МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего образования
«МИРЭА – Российский технологический университет»

ИНСТИТУТ КИБЕРНЕТИКИ

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 1
по курсу «**Теория вероятностей и математическая статистика, часть 2**»

Тема: ________**Первичная обработка выборки из**_______________

________**дискретной генеральной совокупности**___________

Выполнил:
Студент 3-го курса
Петров С.В.

Группа: КМБО-03-17

М О С К В А   2 0 2 0

Лабораторная работа по Математической статистике № 1

«Первичная обработка выборки из дискретной генеральной совокупности»

Задание 1. Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по биномиальному закону с параметрами n и p .

$n=5+ V \bmod 16$ $p=0{,}3 + 0{,}005V$

Задание 2. Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по геометрическому закону с параметром p.

$p=0{,}3 + 0{,}005V$

Задание 3. Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по закону Пуассона с параметром λ.

$\lambda = 0{,}5 + 0{,}01V$

Для всех выборок построить:

1) статистический ряд;

2) полигон относительных частот;

3) график эмпирической функции распределения;

найти:

1) выборочное среднее;

2) выборочную дисперсию;

3) выборочное среднее квадратическое отклонение;

4) выборочную моду;

5) выборочную медиану;

6) выборочный коэффициент асимметрии;

7) выборочный коэффициент эксцесса.

Следуя Указаниям провести сравнение рассчитанных характеристик с теоретическими значениями. V=54 – номер варианта.

Вычисления проводить с точностью до 0,00001.

Краткие теоретические сведения

Полученную выборку $\{x_1, x_2, x_3, ..., x_N\}$ упорядочить по возрастанию, определить частоты n_i и относительные частоты w_i, построить статистический ряд вида:

x_k^*	n_k	w_k	s_k
x_1^*	n_1	w_1	s_1
x_2^*	n_2	w_2	s_2
…	…	…	…
x_m^*	n_m	w_m	s_m

$x_i^* < x_j^*$ при i < j, n_i– частота x_i^* (число значений x_i^*, встречающихся в выборке), $\sum_{i=1}^{m} n_i$; w_i – относительная частота (частотность) значения x_i^*, $w_i = \frac{n_i}{N}$, $\sum_{i=1}^{m} w_i = 1$; $s_k = \sum_{j=1}^{k} w_j$, $s_1 = w_1$, $s_m = 1$

Полигон относительных частот – ломаная линия, соединяющая последовательно точки с координатами $(0,\widetilde{w}_0)$, $(1,\widetilde{w}_1)$, …, $(M,\widetilde{w}_M)$, где $M = x_m^* = \max\{x_i^* : 1 \le i \le m\}$; $\widetilde{w}_j = w_i$, если существует такое x_i^* , что $j = x_i^*$ и $\widetilde{w}_j = 0$ в противном случае.

Эмпирическая функция распределения

$$F_N^{Э}(x) = \sum_{x_i^* \le x} w_i = \begin{cases} 0, & x < x_1^*, \\ w_1, & x_1^* \le x < x_2^*, \\ w_1 + w_2, & x_2^* \le x < x_3^*, \\ w_1 + w_2 + w_3, & x_3^* \le x < x_4^*, \\ \dots\dots\dots\dots\dots\dots, & \\ 1, & x \ge x_m^*. \end{cases}$$

Выборочное среднее

$$\overline{x} = \frac{1}{N}\sum_{i=1}^{m} x_i^* \cdot n_i = \sum_{i=1}^{m} x_i^* \cdot w_i$$

Выборочная дисперсия

$$D_B = \sum_{i=1}^{m}(x_i^* - \overline{x})^2 \cdot w_i = \sum_{i=1}^{m}(x_i^*)^2 \cdot w_i - (\sum_{i=1}^{m} x_i^* \cdot w_i)^2 = \overline{x^2} - (\overline{x})^2$$

Выборочный момент k-ого порядка

$$\overline{\mu}_k = \overline{x^k} = \sum_{i=1}^{m}(x_i^*)^k \cdot w_i\ ,\ \overline{\mu}_1 = \overline{x}$$

Выборочный центральный момент k-ого порядка

$$\begin{aligned}
&\overline{\mu}_k^0 = \sum_{i=1}^{m}(x_i^* - \overline{x})^k \cdot w_i\ ,\ \overline{\mu}_1^0 = 0\ ,\ \overline{\mu}_2^0 = D_B = \overline{\mu}_2 - (\overline{\mu}_1)^2,\\
&\overline{\mu}_3^0 = \sum_{i=1}^{m}((x_i^*)^3 - 3(x_i^*)^2\overline{\mu}_1 + 3x_i^*\left(\overline{\mu}_1\right)^2 - (\overline{\mu}_1)^3)\cdot w_i = \overline{\mu}_3 - 3\overline{\mu}_2\overline{\mu}_1 + 2(\overline{\mu}_1)^3,\\
&\overline{\mu}_4^0 = \sum_{i=1}^{m}((x_i^*)^4 - 4(x_i^*)^3\overline{\mu}_1 + 6(x_i^*)^2\left(\overline{\mu}_1\right)^2 - 4x_i^*(\overline{\mu}_1)^3 + (\overline{\mu}_1)^4)\cdot w_i =\\
&= \overline{\mu}_4 - 4\overline{\mu}_3\overline{\mu}_1 + 6\overline{\mu}_2(\overline{\mu}_1)^2 - 3(\overline{\mu}_1)^4
\end{aligned}$$

Выборочное среднее квадратическое отклонение

$$\overline{\sigma} = \sqrt{D_B}$$

Выборочная медиана

$$\overline{M}_e = \begin{cases} x_i^*, & F_N^{Э}(x_{i-1}^*) < 0{,}5 < F_N^{Э}(x_i^*), \\ \frac{1}{2}(x_i^* + x_{i+1}^*), & F_N^{Э}(x_i^*) = 0{,}5\,. \end{cases}$$

Выборочная мода $\bar{M}_0$ (это значение x_i^*, которому соответствует наибольшая частота)

$$\bar{M}_0 = \{x_i^* \mid n_i = \max n_k\}, \text{ если } \; n_i = \max n_k > n_j,\ i \neq j;$$

$$\text{если } n_i = n_{i+1} = \ldots = n_{i+j} = \max n_k, \text{ то } \bar{M}_0 = \frac{1}{2}(x_i^* + x_{i+j}^*),$$

$$\text{если } n_i = n_j = \max n_k > n_l,\ i < l < j, \text{ то } \bar{M}_0 - \text{ не существует.}$$

Выборочный коэффициент асимметрии

$$\bar{\gamma}_1 = \frac{\bar{\mu}_3^0}{\bar{\sigma}^3}$$

Выборочный коэффициент эксцесса

$$\bar{\gamma}_2 = \frac{\bar{\mu}_4^0}{\bar{\sigma}^4} - 3$$

Биномиальное распределение

Характеристика	Значение
Математическое ожидание	np
Дисперсия	npq , $q = 1 - p$
Среднее квадратичное отклонение	$\sqrt{npq}$
Мода	$[(n+1)p]$, *если* $(n+1)p$ – дробное $(n+1)p - \frac{1}{2}$, *если* $(n+1)p$ - целое
Медиана	*Round(np)*
Коэффициент асимметрии	$\frac{q - p}{\sqrt{npq}}$
Коэффициент эксцесса	$\frac{1 - 6pq}{npq}$

Геометрическое распределение

Характеристика	Значение
Математическое ожидание	$\frac{q}{p}$, $q = 1 - p$
Дисперсия	$\frac{q}{p^2}$, $q = 1 - p$
Среднее квадратичное отклонение	$\frac{\sqrt{q}}{p}$
Мода	0
Медиана	$[-\frac{\ln 2}{\ln q}]$, если $\frac{\ln 2}{\ln q}$ – дробное $-\frac{\ln 2}{\ln q} - \frac{1}{2}$, если $\frac{\ln 2}{\ln q}$ - целое
Коэффициент асимметрии	$\frac{2\ -\ p}{\sqrt{q}}$
Коэффициент эксцесса	$6 + \frac{p^2}{q}$

Распределение Пуассона

Характеристика	Значение
Математическое ожидание	λ
Дисперсия	λ
Среднее квадратичное отклонение	$\sqrt{\lambda}$
Мода	$[\lambda]$
Медиана	$[\lambda + \frac{1}{3} - \frac{0.02}{\lambda}]$
Коэффициент асимметрии	$\lambda^{-\frac{1}{2}}$
Коэффициент эксцесса	λ^{-1}

Средства высокоуровневого интерпретируемого языка программирования Python, которые использованы в программе расчета

numpy – модуль для научных вычислений

math – модуль с основными математическими функциями и операциями

matplotlib – модуль для работы с графиками

numpy.random.binomial(n,p,200) – генерация N=200 псевдослучайных чисел, распределенных по биномиальному закону с параметрами n и p;

numpy.random.geometric(p,200) – генерация N=200 псевдослучайных чисел, распределенных по геометрическому закону с параметром p;

numpy.random.poisson(lambda,200) – генерация N=200 псевдослучайных чисел, распределенных по закону Пуассона с параметром lambda.

sorted(x) – упорядочение по возрастанию коллекции x;

max(x) – выбор максимального значения в коллекции x;

print("text") – вывод в командное окно строки "text";

for: … , while: … – циклы с предусловием;

if: … elif: … else: … – конструкция условного оператора;

break – оператор прерывания цикла;

math.factorial(n) – вычисление значения факториала n;

math.exp(n) – вычисление экспоненты в степени n

zip() – итератор по нескольким коллекциям

list() - конструктор списка

[<some_expression> for <some_iterator> in <some_collection>] – генератор списка

pyplot.axis([xmin,xmax,ymin,ymax]) - вывод части графика, определяемую прямоугольной областью $xmin \le x \le xmax$, $ymin \le y \le ymax$;

pyplot.grid(True) - нанесение сетки на график;

pyplot.figure() –создание графического окна;

pyplot.plot(x, y) – создание графика функции

pyplot.arrow(x, y, dx, dy) – создание стрелки

pyplot.show() – отображение всех графиков

Результаты расчетов

Задание 1) Распределение по биномиальному закону

n=11 p=0.57

Полученная выборка

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

Упорядоченная выборка

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

Статистический ряд

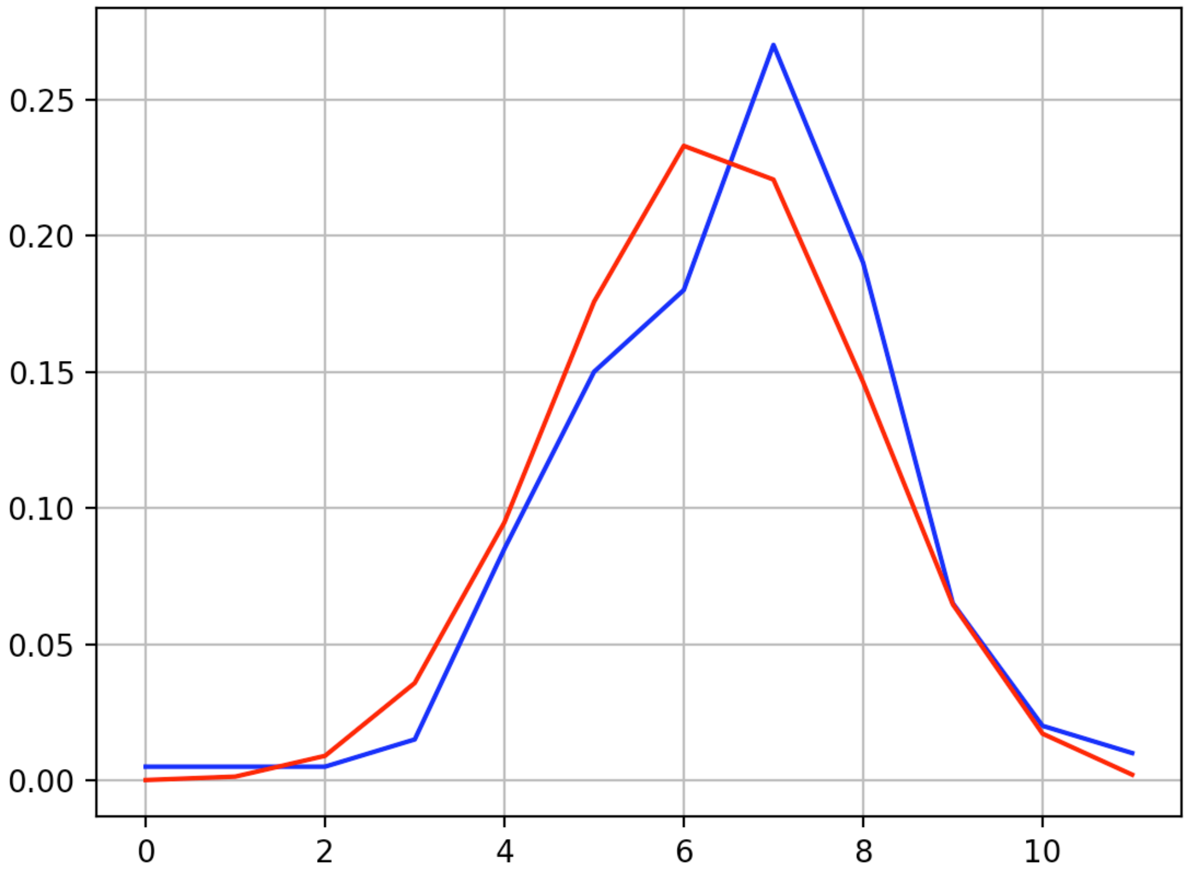
x_k^*	n_k	w_k	s_k
0	1	0.005	0.005
1	1	0.005	0.01
2	1	0.005	0.015
3	3	0.015	0.03

4	17	0.085	0.115
5	30	0.15	0.265
6	36	0.18	0.445
7	54	0.27	0.715
8	38	0.19	0.905
9	13	0.065	0.97
10	4	0.02	0,99
11	2	0.01	1

Результаты расчетов требуемых характеристик

Характеристика	Значение
выборочное среднее	6.535
выборочная дисперсия	2.908775
выборочное среднее квадратическое отклонение	1.70551
выборочная мода	7
выборочная медиана	7
выборочный коэффициент асимметрии	-0.4045
выборочный коэффициент эксцесса	0.87332

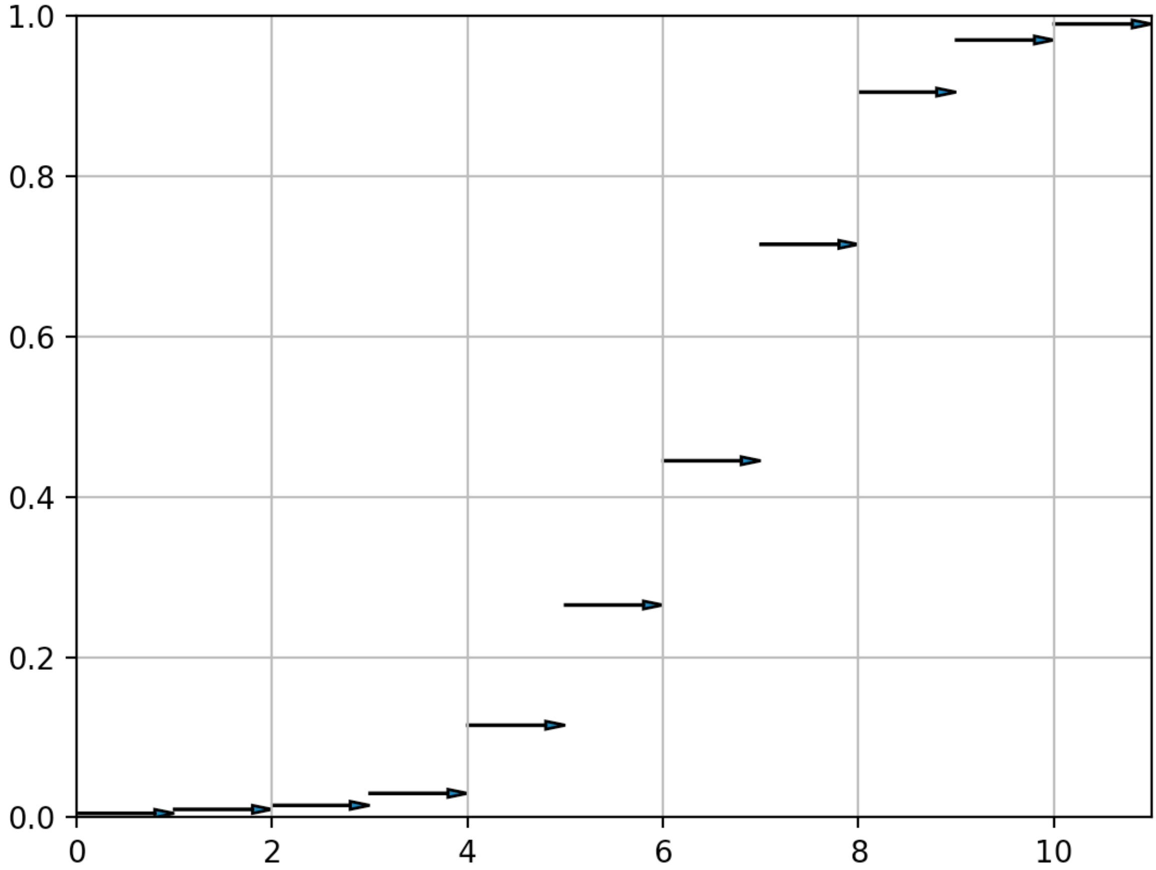
График полигона относительных частот обозначен синим цветом, а график полигона соответствующих теоретических вероятностей – красным.



Эмпирическая функция распределения:

$$F_{200}^{Э}(x) = \sum_{x_i^* \le x} w_i = \begin{cases} 0, x < 0 \\ 0.005, 0 \le x < 1 \\ 0.01, 1 \le x < 2 \\ 0.015, 2 \le x < 3 \\ 0.03, 3 \le x < 4 \\ 0.115, 4 \le x < 5 \\ 0.265, 5 \le x < 6 \\ 0.445, 6 \le x < 7 \\ 0.715, 7 \le x < 8 \\ 0.905, 8 \le x < 9 \\ 0.97, 9 \le x < 10 \\ 0.99, 10 \le x < 11 \\ 1, x \ge 11 \end{cases}$$

График эмпирической функции



Задание 2) Распределение по геометрическому закону

p=0.57

Полученная выборка

0	0	2	0	0	0	0	1	0	1
0	0	3	3	0	2	0	0	1	0
0	1	0	0	1	0	1	0	2	0
0	0	0	1	0	2	1	0	4	1
0	0	1	3	0	0	0	1	0	0
1	0	0	0	0	0	2	0	3	2
0	1	6	4	4	1	0	0	0	3

1	1	1	2	0	0	1	2	0	0
0	1	0	0	0	1	0	1	4	0
0	3	1	1	1	1	0	2	1	0
4	0	2	1	0	0	0	0	1	0
0	0	0	0	2	0	1	4	4	2
0	3	1	0	0	1	1	0	2	0
0	0	0	0	0	0	5	1	0	4
2	1	1	1	0	2	2	2	0	1
0	0	0	0	2	1	0	0	2	3
1	0	1	2	1	0	0	0	1	0
1	1	0	0	0	1	2	1	1	1
0	0	0	3	0	3	0	0	0	1
0	0	2	1	1	0	2	2	1	1

Упорядоченная выборка

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	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	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	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	1
1	1	1	1	1	1	1	1	1	1

1	1	1	1	1	1	1	1	1	1
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	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	5	6

Статистический ряд

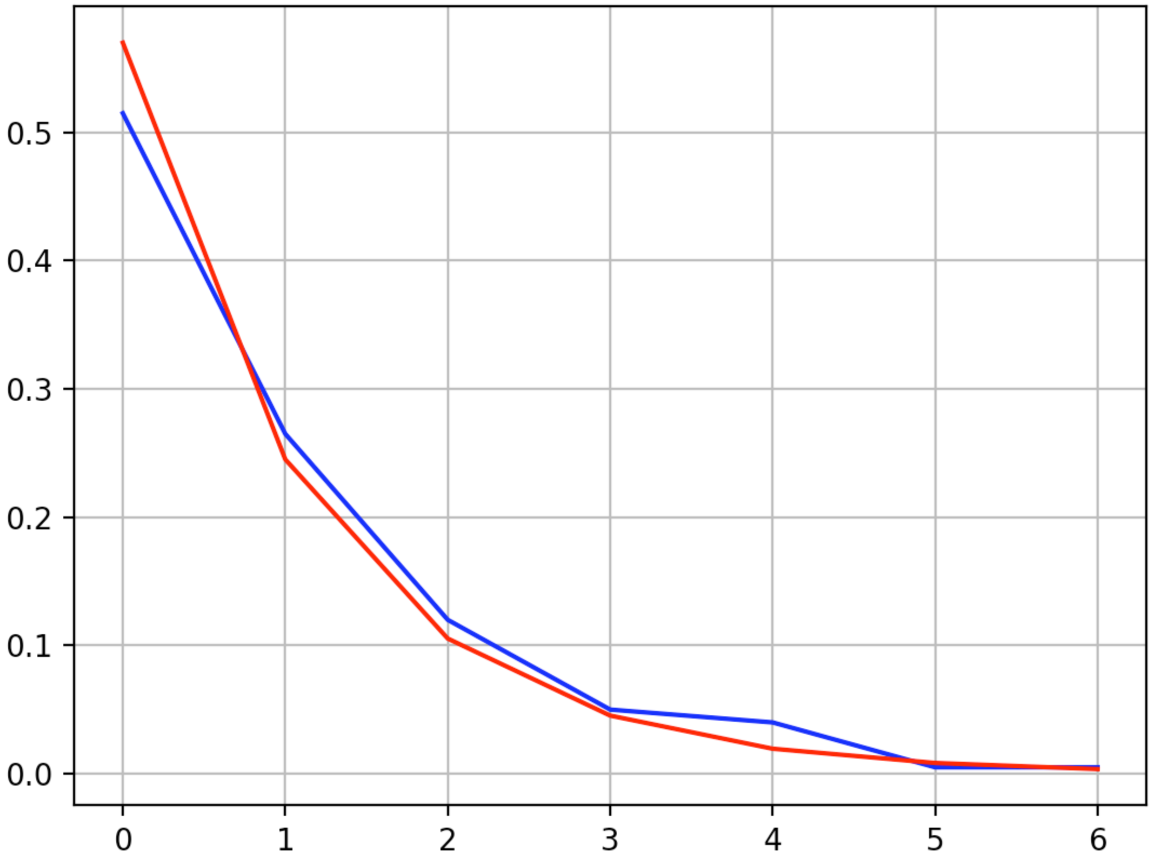
x_k^*	n_k	w_k	s_k
0	103	0.515	0.515
1	53	0.265	0.78
2	24	0.12	0.9
3	10	0.05	0.95
4	8	0.04	0.99
5	1	0.005	0.995
6	1	0.005	1

Результаты расчетов требуемых характеристик

Характеристика	Значение
выборочное среднее	0.87
выборочная дисперсия	1.3831
выборочное среднее квадратическое отклонение	1.17605
выборочная мода	0
выборочная медиана	1

выборочный коэффициент асимметрии	1.58098
выборочный коэффициент эксцесса	2.37213

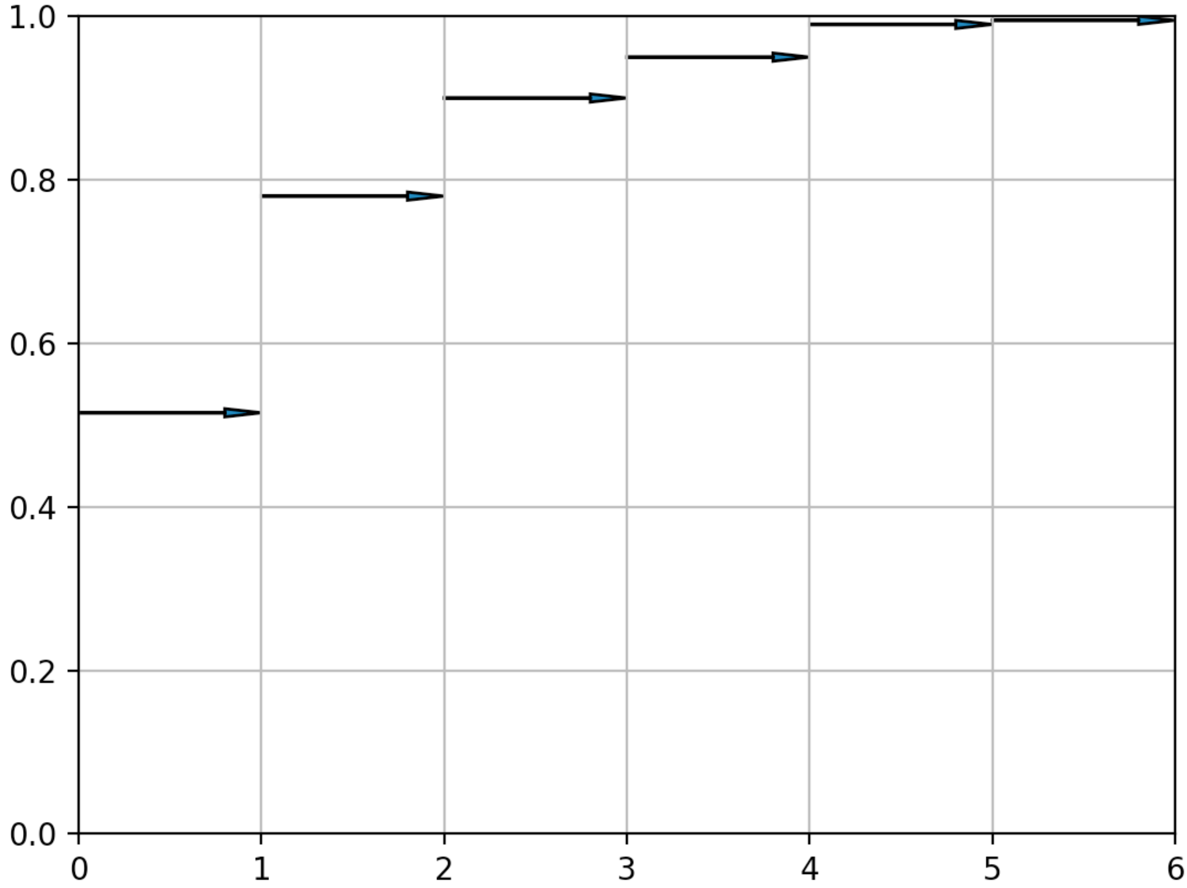
График полигона относительных частот обозначен синим цветом, а график полигона соответствующих теоретических вероятностей – красным.



Эмпирическая функция распределения:

$$F_{200}^{Э}(x) = \sum_{x_i^* \le x} w_i = \begin{cases} 0, x < 0 \\ 0.515, 0 \le x < 1 \\ 0.78, 1 \le x < 2 \\ 0.9, 2 \le x < 3 \\ 0.95, 3 \le x < 4 \\ 0.99, 4 \le x < 5 \\ 0.995, 5 \le x < 6 \\ 1, \quad x \ge 6 \end{cases}$$

График эмпирической функции



Задание 3) Распределение по закону Пуассона

λ = 1.04

Полученная выборка

1	1	0	0	0	2	1	2	0	1
1	1	0	0	2	0	3	0	0	1
2	1	1	1	1	3	0	1	3	0
1	3	1	1	2	0	0	1	0	1
1	0	0	0	0	0	3	0	1	0
0	1	1	0	1	1	1	1	1	0
3	2	2	2	1	1	2	0	0	0

5	1	1	0	1	1	0	0	1	1
1	0	1	2	2	1	0	0	0	2
1	1	0	1	1	0	0	1	0	1
0	1	2	0	4	1	1	0	1	2
0	0	0	1	0	2	0	3	1	2
2	1	1	2	3	0	2	1	2	1
0	1	2	1	0	2	1	2	1	2
2	1	0	2	0	0	0	0	1	1
1	2	2	3	3	1	1	1	1	0
0	0	1	2	1	0	1	5	0	1
2	1	0	2	0	2	1	2	1	1
0	0	0	2	0	1	3	0	1	1
0	1	0	1	0	0	2	1	3	6

Упорядоченная выборка

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	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	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	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1

1	1	1	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	2	2	2	2	2	2	2	2	2
2	2	2	2	3	3	3	3	3	3
3	3	3	3	3	3	4	5	5	6

Статистический ряд

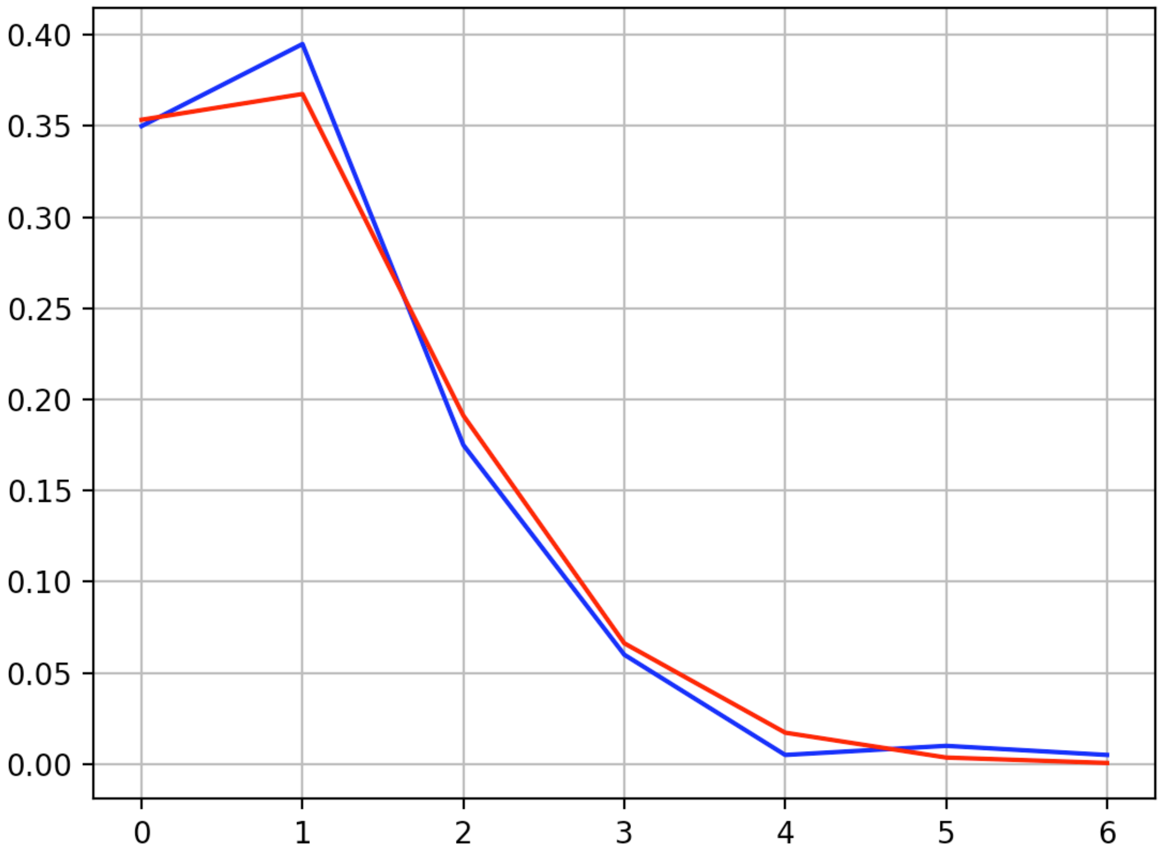
x_k^*	n_k	w_k	s_k
0	70	0.35	0.35
1	79	0.395	0.745
2	35	0.175	0.92
3	12	0.06	0.98
4	1	0.005	0.985
5	2	0.01	0.995
6	1	0.005	1

Результаты расчетов требуемых характеристик

Характеристика	Значение
выборочное среднее	1.025
выборочная дисперсия	1.09437
выборочное среднее квадратическое отклонение	1.04612
выборочная мода	1
выборочная медиана	1

выборочный коэффициент асимметрии	1.41757
выборочный коэффициент эксцесса	3.18592

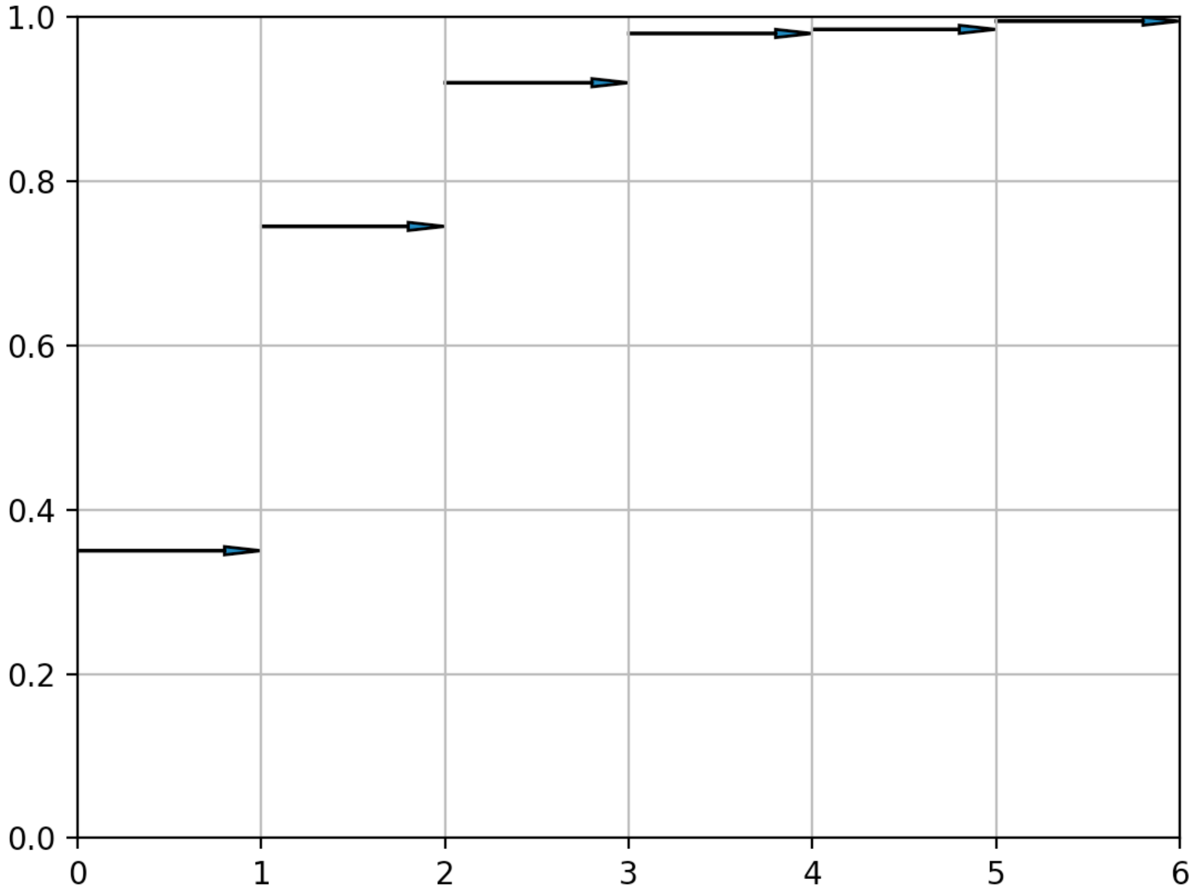
График полигона относительных частот обозначен синим цветом, а график полигона соответствующих теоретических вероятностей – красным.



Эмпирическая функция распределения:

$$F_{200}^{Э}(x) = \sum_{x_i^* \leq x} w_i = \begin{cases} 0, x < 0 \\ 0.35, 0 \leq x < 1 \\ 0.745, 1 \leq x < 2 \\ 0.92, 2 \leq x < 3 \\ 0.98, 3 \leq x < 4 \\ 0.985, 4 \leq x < 5 \\ 0.995, 5 \leq x < 6 \\ 1, \quad x \geq 6 \end{cases}$$

График эмпирической функции



Анализ результатов и выводы

1) Распределение по биномиальному закону

j	$\widetilde{w}_j$	p_j	$\lvert\widetilde{w}_j$ - $p_j\rvert$
0	0.005	0.00009	0.00491
1	0.005	0.00136	0.00364
2	0.005	0.00898	0.00398
3	0.015	0.03572	0.02072
4	0.085	0.09469	0.00969
5	0.15	0.17572	0.02572
6	0.18	0.23293	0.05293

7	0.27	0.22055	0.040445
8	0.19	0.14618	0.04382
9	0.065	0.0646	0.0004
10	0.02	0.01712	0.00288
11	0.01	0.00206	0.00794

$max\{|\widetilde{w}_j - p_j|\} = 0.05293$

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютн-ое отклонение	Относительное отклонение
Выборочное среднее	6.535	6.27	0.265	4.22648%
Выборочная дисперсия	2.908775	2.6961	0.212675	7.88825%
Выборочное среднее квадратичное отклонение	1.70551	1.64198	0.06353	3.86911%
Выборочная мода	7	6	1.00000	16.66667%
Выборочная медиана	7	6	1.00000	16.66667%
Выборочный коэффициент асимметрии	-0.4045	-0.08526	0.31924	374.43115%
Выборочный коэффициент эксцесса	0.87332	-0.17455	1.04787	600.32655%

2) Распределение по геометрическому закону

j	$\widetilde{w}_j$	p_j	$\lvert\widetilde{w}_j - p_j\rvert$
0	0.515	0.57	0.055
1	0.265	0.2451	0.0199
2	0.12	0.10539	0.01461
3	0.05	0.04532	0.00468
4	0.04	0.01948	0.02051
5	0.005	0.00838	0.00338
6	0.005	0.0036	0.0014

$max\{\lvert\widetilde{w}_j - p_j\rvert\} = 0.055$

Название показателя	Эксперимен-тальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	0.87	0.75439	0.11561	15.32496%
Выборочная дисперсия	1.3831	1.23484	0.05962	4.83059%
Выборочное среднее квадратичное отклонение	1.17605	1.15043	0.02562	2.2296%
Выборочная мода	0	0.00000	0.00000	-
Выборочная медиана	0	0	0.00000	-

Выборочный коэффициент асимметрии	1.58098	2.18073	0.59975	27.50226%
Выборочный коэффициент эксцесса	2.37213	6.75558	2.87229	42.5173%

3) Распределение по закону Пуассона

j	$\widetilde{w}_j$	p_j	$\lvert\widetilde{w}_j$ - $p_j\rvert$
0	0.35	0.35345	0.00345
1	0.395	0.36759	0.02741
2	0.175	0.19115	0.01615
3	0.06	0.06626	0.00626
4	0.005	0.01723	0.01223
5	0.01	0.00358	0.00642
6	0.005	0.00062	0.00438

$max\{|\widetilde{w}_j - p_j|\} = 0.02741$

Название показателя	Эксперимен-тальное значение	Теоретичес-кое значение	Абсолютн-ое отклонение	Относитель ное отклонение
Выборочное среднее	1.025	1.04	0.015	1.4423%
Выборочная дисперсия	1.09437	1.04	0.05437	5.22788%
Выборочное среднее	1.04612	1.0198	0.0202	1.98078%

квадратичное отклонение				
Выборочная мода	1	1	0.00000	0%
Выборочная медиана	1	1	0.00000	0%
Выборочный коэффициент асимметрии	1.41757	0.98058	0.43699	44.56437%
Выборочный коэффициент эксцесса	3.18592	0.96154	2.22438	231.33515%

Вывод: теоретические и экспериментальные в основном не сильно отличаются друг от друга, но были случаи, в которых достаточно большие относительное и абсолютные отклонения, но это из-за того, что взяли только 200 чисел.

Список использованной литературы

1. Математическая статистика [Электронный ресурс]: метод. указания по выполнению лаб. работ / А.А. Лобузов — М.: МИРЭА, 2017
2. Боровков А. А. Математическая статистика. — СПб.: Лань, 2010.-704 с.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. — М.: Юрайт, 2013 — 479 с.
4. Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике. — М.: Юрайт, 2013 — 404 с.
5. Емельянов Г.В.Скитович В.П. Задачник по теории вероятностей и математической статистике. — СПб.: Лань, 2007 — 336 с.
6. Ивченко Г. И., Медведев Ю. И. Введение в математическую статистику. — М.: Изд-во ЛКИ, 2010 — 599 с.
7. Кибзун А.И., Горяинова Е.Р., Наумов А.В. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачам. Учебное пособие — М.:ФИЗМАТЛИТ, 2005 — 232 с.

8. Кобзарь А.И. Прикладная математическая статистика: Для инженеров и научных работников — М.: ФИЗМАТЛИТ, 2006 — 816 с.
9. Монсик В.Б., Скрынников А. А. Вероятность и статистика.— М. : БИНОМ, 2015 — 384 с.
10. Сборник задач по теории вероятностей, математической статистике и теории случайных функций: Учеб. пособие для вузов / Под ред. А. А. Свешникова. — СПб.: Лань, 2012 — 472 с.
11. Письменный Д.Т. Конспект лекций по теории вероятностей, математической статистике и случайным процессам: учеб. пособие для вузов. — М.: Айрис-пресс, 2013 — 288 с.
12. Ramachandran Kandethody M., Tsokos Chris P. Mathematical Statistics with Applications in R. — N-Y.: Academic Press, 2009 — 826 p.

Приложение (Листинг программы)

```
from math import sqrt
from matplotlib import pyplot


def get_random_binomial(n, p, size=1):
    '''Генерация size псевдослучайных чисел, распределенных по
    биномиальному закону с параметрами n и p
    '''
    from numpy.random import binomial
    return binomial(n, p, size)



def get_random_geometric(p, size=1):
    '''Генерация size псевдослучайных чисел, распределенных по
    геометрическому закону с параметром p
    '''
    from numpy.random import geometric
    return [i-1 for i in geometric(p, size)]



def get_random_poisson(l, size=1):
```

```
    '''Генерация size псевдослучайных чисел, распределенных по
    закону Пуассона с параметром l
    '''
    from numpy.random import poisson
    return poisson(l, size)


def get_sr(R):
    '''Получение выборочного среднего из коллекции R
    '''
    ans = 0
    for xi, ni, wi in R:
        ans += xi * wi
    return ans


def get_disp(R, sr):
    '''Получение выборочной дисперсии из коллекции R с выборочным средним
sr
    '''
    ans = 0
    for xi, ni, wi in R:
        ans += (xi - sr)*(xi - sr)*wi
    return ans


def get_xi_ni_wi(xi, N):
    '''Получение генерация коллекции R - статистического ряда
    '''
    xi.sort()
```

```
    mx = xi[-1]
    ni = [0 for i in range(mx + 1)]
    wi = [0 for i in range(len(ni))]
    for i in range(mx + 1):
        for j in xi:
            ni[i] = ni[i] + 1 if i == j else ni[i]
        wi[i] = ni[i] / N
    return list(zip(*[(list(range(mx + 1))), (ni), (wi)]))


def get_func(R, x):
    '''Получение значения эмпирической функции распределения в точке x
    '''
    if (x < R[0][0]):
        return 0
    elif x >= R[-1][0]:
        return 1
    else:
        ans = 0
        i = 0
        while R[i][0] <= x:
            ans += R[i][2]
            i += 1
        return ans


def get_med(R):
    '''Получение выборочной медианы из коллекции R
    '''
    ans = 0
```

```
    for i in range(len(R) - 1):
        if get_func(R, R[i][0]) > 0.5:
            ans = R[i][0]
            break
        elif get_func(R, R[i][0]) == 0.5:
            ans = 0.5*(R[i][0] + R[i+1][0] + 1)
            break
    return ans


def get_moda(R):
    '''Получение выборочной моды из коллекции R
    '''
    moda = max([(j, i) for i, j, _ in R])[1]
    count = [i for i, _, _ in R].count(moda)
    if count > 1:
        moda *= 0.5 * count
    return moda


def get_moment(R, k):
    '''Получение выборочного момента порядка k из коллекции R
    '''
    ans = 0
    for xi, ni, wi in R:
        ans += (xi**k)*wi
    return ans


def get_k_asim(R, disp):
```

```
    '''Получение выборочного коэффициента асимметрии из коллекции R
    '''
    ans = get_moment(R, 3) - 3*get_moment(R, 2)*get_moment(R, 1)
    return (ans + 2*(get_moment(R, 1)**3)) / (disp**3)


def get_k_eks(R, disp):
    '''Получение выборочного коэффициента эксцесса из коллекции R
    '''
    ans = get_moment(R, 4) - 4*get_moment(R, 3)*get_moment(R, 1)
    ans += 6* get_moment(R, 2)*(get_moment(R, 1)**2)
    ans -= 3*(get_moment(R, 1)**4)
    return (ans / (disp**4)) - 3


def get_teoretic_binom(n, k, p):
    '''Получение теоретического значения биномиального распределения
    '''
    from math import factorial
    return (factorial(n) * (p**k) * (1 - p)**(n-k)) / (factorial(k)* factorial(n - k))


def get_teoretic_geometric(k, p):
    '''Получение теоретического значения геометрического распределения
    '''
    return (1 - p)**k * p


def get_teoretic_poisson(l, k):
    '''Получение теоретического значения распределения Пуассона
```

```
    '''
    from math import exp, factorial
    return (l**k * exp(-l)) / factorial(k)


def get_info(R, N):
    sr = get_sr(R)
    disp = get_disp(R, sr)
    print("Выборочное среднее: ", sr)
    print("Выборочная дисперсия: ", disp)
    print("Выборочное среднее квадратическое отклонение: ", sqrt(disp))
    print("Выборочная мода:", get_moda(R))
    print("Выборочная медиана:", get_med(R))
    print("Выборочный коэффициент асимметрии: ", get_k_asim(R, sqrt(disp)))
    print("Выборочный коэффициент эксцесса: ", get_k_eks(R, sqrt(disp)))


def draw_arrows(plt, R):
        '''Отрисовка стрелок
        '''
        plt.figure()
        plt.grid(True)
        plt.axis([0, max([i for i, _, _ in R]), 0, 1.0])
        for i in range(len(R)-1):
            x1 = R[i][0]+0.0001
            x2 = R[i+1][0] - 0.0001
            y1 = get_func(R, x1)
            y2 = get_func(R, x2)
            plt.arrow(x1, y1, x2-x1, 0, head_width=0.01, head_length=0.2,
length_includes_head=True)
```

```
def draw(R, teoretic_data):
    '''Отрисовка графиков
    '''

    pyplot.figure()
    pyplot.grid(True)
    pyplot.plot([i for i, _, _ in R], [k for _, _, k in R], color='blue')
    pyplot.plot([i for i, _, _ in R], teoretic_data, color='red')
    draw_arrows(pyplot, R)

    pyplot.figure()
    pyplot.grid(True)
    pyplot.plot([i for i, _, _ in R2], [k for _, _, k in R2], color='blue')
    pyplot.plot([i for i, _, _ in R2], [get_teoretic_geometric(i, p) for i, _, _ in R2],
color='red')
    draw_arrows(pyplot, R2)

    pyplot.figure()
    pyplot.grid(True)
    pyplot.plot([i for i, _, _ in R3], [k for _, _, k in R3], color='blue')
    pyplot.plot([i for i, _, _ in R3], [get_teoretic_poisson(l, i) for i, _, _ in R3],
color='red')
    draw_arrows(pyplot, R3)

    pyplot.show()


N = 200
```

```
v = 54
n = 5 + (v % 16)
p = 0.3 + 0.005 * v
print("n={} ; p={}".format(n , p))

print("Binom")
#binom = get_random_binomial(n, p, N)
binom = [7, 7, 6, 7, 5, 7, 6, 7, 9, 9, 5, 7, 5, 5, 7, 5, 7, 8, 4,
    7, 0, 3, 7, 5, 6, 9, 7, 7, 7, 2, 8, 4, 4, 5, 4, 8, 6, 7,
    6, 7, 6, 8, 8, 8, 7, 5, 6, 4, 6, 4, 8, 5, 4, 9, 5, 8, 9,
    4, 5, 7, 6, 7, 7, 6, 5, 7, 6, 6, 6, 6, 5, 6, 7, 7, 8, 6,
    8, 6, 6, 6, 7, 7, 4, 7, 7, 5, 3, 5, 7, 8, 6, 10, 5, 8, 7,
    7, 8, 9, 10, 8, 4, 8, 4, 7, 7, 5, 8, 10, 6, 5, 9, 6, 5, 6,
    7, 6, 7, 10, 7, 4, 6, 9, 7, 5, 7, 8, 7, 8, 4, 8, 6, 8, 7,
    8, 8, 7, 8, 9, 8, 5, 6, 7, 8, 5, 8, 7, 7, 6, 9, 8, 6, 11,
    5, 8, 5, 6, 8, 4, 7, 5, 8, 9, 6, 9, 5, 7, 5, 4, 5, 8, 7,
    8, 8, 8, 7, 6, 8, 1, 7, 7, 5, 7, 7, 8, 7, 6, 8, 4, 11,
    8, 7, 5, 7, 3, 9, 6, 6, 7, 4, 6]
print(sorted(binom))
R1 = get_xi_ni_wi(binom, N)
print(R1)
get_info(R1, N)

print("Geom")
#geom = get_random_geometric(p, N)
geom = [0, 0, 2, 0, 0, 0, 0, 1, 0, 1, 0, 0, 3, 3, 0, 2, 0, 0, 1,
    0, 0, 1, 0, 0, 1, 0, 1, 0, 2, 0, 0, 0, 0, 1, 0, 2, 1, 0,
    4, 1, 0, 0, 1, 3, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 2,
    0, 3, 2, 0, 1, 6, 4, 4, 1, 0, 0, 0, 3, 1, 1, 1, 2, 0, 0,
    1, 2, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 4, 0, 0, 3, 1, 1, 1,
```

```
        1, 0, 2, 1, 0, 4, 0, 2, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,
        2, 0, 1, 4, 4, 2, 0, 3, 1, 0, 0, 1, 1, 0, 2, 0, 0, 0, 0,
        0, 0, 0, 5, 1, 0, 4, 2, 1, 1, 1, 0, 2, 2, 2, 0, 1, 0, 0,
        0, 0, 2, 1, 0, 0, 2, 3, 1, 0, 1, 2, 1, 0, 0, 0, 1, 0, 1,
        1, 0, 0, 0, 1, 2, 1, 1, 1, 0, 0, 0, 3, 0, 3, 0, 0, 0, 1,
        0, 0, 2, 1, 1, 0, 2, 2, 1, 1]
print(geom)
R2 = get_xi_ni_wi(geom, N)
print(R2)
get_info(R2, N)

print("Poisson")
l = 0.5+0.01*v
print("l = ", l)
#poisson = get_random_poisson(l, N)
poisson = [1, 1, 0, 0, 0, 2, 1, 2, 0, 1, 1, 1, 0, 0, 2, 0, 3, 0,
        0, 1, 2, 1, 1, 1, 1, 3, 0, 1, 3, 0, 1, 3, 1, 1, 2, 0, 0,
        1, 0, 1, 1, 0, 0, 0, 0, 0, 3, 0, 1, 0, 0, 1, 1, 0, 1, 1,
        1, 1, 1, 0, 3, 2, 2, 2, 1, 1, 2, 0, 0, 0, 5, 1, 1, 0, 1,
        1, 0, 0, 1, 1, 1, 0, 1, 2, 2, 1, 0, 0, 0, 2, 1, 1, 0, 1,
        1, 0, 0, 1, 0, 1, 0, 1, 2, 0, 4, 1, 1, 0, 1, 2, 0, 0, 0,
        1, 0, 2, 0, 3, 1, 2, 2, 1, 1, 2, 3, 0, 2, 1, 2, 1, 0, 1,
        2, 1, 0, 2, 1, 2, 1, 2, 2, 1, 0, 2, 0, 0, 0, 0, 1, 1, 1,
        2, 2, 3, 3, 1, 1, 1, 1, 0, 0, 0, 1, 2, 1, 0, 1, 5, 0, 1,
        2, 1, 0, 2, 0, 2, 1, 2, 1, 1, 0, 0, 0, 2, 0, 1, 3, 0, 1,
        1, 0, 1, 0, 1, 0, 0, 2, 1, 3, 6]
print(sorted(poisson))
R3 = get_xi_ni_wi(poisson, N)
print(R3)
get_info(R3, N)
```

```
draw(R1, [get_teoretic_binom(n, i, p) for i, _, _ in R1])
draw(R2, [get_teoretic_geometric(i, p) for i, _, _ in R2])
draw(R3, [get_teoretic_poisson(l, i) for i, _, _ in R3])
```