

Worksheet 1 in R

Erl Syron R. Espadon

1. Set up a vector named age

```
age <- c(34, 28, 22, 36, 27, 18, 52, 39, 42, 29,  
        35, 31, 27, 22, 37, 34, 19, 20, 57, 49,  
        50, 37, 46, 25, 17, 37, 42, 53, 41, 51,  
        35, 24, 33, 41)
```

a. How many data points?

```
length(age)
```

```
## [1] 34
```

2. Find the reciprocal of the values for age

```
reciprocalage <- 1 / age  
reciprocalage
```

```
## [1] 0.02941176 0.03571429 0.04545455 0.02777778 0.03703704 0.05555556  
## [7] 0.01923077 0.02564103 0.02380952 0.03448276 0.02857143 0.03225806  
## [13] 0.03703704 0.04545455 0.02702703 0.02941176 0.05263158 0.05000000  
## [19] 0.01754386 0.02040816 0.02000000 0.02702703 0.02173913 0.04000000  
## [25] 0.05882353 0.02702703 0.02380952 0.01886792 0.02439024 0.01960784  
## [31] 0.02857143 0.04166667 0.03030303 0.02439024
```

3. Assign new_age

```
newage <- c(age, 0, age)  
newage
```

```
## [1] 34 28 22 36 27 18 52 39 42 29 35 31 27 22 37 34 19 20 57 49 50 37 46 25 17  
## [26] 37 42 53 41 51 35 24 33 41 0 34 28 22 36 27 18 52 39 42 29 35 31 27 22 37  
## [51] 34 19 20 57 49 50 37 46 25 17 37 42 53 41 51 35 24 33 41
```

4. Sort the values for age

```
sortedage <- sort(age)  
sortedage
```

```
## [1] 17 18 19 20 22 22 24 25 27 27 28 29 31 33 34 34 35 35 36 37 37 37 39 41 41
```

```
## [26] 42 42 46 49 50 51 52 53 57
```

5. Find the minimum and maximum value for age

```
minage <- min(age)
maxage <- max(age)
minage
```

```
## [1] 17
```

```
maxage
```

```
## [1] 57
```

6. Set up a vector named data

```
data <- c(2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3, 2.5, 2.3, 2.4, 2.7)
```

a. How many data points?

```
length(data)
```

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

7. Generates a new vector for data where you double every value

```
doubleddata <- data * 2
doubleddata
```

```
## [1] 4.8 5.6 4.2 5.0 4.8 4.4 5.0 4.6 5.0 4.6 4.8 5.4
```

8. Generate sequences

```
seq1to100 <- seq(1, 100)
seq20to60 <- seq(20, 60)
mean20to60 <- mean(seq(20, 60))
sum51to91 <- sum(seq(51, 91))
seq1to1000 <- seq(1, 1000)
```

a. How many data points?

```
length(seq1to100)
```

```
## [1] 100
```

```
length(seq20to60)
```

```
## [1] 41
```

```
length(mean20to60)
```

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

```
length(sum51to91)
```

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

b. Output of sequences

```
seq1to100
```

```
## [1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
## [19] 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36
## [37] 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
## [55] 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
## [73] 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
## [91] 91 92 93 94 95 96 97 98 99 100
```

```
seq20to60
```

```
## [1] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44
## [26] 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60
```

```
mean20to60
```

```
## [1] 40
```

```
sum51to91
```

```
## [1] 2911
```

```
#c. 8.5 Maximum data points until 10
```

```
seq1to1000max10 <- seq(1, 10)
```

```
seq1to1000max10
```

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

9. Print a vector with integers not divisible by 3, 5, or 7

```
notdivisible <- Filter(function(i) { all(i %% c(3, 5, 7) != 0) }, seq(1, 100))
notdivisible
```

```
## [1] 1 2 4 8 11 13 16 17 19 22 23 26 29 31 32 34 37 38 41 43 44 46 47 52 53
## [26] 58 59 61 62 64 67 68 71 73 74 76 79 82 83 86 88 89 92 94 97
```

10. Generate a sequence backwards of the integers from 1 to 100

```
backwardseq <- seq(100, 1)
backwardseq
```

```
## [1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83
## [19] 82 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65
## [37] 64 63 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47
## [55] 46 45 44 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29
```

```
## [73] 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11
## [91] 10 9 8 7 6 5 4 3 2 1
```

11. List multiples of 3 or 5 below 25 and find the sum

```
multiples3or5 <- Filter(function(x) x %% 3 == 0 | x %% 5 == 0, seq(1, 24))
summultiples <- sum(multiples3or5)
multiples3or5
```

```
## [1] 3 5 6 9 10 12 15 18 20 21 24
```

```
summultiples
```

```
## [1] 143
```

a. How many data points?

```
length(multiples3or5)
```

```
## [1] 11
```

12. Enter the statement and describe the output

```
x <- {0 + x + 5 + }
```

Output: An error occurs because x is not defined.

13. Set up a vector named score

```
score <- c(72, 86, 92, 63, 88, 89, 91, 92, 75, 75, 77)
x2 <- score[2]
x3 <- score[3]
x2
```

```
## [1] 86
```

```
x3
```

```
## [1] 92
```

14. Create a vector with NA

```
a <- c(1, 2, NA, 4, NA, 6, 7)
# a. Change NA to 999
print(a, na.print="-999")
```

```
## [1] 1 2 -999 4 -999 6 7
```

15. Special function call

```
name <- readline(prompt="Input your name: ")

## Input your name:
age <- readline(prompt="Input your age: ")

## Input your age:
print(paste("My name is", name, "and I am", age, "years old.))

## [1] "My name is  and I am  years old."
print(R.version.string)  # Output: "R version 4.4.1 (2024-06-14 ucrt)"

## [1] "R version 4.4.1 (2024-06-14 ucrt)"
```