

Practice 3

Maria

02/07/2021

Exercise 1

In order to do the exercise, use the following code structure:

```
if () {  
  
  } else {  
  
  }
```

Define 2 numbers: “a” as 33 and “b” as 33.

Compare “a” and “b”

if “b” greater than “a” print “b is greater than a”.

if “a” is equal to “b” print “a and b are equal”

```
## [1] "a and b are equal"
```

Exercise 2

Create the vector “my_vector”, which includes normally distributed 30 random numbers.

To create that vector, you can use the following code.

```
my_vector <- rnorm(n = 30) # n = length
```

Create the empty character vector “my_result”, with length = 1.

If the average of the vector “my_vector” is greater than 20, in “my_result” vector write “My mean is

great”.

If the average of “my_vector” is less than or equal to 20, in “my_result” vector write “My mean is not so great”.

```
## [1] "My mean is not so great"
```

Exercise 3

Write a loop that iterates over the numbers 1 to 7 and prints the cube of each number using.

```
## [1] 1
## [1] 4
## [1] 9
## [1] 16
## [1] 25
## [1] 36
## [1] 49
```

Exercise 4

Define an empty numeric vector s_n of size 25

Create a loop to calculate the sum of the following numbers

1

1 + 2

1 + 2 + 3

.

1 + 2 + 3 + ... + 25

Save the calculated sums in the corresponding positions of s_n vector

s_n[1] = 1

s_n[2] = 1+2

s_n[3] = 1+2+3

```
## [1] "s_n = 1 3 6 10 15 21 28 36 45 55 66 78 91 105 120"
```

Exercise 5

Create a matrix 100 X 10, which contains the numbers from 1 to 1000 and save it in the “my_mat” variable.

Write a loop that calculates the sum for each row of the matrix.

Note that this loop is much faster if you outside the loop create an empty vector of the right size.

Exercise 6

Create a following data frame

```
##  vector1 vector2 vector3
## 1      1    Odd    51
## 2      2   Loop    50
## 3      3     a     51
## 4      4     b     50
## 5      5     c     51
## 6      6     d     50
## 7      7     e     51
## 8      8     f     50
## 9      9     g     51
## 10     10    h     50
```

Write a loop that goes over the columns and reports the mean of the column if it is numeric and NA if it's a character vector.

```
## [1] 5.5 NA 50.5
```