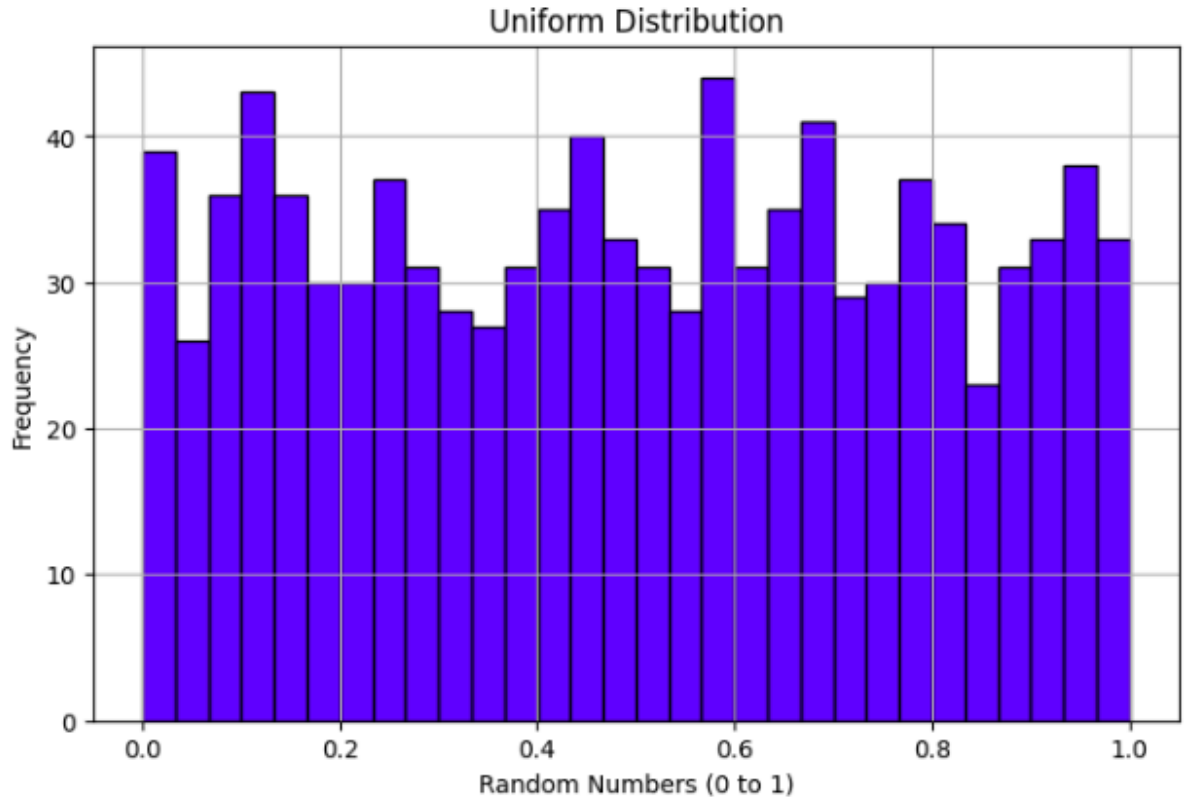
Report: Custom Random Number Generator and Visualizations

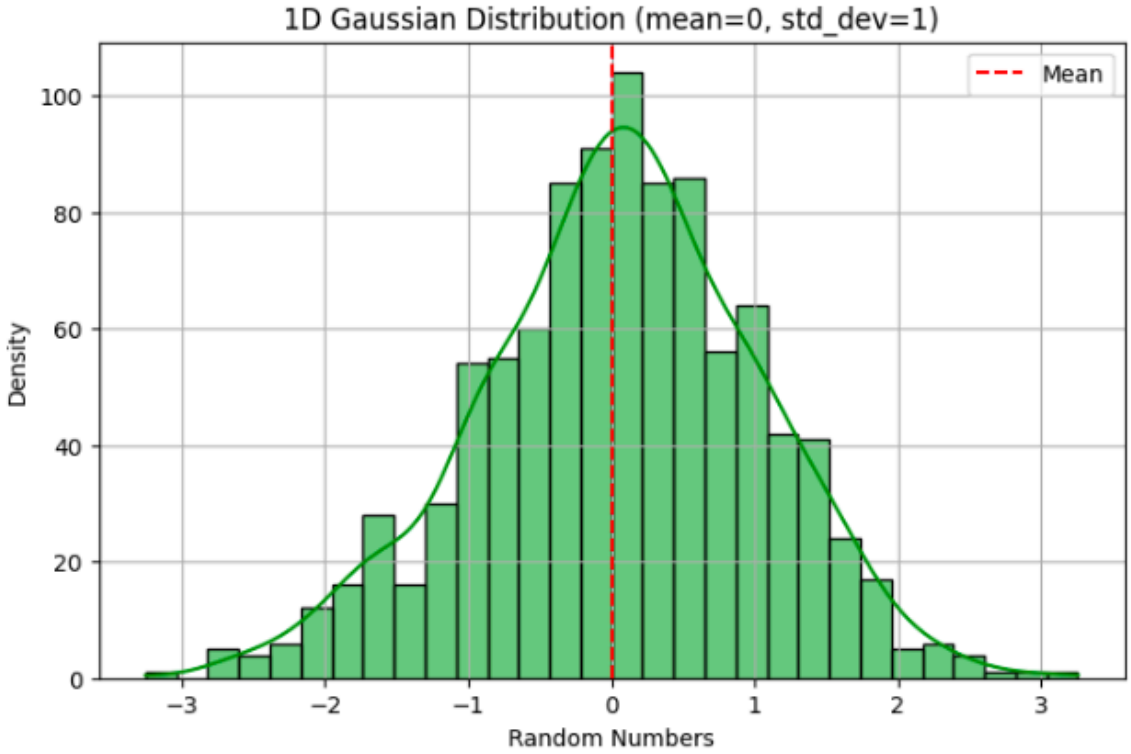
This report outlines the development of a custom random number generator (RNG) in Python, capable of producing uniform, 1D Gaussian, and 2D Gaussian distributions. It also discusses the visualizations created to illustrate these distributions.

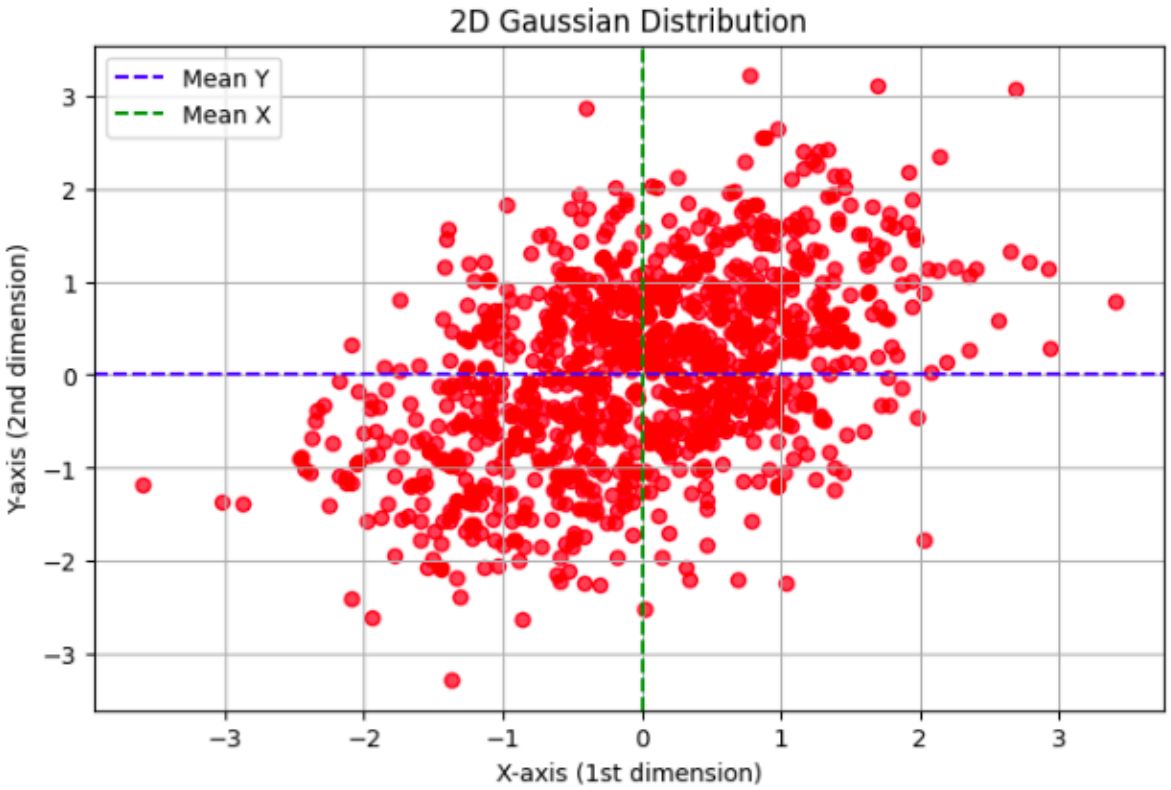
The RNG class is designed to generate random numbers using various mathematical methods. It includes functions for producing uniform random numbers and transforming these into Gaussian distributions.

Key Components:

1. **Uniform Distribution:**
   The uniform method uses a Linear Congruential Generator to create random numbers in the range [0, 1). This is a fundamental building block for generating other types of random numbers.
2. **1D Gaussian Distribution:**
   The Gaussian method employs the Box-Muller transform to convert uniform random numbers into normally distributed ones. Users can specify the mean and standard deviation to shape the distribution.
3. **2D Gaussian Distribution:**
   The mv_gaussian method generates correlated random numbers in two dimensions by utilizing the Cholesky decomposition of a covariance matrix. This ensures that the generated numbers maintain the specified relationships between them.
4. **Visualizations:**
   I use Matplotlib and Seaborn to visualize the generated random numbers, allowing for easy analysis of their distributions.
   - Uniform Distribution:
     The uniform distribution appears flat when plotted, indicating that the numbers are evenly spread across the range.
   - 1D Gaussian Distribution:
     The visualization of the 1D Gaussian distribution shows a classic bell curve, highlighting how the numbers cluster around the mean.
   - 2D Gaussian Distribution:
     In the scatter plot for the 2D Gaussian distribution, points are distributed according to the covariance matrix, illustrating the relationship between the two variables. Lines indicating the mean help visualize where the majority of the data points fall.
     These are my results for 1000 random number generator for 1D and 2D arrays:


Uniform Distribution
Frequency
Random Numbers (0 to 1)
0
10
20
30
40
0.0
0.2
0.4
0.6
0.8
1.0


1D Gaussian Distribution (mean=0, std_dev=1)
Mean
Density
Random Numbers
0
20
40
60
80
100
−3
−2
−1
0
1
2
3



5. **Error Handling:**
   The RNG class includes basic error handling to ensure:

- The number of generated random numbers (count) is positive.
- The standard deviation is non-negative.
- The mean vector and covariance matrix dimensions are compatible for multivariate generation.

Conclusion:

The custom RNG class efficiently generates and visualizes random numbers across different distributions. The visualizations confirm that the generated numbers align with expected statistical properties, making it a valuable tool for simulations and analyses involving randomness.