Homework 4

This assignment was locked Feb 21 at 11:59pm.

Create a GUI that calculates your wealth for each of the 70 years after you start work given the following inputs into the GUI:

1) Mean Return (%)       This is the average annual return of the investment above inflation

2) Std Dev Return (%)     This is a measure of the annual volatility of the investment

3) Yearly Contribution ($)

4) No. of Years of Contribution

5) No. of Years to Retirement

6) Annual Spend in Retirement

There should be Quit and Calculate buttons at the bottom. When you press calculate, a function should be called which uses the specified values to run the analysis 10 times, each time for 70 years. The GUI should display the average of those ten times of your wealth at **retirement**. The retirement should be in a label, not in an entry field!

Your python code should also plot your wealth as a function of year for each of the 10 analyses. The plot must show how much money is in the account for all 70 years. (If the money runs out before 70 years for a particular analysis, end the plot for that analysis the year the money runs out and have that curve terminate at the x axis. That is, you can't go below $0! And don't continue the line from where it goes to $0 to the end of the 70 years. The line ends when it hits the x axis.)

Following are example formulas to use for your calculations. Note that this is not the way you have to implement your code, just a way to clarify how values are calculated. (In fact, using the below equations in your code would not be very elegant...) The wealth in year (i+1) can be calculated from the wealth at year i as follows:

(1) From the start until contributions end: Wi+1=Wi(1+(r/100)+noise[i]) + Y

(2) From end of contributions until retirement: Wi+1=Wi(1+(r/100)+noise[i])

(3) From retirement to the end: Wi+1=Wi(1+(r/100)+noise[i]) - S

where r is the rate entered into the GUI, Y is the amount contributed each year (from the start until contributions stop), S is the amount spent each year in retirement (only following retirement), and noise is an array of 70 random values generated with

noise = (sigma/100)\*np.random.randn(MAX\_YEARS)

where sigma is the standard deviation entered into the GUI, and MAX\_YEARS is 70 which, in your script, should be a constant!

NOTES:

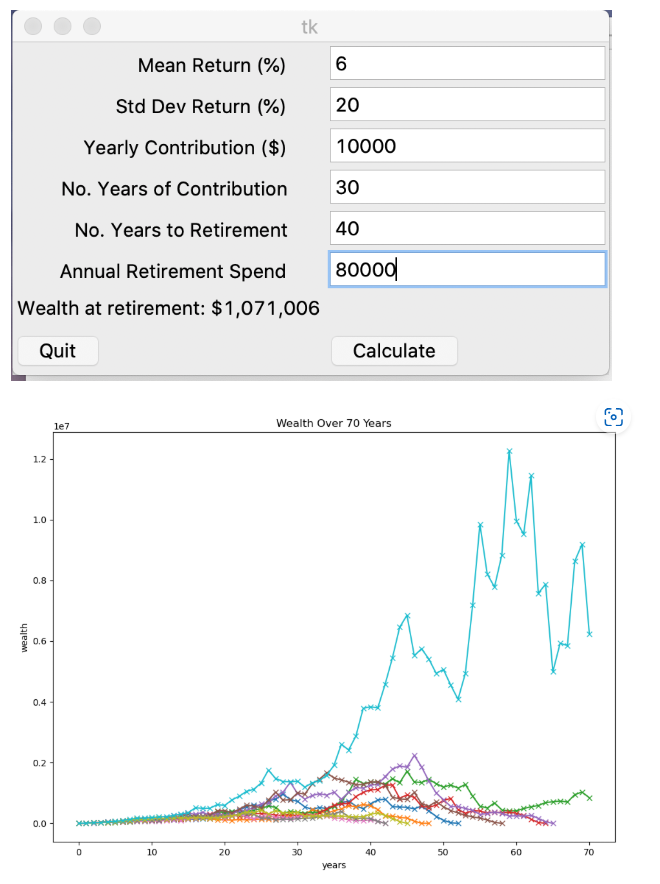
* Start with $0 in year 0
* Contributions must stop at, or prior to, retirement.
* Withdrawals start the year **after** retirement
* If years to retirement is the same as years of contributions, then the function of equation (2) above is skipped

**You may use either tkinter or PySimpleGui.**

Write efficient code, that is, code that executes efficiently. Add comments!

Turn in a file called hw4.py.

Here are examples of the GUI and the resulting graphs. Note that, due to randomness, you'll not be able to duplicate these exact numbers. Note these are not drawn to scale...



**Some installations of Python have a strange problem where the update to the average wealth at retirement doesn't show up until after the plot is closed. There are two possible ways to solve this problem. One, or both, may work for you:**

Try either importing matplotlib like this:

import matplotlib  
matplotlib.use("TkAgg")  
import matplotlib.pyplot as plt

Or try doing plt.show() like this:

plt.show(block=False)

[Previous](https://canvas.asu.edu/courses/177204/modules/items/12670998)[Next](https://canvas.asu.edu/courses/177204/modules/items/12671000)