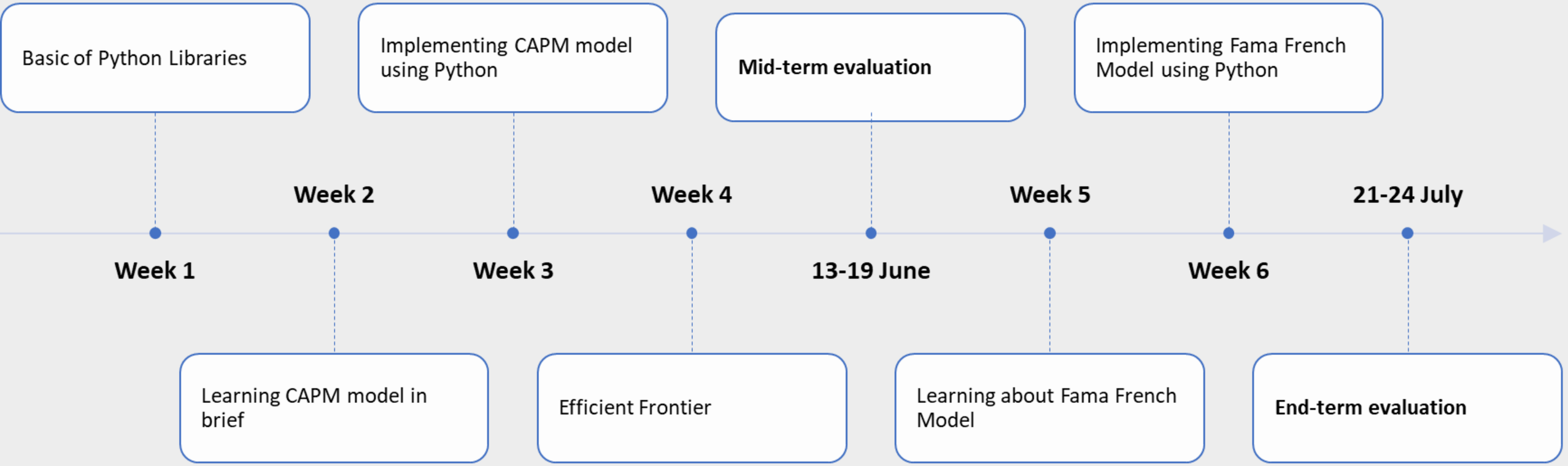
Predicting stock returns using CAPM and FAMA French

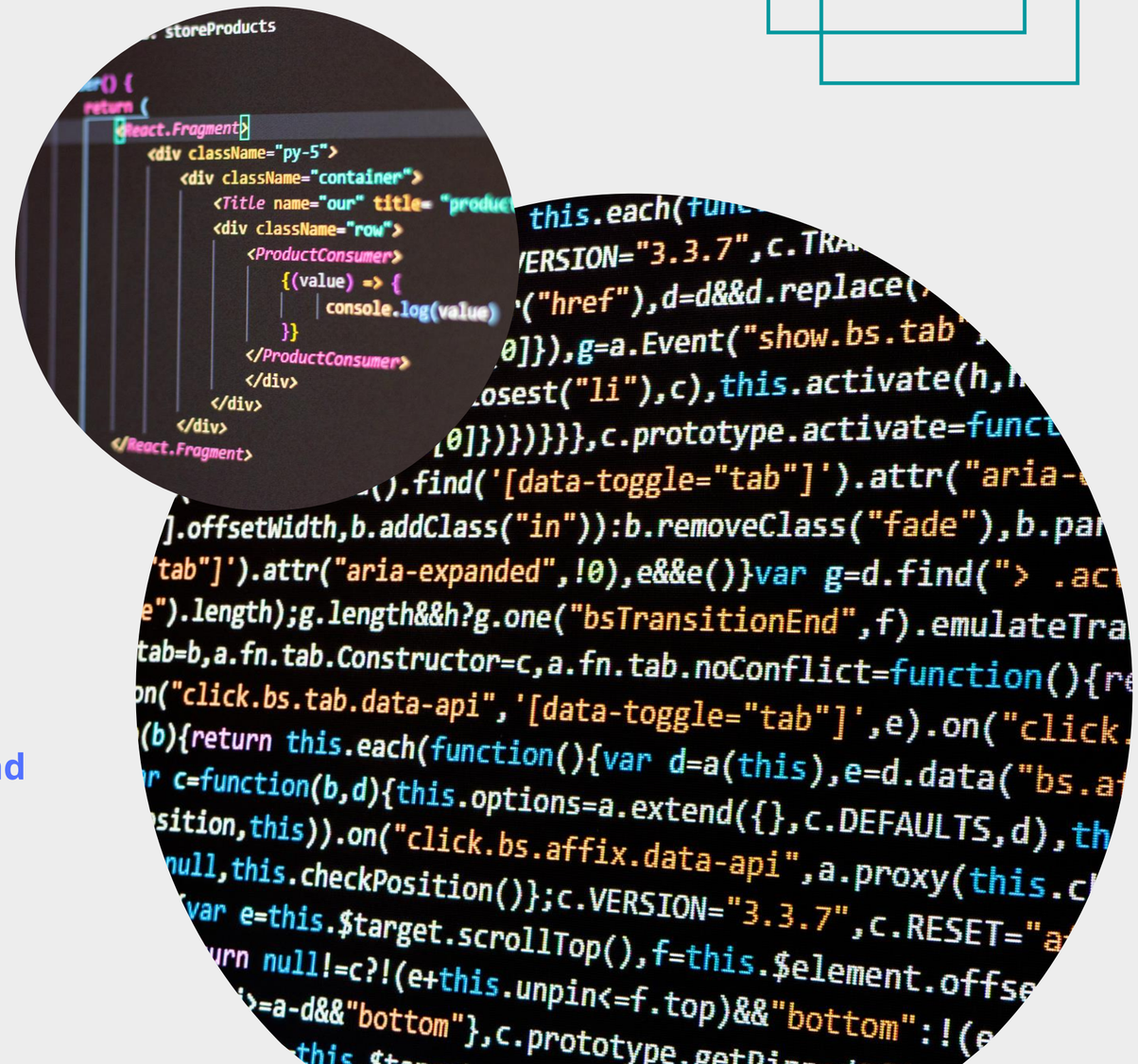
What we have learnt.....



Week 1: Pyhton Libraries

In week 1 we learned about some python libraries which were Matplotlib, Scipy, Pandas, pandas_datareader and datetime.

Week 2: CAPM Model



Our objective was to learn-

Getting Started with the CAPM model.

> What is the CAPM model?

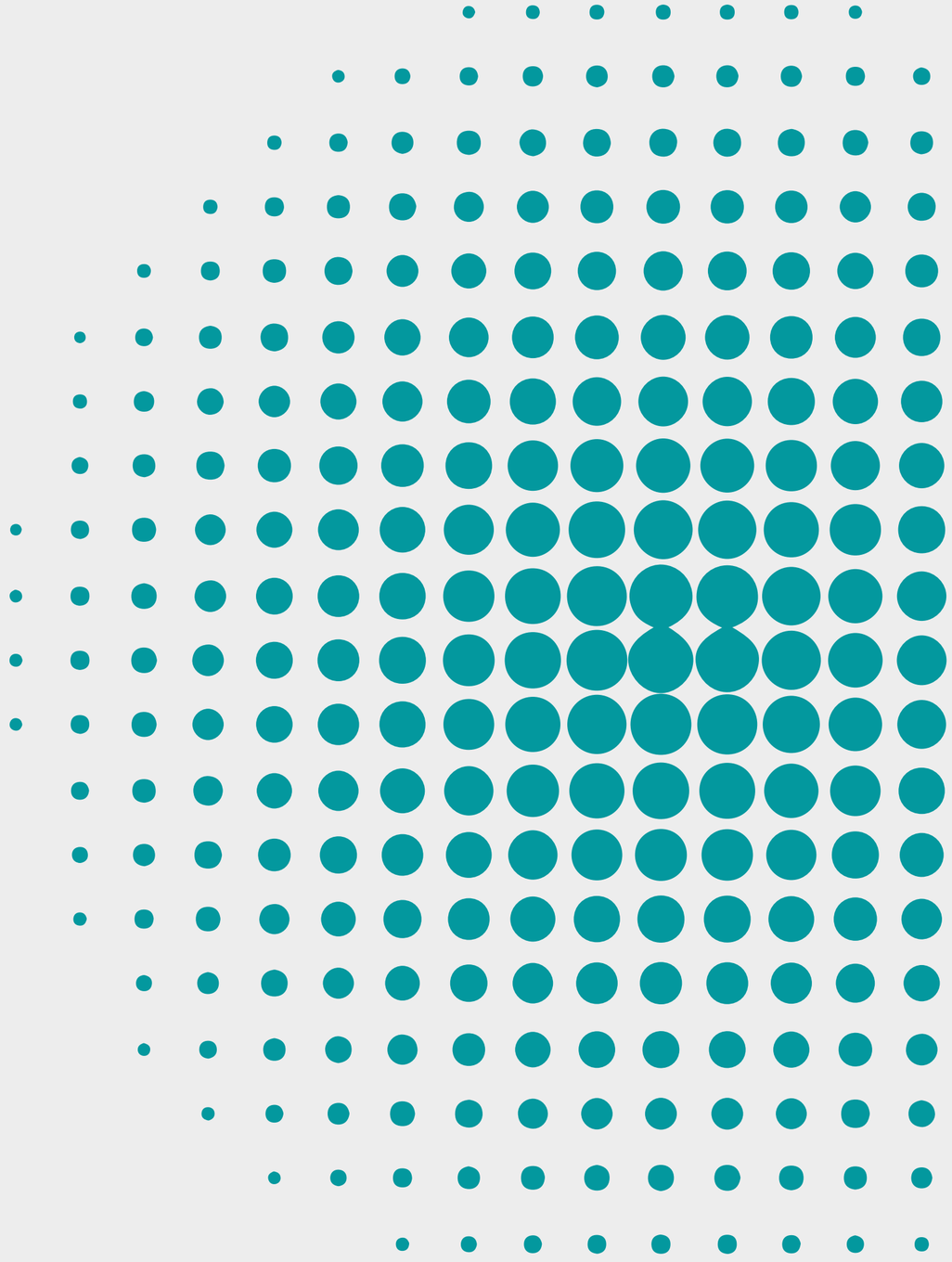
> What is Security Market Line? (SML) Basics Statistics (Linear Regression, R-squared, P-Value)

What is Excess Return? (Terminologies like Sharpe Ratio, alpha and beta) Efficient Frontier

Capital Asset Pricing Model

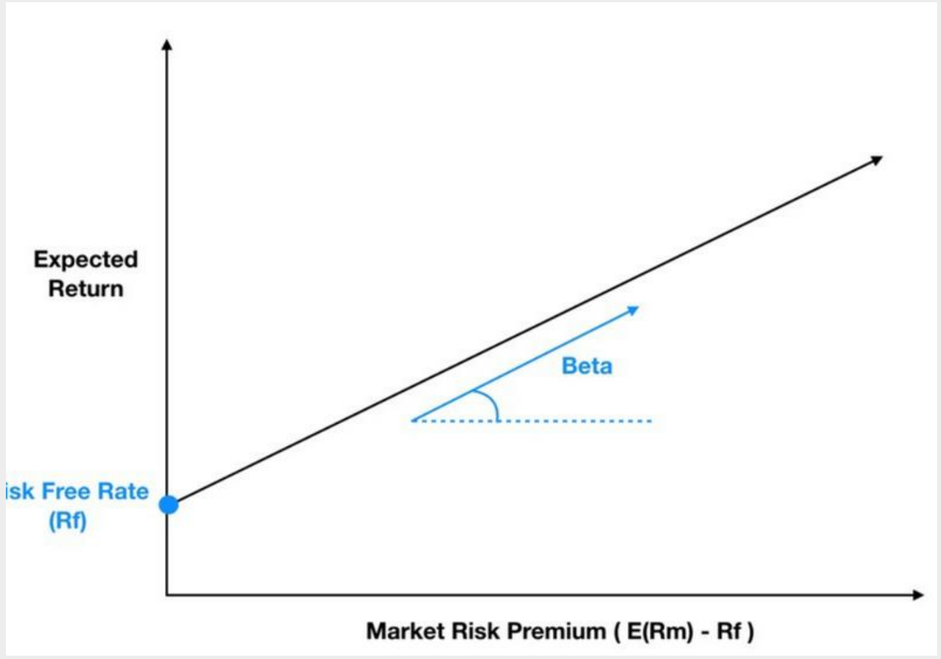
In finance, the capital asset pricing model (CAPM) is a model used to determine a theoretically appropriate required rate of return of an asset, to make decisions about adding assets to a well-diversified portfolio.

The model takes into account the asset's sensitivity to non-diversifiable risk (also known as systematic risk or market risk), often represented by the quantity beta (β) in the financial industry, as well as the expected return of the market and the expected return of a theoretical risk-free asset.



Linear Regression :

Regression is a statistical method used in finance, investing, and other disciplines that attempts to determine the strength and character of the relationship between one dependent variable (usually denoted by Y) and a series of other variables (known as independent variables.

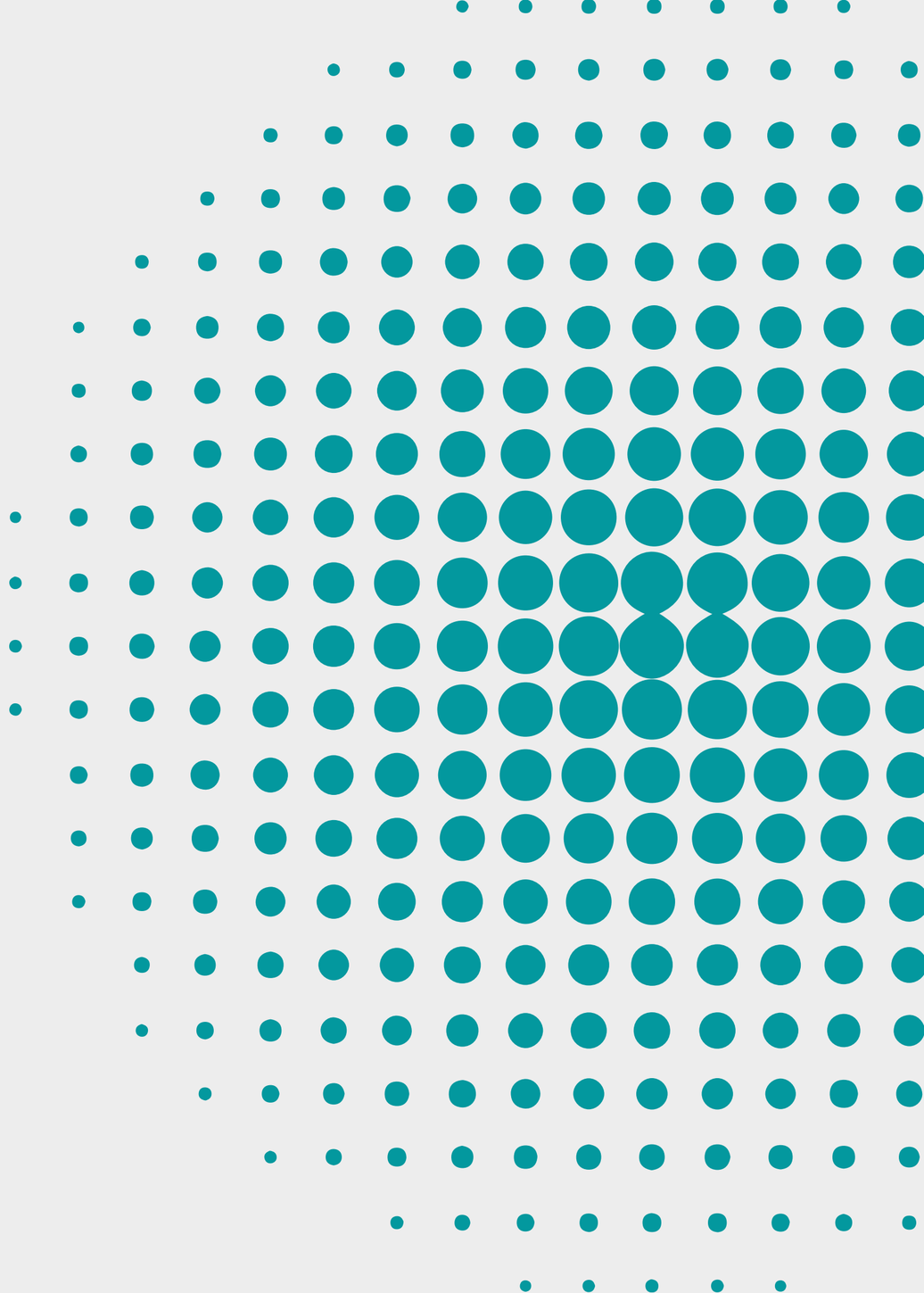


Sharpe Ratio

Sharpe ratio is the measure of risk-adjusted return of a financial portfolio. A portfolio with a higher Sharpe ratio is considered superior relative to its peers.

Ratio: $(R_p - R_f)/\sigma_p$

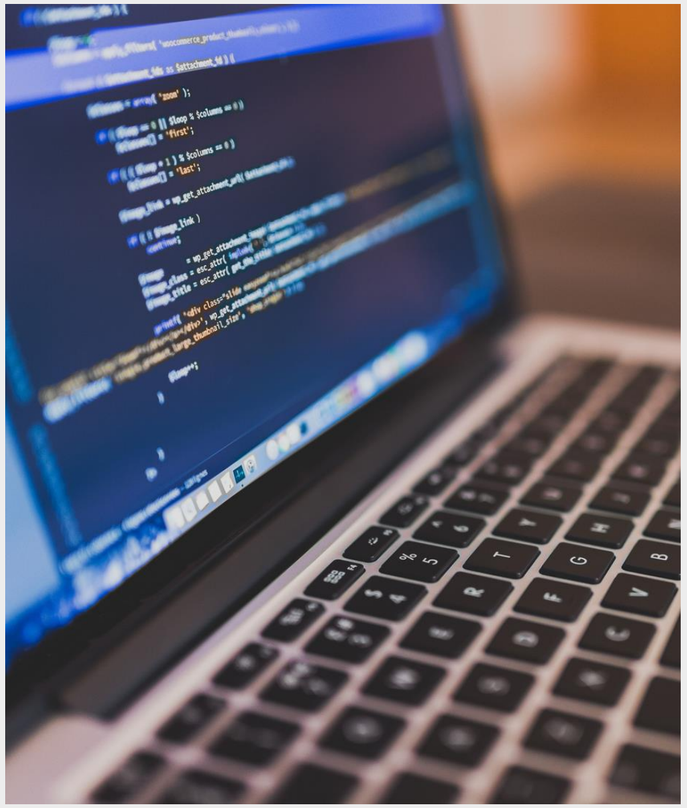
where:R_p=return of portfolio
R_f=risk-free rate
σ_p=standard deviation of the portfolio's excess return



Week 3: Implementing CAPM Model using Python

```
from scipy import stats
import pandas as pd
import pandas_datareader as web
import datetime
import matplotlib.pyplot as plt
%matplotlib inline
```

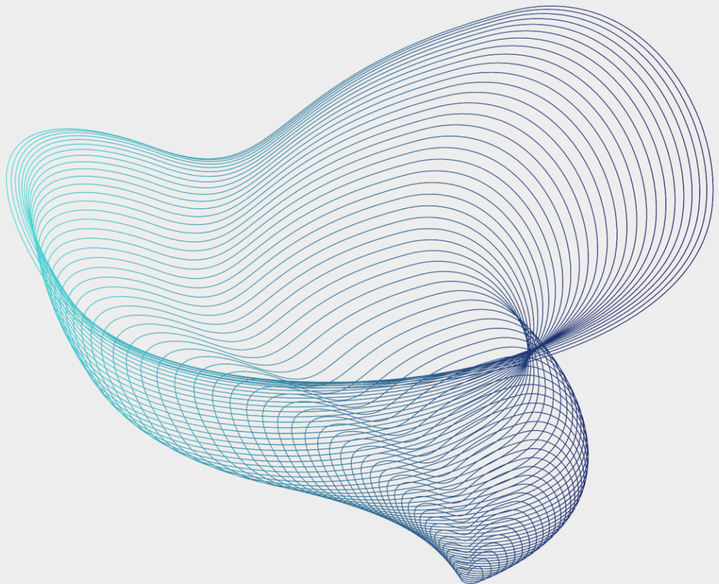
```
# Took 5 year data into consideration.

start = datetime.datetime(2021,1,1)
end = datetime.datetime(2022,1,1)
```



Importing Data from Yahoo

```
# Used Yahoo Finance and took Nifty 50 as index for market return, and Yahoo Finance as the data source provider.

df_nse = web.DataReader('^NSEI','yahoo',start,end)
df_nse.head()
```

Date	High	Low	Open	Close	Volume	Adj Close
2021-01-01	14049.849609	13991.349609	13996.099609	14018.500000	358100	14018.500000
2021-01-04	14147.950195	13953.750000	14104.349609	14132.900391	495000	14132.900391
2021-01-05	14215.599609	14048.150391	14075.150391	14199.500000	492500	14199.500000
2021-01-06	14244.150391	14039.900391	14240.950195	14146.250000	632300	14146.250000
2021-01-07	14256.250000	14123.099609	14253.750000	14137.349609	559200	14137.349609



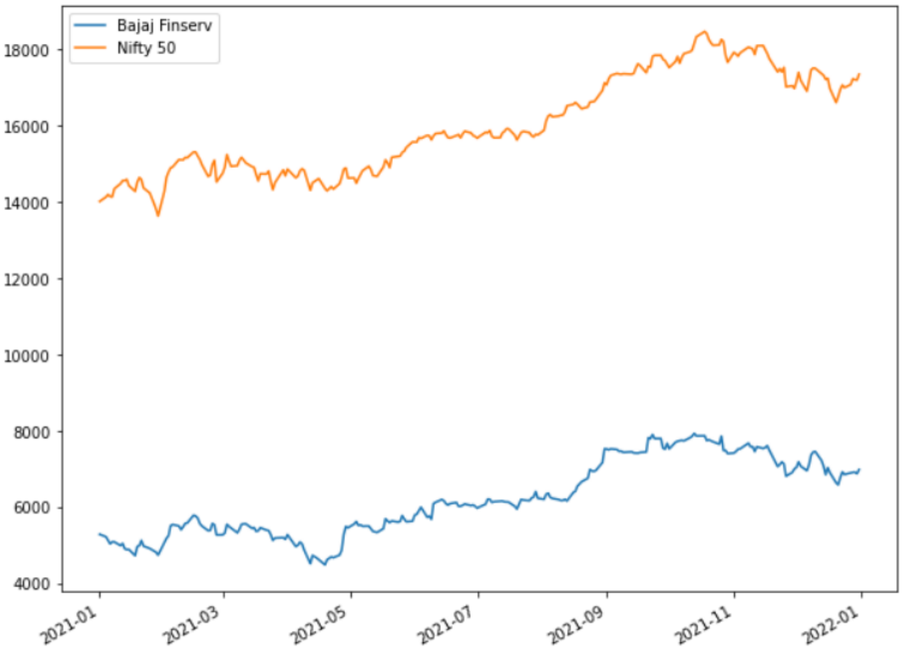
```
[ ]  # Took Bajaj Finance Ltd. as the stock in consideration.

df_stock = web.DataReader('BAJFINANCE.NS','yahoo',start,end)
df_stock.head()
```

Date	High	Low	Open	Close	Volume	Adj Close
2021-01-01	5338.000000	5250.000000	5310.200195	5280.149902	1447187.0	5271.631836
2021-01-04	5324.000000	5196.200195	5324.000000	5216.200195	2333659.0	5207.785156
2021-01-05	5224.200195	5062.500000	5218.000000	5119.000000	3953400.0	5110.741699
2021-01-06	5168.350098	4990.299805	5130.000000	5030.299805	3607923.0	5022.184570
2021-01-07	5131.549805	5021.299805	5065.000000	5081.000000	3035750.0	5072.803223

Plotting Closing Values Curve for both Stocks

```
# Absolute plot of Stocks price.

df_stock['Close'].plot(label = 'Bajaj Finserv', figsize=(10,8))
df_nse['Close'].plot(label = 'Nifty 50')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x27dbeaf2c40>
```

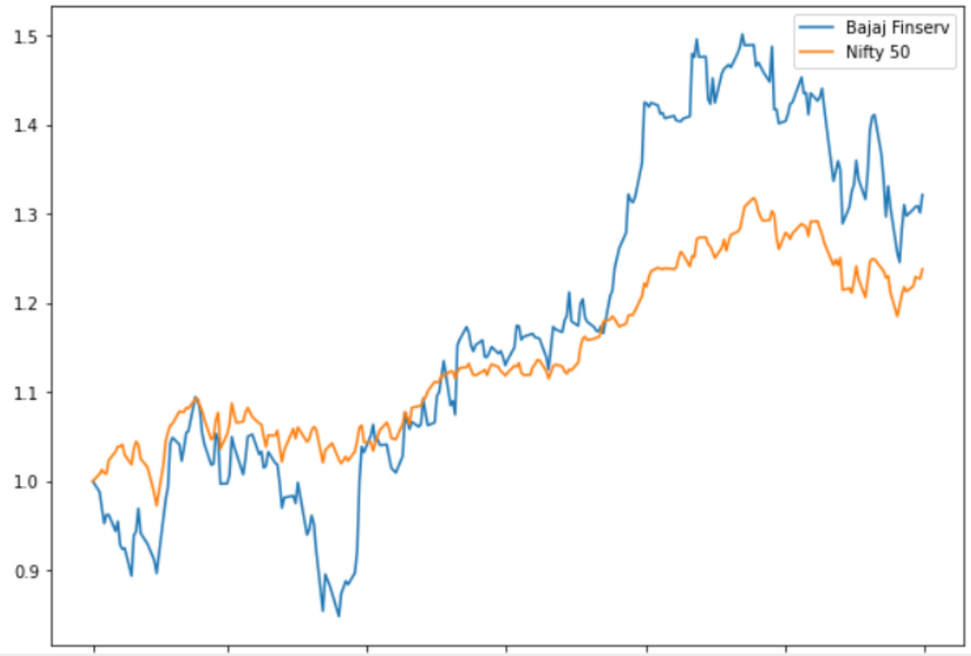


Adjusting Values of Close and Plotting Curve of Adjusted Values

```
df_stock['Cumu'] = df_stock['Close']/df_stock['Close'].iloc[0] #cumulative return, rrelative to Day0 closing.
df_nse['Cumu'] = df_nse['Close']/df_nse['Close'].iloc[0]
```

```
# Cummulative plot graph for vizulaizing relative gain/loss of a stocks.

df_stock['Cumu'].plot(label = 'Bajaj Finserv', figsize=(10,8))
df_nse['Cumu'].plot(label = 'Nifty 50')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x27dbdc4c8e0>
```



Introducing a new Column for Daily Return

```
[ ]  # Daily changes in cummulative price of stocks.

     df_stock['daily_ret'] = df_stock['Close'].pct_change(1)
     df_nse['daily_ret'] = df_nse['Close'].pct_change(1)
```

```
df_nse
```

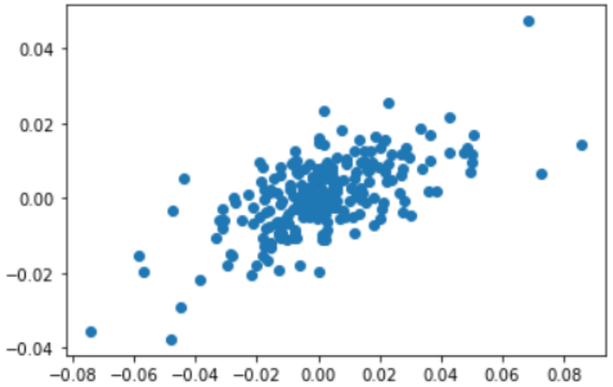
Date	High	Low	Open	Close	Volume	Adj Close	Cumu	daily_ret
2021-01-01	14049.849609	13991.349609	13996.099609	14018.500000	358100	14018.500000	1.000000	NaN
2021-01-04	14147.950195	13953.750000	14104.349609	14132.900391	495000	14132.900391	1.008161	0.008161
2021-01-05	14215.599609	14048.150391	14075.150391	14199.500000	492500	14199.500000	1.012912	0.004712
2021-01-06	14244.150391	14039.900391	14240.950195	14146.250000	632300	14146.250000	1.009113	-0.003750
2021-01-07	14256.250000	14123.099609	14253.750000	14137.349609	559200	14137.349609	1.008478	-0.000629
...	...	...	...	...	...	...	...	...
2021-12-27	17112.050781	16833.199219	16937.750000	17086.250000	144800	17086.250000	1.218836	0.004852
2021-12-28	17250.250000	17161.150391	17177.599609	17233.250000	176000	17233.250000	1.229322	0.008603
2021-12-29	17285.949219	17176.650391	17220.099609	17213.599609	161700	17213.599609	1.227920	-0.001140
2021-12-30	17264.050781	17146.349609	17201.449219	17203.949219	320800	17203.949219	1.227232	-0.000561
2021-12-31	17400.800781	17238.500000	17244.500000	17354.050781	167000	17354.050781	1.237939	0.008725

248 rows × 8 columns

Plotting Scatter plot for Daily Returns and Using linear regression to find Alpha and Beta

```
# Scatter plots of daily returns of stocks.

plt.scatter(df_stock['daily_ret'],df_nse['daily_ret'])
```

```
<matplotlib.collections.PathCollection at 0x27dbeb2f730>
```




```
[ ] # Plotting Linear regression of the above graph

    LR = stats.linregress(df_stock['daily_ret'].iloc[1:],df_nse['daily_ret'].iloc[1:])
```

```
[ ] # Values obtaibed

    LR
```

```
    LinregressResult(slope=0.3028960742957345, intercept=0.0005058653732726757, rvalue=0.6396327955262322, pvalue=8.058089332556864e-30, stderr=0.023255506783162176)
```

```
[ ] # Initializing the values to variables.

    beta,alpha,r_val,p_val,std_err = LR
```

Uploading data of a particular date to compare return Values

```
# Slope of graph

beta
```

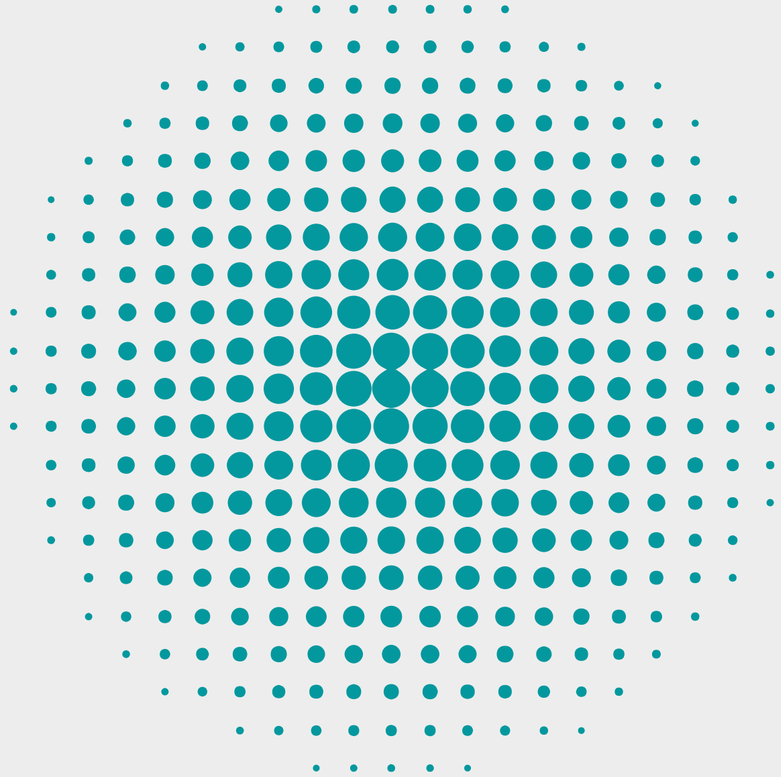
```
0.3028960742957345
```

```
[ ] # Intercept of linesr regression

alpha #not for use as of now.in CAPM model, FF me use ayega.
```

```
0.0005058653732726757
```

```
[ ] # For predicting return on an asset on May 11th, we made a dataframe for that time.

date_1 = datetime.datetime(2022,5,11)
df_stock_1 = web.DataReader('BAJFINANCE.NS','yahoo',date_1, date_1)
df_stock_1.head()
```

Date	High	Low	Open	Close	Volume	Adj Close
2022-05-11	5996.600098	5737	5957	5807	1861173	5807



Comparing Original and CAPM return

```
# Original return obtained.

original_return = df_stock_1.iloc[0].at["Close"]/df_stock.iloc[0].at["Close"]
original_return
```

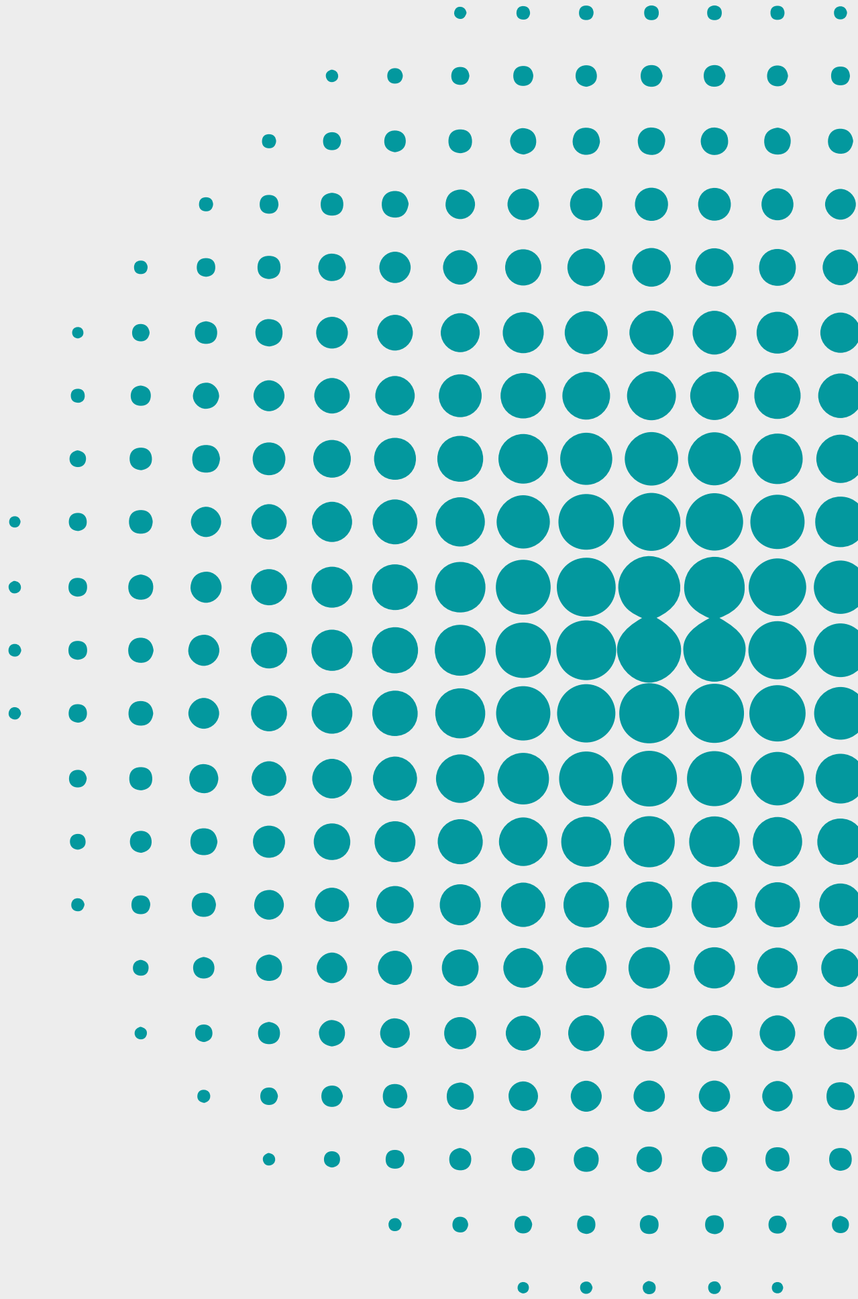
```
1.0997793826691156
```

```
#Using Formula, taking risk free return as 1.

capm_return = 1 + beta * (df_stock.iloc[-1].at["Cumu"] - 1)
capm_return
```

```
1.097357092586165
```

Week 4
Efficient Frontier

OUR TASK WAS TO REPLICATE THE EFFICIENT FRONTIER GRAPH IN EXCEL



WHAT IS THE EFFICIENT FRONTIER?

The efficient frontier is the set of optimal portfolios that offer the highest expected return for a defined level of risk or the lowest risk for a given level of expected return. Portfolios that lie below the efficient frontier are sub-optimal because they do not provide enough return for the level of risk

EFFICIENT FRONTIER

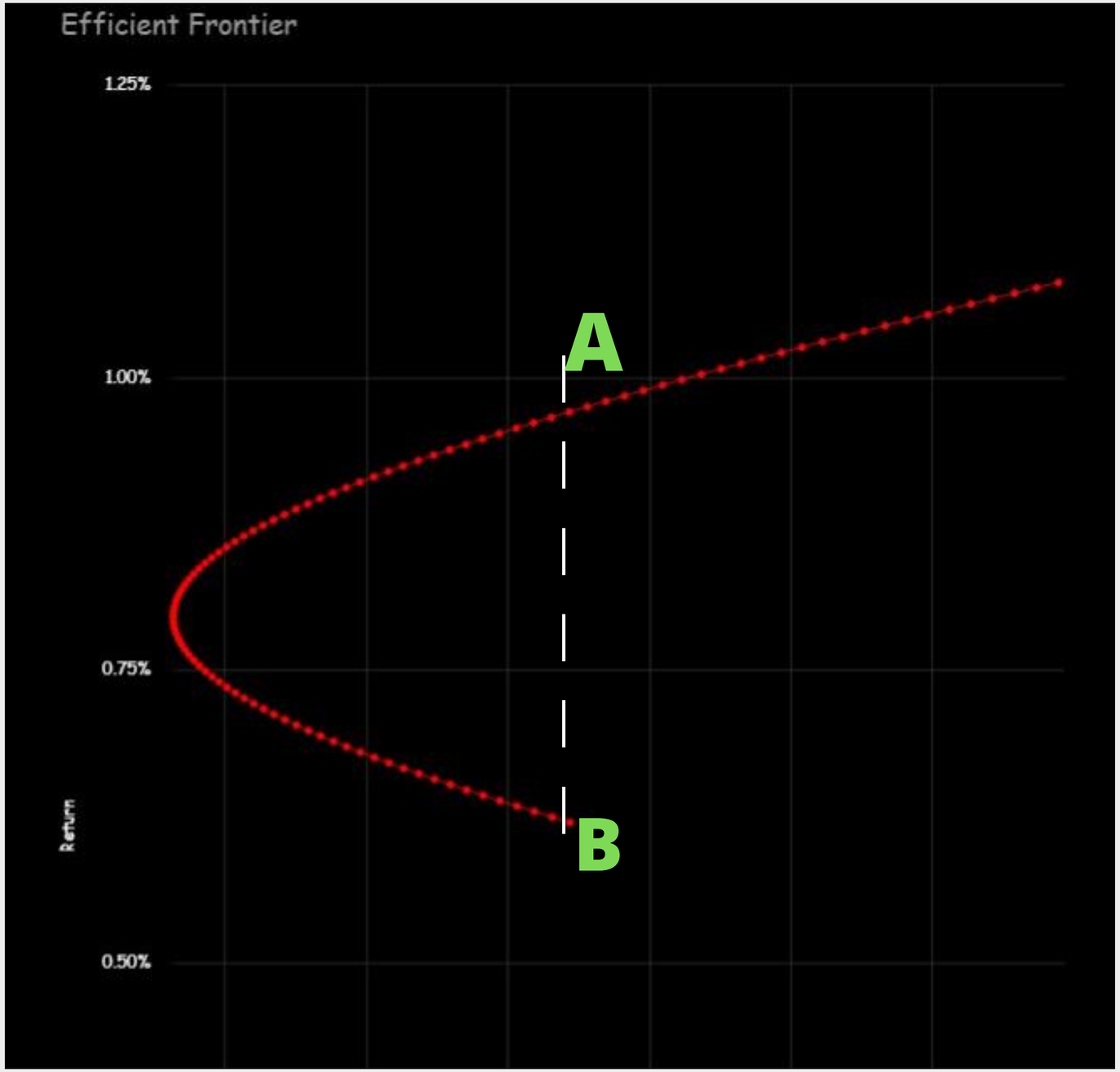
¨WE WERE PROVIDED WITH DATA OF TWO STOCKS: STOCK A & STOCK B. FIRSTLY, WE NEED TO PLOT THE ADJUSTED CLOSE DATA FOR THE TWO STOCK.THEN, IN THE 'MODEL' PAGE, CALCULATE THE MEAN, STDEV, VARIANCE, SHARPE RATIO, COVARIENCE, AND CO-RELATION OF THE TWO STOCK. (REMEMBER, THAT THE PRICE OF BOTH THE STOCK IS DIFFERENT, SO WE MUST TAKE ADJUSTED VALUE IN CALCULATION OF ALL THIS DATA.)THEN, VARY THE SHARE OF STOCK A FROM 0,1..100 IN THE PORTFOLIO. (SHARE OF STOCK B = 100- STOCK A)THEN, FROM THE VALUES CALCULATED, PLOT THE GRAPH OF MEAN V/S STDEV.FROM THIS GRAPH, CALCULATE THE VALUE OF MAXIMUM SHARPE RATIO. (MIN RISK MAX RETURN)

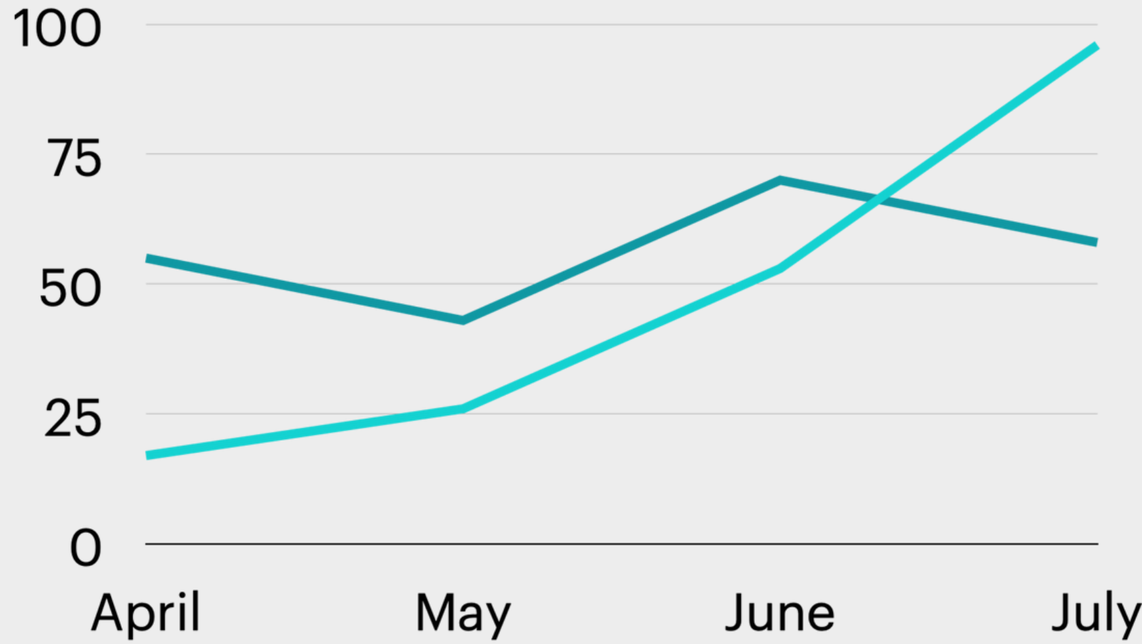
Mean of particular Portfolio

= (percentage of stock A*Mean return of stock A)+

(percentage of stock B*Mean return of stock B)

$$\text{Portfolio variance} = w_1^2\sigma_1^2 + w_2^2\sigma_2^2 + 2w_1w_2\text{Cov}.$$


Efficient Frontier
1.25%
1.00%
0.75%
0.50%
Return
A
B



WEEK 5 :

FAMA FRENCH MODEL

Fama and French Three Factor Model :

The Fama and French Three-factor Model is an asset pricing model developed in 1992 that expands on the CAPM Model by adding the size risk and the value risk factors to the market risk factors. Three factors being used are as follows :

1. **SMALL MINUS BIG (SMB)**
2. **HIGH MINUS LOW (HML)**
3. **PORTFOLIO'S RETURN LESS THE RISK-FREE RATE OF RETURN**

The formula is:

$$R_{it} - R_{ft} = \alpha_{it} + \beta_1(R_{mt} - R_{ft}) + \beta_2(SMB_t) + \beta_3(HML_t) + \varepsilon_{it}$$

R_{it} = Total Return of a stock or portfolio i at a time t

R_{ft} = risk free rate of return at time t

R_{mt} = Total Market portfolio return at time t

$R_{it} - R_{ft}$ = Expected excess return

$R_{mt} - R_{ft}$ = excess return on the market portfolio (index)

SMB_t = size premium (small minus big)

HML_t = value premium (high minus low)

$\beta_{1,2,3}$ = Factor Coefficients

High Minus Low(HML)

1. High Minus Low (HML), also referred to as value premium, is one of the three factors used in the fama French model
2. This system argues that the companies with High Book to market ratios, also known as **Value stocks,** outperform those with lower book to market ratios, known as **Growth Stocks**
3. Along with another factor, called SMB, HML is used to estimate portfolio manger's excess returns
4. If the Manager is buying only value stocks, the model regression shows a positive relation to the HML factor, which explains that the portfolio's returns are attributable to the value premium. Since the model can explain more of the portfolio's return, the original excess return of the manager decreases

Small Minus Big (SMB)

1. Small Minus Big (SMB) is one of three factors in the fama French model. It is also referred to as “**SMALL FIRM EFFECT**” or the “**SIZE EFFECT**”, where size is based on company’s market capitalization
2. The model says that smaller companies outperform larger ones in the long run
3. Small Minus Big is the excess return that the smaller market capitalisation companies return versus larger companies.

Alpha (α)

1. Alpha is a term used in investing to describe an investment strategy's ability to beat the market. Thus alpha is referred to as "Excess return" or "abnormal rate of return", which refers to the idea that the markets are efficient, and so there is no way to systematically earn returns that exceed the broad market as a whole.
2. Alpha represents the performance of a portfolio relative to a benchmark, it is often considered to represent the value that a portfolio's manager adds or subtracts from a fund's return.
3. If Alpha is Positive(+) then the portfolio has outperformed the Benchmark index and if it is Negative(-) then it Underperformed the Benchmark index

Week 6:Implementing Fama French Model Using Python and Finding Error

Importing data of Nifty and Bajaj Auto from yahoo finance

```
#Used Yahoo Finance and took Nifty50 as market return, and Bajaj Auto as refernce stock

df1= web.DataReader('BAJAJ-AUTO.NS','yahoo', start, end)
df2= web.DataReader('^NSEI','yahoo', start, end)
```

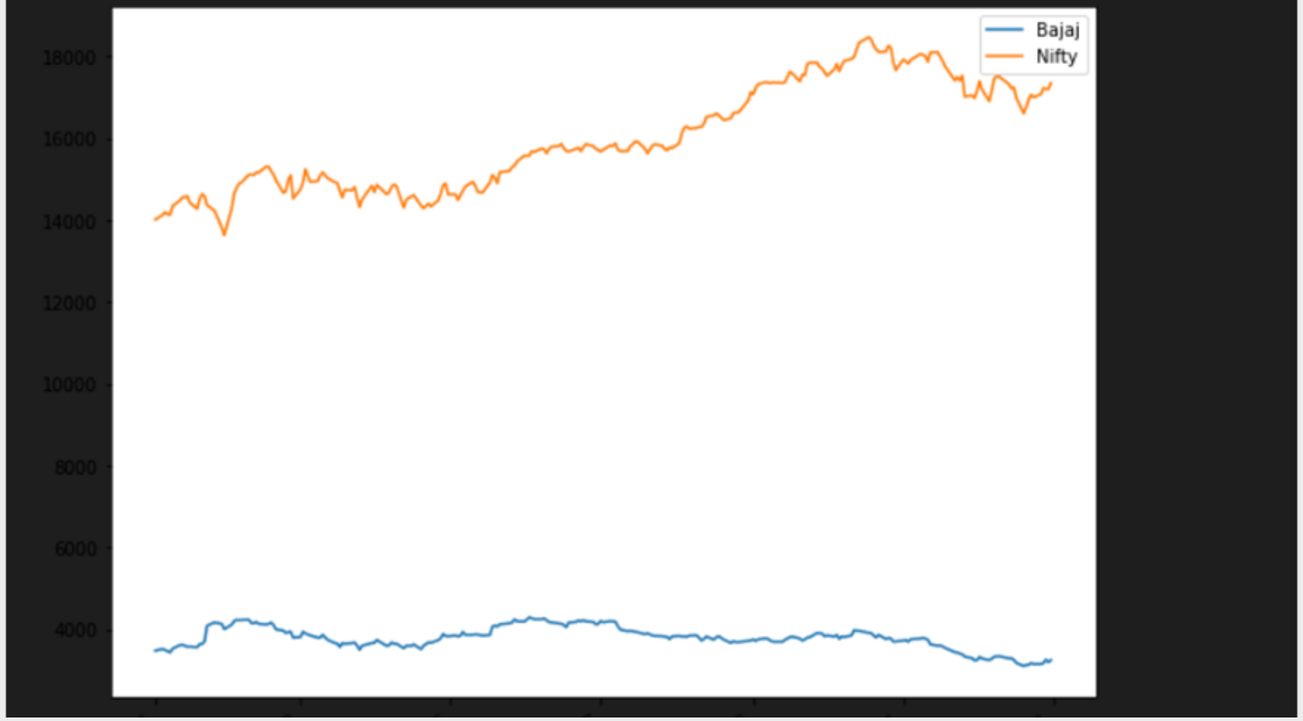
```
df1.head()
```

Date	High	Low	Open	Close	Volume	Adj Close
2021-01-01	3494.000000	3446.000000	3446.000000	3481.250000	421643.0	3243.078613
2021-01-04	3528.000000	3465.000000	3490.000000	3522.449951	647829.0	3281.459961
2021-01-05	3505.000000	3475.000000	3500.250000	3492.649902	561562.0	3253.698730
2021-01-06	3527.000000	3435.899902	3492.649902	3462.699951	591620.0	3225.797852
2021-01-07	3507.350098	3428.250000	3500.000000	3437.949951	531361.0	3202.741211

Plotting absolute closing value curves for both stocks

```
#Absolute plot of Stocks price

df1['Close'].plot(label='Bajaj', figsize=(10, 8))
df2['Close'].plot(label='Nifty')
plt.legend()
```

```
<matplotlib.legend.Legend at 0x2643deab0a0>
```



Adjusting Closing Values

```
#Added adjusted closing values

df1['Cumu'] = df1['Close']/df1['Close'].iloc[0]
df2['Cumu'] = df2['Close']/df2['Close'].iloc[0]
```

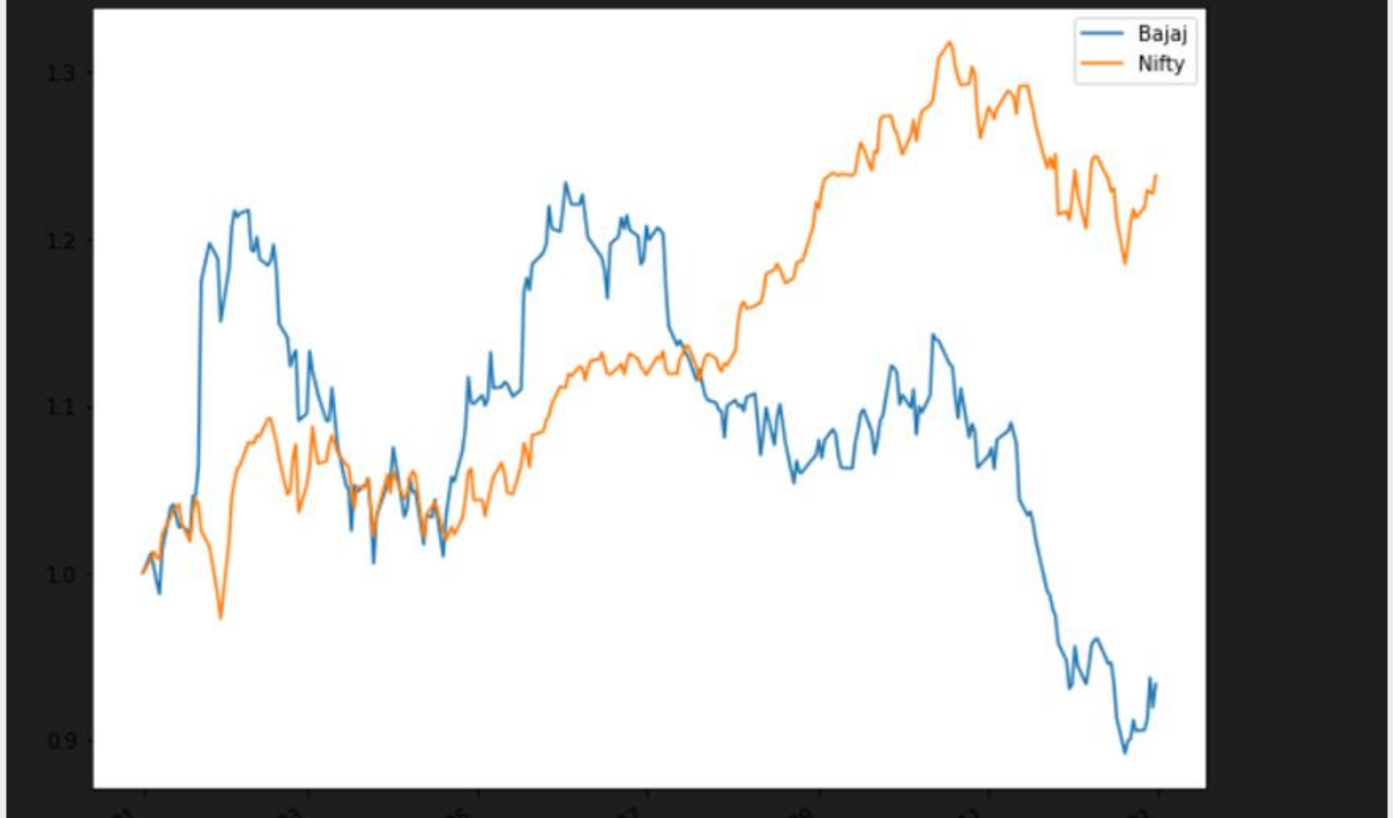
```
df1
```

Date	High	Low	Open	Close	Volume	Adj Close	Cumu
2021-01-01	3494.000000	3446.000000	3446.000000	3481.250000	421643.0	3243.078613	1.000000
2021-01-04	3528.000000	3465.000000	3490.000000	3522.449951	647829.0	3281.459961	1.011835
2021-01-05	3505.000000	3475.000000	3500.250000	3492.649902	561562.0	3253.698730	1.003275
2021-01-06	3527.000000	3435.899902	3492.649902	3462.699951	591620.0	3225.797852	0.994671
2021-01-07	3507.350098	3428.250000	3500.000000	3437.949951	531361.0	3202.741211	0.987562
...	...	...	...	...	...	...	...
2021-12-27	3174.149902	3122.000000	3145.000000	3154.100098	94069.0	3039.926025	0.906025
2021-12-28	3187.550049	3150.000000	3161.199951	3176.050049	214640.0	3061.081543	0.912330
2021-12-29	3269.000000	3152.050049	3188.750000	3262.500000	563709.0	3144.402100	0.937163
2021-12-30	3279.149902	3187.000000	3278.000000	3200.800049	594987.0	3084.935547	0.919440
2021-12-31	3264.699951	3217.000000	3230.000000	3249.250000	248103.0	3131.631592	0.933357

Plotting adjusted closing values

```
#Plotted adjusted values

df1['Cumu'].plot(label = 'Bajaj', figsize=(10,8))
df2['Cumu'].plot(label = 'Nifty')
plt.legend()
```

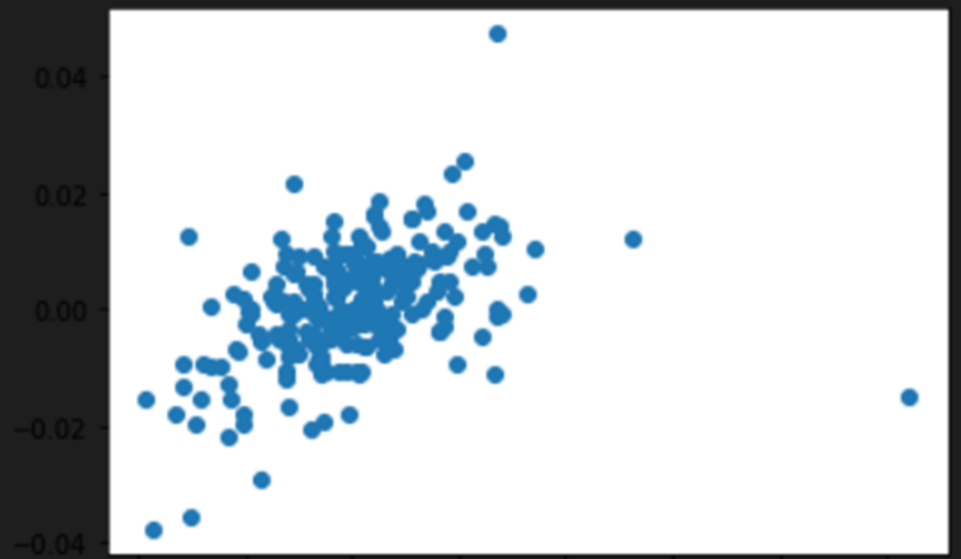
```
<matplotlib.legend.Legend at 0x2641ce90160>
```



Plotting scatter plots from daily return of stocks

```
#Scatter plots of daily return of stocks

df1['daily_ret'] = df1['Close'].pct_change(1)
df2['daily_ret'] = df2['Close'].pct_change(1)
plt.scatter(df1['daily_ret'],df2['daily_ret'])
```

```
<matplotlib.collections.PathCollection at 0x2643e0e1f90>
```



Using Linear Regression to find alpha and beta

```
LR = stats.linregress(df1['daily_ret'].iloc[1:],df2['daily_ret'].iloc[1:])
```

```
LR
```

```
LinregressResult(slope=0.27385776221066616, intercept=0.0009568021412857251, rvalue=0.4325599174929512, pvalue=1.1029472727242061e-12, stderr=0.03646801644229375,
intercept_stderr=0.0005679889271025377)
```

```
beta,alpha,r_val,p_val,std_err = LR
```

```
#Slope of Graph

beta
```

```
0.27385776221066616
```

```
#Intercept of Linear Regression

alpha
```

```
0.0009568021412857251
```

Uploading data of particular date to compare return values

```
#For predicting return on an asset on May 11th, we made a timeframe for that line

date = datetime.date(2022, 5, 11)
df11 = web.DataReader('BAJAJ-AUTO.NS', 'yahoo', date, date)
df11
```

Date	High	Low	Open	Close	Volume	Adj Close
2022-05-11	3627.949951	3537.300049	3600	3612.850098	437746	3482.069824

CAPM return and actual return

```
#Calculated return through CAPM

creturn = 1 + beta*(df1.iloc[-1].at["Cumu"] - 1)
```

```
creturn
```

```
0.9817493713945065
```

```
#Actual return from data

rreturn = (df11.iloc[0].at["Close"])/(df1.iloc[0].at["Close"])
```

```
rreturn
```

```
1.0378025415170558
```

Calculating return using FF Model

```
#SMB and HML factors from data provided

SMB = 0.17
HML = 0.15
```

```
#Calculated return using FF Model

freturn = 1 + alpha + beta*(df1.iloc[-1].at["Cumu"] - 1) + beta*SMB + beta*HML
```

```
freturn
```

```
1.07034065744320555
```

Comparison of error b/w CAPM and FF Model

```
#Error in CAPM

cerror = rreturn - creturn
cerror
```

```
0.056053170122549 29
```

```
#Percentage of error in CAPM

cperror = (cerror/rreturn)*100
cperror
```

```
5.401140186129337
```

```
#Error in FF Model

ferror = abs(freturn - rreturn)
ferror
```

```
0.032538115926149 75
```

```
#Percentage of Error in FF Model

fperror = (ferror/rreturn)*100
fperror
```

```
3.13528967452572
```

Thank you!