Intro

Exercise

In the RStudio you should type R code to solve the exercises. When you hit the 'Run' button, every line of code is interpreted and executed by R and you get a message whether or not your code was correct. The output of your R code is shown in the console in the lower “Console” corner.

R makes use of the # sign to add comments, so that you and others can understand what the R code is about. Just like Twitter! Comments are not run as R code, so they will not influence your result. For example, “#Calculate 3 + 4” in the editor is a comment.

You can also execute R commands straight in the console. This is a good way to experiment with R code

Instruction

* In the Script “Part1.R” there is already some sample code. Can you see which lines are actual R code and which are comments?
* Add a line of code that calculates the sum of 6 and 12, and hit the 'Run' button.

#Arithmetic With R

**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 below to complete the exercise.

##Instructions

* Type 2^5 in the editor to calculate 2 to the power 5.
* Type 28 %% 6 to calculate 28 modulo 6.
* Click 'Run' for each and have a look at the R output in the console.
* Note how the # symbol is used to add comments on the R code.

**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

###instruction

Over to you: complete the code in the editor such that it assigns the value 42 to the variable x in the editor. Click 'Run'. Notice that when you ask R to print x, the value 42 appears.

# 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

* 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.
* Click 'Run' on each line, and look at the console: 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

###Instruction

* 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.

# 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.

###Instructions

* Click 'Run' on each 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.

**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**.
* Natural 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 on the right indicate that "some text" is a character.

##Instructions

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!

# 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 this section shows.

### Instruction

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