Simple interest & compound interest

BOND VALUATION AND ANALYSIS IN PYTHON

Joshua Mayhew
Options Trader

Simple interest

Simple interest depends only on the initial deposit or loan.

We deposit USD 1,000 in a savings account

The account pays 5% simple interest each month

How much interest will we have earned after 1 year?

How much will our account be worth?

Simple interest

PV = Present Value = how much our money is worth today

FV = Future Value = how much our money is worth in the future

r = Interest Rate Per Period

n = number of periods

$$\text{Simple Interest Earned} = PV \times r \times n$$

$$\text{Future Value} = \text{Present Value} + \text{Simple Interest Earned}$$

Simple interest

```
pv = 1000
r = 0.05
n = 12
interest = pv * r * n
print(interest)
```

```
600
```

```
fv = pv + interest
print(fv)
```

```
1600
```

Compound interest

Compound interest means earning interest on our interest!

Deposit USD 1,000 in a bank account earning 5% compound interest per month.

Month	Starting Amount	Interest Earned	Ending Amount
1	1,000.00	1,000.00 * 0.05 = 50.00	1,000.00 + 50.00 = 1,050.00
2	1,050.00	1,050.00 * 0.05 = 52.50	1,050.00 + 52.50 = 1,102.50
3	1,102.50	1,102.50 * 0.05 = 55.13	1,102.50 + 55.13 = 1,157.63
...	...	...	...
12	1,710.34	1,710.34 * 0.05 = 85.52	1,710.35 + 85.52 = 1,795.86

USD 1,795.86 – USD 1,000.00 = USD 795.86 in compound interest

Compound interest

For 1 Period:

1,000 + (1,000 * 0.05) = 1,000 * 1.05 = 1,050

For 2 Periods:

1,050 * 1.05

= 1,000 * 1.05 * 1.05

= 1,000 * 1.05 ^ 2

For n Periods:

1,000 * 1.05 ^ n

Compound interest

The General Formula:

$$FV = PV \times (1 + r)^n$$

Compound interest

```
pv = 1000, r = 0.05, n = 12
fv = pv * (1 + r) ** n
print(fv)
```

```
1795.86
```

Let's practice!

BOND VALUATION AND ANALYSIS IN PYTHON

Future value & compounding frequencies

BOND VALUATION AND ANALYSIS IN PYTHON

Joshua Mayhew
Options Trader

Compound interest with multiple cash flows

- USD 1,000 deposit
- 3% interest rate paid monthly
- USD 100 top ups (extra deposits) at the end of each month
- How much do we have after 3 months?

Compound interest with multiple cash flows

```
deposit_fv = 1000 * (1 + 0.03) ^ 3
topup_1_fv = 100 * (1 + 0.03) ^ 2
topup_2_fv = 100 * (1 + 0.03) ^ 1
topup_3_fv = 100
print(deposit_fv + topup_1_fv + topup_2_fv + topup_3_fv)
```

```
1401.82
```

```
print(deposit_fv + topup_1_fv + topup_2_fv + topup_3_fv - 1000 - 100 - 100 - 100)
```

```
101.82
```

The future value function

- Previous approach can get very repetitive
- NumPy Financial can help simplify these calculations

```
import numpy_financial as npf
?npf.fv
```

```
Signature: npf.fv(rate, nper, pmt, pv)
Given:
 * an interest `rate` compounded once per period, of which there are
 * `nper` total
 * a (fixed) payment, `pmt`
 *  a present value, `pv`

Return:
   the value at the end of the `nper` periods
```

The future value function

- Rate: 3% per period (per month)
- Number of periods: 3 months
- Payment: USD -100 top ups at the end of each month
- PV: USD -1,000 deposit

The future value function

```
npf.fv(rate=0.03, nper=3, pmt=-100, pv=-1000)
```

```
1401.82
```

Compounding frequencies

How much do we have after 10 years investing $1,000 (no top-ups) at:

- 5% annual interest paid annually
- 5% annual interest paid monthly
- 5% annual interest paid daily

Compounding frequencies

- The rate is divided by the frequency and the number of periods multiplied by the frequency

```
# Using annual compounding frequency
npf.fv(rate=0.05, nper=10, pmt=0, pv=-1000)
```

```
1628.89
```

```
# Using monthly compounding frequency
npf.fv(rate=0.05/12, nper=10*12, pmt=0, pv=-1000)
```

```
1647.01
```

```
# Using daily compounding frequency
npf.fv(rate=0.05/365, nper=10*365, pmt=0, pv=-1000)
```

```
1648.66
```

Let's practice!

BOND VALUATION AND ANALYSIS IN PYTHON

More financial functions

BOND VALUATION AND ANALYSIS IN PYTHON

Joshua Mayhew
Options Trader

The nper() function

- Tells you number of periods required to grow PV to FV

```
import numpy_financial as npf
?npf.nper
```

```
Signature: npf.nper(rate, pmt, pv, fv=0)

Compute the number of periodic payments.

Parameters
rate : Rate of interest (per period)
pmt : Payment
pv : Present value
fv : (optional) Future value
```

The nper() function

- How long to save USD 7,000 investing USD 270 per month at 5% per year compounded monthly?

```
import numpy_financial as npf
npf.nper(rate=0.05/12, pmt=-270, pv=0, fv=7000)
```

```
24.67
```

The pmt() function

- Tells you the payment amount required to grow PV to FV

```
import numpy_financial as npf
?npf.pmt
```

```
Signature: npf.pmt(rate, nper, pv, fv=0)

Compute the payment against loan principal plus interest.

Given:
 * an interest `rate` compounded once per period, of which there are
 * `nper` total
 * a present value, `pv` (e.g., an amount borrowed)
 * a future value, `fv` (e.g., 0)

Return:
   the (fixed) periodic payment.
```

The pmt() function

- We have borrowed USD 275,000 at an annual rate of 3.5% compounded monthly
- What monthly payment to pay off a mortgage in ten years?

```
import numpy_financial as npf
npf.pmt(rate=0.035/12, nper=10*12, pv=275000, fv=0)
```

```
-2719.36
```

The rate() function

- Tells you interest rate required to grow PV to FV

```python
import numpy_financial as npf
?npf.rate
```

```
Signature: npf.rate(nper, pmt, pv, fv)

Compute the rate of interest per period.

Parameters
nper : Number of compounding periods
pmt : Payment
pv : Present value
fv : Future value
```

The rate() function

- What investment return to retire in 30 years?
- You save USD 1,500 each month and want to end up with USD 1 million.
- Assume the investments you make have monthly compounding.

```
import numpy_financial as npf
12 * npf.rate(nper=30*12, pmt=-1500, pv=0, fv=1000000)
```

```
0.0377
```

Let's practice!

BOND VALUATION AND ANALYSIS IN PYTHON