



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

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 $\{x_1, x_2, x_3, \dots, x_N\}$ упорядочить по возрастанию, определить частоты n_i и относительные частоты (частости) w_i , построить статистический ряд:

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

Выборочное среднее квадратическое отклонение

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

Выборочная медиана

$$\overline{M_e} = \begin{cases} x_i^*, & F_N^3(x_{i-1}^*) < 0,5 < F_N^3(x_i^*) \\ \frac{1}{2}(x_i^* + x_{i+1}^*), & F_N^3(x_i^*) = 0,5 \end{cases}$$

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

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

Выборочный коэффициент асимметрии

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

Выборочный коэффициент эксцесса

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

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 хвост 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: $\left[-\frac{\ln 2}{\ln q} \right]$, если $\frac{\ln 2}{\ln q}$ – дробное; $-\frac{\ln 2}{\ln q} - \frac{1}{2}$, если $\frac{\ln 2}{\ln q}$ – целое

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

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 средней 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:

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

Упорядоченная:

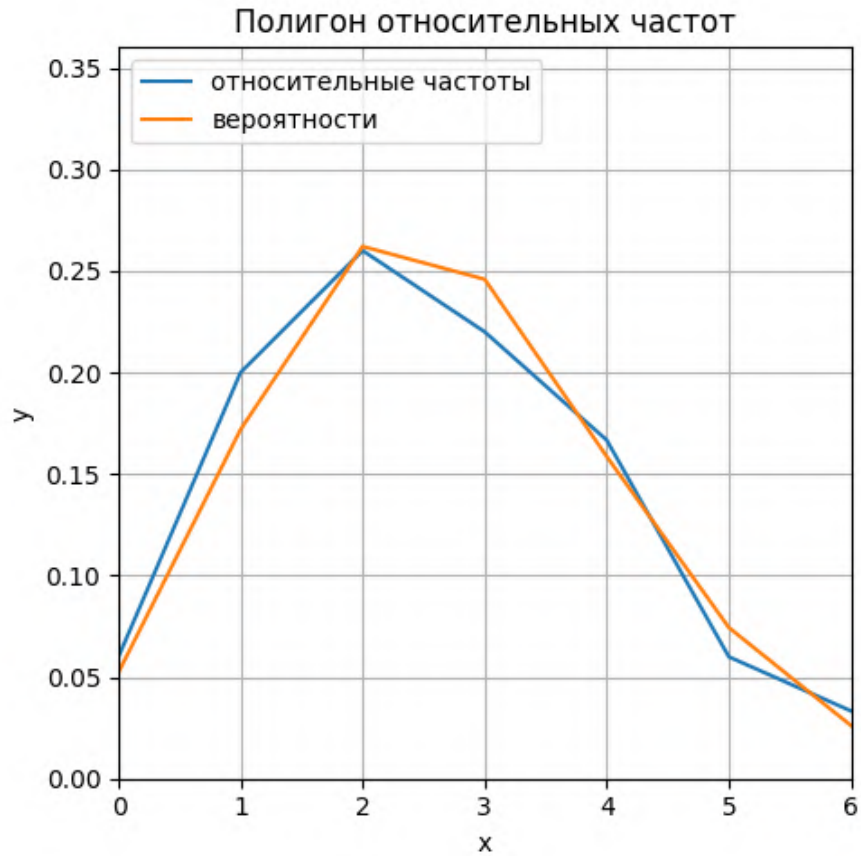
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	4	4	4	4	4	4	4	4	4
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
4	5	5	5	5	5	5	5	5	5	6	6	6	6	6

Статистический ряд:

x_i	0	1	2	3	4	5	6
n_i	9	30	39	33	25	9	5

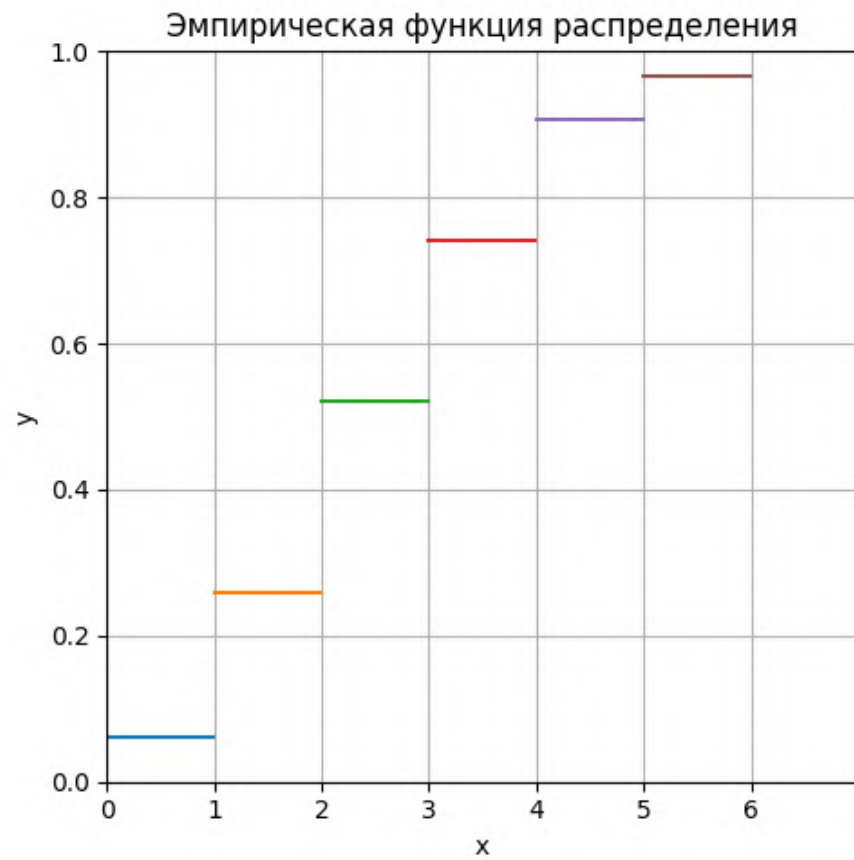
w_i	0.06	0.2	0.26	0.22	0.1666	0.06	0.0333
-------	------	-----	------	------	--------	------	--------

Полигон относительных частот:



Эмпирическая функция распределения:

$$F_{150}^{\partial}(x) = \sum_{x_i \leq x} w_i = \begin{cases} 0, & x < 0 \\ 0.06, & 0 \leq x < 1 \\ 0.26, & 1 \leq x < 2 \\ 0.52, & 2 \leq x < 3 \\ 0.74, & 3 \leq x < 4 \\ 0.9066, & 4 \leq x < 5 \\ 0.9666, & 5 \leq x < 6 \\ 1, & x \geq 6 \end{cases}$$



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

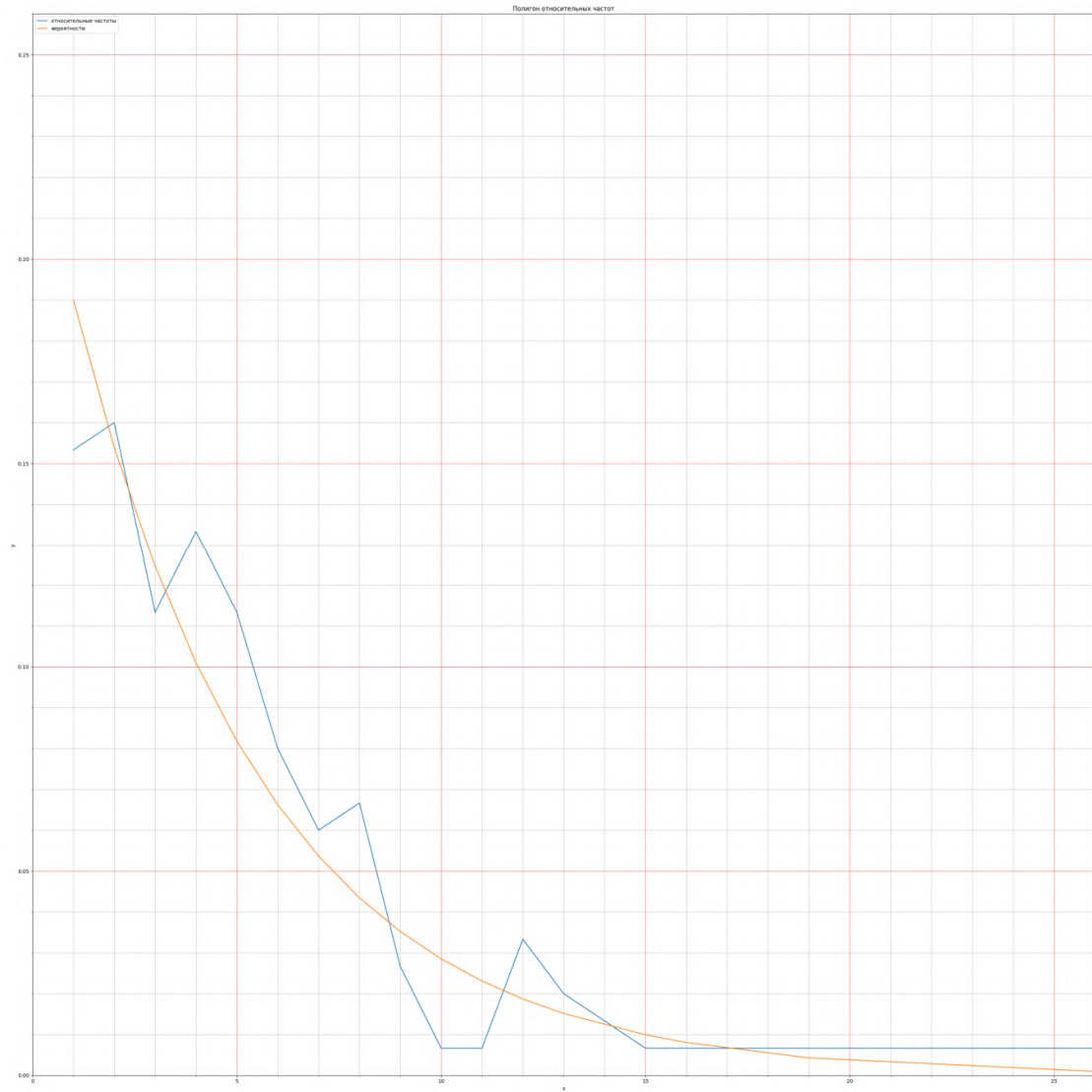
w_i	0.08	0.06	0.06	0.02	0.0066	0.0066
-------	------	------	------	------	--------	--------

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

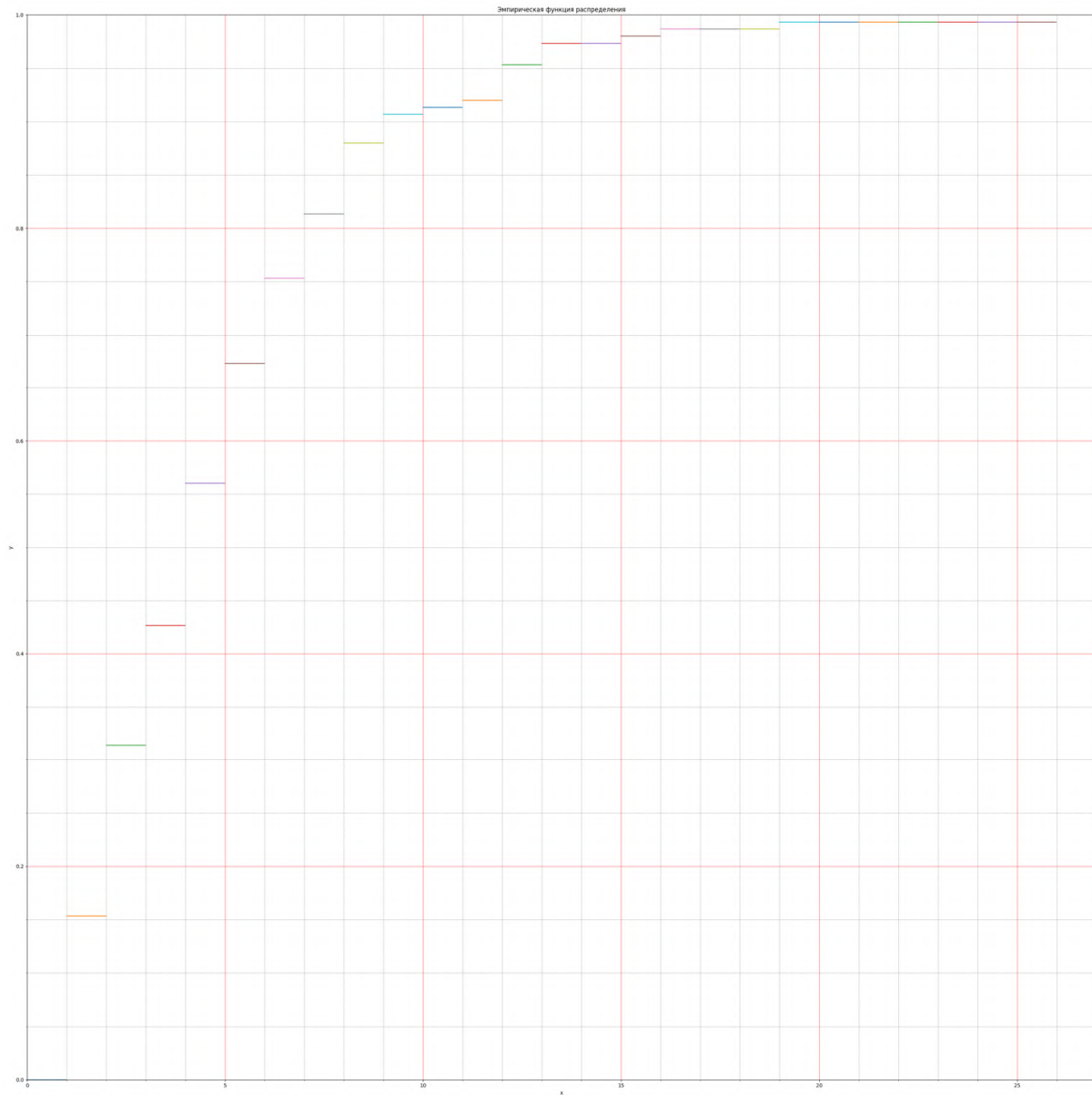
x_i	24	25	26
n_i	0	0	1
w_i	0.0	0.0	0.0066

Полигон относительных частот:



Эмпирическая функция распределения:

$$F_{150}^{\mathfrak{A}}(x) = \sum_{x_i \leq x} w_i = \left\{ \begin{array}{ll} 0, & x < 0 \\ 0, & 0 \leq x < 1 \\ 0.1533, & 1 \leq x < 2 \\ 0.3133, & 2 \leq x < 3 \\ 0.4266, & 3 \leq x < 4 \\ 0.56, & 4 \leq x < 5 \\ 0.6733, & 5 \leq x < 6 \\ 0.7533, & 6 \leq x < 7 \\ 0.8133, & 7 \leq x < 8 \\ 0.8799, & 8 \leq x < 9 \\ 0.9066, & 9 \leq x < 10 \\ 0.9133, & 11 \leq x < 12 \\ 0.9199, & 12 \leq x < 13 \\ 0.9533, & 13 \leq x < 14 \\ 0.9733, & 14 \leq x < 15 \\ 0.9733, & 15 \leq x < 16 \\ 0.98, & 17 \leq x < 18 \\ 0.9866, & 18 \leq x < 19 \\ 0.9866, & 19 \leq x < 20 \\ 0.9866, & 20 \leq x < 21 \\ 0.9933, & 21 \leq x < 22 \\ 0.9933, & 22 \leq x < 23 \\ 0.9933, & 23 \leq x < 24 \\ 0.9933, & 24 \leq x < 25 \\ 0.9933, & 25 \leq x < 26 \\ 1, & x \geq 26 \end{array} \right.$$



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

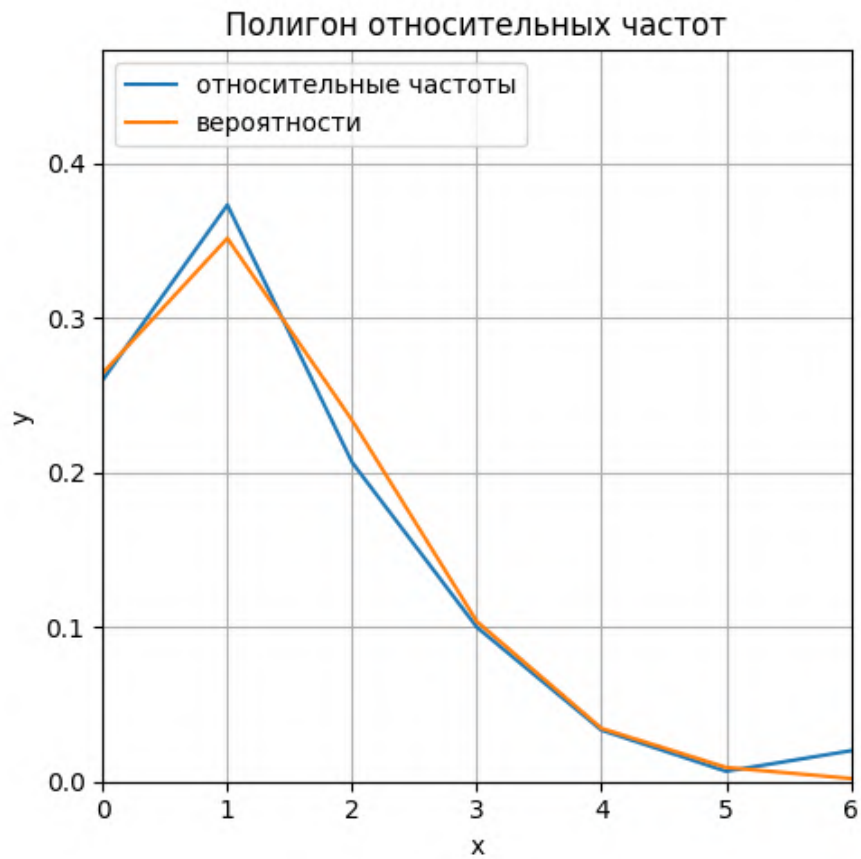
Статистический ряд:

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

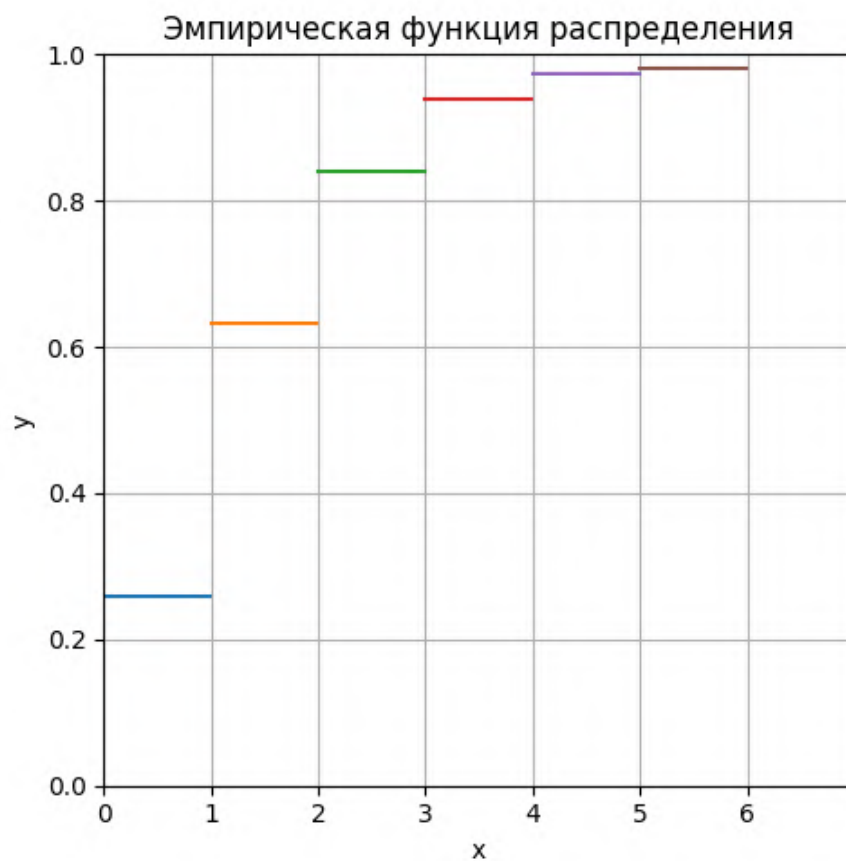
w_i	0.0067	0.02
-------	--------	------

Полигон относительных частот:



Эмпирическая функция распределения:

$$F_{150}^{\exists}(x) = \sum_{x_i \leq x} w_i = \begin{cases} 0, & x < 0 \\ 0.26, & 0 \leq x < 1 \\ 0.6333, & 1 \leq x < 2 \\ 0.84, & 2 \leq x < 3 \\ 0.94, & 3 \leq x < 4 \\ 0.9733, & 4 \leq x < 5 \\ 0.98, & 5 \leq x < 6 \\ 1, & x \geq 6 \end{cases}$$



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

Анализ результатов

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%

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

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

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%

$\{|w_i - p_i|\} = [0.0366, 0.006, 0.0113, 0.032, 0.0315, 0.0137, 0.00639, 0.0232, 0.0085, 0.0218, 0.0164, 0.0146, 0.0048, 0.0032, 0.0013, 0.002, 0.0056]$
 $\max \{|w_i - p_i|, i=1, \dots, m\} = 0.0366$

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%

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

$$\max\{|w_i - p_i|, i=1, \dots, m\} = 0.0179 \text{ чы}$$

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

```

1 #!/usr/bin/python
2 # -*- coding: UTF-8 -*-
3
4 import sys
5 import argparse
6 from collections import Counter
7 from itertools import islice
8
9 from functools import reduce
10 from scipy import stats as st
11 import numpy as np
12 from glob import glob
13 import pickle
14 import re
15
16 import matplotlib.pyplot as plt
17
18 variant = 9
19
20 def createParser():
21     parser = argparse.ArgumentParser()
22     parser.add_argument('-w', '--write', default=0, type=int)
23     parser.add_argument('-r', '--read', default=1, type=int)
24     return parser
25
26 def save(r, type_ryad):
27     reg = 'data/' + type_ryad + '*'
28     files = glob(reg) # type: ryadN.pickle
29     if files:
30         n = int(re.sub(r'^\d+', '', files[0]))
31         n += 1
32     else:
33         n = 0
34     filename = reg[:-1] + str(n) + '.pickle'
35     with open(filename, 'wb') as f:
36         pickle.dump(r, f)
37
38 def load(type_ryad, n = -1):
39     reg = 'data/' + type_ryad + '*'
40     files = glob(reg)
41     if not files:
42         return
43     if abs(n) > len(files):
44         return
45     with open(files[n], 'rb') as f:
46         return(pickle.load(f))
47
48 def draw_poligon(count, pr, name, size_x=5, size_y=5):
49     X = list(count.keys())

```

```

50     Y = [count[x][1] for x in X]
51     fig, ax = plt.subplots(figsize=(sizeX, sizeY))
52     ax.plot(X, Y, label='относительные частоты')
53     ax.plot(X, pr, label='вероятности')
54     ax.set_title('Полигон относительных частот')
55     ax.legend(loc='upper left')
56     ax.set_ylabel('y')
57     ax.set_xlabel('x')
58     ax.set_xlim(xmin=0, xmax=max(X))
59     ax.set_ylim(ymin=0, ymax=max(Y) + 0.1)
60     if sizeX != 5:
61         # Don't allow the axis to be on top of your data
62         ax.set_axisbelow(True)
63
64         # Turn on the minor TICKS, which are required for the minor GRID
65         ax.minorticks_on()
66
67         # Customize the major grid
68         ax.grid(which='major', linestyle='-', linewidth='0.5', color='red')
69         # Customize the minor grid
70         ax.grid(which='minor', linestyle=':', linewidth='0.5',
71 color='black')
72     else:
73         ax.grid()
74     fig.tight_layout()
75     fig.savefig('data/poligon_' + name + '.png')
76
77 def draw_cdf(Xlist, Ylist, name, sizeX=5, sizeY=5):
78     fig, ax = plt.subplots(figsize=(sizeX, sizeY))
79
80     for X, Y in zip(Xlist, Ylist):
81         ax.plot(X, Y, label='')
82     ax.set_title('Эмпирическая функция распределения')
83     ax.legend(loc='upper left')
84     ax.set_ylabel('y')
85     ax.set_xlabel('x')
86     ax.set_xlim(xmin=0, xmax=max(X))
87     ax.set_ylim(ymin=0, ymax=1)
88     fig.tight_layout()
89     if sizeX != 5:
90         # Don't allow the axis to be on top of your data
91         ax.set_axisbelow(True)
92
93         # Turn on the minor TICKS, which are required for the minor GRID
94         ax.minorticks_on()
95
96         # Customize the major grid
97         ax.grid(which='major', linestyle='-', linewidth='0.5', color='red')
98         # Customize the minor grid
99         ax.grid(which='minor', linestyle=':', linewidth='0.5',
100 color='black')

```

```

101     else:
102         ax.grid()
103     fig.savefig('data/cdf_' + name + '.png')
104
105 def expect(nv, key):
106     if key in nv.keys():
107         return nv[key][1]
108     else:
109         return 0.0
110     # elif key not in nv.keys() and key != 0:
111     #     return expect(nv, key-1)
112     # else:
113     #     return 0.0
114
115 def generate_ef(nv, r):
116     delta = 0.01
117     items = list(zip(list(range(max(r) + 2)), islice(list(range(max(r) +
118 2)), 1, None))) # [(0, 1), (1, 2), ...]
119     keys = list(range(list(nv.keys())[-1] + 1)) # list(nv.keys())
120     Y = [sum([expect(nv, key) for key in keys[:i + 1]]) for i, el in
121 enumerate(keys)]
122     Xlist = [[x * delta for x in range(item[0] * int(1 / delta), item[1] *
123 int(1 / delta))]
124             for item in items]
125     Ylist = [[y] * len(Xlist[i]) for i, y in enumerate(Y)]
126     # Ylist = [[st.binom.cdf(x, n, p) for x in line] for line in Xlist]
127     return Xlist, Ylist, Y
128
129 def exp_stats(nv, r, st_r):
130     mean = sum([key * val[1] for key, val in nv.items()]) # mean
131     var = sum([(key - mean) ** 2 * val[1] for key, val in nv.items()])
132     standart = var ** 0.5
133     mu = lambda k: sum([(key - mean) ** k * val[1] for key, val in
134 nv.items()])
135     skew = mu(3) / (standart ** 3)
136     kurtosis = mu(4) / (standart ** 4) - 3
137     mode = st.mode(r)
138     if len(st_r) % 2 != 0:
139         med = st_r[int(len(st_r) / 2)]
140     else:
141         med = 0.5 * (st_r[int(len(st_r) / 2)] + st_r[int(len(st_r) / 2) -
142 1])
143     return mean, var, standart, skew, kurtosis, mode, med
144
145 def binomial(np, size, write, read):
146     n, p = np
147     r = load('binomial') if read else st.binom.rvs(n, p, size=size)
148     if write:
149         save(r, 'binomial')
150
151     st_r = sorted(r)

```

```

152     p_ch = Counter(r)
153     nv = {key: (val, val / sum(p_ch.values())) for key, val in p_ch.items()}
154
155     ex_mean, ex_var, ex_standart, ex_skew, ex_kurtosis, ex_mode, ex_med =
156 exp_stats(nv, r, st_r)
157
158     pr = [st.binom.pmf(x, n, p) for x in list(nv.keys())]
159     draw_poligon(nv, pr, 'binomial')
160
161     # delta = 0.01
162     # items = list(zip(list(range(max(r) + 2)), islice(list(range(max(r) +
163 2)), 1, None))) # [(0, 1), (1, 2), ...]
164     # Xlist = [[x * delta for x in range(item[0] * int(1 / delta), item[1] *
165 int(1 / delta))]
166     #         for item in items]
167     # Ylist = [[st.binom.cdf(x, n, p) for x in line] for line in Xlist]
168
169     Xlist, Ylist, Y = generate_ef(nv, r)
170
171     draw_cdf(Xlist, Ylist, 'binomial')
172
173     cdf_func = [el[0] for el in Ylist]
174     # cdf_func.append(1.0)
175     # cdf_func.insert(0, 0.0)
176
177     mean, variance, skew, kurtosis = st.binom.stats(n, p, moments='mvsk') #
178 среднее, дисперсия, асимметрия, эксцесс
179     standart, med = st.binom.std(n, p), st.binom.median(n, p) # стантартное
180 отклонение, медиана
181     mode = st.mode(r) # мода
182
183     pmf = [abs(val[1] - st.binom.pmf(key, n, p)) for key, val in nv.items()]
184
185     return (r, st_r, cdf_func, nv, float(mean),
186           float(variance),
187           float(skew),
188           float(kurtosis),
189           standart,
190           med,
191           float(mode.mode), ex_mean,
192           ex_var,
193           ex_standart,
194           ex_skew,
195           ex_kurtosis,
196           float(ex_mode.mode),
197           ex_med, pmf, Y)
198
199 def geometr(p, size, write, read):
200     r = load('geometr') if read else st.geom.rvs(p, size=size)
201     if write:
202         save(r, 'geometr')

```

```

203
204     st_r = sorted(r)
205     p_ch = Counter(r)
206     nv = {key: (val, val / sum(p_ch.values())) for key, val in p_ch.items()}
207
208     ex_mean, ex_var, ex_standart, ex_skew, ex_kurtosis, ex_mode, ex_med =
209 exp_stats(nv, r, st_r)
210
211     pr = [st.geom.pmf(x, p) for x in list(nv.keys())]
212     draw_poligon(nv, pr, 'geometr', 30, 30)
213
214     # delta = 0.01
215     # items = list(zip(list(range(max(r) + 2)), islice(list(range(max(r) +
216 2)), 1, None))) # [(0, 1), (1, 2), ...]
217     # Xlist = [[x * delta for x in range(item[0] * int(1 / delta), item[1] *
218 int(1 / delta))]
219     #         for item in items]
220     # Ylist = [[st.geom.cdf(x, p) for x in line] for line in Xlist]
221
222     Xlist, Ylist, Y = generate_ef(nv, r)
223
224     draw_cdf(Xlist, Ylist, 'geometr', 30, 30)
225
226     cdf_func = [el[0] for el in Ylist]
227     cdf_func.append(1.0)
228     cdf_func.insert(0, 0.0)
229
230     mean, variance, skew, kurtosis = st.geom.stats(p, moments='mvsk') #
231 среднее, дисперсия, асимметрия, эксцесс
232     standart, med = st.geom.std(p), st.geom.median(p) # стандартное
233 отклонение, медиана
234     mode = st.mode(r) # мода
235
236     pmf = [abs(val[1] - st.geom.pmf(key, p)) for key, val in nv.items()]
237
238     return (r, st_r, cdf_func, nv, float(mean),
239           float(variance),
240           float(skew),
241           float(kurtosis),
242           standart,
243           med,
244           float(mode.mode), ex_mean,
245           ex_var,
246           ex_standart,
247           ex_skew,
248           ex_kurtosis,
249           float(ex_mode.mode),
250           ex_med, pmf, Y)
251
252 def puasson(mu, size, write, read):
253     r = load('puasson') if read else st.poisson.rvs(mu, size=size)

```

```

254     if write:
255         save(r, 'puasson')
256
257     st_r = sorted(r)
258     p_ch = Counter(r)
259     nv = {key: (val, val / sum(p_ch.values())) for key, val in p_ch.items()}
260
261     ex_mean, ex_var, ex_standart, ex_skew, ex_kurtosis, ex_mode, ex_med =
262 exp_stats(nv, r, st_r)
263
264     pr = [st.poisson.pmf(x, mu) for x in list(nv.keys())]
265     draw_poligon(nv, pr, 'puasson')
266
267     # delta = 0.01
268     # items = list(zip(list(range(max(r) + 2)), islice(list(range(max(r) +
269 2)), 1, None))) # [(0, 1), (1, 2), ...]
270     # Xlist = [[x * delta for x in range(item[0] * int(1 / delta), item[1] *
271 int(1 / delta))]
272     #         for item in items]
273     # Ylist = [[st.poisson.cdf(x, mu) for x in line] for line in Xlist]
274
275     Xlist, Ylist, Y = generate_ef(nv, r)
276
277     draw_cdf(Xlist, Ylist, 'puasson')
278
279     cdf_func = [el[0] for el in Ylist]
280     cdf_func.append(1.0)
281     cdf_func.insert(0, 0.0)
282
283     mean, variance, skew, kurtosis = st.poisson.stats(mu, moments='mvsk') #
284 среднее, дисперсия, асимметрия, эксцесс
285     standart, med = st.poisson.std(mu), st.poisson.median(mu) # стандартное
286 отклонение, медиана
287     mode = st.mode(r) # мода
288
289     pmf = [abs(val[1] - st.poisson.pmf(key, mu)) for key, val in nv.items()]
290
291     return (r, st_r, cdf_func, nv, float(mean),
292           float(variance),
293           float(skew),
294           float(kurtosis),
295           standart,
296           med,
297           float(mode.mode), ex_mean,
298           ex_var,
299           ex_standart,
300           ex_skew,
301           ex_kurtosis,
302           float(ex_mode.mode),
303           ex_med, pmf, Y)
304

```

```

305 def save_result(result):
306     reg = 'data/result*'
307     files = glob(reg)
308     if files:
309         n = int(re.sub(r'^\d+', '', files[0]))
310         n += 1
311     else:
312         n = 0
313     filename = reg[:-1] + str(n) + '.txt'
314     with open(filename, 'w') as f:
315         f.write(result)
316
317
318 if __name__ == '__main__':
319     parser = createParser()
320     namespace = parser.parse_args(sys.argv[1:])
321
322     result = ['выборка: {0}',
323              'упорядоченная: {1}',
324              'эмпирическая функция распределения: {2}',
325              'статистический ряд: {3}',
326              'теор. среднее: {4} <=> среднее: {11}',
327              'теор. дисперсия: {5} <=> дисперсия: {12}',
328              'теор. асимметрия: {6} <=> асимметрия: {14}',
329              '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}',
              'ex_pmf {19}']

    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)

```