

# RWorksheet\_Quebral#1

Myles Andrei Quebral

2024-09-20

#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?

# 34 data points

# b. Write the R code and its 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)
print(age)
```

# 2. Find the reciprocal of the values for age. # Write the R code and its output.

```
1 / age
```

# 3. Assign also new\_age <- c(age, 0, age). # What happen to the new\_age

```
# It contains the value for two age with 0 in the middle
```

# 4. Sort the values for age.

```
sort(age)
```

# 5. Find the minimum and maximum value for age. # Write the R code and its output.

```
min(age)
max(age)
```

# 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?

```
# 12 data points
```

# 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)

```
print(data)
```

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

```
# Doubled by itself.
```

```
num <- data*2
print(num)
```

# 8. Generate a sequence for the following scenario:

# 8.1 Integers from 1 to 100.

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

# 8.2 Numbers from 20 to 60

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

# 8.3 Mean of numbers from 20 to 60

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

# 8.4 Sum of numbers from 51 to 91

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

# 8.5 Integers from 1 to 1,000

```
seq(1:1000)
```

# a. How many data points from 8.1 to 8.4?\_\_\_\_\_

```
# 143 data points
```

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

```
seq(1:100)
seq(20,60)
mean(20:60)
sum(51:91)
```

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

```
numbers <- seq(1:1000)
max_num <- numbers[1:10]
max(max_num)
```

# 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.

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

# 10. Generate a sequence backwards of the integers from 1 to 100. # Write the R code and its output.

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

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

```
numb <- 1:24
num1 <- numb[numb %% 3==0 | numb %% 5==0]
print(num1)
```

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

```
# 11 data points
```

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

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

numb <- 1:24
num1 <- numb[numb %% 3==0 | numb %% 5==0]
print(num1)
```

# 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:  $x \leftarrow \{0 + x + 5 + \}$

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

#It has an error, the x isn't initialize and the + operator doesn't have a value to add.

# 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.

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

# 14. \*Create a vector  $a = c(1,2,NA,4,NA,6,7)$ . # a. Change the NA to 999 using the codes `print(a,na.print="-999")`.

# b. Write the R code and its output.

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

```
# Describe the output.
```

```
# The NA words are replaced by -999 and must be smaller "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?

```
# User prompted on what is the output in the name and age, and it created
# a file named 'My name is Myles and I am 13 years old..Rd'. The print(R.version.string)
# showed the version of R .
```