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

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
Жолковский Д. А.

Группа: КМБО-01-16

МОСКВА 2019

Лабораторная работа по Математической статистике № 1 «Первичная обработка выборки из дискретной генеральной совокупности»

Задание 1. Получить выборку, сгенерировав 150 псевдослучайных чисел, распределенных по биномиальному закону с параметрами n и p. $n=5 + V \bmod 17$ $p=0,1 + 0,01V$

Задание 2. Получить выборку, сгенерировав 150 псевдослучайных чисел, распределенных по геометрическому закону с параметром p.

$p =0,1 + 0,01V$

Задание 3. Получить выборку, сгенерировав 150 псевдослучайных чисел, распределенных по закону Пуассона с параметром λ . $\lambda = 0,7 + 0,07V$

Для всех выборок построить:

1) статистический ряд;

2) полигон относительных частот;

3) эмпирическую функцию 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.

V – номер варианта. Вычисления проводить с точностью до 0,00001 .

Теоретические сведения

Полученную выборку $\{x1 , x2 , x3 , ..., xN\}$ упорядочить по возрастанию, определить частоты n_i и относительные частоты (частости) w_i, построить статистический ряд:

x_i	x_1^*	…	x_m^*
n_i	n_1	…	n_m
w_i	w_1	…	w_m

Полигон относительных частот – ломаная линия, соединяющая последовательно точки с координатами $(x_1^*, w_1), (x_2^*, w_2), \ldots, (x_m^*, w_m)$.

Эмпирическая функция распределения

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

Выборочное среднее

$$\bar{x} = \sum_{i=1}^{m} x_i^* w_i$$

Выборочная дисперсия

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

Выборочный момент k-ого порядка

$$\overline{\mu_k} = \sum_{i=1}^{m} (x_i^*)^k w_i$$

Выборочный центральный момент k-ого порядка

$$\overline{\mu_k^o} = \sum_{i=1}^{m} (x_i^* - \bar{x})^k w_i$$

Выборочное среднее кврадратическое отклонение

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

Выборочная медиана

$$\overline{M_e} = \begin{cases} x_i^*, & F_N^э(x_{i-1}^*) < 0{,}5 < F_N^э(x_i^*) \\ \dfrac{1}{2}(x_i^* + x_{i+1}^*), & F_N^э(x_i^*) = 0{,}5 \end{cases}$$

Выборочная мода – это значение x_i, которому соответствует максимальная частота.

$$\begin{cases} \text{если } n_i = \max n_k > n_j, \text{i} \neq \text{j}, & \overline{M_0} = \{x_i^* \mid n_i = \max n_k\} \\ \text{если } n_i = n_i + 1 = \ldots = n_{i+j} = \max n_k, & \text{то } \overline{M_0} = \dfrac{1}{2}(x_i^* + x_{i+1}^*) \\ \text{если } n_i = n_j = \max n_k > n_l, i < \text{l} < j, & \text{то } \overline{M_0} - \text{не существует} \end{cases}$$

Выборочный коэффициент асимметрии

$$\overline{\alpha_s} = \frac{\overline{\mu_3^o}}{\bar{\sigma}^3}$$

Выборочный коэффициент эксцесса

$$\overline{\varepsilon_k} = \frac{\overline{\mu_4^o}}{\bar{\sigma}^4} - 3$$

Ряд распределения - структурная группировка с целью выделения характерных свойств и закономерностей изучаемой совокупности.

Математическое ожидание – понятие среднего значения случайной величины в теории вероятностей.

Дисперсия – отклонение величины от ее математического ожидания.

Среднеквадратическое отклонение – показатель рассеивания значений случайной величины относительно ее математического ожидания.

Мода – значение во множестве наблюдений, которое встречается наиболее часто.

Медиана – возможное значение признака, которое делит вариационный ряд выборки на две равные части.

Коэффициент асимметрии используется для проверки распределения на симметричность, а также для грубой предварительной проверки на нормальность.

Если плотность распределения симметрична, то выборочный коэффициент асимметрии равен нулю, если левый хост распределения тяжелее – больше нуля, легче – меньше.

Коэффициент эксцесса используется для проверки на нормальность.

Нормальное распределение имеет нулевой эксцесс. Если хвосты распределения «легче», а пик острее, чем у нормального распределения, то коэффициент эксцесса положительный; если хвосты распределения «тяжелее», пик «приплюснутый», чем у нормального распределения, то отрицательный.

Биномиальное распределение

Биномиальное распределение – распределение количества «успехов» в последовательности из n независимых случайных экспериментов, таких что вероятность «успеха» в каждом из них равна p.

Математическое ожидание: np

Дисперсия: npq, $q=1-p$

Среднеквадратическое отклонение: $\sqrt{npq}$

Мода: [(n+1)p], если (n+1)p – дробное; (n+1)p- $\frac{1}{2}$, если np – целое;

Медиана: $Round(np)$

Коэффициент асимметрии: $\frac{q-p}{\sqrt{npq}}$

Коэффициент эксцесса: $\frac{1-6pq}{npq}$

Геометрическое распределение

Геометрическое распределение – распределение величины, равной количеству испытаний случайного эксперимента до наблюдения первого «успеха».

Математическое ожидание: $\frac{q}{p}$, *q=1-p*

Дисперсия: $\frac{q}{p^2}$, *q=1-p*

Среднее квадратичное отклонение: $\sqrt{\frac{q}{p^2}}$

Мода: 0

Медиана: $\left[-\frac{ln2}{lnq}\right]$, если $\frac{ln2}{lnq}$ – дробное; $-\frac{ln2}{lnq}-\frac{1}{2}$, если $\frac{ln2}{lnq}$ - целое

Коэффициент асимметрии: $\frac{2-p}{\sqrt{1-p}}$

Коэффициент эксцесса: $6+\frac{p^2}{1-p}$

Распределение Пуассона

Распределение Пуассона – вероятностное распределение дискретного типа, моделирует случайную величину, представляющую собой число событий, произошедших за фиксированное время, при условии, что данные события происходят с некоторой фиксированной средне интенсивностью и независимо друг от друга.

Математическое ожидание: λ

Дисперсия: λ

Среднеквадратическое отклонение: $\sqrt{\lambda}$

Мода: $\lfloor\lambda\rfloor$

Медиана: $\left\lfloor \lambda + \frac{1}{3} - \frac{0.002}{\lambda} \right\rfloor$

Коэффициент асимметрии: $\lambda^{-\frac{1}{2}}$

Коэффициент эксцесса: λ^{-1}

Средства языка Octave

В программе расчёта используются следующие средства языка:

Функции:

- binornd(n, p, s, z) - возвращает матрицу случайных значений из биномиального распределения с параметрами n и p , где n есть число испытаний и p есть вероятность успеха, s – количество строк в возвращаемой матрице, z – количество столбцов.
- geornd(p, s, z) - возвращает матрицу случайных значений из геометрического распределения с параметром p, s – количество строк в возвращаемой матрице, z – количество столбцов.
- poissrnd(λ, s, z) - возвращает матрицу случайных значений из распределения Пуассона с параметром λ, s – количество строк в возвращаемой матрице, z – количество столбцов.
- sort(x) - возвращает копию x с элементами, расположенными в порядке возрастания.
- sqrt(x) – возвращает квадратный корень из числа x.
- max(X) в случае одномерного массива возвращает наибольший элемент; в случае двумерного массива - это вектор-строка, содержащая максимальные элементы каждого столбца.

Операторы управления:

- for – endfor
- if – else – endif
- break

А также арифметические и логические операторы.

Результаты расчетов с комментариями

Задание 1:

n = 14, p = 0,19

Выборка:

Неупорядоченная:

1	1	4	3	3	1	4	2	2	1	1	3	3	1	3
2	3	2	0	5	2	0	2	3	3	1	2	1	3	5
9	4	1	5	3	0	1	1	2	2	2	3	2	2	1
1	4	4	2	2	1	0	2	4	5	2	2	1	2	4
4	3	1	1	1	4	5	0	1	2	2	3	4	2	5
2	4	5	0	3	3	1	1	2	1	4	5	3	3	2
3	2	1	4	4	3	4	2	2	1	4	0	2	5	1
1	6	1	4	0	2	3	2	2	2	3	4	3	2	4
3	3	1	1	4	2	2	3	3	3	3	4	3	7	3
2	0	3	4	5	3	4	2	2	1	3	2	5	3	1

Упорядоченная:

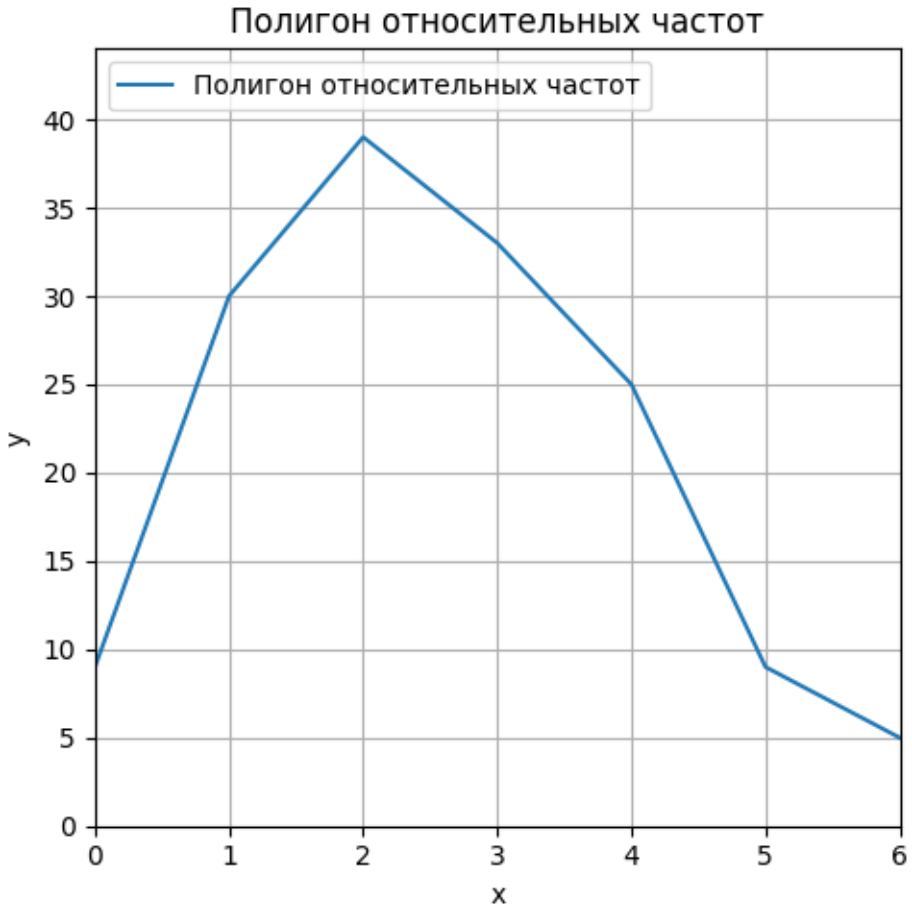
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	5	5	5	5	5	5	5	5	5	5	5	6	7	9

Статистический ряд:

x_i	0	1	2	3	4	5	6	7	8	9
n_i	9	31	39	34	23	11	1	1	0	1

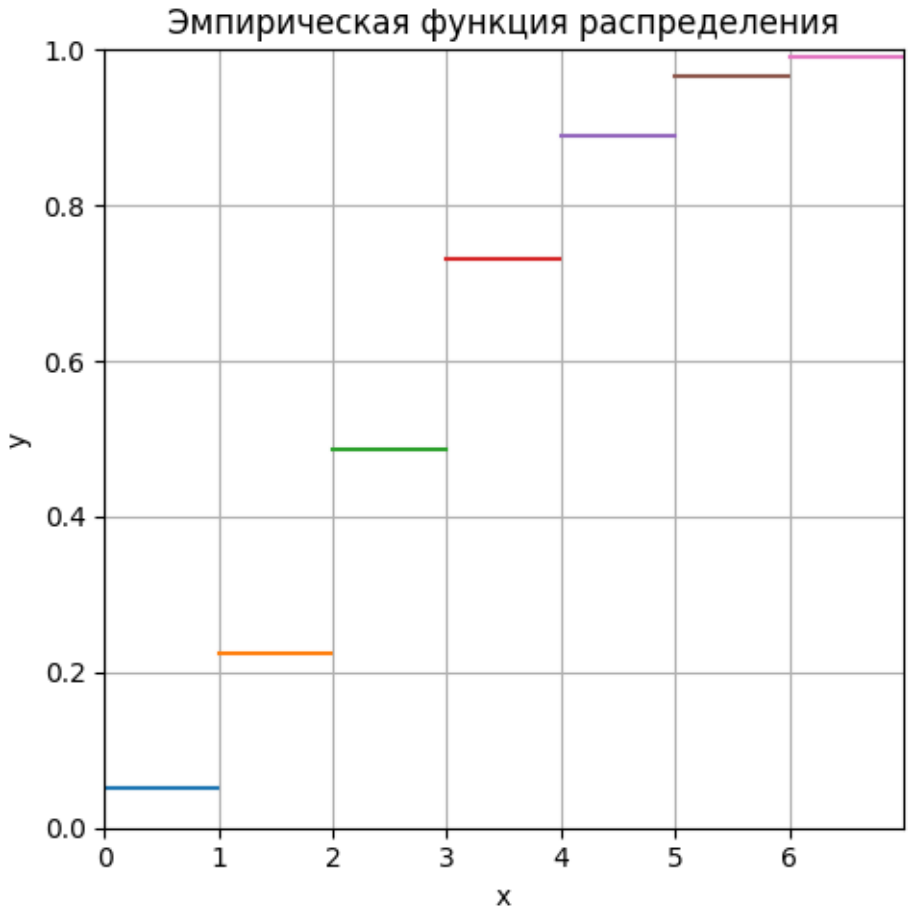
w_i	0.06	0.2066	0.26	0.2266	0.1533	0.0733	0.0066	0.0066	0	0.0066

Полигон относительных частот:



Эмпирическая функция распределения:

$$F^{э}_{150}(x) = \sum_{x_i \le x} w_i = \begin{cases} 0, & x < 0 \\ 0.0523,, & 0 \le x < 1 \\ 0.2241, & 1 \le x < 2 \\ 0.4862, & 2 \le x < 3 \\ 0.7321, & 3 \le x < 4 \\ 0.8907, & 4 \le x < 5 \\ 0.9651, & 5 \le x < 6 \\ 0.9912, & 6 \le x < 7 \\ 0.9983, & 7 \le x < 8 \\ 0.9997, & 8 \le x < 9 \\ 1, & x \ge 9 \end{cases}$$



Выборочное среднее: 2.5466

Выборочная дисперсия: 2.1011

Выборочное среднее квадратическое отклонение: 1.4495

Выборочная мода: 2

Выборочная медиана: 2

Выборочный коэффициент асимметрии: 0.3340

Выборочный коэффициент эксцесса: -0.4272

Задание 2

p = 0.19

Выборка:

Неупорядоченная:

4	3	3	4	5	2	6	4	2	1	6	1	3	19	7
8	1	4	4	2	2	4	8	7	8	1	1	13	5	4
2	1	4	9	4	2	4	7	2	7	8	3	1	6	12
6	1	5	2	5	8	8	4	9	1	2	2	4	7	3
2	1	1	4	6	3	3	6	7	5	5	1	2	11	2
5	12	6	8	2	6	5	3	2	4	2	4	5	5	5
13	26	8	1	2	3	12	2	4	3	1	3	6	2	5
7	4	2	3	3	4	1	6	8	8	5	13	5	6	7
2	1	4	1	9	15	12	12	10	1	5	3	5	3	9
5	4	2	7	3	1	6	3	1	2	1	1	1	16	2

Упорядоченная:

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	3	3	3
3	3	3	3	4	4	4	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	6	6	6	6
6	6	6	6	6	6	6	6	7	7	7	7	7	7	7
7	7	8	8	8	8	8	8	8	8	8	8	9	9	9
9	10	11	12	12	12	12	12	13	13	13	15	16	19	26

Статистический ряд:

x_i	0	1	2	3	4	5
n_i	0	23	24	17	20	17
w_i	0.0	0.1533	0.16	0.1133	0.1333	0.1133

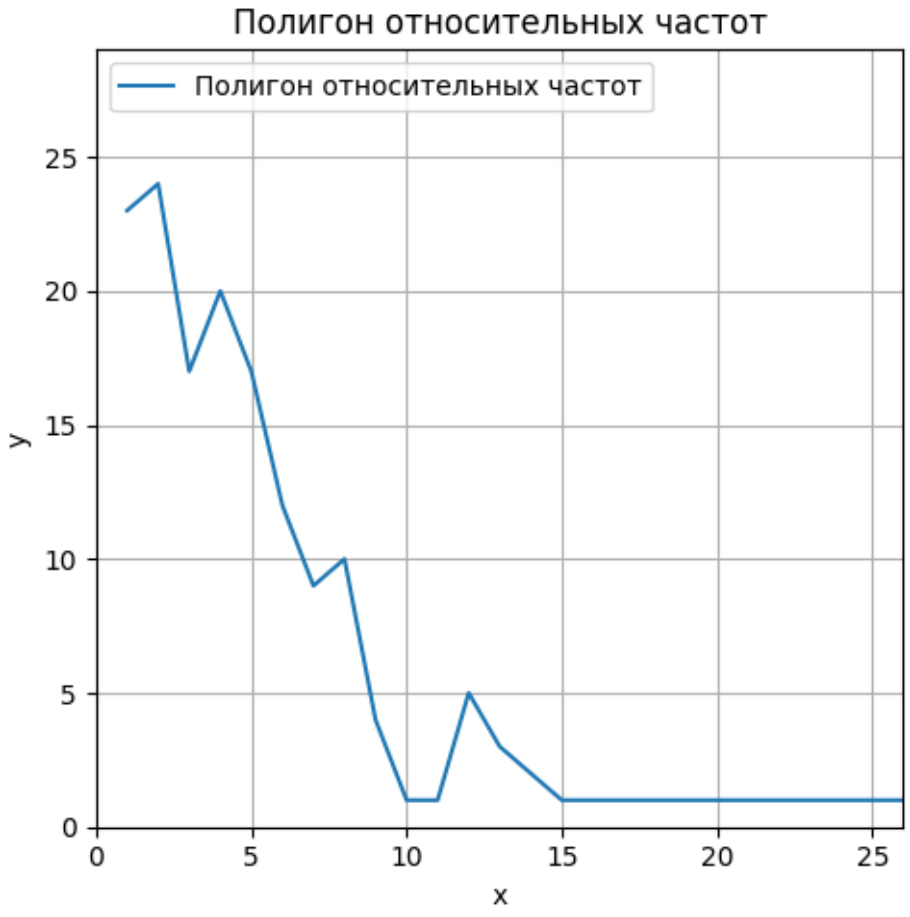
x_i	6	7	8	9	10	11
n_i	12	9	10	4	1	1
w_i	0.08	0.06	0.06	0.02	0.0066	0.0066

x_i	12	13	14	15	16	17
n_i	5	3	0	1	1	0
w_i	0.0333	0.02	0.0	0.0066	0.0066	0.0

x_i	18	19	20	21	22	23
n_i	0	1	0	0	0	0
w_i	0.0	0.0066	0.0	0.0	0.0	0.0

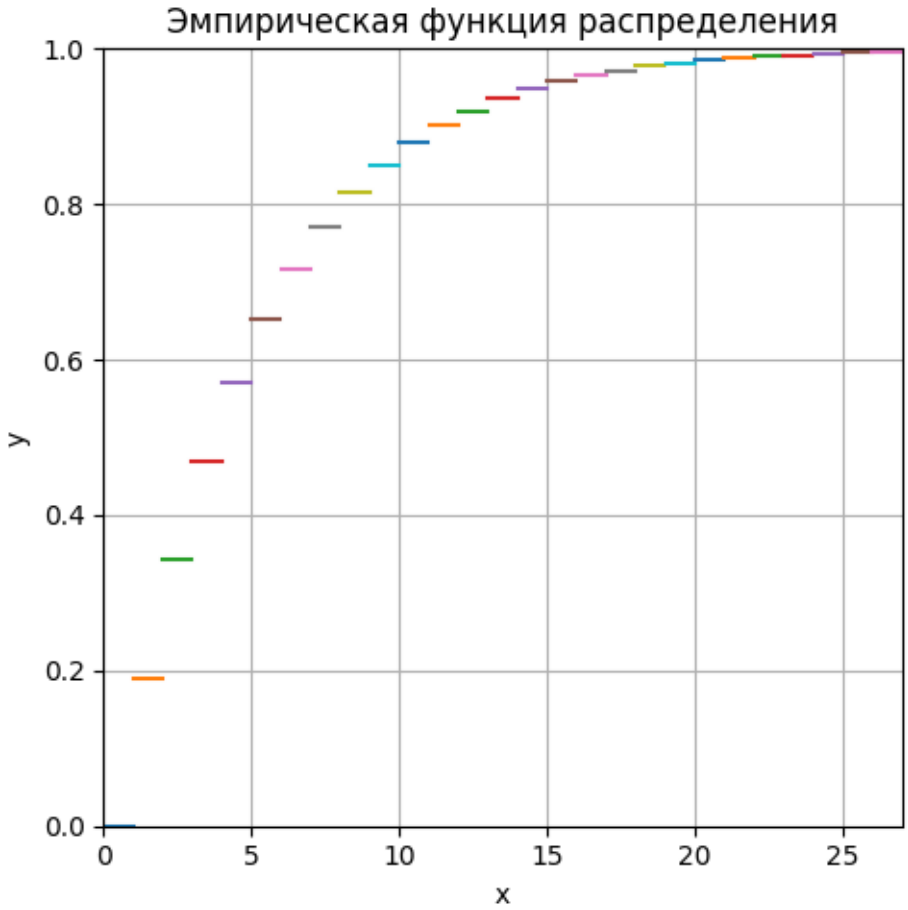
x_i	24	25	26
n_i	0	0	1
w_i	0.0	0.0	0.0066

Полигон относительных частот:



Эмпирическая функция распределения:

$$F_{150}^{Э}(x) = \sum_{x_i \le x} w_i = \begin{cases} 0, & x < 0 \\ 0, & 0 \le x < 1 \\ 0.19, & 1 \le x < 2 \\ 0.3439, & 2 \le x < 3 \\ 0.4685, & 3 \le x < 4 \\ 0.5695, & 4 \le x < 5 \\ 0.6513, & 5 \le x < 6 \\ 0.7175, & 6 \le x < 7 \\ 0.7712, & 7 \le x < 8 \\ 0.8146, & 8 \le x < 9 \\ 0.8499, & 9 \le x < 10 \\ 0.8784, & 11 \le x < 12 \\ 0.9015, & 12 \le x < 13 \\ 0.9202, & 13 \le x < 14 \\ 0.9353, & 14 \le x < 15 \\ 0.9476, & 15 \le x < 16 \\ 0.9576, & 17 \le x < 18 \\ 0.9656, & 18 \le x < 19 \\ 0.9721, & 19 \le x < 20 \\ 0.9774, & 20 \le x < 21 \\ 0.9817, & 21 \le x < 22 \\ 0.9852, & 22 \le x < 23 \\ 0.9880, & 23 \le x < 24 \\ 0.9903, & 24 \le x < 25 \\ 0.9921, & 25 \le x < 26 \\ 1, & x \ge 26 \end{cases}$$



Выборочное среднее: 4.8933

Выборочная дисперсия: 14.8552

Выборочное среднее квадратическое отклонение: 3.8542

Выборочная мода: 2

Выборочная медиана: 4

Выборочный коэффициент асимметрии: 1.9765

Выборочный коэффициент эксцесса: 6.0918

Задание 3

$\lambda = 1.33$

Выборка:

Неупорядоченная:

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

Упорядоченная:

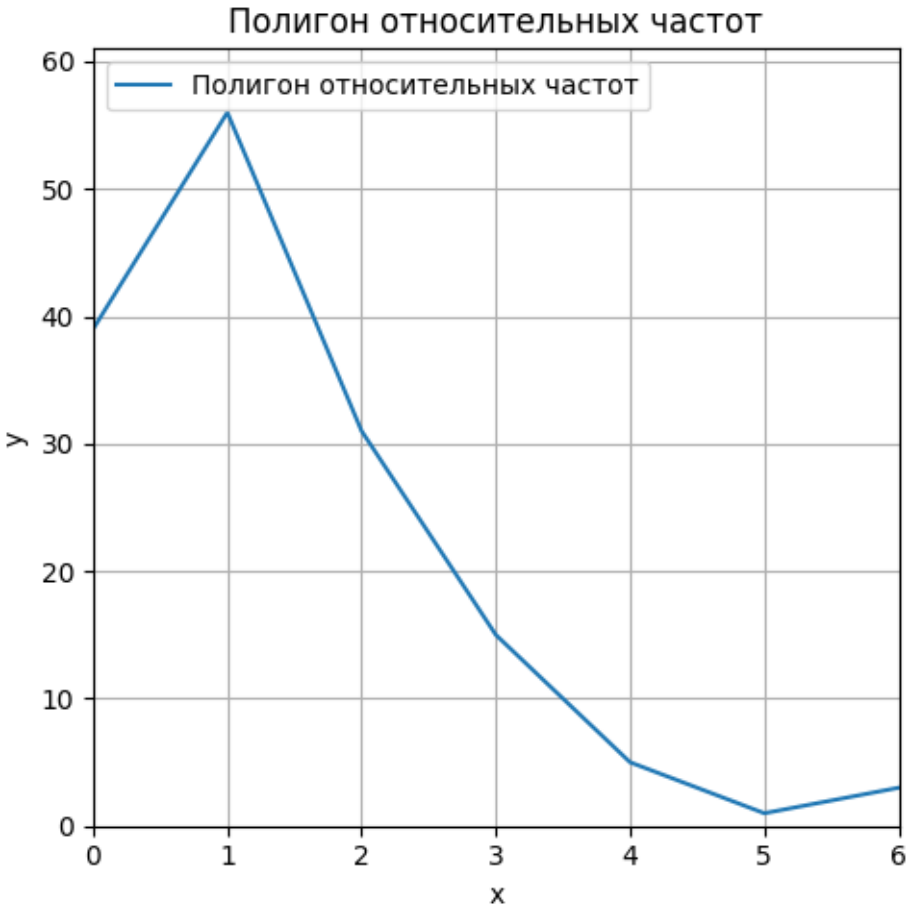
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	4	4	4	4	4	5	6	6	6

Статистический ряд:

x_i	0	1	2	3	4
n_i	39	56	31	15	5
w_i	0.426	0.3733	0.2066	0.01	0.0333

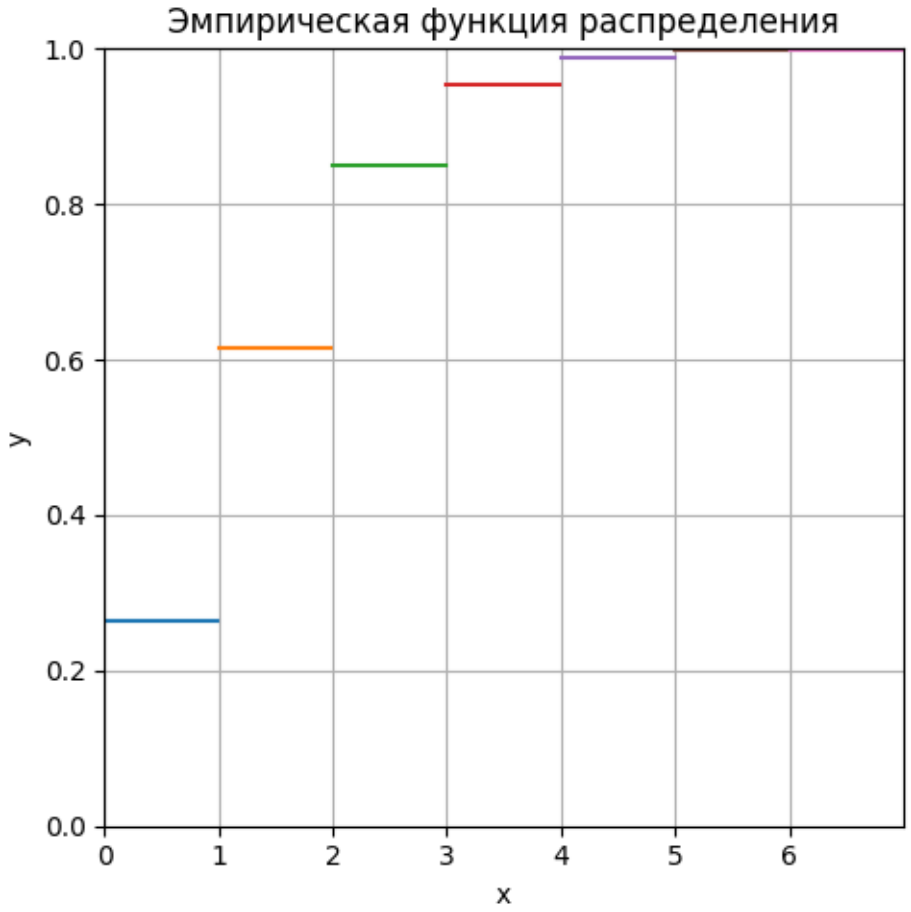
x_i	5	1
n_i	1	3
w_i	0.0067	0.02

Полигон относительных частот:



Эмпирическая функция распределения:

$$F_{150}^{\text{э}}(x) = \sum_{x_i \le x} w_i = \begin{cases} 0, & x < 0 \\ 0.2644, & 0 \le x < 1 \\ 0.6162, & 1 \le x < 2 \\ 0.8501, & 2 \le x < 3 \\ 0.9538, & 3 \le x < 4 \\ 0.9883, & 4 \le x < 5 \\ 0.9975, & 5 \le x < 6 \\ 1, & x \ge 6 \end{cases}$$



Выборочное среднее: 1.3733

Выборочная дисперсия: 1.6339

Выборочное среднее квадратическое отклонение: 1.2782

Выборочная мода: 1

Выборочная медиана: 1

Выборочный коэффициент асимметрии: 1.2885

Выборочный коэффициент эксцесса: 2.0822

Выводы

Задание 1

n = 14, q = 0,81, p = 0,19

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	2.5466	2.66	0.1134	4.4529%
Выборочная дисперсия	2.1011	2.1546	0.0535	2.5462%
Выборочное среднее квадратичное отклонение	1.4495	1.4678	0.0183	1.2625%
Выборочная мода	2	2	0	0%
Выборочная медиана	2	3	1	50%
Выборочный коэффициент асимметрии	0.334	0.4223	0.3889	26.4371%
Выборочный коэффициент эксцесса	-0.4272	0.0355	0.4627	203.3802%

$\{|w_i-p_i|\}$=[0.0076, 0.0281, 0.0020407421518351954, 0.0258, 0.008, 0.0144, 0.0071]

$\max\{|w_i-p_i|, i=1,\ldots,m\}$=0.0281

Задание 2

q = 0,81, p = 0,19

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	4.8933	5.2631	0.3698	7.5572%
Выборочная дисперсия	14.8552	22.4376	7.5824	51.042%
Выборочное среднее квадратичное отклонение	3.8542	4.7368	0.8826	22.8996%
Выборочная мода	2	2	0	0%
Выборочная медиана	4	4	0	0%
Выборочный коэффициент асимметрии	1.9765	2.0111	0.0346	1.7505%
Выборочный коэффициент эксцесса	6.0918	6.0445	0.0473	0.7825%

$\{|w_i-p_i|\}$=[0.0366, 0.006, 0.0113, 0.032, 0.0315, 0.0137, 0.00639, 0.0232, 0.0085, 0.0218, 0.0164, 0.0146, 0.0048, 0.0032, 0.0013, 0.002, 0.0056]

$\max\{|w_i-p_i|, i=1,\ldots,m\}$=0.0366

Задание 3

$\lambda = 1.33$

Название показателя	Экспериментальное значение	Теоретическое значение	Абсолютное отклонение	Относительное отклонение
Выборочное среднее	1.3733	1.33	0.0433	3.2556%
Выборочная дисперсия	1.6339	1.33	0.3039	22.8496%
Выборочное среднее квадратичное отклонение	1.2782	1.1532	0.125	10.8394%
Выборочная мода	1	1	0	0%
Выборочная медиана	1	1	0	0%
Выборочный коэффициент асимметрии	1.2885	0.8671	0.4214	48.5987%
Выборочный коэффициент эксцесса	2.0822	0.7518	1.3304	76.9619%

$\{|w_i-p_i|\}=[0.0044, 0.0215, 0.0272, 0.0037, 0.0011, 0.0025, 0.0179]$

$\max\{|w_i-p_i|, i=1,\ldots,m\}=0.0179$

Вывод

В ходе лабораторной работы выяснилось, что полученные эксперементальным путем данные соответствуют заданным распределениям, если принимать в расчет отклонения от теоретического значения.

Экспериментальная оценка выборочных показателей может сильно отличаться от теоретического значения, в силу того, что выборки из 150 элементов недостаточно для проведения точных расчетов.

Для выборки из 150 элементов безошибочно получается определить моду.

С увеличением выборки точность будет улучшаться.

Литература по математической статистике

1. Гмурман В.Е. Теория вероятностей и математическая статистика: Учеб. пособие
для вузов — М.: Высш. образов., 2006. — 480 с.
2. Математическая статистика [Электронный ресурс]: метод. указания по выполнению лаб. работ / А.А.Лобузов — М.: МИРЭА, 2017. — Электрон. опт. диск (ISO)
3. Справочное пособие по теории вероятностей и математической статистике
(законы распределения): Учеб.пособие для вузов / Г.А.Соколов, Н.А. Чистякова. —
М.: Высш. шк., 2007. — 248 с.

Приложение

```
#!/usr/bin/python
# -*- coding: UTF-8 -*-

import sys
import argparse
from collections import Counter
from itertools import islice

from functools import reduce
from scipy import stats as st
import numpy as np
from glob import glob
import pickle
import re

import matplotlib.pyplot as plt

variant = 9

def createParser():
    parser = argparse.ArgumentParser()
    parser.add_argument('-w', '--write', default=0,
type=int)
    parser.add_argument('-r', '--read', default=1,
type=int)
    return parser

def save(r, type_ryad):
    reg = 'data/' + type_ryad + '*'
    files = glob(reg) # type: ryadN.pickle
    if files:
        n = int(re.sub(r'[^\d]', '', files[0]))
        n += 1
    else:
        n = 0
```

```
36      filename = reg[:-1] + str(n) + '.pickle'
37      with open(filename, 'wb') as f:
38          pickle.dump(r, f)
39
40  def load(type_ryad, n = -1):
41      reg = 'data/' + type_ryad + '*'
42      files = glob(reg)
43      if not files:
44          return
45      if abs(n) > len(files):
46          return
47      with open(files[n], 'rb') as f:
48          return(pickle.load(f))
49
50  def draw_poligon(count, name):
51      X = list(count.keys())
52      Y = [count[x] for x in X]
53      fig, ax = plt.subplots(figsize=(5, 5))
54      ax.plot(X, Y, label='Полигон относительных
55  частот')
56      ax.set_title('Полигон относительных частот')
57      ax.legend(loc='upper left')
58      ax.set_ylabel('y')
59      ax.set_xlabel('x')
60      ax.set_xlim(xmin=0, xmax=max(X))
61      ax.set_ylim(ymin=0, ymax=max(Y) + 5)
62      ax.grid()
63      fig.tight_layout()
64      fig.savefig('data/poligon_' + name + '.png')
65
66  def draw_cdf(Xlist, Ylist, name):
67      fig, ax = plt.subplots(figsize=(5, 5))
68
69      for X, Y in zip(Xlist, Ylist):
70          ax.plot(X, Y, label='')
71      ax.set_title('Эмпирическая функция
```

```
72 распределения')
73     ax.legend(loc='upper left')
74     ax.set_ylabel('y')
75     ax.set_xlabel('x')
76     ax.set_xlim(xmin=0, xmax=max(X))
77     ax.set_ylim(ymin=0, ymax=1)
78     fig.tight_layout()
79     ax.grid()
80     fig.savefig('data/cdf_' + name + '.png')
81
82 def exp_stats(nv, r, st_r):
83     mean = sum([key * val[1] for key, val in
84 nv.items()]) # mean
85     var = sum([(key - mean) ** 2 * val[1] for key,
86 val in nv.items()])
87     standart = var ** 0.5
88     mu = lambda k: sum([(key - mean) ** k * val[1]
89 for key, val in nv.items()])
90     skew = mu(3) / (standart ** 3)
91     kurtosis = mu(4) / (standart ** 4) - 3
92     mode = st.mode(r)
93     if len(st_r) % 2 != 0:
94         med = st_r[int(len(st_r) / 2)]
95     else:
96         med = 0.5 * (st_r[int(len(st_r) / 2)] +
97 st_r[int(len(st_r) / 2) - 1])
98     return mean, var, standart, skew, kurtosis, mode,
99 med
100
101 def binomial(np, size, write, read):
102     n, p = np
103     r = load('binomial') if read else st.binom.rvs(n,
104 p, size=size)
105     if write:
106         save(r, 'binomial')
107
```

```
    st_r = sorted(r)
    p_ch = Counter(r)
    nv = {key: (val, val / sum(p_ch.values())) for
key, val in p_ch.items()}

    ex_mean, ex_var, ex_standart, ex_skew,
ex_kurtosis, ex_mode, ex_med = exp_stats(nv, r, st_r)

    draw_poligon(p_ch, 'binomial')

    delta = 0.01
    items = list(zip(list(range(max(r) + 2)),
islice(list(range(max(r) + 2)), 1, None))) # [(0, 1),
(1, 2), ...]
    Xlist = [[x * delta for x in range(item[0] *
int(1 / delta), item[1] * int(1 / delta))]
                for item in items]
    Ylist = [[st.binom.cdf(x, n, p) for x in line]
for line in Xlist]

    draw_cdf(Xlist, Ylist, 'binomial')

    cdf_func = [el[0] for el in Ylist]
    # cdf_func.append(1.0)
    # cdf_func.insert(0, 0.0)

    mean, variance, skew, kurtosis =
st.binom.stats(n, p, moments='mvsk') # среднее,
дисперсия, ассиметрия, эксцесс
    standart, med = st.binom.std(n, p),
st.binom.median(n, p) # стантартное отклонение,
медиана
    mode = st.mode(r) # мода

    pmf = [abs(val[1] - st.binom.pmf(key, n, p)) for
key, val in nv.items()]
```

```
144
145     return (r, st_r, cdf_func, nv, float(mean),
146                                        float(variance),
147                                        float(skew),
148                                        float(kurtosis),
149                                        standart,
150                                        med,
151                                        float(mode.mode),
152 ex_mean,
153
154 ex_var,
155
156 ex_standart,
157
158 ex_skew,
159
160 ex_kurtosis,
161
162 float(ex_mode.mode),
163
164 ex_med, pmf)
165
166 def geometr(p, size, write, read):
167     r = load('geometr') if read else st.geom.rvs(p,
168 size=size)
169     if write:
170         save(r, 'geometr')
171
172     st_r = sorted(r)
173     p_ch = Counter(r)
174     nv = {key: (val, val / sum(p_ch.values())) for
175 key, val in p_ch.items()}
176
177     ex_mean, ex_var, ex_standart, ex_skew,
178 ex_kurtosis, ex_mode, ex_med = exp_stats(nv, r, st_r)
179
```

```
180     draw_poligon(p_ch, 'geometr')
181
182     delta = 0.01
183     items = list(zip(list(range(max(r) + 2)),
184 islice(list(range(max(r) + 2)), 1, None))) # [(0, 1),
185 (1, 2), ...]
186     Xlist = [[x * delta for x in range(item[0] *
187 int(1 / delta), item[1] * int(1 / delta))]
188                 for item in items]
189     Ylist = [[st.geom.cdf(x, p) for x in line] for
190 line in Xlist]
191
192     draw_cdf(Xlist, Ylist, 'geometr')
193
194     cdf_func = [el[0] for el in Ylist]
195     cdf_func.append(1.0)
196     cdf_func.insert(0, 0.0)
197
198     mean, variance, skew, kurtosis = st.geom.stats(p,
199 moments='mvsk') # среднее, дисперсия, ассиметрия,
200 эксцесс
201     standart, med = st.geom.std(p), st.geom.median(p)
202 # стантартное отклонение, медиана
203     mode = st.mode(r) # мода
204
205     pmf = [abs(val[1] - st.geom.pmf(key, p)) for key,
206 val in nv.items()]
207
208     return (r, st_r, cdf_func, nv, float(mean),
209                                     float(variance),
210                                     float(skew),
211                                     float(kurtosis),
212                                     standart,
213                                     med,
214                                     float(mode.mode),
215 ex_mean,
```

```
216
217 ex_var,
218
219 ex_standart,
220
221 ex_skew,
222
223 ex_kurtosis,
224
225 float(ex_mode.mode),
226
227 ex_med, pmf)
228
229 def puasson(mu, size, write, read):
230     r = load('puasson') if read else
231 st.poisson.rvs(mu, size=size)
232     if write:
233         save(r, 'puasson')
234
235     st_r = sorted(r)
236     p_ch = Counter(r)
237     nv = {key: (val, val / sum(p_ch.values())) for
238 key, val in p_ch.items()}
239
240     ex_mean, ex_var, ex_standart, ex_skew,
241 ex_kurtosis, ex_mode, ex_med = exp_stats(nv, r, st_r)
242
243     draw_poligon(p_ch, 'puasson')
244
245     delta = 0.01
246     items = list(zip(list(range(max(r) + 2)),
247 islice(list(range(max(r) + 2)), 1, None))) # [(0, 1),
248 (1, 2), ...]
249     Xlist = [[x * delta for x in range(item[0] *
250 int(1 / delta), item[1] * int(1 / delta))]
251                 for item in items]
```

```
252     Ylist = [[st.poisson.cdf(x, mu) for x in line]
253 for line in Xlist]
254 
255     draw_cdf(Xlist, Ylist, 'puasson')
256 
257     cdf_func = [el[0] for el in Ylist]
258     cdf_func.append(1.0)
259     cdf_func.insert(0, 0.0)
260 
261     mean, variance, skew, kurtosis =
262 st.poisson.stats(mu, moments='mvsk') # среднее,
263 дисперсия, ассиметрия, эксцесс
264     standart, med = st.poisson.std(mu),
265 st.poisson.median(mu) # стантартное отклонение,
266 медиана
267     mode = st.mode(r) # мода
268 
269     pmf = [abs(val[1] - st.poisson.pmf(key, mu)) for
270 key, val in nv.items()]
271 
272     return (r, st_r, cdf_func, nv, float(mean),
273                                     float(variance),
                                        float(skew),
                                        float(kurtosis),
                                        standart,
                                        med,
                                        float(mode.mode),
    ex_mean,

    ex_var,

    ex_standart,

    ex_skew,

    ex_kurtosis,
```

```
float(ex_mode.mode),

ex_med, pmf)

def save_result(result):
    reg = 'data/result*'
    files = glob(reg)
    if files:
        n = int(re.sub(r'[^\d]', '', files[0]))
        n += 1
    else:
        n = 0
    filename = reg[:-1] + str(n) + '.txt'
    with open(filename, 'w') as f:
        f.write(result)


if __name__ == '__main__':
    parser = createParser()
    namespace = parser.parse_args(sys.argv[1:])

    result = ['выборка: {0}',
              'упорядоченная: {1}',
              'эмпирическая функция распределения:
{2}',
              'статистический ряд: {3}',
              'теор. среднее: {4} <=> среднее: {11}',
              'теор. дисперсия: {5} <=> дисперсия:
{12}',
              'теор. ассиметрия: {6} <=> ассиметрия:
{14}',
              'теор. эксцесс: {7} <=> эксцесс: {15}',
              'теор. среднее квадратичное отклонение:
{8} <=> среднее квадратичное отклонение: {13}',
              'теор. медиана: {9} <=> медиана: {17}',
```

```
                'теор. мода: {10} <=> мода: {16}',
                'pmf: {18}']

    line = '\n\n'.join(result)

    task = {'binomial': (binomial, [5 + variant % 17,
0.1 + 0.01 * variant]),
            'geometr': (geometr, 0.1 + 0.01 *
variant),
            'puasson': (puasson, 0.7 + 0.07 *
variant)}

    out = ''
    for r_type, fdata in task.items():
        func, params = fdata
        res = func(params, size=150,
                           write = namespace.write,
                           read = namespace.read)

        out += r_type + '\n\n' + line.format(*res) +
'\n\n<' + '='*50 + '>\n\n'

    save_result(out)
```