# Laboratory 7: Tests for Random Numbers using Frequency Tests

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

**K-S Test**

Design and implement a Java program to test the generated random numbers 0.44, 0.81, 0.14, 0.05, 0.93 for uniformity by using the Kolmogorov-Smirnov test with the level of significance α= 0.10

**Chi-Square test**

A public opinion poll surveyed a random sample of 1000 voters. Respondents were classified by gender (male or female) and by voting preference (BJP, Congress and AAP). Results are shown below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Programing language Preferences | | | Row total |
| BJP | Congress | AAP |
| Male | 200 | 150 | 50 | 400 |
| Female | 250 | 300 | 50 | 600 |
| Column total | 450 | 450 | 100 | 1000 |

Design and implement a Java program to conduct chi-square test with level of significance and determine if there is a gender gap. Identify whether the men’s preferences differ significantly from the women's preferences.

1. Algorithms

**KS-Test**

-> Rank the N random numbers in ascending order.

-> Calculate D+ as max(i/N-Ri) for all i in(1, N)

-> Calculate D- as max(Ri-((i-1)/N)) for all i in(1, N)

-> Calculate D as max(D+, D-)

-> If D>D(alpha)

Rejects Uniformity

else

It fails to reject the Null Hypothesis.

**Chi-Square Test**

First calculate the expected frequencies for the groups, then determining whether the division of the groups, called the observed frequencies, matches the expected frequencies.

The result of the test is a test statistic that has a chi-squared distribution and can be interpreted to reject or fail to reject the assumption or null hypothesis that the observed and expected frequencies are the same.

1. 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 lab07;

import java.util.Arrays;

/\*\*

 \*

 \* @author shadowleaf

 \*/

public class Lab07 {

    /\*\*

     \* @param args the command line arguments

     \*/

    public static void main(String[] args) {

        {

            System.out.println("KS Test");

            Double[] rn = {0.44, 0.81, 0.14, 0.05, 0.93};

            Arrays.sort(rn);

            Integer N = rn.length;

            Double Dplus = Double.MIN\_VALUE;

            Double Dminus = Double.MIN\_VALUE;

            Double i = 1.0;

            for (int idx = 0; idx < N; idx++) {

                Double newDplus = ((i / N) - rn[idx]);

                Double newDminus = rn[idx] - ((i - 1) / N);

                Dplus = Math.max(Dplus, newDplus);

                Dminus = Math.max(Dminus, newDminus);

//            System.out.printf("newDplus = %.10f, newDminus = %.10f\n", newDplus, newDminus);

                System.out.printf("Dplus\_%d = %.10f, Dminus\_%d = %.10f\n", idx + 1, newDplus, idx + 1, newDminus);

                i += 1.0;

            }

            Double D = Math.max(Dplus, Dminus);

            System.out.printf("Dplus : %.10f\nDminus : %.10f\nD : %.10f\n", Dplus, Dminus, D);

            Double alpha = 0.05;

        }

        {

            System.out.println("\nChi-Square Test");

            Integer[][] data\_obs = {{200, 150, 50}, {250, 300, 50}};

            Integer nrows = data\_obs.length;

            Integer ncols = data\_obs[0].length;

            Integer[] col\_total = new Integer[ncols];

            Integer[] row\_total = new Integer[nrows];

            Integer total = 0;

            int row, col;

            // calculate col totals

            for (col = 0; col < col\_total.length; col++) {

                col\_total[col] = 0;

                for (row = 0; row < nrows; row++) {

                    col\_total[col] += data\_obs[row][col];

                }

                total += col\_total[col];

            }

            // calculate row totals

            for (row = 0; row < row\_total.length; row++) {

                row\_total[row] = 0;

                for (col = 0; col < ncols; col++) {

                    row\_total[row] += data\_obs[row][col];

                }

            }

//            System.out.println(Arrays.toString(row\_total));

//            System.out.println(Arrays.toString(col\_total));

//            System.out.println(total);

            Double[][] data\_exp = new Double[nrows][ncols];

            // calculate expected values

            for (row = 0; row < nrows; row++) {

                for (col = 0; col < ncols; col++) {

                    data\_exp[row][col] = col\_total[col] \* row\_total[row] / (double) total;

//                    System.out.printf("%.5f ", data\_exp[row][col]);

                }

//                System.out.println();

            }

            Double chi\_sqr = 0.0;

            // calculate chi square

            for (row = 0; row < nrows; row++) {

                for (col = 0; col < ncols; col++) {

                    Double Oi = (double) data\_obs[row][col];

                    Double Ei = data\_exp[row][col];

                    chi\_sqr += (Oi - Ei) \* (Oi - Ei) / Ei;

                }

            }

            System.out.printf("Chi-Square = %.10f\n", chi\_sqr);

            Integer dfree = (nrows - 1) \* (ncols - 1);

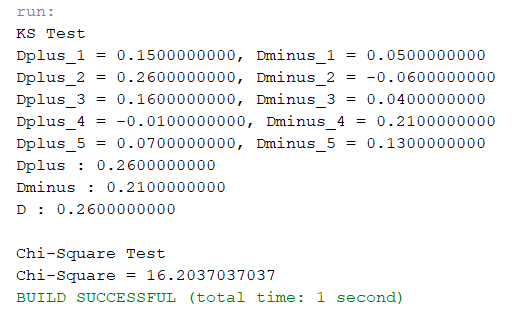
            Double alpha = 0.05;

        }

    }

}

1. Analysis and Discussions



1. Conclusions

**KS-Test**

Kolmogorov–Smirnov test a very efficient way to determine if two samples are significantly different from each other. It is usually used to check the uniformity of random numbers. Uniformity is one of the most important properties of any random number generator and Kolmogorov–Smirnov test can be used to test it.

The Kolmogorov–Smirnov test may also be used to test whether two underlying one-dimensional probability distributions differ. It is a very efficient way to determine if two samples are significantly different from each other.

The Kolmogorov–Smirnov statistic quantifies a distance between the empirical distribution function of the sample and the cumulative distribution function of the reference distribution, or between the empirical distribution functions of two samples.

H0(Null Hypothesis): Null hypothesis assumes that the numbers are uniformly distributed between 0-1. If we are able to reject the Null Hypothesis, this means that the numbers are not uniformly distributed between 0-1. Failure to reject the Null Hypothesis although does not necessarily mean that the numbers follow the uniform distribution.

**Chi-square Test**

Chi-square Test for Feature Extraction:

Chi-square test is used for categorical features in a dataset. We calculate Chi-square between each feature and the target and select the desired number of features with best Chi-square scores. It determines if the association between two categorical variables of the sample would reflect their real association in the population.

