1: Visualize the daily temperature changes over time in a city and give your conclusion

Input: days = list(range(1, 32))

# Daily temperature data (replace with your own data)

temperature = [65, 68, 70, 72, 75, 76, 78, 80, 81, 79, 75, 72, 70, 68, 67, 69, 70, 73, 75, 76, 78, 80, 81, 82, 83, 82, 80, 78, 76, 74, 71]

Code:

import matplotlib.pyplot as plt

days = list(range(1, 32))

temperature = [65, 68, 70, 72, 75, 76, 78, 80, 81, 79, 75, 72, 70, 68, 67, 69, 70, 73, 75, 76, 78, 80, 81, 82, 83, 82, 80, 78, 76, 74, 71]

plt.figure(figsize=(10, 5))

plt.plot(days, temperature, marker='o', linestyle='-', color='b')

plt.title('Daily Temperature Changes Over Time')

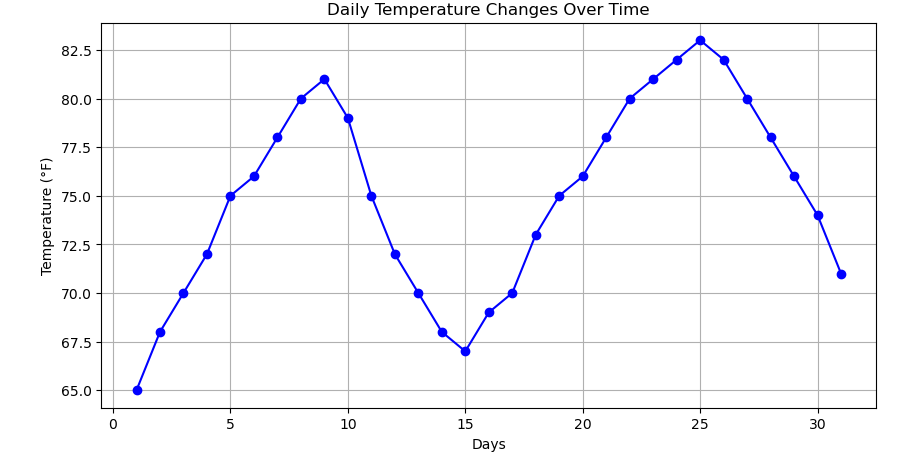
plt.xlabel('Days')

plt.ylabel('Temperature (°F)')

plt.grid(True)

plt.show()

Output:



2: Create a line plot to visualize the daily closing prices of a stock over a year and give your conclusion.

Input: days = list(range(1, 78))

# Daily closing prices of a stock (replace with your own data)

stock\_prices = [100, 105, 110, 115, 112, 120, 118, 125, 128, 130, 132, 135, 138, 140, 142, 144, 145, 148, 150, 155, 160, 158, 162, 165, 170, 172, 175, 178, 180, 182, 185, 188, 190, 192, 195, 198,200, 198, 195, 193, 190, 188, 185, 182, 180, 178, 175, 172, 170, 168, 165, 162, 160, 158, 155, 152, 150, 148, 145, 143, 140, 138, 135, 132, 130, 128, 125, 123, 120, 118, 115, 112, 110, 108, 105, 103, 100]

Code:

import matplotlib.pyplot as plt

# Data: days and corresponding stock prices

days = list(range(1, 78))

stock\_prices = [100, 105, 110, 115, 112, 120, 118, 125, 128, 130, 132, 135, 138, 140, 142, 144, 145, 148, 150, 155, 160, 158, 162, 165, 170, 172, 175, 178, 180, 182, 185, 188, 190, 192, 195, 198, 200, 198, 195, 193, 190, 188, 185, 182, 180, 178, 175, 172, 170, 168, 165, 162, 160, 158, 155, 152, 150, 148, 145, 143, 140, 138, 135, 132, 130, 128, 125, 123, 120, 118, 115, 112, 110, 108, 105, 103, 100]

# Create a plot with specified size

plt.figure(figsize=(12, 6))

# Plotting the stock price data

# days on x-axis and stock\_prices on y-axis

plt.plot(days, stock\_prices, marker='o', linestyle='-', color='g')

# Adding a title to the plot

plt.title('Daily Closing Prices of a Stock Over 78 Days')

# Labeling the x-axis

plt.xlabel('Days')

# Labeling the y-axis

plt.ylabel('Stock Prices ($)')

# Adding a grid for better readability of the plot

plt.grid(True)

# Display the plot

plt.show()



3: Create a bar chart to represent monthly expenses in different spending categories and give your conclusion.

Input: categories = ['Rent', 'Groceries', 'Utilities', 'Entertainment', 'Transportation']

# Monthly expenses in dollars (replace with your own data)

expenses = [1200, 400, 200, 150, 250]

Code:

import matplotlib.pyplot as plt

# Data: categories and corresponding expenses

categories = ['Rent', 'Groceries', 'Utilities', 'Entertainment', 'Transportation']

expenses = [1200, 400, 200, 150, 250]

# Create a bar chart

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

plt.bar(categories, expenses, color='skyblue')

# Adding a title to the chart

plt.title('Monthly Expenses by Category')

# Labeling the y-axis

plt.ylabel('Expenses ($)')

# Adding values on top of the bars

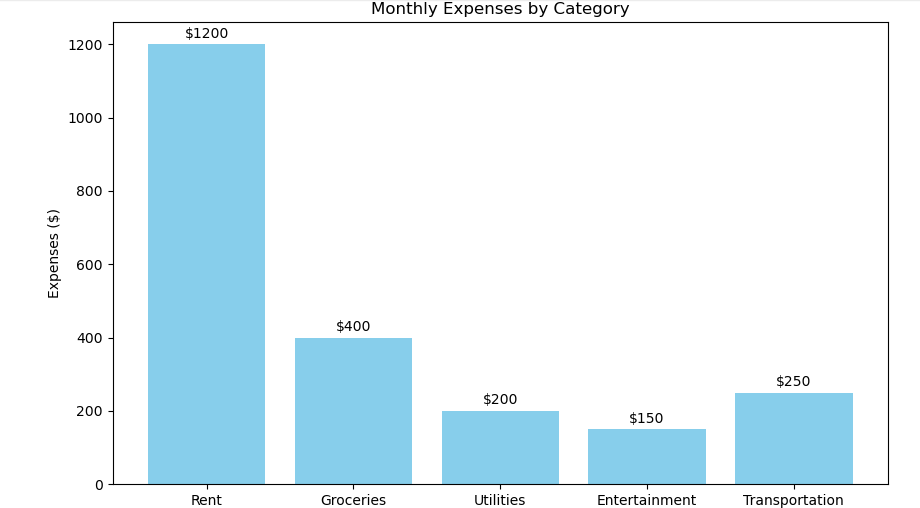
for i, expense in enumerate(expenses):

plt.text(i, expense + 10, f'${expense}', ha='center', va='bottom')

# Display the chart

plt.show()

Output:



4: Create a histogram to represent the distribution of product prices in a retail store and give your conclusion.

Input: product\_prices = [24.99, 34.99, 49.99, 64.99, 39.99, 54.99, 79.99, 99.99, 29.99, 44.99, 59.99, 69.99, 84.99, 109.99, 119.99, 89.99, 74.99, 124.99, 69.99, 54.99]

Code:

import matplotlib.pyplot as plt

# Data: product prices

product\_prices = [24.99, 34.99, 49.99, 64.99, 39.99, 54.99, 79.99, 99.99, 29.99, 44.99, 59.99, 69.99, 84.99, 109.99, 119.99, 89.99, 74.99, 124.99, 69.99, 54.99]

# Create a histogram with specified bins

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

plt.hist(product\_prices, bins=8, edgecolor='black', color='lightblue')

# Adding a title to the histogram

plt.title('Distribution of Product Prices in a Retail Store')

# Labeling the x-axis

plt.xlabel('Product Prices ($)')

# Labeling the y-axis

plt.ylabel('Frequency')

# Adding grid lines for better readability

plt.grid(True)

# Display the histogram

plt.show()

Output:

