S&SA Individual Assignment

Deadline: 22/11/2024 at 16:00h

Before you begin:

- ➔ This is an individual assignment that will count for 25% of your grade. Cooperating is not allowed. Copying code from others, other sources, or getting assistance, is not allowed. These actions constitute fraud. Please see the rules and regulations and this video on the difference between helping and fraud. We would like to re-iterate the following from the syllabus: "We expect you to do things in terms of the material as we discuss it, and there might be different versions around. As a concrete example: there are various runs tests and poker tests with varying degree of difficulty. We picked one and expect you to stick to it."
- ➔ Hand in a single .zip file called "assignment.zip" that solves this assignment on Canvas. Use subfunctions in separate files as the provided template does. Place your motivations and reflections in the comments. Please ensure all the .m files are in the highest-level directory, this means no subfolders/subdirectories in the submission file, if these are present, they will be ignored.
- ➔ Do not put any personal information in the .m file like your name or student ID! By submitting it you agree that it will be accessible by other parties like teaching assistants and plagiarism checking tools. The plagiarism check is also the reason why we do not allow LiveScripts (.mlx files).
- ➔ This assignment is to show your skills, hence don't let MATLAB run tests for you, but do them yourself. In the cases when you can use toolboxes, this is explicitly mentioned. **Hence, the default is that no toolboxes are allowed.** In the syllabus it is shown how you can see whether something is in a toolbox or not. Ensure that you implement distributions yourself.
- ➔ Be aware: 22/11/2024 at 16:00:01h is after the deadline in the way Canvas handles this, so don't wait till the last minute. Note that in Canvas there is a distinction between when the deadline is and till when an assignment is available.
- ➔ Ensure you test your indeed make the functions needed in separate .m files and test them thoroughly since we will use unit testing and if it fails our test that rubric will be a fail. The automated portions of the checks are case sensitive, so ensure that your function signatures defined **exactly** the same as listed below.

Task 1

Ensure that in your .zip file you have a MATLAB script *task1.m* that calls functions from other .m files to implement the task 1a and 1b.

In (L'Ecuyer, 1999), several approaches for combined LFSRs are given. In this task we will restrict ourselves to the first example of ME-CF generators as indicated in Table 1 of this paper. We make the following choices for the parameters: $L = 32$, $J = 4$, $s_1 = 18$, $s_2 = 2$, $s_3 = 7$, $s_4 = 13$. Furthermore, we take as seed $z_1 = 2957$, $z_2 = 646$, $z_3 = 3847$, $z_4 = 947$.

a) Implement this combined generator in MATLAB and generate 10,000 random numbers with this generator.

You should implement this generator in the MATLAB file *lEcuyer.m* with the following function definition:

```
function u = lEcuyer(z1, z2, z3, z4, n)
```

Where *n* is the number of random numbers to be generated, and *z1*, *z2*, *z3*, and *z4* the seeds. The output *u* is a **nx1 vector** of U(0,1) random numbers.

Toolboxes: For task 1 a) you are permitted to use the '*bitsll, bitsrl, bitand* and *bitxor'* functions from the Fixed-Point Designer Toolbox.

b) Implement a function `runsTest' in a MATLAB file `runsTest.m' that performs a runs test as in the lecture notes to assess the quality of the generated data from the random number generator.

```
function [reject, R] = runsTest(u, a)
```

Where u is the random numbers generated with your solution to 1a), a is the α-value, reject is a Boolean that indicates true if we reject H_0 and R is the test statistic.

Clearly state what your null-hypothesis is and what your conclusion is using $\alpha = 0.05$. Display the value of the test statistic, the critical value and the conclusion clearly. The input *u* is a **nx1 vector**, and both outputs are scalars. For this particular subquestion you are allowed to use the `norminv` function.

Task 2

Consider the following casus: the data covers the responses of a gas multi-sensor device deployed on the field in an Italian city. Hourly responses averages are recorded along with gas concentrations references from a certified analyzer. In this assignment, you are given the carbon monoxide (CO) and benzene (C_6H_6) measurements from the sensor.

All code and explanations for this task should be included in the file *task2.m*.

a) Load the array *dat* from *dataIndAss2425.mat* as it contains all measurement data. Use data exploration techniques to analyze, clean and visualize the relevant data. Use at least three different visualization techniques for the benzene (C_6H_6) measurements. You are allowed to use toolboxes for the visualizations, but not for anything else. Deal appropriately with outliers in the data in case there are any and argue why these are outliers and how and why you dealt with them. If there are none, then argue why.

b) Perform data exploration on the cleaned data and compute the following descriptive statistics: the seven-number summary, the mean, variance and (sample) skewness. Based on this data exploration, form a hypothesis from which distribution the processing times are coming and explain why you think this. This hypothesis should of course be reasonable. Limit the candidates to distributions from the textbook.

c) Perform a visual inspection of the comparative values of the carbon monoxide (CO) and benzene (C_6H_6) measurements in case both hourly measurements are available. If there are any outliers in these observations, describe at which times these observations were recorded.

Task 3

Consider a manufacturing plant in which products are produced by a single machine. We will model this plant using a $M|M|1$-queue in which machines can break down. This notation indicates that the inter-arrival times and service times are both Markovian (hence the M) and thus follow an exponential distribution. Moreover, the 1 indicates that there is a single server that processes customers.

Upon the breakdown of a machine, any product that is being processed will have to be finished once the machine is repaired, thus the remaining service time of this product has to be served first, before a new product is being put into service. To generate the inter-arrival and service times, you can use functions that are provided in the assignment, using the files *arrival.m* and *service.m*, respectively. Break-down and repair times can be generated using the files *breakdown.m* and *repair.m*.

Perform discrete event simulation for this manufacturing plant. Use the end of the busy period in which product #10,000 is constructed as the stopping criterion for your simulation. You should implement this generator in the MATLAB file *DES.m* with the following function definition:

```
function [avgWaitingTime, avgQueueLength] = DES()
```

Perform 20 iterations of the simulation and report on the distribution of the following performance indicators of your simulations:

- ◊ The average waiting time of customers in the queue
- ◊ The average queue length

References

L'Ecuyer, P. (1999). Tables of maximally equidistributed combined LFSR generators. *Math. Comput.*, *68*(225), 261–269. https://doi.org/10.1090/s0025-5718-99-01039-x

---