EM384: Analytical Methods for Engineering Management
AT 23-2
Name:
Section :

Homework Set 10

This assignment is worth 20 points + **10 bonus points**, and is due NLT 1700 the day of Lesson 39 (Wednesday 10 May). Late submissions will be penalized one point per day until lesson 40. No submissions will be accepted past 1700, Thursday 11 May.

Your code must run without errors to be awarded points. It is highly recommended to start early on this homework set to allow time to troubleshoot any modeling or syntax errors you may encounter. You will be graded on the correctness of your model(s) as well as the clarity of your code and outputs. Inline comments using # are highly encouraged to add clarity to your code. Your outputs printed to the console should include explanatory text.

- **Documentation**. This deliverable is an individual assignment. Any assistance received must be documented in detail. Document all sources in accordance with the Office of the Dean Pamphlet "Documentation of Academic Work," (June 2015), Appendix E, and course guidance. e-Acknowledge documentation must be turned in through CIS at the time of submission. The deliverable is considered late until all portions of the assignment and the documentation are submitted.
- **Turn-In Requirement**: You will turn in **one file** to microsoft Teams: **a Python file** (extension .py), with the following naming convention:

  Section_LastName_FirstName_EM384_Homework_10.py

  Remember that engineering management is about communicating. You will be graded on the clarity and structure of your work.
- **Acknowledgement Statement**: This assignment must be accompanied by a signed e-Acknowledgement Statement (DAW) to be eligible for graded credit. If you submit your files(s) but fail to sign the e-Acknowledgement Statement, your assignment will be considered late until the e-Acknowledgement Statement is signed.
- **Guidelines for Documenting Assistance**: For this assignment, individual work is highly encouraged, but collaboration between individuals is allowed. ALL collaboration must be documented. Any discussion of this problem set with anyone other than an EM384 instructor requires documentation. Documentation must be specific and detail the topics discussed and actions taken.
- You must be very specific (which problem, what assistance, etc.) when explaining any assistance used in your documentation or you will be deducted at a higher penalty. Assistance *may* result in a deduction of points in accordance with a holistic assessment by your instructor.
- Sharing of electronic files via email or any other electronic means is strongly discouraged. Using, copying, or being dictated someone else's work will result in a greater point deduction.

Recall Homework Set 1 which modeled a deterministic portfolio for your Roth IRA (see homework 1 here). You will now be asked to create a stochastic version of your model for this homework in Python. You may use the same starter code from Homework 9.

You are a newly commissioned second lieutenant and have been advised to open a retirement savings account. In addition to the company match for your Thrift Savings Plan, you have decided (upon advice of your Systems Engineering instructor) to Open an Individual Retirement Account (IRA). There are two types of IRAs (You can read more about IRAs here): A *traditional* IRA and a *Roth* IRA.

The main differences between the two types of accounts are listed in Table 1 below.

	Traditional IRA	Roth IRA
Income Limit	No	Yes
Contributions	Pre-tax	After-tax
Distributions are Taxed	Yes	No

Table 1: IRA Differences

You decide to invest in a Roth IRA, and make the following additional assumptions:

- Assume an annual investment after tax of $6500 in the years you decide to contribute (do not adjust for inflation). This is the maximum allowed in dollars per year as of 2023.
- Assume an annual discount rate of 3% equal to the rate of inflation.
- Assume an annual portfolio return on investment of X, where X is a Triangular random variable each year with a min of -0.075, a max of 0.25, and a mode of 0.08 (See here for Python syntax to generate Triangular random variable(s) with the Numpy library).
- Assume that your portfolio compounds annually.
- Assume that starting at age 58, you make annual withdrawals equal to Y% of your portfolio, or $80000, whichever is greater, where Y is a continuous uniform random variable between between 1 and 3. If there is less than $80000 in the account in any year, then you withdraw the remaining amount that year.
- Assume that zero dollars is the lowest amount that your portfolio can go.
- For simplicity, assume that age change occurs at the beginning of the year.
- Assume that you make contributions and withdrawals to your account at the beginning of each year after your age change.
- Assume that investment returns or losses are compounded at the end of the year.

1. **(20 Points)** Build a Monte Carlo Simulation model (10,000 iterations) in Python that simulates the NPV of your Roth IRA portfolio balance at age 65, **when contributing $6500 a year from your *current age* through age 31**.

   (a) Print the expected NPV of your portfolio at age 65 (end of year) to the console.

   (b) Create a histogram of your portfolio NPV at age 65 (end of year) based on the output of your simulation. Pick a title and a number of bins that make sense.

   (c) Use your model to test what would happen if you change your starting age (try both directions). What do you observe? Why? Print your answer to the console. After this step, reset your starting age to your current age.

   (d) Create a plot of the ECDF of your portfolio NPV at age 65 (end of year). Pick a title and range that make sense.

   (e) Using your ECDF, what is the probability that you run out of money on or before age 65 (end of year)? Print the answer to the console.

2. **(BONUS: 10 Points) Any assistance received on this part from anyone other than from an EM384 instructor will result in reduced bonus points. You must clearly distinguish assistance received on this part from part 1 in your e-acknowledge statement.** You decide that you want to leave an inheritance for your loved ones so you are now interested in the expected NPV of your portfolio when you die. Build a Monte Carlo Simulation model (10,000 iterations) in Python that simulates this, **when contributing $6500 a year from your current age through age 31**. Make the following changes to your initial assumptions:

   - Your life expectancy is normally distributed with a mean of 87 years and a standard deviation of 10 years.
   - The age at which you start withdrawing money is a discrete uniform random variable between the values of 58 and 68 (included).
   - In any given year of your life there is a 0.02 probability that you need to make a disaster withdrawal equal to 50% of your portfolio balance.
   - Assume that if you live a partial year (e.g. 92.3), that make any required withdrawals but do not make any investment returns/losses that year.

   (a) Print the expected NPV of your portfolio at death to the console.

   (b) Create a histogram of your portfolio NPV at death based on the output of your simulation. Pick a title and a number of bins that make sense.

   (c) Create a plot of the ECDF of your portfolio NPV at death. Pick a title and range that make sense.

   (d) Using your ECDF, what is the probability that you run out of money before you die? Print the answer to the console.