Stat 291 - Recitation 2

Orçun Oltulu

05 / 11 / 2021

Exercise 1: Creating a Sequence

You can use 'c()', 'seq' and 'rep' functions to construct vectors. By using those functions create following vectors; (use help menu to read descriptions and default values for 'seq' and 'rep' functions)

- a vector of sequence from 2 to 7 and from 20 to 12.
- a vector of sequence from 0 to 20 by 2 increments.
- a vector of sequence from 1 to 10 by 3 increments, each repeated 4 times.
- a vector of sequence from 5 to 10, 4 times.
- a vector of sequence from -10 to 10 by 2 increments, 3 times
- {1 2 2 3 3 3 4 4 4 4 5 5 5 5 5}

```
vec1 <- c(2:7, 20:12) # or c(seq(2, 7), seq(20,12))
print(vec1)
```

```
##  [1]  2  3  4  5  6  7 20 19 18 17 16 15 14 13 12
```

```
vec2 <- seq(from=0,to=20,by=2)
print(vec2)
```

```
##  [1]  0  2  4  6  8 10 12 14 16 18 20
```

```
vec3 <- rep(seq(1,10,3),each=4)
print(vec3)
```

```
##  [1]  1  1  1  1  4  4  4  4  7  7  7  7 10 10 10 10
```

```
vec4 <- rep(seq(5,10),4)
print(vec4)
```

```
##  [1]  5  6  7  8  9 10  5  6  7  8  9 10  5  6  7  8  9 10  5  6  7  8  9 10
```

```
vec5 <- rep(seq(-10,10,2),3)
print(vec5)
```

```
##  [1] -10  -8  -6  -4  -2   0   2   4   6   8  10 -10  -8  -6  -4  -2   0   2   4
## [20]   6   8  10 -10  -8  -6  -4  -2   0   2   4   6   8  10
```

```
vec6 <- rep(c(1,2,3,4,5),c(1,2,3,4,5))
print(vec6)
```

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

Exercise 2: Operation on Vectors

Exercise 2.1:

Create 3 vectors and calculate following;

$$x = \{5, -6, 1, 2\},\ y = \{2, 3, -1, 8\},\ z = \{1, 2, 3, 4\},\ t = \{-1, 1\}$$

- z + 2
- x + y
- y * z
- z - y
- z + t
- z / t

```
x <- c(5,-6,1,2); y <- c(2,3,-1,8); z <- 1:4; t <- c(-1,1)
z + 2 # Scalar 2 is recycled to 2,2,2,2
```

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

```
x + y
```

```
## [1]  7 -3  0 10
```

```
y * z
```

```
## [1]  2  6 -3 32
```

```
z - y
```

```
## [1] -1 -1  4 -4
```

```
z + t # Element of t is recycled to -1,1,-1,1
```

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

```
z / t
```

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

Exercise 2.2:

Calculate the sum $\sum_{i=1}^{50} i^2$ by using R. (*Hint:* use 'sum' function) Then, compare your result with the formula $\frac{n(n+1)(2n+1)}{6}$.

```
n <- 50

total1 <- sum(seq(1,50)^2)

total2 <- n * (n+1) * (2*n+1) / 6

total1 == total2
```

```
## [1] TRUE
```

Exercise 2.3:

The weights of five people before and after a diet program are given in the following. Read the 'before' and 'after' values into two different vectors called before and after. Use R to evaluate the amount of weight lost for each participant. What is the average amount of weight lost?

- Before: 78 72 78 79 105
- After: 67 65 79 70 93

```
before <- c(78,72 ,78 ,79 ,105)
after <- c(67, 65 ,79 ,70 ,93)

weight_lost <- before - after
weight_lost
```

```
## [1] 11  7 -1  9 12
```

```
av_weight_lost <- mean(weight_lost)
av_weight_lost
```

```
## [1] 7.6
```

Exercise 3: Indexing for vectors

Exercise 3.1:

Consider the following vectors;

- x1 = {10 15 20 25 30 35 40 45 50}
- x2 = {1 2 3 11 12 13 -1 -2 -3 -11 -12 -13}
- x3 = {5 -8 3.3 -5.6 8.2 4.4 -2.2 3.0 -0.5}

- x4 = {“Michael”, “Jim”, “Pam”, “Dwigth”, “Kevin”, “Angela”, “Kelly”, “Andy”, “Oscar”, “Meredith”}

```
x1 <- seq(from = 10, to = 50, by = 5)
x2 <- c(1:3, 11:13, -1:-3, -11:-13)
x3 <- c(5, 8, 2, 3.3, 5.6, 8.2, 4.4, -2.2, -3, -.5)
x4 <- c("Michael", "Jim", "Pam", "Dwigth", "Kevin", "Angela", "Oscar")
```

Part A.

Given x1 form a new vector y1 whose entries are the entries of x1 with odd index;

(i.e., x1[1], x1[3], ….)

```
length(x1)
```

```
## [1] 9
```

```
y1 <- c(x1[1], x1[3], x1[5], x1[7], x1[9]) # or
y1 <- x1[seq(1,length(x1), by=2)]
y1
```

```
## [1] 10 20 30 40 50
```

Part B.

Take the third elements of those 4 vectors above and create a vector y2.

```
y2 <- c(x1[3], x2[3], x3[3], x4[3])
y2
```

```
## [1] "20"  "3"   "2"   "Pam"
```

Part C.

Form a vector y3 such that for each j;

$$y3[j] = \begin{cases} y3[j] = x3[j] & if \quad x3[j] > 0 \\ y3[j] = 0 & if \quad x3[j] \leq 0 \end{cases}$$

```
y3 <- x3
y3[x3 <= 0] <- 0
y3
```

```
##  [1] 5.0 8.0 2.0 3.3 5.6 8.2 4.4 0.0 0.0 0.0
```

Part D.

x4 consists of some characters from a series called The Office. From x4 create a new vector, say y4, such that y4 has only {Jim Pam and Kevin}.

```
y4 <- x4[x4 == "Jim" | x4 == "Pam" | x4 == "Kevin"] # or
y4 <- x4[x4 %in% c("Jim","Pam","Kevin")]
y4
```

```
## [1] "Jim"   "Pam"   "Kevin"
```

Exercise 3.2:

Create a vector called 'plate_numbers' = {6, 44, 69, 33, 57, 34}. Then, using *names* function assign each element city names, {"Ankara", "Malatya", "Bayburt", "Mersin", "Sinop", "Istanbul"} respectively.

```
plate_numbers <- c(6, 44, 69, 33, 57, 34)
names(plate_numbers) <- c("Ankara", "Malatya", "Bayburt",
                          "Mersin", "Sinop", "Istanbul")
plate_numbers
```

```
##   Ankara  Malatya  Bayburt   Mersin    Sinop Istanbul
##        6       44       69       33       57       34
```

Part A.

Among those 6 cities, whose plate number is less than 35?

```
plate_numbers[plate_numbers<35]
```

```
##   Ankara   Mersin Istanbul
##        6       33       34
```

```
# or if you want to see only the city names
names(plate_numbers[plate_numbers < 35])
```

```
## [1] "Ankara"   "Mersin"   "Istanbul"
```

Part B.

What is the plate number of "Sinop"?

```
plate_numbers["Sinop"]
```

```
## Sinop
##    57
```

Part C.

What are the plate numbers of "Bayburt" and "Malatya"?

```
plate_numbers[c("Bayburt","Malatya")]
```

```
## Bayburt Malatya
##      69      44
```

Part D.

Form a vector such that it consists only plates which are odd number.

```
odd_numbers <- plate_numbers[plate_numbers %% 2 == 1]
odd_numbers
```

```
## Bayburt  Mersin   Sinop
##      69      33      57
```

```
# or if you want to see only the plate numbers
as.numeric(odd_numbers)
```

```
## [1] 69 33 57
```

Part E.

Print 'plate_numbers' vector without the second and the forth item.

```
plate_numbers[c(-2,-4)]
```

```
##   Ankara  Bayburt    Sinop Istanbul
##        6       69       57       34
```

Exercise 4: Matrix

Exercise 4.1:

Create following matrices;

- a 5 x 5 matrix of zeros
- a 5 x 5 matrix of ones
- a 5 x 5 diagonal matrix having each element as 5
- a 5 x 5 diagonal matrix having each element as 1 to 5
- Also,

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

```r
mat1 <- matrix(rep(0,25), nrow = 5, ncol = 5); mat1
```

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

```r
mat2 <- matrix(1, nrow = 5, ncol = 5); mat2
```

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

```r
mat3 <- diag(5, 5); mat3
```

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

```r
mat4 <- diag(1:5, 5); mat4
```

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

```r
mat5 <- matrix(rep(c(1,0,1),3), nrow = 3, byrow = TRUE); mat5
```

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

Exercise 4.2:

Create the following matrix

$$\begin{bmatrix} 1.1 & 1.3 & 2.2 \\ 1.2 & 2.1 & 2.3 \\ 1.3 & 1.5 & 2.9 \\ 0.5 & 1.2 & 3.1 \end{bmatrix}$$

```
mat <- matrix(c(1.1,1.2,1.3,0.5,
                1.3,2.1,1.5,1.2,
                2.2,2.3,2.9,3.1),
              nrow = 4, ncol = 3, byrow = F)
print(mat)
```

```
##      [,1] [,2] [,3]
## [1,]  1.1  1.3  2.2
## [2,]  1.2  2.1  2.3
## [3,]  1.3  1.5  2.9
## [4,]  0.5  1.2  3.1
```

Part A.

Print out the second column.

```
print(mat[,2])
```

```
## [1] 1.3 2.1 1.5 1.2
```

Part B.

Print the matrix without the second row.

```
print(mat[-2,])
```

```
##      [,1] [,2] [,3]
## [1,]  1.1  1.3  2.2
## [2,]  1.3  1.5  2.9
## [3,]  0.5  1.2  3.1
```

Part C.

Print the element at (2,3) position in the matrix.

```
print(mat[2,3])
```

```
## [1] 2.3
```

Part D.

Print the first and the third columns in the matrix.

```
print(mat[,c(1,3)])
```

```
##      [,1] [,2]
## [1,]  1.1  2.2
## [2,]  1.2  2.3
## [3,]  1.3  2.9
## [4,]  0.5  3.1
```

```
# or
print(mat[,-2])
```

```
##      [,1] [,2]
## [1,]  1.1  2.2
## [2,]  1.2  2.3
## [3,]  1.3  2.9
## [4,]  0.5  3.1
```

Part F.

Replace the first column with the following sequence; {1,2,3,4}

```
mat[,1] <- 1:4
print(mat)
```

```
##      [,1] [,2] [,3]
## [1,]    1  1.3  2.2
## [2,]    2  2.1  2.3
## [3,]    3  1.5  2.9
## [4,]    4  1.2  3.1
```

Part E.

Replace the first and the last columns with each other.

```
mat[,c(1,3)] <- mat[,c(3,1)]
mat
```

```
##      [,1] [,2] [,3]
## [1,]  2.2  1.3    1
## [2,]  2.3  2.1    2
## [3,]  2.9  1.5    3
## [4,]  3.1  1.2    4
```

Part F.

Find dimension of this matrix.

```r
dim(mat)
```

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

Exercise 4.3: rbind() & cbind()

Create following sequences and store each in vectors; vec1, vec2 and vec3

- {1, 2, 3, 4, 5, 6}
- {10, 20, 30, 40, 50, 60}
- {1, 1, 1, 2, 2, 2}

```r
vec1 <- 1:6
vec2 <- seq(10, 60, 10)
vec3 <- rep(1:2, each = 3)
```

Part A.

Combine vec1 and vec2 to achieve 2x6 matrix.

```r
newMatrix1 <- rbind(vec1, vec2)
print(newMatrix1)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## vec1    1    2    3    4    5    6
## vec2   10   20   30   40   50   60
```

Part B.

Combine all three vectors to achieve 6x3 matrix.

```r
newMatrix2 <- cbind(vec1, vec2, vec3)
print(newMatrix2)
```

```
##      vec1 vec2 vec3
## [1,]    1   10    1
## [2,]    2   20    1
## [3,]    3   30    1
## [4,]    4   40    2
## [5,]    5   50    2
## [6,]    6   60    2
```

Part C.

Using first 4 elements in each vector, create 4x3 matrix.

```r
newMatrix3 <- cbind(vec1[1:4],
                    vec2[1:4],
```

```
                    vec3[1:4])
print(newMatrix3)
```

```
##      [,1] [,2] [,3]
## [1,]    1   10    1
## [2,]    2   20    1
## [3,]    3   30    1
## [4,]    4   40    2
```

```
# or
# cbind(vec1,vec2,vec3)[,1:4]
```

Exercise 5: Matrix Operations

Exercise 5.1:

Create the following matrix and assign to X;

```
      [,1] [,2] [,3] [,4]
[1,]   -2    4   -6    8
[2,]  -10   12  -14   16
[3,]    1    2    3    4
[4,]    5    6    7    8
```

```
X <- matrix(c(seq(2,16, by = 2) * c(-1,1), 1:8),
                 nrow=4, ncol=4, byrow = TRUE)
X
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   -2    4   -6    8
## [2,]  -10   12  -14   16
## [3,]    1    2    3    4
## [4,]    5    6    7    8
```

- Take Transpose of X;

```
t(X)
```

```
##      [,1] [,2] [,3] [,4]
## [1,]   -2  -10    1    5
## [2,]    4   12    2    6
## [3,]   -6  -14    3    7
## [4,]    8   16    4    8
```

- Take Inverse of X;

```
solve(X)
```

```
##          [,1]     [,2]    [,3]    [,4]
## [1,]  0.21875 -0.09375 -0.4375  0.1875
## [2,] -0.25000  0.12500 -0.5000  0.2500
## [3,] -0.15625  0.03125  0.3125 -0.0625
## [4,]  0.18750 -0.06250  0.3750 -0.1250
```

Exercise 5.2:

Create 3 matrices A, B and C such that

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

$$B = \begin{pmatrix} 1 & 0 \\ 2 & -2 \end{pmatrix}$$

$$C = \begin{pmatrix} -4 & 1 \\ 3 & -1 \end{pmatrix}$$

```
A <- matrix(c(2,-1,-3,5), nrow = 2)
B <- matrix(c(1, 2, 0, -2), ncol = 2)
C <- matrix(c(-4, 1, 3, -1), nrow = 2, byrow = TRUE)
```

Part A.

Calculate A + B

```
A + B
```

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

Part B.

Calculate B - C

```
B - C
```

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

Part C.

Calculate B * C

```
B %*% C
```

```
##      [,1] [,2]
## [1,]   -4    1
## [2,]  -14    4
```

Part D.

Calculate $A * B^{-1} + C^T$

```
A %*% solve(B) + t(C)
```

```
##      [,1] [,2]
## [1,]   -5  4.5
## [2,]    5 -3.5
```