CODE:

**Arithmetic with R**

In its most basic form, R can be used as a simple calculator. Consider the following arithmetic operators:

* Addition: +
* Subtraction: -
* Multiplication: \*
* Division: /
* Exponentiation: ^
* Modulo: %%

The last two might need some explaining:

* The ^ operator raises the number to its left to the power of the number to its right: for example 3^2 is 9.
* The modulo returns the remainder of the division of the number to the left by the number on its right, for example 5 modulo 3 or 5 %% 3 is 2.

With this knowledge, follow the instructions to complete the exercise.

# An addition

5 + 5

# A subtraction

5 - 5

# A multiplication

3 \* 5

# A division

(5 + 5) / 2

# Exponentiation

2^5

# Modulo

28%%6

# Variable assignment

A basic concept in (statistical) programming is called a **variable**.

A variable allows you to store a value (e.g. 4) or an object (e.g. a function description) in R. You can then later use this variable's name to easily access the value or the object that is stored within this variable.

You can assign a value 4 to a variable my\_var with the command

my\_var <- 4

# Assign the value 42 to x

x <- 42

# Print out the value of the variable x

x

# Variable assignment (2)

Suppose you have a fruit basket with five apples. As a data analyst in training, you want to store the number of apples in a variable with the name my\_apples.

##### Instructions

**100 XP**

* Type the following code in the editor: my\_apples <- 5. This will assign the value 5 to my\_apples.
* Type: my\_apples below the second comment. This will print out the value of my\_apples.
* Submit your answer, and look at the output: you see that the number 5 is printed. So R now links the variable my\_apples to the value 5.

# Variable assignment (3)

Every tasty fruit basket needs oranges, so you decide to add six oranges. As a data analyst, your reflex is to immediately create the variable my\_oranges and assign the value 6 to it. Next, you want to calculate how many pieces of fruit you have in total. Since you have given meaningful names to these values, you can now code this in a clear way:

my\_apples + my\_oranges

##### Instructions

**100 XP**

* Assign to my\_oranges the value 6.
* Add the variables my\_apples and my\_oranges and have R simply print the result.
* Assign the result of adding my\_apples and my\_oranges to a new variable my\_fruit.
* # Assign a value to the variables my\_apples and my\_oranges
* my\_apples <- 5
* my\_oranges <- 6
* # Add these two variables together
* my\_apples + my\_oranges
* # Create the variable my\_fruit
* my\_fruit <- my\_apples + my\_oranges

# Apples and oranges

Common knowledge tells you not to add apples and oranges. But hey, that is what you just did, no :-)? The my\_apples and my\_oranges variables both contained a number in the previous exercise. The + operator works with numeric variables in R. If you really tried to add "apples" and "oranges", and assigned a text value to the variable my\_oranges (see the editor), you would be trying to assign the addition of a numeric and a character variable to the variable my\_fruit. This is not possible.

* Submit the answer and read the error message. Make sure to understand why this did not work.
* Adjust the code so that R knows you have 6 oranges and thus a fruit basket with 11 pieces of fruit.



# Assign a value to the variable my\_apples

my\_apples <- 5

# Fix the assignment of my\_oranges

my\_oranges <- 6

# Create the variable my\_fruit and print it out

my\_fruit <- my\_apples + my\_oranges

my\_fruit

**Basic data types in R**

R works with numerous data types. Some of the most basic types to get started are:

* Decimal values like 4.5 are called **numerics**.
* Whole numbers like 4 are called **integers**. Integers are also numerics.
* Boolean values (TRUE or FALSE) are called **logical**.
* Text (or string) values are called **characters**.

Note how the quotation marks in the editor indicate that "some text" is a string.

Change the value of the:

* my\_numeric variable to 42.
* my\_character variable to "universe". Note that the quotation marks indicate that "universe" is a character.
* my\_logical variable to FALSE.

Note that R is case sensitive!

# Change my\_numeric to be 42

my\_numeric <- 42.5

# Change my\_character to be "universe"

my\_character <- "some text"

# Change my\_logical to be FALSE

my\_logical <- TRUE

# Change my\_numeric to be 42

my\_numeric <- 42

# Change my\_character to be "universe"

my\_character <- "universe"

# Change my\_logical to be FALSE

my\_logical <- FALSE

# What's that data type?

Do you remember that when you added 5 + "six", you got an error due to a mismatch in data types? You can avoid such embarrassing situations by checking the data type of a variable beforehand. You can do this with the class() function, as the code in the editor shows.

Complete the code in the editor and also print out the classes of my\_character and my\_logical.

# Declare variables of different types

my\_numeric <- 42

my\_character <- "universe"

my\_logical <- FALSE

# Check class of my\_numeric

class(my\_numeric)

# Check class of my\_character

class(my\_character)

# Check class of my\_logical

class(my\_logical)