

RWorksheet_noblezada#1

Necel Kate Noblezada

2024-09-04

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)
```

a. How many data points?

answer: 34 data points

```
length(age)
```

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

2. Find the reciprocal of the values for age.

```
reciprocal<-1/age  
reciprocal
```

```
## [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 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
```

What happen to the new_age?

answer: It duplicates the age vector with 0 in the middle.

4. 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
```

5. Find the minimum and maximum value for age.

```
min(age)

## [1] 17
max(age)

## [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?

answer: 12 data points

```
length(data)

## [1] 12
```

7. Generate a new vector for data where you double every value of the data.

```
new_vector <- data
data * 2

## [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
```

What happened to the data?

answer: The data vector results in a new vector where each element is twice the original value.

8. Generate a sequence for the following scenario:

8.1 - Integers from 1 to 100.

```
a <- seq(1:100)
```

8.2 - Numbers from 20 to 60.

```
l <- seq(20,60)
```

8.3 - Mean of numbers from 20 to 60.

```
i <- mean(20,60)
```

8.4 - Sum of numbers from 51 to 91.

```
k <- sum(51:91)
```

8.5 - Integers from 1 to 1,000.

```
t <- seq(1:1000)
```

a. How many data points from 8.1 to 8.4?

answer: 143 data points

```
length1 <- length(a)
length2 <- length(l)
length3 <- length(i)
length4 <- length(k)
sum (length1+length2+length3+length4)
```

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

b. Write the R code and its output from 8.1 to 8.4.

```
a <- seq(1:100)
l <- seq(20,60)
i <- mean(20,60)
k <- sum(51:91)
```

c. For 8.5 find only maximum data points until 10.

```
t <- 1:1000
answer <- max(t[t <= 10])
answer
```

```
## [1] 10
```

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.

```
rev(seq(1:100))
```

```
## [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 all the natural numbers below 25 that are multiples of 3 or 5.

```
upper_limit <- 25
num <- 1:(upper_limit - 1)
multiples <- num[num %% 3 == 0 | num %% 5 == 0]
multiples
```

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

a. How many data points from 10 to 11?

answer: 11 data points

```
length(multiples)
```

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

b. Write the R code and its output from 10 and 11.

12.

Describe the output.

answer: the line is incomplete. Thus, it caused an error.

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]
```

```
print(x2)
```

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

```
print(x3)
```

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

14. Create a vector

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

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

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

b. describe the output.

answer: The output displays the vector a with NA values shown as -999, while the actual vector remains unchanged with NA values.

15.

```
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)"
```

What is the output of the above code?

answer: The output of the code is a message stating the user's name and age followed by the R version string, e.g., "My name is "name" and I am "age" years old." and "R version 4.3.1 (2024-06-01)".