

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Ижевский государственный технический университет
имени М.Т. Калашникова»
(ФГБОУ ВО «ИжГТУ имени М.Т. Калашникова»)
Факультет «Информационные технологии»
Кафедра «Информационные системы»

ОТЧЕТ
по лабораторной работе №1
на тему «Розыгрыш стандартных дискретных случайных величин»
по дисциплине «Теория вероятностей и математическая статистика»

Выполнил

студент группы Б22-791-1

Д.С. Леонтьев

Проверил

доктор тех. наук, проф.

В.Г. Суфиянов

ИЖЕВСК 2024

Постановка задачи

Написать программу для розыгрыша геометрического распределения с параметром $p = 0,25$. Записать в файл 100 разыгранных значений.

Краткие теоретические сведения

Геометрическое распределение с параметром p .

$$P(\xi = m) = pq^{m-1}, \text{ где } m = 1, 2, \dots$$

Ход работы:

Ниже представлен алгоритм для розыгрыша геометрического распределения с параметром p .

```
1   $\alpha[1..100]$  (массив заполненный случайными значениями от 0 до 1)
2   $\xi[1..100]$ 
3  for  $i = 1$  to 100
4       $m = 1$ 
5       $P_m = p$ 
6       $\gamma = P_m$ 
7      while  $\alpha[i] \geq \gamma$ 
8           $m = m + 1$ 
9           $P_m = P_m * q$ 
10          $\gamma = \gamma + P_m$ 
11      $\xi[i] = m$ 
12 return  $\xi$ 
```

Результаты работы

Ниже представлена программа на языке C#, реализующая алгоритм, описанный выше.

```
// Генерируем случайное число и добавляем его в список
Random random = new Random();
```

```

List<double> P = new List<double>();
List<double> alpha = new List<double>();
List<int> ksi = new List<int>(new int[100]);

double p = 0.25;
double q = 1 - p;

string path = @"C:\Users\leone\Desktop\Учеба\ТИ и
тервер\ТерВер\Demo_2\content.txt";
string path1 = @"C:\Users\leone\Desktop\Учеба\ТИ и
тервер\ТерВер\Demo_2\content1.txt";

for (int i = 0; i < 100; i++)
{
    double rnd = random.NextDouble();
    alpha.Add(rnd);
}

for (int i = 0; i < 100; i++)
{
    int m = 0;
    double pm = p;
    double gamma = pm;
    while (alpha[i] >= gamma)
    {
        m++;
        pm *= q;
        gamma += pm;
    }
    ksi[i] = m;
}

for (int i = 0; i < 100; i++)
{
    numbersList.Add(ksi[i]);
}
File.WriteAllLines(path1, ksi.ConvertAll(x => x.ToString()));
File.WriteAllText(path1, string.Join(" ", ksi));
ksi.Sort();
File.WriteAllLines(path, ksi.ConvertAll(x => x.ToString()));
File.WriteAllText(path, string.Join(" ", ksi));

// Создание словаря для хранения частоты появления чисел
Dictionary<int, int> frequency = new Dictionary<int, int>();

foreach (int num in numbersList)
{
    if (frequency.ContainsKey(num))
    {
        frequency[num]++;
    }
    else
    {
        frequency[num] = 1;
    }
}
// Очистка графика перед построением нового

chart1.Series.Clear();
chart1.ChartAreas[0].AxisX.Interval = 1;

```

```
// Создание серии для отображения данных на графике
Series series = new Series("Частота появления чисел");
series.ChartType = SeriesChartType.Column;

// Добавление точек на график (значение x – число, значение y – частота)

foreach (KeyValuePair<int, int> entry in frequency)
{
    series.Points.AddXY(entry.Key, (float)entry.Value/100);
}
chart1.ChartAreas[0].AxisX.Title = "Значение";
chart1.ChartAreas[0].AxisY.Title = "Частота";
// Добавление серии на график
chart1.Series.Add(series);
frequency.Clear();
numbersList.Clear();
```

Величины, полученные в результате работы программы, представлены в виде дискретного вариационного ряда и изображены на рисунке 1 и вариационный ряд на рисунке 2.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	3	3	4
4	4	4	4	5	5	5	6	6	6	7	7	7	7	8	8	8	9	10
11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	11	13

Рисунок 1 – Дискретный вариационный ряд

8	2	11	2	4	0	0	0	0	3	1	1	0	4	1	2	1	4
4	2	3	2	0	3	0	2	0	3	1	0	6	0	0	3	0	2
7	6	1	0	10	4	13	0	0	3	9	2	2	7	0	7	3	
0	2	1	1	1	1	0	0	2	6	11	5	1	1	2	3	5	
3	2	1	2	2	8	3	11	1	2	2	2	0	2	0	11	2	
0	5	0	2	2	0	0	1	7	0	1	8						

Рисунок 2 – Вариационный ряд

В результате работы программы была построена гистограмма относительных частот, которая изображена на рисунке 3.

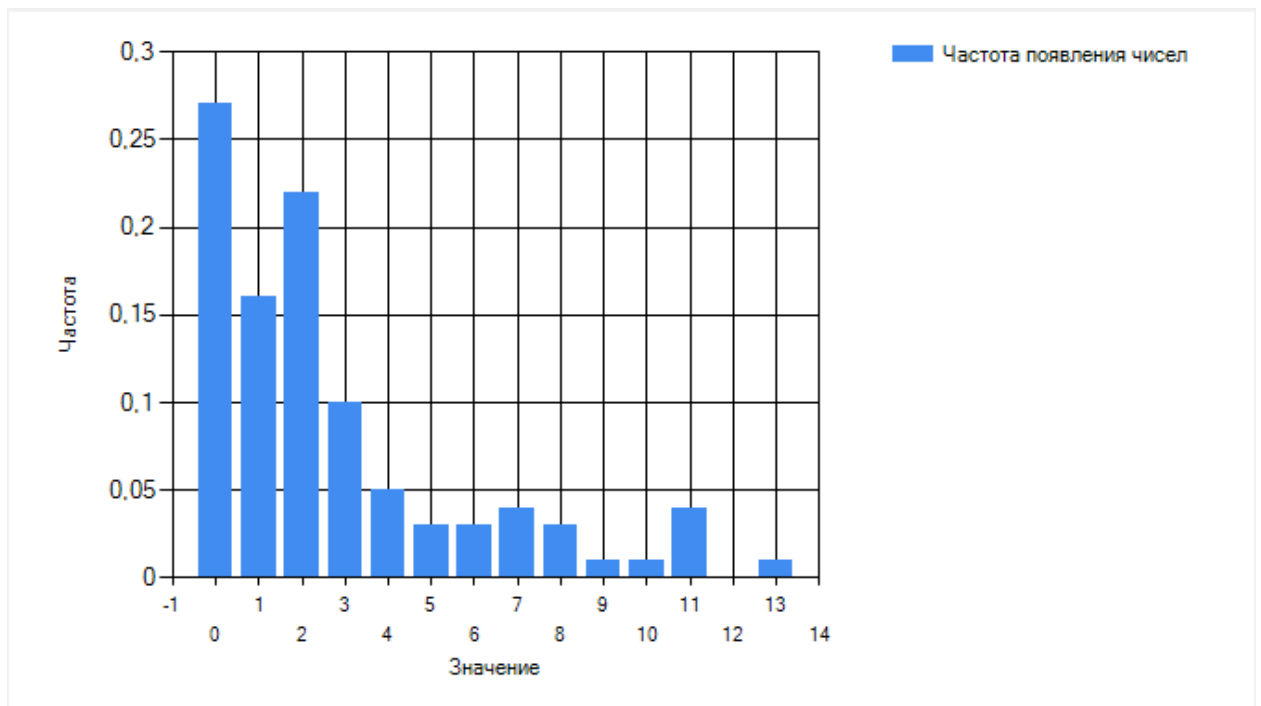


Рисунок 2 – Гистограмма относительных частот

Выводы

В результате выполнения лабораторной работы была разработана программа для генерации геометрического распределения с параметром $p = 0.25$. А также были изучены методы генерации дискретных случайных величин. Кроме того, были записаны 100 разыгранных значений в файл и построена гистограмма частот для визуализации распределения. Полученные результаты демонстрируют успешную реализацию поставленных целей и позволяют провести анализ полученных данных