

Retirement Planning

You will likely spend 1/4 to 1/3 of your life in retirement - collecting money while *not* working. Where does your income come from in retirement? This activity investigates a special type of retirement planning scenario: pay yourself first!

Objectives:

- Determine the amount of money you will accrue in a retirement account by controlling various parameters.
- Examine the impact of management fees on your retirement savings.

This might sound like a strange topic for high school students, but it is *never* too soon to plan for retirement!

Retirement income can come from many different sources: a pension plan, a 401K or similar plan, IRAs (Individual Retirement Accounts), USA's Social Security or the Canada Pension Plan (other countries have similar government-managed plans). But company-sponsored pension plans can be risky (see Enron), and government plans are usually not sufficient to live on comfortably.

Your personal planning and saving can assure you a secure and rewarding retirement. It's never too soon (or too late) to start saving for retirement. But sooner is better because you will tap 'the power of compounding' as you will see in this activity.

Teacher Note: The built-in **Finance...** App lets users solve TVM problems using a simple formula and can solve for any of the feature's parameters. But it is limited in scope, especially in the **pmt** field: it assumes all payments are equal. This activity takes a different approach: as your income increases over time, so should your retirement plan contribution. And, in retirement, your income should *increase* to keep pace with inflation.

The data that the program requires are not just 'pulled out of thin air.' Some preliminary instruction about salaries, taxes, investment options and fees, and economic factors like inflation should be covered.



Beyond the Time Value of Money

TI-84 PLUS CE PYTHON TECHNOLOGY

1. Run the Python program named **RETPLAN** on your TI-84 Plus CE Python. (Be sure you have installed the RETPLAN.8xv program file to your calculator using [TI Connect CE](#).)

The result of the program is the screen shown here. You will change some of the values that the program uses to calculate these results by editing the code.

Press **<Editor>** to edit the program.

2. The data that the program uses is stored in a Python dictionary named **data** which conveniently labels each of the input values. This variable is defined near the top of the code. *Edit the program and change only the numbers in the variable **data**, not the text, punctuation, or indentation!*

The values represent:

Start age: the age at which you open a retirement account.

Retire age: the age when you plan to retire.

Final age: the age when you expect the plan to run out of money. Your death?

Current Savings: the opening deposit in the account.

Current Salary: your total annual salary for the first year of saving.

Percents are entered in 'percent form,' not their decimal equivalents. For 12% enter 12, not 0.12. For 2½% enter 2.5, not 0.025.

% to save each year: percent of annual salary to save. 12% of annual salary is the default.

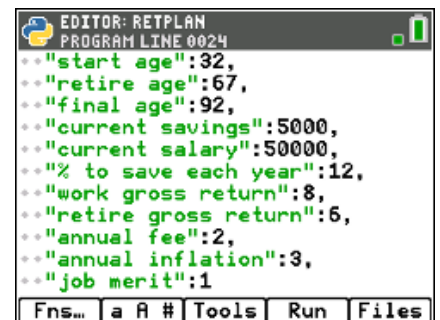
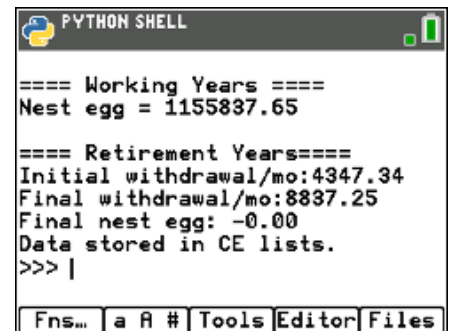
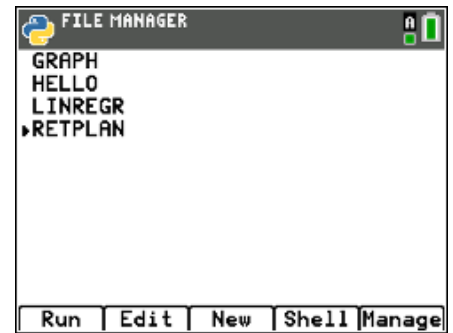
Work gross return %: estimated rate of return on investment during working years. Default is 8%.

Retire gross return %: estimated rate of return on investment during retirement years. Default is 6%.

Annual fee %: percent of the investment company's annual fees. Default is 2%.

Retirement Planning

TEACHER NOTES





Annual inflation %: estimate of annual inflation rate. Default is 3%.

Job merit %: average increase in salary/year. Default is 1%.

Note: (inflation+merit) are combined to account for annual increases in salary and your retirement contribution.

Teacher Notes:

A copy of the **data** dictionary can be found at the end of the program. It has no effect on the program but is a good reminder of the original values that were used.

Scenario 1: You graduate from college at age 22. Your first job pays \$40,000 a year. You immediately (smart!) open a retirement account starting with \$1,000. You will deposit only \$100 the first year and plan to increase that deposit thanks to your (predicted) increases in salary to cover inflation at 3% plus merit raises of 2%. Your retirement plan charges 3% fees annually. Assume the gross returns on the investment are 8% while working and 6% when retired. You plan to retire at age 70 and expect your retirement plan to last to age 100 (with annual increases to cover inflation).

Your answers:

- a) What percent of your salary are you depositing each year? Use this value as the **% to save each year**.

a) _____

Enter the **data** into the dictionary and run the Python program.

- b) How large is your **nest egg** at retirement?

b) _____

- c) What is your first monthly retirement payment?

c) _____

Are you happy with this result?

Teacher Notes:

Answers:

- a) $100/40000 = 0.25\%$. That's only $\frac{1}{4}$ of one percent. Enter 0.25 for the % to save each year, not 25!

- b) \$55,376.45

- c) \$152.22

Not a very promising plan!



Scenario 2: Suppose you find a retirement plan that charges a very small fee for administration: 0.1% (one tenth of one percent). Change only this value in your input sheet from Scenario 1 and re-run the program. Report your new

- a) Nest egg value
- b) First month retirement payment
- c) By what percent does your nest egg increase?

Your answers:

a) _____

b) _____

c) _____

Teacher Notes:

Answers:

- a) \$138,325.36
- b) \$575.55
- c) $(138325.36 - 55376.45) / 55376.45 = 1.4979...$ or approximately 150%!

While still not a magnificent total the nest egg is now 2.5 times larger just from smaller fees!

Scenario 3: What will it take to retire with one million dollars?

At age 25 you open a retirement account with \$2,000. Your salary is currently \$50,000 a year, and you confidently expect 2% merit increases each year on top of inflation increases. What *minimum yearly* deposit should you start with to acquire a nest egg of \$1,000,000 at age 70?

\$1,000,000.00

Assume the following additional parameters:

Final age: 100

Work Gross Return: 8%

Retire Gross Return: 6%

Annual Fees: 2%

Annual Inflation: 2%

- a) Work with the “% to save each year” value to get the nest egg just above \$1,000,000. Calculate that % of \$50,000 to arrive at the first year’s deposit.
- b) Suppose you can get an annual fee of only 1%. How does this impact the nest egg?

Answer a) _____

Answer b) _____

Teacher Notes:

Inputs:

"Age now" 25

"Retirement age" 70

"Final age" 100

"Current savings" 2000

"Current salary" 50000

"% to save each year" --- *Work on this number to get just over a \$1,000,000 nest egg.*

"Work gross return %" 8

"Retire gross return %" 6

"Annual fee %" 2

"Annual inflation %" 2

"Job merit %" 2

- a) A value of 5% (to invest each year) results in a nest egg of \$1,004,634.92. That's a starting yearly deposit of $.05 * \$50,000 = \$2,500$ or about \$208 per month.
- b) The nest egg increases to \$1,317,962.80. That's over a 30% increase just from reducing the fees!

Stat Plots

After each run of the program several lists are stored into the calculator. Quit the Python App and look at the **[list]** menu (press **[2nd]** **[stat]**). The seven *named* lists (below **L₁** ... **L₆**) are:

ACCUM: the amounts in the account during the working years

LIFE: ages from "start age" to "final age": $LIFE = WORK + RETIR$

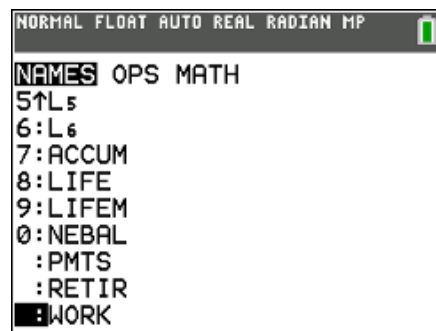
LIFEM: lifetime money - amounts in the account

NEBAL: the nest egg balance during retirement

PMTS: the monthly payments during retirement

RETIR: the retirement ages

WORK: the working ages



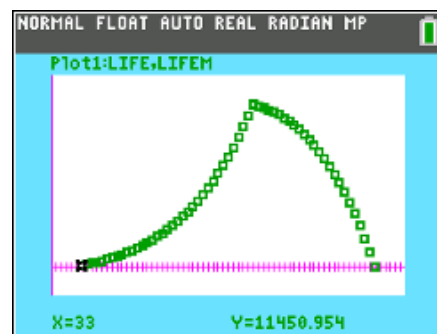
There are several **stat plots** possible:

1. Growth of the nest egg while working:
x-list: **WORK**
y-list: **ACCUM**
2. Decline in the nest egg balance during retirement:
x-list: **RETIR**
y-list: **NEBAL**

3. Growth of monthly retirement payments:
x-list: **RETIR**
y-list: **PMTS**
4. The whole lifetime values in the account (*image to the right*):
x-list: **LIFE**
y-list: **LIFEM**

What types of graphs are displayed?

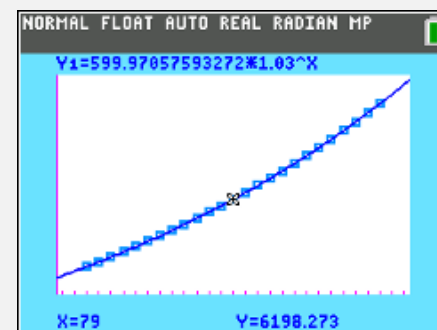
Use the TI-84 Plus CE **[stat] > CALC** tools to determine an appropriate mathematical model for each graph.



Teacher Notes:

The first two plots are some forms of exponential functions. The third, 'monthly retirement payments,' produces a good exponential function but the others will require some data transformation or combinations of exponential functions to get a good fit.

The last one (**LIFE, LIFEM**) makes a good piecewise function activity!



Summary:

Some things are out of your control when it comes to retirement planning, but the factors that you *can* control greatly influence the final outcome: the age you start saving, the percent of your income you can put away each year, and the fees that the company charges you to invest your money all have an impact on nest egg values.

Regardless of the actual numbers you use, your retirement plan *with increasing monthly payments* will look like the screen to the right. The final question is: how high can it go? It's up to you: forewarned is forearmed.

