Stochastic Methods Lab, Jacobs University Bremen, Fall 2022 — Session 2, Sep. 8, 2022

Prof. Dr. Soeren Petrat

0.2 Scientific Python

- optimized for vectorized operations, using NumPy arrays (runs near machine speed, even though it's an interpreted language)
- SciPy package has functionality most aspects of scientific computing

Practically: Install Anaconda package
- ↳ SciPy libraries included
- ↳ spyder editor (development environment) ← rec. for HW submission
- ↳ jupyter notebooks (editor + comment (mark up) + run code fragments separately)

For homework: always submit .py files
(If you use jupyter notebooks, export them as .py files.)

1. Basics of Financial Math

1.1 Time Value of Money

r = annual interest rate , FV = future value , PV = present value

After n years interest compounds: $\boxed{FV = PV(1+r)^n}$

(the "value" of money changes over time) $\Longleftrightarrow PV = FV(1+r)^{-n}$

If interest is compounded m times per year: $FV = PV\left(1+\frac{r}{m}\right)^{n \cdot m}$

Terminology:
- BEY (bond equivalent yield)
  ↳ annualized yield, compounded semiannually (twice a year, $m=2$)
- MEY (mortgage equivalent yield)
  ↳ annualized yield, compounded monthly ($m=12$)

Q.: How to compare financial instruments with different compounding standards?

A.: Compare to an effective annual interest rate r_{eff}:

$$(1+r_{eff})^n = \left(1+\frac{r}{m}\right)^{nm} \Longrightarrow 1 + r_{eff} = \left(1+\frac{r}{m}\right)^m$$

$$\Longrightarrow \boxed{r_{eff} = \left(1+\frac{r}{m}\right)^m - 1}$$

Example: $r = 10\%$, BEY

$$\Longrightarrow r_{eff} = \left(1+\frac{0.1}{2}\right)^2 - 1 = 0.1025 = 10.25\%$$

Is r_{eff} always bigger than r? $(m \geq 1)$

$$r_{eff} = \left(1+\frac{r}{m}\right)^m - 1 = 1 + m\frac{r}{m} + \underbrace{rest}_{\geq 0 \text{ for } r \geq 0} - 1 \geq r$$ Yes (for $r \geq 0$)

$\underbrace{(1+x)^m \geq 1+mx}_{\text{Bernoulli's inequality}}$ holds for $x > -1$, so $\boxed{r_{eff} \geq r \text{ even for } \frac{r}{m} > -1}$

One often useful idealization is continuous compounding, i.e., take limit $m \to \infty$.

$$\Rightarrow FV = PV \lim_{m \to \infty} \left(1+\frac{r}{m}\right)^{n \cdot m}$$

$$= PV \Bigg[\underbrace{\lim_{m \to \infty} \left(1+\frac{r}{m}\right)^m}_{= e^r = \exp(r) \text{ exponential function}}\Bigg]^n$$

$$= PV \left(e^r\right)^n$$

$$\Rightarrow \boxed{FV = PV\, e^{rn}}$$

1.2 General Cash Flows

n years, r yearly interest rate ← In our idealization this is fixed (for whole period under consideration)

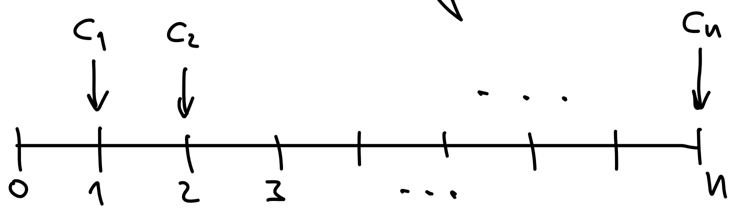
Suppose at the end of the year j, there is a cash flow C_j.

Then
$$\boxed{PV = \frac{C_1}{1+r} + \frac{C_2}{(1+r)^2} + \ldots + \frac{C_n}{(1+r)^n} = \sum_{j=1}^{n} \frac{C_j}{(1+r)^j}}.$$

Note: Future value after n years

$$\boxed{FV_n = C_1 (1+r)^{n-1} + C_2 (1+r)^{n-2} + \ldots + C_n = \sum_{j=1}^{n} C_j (1+r)^{n-j}}$$

Ex.: Financial instrument paying C_j at end of each year to you.



$$PV = \text{price of such an instrument} = \frac{C_1}{1+r} + \frac{C_2}{(1+r)^2} + \ldots + \frac{C_n}{(1+r)^n}$$

$$FV_n = \text{value of financial instrument at end of } n \text{ years} = PV\,(1+r)^n.$$

In python, how do we evaluate polynomials, such as $\sum_{j=1}^{n} c_j x^j$, with $x = \frac{1}{1+r}$?

- explicit "for" loop (discouraged, very slow)

- best: vectorized operations

  define vector $j = \text{arange}(1, n+1)$ $(= (1, \dots, n))$

  use given vector $C = (c_1, \dots, c_n)$

  we compute new vector $\underbrace{x ** j}$ $= (x^1, x^2, \dots, x^n)$

  number to the power of a vector

  (does not make mathematical sense, but in python it is interpreted component-wise)

  use dot product (scalar product) to evaluate sum: $PV = \text{dot}(C, x**j)$

- Horner's scheme:

$$PV = (\dots((c_n x + c_{n-1})x + c_{n-2})x \dots + c_1)x$$

  (fewer operations than explicit loop)

- polyval (fct. from SciPy), uses optimized version of Horner's scheme

Special case of a cash flow: <u>Annuity</u>: yearly payments, all $C_j = C$.

Types:

- **ordinary annuity** (usually assumed): pays C at end of the year

$$FV = \sum_{j=1}^{n} C(1+r)^{n-j} \overset{i:=n-j}{=} C \underbrace{\sum_{i=0}^{n-1} (1+r)^i}_{\text{geometric series}}$$

recall: $x \sum_{i=0}^{N} x^i = \sum_{i=0}^{N} x^{i+1} = \sum_{i=1}^{N+1} x^i = \sum_{i=0}^{N} x^i - 1 + x^{N+1}$

$$\Rightarrow (x-1) \sum_{i=0}^{N} x^i = x^{N+1} - 1$$

$$\Rightarrow \sum_{i=0}^{N} x^i = \frac{x^{N+1}-1}{x-1}$$

$$\Rightarrow \boxed{FV = C \frac{(1+r)^n - 1}{1+r-1} = C \frac{(1+r)^n - 1}{r}}$$

- annuity due: pays at beginning of year

$$FV = C \sum_{i=1}^{n} (1+r)^i = C(1+r) \sum_{i=0}^{n-1} (1+r)^i = C(1+r) \frac{(1+r)^n - 1}{r}$$

- general (ordinary) annuity: m payments per year (at end of period)

$$FV = C \sum_{i=0}^{nm-1} \left(1+\frac{r}{m}\right)^i = Cm \left(\frac{\left(1+\frac{r}{m}\right)^{nm} - 1}{r} \right)$$

$$PV = \sum_{j=1}^{nm} C\left(1+\frac{r}{m}\right)^{-j} = C \sum_{j=1}^{nm}\left(\frac{1}{1+\frac{r}{m}}\right)^{j}$$

$$= C\,\frac{1}{1+\frac{r}{m}} \sum_{j=0}^{nm-1}\left(\frac{1}{1+\frac{r}{m}}\right)^{j}$$

recall $\sum_{i=0}^{N} x^i = \frac{x^{N+1}-1}{x-1}$,

here: $N = nm-1$, $x = \frac{1}{1+\frac{r}{m}}$

$$= C\,\frac{1}{1+\frac{r}{m}}\;\frac{\left(1+\frac{r}{m}\right)^{-nm}-1}{\frac{1}{1+\frac{r}{m}}-1}$$

$$= C\,\frac{\left(1+\frac{r}{m}\right)^{-nm}-1}{1-\left(1+\frac{r}{m}\right)}$$

$$\Rightarrow \boxed{PV = Cm\left(\frac{1-\left(1+\frac{r}{m}\right)^{-nm}}{r}\right)}$$

- perpetual annuity: pays C every period (end of year) forever

$\Rightarrow \lim_{n\to\infty}$ in general ordinary annuity

$$\Rightarrow PV = \lim_{n\to\infty} \frac{Cm}{r}\left(1-\left(1+\frac{r}{m}\right)^{-nm}\right) = \frac{Cm}{r}, \text{ assuming } r>0.$$