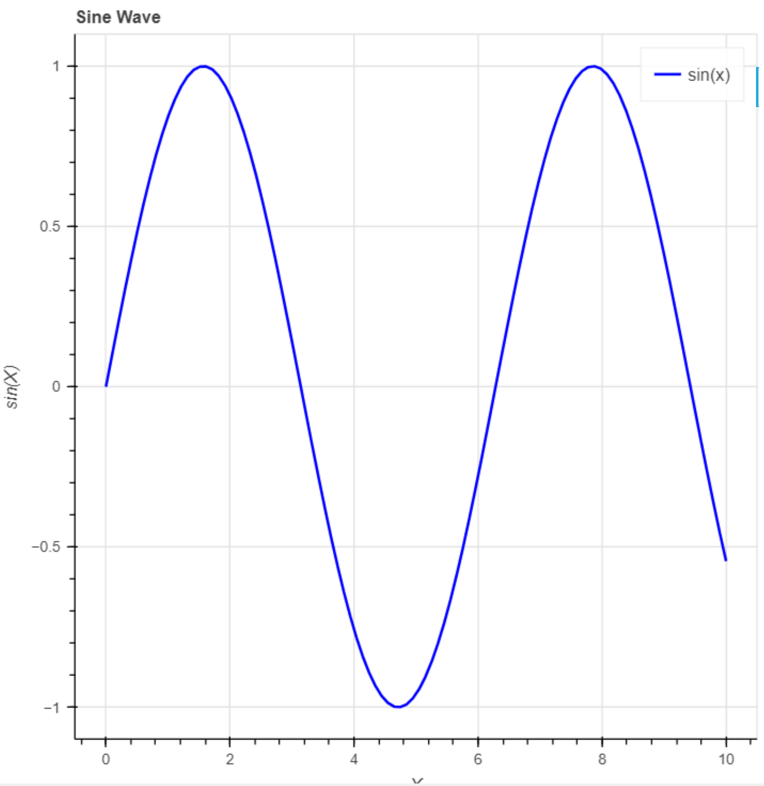
BOKEH ASSIGNMENT

ques 1.Create a Bokeh plot displaying a sine wave. Set x-values from 0 to 10 and y-values as the sine of x.

```
import numpy as np
from bokeh.plotting import figure, show
from bokeh.io import output_notebook

# Prepare the output to be displayed in a Jupyter notebook (or use output_file for HTML)
output_notebook()

# Generate data
x = np.linspace(0, 10, 100)  # 100 points from 0 to 10
y = np.sin(x)  # Sine of x

# Create a Bokeh plot
p = figure(title='Sine Wave', x_axis_label='X', y_axis_label='sin(X)')

# Add a line renderer
p.line(x, y, legend_label='sin(x)', line_width=2, line_color='blue')

# Show the plot
show(p)
```

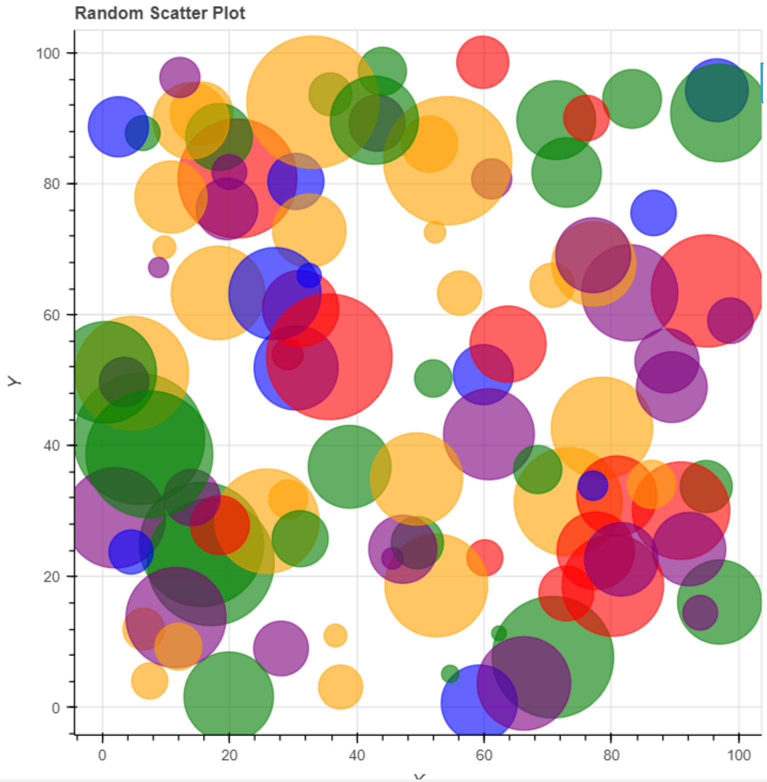


ques 2.Create a Bokeh scatter plot using randomly generated x and y values. Use different sizes and colors for the markers based on the 'sizes' and 'colors' columns.

```
import numpy as np
import pandas as pd
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
from bokeh.models import ColumnDataSource

# Prepare the output to be displayed in a Jupyter notebook (or use output_file for HTML)
output_notebook()

# Generate random data
np.random.seed(42)  # For reproducibility
```

```
n = 100  # Number of points
x = np.random.rand(n) * 100  # Random x values
y = np.random.rand(n) * 100  # Random y values
sizes = np.random.randint(10, 100, n)  # Random sizes for markers
colors = np.random.choice(['red', 'green', 'blue', 'orange', 'purple'], n)  # Random colors

# Create a ColumnDataSource
source = ColumnDataSource(data={'x': x, 'y': y, 'sizes': sizes, 'colors': colors})

# Create a Bokeh scatter plot
p = figure(title='Random Scatter Plot', x_axis_label='X', y_axis_label='Y')

# Add circle markers with varying sizes and colors
p.circle('x', 'y', size='sizes', color='colors', alpha=0.6, source=source)

# Show the plot
show(p)
```

```
BokehDeprecationWarning: 'circle() method with size value' was deprecated in Bokeh 3.4.0 and will be removed, use 'scatter(size=...) ins
```

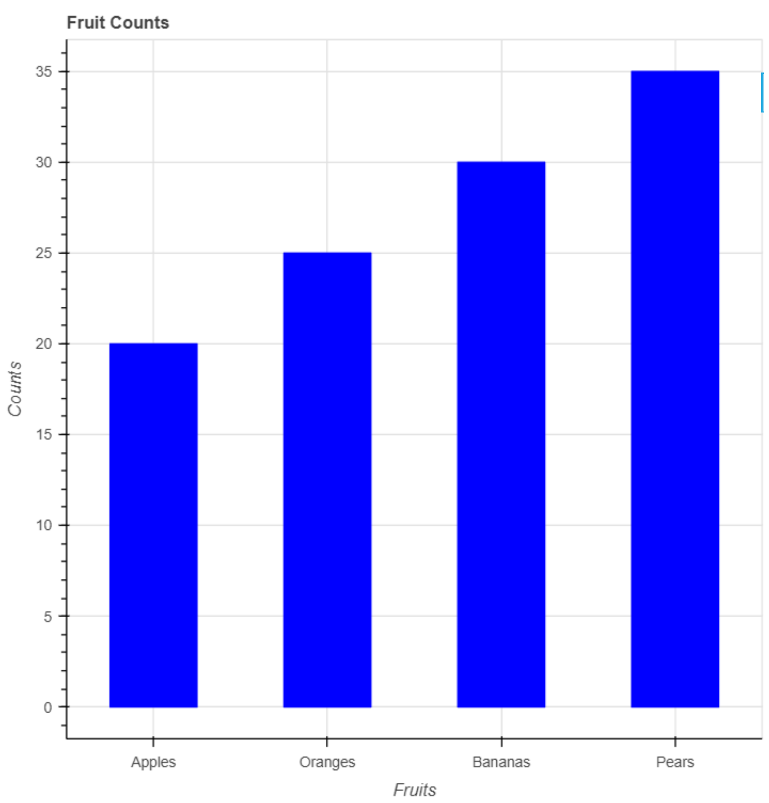


ques 3. Generate a Bokeh bar chart representing the counts of different fruits using the following dataset.

fruits = ['Apples', 'Oranges', 'Bananas', 'Pears']

counts = [20, 25, 30, 35]

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook

# Prepare the output to be displayed in a Jupyter notebook (or use output_file for HTML)
output_notebook()
fruits = ['Apples', 'Oranges', 'Bananas', 'Pears']
counts = [20, 25, 30, 35]
p = figure(x_range=fruits, title='Fruit Counts',
           x_axis_label='Fruits', y_axis_label='Counts')
# Add bars to the chart
p.vbar(x=fruits, top=counts, width=0.5, color='blue')
show(p)
```

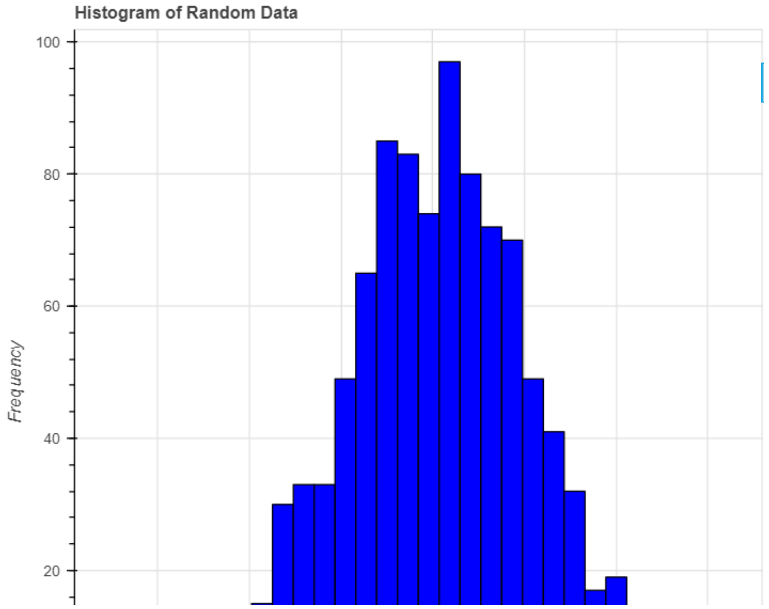


ques 4. Create a Bokeh histogram to visualize the distribution of the given data.

data_hist = np.random.randn(1000)

hist, edges = np.histogram(data_hist, bins=30)

```
import numpy as np
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
output_notebook()
data_hist = np.random.randn(1000)
hist, edges = np.histogram(data_hist, bins=30)
p = figure(title='Histogram of Random Data',
           x_axis_label='Value', y_axis_label='Frequency')
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:], fill_color='blue', line_color='black')
show(p)
```


Histogram of Random Data
100
80
60
40
20
Frequency