

EE4017 Internet Finance

Lab 1 - Computing Time Value of Money with Python

1. Overview

In this laboratory, students will be walked through various financial concepts and learn to solve different computational problems with Python and its modules.

2. Environment and Tools

This laboratory will make use of Python 3.6.8 with modules SciPy, NumPy, SymPy and Matplotlib.

3. Installing the Dependencies

Open the Terminal, make a new directory as our project directory and change to that directory:

```
$ mkdir <project_directory>
$ cd <project_directory>
```

Create a new Python 3 virtual environment named `.env` by invoking the `venv` module and activate the environment:

```
$ python3 -m venv .env
$ . .env/bin/activate
```

After activating the virtual environment, install the required Python modules using `pip`:

```
$ pip install scipy numpy sympy matplotlib
```

Once the modules are installed properly, you can try to import them using the interpreter and display the module manual using the `help()` function. You may also want to look into further documentations by referring to the specified link.

```
>>> import scipy as sp
```

```
>>> help(sp)
Help on package scipy:

NAME
    scipy

DESCRIPTION
    SciPy: A scientific computing package for Python
    =====

    Documentation is available in the docstrings and
    online at https://docs.scipy.org.
[snipped]
```

4. Time Value of Money

Consider the following problem, assume that \$100 is deposited in a bank today with an annual interest rate of 10%. What is the value of the deposit two year later?

$$FV = PV(1+R)^n \text{ ---- (1)}$$

Here, FV is the future value, PV is the present value, R is the period rate and n is the number of periods.

Therefore,

$$\begin{aligned} FV (\text{at the end of year 2}) &= 100 * (1 + 0.1)^2 \\ &= 121 \end{aligned}$$

We can use the corresponding function in the *SciPy* module, called $fv()$, to estimate the future value. Open the Terminal, change to the project directory and activate the Python virtual environment:

```
$ cd <project_directory>
$ . env/bin/activate
```

Run the Python interpreter and import the SciPy module:

```
>>> import scipy as sp
>>> sp.fv(0.1,2,0,100)
-121.00000000000001
```

You can use the *help()* function to find out more information on *scipy.fv()* or refer to the documentation:

```
>>> help(sp.fv)
```

Q1 Explain why the output of *sp.fv()* above is negative and why the result is not an integer?

Q2 John deposits \$5,000 in the bank for 25 years. If the annual rate is 0.25% per year, what is the future value? Write down your input of *sp.fv()* and the output result.

To find out the present value, we can use the function *scipy.pv()* which is corresponding to the following formula, derived from equation (1).

$$PV = FV/(1+R)^n \text{ ---- (2)}$$

Q3 Consult the help manual of *scipy.pv()* and compute the following (write down your input of *scipy.pv()* as well):

If we plan to have \$234 at the end of year five and the interest rate is 1.45% per year, how much we have to deposit today?

5. Perpetuity and Annuity

The concept of perpetuity is defined as the same constant cash flows at the same intervals forever. Applying equation (2) and summing up all the future cash flows *C*,

$$PV(\text{perpetuity}) = [C/(1+R)] + [C/(1+R)^2] + [C/(1+R)^3] + \dots \text{ ---- (I)}$$

Multiply both sides by $1 / (1 + R)$,

$$PV(\text{perpetuity}) / (1+R) = [C/(1+R)^2] + [C/(1+R)^3] + \dots \text{ ---- (II)}$$

Subtract equation (I) with (II),

$$PV(\text{perpetuity}) - PV(\text{perpetuity}) / (1+R) = C / (1+R)$$

By reducing the above equation, we can formulate an estimation of the present value of perpetuity:

$$PV(\text{perpetuity}) = C/R \text{ ----- (3)}$$

Q4 John plans to donate \$3,000 per year to his alma mater to have a welcoming party for the forthcoming MBA students at the end of the year forever. If the annual discount rate is 2.5% and the first party will occur at the end of the first year, how much he should donate today?

Assume that the first cash flow is C and the following cash flows enjoy a constant growth rate of g , then the present value of the growing perpetuity is:

$$PV(\text{growing perpetuity}) = C / (R - g) \text{ ----- (4)}$$

If the first cash flow happens at the end of k^{th} period:

$$PV(\text{perpetuity}, k^{\text{th}}) = [1 / (1+R)^{k-1}] [C / (R - g)] \text{ ----- (5)}$$

An annuity is defined as the same cash flows at the same intervals for n periods:

$$PV(\text{annuity}) = (C/R) [1 - (1+R)^{-n}] \text{ ----- (6)}$$

Q5 Assume that the discount rate is 1% per year. The annual cash flow is \$20 for the next 10 years. The first cash flow will be at the end of the first year. What is the present value of annuity? Write down the input of $sp.pv()$ and calculate the result.

Annuity and perpetuity are called annuity due and perpetuity due when the cash flows happen at the beginning of each time period. The formula is as follows:

$$PV(\text{annuity due}) = (C/R) [1 - (1+R)^{-n}] (1+R) \text{ ----- (7)}$$

Q6 Assume that the discount rate is 3.5% per year. The annual cash flow is \$20 for the next 15 years. What is the present value of annuity due? Write down the input of `sp.pv()` and calculate the result.

We have now walked through the basic formulae concerning present and future value. Instead of using the functions provided by SciPy or NumPy, we can write our own financial calculator without using those modules, since we already have those formulae.

Create a new file called *financial.py* in the project directory (You may use any text editors by your own choice):

```
$ cd <project_directory>
$ vim financial.py
```

The function *fv()* below serves as an example of a self-implemented function of future value according to equation (1). Copy the code into *financial.py* and save the file:

```
def fv(pv, r, n):
    """
        Objective : estimate future value
        pv : present value
        r : discount period rate
        n : number of periods
        formula : pv*(1+r)**n
        e.g.,
        >>>fv_f(100,0.1,1)
        110.00000000000001
    """
    return pv * (1 + r) ** n
```

We can invoke the our *fv()* by importing *financial.py* in the interpreter:

```
>>> import financial as fin
>>> help(fin.fv)
>>> fin.fv(100,0.1,1)
```

Task 1 Follow the example of $fv()$ given above, create six new functions named $pv()$, $pv_perpetuity()$, $pv_grow_perpetuity()$, $pv_perpetuity_kth_period()$, $pv_annuity()$ and $pv_annuity_due()$ in *financial.py*, write and complete your own implementation of the functions based on equation (2)-(7) without using any Python modules (i.e. Usage of SciPy, NumPy is not permitted).

Task 2 Formulate the equation of future value of annuity and future value of annuity due, explain how you can derive those formulae, create and implement your own functions $fv_annuity$ and $fv_annuity_due()$ in *financial.py* without using any Python modules.

*Remember to save and keep your work of *financial.py*, it will be used in the subsequent sections.