Probability distribution and data

By

Dr Shaik A Qadeer

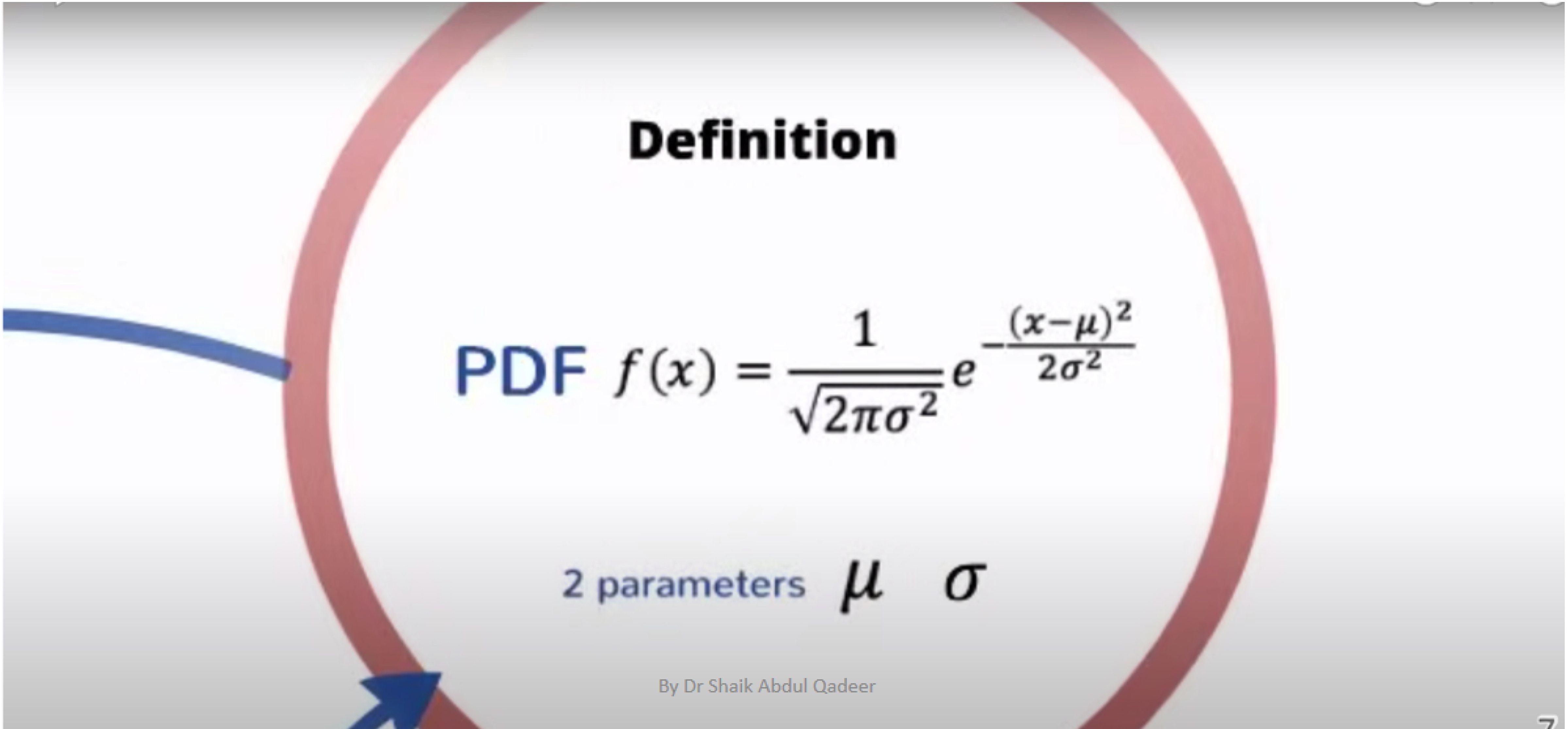
Professor MJCET

Normal distribution: Intro

- Also known as Gaussian distribution
- A continuous distribution
- Normal distribution is observed across many naturally occurring measures like: age, salary, sale volume, birth weight, height, etc.
- Popularly known as bell curve

By Dr Shaik Abdul Qadeer

Normal distribution: Intro.. PDF of it is



Definition

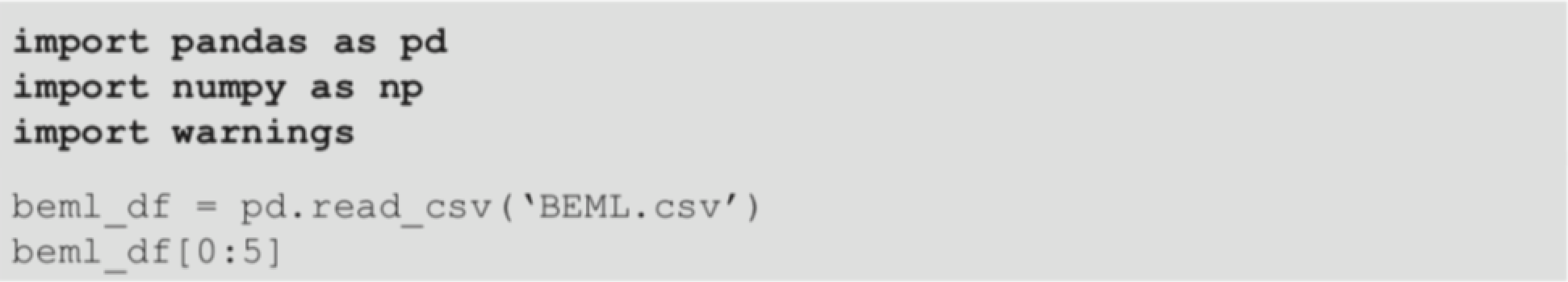
PDF $$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

2 parameters μ σ

Normal distribution: Let us dive into normal distribution with a case study

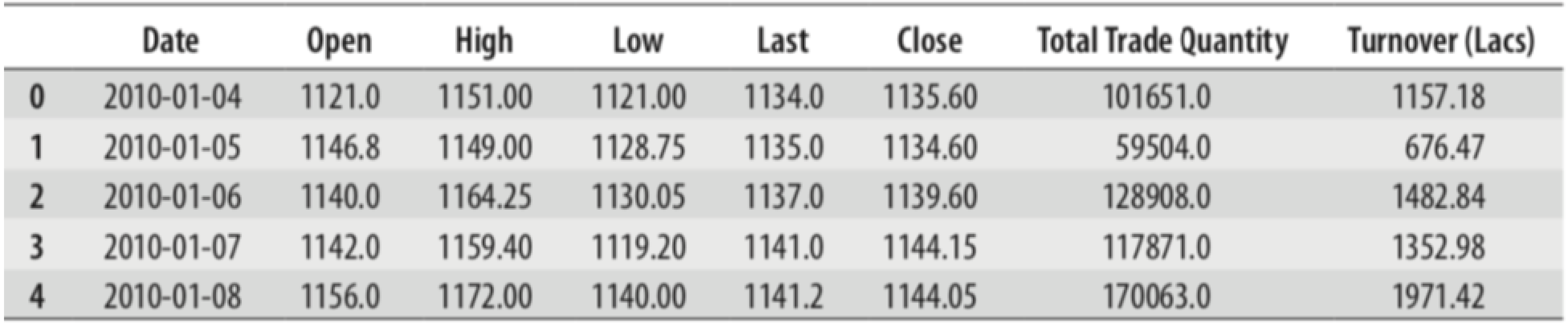
- Imagine a scenario where an investor wants to understand the risks and returns associated with various stocks before investing in them.
- We will evaluate two stocks: BEML and GLAXO.
- The daily trading data for each stock is taken for the period starting from 2010 to 2016 from BSE site.
- Reference: (www.bseindia.com)

Normal distribution..
Solution: loading the data(BEML)



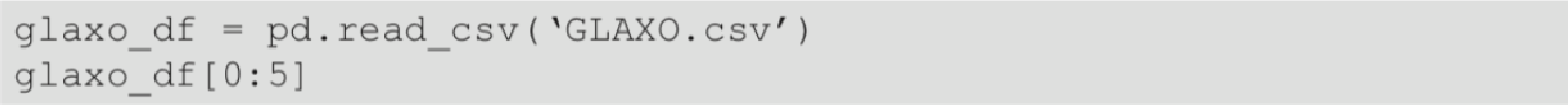
```
import pandas as pd
import numpy as np
import warnings

beml_df = pd.read_csv(‘BEML.csv’)
beml_df[0:5]
```

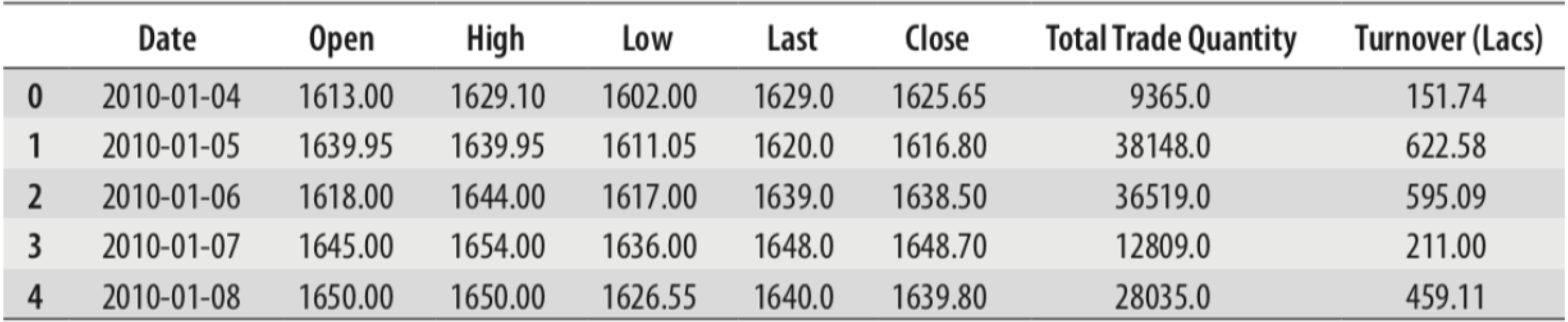


	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2010-01-04	1121.0	1151.00	1121.00	1134.0	1135.60	101651.0	1157.18
1	2010-01-05	1146.8	1149.00	1128.75	1135.0	1134.60	59504.0	676.47
2	2010-01-06	1140.0	1164.25	1130.05	1137.0	1139.60	128908.0	1482.84
3	2010-01-07	1142.0	1159.40	1119.20	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

Normal distribution..
Solution: loading the data(GLAXO)..



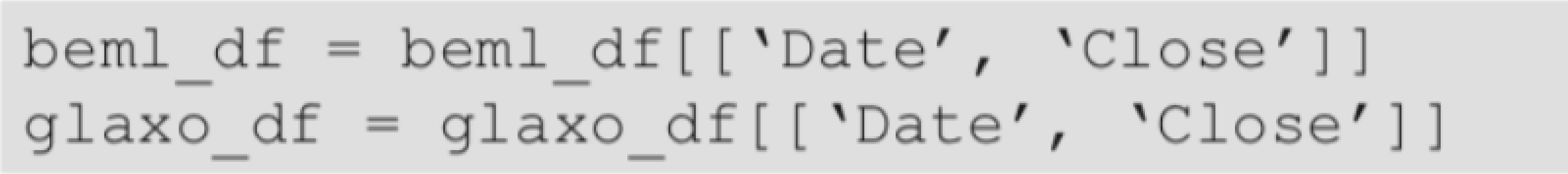
```
glaxo_df = pd.read_csv('GLAXO.csv')
glaxo_df[0:5]
```



	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

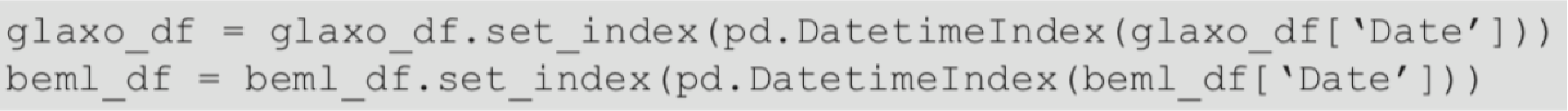
Normal distribution..
Solution:..

- Selecting Date and Close columns from the DataFrames, since the analysis will involve only daily prices.



```
beml_df = beml_df[[‘Date’, ‘Close’]]
glaxo_df = glaxo_df[[‘Date’, ‘Close’]]
```

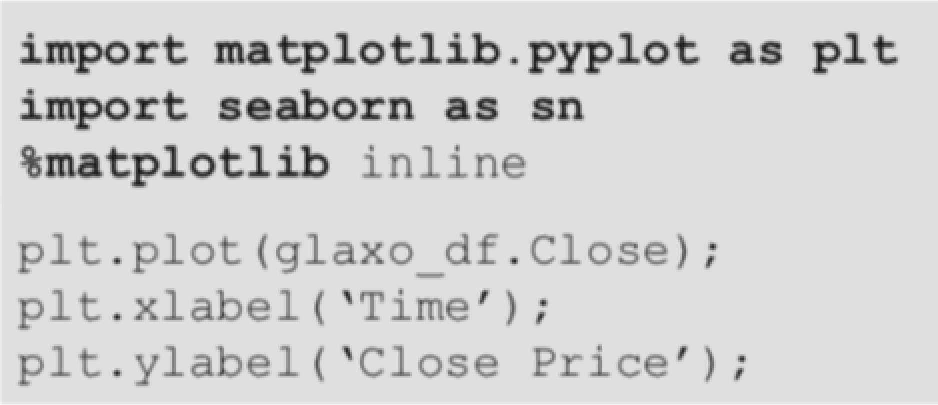
- Setting the Datetime Index



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

Normal distribution..
Solution:..

- Plotting the trend of close prices of GLAXO stock.



```
import matplotlib.pyplot as plt
import seaborn as sn
%matplotlib inline

plt.plot(glaxo_df.Close);
plt.xlabel('Time');
plt.ylabel('Close Price');
```

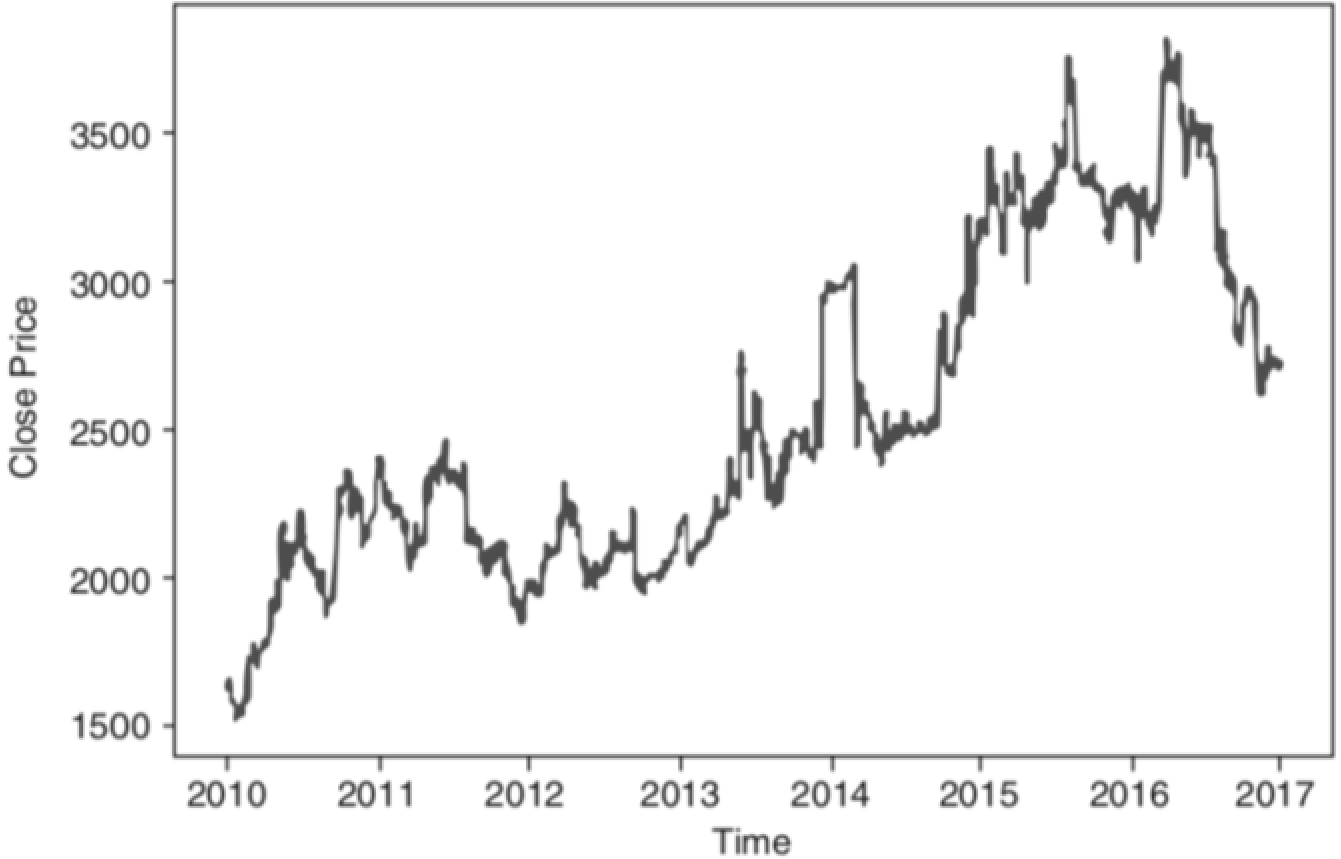
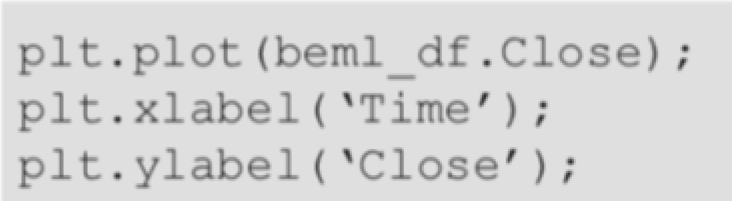


FIGURE 3.4 Close price trends of GLAXO stock.

Normal distribution..
Solution:..

- Plotting the trend of close prices of BEML stock.



```
plt.plot(beml_df.Close);
plt.xlabel('Time');
plt.ylabel('Close');
```

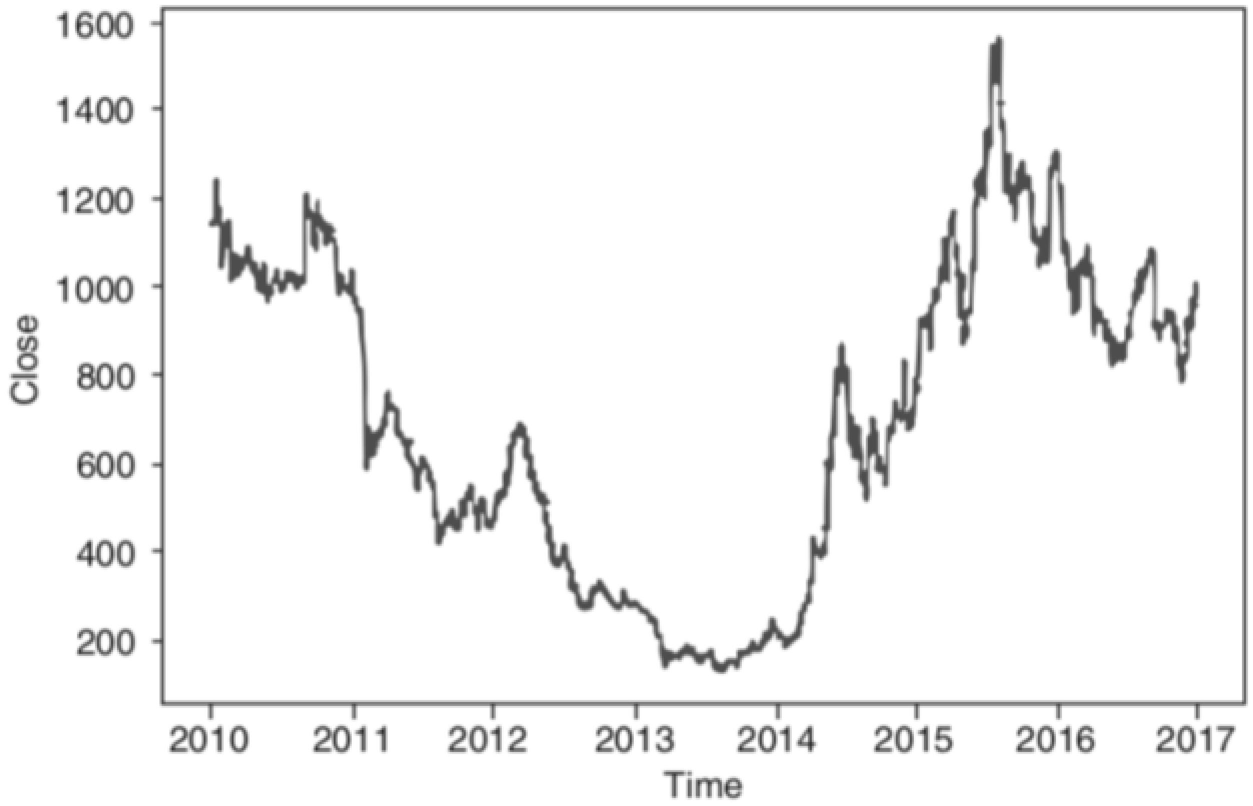


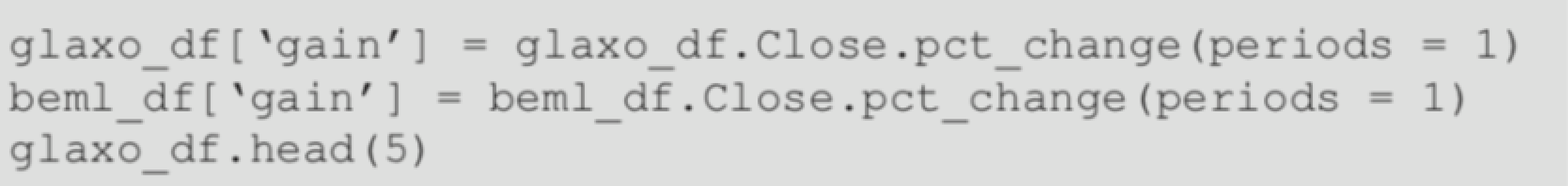
FIGURE 3.5 Close price trends of BEML stock.

ND Solution:..

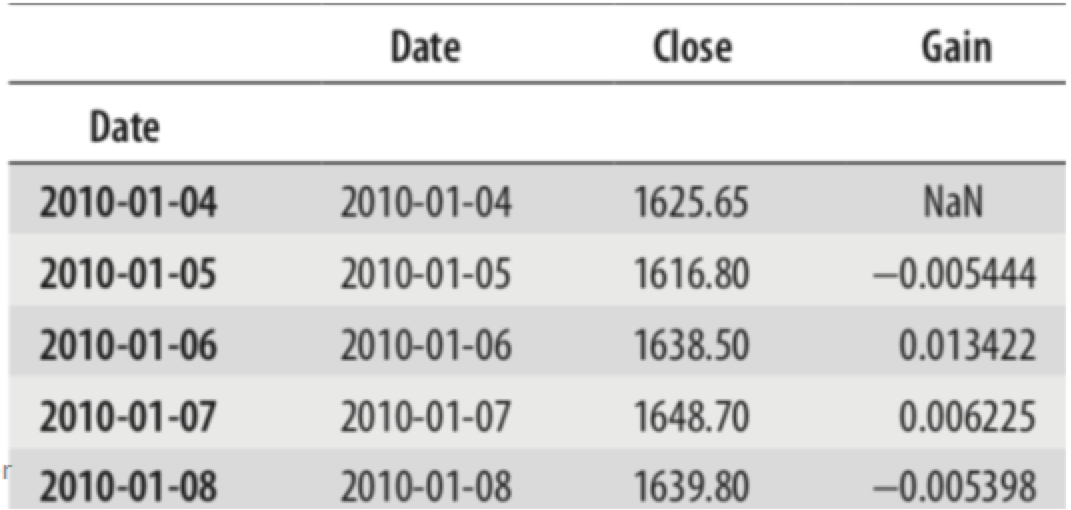
- The behavior of daily returns on the stocks is called Gain.

$$gain = \frac{ClosePrice_t - ClosePrice_{t-1}}{ClosePrice_{t-1}}$$

- In Pandas it can be calculated as



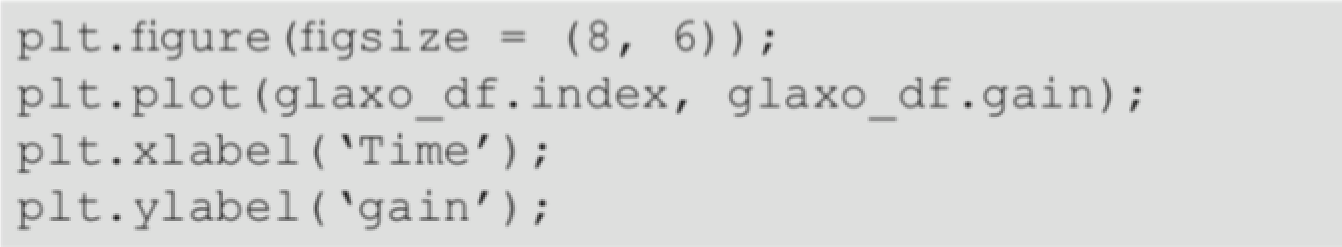
```
glaxo_df['gain'] = glaxo_df.Close.pct_change(periods = 1)
beml_df['gain'] = beml_df.Close.pct_change(periods = 1)
glaxo_df.head(5)
```



	Date	Close	Gain
Date			
2010-01-04	2010-01-04	1625.65	NaN
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

ND Solution:..

- Plotting gain against time



```
plt.figure(figsize = (8, 6));
plt.plot(glaxo_df.index, glaxo_df.gain);
plt.xlabel(‘Time’);
plt.ylabel(‘gain’);
```

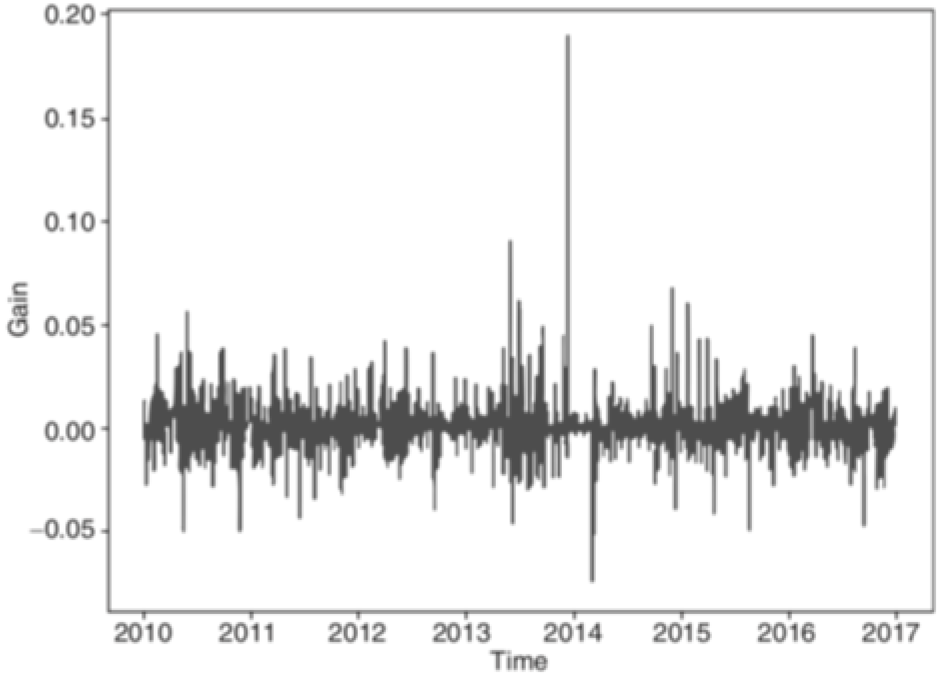
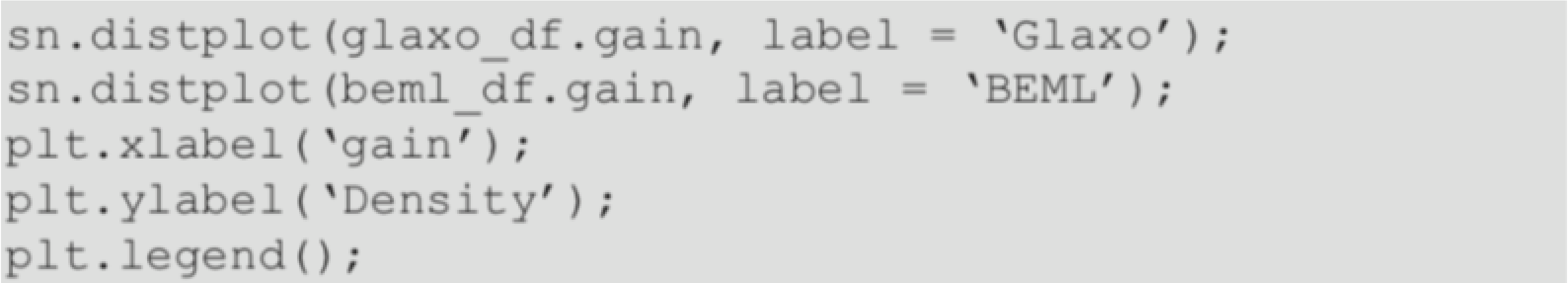


Figure: Daily gain of Glaxo stock

ND Solution:..

- Distribution plot of gain for both BEML and GLAXO stocks



```
sn.distplot(glaxo_df.gain, label = ‘Glaxo’);
sn.distplot(beml_df.gain, label = ‘BEML’);
plt.xlabel(‘gain’);
plt.ylabel(‘Density’);
plt.legend();
```

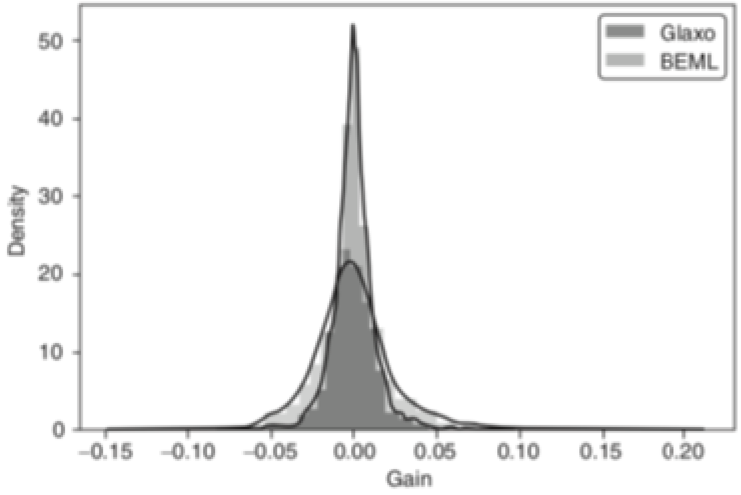


FIGURE 3.7 Distribution plot of daily gain of BEML and Glaxo stocks.

- Gain seems to be normally distributed for both the stocks with a mean around 0.00
- BEML seems to have a higher variance than GLAXO

ND Solution:..

- The sample mean of a normal distribution is given by

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

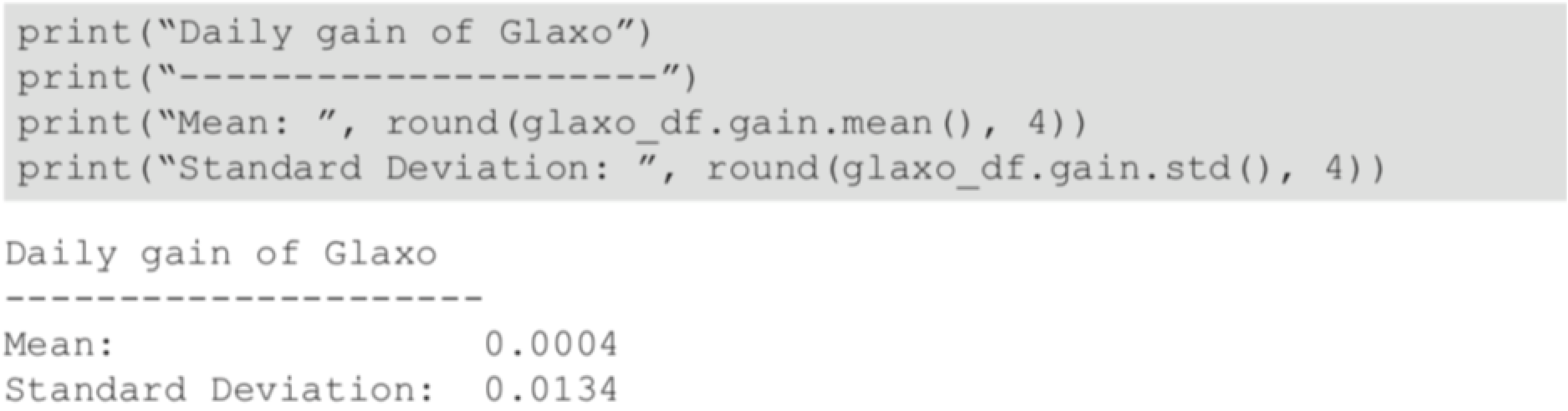
- Variance is given by

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2$$

ND Solution:..

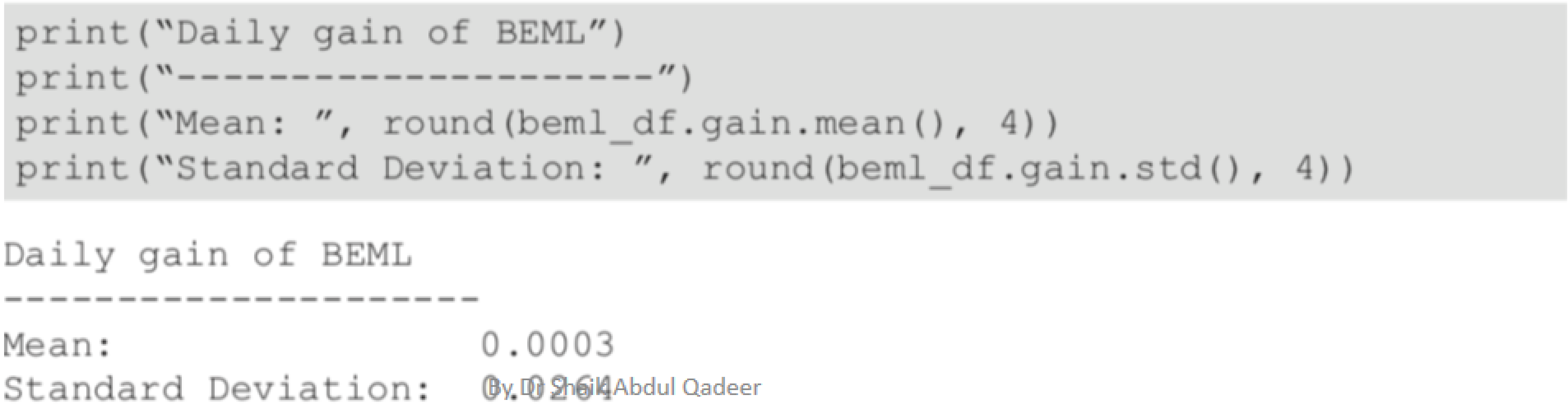
- In Pandas, the sample mean and standard deviation for daily returns

for GLAXO and BEML are



```
print(“Daily gain of Glaxo”)
print(“---------------------”)
print(“Mean: ”, round(glaxo_df.gain.mean(), 4))
print(“Standard Deviation: ”, round(glaxo_df.gain.std(), 4))
```

```
Daily gain of Glaxo
---------------------
Mean:                    0.0004
Standard Deviation:      0.0134
```

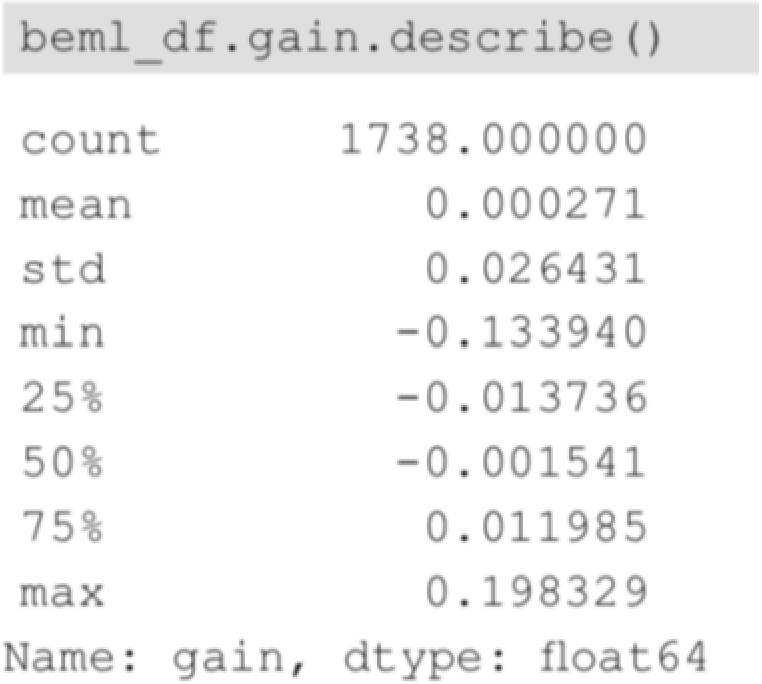


```
print(“Daily gain of BEML”)
print(“---------------------”)
print(“Mean: ”, round(beml_df.gain.mean(), 4))
print(“Standard Deviation: ”, round(beml_df.gain.std(), 4))
```

```
Daily gain of BEML
---------------------
Mean:                    0.0003
Standard Deviation:      0.0264
```

ND Solution:..

- The describe() method of DataFrame returns the detailed statistical summary of a variable



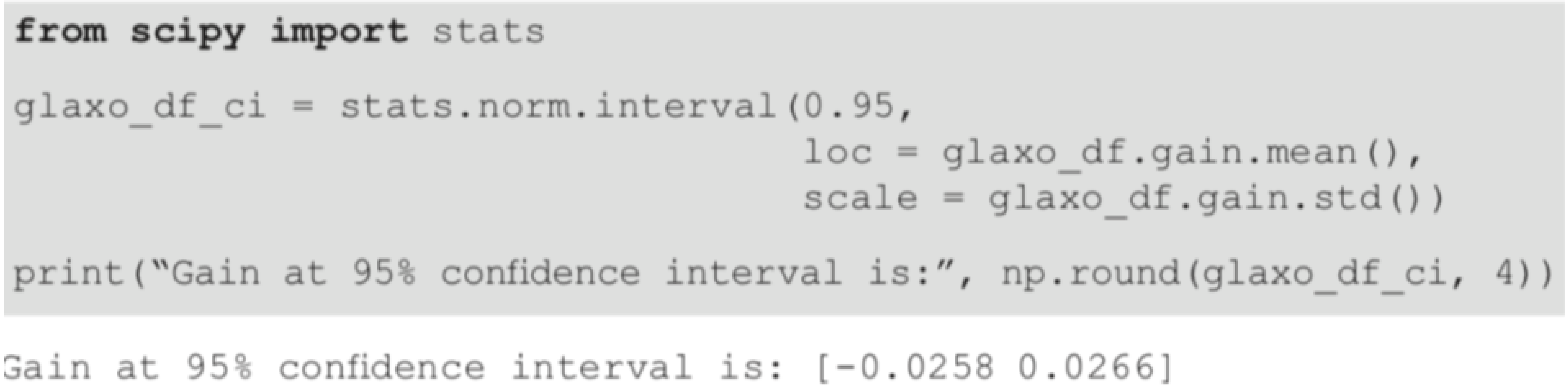
```
beml_df.gain.describe()
```

```
count    1738.000000
mean        0.000271
std         0.026431
min        -0.133940
25%        -0.013736
50%        -0.001541
75%         0.011985
max         0.198329
Name: gain, dtype: float64
```

- BEML stock has higher risk as standard deviation of BEML is 2.64% whereas the standard deviation for GLAXO is 1.33%

ND Solution:..

- Gain at confidence interval 95% for a GLAXO stocks is given as



```
from scipy import stats

glaxo_df_ci = stats.norm.interval(0.95,
                                  loc = glaxo_df.gain.mean(),
                                  scale = glaxo_df.gain.std())

print(“Gain at 95% confidence interval is:”, np.round(glaxo_df_ci, 4))
```

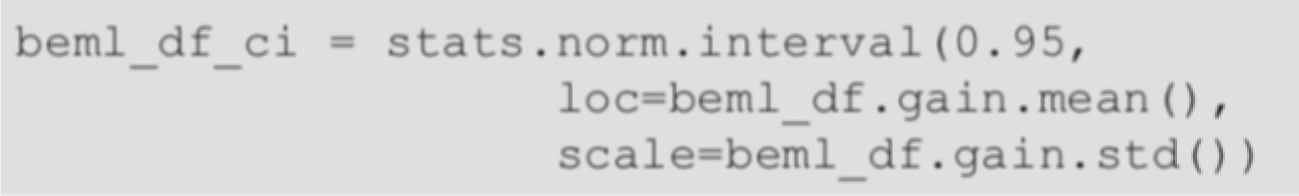
```
Gain at 95% confidence interval is: [-0.0258 0.0266]
```

Stats.norm.interval() takes three parameters and return gain at given interval

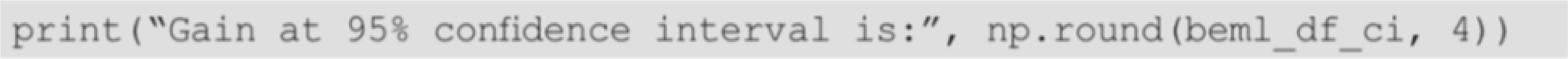
- **Alpha:** it is the interval
- **Loc:** location parameter, i.e. mean for normal distribution
- **Scale:** Scale parameter, i.e. standard deviation for normal distribution.

ND Solution:..

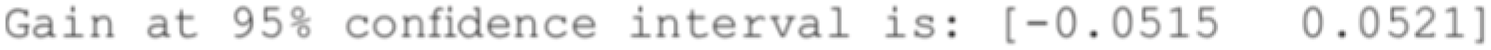
- Gain at confidence interval 95% for a GLAXO stocks is given as



```
beml_df_ci = stats.norm.interval(0.95,
                                 loc=beml_df.gain.mean(),
                                 scale=beml_df.gain.std())
```



```
print(“Gain at 95% confidence interval is:”, np.round(beml_df_ci, 4))
```



```
Gain at 95% confidence interval is: [-0.0515  0.0521]
```

- For 95% confidence interval, gain of GLAXO remains between -2.58% and 2.66% whereas gain of BEML remains between -5.15% and 5.21%.

- Thanks