

Week 1 Exercises

Patrick Sheehan

March 15, 2024

Please complete all exercises below WITHOUT using any libraries/packages.

Exercise 1

subheader

heavily nested

Assign 10 to the variable x. Assign 5 to the variable y. Assign 20 to the variable z.

For this first question I used the <- operator to assign the proper values to x/y/z.

```
x <- 10
y <- 5
z <- 20
```

Exercise 2

Show that x is less than z but greater than y.

Note: your output must be a SINGLE boolean, do not output a boolean for each expression.

For this exercise, I wrote x is less than z and z is greater than y. I used the & operator between the two arguments to show that both are true.

```
(x < z) & (z > y)
```

```
## [1] TRUE
```

Exercise 3

Show that x and y do not equal z.

For this question I used the != operator to write that x is not equal to z and that y is not equal to z. I used the & operator between them to show that both arguments are true.

Note: your output must be a SINGLE boolean, do not output a boolean for each expression.

```
(x != z) & (y != z)
```

```
## [1] TRUE
```

Exercise 4

Show that the formula $x + 2y = z$.

For this question I put parentheses where necessary and used the == operator to show that the left part of the equation equaled the right.

Note: your output must be a SINGLE boolean

```
(x + (2*y)) == z
```

```
## [1] TRUE
```

Exercise 5

I have created a vector (test_vector) of integers for you. Determine if any of x, y, or z are in the vector.

For this question I called the test_vector shown in the recorded lecture and then used the %in% operator to check if x,y,or z (use of | in between each argument) were in the test_vector.

Note: your output must be a SINGLE boolean, do not output a boolean for each expression.

```
test_vector <- c(1,5,11:22)
(x %in% test_vector) | (y %in% test_vector) | (z %in% test_vector)
```

```
## [1] TRUE
```

Exercise 6

Show which value is contained in the test vector. To do this you will need to create an element-wise logical vector using operators. `x == vector`. Once you have done that you will need to use slicing to return all indices that have matches.

For this exercise I used the or operator (|) to separate x/y/z == test_vector. By having test_vector outside of the brackets, I am checking to see if the elements included in each argument are included in the test_vector.

Note: your output should be two integers

```
test_vector[(x==test_vector) | (y==test_vector) | (z==test_vector)]
```

```
## [1] 5 20
```