

# Lab 8

Daniel Mehta n01753264

```
In [22]: import numpy as np
import scipy.stats as stats
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
```

## Question 1

A teacher wants to analyze the performance of students in a math test. Given the scores: scores = [78, 85, 90, 92, 88, 76, 95]

Compute and print the variance to understand the spread of scores.

```
In [4]: scores = [78, 85, 90, 92, 88, 76, 95]

variance = np.var(scores)

print(f"Variance of scores: {variance}")
```

Variance of scores: 43.06122448979592

## Question 2:

The teacher also wants to measure how much the scores deviate from the average.

Compute and print the standard deviation of the scores.

```
In [5]: std_dev = np.std(scores)
print(f"Standard Deviation of scores: {std_dev}")
```

Standard Deviation of scores: 6.562105187346201

## Question 3:

A company wants to analyze the relationship between employee experience (in years) and their productivity score.

experience = [1, 2, 3, 4, 5]

productivity = [50, 55, 65, 70, 80]

Compute and print Pearson correlation to assess the linear relationship.

```
In [7]: experience = [1, 2, 3, 4, 5]
productivity = [50, 55, 65, 70, 80]

correlation = np.corrcoef(experience, productivity)[0, 1]

print(f"Pearson Correlation Coefficient: {correlation}")
```

Pearson Correlation Coefficient: 0.9933992677987827

## Question 4:

The HR department is interested in determining if there is a monotonic relationship between employee experience and productivity.

Compute and print Spearman correlation

```
In [12]: experience = [1, 2, 3, 4, 5]
productivity = [50, 55, 65, 70, 80]

spearman_corr, _ = stats.spearmanr(experience, productivity)

print(f"Spearman Correlation Coefficient: {spearman_corr}")
```

Spearman Correlation Coefficient: 0.9999999999999999

## Question 5:

A business analyst wants to visualize the relationship between advertising spending and sales.

ad\_spending = [100, 200, 300, 400, 500]

sales = [20, 40, 60, 80, 100]

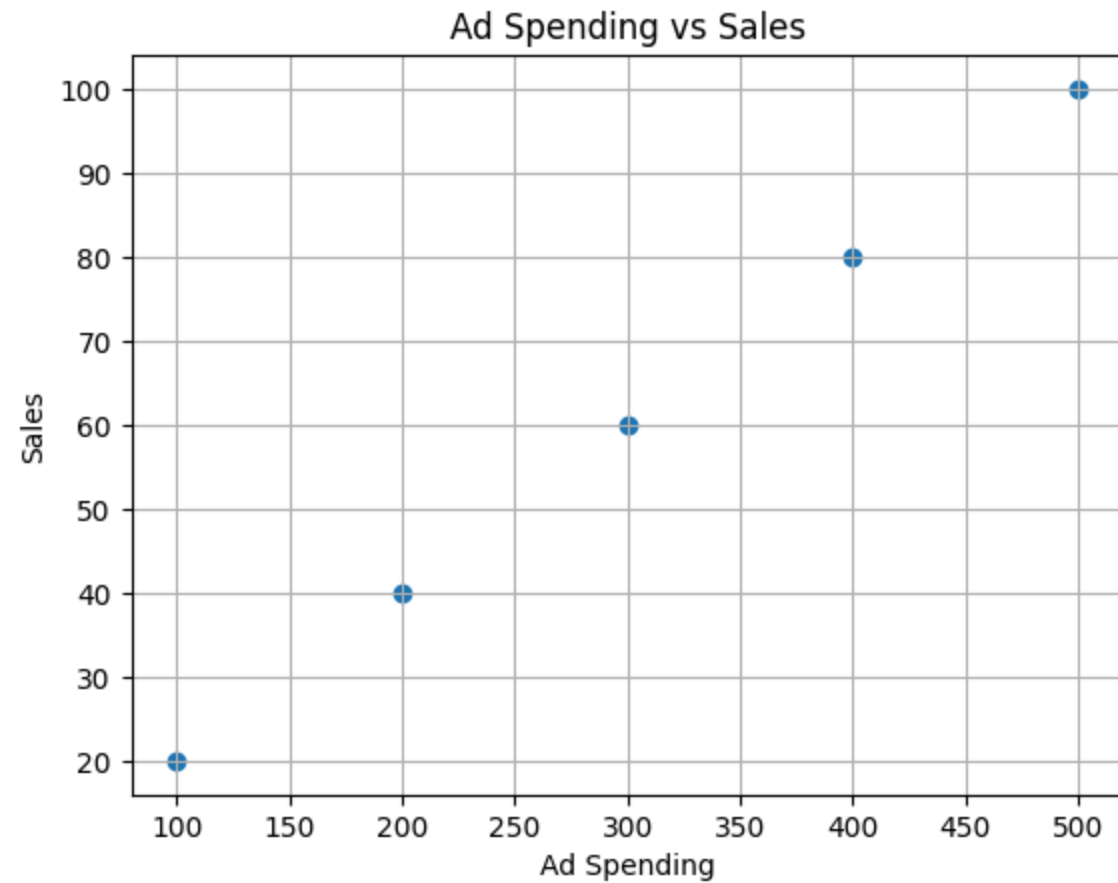
Generate and display a scatter plot.

```
In [17]: ad_spending = [100, 200, 300, 400, 500]
sales = [20, 40, 60, 80, 100]
```

```
plt.scatter(ad_spending, sales)

plt.xlabel("Ad Spending")
plt.ylabel("Sales")
plt.title("Ad Spending vs Sales")

plt.grid(True)
plt.show()
```



## Question 6:

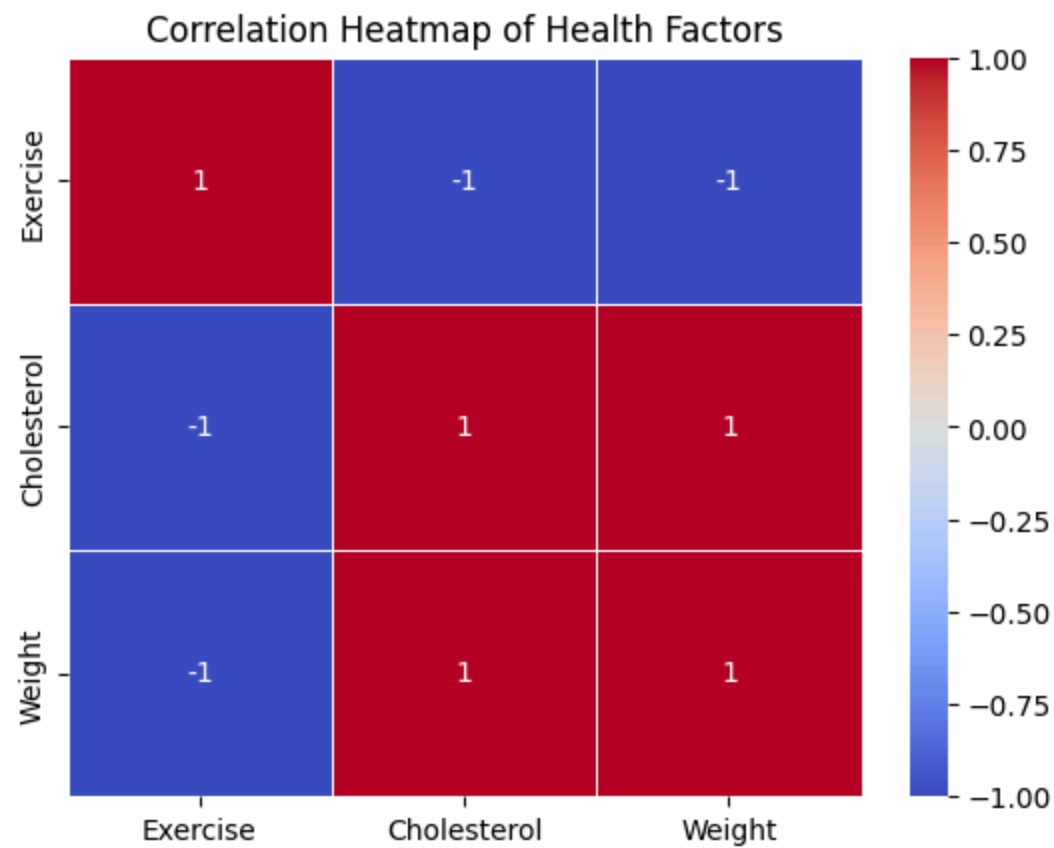
A health researcher is analyzing the correlation between daily exercise (minutes) and cholesterol levels.

```
df = pd.DataFrame({'Exercise': [30, 40, 50, 60, 70], 'Cholesterol': [220, 210, 200, 190, 180], 'Weight': [80, 78, 75, 73, 70]})
```

Generate and display a correlation heatmap to identify relationships.

```
In [33]: df = pd.DataFrame({'Exercise': [30, 40, 50, 60, 70], 'Cholesterol': [220, 210, 200, 190, 180], 'Weight': [80, 78, 75, 73, 70]})

correlation_matrix = df.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm',linewidths=0.5)
plt.title("Correlation Heatmap of Health Factors")
plt.show()
```



## Question 7

A financial analyst wants to analyze the volatility of stock prices over time.

stocks = [100, 102, 105, 98, 97, 110, 115]

Compute the rolling variance using a window size of 3.

```
In [35]: stocks = [100, 102, 105, 98, 97, 110, 115]
```

```
df = pd.DataFrame({'Stock Prices': stocks})

df['Rolling Variance'] = df['Stock Prices'].rolling(window=3).var()

print(df)
```

	Stock Prices	Rolling Variance
0	100	NaN
1	102	NaN
2	105	6.333333
3	98	12.333333
4	97	19.000000
5	110	52.333333
6	115	86.333333

## Question 8:

A data scientist wants to study the impact of study hours on exam scores and visualize the trend.

```
study_hours = [1, 2, 3, 4, 5]
```

```
exam_scores = [50, 60, 70, 80, 90]
```

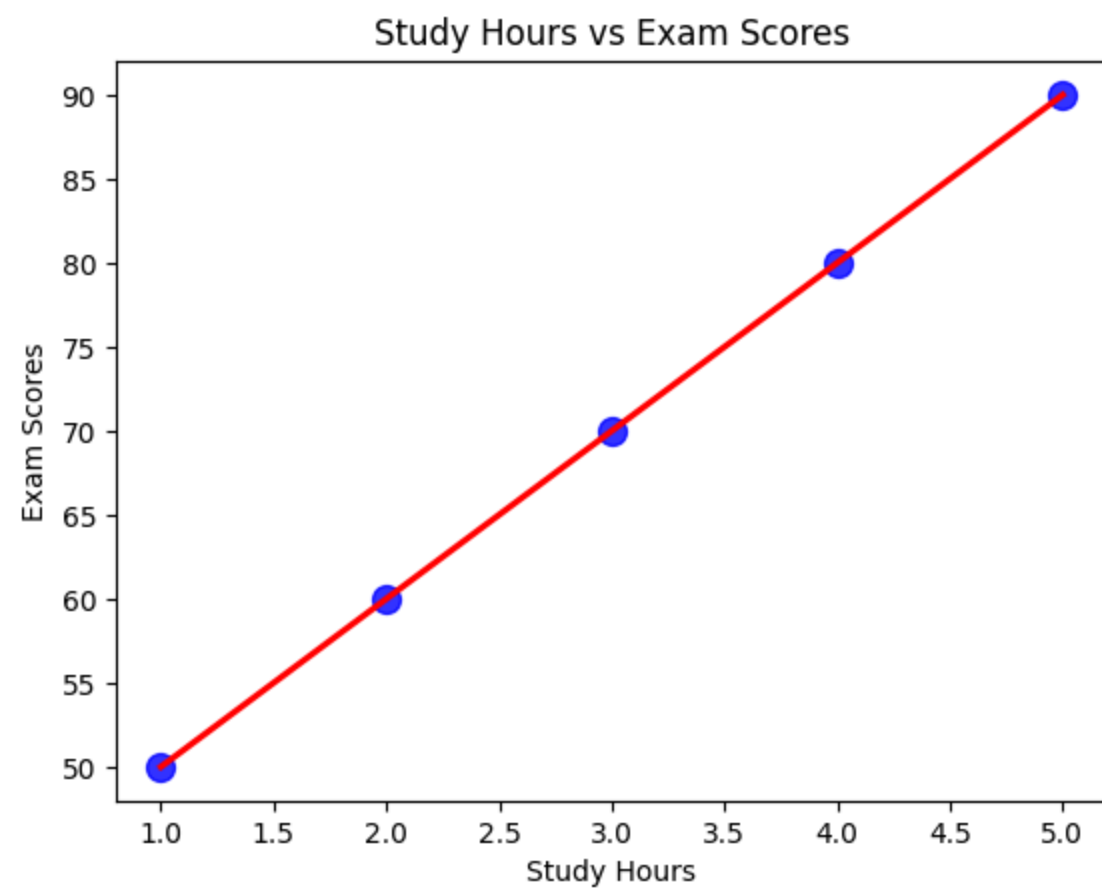
Generate and display a scatter plot with a regression line.

```
In [39]: study_hours = [1, 2, 3, 4, 5]
exam_scores = [50, 60, 70, 80, 90]

sns.regplot(x=study_hours, y=exam_scores, color='blue', scatter_kws={'s': 100}, line_kws={'color': 'red'})

plt.xlabel("Study Hours")
plt.ylabel("Exam Scores")
plt.title("Study Hours vs Exam Scores")

plt.show()
```



In [ ]: