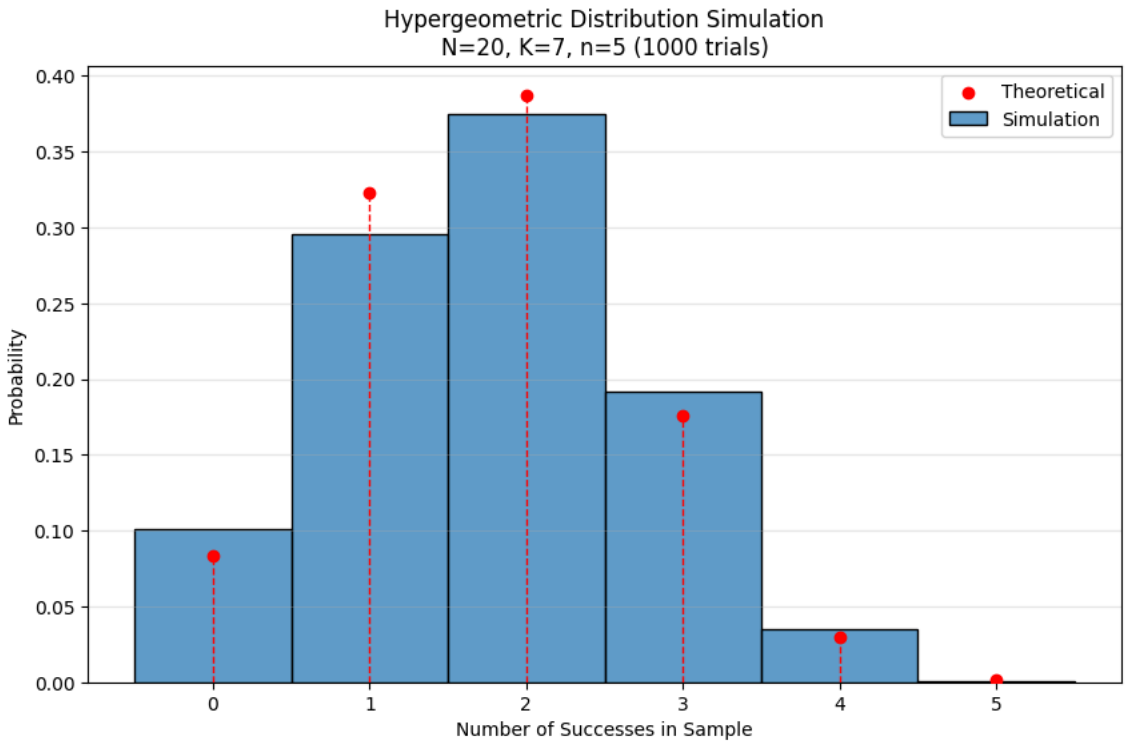
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from math import comb  # Using Python's built-in comb function

# Set the parameters
population_size = 20  # Total objects (N)
num_successes = 7     # Successes in population (K)
sample_size = 5       # Number of draws (n)
num_simulations = 1000  # Number of experiments

# Simulate the hypergeometric distribution
results = np.random.hypergeometric(
    ngood=num_successes,
    nbad=population_size - num_successes,
    nsample=sample_size,
    size=num_simulations
)

# Calculate theoretical probabilities using math.comb
x_values = np.arange(0, min(num_successes, sample_size) + 1)
theoretical_probs = [
    (comb(num_successes, k) * comb(population_size - num_successes, sample_size - k)) /
    comb(population_size, sample_size)
    for k in x_values
]

# Plot the results
plt.figure(figsize=(10, 6))
sns.histplot(results, bins=np.arange(-0.5, sample_size + 1.5, 1),
             stat='probability', alpha=0.7, label='Simulation')

# Overlay theoretical distribution
plt.scatter(x_values, theoretical_probs, color='red', zorder=5, label='Theoretical')
plt.vlines(x_values, 0, theoretical_probs, colors='red', linestyles='dashed', linewidth=1)

# Formatting
plt.title(f'Hypergeometric Distribution Simulation\nN={population_size}, K={num_successes}, n={sample_size} (1000 trials)')
plt.xlabel('Number of Successes in Sample')
plt.ylabel('Probability')
plt.xticks(range(0, sample_size + 1))
plt.legend()
plt.grid(axis='y', alpha=0.3)

plt.show()
```


Hypergeometric Distribution Simulation
N=20, K=7, n=5 (1000 trials)
Theoretical
Simulation
Probability
Number of Successes in Sample