2. (Use software such as Python, R, Matlab, etc ..) Suppose X1, · · · ,Xn are from N(0,1). The distribution of the maximum X(n) does not have a closed form pdf. However, we can consider two ways to approximate the pdf. Let n = 30.

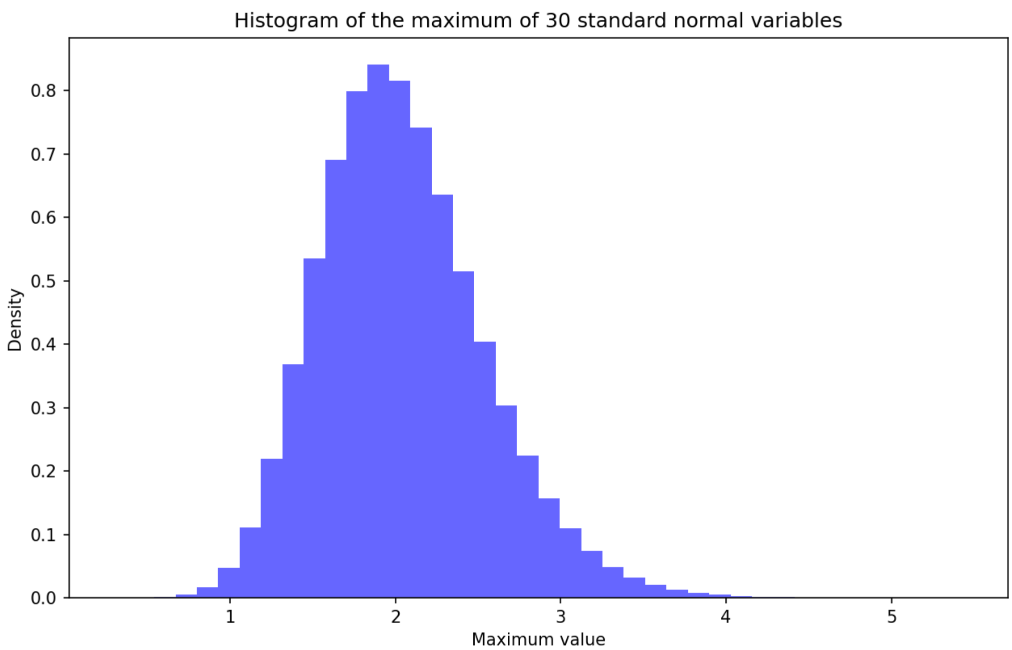
(a) Approximate the distribution of X(n) by simulation, that is, repeat the data generation many times and draw a histogram of X(n).

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt

n = 30
num_simulations = 1000000

maxima = np.max(np.random.randn(num_simulations, n), axis=1)

#2-(a)
plt.figure(figsize=(10, 6))
plt.hist(maxima, bins=40, density=True, alpha=0.6, color='blue')
plt.title('Histogram of the maximum of 30 standard normal variables')
plt.xlabel('Maximum value')
plt.ylabel('Density')
plt.show()
```



(b) Write the pdf of X(n), as learned from the class. Draw the pdf by numerically evaluating the function over a grid. Compare this with the histogram in the above.

- Pdf of X(n) is $f_{X(n)} = n \times \phi(x) \times [\Phi(x)]^{n-1}$
- $\phi(x)$is the PDF of the standard normal distribution and $\Phi(x)$ is its CDF.
- Both plots demonstrate that the theoretical PDF is consistent with the simulated data, validating the use of the given mathematical model to describe the behavior of the maximum of standard normal variables for n=30.

```
from scipy.stats import norm
import numpy as np
import matplotlib.pyplot as plt

n = 30
num_simulations = 1000000

maxima = np.max(np.random.randn(num_simulations, n), axis=1)

'''...

#2-(b)
x_values = np.linspace(-2, 5, 1000)
pdf_values = n * norm.pdf(x_values) * norm.cdf(x_values)**(n-1)

plt.figure(figsize=(10, 6))
plt.plot(x_values, pdf_values, label='Theoretical PDF', color='red')
plt.hist(maxima, bins=40, density=True, alpha=0.4, color='blue', label='Simulated Histogram')
plt.title('Comparison of Theoretical PDF and Simulated Histogram')
plt.xlabel('Maximum value')
plt.ylabel('Density')
plt.legend()
plt.show()
```