

Basic R: Matrices

Anthea Yichen Li

January 25, 2018

Matrix problems

1. Suppose

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

(a) Check that $A^3 = \mathbf{0}$

(b) Replace the third column of A by the sum of the second and third columns

First, produce A

```
A <- matrix(c(1,1,3,5,2,6,-2,-1,-3), nrow = 3, byrow = TRUE)
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    3
## [2,]    5    2    6
## [3,]   -2   -1   -3
```

Check whether $A^3 = \mathbf{0}$

```
A^3
```

```
##      [,1] [,2] [,3]
## [1,]    1    1   27
## [2,]   125    8  216
## [3,]   -8   -1  -27
```

Then, add the columns 2 and 3 and assign the sum to the third column

```
A[,3] <- A[,2] + A[,3]
```

```
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    4
## [2,]    5    2    8
## [3,]   -2   -1   -4
```

2. Create the following matrix B with 15 rows

$$B = \begin{bmatrix} 10 & -10 & 10 \\ 10 & -10 & 10 \\ \dots & \dots & \dots \\ 10 & -10 & 10 \end{bmatrix}$$

Calculate the 3x3 matrix $B^T B$. You can make this calculation with the function `crossprod()`. See the documentaion.

```
tmp <- matrix(c(10,-10,10), b = T, nc = 3, nr = 15)
t(tmp) %*% tmp
```

```
##      [,1] [,2] [,3]
## [1,] 1500 -1500 1500
## [2,] -1500 1500 -1500
## [3,] 1500 -1500 1500
```

3. Create a 6 x 6 matrix `matE` with every element equal to 0. check what the functions `row()` and `col()` return when applied to `matE`.

Now, create the 6 x 6 matrix:

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

Here is `matE`, a 6x6 matrix of 0's followed by `row(matE)` and `col(matE)`

```
matE <- matrix(rep(0,36), nrow = 6, byrow = TRUE)
```

```
# Note what the functions row() and col() do
row(matE)
```

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

```
col(matE)
```

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

```
# With a little experimentation you would see
# that the specified pattern is in the /1/'s
row(matE)-col(matE)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0   -1   -2   -3   -4   -5
## [2,]    1    0   -1   -2   -3   -4
## [3,]    2    1    0   -1   -2   -3
## [4,]    3    2    1    0   -1   -2
## [5,]    4    3    2    1    0   -1
## [6,]    5    4    3    2    1    0
```

```
# so you use the locations of the 1's to modify matE
matE[abs(row(matE)-col(matE))==1] <- 1
matE
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    0    1    0    0    0    0
## [2,]    1    0    1    0    0    0
## [3,]    0    1    0    1    0    0
## [4,]    0    0    1    0    1    0
## [5,]    0    0    0    1    0    1
## [6,]    0    0    0    0    1    0
```

4. Look at the help for the function `outer()`. Now, create the following patterned matrix:

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 6 \\ 3 & 4 & 5 & 6 & 7 \\ 4 & 5 & 6 & 7 & 8 \end{bmatrix}$$

```
a <- 0:4
A <- outer(a,a,"+")
A
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    1    2    3    4
## [2,]    1    2    3    4    5
## [3,]    2    3    4    5    6
## [4,]    3    4    5    6    7
## [5,]    4    5    6    7    8
```

Use `outer()` a little more to make sure you get it.

```
B <- outer(a,a, "*")
B
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    1    2    3    4
## [3,]    0    2    4    6    8
## [4,]    0    3    6    9   12
## [5,]    0    4    8   12   16
```

```
# and
b <- 5:10
C <- outer(a,b,"+")
C
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,]    5    6    7    8    9   10
## [2,]    6    7    8    9   10   11
## [3,]    7    8    9   10   11   12
## [4,]    8    9   10   11   12   13
## [5,]    9   10   11   12   13   14
```

```
# and finally -- make sure you check the values.
D <- outer(b,a, "%%")
D
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]  NA   0   1   2   1
## [2,]  NA   0   0   0   2
## [3,]  NA   0   1   1   3
## [4,]  NA   0   0   2   0
## [5,]  NA   0   1   0   1
## [6,]  NA   0   0   1   2
```

5. Create the following patterned matrices. Your solutions should be generalizable to enable creating larger matrices with the same structure.

(a)

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

```
a <- 0:4
A <- outer(a,a,"+") %%5
A
```

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

(b)

$$\begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 8 & 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 9 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{bmatrix}$$

```
a <- 0:9
A <- outer(a,a,"+")%%10
A
```

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

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

(c)

$$\begin{bmatrix} 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 & 2 \\ 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 & 3 \\ 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 & 4 \\ 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 & 5 \\ 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 & 7 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 8 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix}$$

```
a <- 0:8
A <- outer(a,a,"-")%%9
A
```

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

6. Solve the following system of linear equations by setting up and solving the matrix equation $Ax = y$.

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 &= 7 \\ 2x_1 + x_2 + 2x_3 + 3x_4 + 4x_5 &= -1 \\ 3x_1 + 2x_2 + x_3 + 2x_4 + 3x_5 &= -3 \\ 4x_1 + 3x_2 + 2x_3 + x_4 + 2x_5 &= 5 \\ 5x_1 + 4x_2 + 3x_3 + 2x_4 + x_5 &= 17 \end{aligned}$$

```
yVec <- c(7, -1, -3, 5, 17)
AMat <- matrix(0, nr = 5, nc = 5)
AMat <- abs(col(AMat) - row(AMat)) + 1
AMat
```

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

```
solve(AMat, yVec)
```

```
## [1] -2 3 5 2 -4
```

7. Create a 6 x 10 matrix of random integers chosen from 1,2,...,10 by executing the following two lines of code:

```
set.seed(75)
aMat <- matrix(sample(10, size=60, replace=TRUE), nr=6)
```

Use the matrix you have created to answer these questions:

(a) Find the number of entries in each row which are greater than 4.

```
apply(aMat, 1, function(x){sum(x>4)})
```

```
## [1] 4 7 6 2 6 7
```

(b) Which rows contain exactly two occurrences of the number seven?

```
which ( apply(aMat, 1, function(x) {sum(x==7) == 2}))
```

```
## [1] 5
```

(c) Find those pairs of columns whose total (over both columns) is greater than 75. The answer should be a matrix with two columns; so, for example, the row (1,2) in the output matrix means that the sum of columns 1 and 2 in the original matrix is greater than 75. Repeating a column is permitted; so, for example, the final output matrix could contain the rows (1,2), (2,1), and (2,2).

```
aMatColSums <- colSums(aMat)
which( outer(aMatColSums,aMatColSums,"+" ) >75, arr.ind = T)
```

```
##      row col
## [1,]    2    2
## [2,]    6    2
## [3,]    8    2
## [4,]    2    6
## [5,]    8    6
## [6,]    2    8
## [7,]    6    8
## [8,]    8    8
```

What if repetitions are not permitted? Then only (1,2) from (1,2),(2,1) and (2,2) would be permitted.

```
aMatColSums <- colSums(aMat)
logicalMat <- outer(aMatColSums,aMatColSums,"+" ) >75
logicalMat[lower.tri(logicalMat, diag = T)] <- F
which(logicalMat, arr.ind = T)
```

```
##      row col
## [1,]    2    6
## [2,]    2    8
## [3,]    6    8
```

8. Calculate

$$(a) \sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+j)}$$

```
sum((1:20)^4) * sum(1/(3+(1:5)))
```

```
## [1] 639215.3
```

```
# or
```

```
sum(outer((1:20)^4, (3+(1:5)), "/"))
```

```
## [1] 639215.3
```

$$(b) \sum_{i=1}^{20} \sum_{j=1}^5 \frac{i^4}{(3+ij)}$$

```
sum( (1:20)^4 / (3 + outer(1:20, 1:5, "*")) )
```

```
## [1] 89912.02
```

$$(c) \sum_{i=1}^{10} \sum_{j=1}^i \frac{i^4}{(3+ij)}$$

```
sum(outer(1:10, 1:10, function(i,j) { (i>=j)*i^4/(3+i*j) } ))
```

```
## [1] 6944.743
```