The Custom Cash Company (C3) is an insurance company that sells annuities to individuals and organizations. An [annuity (Links to an external site.)](https://en.wikipedia.org/wiki/Annuity_(American)) is typically an exchange of one large up-front payment by the customer for monthly payments from the insurance company for a fixed term or for life. C3 has these products, but wants to break into a specialty niche market that allows the products to be tailored to high-value customers who have a known series of future cash flows that they would like to amortize out to a fixed monthly amount for a fixed period. C3 also wants to include annual inflation adjustments to the monthly payments since its marketing team thinks that will help them sell these niche products. To start, C3 will only offer a fixed-term payout which is guaranteed to a beneficiary if the original customer dies during the term of the annuity.

For example, say a customer, Sarah, has sold a real estate property and taken back a mortgage for the purchaser which is comprised of annual payments every year for 7 years starting in 5 years. She also forsees paying for her daughter's education at Seattle University beginning in 2 years from now and for the following 4 years after that. All of these cash flows are fairly certain (incoming mortgage payments from the property purchaser and outgoing quarterly payments for SU tuition in the future). So C3 would agree to take over all of Sarah's cash flows in exchange for providing a monthly fixed annuity for 20 years to Sarah. Essentially, C3 is paying out what's left over from the mortgage receipts after paying Sarah's daughters tuition. Of course, they have to take into account the time-value of the cash flows they are taking on from Sarah and also the time-value of the money being paid out to Sarah over time in the monthly annuity payments. Sarah also wants the fixed payments she's getting from C3 to go up by 3% after 12 months, and then another 3% after 24 months, and every year until the end.

C3 wants to have the tools to figure out the fair amount of the initial monthly payment for such a contract given a list of arbitrary cash flows from the customer and an annual inflation adjustment percentage. Of course, C3 will also take out a small fee, too, from the payments, but that can be calculated separately. Your problem is to tell them the fair amount of the monthly payments.

The time-value of the monies is determined by the interest rate curve for C3 which can easily be obtained for any term and will be provided in another data file.

Analysis

There are number of ways to approach this analytically, but one very simple way comes to mind since we know how to program a computer to compute the net present value (NPV), i.e., the time-value of a set of future cash flows, of all the customer-provided cash flows along with the monthly C3-paid payments together. This NPV will be exactly zero if the payments are fair. We just have to find the payment amount that makes the NPV exactly zero. For this we can use bisection search. It's easy to bracket the root, since we know that a payment of zero is way too low and a payment of the entire sum of the customer-provided cash flows is way too high--and, of course, we know that the just right payment amount is somewhere in between.

NPV

Given an interest rate, we can calculate the present value of a future cash flow. Logistically, if we wanted to make a cash payment t periods in the future with some funds today, what we'd do is deposit some amount of money in an interest-bearing account and withdraw it at the end of t periods with the accumulated interest and have the exact amount required to make the cash payment at that time. Here's the formula:

p v = f v ⋅ e − r t

where *fv* is the future cashflow, *r* is the interest rate per period, and*t* is the number of periods. For our purposes, *pv* and *fv* are in US dollars and *t* is in months. Since *t*'s units are months, we need *r* to be in interest rate per month. For example, if *t* is 18 months and the interest rate for a 18-month term deposit is 6% per annum, then we would use r = 0.06 12 = 0.005. In C3 world, their effective interest rates are a term structure, where short term rates are typically less than longer-term rates. Each monthly payment thus has a *pv* calculated with the term rate for the period between now and when the payment will be made. These rates are market variables and will be provided to you in a file.

The net present value is just the sum of the present value of each of the cash flows:

n p v = ∑ i = 0 n p v i

Communication

The envisioned system would be a web-based tool where the sales team would enter the customer's cash flows, the term of the annuity, and the inflation factor and instantly get the monthly annuity amount (with C3's fee already deducted). That would be built by a software development team and they would then use your software to calculate the fair market annuity. For now, though, you just need to write the calculation software and that will be executed manually whenever required. The cashflows can be provided in a comma-separated value (CSV) file. So too, can the interest rates for all the monthly terms be provided in a CSV file and also the annuity parameters: annuity term (number of months) and annual inflation adjustment percentage. Your software just needs to print out the answer and a little explanatory report.

Management would also like a few paragraphs with your thoughts about future direction of this project. What are the flaws and drawbacks of this approach and what enhancements could you imagine going forward? Perhaps you have some ideas about things you could easily do that would add features for the marketers and sales. Just a paragraph or two with your thoughts is sufficient.

We look at the first test case and imagine a bare-bones verbose report like this and run it by the stake holders to make sure everyone has the same idea. We explain we're going to use bisection search, so we will print that out for now so everyone can see it working correctly. Here's our imagined output from the first test case:

months: 1 cashflow: 50088.12 discount: 0.9986675551606254 npv: 50021.38034299203

months: 2 cashflow: -98000.0 discount: 0.9970875826732489 npv: -97714.5831019784

months: 9 cashflow: 140000.0 discount: 0.9851858257692295 npv: 137926.01560769213

NPV of customer-provided cash flows: $90232.81

Total customer-provided cash flows: $92088.12

Searching for correct beginning monthly annuity payment:

1: With annuity of (0.00+92088.12)/2 = $46044.06, NPV is $-1087730.11

2: With annuity of (0.00+46044.06)/2 = $23022.03, NPV is $-498748.64

3: With annuity of (0.00+23022.03)/2 = $11511.01, NPV is $-204257.79

4: With annuity of (0.00+11511.01)/2 = $5755.51, NPV is $-57012.68

5: With annuity of (0.00+5755.51)/2 = $2877.76, NPV is $16609.93

6: With annuity of (2877.76+5755.51)/2 = $4316.64, NPV is $-20201.50

7: With annuity of (2877.76+4316.64)/2 = $3597.20, NPV is $-1795.71

8: With annuity of (2877.76+3597.20)/2 = $3237.48, NPV is $7407.18

9: With annuity of (3237.48+3597.20)/2 = $3417.34, NPV is $2805.72

10: With annuity of (3417.34+3597.20)/2 = $3507.27, NPV is $505.02

11: With annuity of (3507.27+3597.20)/2 = $3552.23, NPV is $-645.23

12: With annuity of (3507.27+3552.23)/2 = $3529.75, NPV is $-70.17

13: With annuity of (3507.27+3529.75)/2 = $3518.51, NPV is $217.34

14: With annuity of (3518.51+3529.75)/2 = $3524.13, NPV is $73.66

15: With annuity of (3524.13+3529.75)/2 = $3526.94, NPV is $1.74

16: With annuity of (3526.94+3529.75)/2 = $3528.35, NPV is $-34.28

17: With annuity of (3526.94+3528.35)/2 = $3527.64, NPV is $-16.21

18: With annuity of (3526.94+3527.64)/2 = $3527.29, NPV is $-7.18

19: With annuity of (3526.94+3527.29)/2 = $3527.11, NPV is $-2.58

20: With annuity of (3526.94+3527.11)/2 = $3527.03, NPV is $-0.54

21: With annuity of (3526.94+3527.03)/2 = $3526.99, NPV is $0.47

22: With annuity of (3526.99+3527.03)/2 = $3527.01, NPV is $-0.04

23: With annuity of (3526.99+3527.01)/2 = $3527.00, NPV is $0.22

24: With annuity of (3527.00+3527.01)/2 = $3527.01, NPV is $-0.04

25: With annuity of (3527.00+3527.01)/2 = $3527.01, NPV is $-0.04

First year pay $3527.01 per month

Next year pay $3562.28 per month

Next year pay $3597.90 per month

For a total of $92267.29

Resources

C3 will provide some test cases for you to get started. Each test case consists of three data files:

* Series of discrete cash flows for each month where there is a customer-supplied cash flow. A typical file name would be p4\_test1\_cash\_flows.csv. This is a 2-column csv-file with the first column being the month number (number of months from now) and the amount of the cash flow in US dollars. A positive cash flow is one that will be given to C3 and a negative one is a payment by C3. There are only entries for months with cash flows and if there are multiple cash flows for the same month they have been netted together in a single entry. This file has also been sorted by month number.
* Term interest rates for maturities at each month. This is a 2-column csv-file with the first column being the month number (number of months from now) and the percentage annualized interest rate between now and that month. There are only entries for months where the rate changes from the previous month. This file is sorted by month number. A typical file name would be p4\_test1\_interest\_rates.csv.
* Terms of the annuity are in a third file. A typical one would be named p4\_test1\_terms.csv. It has entries for the number of months of payments and the inflation adjustment percentage to adjust up the payments every year.

Requirements

You must:

* Use bisection search as described above. No other analytical methods are allowed (though you could mention them in your textual paragraphs).
* You may only use Python features we've learned so far, so, specifically: no tuples, lists, dictionaries, or functions.
* Your output should be nearly identical to that provided in the guidance for test case 1. The expected output for test case 2 is not provided and the secret test case 3 is not provided at all but will be used to evaluate your programs.
* You must hand in three files:
  1. p4\_create\_dense\_files.py, a program that takes the given test case files and produces two files that includes those data and also has entries for all the implied data for the "missing" months. You must follow the outline as shown in the guidance.
  2. p4\_search\_annuity.py, a program that uses bisection search to find the correct beginning year's monthly annuity payment in US dollars. You must follow the outline as shown in the guidance.
  3. A document with your thoughts as required above in the communications section. Just a paragraph or two is sufficient.

Guidance

* Download the instructor provided files: [p4\_instructor\_files.zip](https://seattleu.instructure.com/courses/1592277/files/65846233/download?wrap=1).
* Open p4\_hints.py to see hints on the structure of the programs.
* The provided test case files are also present in that zip folder.

We'll do this project in two passes.

1. The first pass takes the rather sparse input files and expands them so we have two files: one for cash flows for every month (most of the lines being 0.0) and one for the interest rates for every month (if a month was excluded in the sparse input file, we copy the value from the previous month). This is the program p4\_create\_dense\_files.py. Get this working first before proceeding to pass 2. The expected results for test case 1 are in p4\_test1\_dense\_cf.txt and p4\_test1\_dense\_rates.txt.
2. The second pass takes the fully-expanded dense data files and reads them to calculate NPV's. (It does this once for the customer-provided cash flows since that doesn't ever change, but it does it repeatedly for the various guesses it has for the annuity payment amounts.) This is the program p4\_search\_annuity.py. It's expected output for test case 1 is in p4\_test1\_results.txt.