Data Science Assignment (3)

Full Name: Ibrahim Qahtan Adnan

Theoretical Group: B

1) What did you learn from the assigned reading?

From Chapter 3, "Describing Data with Statistics," I learned how to use Python to calculate fundamental statistical measures to describe and understand datasets.

Measures of Central Tendency: I learned how to find the mean (the average) , the median (the middle value in a sorted list) , and the mode (the most frequently occurring value).

Frequency Distribution: The chapter taught me how to find the most common elements in a dataset and present this as a frequency table.

Measures of Dispersion (Spread): I learned how to quantify how "spread out" a dataset is by calculating:

Range: The difference between the highest and lowest values.

Variance: The average of the squared differences from the mean, which measures how far the data is dispersed from its average.

Standard Deviation: The square root of the variance, which is a more intuitive measure of spread in the same units as the original data.

Correlation: I learned how to calculate the correlation coefficient, a value between -1 and 1 that measures the strength and direction of a linear relationship between two sets of data. The chapter also warns that "correlation doesn’t imply causation".

Data Visualization: The chapter emphasizes the importance of scatter plots to visually check for relationships that statistical numbers might miss. It uses Anscombe's quartet as a key example, showing four datasets with identical statistics (mean, variance, correlation) but wildly different visual plots.

Reading Data from Files: Crucially, I learned how to make these programs practical by reading data from external files, instead of "hardcoding" it. This includes reading from:

Simple .txt files.

.csv (Comma-Separated Values) files using Python's built-in csv module, including how to skip a header row.

2) Can your learning help you solve real-life problems?

A) If yes, describe how (Write approximately 200–300 words):

Yes, the concepts in Chapter 3 are foundational to solving countless real-life problems. Almost any field that collects data uses these exact statistical measures to make sense of it.

For instance, a business manager could use these skills to analyze sales data. They could calculate the mean and median daily sales to understand typical performance and find the standard deviation to see how consistent or volatile their sales are.

The ability to read CSV files is a critical real-world skill. Most data from spreadsheets, databases, or online analytics tools (like Google Trends, as mentioned in the book) is exported as a CSV. This chapter provides the tools to read that data directly into Python for analysis.

Furthermore, understanding correlation and scatter plots is essential for making good decisions. A scientist could plot temperature against crop yield to see if there's a relationship. The book itself uses the example of analyzing the correlation between high school grades and college admission test scores. This analysis helps determine if one variable can predict another. The warning about "correlation vs. causation" is a vital piece of wisdom for anyone interpreting data, preventing them from making false conclusions.

B) If you are able to answer question A, provide an example with a clear explanation and include Python code to support your answer.

Example:

A small e-commerce business owner has a .csv file named web_traffic.csv. This file tracks the number of daily visitors to their website (Column 1) and the total sales for that day (Column 2). They want to know if there is a relationship between website traffic and sales.

Explanation and Solution:

We can use the methods from Chapter 3 to solve this:

Read the CSV file: We'll use the read_csv technique from the chapter to read web_traffic.csv, skipping the header.

Store the data: We'll put the traffic data in one list and the sales data in another.

Calculate Correlation: We'll implement a function based on the chapter's principles to calculate the correlation coefficient between traffic and sales. This will give us a single number telling us how strong the relationship is.

Create a Scatter Plot: We'll use matplotlib to create a scatter plot, just as the chapter does, to visualize the relationship. This helps confirm if the correlation number makes sense.

C) Python Code:

```
import csv
import matplotlib.pyplot as plt
def read_web_data(filename):
    traffic = []
    sales = []
    try:
        with open(filename) as f:
            reader = csv.reader(f)
            next(reader)
            for row in reader:
                traffic.append(float(row[0]))
                sales.append(float(row[1]))
    except FileNotFoundError:
        print(f"Error: The file '{filename}' was not found.")
        return None, None
    except Exception as e:
        print(f"An error occurred: {e}")
        return None, None

    return traffic, sales

def find_correlation(x, y):
    n = len(x)
    if n == 0 or len(y) != n:
        return 0
    sum_x = sum(x)
    sum_y = sum(y)
    sum_x_sq = sum([xi**2 for xi in x])
    sum_y_sq = sum([yi**2 for yi in y])
    sum_prod_xy = sum([xi*yi for xi, yi in zip(x, y)])
    numerator = n * sum_prod_xy - sum_x * sum_y
    denominator_term1 = (n * sum_x_sq - sum_x**2)**0.5
    denominator_term2 = (n * sum_y_sq - sum_y**2)**0.5

    if denominator_term1 == 0 or denominator_term2 == 0:
        return 0

    correlation = numerator / (denominator_term1 * denominator_term2)
    return correlation

def create_scatter_plot(x, y):
    plt.scatter(x, y)
    plt.title('Website Visitors vs. Daily Sales')
```

```
    plt.xlabel('Daily Visitors')
    plt.ylabel('Daily Sales ($)')
    plt.grid(True)
    plt.show()
if __name__ == '__main__':
    try:
        with open('web_traffic.csv', 'w', newline='') as f:
            writer = csv.writer(f)
            writer.writerow(['Visitors', 'Sales'])
            writer.writerow([100, 1200])
            writer.writerow([120, 1500])
            writer.writerow([110, 1450])
            writer.writerow([150, 2100])
            writer.writerow([160, 2250])
            writer.writerow([130, 1700])
            writer.writerow([180, 2500])
    except IOError:
        print("Error: Could not write dummy CSV file.")
    traffic_data, sales_data = read_web_data('web_traffic.csv')

    if traffic_data:
        corr = find_correlation(traffic_data, sales_data)
        print(f"--- Business Analysis ---")
        print(f"Correlation between Visitors and Sales: {corr:.4f}")
        if corr > 0.8:
            print("This indicates a very strong positive linear relationship.")
            print("Conclusion: More website traffic is highly correlated with
more sales.")
        elif corr > 0.5:
            print("This indicates a moderate positive relationship.")
        else:
            print("The relationship is weak or not positive.")

        print("\nDisplaying scatter plot for visual confirmation...")
        create_scatter_plot(traffic_data, sales_data)
```