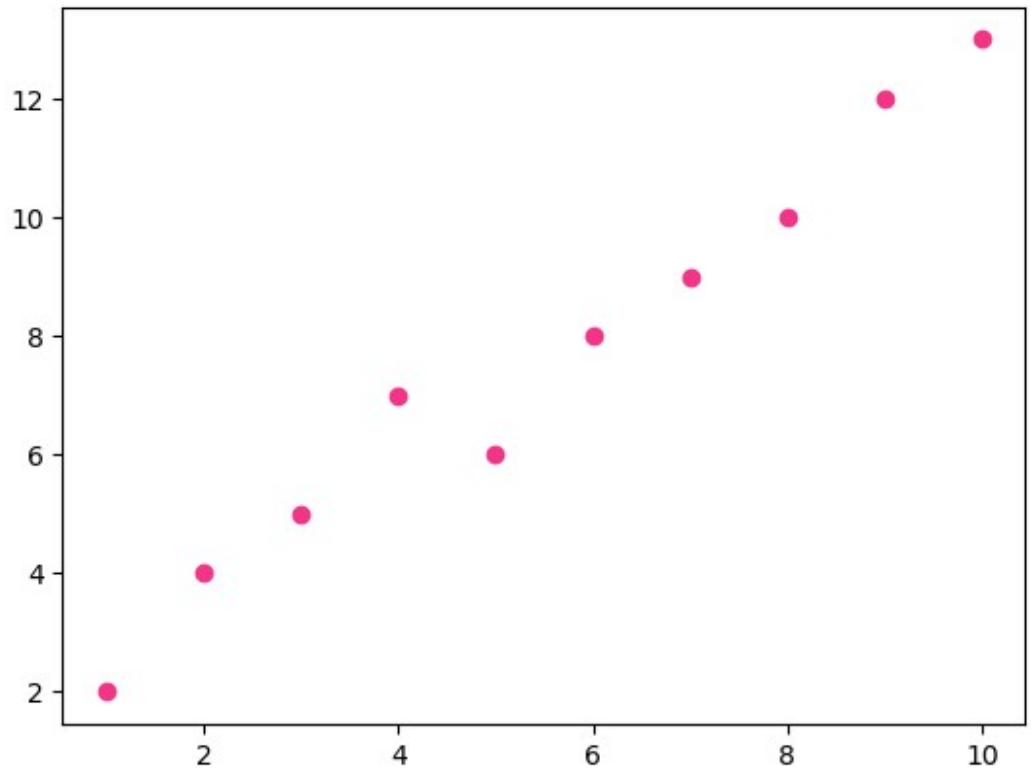
MATPLOTLIB ASSIGNMENT:

Q.1. Create a scatter plot using Matplotlib to visualize the relationship between two arrays, x and y for the given data

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [2, 4, 5, 7, 6, 8, 9, 10, 12, 13]

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [2, 4, 5, 7, 6, 8, 9, 10, 12, 13]
plt.scatter(x,y ,c='#F33488')
```

```
<matplotlib.collections.PathCollection at 0x783bff6e1d50>
```

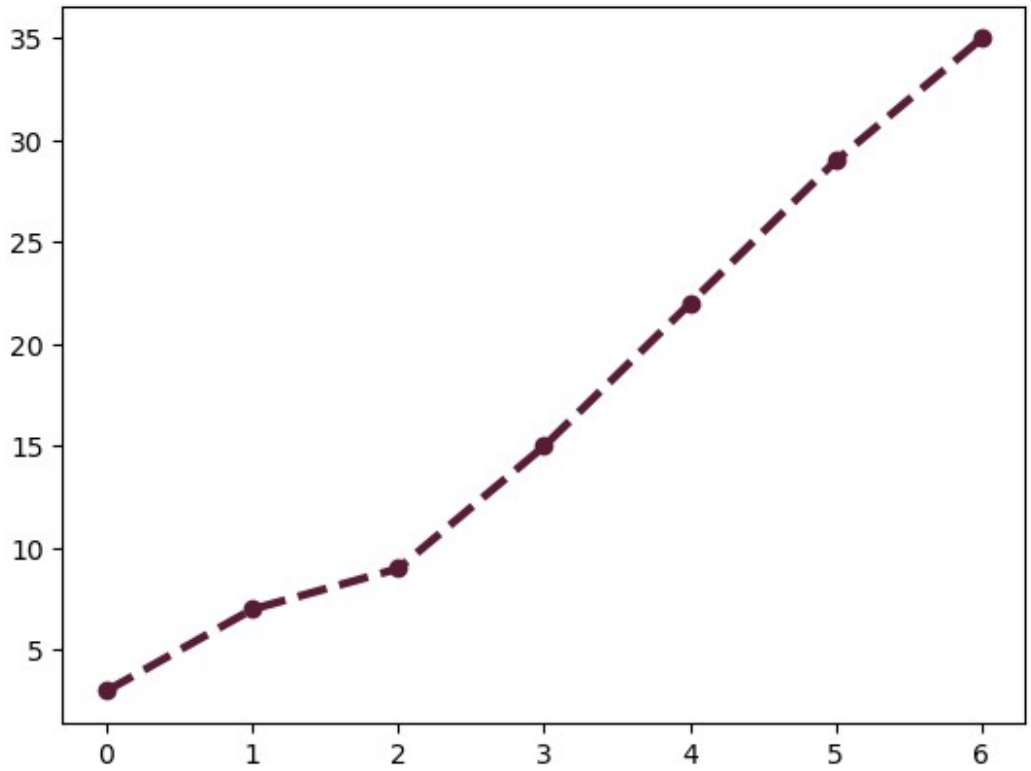


2. Generate a line plot to visualize the trend of values for the given data.

data = np.array([3, 7, 9, 15, 22, 29, 35])

```
data = np.array([3, 7, 9, 15, 22, 29, 35])

plt.plot(data, color = '#591B36', marker = "o", linestyle = '--',
linewidth = 3)
```

```
[<matplotlib.lines.Line2D at 0x783bff380c10>]
```



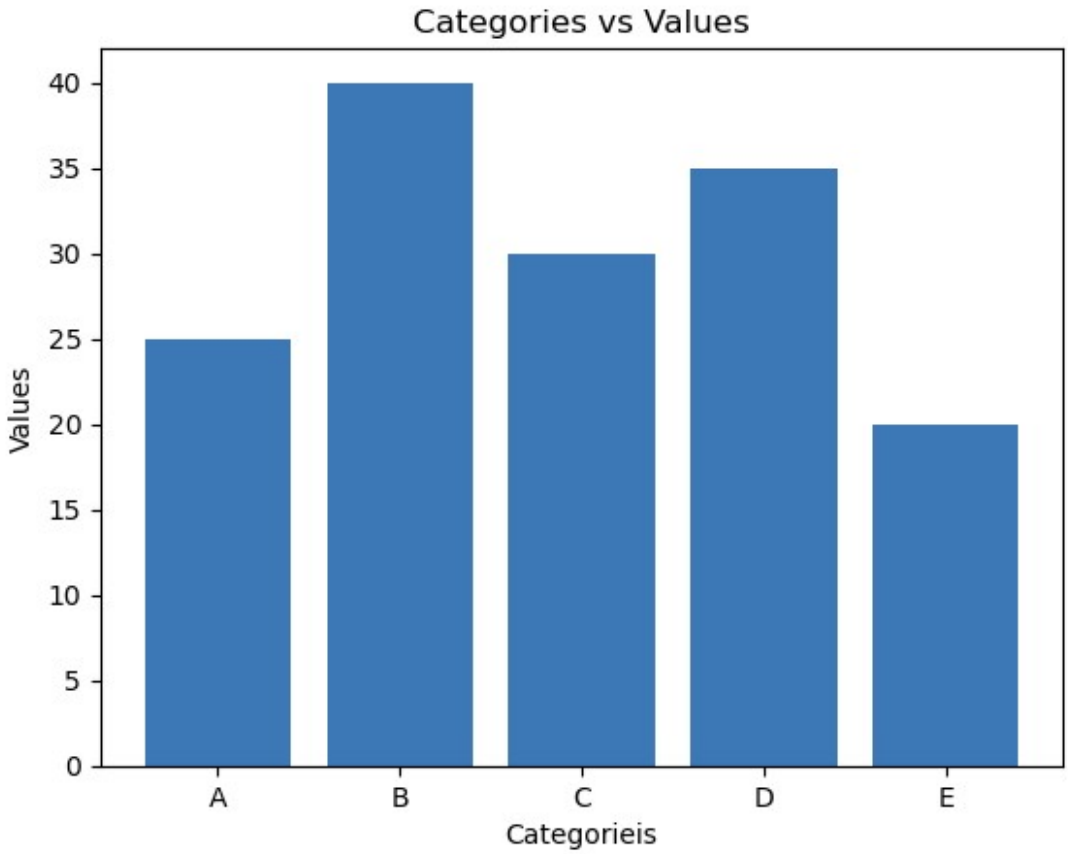
Q.3. Display a bar chart to represent the frequency of each item in the given array categories.

categories = ['A', 'B', 'C', 'D', 'E']

values = [25, 40, 30, 35, 20]

```
categories = ['A', 'B', 'C', 'D', 'E']
values = [25, 40, 30, 35, 20]
```

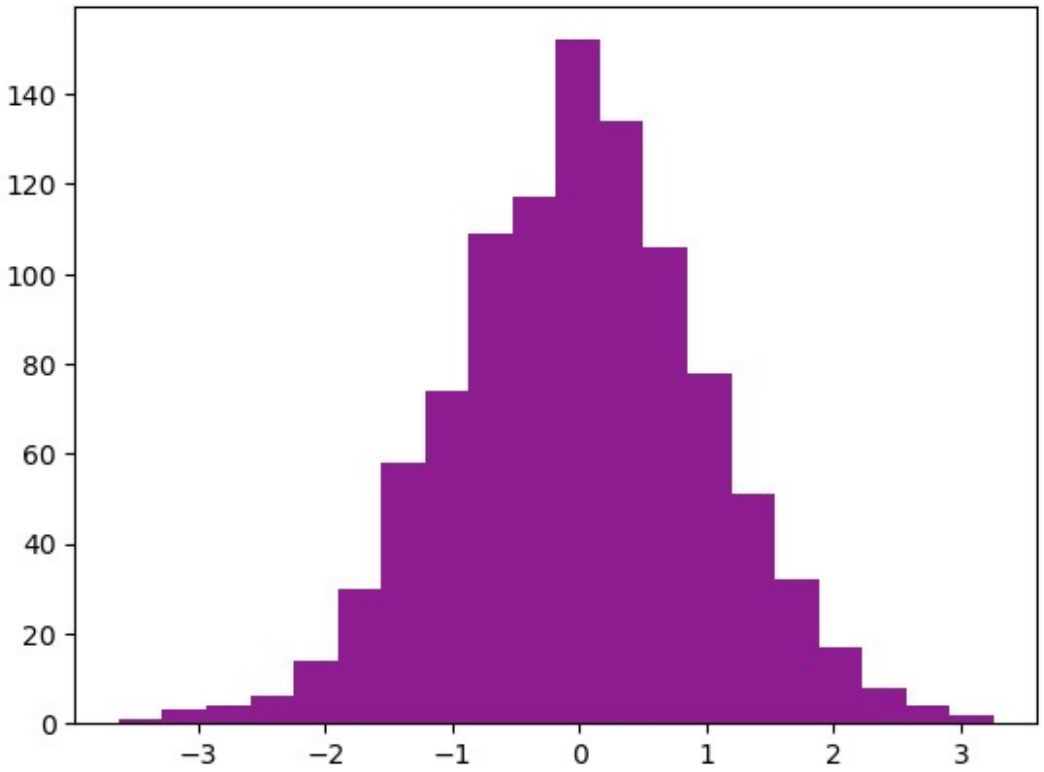
```
plt.bar(categories,values, color= '#3C78B5')
plt.xlabel('Categorieis')
plt.ylabel('Values')
plt.title('Categories vs Values')
plt.show()
```



Q.4.Create a histogram to visualize the distribution of values in the array data.

data = np.random.normal(0, 1, 1000)

```
data = np.random.normal(0, 1, 1000)
plt.hist(data, color = '#8E1D92', bins=20)
plt.show()
```

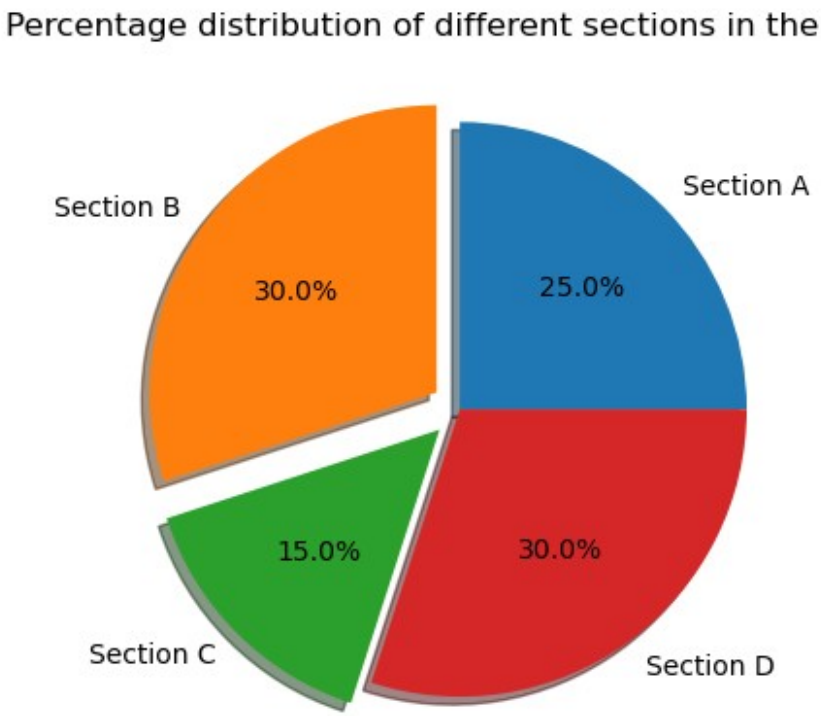


Q.5. Show a pie chart to represent the percentage distribution of different sections in the array 'sections'.

sections = ['Section A', 'Section B', 'Section C', 'Section D']

sizes = [25, 30, 15, 30]

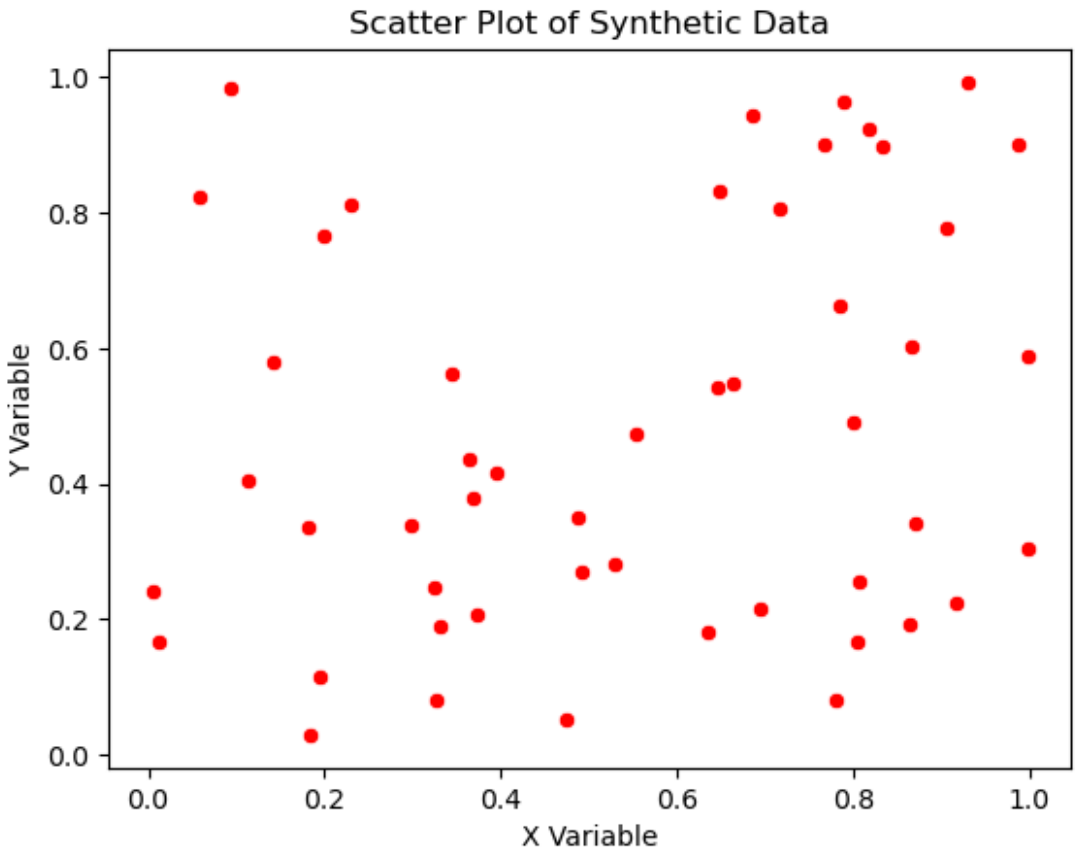
```
sections = ['Section A', 'Section B', 'Section C', 'Section D']
sizes = [25, 30, 15, 30]

explode = (0.0, 0.1, 0.1, 0.0)

plt.pie(sizes, labels=sections, autopct='%1.1f%%', explode=explode,
shadow=True)
plt.title("Percentage distribution of different sections in the
array")
plt.show()
```



SEABORN ASSIGNMENT:

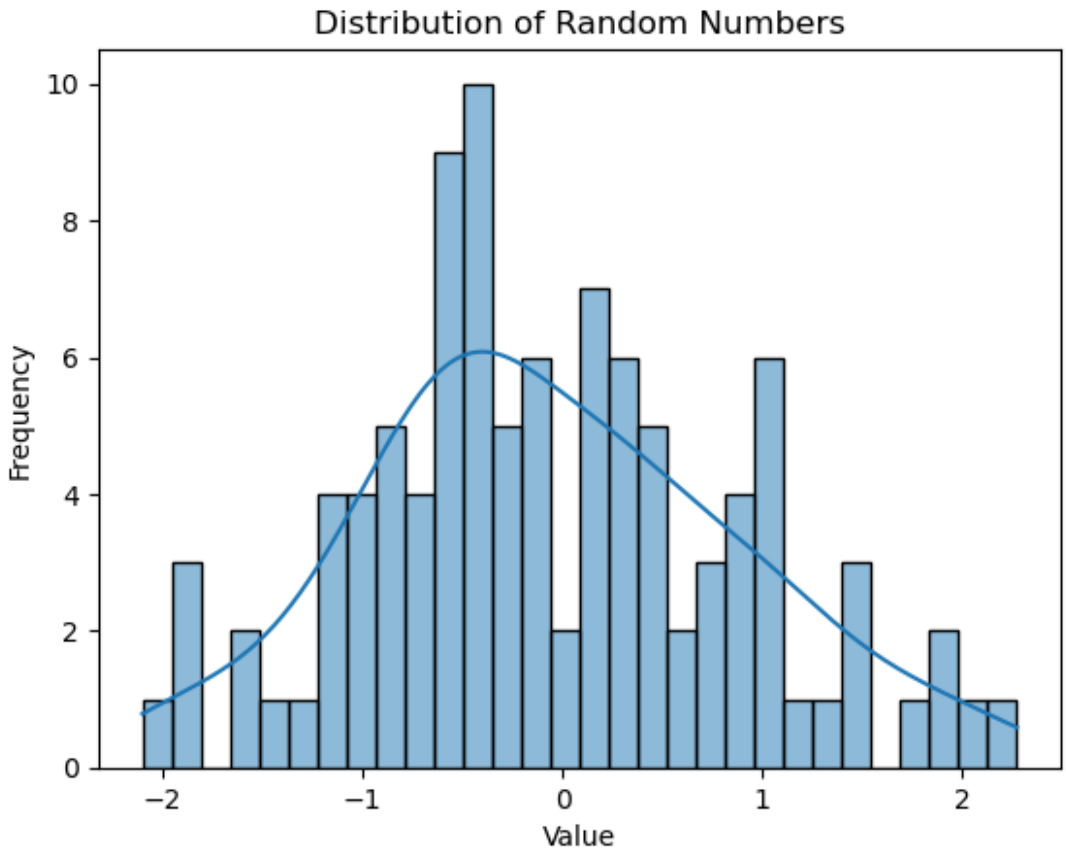
1. Create a scatter plot to visualize the relationship between two variables, by generating a synthetic dataset.

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

import seaborn as sns

data = pd.DataFrame({'X': x, 'Y': y})

sns.scatterplot(x='X', y='Y', data=data,c='r')

plt.title("Scatter Plot of Synthetic Data")
plt.xlabel("X Variable")
plt.ylabel("Y Variable")

plt.show()
```



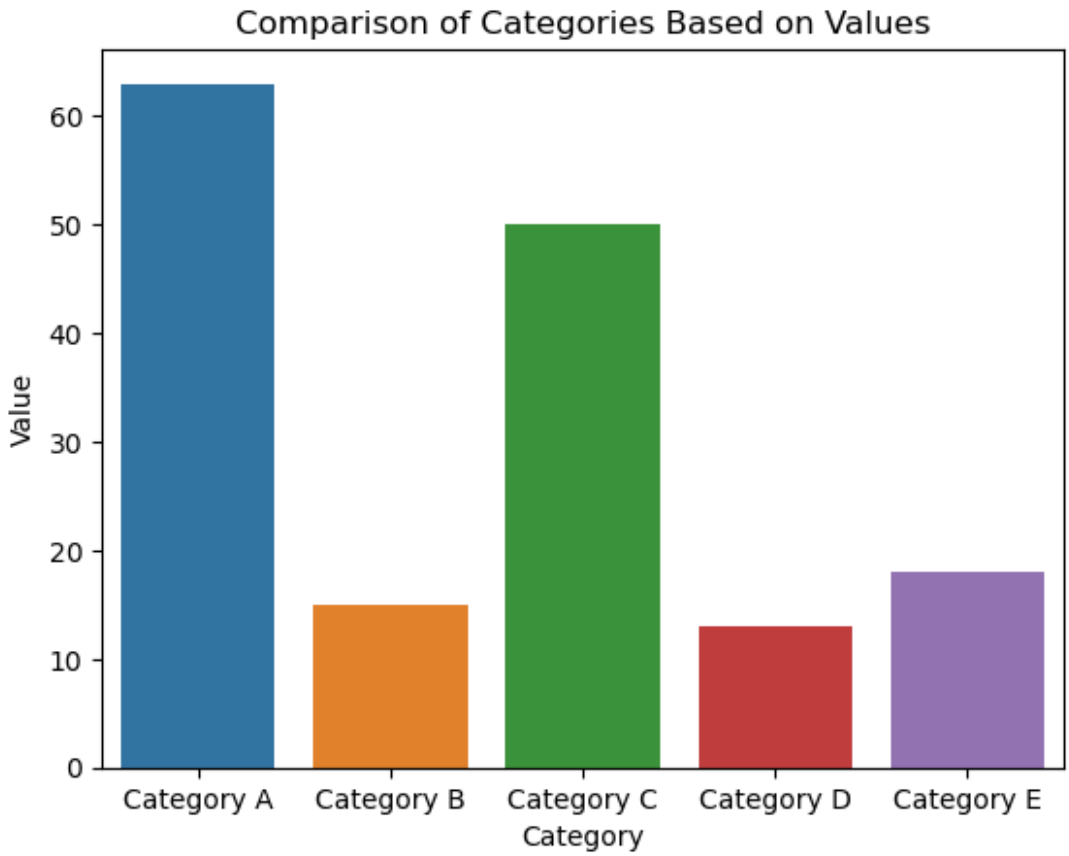
2.Generate a dataset of random numbers. Visualize the distribution of a numerical variable.

```
data = np.random.randn(100)

sns.histplot(data, kde=True, bins=30)

plt.title("Distribution of Random Numbers")
plt.xlabel("Value")
plt.ylabel("Frequency")

plt.show()
```

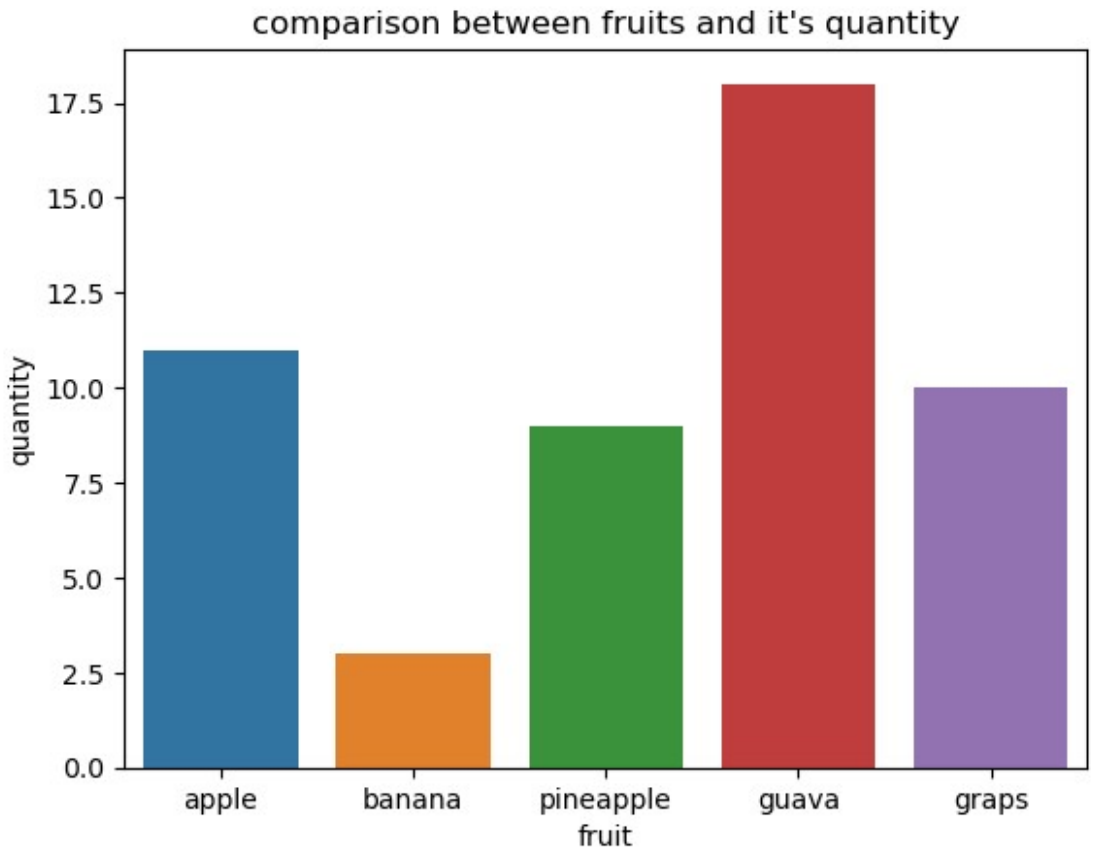


3. Create a dataset representing categories and their corresponding values. Compare different categories based on numerical values.

```
categories = ['Category A', 'Category B', 'Category C', 'Category D',
'Category E']
values = np.random.randint(10, 100, size=5)

data = pd.DataFrame({'Category': categories, 'Value': values})

sns.barplot(x='Category', y='Value', data=data)


plt.title("Comparison of Categories Based on Values")
plt.xlabel("Category")
plt.ylabel("Value")

plt.show()
```

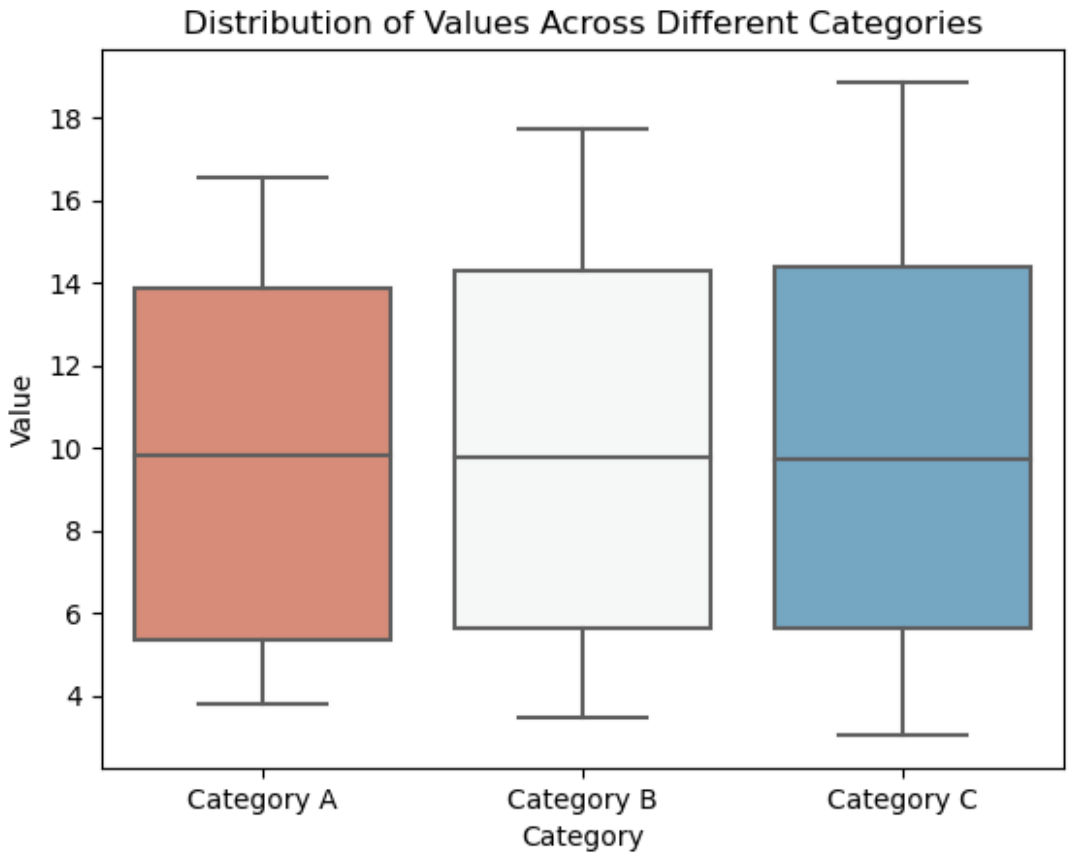



```
fruits = ['apple', 'banana', 'pineapple',  'guava','graps']
number_of_fruits = np.random.randint(2,20,size = 5)

data = pd.DataFrame({'fruit':fruits, 'quantity':number_of_fruits})
sns.barplot(x = 'fruit',y = 'quantity', data=data)
plt.title("comparison between fruits and it's quantity")
plt.xlabel('fruit')
plt.ylabel('quantity')
plt.show()
```

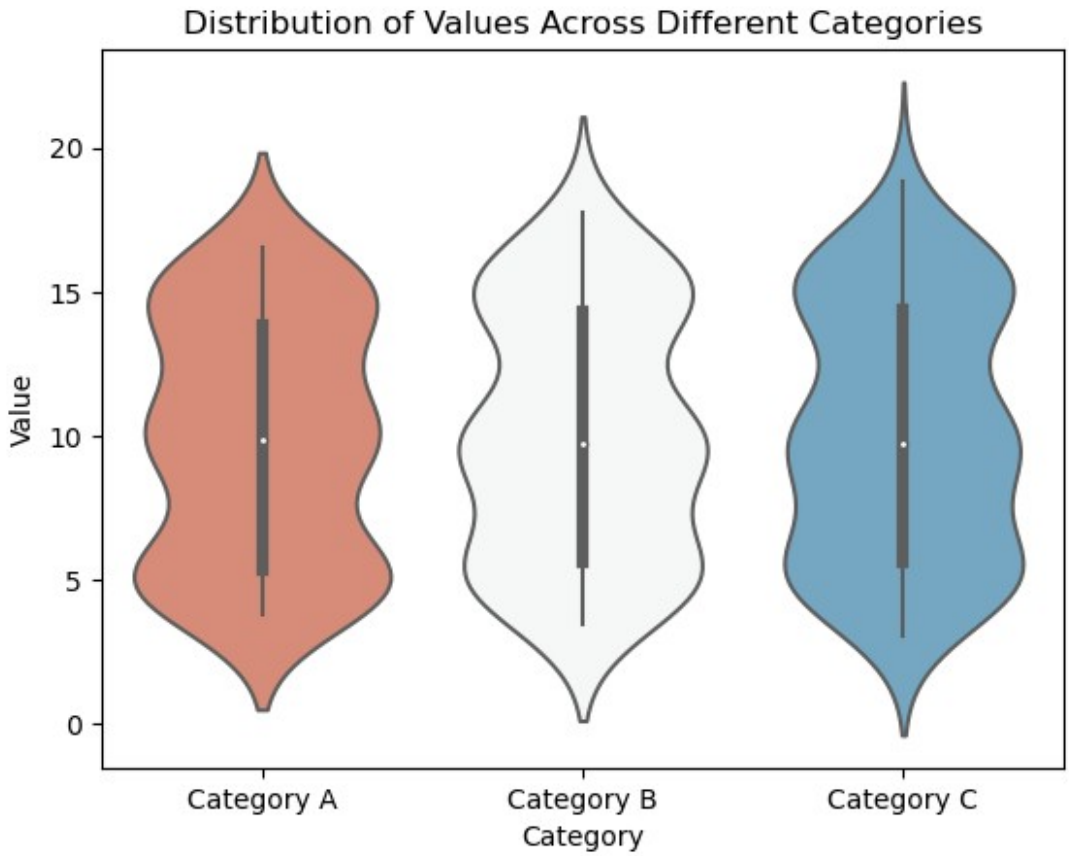


4. Generate a dataset with categories and numerical values. Visualize the distribution of a numerical variable across different categories.

```
np.random.seed(42)
categories = np.repeat(['Category A', 'Category B', 'Category C'],
100)
values = np.random.randn(300) + np.tile([5, 10, 15], 100)

data = pd.DataFrame({'Category': categories, 'Value': values})

sns.boxplot(x='Category', y='Value', data=data, palette="RdBu")

plt.title("Distribution of Values Across Different Categories")
plt.xlabel("Category")
plt.ylabel("Value")

plt.show()
```



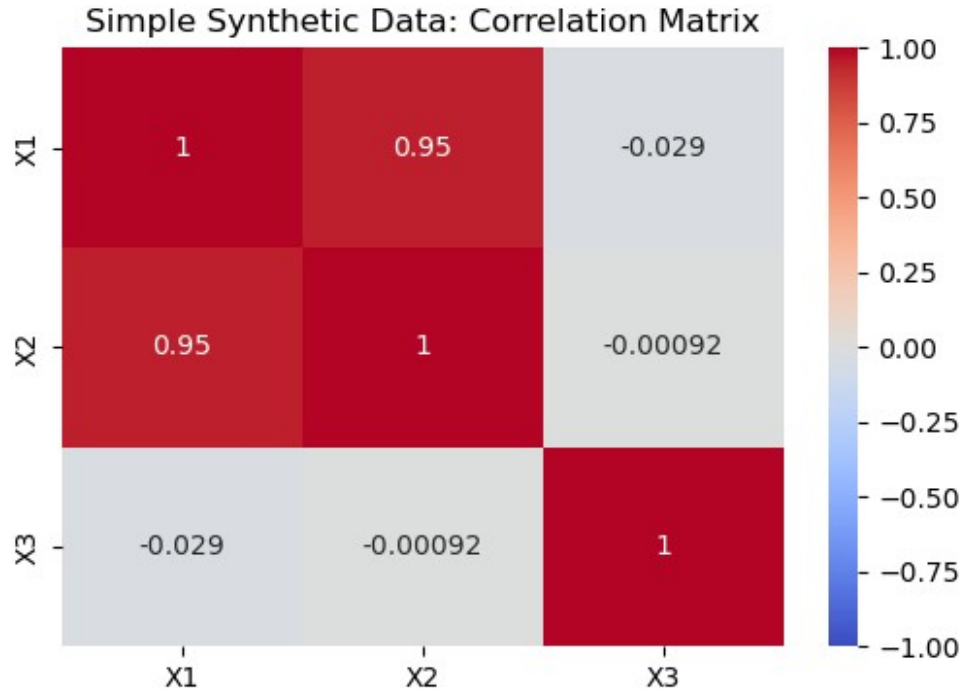

```
sns.violinplot(x='Category', y='Value', data=data, palette="RdBu")

plt.title("Distribution of Values Across Different Categories")
plt.xlabel("Category")
plt.ylabel("Value")

plt.show()
```



5. Generate a synthetic dataset with correlated features. Visualize the correlation matrix of a dataset using a heatmap.

```
np.random.seed(42)


X1 = np.random.rand(100)
X2 = X1 + np.random.normal(0, 0.1, 100)
X3 = np.random.rand(100)

data = pd.DataFrame({
    'X1': X1,
    'X2': X2,
    'X3': X3
})

plt.figure(figsize=(6, 4))
sns.heatmap(data.corr(), annot=True, cmap='coolwarm', vmin=-1, vmax=1)
```

```
plt.title("Simple Synthetic Data: Correlation Matrix")
plt.show()
```



PLOTLY ASSIGNMENT:

1. Using the given dataset, to generate a 3D scatter plot to visualize the distribution of data points in a three dimensional space.

np.random.seed(30)

data = {

```
'X': np.random.uniform(-10, 10, 300),
'Y': np.random.uniform(-10, 10, 300),
'Z': np.random.uniform(-10, 10, 300)
```

} df = pd.DataFrame(data)

```
import plotly.express as px
import plotly.graph_objects as go

# Generate the dataset
np.random.seed(30)
data = {
    'X': np.random.uniform(-10, 10, 300),
    'Y': np.random.uniform(-10, 10, 300),
    'Z': np.random.uniform(-10, 10, 300)
}
df = pd.DataFrame(data)

# Create the 3D scatter plot using Plotly Express
fig = px.scatter_3d(df, x='X', y='Y', z='Z', color='X')

# Customize the plot
fig.update_layout(scene=dict(
                    xaxis_title='X Axis',
                    yaxis_title='Y Axis',
                    zaxis_title='Z Axis'))

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

2. Using the Student Grades, create a violin plot to display the distribution of scores across different grade categories.

np.random.seed(15)

data = {

```
'Grade': np.random.choice(['A', 'B', 'C', 'D', 'F'], 200),
'Score': np.random.randint(50, 100, 200)}
```

df = pd.DataFrame(data❯)

❮

Using the sales data, generate a heatmap to visualize the variation in sales across different months and days.

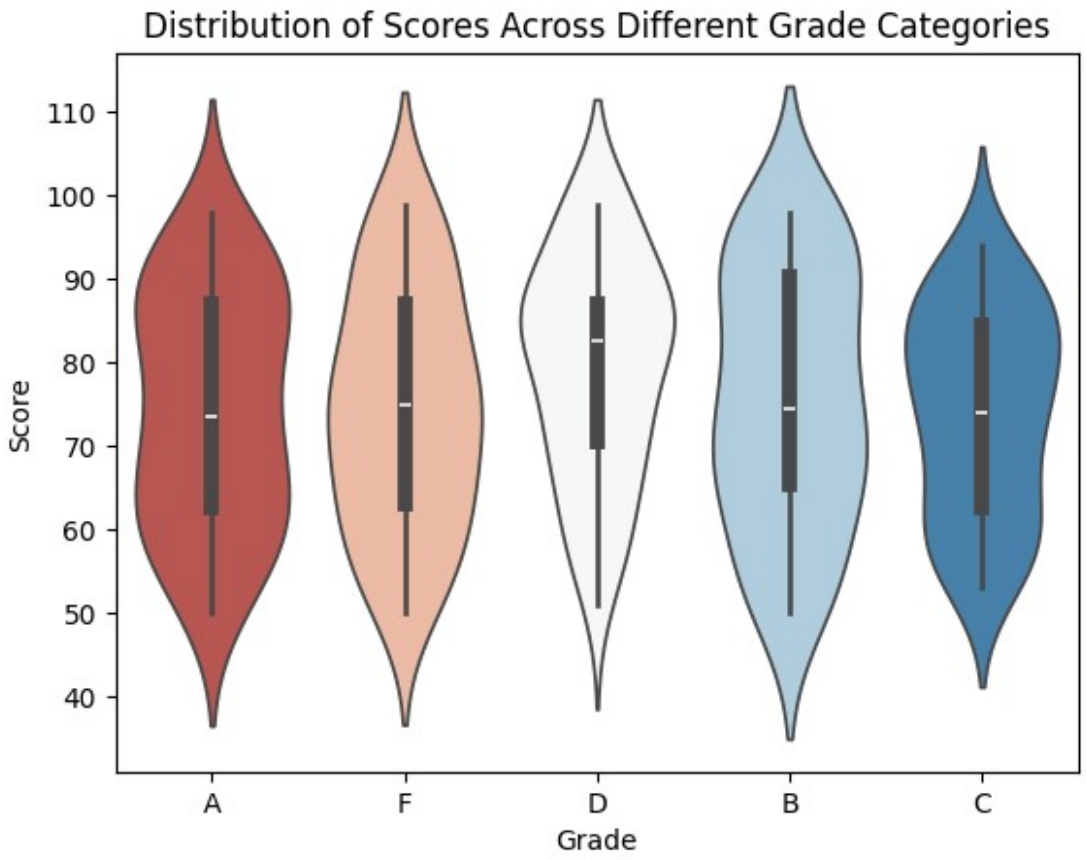
np.random.seed(20)

data = {

```
'Month': np.random.choice(['Jan', 'Feb', 'Mar', 'Apr', 'May'], 100),
'Day': np.random.choice(range(1, 31), 100),
'Sales': np.random.randint(1000, 5000, 100)}
```

df = pd.DataFrame(data)

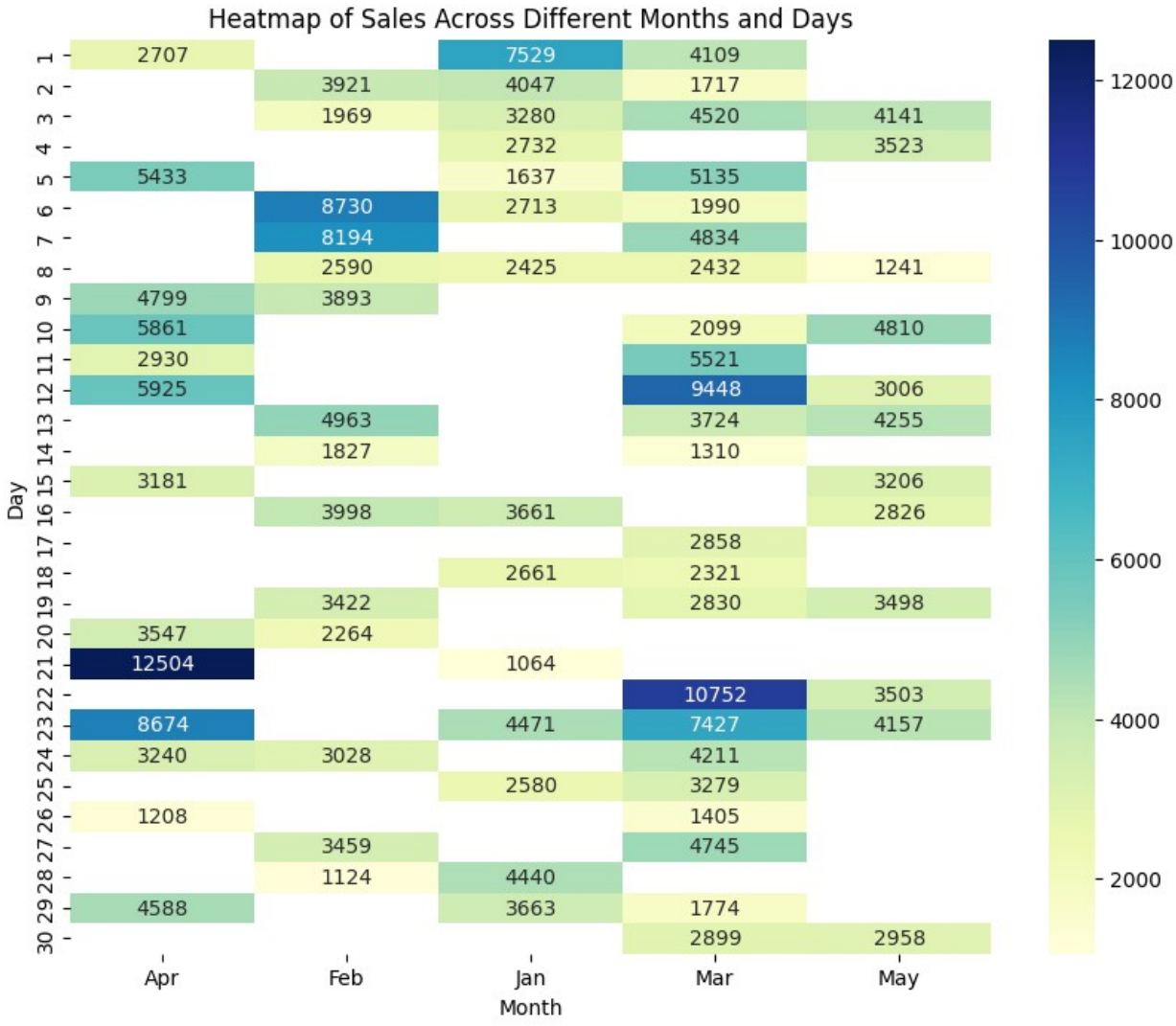
```
np.random.seed(15)
data = {
    'Grade': np.random.choice(['A', 'B', 'C', 'D', 'F'], 200),
    'Score': np.random.randint(50, 100, 200)
}
df = pd.DataFrame(data›)
{ ## Using the sales data, generate a heatmap to visualize the
variation in sales across different months and days.

np.random.seed(20)
data = {
    'Month': np.random.choice(['Jan', 'Feb', 'Mar', 'Apr', 'May'],
100),
    'Day': np.random.choice(range(1, 31), 100),
    'Sales': np.random.randint(1000, 5000, 100)
}
df = pd.DataFrame(data)

np.random.seed(15)
data = {
    'Grade': np.random.choice(['A', 'B', 'C', 'D', 'F'], 200),
    'Score': np.random.randint(50, 100, 200)}
df = pd.DataFrame(data)

sns.violinplot(x='Grade', y='Score', data=df, palette="RdBu")

plt.title("Distribution of Scores Across Different Grade Categories")
plt.xlabel("Grade")
plt.ylabel("Score")

plt.show()
```



3. Using the sales data, generate a heatmap to visualize the variation in sales across different months and days.

```
## Using the sales data, generate a heatmap to visualize the variation
in sales across different months and days.

np.random.seed(20)
data = {
    'Month': np.random.choice(['Jan', 'Feb', 'Mar', 'Apr', 'May'],
100),
    'Day': np.random.choice(range(1, 31), 100),
    'Sales': np.random.randint(1000, 5000, 100)
}
df = pd.DataFrame(data)
sales_pivot = df.pivot_table(index='Day', columns='Month',
values='Sales',aggfunc='sum')

plt.figure(figsize=(10, 8))
sns.heatmap(sales_pivot, cmap='YlGnBu', annot=True, fmt='.0f')
```

```
plt.title("Heatmap of Sales Across Different Months and Days")
plt.xlabel("Month")
plt.ylabel("Day")

plt.show()
```



4. Using the given x and y data, generate a 3D surface plot to visualize the function .

x = np.linspace(-5, 5, 100) y = np.linspace(-5, 5, 100) x, y = np.meshgrid(x, y) z = np.sin(np.sqrt(x**2 + y**2)) data = { 'X': x.flatten(), 'Y': y.flatten(), 'Z': z.flatten() } df = pd.DataFrame(data)

```
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))
data = {
    'X': x.flatten(),
    'Y': y.flatten(),
    'Z': z.flatten()
}
df = pd.DataFrame(data)

fig = go.Figure(go.Surface(x=x, y=y, z=z))

fig.update_layout(
    title="3D Surface Plot of sin(sqrt(x² + y²))",
    scene=dict(
        xaxis_title='X Axis',
        yaxis_title='Y Axis',
        zaxis_title='Z Axis'
    )
)

fig.show()
```

5. Using the given dataset, create a bubble chart to represent each country's population (y-axis), GDP (xaxis), and bubble size proportional to the population.

np.random.seed(25) data = { 'Country': ['USA', 'Canada', 'UK', 'Germany', 'France'], 'Population': np.random.randint(100, 1000, 5), 'GDP': np.random.randint(500, 2000, 5) } df = pd.DataFrame(data)

```
np.random.seed(25)
data = {
    'Country': ['USA', 'Canada', 'UK', 'Germany', 'France'],
    'Population': np.random.randint(100, 1000, 5),
    'GDP': np.random.randint(500, 2000, 5)
}
df = pd.DataFrame(data)

# Create a bubble chart using Plotly Express
fig = px.scatter(
    df,
    x='GDP',
    y='Population',
    size='Population',
```

```
    hover_name='Country',
    title="Bubble Chart: Population vs GDP",
    labels={'GDP': 'GDP (in billion USD)', 'Population': 'Population
(in millions)'}
)

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

BOKEH ASSIGNMENT:

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 bokeh.io
import bokeh.plotting
bokeh.io.output_notebook()
from bokeh.plotting import figure, output_file, show

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

x = np.linspace(0, 10, 100)
y = np.sin(x)

plot = figure(title="Sine Wave", x_axis_label='X', y_axis_label='Y')

plot.line(x, y, legend_label="Sine Wave", line_width=2, color='blue')

output_notebook()
show(plot)
```

```
"'use strict';\n(function(root) {\n  function now() {\n    return new
Date();\n  }\n\n  const force = true;\n\n  if (typeof
root._bokeh_onload_callbacks === \"undefined\" || force === true) {\n
root._bokeh_onload_callbacks = [];\n    root._bokeh_is_loading =
undefined;\n  }\n\n\n  if (typeof (root._bokeh_timeout) ===
\"undefined\" || force === true) {\n    root._bokeh_timeout =
Date.now() + 5000;\n    root._bokeh_failed_load = false;\n  }\n\n
const NB_LOAD_WARNING = {'data': {'text/html':\n     \"<div
style='background-color: #fdd'>\\n\"+\n     \"<p>\\n\"+\n
\"BokehJS does not appear to have successfully loaded. If loading
BokehJS from CDN, this \\n\"+\n     \"may be due to a slow or bad
network connection. Possible fixes:\\n\"+\n     \"</p>\\n\"+\n
\"<ul>\\n\"+\n     \"<li>re-rerun `output_notebook()` to attempt to
load from CDN again, or</li>\\n\"+\n     \"<li>use INLINE resources
instead, as so:</li>\\n\"+\n     \"</ul>\\n\"+\n     \"<code>\\n\"+\n
```

```
\"from bokeh.resources import INLINE\\n\"+\n
\"output_notebook(resources=INLINE)\\n\"+\n     \"</code>\\n\"+\n
\"</div>\"}};\n\n  function display_loaded(error = null) {\n    const
el = document.getElementById(null);\n    if (el != null) {\n
const html = (() => {\n        if (typeof root.Bokeh ===
\"undefined\") {\n          if (error == null) {\n            return
\"BokehJS is loading ...\";\n          } else {\n            return
\"BokehJS failed to load.\";\n          }\n        } else {\n
const prefix = `BokehJS ${root.Bokeh.version}`;\n          if (error
== null) {\n            return `${prefix} successfully loaded.`;\n
} else {\n            return `${prefix} <b>encountered errors</b>
while loading and may not function as expected.`;\n          }\n
}\n      })();\n      el.innerHTML = html;\n\n      if (error != null)
{\n        const wrapper = document.createElement(\"div\");\n
wrapper.style.overflow = \"auto\";\n        wrapper.style.height =
\"5em\";\n        wrapper.style.resize = \"vertical\";\n        const
content = document.createElement(\"div\");\n
content.style.fontFamily = \"monospace\";\n
content.style.whiteSpace = \"pre-wrap\";\n
content.style.backgroundColor = \"rgb(255, 221, 221)\";\n
content.textContent = error.stack ?? error.toString();\n
wrapper.append(content);\n        el.append(wrapper);\n      }\n    }
else if (Date.now() < root._bokeh_timeout) {\n      setTimeout(() =>
display_loaded(error), 100);\n    }\n  }\n\n  function run_callbacks()
{\n    try {\n
root._bokeh_onload_callbacks.forEach(function(callback) {\n        if
(callback != null)\n          callback();\n      });\n    } finally {\
n      delete root._bokeh_onload_callbacks\n    }\n
console.debug(\"Bokeh: all callbacks have finished\");\n  }\n\n
function load_libs(css_urls, js_urls, callback) {\n    if (css_urls ==
null) css_urls = [];\n    if (js_urls == null) js_urls = [];\n\n
root._bokeh_onload_callbacks.push(callback);\n    if
(root._bokeh_is_loading > 0) {\n      console.debug(\"Bokeh: BokehJS
is being loaded, scheduling callback at\", now());\n      return
null;\n    }\n    if (js_urls == null || js_urls.length === 0) {\n
run_callbacks();\n      return null;\n    }\n
console.debug(\"Bokeh: BokehJS not loaded, scheduling load and
callback at\", now());\n    root._bokeh_is_loading = css_urls.length +
js_urls.length;\n\n    function on_load() {\n
root._bokeh_is_loading--;\n      if (root._bokeh_is_loading === 0) {\n
console.debug(\"Bokeh: all BokehJS libraries/stylesheets loaded\");\n
run_callbacks()\n      }\n    }\n\n    function on_error(url) {\n
console.error(\"failed to load \" + url);\n    }\n\n    for (let i =
0; i < css_urls.length; i++) {\n      const url = css_urls[i];\n
const element = document.createElement(\"link\");\n
element.onload = on_load;\n      element.onerror = on_error.bind(null,
url);\n      element.rel = \"stylesheet\";\n      element.type =
\"text/css\";\n      element.href = url;\n      console.debug(\"Bokeh:
injecting link tag for BokehJS stylesheet: \", url);\n
```

```
document.body.appendChild(element);\n    }\n\n    for (let i = 0; i < js_urls.length; i++) {\n      const url = js_urls[i];\n      const element = document.createElement('script');\n      element.onload = on_load;\n      element.onerror = on_error.bind(null, url);\n      element.async = false;\n      element.src = url;\n      console.debug(\"Bokeh: injecting script tag for BokehJS library: \", url);\n      document.head.appendChild(element);\n    }\n  };\n\n  function inject_raw_css(css) {\n    const element = document.createElement(\"style\");\n    element.appendChild(document.createTextNode(css));\n    document.body.appendChild(element);\n  }\n\n  const js_urls = [\"https://cdn.bokeh.org/bokeh/release/bokeh-3.4.3.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-gl-3.4.3.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-widgets-3.4.3.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-tables-3.4.3.min.js\", \"https://cdn.bokeh.org/bokeh/release/bokeh-mathjax-3.4.3.min.js\"];\n  const css_urls = [];\n\n  const inline_js = [    function(Bokeh) {\n    Bokeh.set_log_level(\"info\");\n    },\nfunction(Bokeh) {\n    }\n  ];\n\n  function run_inline_js() {\n    if (root.Bokeh !== undefined || force === true) {\n      try {\n            for (let i = 0; i < inline_js.length; i++) {\n        inline_js[i].call(root, root.Bokeh);\n    }\n\n      } catch (error) {throw error;\n      }} else if (Date.now() < root._bokeh_timeout) {\n    setTimeout(run_inline_js, 100);\n    } else if (!root._bokeh_failed_load) {\n      console.log(\"Bokeh: BokehJS failed to load within specified timeout.\");\n      root._bokeh_failed_load = true;\n    } else if (force !== true) {\n      const cell = $(document.getElementById(null)).parents('.cell').data().cell;\n    cell.output_area.append_execute_result(NB_LOAD_WARNING)\n    }\n  }\n\n  if (root._bokeh_is_loading === 0) {\n    console.debug(\"Bokeh: BokehJS loaded, going straight to plotting\");\n    run_inline_js();\n  } else {\n    load_libs(css_urls, js_urls, function() {\n    console.debug(\"Bokeh: BokehJS plotting callback run at\", now());\n    run_inline_js();\n    });\n  }\n}(window));"

""
```

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.

```
from bokeh.io import output_notebook
from bokeh.transform import factor_cmap

np.random.seed(42)
n = 100
```

```
x = np.random.uniform(0, 10, n)
y = np.random.uniform(0, 10, n)
sizes = np.random.uniform(5, 50, n)
colors = np.random.choice(['red', 'green', 'blue', 'orange',
'purple'], n)

data = pd.DataFrame({'x': x, 'y': y, 'sizes': sizes, 'colors':
colors})

plot = figure(title="Scatter Plot with Different Sizes and Colors",
x_axis_label='X', y_axis_label='Y')
plot.scatter(
    x='x',
    y='y',
    size='sizes',
    color='colors',
    source=data,
    fill_alpha=0.6,
    legend_field='colors',
    line_color=None
)

output_notebook()
show(plot)
```

```
"'use strict';\n(function(root) {\n  function now() {\n    return new
Date();\n  }\n\n  const force = true;\n\n  if (typeof
root._bokeh_onload_callbacks === \"undefined\" || force === true) {\n
root._bokeh_onload_callbacks = [];\n    root._bokeh_is_loading =
undefined;\n  }\n\n\n  if (typeof (root._bokeh_timeout) ===
\"undefined\" || force === true) {\n    root._bokeh_timeout =
Date.now() + 5000;\n    root._bokeh_failed_load = false;\n  }\n\n
const NB_LOAD_WARNING = {'data': {'text/html':\n     \"<div
style='background-color: #fdd'>\\n\"+\n     \"<p>\\n\"+\n
\"BokehJS does not appear to have successfully loaded. If loading
BokehJS from CDN, this \\n\"+\n     \"may be due to a slow or bad
network connection. Possible fixes:\\n\"+\n     \"</p>\\n\"+\n
\"<ul>\\n\"+\n     \"<li>re-rerun `output_notebook()` to attempt to
load from CDN again, or</li>\\n\"+\n     \"<li>use INLINE resources
instead, as so:</li>\\n\"+\n     \"</ul>\\n\"+\n     \"<code>\\n\"+\n
\"from bokeh.resources import INLINE\\n\"+\n
\"output_notebook(resources=INLINE)\\n\"+\n     \"</code>\\n\"+\n
\"</div>\"}};\n\n  function display_loaded(error = null) {\n    const
el = document.getElementById(null);\n    if (el != null) {\n
const html = (() => {\n        if (typeof root.Bokeh ===
\"undefined\") {\n          if (error == null) {\n            return
\"BokehJS is loading ...\";\n          } else {\n            return
\"BokehJS failed to load.\";\n          }\n        } else {\n
const prefix = `BokehJS ${root.Bokeh.version}`;\n          if (error
== null) {\n            return `${prefix} successfully loaded.`;\n
```

```
} else {\n            return `${prefix} <b>encountered errors</b>
while loading and may not function as expected.`;\n          }\n
}\n      })();\n      el.innerHTML = html;\n\n      if (error != null)
{\n        const wrapper = document.createElement(\"div\");\n
wrapper.style.overflow = \"auto\";\n        wrapper.style.height =
\"5em\";\n        wrapper.style.resize = \"vertical\";\n        const
content = document.createElement(\"div\");\n
content.style.fontFamily = \"monospace\";\n
content.style.whiteSpace = \"pre-wrap\";\n
content.style.backgroundColor = \"rgb(255, 221, 221)\";\n
content.textContent = error.stack ?? error.toString();\n
wrapper.append(content);\n        el.append(wrapper);\n      }\n    }
else if (Date.now() < root._bokeh_timeout) {\n      setTimeout(() =>
display_loaded(error), 100);\n    }\n  }\n\n  function run_callbacks()
{\n    try {\n
root._bokeh_onload_callbacks.forEach(function(callback) {\n        if
(callback != null)\n          callback();\n      });\n    } finally {\
n      delete root._bokeh_onload_callbacks\n    }\n
console.debug(\"Bokeh: all callbacks have finished\");\n  }\n\n
function load_libs(css_urls, js_urls, callback) {\n    if (css_urls ==
null) css_urls = [];\n    if (js_urls == null) js_urls = [];\n\n
root._bokeh_onload_callbacks.push(callback);\n    if
(root._bokeh_is_loading > 0) {\n      console.debug(\"Bokeh: BokehJS
is being loaded, scheduling callback at\", now());\n      return
null;\n    }\n    if (js_urls == null || js_urls.length === 0) {\n
run_callbacks();\n      return null;\n    }\n
console.debug(\"Bokeh: BokehJS not loaded, scheduling load and
callback at\", now());\n    root._bokeh_is_loading = css_urls.length +
js_urls.length;\n\n    function on_load() {\n
root._bokeh_is_loading--;\n      if (root._bokeh_is_loading === 0) {\n
console.debug(\"Bokeh: all BokehJS libraries/stylesheets loaded\");\n
run_callbacks()\n      }\n    }\n\n    function on_error(url) {\n
console.error(\"failed to load \" + url);\n    }\n\n    for (let i =
0; i < css_urls.length; i++) {\n      const url = css_urls[i];\n
const element = document.createElement(\"link\");\n
element.onload = on_load;\n      element.onerror = on_error.bind(null,
url);\n      element.rel = \"stylesheet\";\n      element.type =
\"text/css\";\n      element.href = url;\n      console.debug(\"Bokeh:
injecting link tag for BokehJS stylesheet: \", url);\n
document.body.appendChild(element);\n    }\n\n    for (let i = 0; i <
js_urls.length; i++) {\n      const url = js_urls[i];\n      const
element = document.createElement('script');\n      element.onload =
on_load;\n      element.onerror = on_error.bind(null, url);\n
element.async = false;\n      element.src = url;\n
console.debug(\"Bokeh: injecting script tag for BokehJS library: \",
url);\n      document.head.appendChild(element);\n    }\n  };\n\n
function inject_raw_css(css) {\n    const element =
document.createElement(\"style\");\n
element.appendChild(document.createTextNode(css));\n
```

```
document.body.appendChild(element);\n  }\n\n  const js_urls =
[\"https://cdn.bokeh.org/bokeh/release/bokeh-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-gl-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-widgets-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-tables-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-mathjax-3.4.3.min.js\"];\n
const css_urls = [];\n\n  const inline_js = [    function(Bokeh) {\n
Bokeh.set_log_level(\"info\");\n    },\nfunction(Bokeh) {\n    }\
n  ];\n\n  function run_inline_js() {\n    if (root.Bokeh !==
undefined || force === true) {\n      try {\n            for (let i =
0; i < inline_js.length; i++) {\n      inline_js[i].call(root,
root.Bokeh);\n    }\n\n      } catch (error) {throw error;\n      }}
else if (Date.now() < root._bokeh_timeout) {\n
setTimeout(run_inline_js, 100);\n    } else if (!
root._bokeh_failed_load) {\n      console.log(\"Bokeh: BokehJS failed
to load within specified timeout.\");\n      root._bokeh_failed_load =
true;\n    } else if (force !== true) {\n      const cell = $
(document.getElementById(null)).parents('.cell').data().cell;\n
cell.output_area.append_execute_result(NB_LOAD_WARNING)\n    }\n  }\n\
n  if (root._bokeh_is_loading === 0) {\n    console.debug(\"Bokeh:
BokehJS loaded, going straight to plotting\");\n    run_inline_js();\n
} else {\n    load_libs(css_urls, js_urls, function() {\n
console.debug(\"Bokeh: BokehJS plotting callback run at\", now());\n
run_inline_js();\n    });\n  }\n}(window));"

""
```

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]

data = pd.DataFrame({'fruit': fruits, 'quantity': counts})

plot = figure(x_range=fruits, title="Counts of Different Fruits",
x_axis_label='Fruit', y_axis_label='Quantity')
plot.vbar(
    x='fruit',
    top='quantity',
    source=data,
    width=0.5,
    fill_alpha=0.6,
    color=factor_cmap('fruit', palette=['blue', 'green', 'orange',
'red', 'purple'], factors=fruits)
)
```

```
output_notebook()
show(plot)
```

```
"'use strict';\n(function(root) {\n  function now() {\n    return new
Date();\n  }\n\n  const force = true;\n\n  if (typeof
root._bokeh_onload_callbacks === \"undefined\" || force === true) {\n
root._bokeh_onload_callbacks = [];\n    root._bokeh_is_loading =
undefined;\n  }\n\n\n  if (typeof (root._bokeh_timeout) ===
\"undefined\" || force === true) {\n    root._bokeh_timeout =
Date.now() + 5000;\n    root._bokeh_failed_load = false;\n  }\n\n
const NB_LOAD_WARNING = {'data': {'text/html':\n     \"<div
style='background-color: #fdd'>\\n\"+\n     \"<p>\\n\"+\n
\"BokehJS does not appear to have successfully loaded. If loading
BokehJS from CDN, this \\n\"+\n     \"may be due to a slow or bad
network connection. Possible fixes:\\n\"+\n     \"</p>\\n\"+\n
\"<ul>\\n\"+\n     \"<li>re-rerun `output_notebook()` to attempt to
load from CDN again, or</li>\\n\"+\n     \"<li>use INLINE resources
instead, as so:</li>\\n\"+\n     \"</ul>\\n\"+\n     \"<code>\\n\"+\n
\"from bokeh.resources import INLINE\\n\"+\n
\"output_notebook(resources=INLINE)\\n\"+\n     \"</code>\\n\"+\n
\"</div>\"}};\n\n  function display_loaded(error = null) {\n    const
el = document.getElementById(null);\n    if (el != null) {\n
const html = (() => {\n        if (typeof root.Bokeh ===
\"undefined\") {\n          if (error == null) {\n            return
\"BokehJS is loading ...\";\n          } else {\n            return
\"BokehJS failed to load.\";\n          }\n        } else {\n
const prefix = `BokehJS ${root.Bokeh.version}`;\n          if (error
== null) {\n            return `${prefix} successfully loaded.`;\n
} else {\n            return `${prefix} <b>encountered errors</b>
while loading and may not function as expected.`;\n          }\n
}\n      })();\n      el.innerHTML = html;\n\n      if (error != null)
{\n        const wrapper = document.createElement(\"div\");\n
wrapper.style.overflow = \"auto\";\n        wrapper.style.height =
\"5em\";\n        wrapper.style.resize = \"vertical\";\n        const
content = document.createElement(\"div\");\n
content.style.fontFamily = \"monospace\";\n
content.style.whiteSpace = \"pre-wrap\";\n
content.style.backgroundColor = \"rgb(255, 221, 221)\";\n
content.textContent = error.stack ?? error.toString();\n
wrapper.append(content);\n        el.append(wrapper);\n      }\n    }
else if (Date.now() < root._bokeh_timeout) {\n      setTimeout(() =>
display_loaded(error), 100);\n    }\n  }\n\n  function run_callbacks()
{\n    try {\n
root._bokeh_onload_callbacks.forEach(function(callback) {\n        if
(callback != null)\n          callback();\n      });\n    } finally {\
n      delete root._bokeh_onload_callbacks\n    }\n
console.debug(\"Bokeh: all callbacks have finished\");\n  }\n\n
function load_libs(css_urls, js_urls, callback) {\n    if (css_urls ==
null) css_urls = [];\n    if (js_urls == null) js_urls = [];\n\n
root._bokeh_onload_callbacks.push(callback);\n    if
```

```
(root._bokeh_is_loading > 0) {\n      console.debug(\"Bokeh: BokehJS
is being loaded, scheduling callback at\", now());\n      return
null;\n    }\n    if (js_urls == null || js_urls.length === 0) {\n
run_callbacks();\n      return null;\n    }\n
console.debug(\"Bokeh: BokehJS not loaded, scheduling load and
callback at\", now());\n    root._bokeh_is_loading = css_urls.length +
js_urls.length;\n\n    function on_load() {\n
root._bokeh_is_loading--;\n      if (root._bokeh_is_loading === 0) {\n
console.debug(\"Bokeh: all BokehJS libraries/stylesheets loaded\");\n
run_callbacks()\n      }\n    }\n\n    function on_error(url) {\n
console.error(\"failed to load \" + url);\n    }\n\n    for (let i =
0; i < css_urls.length; i++) {\n      const url = css_urls[i];\n
const element = document.createElement(\"link\");\n
element.onload = on_load;\n      element.onerror = on_error.bind(null,
url);\n      element.rel = \"stylesheet\";\n      element.type =
\"text/css\";\n      element.href = url;\n      console.debug(\"Bokeh:
injecting link tag for BokehJS stylesheet: \", url);\n
document.body.appendChild(element);\n    }\n\n    for (let i = 0; i <
js_urls.length; i++) {\n      const url = js_urls[i];\n      const
element = document.createElement('script');\n      element.onload =
on_load;\n      element.onerror = on_error.bind(null, url);\n
element.async = false;\n      element.src = url;\n
console.debug(\"Bokeh: injecting script tag for BokehJS library: \",
url);\n      document.head.appendChild(element);\n    }\n  };\n\n
function inject_raw_css(css) {\n    const element =
document.createElement(\"style\");\n
element.appendChild(document.createTextNode(css));\n
document.body.appendChild(element);\n  }\n\n  const js_urls =
[\"https://cdn.bokeh.org/bokeh/release/bokeh-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-gl-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-widgets-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-tables-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-mathjax-3.4.3.min.js\"];\n
const css_urls = [];\n\n  const inline_js = [    function(Bokeh) {\n
Bokeh.set_log_level(\"info\");\n    },\nfunction(Bokeh) {\n    }\
n  ];\n\n  function run_inline_js() {\n    if (root.Bokeh !==
undefined || force === true) {\n      try {\n            for (let i =
0; i < inline_js.length; i++) {\n      inline_js[i].call(root,
root.Bokeh);\n    }\n\n      } catch (error) {throw error;\n      }}
else if (Date.now() < root._bokeh_timeout) {\n
setTimeout(run_inline_js, 100);\n    } else if (!
root._bokeh_failed_load) {\n      console.log(\"Bokeh: BokehJS failed
to load within specified timeout.\");\n      root._bokeh_failed_load =
true;\n    } else if (force !== true) {\n      const cell = $
(document.getElementById(null)).parents('.cell').data().cell;\n
cell.output_area.append_execute_result(NB_LOAD_WARNING)\n    }\n  }\n\
n  if (root._bokeh_is_loading === 0) {\n    console.debug(\"Bokeh:
BokehJS loaded, going straight to plotting\");\n    run_inline_js();\n
} else {\n    load_libs(css_urls, js_urls, function() {\n
```

```
console.debug(\"Bokeh: BokehJS plotting callback run at\", now());\n
run_inline_js();\n    });\n  }\n}(window));"

""
```

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)

```
data_hist = np.random.randn(1000)
hist, edges = np.histogram(data_hist, bins=30)

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


hist_data = pd.DataFrame({'left': edges[:-1], 'right': edges[1:],
'top': hist})

plot = figure(title="Histogram of Random Data", x_axis_label='Value',
y_axis_label='Frequency')

plot.quad(
    top='top',
    bottom=0,
    left='left',
    right='right',
    source=hist_data,
    fill_alpha=0.6,
    line_color='black'
)

output_notebook()
show(plot)

"'use strict';\n(function(root) {\n  function now() {\n    return new
Date();\n  }\n\n  const force = true;\n\n  if (typeof
root._bokeh_onload_callbacks === \"undefined\" || force === true) {\n
root._bokeh_onload_callbacks = [];\n    root._bokeh_is_loading =
undefined;\n  }\n\n\n  if (typeof (root._bokeh_timeout) ===
\"undefined\" || force === true) {\n    root._bokeh_timeout =
Date.now() + 5000;\n    root._bokeh_failed_load = false;\n  }\n\n
const NB_LOAD_WARNING = {'data': {'text/html':\n     \"<div
style='background-color: #fdd'>\\n\"+\n     \"<p>\\n\"+\n
\"BokehJS does not appear to have successfully loaded. If loading
BokehJS from CDN, this \\n\"+\n     \"may be due to a slow or bad
network connection. Possible fixes:\\n\"+\n     \"</p>\\n\"+\n
```

```
\"<ul>\\n\"+\n     \"<li>re-rerun `output_notebook()` to attempt to
load from CDN again, or</li>\\n\"+\n     \"<li>use INLINE resources
instead, as so:</li>\\n\"+\n     \"</ul>\\n\"+\n     \"<code>\\n\"+\n
\"from bokeh.resources import INLINE\\n\"+\n
\"output_notebook(resources=INLINE)\\n\"+\n     \"</code>\\n\"+\n
\"</div>\"}};\n\n  function display_loaded(error = null) {\n    const
el = document.getElementById(null);\n    if (el != null) {\n
const html = (() => {\n        if (typeof root.Bokeh ===
\"undefined\") {\n          if (error == null) {\n            return
\"BokehJS is loading ...\";\n          } else {\n            return
\"BokehJS failed to load.\";\n          }\n        } else {\n
const prefix = `BokehJS ${root.Bokeh.version}`;\n          if (error
== null) {\n            return `${prefix} successfully loaded.`;\n
} else {\n            return `${prefix} <b>encountered errors</b>
while loading and may not function as expected.`;\n          }\n
}\n      })();\n      el.innerHTML = html;\n\n      if (error != null)
{\n        const wrapper = document.createElement(\"div\");\n
wrapper.style.overflow = \"auto\";\n        wrapper.style.height =
\"5em\";\n        wrapper.style.resize = \"vertical\";\n        const
content = document.createElement(\"div\");\n
content.style.fontFamily = \"monospace\";\n
content.style.whiteSpace = \"pre-wrap\";\n
content.style.backgroundColor = \"rgb(255, 221, 221)\";\n
content.textContent = error.stack ?? error.toString();\n
wrapper.append(content);\n        el.append(wrapper);\n      }\n    }
else if (Date.now() < root._bokeh_timeout) {\n      setTimeout(() =>
display_loaded(error), 100);\n    }\n  }\n\n  function run_callbacks()
{\n    try {\n
root._bokeh_onload_callbacks.forEach(function(callback) {\n        if
(callback != null)\n          callback();\n      });\n    } finally {\
n      delete root._bokeh_onload_callbacks\n    }\n
console.debug(\"Bokeh: all callbacks have finished\");\n  }\n\n
function load_libs(css_urls, js_urls, callback) {\n    if (css_urls ==
null) css_urls = [];\n    if (js_urls == null) js_urls = [];\n\n
root._bokeh_onload_callbacks.push(callback);\n    if
(root._bokeh_is_loading > 0) {\n      console.debug(\"Bokeh: BokehJS
is being loaded, scheduling callback at\", now());\n      return
null;\n    }\n    if (js_urls == null || js_urls.length === 0) {\n
run_callbacks();\n      return null;\n    }\n
console.debug(\"Bokeh: BokehJS not loaded, scheduling load and
callback at\", now());\n    root._bokeh_is_loading = css_urls.length +
js_urls.length;\n\n    function on_load() {\n
root._bokeh_is_loading--;\n      if (root._bokeh_is_loading === 0) {\n
console.debug(\"Bokeh: all BokehJS libraries/stylesheets loaded\");\n
run_callbacks()\n      }\n    }\n\n    function on_error(url) {\n
console.error(\"failed to load \" + url);\n    }\n\n    for (let i =
0; i < css_urls.length; i++) {\n      const url = css_urls[i];\n
const element = document.createElement(\"link\");\n
element.onload = on_load;\n      element.onerror = on_error.bind(null,
```

```
url);\n      element.rel = \"stylesheet\";\n      element.type =
\"text/css\";\n      element.href = url;\n      console.debug(\"Bokeh:
injecting link tag for BokehJS stylesheet: \", url);\n
document.body.appendChild(element);\n    }\n\n    for (let i = 0; i <
js_urls.length; i++) {\n      const url = js_urls[i];\n      const
element = document.createElement('script');\n      element.onload =
on_load;\n      element.onerror = on_error.bind(null, url);\n
element.async = false;\n      element.src = url;\n
console.debug(\"Bokeh: injecting script tag for BokehJS library: \",
url);\n      document.head.appendChild(element);\n    }\n  };\n\n
function inject_raw_css(css) {\n    const element =
document.createElement(\"style\");\n
element.appendChild(document.createTextNode(css));\n
document.body.appendChild(element);\n  }\n\n  const js_urls =
[\"https://cdn.bokeh.org/bokeh/release/bokeh-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-gl-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-widgets-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-tables-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-mathjax-3.4.3.min.js\"];\n
const css_urls = [];\n\n  const inline_js = [    function(Bokeh) {\n
Bokeh.set_log_level(\"info\");\n    },\nfunction(Bokeh) {\n    }\
n  ];\n\n  function run_inline_js() {\n    if (root.Bokeh !==
undefined || force === true) {\n      try {\n            for (let i =
0; i < inline_js.length; i++) {\n      inline_js[i].call(root,
root.Bokeh);\n    }\n\n      } catch (error) {throw error;\n      }}
else if (Date.now() < root._bokeh_timeout) {\n
setTimeout(run_inline_js, 100);\n    } else if (!
root._bokeh_failed_load) {\n      console.log(\"Bokeh: BokehJS failed
to load within specified timeout.\");\n      root._bokeh_failed_load =
true;\n    } else if (force !== true) {\n      const cell = $
(document.getElementById(null)).parents('.cell').data().cell;\n
cell.output_area.append_execute_result(NB_LOAD_WARNING)\n    }\n  }\n\
n  if (root._bokeh_is_loading === 0) {\n    console.debug(\"Bokeh:
BokehJS loaded, going straight to plotting\");\n    run_inline_js();\n
} else {\n    load_libs(css_urls, js_urls, function() {\n
console.debug(\"Bokeh: BokehJS plotting callback run at\", now());\n
run_inline_js();\n    });\n  }\n}(window));"

""
```

5. Create a Bokeh heatmap using the provided dataset.

data_heatmap = np.random.rand(10, 10) x = np.linspace(0, 1, 10) y = np.linspace(0, 1, 10) xx, yy = np.meshgrid(x, y)

```
from bokeh.transform import linear_cmap

data_heatmap = np.random.rand(10, 10)
x = np.linspace(0, 1, 10)
```

```
y = np.linspace(0, 1, 10)
xx, yy = np.meshgrid(x, y)

data_flat = {
    'x': xx.flatten(),
    'y': yy.flatten(),
    'value': data_heatmap.flatten()
}

data_df = pd.DataFrame(data_flat)

plot = figure(
    title="Heatmap",
    x_axis_label='X',
    y_axis_label='Y',
    x_range=(0, 1),
    y_range=(0, 1),
    tools=""
)

plot.rect(
    x='x',
    y='y',
    width=0.1,
    height=0.1,
    source=data_df,
    fill_color=linear_cmap('value', palette='Viridis256',
low=data_heatmap.min(), high=data_heatmap.max()),
    line_color=None
)

output_notebook()
show(plot)
```

```
"'use strict';\n(function(root) {\n  function now() {\n    return new
Date();\n  }\n\n  const force = true;\n\n  if (typeof
root._bokeh_onload_callbacks === \"undefined\" || force === true) {\n
root._bokeh_onload_callbacks = [];\n    root._bokeh_is_loading =
undefined;\n  }\n\n\n  if (typeof (root._bokeh_timeout) ===
\"undefined\" || force === true) {\n    root._bokeh_timeout =
Date.now() + 5000;\n    root._bokeh_failed_load = false;\n  }\n\n
const NB_LOAD_WARNING = {'data': {'text/html':\n     \"<div
style='background-color: #fdd'>\\n\"+\n     \"<p>\\n\"+\n
\"BokehJS does not appear to have successfully loaded. If loading
BokehJS from CDN, this \\n\"+\n     \"may be due to a slow or bad
network connection. Possible fixes:\\n\"+\n     \"</p>\\n\"+\n
\"<ul>\\n\"+\n     \"<li>re-rerun `output_notebook()` to attempt to
load from CDN again, or</li>\\n\"+\n     \"<li>use INLINE resources
instead, as so:</li>\\n\"+\n     \"</ul>\\n\"+\n     \"<code>\\n\"+\n
\"from bokeh.resources import INLINE\\n\"+\n
```

```
\"output_notebook(resources=INLINE)\\n\"+\n     \"</code>\\n\"+\n
\"</div>\"}};\n\n  function display_loaded(error = null) {\n    const
el = document.getElementById(null);\n    if (el != null) {\n
const html = (() => {\n        if (typeof root.Bokeh ===
\"undefined\") {\n          if (error == null) {\n            return
\"BokehJS is loading ...\";\n          } else {\n            return
\"BokehJS failed to load.\";\n          }\n        } else {\n
const prefix = `BokehJS ${root.Bokeh.version}`;\n          if (error
== null) {\n            return `${prefix} successfully loaded.`;\n
} else {\n            return `${prefix} <b>encountered errors</b>
while loading and may not function as expected.`;\n          }\n
}\n      })();\n      el.innerHTML = html;\n\n      if (error != null)
{\n        const wrapper = document.createElement(\"div\");\n
wrapper.style.overflow = \"auto\";\n        wrapper.style.height =
\"5em\";\n        wrapper.style.resize = \"vertical\";\n        const
content = document.createElement(\"div\");\n
content.style.fontFamily = \"monospace\";\n
content.style.whiteSpace = \"pre-wrap\";\n
content.style.backgroundColor = \"rgb(255, 221, 221)\";\n
content.textContent = error.stack ?? error.toString();\n
wrapper.append(content);\n        el.append(wrapper);\n      }\n    }
else if (Date.now() < root._bokeh_timeout) {\n      setTimeout(() =>
display_loaded(error), 100);\n    }\n  }\n\n  function run_callbacks()
{\n    try {\n
root._bokeh_onload_callbacks.forEach(function(callback) {\n        if
(callback != null)\n          callback();\n      });\n    } finally {\
n      delete root._bokeh_onload_callbacks\n    }\n
console.debug(\"Bokeh: all callbacks have finished\");\n  }\n\n
function load_libs(css_urls, js_urls, callback) {\n    if (css_urls ==
null) css_urls = [];\n    if (js_urls == null) js_urls = [];\n\n
root._bokeh_onload_callbacks.push(callback);\n    if
(root._bokeh_is_loading > 0) {\n      console.debug(\"Bokeh: BokehJS
is being loaded, scheduling callback at\", now());\n      return
null;\n    }\n    if (js_urls == null || js_urls.length === 0) {\n
run_callbacks();\n      return null;\n    }\n
console.debug(\"Bokeh: BokehJS not loaded, scheduling load and
callback at\", now());\n    root._bokeh_is_loading = css_urls.length +
js_urls.length;\n\n    function on_load() {\n
root._bokeh_is_loading--;\n      if (root._bokeh_is_loading === 0) {\n
console.debug(\"Bokeh: all BokehJS libraries/stylesheets loaded\");\n
run_callbacks()\n      }\n    }\n\n    function on_error(url) {\n
console.error(\"failed to load \" + url);\n    }\n\n    for (let i =
0; i < css_urls.length; i++) {\n      const url = css_urls[i];\n
const element = document.createElement(\"link\");\n
element.onload = on_load;\n      element.onerror = on_error.bind(null,
url);\n      element.rel = \"stylesheet\";\n      element.type =
\"text/css\";\n      element.href = url;\n      console.debug(\"Bokeh:
injecting link tag for BokehJS stylesheet: \", url);\n
document.body.appendChild(element);\n    }\n\n    for (let i = 0; i <
```

```
js_urls.length; i++) {\n      const url = js_urls[i];\n      const
element = document.createElement('script');\n      element.onload =
on_load;\n      element.onerror = on_error.bind(null, url);\n
element.async = false;\n      element.src = url;\n
console.debug(\"Bokeh: injecting script tag for BokehJS library: \",
url);\n      document.head.appendChild(element);\n    }\n  };\n\n
function inject_raw_css(css) {\n    const element =
document.createElement(\"style\");\n
element.appendChild(document.createTextNode(css));\n
document.body.appendChild(element);\n  }\n\n  const js_urls =
[\"https://cdn.bokeh.org/bokeh/release/bokeh-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-gl-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-widgets-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-tables-3.4.3.min.js\",
\"https://cdn.bokeh.org/bokeh/release/bokeh-mathjax-3.4.3.min.js\"];\n
const css_urls = [];\n\n  const inline_js = [    function(Bokeh) {\n
Bokeh.set_log_level(\"info\");\n    },\nfunction(Bokeh) {\n    }\
n  ];\n\n  function run_inline_js() {\n    if (root.Bokeh !==
undefined || force === true) {\n      try {\n            for (let i =
0; i < inline_js.length; i++) {\n      inline_js[i].call(root,
root.Bokeh);\n    }\n\n      } catch (error) {throw error;\n      }}
else if (Date.now() < root._bokeh_timeout) {\n
setTimeout(run_inline_js, 100);\n    } else if (!
root._bokeh_failed_load) {\n      console.log(\"Bokeh: BokehJS failed
to load within specified timeout.\");\n      root._bokeh_failed_load =
true;\n    } else if (force !== true) {\n      const cell = $
(document.getElementById(null)).parents('.cell').data().cell;\n
cell.output_area.append_execute_result(NB_LOAD_WARNING)\n    }\n  }\n\
n  if (root._bokeh_is_loading === 0) {\n    console.debug(\"Bokeh:
BokehJS loaded, going straight to plotting\");\n    run_inline_js();\n
} else {\n    load_libs(css_urls, js_urls, function() {\n
console.debug(\"Bokeh: BokehJS plotting callback run at\", now());\n
run_inline_js();\n    });\n  }\n}(window));"

""
```