11 Feb 2023, Saturday

R Programming Lab
Assignment 5 with Concepts
Miscellaneous

Functions:

```
myfun <- function() {
  # No instructions
}
```

The new function() { } is assigned to a new object called myfun.

1. Write a function that takes one input argument x and returns x^power.

2. Create a data frame *persons*:

```
##             height  age    gender
## Renate      155     33.07    1
## Kurt        175     22.36    0
## Hermann     171     18.68    0
## Anja        152     18.96    1
## Andrea      165     45.52    1
## Bertha      155     24.40    1
```

   a. create a new function find_tallest_man() with one input argument called persons to find the man with max height.

If statements:

Structure: **if (<condition>) { <action> }.**
The <condition> has to be a single logical TRUE or FALSE.
If <condition> is evaluated to TRUE, the <action> is executed.

If-else statements:

Structure: if (<condition>) { <action 1> } else { <action 2> }.
If <condition> is evaluated to TRUE, <action 1> is executed. Else <action 2> is executed.

3. Print via cat("x is smaller than 10") that the variable x is smaller than 10 in case this is true.

4. Start with a vector y <- 1:10. If the element in y is odd, add + 1. If even, leave it as it is. The result should look as follows:
   ## [1] 2 2 4 4 6 6 8 8 10 10
   a. Use ifelse() function

For Loops:

Basic usage: for (<value> in <values>) { <action> }

<value>: Current loop variable.
<values>: Set over which the variable iterates. Typically an atomic vector, but it can also be a list.
<action>: Executed for each <value> in <values>.

5. Create a matrix x:

   ```
   ##             Age  Size Weight
   ## Veronica  28   1.62     65
   ## Karl         35   1.53     59
   ## Miriam     13    1.83     72
   ## Peter       13    1.71     83
   ```

   Find the average values for all three columns using a for loop.

Use `next` and `break`

Additional control constructs exist which can be used in combination with loops.

next: Skip current loop iteration and continue with the next one.

break: Break from the entire loop (stop loop and jump to the end).

6. Try to solve the following one without using a computer. What is the final value of x?

```
x <- 1
for (i in 1:10)
{
   if (i <= 8) next
   x <- x + 1
}
```

7. Which value takes z after running the following code? Try to solve without executing the code again.

```
z <- 0
for (i in 1:5)
{
   z <- z + 1
   if (i > 2) break
   z <- z + 0.5
   next
}
```

Factors:

In R, objects for categorical data are called factors, while the different categories in which an observation can fall are called a level.
Factors can be created using the function factor().

```
factor(x = character(),levels, labels=levels,exclude=NA, ordered=is.ordered(x), nmax = NA)
```

`x`: a vector of data, usually taking a small number of distinct values.

`levels`: an optional vector of the unique values (as character strings) that 'x' might have taken. The default is the unique set of values taken by 'as.character(x)', sorted into increasing order *of 'x'*.

`labels`: *either* an optional character vector of labels for the levels (in the same order as 'levels' after removing those in 'exclude') *or* a character string of length 1. Duplicated values in `labels` can be used to map different values of `x` to the same factor level.

`exclude`: a vector of values to be excluded when forming the set of levels. This may be a factor with the same level set as `x` or should be a `character`.

`ordered`: logical flag to determine if the levels should be regarded as ordered (in the order given).

8. Create a character vector that contains the educational attainment of a series of people, coded by the highest university degree.
   degree <- c("master", "master", "bachelor", "phd", "bachelor", "master")

   Convert this character vector into an object of class factor. Observe the object of the class factor.

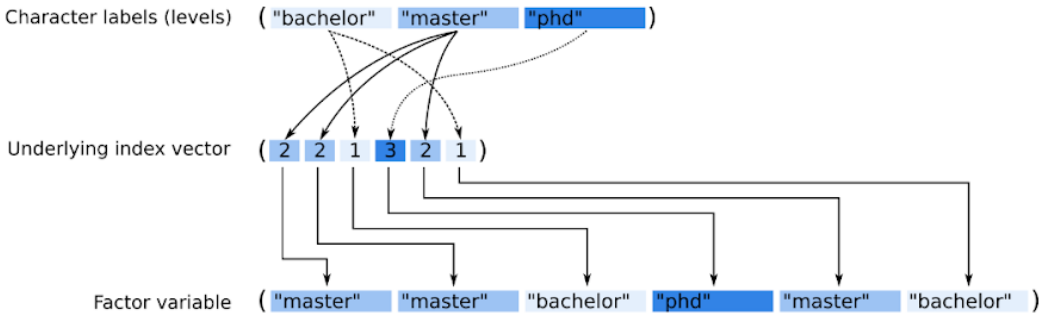

Figure 11.2: Graphical representation of the internal structure of a factor variable.

A factor object is nothing else than an *integer vector* (`[1] 2 2 1 3 1 2`) with an *additional attribute* `levels` containing the names of the three levels. This is nothing else as a 'lookup table'. The integer vector (index vector) tells us that the first entry of our object belongs to category 2, the second to category 2 and so far, and so on.

9. Create yourself a new object new_names <- c("BSc", "MSc", "PhD") (character vector)
    a. Coerce (convert) degree to integer using explicit coercion, store result on idx.
    b. Use idx and new_names to create the same as shown above.

Reading CSV

```
read.csv(file, header = TRUE, sep = ",", quote = "\"", dec = ".", fill = TRUE,
comment.char = "")
```

- `file`: File name or connection or URL with data to be read.
- `header`: Is there a header line with the variable/column names?
- `sep`: Column separator.
- `quote`: Quoting characters.
- `dec`: Decimal separator.
- `fill`: Should incomplete lines be filled with blanks?
- `comment.char`: Character indicating rows with comments.

10. Download the data set (homstart-innsbruck.csv) and read it using read.csv() Store the return on an object clim.
    a. What is the average (mean) January temperature in Innsbruck?
    b. What is the average (mean) temperature in July?
    c. Could we do (1) and (2) dynamically? Try to calculate the monthly average temperature for each month using a for loop.
    d. Try to create a boxplot which shows the climatology, namely plotting "temperature" (y axis) against the name of the month (x axis).

Writing CSV

Instead of only reading CSV files, we can also write CSV-like files. The functions work analogously to the reading functions and come with a wide range of arguments that can be set.
write.csv()

11. Use the “Innsbruck” data set. We first load the content of the file homstart-innsbruck.csv and write another text file into the text file.

    The content of another text file then looks as follows.

    ```
    "","year","month","temperature","precipitation"
    "1",1952,"Jan",-6,54.3
    "2",1952,"Feb",-2.8,67.6
    "3",1952,"Mar",3.7,55.5
    "4",1952,"Apr",11.1,37.1
    ```