ISSN Print: 2394-7500
ISSN Online: 2394-5869
Impact Factor (RJIF): 8.4
IJAR 2024; 10(8): 01-03
www.allresearchjournal.com
Received: 02-05-2024
Accepted: 08-06-2024

K Naga Vihari
Guest Faculty, Department of Statistics, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

A Vani
Research Scholar, Department of Statistics, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

M Bhupathi Naidu
Professor, Directorate of Distance Education, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

Corresponding Author:
K Naga Vihari
Guest Faculty, Department of Statistics, Sri Venkateswara University, Tirupati, Andhra Pradesh, India

Implementing chi-square test for goodness of fit in java

K Naga Vihari, A Vani and M Bhupathi Naidu

DOI: https://doi.org/10.22271/allresearch.2024.v10.i8a.11908

Abstract
The Chi-square test for goodness of fit is a statistical approach that determines if there is a significant difference between the actual and expected frequencies in categorical data. This article describes a Java implementation of the Chi-square test, including the steps for reading user input, computing the Chi-square statistic, comparing it to critical values, and drawing a conclusion based on a specified significance level.

Keywords: Java programming, chi-square statistic, statistical computation, significance level

Introduction
The Chi-square test is a non-parametric test that is widely used to compare observed data to expected data based on a specific hypothesis. This test is very useful for detecting whether observed categorical data differs from the expected distribution. This article describes how to implement the Chi-square test in Java, covering user input, statistical computations, and result interpretation.

Methodology
Initially the user is prompted to enter the number of categories. Next, the observed and predicted frequencies for each category are displayed.

Chi-square calculation
The fundamental step in the Chi-square test is to use the following calculation to get the Chi-square statistic:

$$\chi^2 = \sum(O_i - E_i)^2 / E_i$$

Degrees of Freedom
For the Chi-square test, the number of categories minus one is used to compute the degrees of freedom:

$$df = n-1$$

Critical value
The application makes use of a predetermined database of critical values for typical significance levels. In this implementation, we utilize a 0.05 significance level, which corresponds to a 95% confidence level.

Conclusion
The critical value and the Chi-square statistic are compared. A significant differences between the actual and expected data is determined by the program if the statistic is greater than the critical value.
Implementation:

Source code

```
importjava.util.Scanner;
public class ChiSquareTest {
public static void main(String[] args) {
Scanner scanner = new Scanner(System.in);

// Prompt the user to enter the number of categories
System.out.print("Enter the number of categories: ");
int n = scanner.nextInt();

// Initialize arrays for observed and expected frequencies
double[] observed = new double[n];
double[] expected = new double[n];

// Read observed frequencies from the user
System.out.println("Enter the observed frequencies:");
for (inti = 0; i< n; i++) {
System.out.print("Observed frequency " + (i + 1) + ": ");
observed[i] = scanner.nextDouble();
}

// Read expected frequencies from the user
System.out.println("Enter the expected frequencies:");
for (inti = 0; i< n; i++) {
System.out.print("Expected frequency " + (i + 1) + ": ");
expected[i] = scanner.nextDouble();
}

// Calculate the Chi-square statistic
doublechiSquare = calculateChiSquare(observed, expected, n);
System.out.println("Chi-square statistic: " + chiSquare);

// Degrees of freedom
intdegreesOfFreedom = n - 1;

// Significance level (e.g., 0.05 for 95% confidence level)
doublesignificanceLevel = 0.05;

// Get the critical value for the given degrees of freedom and significance level
doublecriticalValue = getCriticalValue(degreesOfFreedom, significanceLevel);

// Display the conclusion
if (chiSquare>criticalValue) {
System.out.println("Conclusion: The observed data is significantly different from the expected data at the " + (significanceLevel * 100) + "% significance level.");
} else {
System.out.println("Conclusion: There is no significant difference between the observed and expected data at the " + (significanceLevel * 100) + "% significance level.");
}

// Close the scanner
scanner.close();
}

// Method to calculate the Chi-square statistic
public static double calculateChiSquare(double[] observed, double[] expected, int n) {
doublechiSquare = 0.0;
for (inti = 0; i< n; i++) {
double difference = observed[i] - expected[i];
chiSquare += (difference * difference) / expected[i];
}

returnchiSquare;
}

// Method to get the critical value from the Chi-square distribution table
public static double getCriticalValue(intdegreesOfFreedom, double significanceLevel) {
// Chi-square distribution table for significance level 0.05 (95% confidence level)
// Only common values are included for simplicity
double[] criticalValues = {
3.841, // df = 1
5.991, // df = 2
7.815, // df = 3
9.488, // df = 4
11.070, // df = 5
12.592, // df = 6
14.067, // df = 7
15.507, // df = 8
16.919, // df = 9
18.307 // df = 10
};

// EnsuredegreesOfFreedom is within the range of our table
if (degreesOfFreedom>= 1 &&degreesOfFreedom<= criticalValues.length) {
returncriticalValues[degreesOfFreedom - 1];
} else {
System.out.println("Degrees of freedom out of range for critical value table.");
returnDouble.NaN; // Return NaN if degrees of freedom is out of range
}
}
}
```

Output

```
Enter the number of categories: 3
Enter the observed frequencies:
Observed frequency 1: 45
Observed frequency 2: 52
Observed frequency 3: 63
```

Enter the expected frequencies

```
Expected frequency 1: 63
Expected frequency 2: 54
Expected frequency 3: 63
Chi-square statistic: 5.216931216931218
```

Results

There is no significant difference between the observed and expected data at the 5.0% significance level.

Conclusion

The Chi-square test for goodness of fit is efficiently implemented in this Java program, giving users a tool to statistically assess their categorical data. The tool helps determine whether there are significant differences by comparing the observed data against expected frequencies,

which improves decision-making based on statistical evidence. Future enhancements can include connecting with statistical libraries for more accuracy and flexibility or adding a more comprehensive critical value table.

References

1. Pearson K. On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. Lond Edinb Dubl Philos Mag J Sci. 1900;50(302):157-175.
2. Cochran WG. The χ2 test of goodness of fit. Ann Math Stat. 1952;23(3):315-345.
3. Yates F. Contingency tables involving small numbers and the χ2 test. J R Stat Soc. 1934;1(2):217-235.
4. Ali A, Choudhary K, Sharma A. Object oriented based technique for software quality prediction through clustering and chi-square test. In2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT); c2015 Oct 29. p. 238-245.
5. Kembhavi S, Gupta R, Singh G. An Efficient Algorithm for Auto Upload and Chi-Square Test on Application Software. International Journal of Advanced Computer Research. 2013 Jun 1;3(2):121.