# Laboratory 4: Discrete Distributions AND Continuous Distributions

1. Introduction and Purpose of Experiment
2. Aim and Objectives
3. Experimental Procedure

Design and Implement a Java program for the following **Discrete Probability Distribution**

* 1. Binomial distribution
     1. To find the number of successes in **n** independent Bernoulli trials, given that **X** has a binomial distribution
     2. Calculate the
        1. Mean, E(X)
        2. Variance, V(X)
  2. Geometric distribution
     1. To identify the number of Bernoulli trials, *X*, to achieve the 1st success
     2. Calculate the
        1. Mean, *E(X)*
        2. Variance *V(X)*
  3. Negative binomial distribution

1. To identify the number of Bernoulli trials, X, until the kth success
2. Calculate the Mean, E(X) and Variance V(X)
   1. Develop and implement a Java program by selecting suitable distribution function for given scenario:If 40% of the assembled ink-jet printers are rejected at the inspection station. Your program should identify:
3. Probability that the first acceptable ink-jet printer is the third one inspected. Considering each inspection as a Bernoulli trial with q=0.4 and p=0.6.
4. Probability that the third printer inspected is the second acceptable printer
5. Design and Implement a Java program for the following **Continuous Distribution**

A computer repair person is “beeped” each time there is a call for service. If the number of beeps per hour is Poisson distributed (α = 2 beeps per hour). Then design and implement a Java program to determine the following.

* + 1. The probability of exactly three beeps in the next hour:
    2. The probability of two or more beeps in a 1-hour period:

1. Algorithms
2. Presentation of Results

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab04;

import java.util.Scanner;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class Lab04 {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        Distribution.DISTR\_TYPE currDistr;

        Integer choice;

        System.out.print("Enter the Distrubution you want\n1.\tBinomial\n2.\tGeometrical\n3.\tNegative Binomial\n4.\tPoisson\nYour Choice : ");

        choice = input.nextInt();

        Distribution distr = null;

        switch (choice) {

            case 1: {

                // Binomial

                currDistr = Distribution.DISTR\_TYPE.BINOM;

                System.out.print("Enter N : ");

                Long N = input.nextLong();

                System.out.print("Enter X : ");

                Long K = input.nextLong();

                System.out.print("Enter P (success) : ");

                Double P = input.nextDouble();

                distr = new BinomialDistribution(currDistr, N, P, K);

            }

            break;

            case 2: {

                // Geometrical

                currDistr = Distribution.DISTR\_TYPE.GEOMT;

                System.out.print("Enter P (success) : ");

                Double P = input.nextDouble();

                System.out.print("Enter K : ");

                Long K = input.nextLong();

                distr = new GeometricDistribution(currDistr, P, K);

            }

            break;

            case 3: {

                // Negative Binomial

                currDistr = Distribution.DISTR\_TYPE.NBINOM;

                System.out.print("Enter P (success) : ");

                Double P = input.nextDouble();

                System.out.print("Enter N : ");

                Long N = input.nextLong();

                System.out.print("Enter R : ");

                Long R = input.nextLong();

                distr = new NegativeBinomialDistribution(currDistr, P, N, R);

            }

            break;

            case 4: {

                // Poisson

                currDistr = Distribution.DISTR\_TYPE.POISSON;

                System.out.print("Enter lambda : ");

                Double lambda = input.nextDouble();

                System.out.print("Enter K : ");

                Long K = input.nextLong();

                distr = new PoissonDistribution(currDistr, lambda, K);

            }

            break;

            default:

                main(args);

                return;

        }

        if (distr != null) {

            System.out.printf("P( X = %d ) = %.10f\nE[X] = %.10f\nVar[X] = %.10f\n",

                    distr.getParam(), distr.getDistribution(), distr.getExpectance(), distr.getVariance());

        } else {

            System.out.println("DISTR IS NULL");

        }

    }

}

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab04;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public abstract class Distribution {

    public static enum DISTR\_TYPE {

        BINOM, GEOMT, NBINOM, POISSON

    };

    public Distribution(DISTR\_TYPE currentDistr) {

        this.currentDistr = currentDistr;

    }

    public DISTR\_TYPE currentDistr;

    public static Long choose(Long n, Long r) {

        if (r > n/2)

            r = n - r;

        Long ans = 1l;

        for (int i = 1 ; i <= r ; i++) {

            ans \*= (n - r + i);

            ans /= i;

        }

        return ans;

    }

    public abstract Long getParam();

    /\*\*

     \*

     \* Calculates P(X = K) generally speaking

     \*

     \* @return

     \*/

    public abstract Double getDistribution();

    public abstract Double getExpectance();

    public abstract Double getVariance();

}

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab04;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class BinomialDistribution extends Distribution {

    Long N, K;

    Double P;

    public BinomialDistribution(DISTR\_TYPE currentDistr, Long N, Double P, Long K) {

        super(currentDistr);

        this.N = N;

        this.P = P;

        this.K = K;

    }

    public Double binomialDistribution() {

        return choose(N, K) \* (double)Math.pow(P, K) \* (double)Math.pow( 1 - P, N - K);

    }

    @Override

    public Double getDistribution() {

        return binomialDistribution();

    }

    @Override

    public Double getExpectance() {

        return (double)N \* P;

    }

    @Override

    public Double getVariance() {

        return N \* P \* (1.0 - P);

    }

    @Override

    public Long getParam() {

        return K;

    }

}

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab04;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class GeometricDistribution extends Distribution {

    Double P;

    Long K;

    public GeometricDistribution(DISTR\_TYPE currentDistr, Double P, Long K) {

        super(currentDistr);

        this.P = P;

        this.K = K;

    }

    public Double geometricDistribution() {

        return (double)Math.pow(1 - P, K - 1) \* P;

    }

    @Override

    public Double getDistribution() {

        return geometricDistribution();

    }

    @Override

    public Double getExpectance() {

        return (1.0) / P;

    }

    @Override

    public Double getVariance() {

        return (1 - P) / (P \* P);

    }

    @Override

    public Long getParam() {

        return K;

    }

}

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab04;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class NegativeBinomialDistribution extends Distribution {

    Double P;

    Long R;

    Long N;

    public NegativeBinomialDistribution(DISTR\_TYPE currentDistr, Double P, Long N, Long R) {

        super(currentDistr);

        this.P = P;

        this.R = R;

        this.N = N;

    }

    @Override

    public Double getDistribution() {

        return choose(N-1, R-1) \* Math.pow(P, N-R) \* (double)Math.pow(1-P, R);

    }

    @Override

    public Double getExpectance() {

        return R / P;

    }

    @Override

    public Double getVariance() {

        return (R \* (1.0 - P)) / (P \* P);

    }

    @Override

    public Long getParam() {

        return R;

    }

}

/\*

 \* To change this license header, choose License Headers in Project Properties.

 \* To change this template file, choose Tools | Templates

 \* and open the template in the editor.

 \*/

package lab04;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class PoissonDistribution extends Distribution {

    Double lambda;

    Long K;

    public PoissonDistribution(DISTR\_TYPE currentDistr, Double lambda, Long K) {

        super(currentDistr);

        this.lambda = lambda;

        this.K = K;

    }

    @Override

    public Double getDistribution() {

        Double ans = Math.pow(lambda, K) \* Math.exp(-lambda);

        for (Long i = 1l; i <= K; i++) {

            ans /= i;

        }

        return ans;

    }

    @Override

    public Double getExpectance() {

        return lambda;

    }

    @Override

    public Double getVariance() {

        return getExpectance();

    }

    @Override

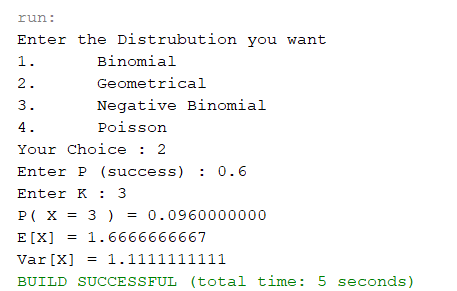
    public Long getParam() {

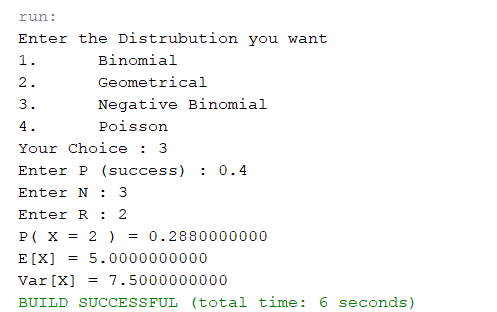
        return K;

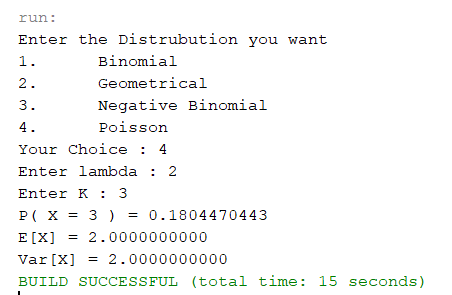
    }

}

1. Analysis and Discussions







1. Conclusions
   1. Bernoulli Distribution
   2. Binomial Distribution
   3. Geometric Distribution
   4. Negative Binomial Distribution
   5. Poisson Distribution