Fachhochschule Südwestfalen
University of Applied Sciences

Programmierung R - Exercise

Code structure

April 26, SS 2022 || Hannah Behrens

Wir geben Impulse

Task 1

Your turn

1.1 When do you use a for-loop and when a while-loop?

1.2 Vector `x` passes the following for-loop:

```
x <- c(3, 14, 16, 32, 12, 2)
sum_e <- 0
for(e in 1:length(x)){
    sum_e <- sum_e + x[e]
}
sum_e
```

```
## [1] 79
```

Change this for-loop into a while-loop!

Task 1 - answer

1.1 A for-loop is used when you **know** the number of iterations. On the contrary, a while-loop is used when you **do not know** the exact number of iterations.

1.2

```
x <- c(3, 14, 16, 32, 12, 2)
sum_e <- 0
e <- 1 # initialization of e
while(e <= length(x)){
  sum_e <- sum_e + x[e]
  e <- e + 1 # increment e
}
sum_e
```

```
## [1] 79
```

Task 2

Your turn

Describe line by line what the following code executes. Describe the first runs for `r = 1` and `r = 2`.

```
1  A <- matrix(data = c(1,2,3,4,5,6,7,8,9,6,5,4,3,2,1,0), nrow = 4)
2  for(r in 1:nrow(A)){
3    for(c in 1:ncol(A)){
4      if(A[r,c] > 1){
5        A[r,c] <- A[c,r]
6      }
7    }
8  }
9  A
```

Task 2 - answer

```r
A <- matrix(data = c(1,2,3,4,5,6,7,8,9,6,5,4,3,2,1,0), nrow = 4) # matrix as input
for(r in 1:nrow(A)){ # for each row
  for(c in 1:ncol(A)){ # for each column
    if(A[r,c] > 1){ # if the element A[r,c] is greater than 1
      A[r,c] <- A[c,r] # substitute the element A[r,c] by A[c,r]
    }
  }
}
A
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    2    3    4
## [2,]    2    6    7    8
## [3,]    3    7    5    1
## [4,]    4    8    1    0
```

Tasks 3.1 and 3.2

Your turn

3.1 Imagine you want to save money for the vacations to go on a trip e.g. to Barcelona, France or elsewhere. Therefore, you are working three days per week during the semester. The vector `work_money` contains the money in euro you have earned after each day of work for your trip. (The hours you work per day vary, so the money you get after each day of work also varies.)
For your trip you need at least 750 euros.
If you have less than 420 euros, you will say: “So many more hours to work!”
If you have between 420 and 600 euros, you will say: “That is already great.” And in the other case you will be happy because you will say: “Only some hours of work left. Let‘s go!”
What will you say? Implement conditional expressions and loops if necessary.

```
1  work_money <- c(25, 25, 35, 40, 62, 48, 80, 50, 50, 40, 50)
```

3.2 Wrap your code from task 3.1 in a function and apply your function to `work_money`.

Task 3.1 - answer

3.1

```r
savings <- sum(work_money)
if(savings < 420){
  print("So many more hours to work!")
}else if(savings >= 420 && savings < 600){
  print("That is already great.")
}else{
    print("Only a few hours of work left. Let`s go!")
  }
```

```
## [1] "That is already great."
```

```r
## Alternative - Outlook - advanced version:
savings <- sum(work_money)
case_when(
  savings < 420 ~ "So many more hours to work!",
  savings >= 420 && savings < 600 ~ "That is already great.",
  TRUE ~ "Only a few hours of work left. Let`s go!"
)
```

Advanced version - R package `dplyr` (Wickham et al. (2021))

```
## [1] "That is already great."
```

Task 3.2 - answer

3.2

```
checkSavings <- function(money_vec){
savings <- sum(money_vec)
if(savings < 420){
  print("So many more hours to work!")
}else if(savings >= 420 && savings < 600){
  print("That is already great.")
}else{
    print("Only a few hours of work left. Let`s go!")
}
}
checkSavings(money_vec = work_money)
```

```
## [1] "That is already great."
```

Task 3.3

Your turn

3.3 So far, your function can handle one vector as input with the money you have saved. But what if you want to check your savings **and** the savings of your friends who want to go with you on the trip?

Write a function: It should handle the following data set called `work_money_friends_and_me` and check for every person the status of his/her savings.

Do not forget to comment your code!

```
work_money_friends_and_me <- data.frame(me = work_money,
                                        Kim = c(50, 25, 35, 30, 35, 48, 40, 55, 55, 40, 40),
                                        Max = c(35, 35, 35, 40, 42, 48, 80, 40, 40, 40, 40),
                                       Emma = c(55, 55, 35, 40, 62, 70, 85, 65, 65, 40, 35))
```

Task 3.3 - answer

```r
checkSavings2 <- function(df){
  num_persons <- ncol(df) # number of persons is equal to number of columns
  person_names <- colnames(df) # column names are equal to names of persons
  for(c in 1:num_persons){
    savings <- sum(df[,c]) # calculate the sum of each column
    print(paste("Check savings of: ", person_names[c]))
    if(savings < 420){
       print("So many more hours to work!")
    }else if(savings >= 420 && savings < 600){
       print("That is already great.")
    }else{
       print("Only a few hours of work left. Let`s go!")
    }
  }
}
checkSavings2(df = work_money_friends_and_me) # apply function to data set
```

```
## [1] "Check savings of:  me"
## [1] "That is already great."
## [1] "Check savings of:  Kim"
## [1] "That is already great."
## [1] "Check savings of:  Max"
## [1] "That is already great."
## [1] "Check savings of:  Emma"
## [1] "Only a few hours of work left. Let`s go!"
```

Task 3.3 - smart answer

```
checkSavingsFriends <- function(df){
  person_names <- colnames(df)
for(n in 1:ncol(df)){
  print(paste("Check savings of: ", person_names[n]))
  checkSavings(df[,n])
}
}
checkSavingsFriends(df = work_money_friends_and_me)
```

```
## [1] "Check savings of:  me"
## [1] "That is already great."
## [1] "Check savings of:  Kim"
## [1] "That is already great."
## [1] "Check savings of:  Max"
## [1] "That is already great."
## [1] "Check savings of:  Emma"
## [1] "Only a few hours of work left. Let`s go!"
```

Prefer `checkSavingsFriends()` to `checkSavings2()`.

Task 4 - 99 bottles of beer

For the introduction to *99 bottles of beer* see the slides of Prof. Dr. Buchwitz *Computational statistics with R*, Chapter 4 (Buchwitz 2021).
Write a function called `bottlesong()` that outputs the famous and complete lyrics of the song 99 Bottles of Beer:

99 bottles of beer on the wall, 99 bottles of beer. Take one down and pass it around, 98 bottles of beer on the wall.

.

.

.

2 bottles of beer on the wall, 2 bottles of beer. Take one down and pass it around, 1 bottle of beer on the wall.

1 bottle of beer on the wall, 1 bottle of beer. Take one down and pass it around, no more bottles of beer on the wall.

No more bottles of beer on the wall, no more bottles of beer. Go to the store and buy some more, 99 bottles of beer on the wall.

Task 5

Comment the functions presented here in this exercise.

References

Buchwitz, B. 2021. *Computational Statistics*. `https://bchwtz.github.io/bchwtz-cswr/`.

Wickham, Hadley, Romain François, Lionel Henry, and Kirill Müller. 2021. *Dplyr: A Grammar of Data Manipulation*. `https://CRAN.R-project.org/package=dplyr`.