

Exercise 2: Elementary data structures and functions

Niels Richard Hansen

December 8, 2016

The *Base R Cheat Sheet* may be helpful.

Elementary data structures

- Create a vector containing the integer sequence from 1 to 10. What is the type of the vector you created?
- Create a data frame with three columns called **name**, **gender** and **age**, with the **name** column containing the names “Niels”, “Peter”, “Hanne” and “Bente”, and the **age** column containing the values 41, 42, 34 and 49. The gender is up to you!
- Subset the data frame to contain the females only.
- Subset the data frame to contain those individuals older than 40 only.
- Select the data frame containing only the **name** and the **age** columns.
- Select the **age** column only. What is the type of the result?

Note that there are many possible ways to arrive at the solutions. Experiment with referring to columns by name, using the **\$** operator, and by index. Experiment with using logical vectors as filters for extracting the correct individuals.

Standard functions

- Compute the mean age of the individuals in the data frame constructed above.
- Use **summary** on the data frame.
- Use **lapply** to compute the type of each column in the data frame. Then use **sapply** – what is the difference?
- Use the **which** function to compute the row indices for the female individuals.

Increasing runs – a challenge

The following code generates a vector of 100 (pseudo) random variables, uniformly distributed between 0 and 1.

```
set.seed(08122016) ## So that we all get the same results
uniforms <- runif(100)
```

An *increasing run* of length k of a sequence x_1, \dots, x_n is a subsequence of (weakly) increasing neighboring elements

$$x_{i+1} \leq x_{i+2} \leq \dots \leq x_{i+k}.$$

A longest increasing run is a run with maximal length among all increasing runs. A single element is, by definition, an increasing run of length 1.

- Write a script that computes the length of the longest increasing run of the vector **uniforms**. (*Hint:* You can use a for-loop to traverse the vector and a counter variable to keep track of the length of the run.)
- Wrap the script up in a function, which returns a list of all the maximal runs.