Lab 1-1 Exercise - R Basic Commands

Ruoxin WANG (Wendy) - 2030026150

2023/09/19

Please input your student name and No. below.
Student Name:
Student No.:

Q1. Let x be defined as follow.
a. Add 3 to just the even-index elements.
b. Compute the square root of each element.

```
x <- c(11,10,5,9,14,12,18,2,17,9,1,7,1,15,16,7,16,12,13,3,8,3,11,1,5,7,7,10,6,10,17,13,1,2,17,
1,15,8,12,3,6,15,10,3,10,12,19,17,15,12,18,17,1,13,5,9,18,2,20,16,15,7,9,12,12,5,5,7,7,9,17,7,1
5,11,10,12,19,8,13,3,7,14,13,18,1,11,18,1,12,18,5,12,5,10,3,6,11,12,1,3)
```

```
# Write your code here
# a
x_3 <- 1:length(x)
x_3 <- x[seq(2,length(x),2)] + 3
print(x_3)
```

```
##  [1] 13 12 15  5 12 10 18 10 15  6  6  4 10 13 13 16  5  4 11  6 18  6 15 20 15
## [26] 20 16 12  5 19 10 15  8 10 12 10 14 15 11  6 17 21 14  4 21 15 13  9 15  6
```

```
# b
x_sqrt <- 1:length(x)
x_sqrt <- sqrt(x)
print(x_sqrt)
```

```
##   [1] 3.316625 3.162278 2.236068 3.000000 3.741657 3.464102 4.242641 1.414214
##   [9] 4.123106 3.000000 1.000000 2.645751 1.000000 3.872983 4.000000 2.645751
##  [17] 4.000000 3.464102 3.605551 1.732051 2.828427 1.732051 3.316625 1.000000
##  [25] 2.236068 2.645751 2.645751 3.162278 2.449490 3.162278 4.123106 3.605551
##  [33] 1.000000 1.414214 4.123106 1.000000 3.872983 2.828427 3.464102 1.732051
##  [41] 2.449490 3.872983 3.162278 1.732051 3.162278 3.464102 4.358899 4.123106
##  [49] 3.872983 3.464102 4.242641 4.123106 1.000000 3.605551 2.236068 3.000000
##  [57] 4.242641 1.414214 4.472136 4.000000 3.872983 2.645751 3.000000 3.464102
##  [65] 3.464102 2.236068 2.236068 2.645751 2.645751 3.000000 4.123106 2.645751
##  [73] 3.872983 3.316625 3.162278 3.464102 4.358899 2.828427 3.605551 1.732051
##  [81] 2.645751 3.741657 3.605551 4.242641 1.000000 3.316625 4.242641 1.000000
##  [89] 3.464102 4.242641 2.236068 3.464102 2.236068 3.162278 1.732051 2.449490
##  [97] 3.316625 3.464102 1.000000 1.732051
```

Q2. Given x = [3 15 9 12 -1 -12 9 6 1], provide the command that will
a) set the values of x that are positive to zero
b) set values that are multiples of 3 to 3

c) multiply the values of x that are even by 5
d) extract the values of x that are greater than 10 into a vector called y
e) Find the index position of elements in x which are larger than 4

```
# Write your code here
x <- c(3,15,9,12,-1,-12,9,6,1)
# a
x_pos <- 1:length(x)
x_pos <- x
x_pos[x_pos > 0] <- 0
print(x_pos)
```

```
## [1]   0   0   0   0  -1 -12   0   0   0
```

```
# b
x_multi <- 1:length(x)
x_multi <- x
x_multi[x_multi%%3 == 0] <- 3
print(x_multi)
```

```
## [1]  3  3  3  3 -1  3  3  3  1
```

```
# c
x_even <- 1:length(x)
x_even <- x
x_even[x_even%%2 == 0] <- 5 * x_even[x_even%%2==0]
print(x_even)
```

```
## [1]   3  15   9  60  -1 -60   9  30   1
```

```
# d
y <- 1:length(x)
y <- x[x>10]
y
```

```
## [1] 15 12
```

```
# e
x_ind <- 1:length(x)
x_ind <- which(x > 4)
x_ind
```

```
## [1] 2 3 4 7 8
```

Q3. Given the following list, please answer the questions below.

```
# Create a list containing a vector, a matrix and a list.
list_data <- list(c("Jan", "Feb", "Mar"), matrix(c(3, 9, 5, 1, -2, 8), nrow = 2), list("green",
12.3))
# Give names to the elements in the list.
names(list_data) <- c("1st Quarter", "A_Matrix", "An Inner list")
```

a. Access the first element of the list using position.

b. Access the last element of the list using name.

c. Add element “New element” at the end of the list.

d. Remove the last element.

```
# Write your code here
# a
list_data[1]
```

```
## $`1st Quarter`
## [1] "Jan" "Feb" "Mar"
```

```
# Write your code here
# b
list_data["An Inner list"]
```

```
## $`An Inner list`
## $`An Inner list`[[1]]
## [1] "green"
##
## $`An Inner list`[[2]]
## [1] 12.3
```

```
# Write your code here
# c
list_data_new <- list_data
list_data_new[4] = "New element"
list_data_new
```

```
## $`1st Quarter`
## [1] "Jan" "Feb" "Mar"
##
## $A_Matrix
##      [,1] [,2] [,3]
## [1,]    3    5   -2
## [2,]    9    1    8
##
## $`An Inner list`
## $`An Inner list`[[1]]
## [1] "green"
##
## $`An Inner list`[[2]]
## [1] 12.3
##
##
## [[4]]
## [1] "New element"
```

```
# Write your code here
# d
list_data_rm <- list_data
list_data_rm[3] = NULL
list_data_rm
```

```
## $`1st Quarter`
## [1] "Jan" "Feb" "Mar"
##
## $A_Matrix
##      [,1] [,2] [,3]
## [1,]    3    5   -2
## [2,]    9    1    8
```

Q4. Given the matrix A as following

```
a <- c(2,4,1,6,7,2,3,5,9)
A <- matrix(a, nrow=3, ncol=3, byrow=T)
A
```

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

provide the commands needed to
a) assign the last 2 rows of A to a matrix called y
b) compute the sum over the columns of A
c) compute the sum over the rows of A
Hint: you can use function `apply`

```
# Write your code here
# a
y <- A[2:3,]
y
```

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

```
# Write your code here
# b
col_sum <- apply(A, 2, sum)
col_sum
```

```
## [1] 11 16 12
```

```
# Write your code here
# c
row_sum <- apply(A, 1, sum)
row_sum
```

```
## [1]  7 15 17
```