

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KINH TẾ - LUẬT



BÀI THI CUỐI KỲ:

Gói phần mềm ứng dụng trong Tài chính 1

Đề tài:

ĐỊNH GIÁ TÀI SẢN VỚI CAMP VÀ PHÂN TÍCH RỦI RO

CÔNG PHƯƠNG

Giảng viên: Ngô Phú Thanh

Sinh viên: Lương Thị Mỹ Tâm

MSSV: K204141929

Lớp: K20414C

TP Hồ Chí Minh tháng 1 năm 2023

MỤC LỤC

MỤC LỤC	I
Danh sách hình	II
I. ĐỊNH GIÁ TÀI SẢN VỚI CAMP	1
1. Các khái niệm	1
2. Ứng dụng Python vào mô hình định giá tài sản vốn (CAMP)	3
II. PHÂN TÍCH RỦI RO CỔ PHIẾU	12
III. TÀI LIỆU THAM KHẢO	16
IV. PHẦN CODE	16

Danh sách hình

	TÊN HÌNH	trang
1	Hình 1: Dữ liệu thị trường SPY	4
2	Hình 2: Dữ liệu của cổ phiếu Amazon	5
3	Hình 3: Thống kê mô tả của dữ liệu chứng khoán thị trường SPY	6
4	Hình 4: Thống kê mô tả của dữ liệu cổ phiếu Amazon	7
5	Hình 5: Biểu đồ thể hiện mối quan hệ giữa Amazon và SPY	8
6	Hình 6: Biểu đồ biểu diễn lợi nhuận tích lũy của Amazon và SPY	9
7	Hình 7: Biểu đồ tỷ suất lợi nhuận được tính theo ngày	9

8	Hình 8: Mô hình hồi quy tuyến tính	10
9	Hình 9: Kết quả của beta và alpha	11
10	Hình 10: Kết quả mô hình định giá tài sản vốn (CAMP)	11
11	Hình 11: Dữ liệu của các mã cổ phiếu	12
12	Hình 12: Missing Value	13
13	Hình 13: Biểu đồ giá đóng cửa của các mã cổ phiếu	14
14	Hình 14: Giá trị trung bình và độ lệch chuẩn của giá	14
15	Hình 15: Biểu đồ rủi ro của các mã cổ phiếu	15

I. ĐỊNH GIÁ TÀI SẢN VỚI CAMP

1. Các khái niệm:

a) Mô hình định giá tài sản vốn (CAPM)

Mô hình định giá tài sản vốn (Capital asset pricing model- CAPM) mô tả mối quan hệ giữa lợi tức kỳ vọng của tài sản và rủi ro hệ thống của thị trường đối với tài sản. Đặc biệt là cổ phiếu.

CAPM chỉ ra rằng lợi nhuận kỳ vọng của một tài sản bằng lãi suất phi rủi ro cộng với phần bù rủi ro. Giả định của CAPM là các nhà đầu tư hợp lý và muốn tối đa hóa lợi nhuận và giảm thiểu rủi ro càng nhiều càng tốt. Do đó, mục tiêu của CAPM là tính toán lợi nhuận mà một nhà đầu tư có thể mong đợi để kiếm được một phần bù rủi ro nhất định so với lãi suất phi rủi ro.

b) Lợi tức kỳ vọng (R_e)

Lợi tức kỳ vọng là kỳ vọng của riêng nhà đầu tư hoặc dựa trên dữ liệu lịch sử của các chỉ số cụ thể, ví dụ như S&P500.

c) Lãi suất phi rủi ro (R_f)

Đây là loại lãi suất mà rủi ro gần như bằng 0. Tại các nước Châu Âu, chứng khoán kho bạc Hoa Kỳ được sử dụng để đại diện cho mức lãi suất phi rủi ro. Ngoài ra, lãi suất phi rủi ro cũng thường được lấy dựa trên lãi suất trái phiếu kho bạc chính phủ 10 năm.

d) Lợi tức thị trường (R_m)

Lợi tức thị trường bao gồm tất cả các chứng khoán trên thị trường. Một đại diện tốt cho danh mục đầu tư của thị trường Hoa Kỳ là S&P 500, đây là chỉ số tính theo giá trị vốn hóa thị trường của 500 công ty giao dịch công khai lớn nhất của Hoa Kỳ.

e) *Beta*

Beta là thước đo mức độ biến động của cổ phiếu so với thị trường chung, chẳng hạn như S&P 500. Nói cách khác, Beta đại diện cho độ dốc của đường hồi quy, là lợi nhuận thị trường so với lợi nhuận của từng cổ phiếu.

Beta được sử dụng trong CAPM để mô tả mối quan hệ giữa rủi ro hệ thống hoặc rủi ro thị trường và lợi tức kỳ vọng của một tài sản. Theo định nghĩa, chúng ta nói rằng thị trường tổng thể có hệ số beta là 1,0 và các cổ phiếu riêng lẻ được xếp hạng theo mức độ biến động của chúng so với thị trường.

Hệ số Beta = 0: có nghĩa là mức biến động giá của cổ phiếu hoàn toàn độc lập so với mức biến động của thị trường.

Nếu Beta của một cổ phiếu riêng lẻ = 1,0, điều này có nghĩa là giá của nó hoàn toàn tương quan với thị trường, rủi ro bằng mức trung bình của thị trường. Ví dụ khi chỉ số VNI tăng 1% thì cổ phiếu sẽ tăng 1%.

Nếu Beta < 1.0, được gọi là 'phòng thủ', điều này cho thấy về mặt lý thuyết, chứng khoán ít biến động hơn so với thị trường, rủi ro thấp hơn mức trung bình của thị trường. Ví dụ Beta bằng = 0.5 khi chỉ số VNI tăng 1% thì cổ phiếu sẽ tăng 0.5%.

Nếu Beta > 1.0 hoặc 'tích cực', điều này cho thấy giá tài sản dễ biến động hơn thị trường, rủi ro cao hơn mức trung bình của thị trường. Ví dụ Beta bằng = 1.2 khi chỉ số VNI tăng 1% thì cổ phiếu sẽ tăng 1.2%.

f) *Công thức CAPM*

$$Re = Rf + [Beta \times (Rm - Rf)]$$

trong đó:

Re: Lợi tức kỳ vọng

Rf: Lãi suất phi rủi ro

Rm: Lợi tức thị trường

Beta: beta của cổ phiếu so với thị trường

2. Ứng dụng Python vào mô hình định giá tài sản vốn (CAMP)

a) Nhập dữ liệu

Sau khi cài các thư viện cần thiết thì ta sẽ tiến hành tải dữ liệu chứng khoán thông qua Yahoo Finance API.

Yahoo Finance API là nguồn dữ liệu thị trường chứng khoán đáng tin cậy, có thể được gọi là một loạt thư viện, API hoặc phương pháp vì nó cung cấp thông tin tài chính khác, bao gồm tóm tắt thị trường, báo cáo tài chính, báo giá lịch sử và nguồn cập tin tức.

Đầu tiên ta sẽ lấy dữ liệu của một thị trường chẳng hạn như 'SPY' (SPDR S&P 500) là một tập hợp các cổ phiếu, mô phỏng lại chỉ số S&P 500, sau đó ta sẽ lấy dữ liệu của một cổ phiếu cụ thể ở đây ta sẽ dùng dữ liệu của cổ phiếu Amazon với mã chứng khoán là AMZN. Dữ liệu sử dụng sẽ có phạm vi là 5 năm từ năm 2018 đến năm 2022 và chúng ta hoàn toàn có thể điều chỉnh để lựa chọn các khoảng thời gian khác một cách linh hoạt.

```
In [3]: data = yf.download('SPY', '2017-12-31', '2022-12-31')
```

```
[*****100%*****] 1 of 1 completed
```

```
In [4]: data
```

```
Out[4]:
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-01-02	267.839996	268.809998	267.399994	268.769989	246.766602	86655700
2018-01-03	268.959991	270.640015	268.959991	270.470001	248.327438	90070400
2018-01-04	271.200012	272.160004	270.540009	271.609985	249.374176	80636400
2018-01-05	272.510010	273.559998	271.950012	273.420013	251.035995	83524000
2018-01-08	273.309998	274.100006	272.980011	273.920013	251.495071	57319200
...
2022-12-23	379.649994	383.059998	378.029999	382.910004	382.910004	59857300
2022-12-27	382.790009	383.149994	379.649994	381.399994	381.399994	51638200
2022-12-28	381.329987	383.390015	376.420013	376.660004	376.660004	70911500
2022-12-29	379.630005	384.350006	379.079987	383.440002	383.440002	66970900
2022-12-30	380.640015	382.579987	378.429993	382.429993	382.429993	83975100

Hình 1: Dữ liệu thị trường SPY


```
In [8]: df = yf.download('AMZN', '2017-12-31', '2022-12-31')
```

```
[*****100%*****] 1 of 1 completed
```

```
In [9]: df
```

```
Out[9]:
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2018-01-02	58.599998	59.500000	58.525501	59.450500	59.450500	53890000
2018-01-03	59.415001	60.274502	59.415001	60.209999	60.209999	62176000
2018-01-04	60.250000	60.793499	60.233002	60.479500	60.479500	60442000
2018-01-05	60.875500	61.457001	60.500000	61.457001	61.457001	70894000
2018-01-08	61.799999	62.653999	61.601501	62.343498	62.343498	85590000
...
2022-12-23	83.250000	85.779999	82.930000	85.250000	85.250000	57433700
2022-12-27	84.970001	85.349998	83.000000	83.040001	83.040001	57284000
2022-12-28	82.800003	83.480003	81.690002	81.820000	81.820000	58228600
2022-12-29	82.870003	84.550003	82.550003	84.180000	84.180000	54995900
2022-12-30	83.120003	84.050003	82.470001	84.000000	84.000000	62330000

```
1259 rows × 6 columns
```

Hình 2: Dữ liệu của cổ phiếu Amazon

b) Xử lý dữ liệu

Bước tiếp theo là chúng ta sẽ đi kiểm tra xem hai tập dữ liệu có missing value, duplicate hay có vấn đề gì không thông qua việc xem các thông tin về các cột dữ liệu, sử dụng thống kê mô tả để từ đó đưa ra giải pháp xử lý để làm sạch dữ liệu. Bởi vì dữ liệu đầu vào phải chuẩn thì kết quả mới chính xác được.

In [5]: `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1259 entries, 2018-01-02 to 2022-12-30
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
---  --
 0   Open        1259 non-null   float64
 1   High        1259 non-null   float64
 2   Low         1259 non-null   float64
 3   Close       1259 non-null   float64
 4   Adj Close   1259 non-null   float64
 5   Volume      1259 non-null   int64   
dtypes: float64(5), int64(1)
memory usage: 68.9 KB
```

In [6]: `data.describe()`

Out[6]:

	Open	High	Low	Close	Adj Close	Volume
count	1259.000000	1259.000000	1259.000000	1259.000000	1259.000000	1.259000e+03
mean	344.334512	346.514384	341.919245	344.338904	331.588049	8.722074e+07
std	66.593699	66.880381	66.227147	66.595683	71.189159	4.583611e+07
min	228.190002	229.679993	218.259995	222.949997	213.785477	2.027000e+07
25%	283.784988	285.600006	282.314987	283.980011	266.306137	5.775315e+07
50%	326.910004	328.619995	324.500000	326.859985	313.021698	7.581760e+07
75%	403.479996	406.869995	399.190002	402.525009	397.886475	1.023069e+08
max	479.220001	479.980011	476.059998	477.709991	470.083679	3.922207e+08

Overview

Overview		Alerts 6	Reproduction
Dataset statistics		Variable types	
Number of variables	6	Numeric	6
Number of observations	1259		
Missing cells	0		
Missing cells (%)	0.0%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	68.9 KiB		
Average record size in memory	56.0 B		

Hình 3: Thống kê mô tả của dữ liệu chứng khoán thị trường SPY

In [10]: `df.info()`

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1259 entries, 2018-01-02 to 2022-12-30
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype
---  -
 0   Open        1259 non-null   float64
 1   High        1259 non-null   float64
 2   Low         1259 non-null   float64
 3   Close       1259 non-null   float64
 4   Adj Close   1259 non-null   float64
 5   Volume      1259 non-null   int64
dtypes: float64(5), int64(1)
memory usage: 68.9 KB
```

In [11]: `df.describe()`

Out[11]:

	Open	High	Low	Close	Adj Close	Volume
count	1259.000000	1259.000000	1259.000000	1259.000000	1259.000000	1.259000e+03
mean	119.895163	121.332617	118.284385	119.812384	119.812384	8.655625e+07
std	35.655449	36.039141	35.226090	35.592042	35.592042	4.068152e+07
min	58.599998	59.500000	58.525501	59.450500	59.450500	1.762600e+07
25%	89.062252	89.779003	87.965252	88.930500	88.930500	5.893800e+07
50%	107.970001	110.155998	106.294502	107.783501	107.783501	7.499600e+07
75%	158.343750	159.924004	156.199997	158.086998	158.086998	1.025480e+08
max	187.199997	188.654007	184.839493	186.570496	186.570496	3.113460e+08

Overview

Overview

Alerts **5**

Reproduction

Dataset statistics

Number of variables	6
Number of observations	1259
Missing cells	0
Missing cells (%)	0.0%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	68.9 KiB
Average record size in memory	56.0 B

Variable types

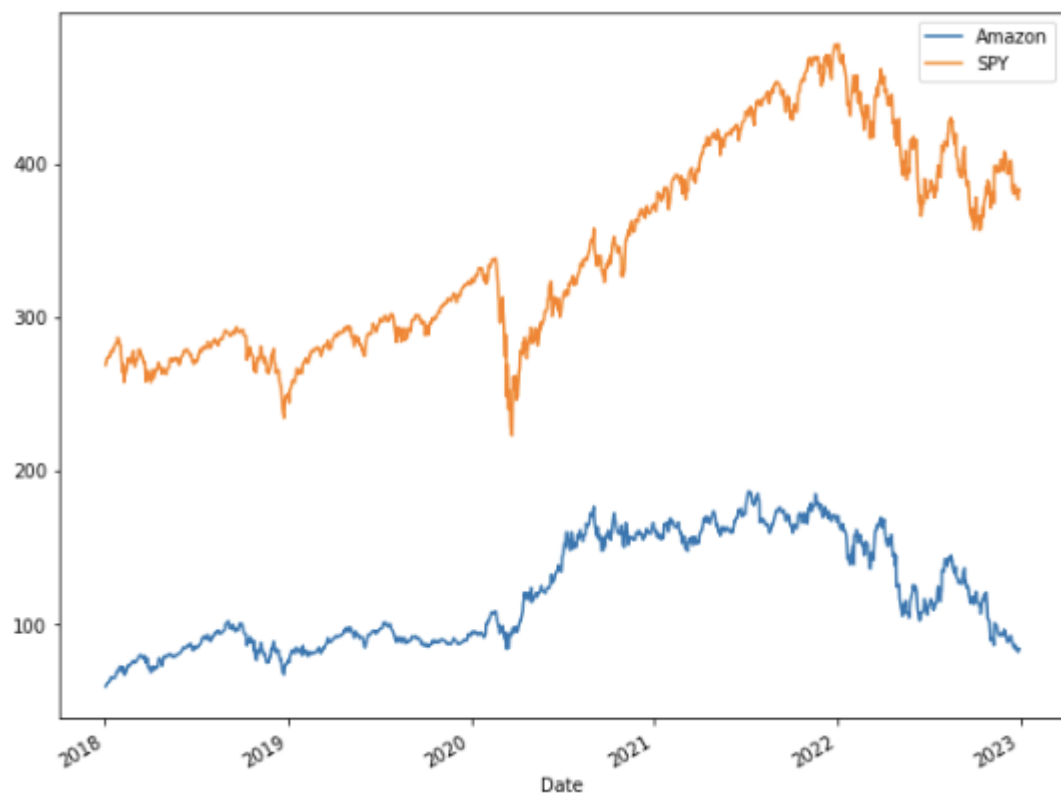
Numeric	6
---------	---

Hình 4: Thống kê mô tả của dữ liệu cổ phiếu Amazon

Từ bảng 3 và bảng 4, chúng ta thấy rằng hai tập dữ liệu đều có 6 cột và 1259 dòng, không có chứa missing value và duplicate. Do đó, hai tập dữ liệu đã sạch không cần phải xử lý gì thêm.

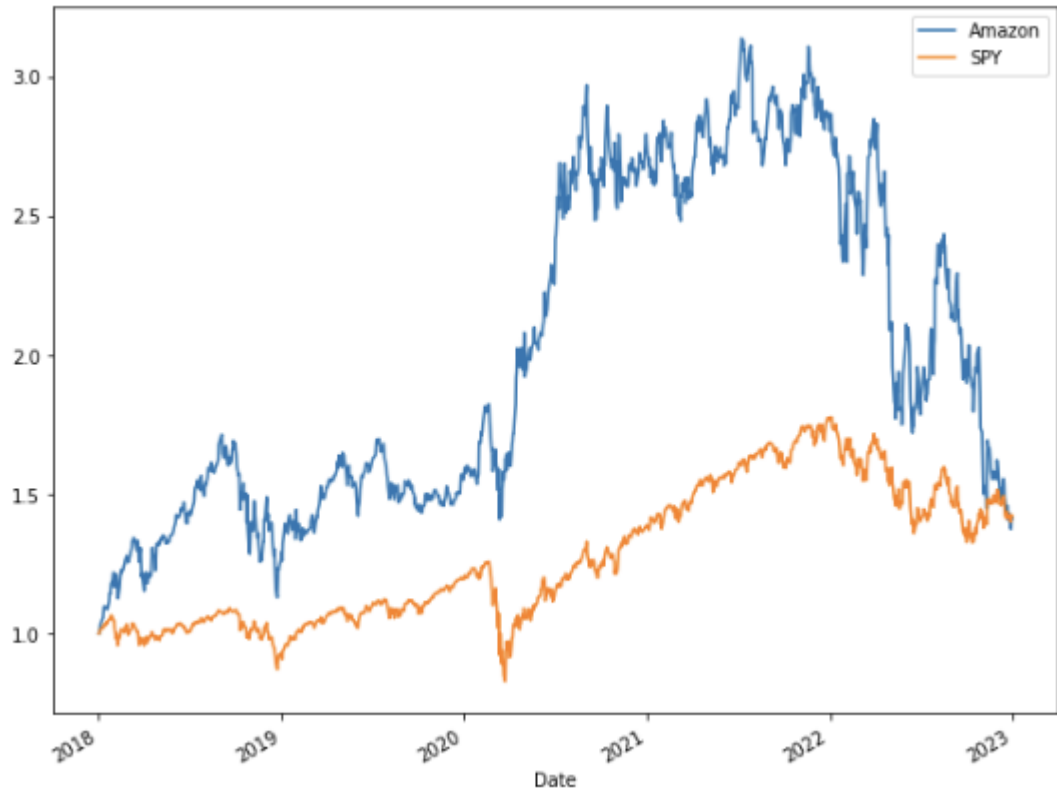
c) Trực quan hóa dữ liệu

Tiếp theo, ta sẽ trực quan hóa dữ liệu để tìm mối quan hệ bởi vì theo CAPM thì cần có một số mối quan hệ giữa hoạt động của cổ phiếu và hoạt động của thị trường sẽ được xem xét trong tương lai.



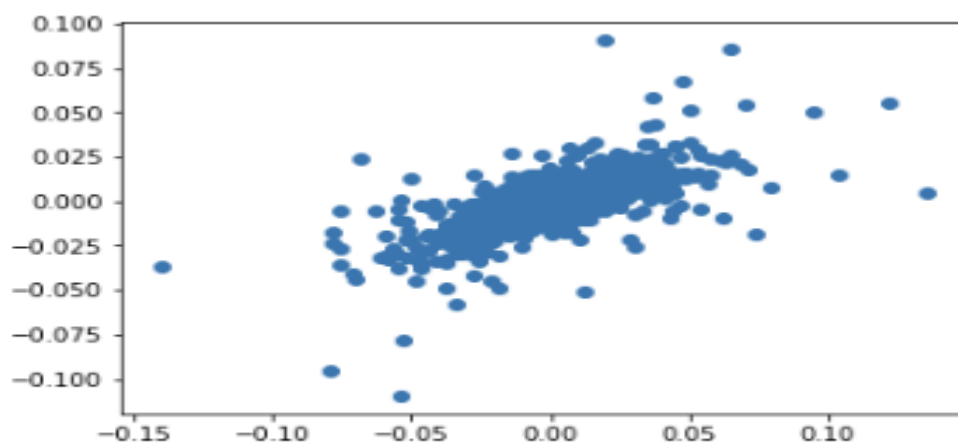
Hình 5: Biểu đồ thể hiện mối quan hệ giữa Amazon và SPY

Từ hình 5, ta thấy được rằng giá của Amazon đang thấp hơn chỉ số thị trường SPY và giá của nó biến động tương tự theo thị trường và đang có xu hướng giảm. Vì vậy, ta có thể so sánh được cổ phiếu Amazon với SPY với nhau. Và lợi nhuận tích lũy được thể hiện dưới hình 6.



Hình 6: Biểu đồ biểu diễn lợi nhuận tích lũy của Amazon và SPY

Và Hiệu quả đầu tư/tỷ suất lợi nhuận được tính hàng ngày cũng được xác định, ta có thể thấy ở hình 7 dưới đây



Hình 7: Biểu đồ tỷ suất lợi nhuận được tính theo ngày

Từ biểu đồ phân tán chỉ ra rằng có một số mối quan hệ giữa lợi nhuận hàng ngày của cổ phiếu và thị trường. Sau đây chúng ta sẽ đi vào xây dựng mô hình CAMP.

d) Xây dựng mô hình CAMP

Đầu tiên chúng ta sẽ đi tìm 2 giá trị là beta và alpha. Các giá trị alpha và beta này sẽ được tìm thấy bằng cách sử dụng gói thống kê của scipy và gọi hàm hồi quy tuyến tính của nó. Trong lúc tìm lợi nhuận hàng ngày, thì chúng ta phát hiện ra rằng hàng đầu tiên có các giá trị NaN và do đó, trong khi chuyển các cột cho việc tính toán hồi quy tuyến tính, mọi thứ từ hàng đầu tiên đều phải được xem xét đến vì vậy chúng ta sẽ dùng iloc để lựa chọn hàng cho đúng.

```
In [18]: LR = stats.linregress(df['daily_ret'].iloc[1:],data['daily_ret'].iloc[1:])
In [19]: LR
Out[19]: LinregressResult(slope=0.40677746068570275, intercept=0.0001584122910723742, rvalue=0.6734645647606573, pvalue=5.0471169390260054e-167, stderr=0.012598554833333424, intercept_stderr=0.00028410849632564626)
```

Hình 8: Mô hình hồi quy tuyến tính

Như chúng ta thấy từ hình 8, mô hình hồi quy tuyến tính đã được xây dựng. Nó có 6 giá trị có thể thu được từ việc giải nén bộ dữ liệu. Sáu giá trị bao gồm beta, alpha, rvalue, pvalue, sai số chuẩn và intercept standard deviation. Vì mô hình đã được xây dựng nên ta dễ dàng tìm được giá trị của beta và alpha. Giá trị của beta và alpha lần lượt là 1.11499 và 0.000112.

Lưu ý rằng như mô hình CAPM đã nói rằng alpha mong đợi là gần bằng 0 và ta có thể thấy giá trị alpha khi chạy mô hình cũng là một con số gần bằng 0. Và bên cạnh đó thì nếu giá trị beta cao thì cổ phiếu biến động nhiều hơn thị trường. Do đó, giá trị beta ở đây là $1.11499 > 1.0$ vì vậy ta có thể kết luận rằng, cổ phiếu Amazon biến động nhiều hơn thị trường, rủi ro cao hơn mức trung bình của thị trường.

```

In [20]: beta,alpha,r_val,p_val,std_err = LR
In [21]: beta
Out[21]: 1.1149942261395382
In [22]: alpha
Out[22]: 0.00011248461545313441

```

Hình 9: Kết quả của beta và alpha

Bây giờ chúng ta đã tính được lợi nhuận hàng ngày và hệ số beta của cổ phiếu Amazon do đó chúng ta có thể tính toán mô hình định giá tài sản vốn (CAMP). Đầu tiên, chúng ta tính lợi nhuận trung bình hàng ngày của thị trường. Sau đó, chúng ta tính lợi tức này hàng năm bằng cách nhân nó với số ngày giao dịch trong một năm là 252 ngày. Giả sử lãi suất phi rủi ro là 0, thì chúng ta có thể tính CAMP cho AMZN bằng công thức đã nêu ở phần khái niệm phía trên. Và kết quả CAMP của Amazon là bằng 0.10498 như ta có thể theo dõi hình 10 dưới đây.

```

In [23]: data['daily_ret'].mean()
Out[23]: 0.00037363013288758864
In [24]: rm = data['daily_ret'].mean() * 252
          rm
Out[24]: 0.09415479348767233
In [25]: rf = 0
          CAMP_AMZN = rf + (beta * (rm-rf))
In [26]: CAMP_AMZN
Out[26]: 0.10498205110211525

```

Hình 10: Kết quả mô hình định giá tài sản vốn (CAMP)

Tóm lại, kết quả thể hiện rằng một nhà đầu tư nên kỳ vọng tỷ suất sinh lợi của cổ phiếu Amazon là 10,49% để bù đắp cho rủi ro bổ sung mà họ đang chấp

nhận.

II. PHÂN TÍCH RỦI RO CỔ PHIẾU

a) Nhập dữ liệu

Tương tự như mô hình CAMP, chúng ta sẽ lấy dữ liệu của các mã cổ phiếu từ Yahoo Finance. Chúng ta có thể lấy bao nhiêu mã tùy ý nhưng ở đây ta sẽ lấy 7 mã cổ phiếu bao gồm cổ phiếu của Amazon, SPY, Microsoft, Apple, Facebook, Nvidia và Adobe. Dữ liệu cũng được lấy trong 5 năm từ 2018 đến 2022 để dễ so sánh.

In [27]: `stocks = yf.download(['AMZN', 'ADBE', 'MSFT', 'NVDA', 'AAPL', 'META', 'SPY'], '2017-12-31', '2023-01-01')`
 [*****100%*****] 7 of 7 completed

In [28]: `stocks`

Out [28]:

	Adj Close							Close			...	Open		
	AAPL	ADBE	AMZN	META	MSFT	NVDA	SPY	AAPL	ADBE	AMZN	...	MSFT	NVDA	SPY
Date														
2018-01-02	40.950500	177.699997	59.450500	181.419998	80.940399	49.334675	246.766647	43.064999	177.699997	59.450500	...	86.129997	48.945000	267.8
2018-01-03	40.943375	181.039993	60.209999	184.669998	81.317062	52.581573	248.327530	43.057499	181.039993	60.209999	...	86.059998	51.025002	268.9
2018-01-04	41.133541	183.220001	60.479500	184.330002	82.032776	52.858757	249.374084	43.257500	183.220001	60.479500	...	86.589996	53.939999	271.2
2018-01-05	41.601871	185.339996	61.457001	186.850006	83.049820	53.306683	251.035934	43.750000	185.339996	61.457001	...	87.660004	53.547501	272.5
2018-01-08	41.447342	185.039993	62.343498	188.279999	83.134590	54.940041	251.495026	43.587502	185.039993	62.343498	...	88.199997	55.099998	273.3
...
2022-12-23	131.860001	338.450012	85.250000	118.040001	238.729996	152.059998	382.910004	131.860001	338.450012	85.250000	...	236.110001	151.960007	379.6
2022-12-27	130.029999	335.089996	83.040001	116.879997	236.960007	141.210007	381.399994	130.029999	335.089996	83.040001	...	238.699997	150.740005	382.7
2022-12-28	126.040001	328.329987	81.820000	115.620003	234.529999	140.360001	376.660004	126.040001	328.329987	81.820000	...	236.889999	139.270004	381.3
2022-12-29	129.610001	337.579987	84.180000	120.260002	241.009995	146.029999	383.440002	129.610001	337.579987	84.180000	...	235.649994	144.020004	379.6
2022-12-30	129.929993	336.529999	84.000000	120.339996	239.820007	146.139999	382.429993	129.929993	336.529999	84.000000	...	238.210007	143.339996	380.6

1259 rows x 42 columns

Hình 11: Dữ liệu của các mã cổ phiếu

b) Xử lý dữ liệu:

Ở bước này, chúng ta sẽ tiếp tục kiểm tra missing value, duplicate và các lỗi khác có thể ảnh hưởng đến quá trình sử dụng dữ liệu.


```
In [31]: stocks.isnull().sum()
```

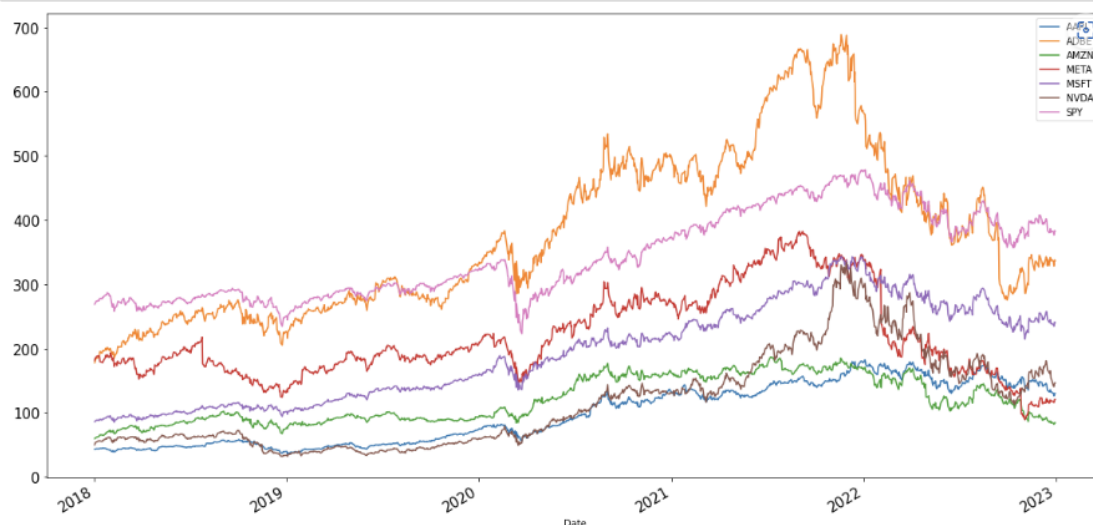
```
Out[31]: Adj Close AAPL 0
          ADBE 0
          AMZN 0
          META 0
          MSFT 0
          NVDA 0
          SPY 0
          Close AAPL 0
                ADBE 0
                AMZN 0
                META 0
                MSFT 0
                NVDA 0
                SPY 0
          High AAPL 0
                ADBE 0
                AMZN 0
                META 0
                MSFT 0
                NVDA 0
                SPY 0
          Low AAPL 0
              ADBE 0
              AMZN 0
              META 0
              MSFT 0
              NVDA 0
              SPY 0
          Open AAPL 0
               ADBE 0
               AMZN 0
               META 0
               MSFT 0
               NVDA 0
               SPY 0
          Volume AAPL 0
                 ADBE 0
                 AMZN 0
                 META 0
                 MSFT 0
                 NVDA 0
                 SPY 0
          dtype: int64
```

Hình 12: Missing Value

Từ hình 12 ta thấy dữ liệu đã được làm sạch và sẵn sàng được đem vào sử dụng. Tiếp theo, chúng ta sẽ thực hiện phân tích của mình dựa trên giá đóng cửa của các cổ phiếu. Vì vậy, hãy chỉ lấy dữ liệu của cột Close và vẽ biểu đồ giá đóng cửa của các cổ phiếu để có ý tưởng về vị trí của từng cổ phiếu này và điều này có thể được thực hiện ở hình dưới:

```
In [32]: stocksData = stocks.loc[:, 'Close'].copy()
```

```
In [33]: stocksData.plot(figsize= (20,10), fontsize=15)
pyplot.style.use('classic')
pyplot.show()
```



Hình 13: Biểu đồ giá đóng cửa của các mã cổ phiếu

Từ hình 13, chúng ta có thể thấy rằng mã cổ phiếu ADBE và SPY là hai mã cổ phiếu có giá cao hơn so với các mã còn lại. Tiếp theo, chúng ta sẽ kiểm tra xem giá đã thay đổi bao nhiêu vào mỗi ngày và sẽ xóa tất cả các hàng có giá trị bị thiếu. Tại đây, chúng ta sẽ nhận được các giá trị còn thiếu cho những ngày không giao dịch và chúng ta sẽ tính độ lệch và giá trị trung bình cho mỗi ngày này. Điều này chúng ta thực hiện bằng cách sử dụng dòng code trong hình 14 dưới đây:

```
In [34]: data = stocksData.pct_change().dropna()
```

```
In [35]: data = data.describe().T.loc[:, ['mean', 'std']]
```

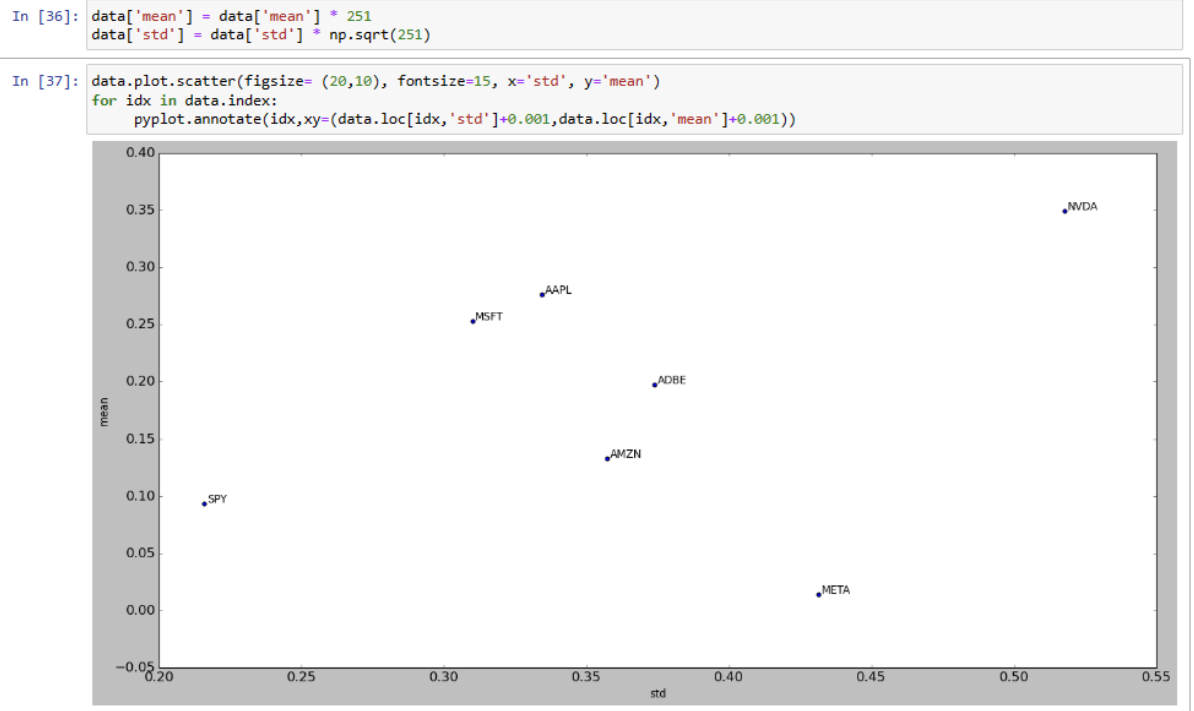
```
In [38]: data
```

Out[38]:

	mean	std
AAPL	0.276298	0.334285
ADBE	0.197726	0.373889
AMZN	0.132799	0.357317
META	0.014161	0.431449
MSFT	0.252861	0.309946
NVDA	0.349616	0.517877
SPY	0.093781	0.215822

Hình 14: Giá trị trung bình và độ lệch chuẩn của giá

Bảng trên hiển thị dữ liệu cho một ngày. Vì vậy, chúng ta cần tính toán nó cho cả năm, tương đương với khoảng 251 ngày giao dịch và sau đó vẽ giá trị trung bình và độ lệch chuẩn của chúng.



Hình 15: Biểu đồ rủi ro của các mã cổ phiếu

Nhận xét:

Nhìn vào hình 14 và hình 15, chúng ta có thể thấy rằng mã NVDA mặc dù giá không phải là cao nhất thậm chí có năm giá của NVDA là thấp nhất so với các mã cổ phiếu khác tuy nhiên NVDA lại là mã cổ phiếu có sự biến động mạnh nhất, rủi ro cao nhất trong số các mã cổ phiếu. Còn mã cổ phiếu có ít sự biến động nhất là SPY. Bên cạnh đó, chúng ta có thể thấy rằng hầu hết các mã cổ phiếu đều có mức độ rủi ro cao hơn so với thị trường. Và kết quả này cũng tương đồng với mô hình CAMP mà chúng ta đã thực hiện ở trên đối với mã AMZN và mã SPY cũng cho rằng mã cổ phiếu của Amazon là rủi ro cao hơn thị trường.

III. TÀI LIỆU THAM KHẢO

<https://towardsai.net/p/l/capital-assets-pricing-model-capm%E2%80%8A-%E2%80%8Ausing-python>

<https://www.c-sharpcorner.com/article/risk-analysis-on-stocks/>

<https://pinetree.vn/post/20220227/y-nghia-va-ung-dung-cua-he-so-beta-chung-khoan/>

IV. PHẦN CODE

```

pip install pandas-profiling[notebook]
import matplotlib.pyplot as pyplot
from scipy import stats
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import yfinance as yf
import pandas_profiling as pp
%matplotlib inline
data = yf.download('SPY','2017-12-31','2023-01-01')
data
data.info()
data.describe()
pp.ProfileReport(data)
df = yf.download('AMZN','2017-12-31','2023-01-01')
df
df.info()
df.describe()
pp.ProfileReport(df)
df['Close'].plot(label = 'Amazon', figsize=(10,8))
data['Close'].plot(label = 'SPY')
plt.legend()

```

```

df['Cumu'] = df['Close']/df['Close'].iloc[0]
data['Cumu'] = data['Close']/data['Close'].iloc[0]
df['Cumu'].plot(label = 'Amazon', figsize=(10,8))
data['Cumu'].plot(label = 'SPY')
plt.legend()
df['daily_ret'] = df['Close'].pct_change(1)
data['daily_ret'] = data['Close'].pct_change(1)
plt.scatter(df['daily_ret'],data['daily_ret'])
LR = stats.linregress(data['daily_ret'].iloc[1:],df['daily_ret'].iloc[1:])
LR
beta,alpha,r_val,p_val,std_err = LR
beta
alpha
data['daily_ret'].mean()
rm = data['daily_ret'].mean() * 252
rm
rf = 0
CAMP_AMZN = rf + (beta * (rm-rf))
CAMP_AMZN
stocks =
yf.download(['AMZN','ADBE','MSFT','NVDA','AAPL','META','SPY','TSLA'],'2017-
12-31','2023-01-01')
stocks
stocks.info()
stocks.describe()
stocks.isnull().sum()
stocksData = stocks.loc[:, 'Close'].copy()
stocksData.plot(figsize= (20,10), fontsize=15)
pyplot.style.use('classic')
pyplot.show()
data = stocksData.pct_change().dropna()

```

```
data = data.describe().T.loc[:,['mean', 'std']]  
data  
data['mean'] = data['mean'] * 251  
data['std'] = data['std'] * np.sqrt(251)  
data.plot.scatter(figsize= (20,10), fontsize=15, x='std', y='mean')  
for idx in data.index:  
    pyplot.annotate(idx,xy=(data.loc[idx,'std']+0.001,data.loc[idx,'mean']+0.001))
```