Національний технічний університет України

«Київський політехнічний інститут ім. Ігоря Сікорського»

Кафедра цифрових технологій в енергетиці

ЗВІТ   
з виконання лабораторної роботи №4  
з дисципліни «Математичні методи в психології»

«Перевірка статистичних гіпотез (параметричні критерії)»

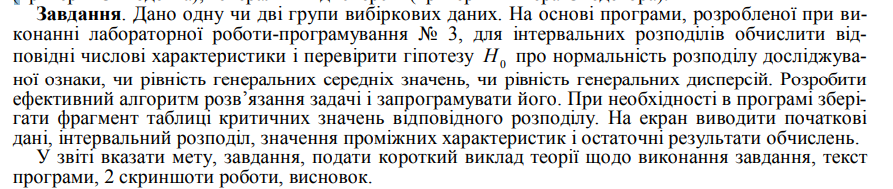
Варіант 8

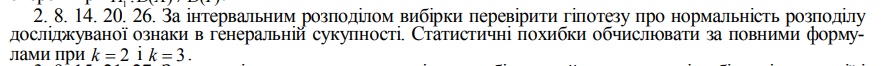
Виконав: студент групи ТР-12 Руденко В.І

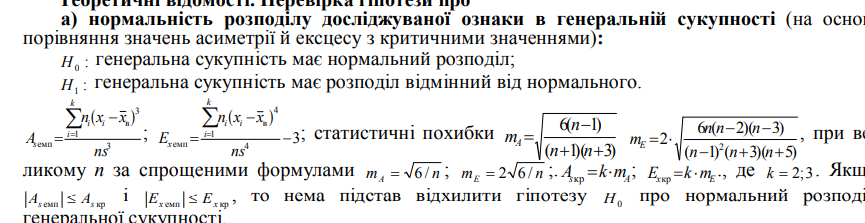
**Мета:**

Розробити програмні засоби для перевірки статистичних гіпотез про нормальність розподілу досліджуваної ознаки в генеральній сукупності (на основі порівняння значень асиметрії й ексцесу з критичними значеннями), рівність генеральних середніх значень (критерій Стьюдента), генеральних дисперсій (критерій Фішера-Снедекора).

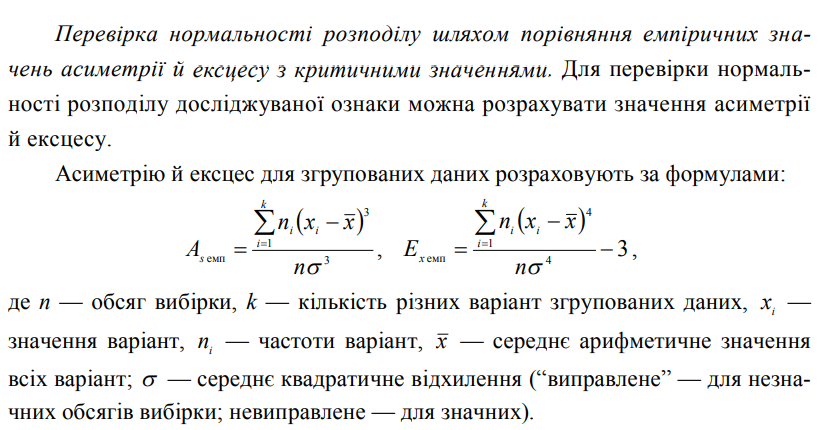
**Завдання:**

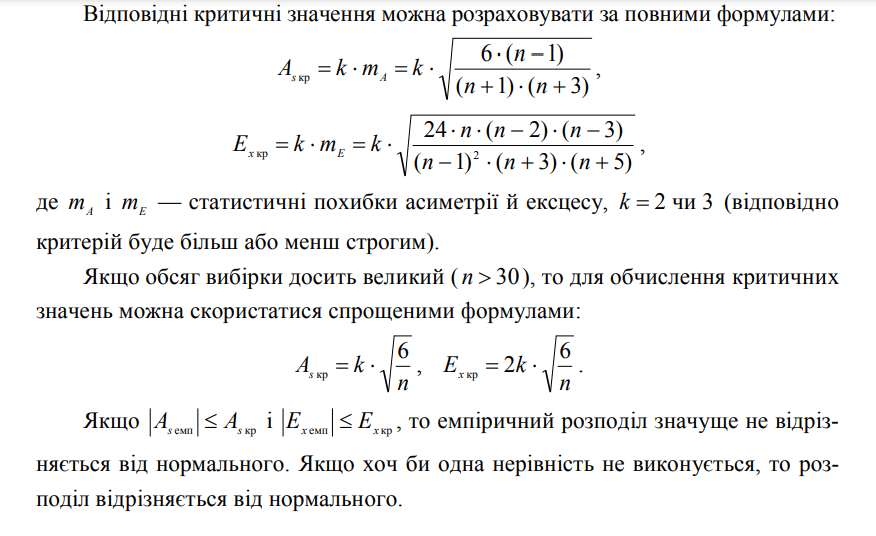






**Теорія:**





**Хід Роботи:**

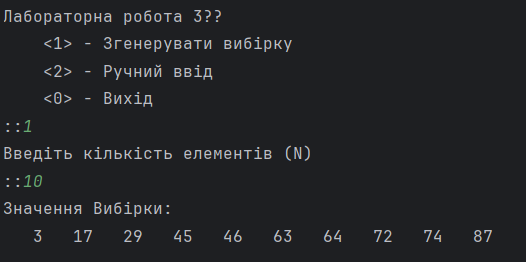


Table 1 Генерація Значень

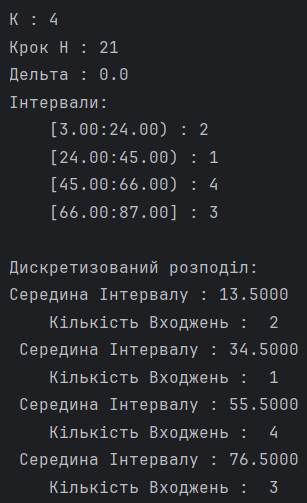


Table 2 Пошук Інтервалів

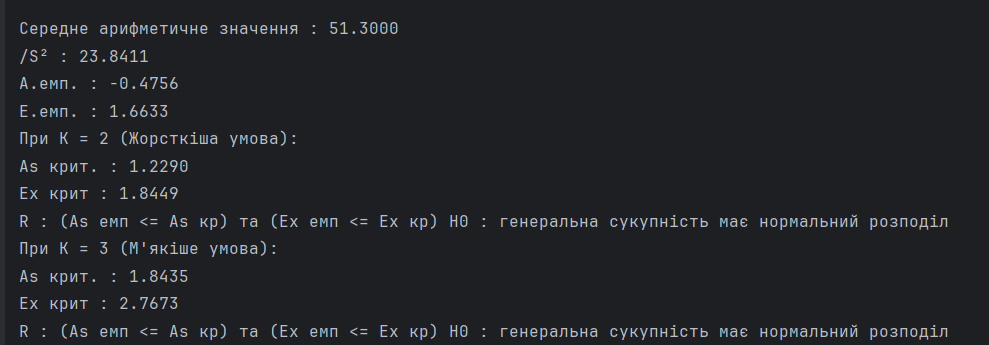


Table 3 Результат опрацювання

**Висновок:**

У ході лабораторної роботи було розроблено програмні засоби для перевірки статистичних гіпотез про нормальність розподілу досліджуваної ознаки в генеральній сукупності, а також про рівність генеральних середніх та дисперсій.

В результаті роботи за інтервальним розподілом перевірено гіпотезу про нормальність розподілу дос. Ознаки в генеральній сукупності. Програмний код здатен як і на автоматичне заповнення та опрацювання, так і на використання користувацьких значень.

**Код Програми:**  
*package LR5;  
  
  
import java.util.Arrays;  
import java.util.Random;  
import java.util.Scanner;  
  
public class Main  
{  
 enum UIState  
 {  
 Exit,  
 RandomGeneration,  
 ManualInput,  
 StergerGeneration,  
 InputWait;  
  
 public static UIState getByValue(int value) {  
 switch (value) {  
 case 0:  
 return Exit;  
 case 1:  
 return RandomGeneration;  
 case 2:  
 return ManualInput;  
 case 3:  
 return StergerGeneration;  
 default:  
 return InputWait;  
 }  
 }  
 }  
  
 private static int DiapasonCounter(double Array[], double LowerBound, double UpperBound, boolean lastBound)  
 {  
 int count = 0;  
 for (int j = 0; j < Array.length; j++) {  
 if (Array[j] >= LowerBound && Array[j] < UpperBound)  
 {  
 count++;  
 }  
 if(lastBound && Array[j] == UpperBound)  
 {  
 count++;  
 }  
 }  
 return count;  
 }  
  
  
 public static void main(String[] args)  
 {  
 System.out.println("<><><><>");  
 ConsoleUI();  
 }  
  
 private static void ConsoleUI()  
 {  
 Scanner scan = new Scanner(System.in);  
 UIState CurrentProgramPart = UIState.InputWait;  
 while (true) {  
 System.out.print("Лабораторна робота 3??\n\t<1> - Згенерувати вибірку\n\t<2> - Ручний ввід\n\t<0> - Вихід\n::");  
 CurrentProgramPart = UIState.getByValue(scan.nextInt());  
 switch (CurrentProgramPart) {  
 case UIState.Exit:  
 return;  
 case UIState.InputWait:  
 System.out.println("Невіриний ввід....");  
 break;  
 default:  
 MainRun(CurrentProgramPart, scan);  
 break;  
 }  
  
 }  
 }  
  
 private static UIState WorkTypeUI(UIState currentProgramPart)  
 {  
 Scanner scan = new Scanner(System.in);  
 while(true)  
 {  
 System.out.print("\n\t<1> - Автоматичний метод Стерджерса\n\t<2> - Ручний ввід\n\t<0> - Вихід\n::");  
 currentProgramPart = UIState.getByValue(scan.nextInt()) ;  
 switch (currentProgramPart)  
 {  
 case UIState.Exit:  
 System.exit(0);  
 case UIState.RandomGeneration:  
 return UIState.RandomGeneration;  
 case ManualInput:  
 return UIState.ManualInput;  
 default:  
 System.out.println("Невіриний ввід....");  
 break;  
 }  
 System.out.println("<><><><>");  
  
 }  
 }  
  
 private static void MainRun(UIState currentProgramPart, Scanner scan)  
 {  
 int N;  
 do  
 {  
 System.out.print("Введіть кількість елементів (N)\n::");  
 N = scan.nextInt();  
 } while(N < 1);  
 double[] Array = new double[N];  
 switch (currentProgramPart) {  
 case UIState.RandomGeneration:  
 Random rand = new Random(System.currentTimeMillis());  
 for (int i = 0; i < N; i++) {  
 Array[i] = rand.nextInt(N\*10);  
 }  
 break;  
 case UIState.ManualInput:  
 if (N < 10) {  
 System.out.printf("Введіть %d Значеннь підряд\n::", N);  
 }  
 for (int i = 0; i < N; i++) {  
 if (N > 10) {  
 System.out.printf("Введіть %d Значення\n::", i + 1);  
 }  
 Array[i] = scan.nextDouble();  
 }  
 scan.nextLine();  
 break;  
 default:  
 return;  
 }  
 Arrays.sort(Array);  
 //Base Output//////////  
 System.out.println("Значення Вибірки:");  
 for (int i = 0; i < N; i++) {  
 if (Array[i] == Math.floor(Array[i])) {  
 System.out.printf("%4.0f ", Array[i]);  
 } else {  
 System.out.printf("%4.1f ", Array[i]);  
 }  
 }  
 System.out.println();  
 ///////////////////////  
 currentProgramPart = WorkTypeUI(currentProgramPart);  
 String OutputString[];  
 double workSpace[][];  
 if (currentProgramPart == UIState.RandomGeneration)  
 {  
 int k = (int) Math.round(1 + (Math.log(N) / Math.log(2)));  
 System.out.println("K : " + k);  
 OutputString = new String[k];  
 workSpace = new double[k][2];  
 double Max = Arrays.stream(Array).max().getAsDouble();  
 double Min = Arrays.stream(Array).min().getAsDouble();  
 int h = (int) Math.ceil((Max - Min) / k);  
 System.out.println("Крок H : " + h);  
 double delta = (k \* h - (Max - Min)) / 2;  
 System.out.println("Дельта : " + delta);  
 double startRange = (double) (Min - delta);  
 for (int i = 0; i < k; i++) {  
 double endRange = startRange + h;  
 int count = DiapasonCounter(Array, startRange, endRange, (k - 1 == i));  
 OutputString[i] = String.format("[%.2f:%.2f%s : %d\n", startRange, endRange, (k - 1 == i ? "]" : ")"), count);  
 workSpace[i] = new double[]{((startRange + endRange) / 2), count};  
 startRange = endRange;  
 }  
 }  
 else  
 {  
 double CurrentMin = Array[0];  
 double CurrentMax = Array[0];  
 CurrentMax = Arrays.stream(Array).max().orElse(Double.NaN);  
 CurrentMin = Arrays.stream(Array).min().orElse(Double.NaN);  
  
 System.out.print("ВВедіть бажану кількість інтервалів\n::");  
 int IntervalNumber = scan.nextInt();  
 scan.nextLine();  
 OutputString = new String[IntervalNumber];  
 workSpace = new double[IntervalNumber][2];  
 if (IntervalNumber < 1) {  
 System.out.println("Пора спати");  
 System.exit(1);  
 }  
  
 for (int i = 0; i < IntervalNumber; ) {  
 System.out.printf("Введіть межі %d Інтервалу\n::", (int) (i + 1));  
 double LowBorder = scan.nextInt(),  
 HighBorder = scan.nextInt();  
 scan.nextLine();  
 if ((LowBorder == HighBorder) || (i == 0 && LowBorder > CurrentMin) || (i + 1 == IntervalNumber && HighBorder < CurrentMax) || (i != 0 && LowBorder != CurrentMin)) {  
 System.out.println("Нє");  
 continue;  
 }  
 int count = DiapasonCounter(Array, LowBorder, HighBorder, (i + 1 == IntervalNumber));  
 workSpace[i] = new double[]{((LowBorder + HighBorder) / 2), count};  
 OutputString[i] = String.format("[%.2f:%.2f%s : %d\n", LowBorder, HighBorder, (i + 1 == IntervalNumber ? "]" : ")"), count);  
 CurrentMin = HighBorder;  
 i++;  
 }  
  
 }  
 String[] OutputSupportString = new String[OutputString.length];  
 System.out.println("Інтервали:");  
 for (int i = 0; i < OutputString.length; i++) {  
 System.out.printf("\t%s", OutputString[i]);  
 }  
 System.out.println();  
 ///////////////////  
 double Xc = 0;  
 System.out.println("Дискретизований розподіл:");  
 {  
 int i = 0;  
 for (double[] entry : workSpace)  
 {  
 Xc += entry[0]\*entry[1];  
 OutputSupportString[i++] = String.format("Середина Інтервалу : %2.4f\n\tКількість Входжень : %2.0f\n", (double) entry[0],(double) entry[1]);  
 }  
 for(i = 0; i < OutputSupportString.length; i++)  
 {  
 System.out.printf("%s ",OutputSupportString[i]);  
 }  
 }  
 System.out.println();  
 Xc/=N;  
 System.out.println("Середне арифметичне значення : "+String.format("%.4f",Xc));  
 double S2 = 0;  
 for (double[] entry : workSpace)  
 {  
 S2 += entry[1] \* Math.pow(entry[0] - Xc, 2);  
 }  
 S2 = Math.sqrt(S2/(N-1));  
 System.out.printf("/S² : %.4f\n",S2);  
 double Ae = 0, Ee = 0;  
 for (double[] entry : workSpace)  
 {  
 Ae += entry[1] \* Math.pow(entry[0] - Xc, 3);  
 Ee += entry[1] \* Math.pow(entry[0] - Xc, 4);  
 }  
 Ae = Ae / (N\*Math.pow(S2,3));  
 Ee = Ee / (N\*Math.pow(S2,4));  
 System.out.println("A.емп. : "+String.format("%.4f",Ae));  
 System.out.println("E.емп. : "+String.format("%.4f",Ee));  
 double Ma = 0, Me = 0;  
 if(N < 30)  
 {  
 Ma = Math.sqrt((double) (6 \* (N - 1)) /((N+1)\*(N+3)));  
 Me = Math.sqrt((24\*N\*(N-2)\*(N-3))/((N+3)\*(N+5)\*Math.pow((N-1),2)));  
 }  
 else  
 {  
 Ma = Math.sqrt((float)6/N);  
 Me = Ma\*2;  
 }  
 double Ak = Ma, Ek = Me;  
 {  
 System.out.println("При K = 2 (Жорсткіша умова):");  
 Ak \*= 2;  
 Ek \*= 2;  
 FinalResulOut(Ae,Ee,Ak,Ek);  
 }  
 Ak = Ma;  
 Ek = Me;  
 {  
 System.out.println("При K = 3 (М'якіше умова):");  
 Ak \*= 3;  
 Ek \*= 3;  
 FinalResulOut(Ae,Ee,Ak,Ek);  
 }  
 }  
 static void FinalResulOut(double Ae, double Ee,double Ak, double Ek)  
 {  
 String  
 H0 = "(As емп <= As кр) та (Ex емп <= Ex кр) H0 : генеральна сукупність має нормальний розподіл",  
 H1 = "(As емп > As кр) та (Ex емп > Ex кр) H1 : генеральна сукупність має розподіл відмінний від нормального";  
 String result = "";  
 if(Math.abs(Ae) <= Ak && Math.abs(Ee) <= Ek)  
 {  
 result = H0;  
 }  
 else  
 {  
 result = H1;  
 }  
 System.out.println("As крит. : "+String.format("%.4f",Ak));  
 System.out.println("Ex крит : "+String.format("%.4f",Ek));  
 System.out.printf("R : %s",result);  
 System.out.println();  
 }  
}*