Examination Computational Statistics

Linköpings Universitet, IDA, Statistik

Course:	732A90 Computational Statistics
Date:	2021/01/12, 8–13
Teacher:	Krzysztof Bartoszek
Provided aids:	material in the zip file **exam_material_732A90.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:	Provide a detailed report that includes plots, conclusions and interpretations. If you are unable to include a plot in your solution file clearly indicate the section of R code that generates it. Give motivated answers to the questions. If an answer is not motivated, the points are reduced. Provide all necessary codes in an appendix. In a number of questions you are asked to do plots. Make sure that they are informative, have correctly labelled axes, informative axes limits and are correctly described. Points may be deducted for poorly done graphs. Name your solution files as: **[your id]_[own file description].[format]** If you have problems with creating a pdf you may submit your solutions in text files with unambiguous references to graphics and code that are saved in separate files. There are **TWO** assignments (with sub–questions) to solve. Provide a separate solution file for each assignment. Include all R code that was used to obtain your answers in your solution files. Make sure it is clear which code section corresponds to which question. If you also need to provide some hand–written derivations please number each page according to the pattern: Question number . page in question number i.e. Q1.1, Q1.2, Q1.3,. . ., Q2.1, Q2.2, . . ., Q3.1, ... Scan/take photos of such derivations preferably into a single pdf file but if this is not possible multiple pdf or .bmp/.jpg/.png files are fine. Please do not use other formats for scanned/photographed solutions. Please submit all your solutions via LISAM or e–mail. If emailing, please email them to **BOTH** krzysztof.bartoszek@liu.se and KB_LiU_exam@protonmail.ch . During the exam you may ask the examiner questions by emailing them to KB_LiU_exam@protonmail.ch **ONLY**. Other exam procedures in LISAM.

NOTE: If you fail to do a part on which subsequent question(s) depend on describe (maybe using dummy data, partial code e.t.c.) how you would do them given you had done that part. You *might* be eligible for partial points.

Assignment 1 (10p)

Recall that the binomial coefficient is defined as

$$\binom{n}{m} = \frac{n!}{m!(n-m)!}.$$

It turned out that the following quantity is important to be evaluated

$$S(m,n,r) = \frac{(m+1)!n!(r-m-n)!}{(r-2)!}, \tag{1}$$

for m, n, r positive integers and $m, n, \leq r$.

Question 1.1 (2p)

Implement a function that calculates $S(m,n,r)$ as defined in Eq. (1). Do you need to have any other condition on m, n, r than the one stated above? You might find `R`'s `factorial()` function useful.

Question 1.2 (4p)

Is your function able to calculate the correct value of $S(m,n,r)$ for all m, n, r? What could be the problem? Experiment numerically in order to find for what m, n, r calculations break down, report on this and explain. You will not need to consider values of r exceeding 300.

Question 1.3 (4p)

Can you improve on Eq. (1) to make it numerically more stable? One suggestion is to use `R`'s `choose()` function, which should result in stabler calculations. Implement an improved version of calculating Eq. (1). It will still nearly certainly suffer from numerical issues. Experiment numerically in order to find for what m, n, r calculations break down now, report on this and explain. You will not need to consider values of r exceeding 700.

Assignment 2 (10p)

In the lecture we presented the linear congruential generator that is of the form

$$x_{k+1} = (ax_k + c) \mod m. \tag{2}$$

There are multiple other possible generators and one proposed option was the following additive generator

$$X_k = (X_{k-24} + X_{k-55}) \mod m, k \geq 55, \tag{3}$$

with some choice of $X_0, \ldots, X_{54}$ and *even* m.

Question 2.1 (3p)

Implement a random number generator based on Eq. (3). You need make a choice of the *integers* $X_0, \ldots, X_{54}$ and m. You can generate them using appropriately any of `R`'s inbuilt random number generators.

Question 2.2 (3p)

Compare how random numbers derived from your implemented generator compare with the uniform on $[0, 1]$ distribution. You may use `R`'s `runif()` function to generate uniform numbers for comparison. Report on what you find.

Question 2.3 (4p)

Mathematical theory behind Eq. (3) tells us that the choice of $X_0, \ldots, X_{54}$ *cannot* be arbitrary. In particular they should not all be even. What will happen if they are? Generate a random sample when $X_0, \ldots, X_{54}$ are all even and another one when they are not. Investigate, compare both to the uniform on $[0, 1]$ distribution, you may use `runif()` function to generate uniform numbers for comparison. Report on what you find.