Assignment 2

Kerin Grewal

January 24th

1.

(a) Check that A^3 $=$ 0 where 0 is a 3×3 matrix with every entry equal to 0.

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

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

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

```
x <- A[,1] + A[,2]
A[,3] <- x
A
```

```
##      [,1] [,2] [,3]
## [1,]    1    1    2
## [2,]    5    2    7
## [3,]   -2   -1   -3
```

2.Create the matrix B with 15 rows: Calculate the 3×3 matrix B^T B

```
x <- cbind(rep(10, 15))
y <- cbind(rep(-10, 15))

A <- matrix(
  rep(c(x, y), 8),
  nrow = 15,
  ncol = 15,
  byrow = FALSE
)
B <- crossprod(A, y = NULL)
B
```

```
##        [,1]  [,2]  [,3]  [,4]  [,5]  [,6]  [,7]  [,8]  [,9] [,10] [,11]
##  [1,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
##  [2,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
##  [3,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
##  [4,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
##  [5,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
##  [6,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
##  [7,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
##  [8,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
##  [9,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
## [10,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
## [11,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
## [12,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
## [13,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
## [14,] -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500
## [15,]  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500 -1500  1500
##       [,12] [,13] [,14] [,15]
##  [1,] -1500  1500 -1500  1500
##  [2,]  1500 -1500  1500 -1500
##  [3,] -1500  1500 -1500  1500
##  [4,]  1500 -1500  1500 -1500
##  [5,] -1500  1500 -1500  1500
##  [6,]  1500 -1500  1500 -1500
##  [7,] -1500  1500 -1500  1500
##  [8,]  1500 -1500  1500 -1500
##  [9,] -1500  1500 -1500  1500
## [10,]  1500 -1500  1500 -1500
## [11,] -1500  1500 -1500  1500
## [12,]  1500 -1500  1500 -1500
## [13,] -1500  1500 -1500  1500
## [14,]  1500 -1500  1500 -1500
## [15,] -1500  1500 -1500  1500
```

3.Create a 6 ×6 matrix matE with every entry equal to 0. Check what the functions row and col return when applied to matE.

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

matE[abs(col(matE) - row(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.

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

```
##      [,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
```

5. Create the following patterned matrices. In each case, your solution should make use of the special form of the matrix—this means that the solution should easily generalise to creating a larger matrix with the same structure and should not involve typing in all the entries in the matrix.

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

```
##      [,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
```

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

```
##       [,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
```

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

```
##       [,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 in five unknowns

```
b <- c(7, -1, -3, 5, 17)
A <- matrix(0, nrow = 5, ncol = 5)
A <- abs(col(A)-row(A)) + 1
A
```

```
##      [,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
```

```
x <- solve(A, b)
x
```

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

7. Create a 6 × 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=T), nr=6)
aMat
```

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

a.

```
colSums(aMat>4)
```

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

b.

```
which(colSums(aMat == 7) == 2)
```

```
## [1] 3 4 8
```

c.

```
aColSums <- colSums(aMat)
aColSums
```

```
##  [1] 18 42 32 33 32 36 31 40 31 20
```

```
which(outer(aColSums, aColSums, "+")>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
```

8. Calculate

a.

```
sum(outer((1:20)^4,4:8,"/"))
```

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

b.

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

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

c.

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

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