Lab 3 Normal Distribution

In [78]:

```
import pandas as pd
import matplotlib
import numpy as np
```

1.what is the expected daily rate of return of this stock

2.which stock have higher risk and volatility and daily return and concernt

3.which stock has higher probability of daily return 2% or more

4.which stock has high end pribability of making a loss of 2% or more

In [79]:

```
beml_df = pd.read_csv(r"C:\Users\HOME\Desktop\DSA\Lab3\BEML.csv")
glaxo_df = pd.read_csv(r"C:\Users\HOME\Desktop\DSA\Lab3\GLAXO.csv")

#df.head()
glaxo_df.tail()
```

Out[79]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
1734	2016-12-26	2703.00	2740.00	2677.00	2715.0	2723.50	3953.0	107.15
1735	2016-12-27	2722.95	2725.00	2683.00	2692.0	2701.75	10600.0	286.10
1736	2016-12-28	2701.75	2718.00	2690.00	2698.0	2702.15	6050.0	163.44
1737	2016-12-29	2702.05	2739.00	2691.95	2710.0	2727.90	7649.0	207.87
1738	2016-12-30	2730.00	2740.45	2705.00	2730.0	2729.80	6513.0	177.65

Finding Mean

In [80]:

```
beml_df = beml_df[['Date','Close']]
glaxo_df = glaxo_df[['Date','Close']]
```

In [81]:

```
glaxo_df
```

Out[81]:

	Date	Close
0	2010-01-04	1625.65
1	2010-01-05	1616.80
2	2010-01-06	1638.50
3	2010-01-07	1648.70
4	2010-01-08	1639.80
...	...	...
1734	2016-12-26	2723.50
1735	2016-12-27	2701.75
1736	2016-12-28	2702.15
1737	2016-12-29	2727.90
1738	2016-12-30	2729.80

1739 rows × 2 columns

Converting date to date time index

In [82]:

```
glaxo_df = glaxo_df.set_index(pd.DatetimeIndex(glaxo_df['Date']))
beml_df = beml_df.set_index(pd.DatetimeIndex(beml_df['Date']))
glaxo_df
```

Out[82]:

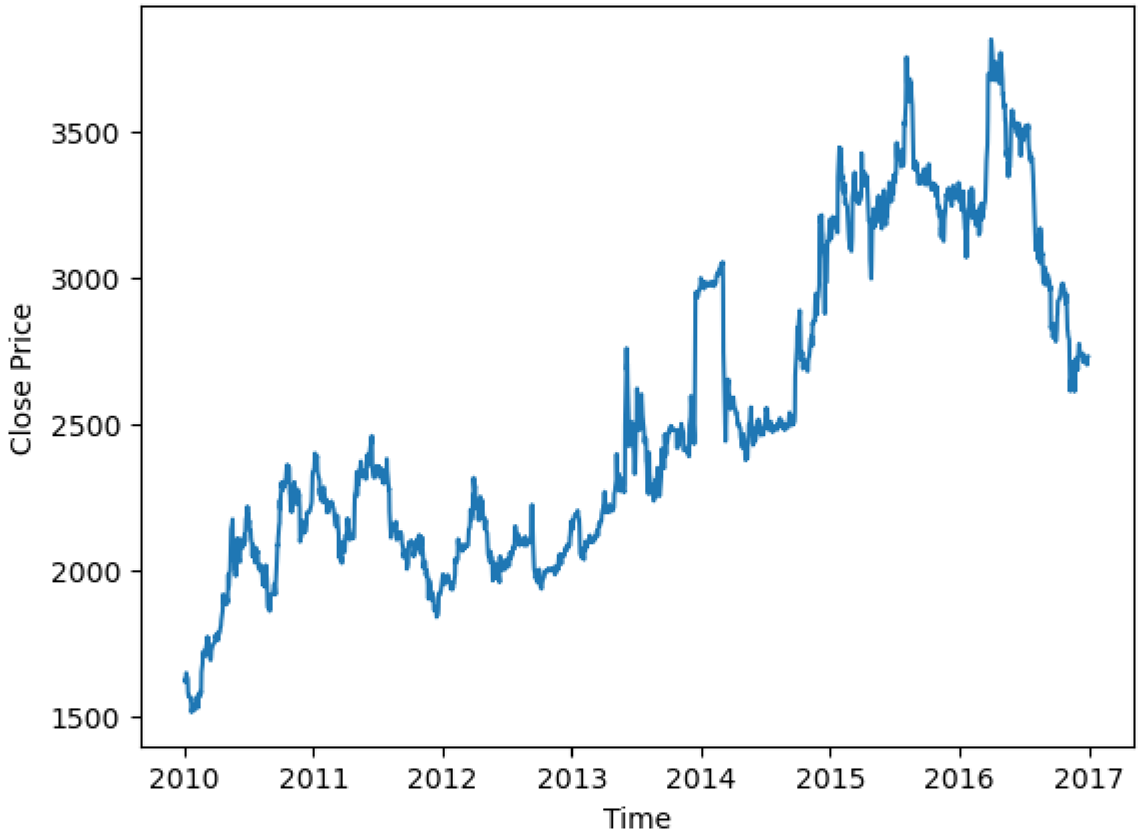
	Date	Close
Date		
2010-01-04	2010-01-04	1625.65
2010-01-05	2010-01-05	1616.80
2010-01-06	2010-01-06	1638.50
2010-01-07	2010-01-07	1648.70
2010-01-08	2010-01-08	1639.80
...	...	...
2016-12-26	2016-12-26	2723.50
2016-12-27	2016-12-27	2701.75
2016-12-28	2016-12-28	2702.15
2016-12-29	2016-12-29	2727.90
2016-12-30	2016-12-30	2729.80

1739 rows × 2 columns

Glaxo plot

In [83]:

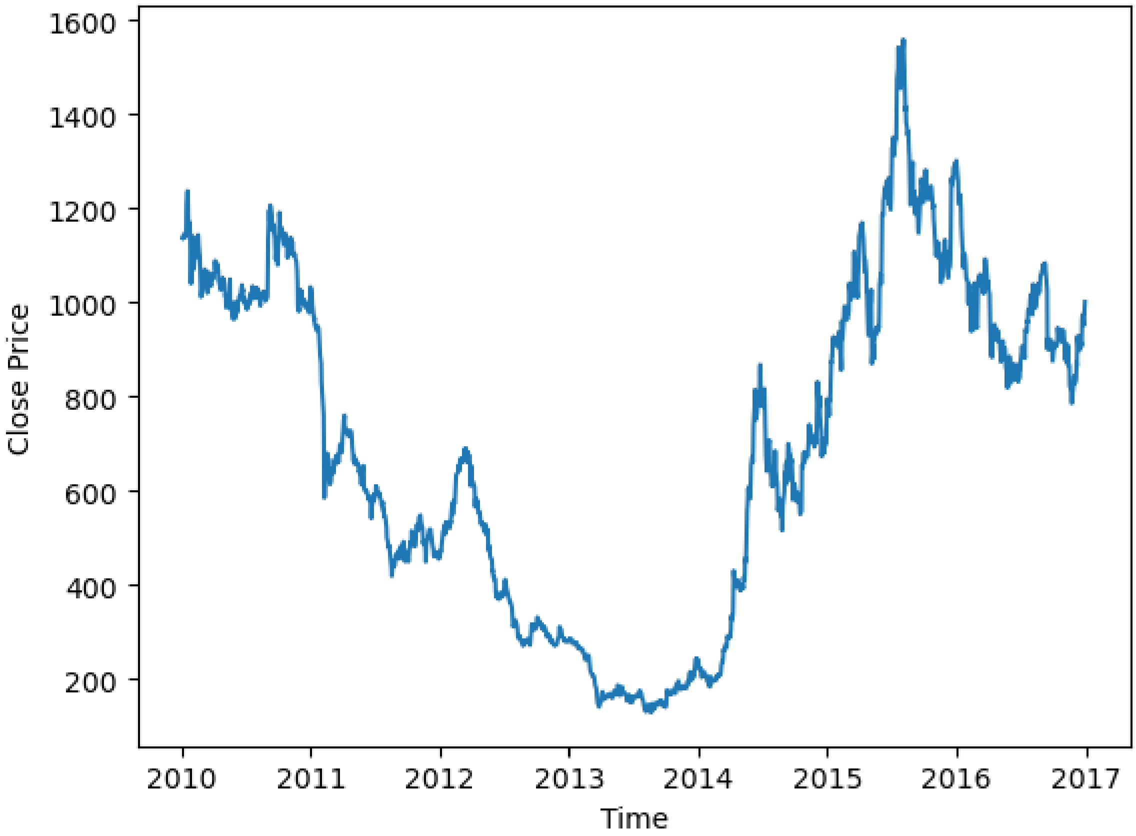
```
import matplotlib.pyplot as plt
import seaborn as sn
%matplotlib inline
plt.plot(glaxo_df.Close);
plt.xlabel('Time');
plt.ylabel('Close Price');
```



BEML plot

In [84]:

```
import matplotlib.pyplot as plt
import seaborn as sn
%matplotlib inline
plt.plot(beml_df.Close);
plt.xlabel('Time');
plt.ylabel('Close Price');
```



In [85]:

```
glaxo_df['gain'] = glaxo_df.Close.pct_change(periods=1)
beml_df['gain'] = beml_df.Close.pct_change(periods=1)
```

In [86]:

```
#dropping first row since null
glaxo_df = glaxo_df.dropna()
beml_df = beml_df.dropna()
glaxo_df
```

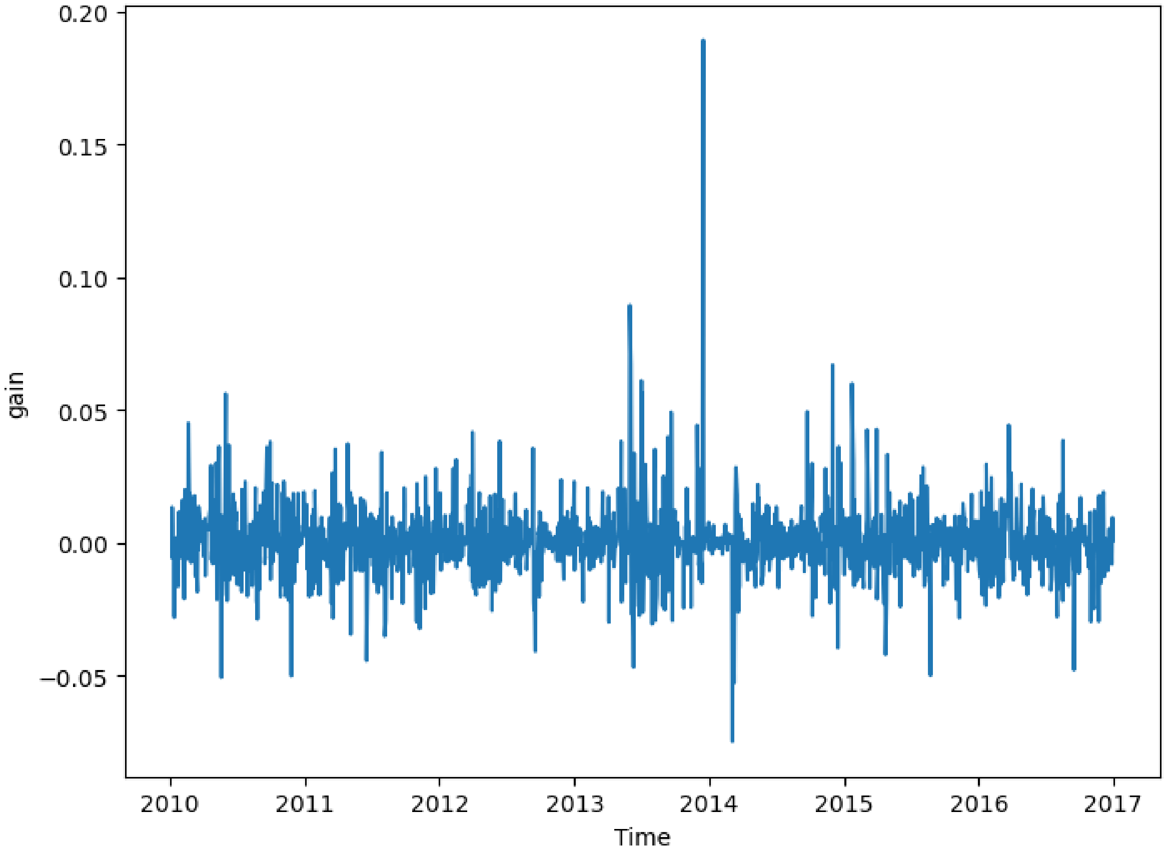
Out[86]:

	Date	Close	gain
Date			
2010-01-05	2010-01-05	1616.80	-0.005444
2010-01-06	2010-01-06	1638.50	0.013422
2010-01-07	2010-01-07	1648.70	0.006225
2010-01-08	2010-01-08	1639.80	-0.005398
2010-01-11	2010-01-11	1629.45	-0.006312
...	...	...	...
2016-12-26	2016-12-26	2723.50	-0.001283
2016-12-27	2016-12-27	2701.75	-0.007986
2016-12-28	2016-12-28	2702.15	0.000148
2016-12-29	2016-12-29	2727.90	0.009529
2016-12-30	2016-12-30	2729.80	0.000697

1738 rows × 3 columns

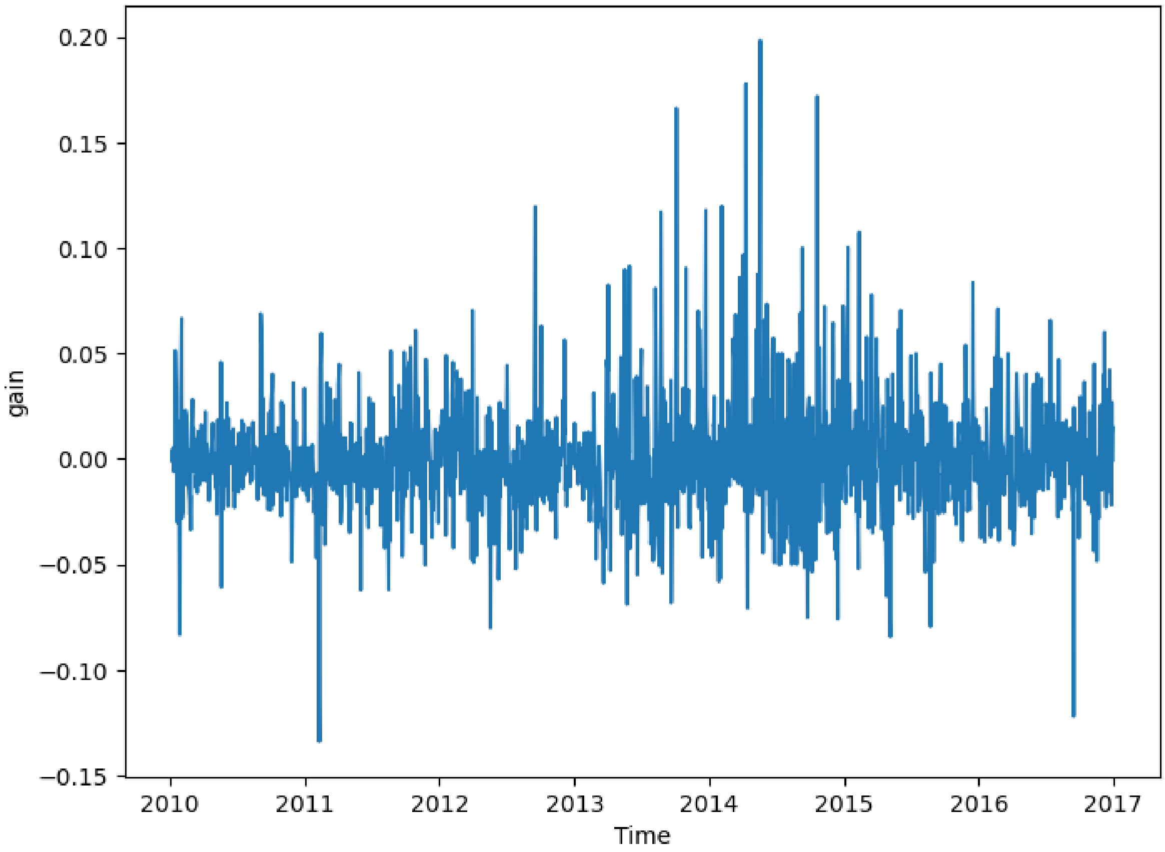
```
In [87]: #plotting gain
         plt.figure(figsize= (8,6));
         plt.plot(glaxo_df.index,glaxo_df.gain);
         plt.xlabel('Time');
         plt.ylabel('gain')

Out[87]: Text(0, 0.5, 'gain')
```



```
In [88]: #plotting gain
         plt.figure(figsize= (8,6));
         plt.plot(beml_df.index,beml_df.gain);
         plt.xlabel('Time');
         plt.ylabel('gain')

Out[88]: Text(0, 0.5, 'gain')
```



```
In [89]: sn.distplot(glaxo_df.gain, label='glaxo')
         plt.xlabel('gain');
         plt.ylabel('Density');
         plt.legend();
```

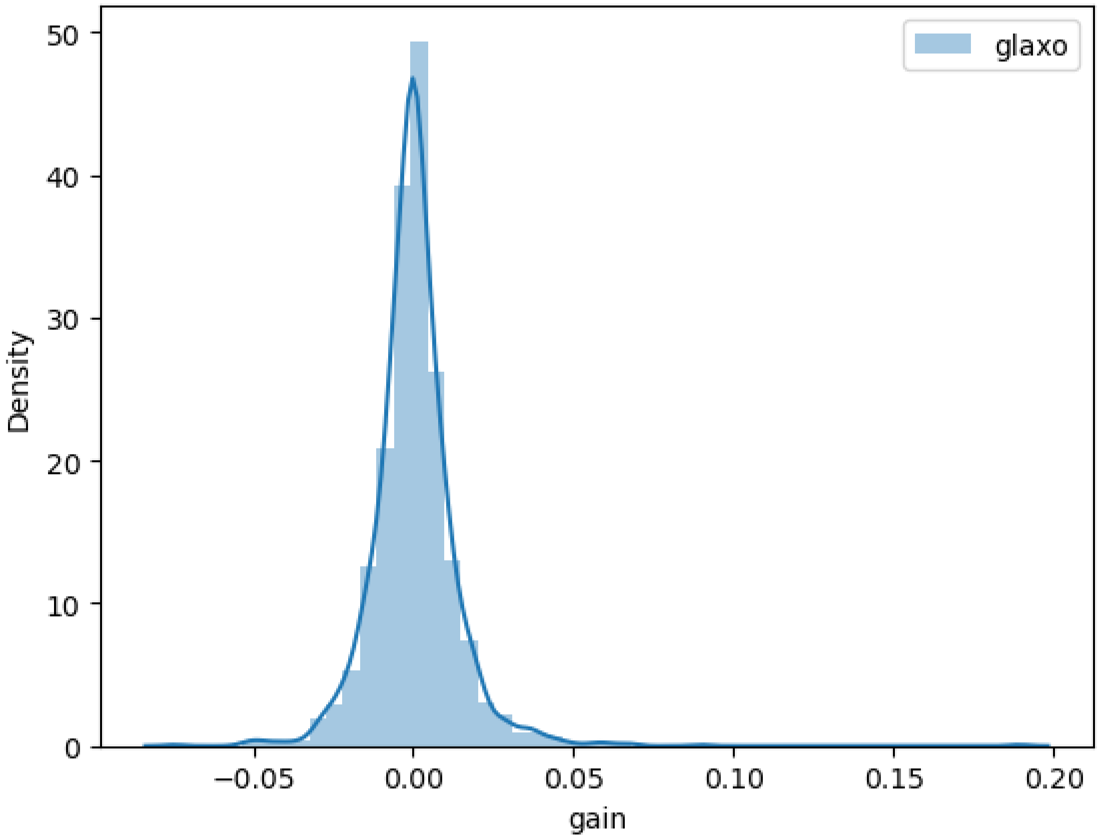
```
C:\Users\HOME\AppData\Local\Temp\ipykernel_1112\1954850185.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sn.distplot(glaxo_df.gain, label='glaxo')
```



In [90]:

```
sn.distplot(beml_df.gain, label='beml')
plt.xlabel('gain');
plt.ylabel('Density');
plt.legend();
```

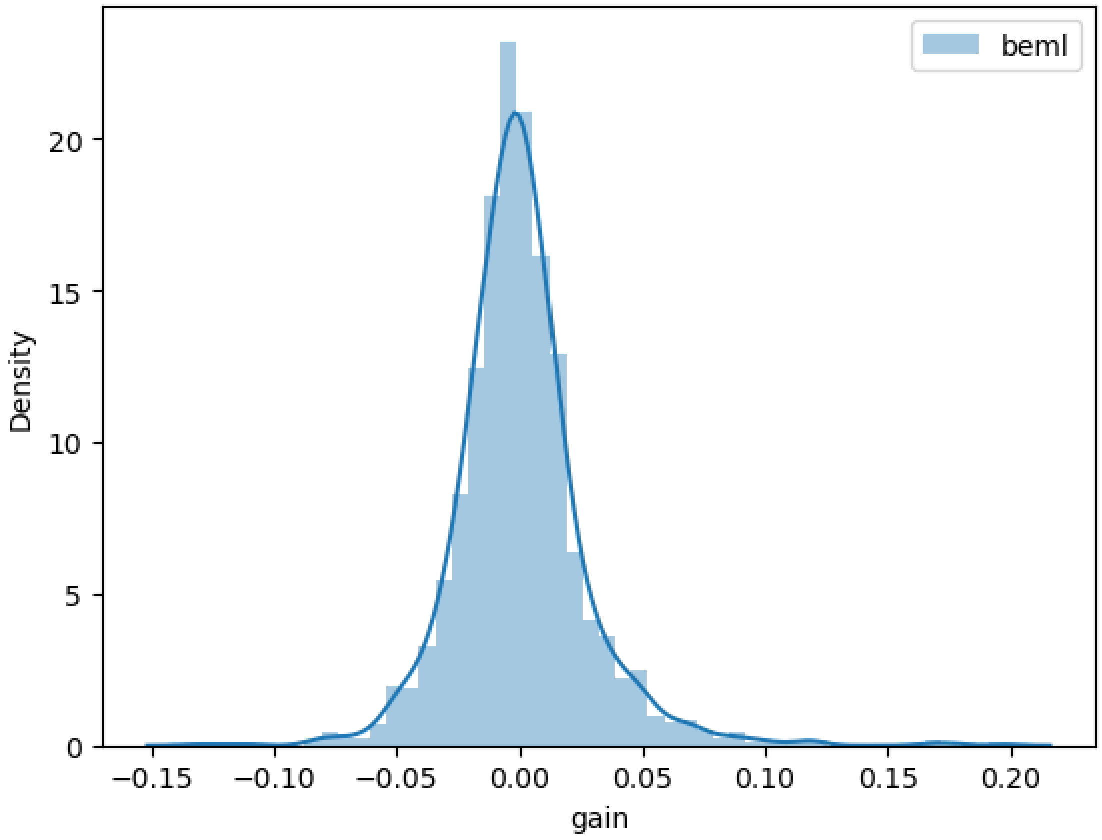
```
C:\Users\HOME\AppData\Local\Temp\ipykernel_1112\2275164977.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gis
t.github.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sn.distplot(beml_df.gain, label='beml')
```



mean and std for glaxo

In [91]:

```
print('Mean', round(glaxo_df.gain.mean(),4))
print('Standard Deviation: ', round(glaxo_df.gain.std(),4))
```

```
Mean 0.0004
Standard Deviation:  0.0134
```

mean and std for beml

In [92]:

```
print('Mean', round(beml_df.gain.mean(),4))
print('Standard Deviation: ', round(beml_df.gain.std(),4))
```

```
Mean 0.0003
Standard Deviation:  0.0264
```

In [93]:

```
from scipy import stats
#probablity of making 2% loss or higher in glaxo
stats.norm.cdf(-0.02,loc=glaxo_df.gain.mean(),scale=glaxo_df.gain.std())
```

Out[93]:

```
0.06352488667177397
```

In [94]:

```
#probablity of making 2% gain or higher in glaxo

1-stats.norm.cdf(0.02,loc=glaxo_df.gain.mean(),scale=glaxo_df.gain.std())
```

Out[94]:

```
0.07104511457618568
```

In [95]:

```
#probablity of making 2% loss or higher in beml
stats.norm.cdf(-0.02,loc=beml_df.gain.mean(),scale=beml_df.gain.std())
```

Out[95]:

```
0.22155987503755287
```

In [96]:

```
#probablity of making 2% gain or higher in beml

1-stats.norm.cdf(0.02,loc=beml_df.gain.mean(),scale=beml_df.gain.std())
```

Out[96]:

```
0.22769829484075355
```