

# RWorksheet\_Elizalde#1

Christian Alimace Elizalde

2024-09-17

#1. Set up a vector named age, consisting of 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.

```
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
```

#b. Write the R code and its output.

#Find the reciprocal values for age.

```
library("MASS")
reciprocalOfage <- 1/age
fractions(reciprocalOfage)
```

```
## [1] 1/34 1/28 1/22 1/36 1/27 1/18 1/52 1/39 1/42 1/29 1/35 1/31 1/27 1/22 1/37
## [16] 1/34 1/19 1/20 1/57 1/49 1/50 1/37 1/46 1/25 1/17 1/37 1/42 1/53 1/41 1/51
## [31] 1/35 1/24 1/33 1/41
```

#Assign also new\_age <- c(age, 0, age)

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

```
## [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
```

#sort the values for age

```
sort(age)
```

```
## [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
```

#Find the minimum and maximum value for age

```
min(age)
```

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

```
max(age)
```

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

#Set up a vector named data, consisting of 2.4, 2.8, 2.1, 2.5, 2.4, 2.2, 2.5, 2.3, 2.5, 2.3, 2.4, and 2.7.

```

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)

#How many data points?
length(data)

## [1] 12

#Generates a new vector for data where you double every value of the data.
doubled_data <- 2 * (data)
doubled_data

## [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

#Generate a sequence for the following scenario:8.1 Integers from 1 to 100.
sequence_1to100 <- seq(1:100)

#8.2 Numbers from 20 to 60
sequence_20to60 <- (20:60)

#8.3 Mean from 20 to 60
mean_sequence <- mean(sequence_20to60)
mean_sequence

## [1] 40

#8.4 Sum of numbers from 51 to 91
summation <- sum(51:91)
summation

## [1] 2911

#8.5 Integers from 1 to 1,000
sequence_1to1000 <- seq(1:1000)

#a. how many data points from 8.1 to 8.4?
length(sequence_1to100) + length(sequence_20to60) + length(mean_sequence) + length(summation)

## [1] 143

#9. *Print a vector with the integers between 1 and 100 that are not divisible by 3, 5 and 7 using filter
option.
Filter(function(i) { all(i %% c(3,5,7) != 0) }, seq(100))

## [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.
backwards <- seq(100:1)
backwards

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

```

```
## [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
```

#11. List all the natural numbers below 25 that are multiples of 3 or 5. Find the sum of these multiples.

```
below_25 <- seq(1:24)
multiplesof3or5 <- below_25[below_25 %% 3 == 0 | below_25 %% 5 == 0]
multiplesof3or5
```

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

```
sum(multiplesof3or5)
```

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

#a. How many data points from 10 to 11?

```
length(backwards) + length(multiplesof3or5)
```

```
## [1] 111
```

#12

```
#x <- {0 + x + 5 + } this code produces error
```

#13 \*Set up a vector named score, consisting of 72, 86, 92, 63, 88, 89, 91, 92, 75, 75 and 77. To access individual elements of an atomic vector, one generally uses the x[i] construction.

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

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

```
score[3]
```

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

#\*Create a vector a = c(1,2,NA,4,NA,6,7).

```
a = c(1,2,NA,4,NA,6,7)
```

#a. Change the NA to 999 using the codes print(a,na.print="-999").

```
print(a,na.print="-999")
```

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

#15. A special type of function calls can appear on the left hand side of the assignment operator as in > class(x) <- "foo".

```
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)
```

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