

SIMPRON, MECHELLE. BSIT 2-A

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.

a. How many data points?

ANSWER: there are 34 data points.

b. Write the R code and its output.

**ANSWER: 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)**

Age

OUTPUT: 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

[24] 25 17 37 42 53 41 51 35 24 33 41

Write the R code and its output.

2. Find the reciprocal of the values for age.

OUTPUT:

**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)**

reciprocal <- 1/age

reciprocal

answer: [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`

What happen to the `new_age`?

ANSWER: it displayed the values of "age", space, and then followed by the same values.

4. Sort the values for `age`.

Write the R code and its output.

ANSWER: `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

[24] 41 41 42 42 46 49 50 51 52 53 57

5. Find the minimum and maximum value for `age`.

Write the R code and its output.

`max(age)`

`min(age)`

answer: maximum is 57 and minimum is 17.

6. 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.

a. How many data points?

ANSWER: 12

b. Write the R code and its output.

```
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)
```

```
data
```

```
output:[1] 2.4 2.8 2.1 2.5 2.4 2.2 2.5 2.3 2.5 2.3 2.4 2.7
```

7. Generates a new vector for data where you double every value of the data. | What happen to the data?

ANSWER:

```
dbld_data <- data*2
```

```
dbld_data
```

answer: the data were doubled.

8 . Generate a sequence for the following scenario:

8.1 Integers from 1 to 100.

answer:

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

8.2 Numbers from 20 to 60.

answer:

```
seq(20,60)
```

8.3 Mean of numbers from 20 to 60

answer:

```
print(mean(20:60))
```

8.4 Sum of numbers from 51 to 91

answer:

```
print(sum(51:91))
```

#8.5 Integers from 1 to 1,000

#answer:

seq(1:1000)

a. How many data points are there in 8.1 and

answer: there are 43 data points from 8.1 to 8.4

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

answer:

seq(1:100)

output:

[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19

[20] 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38

[39] 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57

[58] 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76

[77] 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95

[96] 96 97 98 99 100

seq(20,60)

output

[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 45

[27] 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60

print(mean(20:60))

output:

40

print(sum(51:91))

output:

2911

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

answer: max(1:10)

[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))` Write the R code and its output.

ANSWER: `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 58

[27] 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. Write the R code and its output.

ANSWER:

`seq(100,1)`

output:

[1] 100 99 98 97 96 95 94 93 92 91 90 89 88 87 86 85 84 83 82

[20] 81 80 79 78 77 76 75 74 73 72 71 70 69 68 67 66 65 64 63

[39] 62 61 60 59 58 57 56 55 54 53 52 51 50 49 48 47 46 45 44

[58] 43 42 41 40 39 38 37 36 35 34 33 32 31 30 29 28 27 26 25

[77] 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6

[96] 5 4 3 2 1

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

answer:

`sum(((1 : 25)[((1 : 25)%%3 == 0) | ((1:25)%%5 == 0)])`

[1] 168

a. How many data points from 10 to 11?

answer: 101 data points.

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

answer: 101

10

`seq (100,1)`

output:

```

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

```

12. Statements can be grouped together using braces '{' and '}'. A group of statements is sometimes called a block. Single statements are evaluated when a new line is typed at the end of the syntactically complete statement. Blocks are not evaluated until a new line is entered after the closing brace. Enter this statement: { x <- 0+ x + 5 + } Describe the output.

answer: The output is error. It means that the curly brace is not appropriate with the statement.

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

Find x[2] and x[3]. Write the R code and its output.

ANSWER: score <- c(72, 86, 92, 63, 88, 89, 91, 92, 75, 75, 77)

x[2] = 86 and x[3] = 92

14. 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")

answer:

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

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

output: [1] 1 2 -999 4 -999 6 7

b. Write the R code and its output. Describe the output.

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

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

output: [1] 1 2 -999 4 -999 6 7

The NA was replaced by -999 because we assigned -999 to NA.

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

Follow the codes below:

```
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)
```

What is the output of the above code?

ANSWER: [1] "My name is mechelle simpron and I am 20 years old."