

# The University of British Columbia

Irving K. Barber Faculty of Science

DATA 101

Lab Assignment 2

**Date:** September 21-25, 2020

*Demonstration.* The TA will guide you through the following points. Following the demonstration, there are some exercises for you to complete, with assistance from the TA.

- The  $\sum$  symbol means to “add up” or “sum”. It usually appears with a number at the top and what looks like an equation at the bottom, such as

$$\sum_{j=1}^4.$$

This actually means that there will be 4 numbers to add up, each one corresponding to a different value of  $j$ , starting at 1 and ending at 4. For example,

$$\sum_{j=1}^4 j = 1 + 2 + 3 + 4.$$

If we want to add up the squares of the 4 numbers, we would use

$$\sum_{j=1}^4 j^2 = 1^2 + 2^2 + 3^2 + 4^2.$$

- In R, these calculations can be carried out using the `:` operator and the `sum()` function:

```
sum(1:4)

## [1] 10

sum((1:4)^2)

## [1] 30
```

- Now, we will calculate the sum  $\sum_{j=1}^n j$  and compare with  $n(n+1)/2$ , for  $n = 100, 200, 400, 800$ .

```
n<- c(100,200,400,800)
n*(n+1)/2

## [1] 5050 20100 80200 320400

c(sum(1:100),sum(1:200),sum(1:400),sum(1:800))

## [1] 5050 20100 80200 320400
```

- Referring to the above example, we will use the quick formula to compute  $\sum_{j=1}^n j$  for all values of  $n$  between 1 and 100. Store the 100 values in a vector.

```
n <- 1:100 # this stores all the possible n values
sumsUp2n <- n*(n+1)/2 # this stores each of the sums
```

We can check the first 5 elements of `sumsUp2n` to see that the calculation was done correctly:

```
sumsUp2n[1:5]
## [1]  1  3  6 10 15
```

- This time, we will calculate the sum  $\sum_{j=1}^n j^2$  and compare with  $n(n+1)(2n+1)/6$  for  $n = 10, 20, 40, 80$ .

```
n<- c(10,20,40,80)
n*(n+1)*(2*n+1)/6

## [1]    385    2870   22140  173880

c(sum((1:10)^2),sum((1:20)^2),sum((1:40)^2),sum((1:80)^2))

## [1]    385    2870   22140  173880
```

- Referring to the above example, we will use the quick formula to compute  $\sum_{j=1}^n j^2$  for all values of  $n$  between 1 and 100. Store the 100 values in a vector.

```
n <- 1:100 # this stores all the possible n values
sumSquaresUp2n <- n*(n+1)*(2*n+1)/6 # this stores each of the sums
```

We can check the first 5 elements of `sumSquaresUp2n` to see that the calculation was done correctly:

```
sumSquaresUp2n[1:5]
## [1]  1  5 14 30 55
```

- Now, suppose  $x_1 = 15$ ,  $x_2 = 28$  and  $x_3 = -42$ . Using the summation notation, we would write

$$\sum_{j=1}^3 x_j = x_1 + x_2 + x_3 = 15 + 28 + (-42).$$

In R, we write

```
x <- c(15, 28, -42)
sum(x)

## [1] 1
```

If we want the sum of the squared  $x$  values, we write

$$\sum_{j=1}^3 x_j^2 = x_1^2 + x_2^2 + x_3^2 = 15^2 + 28^2 + (-42)^2.$$

In R, we write

```
sum(x^2)

## [1] 2773
```

- The `seq()` function is similar to `:` but it allows for a larger variety of patterns. If we want a list of the odd numbers from 13 through 26, we can type

```
seq(13, 26, 2)
```

to run through every second number from 13 to 26. The result is

```
## [1] 13 15 17 19 21 23 25
```

- Let's list all the numbers, starting from 1 and ending at 100, if we count by 7's:

```
count1to100by7 <- seq(1, 100, 7)
count1to100by7

## [1] 1 8 15 22 29 36 43 50 57 64 71 78 85 92 99
```

We can use the `length()` function to count the numbers in this sequence:

```
length(count1to100by7)

## [1] 15
```

- The `rep()` function creates repeated patterns. For example, if I want to repeat the number 3, 7 times, I type

```
rep(3, 7)

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

If I want to repeat the sequence (2, 4, 8), 3 times, I type

```
rep(c(2, 4, 8), 3)
```

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

but if want to repeat each element of (2, 4, 8), 3 times, I type

```
rep(c(2, 4, 8), each = 3)
```

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

If I want to repeat each element a different number of times, say 4 2's, 7 4's and 2 8's, I type

```
rep(c(2, 4, 8), c(4, 7, 2))
```

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

I can combine `rep()` and `seq()` to produce lots of different patterns. For example,

```
rep(seq(2, 11, 3), seq(7, 1, -2))
```

```
## [1] 2 2 2 2 2 2 2 5 5 5 5 5 8 8 8 11
```

In each question below, write out (or type) the required lines of R code, together with the answer to the question.

1. Calculate the sum  $\sum_{j=1}^n j^2$  and compare with  $n(n+1)(2n+1)/6$ , for  $n = 200, 400, 600, 800$ .
2. Calculate the sum  $\sum_{j=1}^n \sqrt{j}$  for  $n = 200, 400, 600, 800$ .
3. Suppose  $y_1 = 7$ ,  $y_2 = -4$ ,  $y_3 = 5$  and  $y_4 = 15$ . Using R, find  $\sum_{j=1}^4 y_j$  and  $\sum_{j=1}^4 y_j^2$ .
4. Use the `rep()` and `seq()` functions in R to obtain the following patterned data vectors:

(a) 

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

(b) 

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

(c) 

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

(d) 

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

5. The area of a rectangle can be calculated by multiplying the length by the width. You have 5 rectangles with lengths 3, 7, 12, 15 and 20. The corresponding widths are 2, 5, 8, 11 and 15. Construct appropriate vectors called `reclength` and `recwidth`, and use R to calculate all of the areas of these rectangles.

6. Download a file called `rain.txt` that comes with the lab assignment questions. Then read the data in the file into a data frame called `rain.df` using the `read.table()` function. Use the `header = FALSE` option.

```
rain.df <- read.table("rain.txt", header=FALSE, sep="")  
#head(rain.df)
```