**Group No.: Date:**

**EXPERIMENT NO. 3**

**AIM**: To understand various concepts about Normal Distribution in python .

**SOFTWARE USED**: Jupyter Notebook

**THEORY**:

The normal distribution, also known as the Gaussian distribution, is a very common probability distribution used in statistics. It's characterized by a bell-shaped curve and is described by two parameters: the mean (μ) and the standard deviation (σ). In Python, you can use libraries like NumPy and Matplotlib to work with and visualize normal distributions. is one of the most fundamental concepts in statistics and probability theory. It is widely used in various fields, including natural sciences, social sciences, finance, and engineering. In this comprehensive guide, we will explore the normal distribution in Python using Jupyter Notebooks, focusing on generating random samples, visualizing the distribution, computing statistics, and understanding the probability density function (PDF).

* We import NumPy as np and Matplotlib.pyplot as plt.
* We set the parameters for the normal distribution, mean (μ) and standard deviation (σ). \* We generate random samples from a normal distribution using np.random.normal().
* We plot the histogram of the generated samples using plt.hist(), which shows the distribution of the data.
* We calculate and plot the probability density function (PDF) of the normal distribution using the formula (1 / (sigma \* np.sqrt(2 \* np.pi))) \* np.exp(-(x - mu)\*2 / (2 \* sigma\*2)).
* We add labels and title to the plot.
* Finally, we display the plot using plt.show().

# Import Libraries :

Start by importing the necessary libraries. For working with the normal distribution, you'll typically need NumPy for numerical computations and Matplotlib for visualization.

# Generate Random Samples:

Use NumPy to generate random samples from a normal distribution with specified mean (μ) and standard deviation (σ)

# Plot Histogram ;

Visualize the distribution of the generated samples using a histogram.

# Compute Statistics:

Calculate statistics such as mean and standard deviation of the generated samples using NumPy functions.

# Plot Probability Density Function (PDF):

Compute and plot the probability density function (PDF) of the normal distribution.

# Methodology for Experimenting with the Normal Distribution in Python:

1. **Setting up the Environment:**

* Ensure that you have the necessary libraries installed, including NumPy and Matplotlib.
* Import the required libraries at the beginning of your notebook/script.

# Define Parameters:

* Decide on the parameters for the normal distribution, including mean (μ) and standard deviation (σ).

# Generating Random Samples:

* Specify the mean, standard deviation, and the desired sample size as parameters to the function.

# Visualizing the Distribution:

* Create a histogram to visualize the distribution of the generated samples.

# Computing Statistics:

* Calculate statistical measures such as mean and standard deviation using NumPy functions.

# Understanding the Probability Density Function (PDF):

* + Compute the probability density function (PDF) of the normal distribution using the formula.
  + Use the PDF formula to compute corresponding probabilities for each x value.

# Plotting the PDF:

* + Plot the PDF curve using Matplotlib to visualize the shape of the normal distribution.

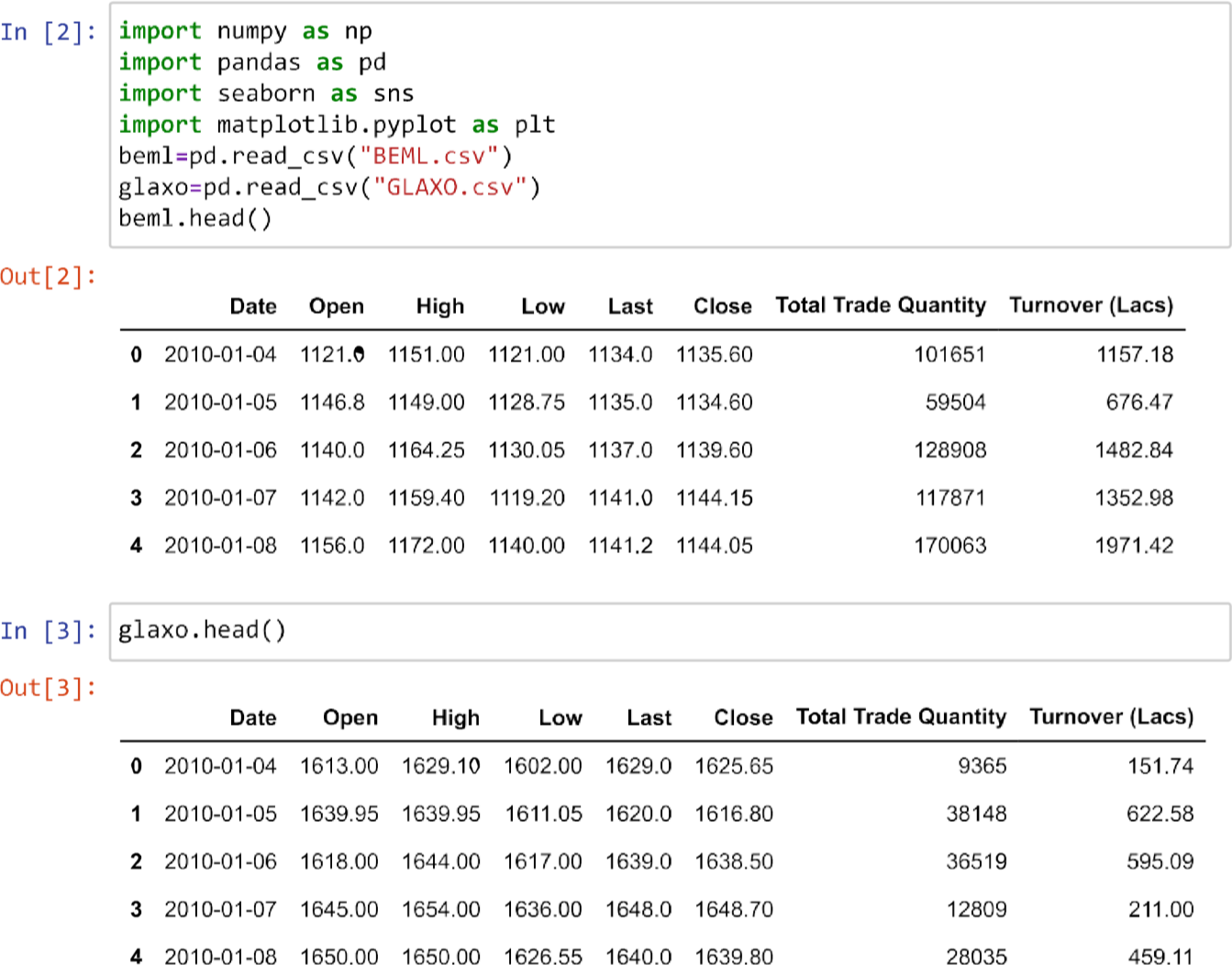
# Interpretation and Analysis:

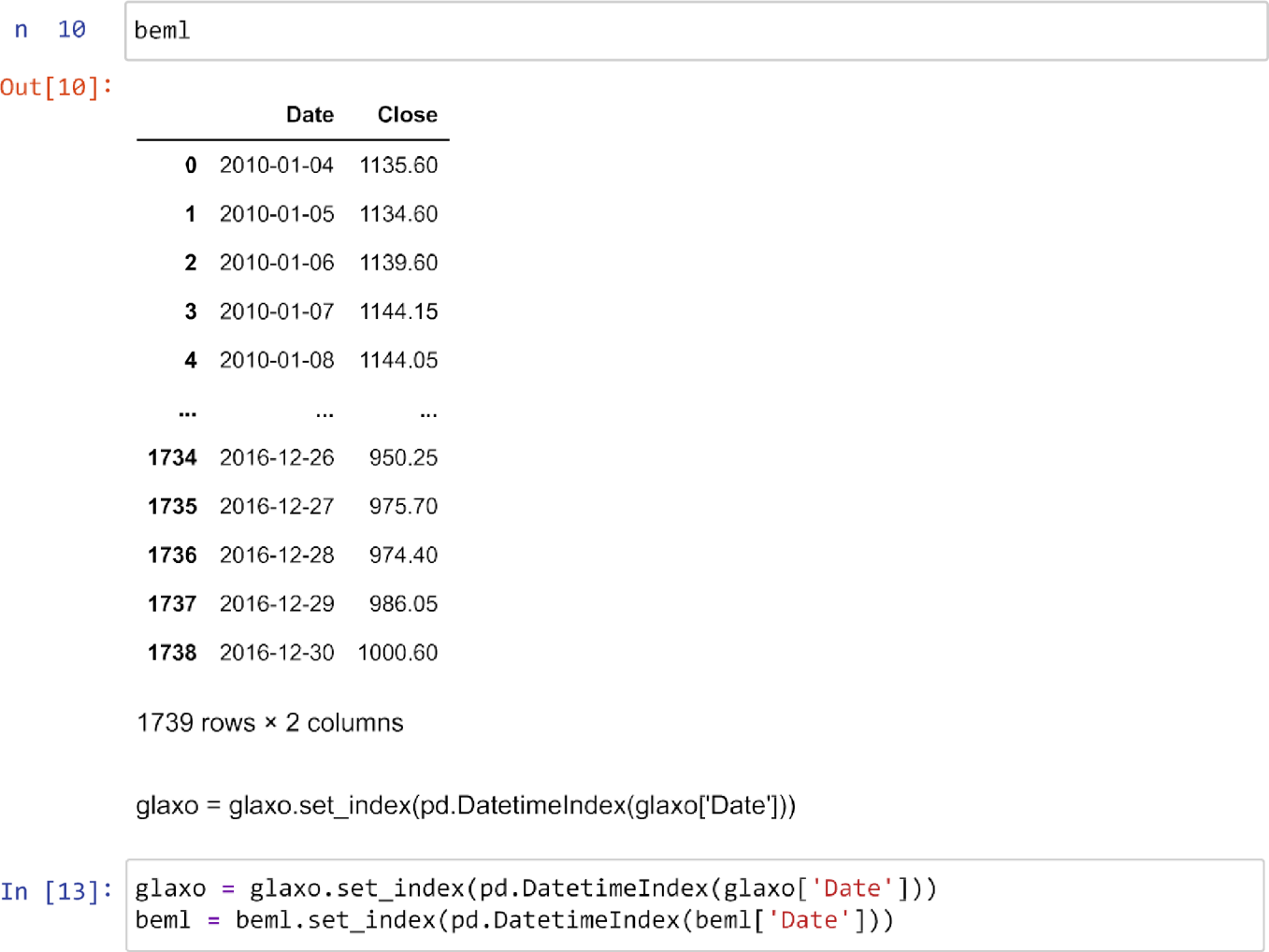
* + Analyze the histogram, PDF plot, and computed statistics to understand the characteristics of the normal distribution.
  + Observe how changes in mean and standard deviation affect the distribution.
  + Compare the theoretical properties of the normal distribution with the experimental results obtained from the generated samples.

# Iteration and Exploration:

* + Experiment with different parameters, sample sizes, and visualization techniques to gain a deeper understanding of the normal distribution.
  + Explore additional functionalities and methods provided by NumPy and Matplotlib for working with probability distribution

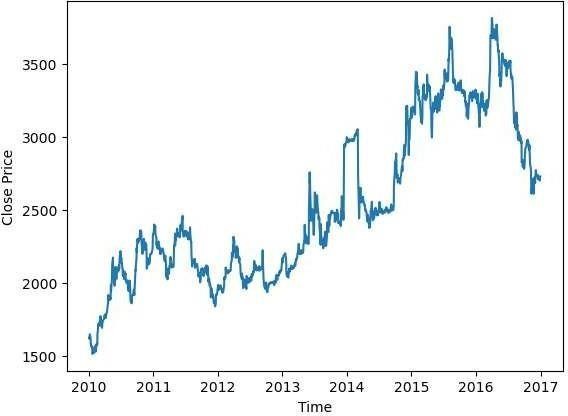
# CODE AND OUTPUT:



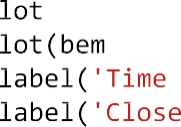


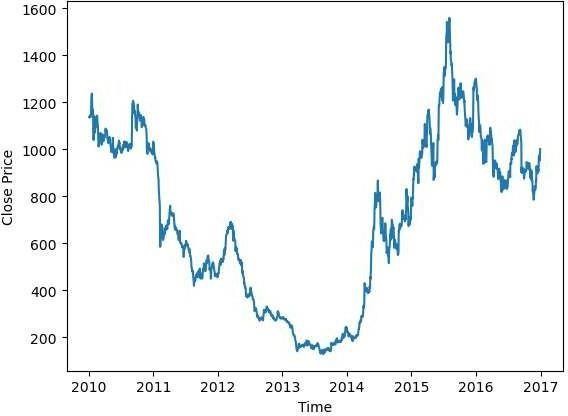






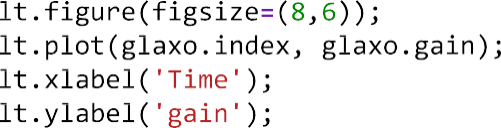


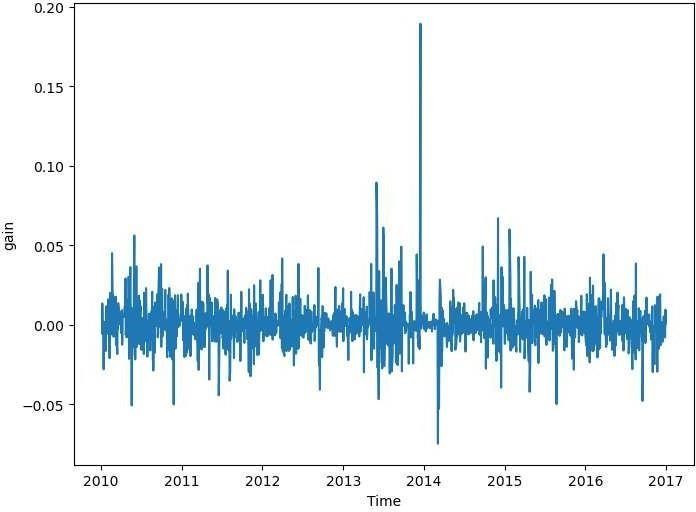




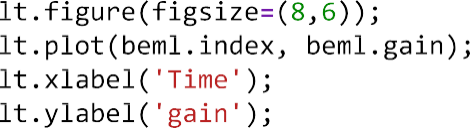




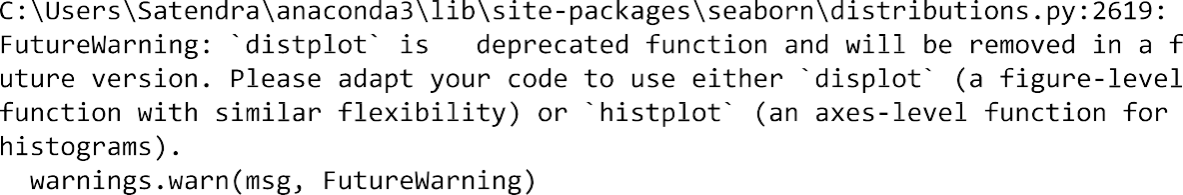
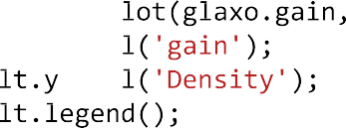
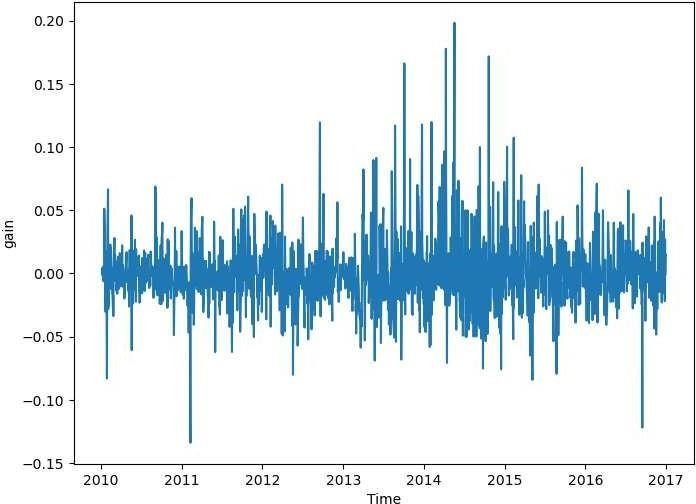




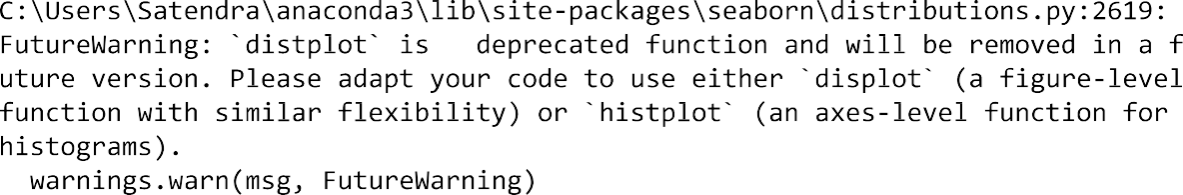
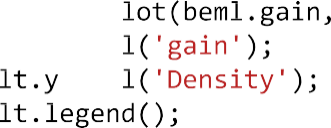
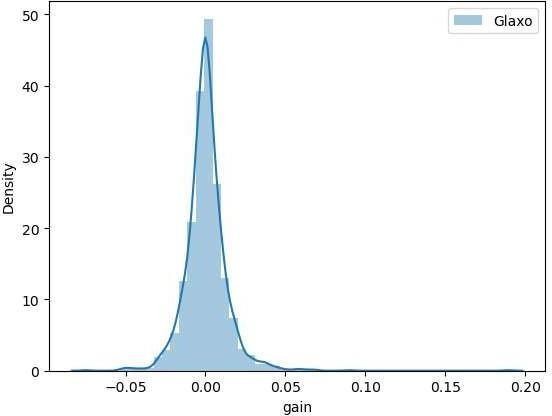


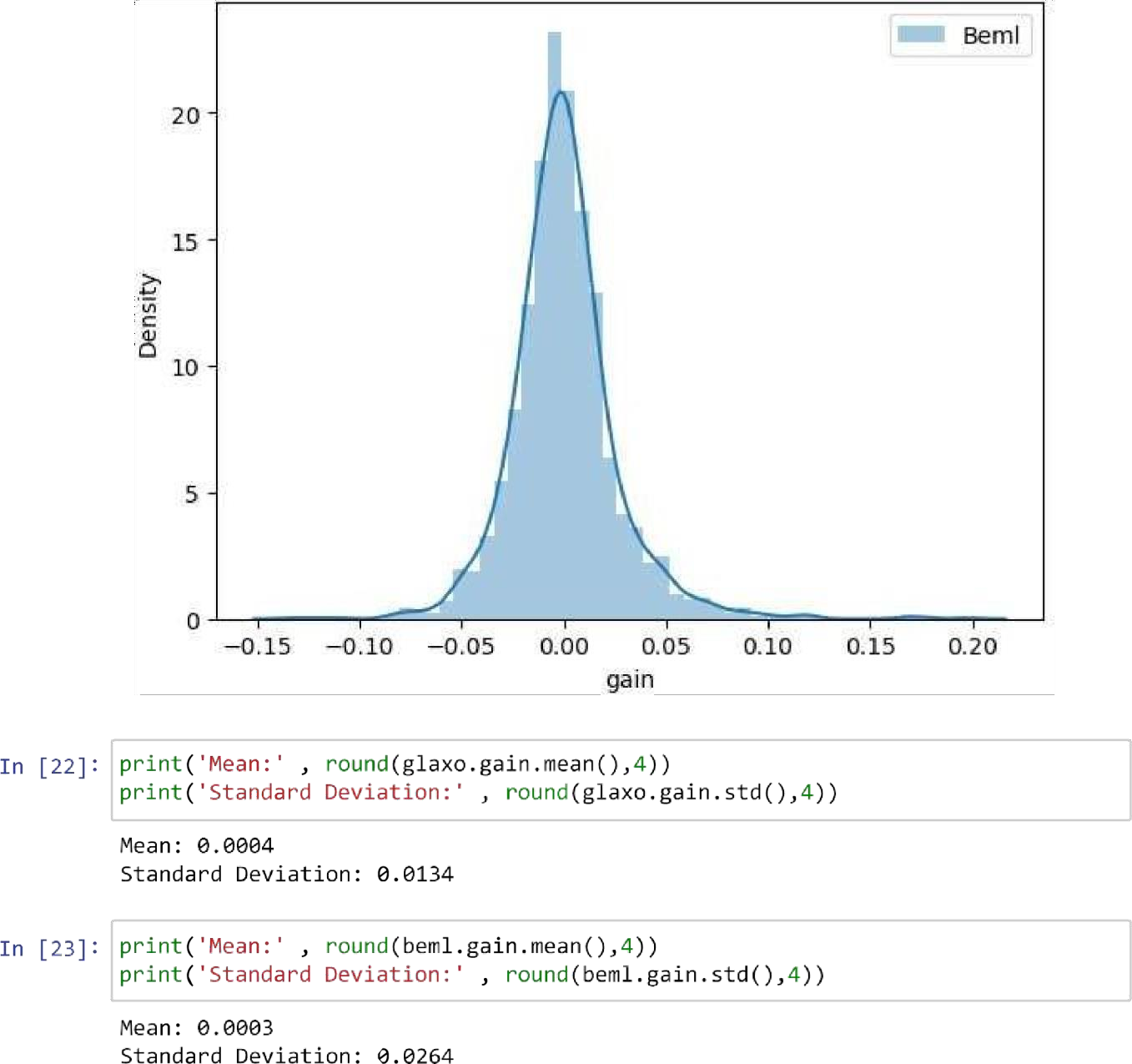




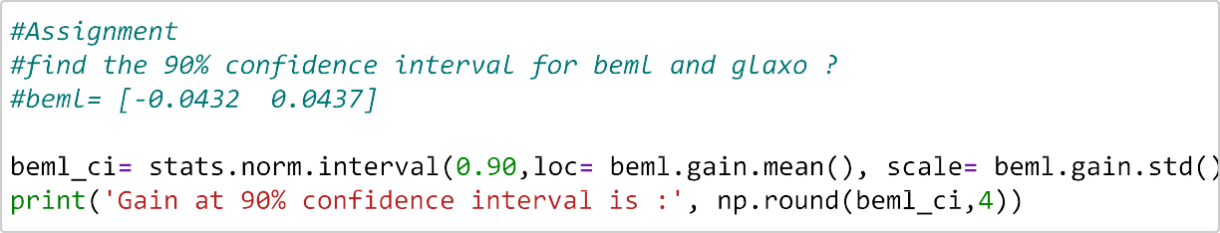






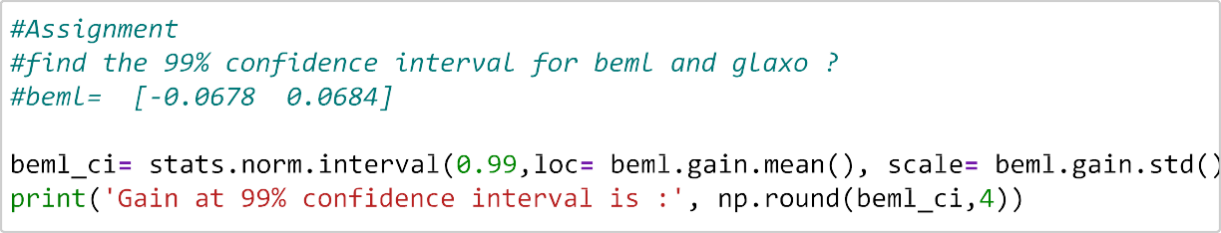
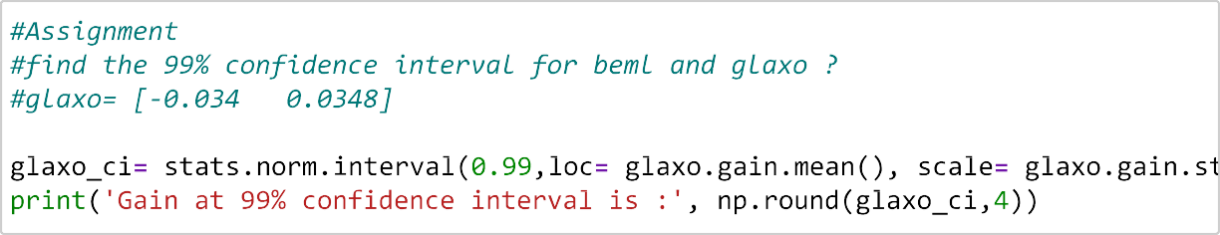










# CONCLUSION:

In this guide, we've explored various aspects of the normal distribution in Python using Jupyter Notebooks. We've learned how to generate random samples, visualize the distribution using histograms, compute statistics such as mean and standard deviation, and understand the probability density function (PDF). Understanding the normal distribution is essential for anyone working with data analysis, statistical modeling, or machine learning, and Python provides powerful tools to explore and analyze such distributions effectively. With this knowledge, we'll be better equipped to handle and interpret data in real-world applications across diverse domains.