StatsLibrary Report

Mouaz Ali

Table of Contents

The provided Java classes continue a statistical library designed to facilitate common calculations and distributions encountered in probability and statistics. The library consists of two classes: StatsLibrary and StatsLibraryTester.

StatsLibrary Class

The StatsLibrary class contains various methods for computing fundamental statistical operations and probability distributions.

Here's an overview of its functionalities:

1. Factorial
    - Method: factorial(int x)
        - Calculates the factorial of a given integer x using BigInteger arithmetic to handle large numbers.
2. Combinations
    - Method: combination(int n, int r)
        - Computes the number of combinations given n total items and r items selected, using the factorial function.
3. Poisson Distribution
    - Methods:
        - poissonDistribution(int lambda, int y): Computes the probability of a Poisson distribution with average rate lambda and y occurrences.
        - poissonExpectedAndVariance(int lambda): Returns the expected value (which is equal to lambda) and variance of a Poisson distribution.
        - poissonStandardDeviation(int lambda): Computes the standard deviation of a Poisson distribution.
4. Chebyshev's
    - Methods:
        - chebyshev(double lowerRange, double upperRange, double mean, double standardDeviation): Calculates the percentage of values between specified intervals using Chebyshev's inequality.
        - chebyshevGivenVariance(double lowerRange, double upperRange, double mean, double variance): Calculates the percentage of values between specified intervals using Chebyshev's inequality, given the variance.
5. Uniform Distribution
    - Methods:
        - uniformDistribution(int a, int b, int c, int d): Computes the probability of an event occurring in a specified interval given a larger total interval.
        - uniformExpectedValue(int c, int d): Calculates the expected value in a given total interval.
        - uniformVariance(int c, int d): Computes the variance given a total interval.

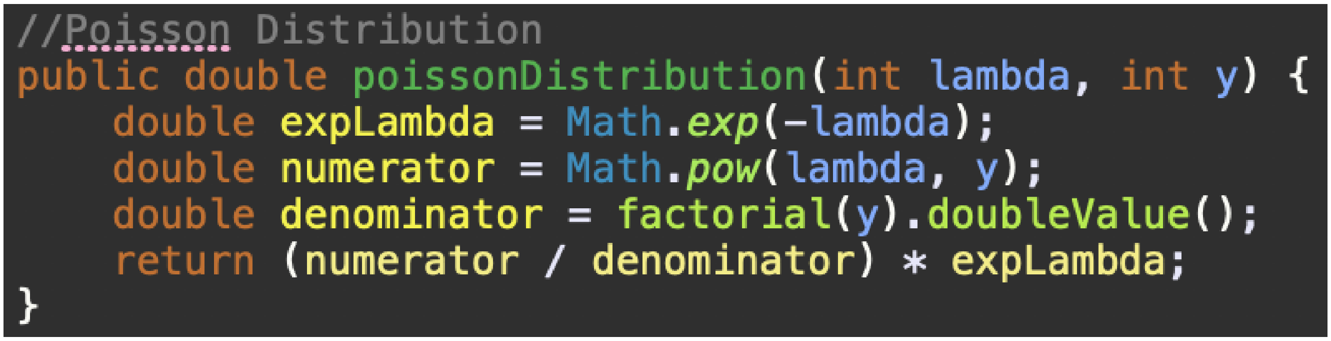
- ○ uniformStandardDeviation(int c, int d): Calculates the standard deviation given a total interval.

StatsLibraryTester Class
The StatsLibraryTester class serves as a test suite for validating the functionality of the methods implemented in the StatsLibrary class. It contains various test cases for each method, providing input parameters and displaying the computed results.

Overall, this statistical library offers a comprehensive set of tools for performing common statistical calculations, making it a valuable resource for applications involving probability and statistics.

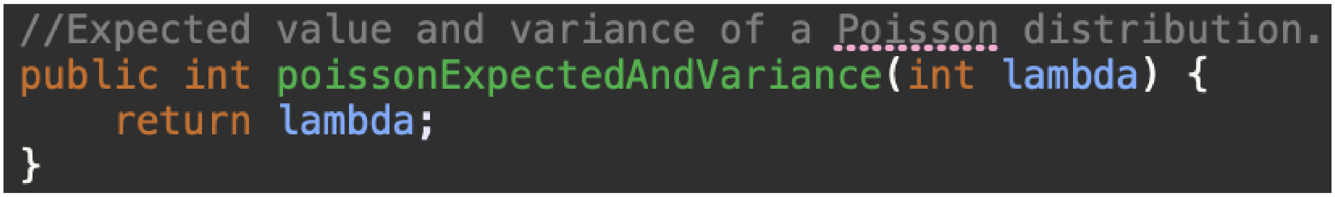
Poisson Distribution
Methods:

Probability



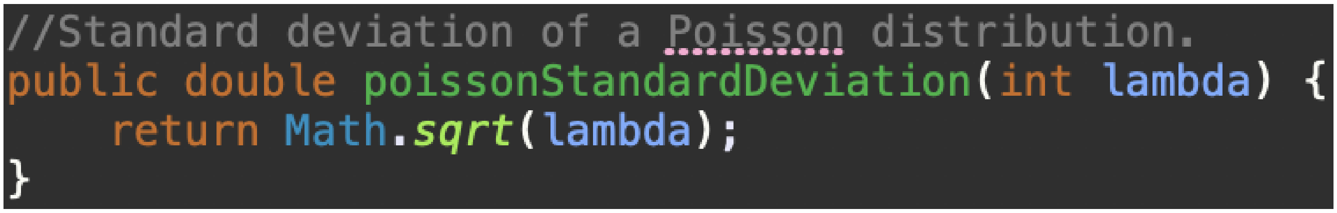
```
//Poisson Distribution
public double poissonDistribution(int lambda, int y) {
    double expLambda = Math.exp(-lambda);
    double numerator = Math.pow(lambda, y);
    double denominator = factorial(y).doubleValue();
    return (numerator / denominator) * expLambda;
}
```

Expected value and variance



```
//Expected value and variance of a Poisson distribution.
public int poissonExpectedAndVariance(int lambda) {
    return lambda;
}
```

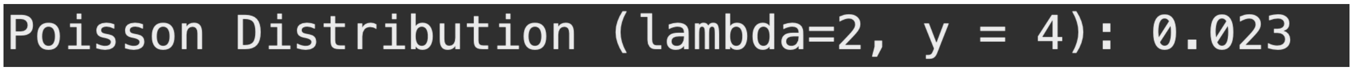
Standard deviation



```
//Standard deviation of a Poisson distribution.
public double poissonStandardDeviation(int lambda) {
    return Math.sqrt(lambda);
}
```

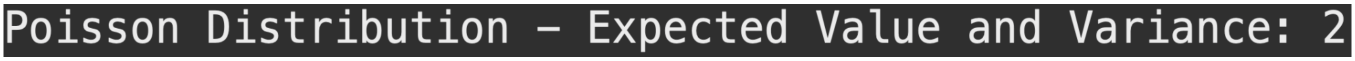
Outputs:

Probability



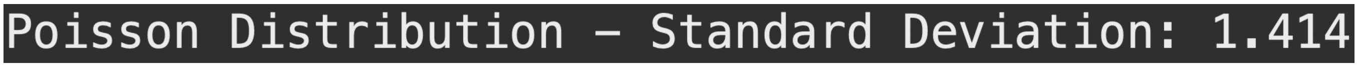
```
Poisson Distribution (lambda=2, y = 4): 0.023
```

Expected value and variance



```
Poisson Distribution - Expected Value and Variance: 2
```

Standard deviation

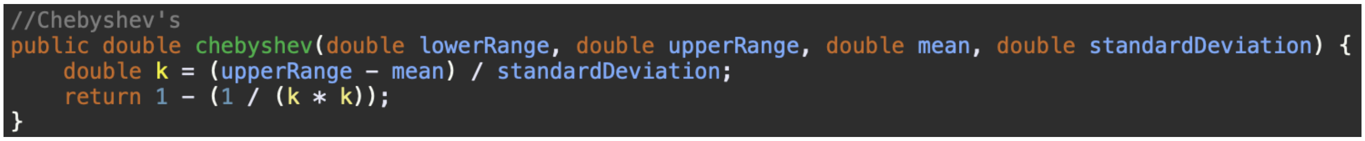


```
Poisson Distribution - Standard Deviation: 1.414
```

Chebyshev's

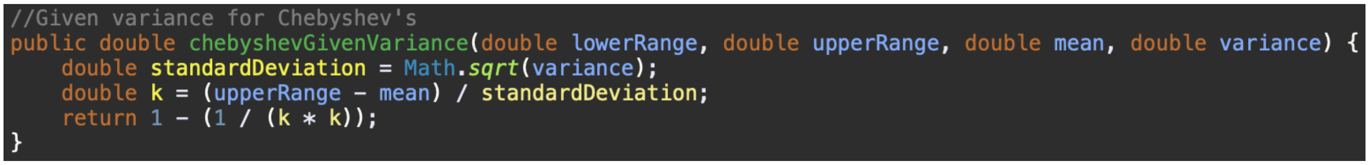
Methods:

Probability



```java
//Chebyshev's
public double chebyshev(double lowerRange, double upperRange, double mean, double standardDeviation) {
    double k = (upperRange - mean) / standardDeviation;
    return 1 - (1 / (k * k));
}
```

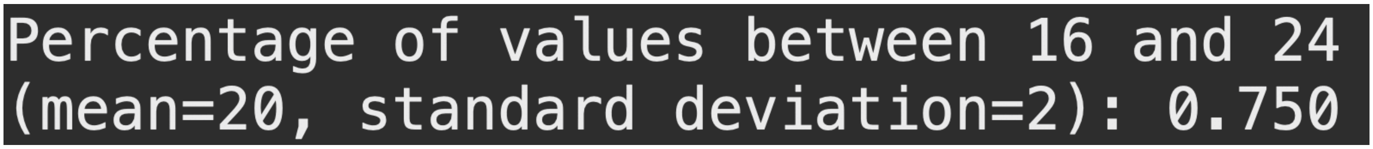
Given variance



```java
//Given variance for Chebyshev's
public double chebyshevGivenVariance(double lowerRange, double upperRange, double mean, double variance) {
    double standardDeviation = Math.sqrt(variance);
    double k = (upperRange - mean) / standardDeviation;
    return 1 - (1 / (k * k));
}
```

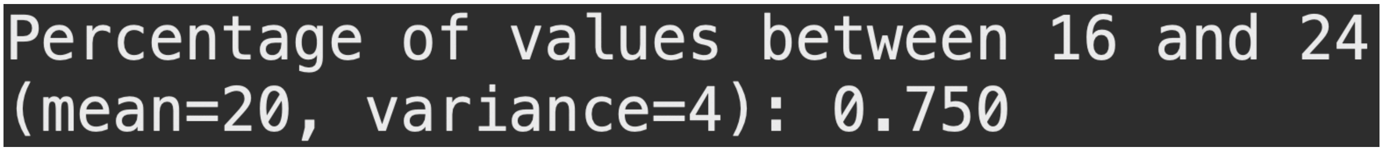
Outputs:

Probability



```
Percentage of values between 16 and 24
(mean=20, standard deviation=2): 0.750
```

Given variance

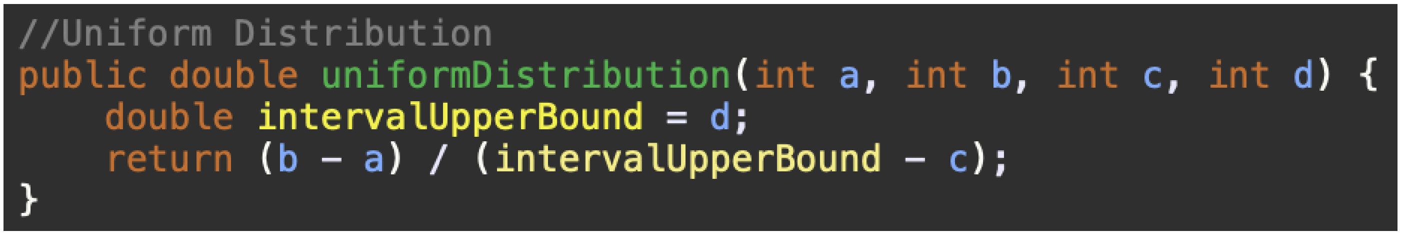


```
Percentage of values between 16 and 24
(mean=20, variance=4): 0.750
```

Uniform Distribution

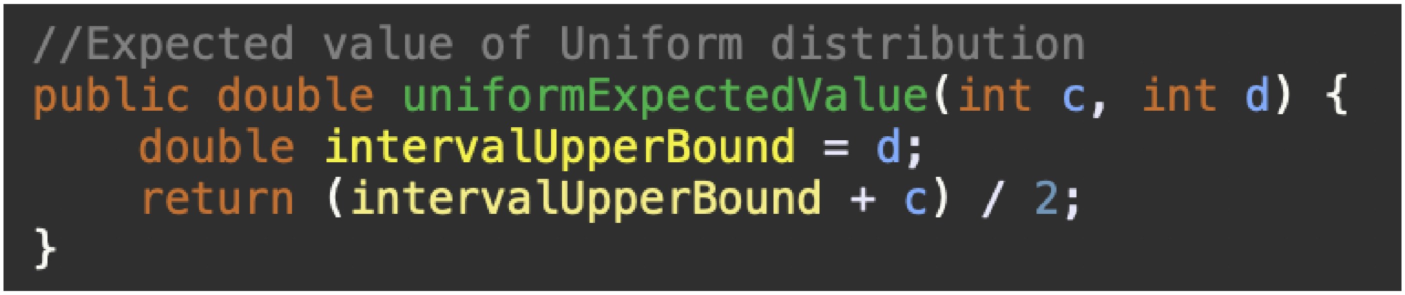
Methods:

Probability



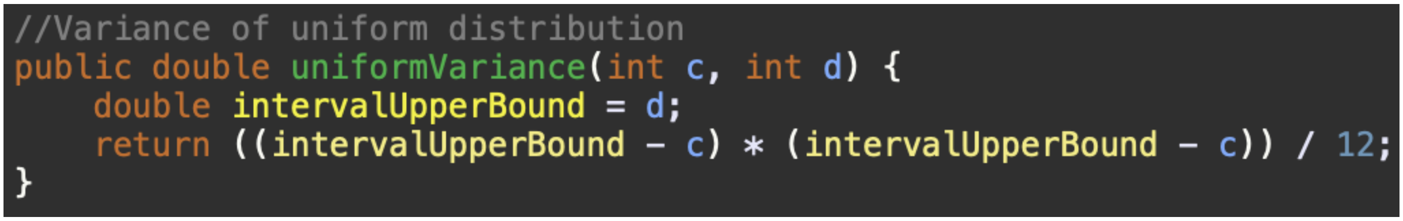
```
//Uniform Distribution
public double uniformDistribution(int a, int b, int c, int d) {
    double intervalUpperBound = d;
    return (b - a) / (intervalUpperBound - c);
}
```

Expected value



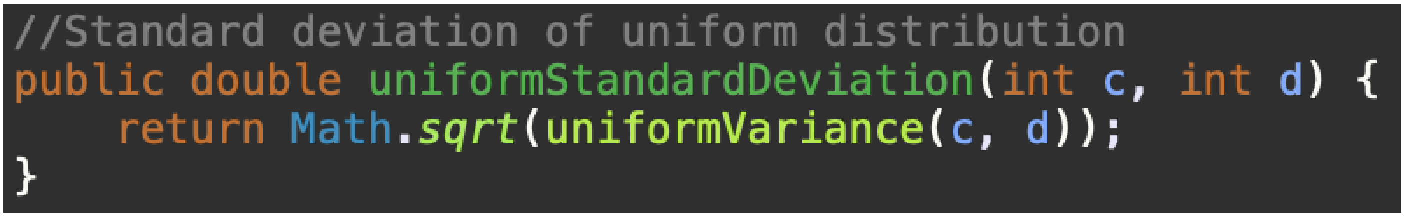
```
//Expected value of Uniform distribution
public double uniformExpectedValue(int c, int d) {
    double intervalUpperBound = d;
    return (intervalUpperBound + c) / 2;
}
```

Variance



```
//Variance of uniform distribution
public double uniformVariance(int c, int d) {
    double intervalUpperBound = d;
    return ((intervalUpperBound - c) * (intervalUpperBound - c)) / 12;
}
```

Standard deviation

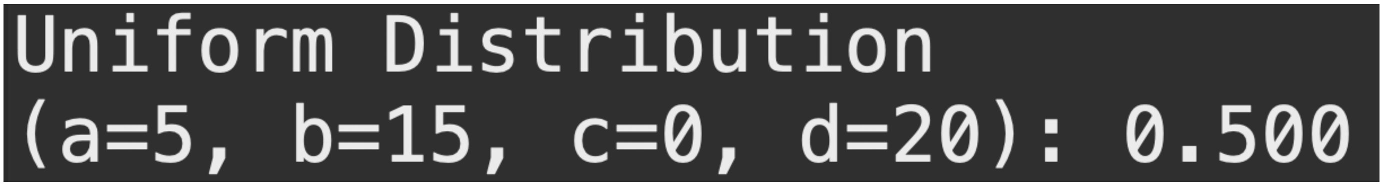


```
//Standard deviation of uniform distribution
public double uniformStandardDeviation(int c, int d) {
    return Math.sqrt(uniformVariance(c, d));
}
```

Outputs:

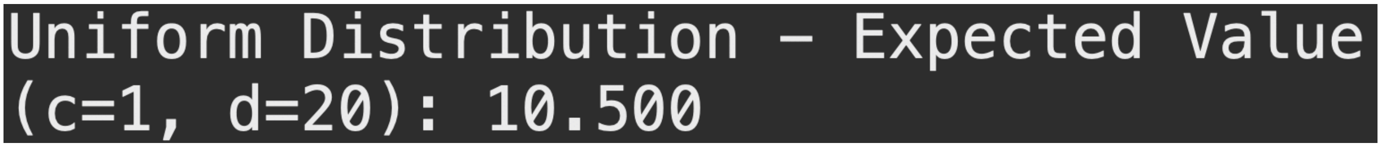
Probability

Internal interval: upper bound b = 15; lower bound a = 5
External interval: upper bound d = 20; lower bound c = 0



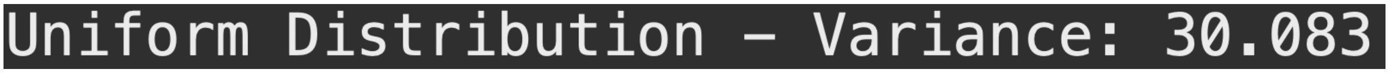
```
Uniform Distribution
(a=5, b=15, c=0, d=20): 0.500
```

Expected value
Upper bound: d = 20
Lower bound: c = 1



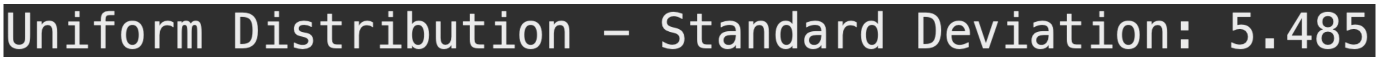
```
Uniform Distribution – Expected Value
(c=1, d=20): 10.500
```

Variance
Upper bound: d = 20
Lower bound: c = 1



```
Uniform Distribution – Variance: 30.083
```

Standard deviation
Upper bound: d = 20
Lower bound: c = 1



```
Uniform Distribution – Standard Deviation: 5.485
```