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]

import matplotlib. pyplot as plt

# Provided data

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]

# Create a line plot

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

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

# Add titles and labels

plt.title('Daily Temperature Changes Over a Month')

plt.xlabel('Day')

plt.ylabel('Temperature (°F)')

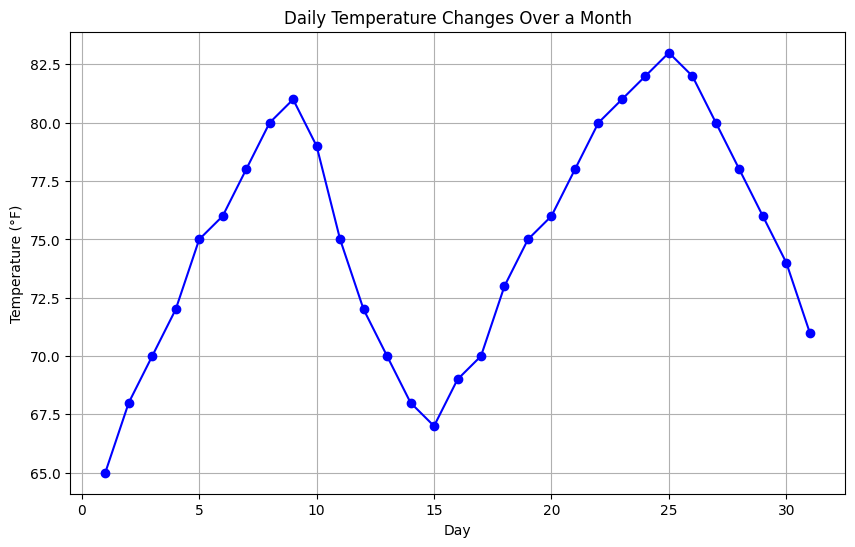
plt.grid(True)

# Show the plot

plt.show()

Output:

The Plot mentioned below shows the output of program1



1. 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, 188,192, 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]

import matplotlib.pyplot as plt

# Provided data

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 line plot

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

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

# Add titles and labels

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

plt.xlabel('Day')

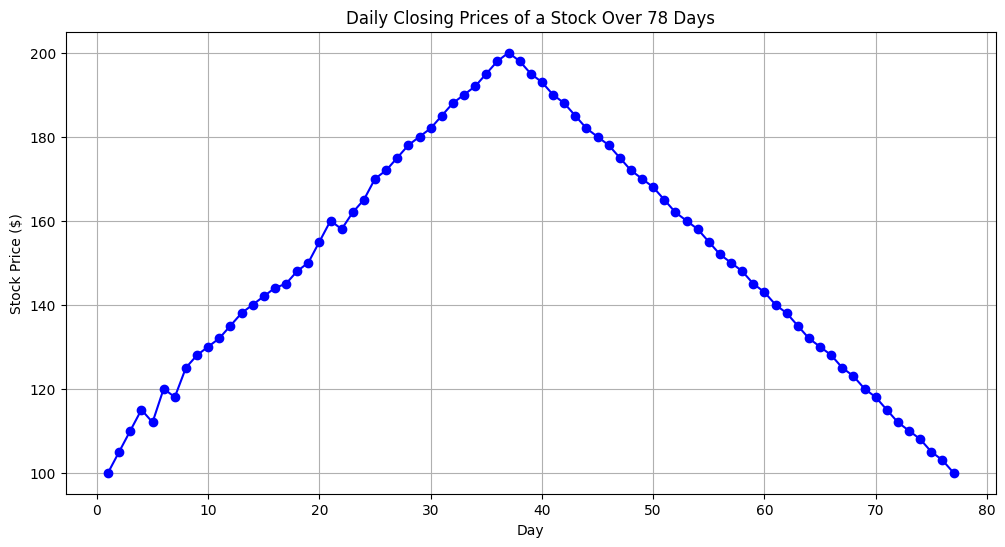
plt.ylabel('Stock Price ($)')

plt.grid(True)

# Show the plot

plt.show()

Output:



1. 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]

import matplotlib.pyplot as pl

# Provided data

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=['blue', 'green', 'red', 'purple', 'orange'])

# Add titles and labels

plt.title('Monthly Expenses by Category')

plt.xlabel('Spending Categories')

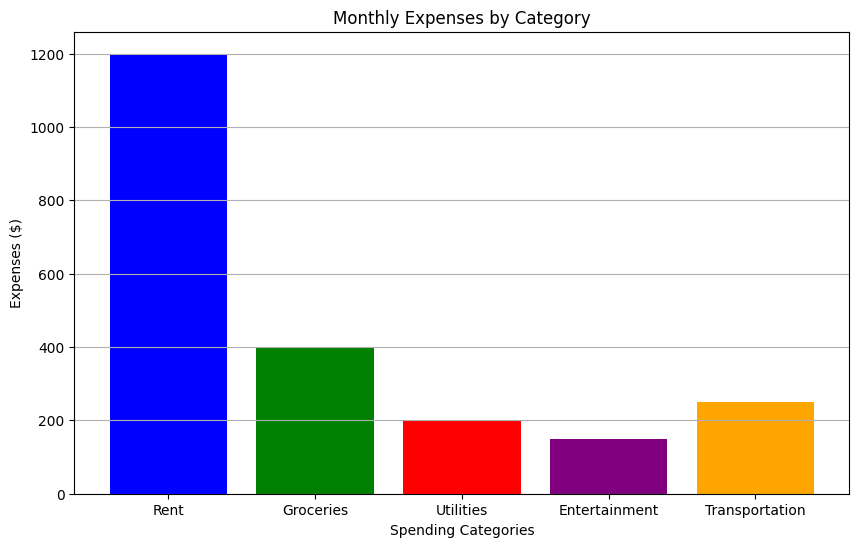
plt.ylabel('Expenses ($)')

plt.grid(axis='y')

# Show the plot

plt.show()

Output:



1. 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]

import matplotlib.pyplot as plt

# Provided data

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

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

plt.hist(product\_prices, bins=10, edgecolor='black', color='blue')

# Add titles and labels

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

plt.xlabel('Price ($)')

plt.ylabel('Frequency')

plt.grid(axis='y')

# Show the plot

plt.show()

Output:



\*\* Using ChatGPT generate the python code to solve the same problem Scenario: Suppose you have two columns named regions and sales with some dummy data frame and you need to generate a bar chart based on these two columns and also generate the percentage. Further, you need to get some inference out of the chart. Create a ChatGPT prompt to generate the code for this scenario. Based on the code generated, ask ChatGPT to give the conclusion/inference.

Note. You can provide the data to ChatGPT or ask it to use sample data.

**### Step 1**: **Create a ChatGPT prompt to generate the Python code**

Here is the prompt to generate the Python code for creating a bar chart and calculating percentages based on two columns (`regions` and `sales`) in a dummy data frame:

\*\*ChatGPT Prompt:\*\*

"Generate Python code to create a bar chart and calculate the percentage distribution based on two columns named `regions` and `sales` in a dummy data frame. Use the following sample data:

data = {

'regions': ['North', 'South', 'East', 'West', 'Central'],

'sales': [150, 200, 250, 175, 225]

}

The code should:

1. Create the data frame using the provided data.

2. Generate a bar chart to visualize the sales for each region.

3. Calculate the percentage of total sales for each region and display these percentages on the chart.

4. Add titles and labels to the chart.

5. Generate the bar chart with a grid for better visualization.

Include comments for each line of code.

**### Step 2: Generate the conclusion/inference based on the code**

After running the generated code, ask ChatGPT to provide conclusions or inferences from the generated bar chart and the calculated percentages.

Here is the complete flow:

\*\*ChatGPT Prompt:\*\*

# Import necessary libraries

import pandas as pd

import matplotlib.pyplot as plt

# Create the data frame using the provided data

data = {'regions': ['North', 'South', 'East', 'West', 'Central'], 'sales': [150, 200, 250, 175, 225]}

df = pd.DataFrame(data)

# Calculate the total sales

total\_sales = df['sales'].sum()

# Calculate the percentage of total sales for each region

df['percentage'] = (df['sales'] / total\_sales) \* 100

# Generate a bar chart to visualize the sales for each region

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

bars = plt.bar(df['regions'], df['sales'], color=['blue', 'green', 'red', 'purple', 'orange'])

# Display the percentages on the chart

for bar, percentage in zip(bars, df['percentage']):

plt.text(bar.get\_x() + bar.get\_width() / 2, bar.get\_height() - 10, f'{percentage:.2f}%', ha='center', color='white', fontsize=12)

# Add titles and labels to the chart

plt.title('Sales Distribution by Region')

plt.xlabel('Regions')

plt.ylabel('Sales ($)')

plt.grid(axis='y')

# Show the plot

plt.show()

**### Step 3: Ask ChatGPT for the conclusion/inference**

\*\*ChatGPT Prompt:\*\* "Based on the generated bar chart and calculated percentages, provide conclusions or inferences about the sales distribution across the regions.

"This approach ensures that the code is generated, and then conclusions are drawn from the resulting visual and data analysis.

