

Basic matrix math in R

This chapter reviews the basic matrix math operations that you will need to understand the course material and how to do these operations in R.

1.1 Creating matrices in R

Create a 3×4 matrix, meaning 3 row and 4 columns, that is all 1s:

```
matrix(1, 3, 4)

      [,1] [,2] [,3] [,4]
[1,]    1    1    1    1
[2,]    1    1    1    1
[3,]    1    1    1    1
```

Create a 3×4 matrix filled in with the numbers 1 to 12 by column (default) and by row:

```
matrix(1:12, 3, 4)

      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

matrix(1:12, 3, 4, byrow=TRUE)

      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```

Create a matrix with one column:

```
matrix(1:4, ncol=1)
```

2 1 Matrix math

```
      [,1]
[1,]    1
[2,]    2
[3,]    3
[4,]    4
```

Create a matrix with one row:

```
matrix(1:4, nrow=1)
```

```
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
```

Check the dimensions of a matrix

```
A=matrix(1:6, 2,3)
```

```
A
```

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

```
dim(A)
```

```
[1] 2 3
```

Get the number of rows in a matrix:

```
dim(A)[1]
```

```
[1] 2
```

```
nrow(A)
```

```
[1] 2
```

Create a 3D matrix (called array):

```
A=array(1:6, dim=c(2,3,2))
```

```
A
```

```
, , 1
```

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

```
, , 2
```

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

```
dim(A)
```

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

Check if an object is a matrix. A dataframe is not a matrix. A vector is not a matrix.

```
A=matrix(1:4, 1, 4)
```

```
A
```

```
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
```

```
class(A)
```

```
[1] "matrix"
```

```
B=data.frame(A)
```

```
B
```

```
  X1 X2 X3 X4
1  1  2  3  4
```

```
class(B)
```

```
[1] "data.frame"
```

```
C=1:4
```

```
C
```

```
[1] 1 2 3 4
```

```
class(C)
```

```
[1] "integer"
```

1.2 Matrix multiplication, addition and transpose

You will need to be very solid in matrix multiplication for the course. If you haven't done it in awhile, google 'matrix multiplication youtube' and you find lots of 5min videos to remind you.

In R, you use the `%*%` operation to do matrix multiplication. When you do matrix multiplication, the columns of the matrix on the left must equal the rows of the matrix on the right. The result is a matrix that has the number of rows of the matrix on the left and number of columns of the matrix on the right.

$$(n \times m)(m \times p) = (n \times p)$$

4 1 Matrix math

```
A=matrix(1:6, 2, 3) #2 rows, 3 columns
B=matrix(1:6, 3, 2) #3 rows, 2 columns
A%%B #this works
```

```
      [,1] [,2]
[1,]   22  49
[2,]   28  64
```

```
B%%A #this works
```

```
      [,1] [,2] [,3]
[1,]    9  19  29
[2,]   12  26  40
[3,]   15  33  51
```

```
try(B%%B) #this doesn't
```

To add two matrices use +. The matrices have to have the same dimensions.

```
A+A #works
```

```
      [,1] [,2] [,3]
[1,]    2    6   10
[2,]    4    8   12
```

```
A+t(B) #works
```

```
      [,1] [,2] [,3]
[1,]    2    5    8
[2,]    6    9   12
```

```
try(A+B) #does not work since A has 2 rows and B has 3
```

The transpose of a matrix is denoted \mathbf{A}^\top or \mathbf{A}' . To transpose a matrix in R, you use t().

```
A=matrix(1:6, 2, 3) #2 rows, 3 columns
t(A) #is the transpose of A
```

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

```
try(A%%A) #this won't work
```

```
A%%t(A) #this will
```

```
      [,1] [,2]
[1,]   35  44
[2,]   44  56
```

1.3 Subsetting a matrix

To subset a matrix, we use []:

```
A=matrix(1:9, 3, 3) #3 rows, 3 columns
#get the first and second rows of A
#it's a 2x3 matrix
A[1:2,]
```

```
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
```

```
#get the top 2 rows and left 2 columns
A[1:2,1:2]
```

```
      [,1] [,2]
[1,]    1    4
[2,]    2    5
```

```
#What does this do?
A[c(1,3),c(1,3)]
```

```
      [,1] [,2]
[1,]    1    7
[2,]    3    9
```

```
#This?
A[c(1,2,1),c(2,3)]
```

```
      [,1] [,2]
[1,]    4    7
[2,]    5    8
[3,]    4    7
```

If you have used matlab, you know you can say something like `A[1,end]` to denote the element of a matrix in row 1 and the last column. R does not have 'end'. To do, the same in R you do something like:

```
A=matrix(1:9, 3, 3)
A[1,ncol(A)]
```

```
[1] 7
```

```
#or
A[1,dim(A)[2]]
```

```
[1] 7
```

Warning R will create vectors from subsetting matrices!

One of the really bad things that R does with matrices is create a vector if you happen to subset a matrix to create a matrix with 1 row or 1 column. Look at this:

```
A=matrix(1:9, 3, 3)
#take the first 2 rows
B=A[1:2,]
#everything is ok
dim(B)

[1] 2 3

class(B)

[1] "matrix"

#take the first row
B=A[1,]
#oh no! It should be a 1x3 matrix but it is not.
dim(B)

NULL

#It is not even a matrix any more
class(B)

[1] "integer"

#and what happens if we take the transpose?
#Oh no, it's a 1x3 matrix not a 3x1 (transpose of 1x3)
t(B)

      [,1] [,2] [,3]
[1,]    1    4    7

#A**B should fail because A is (3x3) and B is (1x3)
A**B

      [,1]
[1,]    66
[2,]    78
[3,]    90

#It works? That is horrible!
```

This will create hard to find bugs in your code because you will look at `B=A[1,]` and everything looks fine. Why is R saying it is not a matrix! To stop R from doing this use `drop=FALSE`.

```

B=A[1,,drop=FALSE]
#Now it is a matrix as it should be
dim(B)

[1] 1 3

class(B)

[1] "matrix"

#this fails as it should (alerting you to a problem!)
try(A%%B)

```

1.4 Replacing elements in a matrix

Replace 1 element.

```

A=matrix(1, 3, 3)
A[1,1]=2
A

```

	[,1]	[,2]	[,3]
[1,]	2	1	1
[2,]	1	1	1
[3,]	1	1	1

Replace a row with all 1s or a string of values

```

A=matrix(1, 3, 3)
A[1,]=2
A

```

	[,1]	[,2]	[,3]
[1,]	2	2	2
[2,]	1	1	1
[3,]	1	1	1

```

A[1,]=1:3
A

```

	[,1]	[,2]	[,3]
[1,]	1	2	3
[2,]	1	1	1
[3,]	1	1	1

Replace group of elements. This often does not work as one expects so be sure look at your matrix after trying something like this. Here I want to replace elements (1,3) and (3,1) with 2, but it didn't work as I wanted.

```
A=matrix(1, 3, 3)
A[c(1,3),c(3,1)]=2
A
```

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

How do I replace elements (1,1) and (3,3) with 2 then? It's tedious. If you have a lot of elements to replace, you might want to use a for loop.

```
A=matrix(1, 3, 3)
A[1,3]=2
A[3,1]=2
A
```

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

1.5 Diagonal matrices and identity matrices

A diagonal matrix is one that is square, meaning number of rows equals number of columns, and it has 0s on the off-diagonal and non-zeros on the diagonal. In R, you form a diagonal matrix with the `diag()` function:

```
diag(1,3) #put 1 on diagonal of 3x3 matrix
```

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

```
diag(2, 3) #put 2 on diagonal of 3x3 matrix
```

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

```
diag(1:4) #put 1 to 4 on diagonal of 4x4 matrix
```

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


The `diag()` function can also be used to replace elements on the diagonal of a matrix:

```
A=matrix(3, 3, 3)
diag(A)=1
A
      [,1] [,2] [,3]
[1,]    1    3    3
[2,]    3    1    3
[3,]    3    3    1
```

```
A=matrix(3, 3, 3)
diag(A)=1:3
A
      [,1] [,2] [,3]
[1,]    1    3    3
[2,]    3    2    3
[3,]    3    3    3
```

```
A=matrix(3, 3, 4)
diag(A[1:3,2:4])=1
A
      [,1] [,2] [,3] [,4]
[1,]    3    1    3    3
[2,]    3    3    1    3
[3,]    3    3    3    1
```

The `diag` function is also used to get the diagonal of a matrix.

```
A=matrix(1:9, 3, 3)
diag(A)
[1] 1 5 9
```

The identity matrix is a special kind of diagonal matrix with 1s on the diagonal. It is denoted \mathbf{I} . \mathbf{I}_3 would mean a 3×3 diagonal matrix. A identity matrix has the property that $\mathbf{AI} = \mathbf{A}$ and $\mathbf{IA} = \mathbf{A}$ so it is like a 1.

```
A=matrix(1:9, 3, 3)
I=diag(3) #shortcut for 3x3 identity matrix
A**%I
      [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

1.6 Taking the inverse of a square matrix

The inverse of a matrix is denoted \mathbf{A}^{-1} . You can think of the inverse of a matrix like $1/a$. $1/a \times a = 1$. $\mathbf{A}^{-1}\mathbf{A} = \mathbf{A}\mathbf{A}^{-1} = \mathbf{I}$. The inverse of a matrix does not always exist; for one it has to be square. We'll be using inverses for variance-covariance matrices and by definition (of a variance-covariance matrix), the inverse of those exist. In R, there are a couple way common ways to take the inverse of a variance-covariance matrix (or something with the same properties). `solve` is the most common probably:

```
A=diag(3,3)+matrix(1,3,3)
invA=solve(A)
invA%%A

      [,1]      [,2] [,3]
[1,] 1.000000e+00 -6.938894e-18  0
[2,] 2.081668e-17  1.000000e+00  0
[3,] 0.000000e+00  0.000000e+00  1

A%%invA

      [,1]      [,2] [,3]
[1,] 1.000000e+00 -6.938894e-18  0
[2,] 2.081668e-17  1.000000e+00  0
[3,] 0.000000e+00  0.000000e+00  1
```

Another option is to use `chol2inv` which uses a Cholesky decomposition¹:

```
A=diag(3,3)+matrix(1,3,3)
invA=chol2inv(chol(A))
invA%%A

      [,1]      [,2]      [,3]
[1,] 1.000000e+00  6.938894e-17  0.000000e+00
[2,] 2.081668e-17  1.000000e+00 -2.775558e-17
[3,] -5.551115e-17  0.000000e+00  1.000000e+00

A%%invA

      [,1]      [,2]      [,3]
[1,] 1.000000e+00  2.081668e-17 -5.551115e-17
[2,] 6.938894e-17  1.000000e+00  0.000000e+00
[3,] 0.000000e+00 -2.775558e-17  1.000000e+00
```

For the purpose of this course, `solve` is fine.

¹ The Cholesky decomposition is a handy way to keep your variance-covariance matrices valid when doing a parameter search. Don't search over the raw variance-covariance matrix. Search over a matrix where the lower triangle is 0, that is what a Cholesky decomposition looks like. Let's call it \mathbf{B} . Your variance-covariance matrix is $\mathbf{t}(\mathbf{B})\mathbf{B}$.

Problems

- 1.1 Build a 4×3 matrix with the numbers 1 through 4 in each row.
- 1.2 Extract the elements in the 1st and 2nd rows and 1st and 2nd columns (you'll have a 2×2 matrix). Show the R code that will do this.
- 1.3 Build a 4×3 matrix with the numbers 1 through 12 by row (meaning the first row will have the numbers 1 through 4 in it).
- 1.4 Extract the 3rd row of the above. Show R code to do this where you end up with a vector and how to do this where you end up with a 1×3 matrix.
- 1.5 Build a 4×3 matrix that is all 1s except a 2 in the (2,3) element (2nd row, 3rd column).
- 1.6 Take the transpose of the above.
- 1.7 Build a 4×4 diagonal matrix with 1 through 4 on the diagonal.
- 1.8 Build a 5×5 identity matrix.
- 1.9 Replace the diagonal in the above matrix with 2 (the number 2).
- 1.10 Build a matrix with 2 on the diagonal and 1s on the offdiagonals.
- 1.11 Take the inverse of the above.
- 1.12 Build a 3×3 matrix with the first 9 letters of the alphabet. First column should be "a", "b", "c". `letters[1:9]` gives you these letters.
- 1.13 Replace the diagonal of this matrix with the word "cat".
- 1.14 Build a 4×3 matrix with all 1s. Multiply by a 3×4 matrix with all 2s.
- 1.15 If \mathbf{A} is a 4×3 matrix, is \mathbf{AA} possible? Is \mathbf{AA}^\top possible? Show how to write \mathbf{AA}^\top in R.
- 1.16 In the equation, $\mathbf{AB} = \mathbf{C}$, let $\mathbf{A} = \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix}$. Build a \mathbf{B} matrix with only 1s and 0s such that the values on the diagonal of \mathbf{C} are 1, 8, 6 (in that order). Show your R code for \mathbf{A} , \mathbf{B} and \mathbf{AB} .
- 1.17 Same \mathbf{A} matrix as above and same equation $\mathbf{AB} = \mathbf{C}$. Build a 3×3 \mathbf{B} matrix such that $\mathbf{C} = 2\mathbf{A}$. So $\mathbf{C} = \begin{bmatrix} 2 & 8 & 14 \\ 4 & 10 & 16 \\ 6 & 12 & 18 \end{bmatrix}$. Hint, \mathbf{B} is diagonal.
- 1.18 Same \mathbf{A} and $\mathbf{AB} = \mathbf{C}$ equation. Build a \mathbf{B} matrix to compute the row sums of \mathbf{A} . So the first 'row sum' would be $1 + 4 + 7$, the sum of all elements in

row 1 of \mathbf{A} . \mathbf{C} will be $\begin{bmatrix} 12 \\ 15 \\ 18 \end{bmatrix}$, the row sums of \mathbf{A} . Hint, \mathbf{B} is a column matrix (1 column).

1.19 Same \mathbf{A} matrix as above but now equation $\mathbf{BA} = \mathbf{C}$. Build a \mathbf{B} matrix to compute the column sums of \mathbf{A} . So the first 'column sum' would be $1+2+3$. \mathbf{C} will be a 1×3 matrix.

1.20 Let $\mathbf{AB} = \mathbf{C}$ equation but $\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$ (so $\mathbf{A} = \text{diag}(3)+1$). Build a \mathbf{B} matrix such that $\mathbf{C} = \begin{bmatrix} 3 \\ 3 \\ 3 \end{bmatrix}$. Hint, you need to use the inverse of \mathbf{A} .