Experiment No 3

Implement scatter plots, bubble plots, waffle charts

Scatter Plot

Purpose: Visualizes the relationship between two continuous variables. Each data point represents a single observation, plotted based on its values for the two variables (X and Y).

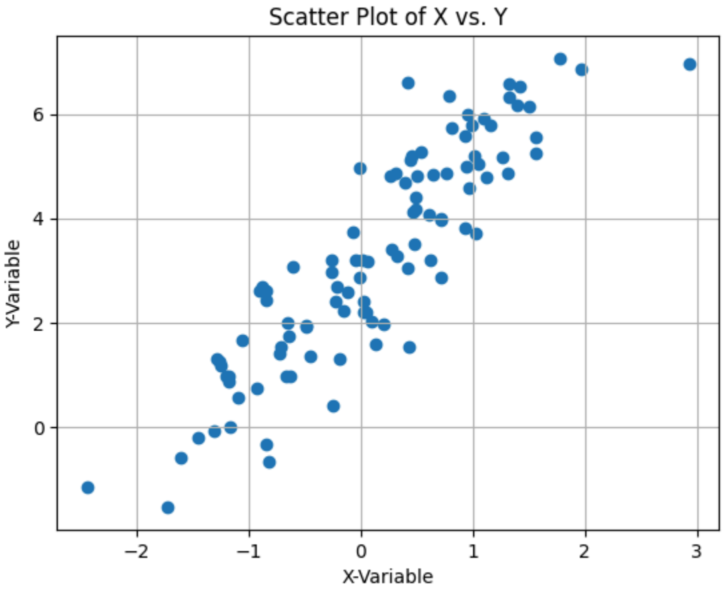
Use Cases:

- Identify correlations or trends between two variables (e.g., weight vs. height).
- Explore relationships between variables without assuming a linear relationship (unlike line graphs).
- Identify clusters or outliers in the data.

```
import matplotlib.pyplot as plt
import numpy as np

# Sample data
x = np.random.randn(100)
y = 2 * x + 3 + np.random.randn(100)  # Simulate linear relationship with noise

# Create the scatter plot
plt.scatter(x, y)
plt.xlabel('X-Variable')
plt.ylabel('Y-Variable')
plt.title('Scatter Plot of X vs. Y')
plt.grid(True)
plt.show()
```



Bubble Plots

Purpose: Similar to a scatter plot, but a third variable is used to determine the size of the bubble markers. This allows visualizing an additional dimension of data on the same plot.

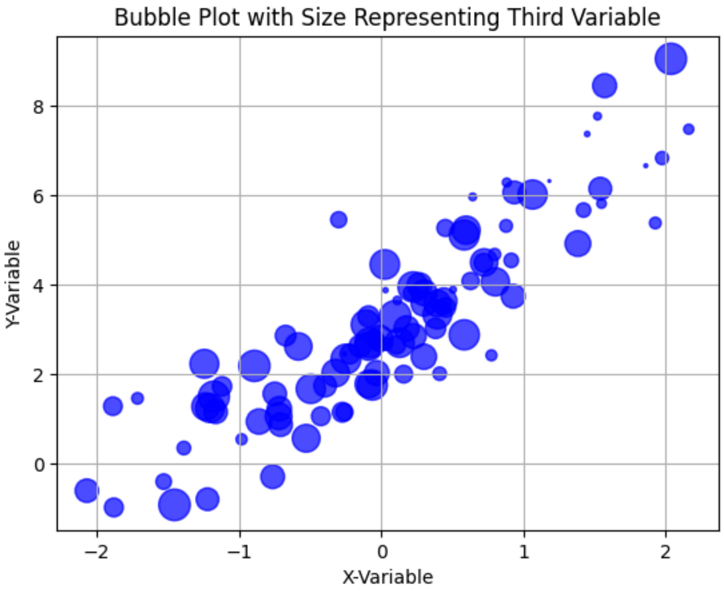
Use Cases:

- Visualize three variables simultaneously (e.g., scatter plot with bubble size representing population of a city).
- Identify trends or relationships between two variables while considering a third influencing factor (e.g., sales vs. advertising spending, with bubble size representing product category).
- Be cautious with too many bubbles, as they can overlap and obscure information.

```
import matplotlib.pyplot as plt
import numpy as np

# Sample data (ensure same length for all variables)
x = np.random.randn(100)
y = 2 * x + 3 + np.random.randn(100)
bubble_size = np.random.rand(100) * 300  # Adjust scaling factor for bubble size

# Create the bubble plot
plt.scatter(x, y, s=bubble_size, c='blue', alpha=0.7)  # Adjust color and transparency
plt.xlabel('X-Variable')
plt.ylabel('Y-Variable')
plt.title('Bubble Plot with Size Representing Third Variable')
plt.grid(True)
plt.show()
```



Waffle Charts

Purpose: Visualizes categorical data using a grid of squares. Each square represents a category, and the squares are filled (often shaded) proportionally to the data values.

Use Cases:

- Visualize the distribution of categorical data in a visually appealing way.
- Compare the proportions of different categories within a dataset (similar to a pie chart, but with better use of space for many categories).
- Effective for displaying percentages or ratios where exact precision is not crucial.

```
from pywaffle import Waffle
import matplotlib.pyplot as plt

# Data
value = [12, 22, 16, 38, 12]

# Waffle chart
plt.figure(
    FigureClass = Waffle,
    rows = 10,
    columns = 10,
    values = value,
    facecolor = 'whitesmoke')

plt.show()
```