…………………………………………………………Assignment………………………………………………………………

1. Analyze the relationship between the size of houses (measured in square footage) and their selling prices in a particular neighborhood. You have collected data on various houses in that neighborhood.Create a scatter plot using the below data and share your conclusion/analysis.

Input:

square\_footage = np.array([1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600, 2800, 3000])

selling\_prices = np.array([250, 290, 315, 380, 410, 450, 500, 525, 570, 610])

//code

import matplotlib.pyplot as plt

import numpy as np

# Data

square\_footage = np.array([1200, 1400, 1600, 1800, 2000, 2200, 2400, 2600, 2800, 3000])

selling\_prices = np.array([250, 290, 315, 380, 410, 450, 500, 525, 570, 610])

# Plotting

plt.figure(figsize=(10, 6))

plt.scatter(square\_footage, selling\_prices, color='blue', marker='o')

plt.title('Relationship Between House Size and Selling Price')

plt.xlabel('Square Footage')

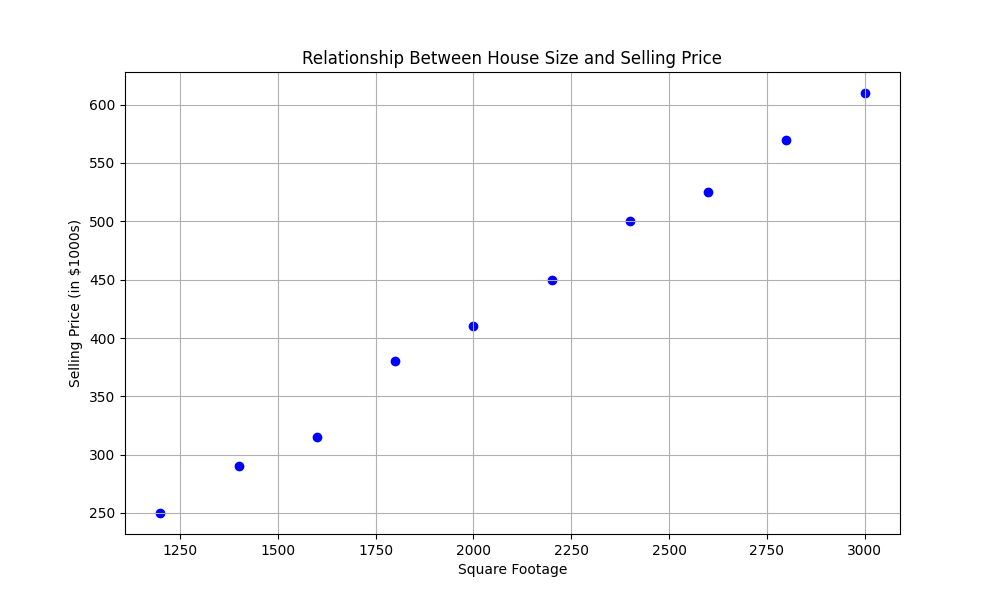
plt.ylabel('Selling Price (in $1000s)')

plt.grid(True)

# Display the plot

plt.show()

output:



…………………………………………………………………………………………………………………………………………………………..

2. Create a pie chart to visualize the distribution of your monthly income by source. You have collected data on the various sources of your income, such as salary, freelance work, investments, and rental income. Share your conclusion/analysis.

Input:

income\_sources = ['Salary', 'Freelance', 'Investments', 'Rental', 'Other']

monthly\_income = [5000, 1500, 1000, 600, 400]

//code

import matplotlib.pyplot as plt

# Data

income\_sources = ['Salary', 'Freelance', 'Investments', 'Rental', 'Other']

monthly\_income = [5000, 1500, 1000, 600, 400]

# Plotting

plt.figure(figsize=(8, 8))

plt.pie(monthly\_income, labels=income\_sources, autopct='%1.1f%%', startangle=140, colors=['blue', 'orange', 'green', 'red', 'purple'])

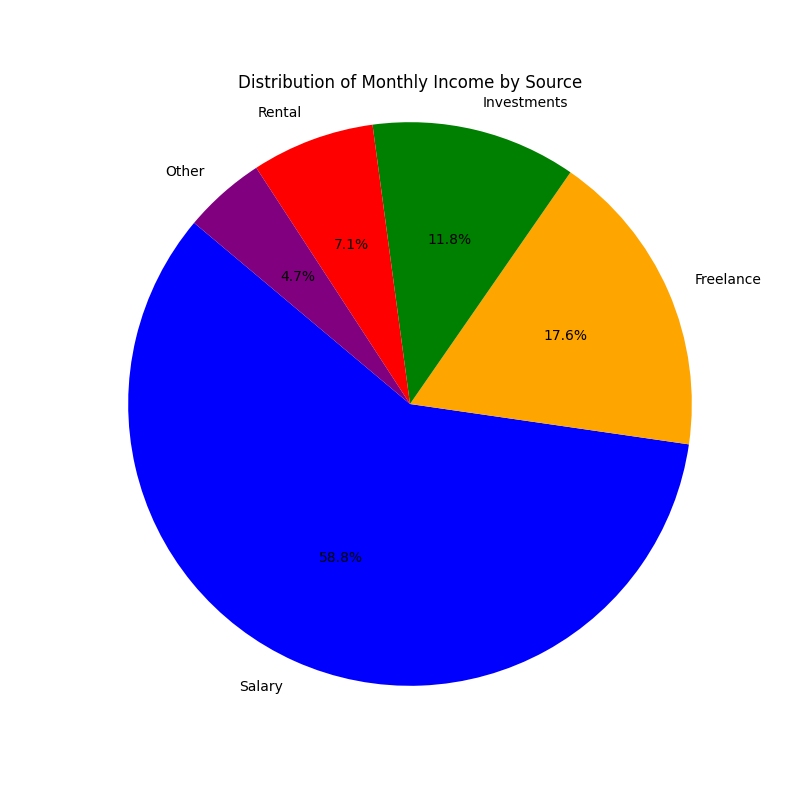
plt.title('Distribution of Monthly Income by Source')

plt.axis('equal') # Equal aspect ratio ensures that pie is drawn as a circle.

# Display the plot

plt.show()

output:



…………………………………………………………………………………………………………………………………………………………..