Chapter 8

Matrix Algebra

In applications, we often find ourselves dealing with very large numbers of equations of the form

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = c$$

where $a_1, a_2, \ldots, a_n$ and c are constants and $x_1, x_2, \ldots, x_n$ are (random or non-random) variables. Matrix algebra is indispensable in such situations. In this chapter we cover the basics: definitions, notation, and elementary operations, and its applications in solving systems of linear equations, and connections with vector spaces. A later chapter deals with more advanced topics like eigenvalues and matrix decompositions.

We will also learn how to work with matrices in Python as we proceed through this chapter. We will use the following libraries:

```
import numpy as np
import scipy.linalg as la
import sympy as sp
```

We import them here, and will not repeat these import statements in the code that follows.

The ideas in this chapter are closely connected to the concepts covered in Section 2.1.5 and Chapter 4 so you may wish to review that material prior to starting on this chapter.

8.1 Definitions and Notation

A **matrix** is a rectangular collection of numbers. The following is a matrix with m rows and n columns:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}.$$

Such a matrix is said to have “dimension” or “order” $m \times n$. The number that appears in the (i, j)th position, i.e., in the ith row and jth column, is called the (i, j)th element/entry/component of the matrix. We count rows from top to bottom, and columns from left to right. If $m = n$, the matrix is a **square matrix**. If $m = 1$ and $n > 1$, we have a **row vector**. If $m > 1$ and $n = 1$, we have a **column vector**. If $m = n = 1$, we have a **scalar**.

The term “vector” is used in many ways in mathematics. Sometimes a vector refers to an ordered list of numbers $(x_1, x_2, \ldots, x_n)$. Such an object has no "shape". It is merely an ordered sequence of n elements. Column and row vectors, on the other hand, are "two-dimensional" objects, in the sense of having a "height" (number of rows) and "width" (number of columns). In the context of matrix algebra, the word "vector" alone usually means a column vector, but not always.

Example 8.1 The matrix A below is a square matrix, b is a column vector and c is a row vector.

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \; b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}, \; c = \begin{bmatrix} c_1 & c_2 & \cdots & c_n \end{bmatrix}.$$

Matrices and vectors are often written in bold lettering, or with some sort of mark to distinguish them from scalars and other objects. We will not do so in this book. The reader will have to rely on context to distinguish scalars from vectors and matrices. Where context is unclear, we will be more explicit.

Some additional notation:

- It is often convenient to indicate an $m \times n$ matrix A by $(a_{ij})_{m\times n}$.
- We can refer to the (i,j)th element of a matrix A by $(A)_{ij}$ or $(A)_{i,j}$.

The utility of these two notational conventions should become clearer as the chapter progresses.

Two matrices of the same dimension are said to be equal if each of their corresponding elements are equal, i.e.,

$$A = B \Leftrightarrow (A)_{ij} = (B)_{ij} \;\text{ for all }\; i = 1, 2, \dots, m \;\text{ and }\; j = 1, 2, \dots, n.$$

Two matrices of different dimensions cannot be equal.

A **zero matrix** is one whose elements are all zero. It is simply written as 0 although sometimes subscripts are added to indicate its dimension.

The **diagonal** of an $n \times n$ square matrix refers to the (i,i)th elements of the matrix, i.e., to the elements $(A)_{ii}$, $i = 1, 2, \dots, n$. A **diagonal matrix** is a square matrix with all off-diagonal elements equal to zero, i.e., a square matrix A is diagonal if $(A)_{ij} = 0$ for all $i \neq j$, $i, j = 1, 2, ..., n$. Diagonal matrices are sometimes written $\text{diag}(a_{11}, a_{22}, ..., a_{nn})$.

Example 8.2 The matrix

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \text{diag}(1, 4, 0)$$

is a diagonal matrix. Note that there is nothing in the definition of a diagonal matrix that says its diagonal elements cannot be zero.[1]

[1] A square zero matrix is therefore technically also a diagonal matrix.

An **identity matrix** is a square matrix with all diagonal elements equal to one and all off-diagonal elements equal to zero, i.e.,

$$I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} = \operatorname{diag}(\underbrace{1, 1, \ldots, 1}_{n \text{ terms}}) \,.$$

We will denote an identity matrix by I. A subscript is sometimes added to indicate its dimension, as we did above, although this is often left out. We will see shortly that the identity matrix plays a role in matrix algebra akin to the role played by the number "1" in the real number system.

A **symmetric matrix** is a square matrix A such that $(A)_{ij} = (A)_{ji}$ for all $i, j = 1, 2, ..., n$.

Example 8.3 The matrix $\begin{bmatrix} 1 & 3 & 2 \\ 3 & 4 & 6 \\ 2 & 6 & 3 \end{bmatrix}$ is symmetric, $\begin{bmatrix} 1 & 3 & 2 \\ 7 & 4 & 6 \\ 2 & 6 & 3 \end{bmatrix}$ is not.

8.1.1 *Addition, Scalar Multiplication and Transpose*

Addition: Matrix addition is defined as element-by-element addition, i.e.., for two matrices $A = (a_{ij})_{m\times n}$ and $B = (b_{ij})_{m\times n}$, we define

$$(A + B)_{ij} = (A)_{ij} + (B)_{ij} \ \text{ for all } \ i = 1, \ldots, m\,;\ j = 1, \ldots, n\,.$$

Matrix addition is defined only for matrices of the same dimensions.

Example 8.4 $\begin{bmatrix} 1 & 4 \\ 3 & 2 \\ 6 & 5 \end{bmatrix} + \begin{bmatrix} 6 & 9 \\ 1 & 2 \\ 1 & 10 \end{bmatrix} = \begin{bmatrix} 1+6 & 4+9 \\ 3+1 & 2+2 \\ 6+1 & 5+10 \end{bmatrix} = \begin{bmatrix} 7 & 13 \\ 4 & 4 \\ 7 & 15 \end{bmatrix}$.

It should also be obvious that

$$A + B = B + A\,,$$
$$(A + B) + C = A + (B + C)\,.$$

This means that *as far as addition is concerned*, we can manipulate matrices in the same way we manipulate ordinary numbers (as long as the matrices being added have the same dimensions).

Scalar Multiplication: For a scalar α and matrix $A = (a_{ij})_{m\times n}$, we define

$$(\alpha A)_{ij} = (A\alpha)_{ij} = \alpha(A)_{ij} \ \text{ for all } \ i = 1, \ldots, m\,;\ j = 1, \ldots, n\,.$$

i.e., the product of a scalar and a matrix is defined to be the multiplication of each element of the matrix by the scalar.

Example 8.5 $b \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} = \begin{bmatrix} ba_{11} & ba_{12} \\ ba_{21} & ba_{22} \\ ba_{31} & ba_{32} \end{bmatrix}$.

We can use scalar multiplication to define **matrix subtraction**:

$$A - B = A + (-1)B.$$

Transpose: When we transpose a matrix, we write its rows as its columns, and its columns as its rows. That is, the transpose of an $(m \times n)$ matrix A, denoted either by A^{T} or A', is defined by

$$(A^{\mathrm{T}})_{ij} = (A)_{ji} \quad \text{for all} \quad i = 1, 2, ..., m \quad \text{and} \quad j = 1, 2, ..., n.$$

Example 8.6 $\begin{bmatrix} 1 & 4 \\ 3 & 2 \\ 6 & 5 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} 1 & 3 & 6 \\ 4 & 2 & 5 \end{bmatrix}$.

In order to use space more efficiently, we will often write a column vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

as $x = \begin{bmatrix} x_1 & x_2 & ... & x_m \end{bmatrix}^{\mathrm{T}}$ or $x^{\mathrm{T}} = \begin{bmatrix} x_1 & x_2 & ... & x_m \end{bmatrix}$.

We can use the transpose operator to define symmetric matrices: a symmetric matrix is simply a square matrix where $A^{\mathrm{T}} = A$.

8.1.2 *Exercises*

Ex. 8.1 What is the dimension of $A = \begin{bmatrix} 7 & 13 \\ 4 & 4 \\ 7 & 15 \end{bmatrix}$? What is $(A)_{1,2}$ and $(A)_{3,1}$?

Ex. 8.2 Suppose $A = (a_{ij})_{2\times 4}$ where $a_{ij} = i + j$. Write out the matrix in full.

Ex. 8.3 Express the following matrices in full:

i. $(a_{ij})_{4\times 4}$ where $a_{ij} = 1$ when $i = j$, 0 otherwise.

ii. $(a_{ij})_{4\times 4}$ where $a_{ij} = 0$ if $i \neq j$ (fill the rest of the entries with "$*$").

iii. $(a_{ij})_{4\times 4}$ where $a_{ij} = 0$ if $i > j$ (fill the rest of the entries with "$*$").

iv. $(a_{ij})_{4\times 4}$ where $a_{ij} = 0$ if $i < j$ (fill the rest of the entries with "$*$").

These are all square matrices. Matrix (c) is a "lower triangular matrix" and (d) is an "upper triangular matrix" (so we have in (c) and (d) matrices that are square and triangular!). Matrix (b) is diagonal, which is both upper and lower triangular.

Ex. 8.4 What is u and v if

$$\begin{bmatrix} u + 2v & 1 & 3 \\ 9 & 0 & 4 \\ 3 & 4 & 7 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 3 \\ 9 & 0 & u + v \\ 3 & 4 & 7 \end{bmatrix}?$$

Ex. 8.5 Let v_1, v_2, v_3, v_4 represent cities and suppose there are one-way flights from v_1 to v_2 and v_3, from v_2 to v_3 and v_4, and two-way flights between v_1 and v_4. Write out a matrix A such that $(A)_{ij} = 1$ if there is a flight from v_i to v_j, and zero otherwise.

Ex. 8.6 Let $A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$ and $B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$. Is $A = B$?

Ex. 8.7 If $2A = \begin{bmatrix} 3 & 4 \\ 2 & 8 \\ 1 & 5 \end{bmatrix}$, what is A? If $B - \frac{1}{2}\begin{bmatrix} 3 & 4 \\ 1 & 8 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 6 & 4 \\ 2 & 5 \\ 3 & 1 \end{bmatrix}$, what is B?

Ex. 8.8 Which of the following matrices are symmetric?

(a) $\begin{bmatrix} 1 & 2 & 3 & 5 \\ 2 & 5 & 4 & b \\ 3 & 4 & 3 & 3 \\ 5 & b & 3 & 1 \end{bmatrix}$ (b) $\begin{bmatrix} 1 & 1 & 3 & 5 \\ 2 & 5 & 4 & b \\ 3 & 4 & 3 & 3 \\ 5 & b & 3 & 1 \end{bmatrix}$ (c) $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ (d) $\begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$

Ex. 8.9 True or False?

i. Symmetric matrices must be square.
ii. A scalar is symmetric.
iii. If A is symmetric, then αA is symmetric.
iv. The sum of symmetric matrices is symmetric.
v. All diagonal matrices are symmetric.
vi. If $(A^{\mathrm{T}})^{\mathrm{T}} = A$, then A is symmetric.

Ex. 8.10 (a) Find A and B if they simultaneously satisfy

$$2A + B = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 3 & 0 \end{bmatrix} \quad \text{and} \quad A + 2B = \begin{bmatrix} 4 & 2 & 3 \\ 5 & 1 & 1 \end{bmatrix} .$$

(b) If $A + B = C$ and $3A - 2B = 0$ simultaneously, find A and B in terms of C.

8.2 Matrix Multiplication

Let A be $m \times n$ and B be $n \times p$ — here we require the number of columns in A and the number of rows in B to be the same. Then the product AB is defined as the $m \times p$ matrix whose (i, j)th element is

$$(AB)_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} .$$

That is, the (i, j)th element of the product AB is defined as the sum of the product of the elements of the ith row of A with the corresponding elements in the jth column of B. Put another way, the $(i, j)th$ element of the product AB is the dot or inner product of the ith row of A with the jth column of B. For example, the $(1, 1)$th element of AB is

$$(AB)_{11} = \sum_{k=1}^{n} a_{1k} b_{k1} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + \cdots + a_{1n}b_{n1} .$$

The $(2,3)$th element of AB is

$$(AB)_{23} = \sum_{k=1}^{n} a_{2k}b_{k3} = a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33} + \cdots + a_{2n}b_{n3}\,,$$

Visually, for a product of a 3×3 matrix and a 3×2 matrix, we have

$$\begin{bmatrix} \boxed{a_{11} \quad a_{12} \quad a_{13}} \\ a_{21} \quad a_{22} \quad a_{23} \\ a_{31} \quad a_{32} \quad a_{33} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} b_{11} \\ b_{21} \\ b_{31} \end{matrix}} & \begin{matrix} b_{12} \\ b_{22} \\ b_{32} \end{matrix} \end{bmatrix} = \begin{bmatrix} \boxed{a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}} & \bullet \\ \bullet & \bullet \\ \bullet & \bullet \end{bmatrix}$$

$$\begin{bmatrix} \boxed{a_{11} \quad a_{12} \quad a_{13}} \\ a_{21} \quad a_{22} \quad a_{23} \\ a_{31} \quad a_{32} \quad a_{33} \end{bmatrix} \begin{bmatrix} \begin{matrix} b_{11} \\ b_{21} \\ b_{31} \end{matrix} & \boxed{\begin{matrix} b_{12} \\ b_{22} \\ b_{32} \end{matrix}} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{3} a_{1k}b_{k1} & \boxed{a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}} \\ \bullet & \bullet \\ \bullet & \bullet \end{bmatrix}$$

$$\begin{bmatrix} a_{11} \quad a_{12} \quad a_{13} \\ \boxed{a_{21} \quad a_{22} \quad a_{23}} \\ a_{31} \quad a_{32} \quad a_{33} \end{bmatrix} \begin{bmatrix} \boxed{\begin{matrix} b_{11} \\ b_{21} \\ b_{31} \end{matrix}} & \begin{matrix} b_{12} \\ b_{22} \\ b_{32} \end{matrix} \end{bmatrix} = \begin{bmatrix} \sum_{k=1}^{3} a_{1k}b_{k1} & \sum_{k=1}^{3} a_{1k}b_{k2} \\ \boxed{a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31}} & \bullet \\ \bullet & \bullet \end{bmatrix}$$

and so on.

Example 8.7 Let $A = \begin{bmatrix} 2 & 8 \\ 3 & 0 \\ 5 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 4 & 7 \\ 6 & 9 \end{bmatrix}$. Then

$$AB = \begin{bmatrix} 2 & 8 \\ 3 & 0 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 4 & 7 \\ 6 & 9 \end{bmatrix} = \begin{bmatrix} 2\cdot 4 + 8\cdot 6 & 2\cdot 7 + 8\cdot 9 \\ 3\cdot 4 + 0\cdot 6 & 3\cdot 7 + 0\cdot 9 \\ 5\cdot 4 + 1\cdot 6 & 5\cdot 7 + 1\cdot 9 \end{bmatrix} = \begin{bmatrix} 56 & 86 \\ 12 & 21 \\ 26 & 44 \end{bmatrix}.$$

Example 8.8 The system of equations

$$\begin{aligned} 2x_1 - x_2 &= 4 \\ x_1 + 2x_2 &= 2 \end{aligned}$$

can be written in matrix form as

$$\begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix}, \quad \text{or} \quad Ax = b$$

where $A = \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}$, $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, and $b = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$.

8.2.1 *Exercises*

These exercises illustrate crucial aspects of matrix multiplication. You should work through each exercise and be sure to understand the point being made.

Ex. 8.11 Let $A = \begin{bmatrix} 2 & 8 \\ 3 & 0 \\ 5 & 1 \end{bmatrix}$, $B = \begin{bmatrix} 2 & 0 \\ 3 & 8 \end{bmatrix}$ and $C = \begin{bmatrix} 7 & 2 \\ 6 & 3 \end{bmatrix}$.

(a) Compute the matrices BC, CB, and AB.

(b) Can BA even be computed?

Remark: This exercise shows that for any two matrices A and B, $AB \neq BA$ in general. That is, we have to distinguish between pre-multiplication and post-multiplication. In the product AB, we say that B is pre-multiplied by A, or that A is post-multiplied by B.

Ex. 8.12 Show that $x^{\mathrm{T}}x \geq 0$ for any vector $x = \begin{bmatrix} x_1 & x_2 & \ldots & x_n \end{bmatrix}^{\mathrm{T}}$. When will $x^{\mathrm{T}}x = 0$?
Remark: For any column vector x, the product $x^{\mathrm{T}}x$ is the sum of the squares of its elements. That is, it is the dot or inner product of the column vector x with itself. In matrix algebra contexts, "inner product" seems to be preferred.

Ex. 8.13

(a) Compute $\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix}$.

(b) Let $A = \begin{bmatrix} 1 & b \\ -\frac{1}{b} & -1 \end{bmatrix}$ where $b \neq 0$. Compute the product $A^2 = AA$.

Remark: This exercise shows that you can multiply two non-zero matrices and end up with a zero matrix. Therefore $AB = 0$ does **not** *imply $A = 0$ or $B = 0$. It is even possible for the square of a non-zero matrix to be a zero matrix. Of course, if $A = 0$ or $B = 0$, then $AB = 0$.*

As you can see, in some ways matrix multiplication does not behave like the usual multiplication of numbers. For instance, the order of multiplication matters, and $AB = 0$ does not imply $A = 0$ or $B = 0$. But in other ways matrix multiplication *does* behave like regular multiplication of numbers, as the next exercise shows.

Ex. 8.14 Prove that

(a) $(AB)C = A(BC)$ where A, B, and C are $m \times n$, $n \times p$ and $p \times q$ respectively.

(b) $A(B + C) = AB + AC$ where A is $m \times n$, and B and C are $n \times p$.

(c) $(A + B)C = (AC + BC)$ where A and B are $m \times n$ and C is $n \times p$.

Remark: Although $(AB)C = A(BC)$, one approach may be more computationally efficient than the other. Suppose A and B are both $n \times n$, and C is $n \times 1$. Computing AB requires calculating n^2 inner products of vectors with n elements. Each of these inner products require n multiplications and $n - 1$ additions. These means the product AB requires $n^2(2n - 1)$ operations. Then multiplying the $n \times n$ matrix AB with C involves n inner products. The total number of operations required to compute $(AB)C$ is therefore $n(n + 1)(2n - 1)$. The product $A(Bc)$ however requires only $2n$ inner products, or $2n(2n - 1)$ operations. The ratio *of the number of operations required to compute $(AB)C$ to that required to compute $A(BC)$ is of order $O(n)$. As an exercise, you should calculate the number of operations required for both when $n = 100$.*

Ex. 8.15 Let A be an $m \times n$ matrix, and let I_n and I_m be identity matrices of dimensions $n \times n$ and $m \times m$ respectively. Show that $I_m A = AI_n = A$.

Ex. 8.16 Show that

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \\ a_{41} & a_{42} & a_{43} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = b_1 \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{bmatrix} + b_2 \begin{bmatrix} a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \end{bmatrix} + b_3 \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{bmatrix}$$

i.e., Ab is a linear combination of the columns of A, with weights given in b.

Ex. 8.17 (a) Show that $(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}$ for any $m \times n$ matrix A and any $n \times p$ matrix B. Verify this equality for the matrices

$$A = \begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \end{bmatrix} \text{ and } B = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_4 & b_5 & b_6 \\ b_7 & b_8 & b_9 \end{bmatrix}.$$

(b) Prove that $(ABC)^{\mathrm{T}} = C^{\mathrm{T}}B^{\mathrm{T}}A^{\mathrm{T}}$.

Ex. 8.18 Explain why $X^{\mathrm{T}}X$ is square and symmetric for any general $n \times k$ matrix X.

Remark: The matrix $X^{\mathrm{T}}X$ is encountered frequently in all statistical disciplines.

Ex. 8.19 The **trace** of an $n \times n$ matrix $A = (a_{ij})_{n\times n}$ is defined to be

$$\mathrm{tr}(A) = \sum_{i=1}^{n} a_{ii}.$$

That is, the trace of a square matrix is simply the sum of its diagonal elements. The trace of a scalar is the scalar itself.

(a) If A and B are square matrices of the same dimensions, show that

$$\mathrm{tr}(A + B) = \mathrm{tr}(A) + \mathrm{tr}(B).$$

(b) If A is a square matrix, show that $\mathrm{tr}(A^{\mathrm{T}}) = \mathrm{tr}(A)$.

(c) If A is $m \times n$ and B is $n \times m$, show that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$.

(d) If x is an $n \times 1$ column vector, show that $x^{\mathrm{T}}x = \mathrm{tr}(xx^{\mathrm{T}})$ by

i. direct multiplication,

ii. using (c) and the fact that the trace of a scalar is the scalar itself.

This odd little matrix operation is surprisingly useful in proofs and for deriving and simplifying matrix equations.

Ex. 8.20 Let i_n be an $n \times 1$ vector of ones, i.e., $i_n = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^{\mathrm{T}}$.

(a) Show that the formula for the sample mean of the elements of the column vector $y = \begin{bmatrix} y_1 & y_2 & \cdots & y_n \end{bmatrix}^{\mathrm{T}}$ can be written as $\overline{y} = (i_n^{\mathrm{T}} i_n)^{-1} i_n^{\mathrm{T}} y$.

(b) Show that $M_0 = I_n - i_n(i_n^{\mathrm{T}} i_n)^{-1} i_n^{\mathrm{T}}$ is symmetric, and that $M_0 M_0 = M_0$.

(c) Show that the sample variance of the data in y can be written as

$$\frac{1}{n-1} \sum_{i=1}^{n} (y_i - \overline{y})^2 = \frac{y^{\mathrm{T}} M_0 y}{n-1}.$$

Ex. 8.21 Prove that $A(\alpha B) = (\alpha A)B = \alpha(AB)$.

8.3 Partitioned Matrices

We can partition the contents of an $m \times n$ matrix into blocks of submatrices. For instance, we can write

$$A = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{bmatrix} = \left[\begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

where

$$A_{11} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}, A_{21} = \begin{bmatrix} 3 \\ 4 \\ 3 \end{bmatrix}, A_{12} = \begin{bmatrix} 3 & 2 & 6 \\ 8 & 2 & 1 \end{bmatrix} \text{ and } A_{22} = \begin{bmatrix} 1 & 2 & 4 \\ 2 & 1 & 3 \\ 1 & 1 & 7 \end{bmatrix}.$$

Partitioned matrices are often called **block matrices**. Of course, there are many ways of partitioning any given matrix. The following is another partition of the matrix A:

$$A = \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{bmatrix} = \left[\begin{array}{cc|cc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ 3 & 1 & 2 & 4 \\ \hline 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array}\right].$$

It can be shown that addition and multiplication of partitioned matrices can be carried out as though the blocks are elements, as long as the matrices are partitioned conformably.

Addition of Partitioned Matrices. Consider two $m \times n$ matrices A and B partitioned in the following manner:

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times n_1} & \underbrace{A_{12}}_{m_1 \times n_2} \\ \underbrace{A_{21}}_{m_2 \times n_1} & \underbrace{A_{22}}_{m_2 \times n_2} \end{bmatrix} \text{ and } B = \begin{bmatrix} \underbrace{B_{11}}_{m_1 \times n_1} & \underbrace{B_{12}}_{m_1 \times n_2} \\ \underbrace{B_{21}}_{m_2 \times n_1} & \underbrace{B_{22}}_{m_2 \times n_2} \end{bmatrix}$$

where $n_1 + n_2 = n$ and $m_1 + m_2 = m$. We emphasize that A and B must be of the same size and partitioned identically. Then

$$A + B = \begin{bmatrix} \underbrace{A_{11} + B_{11}}_{m_1 \times n_1} & \underbrace{A_{12} + B_{12}}_{m_1 \times n_2} \\ \underbrace{A_{21} + B_{21}}_{m_2 \times n_1} & \underbrace{A_{22} + B_{22}}_{m_2 \times n_2} \end{bmatrix}. \tag{8.1}$$

Multiplication of Partitioned Matrices. Now consider two matrices A and B with dimensions $m \times p$ and $p \times n$ respectively, are partitioned as follows:

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times p_1} & \underbrace{A_{12}}_{m_1 \times p_2} \\ \underbrace{A_{21}}_{m_2 \times p_1} & \underbrace{A_{22}}_{m_2 \times p_2} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \underbrace{B_{11}}_{p_1 \times n_1} & \underbrace{B_{12}}_{p_1 \times n_2} \\ \underbrace{B_{21}}_{p_2 \times n_1} & \underbrace{B_{22}}_{p_2 \times n_2} \end{bmatrix}.$$

In particular, the partition is such that the column-wise partition of A matches the row-wise partition of B. Then

$$\begin{aligned} AB &= \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times p_1} & \underbrace{A_{12}}_{m_1 \times p_2} \\ \underbrace{A_{21}}_{m_2 \times p_1} & \underbrace{A_{22}}_{m_2 \times p_2} \end{bmatrix} \begin{bmatrix} \underbrace{B_{11}}_{p_1 \times n_1} & \underbrace{B_{12}}_{p_1 \times n_2} \\ \underbrace{B_{21}}_{p_2 \times n_1} & \underbrace{B_{22}}_{p_2 \times n_2} \end{bmatrix} \\ &= \begin{bmatrix} \underbrace{A_{11}B_{11} + A_{12}B_{21}}_{m_1 \times n_1} & \underbrace{A_{11}B_{12} + A_{12}B_{22}}_{m_1 \times n_2} \\ \underbrace{A_{21}B_{11} + A_{22}B_{21}}_{m_2 \times n_1} & \underbrace{A_{21}B_{12} + A_{22}B_{22}}_{m_2 \times n_2} \end{bmatrix}. \end{aligned} \tag{8.2}$$

Transposition of Partitioned Matrices. It is straightforward to show that

$$A = \begin{bmatrix} \underbrace{A_{11}}_{m_1 \times n_1} & \underbrace{A_{12}}_{m_1 \times n_2} \\ \underbrace{A_{21}}_{m_2 \times n_1} & \underbrace{A_{22}}_{m_2 \times n_2} \end{bmatrix} \quad \Rightarrow \quad A^{\mathrm{T}} = \begin{bmatrix} \underbrace{A_{11}^{\mathrm{T}}}_{n_1 \times m_1} & \underbrace{A_{21}^{\mathrm{T}}}_{n_1 \times m_2} \\ \underbrace{A_{12}^{\mathrm{T}}}_{n_2 \times m_1} & \underbrace{A_{22}^{\mathrm{T}}}_{n_2 \times m_2} \end{bmatrix}. \tag{8.3}$$

Remark on Matrix Multiplication: So far we have spoken of inner products of vectors, scalar multiplication (multiplication of of matrices and vectors with a scalar), and regular matrix multiplication. There are yet other kinds of matrix multiplication concepts. For instance, the **Hadamard**[2] **product**, denoted $\circ$ or $\odot$, refers to element-wise multiplication, e.g.,

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \odot \begin{bmatrix} 2 & 3 \\ 4 & 5 \\ 6 & 7 \end{bmatrix} = \begin{bmatrix} 1 \cdot 2 & 2 \cdot 3 \\ 3 \cdot 4 & 4 \cdot 5 \\ 5 \cdot 6 & 6 \cdot 7 \end{bmatrix} = \begin{bmatrix} 2 & 6 \\ 12 & 20 \\ 30 & 42 \end{bmatrix}.$$

The **Kronecker**[3] **product**, denoted $\otimes$, of an $m \times n$ matrix A with a $p \times q$ matrix B is the $mp \times nq$ block matrix formed by multiplying each element of A by the entire B matrix. For example

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \left[\begin{array}{cc|cc|cc} a_{11} & 0 & a_{12} & 0 & a_{13} & 0 \\ 0 & a_{11} & 0 & a_{12} & 0 & a_{13} \\ \hline a_{21} & 0 & a_{22} & 0 & a_{23} & 0 \\ 0 & a_{21} & 0 & a_{22} & 0 & a_{23} \end{array}\right].$$

---

[2]Named after the French mathematician Jacques Hadamard (1865-1963).

[3]Named after the German mathematician Leopold Kronecker (1823-1891)

8.3.1 *Exercises*

Ex. 8.22 Let

$$A = \left[\begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array}\right] \quad \text{and} \quad B = \left[\begin{array}{c|cc} 2 & 0 & 1 \\ \hline 3 & 1 & 3 \\ 1 & 5 & 4 \\ 4 & 1 & 1 \end{array}\right].$$

Verify the partitioned matrix multiplication formulas by computing AB in the usual way, then compute AB using (8.2). Verify the transposition formula (8.3) for matrix A.

Ex. 8.23 Let A be a $m \times n$ matrix and b be an $n \times 1$ vector. We have shown earlier that Ab is a linear combination of the columns of A. In terms of partitioned matrices, we have

$$Ab = \left[\begin{array}{c|c|c|c} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array}\right] \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \begin{bmatrix} A_{*1} & A_{*2} & \cdots & A_{*n} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

$$= A_{*1}b_1 + A_{*2}b_2 + \cdots + A_{*n}b_n$$

Let $c = \begin{bmatrix} c_1 & c_2 & \ldots & c_m \end{bmatrix}^{\mathrm{T}}$. Show that $c^{\mathrm{T}}A$ is a linear combination of the rows of A.

Ex. 8.24 Let X be an $n \times 3$ data matrix containing n observations of three variables:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \\ \vdots & \vdots & \vdots \\ x_{n1} & x_{n2} & x_{n3} \end{bmatrix}$$

where x_{ij} represents the ith observation of variable j. We can partition this matrix to emphasize the variables by writing X as $X = \begin{bmatrix} X_{*1} & X_{*2} & X_{*3} \end{bmatrix}$ where

$$X_{*1} = \begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \\ \vdots \\ x_{n1} \end{bmatrix}, \; X_{*2} = \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \\ \vdots \\ x_{n2} \end{bmatrix} \; \text{and} \; X_{*3} = \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \\ \vdots \\ x_{n3} \end{bmatrix}.$$

Alternatively, we can partition the data matrix to emphasize the observations:

$$X = \begin{bmatrix} X_{1*} \\ X_{2*} \\ X_{3*} \\ \vdots \\ X_{n*} \end{bmatrix}$$

where $X_{i*} = [x_{i1} \quad x_{i2} \quad x_{i3}]$ is the row vector containing the ith observations of all three variables, $i = 1, 2, ..., n$. Show that the matrix $X^{\mathrm{T}}X$ can be written as

$$X^{\mathrm{T}}X = \begin{bmatrix} X_{*1}^{\mathrm{T}}X_{*1} & X_{*1}^{\mathrm{T}}X_{*2} & X_{*1}^{\mathrm{T}}X_{*3} \\ X_{*2}^{\mathrm{T}}X_{*1} & X_{*2}^{\mathrm{T}}X_{*2} & X_{*2}^{\mathrm{T}}X_{*3} \\ X_{*3}^{\mathrm{T}}X_{*1} & X_{*3}^{\mathrm{T}}X_{*2} & X_{*3}^{\mathrm{T}}X_{*3} \end{bmatrix} = \sum_{i=1}^{n} X_{i*}^{\mathrm{T}}X_{i*}$$

$$= \begin{bmatrix} \sum_{i=1}^n x_{i1}^2 & \sum_{i=1}^n x_{i1}x_{i2} & \sum_{i=1}^n x_{i1}x_{i3} \\ \sum_{i=1}^n x_{i1}x_{i2} & \sum_{i=1}^n x_{i2}^2 & \sum_{i=1}^n x_{i2}x_{i3} \\ \sum_{i=1}^n x_{i1}x_{i3} & \sum_{i=1}^n x_{i2}x_{i3} & \sum_{i=1}^n x_{i3}^2 \end{bmatrix}$$

8.4 Python Programming: Matrix Fundamentals

In this section, you will learn how to create matrices, check their attributes, index them, and extract submatrices using `numpy`. You will also explore how `numpy` and `scipy` can be used for basic matrix algebra.

8.4.1 *NumPy Matrix Creation*

Like vectors, matrices can be created in Python using the NumPy function `np.array()`. The following code defines and prints a 3×2 matrix:

```
A = np.array([[0, 1], [1, 2], [2, 3]])
print(A)
```

```
[[0 1]
 [1 2]
 [2 3]]
```

Notice how the code uses a nested list, where each inner list represents a row of the matrix. This is an example of a 2D NumPy `ndarray`. As discussed in Chapter 4, `np.array()` can also be used to create 3D arrays (matrices stacked along a new dimension) and even higher-dimensional arrays.

Similarly, we can create a corresponding column vector (a 3×1 matrix):

```
x = np.array([[0], [1], [2]])
print(x)
```

```
[[0]
 [1]
 [2]]
```

An alternative to using nested lists is the `.reshape()` method in NumPy for creating matrices. This method returns a new array with the same data but a different shape, where the shape is specified as a tuple:

```
A = np.array([0, 1, 1, 2, 2, 3]).reshape((3, 2))  # Create a 3 x 2 matrix
print(f"A 3 x 2 matrix:\n{A}\n")

x = np.array([0, 1, 2]).reshape((3, 1))  # Create a 3 x 1 matrix (column vector)
print(f"A 3 x 1 matrix (column vector):\n{x}")
```

```
A 3 x 2 matrix:
[[0 1]
 [1 2]
 [2 3]]

A 3 x 1 matrix (column vector):
[[0]
 [1]
 [2]]
```

The `.reshape()` method allows you to easily convert arrays into different shapes. It is not only useful for creating matrices but also for constructing stacked arrays and reshaping data for various purposes.

You can use `np.eye(n)` to create an identity matrix of size $n \times n$:

```
I = np.eye(3)
print(f"A 3 x 3 identity matrix:\n{I}")
```

```
A 3 x 3 identity matrix:
[[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]
```

More generally, `np.eye(m, n)` creates an $m \times n$ matrix with ones on the main diagonal and zeros elsewhere. You can also use NumPy's array creation functions listed in Table 4.2 to create matrices. For example:

```
zeros_matrix = np.zeros(shape=(2, 3), dtype=int)
print(f"A 2 x 3 zero matrix:\n{zeros_matrix}\n")

ones_matrix = np.ones(shape=(3, 2), dtype=int)
print(f"A 3 x 2 ones matrix:\n{ones_matrix}")
```

```
A 2 x 3 zero matrix:
[[0 0 0]
 [0 0 0]]

A 3 x 2 ones matrix:
[[1 1]
 [1 1]
 [1 1]]
```

8.4.2 *NumPy Array Attributes*

Note that the elements of a NumPy `ndarray` must be of the same type. You can check an array's type using its `.dtype` attribute:

```
print(A.dtype)
```

```
int64
```

In Python, `int64` is a 64-bit integer data type commonly used in libraries like `numpy` and `pandas`. It can store whole numbers in the range -2^{63} to $2^{63}-1$, making it suitable for large numerical computations and data analysis.

Table 8.1 lists some `numpy` array attributes:

Table 8.1. Important `numpy` array attributes.

Attribute	Description
`A.ndim`	Number of array dimensions
`A.shape`	Dimensions of the array as a tuple
`A.size`	Number of elements in the array
`A.dtype`	Data type of array elements

The total number of elements in a NumPy array is equal to the product of its dimensions. This is illustrated below using a 3D array consisting of three stacked 2×2 matrices:

```
stacked_matrices = np.array([[[0, 1], [1, 2]],
                             [[1, 2], [2, 3]],
                             [[2, 3], [3, 4]]])
print(stacked_matrices.ndim, stacked_matrices.shape,
      stacked_matrices.size)
```

```
3 (3, 2, 2) 12
```

The object `stacked_matrices` is an example of a tensor. In the context of data science, a **tensor** is an n-dimensional array of numbers:

```
print(stacked_matrices)
```

```
[[[0 1]
  [1 2]]

 [[1 2]
  [2 3]]
```

```
[[2 3]
 [3 4]]]
```

The term originates from mathematics and physics, where it refers to objects that generalize scalars (0D), vectors (1D), and matrices (2D) to higher dimensions. In Python, tensors are typically represented using NumPy's `ndarray`.[4]

For example, imagine you have 10 grayscale images, each of size 28×28 pixels—as in the famous MNIST dataset (*bit.ly/3To36is*) . Each image can be represented as a 2D NumPy array of shape `(28, 28)`, where each value corresponds to the intensity of a pixel. If you stack all 10 images together, you obtain a 3D tensor of shape `(10, 28, 28)`. Similarly, a batch of 100 RGB images of resolution 224×224 (with 3 color channels: Red, Green, and Blue) can be represented as a 4D tensor of shape `(100, 3, 224, 224)`. Such tensors are commonly used as input data for computer vision models that perform tasks such as image classification and object detection.

8.4.3 *NumPy Array Indexing*

In a multi-dimensional NumPy array, elements can be accessed using a comma-separated tuple of indices. For example (since Python uses zero-based indexing):

```python
A_12 = A[(0, 1)]
print(f"The element in row 1, column 2 of A is: {A_12}")
```

```
The element in row 1, column 2 of A is: 1
```

In NumPy, both `A[0, 1]` and `A[(0, 1)]` are functionally equivalent. The tuple `(0, 1)` and the comma-separated indices `0, 1` are interpreted the same way. However, comma-separated indexing is typically preferred because it is cleaner and more readable:

```python
A_21 = A[1, 0]
print(f"The element in row 2, column 1 of A is: {A_21}\n")

print(f"The third element of x is: {x[2, 0]}")
```

```
The element in row 2, column 1 of A is: 1

The third element of x is: 2
```

Similarly, array elements can also be modified:

[4]However, while tensors can be represented by such numerical arrays, they hold a deeper geometric meaning for physicists and mathematicians. To them, tensors are used to describe geometric objects and their transformations.

```
A[0, 0] = 10
print(A)
```

```
[[10  1]
 [ 1  2]
 [ 2  3]]
```

8.4.4 *NumPy Array Slicing*

Recall that the contents of an $m \times n$ matrix can be partitioned into blocks of submatrices. Each submatrix can be extracted using array slicing. For example:

```
D = np.arange(start=1, stop=10, step=1).reshape((3, 3))
print(f"The matrix D is:\n{D}\n")

submatrix = D[1:3, :]
print(f"The last two rows of D are:\n{submatrix}")
```

```
The matrix D is:
[[1 2 3]
 [4 5 6]
 [7 8 9]]

The last two rows of D are:
[[4 5 6]
 [7 8 9]]
```

Here, `D[1:3, :]` uses array slicing to extract rows 2 and 3 (Python uses zero-based indexing and excludes the stop index), along with all columns (`:`). The result is a submatrix containing the second and third rows of `D`.

Similarly, individual rows or columns can be extracted:

```
first_row = D[0, :]
print(f"The first row of D is:\n{first_row}\n")

first_column = D[:, 0]
print(f"The first column of D is:\n{first_column}")
```

```
The first row of D is:
[1 2 3]

The first column of D is:
[1 4 7]
```

Note that `D[:, 0]` extracts a 1D array, not a column vector.

8.4.5 *Basic Matrix Algebra*

We will use the following two matrices for illustration:

```
A = np.array([[2, 2], [3, 1]])
B = np.array([[2, -1], [4, 5]])

print(f"The matrix A is:\n\n{A}\n")
print(f"The matrix B is:\n\n{B}")
```

```
The matrix A is:

[[2 2]
 [3 1]]

The matrix B is:

[[ 2 -1]
 [ 4  5]]
```

In NumPy, the standard Python operators for addition (`+`), subtraction (`-`), multiplication (`*`), division (`/`), and exponentiation (`**`) perform element-wise operations on arrays. For example:

```
print(f"The sum of A and B is:\n\n{A + B}\n")
print(f"The difference between B and A is:\n\n{B - A}")
```

```
The sum of A and B is:

[[4 1]
 [7 6]]

The difference between B and A is:

[[ 0 -3]
 [ 1  4]]
```

If one operand is a scalar, the operation applies to each element of the array. For example:

```
print(f"The product of 2 and B is:\n\n{2 * B}")
```

```
The product of 2 and B is:

[[ 4 -2]
 [ 8 10]]
```

NumPy provides basic tools for linear algebra through its `numpy.linalg` submodule. For more advanced applications, however, it is generally better to use `scipy.linalg` (already imported as `la`) which offers additional functionality and optimized performance. Table 8.2 lists some `scipy` functions for basic matrix algebra:

Table 8.2. Important `scipy` matrix algebra functions.

Expression	Description
`A.T`	Matrix transpose, A^{T}
`A + B`	Matrix addition, $A + B$
`A - B`	Matrix subtraction, $A - B$
`A @ B`	Matrix multiplication, AB
`A * B`	Hadamard (element by element) Product, $A \odot B$
`la.det(A)`	Matrix determinant, $\det(A)$
`la.inv(A)`	Matrix inverse, A^{-1}

```python
print(f"The transpose of B is:\n{B.T}\n")

print(f"The matrix product of A and B is:\n{A @ B}\n")

print(f"The Hadamard product of A and B is:\n{A * B}\n")

# For scalar multiplication, use "*"
print(f"The scalar multiplication of B by 2 is:\n{2 * B}")
```

```
The transpose of B is:
[[ 2  4]
 [-1  5]]

The matrix product of A and B is:
[[12  8]
 [10  2]]

The Hadamard product of A and B is:
[[ 4 -2]
 [12  5]]

The scalar multiplication of B by 2 is:
[[ 4 -2]
 [ 8 10]]
```

For Kronecker products, we use `np.kron`:

```python
K = np.kron(A, B)
print(f"The Kronecker product of A and B is:\n{K}")
```

```
The Kronecker product of A and B is:
```

```
[[ 4 -2  4 -2]
 [ 8 10  8 10]
 [ 6 -3  2 -1]
 [12 15  4  5]]
```

We discuss inverses and determinants in the next section.

8.4.6 *Exercises*

In addition to the exercises provided here, we recommend that you go through all of the exercises in the previous sections and, wherever appropriate, verify your solutions using the Python tools covered in this section.

Ex. 8.25 In linear algebra, a **Hilbert matrix** is an $n \times n$ matrix where the element in row i and column j is given by $h_{ij} = \frac{1}{i+j-1}$. This matrix arises in numerical analysis, especially in problems related to interpolation and approximation.

Use NumPy and a nested list to create and display a 3×3 Hilbert matrix `H`.

Ex. 8.26 Use a nested list comprehension to generate the elements of matrix `H` from Ex. 8.25, and then convert the result into a NumPy 2D array.

Ex. 8.27 In linear algebra, a **block diagonal matrix** is a square matrix partitioned into smaller square matrices (blocks) along its main diagonal, with all off-diagonal blocks being zero matrices of appropriate sizes.

(a) Use `np.zeros()`, `np.ones()`, and array slicing to create an 6×6 block diagonal matrix with three 2×2 blocks of ones along the main diagonal.

(b) Use `np.kron()` to create the same block diagonal matrix.

Ex. 8.28 Use array slicing to extract the submatrix corresponding to the first two rows and columns of the matrix `H` from Ex. 8.25.

Ex. 8.29 Recall that the trace of a square matrix is the sum of its diagonal elements. Use `np.trace(H)` to compute the trace, $\mathrm{tr}(H)$, of the matrix `H` from Ex. 8.25.

8.5 Introduction to Inverses and Determinants

8.5.1 *The Inverse Matrix*

The $n \times m$ matrix B is said to be a **left-inverse** of a $m \times n$ matrix A if $BA = I_n$. The $n \times m$ matrix C is a **right-inverse** of A if $AC = I_m$. If A is $n \times n$, and $BA = AC = I_n$, then it must be the case that $B = C$ since

$$BA = I_n \Rightarrow BAC = I_n C \Rightarrow BI_n = C \Rightarrow B = C.$$

In this case, we call $B = C$ the **two-sided inverse**, or simply the **inverse of A, and give it the special notation A^{-1}. That is, the inverse of an $n \times n$ matrix A, *if it exists*, is the unique matrix A^{-1} such that

$$A^{-1}A = I_n = AA^{-1}\,.$$

We could leave out the second equality from the definition, since as we have already shown, $A^{-1}A = I \Rightarrow AA^{-1} = I$.

Example 8.9 The inverse of the matrix

$$A = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad \text{is} \quad A^{-1} = -\frac{1}{2} \begin{bmatrix} 4 & -3 \\ -2 & 1 \end{bmatrix} .$$

This can be verified by direct multiplication:

$$A^{-1}A = -\frac{1}{2} \begin{bmatrix} 4 & -3 \\ -2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} .$$

We do not have to show $AA^{-1} = I_2$, since it is implied. You may wish to do so nonetheless, as an exercise.

Example 8.10 Let A and B be the matrices

$$A = \begin{bmatrix} 1 & 1 \\ 2 & 1 \\ 4 & 2 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} -1 & 0.2 & 0.4 \\ 2 & -0.2 & -0.4 \end{bmatrix} .$$

You can easily verify (by direct multiplication) that

$$BA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \text{but} \quad AB = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.2 & 0.4 \\ 0 & 0.4 & 0.8 \end{bmatrix} .$$

The matrix B is a left-inverse of A. We give left-inverses the special notation A^{-1}_{left}. Likewise, right-inverses are given the special notation A^{-1}_{right}. We will say more about left- and right-inverses in a later chapter. For this chapter we will focus on (two-sided) inverses. The term "inverse" will always mean a two-sided inverse.

We emphasize that A has a (two-sided) inverse only if it is square. Furthermore, not all square matrices have an inverse. The inverse of an arbitrary 2×2 matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, if it exists, is

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{21} & a_{11} \end{bmatrix} \quad \text{where } \det(A) = a_{11}a_{22} - a_{12}a_{21}. \tag{8.4}$$

You can easily verify this by direct multiplication. It is worth your while to commit formula (8.4) to memory.

The expression $\det(A)$ in (8.4) is called the **determinant** of the 2×2 matrix A. Notice that the inverse exists only if $\det(\mathrm{A}) \neq 0$. If the inverse of A does not exist, we say that A is **singular**. If the inverse exists, we say that A is **non-singular**. An alternative notation for $\det(A)$ is $|A|$. We will use both notations in this book. In particular, we use the latter when indicating the determinant of a matrix written out in full. For instance, the determinant of the matrix $(a_{ij})_{2\times 2}$ is

$$\det(A) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} .$$

Example 8.11 The inverse of the matrix $A = \begin{bmatrix} 1 & 4 \\ 5 & 6 \end{bmatrix}$ is

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} 6 & -4 \\ -5 & 1 \end{bmatrix} = -\frac{1}{14} \begin{bmatrix} 6 & -4 \\ -5 & 1 \end{bmatrix} = \begin{bmatrix} -\frac{3}{7} & \frac{2}{7} \\ \frac{5}{14} & -\frac{1}{14} \end{bmatrix}.$$

Example 8.12 The determinant of the matrix $A = \begin{bmatrix} 1 & 3 \\ 2 & 6 \end{bmatrix}$ is $\det(A) = 1 \cdot 6 - 2 \cdot 3 = 0$, so A does not have an inverse.

When will $\det(A) = 0$? Examining the expression for $\det(A)$ in (8.4), we see that the determinant will be zero if one or both rows or columns are all zero, or if one row is a multiple of the other, or if one column is a multiple of the other.

The inverse of a scalar is obviously just its reciprocal. The following example shows the inverse of a particular 3×3 matrix.

Example 8.13 The inverse of $A = \begin{bmatrix} 0 & 2 & 4 \\ 3 & 1 & 2 \\ 6 & 2 & 1 \end{bmatrix}$ is $A^{-1} = \begin{bmatrix} -\frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{2} & -\frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & -\frac{1}{3} \end{bmatrix}$.
This can be seen by direct multiplication:

$$\begin{bmatrix} -\frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{2} & -\frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & -\frac{1}{3} \end{bmatrix} \begin{bmatrix} 0 & 2 & 4 \\ 3 & 1 & 2 \\ 6 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

How do we find the inverse of a general 3×3 and larger square matrices? There is a formula for the inverse of a general $n \times n$ matrix which we will present in the appendix to this chapter. There is also a computationally efficient *algorithmic* approach based on Gaussian elimination, which we will discuss shortly. Nonetheless, even without seeing the formula or algorithms for computing the inverse of a matrix, we are able to make the following general statements. Suppose the $n \times n$ matrices A and B are non-singular, i.e., their inverses exist. Then

i. $(A^{-1})^{\mathrm{T}} = (A^{\mathrm{T}})^{-1}$,

ii. $(AB)^{-1} = B^{-1}A^{-1}$.

Proof: For i., start with $AA^{-1} = I$. Transpose both sides to get $(A^{-1})^{\mathrm{T}}A^{\mathrm{T}} = I$. Finally post-multiply both sides by $(A^{\mathrm{T}})^{-1}$ to get

$$(A^{-1})^{\mathrm{T}}A^{\mathrm{T}}(A^{\mathrm{T}})^{-1} = I(A^{\mathrm{T}})^{-1} \;\Rightarrow\; (A^{-1})^{\mathrm{T}} = (A^{\mathrm{T}})^{-1}.$$

For ii., pre-multiply AB first by A^{-1} and then by B^{-1}. This gives

$$A^{-1}AB = B$$
$$B^{-1}A^{-1}AB = B^{-1}B = I.$$

This says that $B^{-1}A^{-1}$ is the inverse of AB since multiplying the two gives the identity matrix.

One implication of the first result is that the inverse of a symmetric matrix is symmetric: if A is symmetric, then $A^{\mathrm{T}} = A$, so we have

$$(A^{-1})^{\mathrm{T}} = (A^{\mathrm{T}})^{-1} = A^{-1}$$

which says that A^{-1} is symmetric. For the second result, it is important to keep in mind that this result holds only if A and B are both square. It is possible for A to be $n \times k$ and B to be $k \times n$ such that the square matrix AB is non-singular. But since A and B are not square, they do not have inverses. In that case the statement $(AB)^{-1} = B^{-1}A^{-1}$ is obviously meaningless.

8.5.2 *Systems of Linear Equations*

One application of matrix inverses is to find solutions to systems of linear equations. Consider a system of n equations in n unknowns x_1, x_2, ..., x_n,

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n \end{array} \tag{8.5}$$

which can be written as

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} \quad \text{or} \quad Ax = b\,.$$

To be clear, we are speaking here of systems where there are as many equations as there are unknowns. If the inverse of A exists, then the system has a unique solution, namely $x = A^{-1}b$, since

$$Ax = b \;\;\Rightarrow\;\; A^{-1}Ax = A^{-1}b \;\;\Rightarrow\;\; x = A^{-1}b\,.$$

Example 8.14 Consider the following systems of equations

$$\text{(i)}\;\; \begin{aligned} 2x_1 - \;\; x_2 &= 4 \\ x_1 + 2x_2 &= 2 \end{aligned} \qquad \text{(ii)}\;\; \begin{aligned} 2x_1 + \;\; x_2 &= 4 \\ 6x_1 + 3x_2 &= 12 \end{aligned} \qquad \text{(iii)}\;\; \begin{aligned} 2x_1 + \;\; x_2 &= 4 \\ 6x_1 + 3x_2 &= 10 \end{aligned} \tag{8.6}$$

You can see that system (i) has a unique solution. System (ii) has infinitely many solutions (the graphs of the two equations coincide). System (iii) has no solution; the graphs of the two equations are parallel. The three systems can be written in the matrix form $Ax = b$:

$$\text{(i)}\; \begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 2 \end{bmatrix} \quad \text{(ii)}\; \begin{bmatrix} 2 & 1 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 12 \end{bmatrix} \quad \text{(iii)}\; \begin{bmatrix} 2 & 1 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 10 \end{bmatrix}$$

Since

$$\begin{bmatrix} 2 & -1 \\ 1 & 2 \end{bmatrix}^{-1} = \frac{1}{5}\begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}$$

the unique solution for system (i) is

$$x = A^{-1}b = \frac{1}{5}\begin{bmatrix} 2 & 1 \\ -1 & 2 \end{bmatrix}\begin{bmatrix} 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

For systems (ii) and (iii), we find that the coefficient matrix A does not have an inverse, since

$$\det\begin{bmatrix} 2 & 1 \\ 6 & 3 \end{bmatrix} = 2 \cdot 3 - 1 \cdot 6 = 0\,.$$

Notice that non-existence of the coefficient matrix inverse does not imply that there are no solutions. It could be that there are multiple solutions.

8.5.3 *The Determinant and Cramer's Rule*[5]

Consider now the general 2×2 system

$$\begin{array}{ccccc} a_{11}x_1 & + & a_{12}x_2 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & = & b_2 \end{array} \quad \text{or} \quad Ax = b \tag{8.7}$$

Solving this system (say, by using Gaussian elimination) gives

$$x_1 = \frac{a_{22}\,b_1 - a_{12}\,b_2}{a_{11}a_{22} - a_{12}a_{21}} \quad \text{and} \quad x_2 = \frac{a_{11}\,b_2 - a_{21}\,b_1}{a_{11}a_{22} - a_{12}a_{21}}\,.$$

Of course, this is the solution only if the (common) denominator in both expressions is not zero. The denominator is just the determinant of the matrix A. Notice also that the numerators of the solutions for x_1 and x_2 are, respectively, the determinants of the matrices

$$A_1(b) = \begin{bmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{bmatrix} \quad \text{and} \quad A_2(b) = \begin{bmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{bmatrix}.$$

These are just the matrix A with one column replaced by b. This is **Cramer's Rule** for systems of two equations in two unknowns: for system (8.7), the solutions are

$$x_1 = \frac{\det(A_1(b))}{\det(A)} = \frac{\begin{vmatrix} b_1 & a_{12} \\ b_2 & a_{22} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}} \quad \text{and} \quad x_2 = \frac{\det(A_2(b))}{\det(A)} = \frac{\begin{vmatrix} a_{11} & b_1 \\ a_{21} & b_2 \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}}\,.$$

[5]Gabriel Cramer (1704-1752).

The idea extends to larger systems of equations with as many equations as unknowns. If you work out the solutions for the general three-equations three-unknowns system

$$\begin{array}{ccccccc} a_{11}x_1 & + & a_{12}x_2 & + & a_{13}x_3 & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & a_{23}x_3 & = & b_2 \\ a_{31}x_1 & + & a_{32}x_2 & + & a_{33}x_3 & = & b_3 \end{array}$$

you will find the solutions to be

$$x_1 = \frac{b_1a_{22}a_{33} + a_{12}a_{23}b_3 + a_{13}b_2a_{32} - a_{13}a_{22}b_3 - b_1a_{23}a_{32} - a_{12}b_2a_{33}}{a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}}$$

$$x_2 = \frac{a_{11}b_2a_{33} + b_1a_{23}a_{31} + a_{13}a_{21}b_3 - a_{13}b_2a_{31} - a_{11}a_{23}b_3 - b_1a_{21}a_{33}}{a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}}$$

$$x_3 = \frac{a_{11}a_{22}b_3 + a_{12}b_2a_{31} + b_1a_{21}a_{32} - b_1a_{22}a_{31} - a_{11}b_2a_{32} - a_{12}a_{21}b_3}{a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{11}a_{23}a_{32} - a_{12}a_{21}a_{33}}$$

You do not want to memorize this solution, at least not in this form. But notice two things: first, the denominator is the same for all three expressions. We define the expression in the denominator to be the determinant of the 3×3 coefficient matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

We must have $\det(A) \neq 0$ in order for there to be a unique solution. Second, using this definition for the determinant, the numerators in the solutions for x_1, x_2 and x_3 are, respectively, the determinants of the matrices

$$A_1(b) = \begin{bmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{bmatrix},\ A_2(b) = \begin{bmatrix} a_{11} & b_1 & a_{13} \\ a_{21} & b_2 & a_{23} \\ a_{31} & b_3 & a_{33} \end{bmatrix} \text{ and } A_3(b) = \begin{bmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{bmatrix}.$$

That is,

$$x_1 = \frac{\det(A_1(b))}{\det(A)},\ x_2 = \frac{\det(A_2(b))}{\det(A)} \text{ and } x_3 = \frac{\det(A_3(b))}{\det(A)}.$$

This is **Cramer's Rule** for systems of three equations in three unknowns.

The determinant for larger square matrices can be thought of in a similar way, as the (common) denominator in the solutions to the general n-equations in n-unknowns system $Ax = b$. Furthermore, the solution to such a system is

$$x_i = \frac{\det(A_i(b))}{\det(A)},\ i = 1, 2, \ldots, n$$

where $A_i(b)$ is the determinant of the matrix A with the ith column replaced by b. But what is the formula for the determinant of a general $n \times n$ matrix?

The following (set of) formulas are called the **Laplace expansions**[6], or the **cofactor expansions**, can be used to calculate determinants for general square matrices. First, define the determinant of a scalar to be the scalar itself. Then given an $n \times n$ matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

define the (i, j)th **cofactor** of A, denoted $C_{ij}(A)$, to be the determinant of the matrix with the ith row and jth column removed, multiplied by $(-1)^{i+j}$. Take for example the matrix

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} . \tag{8.8}$$

We have

$$\begin{aligned} C_{21}(A) &= (-1)^{2+1} \begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} = a_{13}a_{32} - a_{12}a_{33} \\ C_{33}(A) &= (-1)^{3+3} \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21} \end{aligned}$$

and so on. Then we have

$$\begin{aligned} \det(A) &= \sum_{i=1}^{n} a_{ij} C_{ij}(A) \;\; \text{for any column } \; j \\ &= \sum_{j=1}^{n} a_{ij} C_{ij}(A) \;\; \text{for any row } \; i \, . \end{aligned}$$

This formula is somewhat unusual compared with most other formulas that we have been working with. First, it is recursive: to calculate the determinant of an $n \times n$ matrix, we need to calculate its cofactors which involve determinants of $n-1 \times n-1$ matrices, and so on. Second, you can choose to expand along any row or any column. They all give you the same result. The first expression is called the Laplace expansion by column j. The second expression is called the Laplace expansion by row i.

For the general 3×3 matrix shown in (8.8), using the Laplace expansion

[6]Pierre-Simon Laplace (1749-1827) made important contributions in physics and astronomy, and several areas of mathematics, including differential equations and probability theory.

along the 1st column, we have

$$\begin{aligned}\det(A) &= a_{11}C_{11}(A) + a_{21}C_{21}(A) + a_{31}C_{31}(A)\\ &= a_{11}(-1)^{1+1}\begin{vmatrix}a_{22} & a_{23}\\ a_{32} & a_{33}\end{vmatrix} + a_{21}(-1)^{1+2}\begin{vmatrix}a_{12} & a_{13}\\ a_{32} & a_{33}\end{vmatrix} + a_{31}(-1)^{1+3}\begin{vmatrix}a_{12} & a_{13}\\ a_{22} & a_{23}\end{vmatrix}\\ &= a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{21}(a_{12}a_{33} - a_{13}a_{32}) + a_{31}(a_{12}a_{23} - a_{13}a_{22})\end{aligned}$$

which you can verify to be the same as what was found earlier by solving the general 3-equations in 3-unknowns system explicitly. As an exercise you should verify that you get the same expressions by expanding along the other columns or along any of the three rows.

Computationally speaking, the Laplace expansion is inefficient in terms of the number of arithmetic operations required. We will show a much more computationally efficient method shortly. Nonetheless, the Laplace expansion is useful for deriving a number of properties of determinants, which we state below:

i. if A has a row of zeros or a column of zeros, then $\det(A) = 0$. You can see this by taking the Laplace expansion along this zero row/column.

ii. if a single row or column of A is multiplied by some constant α, then its determinant also gets multiplied by α. This is also easy to see from the Laplace expansion along the affected row or column. If row i is multiplied by α, then

$$\sum_{j=1}^{n} \alpha a_{ij} C_{ij}(A) = \alpha \sum_{j=1}^{n} a_{ij} C_{ij}(A) = \alpha \det(A)\,.$$

Of course, if a row *and* a column are both multiplied by α, then the determinant gets multiplied by α^2.

iii. The determinant of a triangular matrix is the product of its diagonal elements. For example, expanding along the top row throughout, we have

$$\begin{aligned}\begin{vmatrix}a_{11} & 0 & 0\\ a_{21} & a_{22} & 0\\ a_{31} & a_{32} & a_{33}\end{vmatrix} &= a_{11}\left((-1)^{1+1}\begin{vmatrix}a_{22} & 0\\ a_{32} & a_{33}\end{vmatrix}\right) + 0 + 0\\ &= a_{11}\left(a_{22}\left((-1)^{1+1}\det(a_{33})\right) + 0\right)\\ &= a_{11}a_{22}a_{33}\,.\end{aligned}$$

The following properties also follow from the Laplace expansion, though their proofs are less immediate, and are omitted here:

iv. $\det(A^{\mathrm{T}}) = \det(A)$.

v. Every time we swap the rows of a matrix, its determinant changes sign. Similarly for columns.

vi. Adding a multiple of one row to another row does not change the determinant. Similarly for columns.

vii. If A and B are two square matrices, then $\det(AB) = \det(A)\det(B)$.

You will recognize the actions in ii., v., and vi. to be operations used in Gaussian elimination.

8.5.4 *Gaussian Elimination Revisited*

The Gaussian elimination method for solving systems of equations involves three actions: swapping the order of equations, multiplying an equation by some constant, and adding (or subtracting) a multiple of one equation to another. Here we apply these "elementary row operations" to the rows of $n \times n$ matrices by pre-multiplying the matrix with certain other matrices. We illustrate with the 3×3 matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}.$$

Suppose we wish to swap the second and third rows. What matrix can we pre-multiply A by in order to achieve this? Denote such a matrix by $E_{[2]\leftrightarrow[3]}$. Suppose we apply this matrix to the identity matrix I_3. Since multiplying by the identity matrix does not change anything, we have

$$E_{[2]\leftrightarrow[3]} I_3 = E_{[2]\leftrightarrow[3]}\,.$$

But this says that the matrix for swapping the second and third rows of a matrix is the matrix obtained by swapping the second and third rows of the identity matrix.[7] That is, the 3×3 "row swap" matrix for swapping rows 2 and 3 is

$$E_{[2]\leftrightarrow[3]} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

This extends to the general case: the "row swap" matrix for swapping the i-th and j-th rows of A is the identity matrix with the i-th and j-th rows swapped.

Notice that $E_{[i]\leftrightarrow[j]}$ is its own inverse. If you swap the i-th and j-th rows of a matrix twice, you get back the same matrix. Since the "row swap" matrix is symmetric, we also have

$$E^{-1}_{[i]\leftrightarrow[j]} = E^{\mathrm{T}}_{[i]\leftrightarrow[j]} = E_{[i]\leftrightarrow[j]}\,.$$

[7]This trick works because the desired transformation matrix actually exists.

The "row swap" matrix is an example of a **permutation matrix**. The following permutation matrix

$$P_{3,1,2} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} = E_{[2]\leftrightarrow[3]} E_{[1]\leftrightarrow[3]}$$

re-arranges the rows of a matrix with three rows so that the third row becomes the first, the first is the second, and the second is the third. Permutation matrices can be obtained by applying multiple row swap operations. In general, a permutation matrix is **not** its own inverse. However, we still have $P^{-1} = P^{\mathrm{T}}$ for all permutation matrices.

We can apply the same logic as before to find the matrix that, when pre-multiplied to a matrix, multiplies the j-th row of that matrix by some constant. This transformation matrix is simply the identity matrix with its *j-th* row multiplied by the constant. For example, to multiply the third row of A by α, pre-multiply A with the matrix

$$E_{[3]\leftarrow\alpha[3]} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \alpha \end{bmatrix}.$$

The matrix $E_{[3]\leftarrow\alpha[3]}$ is obviously symmetric, and its inverse (when $\alpha \neq 0$) is easily seen to be

$$E^{-1}_{[3]\leftarrow\alpha[3]} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/\alpha \end{bmatrix}.$$

To find the matrix that, when pre-multiplied to another matrix, has the effect of adding a multiple of one row to another, carry out that same operation on the identity matrix. So, to add α times the first row of A to its third row, pre-multiply A by

$$E_{[3]\leftarrow[3]+\alpha[1]} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & 0 & 1 \end{bmatrix}.$$

You can easily verify that

$$\begin{aligned} E_{[3]\leftarrow[3]+\alpha[1]}A &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \\ &= \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ \alpha a_{11} + a_{31} & \alpha a_{12} + a_{32} & \alpha a_{13} + a_{33} \end{bmatrix} \end{aligned}$$

The inverse of $E_{[3]\leftarrow[3]+\alpha[1]}$ is just the same matrix with the sign of the constant reversed, i.e.,

$$E^{-1}_{[3]\leftarrow[3]+\alpha[1]} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -\alpha & 0 & 1 \end{bmatrix}.$$

This is easy to verify by direct multiplication, but the intuition is also straightforward: if you wish to reverse the action of adding a multiple of row i to row j, just subtract the same multiple of row i from the transformed row j.

8.5.4.1 *Gaussian Elimination and the Inverse* If the square matrix A has an inverse, you will be able to find a sequence of elementary row operations that reduces A down to the identity matrix. Since each row operation can be done by pre-multiplying A by some elementary row operation matrix, A will have an inverse if there is a sequence of "row-operation matrices" $E_1, E_2, \ldots, E_m$ such that

$$E_m E_{m-1} \times \cdots \times E_2 E_1 A = I.$$

It follows that $A^{-1} = E_m E_{m-1} \times \cdots \times E_2 E_1$. Practically, this implies the following procedure for finding the inverse of a matrix A. First place A next to the identity matrix

$$A \mid I\,.$$

Then apply elementary row operations to both sides until the left-hand side reduces to the identity matrix

$$\begin{array}{c|c} A & I \\ E_1 A & E_1 I \\ E_2 E_1 A & E_2 E_1 I \\ \vdots & \vdots \\ \underbrace{E_m E_{m-1} \times \cdots \times E_2 E_1 A}_{I} & \underbrace{E_m E_{m-1} \times \cdots \times E_2 E_1}_{A^{-1}} I. \end{array}$$

Example 8.15 We find the inverse of the matrix in Example 8.13 using Gaussian Elimination. The row echelon form of the matrix is indicated, and the pivots are boxed.

$$
\left[\begin{array}{ccc|ccc} 0 & 2 & 4 & 1 & 0 & 0 \\ 3 & 1 & 2 & 0 & 1 & 0 \\ 6 & 2 & 1 & 0 & 0 & 1 \end{array}\right]
\xrightarrow[{[1]\leftrightarrow[2]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 4 & 1 & 0 & 0 \\ 6 & 2 & 1 & 0 & 0 & 1 \end{array}\right]
$$

$$
\xrightarrow[{[3]\leftarrow[3]-2[1]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 1 & 2 & 0 & 1 & 0 \\ 0 & \boxed{2} & 4 & 1 & 0 & 0 \\ 0 & 0 & \boxed{-3} & 0 & -2 & 1 \end{array}\right] \text{ row echelon form}
$$

$$
\xrightarrow[\substack{[2]\leftarrow[2]+\frac{4}{3}[3] \\ [1]\leftarrow[1]+\frac{2}{3}[3]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 1 & 0 & 0 & -\frac{1}{3} & \frac{2}{3} \\ 0 & \boxed{2} & 0 & 1 & -\frac{8}{3} & \frac{4}{3} \\ 0 & 0 & \boxed{-3} & 0 & -2 & 1 \end{array}\right]
$$

$$
\xrightarrow[{[1]\leftarrow[1]-\frac{1}{2}[2]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 0 & 0 & -\frac{1}{2} & 1 & 0 \\ 0 & \boxed{2} & 0 & 1 & -\frac{8}{3} & \frac{4}{3} \\ 0 & 0 & \boxed{-3} & 0 & -2 & 1 \end{array}\right]
$$

$$
\xrightarrow[\substack{[1]\leftarrow\frac{1}{3}[1] \\ [2]\leftarrow\frac{1}{2}[2] \\ [3]\leftarrow-\frac{1}{3}[3]}]{}
\left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -\frac{1}{6} & \frac{1}{3} & 0 \\ 0 & 1 & 0 & \frac{1}{2} & -\frac{4}{3} & \frac{2}{3} \\ 0 & 0 & 1 & 0 & \frac{2}{3} & -\frac{1}{3} \end{array}\right]
$$

To be able to reduce an $n \times n$ matrix into the identity matrix using Gaussian Elimination, we need n pivots. If we find fewer than n pivots, then the matrix does not have an inverse, i.e., is singular.

Example 8.16 The matrix $\begin{bmatrix} 0 & 2 & 4 \\ 3 & 1 & 2 \\ 3 & 3 & 6 \end{bmatrix}$ has no inverse. If we try to reduce this matrix to an identity matrix using Gaussian Elimination, we get:

$$
\left[\begin{array}{ccc|ccc} 0 & 2 & 4 & 1 & 0 & 0 \\ 3 & 1 & 2 & 0 & 1 & 0 \\ 3 & 3 & 6 & 0 & 0 & 1 \end{array}\right]
\xrightarrow[{[1]\leftrightarrow[2]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 1 & 2 & 0 & 1 & 0 \\ 0 & 2 & 4 & 1 & 0 & 0 \\ 3 & 3 & 6 & 0 & 0 & 1 \end{array}\right]
$$

$$
\xrightarrow[{[3]\leftarrow[3]-[1]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 1 & 2 & 0 & 1 & 0 \\ 0 & \boxed{2} & 4 & 1 & 0 & 0 \\ 0 & 2 & 4 & 0 & -1 & 1 \end{array}\right]
$$

$$
\xrightarrow[{[3]\leftarrow[3]-[2]}]{}
\left[\begin{array}{ccc|ccc} \boxed{3} & 1 & 2 & 0 & 1 & 0 \\ 0 & \boxed{2} & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & -1 & 1 \end{array}\right] \text{ row echelon form}
$$

We cannot proceed any further.

8.5.4.2 *Gaussian Elimination and the Determinant* The row echelon form of a matrix was obtained using elementary row operations. We know adding a multiple of one row to another does not change the determinant of the matrix. Every row swap changes the sign of the determinant, and multiplying one row by a constant multiplies the determinant by the same constant. Finally, the determinant of a triangular matrix (and the row echelon form of a matrix is triangular) is just the product of its diagonal elements. Therefore, we can compute the determinant of a square matrix as the product of the diagonal of its row echelon form, multiplied by -1 for every row swap made in obtaining the row echelon form. If a row was multiplied by a certain value at any stage prior to obtaining the row echelon form, that value must also be *divided* out.

For example, the product of the diagonal elements of the row echelon form of A in Example 8.15 is $3 \times 2 \times -3 = -18$. Since one row switch was made, we multiply this value by -1. Since no row was multiplied by a factor in obtaining the row echelon form, we do not need to divide anything out. The determinant of A is therefore $\det(A) = -18 \times -1 = 18$.

A square matrix has no inverse if (and only if) its row echelon form includes some zeros in the diagonal, i.e., if there aren't enough pivots, as illustrated in Example 8.16. It follows that the determinant of such a matrix is zero.

8.5.5 *Determinants and Inverses in Python*

In Python, we can use `la.det` and `la.inv` to calculate the determinant and inverse of a matrix. Both use the Gaussian elimation approach (via the LU decomposition that we will discuss in Section 10.2.1) to make their calculations, rather than the Laplace expansion for the determinant, or the formula presented in Appendix A for the inverse.

We use `la.det` and `la.inv` to calculate the determinant and inverse of the matrix

$$A = \begin{bmatrix} 0 & 2 & 4 \\ 3 & 1 & 2 \\ 5 & 2 & 1 \end{bmatrix}.$$

```
A = np.array([0,2,4,3,1,2,6,2,1]).reshape((3,3))
print(f"The determinant of A is: {la.det(A)}\n")
print(f"The inverse of A is: \n{la.inv(A)}")
```

```
The determinant of A is: 18.0

The inverse of A is:
[[-0.16666667  0.33333333  0.        ]
 [ 0.5        -1.33333333  0.66666667]
 [ 0.          0.66666667 -0.33333333]]
```

One can also use `sympy` to compute inverses symbolically:

```
A = sp.Matrix([[0,2,4], [3,1,2], [6,2,1]])
print(f"The inverse of A is: \n{A.inv()}")
```

```
The inverse of A is:
Matrix([[-1/6, 1/3, 0], [1/2, -4/3, 2/3], [0, 2/3, -1/3]])
```

You can use the following code to convert the `sympy` inverse matrix back to an `np.array`.

```
A_inverse_np = np.array(A.inv().tolist(), dtype=float)
print(f"The inverse of A is: \n{A_inverse_np}")
```

```
The inverse of A is:
[[-0.16666667  0.33333333  0.        ]
 [ 0.5        -1.33333333  0.66666667]
 [ 0.          0.66666667 -0.33333333]]
```

Suppose we wished to solve the following system of equations

$$\begin{aligned} 2x_2 + 4x_3 &= 4 \\ 3x_1 + \ x_2 + 2x_3 &= 2 \\ 6x_1 + 2x_2 + \ x_3 &= 7 \end{aligned}$$

In matrix form, this is

$$\underbrace{\begin{bmatrix} 0 & 2 & 4 \\ 3 & 1 & 2 \\ 6 & 2 & 1 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 4 \\ 2 \\ 7 \end{bmatrix}}_{b}.$$

Although we can compute the solution to this system using `la.inv(A) @ b`, it is recommended to use `la.solve(A,b)` instead, as the latter is more efficient and numerically stable. As with `la.det(A)` and `la.inv(A)`, `la.inv(A)` uses Gaussian elimination, via the LU decomposition.

```
# Create the matrix A and vector b
A = np.array([0, 2, 4, 3, 1, 2, 6, 2, 1]).reshape((3, 3))
b = np.array([4, 2, 7]).reshape((3, 1))

x = la.solve(A, b)  # Solve the system Ax = b numerically

print(f"SciPy's numerical solution to Ax = b is:\n\n{x}")
```

```
SciPy's numerical solution to Ax = b is:
```

```
[[ 0.]
 [ 4.]
 [-1.]]
```

You may occasionally get very small differences between the numerical solution obtained using `la.solve(A, b)` and the exact analytical solution due to numerical precision. Computers represent real numbers with a finite number of bits, which can lead to small rounding errors during numerical computations.

For a symbolic solution, without floating-point approximations, we can use SymPy:

```
# Create the matrix A and vector b symbolically
A = sp.Matrix([[0, 2, 4], [3, 1, 2], [5, 2, 1]])
b = sp.Matrix([4, 2, 7])

x = A.solve(b)  # Solve the system Ax = b symbolically

print(f"SymPy's symbolic solution to Ax = b is:\n{x}")
```

```
SymPy's symbolic solution to Ax = b is:
Matrix([[0], [4], [-1]])
```

In this code, `sp.Matrix()` creates symbolic matrices. When given a simple list of elements, SymPy interprets it as a column vector. The method `A.solve(b)` returns the exact symbolic solution.

Since SymPy's approach is symbolic, you can also use variables. Suppose we change the coefficient on x_2 in the third equation from 2 to a_{32}. The solution becomes:

```
# Create the matrix A and vector b symbolically
a32 = sp.Symbol('a32')
A = sp.Matrix([[0, 2, 4], [3, 1, 2], [6, a32, 1]])
b = sp.Matrix([4, 2, 7])

x = A.solve(b)  # Solve the system Ax = b symbolically

print(f"SymPy's symbolic solution to Ax = b is:\n{x}")
```

```
SymPy's symbolic solution to Ax = b is:
Matrix([[0], [-144/(12 - 24*a32)], [(2*a32 - 7)/(2*a32 - 1)]])
```

8.5.6 *Exercises*

Ex. 8.30 Find the inverse of the transpose of the matrix $A = \begin{bmatrix} 0 & 2 & 4 \\ 3 & 1 & 2 \\ 6 & 2 & 1 \end{bmatrix}$. (Hint: see Example 8.13.)

Ex. 8.31 Show that the inverse of a diagonal matrix $A = \text{diag}(a_{11}, a_{22}, \ldots, a_{nn})$ is the diagonal matrix

$$A^{-1} = \text{diag}\left(\frac{1}{a_{11}}, \frac{1}{a_{22}}, \ldots, \frac{1}{a_{nn}}\right).$$

Ex. 8.32 Suppose one row of a (square) matrix is a multiple of another row. Explain why this matrix has no inverse.

Ex. 8.33 Consider the following system of equations

$$\begin{array}{rcccccl} 4x_1 & + & & + & x_3 & = & 4 \\ 8x_1 & + & x_2 & + & -3x_3 & = & 3 \\ 12x_1 & + & x_2 & + & & = & 1 \end{array}$$

a. Express this system in the form $Ax = b$ and solve it by finding the inverse of A and then computing $A^{-1}b$.

b. Verify your solution in a. by solving the system using Cramer's Rule.

Ex. 8.34 Suppose A is an $m \times m$ matrix and b and c are $m \times 1$ vectors. Does $Ab = Ac$ imply that $b = c$? If no, give a counterexample.

Ex. 8.35 Use SymPy to solve Ex. 8.33(a).

8.6 Matrix Definiteness

An $n \times n$ symmetric matrix A is said to be **positive definite** if

$$x^{\mathrm{T}} A x > 0 \quad \text{for all } \; n\text{-vectors} \quad x \neq 0_n \,. \tag{8.9}$$

If the inequality in (8.9) is non-strict, then A is **positive semidefinite**. If the inequality in (8.9) is reversed, A is **negative definite**. If it is reversed and made non-strict, then A is called **negative semidefinite**. We emphasize that the conditions must hold for *all* non-zero vectors x.[8]

Example 8.17 The matrix $\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$ is positive definite since

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 2(x_1^2 + x_1 x_2 + x_2^2) = 2[(x_1 + 0.5x_2)^2 + 0.75x_2^2] > 0$$

as long as x_1 and x_2 are not both zero.

Example 8.18 The matrix $\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$ is indefinite (not definite) since

$$Q = \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x_1^2 + 4x_1 x_2 + x_2^2 \,.$$

If $x_1 = 1$ and $x_2 = 1$, then $Q > 0$. If $x_1 = 1$ and $x_2 = -1$, then $Q < 0$.

[8] Expressions of the form $x^{\mathrm{T}} A x$ where x is $n \times 1$ and A is $n \times n$ and symmetric are called quadratic forms.

We will see later that "variance-covariance matrices" are always at least positive semidefiniteness, often positive definite (Section 8.8). The positive or negative definiteness of the "Hessian" of a multivariable function is also an indicator of whether a function is convex or concave, which in turn plays an important role in function optimization (Chapter 11). Definiteness of matrices also play an important role in matrix factorizations, dynamic systems, and many other areas where matrix algebra is used.

One method for checking the definiteness of matrices uses the determinants of certain submatrices of the matrix. Given a square matrix A, an **order r principal minor**, $1 \leq r \leq n$ is the determinant of the $r \times r$ matrix obtained after removing $n-r$ rows and the corresponding $n-r$ columns of A. The **order r leading principal minor** is the determinant of the $r \times r$ matrix obtained after removing the *last* $n-r$ rows and columns of A.

Example 8.19 Consider the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

The principal minors are as follows. The first principal minor listed for each order is the leading principal minor.

$$\text{Order 1}: \ \det(a_{11}) = a_{11}\,, \ \det(a_{22}) = a_{22}\,, \ \det(a_{33}) = a_{33}\,;$$

$$\text{Order 2}: \ \det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}, \ \det \begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix}, \ \det \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix};$$

$$\text{Order 3}: \ \det \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

We will use $LPM(A, r)$ to denote the order r leading principal minor of A, and $PM(A, r)$ to denote any one of the order r principal minors of A (in otherwords, the latter only indicates an order r principal minor, but does not indicate which one). We have the following characterization of definiteness:

Theorem 8.1 *For any $n \times n$ symmetric matrix A, we have*

(a) A is positive definite iff $LPM(A, r) > 0$ for all $r = 1, \dots, n$;

(b) A is positive semidefinite iff $PM(A, r) \geq 0$ for all $r = 1, \dots, n$;

(c) A is negative definite iff $(-1)^r LPM(A, r) > 0$ for all $r = 1, \dots, n$;

(d) A is negative semidefinite iff $(-1)^r PM(A, r) \geq 0$ for all $r = 1, \dots, n$;

The proof of Theorem 8.1 is slightly long, and omitted. A nice presentation of the proof, with steps broken down into Exercise-Solution format can be found in Abadir and Magnus (2005). We will make do with a detailed look at the 2×2 case, after the following remarks:

i. The conditions for definiteness involve only leading principal minors, while the conditions for semidefiniteness involve all principal minors.
ii. Results (c) and (d) follow immediately from (a) and (b), and the fact that A is negative (semi)definite iff $-A$ is positive (semi)definite. The conditions in (c) and (d) are then simply the conditions in (a) and (b) applied to $-A$. Recall that if a row of a square matrix is multiplied by α, then its determinant is multiplied by α. If the *entire* matrix is multiplied by -1, then its determinant is multiplied by $(-1)^r$, where r is the number of rows or columns of the matrix.

We now show Theorem 8.1 for a 2×2 symmetric matrix $A = (a_{ij})_{2\times2}$. Expanding the quadratic form for A, we get

$$\begin{aligned}
Q = x^{\mathrm{T}}Ax &= \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \\
&= a_{11}x_1^2 + 2a_{12}x_1x_2 + a_{22}x_2^2 \\
&= a_{11}\left(x_1^2 + \frac{2a_{12}}{a_{11}}x_1x_2 + \frac{a_{22}}{a_{11}}x_2^2\right) \quad \text{if} \quad a_{11} \neq 0 \\
&= a_{11}\left[\left(x_1 + \frac{a_{12}}{a_{11}}x_2\right)^2 + \left(\frac{a_{11}a_{22} - a_{12}^2}{a_{11}^2}\right)x_2^2\right] \quad \text{if} \quad a_{11} \neq 0\,.
\end{aligned}$$

The third and fourth lines assume $a_{11} \neq 0$. If $a_{11} = 0$, we can factor out a_{22} instead. We have the following results, corresponding to results (b) and (d) of the theorem:

- $a_{11} \geq 0\,,\ a_{22} \geq 0$ and $a_{11}a_{22} - a_{12}^2 \geq 0 \Leftrightarrow A$ is positive semidefinite.

($\Rightarrow$) should be obvious. To show ($\Leftarrow$), suppose $Q \geq 0$ for all $x \neq 0_2$. Setting $x_1 = 1$ and $x_2 = 0$ shows that $a_{11} = Q \geq 0$. Setting $x_1 = 0$ and $x_2 = 1$ shows that $a_{22} = Q \geq 0$. Setting x_1 and x_2 such that $x_1/x_2 = -a_{12}/a_{11}$, we have $Q = [(a_{11}a_{22} - a_{12}^2)/a_{11}]x_2^2$, so $Q \geq 0$ implies $a_{11}a_{22} - a_{12}^2 \geq 0$.

- $a_{11} \leq 0\,,\ a_{22} \leq 0$ and $a_{11}a_{22} - a_{12}^2 \geq 0 \Leftrightarrow A$ is negative semidefinite.

The argument for this is similar to the condition for positive semidefiniteness.

We also have the following, corresponding to (a) and (c) of the theorem:

- $a_{11} > 0$ and $a_{11}a_{22} - a_{12}^2 > 0 \Leftrightarrow A$ is positive definite.

We do not need to include $a_{22} > 0$ as a condition since this is implied by the two given conditions. The implication ($\Rightarrow$) should be obvious. To see ($\Leftarrow$), note that if $Q > 0$ for all $x \neq 0_2$, then choosing $(x_1, x_2) = (1, 0)$ or $(0, 1)$ implies $a_{11} > 0$ and $a_{22} > 0$. Furthermore, if $a_{11}a_{22} - a_{12}^2 \leq 0$, then choosing x_1 and x_2 such that $x_1/x_2 = -a_{12}/a_{11}$ results in $Q \leq 0$. So it must be that $a_{11}a_{22} - a_{12}^2 > 0$.

- $a_{11} < 0$ and $a_{11}a_{22} - a_{12}^2 > 0 \Leftrightarrow A$ is negative definite.

The argument for this is similar to the condition for positive definiteness.

Example 8.20 The matrix

$$A = \begin{bmatrix} -4 & 2 & 0 \\ 2 & -6 & 2 \\ 0 & 2 & -8 \end{bmatrix}.$$

is negative definite, since the signs of its leading principal minors follow the $(-, +, -)$ pattern:

```
import numpy as np
from scipy.linalg import det

A = np.array([[-4, 2, 0],
              [2, -6, 2],
              [0, 2, -8]])
leading_principal_minors = [
  det(A[:i, :i]) for i in range(1, A.shape[0]+1)
]
print("Leading Principal Minors:",
    [float(lpm) for lpm in leading_principal_minors])
```

```
Leading Principal Minors: [-4.0, 20.0, -144.0]
```

In Chapter 10, we discuss how to determine the definiteness of A using its "eigenvalues".

8.6.1 *Exercises*

Ex. 8.36 Suppose X is $n \times k$. Explain why the matrix $X^{\mathrm{T}}X$ is positive semidefinite. Explain why it is positive definite if $Xc \neq 0$ for all k-vectors $c \neq 0_k$. (The next section explains the significance of the condition $Xc \neq 0$ for all k-vectors $c \neq 0_n$.)

8.7 The Rank of a Matrix

We recollect a few ideas from Chapter 4. A point x in $\mathbb{R}^m$ can be thought of as a m-dimensional vector, or "m-vector". If $X = \{x_1, x_2, \ldots, x_n\}$ is a set of n m-vectors, and if at least one of these vectors can be written as a linear combination of the others, i.e., if

$$x_i = c_1x_1 + \cdots + c_{i-1}x_{i-1} + c_{i+1}x_{i+1} + \cdots + c_nx_n\,,$$

then we say that the vectors are linearly dependent. Another way of saying this is that we can find $c_1, c_2, \ldots, c_n$, not all equal to zero, such that

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n = 0\,.$$

If we cannot express any vector in X as a linear combination of the other vectors, then the vectors in X are linearly independent. In that case, the vectors in X will satisfy the condition

$$c_1x_1 + c_2x_2 + \cdots + c_nx_n = 0 \quad \Rightarrow \quad c_1 = c_2 = \cdots = c_n = 0\,.$$

A vector space or subspace is a set of vectors such that linear combinations of vectors in the space always result in a vector in the space.[9] Every vector space or subspace must contain the zero vector. The set of *all* linear combinations of the vectors in X is a vector subspace of $\mathbb{R}^m$. The dimension of this subspace cannot exceed $\min\{m, n\}$. Finally, recall that two vectors are orthogonal if their inner product is zero.

Consider an $m \times n$ matrix A, where possibly $m \neq n$. We can view the columns of A as a collection of n m-vectors:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \left[\begin{array}{c|c|c|c} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{array}\right].$$

Linear combinations of the column vectors of A can be written as Ax where x is some n-vector. If we consider the function

$$y = f(x) = Ax\,,\ x \in \mathbb{R}^n \tag{8.10}$$

mapping n-vectors into m-vectors, then the range of this function is the set of all linear combinations of the columns of A, spanning a vector subspace of $\mathbb{R}^m$ of dimension $r \leq \min\{m, n\}$. We call this subspace the **column space** of A and refer to r as the **column rank** of A.

Likewise, we can view the rows of A as a collection of m n-vectors, i.e.,

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & a_{22} & \cdots & a_{2n} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline a_{m1} & a_{m2} & \cdots & a_{mn} \end{array}\right].$$

Linear combinations of the m row vectors can be written as $y^T A$ or $A^{\mathrm{T}}y$ where y is an m-vector. The range of the function

$$x = g(y) = A^{\mathrm{T}}y\,,\ y \in \mathbb{R}^m \tag{8.11}$$

is the column space of A^{T}, which is also the **row space** of A, since the columns of A^{T} are the rows of A. The dimension of the row space is called the **row rank** of A.

It turns out that for any matrix A, the row and column ranks of A are the same. Suppose the column rank of A is r. This means we can find

[9] When we think in terms of spaces, we move from "matrix algebra" to "linear algebra".

r linearly independent columns in A. Gather these columns into a $m \times r$ matrix C. Since every column of A can be written as a linear combination of the r columns in C, we can write $A = CR$ where R is $r \times n$, each column containing the necessary weights to generate the corresponding columns of A as a linear combination of the vectors in C. However, the fact that $A = CR$ also means that every *row* of A is a linear combination of the rows of R, the necessary weights appearing in the corresponding rows of C. Since R has r rows, the row rank of A also cannot exceed r, i.e.,

$$\text{row rank}(A) \leq r = \text{column rank}(A)\,.$$

Applying a similar argument to A^{T} shows that the row rank of A^{T} must be less than or equal to the column rank of A^{T}. But since the rows of A^{T} are the columns of A, we have

$$\text{column rank}(A) \leq \text{row rank}(A)\,.$$

It follows that

$$\text{column rank}(A) = \text{row rank}(A). \tag{8.12}$$

We can therefore speak unambiguously of the “rank” of a matrix A, and simply write $\text{rank}(A)$, where $0 \leq \text{rank}(A) \leq \min\{m, n\}$. If $\text{rank}(A) = \min\{m, n\}$, then we say that A has **full rank**. If this coincides with the number of columns n, $r = n \leq m$, we can also say that the matrix has **full column rank**. If the rank coincides with the number of rows, $r = m \leq n$, we say that it has **full row rank**.

The most straightforward way to determine the rank of a matrix A is by counting the number of pivots in its row echelon form, which we denote by $REF(A)$. Every row in $REF(A)$ was formed by elementary row operations on the rows of A, which means that every row in $REF(A)$ is a linear combination of the rows of A. It must be, therefore, that $\text{rank}(REF(A)) \leq \text{rank}(A)$. But it is equally true that every row of A is a linear combination of the rows of $REF(A)$, so $\text{rank}(A) \leq \text{rank}(REF(A))$. It follows that

$$\text{rank}(REF(A)) = \text{rank}(A)\,.$$

Since the rank of a row echelon form matrix is just the number of pivots therein, the rank of A is the number of pivots in $REF(A)$.

A square $n \times n$ matrix has an inverse if (and only if) its row echelon form has n pivots, i.e., if A has full rank. The following are three further results regarding matrix rank:

i. For any matrices A and B such that AB exists, we have

$$\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}.$$

This result holds because the columns of AB are linear combinations of the columns of A, therefore $\text{rank}(AB) \leq \text{rank}(A)$. Likewise, the rows of AB are linear combinations of the rows of B, therefore $\text{rank}(AB) \leq \text{rank}(B)$. It follows that $\text{rank}(AB) \leq \min\{\text{rank}(A), \text{rank}(B)\}$.

ii. If A is a full rank $m \times m$ matrix and B is $m \times p$ of rank r, then $\text{rank}(AB) = r$.

To see this, let $E_1, E_2, \ldots, E_m$ represent the row operations that reduce A to the identity matrix. Applying these same operations to AB reduces it to B (since A is reduced to I). But elementary row operations do not change the rank of matrices, so the rank of AB is the same as the rank of B.

iii. For any matrix A, we have

$$\text{rank}(A^{\mathrm{T}}A) = \text{rank}(AA^{\mathrm{T}}) = \text{rank}(A)\,.$$

We will prove this result shortly.

8.7.1 *The Fundamental Theorem of Linear Algebra*

Let A be an $m \times n$ matrix and consider the function

$$y = f(x) = Ax\,,\ x \in \mathbb{R}^n \tag{8.13}$$

mapping n-vectors into m-vectors. The range of this function is the set of all linear combinations of the columns of A, i.e., the column space of A. The dimension of this subspace of $\mathbb{R}^m$ is equal to $\text{rank}(A) = r \leq \min\{m, n\}$. We will denote the column space of A by $C(A, r)$ indicating both the matrix and the dimension of the space. Likewise, the range of the function

$$x = g(y) = A^{\mathrm{T}}y\,,\ y \in \mathbb{R}^m \tag{8.14}$$

is $C(A^{\mathrm{T}}, r)$, the column space of A^{T}, or the row space of A. The dimension of the row space of A is also r, since $\text{rank}(A) = \text{rank}(A^{\mathrm{T}})$.

The problem of solving a system of linear equations can be described as finding the vector or vectors x such that

$$Ax = b \ \text{ for some } \ b \in \mathbb{R}^m\,.$$

If $b \notin C(A, r)$, then there are no solutions. If $b \in C(A, r)$, then there is at least one solution. There are infinitely many solutions if $b \in C(A, r)$ and A does not have full column rank, and exactly one solution if $b \in C(A, r)$ and A has full column rank, i.e., if $r = n \leq m$.

In general, the set of all solutions to the system $Ax = b$, i.e., the set $\{x \in \mathbb{R}^n \mid Ax = b\}$, will not be a vector subspace, which requires inclusion of the zero vector. In contrast, the set of all solutions to the homogeneous system of equations

$$Ax = 0_m$$

where 0_m is the zero m-vector, will be a vector subspace of $\mathbb{R}^n$: If x_1 and x_2 are solutions, i.e., if they satisfy $Ax_1 = 0$ and $Ax_2 = 0$, then $\alpha x_1 + \beta x_2$ is also a solution, since

$$A(\alpha x_1 + \beta x_2) = \alpha A x_1 + \beta A x_2 = 0.$$

We call the set of all solutions to $Ax = 0$ the **null space** of A. The dimension of the null space will be the same as the number of free parameters in the solutions, which is $n - r$, where r is the rank of A. We denote the null space of A by $N(A, n-r) \subset \mathbb{R}^n$.

Notice also that $Ax = 0_m$ means that every row of A is orthogonal to every x satisfying this equation. It follows that every vector x in $C(A^{\mathrm{T}}, r)$, the row space of A, is orthogonal to every vector in $N(A, n-r)$. We say that the subspaces $C(A^{\mathrm{T}}, r)$ and $N(A, n-r)$ are orthogonal subspaces. The zero vector 0_n is the only vector in both $C(A^{\mathrm{T}}, r)$ and $N(A, n-r)$. Since the dimensions of the two orthogonal subspaces $C(A^{\mathrm{T}}, r) \subset \mathbb{R}^n$ and $N(A, n-r) \subset \mathbb{R}^n$ add to n, it follows that every vector $x \in \mathbb{R}^n$ can be written as

$$x = x_r + x_n \quad \text{where} \quad x_r \in C(A^{\mathrm{T}}, r) \quad \text{and} \quad x_n \in N(A, n-r)\,.$$

We say that $C(A^{\mathrm{T}}, r)$ and $N(A, n-r)$ are **orthogonal complements**. Likewise, the set of all y such that

$$A^{\mathrm{T}} y = 0_n$$

is a dimension $m - r$ subspace of $\mathbb{R}^m$ called the null space of A^{T}, denoted $N(A^{\mathrm{T}}, m-r)$. It is orthogonal to $C(A, r)$. The zero vector 0_m is the only vector in both $C(A, r)$ and $N(A^{\mathrm{T}}, m-r)$. Every vector $y \in \mathbb{R}^m$ can be decomposed as

$$y = y_c + y_n \quad \text{where } y_c \in C(A, r) \quad \text{and} \quad y_n \in N(A^{\mathrm{T}}, m-r)\,.$$

i.e., $C(A, r)$ and $N(A^{\mathrm{T}}, m-r)$ are orthogonal complements.

The subspaces $C(A, r)$, $N(A^{\mathrm{T}}, m-r)$, $C(A^{\mathrm{T}}, r)$ and $N(A, n-r)$ are called the **four fundamental subspaces** of a matrix A. Their relationships, as discussed above, are summarized in the **Fundamental Theorem of Linear Algebra** (FTLA).[10]

Theorem 8.2 (Fundamental Theorem of Linear Algebra) *For any $m \times n$ matrix A of rank r, the spaces*

(i) $C(A, r)$ and $N(A^{\mathrm{T}}, m-r)$ are orthogonal complements in $\mathbb{R}^m$,

(ii) $C(A^{\mathrm{T}}, r)$ and $N(A, n-r)$ are orthogonal complements in $\mathbb{R}^n$.

[10] Not to be confused with the Fundamental Theorem of Algebra.

Note that if A has full column rank $r = n \leq m$, then $C(A^{\mathrm{T}}, r = n)$ and $N(A, n - r = 0)$. In particular, the dimension of the null space of A is zero; it comprises only the zero vector 0_n. This also means that $C(A^{\mathrm{T}}, r = n)$ is the entirety of $\mathbb{R}^n$. Likewise, if A has full row rank, $r = m \leq n$, then $N(A^{\mathrm{T}}, m - r = 0) = \{0_m\}$ and $C(A, r = m) = \mathbb{R}^m$. If A is a full rank $n \times n$ matrix, then both $C(A, r = n)$ and $C(A^{\mathrm{T}}, r = n)$ are the whole $\mathbb{R}^n$ space, and $N(A, n - r = 0) = N(A^{\mathrm{T}}, n - r = 0) = \{0_n\}$.

Example 8.21 The column space $C(A, 1)$ of the following rank 1 matrix

$$A = \begin{bmatrix} 1 & -2 \\ 2 & -4 \\ 3 & -6 \end{bmatrix}$$

comprises all vectors of the form

$$y = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} s & 2s & 3s \end{bmatrix}^{\mathrm{T}} \tag{8.15}$$

which is a 1-dimensional subspace of $\mathbb{R}^3$. The null space of A comprises all vectors $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^{\mathrm{T}}$ that satisfy $Ax = 0_3$. These are vectors of the form

$$x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} s & s/2 \end{bmatrix}^{\mathrm{T}}. \tag{8.16}$$

The row space of A is a 1-dimensional subspace $C(A^{\mathrm{T}}, 1)$ of $\mathbb{R}^2$ comprising vectors of the form

$$x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} r & -2r \end{bmatrix}^{\mathrm{T}}. \tag{8.17}$$

The null space of A^{T} is the 2-dimensional subspace $N(A^{\mathrm{T}}, 2) \subset \mathbb{R}^3$ comprising all vectors y that satisfy $A^{\mathrm{T}} y = 0$. These are vectors of the form

$$y = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^{\mathrm{T}} = \begin{bmatrix} t & u & -\frac{t}{3} - \frac{2u}{3} \end{bmatrix}^{\mathrm{T}}. \tag{8.18}$$

The vectors in $C(A, 1)$ are orthogonal to the vectors in $N(A^{\mathrm{T}}, 2)$, since for all $x \in N(A^{\mathrm{T}}, 2)$ and all $y \in C(A, 1)$, we have

$$x \cdot y = x^{\mathrm{T}} y = s\,t + 2s\,u + 3s\left(-\frac{t}{3} - \frac{2u}{3}\right) = 0.$$

Visually, $N(A^{\mathrm{T}}, 2)$ is a plane in $\mathbb{R}^3$, and $C(A, 1)$ is a line in $\mathbb{R}^3$ that is perpendicular to the plane, cutting through the origin. Any vector in $\begin{bmatrix} a & b & c \end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^3$ can be written as a sum of a vector of the form (8.15) and a vector of the form (8.18). You can easily find s, t and u such that

$$\begin{bmatrix} s \\ 2s \\ 3s \end{bmatrix} + \begin{bmatrix} t \\ u \\ -\frac{t}{3} - \frac{2u}{3} \end{bmatrix} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}. \tag{8.19}$$

Likewise, the vectors in $C(A^{\mathrm{T}}, 1)$ are orthogonal to the vectors in $N(A, 1)$. Every vector in $\mathbb{R}^2$ can be written as a sum of a vector of the form (8.16) and a vector of the form (8.17).

As a quick application of the FTLA, we show that for any $n \times k$ matrix A we have

$$\text{rank}(A^{\text{T}}A) = \text{rank}(A) = \text{rank}(AA^{\text{T}})\,.$$

The row spaces of A and $A^{\text{T}}A$ are both subspaces of $\mathbb{R}^k$. Furthermore, they have the same null space: if x satisfies $Ax = 0$, then x satisfies $A^{\text{T}}Ax = 0$; if $A^{\text{T}}Ax = 0$, then $x^{\text{T}}A^{\text{T}}Ax = (Ax)^{\text{T}}Ax = 0$, which says that $Ax = 0$ since $(Ax)^{\text{T}}Ax$ is the sum of the squared terms of the vector Ax. Since A and $A^{\text{T}}A$ have the same null space, their ranks are the same. Likewise, $\text{rank}(AA^{\text{T}}) = \text{rank}(A^{\text{T}})$. Since $\text{rank}(A) = \text{rank}(A^{\text{T}})$, the result follows.

If we consider only the vectors $x \in C(A^{\text{T}}, r)$, i.e., if we restrict the domain of $y = Ax$ to $C(A^{\text{T}}, r)$, then A defines a *one-to-one* relationship from $C(A^{\text{T}}, r)$ onto $C(A, r)$. Suppose $x_1, x_2 \in C(A^{\text{T}}, r)$. Obviously, $x_1 = x_2$ implies $Ax_1 = Ax_2$, which implies $A(x_1 - x_2) = 0_m$. This says that $x_1 - x_2$ is in the null space $N(A, n - r)$. But $x_1 - x_2$ must also be in $C(A^{\text{T}}, r)$, since both x_1 and x_2 are in $C(A^{\text{T}}, r)$. Since 0_n is the only vector in both $N(A, n-r)$ and $C(A^{\text{T}}, r)$, it must be that $x_1 - x_2 = 0_m$, or $x_1 = x_2$.

We can therefore characterize the mapping $y = Ax$ for any arbitrary vectors $x \in \mathbb{R}^n$ as

$$\begin{aligned} Ax &= A(x_r + x_n) \;\text{ for some }\; x_r \in C(A^{\text{T}}, r) \;\text{ and }\; x_n \in N(A, n-r) \\ &= Ax_r + Ax_n \\ &= Ax_r \;\text{ since }\; x_n \in N(A, n-r) \;\text{ implies }\; Ax_n = 0_m\,. \end{aligned}$$

8.7.2 *Exercises*

Ex. 8.37 Consider the following matrix

$$X = \begin{bmatrix} 1 & 0 & 1 & 55.12 \\ 1 & 1 & 0 & 12.02 \\ 1 & 1 & 0 & 3.54 \\ 1 & 0 & 1 & 10.89 \\ 1 & 0 & 1 & 20.8 \\ 1 & 0 & 1 & 19.12 \\ 1 & 1 & 0 & 19.23 \\ 1 & 1 & 0 & 9.62 \end{bmatrix}.$$

Show that $\text{rank}(X) = 3$. What is the rank of $X^{\text{T}}X$? Does $X^{\text{T}}X$ have an inverse?

Ex. 8.38 Use `np.linalg.matrix_rank()` to calculate the rank of the matrix `H` from Ex. 8.25.

8.8 Vectors and Matrices of Random Variables

Organizing large numbers of random variables using matrix algebra provides convenient formulas for manipulating their expectations, variances and covariances, and for expressing their joint pdf.

8.8.1 *Expectations and Variance-Covariance Matrices*

The expectation of a vector x of m random variables $x = \begin{bmatrix} X_1 & X_2 & ... & X_m \end{bmatrix}^{\mathrm{T}}$ is defined as the vector of their expectations, i.e.,

$$E(x) = \begin{bmatrix} E(X_1) & E(X_2) & ... & E(X_m) \end{bmatrix}^{\mathrm{T}}.$$

Likewise, if X is a matrix of random variables, then

$$X = \begin{bmatrix} X_{11} & X_{12} & ... & X_{1n} \\ X_{21} & X_{22} & ... & X_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ X_{m1} & X_{m2} & ... & X_{mn} \end{bmatrix} \Leftrightarrow E(X) = \begin{bmatrix} E(X_{11}) & E(X_{12}) & ... & E(X_{1n}) \\ E(X_{21}) & E(X_{22}) & ... & E(X_{2n}) \\ \vdots & \vdots & \ddots & \vdots \\ E(X_{m1}) & E(X_{m2}) & ... & E(X_{mn}) \end{bmatrix}.$$

With these definitions, we can define the **variance-covariance matrix** of a vector x of random variables. Let

$$\tilde{x} = x - E(x) = \begin{bmatrix} X_1 - E(X_1) \\ X_2 - E(X_2) \\ \vdots \\ X_m - E(X_m) \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \vdots \\ \tilde{X}_m \end{bmatrix}.$$

Then

$$\begin{aligned} E(\tilde{x}\tilde{x}^{\mathrm{T}}) &= E((x - E(x))(x - E(x))^{\mathrm{T}}) \\ &= E \begin{bmatrix} \tilde{X}_1^2 & \tilde{X}_1\tilde{X}_2 & ... & \tilde{X}_1\tilde{X}_m \\ \tilde{X}_2\tilde{X}_1 & \tilde{X}_2^2 & ... & \tilde{X}_2\tilde{X}_m \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{X}_m\tilde{X}_1 & \tilde{X}_m\tilde{X}_2 & ... & \tilde{X}_m\tilde{X}_m \end{bmatrix} \\ &= \begin{bmatrix} E(\tilde{X}_1^2) & E(\tilde{X}_1\tilde{X}_2) & ... & E(\tilde{X}_1\tilde{X}_m) \\ E(\tilde{X}_2\tilde{X}_1) & E(\tilde{X}_2^2) & ... & E(\tilde{X}_2\tilde{X}_m) \\ \vdots & \vdots & \ddots & \vdots \\ E(\tilde{X}_m\tilde{X}_1) & E(\tilde{X}_m\tilde{X}_2) & ... & E(\tilde{X}_m\tilde{X}_m) \end{bmatrix} \qquad (8.20) \\ &= \begin{bmatrix} Var(X_1) & Cov(X_1, X_2) & ... & Cov(X_1, X_m) \\ Cov(X_1, X_2) & Var(X_2) & ... & Cov(X_2, X_m) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_1, X_m) & Cov(X_2, X_m) & ... & Var(X_m) \end{bmatrix}. \end{aligned}$$

In other words, $E((x-E(x))(x-E(x))^{\mathrm{T}})$ is a symmetric matrix containing the variances of all of the variables in x, and their covariances. We denote the variance-covariance matrix of a vector of random variables x by $Var(x)$:

$$Var(x) = E((x - E(x))(x - E(x))^{\mathrm{T}}).$$

Example 8.22 Let X_1, X_2 and X_3 be random variables with

$$E(X_1) = 1, E(X_2) = 3, E(X_3) = 5,$$
$$Var(X_1) = 2,\ Var(X_2) = 3,\ Var(X_3) = 2, \text{ and}$$
$$Cov(X_1, X_2) = 1,\ Cov(X_1, X_3) = 0,\ Cov(X_2, X_3) = 2$$

and let x be the 3×1 vector $\begin{bmatrix} X_1 & X_2 & X_3 \end{bmatrix}^{\mathrm{T}}$. Then

$$E(X) = \begin{bmatrix} 1 \\ 3 \\ 5 \end{bmatrix} \quad \text{and} \quad Var(X) = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 3 & 2 \\ 0 & 2 & 2 \end{bmatrix}.$$

Recall that if X is a (univariate) random variable, then

- $E(aX + b) = aE(X) + b$,
- $Var(aX + b) = a^2\, Var(X)$, and
- $Var(X) = E(X^2) - E(X)^2$.

The following are the matrix analogues of these results. Suppose x is an $m \times 1$ vector of random variables, $A = (a_{ij})_{km}$ is a $k \times m$ matrix of constants, and b is a $k \times 1$ vector of constants. Then

(i) $E(Ax + b) = AE(x) + b$,

(ii) $Var(Ax + b) = A\, Var(x) A^{\mathrm{T}}$,

(iii) $Var(x) = E(xx^{\mathrm{T}}) - E(x)E(x)^{\mathrm{T}}$.

To show (i), we note that the ith element of the $k \times 1$ vector $Ax + b$ is $\sum_{j=1}^{m}(a_{ij}X_j + b_i)$, and the expectation of this term is

$$E\left(\sum_{j=1}^{m}(a_{ij}X_j + b_i)\right) = \sum_{j=1}^{m} a_{ij}E(X_j) + b_i\,,$$

which in turn is the ith element of the vector $AE(x) + b$. For (ii), since $Ax + b - E(Ax + b) = A(x - E(x)) = A\tilde{x}$, we have

$$\begin{aligned} Var(Ax + b) &= E((A\tilde{x})(A\tilde{x})^{\mathrm{T}}) = E(A\tilde{x}\tilde{x}^{\mathrm{T}}A^{\mathrm{T}}) = AE(\tilde{x}\tilde{x}^{\mathrm{T}})A^{\mathrm{T}} \\ &= A\ Var(x) A^{\mathrm{T}}\,. \end{aligned}$$

You are asked to prove (iii) in Ex. 8.39.

Example 8.23 Given a vector of random variables x, the linear combination $c^{\mathrm{T}}x$ of the random variables in x has variance-covariance matrix

$$Var(c^{\mathrm{T}}x) = c^{\mathrm{T}}\, Var(x) c\,.$$

Since variances cannot be negative, we have $c^{\mathrm{T}} Var(x)c \geq 0$ for all c, i.e., $Var(x)$ is a positive semidefinite matrix. If there is a linear combination of the random variables in x that has zero variance, then at least one or more of the variables in x is actually a constant (a "degenerate random variable"), or at least one of the variables in x is a linear combination of the others. Otherwise we have $c^{\mathrm{T}} Var(x)c > 0$ for all $c \neq 0$, i.e., $Var(x)$ is positive definite.

8.8.2 *The Multivariate Normal Distribution*

We presented the pdf of a bivariate normal distribution in Section 7.4.4. We present here the pdf of a general multivariate normal distribution and some associated results. A $k \times 1$ vector of random variables x is said to have a multivariate normal distribution with mean μ and positive definite variance-covariance matrix Σ, denoted $\text{Normal}_k(\mu, \Sigma)$, if its pdf has the form

$$f(x) = (2\pi)^{-\frac{k}{2}} \det(\Sigma)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right\}.$$

We list a few results below, omitting proofs:

(a) If Σ is diagonal, then $X_1, X_2, \ldots, X_k$ are independent random variables.

(b) If $x \sim \text{Normal}_k(\mu, \Sigma)$, then for $A_{m\times k}$ and $b_{m\times 1}$,

$$Ax + b \ \sim\ \text{Normal}_m(A\mu + b, A\Sigma A^{\mathrm{T}}).$$

(c) If we partition x as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \sim \text{Normal}_k\left(\begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right)$$

where x_1 is $k_1 \times 1$ and x_2 is $k_2 \times 1$, with $k_1 + k_2 = k$, then the marginal distribution of x_1 is $\text{Normal}_{k_1}(\mu_1, \Sigma_{11})$, and the conditional distribution of x_2 given x_1 is

$$x_2 \mid x_1 \sim \text{Normal}_{k_2}(\mu_{2|1}, \Sigma_{22|1})$$

where $\mu_{2|1} = \mu_2 + \Sigma_{21}\Sigma_{11}^{-1}(x_1 - \mu_1)$ and $\Sigma_{22|1} = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$.

(d) If $x \sim \text{Normal}_k(0, I)$ and A is a rank v symmetric matrix such that $AA = A$, then the scalar $x^{\mathrm{T}}Ax$ is distributed $\chi^2(v)$:

$$x^{\mathrm{T}}Ax \ \sim\ \chi^2(v).$$

Matrices A such that $AA = A$ are said to be **idempotent**.

(e) If $x \sim \text{Normal}_k(\mu, \Sigma)$, then $(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu) \sim \chi^2(k)$.

8.8.3 *Exercises*

Ex. 8.39 Show that $Var(x) = E(xx^{\mathrm{T}}) - E(x)E(x)^{\mathrm{T}}$.

Ex. 8.40 Show that $E(\mathrm{trace}(X)) = \mathrm{trace}(E(X))$ where $X = (X_{ij})_{n\times n}$ is a matrix of random variables.

8.9 Appendix: A Formula for the Inverse Matrix

Recall that the (i, j)th cofactor of a square matrix A is the determinant of the same matrix with the ith row and jth column removed, times $(-1)^{i+j}$. For example, the $(2, 3)$th cofactor of the 4×4 matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

is $$C_{23}(A) = (-1)^{2+3} \begin{vmatrix} a_{11} & a_{12} & \cdot & a_{14} \\ \cdot & \cdot & \cdot & \cdot \\ a_{31} & a_{32} & \cdot & a_{34} \\ a_{41} & a_{42} & \cdot & a_{44} \end{vmatrix} = - \begin{vmatrix} a_{11} & a_{12} & a_{14} \\ a_{31} & a_{32} & a_{34} \\ a_{41} & a_{42} & a_{44} \end{vmatrix}.$$

The determinant of A can be computed using the Laplace expansion over any row or column. For instance, we can take the Laplace expansion along the second column to get

$$\det(A) = a_{12}C_{12}(A) + a_{22}C_{22}(A) + a_{32}C_{32}(A) + a_{42}C_{42}(A)\,.$$

The matrix A and its **cofactor matrix** $C(A)$ is shown below, where we have boxed the second column of both.

$$A = \begin{bmatrix} a_{11} & \boxed{\begin{matrix} a_{12} \\ a_{22} \\ a_{32} \\ a_{42} \end{matrix}} & \begin{matrix} a_{13} \\ a_{23} \\ a_{33} \\ a_{43} \end{matrix} & \begin{matrix} a_{14} \\ a_{24} \\ a_{34} \\ a_{44} \end{matrix} \end{bmatrix}$$

$$C(A) = \begin{bmatrix} \begin{matrix} C_{11}(A) \\ C_{21}(A) \\ C_{31}(A) \\ C_{41}(A) \end{matrix} & \boxed{\begin{matrix} C_{12}(A) \\ C_{22}(A) \\ C_{32}(A) \\ C_{42}(A) \end{matrix}} & \begin{matrix} C_{13}(A) \\ C_{23}(A) \\ C_{33}(A) \\ C_{43}(A) \end{matrix} & \begin{matrix} C_{14}(A) \\ C_{24}(A) \\ C_{34}(A) \\ C_{44}(A) \end{matrix} \end{bmatrix}$$

What happens if we were to expand along the second column, but multiply each cofactor by the corresponding elements of a *different column*? We can see the effects of this by considering the determinant of a modified A matrix, with the second column replaced by the first, as shown below along with the corresponding modified cofactor matrix:

$$\widetilde{A} = \begin{bmatrix} a_{11} & \boxed{\begin{matrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{41} \end{matrix}} & a_{13} & a_{14} \\ a_{21} & & a_{23} & a_{24} \\ a_{31} & & a_{33} & a_{34} \\ a_{41} & & a_{43} & a_{44} \end{bmatrix}$$

$$C(\widetilde{A}) = \begin{bmatrix} C_{11}(\widetilde{A}) & \boxed{\begin{matrix} C_{12}(\widetilde{A}) \\ C_{22}(\widetilde{A}) \\ C_{32}(\widetilde{A}) \\ C_{42}(\widetilde{A}) \end{matrix}} & C_{13}(\widetilde{A}) & C_{14}(\widetilde{A}) \\ C_{21}(\widetilde{A}) & & C_{23}(\widetilde{A}) & C_{24}(\widetilde{A}) \\ C_{31}(\widetilde{A}) & & C_{33}(\widetilde{A}) & C_{34}(\widetilde{A}) \\ C_{41}(\widetilde{A}) & & C_{43}(\widetilde{A}) & C_{44}(\widetilde{A}) \end{bmatrix}$$

The determinant of $\widetilde{A}$ is zero, since there are two identical columns. The cofactors of $\widetilde{A}$ along the second columns are the same as the corresponding cofactors of A. This means that

$$\det(\widetilde{A}) = a_{11}C_{12}(A) + a_{21}C_{22}(A) + a_{31}C_{32}(A) + a_{41}C_{42}(A) = 0\,.$$

You can see that the same result will hold if we multiplied the cofactors in the second column with the corresponding elements of columns 3 or 4. In general, *the sum of the products of the cofactors in one column (row) of the cofactor matrix $C(A)$ with the corresponding elements of a different column (row) of the matrix A is zero.*

Now consider pre-multiplying A with the transpose of $C(A)$. We have

$$C(A)^{\mathrm{T}}A = \begin{bmatrix} C_{11}(A) & C_{21}(A) & C_{31}(A) & C_{41}(A) \\ C_{12}(A) & C_{22}(A) & C_{32}(A) & C_{42}(A) \\ C_{13}(A) & C_{23}(A) & C_{33}(A) & C_{43}(A) \\ C_{14}(A) & C_{24}(A) & C_{34}(A) & C_{44}(A) \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$$
$$= \begin{bmatrix} \det(A) & 0 & 0 & 0 \\ 0 & \det(A) & 0 & 0 \\ 0 & 0 & \det(A) & 0 \\ 0 & 0 & 0 & \det(A) \end{bmatrix}.$$

It follows that the inverse of A is the transpose of its cofactor matrix divided by the determinant

$$A^{-1} = \frac{1}{\det(A)} C(A)^{\mathrm{T}}\,.$$

Cramer's rule pops right out of this formula. Consider the n-equations n-unknowns system $Ax = b$ where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

If a unique solution exists, then

$$\begin{bmatrix} x_1 \\ \boxed{x_2} \\ \vdots \\ x_n \end{bmatrix} = A^{-1}b = \frac{1}{\det(A)}C(A)^{\mathrm{T}}b = \frac{1}{\det(A)}\begin{bmatrix} C_{11}(A) & C_{21}(A) & \dots & C_{n1}(A) \\ C_{12}(A) & C_{22}(A) & \dots & C_{n2}(A) \\ \vdots & \vdots & \ddots & \vdots \\ C_{1n}(A) & C_{2n}(A) & \dots & C_{nn}(A) \end{bmatrix}\begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

where we have indicated the solution for x_2. You can see that

$$x_2 = \frac{1}{\det(A)}\left[b_1C_{12}(A) + b_2C_{22}(A) + \dots + b_nC_{n2}(A)\right] = \frac{\det(A_2(b))}{\det(A)}$$

where $A_2(b)$ is the matrix A with the second column replaced by b. Similar remarks can be made for the other elements of the solution x.

8.10 Solutions to Exercises

Ex. 8.1: $\dim(A)$ is 3×2. $(A)_{1,2} = 13$. $(A)_{3,1} = 7$.

Ex. 8.2: $A = \begin{bmatrix} 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \end{bmatrix}$.

Ex. 8.3:
i. $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ ii. $\begin{bmatrix} * & 0 & 0 & 0 \\ 0 & * & 0 & 0 \\ 0 & 0 & * & 0 \\ 0 & 0 & 0 & * \end{bmatrix}$ iii. $\begin{bmatrix} * & * & * & * \\ 0 & * & * & * \\ 0 & 0 & * & * \\ 0 & 0 & 0 & * \end{bmatrix}$ iv. $\begin{bmatrix} * & 0 & 0 & 0 \\ * & * & 0 & 0 \\ * & * & * & 0 \\ * & * & * & * \end{bmatrix}$.

Ex. 8.4: Solving $u + 2v = 1$ and $u + v = 4$ gives $u = 7$ and $v = -3$.

Ex. 8.5: $A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$. Question does not define the $(i, i)th$ element; we've decided on $(A)_{ii} = 1$.

Ex. 8.6: Even though they are both zero matrices, A and B are not equal because their dimensions are not the same.

Ex. 8.7: $A = \frac{1}{2}\begin{bmatrix} 3 & 4 \\ 2 & 8 \\ 1 & 5 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} & 2 \\ 1 & 4 \\ \frac{1}{2} & \frac{5}{2} \end{bmatrix}$; $B = \begin{bmatrix} 6 & 4 \\ 2 & 5 \\ 3 & 1 \end{bmatrix} + \frac{1}{2}\begin{bmatrix} 3 & 4 \\ 1 & 8 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 7.5 & 6 \\ 2.5 & 9 \\ 3.5 & 3 \end{bmatrix}$.

Ex. 8.8: Only (a) and (c) are symmetric.

Ex. 8.9: (a) - (e) are all true. (f) is false: $(A^{\mathrm{T}})^{\mathrm{T}} = A$ holds for all matrices.

Ex. 8.10: (a) $A = \frac{2}{3}C - \frac{1}{3}D$ and $B = \frac{2}{3}D - \frac{1}{3}C$ where $C = \begin{bmatrix} 1 & 2 & 1 \\ 4 & 3 & 0 \end{bmatrix}$ and $D = \begin{bmatrix} 4 & 2 & 3 \\ 5 & 1 & 1 \end{bmatrix}$. (b) $A = \frac{2}{5}C$, $B = \frac{3}{5}C$.

Ex. 8.11: (a) $BC = \begin{bmatrix} 14 & 4 \\ 69 & 30 \end{bmatrix}$, $CB = \begin{bmatrix} 20 & 16 \\ 21 & 24 \end{bmatrix}$, $AB = \begin{bmatrix} 28 & 64 \\ 6 & 0 \\ 13 & 8 \end{bmatrix}$.

(b) BA cannot be computed, because number of columns in B does not equal number of rows in A.

Ex. 8.12: We have

$$x^{\mathrm{T}}x = \begin{bmatrix} x_1 & x_2 & \dots & x_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \sum_{i=1}^{n} x_i^2$$

which is non-negative because it is a sum of squares. $x^{\mathrm{T}}x = 0$ if and only if every element of x is zero.

Ex. 8.13: (a) $\begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix} = \begin{bmatrix} 2\cdot -2 + 4\cdot 1 & 2\cdot 4 + 4\cdot -2 \\ 1\cdot -2 + 2\cdot 1 & 1\cdot 4 + 2\cdot -2 \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

(b) $\begin{bmatrix} 1 & b \\ -\frac{1}{b} & -1 \end{bmatrix} \begin{bmatrix} 1 & b \\ -\frac{1}{b} & -1 \end{bmatrix} = \begin{bmatrix} 1\cdot 1 + b(-1/b) & 1\cdot b + b(-1) \\ (-\frac{1}{b})1 + (-1)(-\frac{1}{b}) & (-\frac{1}{b})b + (-1)(-1) \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$.

Ex. 8.14:

(a)
$$\begin{aligned} ((AB)C)_{ij} &= \sum_{k=1}^{p} (AB)_{ik}(C)_{kj} = \sum_{k=1}^{p} \left(\sum_{l=1}^{n} (A)_{il}(B)_{lk} \right) (C)_{kj} \\ &= \sum_{l=1}^{n} (A)_{il} \left(\sum_{k=1}^{p} (B)_{lk}(C)_{kj} \right) = \sum_{l=1}^{n} (A)_{il}(BC)_{lj} = (A(BC))_{ij} \end{aligned}$$

(b)
$$\begin{aligned} (A(B+C))_{ij} &= \sum_{k=1}^{n} (A)_{ik}(B+C)_{kj} = \sum_{k=1}^{n} (A)_{ik}((B)_{kj} + (C)_{kj}) \\ &= \sum_{k=1}^{n} (A)_{ik}(B)_{kj} + \sum_{k=1}^{n} (A)_{ik}(C)_{kj} = (AB + AC)_{ij} \end{aligned}$$

(c)
$$\begin{aligned} ((A+B)C))_{ij} &= \sum_{k=1}^{n} (A+B)_{ik}(C)_{kj} = \sum_{k=1}^{n} ((A)_{ik} + (B)_{ik})(C)_{kj} \\ &= \sum_{k=1}^{n} (A)_{ik}(C)_{kj} + \sum_{k=1}^{n} (B)_{ik}(C)_{kj} = (AC + BC)_{ij} \end{aligned}$$

Ex. 8.15 (a) $(I_m A)_{ij} = \sum_{k=1}^{m} (I_m)_{ik}(A)_{kj} = (I_m)_{ii}(A)_{ij} = (A)_{ij}$.
(b) $(AI_n)_{ij} = \sum_{k=1}^{n} (A)_{ik}(I_n)_{kj} = (A)_{ij}(I_n)_{jj} = (A)_{ij}$.

Ex. 8.16: Both RHS and LHS equal $\begin{bmatrix} a_{11}b_1 + a_{12}b_2 + a_{13}b_3 \\ a_{21}b_1 + a_{22}b_2 + a_{23}b_3 \\ a_{31}b_1 + a_{32}b_2 + a_{33}b_3 \\ a_{41}b_1 + a_{42}b_2 + a_{43}b_3 \end{bmatrix}$.

Ex. 8.17: (a) We want to show that the (i, j)th element of $(AB)^{\mathrm{T}}$ is equal to the (i, j)th element of $B^{\mathrm{T}}A^{\mathrm{T}}$. By definition of the transpose, the (i, j)th element of $(AB)^{\mathrm{T}}$ is the (j, i)th element of AB, therefore

$$\begin{aligned} ((AB)^{\mathrm{T}})_{ij} = (AB)_{ji} &= \sum_{k=1}^{n} a_{jk}b_{ki} \\ &= \sum_{k=1}^{n} b_{ki}a_{jk} = \sum_{k=1}^{n} (B^{\mathrm{T}})_{ik}(A^{\mathrm{T}})_{kj} = (B^{\mathrm{T}}A^{\mathrm{T}})_{ij}. \end{aligned}$$

For (b), we have $(ABC)^{\mathrm{T}} = ((AB)C)^{\mathrm{T}} = C^{\mathrm{T}}(AB)^{\mathrm{T}} = C^{\mathrm{T}}B^{\mathrm{T}}A^{\mathrm{T}}$.

Ex. 8.18: Since X is $n \times k$, $X^{\mathrm{T}}X$ is $k \times k$. $X^{\mathrm{T}}X$ is symmetric since $(X^{\mathrm{T}}X)^{\mathrm{T}} = X^{\mathrm{T}}(X^{\mathrm{T}})^{\mathrm{T}} = X^{\mathrm{T}}X$.

Ex. 8.19: (a) $\text{tr}(A+B) = \sum_{i=1}^{n}(A+B)_{ii} = \sum_{i=1}^{n}(A)_{ii} + (B)_{ii} = \text{tr}(A) + \text{tr}(B)$.
(b) $(A)_{ii} = (A^{\text{T}})_{ii}$, so the trace of A and $A^{\text{T}}$are the same.
(c) $\text{tr}(AB) = \sum_{i=1}^{n}(AB)_{ii} = \sum_{k=1}^{n}\sum_{i=1}^{n}(B)_{ki}(A)_{ik} = \sum_{k=1}^{n}(B)_{ki}(A)_{ik} = \text{tr}(BA)$.
(d.i) Multiplying out xx^{T} will show that its diagonal elements are x_i^2, $i = 1, 2, ..., n$, so the trace of xx^{T} is $\sum_{i=1}^{n} x_i^2 = x^{\text{T}}x$. For (d.ii), using results from part (c), we have $\text{tr}(x^{\text{T}}x) = \text{tr}(xx^{\text{T}})$, but since $x^{\text{T}}x$ is a scalar, we have $\text{tr}(x^{\text{T}}x) = x^{\text{T}}x$.

Ex. 8.20: (a) Since $(i_n^{\text{T}}i_n)^{-1} = 1/n$ and $i_n^{\text{T}}y = \sum_{i=1}^{n} y_i$, we have $(i_n^{\text{T}}i_n)^{-1}i_n^{\text{T}}y = (1/n)\sum_{i=1}^{n} y_i = \overline{y}$.
(b) We first note that $M_0 = I_n - i_n(i_n^{\text{T}}i_n)^{-1}i_n^{\text{T}} = I_n - \frac{1}{n}i_n i_n^{\text{T}}$. We have,

$$\text{Symmetry: } M_0^{\text{T}} = \left(I_n - \frac{1}{n}i_n i_n^{\text{T}}\right)^{\text{T}} = I_n^{\text{T}} - \frac{1}{n}(i_n^{\text{T}})^{\text{T}}i_n^{\text{T}} = I_n^{\text{T}} - \frac{1}{n}i_n i_n^{\text{T}} = M_0\,.$$

$$\begin{aligned}\text{Idempotence: } M_0M_0 &= \left(I_n - \frac{1}{n}i_n i_n^{\text{T}}\right)\left(I_n - \frac{1}{n}i_n i_n^{\text{T}}\right)\\ &= I_n - \frac{1}{n}i_n i_n^{\text{T}} - \frac{1}{n}i_n i_n^{\text{T}} + \frac{1}{n^2}i_n i_n^{\text{T}} i_n i_n^{\text{T}}\\ &= I_n - \frac{2}{n}i_n i_n^{\text{T}} + \frac{1}{n^2}i_n(n)i_n^{\text{T}} = I_n - \frac{1}{n}i_n i_n^{\text{T}} = M_0\,.\end{aligned}$$

(c) First, note that $M_0y = (I_n - i_n(i_n^{\text{T}}i_n)^{-1}i_n^{\text{T}})\,y = y - i_n\overline{y} = \begin{bmatrix} y_1 - \overline{y} \\ y_2 - \overline{y} \\ \vdots \\ y_n - \overline{y}\end{bmatrix}$. Therefore

$$\sum_{i=1}^{n}(y_i - \overline{y})^2 = (M_0y)^{\text{T}}M_0y = y^{\text{T}}M_0^{\text{T}}M_0y = y^{\text{T}}M_0M_0y = y^{\text{T}}M_0y\,.$$

Ex. 8.21: The result follows from the following equalities:

$$\sum_{i=1}^{n}(A)_{ik}(\alpha B)_{kj} = \sum_{i=1}^{n}(A)_{ik}\alpha(B)_{kj} = \sum_{i=1}^{n}(\alpha A)_{ik}(B)_{kj} = \alpha\sum_{i=1}^{n}(A)_{ik}(B)_{kj}\,.$$

Ex. 8.22: Multiplying out fully, we have $AB = \begin{bmatrix} 37 & 19 & 24 \\ 34 & 19 & 35 \\ 27 & 15 & 18 \\ 27 & 10 & 17 \\ 38 & 13 & 17 \end{bmatrix}$.

Writing A and B as

$$A = \left[\begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array}\right] = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \quad \text{and} \quad B = \left[\begin{array}{c|cc} 2 & 0 & 1 \\ \hline 3 & 1 & 3 \\ 1 & 5 & 4 \\ 4 & 1 & 1 \end{array}\right] = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

it is straightforward to verify that

$$A_{11}B_{11} + A_{12}B_{21} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} + \begin{bmatrix} 35 \\ 30 \end{bmatrix} = \begin{bmatrix} 37 \\ 34 \end{bmatrix}$$

and

$$A_{11}B_{12} + A_{12}B_{22} = \begin{bmatrix} 0 & 1 \\ 0 & 2 \end{bmatrix} + \begin{bmatrix} 19 & 23 \\ 19 & 33 \end{bmatrix} = \begin{bmatrix} 19 & 24 \\ 19 & 35 \end{bmatrix}$$

and likewise for $A_{21}B_{11} + A_{22}B_{21}$ and $A_{21}B_{12} + A_{22}B_{22}$. Furthermore

$$A^{\mathrm{T}} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^{\mathrm{T}} = \left[\begin{array}{c|ccc} 1 & 3 & 2 & 6 \\ 2 & 8 & 2 & 1 \\ \hline 3 & 1 & 2 & 4 \\ 4 & 2 & 1 & 3 \\ 3 & 1 & 1 & 7 \end{array}\right]^{\mathrm{T}} = \left[\begin{array}{cc|ccc} 1 & 2 & 3 & 4 & 3 \\ \hline 3 & 8 & 1 & 2 & 1 \\ 2 & 2 & 2 & 1 & 1 \\ 6 & 1 & 4 & 3 & 7 \end{array}\right] = \begin{bmatrix} A_{11}^{\mathrm{T}} & A_{21}^{\mathrm{T}} \\ A_{12}^{\mathrm{T}} & A_{22}^{\mathrm{T}} \end{bmatrix}.$$

Ex. 8.23: Write A as

$$A = \left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ \hline a_{21} & a_{22} & \cdots & a_{2n} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline a_{m1} & a_{m2} & \cdots & a_{mn} \end{array}\right] = \begin{bmatrix} a_{1*} \\ a_{2*} \\ \vdots \\ a_{m*} \end{bmatrix} \quad \text{where} \quad a_{i*} = \begin{bmatrix} a_{i1} & a_{i2} & \dots & a_{in} \end{bmatrix}.$$

Then $c^{\mathrm{T}}A = \begin{bmatrix} c_1 & c_2 & \dots & c_m \end{bmatrix} \begin{bmatrix} a_{1*} \\ a_{2*} \\ \vdots \\ a_{m*} \end{bmatrix} = c_1 a_{1*} + c_2 a_{2*} + \cdots + c_m a_{m*}$.

Ex. 8.24: The first equality can be obtained by multiplying

$$X^{\mathrm{T}}X = \begin{bmatrix} X_{*1}^{\mathrm{T}} \\ X_{*2}^{\mathrm{T}} \\ X_{*3}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X_{*1} & X_{*2} & X_{*3} \end{bmatrix} = \begin{bmatrix} X_{*1}^{\mathrm{T}}X_{*1} & X_{*1}^{\mathrm{T}}X_{*2} & X_{*1}^{\mathrm{T}}X_{*3} \\ X_{*2}^{\mathrm{T}}X_{*1} & X_{*2}^{\mathrm{T}}X_{*2} & X_{*2}^{\mathrm{T}}X_{*3} \\ X_{*3}^{\mathrm{T}}X_{*1} & X_{*3}^{\mathrm{T}}X_{*2} & X_{*3}^{\mathrm{T}}X_{*3} \end{bmatrix}.$$

The second equality comes from the partition by observations:

$$X^{\mathrm{T}}X = \begin{bmatrix} X_{1*}^{\mathrm{T}} & X_{2*}^{\mathrm{T}} & X_{3*}^{\mathrm{T}} & \dots & X_{n*}^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} X_{1*} \\ X_{2*} \\ X_{3*} \\ \vdots \\ x_{n*} \end{bmatrix} = \sum_{i=1}^{n} X_{i*}^{\mathrm{T}} X_{i*}.$$

The last expression comes directly from the second expression.

Ex. 8.25:

```
H = np.array([[1, 1/2, 1/3], [1/2, 1/3, 1/4], [1/3, 1/4, 1/5]])
print(H)
```

```
[[1.         0.5        0.33333333]
 [0.5        0.33333333 0.25      ]
 [0.33333333 0.25       0.2       ]]
```

Ex. 8.26:

```
n = 3
elements = [[1 / (i + j - 1) for j in range(1, n + 1)] for i in range(1, n + 1)]
H = np.array(elements)
```

This approach is generally better than hardcoding the elements of `H` because it scales well. For example, try changing the value of `n` to `6` and display the matrix.

Ex. 8.27: (a) and (b)

```
B = np.ones((2, 2), dtype=int)
D1 = np.zeros((6, 6), dtype=int)

for i in range(0, 6, 2):  # start, stop, step
    D1[i:i+2, i:i+2] = B
print(D1)

D2 = np.kron(np.eye(3), B)
print(D2)
```

```
[[1 1 0 0 0 0]
 [1 1 0 0 0 0]
 [0 0 1 1 0 0]
 [0 0 1 1 0 0]
 [0 0 0 0 1 1]
 [0 0 0 0 1 1]]
[[1. 1. 0. 0. 0. 0.]
 [1. 1. 0. 0. 0. 0.]
 [0. 0. 1. 1. 0. 0.]
 [0. 0. 1. 1. 0. 0.]
 [0. 0. 0. 0. 1. 1.]
 [0. 0. 0. 0. 1. 1.]]
```

Ex. 8.28:

```
print(f"The first two rows and columns of H are:\n{H[:2, :2]}")
```

```
The first two rows and columns of H are:
[[1.         0.5       ]
 [0.5        0.33333333]]
```

Using `H[:2, :2]` is generally preferred over `H[0:2, 0:2]` for array slicing in Python, as the former is more concise and still clear.

Ex. 8.29:

```
H_trace = np.trace(H)
print(f"The trace of the matrix H is: {H_trace}")
```

```
The trace of the matrix H is: 1.5333333333333332
```

Ex. 8.30: In Example 8.13 we showed that the inverse of A is $A^{-1} = \begin{bmatrix} -\frac{1}{6} & \frac{1}{3} & 0 \\ \frac{1}{2} & -\frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & -\frac{1}{3} \end{bmatrix}$. Therefore the inverse of A^{T} is

$$(A^{\mathrm{T}})^{-1} = (A^{-1})^{\mathrm{T}} = \begin{bmatrix} -\frac{1}{6} & \frac{1}{2} & 0 \\ \frac{1}{3} & -\frac{4}{3} & \frac{2}{3} \\ 0 & \frac{2}{3} & -\frac{1}{3} \end{bmatrix}.$$

Ex. 8.31: This is easily seen by direct multiplication.

Ex. 8.32: Adding a multiple of one row to another row does not change the determinant of a matrix. If row i of a (square) matrix is a multiple of row j, then subtracting that multiple of row j from row i reduces row i into a zero row without changing the determinant. The determinant of any matrix with a zero row is zero. Matrices with zero determinant do not have an inverse.

Ex. 8.33: (a) The system can be written as $Ax = b$ as shown below

$$\begin{bmatrix} 4 & 0 & 1 \\ 8 & 1 & -3 \\ 12 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}.$$

To find the inverse of A:

$$\left[\begin{array}{ccc|ccc} \boxed{4} & 0 & 1 & 1 & 0 & 0 \\ 8 & 1 & -3 & 0 & 1 & 0 \\ 12 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \xrightarrow[\substack{[2]\leftarrow[2]-2[1] \\ [1]\leftarrow[3]-3[3]}]{} \left[\begin{array}{ccc|ccc} \boxed{4} & 0 & 1 & 1 & 0 & 0 \\ 0 & \boxed{1} & -5 & -2 & 1 & 0 \\ 0 & 1 & -3 & -3 & 0 & 1 \end{array}\right]$$

$$\xrightarrow[{[3]\leftarrow[3]-[2]}]{} \left[\begin{array}{ccc|ccc} \boxed{4} & 0 & 1 & 1 & 0 & 0 \\ 0 & \boxed{1} & -5 & -2 & 1 & 0 \\ 0 & 0 & \boxed{2} & -1 & -1 & 1 \end{array}\right] \begin{array}{c}\text{row}\\ \text{echelon}\\ \text{form}\end{array}$$

$$\xrightarrow[\substack{[1]\leftarrow[1]-\frac{1}{2}[3] \\ [2]\leftarrow[2]+\frac{5}{2}[3]}]{} \left[\begin{array}{ccc|ccc} \boxed{4} & 0 & 0 & \frac{3}{2} & \frac{1}{2} & -\frac{1}{2} \\ 0 & \boxed{1} & 0 & -\frac{9}{2} & -\frac{3}{2} & \frac{5}{2} \\ 0 & 0 & \boxed{2} & -1 & -1 & 1 \end{array}\right]$$

$$\xrightarrow[\substack{[1]\leftarrow\frac{1}{4}[1] \\ [3]\leftarrow\frac{1}{2}[2]}]{} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & \frac{3}{8} & \frac{1}{8} & -\frac{1}{8} \\ 0 & 1 & 0 & -\frac{9}{2} & -\frac{3}{2} & \frac{5}{2} \\ 0 & 0 & 1 & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{array}\right]$$

That is,

$$A^{-1} = \begin{bmatrix} \frac{3}{8} & \frac{1}{8} & -\frac{1}{8} \\ -\frac{9}{2} & -\frac{3}{2} & \frac{5}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix}$$

The solution to the system is therefore

$$x = A^{-1}b = \begin{bmatrix} \frac{3}{8} & \frac{1}{8} & -\frac{1}{8} \\ -\frac{9}{2} & -\frac{3}{2} & \frac{5}{2} \\ -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 4 \\ 3 \\ 1 \end{bmatrix}, \quad \text{i.e., } x_1 = \frac{7}{4}\,,\ x_2 = -20\,,\ x_3 = -3\,.$$

(b) The determinant of A is the product of the diagonal of the row echelon form, i.e., $\det(A) = 8$ (we did not swap rows or multiple any row by a factor in obtaining the row echelon form). Likewise, you can show that the determinants of $A_1(b)$, $A_2(b)$ and $A_3(b)$ are

$$\det(A_1(b)) = \begin{vmatrix} 4 & 0 & 1 \\ 3 & 1 & -3 \\ 1 & 1 & 0 \end{vmatrix} = 14\,,\ \det(A_2(b)) = \begin{vmatrix} 4 & 4 & 1 \\ 8 & 3 & -3 \\ 12 & 1 & 0 \end{vmatrix} = -160$$

$$\text{and}\quad \det(A_3(b)) = \begin{vmatrix} 4 & 0 & 4 \\ 8 & 1 & 3 \\ 12 & 1 & 1 \end{vmatrix} = -24$$

which gives

$$x_1 = 14/8 = 7/4\,,\ x_2 = -160/8 = -20 \ \text{ and } \ x_3 = -24/8 = -3\,.$$

Ex. 8.34: No. $Ab = Ac$ implies $b = c$ only if A is non-singular, in which case

$$Ab = Ac \ \Rightarrow\ A^{-1}Ab = A^{-1}Ac \ \Rightarrow\ b = c.$$

But if A is singular, then it is possible that $Ab = Ac$ but $b \neq c$. For example,

$$\begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 2 \\ 0 \end{bmatrix}.$$

Ex. 8.35:

```
A = sp.Matrix([[4, 0, 1], [8, 1, -3], [12, 1, 0]])
b = sp.Matrix([4, 3, 1])

x = A.solve(b)
sp.pprint(x)
```

```
7/4

-20

-3
```

Ex. 8.36: The quadratic form $Q = c^{\mathrm{T}} X^{\mathrm{T}} X c = (Xc)^{\mathrm{T}}(Xc)$. Since Xc is $n \times 1$ for all c, Q is a sum of squares which cannot be negative. If $Xc \neq 0$ for all k-vectors $c \neq 0_k$, then Xc is a non-zero $n \times 1$ vector, and the sum of squares $(Xc)^{\mathrm{T}}(Xc)$ will be strictly positive.

Ex. 8.37: We can find the rank by Gaussian elimination. Swap the first and second rows of X. Then (a) subtract first row from all rows below it, (b) subtract row 2 from rows 4, 5 and 6, (c) use the pivot in column 4 to eliminate everything below it (the last step is not shown):

$$
\begin{bmatrix}
\boxed{1} & 1 & 0 & 12.02 \\
1 & 0 & 1 & 55.12 \\
1 & 1 & 0 & 3.54 \\
1 & 0 & 1 & 10.89 \\
1 & 0 & 1 & 20.8 \\
1 & 0 & 1 & 19.12 \\
1 & 1 & 0 & 19.23 \\
1 & 1 & 0 & 9.62
\end{bmatrix}
\to
\begin{bmatrix}
\boxed{1} & 1 & 0 & 12.02 \\
0 & \boxed{-1} & 1 & 43.1 \\
0 & 0 & 0 & -8.48 \\
0 & -1 & 1 & -1.13 \\
0 & -1 & 1 & 8.78 \\
0 & -1 & 1 & 7.1 \\
0 & 0 & 0 & 7.21 \\
0 & 0 & 0 & -2.4
\end{bmatrix}
\to
\begin{bmatrix}
\boxed{1} & 1 & 0 & 12.02 \\
0 & \boxed{-1} & 1 & 43.1 \\
0 & 0 & 0 & \boxed{-8.48} \\
0 & 0 & 0 & -44.23 \\
0 & 0 & 0 & -34.32 \\
0 & 0 & 0 & -36.0 \\
0 & 0 & 0 & 7.21 \\
0 & 0 & 0 & -2.4
\end{bmatrix}.
$$

This shows that $\mathrm{rank}(X) = 3$. An quicker way is to note that the first column of X is obviously the sum of its second and third column, whereas columns 2 to 4 are clearly independent, so the rank is three. The rank of the 4×4 matrix $X^{\mathrm{T}}X$ is also 3. Since $X^{\mathrm{T}}X$ is not full rank, it does not have an inverse.

Ex. 8.38:

```
H_rank = np.linalg.matrix_rank(H)
print(f"The rank of the matrix H is: {H_rank}")
```

```
The rank of the matrix H is: 3
```

Ex. 8.39: We have

$$
\begin{aligned}
Var(x) &= E((x - E(x))(x - E(x))^{\mathrm{T}}) \\
&= E(xx^{\mathrm{T}} - xE(x)^{\mathrm{T}} - E(x)x^{\mathrm{T}} + E(x)E(x)^{\mathrm{T}}) \\
&= E(xx^{\mathrm{T}}) - E(x)E(x)^{\mathrm{T}} - E(x)E(x)^{\mathrm{T}} + E(x)E(x)^{\mathrm{T}} \\
&= E(xx^{\mathrm{T}}) - E(x)E(x)^{\mathrm{T}}
\end{aligned}
$$

Ex. 8.40: $\mathrm{trace}(X) = X_{11} + X_{22} + \cdots + X_{nn}$. Therefore

$$
\begin{aligned}
E(\mathrm{trace}(X)) &= E(X_{11} + X_{22} + \cdots + X_{nn}) \\
&= E(X_{11}) + E(X_{22}) + \cdots + E(X_{nn}) = \mathrm{trace}(E(X))\,.
\end{aligned}
$$