Matrices and Their Statistical Applications

Marco R. Steenbergen

2019-08-07

Contents

Chapter 1

Introduction

Much of quantitative social science concerns itself with multivariate statistical procedures. By this, we mean that we consider multiple (i.e., more than 2) variables simultaneously. While this is desirable, as it permits testing complex theories, there also is a price to pay: increased mathematical complexity. Fortunately, we can reduce the complexity by switching to matrices.

At first, matrix algebra will look like a completely new language. It takes some time to get used to a new notation and novel operations. Operations that can be performed without any difficulties in "ordinary" algebra may not always work for matrices. Some operations have multiple gestations in the world of matrices. It can all be quite confusing.

In the long run, you will discover that it is not all that complex. You will also realize the benefits of working with matrices. Aside from notational elegance, matrices open up new perspectives, especially in data analysis. Moreover, mathematical and statistical computations with matrices feel quite natural.

The present notes serve as an introduction to those aspects of matrix algebra that I have found useful for social science methodology. I do not intend to offer an exhaustive view of the topic. At the same time, my goal is to move beyond the typical discussions of matrix algebra in the social sciences. Those tend to stop at the end of Chapter 4 of the current manuscript, leaving out numerous topics that are particularly relevant for data reduction and unsupervised machine learning.

The notes show numerous hand computations. It is useful to work through those computations at least once. In practice, however, we leave the compu-

tational side to software. Programs such as R are extremely adept at matrix computations. The last chapter of the book, then, shows how R can be used to perform matrix algebra. Knowing some simple syntax goes a long way toward turning the ideas in these notes into highly useful tools for data analysis.

Chapter 2

Introducing Matrices

2.1 What Is a Matrix?

Matrices are rectangular arrays, usually of numbers, treated as a single entity. The most common example of such an array in statistics is our data. The most prevalent data structure is a two-way, two-mode matrix that distinguishes variables in the columns from observational units in the rows. Each cross-section of a row and column gives an observation, i.e., a data value for some unit on some variable. Although there are obviously numerous data values (for a typical survey there are often as many as 1,000,000 observations), these values are united in that they constitute our data. Therefore, we are inclined to treat the numbers as a single object: a matrix. This allows us to conceptualize mathematical operations on the numbers as if they take place on a single object.

The example of a **data matrix** gives away several characteristics of matrices. First, matrices consist of rows and columns. Second, a matrix contains values. An example of a matrix is the following:

$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

This matrix has three rows and two columns that could, for example, identify the values of three individuals (rows) on two different variables (columns). Of course, without further annotation it is not clear what information is exactly contained in the matrix. Therefore, it is important to always explain what

a matrix means—what sort of information is indicated? Hereby, it is useful to restrict the matrix to a single type of content, for example, the data or observations on predictors in a model.

It is useful to introduce some jargon at this point. First, rather than saying that a matrix has m rows and n columns, mathematicians tend to say that the matrix is of **order** $m \times n$. Expressing the number of rows and columns as a product is useful, because expansion then identifies the total number of values in the matrix. These values are referred to as **elements** and typically they are denoted by a lowercase letter and two subscripts, the first of which denotes the row in which the value occurs and the second of which refers to the column. The matrix itself is usually denoted by a capital, bold-face letter.

We can now revisit to the sample matrix considered earlier. This matrix had three rows and two columns and hence is of order 3×2. The value of this product indicates that there are 6 elements in total. If we label the matrix $\mathbf{A}$, then each element in the matrix may be labeled by a_{ij} , where i refers to a particular row and j to a particular column. Thus,

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

If we want to look up the score of the second individual on the first variable, we should look for element a_{21}; the value of this element is 2.

2.2 Types of Matrices

It is useful to distinguish between several types of matrices that play a distinctive role in matrix algebra and multivariate statistical analysis.

2.2.1 Vectors

A matrix that consists of only one row or column is called a **vector**. Matrices that consist of a single row are sometimes referred to as row vectors or row matrices, while matrices that consist of a single column are sometimes called column vectors or column matrices. An example of a row vector is:

$$\mathbf{u} = \begin{bmatrix} 1 & 4 \end{bmatrix}$$

This vector, which is of order 1×2, gives the values on all variables for the first individual in matrix $\mathbf{A}$ above.

An example of a column vector is:

$$\mathbf{v} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

This vector (of order 3×1) gives the values of all individuals on variable 1 in matrix $\mathbf{A}$ above. As these two examples illustrate, it is common to denote vectors by boldface, lowercase letters, although alternative notations such as $(\vec{u})$ are also used. In statistics, it is often assumed that all vectors are column vectors, a practice that we adopt for these notes as well.

2.2.2 Square Matrices

If a matrix has as many columns as it has rows—i.e., the matrix is of order $m \times m$—it is called a square matrix. (Matrices that are not square are generally referred to as **rectangular**. Rectangular matrices are all matrices of order $m \times n$, where $m \neq n$.) An example of a square matrix is:

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

Generally, data matrices are not square because there are typically more observational units than variables. An example of a square matrix in statistics is the correlation matrix, $\mathbf{R}$, which has m variables listed in its rows and in its columns.

The (main) **diagonal** of a symmetric matrix consists of those elements where the row and column indices are the same. In our example, we have three such elements: $a_{11} = 1$, $a_{22} = 5$, and $a_{33} = 1$.

2.2.3 Symmetric Matrices

Square matrices that are symmetric when "mirrored" at the main diagonal are called symmetric matrices. For such matrices, $a_{ij} = a_{ji}$. Consider, for example,

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

It is easily verified that this is a symmetric matrix. For instance, $a_{21} = 2$ and $a_{12} = 2$, so that $a_{12} = a_{21}$.

A statistical example of a symmetric matrix is again the correlation matrix, $\mathbf{R}$. As discussed earlier, the rows and columns of this matrix denote variables. The off-elements of the matrix denote the correlations between those variables. For example, r_{12} is the correlation between variable 1 and variable 2. It is obvious that the correlation between variable 2 and variable 1, r_{21}, is identical to r_{12}. Thus, the correlation matrix is symmetric in nature. The diagonal elements of $\mathbf{R}$ give the correlations of variables with themselves and are equal to 1, by definition.

2.2.4 Diagonal Matrices

Diagonal matrices are a sub-set of symmetric matrices. These matrices have non-zero values only on the diagonal. The off-diagonal elements are all zero.

Scalar matrices are a special case of diagonal matrices. In these matrices, all of the diagonal elements are equal to the same scalar: $a_{ii} = k$ for all i. The following is an example of a 3×3 scalar matrix with $k = 2$:

$$\mathbf{A} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

A scalar matrix for which $k = 1$ is called an **identity matrix**. This type of matrix is abbreviated as $\mathbf{I}$, which is sometimes followed by a subscript to indicate the order of the matrix. For example, $\mathbf{I}_4$ is the 4×4 identity matrix:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Identity matrices play a central role in matrix algebra, playing a role that is similar to the constant 1 in ordinary algebra.

2.2.5 Triangular Matrices

A triangular matrix is one that has non-zero values only on one side of the diagonal.[1] In a **lower-triangular matrix**, the elements above the diagonal are all 0. Thus, a lower-triangular matrix, $\mathbf{L}$, is a matrix with elements

$$l_{ij} = \begin{cases} a_{ij} & \text{if } i \geq j \\ 0 & \text{if } i < j \end{cases}$$

where a_{ij} is a scalar. For instance,

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix}$$

is a lower-triangular matrix. The elements of an **upper-triangular matrix**, $\mathbf{U}$, are such that everything below the diagonal is 0. That is,

$$u_{ij} = \begin{cases} a_{ij} & \text{if } i \leq j \\ 0 & \text{if } i > j \end{cases}$$

An example of an upper-traingular matrix is

$$\mathbf{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

2.2.6 Partitioned Matrices

A partitioned matrix is a matrix whose elements are not scalars but other matrices (or vectors). Any matrix can be partitioned into two or more sub-matrices. For instance, our data matrix can be partitioned as:

$$\begin{bmatrix} \mathbf{x}_1 & \mathbf{x}_2 \end{bmatrix}$$

[1] In some textbooks, the diagonal elements are required to be 0 as well.

where $\mathbf{x}_1$ is a column vector with elements 1, 2, 3, and $\mathbf{x}_2$ is a column vector with elements 4, 5, 6. One can think of $\mathbf{x}_1$ as a container holding all of the data on the first variable, whereas $\mathbf{x}_2$ holds the data on the second variable.

An alternative way of partitioning the data matrix is

$$\begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix}$$

Here, $\mathbf{x}_1 = (1, 4)$ contains the data on all of the variables for the first individual. Likewise, $\mathbf{x}_2 = (2, 5)$ and $\mathbf{x}_3 = (3, 6)$ contain the data for the second and third individuals.

In general, any matrix $\mathbf{A}$ can be partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} & \cdots & \mathbf{A}_{1K} \\ \mathbf{A}_{21} & \mathbf{A}_{22} & \cdots & \mathbf{A}_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{K1} & \mathbf{A}_{K2} & \cdots & \mathbf{A}_{KK} \end{bmatrix}$$

where K is some constant. If the off-diagonal matrices $\mathbf{A}_{ij}$ always consist entirely of zeros, we can write

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} & \cdots & \mathbf{0} \\ \text{`}\mathbf{0} & \mathbf{A}_{22} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{KK} \end{bmatrix}$$

We call this a **block-diagonal** matrix. As the name implies, the matrix is diagonal with the diagonal elements constituting blocks in the form of sub-matrices.

2.3 Sparse Matrices

Sparse matrices are matrices that consist mostly of 0s. More precisely, a sparse matrix is any matrix with a **sparsity** of greater than 0.5, where sparsity is the proportion of 0 elements over the total number of elements in the matrix.

Example 2.1. The matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & 2 & 1 \\ 0 & -1 & 2 & 0 & 0 \\ -3 & -1 & 10 & 0 & 0 \\ 0 & 0 & 0 & 5 & 1 \\ 1 & 0 & 0 & 0 & 3 \end{bmatrix}$$

is a sparse matrix. It has 25 elements but 13 of those are 0, resulting in a sparsity of 0.52.

Sparse matrices are frequently encountered in text analysis in the form of **document-term** matrices. Such matrices list documents in the rows and terms used in those documents in the columns. The elements are frequencies. Since not all documents use all terms, you will often find that many of the elements are zero.

From a mathematical perspective, we do not have to worry a great deal about sparse matrices. However, from a computational perspective they may require special handling. Frequently sparse matrices are also very large, document-term matrices being a prime example. It is useful to store those matrices in a compact format, lest the computer quickly runs out of memory.

Chapter 3

Matrix Operations

Matrices are only useful in multivariate statistical analysis if we can perform mathematical operations on them. Those operations include transposing a matrix, computing the trace, addition, multiplication, inversion, and so-called elementary row and column operations.

3.1 The Transpose of a Matrix

The transpose of a matrix is a new matrix that interchanges the rows and the columns of the original matrix. Thus, if $\mathbf{A}$ is of order $m \times n$, its transpose, $\mathbf{A}^\top$ (or $\mathbf{A}'$) is of order $n \times m$. The element a_{ij} in $\mathbf{A}$ becomes the element $a_{ji}^\top$ in $\mathbf{A}^\top$.

Example 3.1. Consider

$$\mathbf{A}_{2\times 3} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

The transpose of this matrix is a 3×2 matrix. The ith row of this new matrix is equal to the ith column in $\mathbf{A}$. Equivalently, the jth column of $\mathbf{A}^\top$ is identical to the jth row of $\mathbf{A}$. Hence,

$$\mathbf{A}_{3\times 2}^\top = \begin{bmatrix} 1 & 4 \\ 2 & 3 \\ 3 & 6 \end{bmatrix}$$

We see that the 1st row in $\mathbf{A}$ is now the 1st column in $\mathbf{A}^\top$, etcetera.

Transposes are widely used in multivariate statistics. One use is to turn column vectors into row vectors. In the previous chapter, we saw that statisticians often assume vectors to be column vectors. Should we desire to change this, for example to perform some of the multiplications we shall see later, then one simply takes the transpose: $\mathbf{u}^\top$. The column vector $\mathbf{u}$ is now flipped on its side and becomes a row vector.

Transposes have several important properties.

1. Transposition is reflexive: $\left(\mathbf{A}^\top\right)^\top = \mathbf{A}$.
2. The transpose of a diagonal matrix is the diagonal matrix. The two matrices are identical, meaning that (1) they are of the same order and (2) the corresponding elements are the same.
3. The transpose of a partitioned matrix is the transposed matrix of the transposed sub-matrices. For example, let $\mathbf{X} = [\mathbf{A}\ \mathbf{B}]$. Then

$$\mathbf{X}^\top = \begin{bmatrix} \mathbf{A} & \mathbf{B} \end{bmatrix}^\top = \begin{bmatrix} \mathbf{A}^\top \\ \mathbf{B}^\top \end{bmatrix}$$

3.2 The Trace of a Matrix

The trace of a square matrix is equal to the sum of the diagonal elements. Thus, given $\mathbf{A}_{m\times m}$, the trace is

$$\mathrm{Tr}(\mathbf{A}) = \sum_{i=1}^{m} a_{ii}$$

Example 3.2. Consider

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -1 & 0 \\ 3 & 1 & 2 \end{bmatrix}$$

Then

$$\mathrm{Tr}(\mathbf{A}) = a_{11} + a_{22} + a_{33} = 1 + (-1) + 2 = 2$$

Keep in mind that the trace is defined only for square matrices. This is one of the instances where matrix operations exist only for a subset of matrices.

3.3 Matrix Addition and Subtraction

Any two matrices of the same order can be added or subtracted (in this case the matrices are **conformable** for addition/subtraction). The resulting matrix is of the same order as the two matrices that are added or subtracted. Its elements are given by the sum or difference of the corresponding elements in the original matrices.

Example 3.3. For example,

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

Thus, we start with two 2×2 matrices and obtain a 2×2 matrix. The elements of this matrix are equal to the sums of the corresponding elements in the matrices on the left-hand side of the equation.

Example 3.4. Now consider a case of subraction:

$$\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} - \begin{bmatrix} 3 & 3 \\ 3 & 3 \end{bmatrix} = \begin{bmatrix} 1-3 & 2-3 \\ 2-3 & 1-3 \end{bmatrix} = \begin{bmatrix} -2 & -1 \\ -1 & -2 \end{bmatrix}.$$

In general, the following rules describe the addition or subtraction of two matrices:

1. Two matrices $\mathbf{A}$ and $\mathbf{B}$ can be added or subtracted if both are of order $m \times n$.
2. The result of the addition or subtraction is a new matrix $\mathbf{C}$ that also is of order $m \times n$.
3. The elements of $\mathbf{C}$ are given by $c_{ij} = a_{ij} \pm b_{ij}$, where $\pm$ is positive for addition and negative for subtraction.

In the case of partitioned matrices, these rules are applied to the sub-matrices, as the following example illustrates.

Example 3.5. Imagine that we have collected verbal and mathematical test scores from pupil samples in two classrooms. The first three observations pertain to classroom 1 and the last two to classroom 2. The verbal scores are

$\mathbf{x}^\top = (100, 50, 80, 30, 60)$, while the math scores are $\mathbf{y}^\top = (90, 70, 40, 10, 80)$. We partition the two vectors as follows

$$\mathbf{x} = \left[\begin{array}{c} 100 \\ 50 \\ 80 \\ \hline 30 \\ 60 \end{array}\right] = \left[\begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \end{array}\right] \text{ and } \mathbf{y} = \left[\begin{array}{c} 90 \\ 70 \\ 40 \\ \hline 10 \\ 80 \end{array}\right] = \left[\begin{array}{c} \mathbf{y}_1 \\ \mathbf{y}_2 \end{array}\right]$$

Vectors with the subscript 1 contain test scores from the first classroom. Those with the subscript 2 pertain to the second classroom. Since the verbal and math score vectors are of equal length in each classroom, they are conformable for addition. Thus, we can compute a total test score as follows:

$$\mathbf{x} + \mathbf{y} = \left[\begin{array}{c} \mathbf{x}_1 \\ \mathbf{x}_2 \end{array}\right] + \left[\begin{array}{c} \mathbf{y}_1 \\ \mathbf{y}_2 \end{array}\right] = \left[\begin{array}{c} \mathbf{x}_1 + \mathbf{y}_1 \\ \mathbf{x}_2 + \mathbf{y}_2 \end{array}\right] = \left[\begin{array}{c} 100 + 90 \\ 50 + 70 \\ 80 + 40 \\ \hline 30 + 10 \\ 60 + 80 \end{array}\right]$$

Matrix addition and subtraction have several important properties.

1. Matrix addition complies with the commutative law: $\mathbf{A} + \mathbf{B} = \mathbf{B} + \mathbf{A}$.
2. Matrix addition complies with the associative law: for conformable matrices, $(\mathbf{A} + \mathbf{B}) + \mathbf{C} = \mathbf{A} + (\mathbf{B} + \mathbf{C})$.
3. The transpose of a sum/differences of conformable matrices is equal to the sum/difference of the transposes: $(\mathbf{A} \pm \mathbf{B})^\top = \mathbf{A}^\top \pm \mathbf{B}^\top$.
4. When it exists, the trace of a matrix sum/difference is equal to the sum of the traces: $\text{Tr}\,(\mathbf{A} \pm \mathbf{B}) = \text{Tr}(\mathbf{A}) \pm \text{Tr}(\mathbf{B})$.

3.4 Matrix Multiplication

Matrix multiplication is another important operation. However, there is not just one type of product. Here, I shall review the most important product types.

3.4.1 Scalar Multiplication

The simplest form of multiplication is multiplication of a matrix by a scalar, i.e., any real number. The result of this operation is a new matrix, of the same order as the original matrix, whose elements are equal to the product of the original elements and the constant. Specifically, let k denote a scalar and let $\mathbf{B} = k \cdot \mathbf{A}$, where $\mathbf{A}$ is of order $m \times n$. Then

1. $\mathbf{B}$ is of order $m \times n$.
2. $b_{ij} = k \cdot a_{ij}$.

Example 3.6.

$$3 \cdot \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 3 \times 1 & 3 \times 2 \\ 3 \times 2 & 3 \times 1 \end{bmatrix} = \begin{bmatrix} 3 & 6 \\ 6 & 3 \end{bmatrix}.$$

Scalar multiplication lies at the heart of scalar matrices. That is, any scalar matrix can be obtained by performing a scalar multiplication of an identity matrix of the appropriate order. Let $\mathbf{S}$ be a scalar matrix of order $n \times n$ with diagonal elements k. Then $\mathbf{S} = k \cdot \mathbf{I}_n$.

Scalar multiplication has several important properties:

1. For conformable matrices, $\mathbf{A}$ and $\mathbf{B}$, $k \cdot (\mathbf{A} + \mathbf{B}) = k \cdot \mathbf{A} + k \cdot \mathbf{B}$.
2. $(k \times \mathbf{A})^\top = k \cdot \mathbf{A}^\top$.

3.4.2 Dot Products

Dot products involve the multiplication of two vectors of equal length and result in a scalar outcome.[1] The first vector in the product should be a row vector, while the second vector should be a column vector. The inner product is now equal to the sum of the products between corresponding elements in the vectors. That is, the first element in the row vector is multiplied by the

[1]Dot products are sometimes also called inner products. However, technically they are a special case of inner-products that arises when the result is a real number. In Eculidean space, the distinction is irrelevant but in other vector spaces, inner-products take the place of dot products. They are typically indicated by $\langle \mathbf{u}, \mathbf{v} \rangle$, a notation that you often see in discussions of support vector machines in machine learning.

first element of the column vector, the second element of the row vector is multiplied by the second element of the column vector, etcetera. All of these product terms are summed so that a scalar is returned by the inner product. Thus,

$$\mathbf{u}^\top \cdot \mathbf{v} = \sum_{i=1}^{n} u_{1i}^\top \cdot v_{i1}$$

Note that the dot between the two vector is frequently omitted.

Example 3.7. Consider the vectors $\mathbf{u}^\top = (1, 2, 3)$ and $\mathbf{v}^\top = (1, 0, 1)$. The inner product $\mathbf{u}^\top \cdot \mathbf{v}$ exists because both vectors are of equal length, containing 3 elements each. The product is given by $1 \cdot 1 + 2 \cdot 0 + 3 \cdot 1 = 4$.

Dot products have several important properties:

1. If $\mathbf{u}$ and $\mathbf{v}$ are identical, then the inner product gives the sum of squares of the elements of $\mathbf{u}$ (or $\mathbf{v}$).
2. Let k and m be two scalars. Then $\left(k \cdot \mathbf{u}^\top\right)(m \cdot \mathbf{v}) = k \cdot m \cdot \mathbf{u}^\top \mathbf{v}$.

3.4.3 Matrix Products

When matrices rather than vectors are multiplied we can conceptualize this in terms of a series of dot products. Specifically, if $\mathbf{C} = \mathbf{AB}$ exists, then c_{ij} is equal to the inner product of the ith row in $\mathbf{A}$ and the jth column in $\mathbf{B}$. Existence depends on the ith row of $\mathbf{A}$ and jth column of $\mathbf{B}$ being of equal length, so that it follows that $\mathbf{AB}$ exists only if there are as many columns in $\mathbf{A}$ as there are rows in $\mathbf{B}$. In this case, we say that the two matrices are conformable for multiplication. To summarize,

1. Consider two matrices $\mathbf{A}_{m \times n}$ and $\mathbf{B}_{p \times q}$. The matrix product $\mathbf{AB}$ exists only if $n = p$.
2. The output from the product, if it exists, is a matrix $\mathbf{C}_{m \times q}$. Thus, the output matrix thus has as many rows as $\mathbf{A}$ and as many columns as $\mathbf{B}$.
3. The elements of $\mathbf{C}$ are inner products of the type

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

In the setup that we just showed, we say that **B** is **pre-multiplied** by **A**. We could also reverse that operation and pre-multiply **A** with **B**: **BA**. However, just because **AB** exists does not mean that **BA** does as well. Moreover, even if both matrix products exist than they do not have to generate the same output matrix.

Example 3.8. To illustrate the process, consider

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

and

$$\mathbf{B} = \begin{bmatrix} 1 & 3 & 5 \\ 2 & 4 & 6 \end{bmatrix}$$

We are interested in the product **AB**. We first verify that it exists. **A** has 2 columns and **B** has 2 rows, which means that the matrices are conformable for the product **AB**. The product results in a 2×3 matrix with the following elements:

$$\begin{aligned}
c_{11} &= \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 2 \cdot 1 + 1 \cdot 2 = 4 \\
c_{12} &= \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 2 \cdot 3 + 1 \cdot 4 = 10 \\
c_{13} &= \begin{bmatrix} 2 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = 2 \cdot 5 + 1 \cdot 6 = 16 \\
c_{21} &= \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} = 3 \cdot 1 + 4 \cdot 2 = 11 \\
c_{22} &= \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \end{bmatrix} = 3 \cdot 3 + 4 \cdot 4 = 25 \\
c_{23} &= \begin{bmatrix} 3 & 4 \end{bmatrix} \begin{bmatrix} 5 \\ 6 \end{bmatrix} = 3 \cdot 5 + 4 \cdot 6 = 39
\end{aligned}$$

Hence,

$$\mathbf{C} = \begin{bmatrix} 4 & 10 & 16 \\ 11 & 25 & 39 \end{bmatrix}$$

Having seen the product **AB** in Example 3.8, one might wonder about the product **BA**. This product, however, does not exist: **B** has 3 columns, whereas

A has 2 rows only. Here we see that order matters in matrix products. The commutative law does not apply.

How about the multiplication of partitioned matrices? The logic is similar but defined in terms of the sub-matrices that make up the partitions. Consider, for instance

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_{11} \\ \mathbf{B}_{21} \end{bmatrix}.$$

If the sub-matrices are conformable for multiplication, then the matrix product is given by:

$$\mathbf{AB} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{B}_{11} + \mathbf{A}_{12}\mathbf{B}_{21} \\ \mathbf{A}_{21}\mathbf{B}_{11} + \mathbf{A}_{22}\mathbf{B}_{21} \end{bmatrix}$$

In the case of partitioned matrices, conformability is ensured if the partitioning of **A** along its columns is the same as the partitioning of **B** along its rows.

Matrix multiplication has several important proprties:

1. The product of a matrix and a conformable identity matrix is the matrix itself: $\mathbf{AI} = \mathbf{A}$.
2. Matrix multiplication follows the associative law: given conformable matrices, $\mathbf{A}(\mathbf{BC}) = (\mathbf{ABC})$.
3. Matrix multiplication follows the distributive law: given conformable matrices, $\mathbf{A}(\mathbf{B} + \mathbf{C}) = \mathbf{AB} + \mathbf{AC}$.
4. The transpose of a product is equal to the product of the transposes in reversed order: $(\mathbf{AB})^\top = \mathbf{B}^\top\mathbf{A}^\top$.
5. If the matrix products **AB** and **BA** both exist, then they have identical traces.
6. If $\mathbf{AA} = \mathbf{A}$, then **A** is **idempotent**.

3.4.4 Outer-Products

We have seen that the product of a row vector into a column vector of equal length is the dot product, which often is also called the inner-product. If we multiply the column vector into the row vector, then we obtain the outer

product. Whereas the inner product is a scalar, the outer product is a square matrix, as the following example illustrates.

Example 3.9. Consider the vectors $\mathbf{u}^\top = (1, 2, 3)$ and $\mathbf{v}^top = (1, 0, 1)$. The outer product is

$$\mathbf{vu}^\top = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 1 \cdot 1 & 1 \cdot 2 & 1 \cdot 3 \\ 0 \cdot 1 & 0 \cdot 2 & 0 \cdot 3 \\ 1 \cdot 1 & 1 \cdot 2 & 1 \cdot 3 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 0 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

3.4.5 Kronecker Products

Kronecker products play a role in the mathematical representation of, for example, variance component analysis. The product, which is typically denoted by $\mathbf{A} \otimes \mathbf{B}$, takes the elements of the first matrix, $\mathbf{A}$, to create scalar products with the second matrix, $\mathbf{B}$.

Example 3.10. Let $\mathbf{A}$ be of order 2×3 and $\mathbf{B}$ be of order 2×2, then $\mathbf{A} \otimes \mathbf{B}$ is given by

$$\begin{bmatrix} a_{11} \cdot \mathbf{B} & a_{12} \cdot \mathbf{B} & a_{13} \cdot \mathbf{B} \\ a_{21} \cdot \mathbf{B} & a_{22} \cdot \mathbf{B} & a_{23} \cdot \mathbf{B} \end{bmatrix} =$$

$$\begin{bmatrix} a_{11} \cdot b_{11} & a_{11} \cdot b_{12} & a_{12} \cdot b_{11} & a_{12} \cdot b_{12} & a_{13} \cdot b_{11} & a_{13} \cdot b_{12} \\ a_{11} \cdot b_{21} & a_{11} \cdot b_{22} & a_{12} \cdot b_{21} & a_{12} \cdot b_{22} & a_{13} \cdot b_{21} & a_{13} \cdot b_{22} \\ a_{21} \cdot b_{11} & a_{21} \cdot b_{12} & a_{22} \cdot b_{11} & a_{22} \cdot b_{12} & a_{23} \cdot b_{11} & a_{23} \cdot b_{12} \\ a_{21} \cdot b_{21} & a_{21} \cdot b_{22} & a_{22} \cdot b_{21} & a_{22} \cdot b_{22} & a_{23} \cdot b_{21} & a_{23} \cdot b_{22} \end{bmatrix}.$$

As is evident from this example, if $\mathbf{A}$ is of order $m \times n$ and $\mathbf{B}$ is of order $p \times q$, then $\mathbf{A} \otimes \mathbf{B}$ is of order $(m \times p) \times (n \times q)$. Kronecker products can be obtained from any two matrices, but the product will generally differ depending on which matrix is placed in front of the product operator.

3.4.6 Hadamard Product

The Hadamard or Schur product involves the element-wise multiplication of two matrices of the same order. Let $\mathbf{A}$ and $\mathbf{B}$ be matrices of order $m \times n$. Then the Hadamard product $\mathbf{A} \circ \mathbf{B}$ is a $m \times n$ matrix with elements $c_{ij} = a_{ij} \cdot b_{ij}$.

Example 3.11. Given

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 1 & -1 \\ 1 & -1 & 0 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 3 & 2 & 1 \end{bmatrix}$$

it follows that

$$\mathbf{A} \circ \mathbf{B} = \begin{bmatrix} -1 \cdot 1 & 0 \cdot 2 & 1 \cdot 3 \\ 0 \cdot 4 & 1 \cdot 5 & -1 \cdot 6 \\ 1 \cdot 3 & -1 \cdot 2 & 0 \cdot 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 3 \\ 0 & 5 & -6 \\ 3 & -2 & 0 \end{bmatrix}$$

3.4.7 Frobenius Inner-Product

In convolutional network analysis, a form of deep learning, we rely on the Frobenius inner-product. This is the inner-product of two matrices and the result is a scalar. Given two matrices, $\mathbf{A}$ and $\mathbf{B}$, of the same order, the Frobenius inner-product is defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = \sum_i \sum_j a_{ij} \cdot b_{ij}$$

Example 3.12. Consider

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix}$$

and

$$\mathbf{B} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

the Frobenius inner-product is defined as

$$\langle \mathbf{A}, \mathbf{B} \rangle_F = 1 \cdot 1 + 2 \cdot 0 + 3 \cdot 0 + 4 \cdot 1 + 5 \cdot 0 + 6 \cdot 0 = 5$$

3.4.8 Statistical Applications of Matrix Multiplication

3.4.8.1 Summation, Sums of Squares, and Cross-Products

Inner products can be used to define **sums** of values, which is very useful in statistics. If we want to sum the values of a variable X this can be done by defining the inner product of two vectors. The first vector contains the observations on the variable, x_i, and is of order $n \times 1$ (assuming a sample size of n):

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}.$$

the second vector is also of order $n \times 1$ and consists entirely one ones:

$$\boldsymbol{\iota} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}.$$

The definition of inner products then implies:

$$\begin{aligned} \boldsymbol{\iota}^\top \mathbf{x} &= \sum_{i=1}^{n} \iota_{1i} \cdot x_{i1} \\ &= \sum_{i=1}^{n} 1 \times x_{i1} \\ &= \sum_{i=1}^{n} x_i \end{aligned}$$

(In the last equation we dropped the subscript 1 on x because there is no possibility of confusion.)

To obtain the **sum of squares** of the elements of a $n \times 1$ vector $\mathbf{x}$, we use the following inner product:

$$\mathbf{x}^\top \mathbf{x} = \sum_{i=1}^{n} x_i^2$$

To obtain the sum of **cross-products** between the elements of two $n \times 1$ vectors $\mathbf{x}$ and $\mathbf{y}$. the following inner product can be used:

$$\mathbf{x}^\top \mathbf{y} = \sum_{i=1}^{n} x_i y_i$$

It is easy to generalize this to a set of K variables. Let $\mathbf{X}$ denote the partitioned matrix

$$\left[\begin{array}{cccc} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_K \end{array} \right],$$

where each column in the matrix is a vector consisting of observations on a particular variable. Then the matrix of sums of squares and cross-products is given by

$$\mathbf{C} = \mathbf{X}^\top \mathbf{X}$$

Example 3.13. As an example, consider data on political stability (X_1) and voice and accountability (X_2) in three random countries: Brazil, Namibia, and Sweden. The data matrix is

$$\mathbf{X} = \left[\begin{array}{cc} 30 & 62 \\ 70 & 67 \\ 82 & 100 \end{array} \right] = \left[\begin{array}{cc} \mathbf{x}_1 & \mathbf{x}_2 \end{array} \right]$$

To obtain the sums, we define $\boldsymbol{\iota}^\top = (1, 1, 1)$. The sum on stability is now

$$\left[\begin{array}{ccc} 1 & 1 & 1 \end{array} \right] \left[\begin{array}{c} 30 \\ 70 \\ 82 \end{array} \right] = 182$$

A similar computation produces a total voice and accountability score of 229. The matrix of sums of squares and cross-products is

$$\mathbf{X}^\top \mathbf{X} = \left[\begin{array}{c} \mathbf{x}_1^\top \\ \mathbf{x}_2^\top \end{array} \right] \left[\begin{array}{cc} \mathbf{x}_1 & \mathbf{x}_2 \end{array} \right] = \left[\begin{array}{cc} \mathbf{x}_1^\top \mathbf{x}_1 & \mathbf{x}_1^\top \mathbf{x}_2 \\ \mathbf{x}_2^\top \mathbf{x}_1 & \mathbf{x}_2^\top \mathbf{x}_2 \end{array} \right]$$

Here, $\mathbf{x}_1^\top \mathbf{x}_1 = 30^2 + 70^2 + 82^2 = 12524$, $\mathbf{x}_1^\top \mathbf{x}_2 = \mathbf{x}_2^\top \mathbf{x}_1 = 30 \cdot 62 + 70 \cdot 67 + 82 \cdot 100 = 14750$, and $\mathbf{x}_2^\top \mathbf{x}_2 = 62^2 + 67^2 + 100^2 = 18333$. The diagonal elements are the sums of squares, whereas the off-diagonal elements are the cross-products.

3.4.8.2 Means and Deviation Scores

The sample mean is the sum over the scores of a variable divided by the sample size. Thus,

$$\bar{x} = \frac{1}{n} \boldsymbol{\iota}^\top \mathbf{x}$$

It is easy to generalize this to a set of K variables. Consider the partitioned matrix $\mathbf{X}$ from before. Then

$$\bar{\mathbf{x}} = \frac{1}{n}\mathbf{X}^\top \boldsymbol{\iota},$$

where $\bar{\mathbf{x}}$ is a column vector.

Example 3.14. For the country sample from 3.13 we get:

$$\bar{\mathbf{x}} = \frac{1}{3}\begin{bmatrix} 30 & 70 & 82 \\ 62 & 67 & 100 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 182 \\ 229 \end{bmatrix} = \begin{bmatrix} 60.67 \\ 76.33 \end{bmatrix}$$

Often we center variables about their means. This generates so-called mean deviation scores. For a data matrix $\mathbf{X}$, the deviation scores are obtained from

$$\mathbf{D} = \mathbf{X} - \boldsymbol{\iota}\bar{\mathbf{x}}^\top$$

Example 3.15. In Example 3.14,

$$\mathbf{D} = \begin{bmatrix} 30 & 62 \\ 70 & 67 \\ 82 & 100 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} 60.67 & 76.33 \end{bmatrix} = \begin{bmatrix} -30.67 & -14.33 \\ 9.33 & -9.33 \\ 21.33 & 23.67 \end{bmatrix}$$

Note that the deviation scores sum to 0 within each column of $\mathbf{D}$.

3.4.8.3 Variances and Covariances

By definition, the sample variance of a variable X os defined as

$$s_X^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

To capture this equation through matrices, we partition the deviation scores as

$$\mathbf{D} = \begin{bmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \cdots & \mathbf{d}_K \end{bmatrix}$$

The variance is then given by

$$s_i^2 = \frac{1}{n-1}\mathbf{d}_i^\top \mathbf{d}_i$$

Example 3.16. Using the data from Example 3.13, the variance of political stability is given by:

$$s_1^2 = \frac{1}{3-1}\begin{bmatrix} -30.67 & 9.33 & 21.33 \end{bmatrix}\begin{bmatrix} -30.67 \\ 9.33 \\ 21.33 \end{bmatrix} = 741.33$$

The sample covariance is given by

$$s_{XY} = \frac{1}{n-1}(x_i - \bar{x})(y_i - \bar{y})$$

Using matrix notation, this can be expressed as follows:

$$s_{ij} = \frac{1}{n-1}\mathbf{d}_i^\top \mathbf{d}_j$$

The computation is analogous to the variance.

Rather than computing variances and covariances on specific elements of $\mathbf{X}^*$, we can obtain the entire variance-covariance matrix in one swell swoop:

$$\mathbf{S} = \frac{1}{n-1}\mathbf{D}^\top \mathbf{D}$$

Example 3.17. For Example 3.13,

$$\mathbf{S} = \frac{1}{3-1}\begin{bmatrix} -30.67 & 9.33 & 21.33 \\ -14.33 & -9.33 & 23.67 \end{bmatrix}\begin{bmatrix} -30.67 & -14.33 \\ 9.33 & -9.33 \\ 21.33 & 23.67 \end{bmatrix}$$

or

$$\mathbf{S} = \begin{bmatrix} 741.33 & 428.67 \\ 428.67 & 426.33 \end{bmatrix}$$

The diagonal contains the variances of political stability and voice/accountability, respectively. The off-diagonal elements give the covariance between those attributes.

3.4.8.4 Markov Chains

Markov chains are models that specify the dynamics of a transitional process over time, predicting a future state of a system from a past state, specifically

Table 3.1: Butler and Stokes Subjective Class Data

	Middle64	Working64
Middle63	198	64
Working63	94	639

the state immediately preceding the current time point. These models often yield interesting insights for panel or repeated measures data, in which the same units of analysis are studied at multiple points in time.

The key to Markov chains is the specification of a matrix containing the probability of changing from one state to another. This matrix is called the transition matrix and specifies the dynamics of the change process. Once known, the transition matrix is multiplied by a vector specifying the state distribution at time t. The resulting matrix contains the predictions of the state distribution at time $t + 1$.

Example 3.18. Butler and Stokes (1971) present data concerning subjective class identification in the United Kingdom. We have panel data for 1963-1964 (see Table 3.1). The transition probabilities are defined as conditional probabilities for staying in or moving out of a particular state that a person attained at an earlier time point. This transition matrix is given by:

$$\mathbf{R} = \begin{bmatrix} .76 & .24 \\ .13 & .87 \end{bmatrix}$$

(conditioning on the row totals from Table 3.1). To obtain the probability distribution of class in 1965, we pre-multiply $\mathbf{R}$ with the probability distribution of class in 1964, the most recent year for which we have data. We define this probability distribution as the column vector

$$\mathbf{p} = \begin{bmatrix} .29 \\ .71 \end{bmatrix}$$

(using the sample size and column totals to derive estimates of the probabilities). Using $\mathbf{p}$ and $\mathbf{R}$, we can now obtain the predicted future distribution of class by applying the following formula:

$$\mathbf{p}^\top \mathbf{R} = \begin{bmatrix} .29 & .71 \end{bmatrix} \begin{bmatrix} .76 & .24 \\ .13 & .87 \end{bmatrix} = \begin{bmatrix} .31 & .69 \end{bmatrix}.$$

Thus, we expect a slight increase (by 2 percentage points) of middle class identifiers and a corresponding decline in working class identifiers.

3.5 Matrix Powers

Given a matrix $\mathbf{A}$ and a constant $k > 0$, the matrix power is

$$\mathbf{A}^k = \underbrace{\mathbf{A}\mathbf{A}\cdots\mathbf{A}}_{k \text{ Terms}}$$

Example 3.19. Consider

$$\mathbf{a} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

Then

$$\mathbf{A}^3 = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 4 \\ 4 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 8 \\ 8 & 1 \end{bmatrix}$$

Note that $\mathbf{A}^0 = \mathbf{I}$.

3.6 Vectorization of Matrices

Matrix vectorization allows large matrices to be stored as vectors, which has certain computational advantages (e.g., space and speed). There are two vectorization operators: vect and vecth. Given a $m \times n$ matrix $\mathbf{A}$, vect($\mathbf{A}$) generates a $mn \times 1$ vector that stacks the columns of $\mathbf{A}$ on top of each other.

Example 3.20. If

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix},$$

then

$$\text{vect}(\mathbf{A}) = \begin{bmatrix} a_{11} \\ a_{12} \\ a_{21} \\ a_{22} \end{bmatrix}$$

The vecth operator is suitable for symmetric matrices. It is a vector half operator, meaning that it incorporates only the diagonal elements and half of the off-diagonal elements.

Example 3.21. Consider the $m \times m$ symmetric matrix

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

(The upper diagonal replicates the lower diagonal.) Now

$$\text{vecth}(\mathbf{A}) = \begin{bmatrix} a_{11} \\ a_{21} \\ a_{31} \\ a_{22} \\ a_{32} \\ a_{33} \end{bmatrix}$$

We get this by taking the first column. We stack this on top of the second column, starting with the diagonal element, etcetera. In this way, redundant elements (everything above the diagonal) are not replicated.

3.7 Elementary Row and Column Operations

In linear algebra, there are several mathematical operations that are so fundamental that they are considered elementary. These operations are applied to the rows or the columns of a matrix, whence the name **elementary row operations** or **elementary column operations**. They include the following:

1. Interchanging two rows or two columns.
2. Multiplying each row or column by a non-zero scalar.
3. Adding a non-zero multiple of one row to another row or adding a non-zero multiple of one column to another column.

These operations play an important role in solving systems of equations (the topic of Chapter 4). Each of these operations can be performed through matrix multiplication. The following examples show how this is done.

Example 3.22. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}.$$

To interchange the rows of this matrix, we use the following multiplication:

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \end{bmatrix}$$

(you should verify this result).

Example 3.23. To multiply the first row of **A** in Example 3.22 by 4 and the second row by -1, we use the following multiplication:

$$\begin{bmatrix} 4 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 4 & 8 & 12 \\ -4 & -5 & -6 \end{bmatrix}.$$

Example 3.24. To add -4 times the first row to the second row of **A**, we perform this multiplication:

$$\begin{bmatrix} 1 & 0 \\ -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 6 \end{bmatrix}$$

It is possible to perform similar operations on the columns of a matrix. We illustrate this for the operation of interchanging the columns of a matrix.

Example 3.25. To interchange the first two columns of **A**, we perform the multiplication

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 3 \\ 5 & 4 & 6 \end{bmatrix}$$

All of the elementary row and column operations can be represented through matrix products involving the matrix that should be transformed and a matrix containing the elementary operators. We shall call this second matrix **E**. It has the following properties:

1. **E** is a square matrix.
2. To perform elementary row operations on a matrix **A** of order $m \times n$, **A** is pre-multiplied by a matrix **E** of order $m \times m$. To perform elementary column operations on a matrix **A** of order $m \times n$, **A** is post-multiplied by a matrix **E** of order $n \times n$.

3. Rows are interchanged by specifying $\mathbf{E}$ as an identity matrix with the ith and jth rows interchanged. Columns are interchanged by specifying $\mathbf{E}$ as an identity matrix with the ith and jth columns interchanged.

4. To multiply rows and columns by a scalar, $\mathbf{E}$ should be specified as a scalar matrix with the appropriate diagonal elements.

5. To add a multiple of one row to another row, or to add a multiple of one column to another column, $\mathbf{E}$ is specified as a triangular with one or more non-zero diagonal elements of approprate sign and magnitude.

6. A row remains unchanged if the same row in $\mathbf{E}$ is as one would find in an identity matrix. A column remains unchanged if the same column in $\mathbf{E}$ is specified as in an identity matrix.

3.8 The Determinant of a Matrix

The determinant is a single number that characterizes a square matrix. In Chapter 6, we shall consider the geometric meaning of this number. For now, however, it suffices to say that the determinant provides essential information about a matrix. Specifically, a determinant of 0 means there are redundancies in the matrix.

3.8.1 Computing the Determinant

Notions vary but we shall denote the determinant of a matrix $\mathbf{A}$ as $\det(\mathbf{A})$.[2] Computation of the determinant is easiest in a 2×2 matrix. Consider

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

The determinant is then equal to the product of the diagonal elements minus the product of the off-diagonal elements:

$$\det(\mathbf{A}_{2\times 2}) = a_{11} \cdot a_{22} - a_{12} \cdot a_{21}$$

[2]Sometimes, authors use $|\mathbf{A}|$ to denote the determinant.

Example 3.26.

$$\det \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} = 1 \cdot 4 - 2 \cdot 3 = -2$$

For larger matrices, we can use the **method of cofactors** to compute the determinant. Consider the generic $m \times m$ matrix

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

For this matrix the **minor**, M_{ij}, is defined as the determinant of the sub-matrix of $\mathbf{A}$ that results from deleting the ith row and the jth column. The **cofactor**, C_{ij}, is a signed version of the minor:

$$C_{ij} = (-1)^{i+j} M_{ij} = (-1)^{i+j} \det\left(\mathbf{A}_{-i,-j}\right)$$

The determinant can be obtained through cofactor expansion along a particular row or column of $\mathbf{A}$. Specifically, using cofactor expansion along the ith row, we get

$$\det(vmA) = \sum_j a_{ij} \cdot C_{ij}$$

Using cofactor expansion along the jth column, we obtain

$$\det(\mathbf{A}) = \sum_i a_{ij} \cdot C_{ij}$$

Either method results in the same value of the determinant.

There is, of course, a circular element to this method of calculating the determinant: cofactors are themselves based on determinants, so that we have shifted the problem of finding the determinant of a larger matrix to the problem of finding it for a smaller matrix. The only way to break this circularity is to define an elementary determinant that requires no knowledge of the determinants of any of its sub-matrices. This is where the formula for the determinant of a 2×2 matrix comes into play.

Example 3.27. Consider the 3×3 matrix

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

We shall perform cofactor expansion along the first row, meaning that

$$\det(\mathbf{A}) = 1 \cdot C_{11} + 2 \cdot C_{12} + 3 \cdot C_{13}$$

Here,

$$\begin{aligned} C_{11} &= (-1)^{1+1} \cdot M_{11} \\ C_{12} &= (-1)^{1+2} \cdot M_{12} \\ C_{13} &= (-1)^{1+3} \cdot M_{13} \end{aligned}$$

To obtain the minors, we need to generate the appropriate sub-matrices. To compute M_{11}, for example, we remove the 1st row and 1st column of $\mathbf{A}$. This produces

$$\begin{bmatrix} 5 & 4 \\ 2 & 1 \end{bmatrix}$$

Using the formula for determinants of 2×2 matrices, the minpr is now

$$M_{11} = 5 \cdot 1 - 4 \cdot 2 = -3$$

Similarly, M_{12} is the determinant of the sub-matrix that is formed by eliminating the 1st row and the 2nd column:

$$M_{12} = \det \begin{bmatrix} 4 & 4 \\ 3 & 1 \end{bmatrix} = 4 \cdot 1 - 4 \cdot 3 = -8$$

Finally,

$$M_{13} = \det \begin{bmatrix} 4 & 5 \\ 3 & 2 \end{bmatrix} = 4 \cdot 2 - 5 \cdot 3 = -7$$

Putting everything together, we get

$$\det(\mathbf{A}) = 1 \cdot (-1)^{1+1} \cdot -3 + 2 \cdot (-1)^{1+2} \cdot -8 + 3 \cdot (-1)^{1+3} \cdot -7 = -8$$

Things become more complex for even bigger matrices. The ideas remain the same, but recursion is generally required. For example, if we want to compute the determinant of a 4×4 matrix, we could again perform co-factor expansion along the 1st row. In this computation, the cofactors are still signed versions of the minors. The minors are the determinants of sub-matrices that are obtained by dropping the 1st row of the matrix and each of the columns. However, those are now 3×3 matrices. Figuring out the determinants of those matrices requires another set of cofactor expansions. This can get ugly in a hurry. Fortunately, we can leave the task to the computer, as we shall see in the last chapter of these notes.

Determinants have several important properties:

1. The determinant of any identity matrix is 1: $\det(\mathbf{I}) = 1$.
2. The determinant of the transpose of a matrix is identical to the determinant of the original matrix: $\det\left(\mathbf{A}^\top\right) = \det(\mathbf{A})$.
3. The determinant of the product of two matrices is equal to the product of their determinants: $\det(\mathbf{AB}) = \det(\mathbf{A}) \cdot \det(\mathbf{B})$.
4. Let $\mathbf{A}$ be of order $m \times m$ and let k be a scalar. Then $\det(k\mathbf{A}) = k^m \det(\mathbf{A})$.

3.8.2 Zero Determinants and Matrix Ranks

Let us look at another example of computing the determinant. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 4 \\ 2 & 1 & -2 \end{bmatrix}$$

This matrix is identical to the matrix in 3.27, except that we have changed the last row. Using co-factor expansion along the 1st row, we obtain a determinant of

$$\begin{array}{ll} 1 \cdot (-1)^{1+1} \cdot (5 \cdot -2 - 4 \cdot 1) & + \\ 2 \cdot (-1)^{1+2} \cdot (4 \cdot -2 - 4 \cdot 2) & + \\ 3 \cdot (-1)^{1+3} \cdot (4 \cdot 1 - 5 \cdot 2) & = \\ 0 & \end{array}$$

Determinants of 0 like the one shown here have a special meaning in linear algebra. They show that there are **linear dependencies** in the matrix. Linear dependencies arise when (1) a row is an exact linear function of other rows in the matrix or (2) a column is an exact linear function of other columns in the the matrix. In our example, the 3rd row is equal to the 2nd row minus two times the 1st row. That means no new information is contained in the 3rd row: once we know the content of the 1st and 2nd rows, we know exactly what is in the 3rd row.

In general, a linear dependency arises if there is a set of weights, λ, such that:

$$\sum_i \lambda_i a_{ij} = 0$$

for *every* possible value of j. Alternatively,

$$\sum_j \lambda_j a_{ij} = 0$$

for all values of i. In our case,

$$\lambda_1 a_{1j} + \lambda_2 a_{2j} + \lambda_3 a_{3j} = 0$$

is satisfied for all of the columns if we set $\lambda_1 = -2$, $\lambda_2 = 1$, and $\lambda_3 = -1$. For instance, focusing on the 1st column, $-2 \times 1 + 1 \times 4 - 1 \times 2 = 0$. Applying the expression to the other columns, we see that it holds in every single case. (You should verify this.) Note that we only speak of a linear dependency if the linear equation with weights λ_i holds true for every column in the matrix. If it holds true sometimes but not at other times, there is no linear dependency.

We have approached linear dependencies from the perspective of the 3rd row in the matrix. It is important to realize that linear dependencies are not properties of rows (or columns) but of the entire matrix. We could just as easily have said that the 1st row is a perfect linear function of the 2nd and 3rd rows because, indeed, it is: multiplying the 2nd row by half and subtracting half times the 3rd row perfectly replicates the entries in the 1st row. Then again, we could also have reasoned from the perspective of the 2nd row: two times the 1st row plus the 3rd row yields the 2nd row.

By the same token, if we find that there are linear dependencies between the rows then there also have to be between the columns. Remember, linear dependencies are a property of the matrix. In our example, it is easily shown that the 1st column is equal to 8/7 times the 2nd column minus 3/7 times the 3rd column. I leave it as an exercise to verify this result.

Our focus so far has been on the linear dependencies. We can also reverse our perspective and focus on the number of **linearly independent** or so-called LIN vectors. In our example, there are two LIN vectors that together determine the third vector.

The number of LIN vectors determines the **rank** of the matrix. When there is one LIN vector, then the rank is 1. If there are two LIN vectors, then the rank is 2, etcetera. In our example, the matrix **A** has a rank of 2.

The rank of a matrix has several important properties:

1. The rank of a matrix is identical, no matter whether one focuses on the number of LIN row vectors or the number of LIN column vectors. This is because the number of LIN row vectors is always equal to the number of LIN column vectors.

2. The rank can never be greater than the smallest number that determines the order of the matrix. We can compute the rank of any rectangular

matrix, $\mathbf{A}_{m\times n}$. If $m < n$, then the rank is at most m. If $n < m$, then the rank is at most n.

3. When the rank is smaller than $\min(m, n)$, then we say that the matrix is **not full rank**.

Ranks play several important roles in statistics, for example, in determining whether a statistical model is identified: do we have enough information to estimate the parameters. It also plays a crucial role for the next topic, to wit inverses.

3.9 The Inverse of a Matrix

By now we have seen many matrix operations but one has been conspicuously absent: division. Technically, there is not matrix division in the usual sense of the term. We can think of "division" as a form of multiplication. In ordinary algebra, if we want to divide a by b we may write this as $a \cdot b^{-1}$. We can do something similar in matrix algebra: we can define $\mathbf{AB}^{-1}$. Here $\mathbf{B}^{-1}$ is known as the inverse of $\mathbf{B}$. In this section, we discuss when the inverse exists and how it can be computed.

3.9.1 Definition and Computation of the Inverse

The inverse of a matrix stands in an exact mathematical relationship to the matrix itself. In ordinary algebra, $b \cdot b^{-1} = b^{-1} \cdot b1$ (for $b \neq 0$). Similarly, for a *square* matrix, $\mathbf{B}$, the inverse, $\mathbf{B}^{-1}$, is another square matrix such that

$$\mathbf{B}\mathbf{B}^{-1} = \mathbf{B}^{-1}\mathbf{B} = \mathbf{I}$$

We can find the inverse in a variety of different ways, but we shall focus on the following equation:

$$\mathbf{B}^{-1} = \frac{1}{\det(\mathbf{B})}\text{adj}(\mathbf{B})$$

Here adj is the so-called **adjoint matrix**, which is the transpose of the matrix of cofactors.

Before we illustrate the use of the formula, it is worth calling attention to one of its implications. If $\det(\mathbf{B}) = 0$, then the inverse does not exist. After all, 0^{-1} is not defined in mathematics. In this case, we say that the matrix is

singular. While it is not possible to compute the regular inverse of a singular matrix, it may still be possible to calculate the generalized inverse, as we shall see momentarily.

Example 3.28. Consider,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 3 \end{bmatrix}$$

We start by generating the cofactors. While we only need a subset of those for computing the determinant, we need all possible co-factors to generate the adjoint matrix. Thus,

$$\begin{aligned}
C_{11} &= (-1)^{1+1}\det\begin{bmatrix} 1 & 0 \\ 2 & 3 \end{bmatrix} = 3 \\
C_{12} &= (-1)^{1+2}\det\begin{bmatrix} 0 & 0 \\ 1 & 3 \end{bmatrix} = 0 \\
C_{13} &= (-1)^{1+3}\det\begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} = -1 \\
C_{21} &= (-1)^{2+1}\det\begin{bmatrix} 0 & 1 \\ 2 & 3 \end{bmatrix} = 2 \\
C_{22} &= (-1)^{2+2}\det\begin{bmatrix} 1 & 1 \\ 1 & 3 \end{bmatrix} = 2 \\
C_{23} &= (-1)^{2+3}\det\begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} = -2 \\
C_{31} &= (-1)^{1+3}\det\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = -1 \\
C_{32} &= (-1)^{3+2}\det\begin{bmatrix} 1 & 1 \\ 0 & 0 \end{bmatrix} = 0 \\
C_{33} &= (-1)^{3+3}\det\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 1
\end{aligned}$$

and

$$\mathbf{C} = \begin{bmatrix} 3 & 0 & -1 \\ 2 & 2 & -2 \\ -1 & 0 & 1 \end{bmatrix}$$

The adjoint matrix is the transpose of this matrix:

$$\text{adj}(\mathbf{C}) = \mathbf{C}^\top = \begin{bmatrix} 3 & 2 & -1 \\ 0 & 2 & 0 \\ -1 & -2 & 1 \end{bmatrix}$$

All we have to do now is to compute the determinant. Cofactor expansion along the 1st row yields:

$$\det(\mathbf{B}) = 1 \cdot 3 + 0 \cdot 0 + 1 \cdot -1 = 2$$

The inverse is then

$$\mathbf{B}^{-1} = \frac{1}{2} \begin{bmatrix} 3 & 2 & -1 \\ 0 & 2 & 0 \\ -1 & -2 & 1 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.0 & -0.5 \\ 0.0 & 1.0 & 0.0 \\ -0.5 & -1.0 & 0.5 \end{bmatrix}$$

You can verify this is the determinant by evaluating the product $\mathbf{B}\mathbf{B}^{-1}$ and checking whether it produces the 3×3 identity matrix.

3.9.2 Inverses of Diagonal Matrices

The easiest inverses are those of diagonal matrices. Let

$$\mathbf{A} = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{mm} \end{bmatrix}$$

then

$$\mathbf{A}^{-1} = \begin{bmatrix} \frac{1}{a_{11}} & 0 & \cdots & 0 \\ 0 & \frac{1}{a_{22}} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{1}{a_{mm}} \end{bmatrix}$$

3.9.3 Inverses of Triangular Matrices

Finding the inverse of a triangular matrix is quite a bit simpler than that of a typical square matrix. Consider, the generic upper-triangular matrix

$$\mathbf{U} = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ 0 & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{mm} \end{bmatrix}$$

To find the inverse of this matrix, we apply the following steps:

1. The diagonal elements are given by $b_{ii} = u_{ii}^{-1}$.
2. For $j < i$, $b_{ij} = 0$.
3. For $j > i$, $b_{ij} = -\frac{1}{u_{ii}} \sum_{k=i+1}^{j} b_{kj} u_{ik}$

Here, b_{ij} is the element in the ith row and jth column of $\mathbf{U}^{-1}$.

Example 3.29. Let

$$\mathbf{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$$

Then the inverse is the matrix

$$\mathbf{U}^{-1} = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

From the first rule, we know that $b_{11} = 1^{-1}$, $b_{22} = 4^{-1}$, and $b_{33} = 6^{-1}$. Hence

$$\mathbf{U}^{-1} = \begin{bmatrix} 1 & b_{12} & b_{13} \\ b_{21} & \frac{1}{4} & b_{23} \\ b_{31} & b_{32} & \frac{1}{6} \end{bmatrix}$$

From the second rule, we know that $b_{21} = b_{31} = b_{32} = 0$. Thus,

$$\mathbf{U}^{-1} = \begin{bmatrix} 1 & b_{12} & b_{13} \\ 0 & \frac{1}{4} & b_{23} \\ 0 & 0 & \frac{1}{6} \end{bmatrix}$$

Now we have to find the remaining elements:

$$\begin{aligned} b_{12} &= -\frac{\left(\sum_{k=1+1}^{2} b_{k2} u_{1k}\right)}{u_{11}} = -\frac{b_{22} u_{12}}{u_{11}} = -\frac{\frac{1}{4} \cdot 2}{1} = -\frac{1}{2} \\ b_{23} &= -\frac{\left(\sum_{k=2+1}^{3} b_{k3} u_{2k}\right)}{u_{22}} = -\frac{b_{33} u_{23}}{u_{22}} = -\frac{\frac{1}{6} \cdot 5}{4} = -\frac{5}{24} \\ b_{13} &= -\frac{\left(\sum_{k=1+1}^{3} b_{k3} u_{1k}\right)}{u_{11}} = -\frac{b_{23} u_{12}}{u_{11}} - \frac{b_{33} u_{13}}{u_{11}} \\ &= -\frac{-\frac{5}{24} \cdot 2}{1} - \frac{\frac{1}{6} \cdot 3}{1} = -\frac{1}{12} \end{aligned}$$

Hence,

$$\mathbf{U}^{-1} = \begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{12} \\ 0 & \frac{1}{4} & -\frac{5}{24} \\ 0 & 0 & \frac{1}{6} \end{bmatrix}$$

(You should verify that $\mathbf{U}^{-1}\mathbf{U} = \mathbf{I}$.)

A similar approach is in place for lower-triangular matrices, $\mathbf{L}$:

1. $b_{ii} = l_{ii}^{-1}$
2. $b_{ij} = 0$ for $j > i$
3. $b_{ij} = -\frac{1}{l_{ii}} \sum_{k=j}^{i-1} l_{ik} b_{kj}$ for $i > j$

where b_{ij} denotes an element of $\mathbf{L}^{-1}$.

Example 3.30. The inverse of

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix}$$

is given by

$$\mathbf{L}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ -\frac{2}{3} & \frac{1}{3} & 0 \\ -\frac{1}{9} & -\frac{5}{18} & \frac{1}{6} \end{bmatrix}$$

(You should verify this result.)

3.9.4 Inverses of Partitioned Matrices

Inverses of partitioned matrices play an important role in statistics. The simplest case arises for block diagonal matrices. Given a matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22} \end{bmatrix},$$

the inverse is given by

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_{22}^{-1} \end{bmatrix}$$

Considerably more complex are the inverses of partitioned matrices that do not have zero off-diagonal elements. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

The inverse of this matrix is given by:

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{A}_{11}^{-1}\left(\mathbf{I} + \mathbf{A}_{12}\mathbf{F}\mathbf{A}_{21}\mathbf{A}_{11}^{-1}\right) & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{F} \\ -\mathbf{F}\mathbf{A}_{21}\mathbf{A}_{11}^{-1} & \mathbf{F} \end{bmatrix},$$

where

$$\mathbf{F} = \left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right)^{-1}$$

Computing such complex expressions is best left to a computer.

3.9.5 Properties of Inverses

1. The inverse is a square matrix.
2. The inverse is unique: a matrix has only one inverse.
3. The inverse of the identity matrix is the identity matrix: $\mathbf{I}^{-1} = \mathbf{I}$.
4. The inverse of a transpose is equal to the transpose of the inverse: $\left(\mathbf{A}^{\top}\right)^{-1} = \left(\mathbf{A}^{-1}\right)^{\top}$.
5. the inverse of a product is equal to the product of the inverses in reversed order: $(\mathbf{AB})^{-1} = \mathbf{B}^{-1}\mathbf{A}^{-1}$.
6. The determinant of the inverse is equal to the inverse of the determinant of the original matrix: $\det\left(\mathbf{A}^{-1}\right) = 1/\det(\mathbf{A})$.

3.10 The Generalized Inverse

Although the normal inverse does not exist for singular matrices, it is still possible to define the generalized inverse. The **reflexive generalized inverse**, $\mathbf{A}^{-}$, of a matrix $\mathbf{A}$ is a matrix with the following characteristics:

1. $\mathbf{A}^{-}$ is a diagonal matrix of the same order as $\mathbf{A}$. Hereby, we note that $\mathbf{A}$ does not have to be a square matrix.

2. Let r denote the rank of the matrix. Then, the first r rows and columns are defined by a matrix $\mathbf{D}_r^{-1}$, which is the inverse of the sub-matrix of that consists of the r non-redundant rows or columns of $\mathbf{A}$.

3. The remaining rows and columns are filled with zeros.

Thus,

$$\mathbf{A}^- = \begin{bmatrix} \mathbf{D}_r^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Example 3.31. Consider

$$\mathbf{A} = \begin{bmatrix} 1 & -1 & 3 \\ 1 & 1 & 2 \\ 3 & -1 & 8 \end{bmatrix}$$

This matrix is not full rank because one of the rows is redundant. For example, the third row is equal to the sum of two times the first row and the second row. Hence, $r = 2$. Treating the third row as the redundant row, the generalized inverse is given by:

$$\mathbf{A}^- = \begin{bmatrix} \mathbf{D}_2^{-1} & 0 \\ 0 & 0 \end{bmatrix}$$

Here

$$\mathbf{D}_2^{-1} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}^{-1},$$

so that we can also write the generalized inverse as

$$\mathbf{A}^- = \begin{bmatrix} \lceil 1 & -1 \rceil & -1 & 0 \\ \lfloor 1 & 1 \rfloor & & 0 \\ 0 & 0 & & 0 \end{bmatrix}.$$

A special case of the generalized inverse is the so-called **Moore-Penrose** inverse. Let $\mathbf{G}$ denote the generalized inverse. For any generalized inverse, $\mathbf{AGA} = \mathbf{A}$. The Moore-Penrose, $\mathbf{M}$, inverse requires:

1. $\mathbf{AMA} = \mathbf{A}$.

2. $\mathbf{MAM} = \mathbf{M}$.

3. $\mathbf{AM}$ and $\mathbf{MA}$ are symmetric matrices.

The most important reason for discussing the Moore-Penrose inverse is that R uses a different command for this inverse.

Chapter 4

Systems of Equations

One of the most important applications of linear algebra is systems of equations. This chapter describes how we can represent systems of equations through matrices. It then describes several strategies for solving such systems.

4.1 Matrix Representation

A system of equations consists of multiple equations (which are typically linear in nature) that have multiple unknowns or **parameters**. The objective is to solve for these parameters—i.e., to find values that make the equations true. Systems of equations play a central role in statistics, for example, in the representation of statistical models.

As an example, consider the following system of three equations:

$$\begin{aligned} -u + v + w &= 0 \\ 2u + v - w &= 4 \\ u + 3v + 2w &= 7 \end{aligned}$$

This equation has three parameters u, v, and w. These parameters carry different weights in each of the equations—i.e., they have different **coefficients**. For instance, in the first equation u has a weight of -1, and v and w have weights of 1. Finally, the outcomes of the equations appear on the right-hand side.

The solutions to this system of equations are $u = 2$, $v = 1$, and $w = 1$.[1] In this simple case, it is relatively easy to obtain these solutions. In many cases, however, systems of equations contain many equations and parameters. In these cases, it often pays off to represent the system in terms of matrices.

The general principles for representing a systems of equations as matrices are straightforward. For a system in m equations with n parameters, the following steps describe the process:

1. Group all of the outcomes in the $m \times 1$ column vector $\mathbf{y}$.
2. Group all of the parameters in the $n \times 1$ column vector $\mathbf{b}$.
3. Group the coefficients in the $m \times n$ matrix $\mathbf{X}$. The 1st row in this matrix lists the coefficients for the 1st equation, in the order in which they appear in the equation. The 2nd row does the same for the 2nd equation, etcetera.
4. Now represent the system as

$$\mathbf{y} = \mathbf{X}\mathbf{b}$$

Example 4.1. The previous system of equations can be represented through the following vectors and matrix:

$$\mathbf{y} = \begin{bmatrix} 0 \\ 4 \\ 7 \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} u \\ v \\ w \end{bmatrix},$$

[1] Throughout this report, I shall assume that the system of equations is **consistent**. Consistent systems contain equations that can be true simultaneously. For example,

$$\begin{aligned} u + v &= 5 \\ 2u + 2v &= 10 \end{aligned}$$

is consistent. On the other hand,

$$\begin{aligned} u + v &= 5 \\ 2u + 2v &= 5 \end{aligned}$$

is inconsistent: if the first equation is true, then the second equation cannot be true and vice versa.

and

$$\mathbf{X} = \begin{bmatrix} -1 & 1 & 1 \\ 2 & 1 & -1 \\ 1 & 3 & 2 \end{bmatrix}$$

4.2 Solution Approaches

4.2.1 Gauss-Jordan Elimination

The starting point of the Gauss-Jordan approach is to create a so-called **augmented matrix**. This matrix contains all the known quantities in the system, i.e., $\mathbf{X}$ and $\mathbf{y}$. We combine these quantities in the partitioned matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{X} & \mathbf{y} \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} & y_1 \\ x_{21} & x_{22} & \cdots & x_{2n} & y_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} & y_m \end{bmatrix}$$

Example 4.2. For the previous example, the augmented matrix is given by

$$\mathbf{A} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 2 & 1 & -1 & 4 \\ 1 & 3 & 2 & 7 \end{bmatrix}$$

The next step in Gauss-Jordan elimination is to use elementary row operations that will turn the sub-matrix $\mathbf{X}$ in $\mathbf{A}$ into an identity matrix. We can accomplish this through a series of elementary row operations.

Example 4.3. In Example 4.2, we need seven elementary row operations to accomplish Gauss-Jordan elimination. Denoting each operation by the subscript (i) and denoting the original augmented matrix as $\mathbf{A}_{(0)}$, we perform the following computations.

1. Change the first element in the first row of $\mathbf{A}_{(0)}$ to a 1 (since this is what one would find in an identity matrix). To perform this operation we define:

$$\mathbf{E}_{(1)} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We then compute:

$$\begin{aligned}
\mathbf{A}_{(1)} &= \mathbf{E}_{(1)}\mathbf{A}_{(0)} \\
&= \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 & 0 \\ 2 & 1 & -1 & 4 \\ 1 & 3 & 2 & 7 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -1 & -1 & 0 \\ 2 & 1 & -1 & 4 \\ 1 & 3 & 2 & 7 \end{bmatrix}
\end{aligned}$$

2. Subtract -2 times the first row in $\mathbf{A}_{(1)}$ from the second row, and -1 times the first row from the third row. This produces 0s in the first column of the second and third row, just as in the identity matrix. To perform this operation we define:

$$\mathbf{E}_{(2)} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

We then compute:

$$\begin{aligned}
\mathbf{A}_{(2)} &= \mathbf{E}_{(2)}\mathbf{A}_{(1)} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 0 \\ 2 & 1 & -1 & 4 \\ 1 & 3 & 2 & 7 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 3 & 1 & 4 \\ 0 & 4 & 3 & 7 \end{bmatrix}
\end{aligned}$$

3. Change the second element in the second row of $\mathbf{A}_{(2)}$ to a 1 (since this is what one would find in an identity matrix). To perform this operation we define:

$$\mathbf{E}_{(3)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We then compute:

$$\begin{aligned}
\mathbf{A}_{(3)} &= \mathbf{E}_{(3)}\mathbf{A}_{(2)} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \frac{1}{3} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 3 & 1 & 4 \\ 0 & 4 & 3 & 7 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 4 & 3 & 7 \end{bmatrix}
\end{aligned}$$

4. Subtract -4 times the second row in $\mathbf{A}_{(3)}$ from the third row. This produces a 0 in the second column of the third row, just as in the identity matrix. To perform this operation we define:

$$\mathbf{E}_{(4)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix}$$

We then compute:

$$\begin{aligned}
\mathbf{A}_{(4)} &= \mathbf{E}_{(4)}\mathbf{A}_{(3)} \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -4 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 4 & 3 & 7 \end{bmatrix}] \\
&= \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 0 & \frac{5}{3} & \frac{5}{3} \end{bmatrix}
\end{aligned}$$

5. Change the third element in the third row of $\mathbf{A}_{(4)}$ to a 1 (since this is what one would find in an identity matrix). To perform this operation we define:

$$\mathbf{E}_{(5)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{3}{5} \end{bmatrix}$$

We then compute:

$$\begin{aligned}\mathbf{A}_{(5)} &= \mathbf{E}_{(5)}\mathbf{A}_{(4)} \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{3}{5} \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 0 & \frac{5}{3} & \frac{5}{3} \end{bmatrix} \\ &= \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 0 & 1 & 1 \end{bmatrix}\end{aligned}$$

6. Add negative one-third times the third row to the second row, and one times the third row to the first row of $\mathbf{A}_{(5)}$. This produces 0s in the third column of the first and second rows, just as in the identity matrix. To perform this operation we define:

$$\mathbf{E}_{(6)} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix}$$

We then compute:

$$\begin{aligned}\mathbf{A}_{(6)} &= \mathbf{E}_{(6)}\mathbf{A}_{(5)} \\ &= \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 0 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}\end{aligned}$$

7. Add one times the second row to the first row of $\mathbf{A}_{(6)}$. This produces a 0 in the second column of the first row, just as in the identity matrix. To perform this operation we define:

$$\mathbf{E}_{(7)} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We then compute:

$$\begin{aligned}
\mathbf{A}_{(7)} &= \mathbf{E}_{(7)}\mathbf{A}_{(6)} \\
&= \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \\
&= \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}
\end{aligned}$$

We see that the first three columns in $\mathbf{A}_{(7)}$ constitute an identity matrix, just as was desired. This allows us to determine the solution to the system of equations. First, we note from the definition of the augmented matrix that

$$\mathbf{A}_{(7)} = \begin{bmatrix} \mathbf{X}_{(7)} & \mathbf{y}_{(7)} \end{bmatrix}$$

where

$$\mathbf{X}_{(7)} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{I}_3$$

and

$$\mathbf{y}_{(7)} = \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix}$$

Next, we apply the formula for the system of equations:

$$\begin{aligned}
\mathbf{y}_{(7)} &= \mathbf{X}_{(7)}\mathbf{b} \\
&= \mathbf{I}\mathbf{b} \\
&= \mathbf{b}
\end{aligned}$$

Thus, it follows that $\mathbf{y}_{(7)} = \mathbf{b}$:

$$\begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix},$$

so that we see that $u = 2$, $v = 1$, and $w = 1$. This completes the process of Gauss-Jordan elimination.

In Example 4.3, we could have actually obtained the solution in step (5) of the elimination process, when we had the following augmented matrix:

$$\mathbf{A}_{(5)} = \begin{bmatrix} 1 & -1 & -1 & 0 \\ 0 & 1 & \frac{1}{3} & \frac{4}{3} \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Since,

$$\mathbf{A}_{(5)} = \begin{bmatrix} \mathbf{X}_{(5)} & \mathbf{y}_{(5)} \end{bmatrix},$$

it follows that:

$$\mathbf{y}_{(5)} = \mathbf{X}_{(5)}\mathbf{b}$$

Expansion gives:

$$\begin{bmatrix} 0 \\ \frac{4}{3} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & \frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

This amounts to the equation system

$$\begin{aligned} u - v - w &= 0 \\ v + \frac{1}{3}w &= \frac{4}{3} \\ w &= 1 \end{aligned}$$

The last equation gives the solution for w. By substituting this result into the second equation, it is possible to solve for v. By substituting the solutions for v and w in the first equation, it is possible to solve for u. The augmented matrix $\mathbf{A}_{(5)}$ is called the **echelon form**. If we stop the elementary row operations after we have attained this form, we say that we have performed Gaussian elimination. (Gauss-Jordan elimination requires that we perform additional operations that turn $\mathbf{X}$ into an identity matrix.)

4.2.2 Using the Inverse

A second way of solving systems of equations is through inverses. Assuming $\mathbf{X}$ is regular, we can use it to pre-multiply $\mathbf{y} = \mathbf{Xb}$. This results in the following solution:

$$\begin{aligned} \mathbf{X}^{-1}\mathbf{y} &= \mathbf{X}^{-1}\mathbf{Xb} \\ \mathbf{X}^{-1}\mathbf{y} &= \mathbf{Ib} \\ \mathbf{X}^{-1}\mathbf{y} &= \mathbf{b} \end{aligned}$$

Example 4.4. Consider again the equation system defined by

$$\mathbf{y} = \begin{bmatrix} 0 \\ 4 \\ 7 \end{bmatrix},$$

$$\mathbf{b} = \begin{bmatrix} u \\ v \\ w \end{bmatrix},$$

and

$$\mathbf{X} = \begin{bmatrix} -1 & 1 & 1 \\ 2 & 1 & -1 \\ 1 & 3 & 2 \end{bmatrix}$$

The inverse of $\mathbf{X}$ is given by

$$\mathbf{X}^{-1} = \begin{bmatrix} -1 & -\frac{1}{5} & \frac{2}{5} \\ 1 & \frac{3}{5} & -\frac{1}{5} \\ -1 & -\frac{4}{5} & \frac{3}{5} \end{bmatrix}$$

Hence,

$$\begin{aligned} \begin{bmatrix} u \\ v \\ w \end{bmatrix} &= \begin{bmatrix} -1 & -\frac{1}{5} & \frac{2}{5} \\ 1 & \frac{3}{5} & -\frac{1}{5} \\ -1 & -\frac{4}{5} & \frac{3}{5} \end{bmatrix} \begin{bmatrix} 0 \\ 4 \\ 7 \end{bmatrix} \\ &= \begin{bmatrix} 2 \\ 1 \\ 1 \end{bmatrix} \end{aligned}$$

4.3 Singular Matrices

If $\mathbf{X}$ is singular, then a unique solution to a system of equations cannot be found. However, it is still possible to use the generalized inverse to obtain the structure of a solution, even though it is not unique.

Example 4.5. Consider the following system of linear equations:

$$\begin{aligned} -u + v + w &= 0 \\ 2u + v - w &= 1 \\ u + 2v &= 1 \end{aligned}$$

This equation system is identical to that discussed before, except for the third equation. The matrix representation for this system is:

$$\begin{array}{ccc} \mathbf{y} & = & \mathbf{Xb} \\ \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} & = & \begin{bmatrix} -1 & 1 & 1 \\ 2 & 1 & -1 \\ 1 & 2 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \end{array}$$

The problem in this equation system is that the third row of $\mathbf{X}$ is identical to the sum of the first two rows. Thus, $\mathbf{X}$ is not full rank and the inverse of $\mathbf{X}$ does not exist.

Gauss-Jordan elimination will not yield a solution either. Using the usual sequence of elementary row operations, we have:

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

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

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

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

We can go no further with the Gauss-Jordan elimination process; there is no elementary row operation that will turn the element in the third row and third column into a one. Thus, the solution process ends.

When we consider the augmented matrix produced in step 4, it represents the

following system of equations:

$$\begin{bmatrix} 1 & -1 & -1 \\ 0 & 1 & \frac{1}{3} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{3} \\ 0 \end{bmatrix}$$

or

$$\begin{aligned} u - v - w &= 0 \\ v + \frac{1}{3}w &= \frac{1}{3} \end{aligned}$$

This is a system in two equations with three parameters, which means that a unique solution cannot be obtained.

It is possible, however, to determine the general structure of the solution. If we parameterize $w = \theta$, then the solution to the system of equations implied by the computations above is:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{3} + \frac{2}{3}\theta \\ \frac{1}{3} - \frac{1}{3}\theta \\ \theta \end{bmatrix}$$

Thus, the particular value of $w = \theta$ constrains the values of u and v. For instance, if $w = \theta = 3$, it follows that $v = -\frac{2}{3}$ and $u = 2\frac{1}{3}$. The constrains on u and v imply that the solution to the system is to some extent determined.

We can use these ideas to develop a general solution strategy for systems of equations with a singular coefficient matrix. Consider again the equation system from the beginning of this section. If we treat the third equation as redundant this system can be represented in terms of the first two equations:

$$\begin{aligned} -u + v + w &= 0 \\ 2u + v - w &= 1 \end{aligned}$$

Since this system has one more parameter than it has equations, we need to eliminate one of the parameters. If we parameterize $w = \theta$, then the equation system can be written as

$$\begin{aligned} -u + v + \theta &= 0 \\ 2u + v - \theta &= 1 \end{aligned}$$

or equivalently

$$\begin{aligned} -u + v &= -\theta \\ 2u + v &= 1 + \theta \end{aligned}$$

The matrix representation of this system is

$$\begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} -\theta \\ 1+\theta \end{bmatrix}$$

The coefficient matrix in this system is regular (it has a determinant of -3). Hence, it is possible to apply the inverse solution:

$$\begin{aligned} \begin{bmatrix} u \\ v \end{bmatrix} &= \begin{bmatrix} -1 & 1 \\ 2 & 1 \end{bmatrix}^{-1} \begin{bmatrix} -\theta \\ 1+\theta \end{bmatrix} \\ &= \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} \\ \frac{2}{3} & \frac{1}{3} \end{bmatrix} \begin{bmatrix} -\theta \\ 1+\theta \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{3}+\frac{2}{3}\theta \\ \frac{1}{3}-\frac{1}{3}\theta \end{bmatrix}, \end{aligned}$$

which is what we obtained before.[2]

One particular solution to this system is obtained when we set $\theta = 0$. In this case,

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \end{bmatrix}$$

This solution can be represented in terms of the generalized inverse:

$$\mathbf{b} = \mathbf{X}^{-}\mathbf{y},$$

where

$$\mathbf{X}^{-} = \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

The solution is then

$$\begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \end{bmatrix},$$

which is the solution we would have obtained by setting $w = \theta = 0$, $v = \frac{1}{3} - \frac{1}{3}\theta = \frac{1}{3}$, and $u = \frac{1}{3} + \frac{2}{3}\theta = \frac{1}{3}$.

[2]It does not matter which equation is treated as redundant. You can verify this, for example, by treating the 1st equation as redundant.

Of course, setting $w = \theta = 0$ is arbitrary. It can be demonstrated that the general solution to a systems of equations in terms of the generalized inverse is:

$$\mathbf{b} = \mathbf{X}^{-}\mathbf{y} + (\mathbf{I} - \mathbf{X}^{-}\mathbf{X})\mathbf{g},$$

where $\mathbf{g}$ is an arbitrary vector. This formula is widely used to find solutions to systems of equations in which the coefficient matrix is singular.

Example 4.6. For instance, setting

$$\mathbf{g} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix},$$

the general solution to the system of equations above is given by:

$$\begin{aligned}
\mathbf{b} &= \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + \left(\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \right. \\
&\quad \left. \begin{bmatrix} -\frac{1}{3} & \frac{1}{3} & 0 \\ \frac{2}{3} & \frac{1}{3} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} -1 & 1 & 1 \\ 2 & 1 & -1 \\ 1 & 3 & 2 \end{bmatrix} \right) \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & \frac{2}{3} \\ 0 & 0 & -\frac{1}{3} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{3} + \frac{2}{3}\theta_3 \\ \frac{1}{3} - \frac{1}{3}\theta_3 \\ \theta_3 \end{bmatrix}
\end{aligned}$$

where θ_3 can be any arbitrary number.

4.4 Homogeneous Equations

One variation of systems of equations deserves additional attention, namely, homogeneous equations. A system of homogeneous equations is given by:

$$\mathbf{Xb} = \mathbf{0}$$

This means that the outcomes of all equations in the system are zeroes.

Example 4.7. For instance,

$$\begin{aligned} u + v &= 0 \\ 2u - v &= 0 \end{aligned}$$

is a system of homogeneous equations that can be represented through

$$\begin{bmatrix} 1 & 1 \\ 2 & -1 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Systems of homogeneous equations always have at least one solution. If we set all of the parameters equal to 0, then the equations in the system will all hold true. For example, by setting $u = v = 0$, we obtain a valid solution for the system above. Of course, this solution is not very interesting because it is automatically true. Hence, we call this the **trivial** or **null solution** to the system.

An important question is if there are also **non-trivial** or **non-null solutions** to a system of homogeneous equations. A key result in linear algebra is that such non-null solutions can be found only when the rank of $\mathbf{X}$ is less than the number of columns in $\mathbf{X}$: $r < n$, where n is the number of columns. An equivalent statement is that the rank of $\mathbf{X}$ should be less than the number of parameters. Alternatively, we can state that the prerequisite for finding non-null solutions is that $\det(\mathbf{X}) = 0$—a condition that is met when a matrix is not full rank. A couple examples will illustrate this point.

Example 4.8. First, consider the homogeneous set of equations above. The rank of this system is 2, the same as the number of columns in $\mathbf{X}$. The augmented matrix is

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 0 \\ 2 & -1 & 0 \end{bmatrix}$$

Gauss-Jordan elimination results in the following transformation of the augmented matrix:

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

(you should verify this result). It then follows that there is only one solution: $u = v = 0$—the null solution.

Example 4.9. Next consider the equation system given by

$$\begin{aligned} u + v + w &= 0 \\ u - v - 2w &= 0 \\ 3u - v - 3w &= 0, \end{aligned}$$

which can be represented by

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -2 \\ 3 & -1 & -3 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

The coefficient matrix, $\mathbf{X}$, is not full rank: the third row is equal to the sum of the first row and 2 times the second row. Thus, the rank is 2 and is less than the number of columns in $\mathbf{X}$. We can use the generalize inverse to obtain the solution to this system. Given

$$\mathbf{X}^{-} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 \\ \frac{1}{2} & -\frac{1}{2} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

the solution $\mathbf{b} = \mathbf{X}^{-}\mathbf{y} + \left(\mathbf{I} - \mathbf{X}^{-}\mathbf{X}\right)\theta$ yields

$$\mathbf{b} = \begin{bmatrix} \frac{1}{2}\theta_3 \\ -\frac{3}{2}\theta_3 \\ \theta_3 \end{bmatrix}$$

Thus, there are multiple solutions to the system of equations, including non-null solutions. For example, if $w = \theta_3 = 1$, then $u = .5$ and $v = -1.5$. If $w = \theta_3 = 4$, then $u = 2$ and $v = -6$, etcetera.

4.5 Bilinear and Quadratic Forms

So far, we have discussed systems of linear equations. However, equations can also be non-linear in form. How would one represent such equations in terms of matrices?

First, consider the equation

$$y = a_{11}x_1^2 + (a_{12} + a_{21})\, x_1 x_2 + a_{22}x_2^2$$

This is a quadratic equation. It can be represented in matrix form by defining a vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

and a matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

The matrix representation of the quadratic equation is now:

$$y = \mathbf{x}^\top \mathbf{A} \mathbf{x}$$

This is known as the **quadratic form**. In general, the quadratic form requires the specification of an $m \times 1$ vector, $\mathbf{x}$, of variables and an $m \times m$ matrix, $\mathbf{A}$, of coefficients.

Next, consider an equation that mixes two kinds of variables, namely x and z. An example of such an equation is

$$\begin{aligned} y = {} & a_{11} z_1 x_1 + a_{12} z_2 x_1 + a_{13} z_3 x_1 + \\ & a_{21} z_1 x_2 + a_{22} z_2 x_2 + a_{23} z_3 x_2. \end{aligned}$$

To represent this system, we define a vector

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix},$$

another vector

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix},$$

and a matrix

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

The matrix representation is then

$$y = \mathbf{x}^\top \mathbf{A} \mathbf{z}$$

This is known as the **bilinear form**. In general, bilinear forms require the specification of an $m \times 1$ vector of variables, $\mathbf{x}$, an $n \times 1$ vector of variables, $\mathbf{z}$, and a $m \times n$ coefficient matrix, $\mathbf{A}$. It is easily verified that the quadratic form is a special case of the bilinear form that arises when $\mathbf{z} = \mathbf{x}$. In this case, $\mathbf{A}$ is a square matrix.

4.6 The Linear Regression Model

One important application of systems of equations in statistics is the linear regression model. This model is conventionally written as

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_P x_{iP} + \varepsilon_i$$

where i refers to a particular sample unit. Given that we have n such units in our sample, the regression model actually represents a system of n linear equations:

$$\begin{aligned} y_1 &= \beta_0 + \beta_1 x_{11} + \cdots + \beta_P x_{1P} + \varepsilon_1 \\ y_2 &= \beta_0 + \beta_1 x_{21} + \cdots + \beta_P x_{2P} + \varepsilon_2 \\ \vdots &\quad \vdots \\ y_n &= \beta_0 + \beta_1 x_{n1} + \cdots + \beta_P x_{nP} + \varepsilon_n \end{aligned}$$

This is similar to the systems we have considered so far. The key difference is that we have an additional term on the right-hand side, in the form of ε_i.

To render the regression model in matrix form, we start by writing the system of equations slightly differently:

$$\begin{aligned} y_1 &= \beta_0 \cdot 1 + \beta_1 x_{11} + \cdots + \beta_P x_{1P} + \varepsilon_1 \\ y_2 &= \beta_0 \cdot 1 + \beta_1 x_{21} + \cdots + \beta_P x_{2P} + \varepsilon_2 \\ \vdots &\quad \vdots \\ y_n &= \beta_0 \cdot 1 + \beta_1 x_{n1} + \cdots + \beta_P x_{nP} + \varepsilon_n \end{aligned}$$

We have multiplied the constant by 1 so that it is followed by a term just like the other regression coefficients in the system. We now simply put similar elements into different containers. Starting with the response variable, we collect all of the responses in a $n \times 1$ column vector:

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Similarly, we collect the error terms in

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_n \end{bmatrix}$$

Next, we collect the regression coefficients in a $(P+1) \times 1$ column vector:

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_P \end{bmatrix}$$

We conclude by collecting the information about the predictors (including the constant) in a $n \times (P+1)$ matrix:

$$\mathbf{X} = \begin{bmatrix} 1 & x_{11} & \cdots & x_{1P} \\ 1 & x_{21} & \cdots & x_{2P} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & \cdots & x_{nP} \end{bmatrix}$$

We clearly see that the first column captures the 1s that we inserted earlier. This column captures the constant. The regression madel can now be expressed as

$$\mathbf{y} \;=\; \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

(You should verify that this recovers the system of equations shown earlier.)

Chapter 5

Matrix Differentiation

It is frequently necessary to find the maximum or minimum of a function. In ordinary calculus, we accomplish this through differentiation. This is not different when the function is represented through matrices. However, the differentiation of matrices is a bit more complex than ordinary differentiation. Hence, it is useful to review some rules of matrix differentiation.

5.1 Scalar-Vector Differentiation

Consider the following equation in three variables:

$$y = a_1x_1 + a_2x_2 + a_3x_3$$

From calculus, we know that the partial derivatives of y with respect to the variables are given by:

$$\begin{aligned} \frac{\partial y}{\partial x_1} &= a_1 \\ \frac{\partial y}{\partial x_2} &= a_2 \\ \frac{\partial y}{\partial x_3} &= a_3 \end{aligned}$$

We could also have presented this equation in matrix terms. Let

$$\mathbf{a} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

and

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Then

$$y = \mathbf{a}^\top \mathbf{x}$$

We can now take the derivative of the scalar y with respect to the elements of the vector $\mathbf{x}$. In light of the earlier results from calculus, this should give us a vector of partial derivatives, containing elements a_1, a_2, and a_3. That is,

$$\frac{\partial y}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y}{\partial x_1} \\ \frac{\partial y}{\partial x_2} \\ \frac{\partial y}{\partial x_3} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \mathbf{a}$$

Thus, we have condensed the three partial derivatives into a vector.

This example reflects an important general result about scalar-vector differentiation. Given $y = \mathbf{a}^\top \mathbf{x}$,

$$\frac{\partial y}{\partial \mathbf{x}} = \mathbf{a}$$

Note that we differentiate with respect to a column vector, so that the result also is a column vector. Had we differentiated with respect to a row vector, then the result would have been a row vector. The general result explains how we should address the differentiation of dot products.

A related result concerns a situation where a vector is a function of a scalar. For instance, let

$$y_1 = a_1 x$$
$$y_2 = a_2 x$$

This system can be represented as $\mathbf{y} = x\mathbf{a}$, where x is a scalar, $\mathbf{y}^\top = [y_1 \; y_2]$, and $\mathbf{a}^\top = [a_1 \; a_2]$. In this case,

$$\frac{\partial \mathbf{y}}{\partial x} = \begin{bmatrix} \frac{\partial y_1}{\partial x} \\ \frac{\partial y_2}{\partial x} \end{bmatrix} = \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \mathbf{a}$$

In general, given a $m \times 1$ vector y, a scalar x, and a $m \times 1$ coefficient vector $\mathbf{a}$, we have

$$\frac{\partial \mathbf{y}}{\partial x} = \mathbf{a}$$

5.2 Vector-Vector Differentiation

Consider a system of equations in which two different outcomes are predicted by three variables:

$$y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3$$
$$y_2 = a_{21}x_1 + a_{22}x_2 + a_{23}x_3$$

We have six different partial derivatives:

$$\begin{array}{lll} \frac{\partial y_1}{\partial x_1} = a_{11} & \frac{\partial y_1}{\partial x_2} = a_{12} & \frac{\partial y_1}{\partial x_3} = a_{13} \\ \frac{\partial y_2}{\partial x_1} = a_{21} & \frac{\partial y_2}{\partial x_2} = a_{22} & \frac{\partial y_2}{\partial x_3} = a_{23} \end{array}$$

In matrix form, the system of equations may be written as $\mathbf{y} = \mathbf{A}\mathbf{x}$, where

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix},$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix},$$

and

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

The partial derivatives are given by

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \frac{\partial y_1}{\partial x_3} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \frac{\partial y_2}{\partial x_3} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = \mathbf{A}$$

In general, let $\mathbf{y}$ be a $m \times 1$ vector and let $\mathbf{x}$ be a $n \times 1$ vector. Further, let $\mathbf{A}$ be a $n \times m$ vector such that $\mathbf{y} = \mathbf{A}\mathbf{x}$. Then

$$\frac{\partial \mathbf{y}}{\partial \mathbf{x}} = \mathbf{A}$$

with

$$a_{ij} = \frac{\partial y_i}{\partial x_j}$$

5.3 Differentiating the Quadratic and Bilinear Forms

Consider the quadratic form $y = \mathbf{x}^\top \mathbf{A}\mathbf{x}$. To find the partial derivatives of y with respect to the elements in $\mathbf{x}$, we can use the principle of differentiation by parts.[1] Let $\mathbf{u} = \mathbf{A}\mathbf{x}$ be a $m \times 1$ vector and let $\mathbf{v} = \mathbf{x}^\top \mathbf{A}$ be a $1 \times m$ vector. Then y can be expressed in two different ways: (1) $y = \mathbf{x}^\top \mathbf{u}$ and (2) $y = \mathbf{v}\mathbf{x}$. Both expressions fall in the category of scalar-vector differentiation and hence we know that

$$\begin{aligned} \frac{\partial \mathbf{x}^\top \mathbf{u}}{\partial \mathbf{x}} &= \mathbf{u} \\ \frac{\partial \mathbf{v}\mathbf{x}}{\partial \mathbf{x}} &= \mathbf{v}^\top \end{aligned}$$

(We have taken the transpose of $\mathbf{v}$ because this is a row vector and the result we are looking for is a column vector.) We now simply add the results, so that

$$\frac{\partial y}{\partial \mathbf{x}} = \mathbf{u} + \mathbf{v}^\top = \mathbf{A}\mathbf{x} + \mathbf{A}^\top \mathbf{x} = \left(\mathbf{A} + \mathbf{A}^\top\right)\mathbf{x}$$

A special case of the result arises when $\mathbf{A}$ is symmetric. In this case, $\mathbf{A}^\top = \mathbf{A}$ and we get $\partial y / \partial \mathbf{x} = 2\mathbf{A}\mathbf{x}$.

For blinear forms, we have $y = \mathbf{x}^\top \mathbf{A}\mathbf{z}$. In this model, there are two sets of variables, contained in the vectors $\mathbf{x}$ and $\mathbf{z}$, respectively. When we differentiate y with respect to $\mathbf{x}$, then $\mathbf{z}$ is held constant. Thus, we can write $\mathbf{A}\mathbf{z} = \mathbf{k}_1$, which is a $m \times 1$ vector. Consequently, we can express the bilinear form as $y = \mathbf{x}^\top \mathbf{k}_1$. Using the theorem on vector-vector differentiation, we know that

$$\frac{\partial y}{\partial \mathbf{x}} = \mathbf{k}_1 = \mathbf{A}\mathbf{z}$$

By the same logic, we can write the bilinear form as $y = \mathbf{k}_2 \mathbf{z}$, where $\mathbf{k}_2 = \mathbf{x}^\top \mathbf{A}$ is a $1 \times n$ vector. We now have

$$\frac{\partial y}{\partial \mathbf{z}} = \mathbf{k}_2^\top = \mathbf{A}^\top \mathbf{x}$$

For a symmetric $\mathbf{A}$, the last expression may also be written as $\mathbf{A}\mathbf{x}$.

[1] To refresh memory on this topic consider the following example. Let $y = 3x^2$. Clearly, the derivative of y with respect to x is $6x$. We could have found this by defining a new variable $z = 3x$. In this case $y = zx = xz$. Taking the derivative for zx with respect to x we get z. Taking the derivative of xz with respect to x we get z again. Now the derivative of y with respect to x is equal to the sum of the derivatives for zx and xz. Thus, $\frac{dy}{dx} = z + z = 3x + 3x = 6x$.

5.4 Ordinary Least Squares

In the previous chapter, we saw that the linear regression model may be written as $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$. The utility of this model hinges on our ability to obtain values for the parameters in $\boldsymbol{\beta}$. A widely used approach here is ordinary least squares, which consists of minimizing $S = \sum_{i=1}^{n} \varepsilon_i^2$ with respect to $\boldsymbol{\beta}$.

In matrix terms, we now know that the least squares fit criterion can be written as

$$S = \boldsymbol{\varepsilon}^\top \boldsymbol{\varepsilon}$$

Since, $\boldsymbol{\varepsilon} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$, we can also write the criterion as

$$S = (\mathbf{y} - \mathbf{X}\boldsymbol{\beta})^\top (\mathbf{y} - \mathbf{X}\boldsymbol{\beta}) = \mathbf{y}^\top \mathbf{y} - 2\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{y} + \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$$

Minimization requires that the first order conditions are satisfied. Thus, we take the partial derivative of S with respect to $\boldsymbol{\beta}$ and set the resulting equations equal to 0. For the derivatives, we have

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = \frac{\partial \mathbf{y}^\top \mathbf{y}}{\partial \boldsymbol{\beta}} - 2\frac{\boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{y}}{\partial \boldsymbol{\beta}} + \frac{\partial \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}}{\partial \boldsymbol{\beta}}$$

To facilitate differentiation, we write the elements a bit differently:

1. Let $S_1 = \mathbf{y}^\top \mathbf{y}$. Then the first element on the right-hand side is

$$\frac{\partial S_1}{\partial \boldsymbol{\beta}} = \frac{\partial \mathbf{y}^\top \mathbf{y}}{\partial \boldsymbol{\beta}} = \mathbf{0}$$

   The derivative produces a set of zeros because $\mathbf{y}^\top \mathbf{y}$ is not a function of $\boldsymbol{\beta}$ and as such can be treated as a constant.

2. Let $S_2 = \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{y} = \boldsymbol{\beta}^\top \mathbf{a}$, where $\mathbf{a} = \mathbf{X}^\top \mathbf{y}$ is a $(P+1) \times 1$ vector. Then the rules of scalar-vector differentiation dictate

$$\frac{\partial S_2}{\partial \boldsymbol{\beta}} = \frac{\partial \boldsymbol{\beta}^\top \mathbf{a}}{\partial \boldsymbol{\beta}} = \mathbf{a} = \mathbf{X}^\top \mathbf{y}$$

3. Let $S_3 = \boldsymbol{\beta}^\top \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}^\top \mathbf{C}\boldsymbol{\beta}$, where $\mathbf{C}$ is the $(P+1) \times (P+1)$ matrix of cross-products and sums of squares. This matrix is symmetric. Using the results of matrix-vector differentiation, we have

$$\frac{\partial S_3}{\partial \boldsymbol{\beta}} = \frac{\partial \boldsymbol{\beta}^\top \mathbf{C}\boldsymbol{\beta}}{\partial \boldsymbol{\beta}} = 2\mathbf{C}\boldsymbol{\beta} = 2\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$$

Thus, we have

$$\frac{\partial S}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta}$$

We now set the derivative equal to zero: $-2\mathbf{X}^\top \mathbf{y} + 2\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} = \mathbf{0}$. Rearranging terms, we obtain the **normal equations**:

$$\mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top \mathbf{y}$$

As long as the number of rows in $\mathbf{X}$ is at least as large as the number of columns and there are no linear dependencies among the columns—i.e., the data do not suffere from micro-numerosity and perfect multicollinearity—then $\mathbf{X}^\top \mathbf{X}$ is regular and can be inverted. Thus, using the approach shown in Chapter 4,

$$\begin{aligned} \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{X}\boldsymbol{\beta} &= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y} \\ \hat{\boldsymbol{\beta}} &= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y} \end{aligned}$$

This is the ordinary least squares estimator.

Chapter 6

Vector Geometry

Aside from computational benefits, linear algebra also allows us to obtain useful geometric representations of data. Deriving those representations is the subject of vector geometry. As we shall see, many statistical concepts have geometric representations that we can find through vector geometry and that shed light on patterns in our data.

6.1 Representing Vectors in Space

By now, it should be clear that any matrix can be construed as a collection of row or column vectors. We can represent those vectors geometrically in a space. Doing so requires that we define the **basis** of the space. By basis, we mean a set of **LIN vectors**, i.e., vectors that are linearly independent. Those vectors are perpendicular to one another and they form the axis structure of the space. For any matrix, there are two ways in which the basis may be defined:

1. If we think of the matrix as a set of row vectors, then the columns of the matrix form the basis.

2. If we think of the matrix as a set of column vectors, then the rows form the basis.

The decision on how to interpret the matrix determines the resulting space. If an $m \times n$ matrix is construed in terms of m row vectors, then the space is n-dimensional. We write $\mathbb{R}^n$. By contrast, if the matrix is construed in terms of n column vectors, then the resulting space is $\mathbb{R}^m$.

Figure 6.1: Units Represented in a 2-Space

calc,3d,arrows

Example 6.1. For instance, consider the 3×2 matrix

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

We assume this is a data matrix with rows formed by the units and columns formed by the variables. As a matter of notation, we label the rows by $\mathbf{o}_i^\top$ (for $i = 1, 2, 3$) and the columns by $\mathbf{x}_j$ (for $j = 1, 2$). Thus, one way to think of $\mathbf{A}$ is as a set of three row vectors. When we portray those vectors, we need a 2-dimensional space or a **2-space**, because each row contains two values. Those values are Cartesian coordinates that determine the end point of a vector, as can be seen from Figure 7.1. The starting point is always at the origin. For example, the vector corresponding to $\mathbf{o}_1$ ends at the coordinates (2, 3), since the unit scores 2 on the first variable/horizontal axis and 3 on the second variable/vertical axis. The vectors thus capture how units score on the variables that we measured.

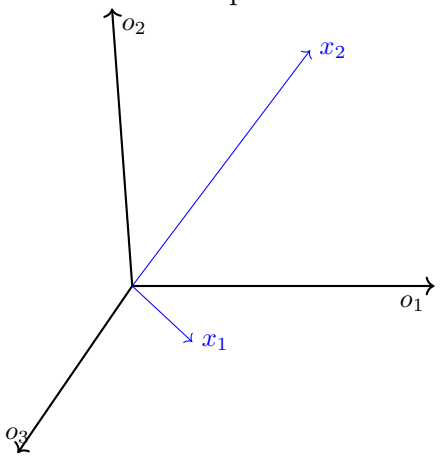
Example 6.2. If we think of $\mathbf{A}$ from 6.1 as a set of two column vectors, then we represent them in a 3-space. The principle is again the same. With two columns, we can draw two vectors. Those vectors originate at (0,0,0). They end at locations determined by the values across the units, as is shown in Figure 7.2. The resulting image shows how each variable behaves across the units.

As is clear from both examples, vectors have both a magnitude and a direction. These can be exploited to understand a variety of statistical concepts, as we shall see.

6.2 Attributes of Vector Spaces

When we depict a matrix through vectors in a space, several attributes of this representation become interesting. First, how long are the vectors? Second, what is the angle between those vectors? Finally, what is the area or volume of the space spanned by the vectors?

Figure 6.2: Variables Represented in a 3-Space



6.2.1 Vector Norms

How long is a vector of data? That is, what is the vector's magnitude? To answer this question I will first consider the simple case of a 2-space. Consider a vector

$$\mathbf{u} = \begin{bmatrix} x \\ y \end{bmatrix}$$

where x gives the coordinate on the x-axis and y gives the coordinate on the y-axis. According to Pythagoras' theorem, the length of this vector is given by $\sqrt{x^2 + y^2}$. In linear algebra, this can be expressed as $\sqrt{\mathbf{u}^\top \mathbf{u}}$. We call this the L_2 norm of the vector. It is often written as $\|\mathbf{u}\|_2$. Later, we shall consider some other norms.

The result generalizes easily to a K-space. Let $\mathbf{u}$ denote a $K \times 1$ vector. The L_2 norm of the vector is given by

$$\|\mathbf{u}\|_2 \quad = \quad \sqrt{\mathbf{u}^\top \boldsymbol{u}} = \sqrt{\sum_{k=1}^{K} u_k^2}$$

This norm gives the Euclidean or shortest line distance from the origin of the space to the coordinates that demarcate the endpoint of the vector.

Example 6.3. Let us return to

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

Now consider the first column of this matrix. The L_2 norm is

$$\|\mathbf{x}_1\|_2 = \sqrt{2^2 + 1^2 + 3^2} = \sqrt{14}$$

Similarly, the L_2 norm for the second column in $\mathbf{A}$ is $\sqrt{26}$ (you should verify this).

In certain applications, it may be useful to give all vectors the same length of 1. In this case, we say that the vectors have been **normalized**. Normalization involves the following transformation

$$\mathbf{v} = \frac{\mathbf{u}}{\|\mathbf{u}\|_2}$$

Example 6.4. For example, if we divide all of the elements in $\mathbf{x}_1$ from Example 6.3 by $\sqrt{14}$, then we obtain a unit L_2 norm:

$$\sqrt{\left(\frac{2}{\sqrt{14}}\right)^2 + \left(\frac{1}{\sqrt{14}}\right)^2 + \left(\frac{3}{\sqrt{14}}\right)^2} = \sqrt{\frac{1}{14}\left(2^2 + 1^2 + 3^2\right)} = \sqrt{\frac{1}{14} \cdot 14} = 1$$

The L_2 norm is not the only norm that one can define. In machine learning, for example, we often use the L_1 norm:

$$\|\mathbf{u}\|_1 = \sum_{k=1}^{K} |x_k|$$

This is also known as the Manhattan or city block norm. In a 2-space, it measures the distance we traverse along the x- and y-axes, respectively, and then adds those two numbers. Most generally, we can define a p-norm:

$$\|\mathbf{u}\|_p = \left(\sum_{k=1}^{K} |x_k|^p\right)^{\frac{1}{p}}$$

where $p \geq 1$. When $p = \infty$, the norm $\|\mathbf{u}\|_\infty$ returns the largest absolute element of $\mathbf{u}$ (this is known as the Chebyshev norm).

Example 6.5. Consider again the first column in matrix $\mathbf{A}$ from Example 6.3. The Manhattan and Chebyshev norms are given by

$$\begin{aligned} L_1 &= |2| + |1| + |3| = 6 \\ L_\infty &= \max(|2|, |1|, |3|) = 3 \end{aligned}$$

6.2.2 Vector Angles

To find the angle between two vectors, we take advantage of the following formula:

$$\mathbf{x}^\top \mathbf{y} \quad = \quad \|\mathbf{x}^\top\|_2 \|\mathbf{y}^\top\|_2 \cos\alpha$$

where $\mathbf{x}$ and $\mathbf{y}$ are $K \times 1$ vectors and α is the angle between them in degrees. Taking the inverse cosine of the left-hand side divided between the product of the L_2 norms should thus yield the angle.

Example 6.6. Consider Figure 6.2. We could ask what the angle is between the two vectors portrayed in this figure. The inner-product between $\mathbf{x}_1$ and $\mathbf{x}_2$ equals 13. We already saw that the L_2 norm of the two vectors are $\sqrt{14}$ and $\sqrt{26}$, respectively. Thus,

$$13 = \sqrt{14} \cdot \sqrt{26} \cos\alpha$$

This means that $\cos\alpha = 0.681$ and $\alpha = 47°$.

From trigonometry, we know that $\cos 90 = 0$. Hence, we obtain perpendicular vectors only when their inner product $\mathbf{x}^\top \mathbf{y} = 0$. In this case, we say that the vectors are **orthogonal**, a property that we saw in principal component analysis. If the vectors also have unit lengths, then we say that they are **orthonormal**.

The vectors that we use to construct the vector space are orthonormal. We call them the **base vectors**. For instance, in a 2-space the base vector for the x-axis takes on the form of $(1, 0)$, while the base vector for the y-axis takes on the form of $(0, 1)$. Clearly, these vectors have unit norms and their inner-product is zero, meaning they are orthonormal.

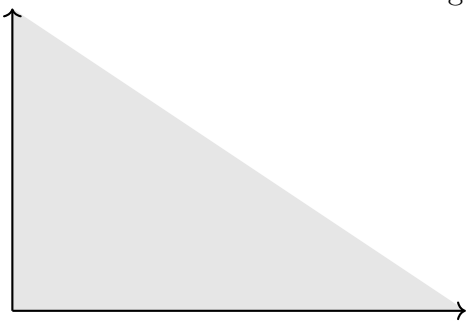
6.3 Areas and Volumes

Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}$$

The two rows in this natrix produce two vectors, as is shown in Figure 6.3. We clearly see that the vectors are orthogonal. We could now connect the end

Figure 6.3: The Area in Between Two Orthogonal Vectors



points of the two vectors to generate the triangle shown in Panel (b). The question is now what is the area of the triangle? From geometry, we know that it is equal to half times the base times the height. The base is equal to 6, which is also the L_2 norm of the vector that runs from west to east. The height is equal to 4, which is the L_2 norm of the vector that runs from south to north. Hence, the area is $\frac{1}{2} \cdot 6 \cdot 4 = 12$.

We observe something quite interesting. The area we just computed is exactly halve of the absolute value of the determinant of $\mathbf{A}$:

$$\text{Area} \quad = \quad \frac{1}{2} |\text{det}\mathbf{A}|$$

Thus, the absolute determinant captures the area between two vectors. We can also say, it captures the degree of separation between those vectors.

Example 6.7. Let us consider a different situation. Let

$$\mathbf{A} = \begin{bmatrix} 2 & 0 \\ 3 & 0 \end{bmatrix}$$

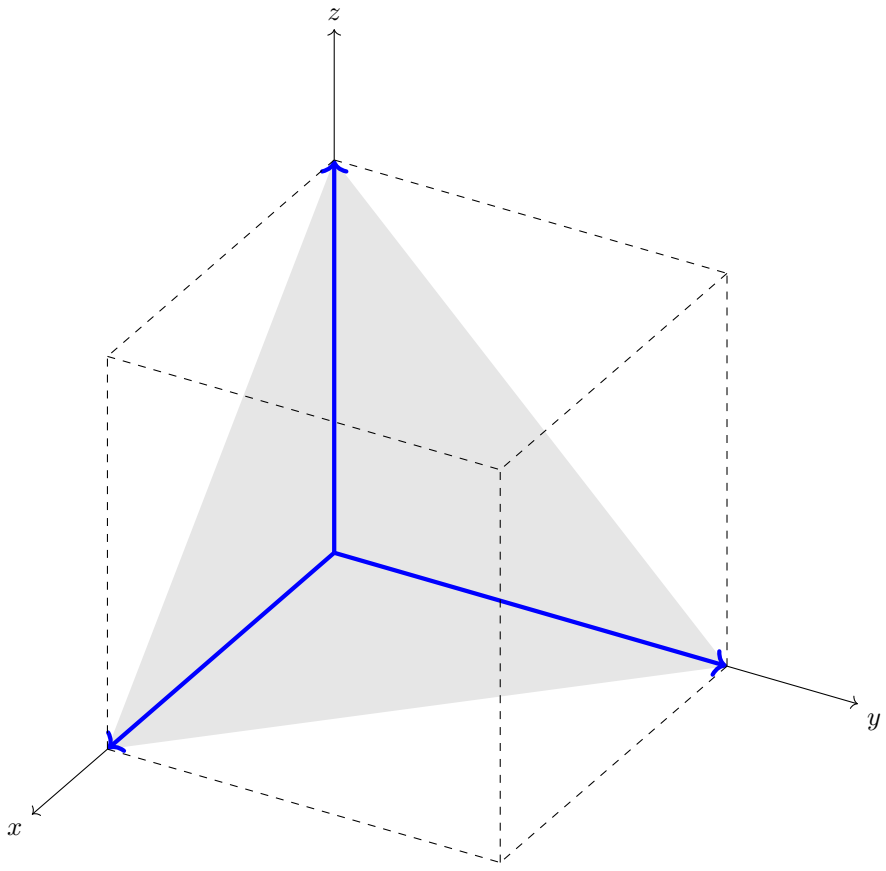
The row vectors are now placed on top of each other; there is no separation between them. We also see that $\frac{1}{2} |\text{det}\mathbf{A}| = 0$.

Example 6.8. Now consider

$$\mathbf{A} = \begin{bmatrix} 3 & 0 \\ 3 & 3 \end{bmatrix}$$

The vectors are at a 45 degree angle and thus the area between the vectors is again half times base times height, or 4.5. (You should verify this.) It is easily verified that $\frac{1}{2} |\text{det}\mathbf{A}| = 4.5$.

Figure 6.4: The Volume in Between Three Orthogonal Vectors



So far, we have looked at vectors in a 2-space. What happens when we generalize things to a 3- or higher dimensional space? In that case, we no longer speak of areas but of volumes. The principle remains the same, however.

Example 6.9. Consider, for example, the following matrix

$$\mathbf{A} = \begin{bmatrix} 4 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 4 \end{bmatrix}$$

Using the rows as the basis, we obtain Figure 6.4. The volume of the space spanned by the vectors is shown in gray. This is half of a cube with volume of 64. Thus, the volume is 32, which is also half of the absolute value of the determinant of $\mathbf{A}$.

6.4 Statistical Applications

Vector geometry can be linked to statistical applications, which often produces interesting insights. In the ensuing discussion, I assume that vectors have been transformed to mean-deviation scores.

6.4.1 Vector Norms and Variation

Consider the matrix of deviation scores that we derived in Example 3.15:

$$\mathbf{D} = \begin{bmatrix} -30.67 & -14.33 \\ 9.33 & -9.33 \\ 21.33 & 23.67 \end{bmatrix}$$

Using the rows as the basis, we obtain two vectors, $\mathbf{d}_1^\top = (-30.67, 9.33, 21.33)$ and $\mathbf{d}_2^\top = (-14.33, -9.33, 23.67)$. Define the variation as

$$S_i^2 = (n-1) \cdot s_i^2 = \mathbf{d}_i^\top \mathbf{d}_i$$

where n is the sample size and s_i^2 is the variance in column i of $\mathbf{D}$. In our example, we have $S_1^2 = 1482.67$ and $S_2^2 = 852.67$. We can now easily see that these numbers are equal to the squared L_2 norms of the two column vectors:

$$S_i^2 \;=\; \|\mathbf{d}_i\|_2^2$$

Thus, the length of a vector captures the degree of variation. In practice, of course, we often work with variance instead of variation. This involves a simple transformation:

$$s_i^2 \;=\; \frac{\|\mathbf{d}_i\|_2^2}{n-1}$$

It also follows that the standard deviation, which is the square root of s_i^2, is equal to

$$s_i \;=\; \frac{\|\mathbf{d}_i\|_2}{\sqrt{n-1}}$$

6.4.2 Vector Angles and Correlation

In Chapter 3, we encountered the concept of covariance: $s_{ij} = (n-1)^{-1}\mathbf{d}_i^\top \mathbf{d}_j$. In many applications, the covariance is transformed into the Pearson product moment correlation:

$$r_{ij} \quad = \quad \frac{s_{ij}}{s_i \cdot s_j} = \frac{\mathbf{d}_i^\top \mathbf{d}_j}{\|\mathbf{d}_i\|_2 \cdot \|\mathbf{d}_j\|_2}$$

(the terms in $n-1$ cancel between the numerator and the denominator.) This coefficient is bounded between -1 and 1, with -1 indicating perfect negative linear association and 1 indicating perfect positive linear association. If $r_{ij} = 0$, then there is no linear association between columns i and j in $\mathbf{D}$.

Let us now look back at the main result for vector angles. Rearranging terms, it is easy to demonstrate that

$$r_{ij} \quad = \quad \cos\alpha$$

Thus the correlation between columns i and j in $\mathbf{D}$ contains information about the vector angle. Specifically,

$$\alpha \quad = \quad \arccos r_{ij}$$

Example 6.10. Using the data from Example 3.15, the covariance between the two columns of $\mathbf{D}$ is given by

$$\frac{1}{3-1} \cdot \begin{bmatrix} -30.67 & 9.33 & 21.33 \end{bmatrix} \begin{bmatrix} -14.33 \\ -9.33 \\ 23.67 \end{bmatrix} = 428.667$$

We know that $s_1^2 = 741.333$ and $s_2^2 = 426.333$. Hence, $r_{ij} = 428.667/(sqrt741.333 \cdot \sqrt{426.333}) = 0.762$. The vector angle is correspondingly 40.3°.

6.4.3 Ordinary Least Squares Revisited

In Chapter 5, we saw that the least squares estimator involves minimizing $\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}$ with respect to $\boldsymbol{\beta}$. Since $\mathbb{E}[\boldsymbol{\epsilon}] = \mathbf{0}$ by assumption, $\boldsymbol{\epsilon}^\top \boldsymbol{\epsilon}$ is the squared L_2 norm of the regression errors. Thus, we can think of least squares estimation as minimizing the (squared) L_2 norm.

Chapter 7

Eigenvalues and Eigenvectors

Eigenvalues and eigenvectors play an important role in data reduction techniques such as principal components analysis. It is therefore useful to discuss these concepts in some detail. You should keep in mind that eigenvalues and eigenvectors are defined for square matrices only.

7.1 Definition

To understand the concepts of eigenvalues and eigenvectors, we should consider the operation of multiplying a vector by a square matrix. Let $\mathbf{c}$ and vA be the vector and matrix, respectively. The product $\mathbf{Ac}$ produces another vector of the same dimensionality as $\mathbf{c}$. Under some circumstances,

$$\mathbf{Ac} \quad = \quad \lambda \mathbf{c},$$

where λ is a scalar. When this happens, we say that λ is the **eigenvalue** and $\mathbf{c}$ is the **eigenvector** that is associated with $\mathbf{A}$.[1]

Example 7.1. Consider the matrix

$$\mathbf{A} = \begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix}$$

[1] Other terms for eigenvector are characteristic vector, proper vector, and latent vector. Other terms for eignevalues are characteristic root, proper root, and latent root.

and the vector

$$\mathbf{c} = \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

It is easy to show that

$$\mathbf{Ac} = \begin{bmatrix} 2 \\ -1 \end{bmatrix} = \mathbf{c}$$

This means we have an eigenvalue of $\lambda = 1$ and that $\mathbf{c}^\top = [2 \ -1]$ is an eigenvector of $\mathbf{A}$. Next consider

$$\mathbf{c} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

Now

$$\mathbf{Ac} = \begin{bmatrix} 10 \\ 5 \end{bmatrix} = 5\mathbf{c}$$

Hence, the eigenvalue is $\lambda = 5$. Further, $\mathbf{c} = [2 \ 1]$ is another eigenvector of $\mathbf{A}$.

7.2 Finding Eigenvalues and Eigenvectors

7.2.1 Eigenvalues

To determine the eigenvalues of a matrix, we rearrange $\mathbf{Ac} = \lambda\mathbf{c}$ into the homgeneous system of equations

$$(\mathbf{A} - \lambda\mathbf{I})\,\mathbf{c} \ = \ \mathbf{0}$$

From the discussion in Chapter 4, we know this system yields non-trivial solutions when $\mathbf{A} - \lambda\mathbf{I}$ is singular, i.e., when

$$\det(\mathbf{A} - \lambda\mathbf{I}) \ = \ 0$$

We can leverage this result to find the eigenvalues λ.

Example 7.2. Consider the matrix $\mathbf{A}$ from Example @ref{exm:eigen1}. We are interested in

$$\det\left(\begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix}\right) = \det\begin{bmatrix} 3-\lambda & 4 \\ 1 & 3-\lambda \end{bmatrix} = 0$$

From the discussion of determinants of 2×2 matrices, we know that this is equivalent to

$$(3-\lambda)^2 - 4 \;=\; \lambda^2 - 6\lambda + 5 = (\lambda - 1)(\lambda - 5) = 0$$

It follows immediately that $\lambda = 1$ and $\lambda = 5$ are the eigenvalues.

7.2.2 Eigenvectors

Once we have found the eigenvalues of a matrix, we can find the eigenvectors. The first step in this process is to substitute a particular solution for the eigenvalues into $\mathbf{Ac} = \lambda\mathbf{c}$. Returning to the earlier example, with $\lambda = 1$ we have

$$\begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

This equals the system

$$\begin{aligned} 3c_1 + 4c_2 &= c_1 \\ c1 + 3c_2 &= c_2 \end{aligned}$$

or

$$\begin{aligned} 2c_1 + 4c_2 &= 0 \\ c1 + 2c_2 &= 0 \end{aligned}$$

The system of equations clearly shows that the 2nd row is a linear function of the 1st row. We thus need to use the generalized inverse to find the eigenvectors. Earlier, we saw that solutions for irregular coefficient matrices are given by $\mathbf{X}^{-}\mathbf{y} + (\mathbf{I} - \mathbf{X}^{-}\mathbf{X})\mathbf{g}$. Since we have a homogeneous system, $\mathbf{y} = \mathbf{0}$ so that the eigenvectors are equal to $(\mathbf{I} - \mathbf{X}^{-}\mathbf{X})\mathbf{g}$. In the case of $\lambda = 1$,

$$\mathbf{X} = \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix},$$

resulting in

$$\mathbf{X}^{-} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}$$

The general solution for the eigenvector is thus

$$\mathbf{c} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 2 & 4 \\ 1 & 2 \end{bmatrix} \right) \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} -2\theta_2 \\ \theta_2 \end{bmatrix}$$

The solution we just derived can be applied to any value of θ_2. To limit the solution, we apply the following **normalizing** constraint:

$$\mathbf{x}^\top \mathbf{c} = 1$$

(We say that $\mathbf{c}$ has a unit norm.) Applying the logic to our result:

$$\begin{bmatrix} -2\theta_2 & \theta_2 \end{bmatrix} \begin{bmatrix} -2\theta_2 \\ \theta_2 \end{bmatrix} = 1$$

Thus, $5\theta_2^2 = 1$ and $\theta_2 = \pm 1/\sqrt{5}$. Retaining the positive value and substituting it into $\mathbf{c}$ we obtain the eigenvector. Thus, in numeric terms $\mathbf{c}^\top = [-0.894\ 0.447]$.

We can summarize what we have done in the following steps:

1. For each eigenvalue, construct the system of equations $(\mathbf{A} - \lambda\mathbf{I})\mathbf{c} = \mathbf{0}$.
2. Let $\mathbf{X} = \mathbf{A} - \lambda\mathbf{I}$. Find the generic eigenvector using $\mathbf{c} = (\mathbf{I} - \mathbf{X}^-\mathbf{X})\mathbf{g}$.
3. Narrow the solution down using $\mathbf{c}^\top\mathbf{c} = 1$.

Example 7.3. Let us apply those steps to find the eigenvector corresponding to $\lambda = 5$. First,

$$\left(\begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} \right) \begin{bmatrix} c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Next, define

$$\mathbf{X} = \begin{bmatrix} 3 & 4 \\ 1 & 3 \end{bmatrix} - \begin{bmatrix} 5 & 0 \\ 0 & 5 \end{bmatrix} = \begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix}$$

It is easily verified that

$$\mathbf{X}^- = \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix}$$

As per the second step,

$$\mathbf{c} = \left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} -\frac{1}{2} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} -2 & 4 \\ 1 & -2 \end{bmatrix} \right) \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 2\theta_2 \\ \theta_2 \end{bmatrix}$$

As per the third step, $5\theta_2^2 = 1$ or $\theta_2 = \pm 1/\sqrt{5}$. Retaining the positive value, we obtain $\mathbf{c}^\top = [0.894\ 0.447]$.

7.2.3 Combining Results

It is customary to combine the results. The eigenvalues are typically placed in a diagonal matrix, with the diagonal sorted by size. The eigenvectors are combined into a single matrix.

Example 7.4. In our example, the eigenvalue matrix takes the form of

$$\mathbf{L} = \begin{bmatrix} 5 & 0 \\ 0 & 1 \end{bmatrix}$$

The eigenvectors corresponding to these values can be combined into

$$\mathbf{C} = \begin{bmatrix} 0.894 & -0.894 \\ 0.447 & 0.447 \end{bmatrix}$$

The first column is the eigenvector corresponding to $\lambda = 5$, while the second column corresponds to $\lambda = 1$.

Using this notation, we have

$$\mathbf{AC} = \mathbf{CL}$$

This is the general representation of the eigenvectors and eigenvalues associated with $\mathbf{A}$.

7.3 Properties

Eigenvalues and eigenvectors have several important properties:

1. The number of non-null eigenvalues in a matrix is equal to the rank of that matrix.
2. If λ is an eigenvalue of $\mathbf{A}$, then λ^k is an eigenvalue of $\mathbf{A}^k$.
3. Let k be a scalar, then the eigenvalues for $k\mathbf{A}$ are k times the eigenvalues of $\mathbf{A}$.
4. The sum of the eigenvalues of $\mathbf{A}$ is equal to the trace of the matrix.
5. The product of the eigenvalues of $\mathbf{A}$ is equal to the determinant of the matrix.

6. For a symmetric matrix, the eigenvectors are independent. For example, let $\mathbf{c}_1$ and $\mathbf{c}_2$ be two eigenvectors, then $\mathbf{c}_1^\top \mathbf{c}_2 = 0$.

The last property has an interesting implication. Imagine we take the eigenvalues and eigenvectors of a covariance matrix, $\mathbf{S}$. This matrix is symmetric. Taking the matrix of eigenvectors, we now have $\mathbf{C}^\top \mathbf{C} = \mathbf{I}$. The off-diagonal elements are all 0 because we have just seen that the eigenvectors are independent. The diagonal elements are all 1 because that is the convention we applied before. Parenthetically, it also follows that $\mathbf{C}^\top = \mathbf{C}^{-1}$. (Why?) The latter property means that $\mathbf{C}$ is a **unitary matrix**.

7.4 Diagonalization and Spectral Decomposition

Consider a symmetric matrix $\mathbf{A}$. It can now be shown that[2]

$$\mathbf{C}^\top \mathbf{A} \mathbf{C} \;=\; \mathbf{L}$$

We call $\mathbf{C}^\top \mathbf{A} \mathbf{C}$ the **diagonalization** of $\mathbf{A}$ because we obtain a diagonal matrix of eigenvalues. The appeal of diagonalization is that the diagonalized matrix makes it extremely simple to compute ranks, determinants, and inverses, among other things.

Example 7.5. We are interested in the determinant of

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

We know that $\det\mathbf{L} = \det(\mathbf{C}^\top)\det\mathbf{A}\det\mathbf{C}$. We also know that $\det(\mathbf{C}^\top)\det\mathbf{C} = \det\mathbf{I}$. Since the identity matrix has a determinant of 1, it follows that $\det\mathbf{C} = \det(\mathbf{C}^\top) = \pm 1$. Thus,

$$\det\, \mathbf{L} = \pm 1 \cdot \det\, \mathbf{A} \cdot \pm 1 = \det\, \mathbf{A}$$

But $\det\mathbf{L}$ is the product of the eigenvalues contained in $\mathbf{L}$. In our example, these eigenvalues are 8.288, -0.558, and -1.730. The product of these numbers is 8, which is also the determinant of $\mathbf{A}$.

[2]The proof of this result is easy. Since $\mathbf{C}^\top = \mathbf{C}^{-1}$, and since $\mathbf{AC} = \mathbf{CL}$, we can write $\mathbf{C}^\top \mathbf{AC} = \mathbf{C}^{-1}\mathbf{CL} = \mathbf{L}$.

For the diagonalization we premultiplied $\mathbf{AC}$ with $\mathbf{C}^\top$. If we post-multiply, we get[3]

$$\mathbf{A} \;=\; \mathbf{CLC}^\top = \sum_{i=1}^{m} \lambda_i \mathbf{c}_i \mathbf{c}_i^\top$$

We call this the **spectral decomposition** or Jordan decomposition of the matrix $\mathbf{A}$. It will come in handy when we engage in principal component analysis.[4] It is one form of matrix decomposition; we shall encounter other ones in the next chapter.

Example 7.6. Let

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

We obtain

$$\mathbf{L} = \begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix}$$

and

$$\mathbf{C} = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

We now demonstrate the spectral decomposition result:

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{3}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{3}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$$

An interesting feature of spectral decomposition is that it can be generalized quite easily to powers of matrix. For example, if $\mathbf{B} = \mathbf{A}^2$, then the spectral decomposition is $\mathbf{B} = \mathbf{CL}^2\mathbf{C}^\top$. In general,

$$\mathbf{A}^n \;=\; \mathbf{CL}^n\mathbf{C}^\top$$

[3]The proof is again straightforward. Since $\mathbf{C}^\top = \mathbf{C}^{-1}$, we know that $\mathbf{ACC}^\top = \mathbf{ACC}^{-1} = \mathbf{A}$. We also know that $\mathbf{AC} = \mathbf{CL}$, so that $\mathbf{ACC}^\top = \mathbf{CLC}^\top$. Hence, $\mathbf{A} = \mathbf{CLC}^\top$.

[4]Spectral decomposition is an example of Schur decomposition. In Schur decomposition, the square matrix $\mathbf{A}$ is decomposed as $\mathbf{QUQ}^\top$, where $\mathbf{U}$ is an upper-triangular matrix. In the case of spectral decomposition, the upper-triangular matrix is reduced to a diagonal matrix.

7.5 Positive (Semi-) Definite Matrices

A quadratic form $\mathbf{x}^\top \mathbf{A}\mathbf{x}$ is said to be positive definite if it is positive for all values of $\mathbf{x}$ other than zero. If the quadratic form is positive or zero for all values of $\mathbf{x}$ other than zero, then it is said to be positive semi-definite. Positive (semi-) definiteness can be demonstrated by extracting the eigenvalues:

- $\mathbf{A}$ is positive definte (pos. def.) if all of the eigenvalues are positive.
- $\mathbf{A}$ is positive semi-definite if all of the eigenvalues are non-negative.

Covariance matrices are positive semi-definite.

7.6 Principal Component Analysis

One of the most important statistical applications of eigenvalues and eigenvectors is **principal components analysis**. Principal components analysis is a data reduction technique in which a set of variables is transformed into a smaller set of linear combinations that accounts for most of the variance of the variables. The mathematical devise for doing so is the extraction of eigenvalues and eigenvectors of the covariance matrix of the variables.[5]

7.6.1 Extracting Principal Components

To determine our thoughts, let us consider the data in Table 7.1. These data represent the scores of key Dutch political parties on a variety of political issues at the time of the 2014 European Elections. The issues are: A = spending verus taxes; B = redistribution; C = urban-rural emphasis; D = immigration; E = law and order; F = social lifestyle; G = religious principle; and H = environment. The question is if these issues break down along a number of dimensions that describe the Dutch political space. Principal components analysis helps to answer this question.

ince principal components analysis is carried out on a covariance matrix, our

[5]It is also possible to perform principal components analysis on the correlation matrix. The general principles for doing so are the same as for covariances.

Table 7.1: Chapel Hill Expert Survey Data for the Netherlands

party	A	B	C	D	E	F	G	H
50Plus	3.3	4.3	4.5	5.0	6.0	3.0	2.3	5.3
CDA	6.7	6.3	7.6	6.5	6.9	5.4	6.9	6.3
CU	4.7	4.1	6.5	3.8	5.8	7.2	8.4	3.8
D66	5.9	5.4	2.7	2.1	2.1	0.4	1.3	4.0
GroenLinks	2.8	3.7	3.5	1.1	1.8	0.9	1.7	1.3
PvdA	4.1	3.8	3.8	4.1	4.3	1.9	2.7	4.8
PvdD	2.8	2.7	3.2	2.3	2.8	2.0	1.3	0.9
PVV	4.4	5.0	5.0	9.9	9.3	3.7	3.6	8.2
SGP	6.3	6.2	7.6	8.4	7.6	9.1	9.9	6.0
SP	1.3	1.7	4.3	4.4	4.0	3.1	2.4	4.9
VVD	7.8	7.4	4.0	7.5	7.8	2.2	2.1	7.3

first task is to construct this matrix. Using the results from Chapter 3, we get

$$\mathbf{S} = \begin{bmatrix} & \underline{A} & \underline{B} & \underline{C} & \underline{D} & \underline{E} & \underline{F} & \underline{G} & \underline{H} \\ A & 3.857 & & & & & & & \\ B & 3.156 & 2.790 & & & & & & \\ C & 1.324 & 1.110 & 2.925 & & & & & \\ D & 2.637 & 2.633 & 2.734 & 7.759 & & & & \\ E & 2.504 & 2.445 & 2.728 & 6.683 & 6.227 & & & \\ F & 1.582 & 1.208 & 4.261 & 4.004 & 3.986 & 7.197 & & \\ G & 2.143 & 1.792 & 4.914 & 3.865 & 3.970 & 7.924 & 9.318 & \\ H & 2.371 & 2.281 & 1.681 & 5.688 & 4.929 & 2.044 & 2.065 & 5.086 \end{bmatrix}$$

(only the main diagonal and lower-triangular elements are shown due to the symmetry of the covariance matrix).

Next, we extract the eigenvalues and eigenvectors. We obtain the following eigenvalues sorted by size:

$$\mathbf{L} = \begin{bmatrix} 29.918 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9.929 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 4.093 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.575 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.278 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.207 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.087 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.072 \end{bmatrix}$$

We see that all the eigenvalues are positive so that $\mathbf{S}$ is a positive definite matrix. The corresponding eigenvectors are:

$$\mathbf{C} = \begin{bmatrix} -0.217 & -0.173 & 0.718 & 0.050 & -0.154 & -0.152 & -0.149 & 0.580 \\ -0.191 & -0.203 & 0.546 & 0.154 & 0.131 & 0.212 & 0.360 & -0.640 \\ -0.273 & 0.221 & -0.010 & -0.198 & 0.664 & 0.510 & 0.143 & 0.340 \\ -0.441 & -0.378 & -0.302 & 0.278 & -0.399 & 0.551 & -0.166 & 0.068 \\ -0.410 & -0.281 & -0.220 & 0.392 & 0.499 & -0.523 & -0.163 & -0.045 \\ -0.414 & 0.438 & -0.128 & 0.175 & -0.304 & -0.230 & 0.651 & 0.146 \\ -0.459 & 0.538 & 0.124 & -0.206 & -0.110 & -0.036 & -0.565 & -0.330 \\ -0.311 & -0.420 & -0.121 & -0.794 & -0.082 & -0.203 & 0.176 & -0.054 \end{bmatrix}$$

It is easily verified that the components are independent, as they should be since $\mathbf{S}$ is symmetric. The interpretation of the elements of the eigenvectors is as follows: c_{ij} gives the contribution of the ith variable to the jth principal component. For instance c_{11} gives the contribution of the spending item to the first principal component. We can use the elements to extract meaning for the component. For instance, when looking at the 3rd principal component, the largest elements (in absolute terms) are c_{13} and c_{23}. Thus, it appears that the component primarily picks up on socioeconomic policies (spending and redistribution).

We can think of principal component analysis in terms of spectral decomposition. Specifically,

$$\mathbf{S} = \mathbf{CLC}^{\top}$$

where the columns of $\mathbf{C}$ are independent of each other.

The matrix $\mathbf{L}$ has an interesting property:

$$\text{Tr } \mathbf{L} = \sum_{i=1}^{P} s_i^2 = \text{Tr } \mathbf{S}$$

Here, Tr $\mathbf{S}$ is the **total variance** in the set of P variables. In line with this result, we say that principal component analysis accounts for the total variation in the data.

We can use the result about the eigenvalues to formulate a measure of each component's contribution to the total variance:

$$\lambda_i^* = \frac{\lambda_i}{\text{Tr } \mathbf{L}}$$

For the data in Table 7.1, Tr $\mathbf{L} = 45.158$. The contribution of the 1st component is thus $29.918/45.158 = 0.663$. This means that the 1st component accounts for 66.3 percent of the total variance in the eight policy domains.

7.6.2 Data Reduction

Now that we have discussed the mechanism for extracting principal components, it is time to discuss the relationship between principal components analysis and data reduction. So far, it appears that no data reduction is associated with this type of analysis. After all, there are as many principal components as there are original variables.[6] However, in practice we only retain those principal components that account for most of the total variance in the variables. In this case, the number of **retained** principal components, K, is less than the number of variables.

An important question is how we decide on the number of principal components that we retain. There are several rules that can guide this decision. Three common rules are:

1. **Cumulative explained variance:** Retain enough principal components so that one is happy with the total explained variance.
2. **Kaiser's rule**: Retain only those principal components with eigenvalues $\lambda \geq 1$.
3. **Scree rule:** Retain only those principal components with eigenvalues that are clearly greater than all other eigenvalues.

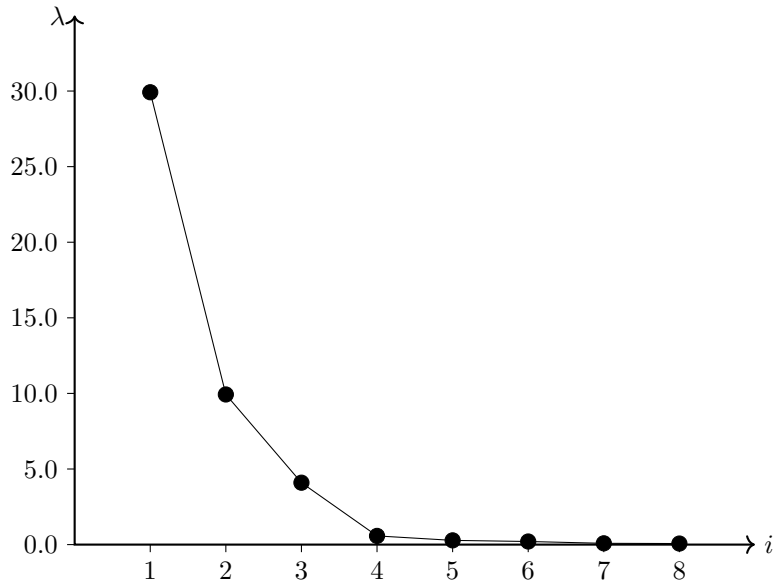
The scree criterion is often assessed on the basis of a **scree plot**. This plot indexes the components on the horizontal axis, in the order in which they were extracted. The vertical axis contains eigenvalues. Since all components have an associated eigenvalue, it is possible to depict this eigenvalue as a point in the plot. We then connect these points to form a curve. The scree criterion is operationalized as the elbow point on the curve. That is prior to this point, the eigenvalues decline rapidly, while they decline more slowly after this point. The criterion states that we should retain only those components up to the elbow point.

Example 7.7. For the data in Table 7.1, the cumulative explained variances are as follows:

1	2	3	4	5	6	7	8
66.3	88.2	97.3	98.6	99.2	99.6	99.8	100.0

[6]The exception to this rule is when **S** is not full rank. This would happen if several variables are perfectly correlated.

Figure 7.1: Scree Plot



With 3 components, then, we explain 97.3% of the variance. The additional components add very little, so that we can easily forego them.

Example 7.8. Looking at the matrix **L**, we observe three eigenvalues that exceed 1. Thus, the Kaiser criterion, too, suggests that we retain three principal components.

Example 7.9. Figure 7.1 shows the scree plot for the Dutch party data. Based on this plot, we observe an elbow at 4, which means that we again retain 3 components. (Remember we retain components to the left of the elbow.)

7.6.3 Interpretation

Given that we retain only components 1-3, the question is how one should interpret these. Table 7.2 shows the first three eigenvectors, retaining only those weights that are at least 0.3 in absolute values. Removing smaller weights has the advantage of clarifying the picture. As we already discussed, the third component is driven primarily by economic issues. The second component is driven mostly by moral issues, although the environment plays somewhat of a role. The first component would seem to approach a GAL-TAN (Green-Alternative-Libertarian versus Traditional-Authoritarian-Nationalist)

Table 7.2: First Three Principal Components with $|c_{ij}| \geq 0.3$

	Principal Component		
	PC1	PC2	PC3
Spending versus taxes			0.718
Redistribution			0.546
Urban-rural emphasis			
Immigration	-0.441	-0.378	-0.302
Law and order	-0.410		
Social lifestyle	-0.414	0.438	
Religious principle	-0.459	0.538	
Environment	-0.311	-0.420	

dimension. Note that some wights have positive signs, while others have negative signs. This depends on the direction of the policy item: sometimes a high score leans toward say the GAL end of the scale and other times it is a low score that captures this.

Chapter 8

Matrix Factorization

Matrix factorization or decomposition is the process of decomposing a matrix into a product of two or more other matrices. The general goal is for those matrices to be simpler than the original matrix. There are many varieties of matrix decomposition; indeed, we have already encountered one variety, namely spectral decomposition. Here we shall consider several other types of decomposition and their applications, namely LU, Cholesky, QR, and singular value decompositions. The differences between these procedures are noted in Table 8.1. Matrix decompositions play a critical role in finding inverses, solving systems of equations, and obtaining least squares estimators, to name just a few applications. Computer programmers rely heavily on matrix decomposition to implement matrix routines. They play a major role in machine learning. Thus, the topic is sufficiently important to consider it in some detail.

Table 8.1: Different Forms of Matrix Factorization

	Requirements of Matrix		
Method	Square	Symmetric	Pos. Def.
LU	Yes	No	No
Cholesky	Yes	Yes	Yes
QR	No	No	No
SVD	No	No	No

8.1 LU Decomposition

8.1.1 Algorithm

LU decomposition is a procedure whereby a square matrix is decomposed into a lower and an upper triangular matrix.[1] Let $\mathbf{A}$ denote the square matrix that we want to decompose. Then

$$\mathbf{A} = \mathbf{LU}$$

is the LU decomposition, provided that $\mathbf{L}$ is a lower-triangular and $\mathbf{U}$ is an upper-triangular matrix. For example, for a 3×3 matrix, the LU decomposition is

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{23} & l_{33} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

LU decompositions are not unique. It is easy to see why. If we were to post-multiply $\mathbf{L}$ with and arbitrary regular matrix $\mathbf{P}$ and we were to pre-multiply $\mathbf{U}$ with $\mathbf{P}^{-1}$, then the solution would again be $\mathbf{A}$: $\mathbf{LPP}^{-1}\mathbf{U} = \mathbf{LU} = \mathbf{A}$.

The non-uniqueness of LU decomposition requires that we impose certain constraints. It is customary to set $l_{ii} = 1$. The remaining elements of $\mathbf{L}$ are then found via

$$l_{ij} = u_{jj}^{-1} \left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj} \right)$$

for $j \leq i - 1$, which reduces to $u_{jj}^{-1} s_{ij}$ if $j - 1 < 1$ (i.e., in the first column). The elements of $\mathbf{U}$ are found via

$$u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj}$$

for $j \geq i$. This reduces to $u_{ij} = a_{ij}$ if $i - 1 < j$ (i.e., in the first row).

Example 8.1. We now work through the LU decomposition of

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

[1] LU decomposition can be generalized to partitioned matrices and, in that context, is often referred to as block LU decomposition.

Starting with the first row, we have

$$\begin{aligned} l_{11} &= 1 \\ l_{12} &= 0 \\ l_{13} &= 0 \end{aligned}$$

All of this follows from the convention that $l_{ii} = 1$ and that the upper-triangular elements of a lower-triangular matrix are all 0. Further,

$$\begin{aligned} u_{11} &= a_{11} = 1 \\ u_{12} &= a_{12} = 2 \\ u_{13} &= a_{13} = 3 \end{aligned}$$

(since $i - 1 < 1$). In the second row, we have

$$\begin{aligned} l_{21} &= u_{11}^{-1} a_{21} = \frac{1}{1} \cdot 4 = 4 \\ l_{22} &= 1 \\ l_{23} &= 0 \end{aligned}$$

Further,

$$\begin{aligned} u_{21} &= 0 \\ u_{22} &= a_{22} - l_{21} u_{12} = 5 - 4 \cdot 2 = -3 \\ u_{23} &= a_{23} - l_{21} u_{13} = 4 - 4 \cdot 3 = -8 \end{aligned}$$

Finally, in the last row we have

$$\begin{aligned} l_{31} &= u_{11}^{-1} a_{31} = \frac{1}{1} \cdot 3 = 3 \\ l_{32} &= u_{22}^{-1} \left(a_{32} - l_{31} u_{12} \right) = -\frac{1}{3}(2 - 3 \cdot 2) = \frac{4}{3} \\ l_{33} &= 1 \end{aligned}$$

and

$$\begin{aligned} u_{31} &= 0 \\ u_{32} &= 0 \\ u_{33} &= a_{33} - (l_{31} u_{13} + l_{32} u_{23}) = 1 - \left(3 \cdot 3 + \frac{4}{3} \cdot -8 \right) = \frac{8}{3} \end{aligned}$$

Thus,

$$\mathbf{L} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 3 & \frac{4}{3} & 1 \end{bmatrix} \quad \mathbf{U} = \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -8 \\ 0 & 0 & \frac{8}{3} \end{bmatrix}$$

(You should verify that $\mathbf{LU} = \mathbf{A}$.)

8.1.2 Applications

One of the most important applications of LU decomposition is in solving systems of equations. Consider the system of equations $\mathbf{y} = \mathbf{Xb}$ in the normal situation that $\mathbf{X}$ is square (i.e. there are as many unknowns as there are equations). Applying the LU decomposition $\mathbf{X} = \mathbf{LU}$, the system of equations may also be written as

$$\mathbf{L}(\mathbf{Ub}) = \mathbf{y}$$

Denoting $\mathbf{Ub} = \mathbf{z}$, a solution can now be found in two steps:

1. Solve for $\mathbf{z}$ in $\mathbf{Lz} = \mathbf{y}$.
2. Once $\mathbf{z}$ is obtained, solve for $\mathbf{b}$ in $\mathbf{Ub} = \mathbf{z}$.

The advantage of solving a system of equations in this manner is that a triangular matrix can be solved using forward or backward substitution. Consider for example the system $\mathbf{Lz} = \mathbf{y}$. This consists of the following linear equations:

$$\begin{aligned} y_1 &= l_{11}z_1 \\ y_2 &= l_{21}z_1 + l_{22}z_2 \\ y_3 &= l_{31}z_1 + l_{32}z_2 + l_{33}z_3 \end{aligned}$$

etcetera. The first equation in this system yields the solution $z_1 = y_1/l_{11}$. By forward substitution, the second equation has the solution $z_2 = (y_2 - l_{21}z_1)/l_{22}$, etcetera. Since $\mathbf{U}$ is triangular as well, the solution for $\mathbf{b}$ in $\mathbf{Ub} = \mathbf{z}$ is equally straightforward.

Example 8.2. Consider the following system of equations:

$$\begin{aligned} a + 2b + 3c &= 5 \\ 4a + 5b + 4c &= 14 \\ 3a + 2b + c &= 7 \end{aligned}$$

This system has $\mathbf{X} = \mathbf{A}$ as its coefficient matrix, where $\mathbf{A}$ is the matrix for which we performed an LU decomposition in Example 8.1. Thus,

$$\begin{aligned} \mathbf{Lz} &= \mathbf{y} \\ \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 3 & \frac{4}{3} & 1 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} &= \begin{bmatrix} 5 \\ 14 \\ 7 \end{bmatrix} \end{aligned}$$

This system describes three equations:

$$\begin{aligned} z_1 &= 5 \\ 4z_1 + z_2 &= 14 \\ 3z_1 + \frac{4}{3}z_2 + z_3 &= 7 \end{aligned}$$

By forward substitution, we know that $z_1 = 5$, $z_2 = -6$, and $z_3 = 0$. We can now solve for the parameters $\mathbf{b}$:

$$\begin{aligned} \mathbf{Ub} &= \mathbf{z} \\ \begin{bmatrix} 1 & 2 & 3 \\ 0 & -3 & -8 \\ 0 & 0 & \frac{8}{3} \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} &= \begin{bmatrix} 5 \\ -6 \\ 0 \end{bmatrix} \end{aligned}$$

This system again comprises three equations:

$$\begin{aligned} a + 2b + 3c &= 5 \\ -3b - 8c &= -6 \\ \frac{8}{3}c &= 0 \end{aligned}$$

By backward substitution, we now find $c = 0$, $b = 2$, and $a = 1$.

8.2 Cholesky Decomposition

Cholesky decomposition is named after the French mathematician André-Louis Cholesky. It plays a major role in computational statistics because of it is speed (by a factor of 2 compared to alternative methods for solving systems of equations). Relative to LU decompisition, it is more restrictive because it requires the matrix to be symmetric and positive definite.

8.2.1 Algorithm

Consider a symmetric, positive definite matrix $\mathbf{A}$. The Cholesky decomposition of that matrix is

$$\mathbf{A} = \mathbf{L}\mathbf{L}^\top$$

where $\mathbf{L}$ is a lower-triangular matrix. One can think of Cholesky decomposition as the special case of LU decomposition that arises when $\mathbf{U} = \mathbf{L}^\top$. Cholesky factorization can also be viewed as taking the square root of a matrix, since $\mathbf{L}$ multiplied into its transpose is like squaring a root in order to obtain the original matrix.

The diagonal elements of $\mathbf{L}$ are found using

$$l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)$$

This reduces to $\sqrt{a_{ii}}$ if $i - 1 < 1$, i.e., in the first column. For the off-diagonal elements, we have

$$l_{ji} = l_{ii}^{-1} \left(a_{ji} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right)$$

This reduces to $l_{ji} = l_{ii}^{-1} a_{ji}$ in the first column.

Example 8.3. To illustrate the procedure, let us consider the following 4×4 matrix:

$$\mathbf{A} = \begin{bmatrix} 49 & 14 & 7 & -14 \\ 14 & 85 & -16 & 5 \\ 7 & -16 & 105 & -34 \\ -14 & 5 & -34 & 158 \end{bmatrix}$$

The matrix is symmetric and positive definite (the eigenvalues are 177.6, 100.4, 77.8, and 41.2, respectively), which means that we can perform a Cholesky decomposition. Starting with the first row, we have

$$\begin{aligned} l_{11} &= \sqrt{a_{11}} = \sqrt{49} = 7 \\ l_{12} &= 0 \\ l_{13} &= 0 \\ l_{14} &= 0 \end{aligned}$$

For the second row, we have

$$\begin{aligned} l_{21} &= l_{11}^{-1} a_{21} = \frac{1}{7} \cdot 14 = 2 \\ l_{22} &= \sqrt{a_{22} - l_{21}^2} = \sqrt{85 - 2^2} = 9 \\ l_{23} &= 0 \\ l_{24} &= 0 \end{aligned}$$

The third row of $\mathbf{L}$ can be computed as

$$\begin{aligned} l_{31} &= l_{11}^{-1} a_{31} = \frac{1}{7} \cdot 7 = 1 \\ l_{32} &= l_{22}^{-1} \left(a_{32} - l_{21} l_{31} \right) = \frac{1}{9}(-16 - 2 \cdot 1) = -2 \\ l_{33} &= \sqrt{a_{33} - (l_{31}^2 + l_{32}^2)} = \sqrt{105 - (1^2 + (-2)^2)} = 10 \\ l_{34} &= 0 \end{aligned}$$

Finally,

$$\begin{aligned} l_{41} &= l_{11}^{-1} a_{41} = \frac{1}{7} \cdot -14 = -2 \\ l_{42} &= l_{22}^{-1} \left(a_{42} - l_{21} l_{41} \right) = \frac{1}{9}(5 - 2 \cdot -2) = 1 \\ l_{43} &= l_{33}^{-1} \left(a_{43} - (l_{31} l_{41} + l_{32} l_{42}) \right) = \frac{1}{10}(-34 - (1 \cdot -2 + -2 \cdot 1)) = 3 \\ l_{44} &= \sqrt{a_{44} - (l_{41}^2 + l_{42}^2 + l_{43}^2)} = \sqrt{158 - ((-2)^2 + 1^2 + 3^2)} = 12 \end{aligned}$$

Hence,

$$\mathbf{L} = \begin{bmatrix} 7 & 0 & 0 & 0 \\ 2 & 9 & 0 & 0 \\ 1 & -2 & 10 & 0 \\ -2 & 1 & -3 & 12 \end{bmatrix}$$

You should verify that $\mathbf{L}\mathbf{L}^\top = \mathbf{A}$.

8.2.2 Applications

8.2.2.1 Systems of Equations

Cholesky decomposition has several important statistical applications. First, it can be used in a manner similar to LU decomposition to solve systems of equations. Specifically, in the system $\mathbf{X}\mathbf{b} = \mathbf{y}$, we can replace a symmetric and positive definite matrix $\mathbf{X}$ with $\mathbf{L}\mathbf{L}^\top$. The system of equations can then be rewritten as $\mathbf{L}\left(\mathbf{L}^\top \mathbf{b}\right) = \mathbf{y}$. Setting $\mathbf{z} = \mathbf{L}^\top \mathbf{b}$, we first solve the system $\mathbf{L}\mathbf{z} = \mathbf{y}$ for $\mathbf{z}$. Once $\mathbf{z}$ is known, we can use $\mathbf{L}^\top \mathbf{b} = \mathbf{z}$ to find $\mathbf{b}$. The advantage of this procedure is similar to LU decomposition, namely that the triangular nature of $\mathbf{L}$ and $\mathbf{L}^\top$ allows us to find solutions using forward and backward substitution.

One application of this logic is least squares estimation. Recall the normal equations from Chapter 5:

$$\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$$

Letting $\mathbf{X}^\top\mathbf{X} = \mathbf{A}$ and $\mathbf{X}^\top\mathbf{y} = \mathbf{c}$, the normal equations may be written as

$$\mathbf{A}\boldsymbol{\beta} = \mathbf{c}$$

Since $\mathbf{A}$ is a matrix of sums-of-squares and cross-products, it is symmetric. In the absence of perfect multicollinearity and micronumerosity, $\mathbf{A}$ is also positive definite. This allows us to solve the normal equations using Cholesky decomposition.[2] This is done in three steps:

1. Obtain the Cholesky decomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$. Rewrite the normal equations as $\mathbf{L}\mathbf{L}^\top\boldsymbol{\beta} = \mathbf{c}$.
2. Let $\mathbf{z} = \mathbf{L}^\top\boldsymbol{\beta}$. Then solve $\mathbf{L}\mathbf{z} = \mathbf{c}$ for $\mathbf{z}$.
3. Solve $\mathbf{z} = \mathbf{L}^\top\boldsymbol{\beta}$ for $\boldsymbol{\beta}$.

Since Cholesky decomposition is fast, this is an attractive approach to finding the least squares estimators, especially when $\mathbf{X}^\top\mathbf{X}$ is large and sparse.

8.2.2.2 Simulating Multivariate Normal Distributions

A second important application of Cholesky decomposition is the simulation of random numbers from multivariate normal distributions. Cholesky decomposition can be used to transform P independent standard normal distributions into a P-variate normal distribution with a theoretical mean of $\boldsymbol{\mu}$ and a theoretical covariance matrix of $\boldsymbol{\Sigma}$.

Since $\boldsymbol{\Sigma}$ is symmetric and positive definite, we can use the decomposition $\boldsymbol{\Sigma} = \mathbf{L}\mathbf{L}^\top$. If we now create a matrix of P standard normal variables, $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$,

[2] Note that Cholesky decomposition may fail even in the absence of perfect multicollinearity. If $\mathbf{X}^\top\mathbf{X}$ is ill-conditioned, i.e., there is near perfect multicollinearity, then rounding error may cause negative values of l_{ii}^2. This makes it impossible to take the square root, as is required, and the algorithm breaks down.

then we obtain a P-variate normal distribution as follows:

$$\begin{aligned}\mathbf{LZ}+\boldsymbol{\mu} &\sim \mathcal{N}\left(\mathbf{L}\mathbb{V}[\mathbf{Z}]\mathbf{L}^\top\right)\\ &\sim \mathcal{N}\left(\boldsymbol{\mu},\mathbf{LIL}^\top\right)\\ &\sim \mathcal{N}\left(\boldsymbol{\mu},\mathbf{LL}^\top\right)\\ &\sim \mathcal{N}\left(\boldsymbol{\mu},\boldsymbol{\Sigma}\right)\end{aligned}$$

This is an attractive procedure since the simulation of univariate standard normal variables is relatively straightforward. Once those variables have been generated they can be turned into any multivariate normal distribution.

Example 8.4. As an example, consider the problem of creating a bivariate normal distribution with

$$\boldsymbol{\mu}=\begin{bmatrix}1\\2\end{bmatrix}\quad\boldsymbol{\Sigma}=\begin{bmatrix}9&4\\4&16\end{bmatrix}$$

(The covariance structure implies a correlation of $\frac{1}{3}$.) We begin by simulating two independent standard normal variables. We do this by taking n draws each from two standard normal distributions, where n is the desired sample size. We place the results in $\mathbf{Z}$. For our example, we assume that we generated the following $n=5$ observations:

$$\mathbf{Z}=\begin{bmatrix}-1.191&0.902\\0.107&-0.538\\0.613&1.040\\-2.102&-0.014\\-0.917&1.364\end{bmatrix}$$

Next, we perform the Cholesky decomposition on $\boldsymbol{\Sigma}$:

$$\mathbf{L}=\begin{bmatrix}3.000&0.000\\1.333&3.771\end{bmatrix}$$

Finally, for each row in $\mathbf{Z}$, we create linear combinations $\mathbf{x}_i=\boldsymbol{\mu}+\mathbf{Lz}_i$, where i denotes a particular row. (In this formula, we continue with the convention that vectors are column vectors. Thus, a row becomes a column.) For example, the 1st row in $\mathbf{Z}$ is transformed as

$$\begin{bmatrix}1.000\\2.000\end{bmatrix}+\begin{bmatrix}3.000&0.000\\1.333&3.771\end{bmatrix}\begin{bmatrix}-1.191\\0.902\end{bmatrix}=\begin{bmatrix}-2.573\\3.814\end{bmatrix}$$

If we proceed the same way for all of the observations, then we get

$$\mathbf{X} = \begin{bmatrix} -2.573 & 3.814 \\ 1.321 & 0.114 \\ 2.839 & 6.739 \\ -5.306 & -0.855 \\ -1.751 & 5.921 \end{bmatrix}$$

With so few observations, the results will not be perfect. We shall revisit the process when we discuss R code and will be in a position to simulate thousands of observations.

8.3 QR Decomposition

QR decomposition of a matrix $\mathbf{A}$, which may be rectangular, entails factorization into an upper triangular matrix, $\mathbf{R}$, and an orthogonal matrix, $\mathbf{Q}$:

$$\mathbf{A} = \mathbf{QR}$$

The orthogonality of $\mathbf{Q}$ implies that $\mathbf{QQ}^\top = \mathbf{I}$. Several methods exist for computing the decomposition. Here, I discuss the Householder and Gram-Schmidt procedures.[3]

8.3.1 Householder Reflections

A Householder reflection or transformation takes a vector and reflects it about some plane.[4] We can do this in such a manner that only one non-zero coordinate remains. This procedure allows us to develop an upper triangular matrix. In the process, we create a Householder matrix, which is orthogonal. Hence, we have all of the ingredients to obtain $\mathbf{Q}$ and $\mathbf{R}$.

Let $\mathbf{a}$ be a m-dimensional vector with a L_2-norm of α. Further, let $\mathbf{e}^\top = (1, 0, \cdots, 0)$ be the first base vector. We now define

$$\begin{aligned} \mathbf{u} &= \mathbf{a} - \alpha\mathbf{e} \\ \mathbf{v} &= \frac{\mathbf{u}}{\|\mathbf{u}\|_2} \\ \mathbf{Q} &= \mathbf{I} - 2\mathbf{v}\mathbf{v}^\top \end{aligned}$$

[3] A third procedure entails so-called Givens rotations.

[4] The procedure was developed by the American mathematician Alston Householder.

Here, $\mathbf{u}\mathbf{u}^\top$ is a symmetric matrix. Further, $\mathbf{Q}$ is a Householder matrix; this matrix is orthogonal. In addition,

$$\mathbf{Q}\mathbf{a} = \mathbf{R}$$

which is an upper-triangular matrix.

In practice, a matrix $\mathbf{A}$ is decomposed by computing successive Householder matrices, which are defined over an ever smaller portion of the original matrix. For the kth Householder matrix, we have

$$\mathbf{Q}_k = \begin{bmatrix} \mathbf{I}_{k-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{B}_k \end{bmatrix}$$

where $\mathbf{B}_k = \mathbf{I} - \mathbf{v}_k\mathbf{v}_k^\top$. Further,

$$\mathbf{R} = \mathbf{Q}_t \cdots \mathbf{Q}_2\mathbf{Q}_1\mathbf{A}$$

and

$$\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2 \cdots \mathbf{Q}_t$$

The procedure may look complicated in the abstract, but an example will show that it is not that difficult in practice.

Example 8.5. Consider the following square matrix

$$\mathbf{A} =] \begin{bmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{bmatrix}$$

Let us take the column vector $\mathbf{a}_1 = (12, 6, -4)^\top$. The Euclidean norm is $\|\mathbf{a}_1\|_2 = \sqrt{12^2 + 6^2 + (-4)^2} = 14$. We take $\mathbf{e}_1 = (1, 0, 0)^\top$, so that

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{a}_1 - \|\mathbf{a}_1\|_2\mathbf{e}_1 \\ &= \begin{bmatrix} 12 \\ 6 \\ -4 \end{bmatrix} - 14 \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} -2 \\ 6 \\ -4 \end{bmatrix} \end{aligned}$$

This vector has a norm of $\|\mathbf{u}_1\|_2 = \sqrt{(-2)^2 + 6^2 + (-4)^2} = 2\sqrt{14}$. Hence,

$$\begin{aligned}
\mathbf{v}_1 &= \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|_2} \\
&= \frac{1}{2\sqrt{14}} \begin{bmatrix} -2 \\ 6 \\ -4 \end{bmatrix} \\
&= \frac{1}{\sqrt{14}} \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix}
\end{aligned}$$

and

$$\begin{aligned}
\mathbf{Q}_1 &= \mathbf{I} - 2\mathbf{v}_1\mathbf{v}_1^\top \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \frac{2}{\sqrt{14}\sqrt{14}} \begin{bmatrix} -1 \\ 3 \\ -2 \end{bmatrix} \begin{bmatrix} -1 & 3 & -2 \end{bmatrix} \\
&= \begin{bmatrix} \frac{6}{7} & \frac{3}{7} & -\frac{2}{7} \\ \frac{3}{7} & -\frac{2}{7} & \frac{6}{7} \\ = \frac{2}{7} & \frac{6}{7} & \frac{3}{7} \end{bmatrix}
\end{aligned}$$

Now

$$\mathbf{Q}_1\mathbf{A} = \begin{bmatrix} \frac{6}{7} & \frac{3}{7} & -\frac{2}{7} \\ \frac{3}{7} & -\frac{2}{7} & \frac{6}{7} \\ = \frac{2}{7} & \frac{6}{7} & \frac{3}{7} \end{bmatrix} \begin{bmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{bmatrix} = \begin{bmatrix} 14 & 21 & -14 \\ 0 & -49 & -14 \\ 0 & 168 & -77 \end{bmatrix}$$

We now have the beginning of an upper-triangular matrix, but we are not quite there yet. To continue the decomposition, we focus on one of the sub-matrices of $\mathbf{Q}_1\mathbf{A}$, to wit

$$\mathbf{A}_{11} = \begin{bmatrix} -49 & -14 \\ 168 & -77 \end{bmatrix}$$

This is the sub-matrix that arises when we drop the 1st row and the 1st column. Let us create the column vector $\mathbf{a}_2 = (-49, 168)^\top$, which has a norm of $\|\mathbf{a}_2\|_2 = \sqrt{(-49)^2 + 168^2} = 175$. Taking the base vector $\mathbf{e}_2 = (1, 0)^\top$, we have

$$\begin{aligned}
\mathbf{u}_2 &= \mathbf{a}_1 - \|\mathbf{a}_2\|_2\mathbf{e}_2 \\
&= \begin{bmatrix} -49 \\ 168 \end{bmatrix} - 175 \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} -224 \\ 168 \end{bmatrix}
\end{aligned}$$

The Euclidean norm of this vector is $\|\mathbf{u}_2\|_2 = \sqrt{(-224)^2 + 168^2} = 280$. Hence,

$$\begin{aligned}\mathbf{v}_2 &= \frac{\mathbf{u}_2}{\|\mathbf{u}_2\|_2} \\ &= \frac{1}{280}\begin{bmatrix} -224 \\ 168 \end{bmatrix} \\ &= \begin{bmatrix} -\frac{4}{5} \\ \frac{3}{5} \end{bmatrix}\end{aligned}$$

and

$$\begin{aligned}\mathbf{B}_2 &= \mathbf{I} - 2\mathbf{v}_2\mathbf{v}_2^\top \\ &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - 2\begin{bmatrix} -\frac{4}{5} \\ \frac{3}{5} \end{bmatrix}\begin{bmatrix} -\frac{4}{5} \\ \frac{3}{5} \end{bmatrix} \\ &= \begin{bmatrix} -\frac{7}{25} & \frac{24}{25} \\ \frac{24}{25} & \frac{7}{25} \end{bmatrix}\end{aligned}$$

It now follows that

$$\mathbf{Q}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{7}{25} & \frac{24}{25} \\ 0 & \frac{24}{25} & \frac{7}{25} \end{bmatrix}$$

Further,

$$\mathbf{Q} = \mathbf{Q}_1\mathbf{Q}_2 = \begin{bmatrix} 14 & 21 & -14 \\ 0 & -49 & -14 \\ 0 & 168 & -77 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & -\frac{7}{25} & \frac{24}{25} \\ 0 & \frac{24}{25} & \frac{7}{25} \end{bmatrix} = \begin{bmatrix} \frac{6}{7} & -\frac{69}{175} & \frac{58}{175} \\ \frac{3}{7} & \frac{158}{175} & -\frac{6}{175} \\ -\frac{2}{7} & \frac{6}{35} & \frac{33}{35} \end{bmatrix}$$

and

$$\begin{aligned}\mathbf{R} &= \mathbf{QA} \\ &= \begin{bmatrix} \frac{6}{7} & -\frac{69}{175} & \frac{58}{175} \\ \frac{3}{7} & \frac{158}{175} & -\frac{6}{175} \\ -\frac{2}{7} & \frac{6}{35} & \frac{33}{35} \end{bmatrix}\begin{bmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{bmatrix} = \begin{bmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & -35 \end{bmatrix}\end{aligned}$$

This concludes the Householder algorithm.

8.3.2 Gram-Schmidt Orthogonalization

Gram-Schmidt orthogonalization is a second procedure for QR decomposition.[5] Orthogonalization means that we take a set of LIN vectors, $\mathcal{S} = \{\mathbf{v}_1, \cdots, \mathbf{v}_K\}$, and find orthogonal vectors $\mathbf{u}_1 \cdots \mathbf{u}_K$ that produce the same subspace as $\mathcal{S}$. Thus, we are **projecting** the original vectors onto an orthogonal space.

Key to the process is the projection operator

$$\text{project}_{\mathbf{v}}\mathbf{u} = \frac{\mathbf{u}^\top \mathbf{v}}{\mathbf{v}^\top \mathbf{v}}\mathbf{v}$$

Using this operator, Gram-Schmidt orthogonalization unfolds as follows:

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{v}_1 & \mathbf{e}_1 &= \tfrac{\mathbf{u}_1}{\|\mathbf{u}_1\|_2} \\ \mathbf{u}_2 &= \mathbf{v}_2 \text{project}_{\mathbf{e}_1}\mathbf{v}_2 & \mathbf{e}_2 &= \tfrac{\mathbf{u}_2}{\|\mathbf{u}_2\|_2} \\ \mathbf{u}_3 &= \mathbf{v}_3 \text{project}_{\mathbf{e}_1}\mathbf{v}_3 - \text{project}_{\mathbf{e}_2}\mathbf{v}_3 & \mathbf{e}_3 &= \tfrac{\mathbf{u}_3}{\|\mathbf{u}_3\|_3} \end{aligned}$$

etcetera. In general,

$$\begin{aligned} \mathbf{u}_k &= \mathbf{v}_k - \sum_{j=1}^{k-1} \text{project}_{\mathbf{e}_j}\mathbf{v}_k \\ \mathbf{e}_k &= \frac{\mathbf{u}_k}{\mathbf{u}_k\|_2} \end{aligned}$$

Here, the vectors $\mathbf{e}_k$ are orthonormal.

Example 8.6. As an example of Gram-Schmidt orthogonalization, consider two vectors $\mathbf{v}_1 = (4, 3)^\top$ and $\mathbf{v}_2 = (2, 1)^\top$. We start by setting

$$\begin{aligned} \mathbf{u}_1 &= \mathbf{v}_1 = \begin{bmatrix} 4 \\ 3 \end{bmatrix} \\ \mathbf{e}_1 &= \frac{1}{\|\mathbf{u}_1\|_2}\mathbf{u}_1 = \frac{1}{5}\begin{bmatrix} 4 \\ 3 \end{bmatrix} = \begin{bmatrix} \frac{4}{5} \\ \frac{3}{5} \end{bmatrix} \end{aligned}$$

To obtain $\mathbf{u}_2$, we begin by computing the projection operator

$$\begin{aligned} \text{project}_{\mathbf{e}_1}\mathbf{v}_2 &= \frac{\mathbf{v}_2^\top \mathbf{e}_1}{\mathbf{e}_1^\top \mathbf{e}_1}\mathbf{e}_1 \\ &= \left(\mathbf{v}_2^\top \mathbf{e}_1\right)\mathbf{e}_1 \\ &= \begin{bmatrix} \frac{44}{25} \\ \frac{33}{25} \end{bmatrix} \end{aligned}$$

[5] The procedure is named after Jörgen Pedersen Gram, a Danish mathematician, and Erhard Schmidt, a German mathematician. The ideas go back further, however, to Laplace and Cauchy.

(The second line follows from the unit norm on $\mathbf{e}_1$.) Now

$$\begin{aligned}\mathbf{u}_2 &= \mathbf{v}_2 - \text{project}_{\ \mathbf{e}_1}\mathbf{v}_2 \\ &= \begin{bmatrix} 2 \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{44}{25} \\ \frac{33}{25} \end{bmatrix} \\ &= \begin{bmatrix} \frac{6}{25} \\ -\frac{8}{25} \end{bmatrix}\end{aligned}$$

It is easily verfied that $\mathbf{u}_1$ and $\mathbf{u}_2$ are orthoginal.

How does Gram-Schmidt orthogonalization play into QR decomposition? The starting point is to consider the matrix that is to be decomposed as a set of column vectors: $\mathbf{A} = (\mathbf{a}_1 \cdots \mathbf{a}_K)$. Each column vector is subjected to Gramm-Schmidt orthogonalization:

$$\begin{aligned}\mathbf{u}_1 &= \mathbf{a}_1 \\ \mathbf{u}_k &= \mathbf{a}_k - \sum_{j=1}^{k-1} \text{project}_{\mathbf{e}_j}\mathbf{a}_k\end{aligned}$$

for $k > 1$, where $\mathbf{e}_k = \mathbf{u}_k/\|\mathbf{u}_k\|_2$. Rearranging terms, we get

$$\begin{aligned}\mathbf{a}_1 &= \mathbf{e}_1\|\mathbf{u}_1\|_2 \\ \mathbf{a}_k &= \sum_{j=1}^{k-1} \text{project}_{\mathbf{e}_j}\mathbf{a}_k + \mathbf{e}_k\|\mathbf{u}_k\|_2\end{aligned}$$

Keeping in mind that $\|\mathbf{e}_k\|_2 = 1$, the projecvtion operator can be written as

$$\text{project}_{\mathbf{e}_j}\mathbf{a}_k = \frac{\mathbf{a}_k^\top \mathbf{e}_j}{\mathbf{e}_j^\top \mathbf{e}_j}\mathbf{e}_j = \left(\mathbf{a}_k^\top \mathbf{e}_j\right)\mathbf{e}_j$$

Thus, the equations for the column vectors in $\mathbf{A}$ can also be written as

$$\begin{aligned}\mathbf{a}_1 &= \mathbf{e}_1\|\mathbf{u}_1\|_2 \\ \mathbf{a}_k &= \sum_{j=1}^{k-1} \left(\mathbf{a}_k^\top \mathbf{e}_j\right)\mathbf{e}_j + \mathbf{e}_k\|\mathbf{u}_k\|_2\end{aligned}$$

We now factor out the base vectors $\mathbf{e}_1 \cdot \mathbf{e}_n$:

$$\begin{aligned}\mathbf{A} &= \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 & \cdots & \mathbf{e}_n \end{bmatrix} \begin{bmatrix} \|\mathbf{u}_1\|_2 & \mathbf{a}_2^\top\mathbf{a}_1 & \mathbf{a}_3^\top\mathbf{e}_1 & \cdots & \mathbf{a}_n^\top\mathbf{e}_1 \\ 0 & \|\mathbf{u}_2\|_2 & \mathbf{a}_3^\top\mathbf{e}_2 & \cdots & \mathbf{a}_n^\top\mathbf{e}_2 \\ 0 & 0 & \|\mathbf{u}_3\|_2 & \cdots & \mathbf{a}_n^\top\mathbf{e}_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \|\mathbf{u}_n\|_2 \end{bmatrix} \\ &= \mathbf{QR}\end{aligned}$$

Since $\mathbf{Q}$ is orthogonal, it follows that $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}$ and that $\mathbf{A} = \mathbf{Q}\mathbf{Q}^\top\mathbf{A} = \mathbf{Q}\mathbf{R}$, so that

$$\mathbf{R} \;=\; \mathbf{Q}^\top\mathbf{A}$$

Thus, the trick is to obtain $\mathbf{Q}$. Once it is known, $\mathbf{R}$ follows immediately.

Example 8.7. Consider the decomposition of

$$\mathbf{A} = \begin{bmatrix} 12 & -51 & 4 \\ 6 & 167 & -68 \\ -4 & 24 & -41 \end{bmatrix}$$

(This is the same matrix that we used for the Householder procedure.) Partitioning $\mathbf{A}$ into the column vectors $\mathbf{a}_1 = (12, -52, 4)^\top$, $\mathbf{a}_2 = (6, 167, -68)^\top$, and $\mathbf{a}_3 = (-4, 24, -41)^\top$, we know that

$$\mathbf{u}_1 \;=\; \mathbf{a}_1$$

Further, $\|\mathbf{u}_1\|_2 = 14$, so that

$$\mathbf{e}_1 = \begin{bmatrix} \frac{6}{7} \\ \frac{3}{7} \\ -\frac{2}{7} \end{bmatrix}$$

It then follows that

$$\begin{aligned} \mathbf{u}_2 &= \mathbf{a}_2 - \text{project}_{\mathbf{e}_1}\mathbf{a}_2 \\ &= \mathbf{a}_2 - \left(\mathbf{a}_2^\top\mathbf{e}_1\right)\mathbf{e}_1 \\ &= \begin{bmatrix} -51 \\ 167 \\ 24 \end{bmatrix} - \begin{bmatrix} 18 \\ 9-6 \end{bmatrix} \\ &= \begin{bmatrix} -69 \\ 158 \\ 30 \end{bmatrix} \end{aligned}$$

This vector has a norm $\|\mathbf{u}_2\|_2 = 175$, so that

$$\mathbf{e}_2 = \begin{bmatrix} -\frac{69}{175} \\ \frac{158}{175} \\ \frac{30}{175} \end{bmatrix}$$

Now, it follows that

$$\begin{aligned}
\mathbf{u}_3 &= \mathbf{a}_3 - \text{project}_{\mathbf{e}_1}\mathbf{a}_3 - \text{project}_{\mathbf{e}_2}\mathbf{a}_3 \\
&= \mathbf{a}_3 = \left(\mathbf{a}_3^\top \mathbf{e}_1\right)\mathbf{e}_1 - \left(\mathbf{a}_3^\top \mathbf{e}_2\right)\mathbf{e}_2 \\
&= \begin{bmatrix} 4 \\ -68 \\ -41 \end{bmatrix} - \begin{bmatrix} -12 \\ -6 \\ 4 \end{bmatrix} - \begin{bmatrix} 27\frac{3}{5} \\ -63\frac{1}{5} \\ -12 \end{bmatrix} \\
&= \begin{bmatrix} -11\frac{3}{5} \\ 1\frac{1}{5} \\ -33 \end{bmatrix}
\end{aligned}$$

For the sake of simplicity, we multiply $\mathbf{u}_3$ by 5. The resulting vector has a Euclidean norm of 175, so that

$$\mathbf{e}_3 = \begin{bmatrix} -\frac{58}{175} \\ \frac{6}{175} \\ -\frac{165}{175} \end{bmatrix}$$

(You should verify that multiplication by 5 does not affect the result.) We can now place $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ into a matrix:

$$\mathbf{Q} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} = \begin{bmatrix} \frac{6}{7} & -\frac{69}{175} & -\frac{58}{175} \\ \frac{3}{7}K & \frac{158}{175} & \frac{6}{175} \\ -\frac{2}{7} & \frac{30}{175} & -\frac{165}{175} \end{bmatrix}$$

Further,

$$\mathbf{R} = \mathbf{Q}^\top \mathbf{A} = \begin{bmatrix} 14 & 21 & -14 \\ 0 & 175 & -70 \\ 0 & 0 & 35 \end{bmatrix}$$

In computer implementations of QR decomposition, Gram-Schmidt orthogonalization sometimes fails because $\mathbf{Q}$ winds up being not quite orthogonal due to rounding errors. For this reason, computational statisticians often prefer Householder reflections.

8.3.3 Applications

One of the most important applications of QR decomposition is ordinary least squares estimation of the linear regression model. Given the model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$$

we can perform a QR decomposition on $\mathbf{X}$. We do this on the $K+1$ columns of the matrix, so that $\mathbf{Q}\mathbf{Q}^\top = \mathbf{I}_{K+1}$. Thus, $\mathbf{X} = \mathbf{Q}\mathbf{R}$, which means that the regression model may be written as

$$\begin{aligned} \mathbf{y} &= \mathbf{Q}\mathbf{R}\boldsymbol{\beta} + \boldsymbol{\varepsilon} \\ &= \mathbf{Q}\mathbf{c} + \boldsymbol{\varepsilon} \end{aligned}$$

where $\mathbf{c} = \mathbf{R}\boldsymbol{\beta}$. The least squares estimator for $\mathbf{c}$ is given by

$$\begin{aligned} \hat{\mathbf{c}} &= \left(\mathbf{Q}^\top\mathbf{Q}\right)^{-1}\mathbf{Q}^\top\mathbf{y} \\ &= \mathbf{Q}^\top\mathbf{y} \end{aligned}$$

(by virtue of the fact that $\mathbf{Q}^\top\mathbf{Q} = \mathbf{I}$). Further,

$$\begin{aligned} \mathbb{V}\left[\hat{\mathbf{c}}\right] &= \sigma^2\mathbf{Q}^\top\mathbf{Q} \\ &= \sigma^2\mathbf{I}_{K+1} \end{aligned}$$

(assuming the errors are homoskedastic and uncorrelated). The last result implies that the elements of $\hat{\mathbf{c}}$ are independent.

We now immediately see the advantages of the QR decompositional approach to least squares estimation. Instead of having to create and invert the matrix $\mathbf{X}^\top\mathbf{X}$, all we have to do is to decompose $\mathbf{X}$. Once the QR decomposition has taken place, we can estimate $\mathbf{c}$ without having to invert anything. This simplifies computation and also tends to produce more stable numerical results.

Of course, we are not interested in $\hat{\mathbf{c}}$ but in $\hat{\boldsymbol{\beta}}$. But the elements of that vector are easily derived:

$$\hat{\boldsymbol{\beta}} = \mathbf{R}^{-1}\hat{\mathbf{c}}$$

Since $\mathbf{R}$ is a triangular matrix, its inverse is relatively easily computed (see the discussion in Chapter 3). To obtain the covariance matrix of $\hat{\boldsymbol{\beta}}$, we compute

$$\begin{aligned} \mathbb{V}\left[\hat{\boldsymbol{\beta}}\right] &= \sigma^2\mathbf{R}^{-1}\left(\mathbf{R}^{-1}\right)^\top \\ &= \sigma^2\left(\mathbf{R}^\top\mathbf{R}\right)^{-1} \end{aligned}$$

which is equal to $\sigma^2\left(\mathbf{X}^\top\mathbf{X}\right)^{-1}$.

Example 8.8. Consider the simple regression model $y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$. Imagine that we have the following data:

$$\mathbf{X} = \begin{bmatrix} 1 & 35 \\ 1 & 81 \\ 1 & 10 \\ 1 & 58 \\ 1 & 11 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 70 \\ 88 \\ 19 \\ 63 \\ 33 \end{bmatrix}$$

QR decomposition of $\mathbf{X}$ using Gram-Schmidt orthogonalization yields

$$\mathbf{Q} = \begin{bmatrix} 0.447 & -0.065 \\ 0.447 & 0.684 \\ 0.447 & -0.473 \\ 0.447 & 0.310 \\ 0.447 & -0.456 \end{bmatrix}$$

and

$$\mathbf{R} = \begin{bmatrix} \frac{5}{\sqrt{5}} & \frac{195}{\sqrt{5}} \\ 0 & \sqrt{3766} \end{bmatrix}$$

With these results in place, we have

$$\begin{aligned} \hat{\mathbf{c}} &= \begin{bmatrix} \frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} & \frac{1}{\sqrt{5}} & -\frac{1}{\sqrt{5}} \\ -\frac{4}{\sqrt{3766}} & \frac{42}{sqrt3766} & -\frac{29}{\sqrt{3766}} & \frac{19}{\sqrt{3766}} & -\frac{28}{\sqrt{3766}} \end{bmatrix} \begin{bmatrix} 70 \\ 88 \\ 19 \\ 63 \\ 33 \end{bmatrix} \\ &= \begin{bmatrix} \frac{273}{\sqrt{5}} \\ \frac{3138}{\sqrt{3766}} \end{bmatrix} \end{aligned}$$

To find the least squares estimator, we take the inverse of $\mathbf{R}$:

$$\mathbf{R}^{-1} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{39}{\sqrt{3766}} \\ 0 & \frac{1}{\sqrt{3766}} \end{bmatrix}$$

Multiplying this inverse into $\hat{\mathbf{c}}$ yields

$$\hat{\boldsymbol{\beta}} = \begin{bmatrix} \frac{1}{\sqrt{5}} & -\frac{39}{\sqrt{3766}} \\ 0 & \frac{1}{\sqrt{3766}} \end{bmatrix} \begin{bmatrix} \frac{273}{\sqrt{5}} \\ \frac{3138}{\sqrt{3766}} \end{bmatrix} = \begin{bmatrix} 22.10 \\ 0.83 \end{bmatrix}$$

To obtain the covariance matrix, we need an estimate of σ^2. Let

$$\mathbf{e} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$$

Then

$$\hat{\sigma}^2 = \frac{\mathbf{e}^\top \mathbf{e}}{n-K-1}$$

where K is the number of predictors. In our case,

$$\mathbf{e} = \begin{bmatrix} 70 \\ 88 \\ 19 \\ 63 \\ 33 \end{bmatrix} - \begin{bmatrix} 1 & 35 \\ 1 & 81 \\ 1 & 10 \\ 1 & 58 \\ 1 & 11 \end{bmatrix} \begin{bmatrix} 22.10 \\ 0.83 \end{bmatrix} = \begin{bmatrix} 18.85 \\ -1.33 \\ -11.40 \\ -7.24 \\ 1.77 \end{bmatrix}$$

We now have

$$\hat{\sigma}^2 = \frac{542.60}{5-1-1} = 180.87$$

It then follows that

$$\mathbb{V}\left[\hat{\mathbf{c}}\right] = \hat{\sigma}^2 \mathbf{I} = \begin{bmatrix} 180.87 & 0.00 \\ 0.00 & 180.87 \end{bmatrix}$$

Further,

$$\begin{aligned} \mathbb{V}\left[\hat{\boldsymbol{\beta}}\right] &= \hat{\sigma}^2 \left(\mathbf{R}^\top \mathbf{R}\right)^{-1} = 180.87 \begin{bmatrix} 5 & 195 \\ 195 & 3766 \end{bmatrix}^{-1} \\ &= \begin{bmatrix} 109.22 & -1.87 \\ -1.87 & 0.05 \end{bmatrix} \end{aligned}$$

The number in the first row and first column is the variance in the intercept estimate, $\hat{\beta}_0$. The number in the second row in the second column is the variance in the slope estimate, $\hat{\beta}_1$. The off-diagonal element captures the covariance between the intercept and slope estimates.

8.4 Singular Value Decomposition

8.4.1 Algorithm

Singular value decomposition or SVD is a powerful decomposition method for matrices that are (near-) singular. It takes a $m \times n$ matrix $\mathbf{A}$, where $m \geq n$,

and decomposes the matrix into the product of $m \times n$ column-orthogonal matrix $\mathbf{U}$, a non-negative diagonal matrix $\mathbf{\Sigma}$, and the transpose of a $n \times n$ orthogonal matrix $\mathbf{V}$:

$$\mathbf{A} \quad = \quad \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$$

The diagonal elements of the matrix $\mathbf{\Sigma}$ are known as the **singular values**.

SVD can be performed in a few simple steps:

1. Compute $\mathbf{C} = \mathbf{A}^\top\mathbf{A}$.
2. Obtain the eigenvalues of $\mathbf{C}$ and arrange them in descending order.
3. Determine the number of non-zero eigenvalues of $\mathbf{C}$; this number, r, is the rank of the matrix.
4. Find the orthogonal eigenvectors of $\mathbf{C}$ that correspond to the eigenvalues and arrange them in the same order. These eigenvectors form the columns of $\mathbf{V}$.
5. Form the diagonal matrix $\mathbf{\Sigma}$ by taking the square root of the eigenvalues: $\sigma_{ii} = \sqrt{\lambda_i}$.
6. Find the first r column vectors of $\mathbf{U}$ by computing

$$\mathbf{u}_i \quad = \quad \sigma_{ii}^{-1}\mathbf{A}\mathbf{v}_i$$

7. Obtain the remaining vectors of $\mathbf{U}$ through Gram-Schmidt orthogonalization.

Example 8.9. Let us decompose

$$\mathbf{A} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

We begin by constructing

$$\mathbf{C} = \mathbf{A}^\top\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

The eigenvalues of $\mathbf{C}$ are 3 and 1. Since both eigenvalues are positive, we have $r = 2$. Further,

$$\mathbf{\Sigma} = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & \sqrt{1} \end{bmatrix}$$

The eigenvector corresponding to $\lambda = 3$ is $\mathbf{v}_1 = (1/\sqrt{2}, 1/\sqrt{2})^\top$, whereas the eigenvector corresponding to $\lambda = 1$ is $\mathbf{v}_2 = (-1/\sqrt{2}, 1/\sqrt{2})^\top$. Hence,

$$\mathbf{V} = \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

Finally,

$$\mathbf{u}_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} \frac{2}{\sqrt{2}\sqrt{3}} \\ \frac{1}{\sqrt{2}\sqrt{3}} \\ \frac{1}{\sqrt{2}\sqrt{3}} \end{bmatrix}$$

and

$$\mathbf{v}_2 = \frac{1}{\sqrt{1}} \begin{bmatrix} 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} \frac{1}{\sqrt{2}} \end{bmatrix} = \begin{bmatrix} 0 \\ \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} \end{bmatrix}$$

Example 8.10. As a second example, consider the matrix $\mathbf{A} = [1, -1]$. The matrix $\mathbf{C} = \mathbf{A}^\top \mathbf{A}$ has eigenvalues $\lambda_1 = 2$ and $\lambda_2 = 0$, so that $r = 1$. The eigenvectors corresponding to the eigenvalues are $\mathbf{v}_1 = (-1/\sqrt{2}, 1/\sqrt{2})^\top$ and $\mathbf{v}_2 = (1/\sqrt{2}, 1/\sqrt{2})^\top$. Consequently,

$$\mathbf{V} = \begin{bmatrix} -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$$

and

$$\mathbf{\Sigma} = \begin{bmatrix} \sqrt{3} & 0 \\ 0 & 0 \end{bmatrix}$$

The first column vector of $\mathbf{U}$ is given by

$$\mathbf{u}_1 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} = -\frac{2}{\sqrt{2}\sqrt{3}}$$

To obtain the second column vector of $\mathbf{U}$, we apply the following Gram-Schmidt orthogonalization: $\mathbf{u}_2 = \mathbf{e}_1 - \left(\mathbf{u}_1^\top \mathbf{e}_1\right) \mathbf{u}_1$. Here, $\mathbf{e}_1 = \mathbf{u}_1 / \|\mathbf{u}_1\|_2$. It is easily verified that $\mathbf{e}_1 = -1$ and $\mathbf{u}_2 = -1/3$. Hence,

$$\mathbf{U} = \begin{bmatrix} -\frac{2}{\sqrt{2}\sqrt{3}} & -\frac{1}{3} \end{bmatrix}$$

Note that if a matrix $\mathbf{A}$ is square, symmetric, and idempotent, then the singular values of that matrix are identical to its eiegenvalues. In that case, SVD amounts to diagonalization.

8.4.2 Applications

8.4.2.1 Text Analysis

SVD is often used in text analysis, for example in finding latent topics in a document-term matrix. For example, consider the following matrix consisting of five documents and three terms:

$$\mathbf{X} = \begin{bmatrix} \underline{Immigration} & \underline{Inequality} & \underline{Problem} \\ 4 & 0 & 4 \\ 2 & 0 & 0 \\ 5 & 0 & 2 \\ 3 & 3 & 2 \\ 0 & 5 & 4 \end{bmatrix}$$

The matrix elements are frequencies of the words in each text. SVD now yields

$$\mathbf{\Sigma} = \begin{bmatrix} 9.49 & 0.00 & 0.00 \\ 0.00 & 5.78 & 0.00 \\ 0.00 & 0.00 & 2.15 \end{bmatrix}$$

as the matrix of singular values. Further

$$\mathbf{U} = \begin{bmatrix} -0.546 & 0.302 & 0.674 \\ -0.140 & 0.222 & -0.359 \\ -0.483 & 0.484 & -0.201 \\ -0.470 & -0.121 & -0.601 \\ -0.478 & -0.781 & 0.127 \end{bmatrix}$$

and

$$\mathbf{V} = \begin{bmatrix} -0.663 & 0.642 & -0.385 \\ -0.400 & -0.738 & -0.543 \\ -0.663 & -0.206 & 0.747 \end{bmatrix}$$

We can think of the columns in $\mathbf{U}$ and $\mathbf{V}$ as the latent topics. The entries in $\mathbf{U}$ indicate how much each document correlates with a latent topic. The entries in $\mathbf{V}$ indicate how much each term correlates with a latent topic. We can use the size of the singular values to decide how many latent topics to retain.

8.4.2.2 Ordinary Least Squares

A second major application of SVD is again least squares estimation. In this context, SVD is particularly valuable when the predictor matrix, $\mathbf{X}$, suffers

from collinearity or micronumerosity problems. When $\mathbf{X}$ is full column rank, it can be demonstrated that the least squares estimator is unique and equal to

$$\hat{\boldsymbol{\beta}} = \sum_{i=1}^{K+1} \frac{1}{\sigma_{ii}} \mathbf{u}_i^\top \mathbf{y} \mathbf{v}_i$$

The singular values, $\mathbf{u}_i$, and $\mathbf{v}_i$ are all generated through SVD of $\mathbf{X}$. Further,

$$\mathbb{V}\left[\hat{\boldsymbol{\beta}}\right] = \hat{\sigma}^2 \sum_{i=1}^{K+1} \frac{\mathbf{v}_i \mathbf{v}_i^\top}{\lambda_{ii}}$$

Here, $\hat{\sigma}^2$ is the estimator of the error variance and $\lambda_{ii} = \sigma_{ii}^2$ are the eigenvalues of $\mathbf{X}$.

Example 8.11. For an illustration, we revisit the data we used in QR decomposition. Specifically,

$$\mathbf{X} = \begin{bmatrix} 1 & 35 \\ 1 & 81 \\ 1 & 10 \\ 1 & 58 \\ 1 & 11 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} 70 \\ 88 \\ 19 \\ 63 \\ 33 \end{bmatrix}$$

SVD of $\mathbf{X}$ yields

$$\mathbf{U} = \begin{bmatrix} 0.33 & 0.31 \\ 0.76 & -0.30 \\ 0.09 & 0.64 \\ 0.54 & 0.00 \\ 0.10 & 0.63 \end{bmatrix} \quad \mathbf{\Sigma} = \begin{bmatrix} 106.65 & 0.00 \\ 0.00 & 1.29 \end{bmatrix} \quad \mathbf{V} = \begin{bmatrix} 0.02 & 1.00 \\ 1.00 & -0.02 \end{bmatrix}$$

The first computation yields

$$\begin{aligned} \frac{1}{\sigma_{11}} \mathbf{u}_1^\top \mathbf{y} \mathbf{v}_1 &= \frac{1}{106.65} \begin{bmatrix} 0.33 & 0.76 & 0.09 & 0.54 & 0.10 \end{bmatrix} \begin{bmatrix} 70 \\ 88 \\ 19 \\ 63 \\ 33 \end{bmatrix} \begin{bmatrix} 0.02 \\ 1.00 \end{bmatrix} \\ &= \begin{bmatrix} 0.02 \\ 1.21 \end{bmatrix} \end{aligned} \tag{8.1}$$

The second computation yields

$$\begin{aligned}\frac{1}{\sigma_{22}}\mathbf{u}_2^\top \mathbf{y}\mathbf{v}_2 &= \frac{1}{1.29}\begin{bmatrix} 0.31 & -0.30 & 0.64 & 0.00 & 0.63 \end{bmatrix}\begin{bmatrix} 70 \\ 88 \\ 19 \\ 63 \\ 33 \end{bmatrix}\begin{bmatrix} 1.00 \\ -0.02 \end{bmatrix} \\ &= \begin{bmatrix} 22.08 \\ -0.38 \end{bmatrix} \end{aligned} \tag{8.2}$$

The OLS estimator is now

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \frac{1}{\sigma_{11}}\mathbf{u}_1^\top \mathbf{y}\mathbf{v}_1 + \frac{1}{\sigma_{22}}\mathbf{u}_2^\top \mathbf{y}\mathbf{v}_2 \\ &= \begin{bmatrix} 0.02 \\ 1.21 \end{bmatrix} + \begin{bmatrix} 22.08 \\ -0.38 \end{bmatrix} \\ &= \begin{bmatrix} 22.10 \\ 0.83 \end{bmatrix}\end{aligned}$$

For the covariance matrix, we recall that $\hat{\sigma}^2 = 180.90$. Hence,

$$\mathbb{V}\left[\hat{\boldsymbol{\beta}}\right] = \hat{\sigma}^2\left[\frac{\mathbf{v}_1\mathbf{v}_1^\top}{\lambda_{11}} + \frac{\mathbf{v}_2\mathbf{v}_2^\top}{\lambda_{22}}\right] = \begin{bmatrix} 109.20 & -1.87 \\ -1.87 & 0.05 \end{bmatrix}$$

If $\mathbf{X}$ is not full column rank, then the least squares estimator is not unique. One could add any vector $\alpha\mathbf{v}_i$ to the least squares estimator and this would produce another least squares estimator. However, the least squares estimator forumula that we derived stands out because it has the smallest norm.

8.4.2.3 The Inverse

A third application of SVD is in finding the inverse. Let $\mathbf{A}$ be an invertible matrix. Byu definition, $\mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$. Using SVD of $\mathbf{A}$, this may also be written as $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\mathbf{A}^{-1} = \mathbf{I}$. Multiplying both sides by $\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^\top$ yields

$$\begin{aligned}\mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^\top\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top\mathbf{A}^{-1} &= \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^\top\mathbf{I} \Leftrightarrow \\ \mathbf{A}^{-1} &= \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^\top\end{aligned}$$

SVD yields all the necessary ingredients. All we have to do is take the appropriate transposes and invert $\boldsymbol{\Sigma}$. The latter step is easy since $\boldsymbol{\Sigma}$ is a diagonal matrix.

8.4.3 Computational Aspects

In practice, most matrix operations such as inverses are carried out on computers. The precision with which these operations are carried out depends greatly on the relative sizes of the singular values of the matrix. In this context, a key concept is the **condition number**, which gives the ratio of the largest and smallest singular values:

$$c = \frac{\max \sigma}{\min \sigma}$$

In general, we do not want this number to be too large. This would indicate the presence of small singular values and that could cause computational mistakes.

For instance, if $1/c$ approaches a computer's **floating-point precision**, which is in the order of 10^{-13}, then numerical routines for matrix inversion become unreliable. In this regard, the condition number is a better diagnostic than the determinant, which may be non-zero even in the presence of very small singular values.

Small singular values also create other problems. In least squares estimation, for example, an error of order $\mathcal{O}(\hat{\sigma})$ will trigger an error in $\hat{\boldsymbol{\beta}}$ of order $\mathcal{O}(\hat{\sigma}/\min \sigma)$. Here, $\hat{\sigma} = \sqrt{\hat{\sigma}^2}$ is the root mean squared error (RMSE) of the regression. This error can be considerable when $\hat{\sigma} \neq 0$ and $\min \sigma \to 0$. In this case, $\mathbb{V}\left[\hat{\boldsymbol{\beta}}\right]$ will also be high, suggesting a great deal of imprecision.

Chapter 9

Matrices in R

R is now the statistical package of choice for many quantitative social scientists. Its assets include great numerical accuracy and enormous flexibility. Matrices are "second nature" to R. Indeed, the default R object is a column vector. It should be no surprise, then, that R is well-suited to performing matrix operations. Here, I have used R version 3.6.1.

9.1 Generating Matrices

The two key procedures for creating matrices in R are to enter them ny hand or to generate them from a data frame.

9.1.1 Entering Matrices by Hand

Entering matrices by hand is easy if the elements are all the same. Consider, for example, a 4×3 matrix consisting of 1s. We enter this via

```
foo <- matrix(1, nrow = 4, ncol = 3)
foo
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    1    1    1
## [3,]    1    1    1
## [4,]    1    1    1
```

If the elements of the matrix are different, then we need to include them using the c() operator. For example, imagine we want to specify

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix}$$

Now the required syntax is

```
A <- matrix(c(1, 2, 3, 4, 5, 6, 7, 8),
            nrow = 2, ncol = 4, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    5    6    7    8
```

The byrow option is important because it ensures that the right values appear in the right places. Leaving this option out results in

```
matrix(c(1, 2, 3, 4, 5, 6, 7, 8), nrow = 2, ncol = 4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    3    5    7
## [2,]    2    4    6    8
```

and this is not the matrix that we want.

Diagonal matrices can also be constructed easily. For instance, the diagonal matrix with elements 1, 2, and 3 can be constructed using

```
diag(c(1, 2, 3))
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    0
## [2,]    0    2    0
## [3,]    0    0    3
```

To generate an identity matrix, we simply specify the number of elements in the parentheses of the diag function. For example,

```
diag(4)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    0    0    0
## [2,]    0    1    0    0
## [3,]    0    0    1    0
```

```
## [4,]    0    0    0    1
```

A vector can be specified using the c() operator. For example,

```
u <- c(1, 2, 3)
u
```

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

9.1.2 Transforming Data Frames

The data frame is probably the best-known data object in R. Although most statistical packages play well with data frames, some require the data to be contained in matrices. It is straightforward to move from data frames to matrices: the `as.matrix` command should do the trick.

Let us consider a data set that is often used to illustrate machine learning, to wit Longley's data on UK employment. This data can be retrieved using the `mlbench` package.

```
library(mlbench)
data(longley)
X <- as.matrix(longley)
X[1:3,1:5]
```

```
##      GNP.deflator     GNP Unemployed Armed.Forces Population
## 1947         83.0 234.289      235.6        159.0    107.608
## 1948         88.5 259.426      232.5        145.6    108.632
## 1949         88.2 258.054      368.2        161.6    109.773
```

Imagine, we plan on doing a regression by hand. We may want to generate a vector with the dependent variable, `Employed`. This is easy:

```
y <- as.matrix(longley$Employed)
y[1:3,]
```

```
## [1] 60.323 61.122 60.171
```

Imagine we want `GNP` and `Armed.Forces` to be our predictors, along with a constant. We accomplish this as follows:

```
X <- cbind(1, longley$GNP, longley$Armed.Forces)
X[1:3,]
```

```
##      [,1]    [,2]  [,3]
## [1,]    1 234.289 159.0
## [2,]    1 259.426 145.6
## [3,]    1 258.054 161.6
```

This shows how one can also use `cbind` to gather variables from a data frame and render them as a matrix.

9.1.3 Sparse Matrices

Consider a 1000×1000 identity matrix. This is a sparse matrix: 0.1% of the elements are non-zero. Storing a matrix of this size can take up a lot of memory:

```
M1<- diag(1000)
object.size(M1)
```

```
## 8000216 bytes
```

We can store this more compactly using the slam package.

```
library(slam)
M2 <- as.simple_triplet_matrix(M1)
object.size(M2)
```

```
## 17152 bytes
```

This takes up less than 1% of the memory (RAM) than the default object.

9.2 Matrix Operations

9.2.1 The Transpose

Imagine we want to take the transpose of the matrix X that we created earlier. This can be done using the `t()` operator:

```
X.T <- t(X)
```

9.2.2 Tracing a Matrix

To compute the trace of a matrix, we implement the following function:

```
trace <- function(X) {
  sum(diag(X))
}
```

Here `X` is a conformable matrix. For instance,

```
A <- matrix(c(1, 2, 3, 4), nrow = 2, ncol = 2, byrow = TRUE)
A
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

```
trace(A)
```

```
## [1] 5
```

9.2.3 Matrix Addition and Subtraction

Addition and subtraction of matrices can be obtained using the `+` and `-` operators. Should a conformability issue arise, then R lets us know. For example,

```
B <- diag(2)
A+B
```

```
##      [,1] [,2]
## [1,]    2    2
## [2,]    3    5
```

```
A-B
```

```
##      [,1] [,2]
## [1,]    0    2
## [2,]    3    3
```

9.2.4 Matrix Multiplication

9.2.4.1 Scalar Multiplication

Consider the vector `y` that we created for the Longley data. The original data show employment as a percentage. Imagine we want to change to employment rates per 1000 instead of 100. Then we should multiply the entries by 10:

```
y.star <- 10*y
y.star[1:3,]
```

```
## [1] 603.23 611.22 601.71
```

9.2.4.2 Dot Products

Imagine we want to create an average employment rate for the Longley data. The first step toward this goal is to sum over the elements in the vector `y`. To that end, we create a vector of 1s that is as long as `y`. We then take the inner-product:

```
one <- rep(1, length(y))
inner <- sum(y*one)
inner
```

```
## [1] 1045.072
```

The `rep` command in the first line creates as many replicates of 1 as there are elements in `y`. To get to the mean, we now divide inner by the sample size:

```
inner/length(y)
```

```
## [1] 65.317
```

9.2.4.3 Outer-Products

Let us construct the outer-product between two vectors with 3 elements each. The easiest way to do this is to use the `outer` command:

```
u <- c(1,2,3)
v <- c(1,0,1)
C <- outer(u,v)
C
```

```
##      [,1] [,2] [,3]
## [1,]    1    0    1
## [2,]    2    0    2
## [3,]    3    0    3
```

9.2.4.4 Matrix Products

Earlier, we created a matrix of predictors from the longley data. Imagine, we want to create the product $\mathbf{X}^\top\mathbf{X}$. This can be done as follows

```
t(X)%*%X
```

```
##           [,1]         [,2]      [,3]
## [1,]   16.000    6203.175    4170.7
## [2,] 6203.175 2553151.560 1663294.5
## [3,] 4170.700 1663294.516 1159816.8
```

This is the matrix of cross-products and sums-of-squares.

9.2.4.5 Kronecker Product

Kronecker products can be obtained using the `kronecker` function:

```
A <- matrix(c(1,2,3,4), nrow = 2, ncol = 2, byrow = TRUE)
A
```

```
##      [,1] [,2]
## [1,]    1    2
## [2,]    3    4
```

```
B <- matrix(c(0,1,1,0), nrow = 2, ncol = 2, byrow = TRUE)
B
```

```
##      [,1] [,2]
## [1,]    0    1
## [2,]    1    0
```

```
kronecker(A,B)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    0    1    0    2
## [2,]    1    0    2    0
```

```
## [3,]    0    3    0    4
## [4,]    3    0    4    0
```

9.2.4.6 Hadamard Product

Hadamard products can be obtained using the `matrixcalc` package. For the matrices `A` and `B` that we just created, the Hadamard product is

```
library(matrixcalc)
hadamard.prod(A,B)
```

```
##      [,1] [,2]
## [1,]    0    2
## [2,]    3    0
```

9.2.4.7 Frobenius Inner-Product

The `matrixcalc` package can also be used to compute the Frobenius inner-product. For example,

```
frobenius.prod(A,B)
```

```
## [1] 5
```

We see this is equal to the sum of the elements of the Hadamard product.

9.2.5 Matrix Inverses

9.2.5.1 Determinants and Ranks

To determine the determinant of a square matrix, we use the `det` function. Taking the matrix `A` from above as our example, we get

```
det(A)
```

```
## [1] -2
```

The matrix rank can be obtained using the `matrixcalc` package:

```
library(matrixcalc)
matrix.rank(A)
```

```
## [1] 2
```

9.2.5.2 Inverses of Regular Matrices

The basic R command for inverses is `solve`. This is not as precise as some of the decomposition methods but if the condition number of the matrix is good, then you should not notice a difference. For the matrix `A` from before, we get

```
A.INV <- solve(A)
A.INV
```

```
##      [,1] [,2]
## [1,] -2.0  1.0
## [2,]  1.5 -0.5
```

We can check the result by verifying that the product of the inverse with the matrix is an identity matrix:

```
A.INV%*%A
```

```
##      [,1]          [,2]
## [1,]    1 4.440892e-16
## [2,]    0 1.000000e+00
```

The upper diagonal is not exactly 0 but sufficiently close that we can say that the result is an identity matrix.

9.2.5.3 Generalized Inverses

The reflexive generalized-inverse can be obtained with the following function, which also computes the usual inverse:[1]

```
inverseAnyMatrix <- function(matrix){
  # create matrix with 0 being the size of the longer side
  ncols = ncol(matrix)
  nrows = nrow(matrix)
  nmax = max(ncols, nrows)
  result = matrix(0, nrow = nmax, ncol = nmax)

  # figure out rank and which cols a unique
```

[1] I would like to thank Benjamin Schlegel for writing this function.

```
  q = qr(matrix)
  rank = q$rank
  cols = q$pivot[1:rank]

  # get full ranked matrix and inverse it
  m.full = matrix[1:rank, cols]
  result[1:rank, 1:rank] = solve(m.full)
  result
}
```

The following illustrates the function.

```
A <- matrix(c(1,-1,3,1,1,2,3,-1,8), nrow = 3, ncol = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1   -1    3
## [2,]    1    1    2
## [3,]    3   -1    8
```

```
det(A)
```

```
## [1] -8.881784e-16
```

```
inverseAnyMatrix(A)
```

```
##      [,1] [,2] [,3]
## [1,]  0.5  0.5    0
## [2,] -0.5  0.5    0
## [3,]  0.0  0.0    0
```

The MASS package in R also has a routine for the generalized inverse. However, this computes the Moore-Penrose inverse, which makes additional assumptions (see Chapter 3).

```
library(MASS)
M <- ginv(A)
M
```

```
##             [,1]       [,2]        [,3]
## [1,] -0.05555556 0.13888889  0.02777778
## [2,] -0.32222222 0.60555556 -0.03888889
## [3,]  0.02222222 0.04444444  0.08888889
```

This matrix satisfies several conditions. First, $\mathbf{AMA} = \mathbf{A}$:

```
A%*%M%*%A
```

```
##      [,1] [,2] [,3]
## [1,]    1   -1    3
## [2,]    1    1    2
## [3,]    3   -1    8
```

Second, $\mathbf{MAM} = \mathbf{M}$:

```
M%*%A%*%M
```

```
##             [,1]       [,2]        [,3]
## [1,] -0.05555556 0.13888889  0.02777778
## [2,] -0.32222222 0.60555556 -0.03888889
## [3,]  0.02222222 0.04444444  0.08888889
```

Finally, $\mathbf{AM}$ and $\mathbf{MA}$ are symmetric matrices:

```
isSymmetric(A%*%M)
```

```
## [1] TRUE
```

```
isSymmetric(M%*%A)
```

```
## [1] TRUE
```

9.3 Systems of Equations

Consider the following system of equations

$$\begin{aligned} a + b - 2c + d &= -1 \\ a - b - c + 3d &= 3 \\ 1 + 6b + c + d &= -2 \\ b + c + d &= 2 \end{aligned}$$

We place the outcomes in a vector $\mathbf{y}$:

```
y <- c(-1,3,-2,2)
```

We place the coefficients into the matrix $\mathbf{X}$:

```
X <- matrix(c(3,1,-2,1,1,-1,-1,3,1,6,1,1,0,1,1,1),
            nrow = 4, ncol = 4, byrow = TRUE)
```

We now make sure that $\mathbf{X}$ is full rank:

```
library(matrixcalc)
matrix.rank(X)
```

```
## [1] 4
```

Since $\mathbf{X}$ is full rank, the solution to the system is given by $\mathbf{X}^{-1}\mathbf{y}$:

```
solve(X)%*%y
```

```
##      [,1]
## [1,]    1
## [2,]   -1
## [3,]    2
## [4,]    1
```

Thus, $a = 1$, $b = -1$, $c = 2$, and $d = 1$.

9.4 Vector Geometry

9.4.1 Vector Norms

Consider the data from Example @ref{exm:dutch}. We focus on two issues: redistribution and the importance of religious principles. We place the party positions on those issues into vectors $\mathbf{u}$ and $\mathbf{v}$, respectively (the order of the parties is the same as in Table 7.1).

```
u <- c(4.3, 6.3, 4.1, 5.4, 3.7, 3.8, 2.7, 5.0, 6.2, 1.7, 7.4)
v <- c(2.3, 6.9, 8.4, 1.3, 1.7, 2.7, 1.3, 3.6, 9.9, 2.4, 2.1)
```

The L_2 norms of the vectors are

```
norm(u, type="2")
```

```
## [1] 16.14497
```

```
norm(v, type="2")
```

```
## [1] 16.06736
```

Since the L_2 norm relates to variation, we can say that Dutch party position show roughly equal amounts of spread on both issues.

9.4.2 Vector Angles

The following function returns the angle between two vectors:

```
vect.angle <- function(x, y, degrees) {
  q <- (x%*%y)/(norm(x, type = "2")*norm(y, type = "2"))
  angle <- acos(q)
  if(degrees) {
    r <- angle*(180/pi)
    return(r)
  }
  else {return(angle)}
}
```

Applying the function to the two vectors of issue positions, we obtain:

```
vect.angle(u,v,degrees=FALSE)
```

```
##           [,1]
## [1,] 0.6014855
```

in radians. To obtain the results in degrees, we issue

```
vect.angle(u,v,degrees=TRUE)
```

```
##          [,1]
## [1,] 34.46258
```

9.5 Eigenvalues and Eigenvectors

9.5.1 Computation

Consider the following symmetric matrix

```
A <- matrix(c(3,4,4,3), nrow = 2, ncol = 2, byrow = TRUE)
A
```

```
##      [,1] [,2]
## [1,]    3    4
## [2,]    4    3
```

We now use the `eigen` function to obtain the eigenvalues and eigenvectors of this matrix:

```
ev <- eigen(A)
ev
```

```
## eigen() decomposition
## $values
## [1]  7 -1
##
## $vectors
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

To create a diagonal matrix of eigenvalues, we do

```
L <- diag(ev$values)
L
```

```
##      [,1] [,2]
## [1,]    7    0
## [2,]    0   -1
```

The eigenvectors can be turned into a matrix:

```
C <- ev$vectors
C
```

```
##           [,1]       [,2]
## [1,] 0.7071068 -0.7071068
## [2,] 0.7071068  0.7071068
```

Since $\mathbf{A}$ is symmetric, the diagonalization theorem implies $\mathbf{C}^\top\mathbf{AC} = \mathbf{L}$:

```
C%*%A%*%t(C)
```

```
##      [,1] [,2]
## [1,]   -1    0
## [2,]    0    7
```

The spectral decomposition theorem implies $\mathbf{CLC}^\top = \mathbf{A}$:

```
C%*%L%*%t(C)
```

```
##      [,1] [,2]
## [1,]    3    4
## [2,]    4    3
```

9.5.2 Principal Component Analysis

In Chapter 7, we discussed a principal component analysis of Dutch party positions on eight issues.

```
dutch <- data.frame(party=c("50Plus", "CDA", "CU", "D66", "GroenLinks", "PvdA", "P
                            "PVV", "SGP", "SP", "VVD"),
                    A=c(3.3, 6.7, 4.7, 5.9, 2.8, 4.1, 2.8, 4.4, 6.3, 1.3, 7.8),
                    B=c(4.3, 6.3, 4.1, 5.4, 3.7, 3.8, 2.7, 5.0, 6.2, 1.7, 7.4),
                    C=c(4.5, 7.6, 6.5, 2.7, 3.5, 3.8, 3.2, 5.0, 7.6, 4.3, 4.0),
                    D=c(5.0, 6.5, 3.8, 2.1, 1.1, 4.1, 2.3, 9.9, 8.4, 4.4, 7.5),
                    E=c(6.0, 6.9, 5.8, 2.1, 1.8, 4.3, 2.8, 9.3, 7.6, 4.0, 7.8),
                    F=c(3.0, 5.4, 7.2, 0.4, 0.9, 1.9, 2.0, 3.7, 9.1, 3.1, 2.2),
                    G=c(2.3, 6.9, 8.4, 1.3, 1.7, 2.7, 1.3, 3.6, 9.9, 2.4, 2.1),
                    H=c(5.3, 6.3, 3.8, 4.0, 1.3, 4.8, 0.9, 8.2, 6.0, 4.9, 7.3))
head(dutch, n=11)
```

party	A	B	C	D	E	F	G	H
50Plus	3.3	4.3	4.5	5.0	6.0	3.0	2.3	5.3
CDA	6.7	6.3	7.6	6.5	6.9	5.4	6.9	6.3
CU	4.7	4.1	6.5	3.8	5.8	7.2	8.4	3.8
D66	5.9	5.4	2.7	2.1	2.1	0.4	1.3	4.0
GroenLinks	2.8	3.7	3.5	1.1	1.8	0.9	1.7	1.3
PvdA	4.1	3.8	3.8	4.1	4.3	1.9	2.7	4.8
PvdD	2.8	2.7	3.2	2.3	2.8	2.0	1.3	0.9
PVV	4.4	5.0	5.0	9.9	9.3	3.7	3.6	8.2
SGP	6.3	6.2	7.6	8.4	7.6	9.1	9.9	6.0
SP	1.3	1.7	4.3	4.4	4.0	3.1	2.4	4.9
VVD	7.8	7.4	4.0	7.5	7.8	2.2	2.1	7.3

We can use the `pca` function to extract the principal components directly from the raw data:

```
dutch.pca <- princomp(dutch[,2:9])
dutch.pca
```

```
## Call:
## princomp(x = dutch[, 2:9])
##
## Standard deviations:
##    Comp.1    Comp.2    Comp.3    Comp.4    Comp.5    Comp.6    Comp.7
## 5.2151982 3.0043809 1.9289822 0.7229157 0.5027918 0.4338732 0.2804293
##    Comp.8
## 0.2550089
##
##  8  variables and  11 observations.
```

The output shows the singular values, i.e., the square roots of the eigenvalues. To obtain the **loadings** or eigenvectors, we issue

```
dutch.pca$loadings
```

```
##
## Loadings:
##   Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## A  0.217  0.173  0.718         0.154  0.152  0.149  0.580
## B  0.191  0.203  0.546  0.154 -0.131 -0.211 -0.360 -0.640
## C  0.273 -0.221        -0.198 -0.664 -0.510 -0.143  0.340
## D  0.441  0.378 -0.302  0.278  0.399 -0.551  0.166
## E  0.410  0.281 -0.220  0.392 -0.499  0.523  0.163
## F  0.414 -0.438 -0.128  0.175  0.304  0.230 -0.651  0.146
## G  0.459 -0.538  0.124 -0.206  0.110         0.565 -0.330
## H  0.311  0.420 -0.121 -0.794         0.203 -0.176
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8
## SS loadings     1.000  1.000  1.000  1.000  1.000  1.000  1.000  1.000
## Proportion Var  0.125  0.125  0.125  0.125  0.125  0.125  0.125  0.125
## Cumulative Var  0.125  0.250  0.375  0.500  0.625  0.750  0.875  1.000
```

Note that small loadings are not shown.

9.6 Matrix Factorization

9.6.1 LU Decomposition

In Chapter 8, we used LU decomposition on

```
A <- matrix(c(1,2,3,4,5,4,3,2,1),
            nrow = 3, ncol = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    4
## [3,]    3    2    1
```

It was quite a bit of work obtaining the lower-triangular matrix **L** and the upper-traingular matrix **U** by hand. Using the `matrixcalc` package, R will do the work for us:

```
library(matrixcalc)
lu.decomposition(A)
```

```
## $L
##      [,1]     [,2] [,3]
## [1,]    1 0.000000    0
## [2,]    4 1.000000    0
## [3,]    3 1.333333    1
##
## $U
##      [,1] [,2]      [,3]
## [1,]    1    2  3.000000
## [2,]    0   -3 -8.000000
## [3,]    0    0  2.666667
```

9.6.2 Cholesky Decomposition

9.6.2.1 Computation

In Chapter 8, we also performed cholesky decomposition on

```
A <- matrix(c(49,14,7,-14,14,85,-16,5,7,-16,105,
              -34,-14,5,-34,158),
            nrow = 4, ncol = 4, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   49   14    7  -14
## [2,]   14   85  -16    5
## [3,]    7  -16  105  -34
## [4,]  -14    5  -34  158
```

Cholesky decomposition requires that the matrix is symmetric and positive definite:

```
isSymmetric(A)
```

```
## [1] TRUE
```

```
is.positive.definite(A)
```

```
## [1] TRUE
```

For the Cholesky decomposition, we can now use the R `base` function `chol`:

```
U <- chol(A)
U
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7    2    1   -2
## [2,]    0    9   -2    1
## [3,]    0    0   10   -3
## [4,]    0    0    0   12
```

This gives the upper-triangular matrix $\mathbf{L}^\top$ from the decomposition $\mathbf{A} = \mathbf{L}\mathbf{L}^\top$. To get $\mathbf{L}$, we perform

```
L <- t(U)
L
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    7    0    0    0
## [2,]    2    9    0    0
## [3,]    1   -2   10    0
## [4,]   -2    1   -3   12
```

9.6.2.2 Application to Regression Analysis

Cholesky decomposition has several important applications, as we saw in Chapter 8. Consider first regression analysis. Imagine we want to regress employment on GNP and armed forces using the Longley data. We define

```
library(mlbench)
data(longley)
y <- as.matrix(longley$Employed, ncol = 1)
X <- cbind(1, longley$GNP, longley$Armed.Forces)
```

The normal equations for the least squares estimator are $\mathbf{X}^\top\mathbf{X}\boldsymbol{\beta} = \mathbf{X}^\top\mathbf{y}$, which may also be written as $\mathbf{A}\boldsymbol{\beta} = \mathbf{c}$. We now proceed in the following steps. First, we generate $\mathbf{A}$ and apply the Cholesky decomposition:

```
A <- crossprod(X)
L <- t(chol(A))
L
```

```
##           [,1]     [,2]     [,3]
## [1,]     4.000   0.0000   0.0000
## [2,] 1550.794 384.9549   0.0000
## [3,] 1042.675 120.3274 241.1782
```

Next, writing the normal equations as $\mathbf{L}\mathbf{L}^\top\boldsymbol{\beta} = \mathbf{c}$ and letting $\mathbf{z} = \mathbf{L}^\top\boldsymbol{\beta}$, we obtain the equations $\mathbf{L}\mathbf{z} = \mathbf{c}$. We first create $\mathbf{c}$:

```
c <- crossprod(X,y)
```

We then compute $\mathbf{L}^{-1}\mathbf{L}\mathbf{z} = \mathbf{L}^{-1}\mathbf{c}$ to obtain a solution for $\mathbf{z}$:

```
z <- solve(L)%*%c
z
```

```
##              [,1]
## [1,] 261.2680000
## [2,]  13.3780673
## [3,]   0.2768617
```

Third, with $\mathbf{z} = \mathbf{L}^\top\boldsymbol{\beta}$ it suffices to pre-multiply $\mathbf{z}$ with the inverse of $\mathbf{L}^\top$ to obtain $\boldsymbol{\beta}$:

```
beta <- solve(t(L))%*%z
beta
```

```
##               [,1]
## [1,] 51.683468731
## [2,]  0.034393472
## [3,]  0.001147955
```

9.6.2.3 Application to the Multivariate Normal Distribution

Another application of Cholesky decomposition is the construction of random draws from an arbitrary multivariate normal distribution. Imagine, we want to simulate 10000 draws from a bivariate normal distribution with mean vector $\boldsymbol{\mu}^\top = (-1, 1)$, variances of 1, and a correlation of 0.5. We begin by generating the standard normal variates that will form the basis of the intended bivariate distribution. Since we rely on the random number generator, replicability requires that we set a seed.

```
set.seed(9817,
         kind = "Mersenne-Twister",
         normal.kind = "Inversion")
z1 <- rnorm(10000, mean = 0, sd = 1)
z2 <- rnorm(10000, mean = 0, sd = 1)
Z <- cbind(z1,z2)
head(Z)
```

```
##                z1          z2
## [1,]  0.04292445 -0.41743364
## [2,]  0.63744164 -0.65349870
## [3,]  0.46773360  0.04305993
## [4,]  0.35463896 -0.21809327
## [5,] -0.43687090  0.63406857
## [6,] -0.52349484  0.56715501
```

Next, we specify the mean and covariance structures:

```
mu <- c(-1,1)
Sigma <- matrix(c(1,0.5,0.5,1), nrow = 2, ncol = 2, byrow = TRUE)
```

We now perform a Cholesky decomposition on `Sigma`:

```
L <- t(chol(Sigma))
L
```

```
##      [,1]      [,2]
```

```
## [1,]  1.0 0.0000000
## [2,]  0.5 0.8660254
```

We now create new variables $\mathbf{X} = \mathbf{ZL} + \boldsymbol{\iota}\boldsymbol{\mu}^\top$, where $\boldsymbol{\iota}$ is a vector of 1s:

```
iota <- matrix(1, nrow = length(z1), ncol = 1)
X <- Z%*%L + iota%*%t(mu)
head(X)
```

```
##             [,1]      [,2]
## [1,] -1.1657924 0.6384919
## [2,] -0.6893077 0.4340535
## [3,] -0.5107364 1.0372910
## [4,] -0.7544077 0.8111257
## [5,] -1.1198366 1.5491195
## [6,] -1.2399173 1.4911706
```

Finally, we check that the resulting random numbers behave more or less as expected:

```
colMeans(X)
```

```
## [1] -0.9991512  1.0051428
```

```
cor(X[,1],X[,2])
```

```
## [1] 0.4493667
```

9.7 QR Decomposition

R's `base` library has a command for QR decomposition. We illustrate it for the following matrix:

```
A <- matrix(c(12,-51,4,6,167,-68,-4,24,-41),
            nrow = 3, ncol = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]   12  -51    4
## [2,]    6  167  -68
## [3,]   -4   24  -41
```

The QR decomposition yields:

```
QR <- qr(A)
QR
```

```
## $qr
##              [,1]          [,2] [,3]
## [1,] -14.0000000  -21.0000000   14
## [2,]   0.4285714 -175.0000000   70
## [3,]  -0.2857143    0.1107692  -35
##
## $rank
## [1] 3
##
## $qraux
## [1]  1.857143  1.993846 35.000000
##
## $pivot
## [1] 1 2 3
##
## attr(,"class")
## [1] "qr"
```

The `qr` bit of the output folds the **Q** and **R** matrices together. To retrieve those matrices, we issue the following syntax:

```
qr.Q(QR)
```

```
##            [,1]       [,2]        [,3]
## [1,] -0.8571429  0.3942857  0.33142857
## [2,] -0.4285714 -0.9028571 -0.03428571
## [3,]  0.2857143 -0.1714286  0.94285714
```

```
qr.R(QR)
```

```
##      [,1] [,2] [,3]
## [1,]  -14  -21   14
## [2,]    0 -175   70
## [3,]    0    0  -35
```

The algorithm uses Householder reflections or pivoting.

An alternative approach is to use the `householder` function in the library `pracma`:

```
library(pracma)
householder(A)
```

```
## $Q
##             [,1]       [,2]        [,3]
## [1,] -0.8571429  0.3942857  0.33142857
## [2,] -0.4285714 -0.9028571 -0.03428571
## [3,]  0.2857143 -0.1714286  0.94285714
##
## $R
##                [,1]          [,2] [,3]
## [1,] -1.400000e+01 -2.100000e+01   14
## [2,]  5.516954e-17 -1.750000e+02   70
## [3,]  6.148928e-18 -3.552714e-15  -35
```

This directly produces readable output. The `pracma` package also lets you do Gram-Schmidt orthogonalization:

```
library(pracma)
gramSchmidt(A)
```

```
## $Q
##             [,1]       [,2]        [,3]
## [1,]  0.8571429 -0.3942857 -0.33142857
## [2,]  0.4285714  0.9028571  0.03428571
## [3,] -0.2857143  0.1714286 -0.94285714
##
## $R
##      [,1] [,2] [,3]
## [1,]   14   21  -14
## [2,]    0  175  -70
## [3,]    0    0   35
```

The `lm` command relies on QR decomposition. Let us run the regression on the Longley data:

```
lm.fit <- lm(Employed~GNP+Armed.Forces,
             data = longley)
lm.fit$qr
```

```
## $qr
##      (Intercept)           GNP  Armed.Forces
```

```
## 1947       -4.00 -1.550794e+03 -1.042675e+03
## 1948        0.25  3.849549e+02  1.203274e+02
## 1949        0.25  2.570757e-01  2.411782e+02
## 1950        0.25  1.881196e-01  1.785390e-01
## 1951        0.25  7.284372e-02 -3.402573e-01
## 1952        0.25  2.602266e-02 -5.121925e-01
## 1953        0.25 -2.173878e-02 -4.587285e-01
## 1954        0.25 -1.583419e-02 -3.812466e-01
## 1955        0.25 -1.050836e-01 -1.925382e-01
## 1956        0.25 -1.614824e-01 -7.322300e-02
## 1957        0.25 -2.227597e-01 -5.168666e-03
## 1958        0.25 -2.273758e-01  6.487075e-02
## 1959        0.25 -3.264991e-01  1.706282e-01
## 1960        0.25 -3.781857e-01  2.231526e-01
## 1961        0.25 -4.186372e-01  2.278802e-01
## 1962        0.25 -5.140276e-01  1.900075e-01
## attr(,"assign")
## [1] 0 1 2
##
## $qraux
## [1] 1.250000 1.253512 1.143583
##
## $pivot
## [1] 1 2 3
##
## $tol
## [1] 1e-07
##
## $rank
## [1] 3
##
## attr(,"class")
## [1] "qr"
```

9.8 Singular Value Decomposition

The `base` library contains the tool to perform SVD. Let us create the matrix of predictors for the Longley data and then apply SVD:

```
X <- cbind(1, longley$GNP, longley$Armed.Forces)
svd(X)
```

```
## $d
## [1] 1913.0604599  230.6150432    0.8417664
##
## $u
##              [,1]        [,2]        [,3]
##  [1,] -0.1480020  0.01127000 -0.49888657
##  [2,] -0.1550616 -0.09748464 -0.48643764
##  [3,] -0.1590973 -0.03642611 -0.45629646
##  [4,] -0.1716342 -0.08791153 -0.40772331
##  [5,] -0.2329023  0.32861551 -0.04550582
##  [6,] -0.2550791  0.46402695  0.08273405
##  [7,] -0.2617198  0.40289659  0.10213665
##  [8,] -0.2550265  0.33723552  0.05879942
##  [9,] -0.2612342  0.14568361  0.05182384
## [10,] -0.2651524  0.02458001  0.04738370
## [11,] -0.2737100 -0.05337998  0.07253948
## [12,] -0.2698216 -0.11577231  0.04283281
## [13,] -0.2839668 -0.23811296  0.08563253
## [14,] -0.2915257 -0.29962351  0.10922607
## [15,] -0.2999820 -0.31608775  0.14540194
## [16,] -0.3233465 -0.31223033  0.25457483
##
## $v
##               [,1]          [,2]         [,3]
## [1,] -0.002042414  0.0006819983 -0.999997682
## [2,] -0.832563259 -0.5539283986  0.001322664
## [3,] -0.553926212  0.8325640300  0.001699158
```

In the output, `d` contains the diagonal elements from $\mathbf{\Sigma}$. Further, `u` captures $\mathbf{U}$ and `v` captures $\mathbf{V}$.

The `pracma` package has an easy way of computing the condition number. For instance, for the matrix of predictors we get

```
library(pracma)
cond(X)
```

```
## [1] 2272.674
```

Ideally, the condition number should be close to 1. That is certainly not the case here, so that numeric precision in least squares estimation could be somewhat of an issue.

References

Butler, David, and Donald Stokes. 1971. *Political Change in Britain.* New York: St. Marin's Press.