Python for Capital Asset Pricing Model: CAPM Part 1

Mamoru Nagano

I CAPM: Capital Asset Pricing Model

- ✓ CAPM defines stock price volatility as the cost of equity and explains the expected return by investing the cost of equity.

- ✓ In other words, the CAPM theoretically explains the trade-off relationship between the expected return and risk of individual stocks.

- ✓ Unlike the Capital Market Line (CML), the Security Market Line (SML) of CAPM denotes the relationship between the expected return and risk (beta) of individual stocks.

- ✓ The CAPM interprets the extent to which individual stocks overperform or underperform all the listed stock market capitalization, i.e., market portfolio, " as risk premium ≒ cost of shareholders' equity.

【Balance Sheet】

【Debit Side】

Current Asset

Cash and Deposits

…

Fixed Asset

Tangible Fixed Assets

Intangible Assets

…

Investment Results

【Credit Side】

Current Liability

…

Fixed Debt

Long-term Borrowing

Bond Issued …

Net Worth

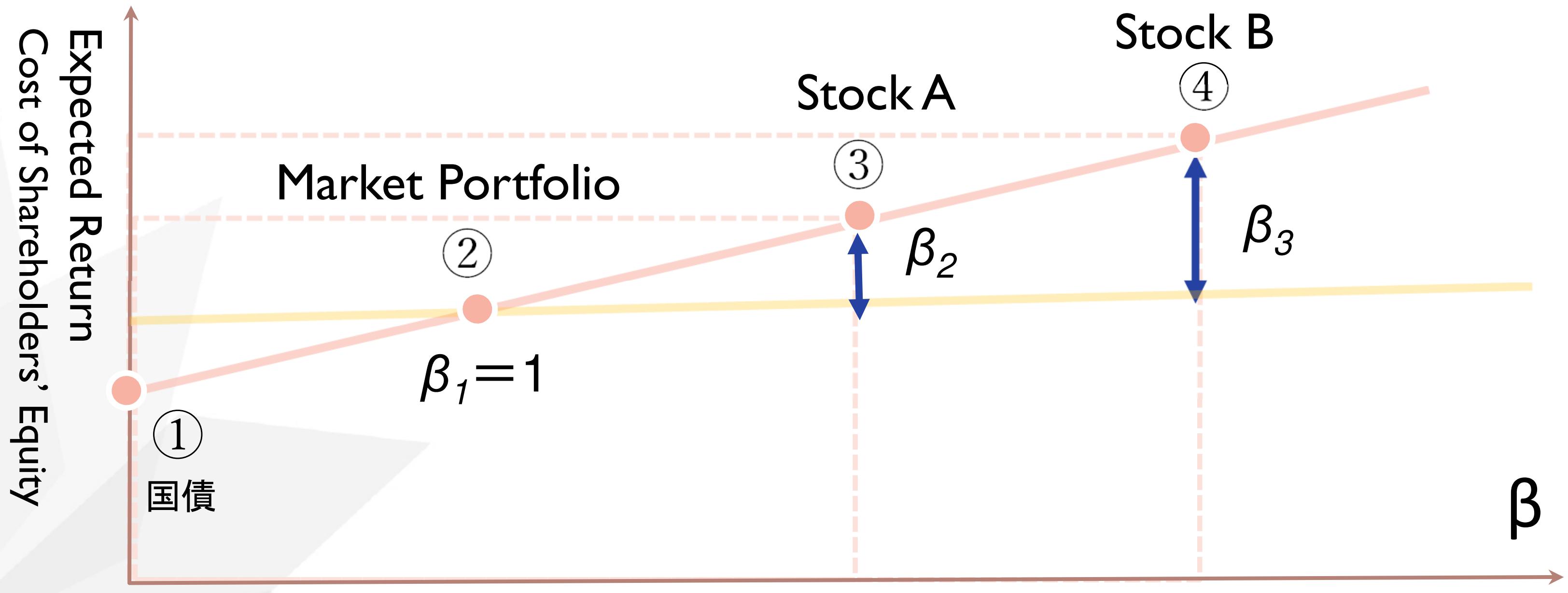
Shareholders' Equity

…

Corporate Value

2 SML and Beta

✓ The degree to which individual stocks overperform or underperform all the listed stock market capitalization, i.e., market portfolio," as risk premium ≒ cost of shareholders' equity.



- ✓ The securities market line (SML) is a straight line that explains the linear relationship between the beta value, which is the risk premium of the expected return, and the expected return.

- ✓ If stock A's expected return matches the market portfolio's risk premium, its beta will be "1"; if it is above (below), its beta will be above (below) "1".

- ✓ Points ③ and ④ on the previous page represent the expected returns of equity investment to firms with higher beta values than the market portfolio.

3 How to Estimate Beta

- ✓ Here, let $E(R_i)$ be the expected return of stock i, $E(R_M)$ be the expected return of the market portfolio, and R_f be the expected return of the risk-free asset.

- ✓ In this case, the market β of stock i can be derived using the following formula.

$$E(R_i) = R_f + \beta_i[E(R_M) - R_f]$$

- ✓ β is the value obtained by dividing the covariance between firm i's stock price and the expected return of the market portfolio by the variance of the market portfolio.

$$\beta_i = \frac{\mathrm{Cov}(R_i, R_M)}{\mathrm{Var}(R_M)}$$

- ✓ As shown on the previous page, $E(R_i)$ consists of two terms: R_f and $\beta_i[E(R_M) - R_f]$.

$$E(R_i) = R_f + \beta_i[E(R_M) - R_f]$$

- ✓ By expanding the latter term, β_i can be directly estimated. $(1 - \beta_i)R_f$ is estimated as a constant term, but it does not have any particularly important meaning.

$$E(R_i) = (1 - \beta_i)R_f + \beta_i E(R_M)$$

4 Estimating Beta: Libraries

- ✓ The following four libraries are used to estimate the market β in this example.
- ✓ You can also use statmodels instead of sklearn.
- ✓ The reason for using sklearn is to allow advanced users to estimate the β value using machine learning.

```
#[1] Library Imports

import pandas as pd
import pandas_datareader as data
import datetime
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression
```

5 Estimating Beta: Getting Stock Price Data

- ✓ Let's compare Tesla (TSLA.US), a 21st century EV maker, and GM (GM.US), a 20th century gasoline car maker.
- ✓ First, obtain stock price data by specifying the sample period using datetime.
- ✓ Tesla (TSLA.US) is listed on the NASDAQ, and GM (GM.US) is listed on the NYSE, so their respective market portfolios are the NASDAQ Composite Index (^NDQ) and the Dow Jones Industrial Average Index (^DJI).

```
#[2] Data Retrieval

tickers=['TSLA.US','^NDQ','GM.US','^DJI']
start=datetime.date(2012,1,1)
end=datetime.date(2022,12,30)
df=data.DataReader(tickers,'stooq',start=start,end=end).sort_values(by='Da
te',ascending=True)
```

6 Estimating Beta: Data Preprocessing

- ✓ Calculate the percentage change from the previous day using only closing price data and remove missing values.
- ✓ Redefine the four variable names 'Tesla', 'Nasdaq', and 'GM', 'DowJones, respectively.
- ✓ the 3rd line instructions to change the variable name to the 4 variables on the 2nd line.

```
#[3] Data Preprocessing

df = df['Close'].pct_change().dropna()*250
company_list=['Tesla','Nasdaq','GM','DowJones']
df.columns = company_list
df.describe()
```

7 Estimating Beta: Tesla

- ✓ Lines 2 and 3 convert the variables df [['Nasdaq']], df[['Tesla']] from the pandas dataframe to ndarrays.
- ✓ Line 4-5 instructions to remove missing values for variable names.

```
#[4] Estimating Beta : Tesla

lr1 = LinearRegression()
X = df[['Nasdaq']].values
Y = df[['Tesla']].values
X1=np.delete(X,0,0)
Y1=np.delete(Y,0,0)
lr1.fit(X1,Y1)
print('β=%.4f'% lr1.coef_[0])
print('intercept = %.4f'% lr1.intercept_)
print('R_squared = %.4f '% lr1.score(X1,Y1))
```

8 Estimating Beta: GM

- ✓ Lines 2 and 3 convert the variables df [[‘DowJones’]], df[[‘GM ']] from the pandas dataframe to ndarrays.
- ✓ Line 4-5 instructions to remove missing values for variable names.

#[5] Estimating Beta : GM

```
lr2 = LinearRegression()
Z = df[['DowJones']].values
W = df[['GM']].values
X2=np.delete(Z,0,0)
Y2=np.delete(W,0,0)
lr2.fit(X2,Y2)
print('β = %.4f'% lr2.coef_[0])
print('intercept = %.4f'% lr2.intercept_)
print('R_squared = %.4f '% lr2.score(X1,Y1))
```

9 Estimating Beta: Results

- ✓ Investors require higher expected returns when investing in Tesla (TSLA.US).
- ✓ GM(GM.US) 's β value is not low by any means, and is much higher than Toyo ta (7203.JP).

Tesla (TSLA.US)

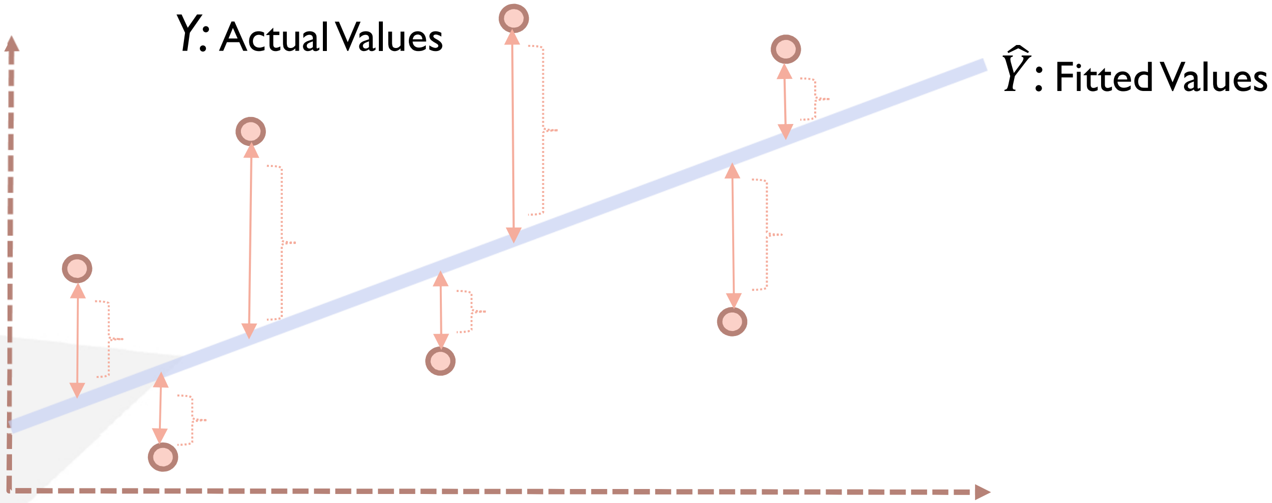
β = 1.4375
intercept = 0.3313
R_squared = 0.2614

GM(GM.US)

β = 1.2683
intercept = -0.0077
R_squared = 0.2561

✓ *R2* (*coefficient of determination*) of the estimation result is calculated using the following formula.

$$R^2 = 1 - \frac{\sum_{i=1}^{N}(Y_i - \hat{Y})^2}{\sum_{i=1}^{N}(Y_i - \bar{Y})^2}$$

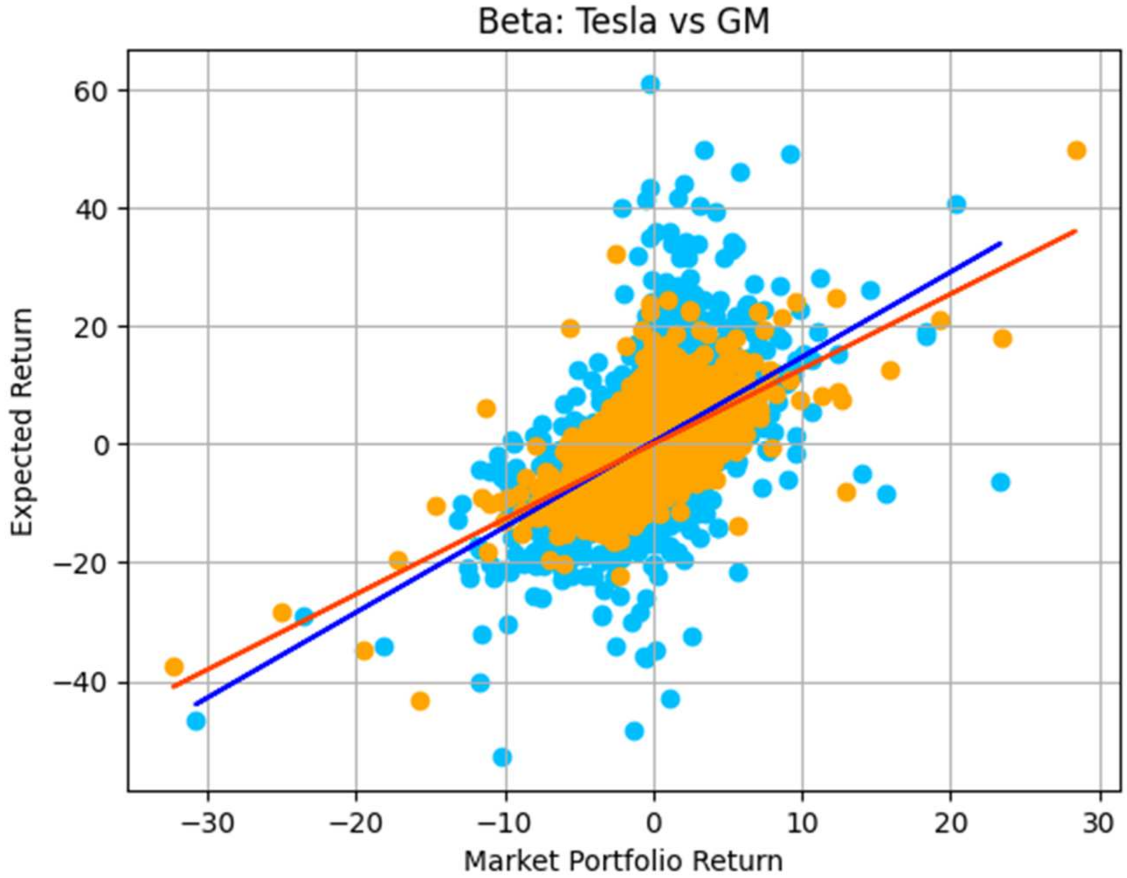


10 Estimating Beta: Visualization

✓ Finally, describe and visualize the scatter diagram and theoretical line.

```
#[6] Visualization

plt.scatter(X1,Y1,color ='deepskyblue')
plt.plot(X1, lr1.predict(X1), color = 'blue')
plt.scatter(X2,Y2, color = 'orange') )
plt.plot(X2, lr2.predict(X2), color = 'orangered')
plt.title('Beta: Tesla vs GM')
plt.xlabel('Market Portfolio Return')
plt.ylabel('Expected Return')
plt.grid()
plt.show()
```


Beta: Tesla vs GM
Expected Return
Market Portfolio Return
60
40
20
0
−20
−40
−30
−20
−10
0
10
20
30

11 Drawing SML: Libraries

✓ In deriving the stock market line (SML) in this example, only matplotlib is used.

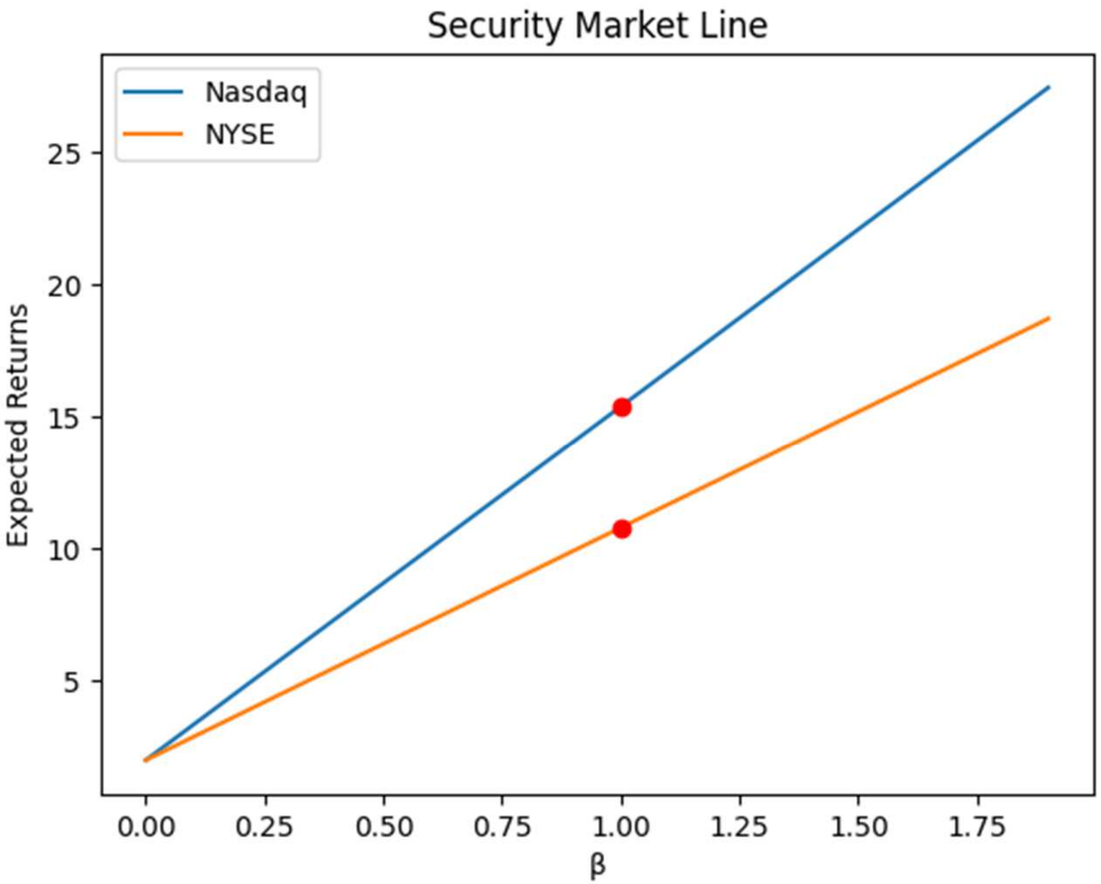
```
import matplotlib.pyplot as plt
```

12 Drawing SML: Libraries

- ✓ Here, the expected return on the 30-year Treasury yield from 2012 to 2022 is 2.0%, and the annual returns for the Nasdaq and New York portfolios are 15.4% and 10.8%, respectively.
- ✓ These values are obtained from the data obtained in the previous section.

```
def SML(rf,rm,label):
    Beta = [x/10 for x in range(20)]
    ExpectedReturns = [rf+(rm-rf)*x for x in Beta]
    plt.plot(Beta,ExpectedReturns,label=label)
    plt.xlabel("β")
        plt.ylabel("Expected Returns")
    plt.title("Security Market Line")
    plt.plot(1,rm,"ro")
```

```
SML(2.0,15.4,"Nasdaq")
SML(2.0,10.8,"NYSE")
plt.legend()
plt.show()
```


Security Market Line
Nasdaq
NYSE
Expected Returns
β
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
5
10
15
20
25

13 Libraries in this Session

pandas_dataframe
datetime

Pandas	.pct_change()	.dropna()	.value()
NumPy	.mean()	.std()	.var()
matplotlib	.plt.scatter()	https://matplotlib.org/	
sklearn	.LinearRegression()	https://scikit-learn.org/stable/	

Contact：

Mamoru Nagano
Professor of Finance,
Seikei University
mnagano@econ.seikei.ac.jp