STAT 40001/ STAT 50001 Statistical Computing Fall -2024

Lab-2

1) Create the following vectors using rep function in R:
   a) V1= 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

```
> V1 <- rep(1:5,5)
> V1
[1] 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5
```

   b) V2= 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6

```
> V2 <- rep(1:6,each = 4)
> V2
[1] 1 1 1 1 2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6
```

   c) V3=5 10 10 15 15 15 20 20 20 20 25 25 25 25 25

```
> V3 <- rep(seq(5,25,5),1:5)
> V3
[1]  5 10 10 15 15 15 20 20 20 20 25 25 25 25 25
```

   d) V4= Math, Math, CS, CS, STAT, STAT, STAT, PHY,PHY,PHY

```
> V4 <- noquote(rep(c("Math","CS","STAT","PHY"),c(2,2,3,3)))
> V4
[1] Math Math CS   CS   STAT STAT STAT PHY  PHY  PHY
```

2) Import the data below in R using scan function
   2 4 5 6 7 8 9 2 3 4 5 6 77 89 45 67 8 9 0 12

```
> y <- scan()
1: 2 4 5 6 7 8 9 2 3 4 5 6 77 89 45 67 8 9 0 12
21:
Read 20 items
```

3) Generate the following sequence of numbers
   a) 1,2,3,…,50.

```
> y <- seq(1,50,1)
> y
[1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 2
1 22 23 24
[25] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47 48
[49] 49 50
```

Or

```
> 1:50
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
21 22 23 24
[25] 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
45 46 47 48
[49] 49 50
```

b) 2,4,6,8,…,50

```
> y <- seq(2,50,2)
> y
[1]  2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 4
2 44 46 48
[25] 50
```

c) "A" "B" "C" "D" "E" "F" "G" "H"

```
> y <- LETTERS[1:8]
> y
[1] "A" "B" "C" "D" "E" "F" "G" "H"
```

d) e f g h i j k l

```
> y <- noquote(letters[5:12])
> y
[1] e f g h i j k l
```

4) Suppose we have the data below

2,5,7,8,9,3,5,8,67,45, 1,NA, 34,23,12,90

a) How many observations are there in the data set?

```
> data <- c(2,5,7,8,9,3,5,8,67,45, 1,NA, 34,23,12,90)
> length(data)
[1] 16
```

b) Is there any missing value? Use R to check it out.

```
> any(is.na(data))
[1] TRUE
```

c) Identify the location of the missing value.

```
> which(is.na(data))
[1] 12
```

d) Identify the smallest and largest observation (both position and the value)

```
> #Largest
```

```
> which.max(data)
[1] 16
> data[which.max(data)]
[1] 90
> #Smallest
> which.min(data)
[1] 11
> data[which.min(data)]
[1] 1
```