Matrix Algebra in R

Andrew Zieffler

To enter a matrix in R, use the `matrix()` function.

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 3 \\ 4 & -5 & -6 \\ 7 & 8 & 9 \\ 0 & 0 & 10 \end{pmatrix}$$

```
> X = matrix(
    data = c(1, 4, 7, 0, -2, -5, 8, 0, 3, -6, 9, 10),
    nrow = 4,
    ncol = 3
    )

> X

     [,1] [,2] [,3]
[1,]    1   -2    3
[2,]    4   -5   -6
[3,]    7    8    9
[4,]    0    0   10
```

By default, elements are filled in columns.

The `byrow=TRUE` argument will fill the elements by rows rather than columns.

$$\mathbf{Y} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

```
> Y = matrix(
    data = c(-5, 1, 3, 2, 2, 6, 7, 3, -4),
    byrow = TRUE,
    nrow = 4
    )

> Y

     [,1] [,2] [,3]
[1,]   -5    1    3
[2,]    2    2    6
[3,]    7    3   -4
```

Some vocabulary:

- Scalar
- Dimension
- Transpose
- Determinant
- Diagonal
- Trace
- Vector
- Dot product

- Square matrix
- Identity matrix
- Inverse
- Idempotent matrix

- Decomposition
- Rank

The `dim()` function will return the dimensions of a matrix.

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 3 \\ 4 & -5 & -6 \\ 7 & 8 & 9 \\ 0 & 0 & 10 \end{pmatrix} \qquad \mathbf{Y} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

```
> dim(X)
[1] 4 3

> dim(Y)
[1] 3 3
```

Matrix **Y** is referred to as a *square matrix*—it has the same number of rows and columns.

The `t()` function will return the transpose of a matrix.

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 3 \\ 4 & -5 & -6 \\ 7 & 8 & 9 \\ 0 & 0 & 10 \end{pmatrix} \qquad \mathbf{Y} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

```
> t(X)

     [,1] [,2] [,3] [,4]
[1,]    1    4    7    0
[2,]   -2   -5    8    0
[3,]    3   -6    9   10

> t(Y)

     [,1] [,2] [,3]
[1,]   -5    2    7
[2,]    1    2    3
[3,]    3    6   -4
```

The `det()` function will return the determinant of a matrix.

$$\mathbf{X} = \begin{pmatrix} 1 & -2 & 3 \\ 4 & -5 & -6 \\ 7 & 8 & 9 \\ 0 & 0 & 10 \end{pmatrix} \qquad \mathbf{Y} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

```
> det(Y)
[1] 156

> det(X)
Error in determinant.matrix(x, logarithm = TRUE, ...) :
  'x' must be a square matrix
```

In order to compute a determinant, the matrix must be square.

The diag() function will return the main diagonal of a matrix. In order to compute the diagonal, the matrix must be square.

$$\mathbf{Y} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

```
> diag(Y)
[1] -5  2 -4
```

The trace of a matrix is the sum of its diagonal elements.

```
> sum(diag(Y))
[1] -7
```

The *identity matrix* is a square matrix with ones on the diagonal, and zeros everywhere else.

$$\mathbf{I} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Adding an argument that specifies the number of rows/columns to the diag() function will return the identity matrix.

```
> 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
```

We can add or subtract matrices that have the same dimensions.

$$\mathbf{A} = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} -5 & 1 & 2 \\ 3 & 0 & 4 \end{pmatrix}$$

```
> A = matrix(data = c(1, 2,  3, 4, 5,  6), byrow = TRUE, nrow = 2)
> B = matrix(data = c(-5, 1, 2, 3, 0, 4), byrow = TRUE, nrow = 2)

> A + B

     [,1] [,2] [,3]
[1,]   -4    3    5
[2,]    7    5   10
```

The sum/difference is a matrix that has the same dimensions and whose elements is the sum/difference of the corresponding elements.

Matrices can be multiplied by a scalar.

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

```
> 3 * B

     [,1] [,2] [,3]
[1,]  -15    3    6
[2,]    9    0   12
```

The product is a matrix that has the same dimensions and whose elements is the product of the corresponding element and the scalar.

Matrix multiplication is more complicated than scalar multiplication. To multiply two matrices together we use the operator `%*%`.

$$\mathbf{B} = \begin{pmatrix} -5 & 1 & 2 \\ 3 & 0 & 4 \end{pmatrix} \qquad \mathbf{Y} = \begin{pmatrix} -5 & 1 & 3 \\ 2 & 2 & 6 \\ 7 & 3 & -4 \end{pmatrix}$$

$2 \times \boxed{3}$ $\qquad$ $\boxed{3} \times 3$

```
> B %*% Y

     [,1] [,2] [,3]
[1,]   41    3  -17
[2,]   13   15   -7

> Y %*% B

Error in Y %*% B : non-conformable arguments
```

Two things: (1) Matrix multiplication is not commutative. (2) The matrices you are multiplying have to be conformable—the first matrix has to have the same number of columns as the second matrix has rows.

Because matrix multiplication is not commutative we use the terms postmultiply and premultiply. In this example, **B** is postmultiplied by **Y**.

Postmultiplying by the identity matrix gives the original matrix.

$$\mathbf{B} = \begin{pmatrix} -5 & 1 & 2 \\ 3 & 0 & 4 \end{pmatrix} \qquad \mathbf{I} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$2 \times \boxed{3} \qquad \boxed{3} \times 3$

```
> B %*% diag(3)

     [,1] [,2] [,3]
[1,]   -5    1    2
[2,]    3    0    4
```

It now might make sense why it is referred to as the identity matrix!

A vector can be input into R using the `c()` function or the `matrix()` function.

$$\mathbf{a} = \mathbf{b} = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 3 \end{pmatrix}$$

```
> a = c(2, 0, 1, 3)
> a

[1] 2 0 1 3

> b = matrix(data = c(2, 0, 1, 3), ncol = 1)
> b

     [,1]
[1,]    2
[2,]    0
[3,]    1
[4,]    3

> dim(a)
NULL

> dim(b)
[1] 4 1
```

Technically, the `c()` function produces a **column vector** and the `matrix()` function produces a **one-column matrix**. The difference is in their classes, and we can, for all intents and purposes, just use the `c()` function.

The dot product between two vectors is the sum of their multiplied elements.

$$\mathbf{a} = \begin{pmatrix} 2 \\ 0 \\ 1 \\ 3 \end{pmatrix} \qquad \mathbf{b} = \begin{pmatrix} -1 \\ 6 \\ 0 \\ 9 \end{pmatrix}$$

```
> a = c(2, 0, 1, 3)
> b = c(-1, 6, 0, 9)

> sum(a * b)
[1] 14
```

We can also compute the dot product via matrix multiplication by postmuliplying the transpose of **a** by **b.**

$$\mathbf{a} \cdot \mathbf{b} = \mathbf{a}^\intercal \mathbf{b}$$

```
> t(a) %*% b

     [,1]
[1,]   14
```

The inverse of a matrix is the matrix that you postmultiply by to get the identity matrix.

$$\begin{pmatrix} 2 & 5 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Finding the inverse in this case requires solving a system of four equations for four unknowns.

$$\begin{aligned} 2a + 5c &= 1 \\ 2b + 5d &= 0 \\ 1a + 3c &= 0 \\ 1b + 3d &= 1 \end{aligned}$$

There are many ways to do this (e.g., substitution, Gaussian elimination, Gauss–Jordan elimination), however none are pleasant when the matrices are of large dimensions.

In R, we can use the solve() function to find the inverse.

```
> D = matrix(c(2, 5, 1, 3), byrow = TRUE, nrow = 2)

> solve(D)

     [,1] [,2]
[1,]    3   -5
[2,]   -1    2
```

This works for most matrices, but there are better (faster) ways to compute an inverse. Many of these methods involve decomposing the matrix into easier things to compute on (e.g., Choleski decomposition, QR decomposition, LU decomposition).

```
# Use QR decomposition
> qr.solve(D)

     [,1] [,2]
[1,]    3   -5
[2,]   -1    2
```

If a matrix has an inverse it is referred to as *nonsingular*. The way we check if a matrix is nonsingular is to compute its determinant. If the determinant is non-zero, the matrix is nonsingular.

```
> det(D)

[1] 1
```

Since the determinant of $\mathbf{D} \neq 0$, we say $\mathbf{D}$ is nonsingular; it has an inverse.

The reason for this is that in computing the inverse of a matrix we use the determinant in the denominator of a fraction. If the determinant is zero, that fraction, and therefore the inverse, is undefined.

References and Source Material

Additional Resources

- Prasad, A. (2012). *Geometric definition of determinants*. YouTube video. https://www.youtube.com/watch?v=xX7qBVa9cQU&feature=youtu.be