(x=cities, top=temperatures, width=0.9)

hover = HoverTool()hover.tooltips = [("City", "@x"), ("Temperature", "@top °C")]p.add\_tools(hover)

show(p)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 6: Create a heatmap to show intensity values for a grid.**

**from** **bokeh.models** **import** ColorBarfrom bokeh.transform

**import** **linear\_cmapfrom** bokeh.palettes **import** **Viridis256**

data = np.random.rand(10, 10)

x, y = np.meshgrid(range(10), range(10))x = x.flatten()y = y.flatten()z = data.flatten()

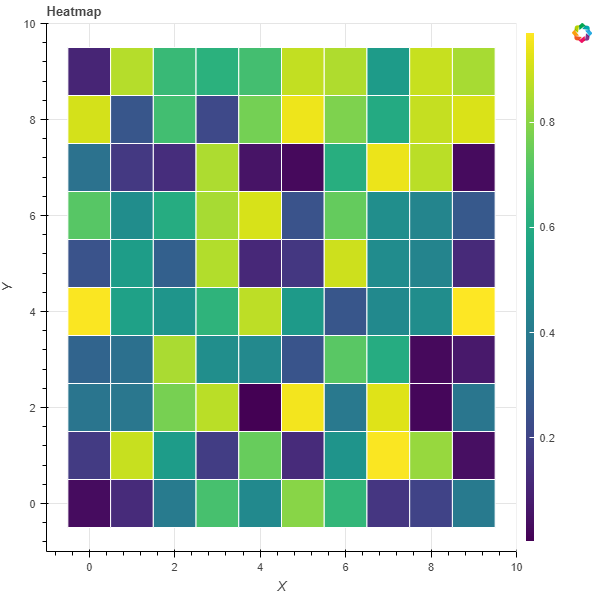
mapper = linear\_cmap(field\_name='z', palette=Viridis256, low=min(z), high=max(z))

p = figure(title="Heatmap", x\_axis\_label="X", y\_axis\_label="Y", tools="")

p.rect(x, y, 1, 1, color=mapper, line\_color="white")

color\_bar = ColorBar(color\_mapper=mapper['transform'], width=8, location=(0, 0))p.add\_layout(color\_bar, 'right')

show(p)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 7: Plot multiple sine and cosine waves on the same figure with a legend.**

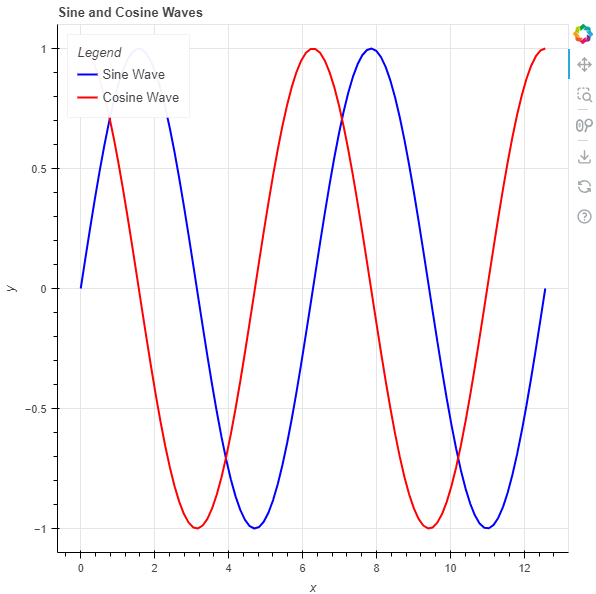
**import** **numpy** **as** **npfrom** bokeh.plotting **import** **figure**, **show**

x = np.linspace(0, 4 \* np.pi, 100)y\_sine = np.sin(x)y\_cosine = np.cos(x)

p = figure(title="Sine and Cosine Waves", x\_axis\_label="x", y\_axis\_label="y")p.line(x, y\_sine, legend\_label="Sine Wave", line\_width=2, color="blue")p.line(x, y\_cosine, legend\_label="Cosine Wave", line\_width=2, color="red")

p.legend.title = "Legend"p.legend.location = "top\_left"

show(p)



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Task 8: Create two linked line plots with panning and zooming.**

**from** **bokeh.plotting** **import** figure, showfrom bokeh.io **import** **output\_filefrom** bokeh.layouts **import** **row**

output\_file("linked\_line\_plots.html")

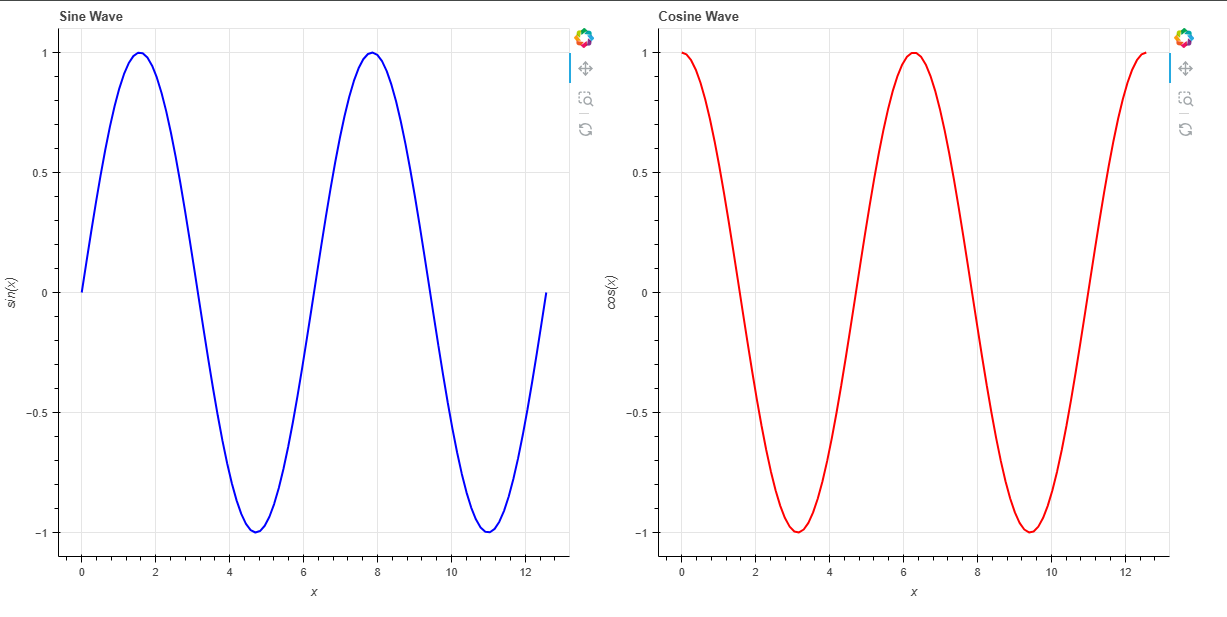
x = np.linspace(0, 4 \* np.pi, 100)y1 = np.sin(x)y2 = np.cos(x)

p1 = figure(title="Sine Wave", x\_axis\_label="x", y\_axis\_label="sin(x)",            tools="pan,box\_zoom,reset")p1.line(x, y1, line\_width=2, color="blue")

p2 = figure(title="Cosine Wave", x\_axis\_label="x", y\_axis\_label="cos(x)",            tools="pan,box\_zoom,reset")p2.line(x, y2, line\_width=2, color="red")

p1.x\_range = p2.x\_rangep1.y\_range = p2.y\_range

show(row(p1, p2))



\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_