**Project Report**

**EE 381**

First Name:\_\_Bryan \_\_\_\_\_\_\_\_\_\_\_\_ Last Name: \_\_\_\_\_\_\_\_\_\_ Tineo Ccasani \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student ID: \_\_\_\_026850068\_\_\_\_\_

I did this final project own my own and did not share with anyone via discord, emails, verbal, or any other means, if I do, I understand that it is considered as cheating, and there will be an action on my academic dishonesty.

Sign \_\_Bryan Tineo Ccasani\_\_\_\_\_\_\_ Date \_\_\_May 6 2024\_\_\_\_\_\_\_\_\_\_\_

Mean Price: Fill the table below. (2 point)

|  |  |
| --- | --- |
| **Year** | **Mean Price** |
| 2001 | $246235.04 |
| 2005 | $364030.13 |
| 2010 | $331657.47 |
| 2015 | $345883.76 |
| 2020 | $529887.73 |

Mean Price: Insert a bar graph below showing yearly mean prices. (2 points)

A graph of blue bars

Description automatically generated

Standard deviation (STD): Fill the table below. (2 point)

|  |  |
| --- | --- |
| **Year** | **STD** |
| 2001 | $587961.37 |
| 2005 | $978403.13 |
| 2010 | $790797.63 |
| 2015 | $1242075.06 |
| 2020 | $2621786.65 |

Standard deviation: Insert a bar graph below showing yearly standard deviations. (2 point)

A graph of a number of blue bars

Description automatically generated with medium confidence

Probability of price ranging from $200,000 to $300,000 inclusive: Fill the table below. (4 points)

|  |  |
| --- | --- |
| **Year** | **Probability** |
| 2001 | 18.31% |
| 2005 | 28.60% |
| 2010 | 22.02% |
| 2015 | 22.66% |
| 2020 | 25.80% |

Insert a bar graph below showing yearly probability of price ranging from $200,000 to $300,000 inclusive. (4 point)

A graph of blue bars

Description automatically generated

Python Code: Provide your code below. (4 points)

import numpy as np  # Importing numpy for numerical calculations

import matplotlib.pyplot as plt  # Importing matplotlib for plotting graphs

import csv  # Importing csv to handle CSV file operations

from collections import defaultdict  # Importing defaultdict for easier grouping of data by year

# Function to load and process data from the CSV file

*def* load\_data(*filepath*):

    data = defaultdict(list)  # Creating a default dictionary to group sale amounts by year

    with open(*filepath*, 'r') as file:  # Opening the CSV file in read mode

        csv\_reader = csv.DictReader(file)  # Creating a CSV reader object that reads the file as a dictionary

        for row in csv\_reader:  # Looping through each row in the CSV file

            year = int(row['List Year'])  # Converting the year from string to integer

            sale\_amount = float(row['Sale Amount'])  # Converting the sale amount from string to float

            data[year].append(sale\_amount)  # Appending the sale amount to the list of its respective year

    return data  # Returning the dictionary containing lists of sale amounts grouped by year

# Function to calculate yearly mean, standard deviation, and probability of specified price range

*def* calculate\_statistics(*data*):

    means = {}  # Dictionary to store mean values for each year

    std\_devs = {}  # Dictionary to store standard deviation values for each year

    probabilities = {}  # Dictionary to store probability values for each year

    for year, prices in *data*.items():  # Looping through each year and its corresponding prices

        means[year] = np.mean(prices)  # Calculating mean for the current year

        std\_devs[year] = np.std(prices)  # Calculating standard deviation for the current year

        # Counting the number of prices within the specified range for the current year

        count\_in\_range = sum(200000 <= price <= 300000 for price in prices)

        probabilities[year] = count\_in\_range / len(prices)  # Calculating probability for the current year

    return means, std\_devs, probabilities  # Returning the calculated values

# Function to plot data using bar graphs

*def* plot\_data(*data*, *title*, *ylabel*):

    years = sorted(*data*.keys())  # Sorting the years to ensure the plot is ordered

    values = [*data*[year] for year in years]  # Extracting values in the order of sorted years

    plt.figure(*figsize*=(10, 5))  # Setting the figure size for the plot

    plt.bar(years, values, *color*='blue')  # Creating a bar chart with the years as x-axis and values as heights

    plt.xlabel('Year')  # Labeling the x-axis as 'Year'

    plt.ylabel(*ylabel*)  # Labeling the y-axis as specified by the function parameter

    plt.title(*title*)  # Setting the title of the plot

    plt.grid(True)  # Enabling the grid for easier visualization

    plt.show()  # Displaying the plot

# Main function to control the flow of the script

*def* main():

    filepath = 'Sales\_01\_20.csv'  # Define the file path to the CSV file

    data = load\_data(filepath)  # Load and process the data from the CSV file

    means, std\_devs, probabilities = calculate\_statistics(data)  # Calculate statistics from the processed data

    # Plotting the results using the plot function

    plot\_data(means, 'Yearly Mean Sale Prices', 'Mean Price ($)')

    plot\_data(std\_devs, 'Yearly Standard Deviation of Sale Prices', 'Standard Deviation ($)')

    plot\_data(probabilities, 'Probability of Sale Price Between $200,000 and $300,000', 'Probability')

# Checking if the script is run as the main program and not as a module

if \_\_name\_\_ == "\_\_main\_\_":

    main()