MGMTMFE 400 Investment

Homework 1

October 4, 2023

Group 2:

Helina (Yiwei) Cheng

Mayuresh Budukh

Ruilin Shi

Jenny (Yunshan) Chen

In [44]:
```
import numpy as np
import numpy_financial as npf
import pandas as pd
```

Assignment 1

Problem 1

you can invest $\$200,000$ in a certificate of deposit offered by your bank. The CD is for two years and the bank quotes you a rate of 4%. How much will you have in two years if the 4% is :

In [28]:
```
c = 200000
r = 0.04
t = 2
```

(a) An EAR?

$$FV = P \times (1 + EAR)^t = 200,000 \times (1 + 4\%)^2 = 216,320$$

In [29]:
```
FV = c*(1+r)**t
print(f'EAR = ${FV:.2f} ')
```

```
EAR = $216320.00
```

(b) A quarterly APR?

$$FV = P \times (1 + \frac{APR}{4})^{4t} = 200,000 \times (1 + \frac{4\%}{4})^{4\times2} = 216,571.34$$

In [31]:
```
FV = c*(1+r/4)**(t*4)
print(f'Quarterly APR = ${FV:.2f} ')
```

```
Quarterly APR = $216571.34
```

(c) A monthly APR?

$$FV = P \times (1 + \frac{APR}{12})^{12t} = 200,000 \times (1 + \frac{4\%}{12})^{12\times2} = 216,628.59$$

In [33]:
```
FV = c*(1+r/12)**(t*12)
print(f'Monthly APR = ${FV:.2f} ')
```

```
Monthly APR = $216628.59
```

Problem 2

you are trying to determine your standard of living after retirement. You make the following assumptions. First, you will earn $\$100,000$ for each of the next 35 years, and save 45% of that amount. Second, payments are annual and the first payment is one year from today. Third, interst rates will be 2% per year forever.

In [40]:
```
pmt = 100000
perc = 0.45
r = 0.02
t = 35
```

(a) Compute how much money you will have saved at the end of 35th year.

Annual Savings:
The amount saved each year is:

$$100,000 \times 45\% = \$45,000$$

Future Value of Annuity:

I save this amount into saving account each year, earning 2% interest. This is an annuity with payment of $\$45,000$ for 35 periods.

This will give me

$$FV = P \times \left(\frac{(1+r)^n - 1}{r} \right)$$

At the end of 35th year, I will have

$$FV = 45,000 \times \left(\frac{(1.02)^{35} - 1}{0.02} \right) = \$2,249,751.49$$

In [41]:
```
def get_FV(pmt, percent, r, t):
    saving = pmt*percent
    FV = saving*((1+r)**(t)-1)/r
    return FV

FV = get_FV(pmt, perc, r, t)
print(f'I will have saved ${FV:.2f} at the end of 35th year')
```

```
I will have saved $2249751.49 at the end of 35th year
```

(b) What is the amount you can consume during each of your retirement years?

Assume that there are 20 retirement years and that consumption takes place at the end of each year.

The amount I saved need to be used for next 20 years.

$$PMT = FV \times \frac{r}{1-(1+r)^{-n}} = FV \times \frac{0.02}{1-(1.02)^{-20}} = \$137587.42$$

In [42]:
```
n = 20 # retirement years
pmt = FV * r/(1-(1+r)**(-n))
print(f'I can consume ${pmt:.2f} during each of the retirement year
```

```
I can consume $137587.42 during each of the retirement years
```

Problem 3

You are considering buying a two-bedroom townhouse for $\$700,000$. You plan to make a $\$200,000$ down payment and take a $\$500,000$ 30-year mortgage for the rest. The interest rate on the mortgage is 3% APR for monthly payments.

(a) What is the effective annual rate?

$$(1+\frac{0.03}{12})^{12} = 3.0416\%$$

In [48]:
```
def effective_annual_rate(APR, n):
    return (1 + APR/n)**n - 1

APR = 0.03
n = 12

EAR = effective_annual_rate(APR, n)
print(f'Effective Annual Rate (EAR) = {EAR*100:.4f}%')
```

```
Effective Annual Rate (EAR) = 3.0416%
```

(b) What is the monthly payment?

The monthly payment calculated from total of principal and interest payments

$$M = \frac{P \times r(1+r)^n}{(1+r)^n - 1} = \frac{500000 \times (\frac{0.03}{12}) \times (1 + \frac{0.03}{12})^{360}}{(1 + \frac{0.03}{12})^{360} - 1} = \$2108.02$$

In Python, we can use built-in function `npf.pmt` **to get the payment**
Below are the codes used to calculate

In [72]:
```
P = 500000  # Principal loan amount
r = APR / 12  # Monthly interest rate
n = 30 * 12  # Total number of monthly payments

M = npf.pmt(r, n, P)
print(f'Monthly payment (M) = ${-M:.2f}')
# Using a negative sign to display as a positive value since npf.pm
```

```
Monthly payment (M) = $2108.02
```

(c) How much do you owe the bank immediately after the twentieth monthly payment?

In [79]:
```
payments = pd.DataFrame(columns=['month', 'total_payment', 'interes
n = 20 #20 payments made

for t in range(n):
    payments = payments.append({'month': t+1, 'total_payment': -M},

for t in range(n):
    payments.iloc[int(t), 2] = P*r
    payments.iloc[int(t), 3] = -M - P*r
    P = P - (-M - P*r)
    payments.iloc[int(t), 4] = P

payments
```

Out[79]:

	month	total_payment	interest_payment	principal_payment	amount_left
0	1.0	2108.020169	1250.000000	858.020169	499141.979831
1	2.0	2108.020169	1247.854950	860.165219	498281.814612
2	3.0	2108.020169	1245.704537	862.315632	497419.498980
3	4.0	2108.020169	1243.548747	864.471421	496555.027559
4	5.0	2108.020169	1241.387569	866.632600	495688.394959
5	6.0	2108.020169	1239.220987	868.799181	494819.595778
6	7.0	2108.020169	1237.048989	870.971179	493948.624599
7	8.0	2108.020169	1234.871561	873.148607	493075.475992
8	9.0	2108.020169	1232.688690	875.331479	492200.144513
9	10.0	2108.020169	1230.500361	877.519807	491322.624706
10	11.0	2108.020169	1228.306562	879.713607	490442.911099
11	12.0	2108.020169	1226.107278	881.912891	489560.998208
12	13.0	2108.020169	1223.902496	884.117673	488676.880535
13	14.0	2108.020169	1221.692201	886.327967	487790.552567
14	15.0	2108.020169	1219.476381	888.543787	486902.008780
15	16.0	2108.020169	1217.255022	890.765147	486011.243633
16	17.0	2108.020169	1215.028109	892.992060	485118.251574
17	18.0	2108.020169	1212.795629	895.224540	484223.027034
18	19.0	2108.020169	1210.557568	897.462601	483325.564433
19	20.0	2108.020169	1208.313911	899.706258	482425.858176

Above is the table that calculate the amount left by monthes, or we can use below formula to get the remaining balance.

In [51]:
```
t = 20  # Number of payments made
n = 30 * 12  # Total number of monthly payments
B = M * (1 - (1 + r)**-(n-t))/r
print(f'Remaining balance after 20 payments (B) = ${-B:.2f}')
```

```
Remaining balance after 20 payments (B) = $482425.86
```

Problem 4

Assume the spots rates are as follows:

$$\begin{aligned} s_1 &= 2.0\% \\ s_2 &= 2.5\% \\ s_3 &= 3.0\% \\ s_4 &= 3.5\% \end{aligned}$$

Spot rates are with annual compounding, coupon payments are annual, and par values are $\$100$.
Compute the prices of the following bonds:

In [21]:
```
spot_rate = [0.02, 0.025, 0.03, 0.035]
par = 100
```

(a) A zero-coupon bond with 3 years to maturity.

$$P_1 = \frac{100}{1.03^3} = \$91.51$$

Below are the codes used to calculate

In [22]:
```
t = 3
price = par/(1+spot_rate[t-1])**t
print(f'The price for a 3 years zero-coupon bond is : ${price:.2f}'
```

```
The price for a 3 years zero-coupon bond is : $91.51
```

(b) A bond with coupon rate 1% and two years to maturity.

$$P_2 = \frac{1}{1.02} + \frac{101}{1.025^2} = \$97.11$$

Below are the codes used to calculate

In [23]:
```
c = 1
t = 2
price = sum(c/(1+spot_rate[t-1])**t for t in range(1, t+1)) + par/(
print(f'The price for a 2 years 1% coupon bond is : ${price:.2f}')
```

```
The price for a 2 years 1% coupon bond is : $97.11
```

(c) A bond with coupon rate 13% and four years to maturity.

$$P_3 = \frac{13}{1.02} + \frac{13}{1.025^2} + \frac{13}{1.03^3} + \frac{113}{1.035^4} = \$135.49$$

Below are the codes used to calculate

In [24]:
```
c = 13
t = 4
price = sum(c/(1+spot_rate[t-1])**t for t in range(1, t+1)) + par/(
print(f'The price for a 4 years 13% coupon bond is : ${price:.2f}')
```

```
The price for a 4 years 13% coupon bond is : $135.49
```

Problem 5

You observe prices for the following bonds:
Bond X: coupon rate = 3%, maturity in 6 months, Price in $\$98.98$
Bond Y: coupon rate = 4%, maturity in 12 months, Price in $\$98.59$

Coupon payments are semi-annual.
Determine the 6-month and 1-year spot rates, both expressed as APRs with semi-annual compounding.

For Bond X

$$98.98 = \frac{101.5}{1 + r_{0.5}}$$
$$r_{0.5} = \frac{101.5}{98.98} - 1 = 0.02546$$

Express spot rate as APR with semi-annual compounding

$$APR_{0.5} = r \times 2 = 5.092\%$$

For Bond Y

$$98.59 = \frac{2}{1 + r_{0.5}} + \frac{102}{(1 + r_1)^2}$$

$$98.59 = \frac{2}{1 + 0.02546} + \frac{102}{(1 + r_1)^2}$$

$$r_1 = 0.0274$$

Express spot rate as APR with semi-annual compounding

$$APR_1 = r \times 2 = 5.472\%$$

Below are the codes used to calculate

In [2]:
```
import numpy as np
def spot_rate(pv,fv,coupon,period): #function to calculate spot rat

    if period == 1 : #if period is 1, discount the fv + coupon that
        r1 = ((fv + coupon)/pv) - 1
        spot_rates.append(r1)

    if period == 2 : #if period is 2, discount the coupon that we g
                        #the fv + coupon that we get in 1 year
            r2 = np.sqrt((fv + coupon)/(pv - (coupon/(1+ spot_rates
            spot_rates.append(r2)

    return np.array(spot_rates) * 2 * 100

spot_rates = [] #create an empty list to store the spot rates
spot_rate(98.98,100,3/2,1) #function call for bond X
spot_rate(98.59,100,4/2,2) #function call for bond Y

print(f'The 2 required spot rates expressed as APRs are : {np.array
```

```
The 2 required spot rates expressed as APRs are : 5.09193776520509
and 5.471880532776252
```

Problem 6

You are holding a 3-year bond with coupon rate of 5%. Coupon payments are annual and par values are $\$100$. Spot rates are: $r_1 = 5\%, r_2 = 6\%, r_3 = 7\%$.

(a) Determine the bond's price and yield-to-maturity

$$P = \frac{5}{1.05} + \frac{5}{1.06^2} + \frac{105}{1.07^3} = \$94.92$$

Plug P into the formula:

$$94.92 = \frac{5}{1+y} + \frac{5}{(1+y)^2} + \frac{105}{(1+y)^3}$$

Using finance calculator,

$$y = 6.93\%$$

Below are the codes used to calculate

In [54]:
```
#imported the library to calculate IRR
#calculated the discount factors assiocated with each spot rate
d1 = 1/(1.05)
d2 = 1/(1.06**2)
d3 = 1/(1.07**3)

par_value = 100
coupon_rate = 5
coupon_dollar = coupon_rate/100 * par_value

#discounted the coupons and the par value to get price of the bond
price = (coupon_dollar*d1) + (coupon_dollar*d2) + ((par_value + cou
print (f'Price of the bond is {price}')

#used the built in library to calculate ytm of the bond
x = np.array([-price,coupon_dollar, coupon_dollar, (par_value + cou
print(f'The YTM of the bond is : {npf.irr(x)*100}')
```

```
Price of the bond is 94.92316403551541
The YTM of the bond is : 6.932138733219806
```

(b) Determine as many forward rates as you can, based on the spot rates above

$$f_{0,1} = r_1 = 5\%$$

$$f_{1,2} = \frac{(1+r_2)^2}{1+r_1} - 1 = 7.0095\%$$

$$f_{2,3} = \frac{(1+r_3)^3}{(1+r_2)^2} - 1 = 9.0284\%$$

$$f_{0,2} = r_2 = 6\%$$

$$f_{1,3} = \sqrt{\frac{(1+r_3)^3}{1+r_1}} - 1 = 8.01424\%$$

$$f_{0,3} = r_3 = 7\%$$

Below are the codes used to calculate

In [55]:
```
#calculate forward rates using the given spot rates
f12 = (1.06**2)/(1.05) - 1
f13 = np.sqrt((1.07**3)/1.05) - 1
f23 = (1.07**3)/(1.06**2) - 1

print(f'The forward rate between year 0 and 1 is {5} %')
print(f'The forward rate between year 1 and 2 is {f12*100} %')
print(f'The forward rate between year 1 and 3 is {f13*100} %')
print(f'The forward rate between year 0 and 2 is {6} %')
print(f'The forward rate between year 2 and 3 is {f23*100} %')
print(f'The forward rate between year 0 and 3 is {7} %')
```

```
The forward rate between year 0 and 1 is 5 %
The forward rate between year 1 and 2 is 7.009523809523821 %
The forward rate between year 1 and 3 is 8.014240683699626 %
The forward rate between year 0 and 2 is 6 %
The forward rate between year 2 and 3 is 9.028390886436455 %
The forward rate between year 0 and 3 is 7 %
```

(c) You would like to get a guaranteed 3-year return on your coupon bond. Explain how this can be achieved using forward rates.

To achieve a guaranteed 3-year return, we need to **reinvest the coupons at the forward rate**.
We need to **invest first year coupon at a forward rate $f_{1,3}$ for 2 years, and the 2nd year coupon at a forward rate $f_{2,3}$ for 1 year.**

In [56]:
```
reinvested_coupons = coupon_dollar * ((1 + f13)**2) + coupon_dollar
future_guar_value = reinvested_coupons + 100 + coupon_dollar
guar_ret = (future_guar_value/price)**(1/3) - 1
print(f'The guaranteed return is {guar_ret * 100}%')
```

```
The guaranteed return is 7.000000000000006%
```