

Examination Advanced R Programming

Linköpings Universitet, IDA, Statistik

Course code and name:	732A94 Advanced R Programming
Date:	2023/10/29, 8–12
Teacher:	Krzysztof Bartoszek
Allowed aids:	The extra material is included in the zip file exam_help_material_732A94.zip
Grades:	A= [18 – 20] points B= [16 – 18) points C= [14 – 16) points D= [12 – 14) points E= [10 – 12) points F= [0 – 10) points
Instructions:	Write your answers in R scripts named according to the pattern [your exam account]_*.R The R code should be complete and readable code, possible to run by calling <code>source()</code> directly on your <code>*.R</code> files. 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 (5p)

- a) (3p)** What is an environment in R? Discuss in particular what happens when a function is called; how are variable names searched for; what happens when a variable that is not defined inside the function is referred to? Illustrate your discussion with short, relevant, pieces of code that can be run.
- b) (2p)** What is the difference between the operators `<-` and `<=<`? Illustrate your answer with short, relevant, pieces of code that can be run.

Problem 2 (11p)

READ THE WHOLE QUESTION BEFORE STARTING TO IMPLEMENT! Remember that your functions should **ALWAYS** check for correctness of user input! For each subquestion please provide **EXAMPLE CALLS!**

It is the X century and the Polish princess, Świętosława, is being married to king Eric the Victorious (of Sweden). As she has to cross the Baltic, both her dowry and personal belongings have to be stored upon the boat she will be travelling on. Not only does the boat has finite storage capacity, but also needs to transport a dowry of appropriate value. You are the courtier that is responsible for making sure everything is correctly packed on the boat. Fortunately, you managed to obtain a computer with R installed on it. After teaching yourself S3, S4 and RC programming you realized that you could write an object oriented program that will help you keep track of the boat's cargo.

a) (3p) In this task you should use object oriented programming in S3, S4 or RC to write code that keeps track of the boat's cargo. The boat has a finite capacity—24tons and the dowry has to be worth at least 50 grzywna (grzywna: weight measure used in Central/Eastern Europe from the early Middle Ages, equalling approximately 200g) of gold. You should record each element of the cargo, its weight; whether it is part of the dowry (in this case also its value) or whether it is part of the personal belongings. Depending on your chosen OO system you can do it through a constructor or by implementing a function `create_cargo()`. The constructing function should not take any arguments. The object should contain for each cargo element, the information describing it, in particular a unique id of it; what it is (e.g., necklace, fur, dog); its weight (a number); whether it is part of the dowry or personal property. If it is part of the dowry, then its value should also be stored. The `cargo` object should also store the current total value, and mass.

```
## example call to create a stocks object
my_cargo <- create_cargo() # S3
my_cargo <- cargo$new() # RC
```

b) (4p) Now implement a function called `add_cargo()` that adds an element to the cargo. The function should have four parameters describing the cargo: description, weight, type, and value (only if needed). Remember that the boat has a finite capacity. If a personal artifact would fill-up the capacity, but the value of the dowry is not met yet, then it cannot be taken.

```
## S3 and RC example calls
my_cargo <-add_cargo(my_cargo,"dog",0.03,"personal")
my_cargo <-add_cargo(my_cargo,"fur",0.1,"dowry",0.15)
## if using RC you may also call in this way
my_cargo$add_cargo("dog",0.03,"personal")
my_cargo$add_cargo("fur",0.1,"dowry",0.15)
```

c) (3p) Now implement a function called `remove_cargo()` that is called when you change your mind about something. The function should take only one parameter, the id of the object to be removed. If you do not have the object on the boat, your code should react accordingly. Also, you cannot remove a dowry element, if it would make the value of the dowry fall short of the required amount.

```
## S3 and RC example call
my_cargo<-remove_cargo(my_cargo,1) #but you are free to have a different id system
## if using RC you may also call in this way
my_cargo$remove_cargo(1) #but you are free to have a different id system
```

d) (1p) Implement a function that displays the state of the cargo on the boat. You are free to choose yourself how to report the state! This function has to also work directly with `print()`.

```
# calls to show state of the stock
my_cargo; print(my_cargo)
```

If you are interested you can read up (after the exam) on the history behind this assignment at <https://en.wikipedia.org/wiki/%C5%9Awi%C4%99tos%C5%82awa> and [https://en.wikipedia.org/wiki/Grzywna_\(unit\)](https://en.wikipedia.org/wiki/Grzywna_(unit)).

Problem 3 (4p)

a) (2p) Please implement a function that given a vector of fitted values, \hat{y}_i , and true observed values, y_i calculates the residual sum of squares

$$\text{RSS}(\hat{y}_i, y_i) = \sum_{i=1}^n (\hat{y}_i - y_i)^2.$$

Your implementation has to be done with a loop, in particular `sum()`, vector operators or similar **CANNOT** be used.

Please provide **EXAMPLE CALLS** to your function.

b) (1p) What is the computational complexity of your code?

c) (1p) Implement a unit test, using R's `%%` operator.