

```
In [1]: import numpy as np
import pandas as pd
```

Authors: Samir Chowdhury, Julanna Zhang, Cathy Zhuang

Question 1

Lanni Products is a start-up computer software development firm. It currently owns computer equipment worth 30,000 and has cash on hand of 20,000 contributed by Lanni's owners. For each of the following transactions, identify the real and/or financial assets that trade hands. Are any financial assets created or destroyed in the transaction?

- (a) [4pts] Lanni takes out a bank loan. It receives \$50,000 in cash and signs a note promising to pay back the loan over 3 years.
- (b) [4pts] Lanni uses the cash from the bank plus \$20,000 of its own funds to finance the development of new financial planning software.
- (c) [4pts] Lanni sells the software product to Microsoft, which will market it to the public under the Microsoft name. Lanni accepts payment in the form of 1,250 shares of Microsoft stock.
- (d) [4pts] Lanni sells the shares of stock for \$100 per share and uses part of the proceeds to pay off the bank loan.

FA = Financial Asset, RA = Real Asset

- (a) The bank loan is an FA
- (b) Lanni trades an FA of cash for an RA of the software
- (c) Lanni trades the RA of intellectual property of the software for an FA that is equity in MSFT
- (d) Lanni trades the FA of equity for the FA of cash and trades that FA of cash to destroy the FA the bank had which was her debt to them.

Question 2

Find the equivalent taxable yield of a short-term municipal bond with a yield of 3% for tax brackets of

- (a) [3pts] zero
- (b) [3pts] 10%
- (c) [3pts] 20%
- (d) [3pts] 40%

Why are high-tax-bracket investors more inclined to invest in municipal bonds than low-bracket investors? [4pts]

Formula for taxable equivalent yield:

$$r_{taxable}(1 - t) = r_{muni} \text{ or } r_{taxable} = \frac{r_{muni}}{(1-t)} \text{ where } t = \text{marginal tax bracket}$$

```
In [2]: # Define function to find taxable yield given tax brackets
def taxable_yield(tax_rate, te_yield=0.03):
    return te_yield/(1-tax_rate)
```

```
In [3]: # Test the yield for multiple tax brackets
for tax_rate in [0, .1, .2, .4]:
    print(f'Tax Bracket: {tax_rate*100:.2f}% -> Equivalent Taxable Yield: {t
```

```
Tax Bracket: 0.00% -> Equivalent Taxable Yield: 3.00%
Tax Bracket: 10.00% -> Equivalent Taxable Yield: 3.33%
Tax Bracket: 20.00% -> Equivalent Taxable Yield: 3.75%
Tax Bracket: 40.00% -> Equivalent Taxable Yield: 5.00%
```

High-tax-bracket investors are more inclined to invest in municipal bonds than low-bracket investors because interest income earned from municipal bonds is tax-free. If two assets provide the same gross yield, the municipal bond will have a higher net yield due to being tax exempt.

Question 3

Looking at the Treasury bond maturing in Feb 2036 at the \$1,000 par value in Figure 1

- (a) [3pts] How much would you have to pay to purchase one of these bonds?
- (b) [3pts] How much would you get to sell one of these bonds?
- (c) [3pts] What was its ask price the previous day?
- (d) [3pts] What is its coupon rate? If the bond makes semiannual coupon payments, then how much would you get in each coupon payment?



Figure 1: BID, ASKED and CHANGE columns are quoted as a percentage of par.

```
In [4]: # Define a function to calculate bid and ask price of a bond
def bid_price(bid_yield, par_value):
    return bid_yield * par_value

def ask_price(ask_yield, par_value):
    return ask_yield * par_value
```

```
In [5]: # Run the functions to find the bid and ask prices
print(f'For a T-Bond w/ Par Value: $1000 maturing on Feb 2036\n(a): Bid Price: {bid_price(1.255313, 1000):.2f}\n(b): Ask Price: {ask_price(1.255313, 1000):.2f}')
```

For a T-Bond w/ Par Value: \$1000 maturing on Feb 2036

(a): Bid Price: \$1254.69

(b): Ask Price: \$1255.31

```
In [6]: # Calculate the previous ask yield off the change column and run the function
print(f'New Ask Yield for One Day Prior: 125.5313-1.5391 = {125.5313-1.5391:}
(c): Prior Day Asking Price: ${ask_price((125.5313-1.5391)/100, 1000):.2f}')
```

New Ask Yield for One Day Prior: 125.5313-1.5391 = 123.99

(c): Prior Day Asking Price: \$1239.92

```
In [7]: # Define function to calculate coupon payments
def coupon_payment(coupon_rate, par_value):
    return coupon_rate * par_value
```

```
In [8]: # Find the coupon payment and divide by 2 for the semiannual payment
print('The coupon rate is the annual yield the bond will pay out based off i
print(f'(d): Semi Annual Payment for $1000 T-Bond w/ Coupon Rate of 6.125: $
```


The coupon rate is the annual yield the bond will pay out based off its par value

(d): Semi Annual Payment for \$1000 T-Bond w/ Coupon Rate of 6.125: \$30.62

Question 4

Looking at the listing for Hess Midstream Partners in Figure 2

- (a) [4pts] How many shares could you buy for \$1,000? Truncate the number of shares to an integer.
- (b) [4pts] What would be your annual dividend income from those shares?
- (c) [4pts] What must be Hess Midstream Partners' earnings per share?
- (d) [4pts] What was the firm's closing price on the day before the listing?

 image Figure 2: Listing of stocks traded on the New York Stock Exchange.

```
In [9]: # (a) integer divide closing price by $1000
print(f'(a): # of purchasable stocks for HESM for $1000: {1000//17.87:.0f}')
```

(a): # of purchasable stocks for HESM for \$1000: 55

```
In [10]: # (b) Define function to find dividend yield and print
print(f'(b): Annual Dividend Yield: ${1.43*55:.2f}')
```

(b): Annual Dividend Yield: \$78.65

To find the Earnings Per Share, we use the following formula

$$P/E = \frac{StockPrice}{EarningsPerShare}$$

We rearrange it to solve for Earnings Per Share

$$EarningsPerShare = \frac{StockPrice}{P/E}$$

Which in our case would result in

$$EarningsPerShare = \frac{17.87}{14.60}$$

```
In [11]: # (c) Print out earnings per share
print(f'(c): Earnings Per Share: ${17.87/14.60:.2f}')
```

(c): Earnings Per Share: \$1.22

```
In [12]: # (d) Subtract net change from today's close to get closing price
print(f'(d): Previous day's close: ${17.87-0.25:.2f}')
```

(d): Previous day's close: \$17.62

Question 5

Consider the three stocks in the following table. P_t represents price at time t , and Q_t represents shares outstanding at time t . Stock Z splits two for one in the last period.

image \

- (a) [5pts] Calculate the rate of return on a price-weighted index of the three stocks for the first period ($t = 0$ to $t = 1$).
- (b) [5pts] What must happen to the divisor for the price-weighted index in year 2?
- (c) [5pts] Calculate the rate of return for the second period ($t = 1$ to $t = 2$).

Calculate the first-period rates of return on the following indexes of the three stocks:

- (a) [5pts] A market-value-weighted index.
- (b) [5pts] An equally weighted index.

```
In [13]: # (a) Store table in array
prices = [[100, 95, 95], [200, 220, 220], [180, 200, 100]]
volume = [[200, 200, 200], [100, 100, 100], [200, 200, 400]]

# Calculate the PWA of each column using defined function
def price_weighted_average(prices, time, divisor=len(prices)):
    prices_at_time = [i[time] for i in prices]
    return np.sum(prices_at_time)/divisor

pwa_0, pwa_1 = price_weighted_average(prices, 0), price_weighted_average(prices, 1)
print(f'PWA at t=0: ${pwa_0:.2f}')
print(f'PWA at t=1: ${pwa_1:.2f}')
print(f'(a): RoR on PWA from t=0 to t=1: {((pwa_1-pwa_0)/pwa_0)*100:.2f}%')
```

PWA at t=0: \$160.00

PWA at t=1: \$171.67

(a): RoR on PWA from t=0 to t=1: 7.29%

(b) The divisor must decrease due to the stock split. This is because since the price of the stock has halved, the divisor must decrease to maintain the same price weighted average. We solve for the divisor below:

```
In [14]: pwa_pre_split = price_weighted_average(prices, time=1)
d = np.sum([i[2] for i in prices])/pwa_pre_split
print(f'Original Divisor: {len(prices)}, New Divisor: {d}')

Original Divisor: 3, New Divisor: 2.41747572815534

In [15]: # (c) Rate of Return from t=1 to t=2 w/ Stock Split
pwa_1 = price_weighted_average(prices, 1)
pwa_2 = price_weighted_average(prices, 2, d)
print(f'PWA, t=1, d={len(prices):.3f}: ${pwa_1:.2f}\nPWA, t=2, d={d:.3f}: ${pwa_2:.2f}')
print(f'(c): RoR on PWA from t=1 to t=2 w/ Stock Split: {(pwa_2-pwa_1)/pwa_1}')

PWA, t=1, d=3.000: $171.67
PWA, t=2, d=2.417: $171.67
(c): RoR on PWA from t=1 to t=2 w/ Stock Split: 0.00

In [25]: # (a) First Period Return Rate on MWA
def market_weighted_average(prices, volume, time):
    prices_at_time = np.asarray([i[time] for i in prices])
    volume_at_time = np.asarray([i[time] for i in volume])
    return np.sum(prices_at_time*volume_at_time)

mwa_0 = market_weighted_average(prices, volume, 0)
mwa_1 = market_weighted_average(prices, volume, 1)
print(f'Index Value, t=1: ${mwa_0:.2f}\nIndex Value, t=2: ${mwa_1:.2f}\n(a): RoR on MWA from t=0 to t=1: {((mwa_1-mwa_0)/mwa_0)*100:.2f}%')

Index Value, t=1: $76000.00
Index Value, t=2: $81000.00
(a): RoR on MWA from t=0 to t=1: 6.58%

In [17]: # (b) First Period Return Rate on EWA
def equally_weighted_average_ror(prices, time1, time2):
    prices_at_time1 = np.asarray([i[time1] for i in prices])
    prices_at_time2 = np.asarray([i[time2] for i in prices])
    ratios = prices_at_time2 / prices_at_time1
    return np.mean(ratios)-1

print(f'(a): RoR on EWA from t=0 to t=1: {equally_weighted_average_ror(prices, 0, 1)*100:.2f}%')

(a): RoR on EWA from t=0 to t=1: 5.37%
```

Question 6

Suppose you buy a February 2023 expiration Apple call option with an exercise price \$135.

- (a) [5pts] Suppose the stock price in September is 140. Will you exercise your call? What is the profit of your position? You can ignore the price of the call.
- (b) [5pts] What if you had bought the February 2023 call with exercise price 145?
- (c) [5pts] What if you had bought a February 2023 put with exercise price 145?

(a) Given an exercise price of 135, I will exercise the call if the price is currently 140 because the current price is higher than the strike price. I will profit $140 - 135 = 5$ dollars

from exercising the call not accounting for the price of the call.

(b) If the exercise price had been 145, I would not exercise the call because the strike price is lower than the current price. If I were to exercise, I would be buying the stock at 145 when it is currently valued at 140, thus losing me 5 dollars so therefore I would not exercise the call and lose 0.

(c) If I had bought a put option with an exercise price at 145, I would exercise the option. I would be able to sell my share for 145 when the share is currently valued at 140, thus resulting in 5 dollar profit.