Basic R Exercises

A. Operations

1. Arithmetic Operators

```r
a <- 10
b <- 4
a + b
```

1.1 Addition

```
## [1] 14
```

```r
a - b
```

1.2 Subtraction

```
## [1] 6
```

```r
a * b
```

1.3 Multiplication

```
## [1] 40
```

```r
a/b
```

1.4 Division

```
## [1] 2.5
```

```
a %% b
```

1.5 Modulus

```
## [1] 2
```

```
# QUESTION 1
# Fix the code below to get an output of 1.
# Hint: You can change the values of the variables to get the output.
# c <- 10 # Original input
c <- 9 # New input
d <- 2
c %% d
```

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

```
## Challenge
# ---
# Question: Fix the code below to get an output of 25.
# Hint: You can change one of the operators to the subtraction operator - and get an answer
# ---
# OUR CODE GOES BELOW
#
a <- 5
b <- 4
# b - a + 10 / 2 * a * 3 + 10 + 59 # Original
b - a + 10 / 2 * a * 3 + 10 - 59
```

```
## [1] 25
```

2. Relational Operators

```
## Example
# Question: Applying less than operator using the x and y variables
# Hint: For the following examples, we will use the previous assigned values of x and y using the opera

# OUR CODE GOES BELOW
#

# Lets find what x is
x <- 3

# Lets find what y is
y <- 4

# And now find out whether x is less than y.
x < y
```

```
## [1] TRUE
## Challenge
# ---
# Question: Find out whether x is greater than y using the Greater than operator >
# ---
x > y
```

2.2 Greater than Code Example

```
## [1] FALSE
```

```
## Challenge
# ---
# Question: Find whether x is less than or equal to y using the less than or equal to operator <=
# ---
x <= y
```

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

```
## Challenge
# ---
# Question: Find out below whether x is greater than or equal to y using the greater than or equal to o
# ---
x >= y
```

2.4 Greater than or Equal to Code Example

```
## [1] FALSE
```

```
## Challenge
# ---
# Question: Find out whether y is equal to y using the equal to operator ==, below
# ---
y == y
```

2.5 Equal to Code Example

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

```
## Challenge
# ---
# Question: Find out whether x is not equal to y using the not equal to operator !=, below
# ---
x != y
```

2.6 Not Equal to Code Example

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

3. Logical Operators

Logical operators are applicable only to vectors of type logical, numeric or complex. Each element of the first vector is compared with the corresponding element of the second vector. The result of comparison is a Boolean value.

```r
## Example
# ---
# Question: Lets create two vectors v and t
# ---
#
v <- c(3,1,TRUE,2+3i)
t <- c(4,1,FALSE,2+3i)

# Then use the element-wise logical and operator & as follows
v&t
```

3.1 Element-wise Logical AND operator Code Example

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

```r
## Example
# ---
# Question: Create again two vectors v and t
# ---
#
v <- c(0,0,TRUE,2+2i)
t <- c(0,3,TRUE,2+3i)

# Then use the element-wise logical or operator | below
v|t
```

3.2 Element-wise Logical OR operator Code Example

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

```r
## Example
# ---
# Question: Let create two vectors v and t
# ---
```

```
#
v <- c(3,1,TRUE,2+3i)
t <- c(4,1,FALSE,2+3i)

# This time, use the logical not operator ||
v||t
```

3.3 The Logical OR operator Code Example

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

```
## Example
# ---
# Question: Create two vectors v and t then use the logical AND operator
# ---
#
v <- c(3,1,TRUE,2+3i)
t <- c(4,1,FALSE,2+3i)

v && t
```

3.4 The Logical AND operator Code Example

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

4. Assignment Operators

```
## Example
# ---
# Question: Create variables v1, v2 and v3, assigning them with vectors
# using the left assignment operators v1, v2 and v3
# ---
#
v1 <- c(3,1,TRUE,2+3i)
v2 <<- c(3,1,TRUE,2+3i)
v3 = c(3,1,TRUE,2+3i)

# Then we print out v1 below
v1
```

4.1 Left Assignment Operator Code Example

```
## [1] 3+0i 1+0i 1+0i 2+3i
```

```
# And print out v2 below
v2
```

```
## [1] 3+0i 1+0i 1+0i 2+3i
```

```
# And now print out v3 below
v3
```

```
## [1] 3+0i 1+0i 1+0i 2+3i
```

```


## Challenge
# ---
# Question: Use the right asignment operators to assign vectors to the variables v1 and v2
# ---
#
c(3,1,TRUE,2+3i) -> v1
c(3,1,TRUE,2+3i) ->> v2

# Then print out variable v1 and see what has happened
# ---
v1
```

4.2 Right Assignment Operator Code Example

```
## [1] 3+0i 1+0i 1+0i 2+3i
```

```
# And also print out variable v2
# ---
v2
```

```
## [1] 3+0i 1+0i 1+0i 2+3i
```

5. Variable Assignment

Variables can be assigned values using leftward, rightward and equal to operator.

```


## Example
# ---
# Question: Use the right assignment operators to assign the vectors to the respective variables as sho
# ---
# OUR CODE GOES HERE
#
variable.1 = c(3,4,5,6)
variable.2 <- c("Hello"," there")
c(TRUE,2) -> variable.3
```

```r
# Uncomment the following lines to see what has happened
# ---
#
variable.1
```

5.1 Right Assignment Operator Code Example

```
## [1] 3 4 5 6
```

```r
variable.2
```

```
## [1] "Hello"  " there"
```

```r
variable.3
```

```
## [1] 1 2
```

6. Basic Data Types

There are several basic data types in R which are of frequent occurrence.

```r
## Example
# ---
# Question: To learn about the numeric data type,
# lets assign the value 62.4 to the variable m as shown below
# ---
# OUR CODE GOES BELOW
#
m = 62.4

# Print out the variable's value below
# ---
m
```

6.1 Numeric Data Type Code Example

```
## [1] 62.4
```

Assigning a decimal value for any variable m, m will become a numeric type as shown above.

6.2 Integer Data Type Code Example Any integer in R is created by invoking the as.integer() function as shown below:

```
## Example
# ---
# Let's now create an integer 3 and assign it to the variable n
# ---

n = as.integer(3)

# Then print n below so as to see what is stored in n
n
```

```
## [1] 3
```

```
# Using another example, lets create convert 3.14 to an interger
# and assign the converted value to the variable p
# ---
p = as.integer(3.14)


# And print out the value p so as to see the value that has been assigned to p
# ---
p
```

```
## [1] 3
```

```


## Example
# ---
# We can also assign a complex number and assign it to the variable k just as shown below
# ---
#
k = 1 + 2i

# Now lets print out k below
k
```

6.3 Complex Data Type Code Example

```
## [1] 1+2i
```

6.4 Logical Data Type Code Example A logical value is created when comparison between variables is done as shown:

```
## Example
# ---
# To create a logical value we are first going to create two variables x and y variables
# ---
#
x = 4;  y = 6

# Now we check whether x is greater than y
```

```r
# ---
#
z = x > y

# And then print out the logical value below
# ---
z
```

```
## [1] FALSE
```

6.5 Character Data Type Code Example A character value can also be created and stored in a variable g.

```r
## Example
# ---
# Convert the value 62.48 to a string and store it a variable g
# ---
#
g = as.character(62.48)

# Then print the character string g
# ---
g
```

```
## [1] "62.48"
```