Class 2

Recap

Bond cash flow:

$$\begin{array}{cccccccc} & \boxed{C} & C & C & C & \cdots & C+F \\ 0 & 1 & 2 & 3 & 4 & \cdots & T \end{array}$$

coupon

Face value or PRINCIPAL

T → or m for MATURITY

COUPON RATE := C/F

Q What is the discount rate? A We solve the NPV eq. for r.

I.e.:

PRESENT VALUE = PRICE

$$\underbrace{\boxed{P}}_{\text{known}} = \frac{C}{1+y} + \frac{C}{(1+y)^2} + \dots + \frac{C}{(1+y)^{m-1}} + \frac{C+F}{(1+y)^m}$$

principal

maturity

SOLVE FOR y:= YIELD to MATURITY

TERMS:
- maturity m
- coupon
- coupon rate
- principal / Face value
- yield to maturity

SPOT RATE: yield on a zero-coupon bond

```
def price_of_bond(C, F, m, Y):
    P = 0
    for i in range(1, m):
        P += C/((1+Y)**(i))
    P += (C+F)/((1+Y)**(m))
    return round(P,2)
```

```
price_of_bond(2, 10, 10, 0.2)
```

```
10.0
```

Def $y = \frac{C}{F}$ ⇝ $P = F$ trading at PAR

$y > C/F$ ⇝ $P < F$ DISCOUNT

$y < C/F$ ⇝ $P > F$ PREMIUM

Rem Since C, F, m are fixed, only P and y change.

⇝ Details on Y: $1+Y = (1+r_{real})(1+\pi)$

real interest rate

(expected) inflation rate

```
from pynverse import inversefunc
```

```python
def yield_of_bond(C, F, m, P):
    price = lambda y : sum([C/((1+y)**i) for i in range(1,m)])+(C+F)/((1+y)**m)
    return float(inversefunc(price, y_values=P))
```

```python
yield_of_bond(2, 100, 2, 97)
```

```
0.03581045971234317
```

Def DURATION $D := \sum_{t=1}^{m} \left(\frac{PV(C_t)}{PV} \right) t$

```python
def NPV(r, C):
    #Calculate the Net Present Value of a sequence (array) C of cash flows
    #with discount rate r.
    NPV = 0
    for i in range(len(C)):
        NPV += C[i]/((1+r)**i)
    return NPV
```

```python
def duration(C, P, r):
    #We assume that we have C[0] to be the flow at time 0.
    D = 0
    for i in range(1,len(C)):
        D += NPV(r, [C[i] if i==j else 0 for j in range(i+1)])*i
    return round(D/P, 2)
```

Question 1
(a) Interest rate on a bond is determined by its coupon rate. True or false? Why?

(b) A bond price tends to rise when interest rate falls. True or false? Why?

(c) If there are two bonds with different maturities, the one with a longer maturity has a higher price sensitivity to interest rate changes. True or false? Why?

a) • COUPON RATE C/F

• INTEREST RATE = YIELD TO MATURITY

= solution in y to $P = C_0 + \frac{1}{(1+y)} C_1 + \dots$

b) TRUE. We care more about the future bc we discount less

c) FALSE. Price sensitivity $\approx$ duration $\neq$ maturity.

Question 2
The following is a list of prices for zero-coupon bonds of various maturities. The face value is $1000.

Maturity (years)	Price of bond ($)
1	943.40
2	898.47
3	847.62
4	792.16

Calculate the yields to maturity of each bond. What is the shape of the yield curve?
Use the given information to compute the price of a 4-year bond with a 4% coupon and the face value of $1000.

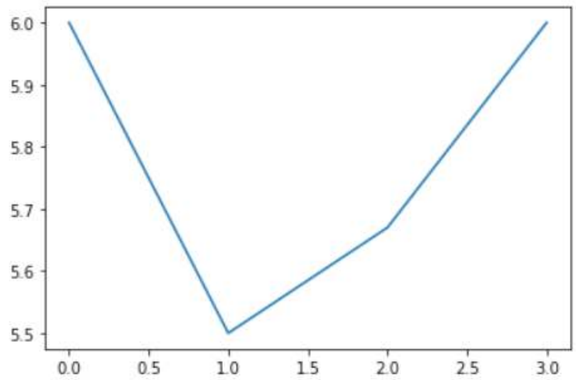
```
P=[943.4, 898.47, 847.62, 792.16]
for i in range(1,5):
    print(yield_of_bond(0, 1000, i, P[i-1]))
```

```
6.0
5.5
5.67
6.0
```

```
P = [943.4, 898.47, 847.62, 792.16]
Y = [0 for i in range(4)]
for i in range(1,5):
    Y[i-1] = yield_of_bond(0, 1000, i, P[i-1])
```

```
plt.plot(Y)
```

```
[<matplotlib.lines.Line2D at 0x7f89d837dfa0>]
```



For the last point, use ADDITIVITY!

Question 3
Consider the following estimates of spot rates:

Year	Spot Rate
1	5.00%
2	5.40%
3	5.70%
4	5.90%
5	6.00%

What can you deduce about the one-year spot interest rate in four years if
(a) The expectations theory of term structure is right?
(b) The liquidity-preference theory of term structure is right?
(c) The term structure contains an inflation uncertainty premium?

1-year interest rate in 4 years

a) we have $I'_4 \; \frac{(1.06)^5}{(1.059)^4} - 1 = 0.064 = 6.4\%$

b) we are now overestimating the I'_4 bc we are not certain

c) Same as b)

Question 4

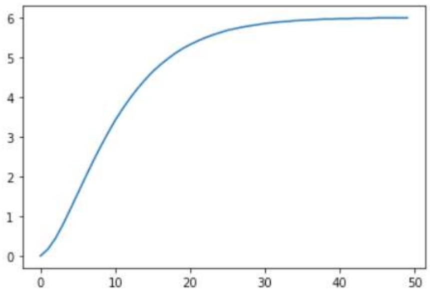
The formula for the duration of a perpetual bond which makes an equal payment each year in perpetuity is (1+yield)/yield.

If bonds yield 5%, which has the longer duration – a perpetual bond or a 15-year zero-coupon bond? What if the yield is 10%?

Do the calculation...

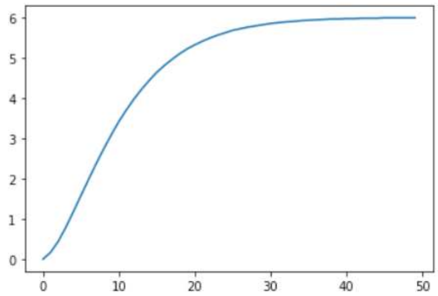
```
n=100
C_p=100
r=0.2
D=[0 for i in range(50)]
for i in range(50):
    C=[0]+[C_p for j in range(i)]
    D[i]=duration(C, C_p/r, r)
plt.plot(D)
```

```
[<matplotlib.lines.Line2D at 0x7f89ca4024f0>]
```



```
n=100
C_p=10
r=0.2
D=[0 for i in range(50)]
for i in range(50):
    C=[0]+[C_p for j in range(i)]
    D[i]=duration(C, C_p/r, r)
plt.plot(D)
```

```
[<matplotlib.lines.Line2D at 0x7f89a847ddc0>]
```



```
n=100
C_p=10
r=0.3
D=[0 for i in range(50)]
for i in range(50):
    C=[0]+[C_p for j in range(i)]
    D[i]=duration(C, C_p/r, r)
plt.plot(D)
```

```
[<matplotlib.lines.Line2D at 0x7f89ca450970>]
```