# DS 4400 Homework 1
# Linear Algebra, Probability and Statistics

Ariv Ahuja

January 2026

**Code Link:** `https://github.com/ArivAhuja1/DS4400/blob/main/HW1/AHUJA_HW1_code.ipynb`

## Problem 1: Probability and Expectation [A] (22 points)

In a bag there are 50 balls: 25 red balls (3 points each), 10 green balls (5 points each), and 15 blue balls (10 points each).

### Part 1: Expected Value of a Random Draw

Let $X$ be the random variable denoting the points from a single draw.

The probabilities are:

$$P(X=3) = P(\text{Red}) = \frac{25}{50} = 0.5$$
$$P(X=5) = P(\text{Green}) = \frac{10}{50} = 0.2$$
$$P(X=10) = P(\text{Blue}) = \frac{15}{50} = 0.3$$

The expected value is:

$$\begin{aligned} E[X] &= \sum_x x \cdot P(X=x) \\ &= 3 \cdot 0.5 + 5 \cdot 0.2 + 10 \cdot 0.3 \\ &= 1.5 + 1.0 + 3.0 \\ &= \boxed{5.5 \text{ points}} \end{aligned}$$

### Part 2: Variance of $X$

First, compute $E[X^2]$:

$$\begin{aligned} E[X^2] &= \sum_x x^2 \cdot P(X=x) \\ &= 3^2 \cdot 0.5 + 5^2 \cdot 0.2 + 10^2 \cdot 0.3 \\ &= 9 \cdot 0.5 + 25 \cdot 0.2 + 100 \cdot 0.3 \\ &= 4.5 + 5.0 + 30.0 \\ &= 39.5 \end{aligned}$$

The variance is:

$$\begin{aligned}
\text{Var}(X) &= E[X^2] - (E[X])^2 \\
&= 39.5 - (5.5)^2 \\
&= 39.5 - 30.25 \\
&= \boxed{9.25}
\end{aligned}$$

## Part 3: Expected Value After Removing 2 Red and 1 Green

After removing 2 red balls and 1 green ball, we have 47 balls remaining:

- Red: $25 - 2 = 23$ balls
- Green: $10 - 1 = 9$ balls
- Blue: $15 - 0 = 15$ balls

The new probabilities are:

$$\begin{aligned}
P(X = 3) &= \frac{23}{47} \\
P(X = 5) &= \frac{9}{47} \\
P(X = 10) &= \frac{15}{47}
\end{aligned}$$

The expected value is:

$$\begin{aligned}
E[X] &= 3 \cdot \frac{23}{47} + 5 \cdot \frac{9}{47} + 10 \cdot \frac{15}{47} \\
&= \frac{69 + 45 + 150}{47} \\
&= \frac{264}{47} \\
&\approx \boxed{5.617 \text{ points}}
\end{aligned}$$

## Part 4: Computing $E[X(X-1)]$ Given $E(X) = 10$ and Var$(X) = 2$

We need to find $E[X(X-1)]$:

$$\begin{aligned}
E[X(X-1)] &= E[X^2 - X] \\
&= E[X^2] - E[X]
\end{aligned}$$

We know that:

$$\text{Var}(X) = E[X^2] - (E[X])^2$$

Solving for $E[X^2]$:

$$E[X^2] = \text{Var}(X) + (E[X])^2 = 2 + 10^2 = 2 + 100 = 102$$

Therefore:

$$E[X(X-1)] = E[X^2] - E[X] = 102 - 10 = \boxed{92}$$

## Problem 2: Conditional Probability and Bayes Theorem [A] (18 points)

Given information:

- $P(S) = 0.80$ (probability email is spam)
- $P(\overline{S}) = 0.20$ (probability email is not spam)
- $P(FM|S) = 0.10$ (probability of "free money" given spam)
- $P(FM|\overline{S}) = 0.01$ (probability of "free money" given not spam)

### Part 1: Probability of "Free Money" in Any Email

Using the law of total probability:

$$\begin{aligned}
P(FM) &= P(FM|S) \cdot P(S) + P(FM|\overline{S}) \cdot P(\overline{S}) \\
&= 0.10 \cdot 0.80 + 0.01 \cdot 0.20 \\
&= 0.080 + 0.002 \\
&= \boxed{0.082}
\end{aligned}$$

### Part 2: Probability Email is Spam Given "Free Money"

Using Bayes' Theorem:

$$\begin{aligned}
P(S|FM) &= \frac{P(FM|S) \cdot P(S)}{P(FM)} \\
&= \frac{0.10 \cdot 0.80}{0.082} \\
&= \frac{0.08}{0.082} \\
&= \frac{80}{82} = \frac{40}{41} \\
&\approx \boxed{0.9756}
\end{aligned}$$

### Part 3: Probability Email is Not Spam Given No "Free Money"

First, compute $P(\overline{FM})$:

$$P(\overline{FM}) = 1 - P(FM) = 1 - 0.082 = 0.918$$

Compute $P(\overline{FM}|\overline{S})$:

$$P(\overline{FM}|\overline{S}) = 1 - P(FM|\overline{S}) = 1 - 0.01 = 0.99$$

Using Bayes' Theorem:

$$\begin{aligned}
P(\overline{S}|\overline{FM}) &= \frac{P(\overline{FM}|\overline{S}) \cdot P(\overline{S})}{P(\overline{FM})} \\
&= \frac{0.99 \cdot 0.20}{0.918} \\
&= \frac{0.198}{0.918} \\
&= \frac{198}{918} = \frac{11}{51} \\
&\approx \boxed{0.2157}
\end{aligned}$$

## Problem 3: Matrices and Vectors [A] (20 points)

Define the matrices:

$$D_1 = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ -1 & 0 & -3 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix}, \quad D_2 = \begin{pmatrix} 1 & -1 & 0 & 2 \\ -1 & 0 & 1 & -1 \\ 0 & -2 & 0 & 1 \\ 3 & 0 & -1 & 0 \end{pmatrix}$$

### Part 1: Linear Independence of Columns

**For $D_1$:** The columns are **linearly dependent**.

To verify, we perform row reduction on $D_1$. Starting with: $D_1 = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ -1 & 0 & -3 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix}$

**Step 1:** $R_3 \leftarrow R_3 + R_1$ (eliminate the $-1$ in position $(3,1)$): $\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix}$

**Step 2:** $R_4 \leftarrow R_4 + \frac{1}{2}R_2$ (eliminate the $-1$ in position $(4,2)$): $\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 1 & -1 \end{pmatrix}$

**Step 3:** $R_4 \leftarrow R_4 + R_3$ (eliminate the 1 in position $(4,3)$): $\begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$

The row echelon form has only **3 pivots** (in columns 1, 2, and 3) with a row of zeros. Therefore, the columns are **linearly dependent**.

**For $D_2$:** The columns are **linearly independent**.

Starting with: $D_2 = \begin{pmatrix} 1 & -1 & 0 & 2 \\ -1 & 0 & 1 & -1 \\ 0 & -2 & 0 & 1 \\ 3 & 0 & -1 & 0 \end{pmatrix}$

**Step 1:** $R_2 \leftarrow R_2 + R_1$: $\begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 1 & 1 \\ 0 & -2 & 0 & 1 \\ 3 & 0 & -1 & 0 \end{pmatrix}$

**Step 2:** $R_4 \leftarrow R_4 - 3R_1$: $\begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 1 & 1 \\ 0 & -2 & 0 & 1 \\ 0 & 3 & -1 & -6 \end{pmatrix}$

**Step 3:** $R_3 \leftarrow R_3 - 2R_2$: $\begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & -2 & -1 \\ 0 & 3 & -1 & -6 \end{pmatrix}$

**Step 4:** $R_4 \leftarrow R_4 + 3R_2$: $\begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & -2 & -1 \\ 0 & 0 & 2 & -3 \end{pmatrix}$

**Step 5:** $R_4 \leftarrow R_4 + R_3$: $\begin{pmatrix} 1 & -1 & 0 & 2 \\ 0 & -1 & 1 & 1 \\ 0 & 0 & -2 & -1 \\ 0 & 0 & 0 & -4 \end{pmatrix}$

The row echelon form has **4 pivots** (in all columns), so all columns are **linearly independent**.

## Part 2: Maximum Number of Linearly Independent Columns

**For $D_1$:** Maximum of **3** linearly independent columns.
**For $D_2$:** Maximum of **4** linearly independent columns (all columns are independent).

## Part 3: Rank of Each Matrix

From the row reduction shown above:

**Rank of $D_1$:** The row echelon form has 3 non-zero rows, so $\text{rank}(D_1) = \boxed{3}$.

**Rank of $D_2$:** The row echelon form has 4 non-zero rows, so $\text{rank}(D_2) = \boxed{4}$.

## Part 4: Vector Products with $\theta$

Given:

$$\theta = \begin{pmatrix} -1 \\ 0 \\ 1 \\ 2 \end{pmatrix}$$

**$\theta D_1$: Cannot be computed.**

$\theta$ is a $4 \times 1$ matrix and $D_1$ is a $4 \times 4$ matrix. For $\theta D_1$, we need the number of columns in $\theta$ (which is 1) to equal the number of rows in $D_1$ (which is 4). Since $1 \neq 4$, this product is undefined.

**$D_1\theta$: Can be computed.**

$D_1$ is $4 \times 4$ and $\theta$ is $4 \times 1$, so the result is $4 \times 1$:

$$D_1\theta = \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ -1 & 0 & -3 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \\ 1 \\ 2 \end{pmatrix}$$

Computing each entry:

$$\begin{aligned}
\text{Row 1:} \quad & 1(-1) + 0(0) + 2(1) + 0(2) = -1 + 0 + 2 + 0 = 1 \\
\text{Row 2:} \quad & 0(-1) + 2(0) + 4(1) + (-2)(2) = 0 + 0 + 4 - 4 = 0 \\
\text{Row 3:} \quad & (-1)(-1) + 0(0) + (-3)(1) + 1(2) = 1 + 0 - 3 + 2 = 0 \\
\text{Row 4:} \quad & 0(-1) + (-1)(0) + (-1)(1) + 0(2) = 0 + 0 - 1 + 0 = -1
\end{aligned}$$

$$D_1\theta = \boxed{\begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}} \quad \text{(dimension: } 4 \times 1\text{)}$$

$\theta^T D_1$**: Can be computed.**
$\theta^T$ is $1 \times 4$ and $D_1$ is $4 \times 4$, so the result is $1 \times 4$:

$$\theta^T D_1 = \begin{pmatrix} -1 & 0 & 1 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 2 & 0 \\ 0 & 2 & 4 & -2 \\ -1 & 0 & -3 & 1 \\ 0 & -1 & -1 & 0 \end{pmatrix}$$

Computing each entry:

$$\begin{aligned}
\text{Col 1:} \quad & (-1)(1) + 0(0) + 1(-1) + 2(0) = -1 + 0 - 1 + 0 = -2 \\
\text{Col 2:} \quad & (-1)(0) + 0(2) + 1(0) + 2(-1) = 0 + 0 + 0 - 2 = -2 \\
\text{Col 3:} \quad & (-1)(2) + 0(4) + 1(-3) + 2(-1) = -2 + 0 - 3 - 2 = -7 \\
\text{Col 4:} \quad & (-1)(0) + 0(-2) + 1(1) + 2(0) = 0 + 0 + 1 + 0 = 1
\end{aligned}$$

$$\theta^T D_1 = \boxed{\begin{pmatrix} -2 & -2 & -7 & 1 \end{pmatrix}} \quad \text{(dimension: } 1 \times 4\text{)}$$

$D_1\theta^T$**: Cannot be computed.**
$D_1$ is $4 \times 4$ and $\theta^T$ is $1 \times 4$. For $D_1\theta^T$, we need the number of columns in $D_1$ (which is 4) to equal the number of rows in $\theta^T$ (which is 1). Since $4 \neq 1$, this product is undefined.

## Problem 4: Matrix Transpose and Inverse [A] (20 points)

### Part 1: Language Matrix

People and their languages:

- Anton: French, German
- Geraldine: English, French, Italian
- James: English, Italian, Spanish
- Lauren: English, German, Italian, Spanish (all except French)

Matrix $A$ with rows = people, columns = languages (English, French, German, Italian, Spanish):

$$A = \begin{pmatrix} & \text{Eng} & \text{Fr} & \text{Ger} & \text{Ita} & \text{Spa} \\ \text{Anton} & 0 & 1 & 1 & 0 & 0 \\ \text{Geraldine} & 1 & 1 & 0 & 1 & 0 \\ \text{James} & 1 & 0 & 0 & 1 & 1 \\ \text{Lauren} & 1 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

**Computing $AA^T$ (4×4 matrix):**

$$AA^T = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 & 0 & 1 \\ 1 & 3 & 2 & 2 \\ 0 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix}$$

**Significance of $AA^T$:** This is a person-by-person matrix where entry $(i, j)$ represents the **number of languages that person $i$ and person $j$ have in common**. The diagonal entries show the total number of languages each person speaks.

For example:

- $(AA^T)_{11} = 2$: Anton speaks 2 languages
- $(AA^T)_{23} = 2$: Geraldine and James share 2 languages (English and Italian)
- $(AA^T)_{13} = 0$: Anton and James share no common languages

**Computing $A^TA$ (5×5 matrix):**

$$A^TA = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 3 & 1 & 1 & 3 & 2 \\ 1 & 2 & 1 & 1 & 0 \\ 1 & 1 & 2 & 1 & 1 \\ 3 & 1 & 1 & 3 & 2 \\ 2 & 0 & 1 & 2 & 2 \end{pmatrix}$$

**Significance of $A^TA$:** This is a language-by-language matrix where entry $(i, j)$ represents the **number of people who speak both language $i$ and language $j$**. The diagonal entries show the total number of people who speak each language.

For example:

- $(A^TA)_{11} = 3$: 3 people speak English
- $(A^TA)_{25} = 0$: No one speaks both French and Spanish
- $(A^TA)_{14} = 3$: 3 people speak both English and Italian

### Part 2: Random Matrices and Inverses

Using NumPy with random seed 42 for reproducibility:

**Matrix 1:**

$$M_1 = \begin{pmatrix} -4 & 9 & 4 \\ 0 & -3 & 10 \\ -4 & 8 & 0 \end{pmatrix}, \quad \det(M_1) = -88$$

$$M_1^{-1} = \begin{pmatrix} 0.9091 & -0.3636 & -1.1591 \\ 0.4545 & -0.1818 & -0.4545 \\ 0.1364 & 0.0455 & -0.1364 \end{pmatrix}$$

$$M_1 \cdot M_1^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \checkmark$$

**Matrix 2:**

$$M_2 = \begin{pmatrix} 0 & 10 & -7 \\ -3 & -8 & 10 \\ -9 & 1 & -5 \end{pmatrix}, \quad \det(M_2) = -525$$

$$M_2^{-1} = \begin{pmatrix} -0.0571 & -0.0819 & -0.0838 \\ 0.2000 & 0.1200 & -0.0400 \\ 0.1429 & 0.1714 & -0.0571 \end{pmatrix}$$

$$M_2 \cdot M_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \checkmark$$

**Matrix 3:**

$$M_3 = \begin{pmatrix} -9 & 10 & -10 \\ 1 & 1 & 6 \\ -1 & 5 & 4 \end{pmatrix}, \quad \det(M_3) = 74$$

$$M_3^{-1} = \begin{pmatrix} -0.3514 & -1.2162 & 0.9459 \\ -0.1351 & -0.6216 & 0.5946 \\ 0.0811 & 0.4730 & -0.2568 \end{pmatrix}$$

$$M_3 \cdot M_3^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \checkmark$$

All three randomly generated matrices are invertible (non-singular), and the products $M_i \cdot M_i^{-1}$ yield the identity matrix as expected.

## Problem 5: Average, Variance, and Correlation [C] (20 points)

Using the `kc_house_data.csv` dataset, excluding the columns `id`, `date`, and `zipcode`.

### Part 1: Feature Statistics

Table 1: Summary Statistics for Each Feature

| Feature | Average | Min | Max | Variance |
|---|---|---|---|---|
| bedrooms | 3.3708 | 0.0000 | 33.0000 | 0.8650 |
| bathrooms | 2.1148 | 0.0000 | 8.0000 | 0.5932 |
| sqft_living | 2079.8997 | 290.0000 | 13540.0000 | 843,533.68 |
| sqft_lot | 15106.9676 | 520.0000 | 1,651,359.00 | 1,715,658,774.18 |
| floors | 1.4943 | 1.0000 | 3.5000 | 0.2916 |
| waterfront | 0.0075 | 0.0000 | 1.0000 | 0.0075 |
| view | 0.2343 | 0.0000 | 4.0000 | 0.5872 |
| condition | 3.4094 | 1.0000 | 5.0000 | 0.4235 |
| grade | 7.6569 | 1.0000 | 13.0000 | 1.3817 |
| sqft_above | 1788.3907 | 290.0000 | 9410.0000 | 685,734.67 |
| sqft_basement | 291.5090 | 0.0000 | 4820.0000 | 195,872.67 |
| yr_built | 1971.0051 | 1900.0000 | 2015.0000 | 862.7973 |
| yr_renovated | 84.4023 | 0.0000 | 2015.0000 | 161,346.21 |
| lat | 47.5601 | 47.1559 | 47.7776 | 0.0192 |
| long | −122.2139 | −122.5190 | −121.3150 | 0.0198 |
| sqft_living15 | 1986.5525 | 399.0000 | 6210.0000 | 469,761.24 |
| sqft_lot15 | 12768.4557 | 651.0000 | 871,200.00 | 745,518,225.34 |

**Features with lowest and highest average:**

- **Lowest average:** `long` $= -122.2139$
- **Highest average:** `sqft_lot` $= 15106.9676$

**Features with lowest and highest variance:**

- **Lowest variance:** `waterfront` $= 0.0075$
- **Highest variance:** `sqft_lot` $= 1,715,658,774.18$

### Part 2: Correlation with Response (price)

Table 2: Correlation Coefficients with Price

| Feature | Correlation with Price |
|---|---:|
| bedrooms | 0.3084 |
| bathrooms | 0.5251 |
| sqft_living | 0.7020 |
| sqft_lot | 0.0897 |
| floors | 0.2568 |
| waterfront | 0.2664 |
| view | 0.3973 |
| condition | 0.0364 |
| grade | 0.6674 |
| sqft_above | 0.6056 |
| sqft_basement | 0.3238 |
| yr_built | 0.0540 |
| yr_renovated | 0.1264 |
| lat | 0.3070 |
| long | 0.0216 |
| sqft_living15 | 0.5854 |
| sqft_lot15 | 0.0824 |

**Positively correlated features:** All features are positively correlated with price: `bedrooms`, `bathrooms`, `sqft_living`, `sqft_lot`, `floors`, `waterfront`, `view`, `condition`, `grade`, `sqft_above`, `sqft_basement`, `yr_built`, `yr_renovated`, `lat`, `long`, `sqft_living15`, `sqft_lot15`.

**Feature with highest positive correlation:** `sqft_living` with correlation coefficient of $\boxed{0.7020}$

### Part 3: Negative Correlations

**No features have a negative correlation with price.** All 17 features analyzed have positive correlation coefficients with the response variable.

## Python Code for Problem 5

```python
import pandas as pd
import numpy as np

# Load the data
df = pd.read_csv('kc_house_data.csv')

# Remove id, date, and zipcode columns
cols_to_remove = ['id', 'date', 'zipcode']
df_analysis = df.drop(columns=cols_to_remove)

# Get feature columns (excluding price which is the response)
features = [col for col in df_analysis.columns if col != 'price']

# Part 1: Compute statistics for each feature
print("--- Part 1: Feature Statistics ---\n")
stats = {}
for col in features:
    stats[col] = {
        'average': df_analysis[col].mean(),
        'min': df_analysis[col].min(),
        'max': df_analysis[col].max(),
        'variance': df_analysis[col].var()
    }

# Print statistics table
print(f"{'Feature':<20} {'Average':<15} {'Min':<15} {'Max':<15} {'Variance':<20}")
print("-"*85)
for col in features:
    print(f"{col:<20} {stats[col]['average']:<15.4f} "
          f"{stats[col]['min']:<15.4f} {stats[col]['max']:<15.4f} "
          f"{stats[col]['variance']:<20.4f}")

# Find lowest and highest average
avg_sorted = sorted(stats.items(), key=lambda x: x[1]['average'])
print(f"\nLowest average: {avg_sorted[0][0]} = {avg_sorted[0][1]['average']:.4f}")
print(f"Highest average: {avg_sorted[-1][0]} = {avg_sorted[-1][1]['average']:.4f}")

# Find lowest and highest variance
var_sorted = sorted(stats.items(), key=lambda x: x[1]['variance'])
print(f"\nLowest variance: {var_sorted[0][0]} = {var_sorted[0][1]['variance']:.4f}")
print(f"Highest variance: {var_sorted[-1][0]} = {var_sorted[-1][1]['variance']:.4f}")

# Part 2: Correlation with response (price)
print("\n--- Part 2: Correlation with Price ---\n")
correlations = {}
for col in features:
```

```python
    correlations[col] = df_analysis[col].corr(df_analysis['price'])

print(f"{'Feature':<20} {'Correlation with Price':<25}")
print("-"*45)
for col in features:
    print(f"{col:<20} {correlations[col]:<25.6f}")

# Positive correlations
pos_corr = {k: v for k, v in correlations.items() if v > 0}
print(f"\nPositively correlated features: {list(pos_corr.keys())}")

# Highest positive correlation
max_corr_feature = max(correlations.items(), key=lambda x: x[1])
print(f"Highest positive correlation: {max_corr_feature[0]} = "
      f"{max_corr_feature[1]:.6f}")

# Part 3: Negative correlations
neg_corr = {k: v for k, v in correlations.items() if v < 0}
print(f"\n--- Part 3: Negative Correlations ---")
if neg_corr:
    print(f"Negatively correlated features: {list(neg_corr.keys())}")
else:
    print("No features with negative correlation found.")
```

## Python Code for Problem 4.2

```python
import numpy as np
np.random.seed(42)  # For reproducibility

for i in range(3):
    M = np.random.randint(-10, 11, size=(3, 3))
    print(f"\nMatrix {i+1}:")
    print(M)

    det = np.linalg.det(M)
    print(f"Determinant: {det:.4f}")

    if abs(det) > 1e-10:
        M_inv = np.linalg.inv(M)
        print(f"Inverse:")
        print(M_inv)
        print(f"M @ M^(-1) (should be identity):")
        print(np.round(M @ M_inv, 10))
    else:
        print("Matrix is singular, no inverse exists.")
```