***Read all the following information before starting the exam:***

* The exam is open book, open notes, open Python documentation, open internet, etc.
* You are encouraged to use ChatGPT or other AI’s to help you with your coding. If you use an AI to write code, you must include a comment in your code such as: “ChatGPT helped write this function.”
* You **MAY NOT** use any form of technology to communicate with, send to, or receive information from another person (e.g., classmates, other instructors, anonymous persons on the internet). HOWEVER, you are **encouraged** to submit written questions to the professor or TA by email and/or have a private help session with the instructor through ZOOM.
* **MODULES/PACKAGES:** You may **NOT** use any Python packages other than random, math and copy and the functions used in our homework. However, you may use **any** Python packages you wish as a check of your answer, but they should not be a part of your submitted solution. You may use/reuse code (with proper attribution, e.g., “this function is copied from Dr. Smay’s Gauss\_Seidel.py file” or “this import is from my HW1 file”)
* **COMMENTS/DOCUMENTATION:** **ALL** of your functions must have docstrings and other comments inside the function as necessary.
* **SUBMISSION:** You must create a **private repository** on github and add [ashley.pennyman@okstate.edu](mailto:ashley.pennyman@okstate.edu) and [jim.smay@okstate.edu](mailto:jim.smay@okstate.edu) as collaborators. For final submission, copy the url of your repository and submit it on CANVAS

1. (50 pts) Create a Python program that simulates an industrial-scale gravel production process where crushed rocks are sieved through a pair of screens: the first screen is a large aperture screen that excludes rocks above a certain size and the second screen has a smaller aperture. The product is the fraction of rocks from between the screens.

Assumptions:

* 1. While the actual gavel is not spherical, we will assume that the rocks are spherical.
  2. Prior to sieving, the gravel follows a log-normal distribution (i.e., loge(D) is N(μ,σ)), where D is the rock diameter, μ=mean of ln(D) and σ= standard deviation of ln(D).
  3. After sieving, the log-normal distribution is now truncated to have a maximum (Dmax) and minimum size (Dmin) imposed by the aperture size of the screens.

Your program should solicit input from the user (with suggested default values) μ, σ, Dmax and Dmin. It should then produce 11 samples of N=100 rocks randomly selected from the truncated log-normal distribution and report to the user through the cli the sample mean (D̅) and variance (S2) of each sample as well as the mean and variance of the sampling mean.

Note:

The standard log-normal probability density function (PDF) is normalized over (0,∞) by:

And the normalized truncated log-normal PDF is given by:

Your grade will be based on your efficient use of imports of the allowed modules, use of functions and function calls, use of lists and list comprehensions, your clarity in your docstrings and comments and your overall approach to the problem. Clearly state your assumptions in the docstring of the main function.

1. (50 points) Using your program from problem 1 and your t-distribution probability calculator from homework 3, create a Python program that assumes two different gravel manufacturers submit their sample results to a buyer. Supplier A used a large aperture screen with 1”x1” openings whereas supplier B used a 7/8”x7/8” mesh. Supplier B claims to have a statistically significant smaller gravel size than supplier A. You must perform a statistical 1-sided t-test with α =0.05 to determine if this is true. Report your findings to the user through the command line interface.

Your grade will depend on your proper formulation of statistical hypothesis testing, clarity of your program (i.e., docstrings, region markers, step-by-step description, importing functions rather than copy/paste of code, and effectiveness of output).