EXPERIMENT 3

AIM: To study and examine probability normal distribution.

SOFTWARE USED: Jupyter Notebook

THEORY:

The normal distribution is a continuous probability distribution characterized by its bell-shaped curve.
It is defined by two parameters: the mean (μ) and the standard deviation (σ).
The probability density function (PDF) of the normal distribution is given by the formula:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Here,
x represents the random variable,
μ is the mean,
σ is the standard deviation, and
π is the mathematical constant pi.

Properties:

- The normal distribution is symmetric about its mean.
- Around 68% of the data falls within one standard deviation of the mean, approximately 95% within two standard deviations, and about 99.7% within three standard deviations (known as the 68-95-99.7 rule).
- It is unimodal, meaning it has a single peak at the mean.
- The mean, median, and mode of a normal distribution are equal.

Standard Normal Distribution:

- A special case of the normal distribution with a mean of 0 and a standard deviation of 1 is known as the standard normal distribution.
- Denoted as Z, random variables from this distribution are commonly denoted as Z~N(0,1).
- It is widely used for statistical calculations and hypothesis testing.

Applications:

- The normal distribution arises naturally in various fields, including natural and social sciences, finance, engineering, and more.
- It is used in statistical inference, hypothesis testing, modeling phenomena with continuous outcomes, and analyzing data in various domains.

Central Limit Theorem:

The normal distribution plays a crucial role in the Central Limit Theorem, which states that the distribution of the sample mean approaches a normal distribution as the sample size increases, regardless of the shape of the population distribution.

Questions:

1. What is the expected daily rate of return of stocks
2. Which stocks have higher risk and volatility as per daily returns are concerned
3. Which stock has a higher probability of making a daily returns of 2% or more
4. Which stock has higher probability of making a loss(risk) of 2% or more

OUTPUT:

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

In [12]:
```
glaxo_df= pd.read_csv(r"C:\Users\Anvika\Downloads\GLAXO.csv")
beml_df= pd.read_csv(r"C:\Users\Anvika\Downloads\BEML.csv")
```

In [13]:
```
beml_df[0:5]
```

Out[13]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2010-01-04	1121.0	1151.00	1121.00	1134.0	1135.60	101651.0	1157.18
1	2010-01-05	1146.8	1149.00	1128.75	1135.0	1134.60	59504.0	676.47
2	2010-01-06	1140.0	1164.25	1130.05	1137.0	1139.60	128908.0	1482.84
3	2010-01-07	1142.0	1159.40	1119.20	1141.0	1144.15	117871.0	1352.98
4	2010-01-08	1156.0	1172.00	1140.00	1141.2	1144.05	170063.0	1971.42

In [15]:
```
glaxo_df.head(5)
```

Out[15]:

	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2010-01-04	1613.00	1629.10	1602.00	1629.0	1625.65	9365.0	151.74
1	2010-01-05	1639.95	1639.95	1611.05	1620.0	1616.80	38148.0	622.58
2	2010-01-06	1618.00	1644.00	1617.00	1639.0	1638.50	36519.0	595.09
3	2010-01-07	1645.00	1654.00	1636.00	1648.0	1648.70	12809.0	211.00
4	2010-01-08	1650.00	1650.00	1626.55	1640.0	1639.80	28035.0	459.11

In [16]:
```
#Answer 1
beml_1= beml_df[['Date', 'Close']]
glaxo_1= glaxo_df[['Date', 'Close']]
```

In [17]:
```
beml_1
```
Out[17]:

	Date	Close
0	2010-01-04	1135.60
1	2010-01-05	1134.60
2	2010-01-06	1139.60
3	2010-01-07	1144.15
4	2010-01-08	1144.05
...	...	...
1734	2016-12-26	950.25
1735	2016-12-27	975.70
1736	2016-12-28	974.40
1737	2016-12-29	986.05
1738	2016-12-30	1000.60

1739 rows × 2 columns

In [18]:
```
glaxo_1
```
Out[18]:

	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

In [19]:
```
glaxo_1=glaxo_1.set_index(pd.DatetimeIndex(glaxo_1['Date']))
beml_1=beml_1.set_index(pd.DatetimeIndex(beml_1['Date']))
```

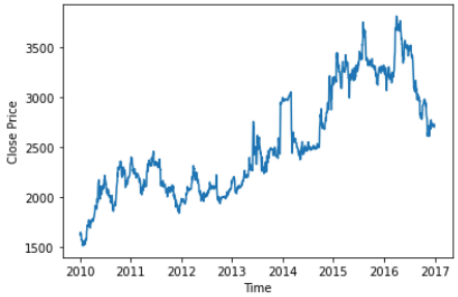
In [20]:
```
glaxo_1
```
Out[20]:

	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

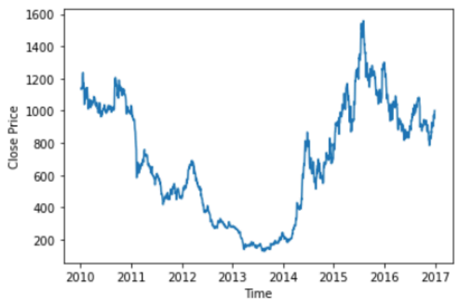
In [22]:

```
%matplotlib inline
plt.plot(glaxo_1.Close);
plt.xlabel('Time');
plt.ylabel('Close Price');
```



In [32]:

```
%matplotlib inline
plt.plot(beml_1.Close);
plt.xlabel('Time');
plt.ylabel('Close Price');
```



In [33]:

```
glaxo_1['Gain']=glaxo_1.Close.pct_change(periods=1)
beml_1['Gain']=beml_1.Close.pct_change(periods=1)
```

In [34]:

```
glaxo_1
```

Out[34]:

	Date	Close	Gain
Date			
2010-01-05	2010-01-05	1616.80	NaN
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

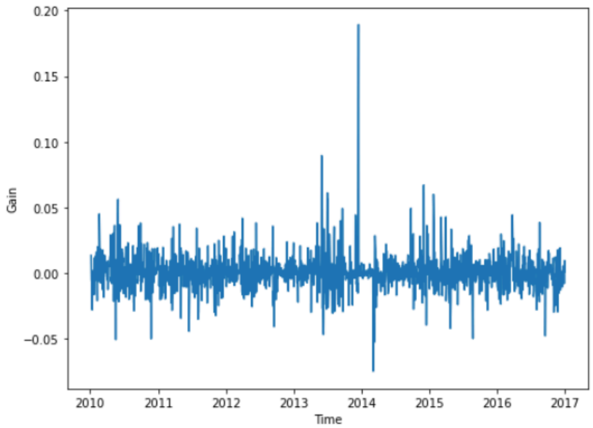
1738 rows × 3 columns

In [35]:

```
#drop first row because it is NA
glaxo_1=glaxo_1.dropna()
beml_1=beml_1.dropna()
```

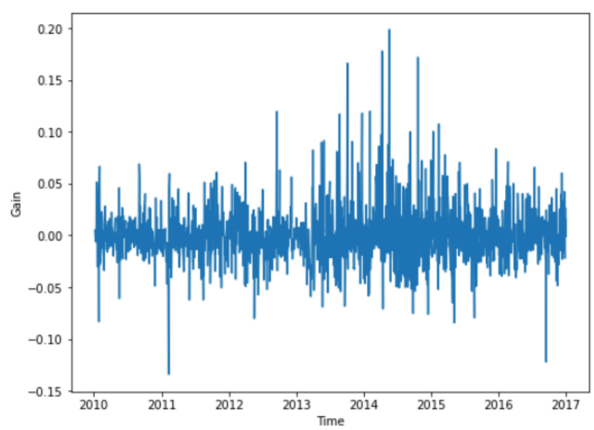
In [36]:

```python
#plot the gains
plt.figure(figsize=(8,6))
plt.plot(glaxo_1.index, glaxo_1.Gain);
plt.xlabel('Time');
plt.ylabel('Gain');
```



In [37]:

```python
plt.figure(figsize=(8,6))
plt.plot(beml_1.index, beml_1.Gain);
plt.xlabel('Time');
plt.ylabel('Gain');
```

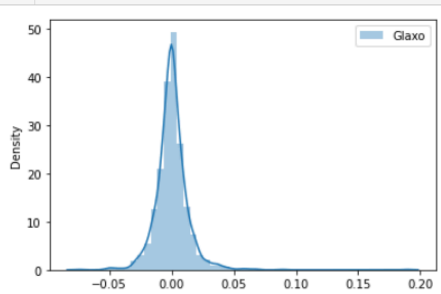


In [ ]:

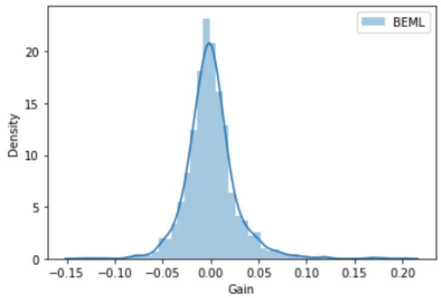
```python
#Answer 2
#BEML is more volatile because above 0 more spikes.
```

In [40]:

```python
sns.distplot(glaxo_1.Gain, label='Glaxo');
plt.xlabel('Gain');
plt.ylabel('Density');
plt.legend();
```



In [41]:
```python
sns.distplot(beml_1.Gain, label='BEML');
plt.xlabel('Gain');
plt.ylabel('Density');
plt.legend();
```



In [42]:
```python
print('Mean:', round(glaxo_1.Gain.mean(),4))
print('Standard Deviation:', round(glaxo_1.Gain.std(),4))
```

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

In [43]:
```python
print('Mean:', round(beml_1.Gain.mean(),4))
print('Standard Deviation:', round(beml_1.Gain.std(),4))
```

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

In [46]:
```python
#Answer 3
#probability of making 2% loss or higher in Glaxo
from scipy import stats
stats.norm.cdf(-0.02,
               loc=glaxo_1.Gain.mean(),
               scale=glaxo_1.Gain.std())
```

Out[46]: `0.06353789851454293`

In [47]:
```python
#probability of making 2% loss or higher in BEML
from scipy import stats
stats.norm.cdf(-0.02,
               loc=beml_1.Gain.mean(),
               scale=beml_1.Gain.std())
```

Out[47]: `0.2216179428118762`

In [49]:
```python
#Answer 4
#probabilty of makinf 2% gain or higher in Glaxo
1-stats.norm.cdf(0.02,
               loc=glaxo_1.Gain.mean(),
               scale=glaxo_1.Gain.std())
```

Out[49]: `0.07112572432274356`

In [50]:
```python
1-stats.norm.cdf(0.02,
               loc=beml_1.Gain.mean(),
               scale=beml_1.Gain.std())
```

Out[50]: `0.2277706340605088`

CONCLUSION:

Probability normal distribution, often referred to as the Gaussian distribution, is a fundamental concept in statistics and probability theory. It describes the distribution of a continuous random variable where data tends to cluster around the mean in a symmetrical, bell-shaped curve. Understanding the properties of this distribution is crucial in various fields such as economics, psychology, and natural sciences. One of the key characteristics of the normal distribution is that it is fully described by two parameters: the mean (μ) and the standard deviation (σ). These parameters determine the center and spread of the distribution, respectively. By analyzing data within the framework of the normal distribution, researchers can make predictions about the likelihood of different outcomes, estimate probabilities, and perform hypothesis testing. Moreover, many statistical methods and models, including linear regression and hypothesis testing, rely on the assumption of normality, making it a cornerstone of statistical analysis.