
Project - 1 MA517M-Basic Programming Laboratory	Last Date: 09 November 2025
Name	Roll No.: MA25M005

Hypergeometric Distribution using C++ Classes and Operator Overloading

Objective: To design a C++ program that implements the hypergeometric distribution using classes, computes PMF, CDF, expectation, variance, and supports operator overloading for comparison.

A hypergeometric random variable $X \sim \text{Hypergeom}(N, K, n)$ counts the number of successes in n draws without replacement from a population of size N containing K successes.

The probability mass function (PMF) is

$$P(X = k) = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}, \quad \max(0, n + K - N) \leq k \leq \min(K, n)$$

Mean and variance:

$$\mathbb{E}[X] = n \frac{K}{N}, \quad \text{Var}(X) = n \frac{K}{N} \frac{N-K}{N} \frac{N-n}{N-1}$$

Problem Description

Design a class `HypergeometricRV` to represent a hypergeometric random variable. The class should allow computation of PMF, CDF, mean, variance, and comparison with other hypergeometric random variables using operator overloading.

Class Specification

- **Class Name:** `HypergeometricRV`
- **Private Data Members:**
 - `int N;`
 - `int K;`
 - `int n;`
- **Public Member Functions:**
 - `HypergeometricRV(int population, int success, int draws);`
 - `double pmf(int k) const;`
 - `double cdf(int k) const;`
 - `double mean() const;`
 - `double variance() const;`
 - `void display() const;`

Operator Overloading

- `operator ==( ), !=( ), <( ), >( )` Compares mean or variance of two hypergeometric random variables
- `operator <<( )` Displays the hypergeometric variable's parameters and statistics

Tasks

1. Create a hypergeometric random variable $X \sim \text{Hypergeom}(N = 50, K = 10, n = 5)$
2. Compute PMF and CDF for selected values of k
3. Compute mean and variance
4. Create another variable $Y \sim \text{Hypergeom}(50, 15, 5)$ and compare X and Y
5. Display results using the overloaded `<<` operator

Expected Output Example

```
X ~ Hypergeom(N=50, K=10, n=5)
```

```
PMF P(X=2) = 0.275
```

```
CDF P(X<=2) = 0.678
```

```
Mean = 1.0, Variance = 0.796
```

```
Y ~ Hypergeom(N=50, K=15, n=5)
```

```
Mean = 1.5, Variance = 0.983
```

```
X < Y : True
```

```
X == Y: False
```

Project - 2: Lights Out Game Using C++ Classes

Problem Statement

Design and implement the **Lights Out Game** using **C++ classes**. The game consists of a grid of lights, where each light can be either *on* or *off*. Pressing a light toggles its state and the state of its adjacent neighbors (up, down, left, right). The objective is to turn all lights off. The project should utilize object-oriented programming concepts such as classes, objects, encapsulation, and methods for handling grid updates and game logic.

Project Requirements

1. Create a `Light` class to represent a single light in the grid.
 - (a) Data member: current state (`on/off`)
 - (b) Methods: `toggle()`, `getState()`
2. Create a `Grid` class to represent the game board.

- (a) Initialize the grid with a random pattern of lights on and off.
 - (b) Display the current state of the grid in a readable format.
 - (c) Toggle a selected light and its adjacent neighbors when a user chooses a cell.
 - (d) Include a method to check if all lights are off (win condition).
3. Create a **Game** class to manage gameplay.
 - (a) Display options for the user: **Play** or **Solution**.
 - (b) If the user chooses **Play**, allow them to select lights to toggle.
 - (c) Allow the user to quit and confirm before exiting.
 - (d) Optionally, allow the user to reset the board or play multiple rounds.
4. Ensure proper encapsulation of light and grid operations within the respective classes.

Suggested Class Structure

1. **Light** Class:

- Data member: `bool isOn`
- Methods: `toggle()`, `getState()`

2. **Grid** Class:

- Data member: 2D array of **Light** objects
- Methods: `initializeGrid()`, `displayGrid()`, `toggleLight(int row, int col)`, `isWin()`

3. **Game** Class:

- Data member: **Grid** object, user choice, move counter
- Methods: `playGame()`, `showSolution()`, `confirmQuit()`, `resetBoard()`

Reference

For more details about the Lights Out game, visit: [https://en.wikipedia.org/wiki/Lights_Out_\(game\)](https://en.wikipedia.org/wiki/Lights_Out_(game))