PYTHON NOTES

READING AND PLOTING DATA

import pandas as pd
import matplotlib.pyplot

df = pd.read_csv("data/IBM.csv",index_col = ‘Date’,parse_dates = “True”,na_values = [‘nan’])
##Parse_dates parameter converts date and time into objects
#na_values = [‘nan’] helps noting that invalid values are given by ‘nan’

df = df.dropna()#Removes all ‘nan’ values
plt.plot(df['High'])
# or df[‘High’].plot

plt.show() # must be called to show plots

For ploting multiple columns in one plot use df.[[‘col1’,’col2’]].plot

Joining two dataframes -> df1.
For eg, a = a.join(b,how =’inner’) where a,b are dataframes and b is joined to a. All rows of ‘a’ are maintained and only those rows from ‘b’ whose indices are present in ‘a’.

‘how’ parameter chooses rows with indices common to both dataframes and removes ‘nan’ values

pd.Dataframe(parameters)   -> To create a dataframe

Row slicing

df.ix[start index :end index]
##When using date as index use YY-MM-DD

Column slicing

df[[‘GOOG’,’IBM’]]
#To print columns

Both
##Use the name of the column for printing a particular coumn within a particular date range or particular index range
df.ix[row range , column range]
df.ix[start index :end index , [‘GOOG’,’IBM’]]

Data plotting

```
def plot_data(df, title="Stock prices"):
    """Plot stock prices with a custom title and meaningful axis
labels."""
    ax = df.plot(title=title, fontsize=12)
    ax.set_xlabel("Date")
    ax.set_ylabel("Price")
    plt.show()
```

Normalizing

```
    return df/df.ix[0:]
```

Let nd1 = df.values which gives values of dataframe except indices and labels. Elements in an N-D array of a dataframe can be accessed using nd1[row,col]

Array creation in numpy

```
np.array([(1,2,3),(2,3,4)])
```

Here each tuple is a row

Empty array creation

```
np.empty((5,4))#Has 5 rows and 4 cols
```

Array of ones

```
np.ones((5,4), dtype=np.int)
```

Use np.zeros to generate zero array

Generate array of random values

numpy.random.random((5,4))

or

numpy.random.rand(5,4)

It generates an array of random values of 5 rows and 4 cols

numpy.random.normal: Normal or Gaussian distribution

numpy.random.randint: Integers from Uniform distribution array.shape -> This return the dimensions of the array as a tuple

array.size -> No of elements present in the array

array.type -> Returns type of array

Operations on N-D arrays

array.sum()

array.sum(axis=0)

axis=0 , Returns sum of each column in an array

array.sum(axis=1)

axis=1, Returns sum of each row in an array

array.min(axis =0)

#Returns min in each column

array.max(), array.mean()

Masking in numpy

array[array <mean] = mean

#This replaces all elements of array < mean with mean value

Arithmetic operations using array

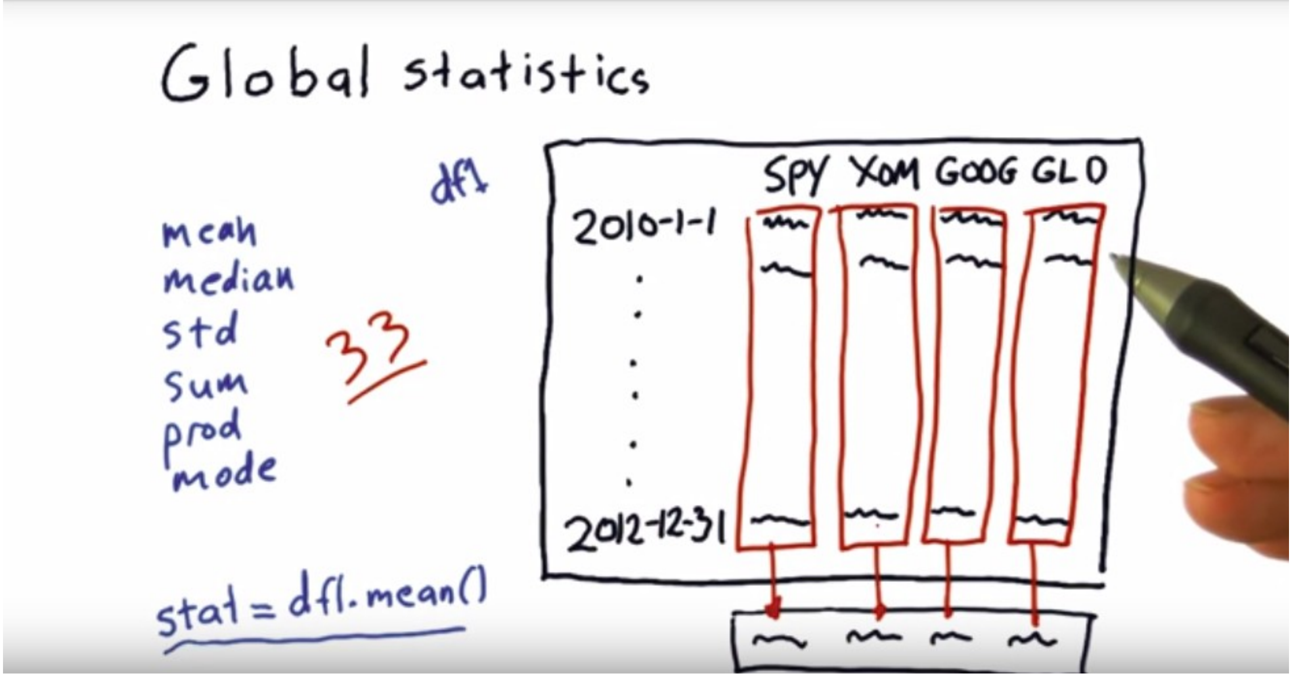
a+b , a-b is normal

a*b returns element wise result

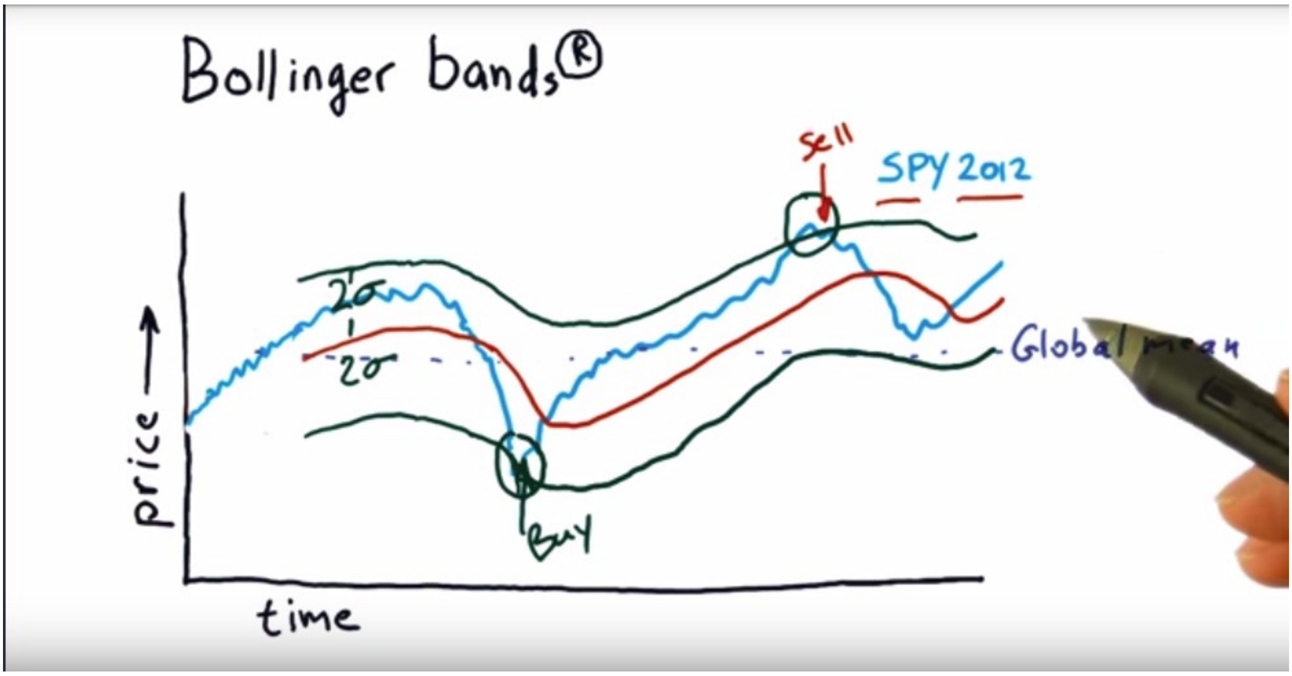
Similarly for division

```
Original array a:
[[ 1  2  3  4  5]
 [10 20 30 40 50]]

Original array b:
[[100 200 300 400 500]
 [  1   2   3   4   5]]

Multiply a and b:
[[ 100  400  900 1600 2500]
 [  10   40   90  160  250]]
```

Statistical Analysis



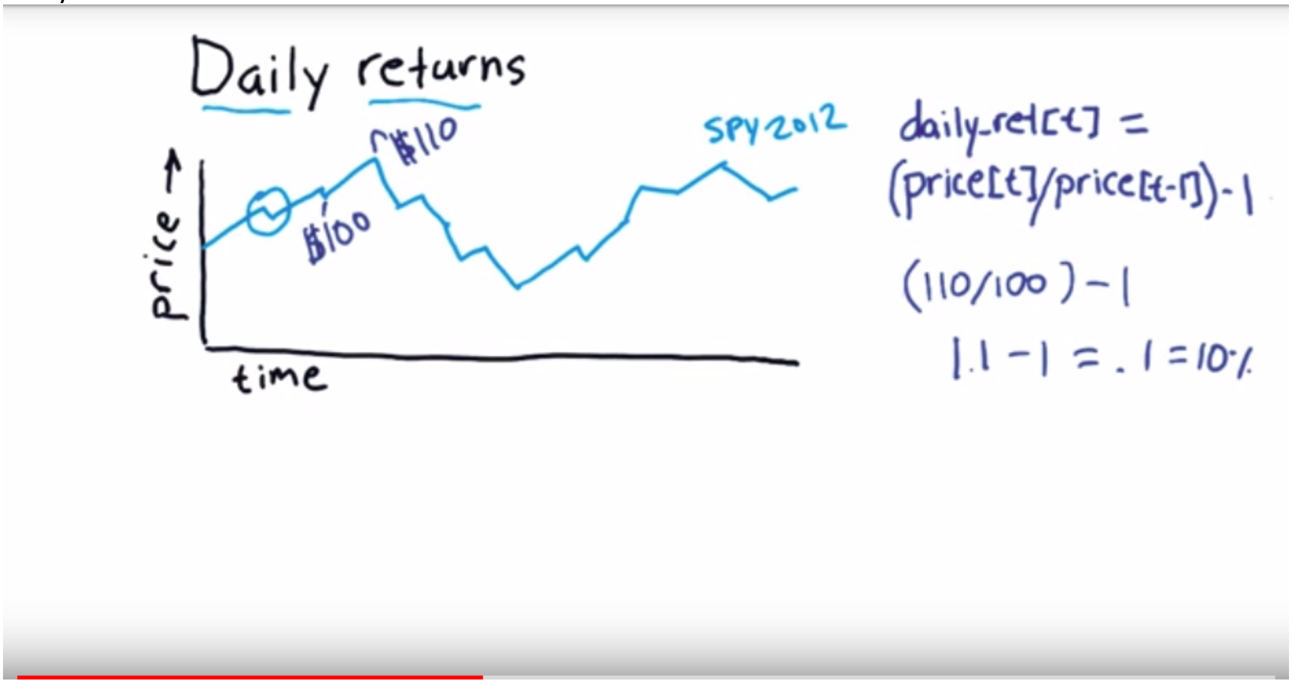
Bollinger bands(Buy/Sell determining)



Code for Bollinger bands

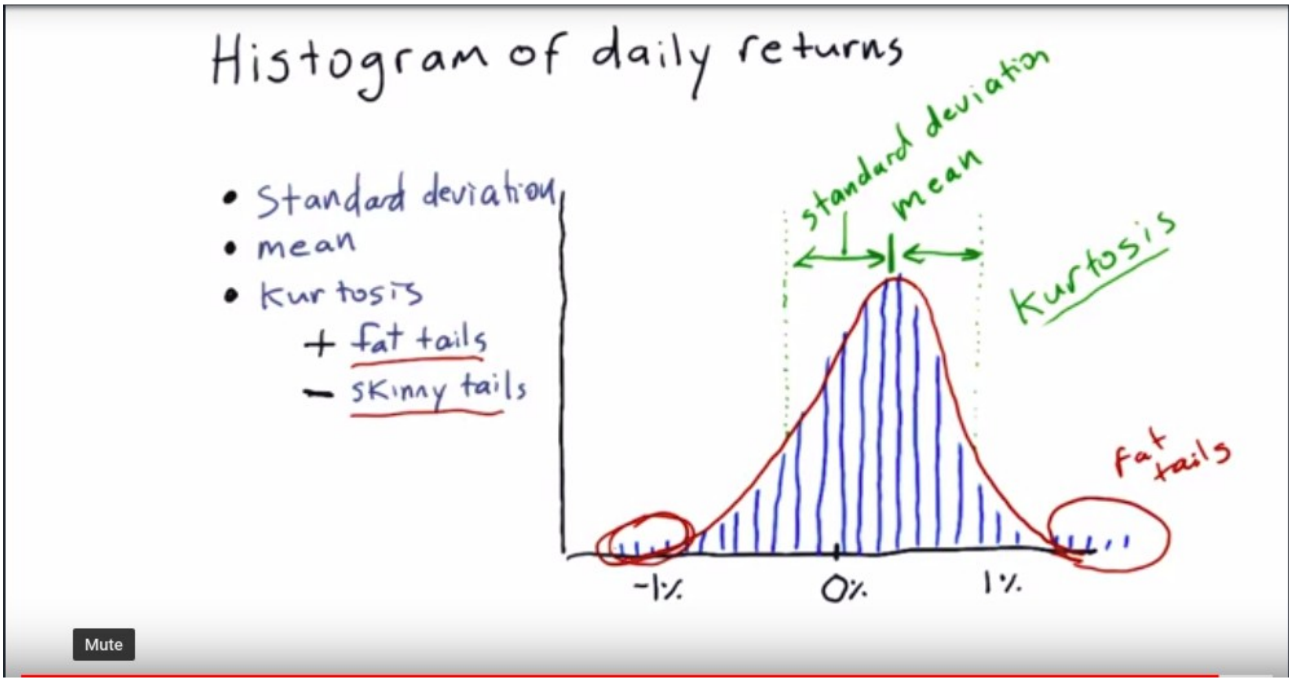
```
def get_bollinger_bands(rm, rstd):
    """Return upper and lower Bollinger Bands."""
    # TODO: Compute upper_band and lower_band
    upper_band = rm + rstd *2
    lower_band = rm - rstd *2
    return upper_band, lower_band
```

Daily return

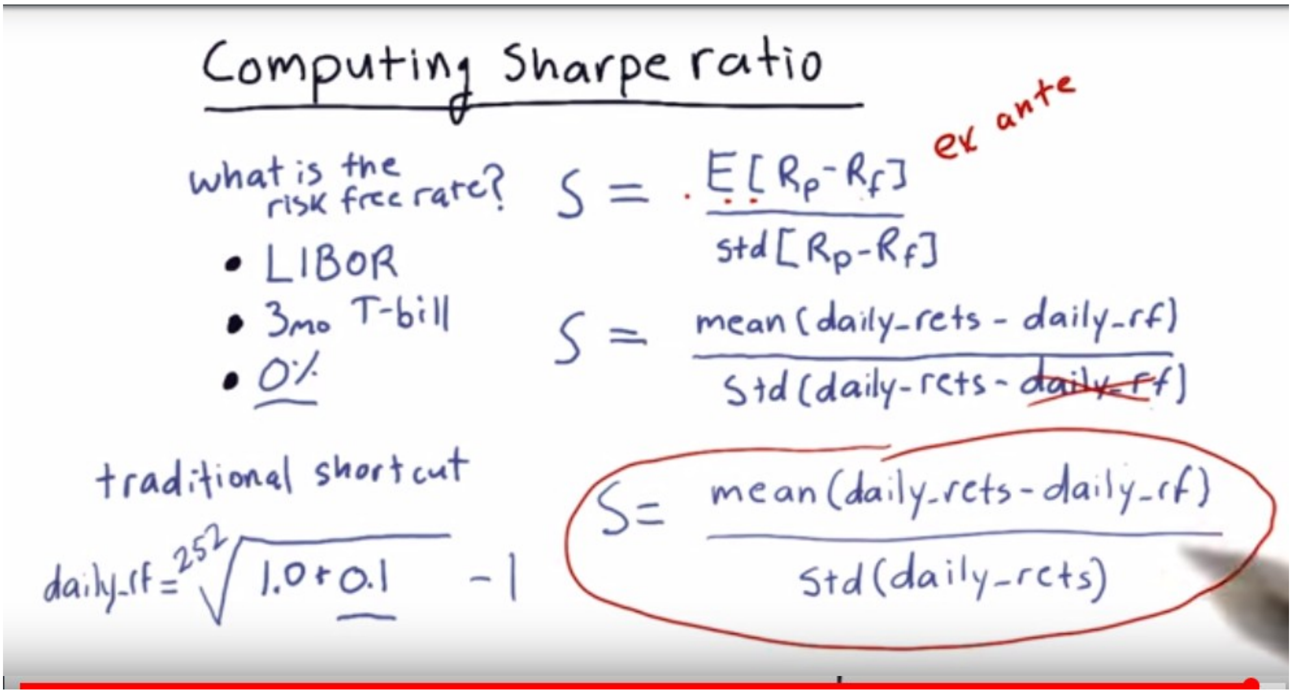


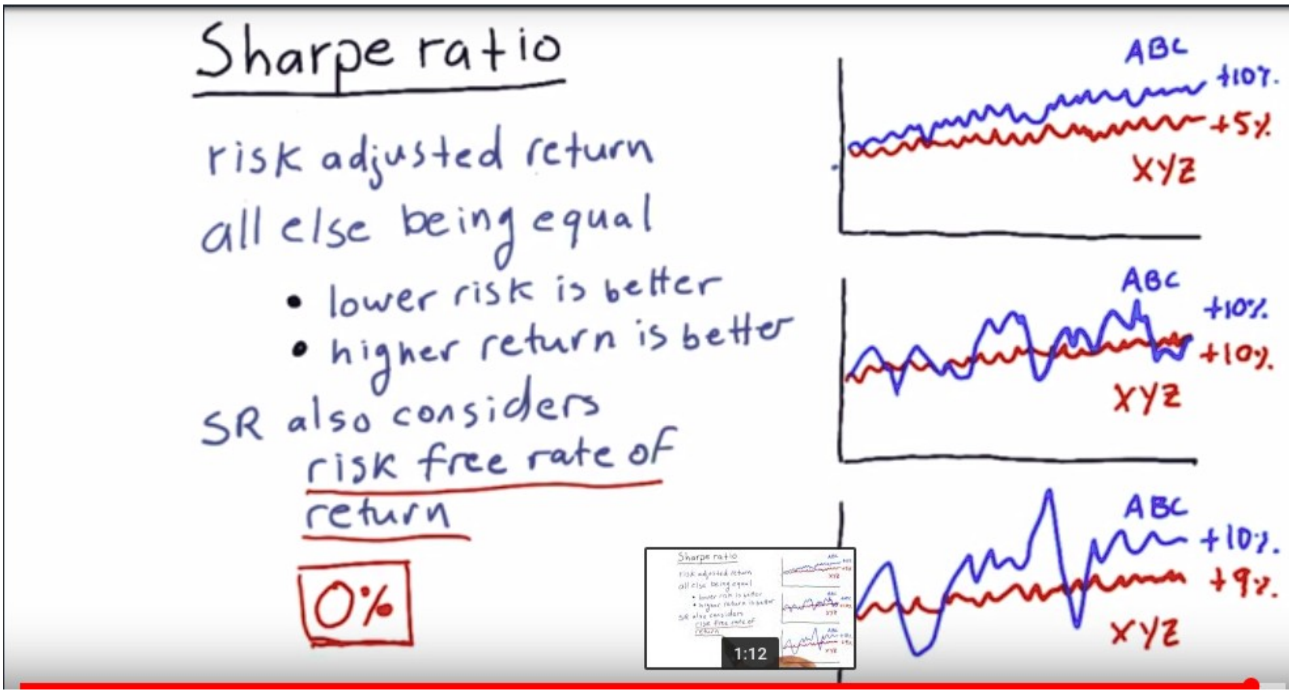
Cumulative returns (For eg , Increase over a year)

Histogram

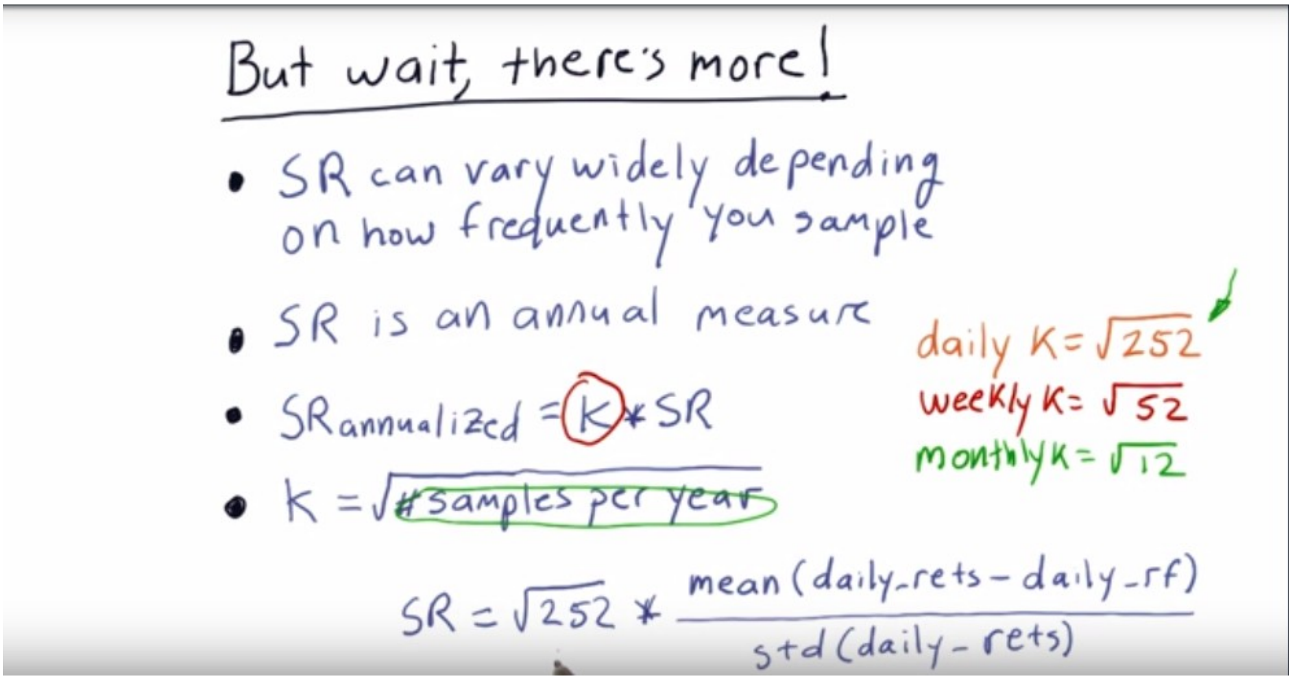

Histogram of daily returns
• Standard deviation
• mean
• Kurtosis
+ fat tails
− skinny tails
standard deviation
mean
Kurtosis
fat tails
-1%
0%
1%
Mute

Important terms

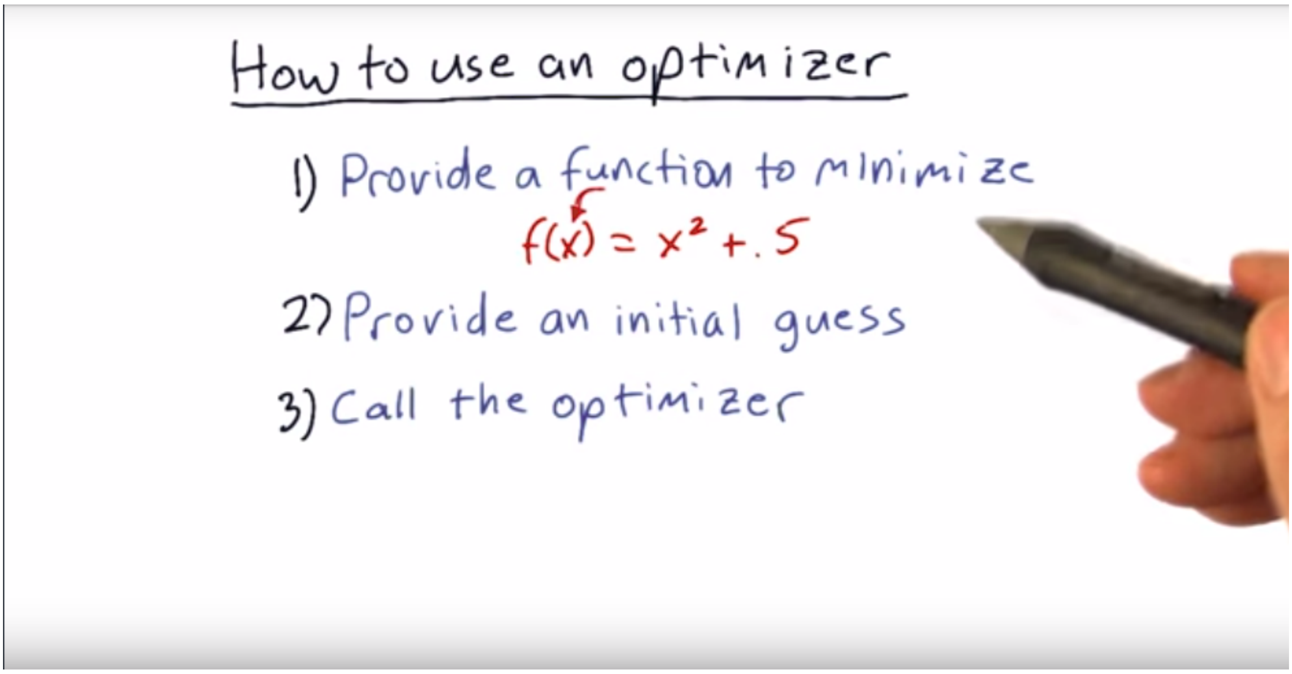




Portfolio : Allocation of funds to a set of stocks



How to use optimizer



python minimizer/optimizer code

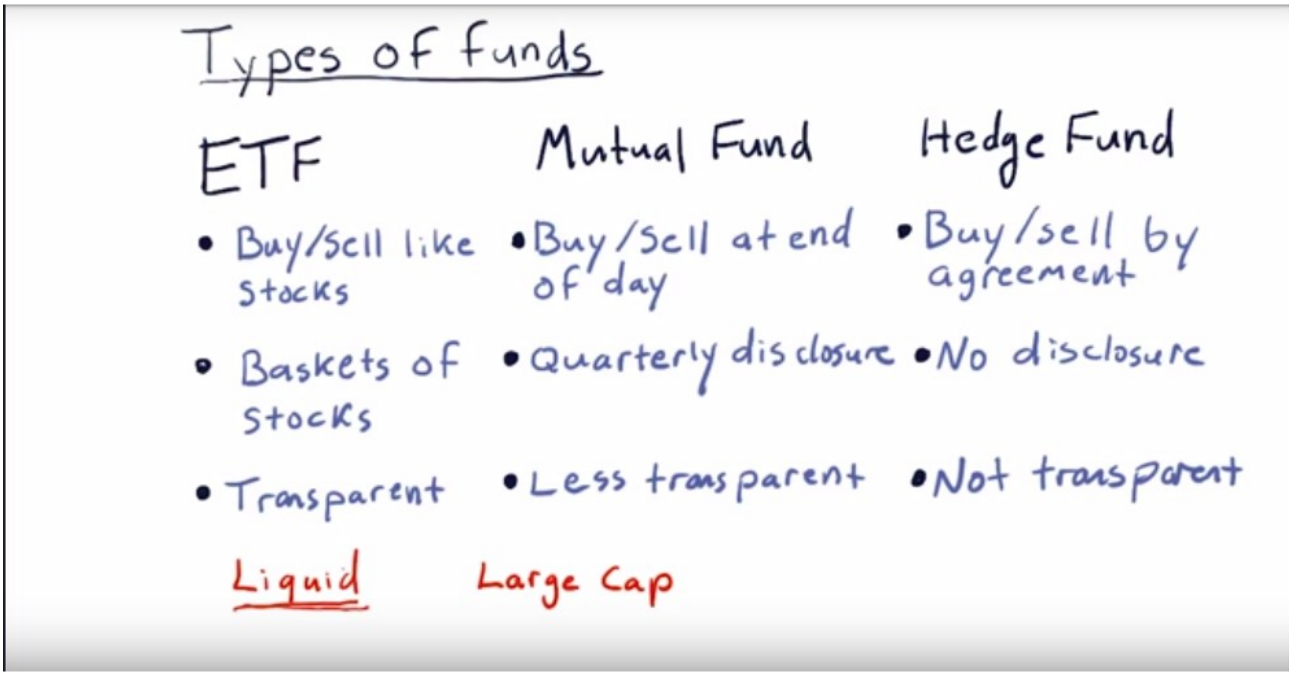
```
    Y = (X - 1.5)**2 + 0.5
    print "X = {}, Y = {}".format(X, Y) # for tracing
    return Y

def test_run():
    Xguess = 2.0
    min_result = spo.minimize(f, Xguess, method='SLSQP', options={'disp': True})
    print "Minima found at:"
    print "X = {}, Y = {}".format(min_result.x, min_result.fun)

    # Plot function values, mark minima
    Xplot = np.linspace(0.5, 2.5, 21)
    Yplot = f(Xplot)
    plt.plot(Xplot, Yplot)
    plt.plot(min_result.x, min_result.fun, 'ro')
    plt.title("Minima of an objective function")
    plt.show()

if __name__ == "__main__":
    test_run()
```

Important terms

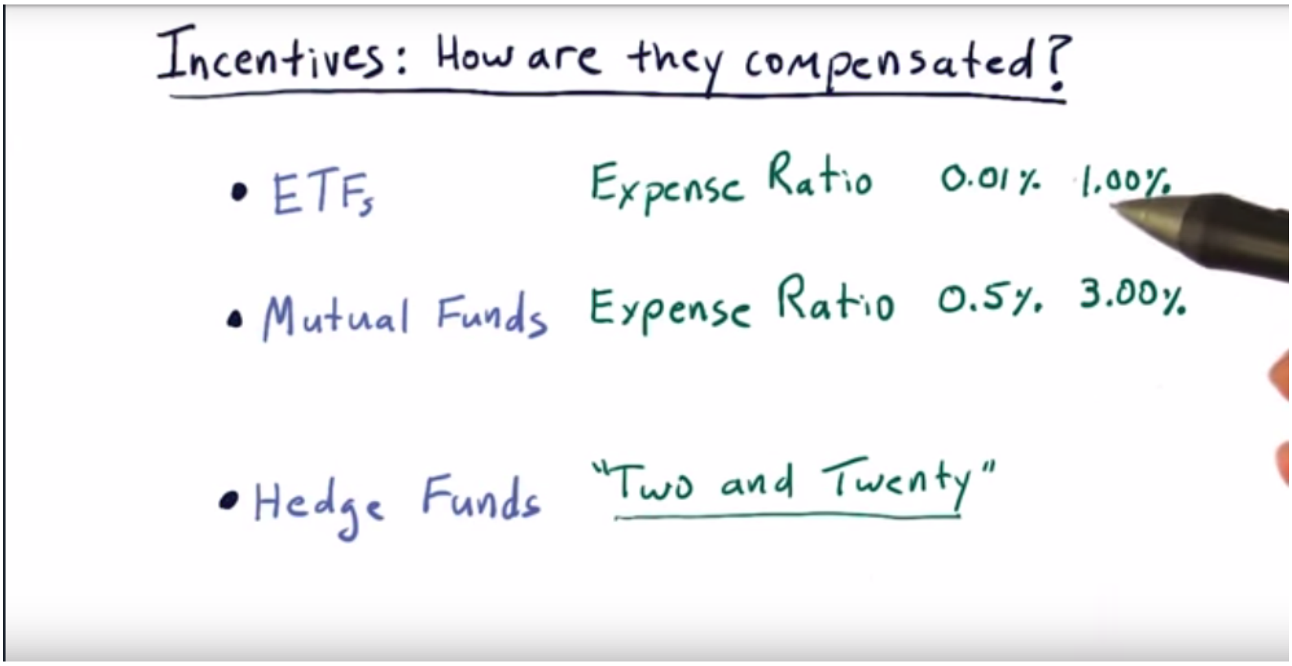
Market capitalization = No of outstanding shares of the company * price of the stock

Liquidity : Ease with which stock can be bought/sold.

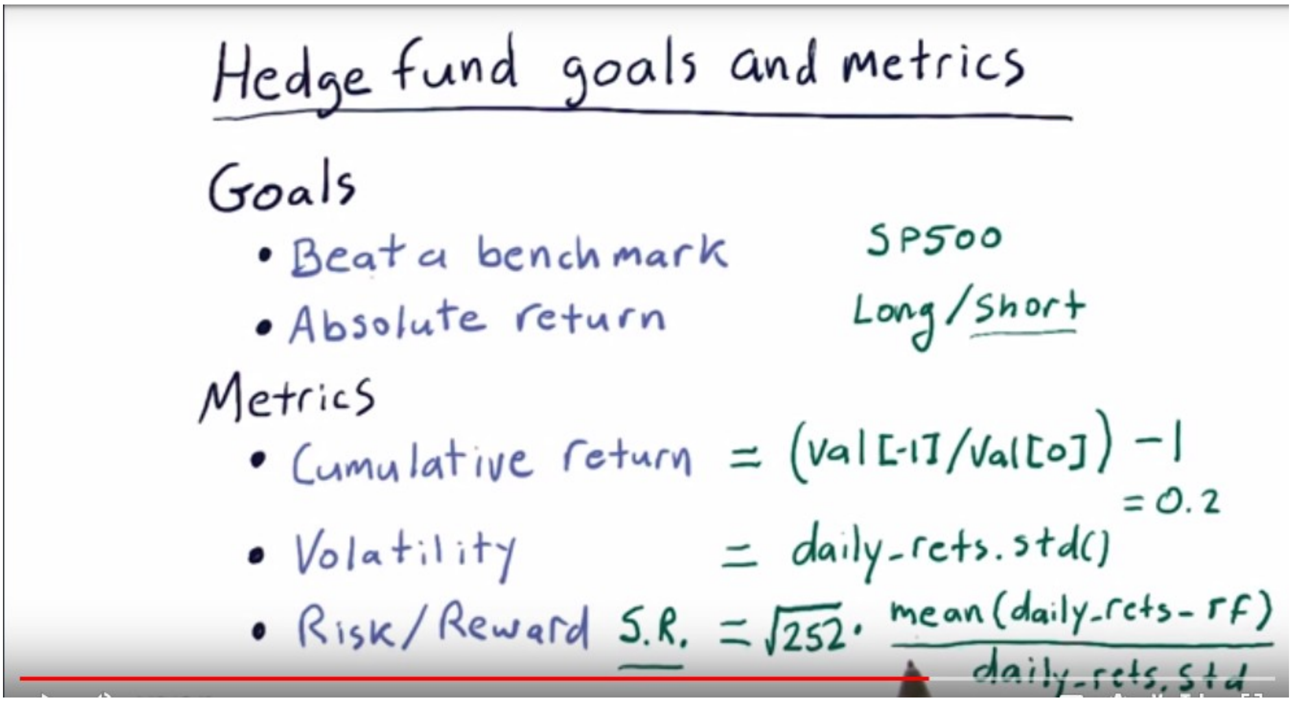
Outstanding shares : This refers to a company's stock currently held by all its shareholders



How fund managers are compensated ?



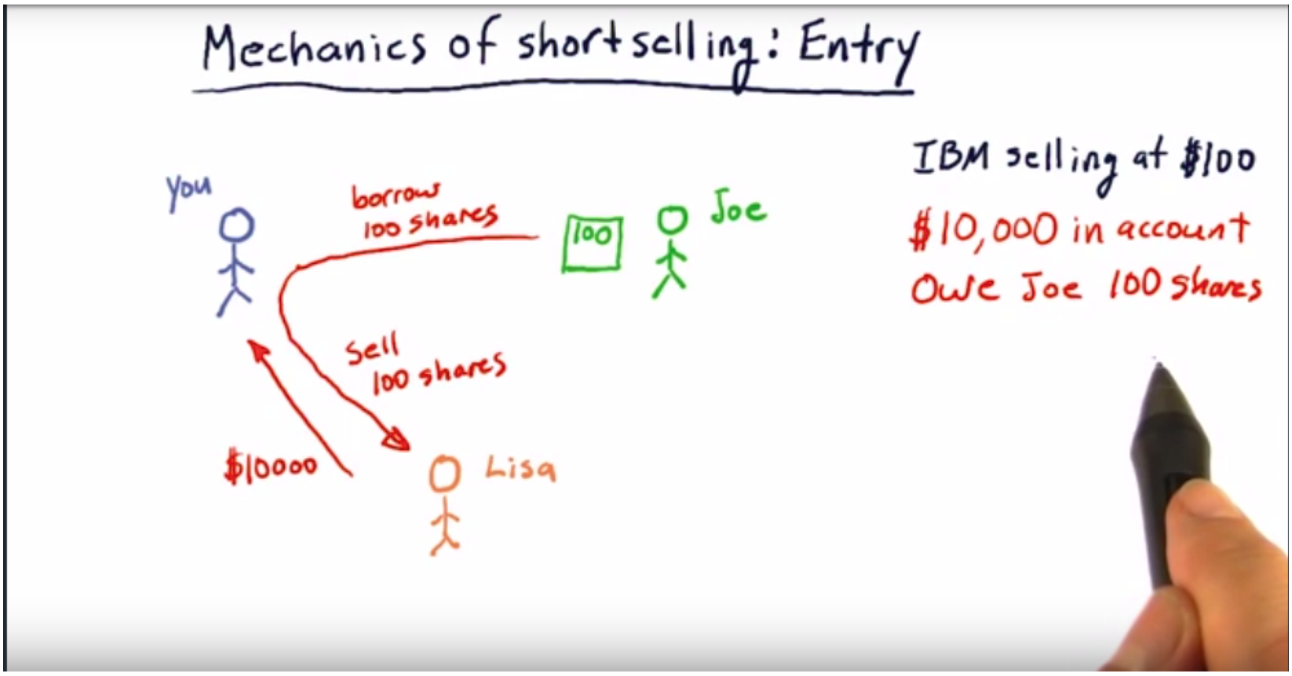
For hedge funds two percent of AUM(assets under management) and 20 percent of profit for fund managers.



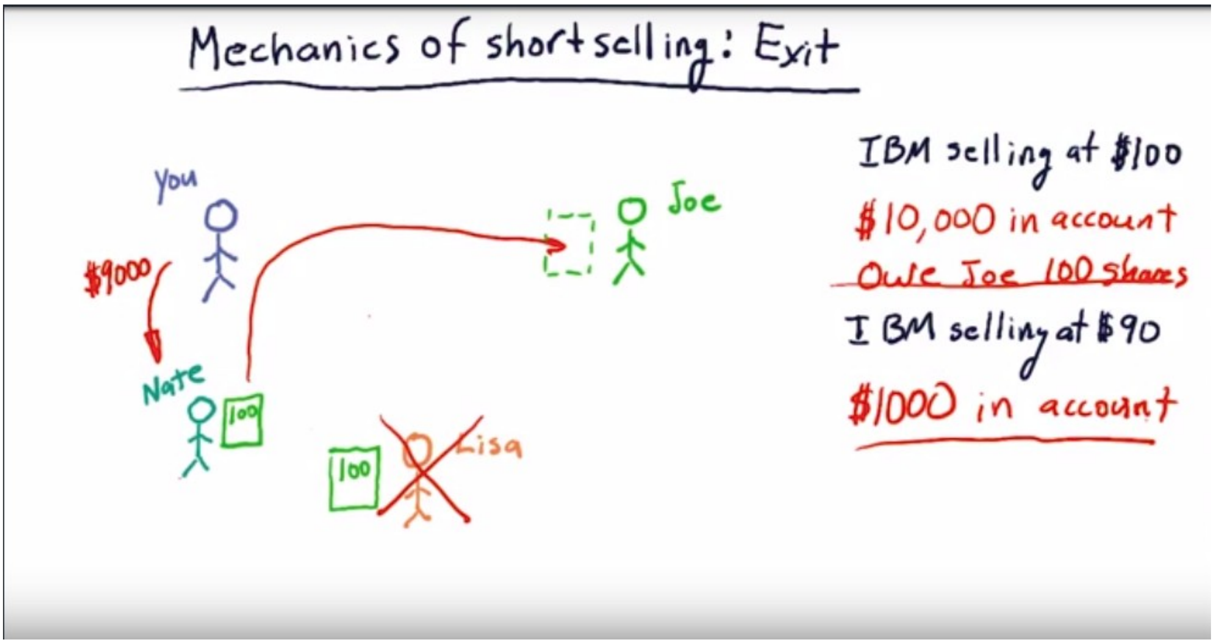
Selling short

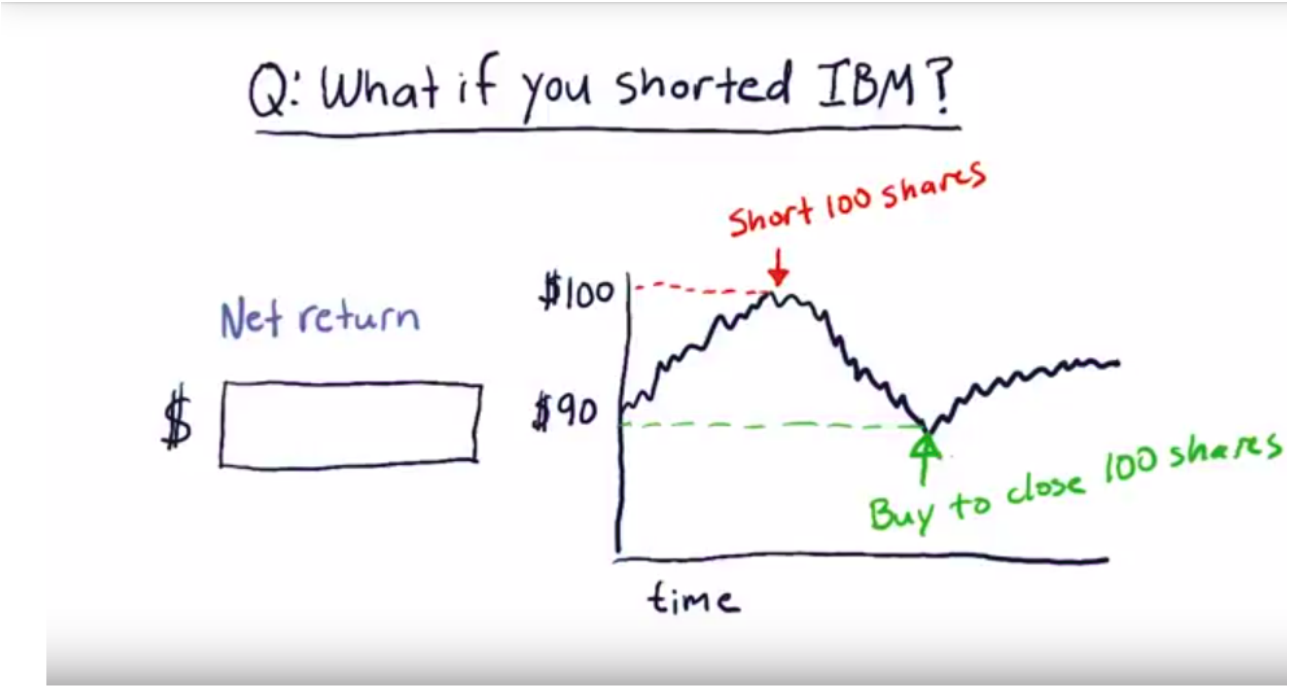
Taking negative position on a stock ie we sell a stock short if we believe that its price will go down. Note that here we are selling stocks that we donot own.

Short selling : Entry

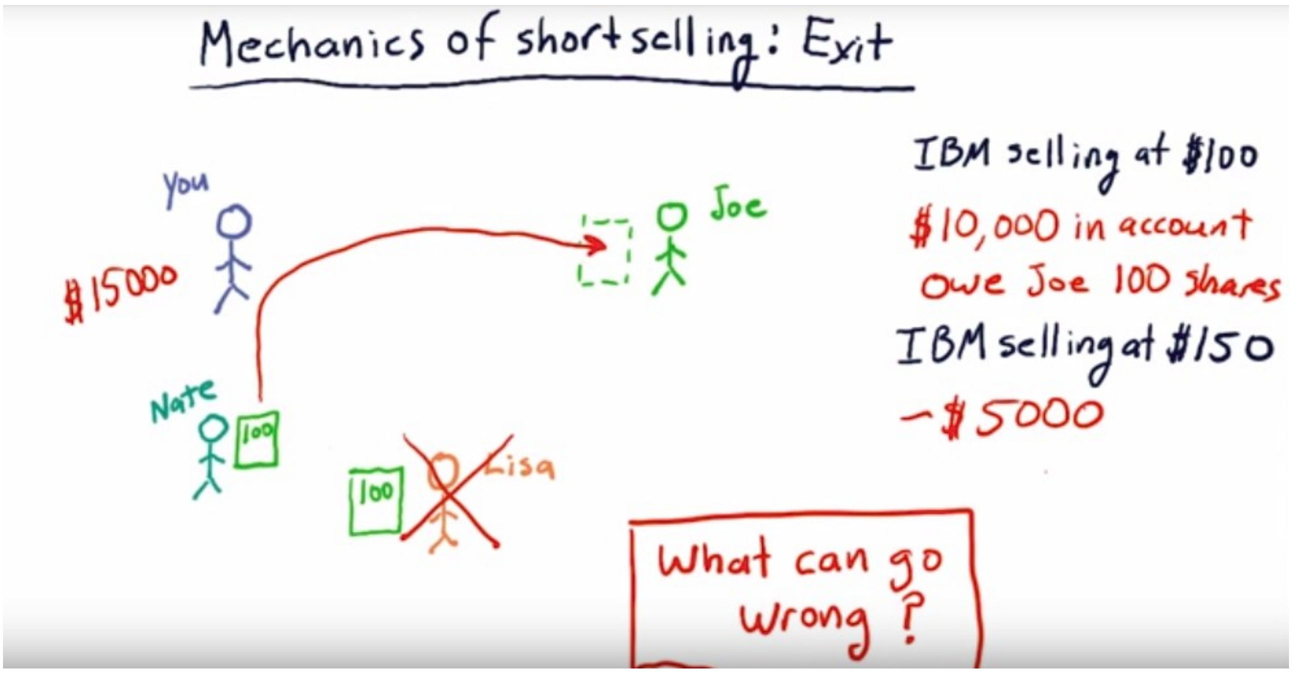


Short selling : Exit





Problem of short selling

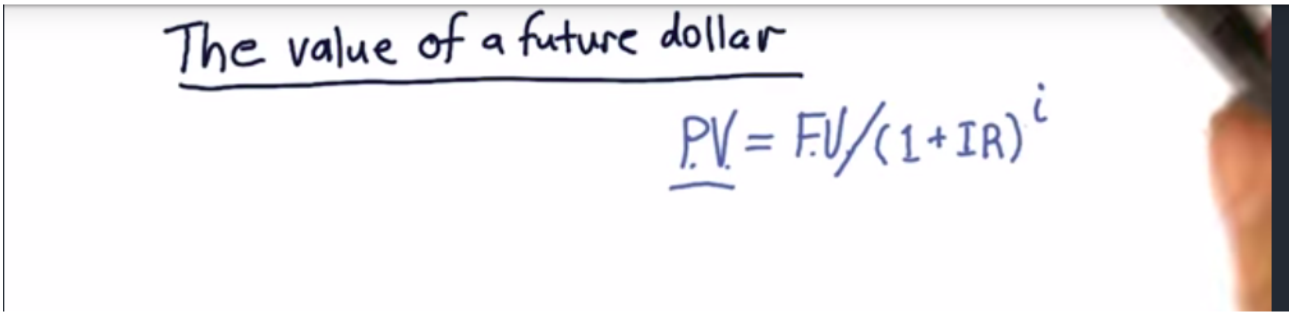


Company value

Intrinsic value : Value of company estimated by future dividends(Amount companies pay to their investors every year based on the number of shares they hold).

Book value : Based on the assets that the company owns ie *Total assets minus intangible assets and liabilities.(intagible assets are value of patent , brand... and total assets = value of properties + patents+brand value)*

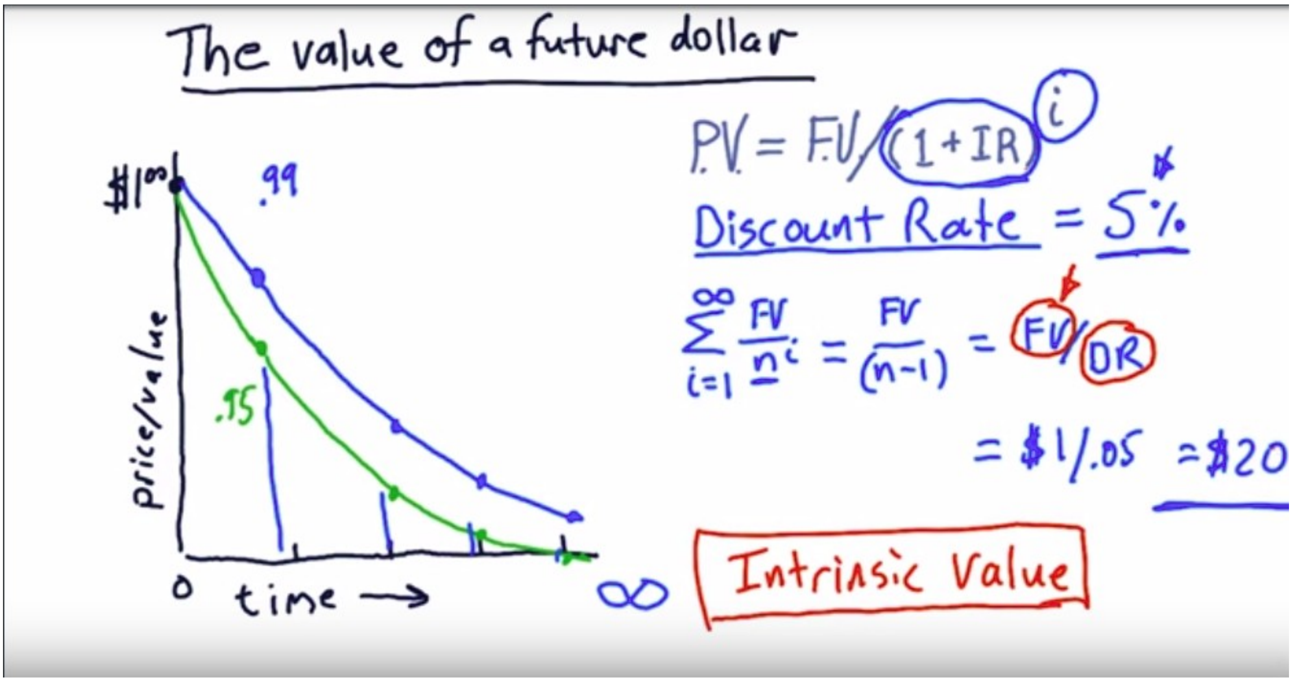
Market cap : Based on value of stock on the market and how many shares are outstanding. (



Where P.V is present value, F.V is future value(or dividend rate), IR is interest rate and i is no of years into the future.

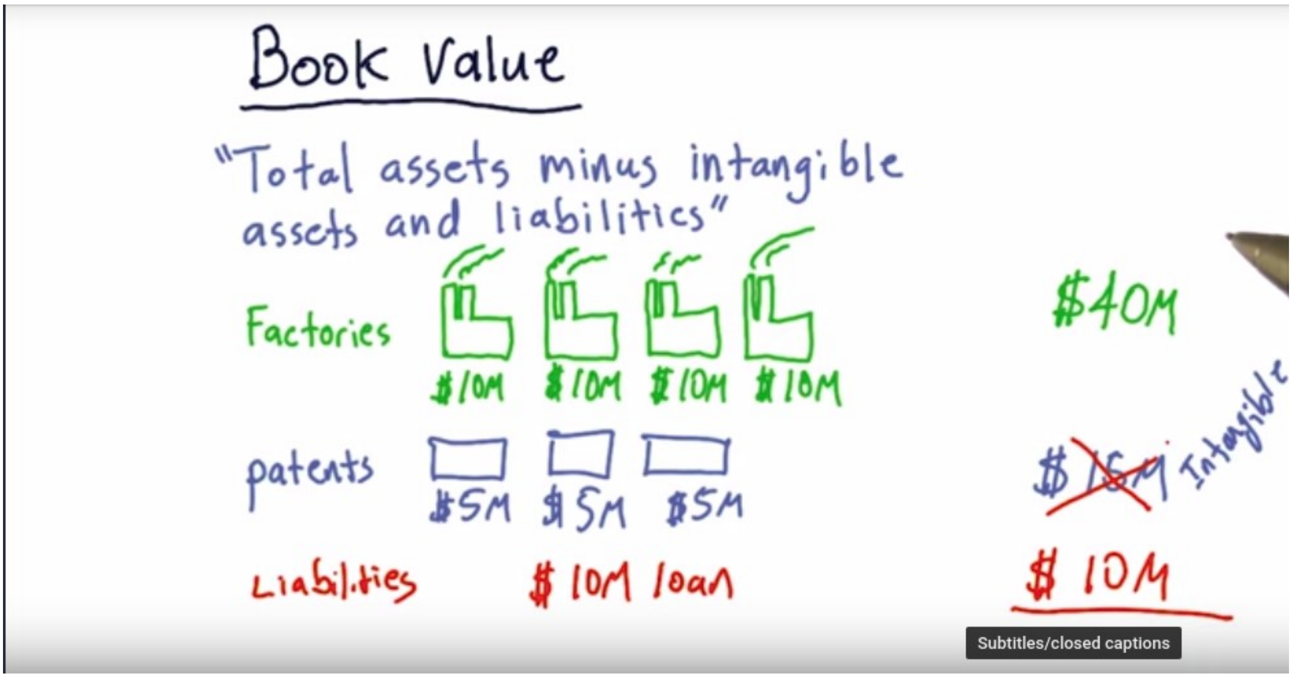
(1+ IR) ^ i ---> Discount rate

Discount rate should be higher for riskier companies



//Here $20 means $20 million

Book value : *Total assets minus intangible assets and liabilities.(intagible assets are value of patent , brand... and total assets = value of properties + patents ,)*



Here book value = total assets - (intagible assets + liabilities)

=(40 + 15) – (15 +10) = 30

Market cap :

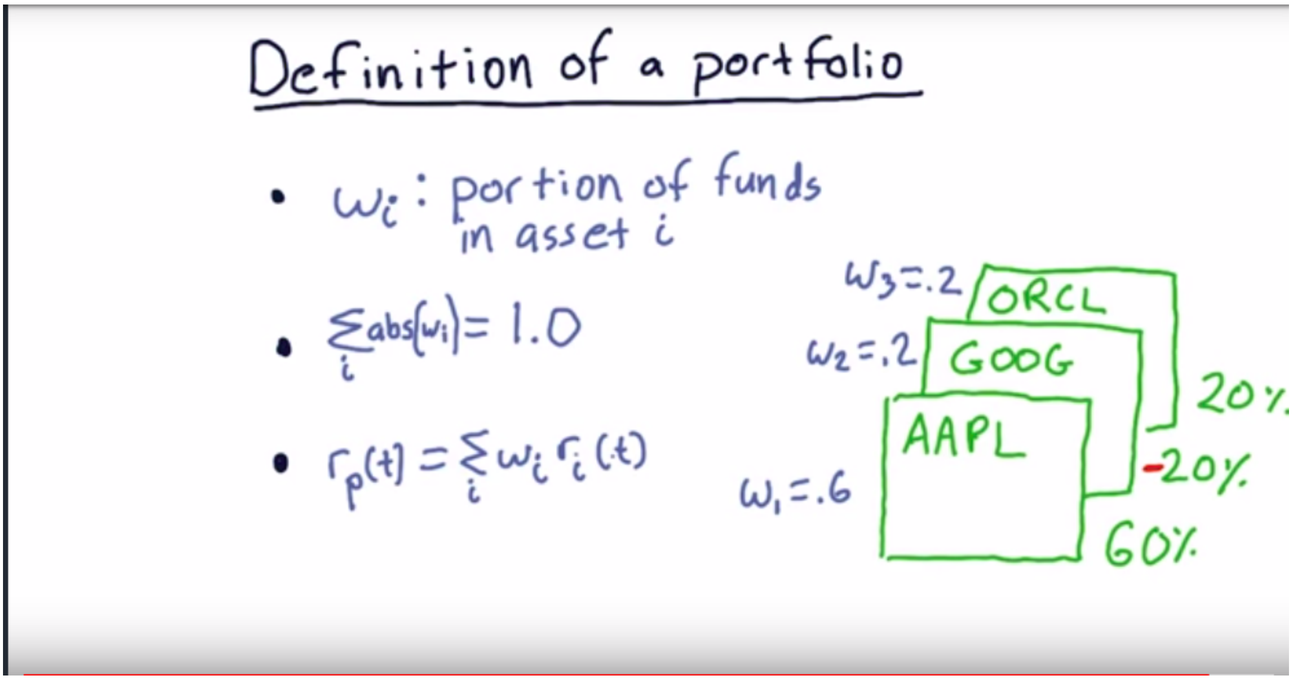
Market capitalization = No of outstanding shares of the company * price of the stock

When intrinsic value is low and market cap is high --> short the stock

High intrinsic value and market cap is low --> Buy the stock

CAP model

Portfolio :



where r_i is the return of that stock and $r_p(t)$ is the total return of that day.

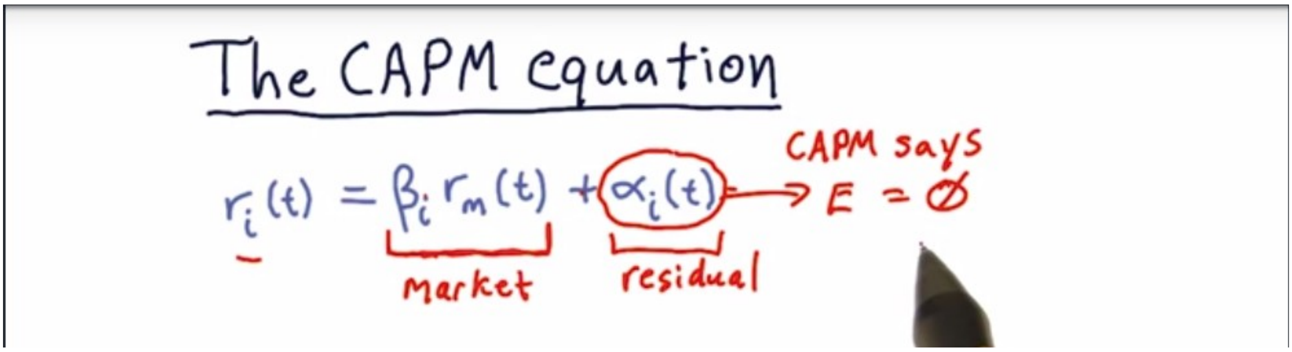
Market portfolio :

Here weight of stocks(w_1,w_2 ...) are set according to market cap of that stock. To calculate weight of any particular stock

w_i = Market cap of that stock / sum of market caps of all stocks where market cap = no of shares * price of stock

Note : The term market just means an index that represents large number of stocks (US : S&P500, UK : FTA ...)

CAPM equation



Return on a particular stock on a particular day ‘ t ’ --> r_i

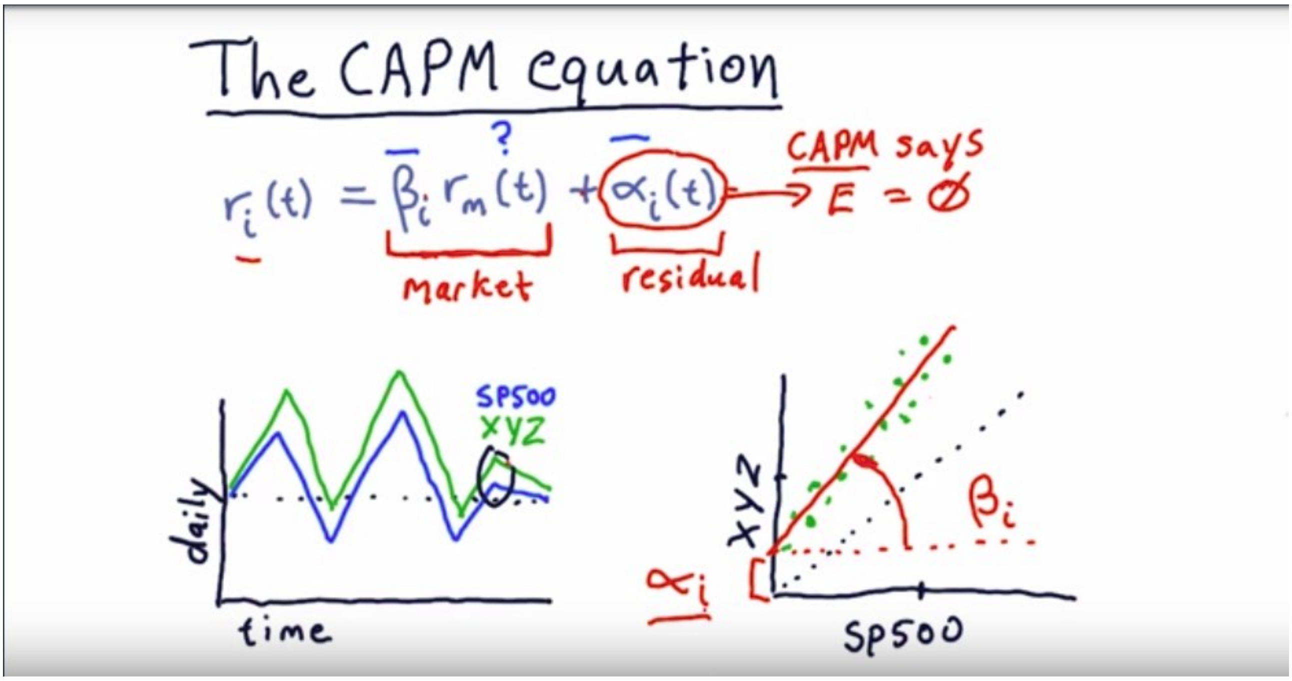
Return on the market on day ‘ t ‘ --> r_m

Beta value is determined when market value goes up or down. Beta(i) represents the relation of a particular stock with market.

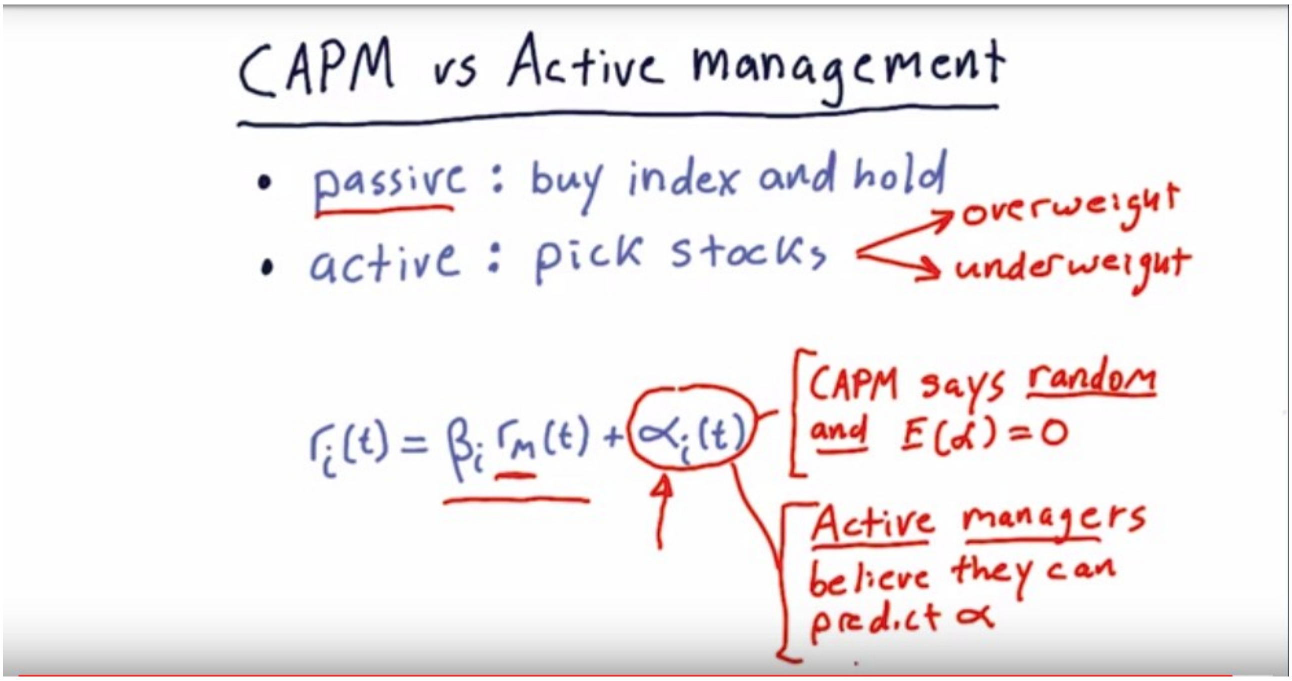
Alpha value expected is mostly 0.

Alpha and beta values are decided based on daily returns(It is decided based on how daily returns relate to daily returns of market)

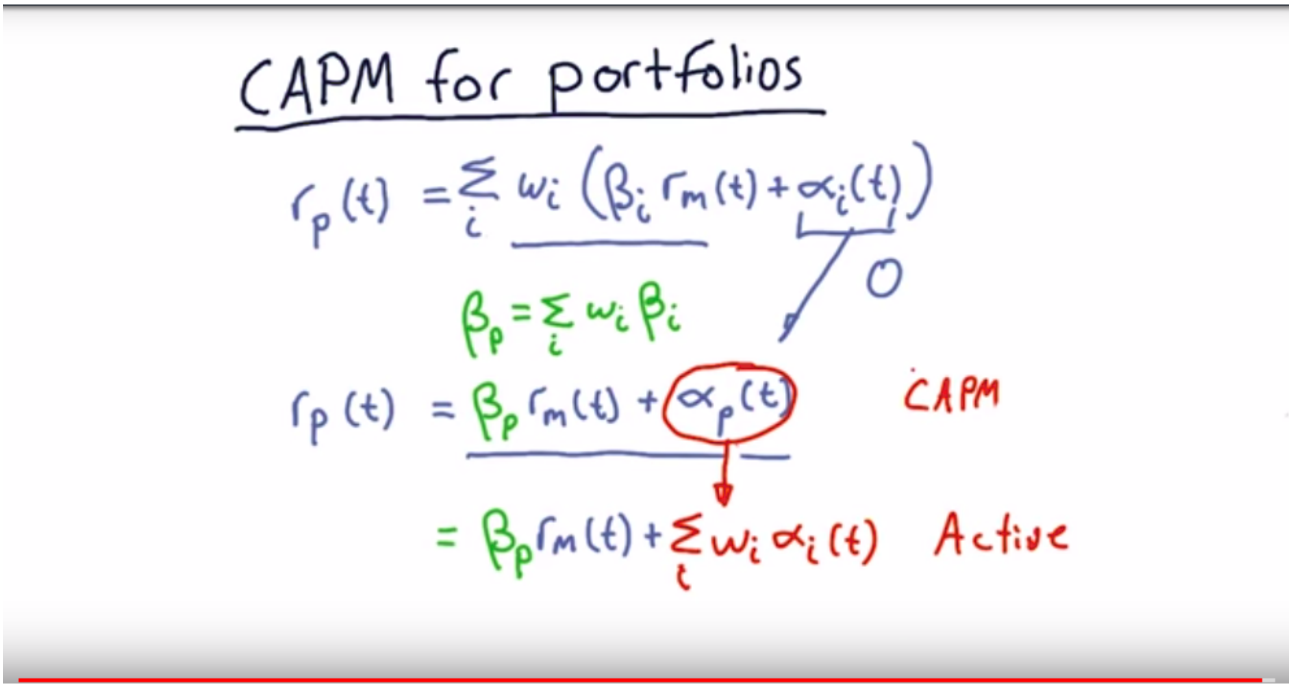
Beta is the slope and alpha is y-intercept



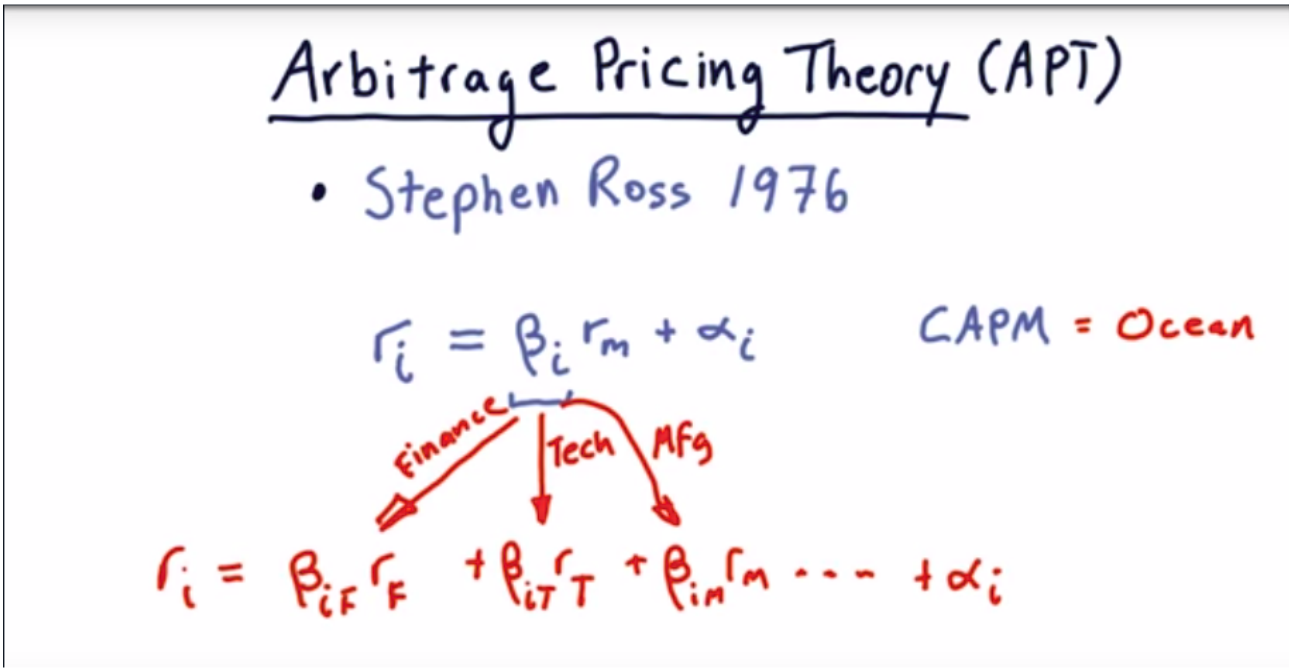
Active Vs Passive investing



CAPM for all portfolios in a particular day

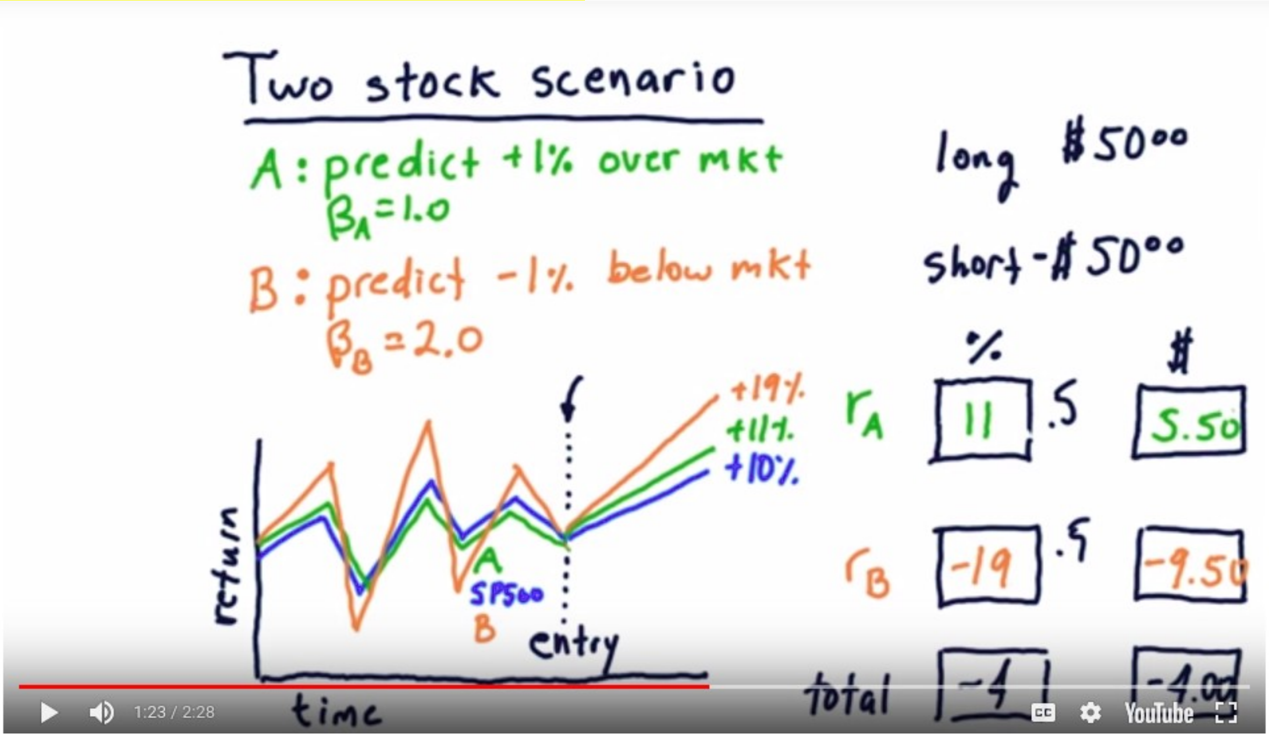


Arbitrage Pricing Theory

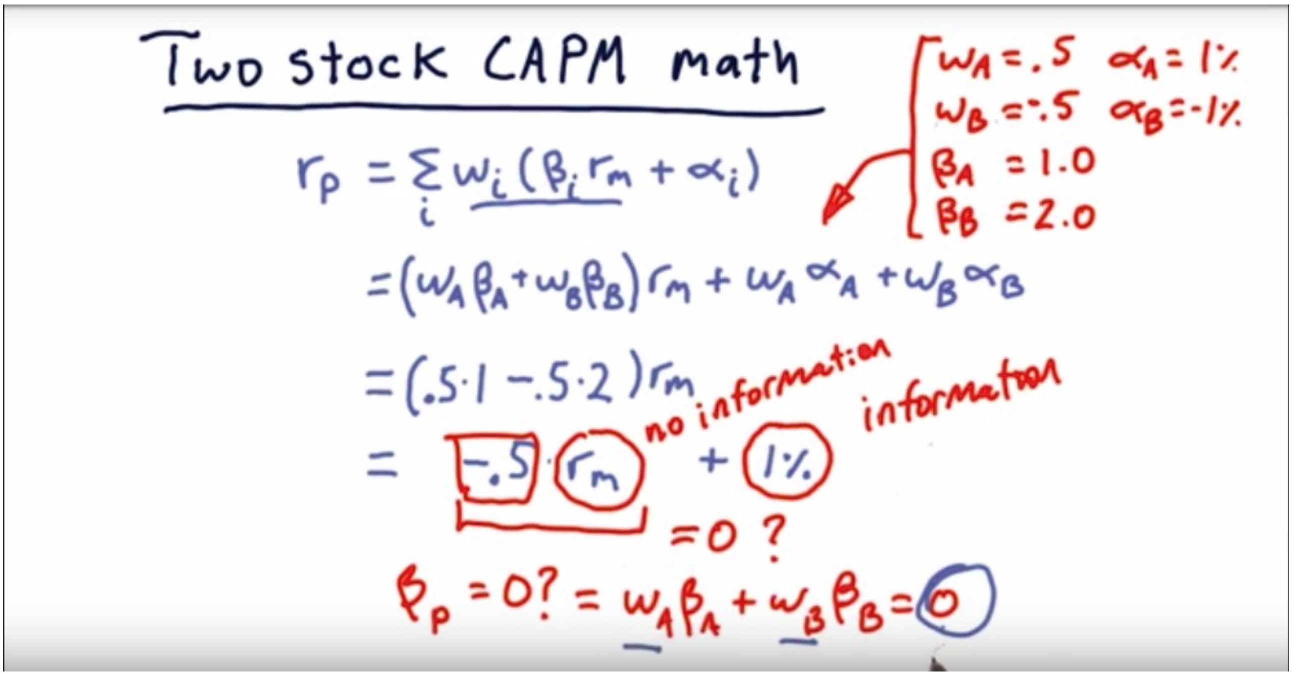


Here Beta(i) is the sum of betas of different sectors (ie sum of islands in the CAPM ocean).

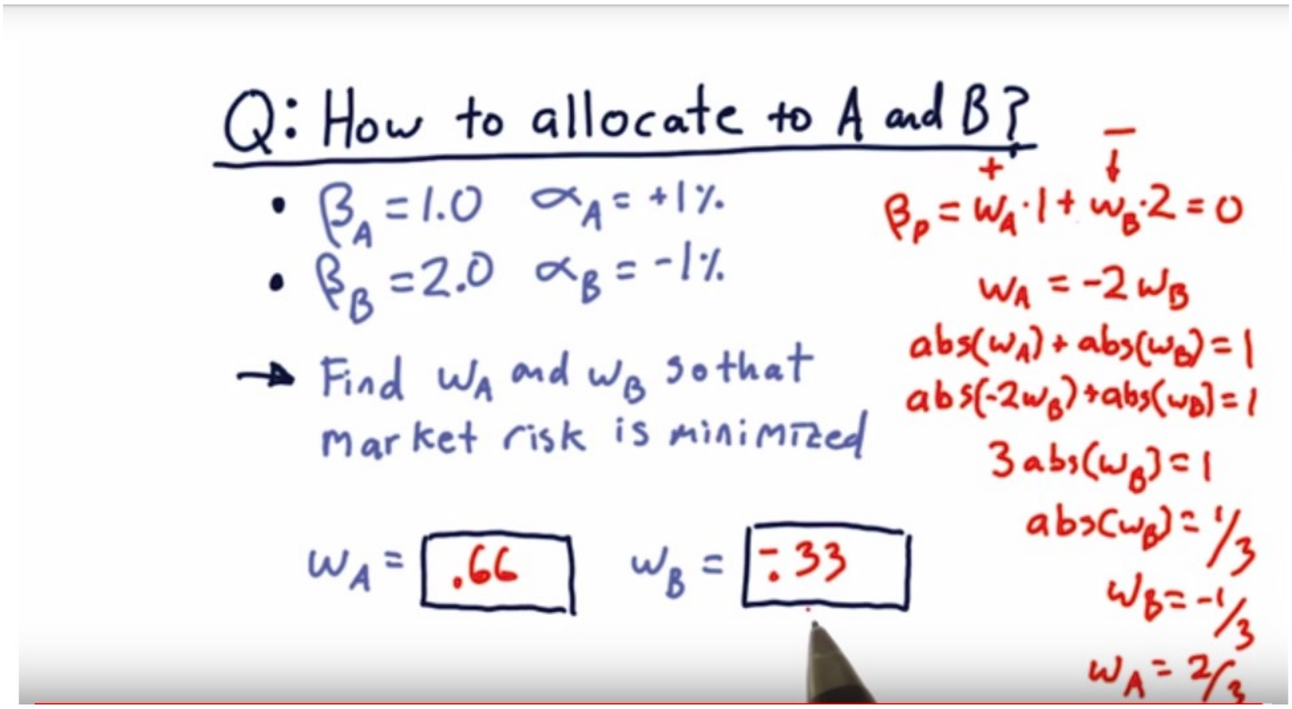
Example on short and long on two stocks



Example on short and long of two stocks using capm method



(same example)Removing market risk



Fundamental Analysis Vs technical analysis

Fundamental Analysis : Here we consider aspects which determine value of company

Technical analysis : Here we consider trends or patterns in price of stock.

Technical analysis

Chars -

1. Only historical price and volume are considered.

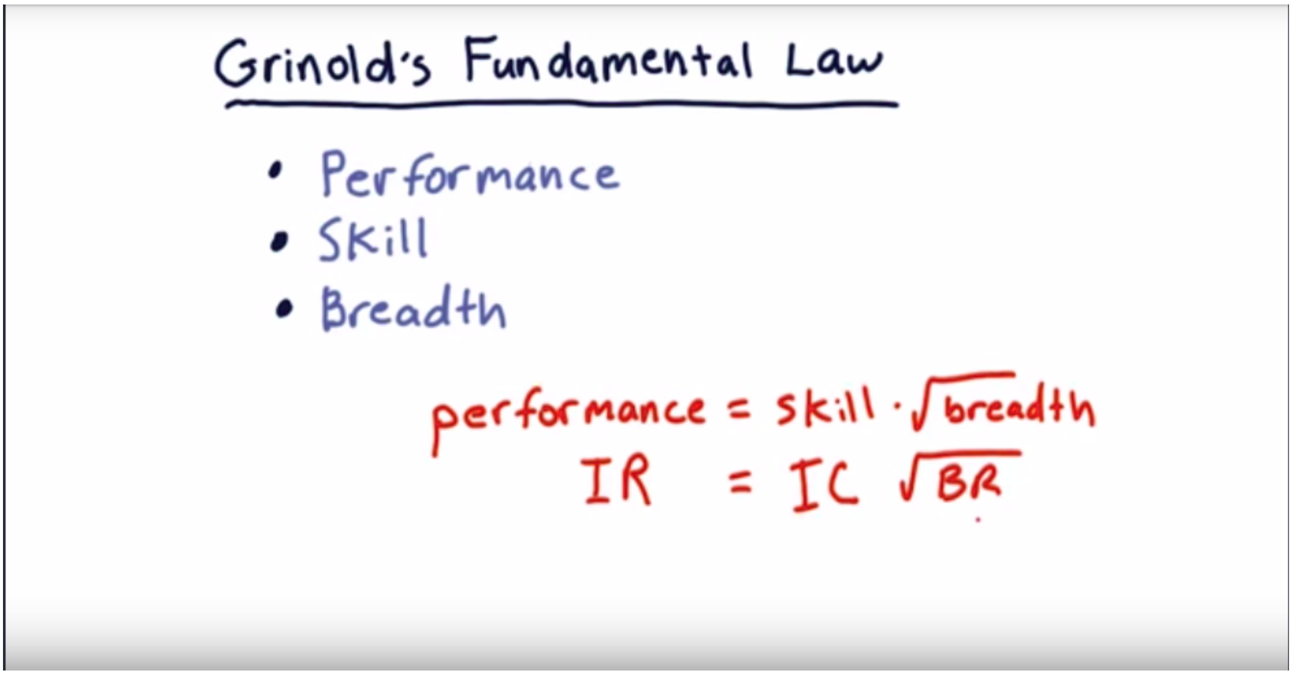
2.Based on this we compute statistics called indicators which indicate a buy/sell opportunity.

Technical analysis indicators

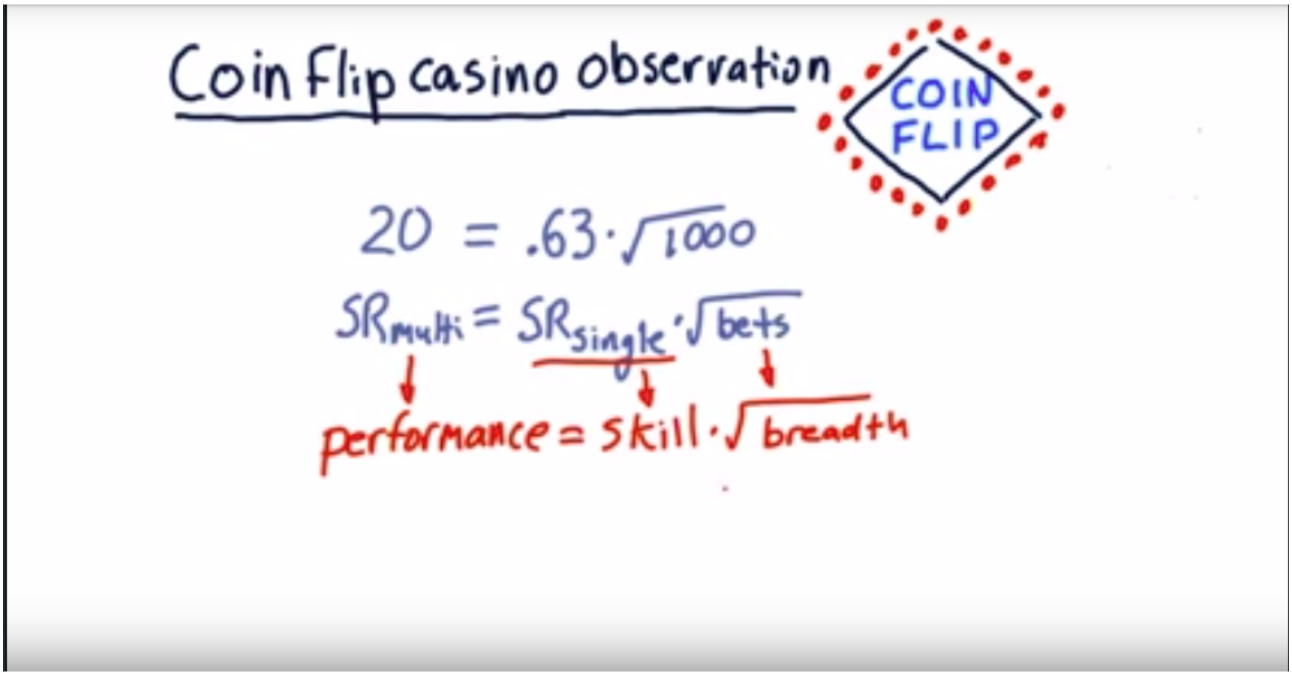
1.Momentum

Fundamental law of active portfolio management

Grinolds fundamental Law

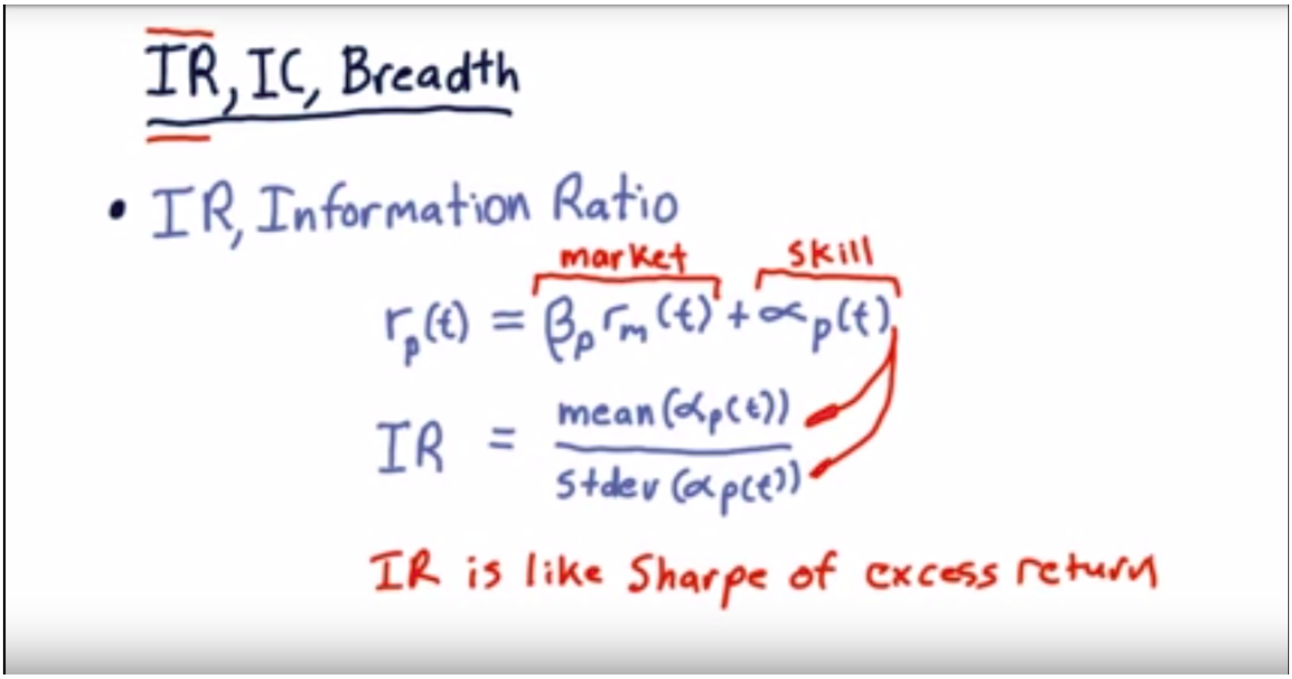


Adjusted risk/reward ratio or sharpe ratio = expected return / risk

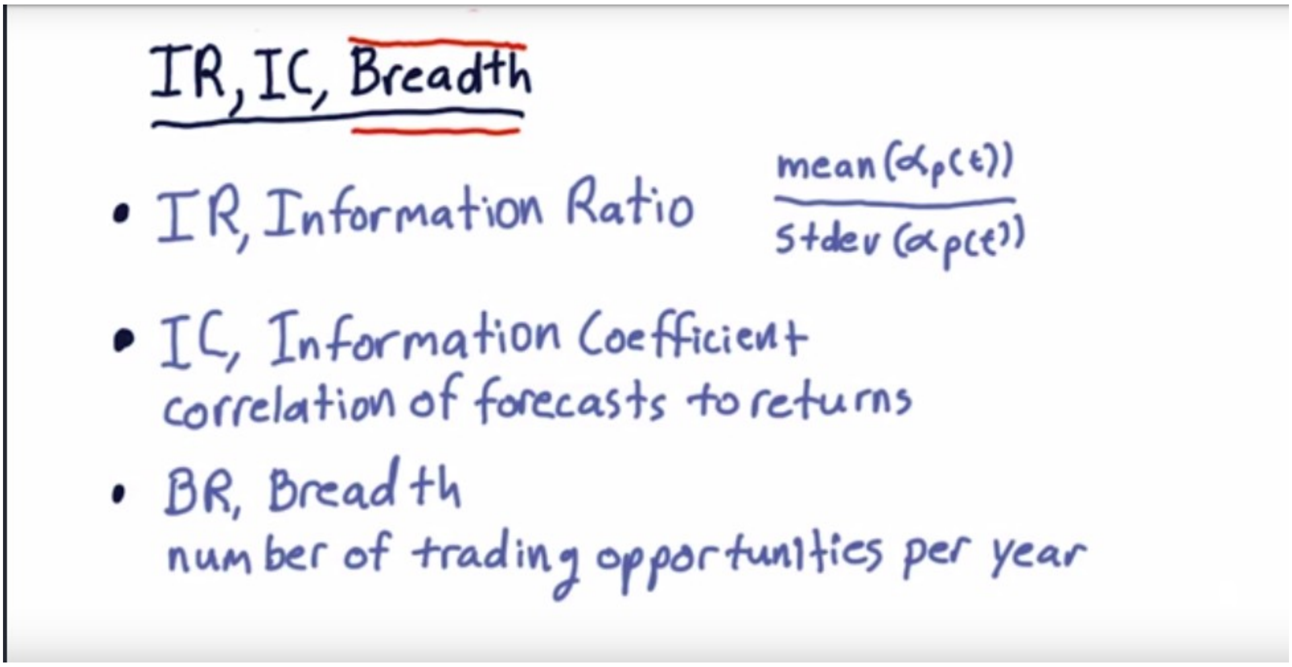


Fundamental terms

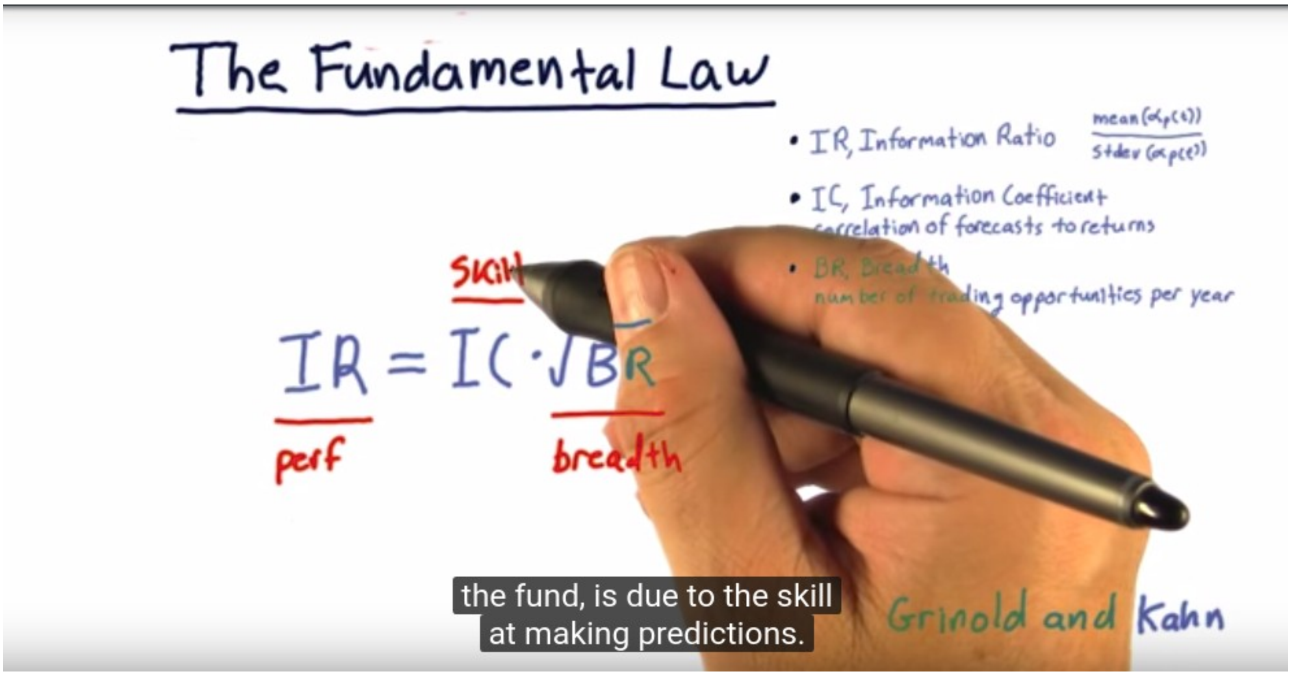
Information ratio



Information coefficient and breadth



Fundamental Law for performance



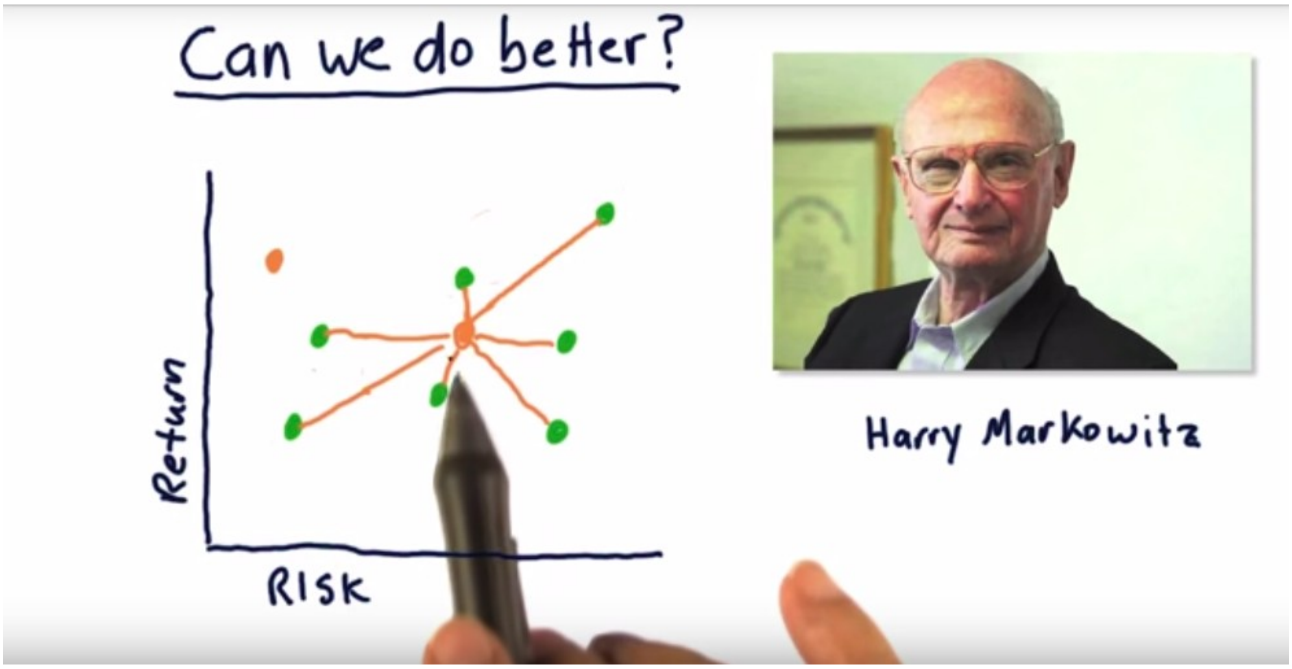
Risk :

(Less volatile stocks have less risk)

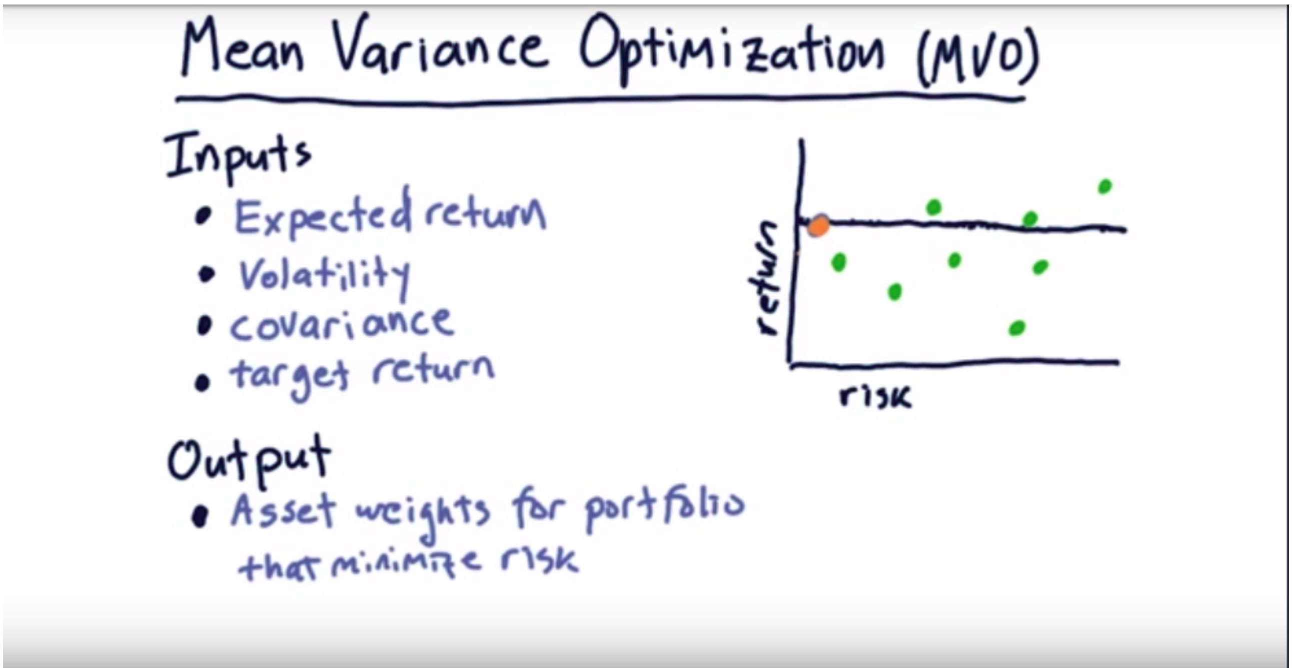
It is the standard deviation of historical daily returns

Can we reduce risk ?

Relationship of stocks in terms of covariance



Mean variance optimization



The efficient Frontier