MCP261 IE Lab I: Exercise 3 **January 24, 2024**

Due date: 11:59 PM January 28, 2024

Introduction to Discrete Event Simulation

Submission Instructions

Each Python file *must* be easy to read: all variables and key quantities of interest must be clearly named, all computations must be commented, all assumptions must be very clearly stated, and finally, you may even add a short explanation of your approach to the problem at the beginning of the .py file. The exercise must be submitted as a zipped folder containing two Python files (.py scripts only – no .txt files or Jupyter notebooks will be entertained) – one for each question - and the "matresults.xlsx" Excel file (see problem description below). Naming convention for the zipped folder: Name_EntryNum_Exercise#.zip. Naming convention for the Python file for each question: Name_Exercise#_Question#.py. Your submission will be heavily penalized if it does not follow the above requirements. Note that the "matresults.xlsx" Excel file must also be provided in the zipped folder so that your code can be run directly from the folder to which your submission is extracted.

Exercise 3 Description

Examine TTF simulation code (in Matlab) provided on Teams. A brief description of the TTF simulation (and simulation in general) is also provided as a PDF document in the Ex 3 folder on Teams. A brief overview of the TTF simulation is also provided here for your quick reference.

Consider a system with two machines – one is an active machine and the other is an inactive spare. The spare machine becomes active when the (currently) active machine fails, while the failed machine immediately starts repair. The failed machine becomes the spare when its repair is completed. Only one component at a time can be repaired, so the system as a whole fails if both components have failed, and it is operational as long as at least one of the components is working.

For both questions below, use a random number seed of 1234. Use the NumPy, Pandas, SciPy and SciPy.Stats packages for the exercise.

1. (7 marks) Go through the Matlab code and develop a simulation of the same system in Python. Show that the results from your Python version of this simulation are statistically identical to that of the Matlab version by conducting a two-sample t-test (assuming equal variances) for the average number of functional components between the results from the Matlab and Python versions, using 60 replications. Print the output of the two-sample t-test.
   The results of the 60 replications (60 values of the average number of functional components) from Matlab should be written to an excel file called "matresults.xlsx". These results should be read in by your Python script for conducting the t-test. Provide the "matresults.xlsx" file along with your Python script.
2. (3 marks) Modify the system in Problem 1 such that time to repair is 3.5 days with probability 0.4 and 1.5 days otherwise. Find the average time to failure by programming 100 replications.