Situation Room Pseudo Code

When ‘simulate’ button is pressed by user:

1. All user input is put into a user object, an accounts object and an array of futSteps where N goes from 1🡪the number of situations they created
2. **duration** is set as the latest end year – curSituation[age]; (example 85 – 23.69 = 56.31)
3. Initiates an array container for all situations;
4. Initiates a loop for **yr** that goes from **strt** to **last** (strt = 1950 & last = 2017)
5. Each **yr** is used as an argument in the function **simulateStarting**(**yr**)
6. The partial year will be determined (ie. 23.69 🡪 24) by taking the **yr** and any steps with start = ‘now’ and doing a calculation based off the amount of that year left (in this case 24 - 23.69 = 0.31). **age** will be appropriately set (in this case 24).
7. Then the **yr** will be increased by one and a loop will be started from **yr** to **yr** + **duration**.
8. Inside of which a call will be made to **simulate**(**yr**)
9. Which will go through and based on the **age** will pull relevant steps and simulate the year based on those steps and the **yr**.
10. **simulate**(**yr**) will then return an object with the information for the end of that year

For each year going from 1950 to 2017

Create a simulation based on the user’s game-plan if they had started in that year

If the user selected a ‘Rest of Life Option’ simulate them living to age 70, 75, 80, 85, 90, 95 and 100.

A loop will start with yr and age increasing by one each loop. Before that happens, each account will change based on the market and inflation for that year. If no data exists that far back then it will start with the first year of data and reset to the beginning once it has reached it’s last year. No data will be used that has less than a 15 year track record.

Put all these simulations into an array together

Once all simulations have been completed pull out the following data:

The best situation

The worst situation

The median situation

The average situation

Output that data into a graph that the user can easily interpret and give a few stats such as average age of paying off debt, average age of hitting a million $, % of cases that user ran out of money, etc.

Here is an example of what one simulation might look like for a simple 5-year debt repayment then 5-year savings plan…the simulations array will be an array of these with starting years from 1950 to 2017

[1950 = {IA Net Worth: -11,500}, 1951 = {IA Net Worth: -7,245}, 1952 = {IA Net Worth: -2450}, 1953 = {IA Net Worth: 0}, 1954 = {IA Net Worth: 0}, 1955 = {IA Net Worth: 0}, 1956 = {IA Net Worth: 4500}, 1957 = {IA Net Worth: 11,000}, 1958 = {IA Net Worth: 18,000}, 1959 = {IA Net Worth: 25,000}, 1960 = {IA Net Worth: 35,000}]

Tests

Converting users Date of birth into their current age to two decimal places:

dobToAge(1994, 04 28) 🡪 **23.82**

Using **simulate**(**yr**, **age**) function to test example plan (my plan):

**simulate**(**1983**, **23.82**) 🡪 { ‘debts’: 16788,

‘sAndP’: 20179,

‘tenYearBond’: 503 }

🡪 **simulate**(**1984**, **24**) 🡪 { ‘debts’: 10001,

‘sAndP’: 43944,

‘tenYearBond’: 572 }

🡪 **simulate**(**1985**, **25**) 🡪 { ‘debts’: 2926,

‘sAndP’: 80387,

‘tenYearBond’: 719 }

🡪 **simulate**(**1986**, **26**) 🡪 { ‘debts’: 0,

‘sAndP’: 117968,

‘tenYearBond’: 894 }

Using **simulateStarting**(**yr**) function to test example plan (my plan) starting in year 1983:

**simulateStarting**(**1983**) 🡪

[24: { Net Worth: 3894

IA Net Worth: 3868

}

25: { Net Worth: 34515

IA Net Worth: 32978

}

26: { Net Worth: 78180

IA Net Worth: 71962

}

27: { Net Worth: 118862

IA Net Worth: 108214

}

28: { Net Worth: 147511

IA Net Worth: 128595

}

29: { Net Worth: 194441

IA Net Worth: 162332

}

30: { Net Worth: 278431

IA Net Worth: 222123

}

31: { Net Worth: 284873

IA Net Worth: 214174

}…]

Using returnAvgHistoricalCase(allSimulationsArray) to generate an ‘average’ case.

If allSimulationsArray = [ [0, 3, 10]

[50, 3, 10]

[100, 4, 10]

]

Then returnAvgHistoricalCase(allSimulationsArray 🡪 [50, 3.33, 10]