Examination Advanced R Programming

Linköpings Universitet, IDA, Statistik

Course code and name:	732A94 Advanced R Programming
Date:	2017/11/29, 8–12
Teacher:	Krzysztof Bartoszek
Allowed aids:	The extra material is included in the zip file **exam_material.zip**
Grades:	A= $[19-20]$ points B= $[17-19)$ points C= $[12-17)$ points D= $[10-12)$ points E= $[8-10)$ points F= $[0-8)$ points **(FAIL)**
Instructions:	Write your answers in an R script file named **[your exam account].R** The R code should be complete and readable code, possible to run by copying directly into a script. Comment directly in the code whenever something needs to be explained or discussed. Follow the instructions carefully. There are **THREE** problems (with sub–questions) to solve.

Problem 1 (6p)

a) (1p) What are the potential risks if you start working in R with a non–empty workspace?
b) (2p) Using the function `package.skeleton()` create an R package that contains only one function. This function should take two numbers and return their sum.
c) (3p) The function `package.skeleton()` creates a number of files and directories. However, other directories are possible. List at least three other directories that can be present in an R package and write **IN YOUR OWN WORDS** a few (not more than three) sentences what is each ones role in an R package.

Problem 2 (8p)

READ THE WHOLE QUESTION BEFORE STARTING TO IMPLEMENT

a) (3p) In this task you should use object oriented programming in S3 or RC to write code that stores and manipulates information concerning a household. The first task is to implement a function called `create_household()` that returns a household object. The `create_household()` function should take two arguments: `address` and `number_of_devices` (the devices are meant to be ones that connect to the internet). The object shall represent each device by a unique integer–valued ID, the time it was connected to the internet (in seconds, initially 0) and a device specific number called `connection_rate`. For each device its `connection_rate` should be drawn from the uniform on $[0, 1]$ distribution—function `runif()`.

```
## S3 and RC call to create_household() function
household_1 <- create_household()(address="Circle Drive 3",number_of_devices=5)
```

b) (3p) Now implement a function called `simulate_internet_usage()` to simulate the connection duration for all the devices in the household. If you use RC, you can add the function to your household object that you implemented in Task 2.a if you like, otherwise use an ordinary function. The function should take two arguments (only one argument (the number of days) is needed if using RC), a household and the number of days. The simulation should loop for "number of days" times. For each day it should loop over each device and increase its internet uptime. The amount of seconds a device was connected on a given day is exponentially, function `rexp()`, distributed with rate equalling the device's `connection_rate`. The function should return a household with the updated connection times for each device in the household.

```
# S3 and RC call to simulate_internet_usage function
household_1 <- simulate_internet_usage(household_1,100)

# It is also OK to use the following OO style of simulation if using RC
# household_1$simulate_internet_usage(100)
```

c) (2p) Implement a plot function for your household objects. The function should use ggplot to visualize the connection times for each device. You are free to choose how to visualize it yourself!

```
# Plotting call
plot(household_1)
```

Problem 3 (6p)

a) (3p) An important operation in linear algebra is the scalar product between two vectors `x` and `y` of the same dimension d. This scalar product operation is defined as

$$\sum_{i=1}^{d} \texttt{x[i]} * \texttt{y[i]}.$$

Your task is to write a function, called `scalar_product()` that takes as its input two vectors and returns their scalar product. To do calculations you may only use ordinary addition, $+$, and multiplication, $*$. The function should check for correctness of input and react appropriately.

```
> x<-1:4
> y<-5:8
> scalar_product(x,y)
[1] 70
```

b) (1p) What is the complexity of your solution in terms of the input vectors' dimension?
c) (2p) If provided with two vectors R's matrix multiplication operation $\%*\%$ calculates their scalar product.

```
> x<-1:4
> y<-5:8
> x%*%y
     [,1]
[1,]   70
```

Implement a unit test that compares your scalar product implementation with R's $\%*\%$. Do not forget that $\%*\%$'s output is a **matrix**!