Programming - Iteration

Data Science and Business Analytics

Exercise 1 – for and while

a) Create a for loop that goes from 10 to 20 and print in each iteration the current number times 2.
b) Create a function with 2 input arguments (`start_value` and `increment`)
   - Create a while loop that starts at `start_value` and counts back to 0 in steps of `increment`
   - In each step, `increment` increases with +1
   - `print` the value at the start of each iteration

Exercise 2 – Fibonacci

Fibonacci is a series where each number is the sum of the previous two numbers.
Starting with 0 and 1 (given), we get:
0, 1, 1, 2, 3, 5, 8 ,13, 21 … et cetera

a) Create a function which
   - given the input argument `length`
   - using a `for` loop
   creates a Fibonacci-series of length `length`.
   What is the outcome when `length` is 10?
   Why is it necessary to use a for loop here? Why could you not vectorize this function?
b) Create a function which
   - given the input argument max_value
   - using a while loop
   creates a Fibonacci-series for which all values are smaller than max_value.
   What is the outcome when max_value is 500?

Exercise 3 – Iteration: purrr

a) Create a vector 1 to 10
b) Create a function that calculates the mean over x random numbers from the standard normal distribution (`rnorm()`), where x is the input argument
c) Map over the vector created in a) to execute the function from b) over all values. Save the outcome to a numeric vector instead of a list

Exercise 4

a) Use `map_df()` to calculate the mean of all columns in `mtcars`, data available in the datasets-package.
b) Back to lectures 3 and 4 on data wrangling. Would you be able to do the same using `dplyr`-functions?
c) What are the advantages of `dplyr`?

Exercise 5

For this exercise we use the starwars dataset standardly available in the datasets-package of R. We first transform this dataset for this particular exercise:

```
library(tidyverse)
starwars_list <- starwars %>%
     transpose() %>%
     setNames(starwars$name)
```

Write a function called `get_movies()` which:
- gives by character in the input argument `characters`, the movies the character plays in
- inputs of the function are `starwars_list` and `characters` (possibly more than one)
- output of the function is a named list with for each character in `characters` an element: the list of movie names

Exercise 6

Again using the starwars dataset, but not the startwars list-transformation.
Write a function called get_tidy_movies() which:
- gives by character in the input argument characters, the movies the character plays in
- use dplyr-functions and map-functions (which map-function?)
- return the results in a tidy data.frame with in each row a character/movie-combination