Determine Beta for CAPM application

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Load Dataset from resident file

In [2]:
```
security_data = pd.read_csv('^GSPC_MSFT Monthly V2 .csv',index_col='Dat
```

In [3]:
```
security_data
```

Out[3]:

Date	^GSPC	MSFT
01/11/2015	2080.409912	49.264046
01/12/2015	2043.939941	50.627274
01/01/2016	1940.239990	50.271385
01/02/2016	1932.229980	46.429623
01/03/2016	2059.739990	50.760994
...	...	...
01/07/2020	3271.120117	203.981583
01/08/2020	3500.310059	224.398651
01/09/2020	3363.000000	209.780777
01/10/2020	3269.959961	201.941315
01/11/2020	3621.629883	213.511017

61 rows × 2 columns

Calculate the returns using the log function and based on differencing method. Check the data is complete for all dates in the series

In [4]:
```
sec_returns =np.log ( security_data/ security_data.shift(1))
```

In [5]:
```
Ret =sec_returns*12
Ret
```

Out[5]:

Date	^GSPC	MSFT
01/11/2015	NaN	NaN
01/12/2015	-0.212228	0.327551
01/01/2016	-0.624811	-0.084653
01/02/2016	-0.049643	-0.953980
01/03/2016	0.766860	1.070287
...	...	...
01/07/2020	0.643642	0.088124
01/08/2020	0.812629	1.144733
01/09/2020	-0.480217	-0.808333
01/10/2020	-0.336668	-0.457031
01/11/2020	1.225757	0.668536

61 rows × 2 columns

Determine the Covariance between the security and the market portoflio. Check date for accuracy and interpret results

In [6]:
```
Ret_secs =  Ret.iloc[1:61] #Index 1 to 61
```

In [7]:
```
Ret_secs.head()
```

Out[7]:

Date	^GSPC	MSFT
01/12/2015	-0.212228	0.327551
01/01/2016	-0.624811	-0.084653
01/02/2016	-0.049643	-0.953980
01/03/2016	0.766860	1.070287
01/04/2016	0.032349	-1.225041

In [8]:
```
cov=Ret_secs.cov()
```

In [9]:
```
cov.round(4)
```

Out[9]:

	^GSPC	MSFT
^GSPC	0.2814	0.2248
MSFT	0.2248	0.3849

CoviM/Varm is the Beta. We apply the ILOC method to the derived covariance of returns. The preceding output is the nominator we require to calculate the beta value

In [10]:
```
#iloc locates the cell that coresponds to the covariance between MSFT a
cov_with_market = cov.iloc[1,0]
```

Note- ILOC is a cell location in a dataframe. Therefore, iloc[1,0] is the second row and first column in the dataframe.

In [11]:
```
cov_with_market.round(4)
```

Out[11]: 0.2248

We require the Variance of the market index to also calculate the BETA. Find this out next.

In [12]:
```
mkt_var= Ret_secs['^GSPC'].var()
```

In [13]:
```
mkt_var
```

Out[13]: 0.28138771172648686

Finally we calculate the Beta value of the stocks by using the formula discussed in class.

In [14]:
```
MSFT_beta= cov_with_market / mkt_var
```

In [15]:
```
MSFT_beta
```

Out[15]: 0.7987772298943374

Interpretation of results: MSFT Beta is Defensive security if beta value is less than 1. Aggressive if higher.

The CAPM Formula ri = rf - (beta(im).(rm-rf))

In the US, we get the 3mth T-bill rate of 0.09% (22nd Dec 2020)

In [16]:
```
RM = Ret_secs['^GSPC'].mean()
RM
```
Out[16]: 0.11087184421142973

In [17]:
```
ri= Ret_secs['MSFT'].mean()
ri
```
Out[17]: 0.29329878165689066

In [18]:
```
rf= 0.0009
```

Determine the Expected return of the security (MSFT_er) using the CAPM

In [19]:
```
#Risk premium is Rm-Rf
RP = RM - rf
RP
```
Out[19]: 0.10997184421142973

In [20]:
```
MSFT_er= rf +MSFT_beta*(RP)
```

In [21]:
```
MSFT_er
```
Out[21]: 0.08874300508557746

In [22]:
```
#8.87% is predicted. Technique can be applied to estimate the return on
```

In [23]:
```
#EXERCISE 2 Apply the Sharpe Ratio in practice
#Sharpe Ratio, S = (Ri-Rf) Vari
```

In [27]:
```
(Ret_secs ['MSFT'].var())**0.5
```
Out[27]: 0.6204211352023237

In [29]:
```
Ret_secs ['MSFT'].std()
```
Out[29]: 0.6204211352023237

In [30]:
```
Sharpe= (MSFT_er - rf) / Ret_secs ['MSFT'].std()
```

In [31]:
```
Sharpe
```
Out[31]: 0.14158609386659796

In [ ]:
```
Sharpe Ratio is used to compare different stocks in portfolios.
```