

How To Use: Data Visualization

Tools, and resources used: Python Crash Course, Third Edition By Eric Matthes, Chat GPT 4.0

How it works: For the most part we use the matplotlib.pyplot library to visualize our code.

The most important methods of the library to are:

1.- ax.plot: This plots the information we give it in a number line format.

2.- matplotlib.pyplot.show(): This give us an output of the visualization

3.- ax.scatter: This plots the information we give it in a vector format. You can use x, and y attributes as well as adjust the color of the plot.

4.- matplotlib.pyplot.bar(): This creates the output in a bar graph format.

Here are some of the programs we did and their outputs:

Die_visual.py:

```
from die import Die

import plotly.express as px

# Create two D6 dice.
die_1 = Die()
die_2 = Die(10)

# Make some rolls, and store results in a list.
results = []
for roll_num in range(50_000):
    result = die_1.roll() + die_2.roll()
    results.append(result)

# Analyze the results.
frequencies = []
max_result = die_1.num_sides + die_2.num_sides
poss_results = range(2, max_result+1)
for value in poss_results:
    frequency = results.count(value)
    frequencies.append(frequency)
```

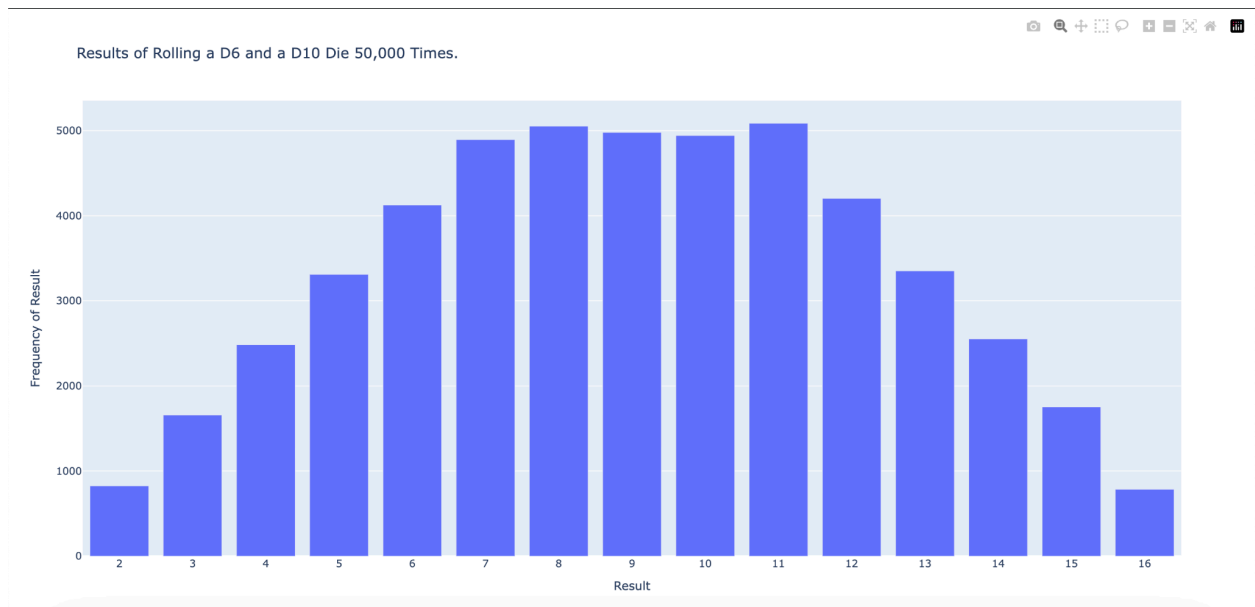
```

# Visualize the results.
title = "Results of Rolling a D6 and a D10 Die 50,000 Times."
labels = {'x': 'Result', 'y': 'Frequency of Result'}
fig = px.bar(x=pos_results, y=frequencies, title=title, labels=labels)

# Further customize chart.
fig.update_layout(xaxis_dtick=1)
fig.show()

```

Output:



Mql_square.py

```

import matplotlib.pyplot as plt

input_values = [1, 2, 3, 4, 5]
squares = [1, 4, 9, 16, 25]

# fig = figure, ax = single plot, plot = plots the data
plt.style.use('seaborn-v0_8')
fig, ax = plt.subplots()
ax.plot(input_values, squares, linewidth = 3)

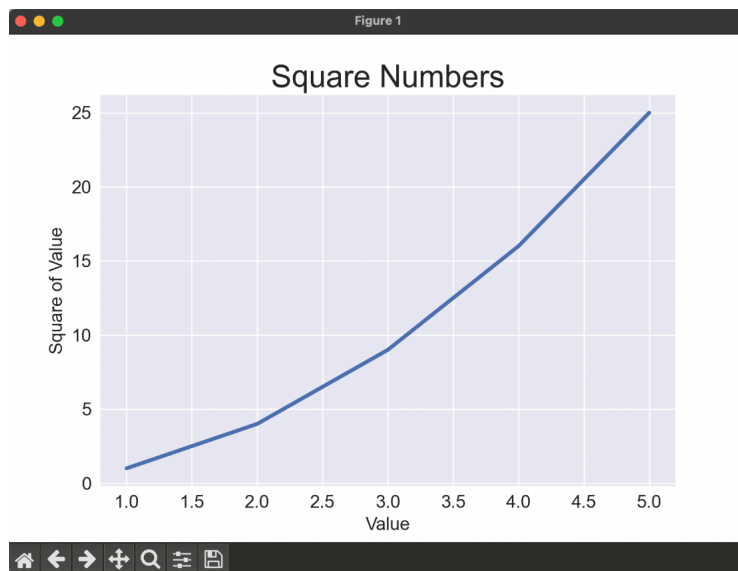
```

```
# Set char title and label axes.
ax.set_title("Square Numbers", fontsize=24)
ax.set_xlabel("Value", fontsize=14)
ax.set_ylabel("Square of Value", fontsize=14)

# Set size of tick labels.
ax.tick_params(labelsize=14)

# Displays the plot
plt.show()
```

Output:



Rw_visual.py

```
import matplotlib.pyplot as plt

from random_walk import RandomWalk

# Keep making new walks, as long as the program is active.
while True:
    # Make a random walk.
    rw = RandomWalk()
    rw = RandomWalk(25_000)
```

```

rw.fill_walk()

# Plot the points in the walk.
plt.style.use('classic')
fig, ax = plt.subplots(figsize=(10,6), dpi=128)
point_numbers = range(rw.num_points)
ax.scatter(rw.x_values, rw.y_values, c=point_numbers,
cmap=plt.cm.Blues, edgecolors='none', s=1)
ax.set_aspect('equal')

# Emphasize the first and last points.
ax.scatter(0, 0, c='green', edgecolors='none', s=100)
ax.scatter(rw.x_values[-1], rw.y_values[-1], c='red', edgecolors='none', s=100)

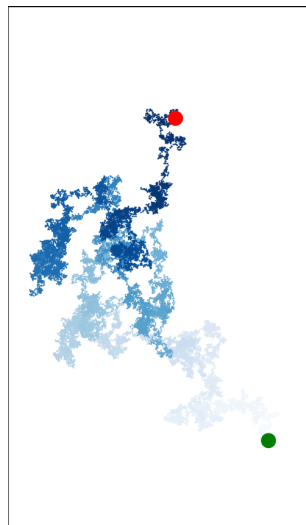
# Remove the axis.
ax.get_xaxis().set_visible(False)
ax.get_yaxis().set_visible(False)

plt.show()

keep_running = input("Make another walk? (y/n): ")
if keep_running == 'n':
    break

```

Output:



Scatter_square.py

```
import matplotlib.pyplot as plt

x_values = range(1, 1001)
y_values = [x**2 for x in x_values]

plt.style.use('seaborn-v0_8')
fig, ax = plt.subplots()
ax.scatter(x_values, y_values, c=y_values, cmap=plt.cm.Blues, s=10)

# Set chart title and label axes.
ax.set_title("Square Numbers", fontsize=24)
ax.set_xlabel("Value", fontsize=14)
ax.set_ylabel("Square of Value", fontsize=14)

# Set size of tick labels.
ax.tick_params(labelsize=14)

# Set the range for each axis.
ax.axis([0, 1100, 0, 1_100_000])

plt.show()
```

Output:

