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) # Output: 34

2. Find the reciprocal of the values for age

reciprocalage <- 1 / age reciprocalage # Output: A vector of reciprocals of age.

3. Assign new_age

newage <- c(age, 0, age) newage # Output: A vector containing age, 0, and age again.

4. Sort the values for age

sortedage <- sort(age) sortedage # Output: Sorted age values, 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 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 # Output: Minimum age value, 17. maxage # Output: Maximum age value, 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) # Output: 12

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

doubleddata <- data * 2 doubleddata # Output: The values of the data doubled, 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) # 8.1 Integers from 1 to 100 seq20to60 <- seq(20, 60) # 8.2 Numbers from 20 to 60 mean20to60 <- mean(seq(20, 60)) # 8.3 Mean of numbers from 20 to 60 sum51to91 <- sum(seq(51, 91)) # 8.4 Sum of numbers from 51 to 91 seq1to1000 <- seq(1, 1000) # 8.5 Integers from 1 to 1000

a. How many data points?

length(seq1to100) # Output: 100 length(seq20to60) # Output: 41 length(mean20to60) # Output: 1 length(sum51to91) # Output: 1

b. Output of sequences

seq1to100 #Output: Integers from 1 to 100 seq20to60 #Output: Numbers from 20 to 60 mean20to60 #Output: 40 sum51to91 #Output: 2911

#c. 8.5 Maximum data points until 10 seq1to1000max10 <- seq(1, 10) seq1to1000max10 # Output: Integers from 1 to 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 # Output: A vector of filtered integers, 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 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 # Output: A vector counting backward, from 100 down to 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 # Output: A vector of multiples 3 and 5 below 25, 3 5 6 9 10 12 15 18 20 21 24 summultiples # Output: Sum of multiples, 143.

a. How many data points?

length(multiples3or5) # Output: Number of multiples, 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] # Output: 86 x3 <- score[3] # Output: 92 x2 x3

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") # Output: NA changed into -999.

15. Special function call

name <- readline(prompt="Input your name:") age <- readline(prompt="Input your age:") print(paste("My name is", name, "and I am", age, "years old.")) print(R.version.string) # Output: "R version 4.4.1 (2024-06-14 ucrt)"