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

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Типовой расчет по математической статистике

Часть 1

ВАРИАНТ 165

Выполнил:
Студент 3-го курса
Баттур Ц.

Группа: КМБО-07-22

МОСКВА – 2025

Содержание

Задание

Задание 1. Получить выборку объёмом N=200, сгенерировав псевдослучайные числа, распределённые по биномиальному закону с параметрами $\boldsymbol{n}$ и $\boldsymbol{p}$:

$$p_k = C_n^k * p^k * q^{n-k}, k = 0,1,2,\ldots,n, n = 5 + V mod 20, p = 0{,}2 + 0{,}003 * V$$

Задание 2. Получить выборку объёмом N=200, сгенерировав псевдослучайные числа, распределённые по геометрическому закону с параметром $\boldsymbol{p}$:

$$p_k = q^k * p, k = 0,1,\ldots, p = 0{,}2 + 0{,}003 * V$$

В заданиях 1 и 2 построить:

1) статистический ряд;
2) график полигона относительных частот с наложенным на него и выделенным красным цветом график полигона теоретических вероятностей;
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;

составить таблицы:

1) сравнения относительных частот и теоретических вероятностей;
2) сравнения рассчитанных характеристик с теоретическими значениями.

Задание 3. Получить выборку объёмом N=200, сгенерировав псевдослучайные числа, распределённые по показательному закону с параметром $\lambda = 1 + (-1)^V * 0{,}003 * V$.

В задании 3 построить:

1) график эмпирической функции распределения;
2) интервальный ряд и ассоциированный статистический ряд;
3) гистограмму относительных частот с наложенным на неё и выделенным красным цветом график плотности распределения;

найти:

1) выборочное среднее;

2) выборочную дисперсия с поправкой Шеппарда;

3) выборочное среднее квадратическое отклонение;

4) выборочную моду;

5) выборочную медиану;

6) выборочный коэффициент асимметрии;

7) выборочный коэффициент эксцесса;

составить таблицы:

1) сравнения относительных частот и теоретических вероятностей попадания в интервалы;

2) сравнения рассчитанных характеристик с теоретическим значениями.

Краткие теоретические сведения

Выборка объёмом N=200 с сгенерированными псевдослучайными числами, распределённые по биномиальному закону с параметрами $\boldsymbol{n}$ и $\boldsymbol{p}$:

$$p_k = C_n^k * p^k * q^{n-k}, k = 0,1,2,\ldots,n, n = 5 + V mod 20, p = 0,2 + 0,003 * V$$

Выборка объёмом N=200 с сгенерированными псевдослучайными числами, распределённые по геометрическому закону с параметром $\boldsymbol{p}$:

$$p_k = q^k * p, k = 0,1,\ldots, p = 0,2 + 0,003 * V$$

Полученные выборки упорядочить по возрастанию, определить частоты n_i и относительные частоты w_i, построить статистический ряд.

x_i^*	n_i	w_i	s_i
x_1^*	n_1	w_1	s_1
x_2^*	n_1	w_2	s_2
…	…	…	…
x_m^*	n_m	w_m	s_m
	$\sum_{i=1}^{m} n_i$	$\sum_{i=1}^{m} w_i$	-

Таблица 1. Статистический ряд.

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:

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

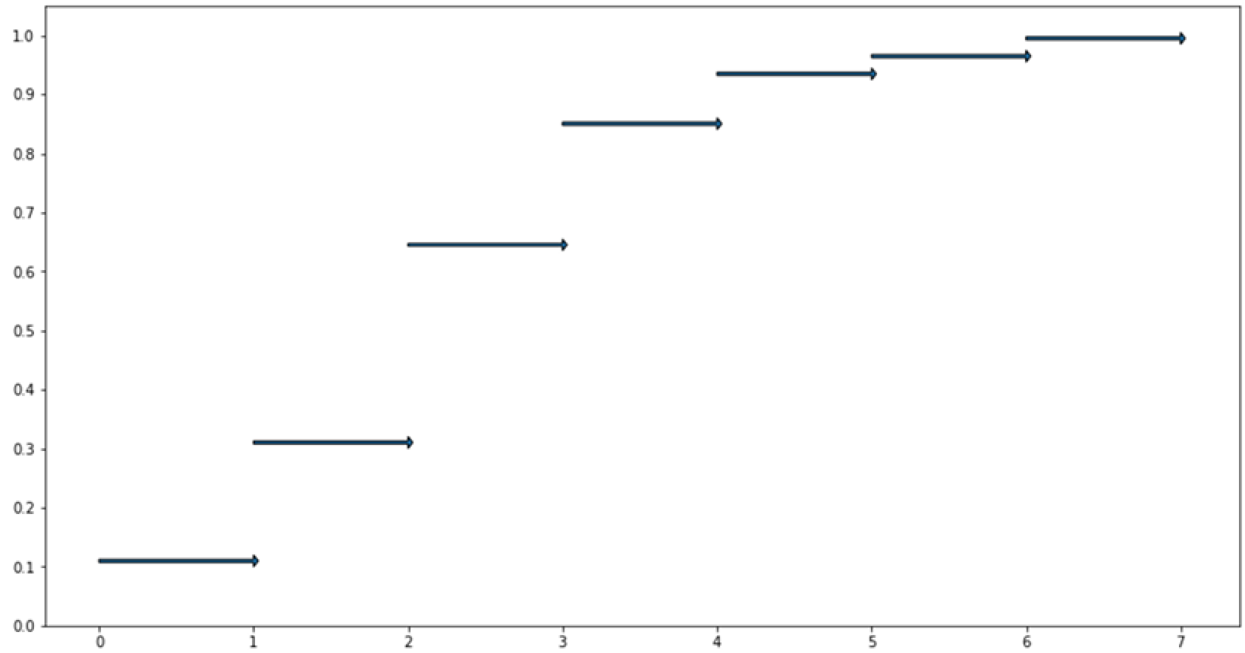

Рисунок 1. Образец графика эмпирической функции распределения.

Выборочное среднее:

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

Выборочный момент k-ого порядка (выборочный k-ый момент):

$$\bar{\mu} = \sum_{i=1}^{m} (x_i^*)^k * w_i \, , \overline{\mu_1} = \bar{x}$$

Выборочная дисперсия:

$$D_B = \sum_{i=1}^{m} (x_i^* - \overline{x})^2 * w_i = \overline{\mu_2} - (\overline{\mu_1})^2$$

Выборочная центральный момент k-ого порядка:

$$\overline{\mu}_k^0 = \sum_{i=1}^{m} (x_i^* - \overline{x})^k * w_i \, , \overline{\mu}_1^0 = 0, \overline{\mu}_2^0 = D_B$$

$$\overline{\mu}_3^0 = \overline{\mu}_3 - 3\overline{\mu}_2\overline{\mu}_1 + 2\left(\overline{\mu}_1\right)^3$$

$$\overline{\mu}_4^0 = \overline{\mu}_4 - 4\overline{\mu}_3\overline{\mu}_1 + 6\overline{\mu}_2\left(\overline{\mu}_1\right)^2 - 3\left(\overline{\mu}_1\right)^4$$

Выборочное среднее квадратическое отклонение:

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

Выборочный коэффициент асимметрии:

$$\overline{\gamma}_1 = \frac{\overline{\mu}_3^0}{\overline{\sigma}^2}$$

Выборочный коэффициент эксцесса:

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

Выборочная мода $\overline{M}_0 = \{x_i^* | n_i = max\, n_k\}$, если $n_i = max\, n_k > n_j, i \neq j$;

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

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

Выборочная медиана:

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

Биномиальное распределение	
Вероятность	$p_k = C_n^k * p^k * q^{n-k}, k = 0,1, \ldots n, q = 1 - p$
Математическое ожидание	np
Дисперсия	npq
Среднее квадратическое отклонение	$\sqrt{npq}$
Мода	$[(n+1)p]$, если $(n+1)p$ - дробное; $\left[(n+1)p - \frac{1}{2}\right]$, если $(n+1)p$ - целое;
Медиана	$min\left\{k: \sum_{i=1}^{k} p_i \geq 0{,}5\right\}$
Коэффициент асимметрии	$\frac{q-p}{\sqrt{npq}}$
Коэффициент эксцесса	$\frac{1-6pq}{npq}$

Таблица 2. Характеристики биномиального распределения.

Геометрическое распределение	
Вероятность	$p_k = q^k * p, k = 0,1,\ldots, q = 1 - p$
Математическое ожидание	$\frac{q}{p}$
Дисперсия	$\frac{q}{p^2}$
Среднее квадратическое отклонение	$\frac{\sqrt{q}}{p}$
Мода	0
Медиана	$min\left\{k: \sum_{i=1}^{k} p_i \geq 0,5\right\}$
Коэффициент асимметрии	$\frac{1+q}{\sqrt{q}}$
Коэффициент эксцесса	$6 + \frac{p^2}{q}$

Таблица 3. Характеристики геометрического распределения.

В задании 3 полученную выборку псевдослучайных чисел, распределённые по показательному закону, упорядочить по возрастанию, определить интервалы $[a_0, a_1], (a_1, a_2], \ldots, (a_{m-1}, a_m]$; число интервалов находится по формуле Стерджеса $m = 1 + [\log_2 N]$; $a_0 = 0, a_m = max\{x_i\}$.

Интервалы	n_i	w_i
$[a_0, a_1]$	n_1	w_1
$(a_1, a_2]$	n_2	w_2
…	…	…
$(a_{m-1}, a_m]$	n_m	w_m
	$\sum_{i=1}^{m} n_i$	$\sum_{i=1}^{m} w_i$

Таблица 4. Интервальный ряд.

n_i - число значений, попавших в i-ый интервал; w_i - относительная частота попадания в i-ый интервал, $w_i = \frac{n_i}{N}$.

x_i^*	n_i	w_i
x_1^*	n_1	w_1
x_2^*	n_2	w_2
…	…	…
x_m^*	n_m	w_m
	$\sum_{i=1}^{m} n_i$	$\sum_{i=1}^{m} w_i$

Таблица 5. Ассоциированный статический ряд, где $x_i^* = \frac{a_{i-1}+a_i}{2}$ - середина интервала.

Эмпирическая функция распределения:

$$F_N^{Э}(x; x_1, x_2, \dots, x_N) = \sum_{x_k \le x} \frac{1}{N} = \begin{cases} 0, & x < x_1, \\ \frac{1}{N}, & x_1 \le x < x_2, \\ \frac{2}{N}, & x_2 \le x < x_3, \\ \dots\dots\dots\dots\dots\dots\dots\dots & \\ 1, & x \ge x_N. \end{cases}$$

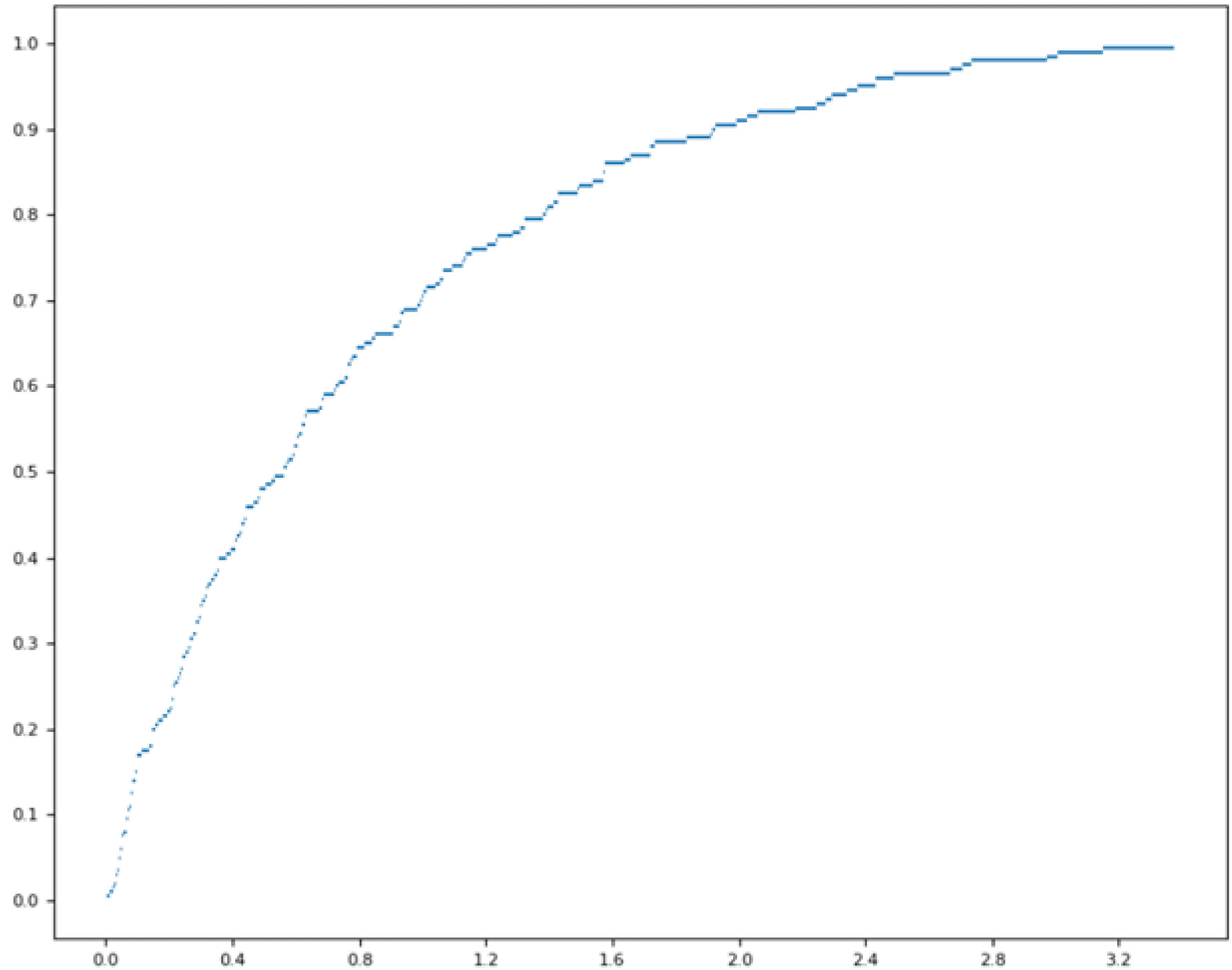

Рисунок 2. Образец графика эмпирической функции распределения.

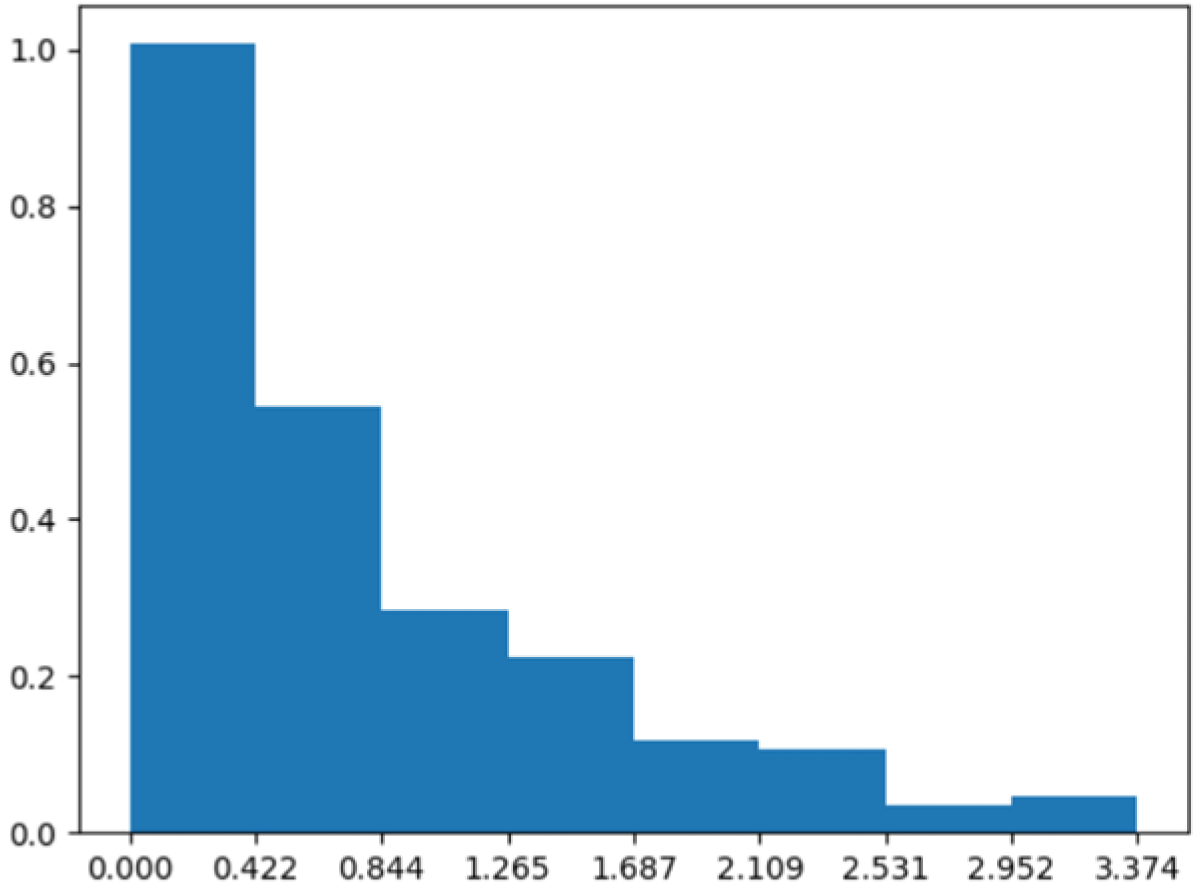

Рисунок 3. Образец гистограммы относительных частот.

Площадь i-ого столбца гистограммы равна w_i, а высота $\frac{w_i}{h}$.

Выборочное среднее:

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

Выборочная дисперсия с поправкой Шеппарда:

$$s_B^2 = \sum_{i=1}^{m} (x_i^* - \overline{x})^2 * w_i - \frac{h^2}{12}, h = \frac{a_m - a_0}{m}$$

Выборочное среднее квадратическое отклонение:

$$\tilde{\sigma} = \sqrt{s_B^2}$$

Выборочная мода:

Если модальный интервал, на котором высота гистограммы максимальна, один, то $\overline{M}_0 = a_k + h\frac{w_{k+1} - w_k}{2w_{k+1} - w_k - w_{k+2}}$, где a_k - левая граница модального интервала

(a_k, a_{k+1}); a_{k+1} - правая граница модального интервала (a_k, a_{k+1}); w_{k+1} - относительная частота на модальном интервале; w_k, w_{k+2} - относительные частоты интервалов слева и справа от модального интервала.

Если модальных интервалов несколько, и все они идут подряд (т. е. интервалы (a_k, a_{k+1}), ..., (a_{k+l-1}, a_{k+l}) - все модальные), то

$$\overline{M}_0 = a_k + l * h * \frac{w_{k+1} - w_k}{2w_{k+1} - w_k - w_{k+2}}$$

Если между модальными интервалами находятся немодальные, то считаем, что выборочной моды не существует.

Выборочная медиана:

$$\overline{M}_e = a_{k-1} + \frac{h}{w_k}\left(\frac{1}{2} - \sum_{i=1}^{k-1} w_i\right), \text{если} \sum_{i=1}^{k-1} w_i < \frac{1}{2} < \sum_{i=1}^{k} w_i$$

$$\overline{M}_e = a_k, \text{если} \sum_{\mathrm{i}=1}^{\mathrm{k}} \mathrm{w_i} = \frac{1}{2}$$

Выборочный момент *k*-ого порядка:

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

Выборочный центральный момент *k*-ого порядка:

$$\overline{\mu}_k^0 = \sum_{i=1}^{m} (x_i^* - \overline{x})^k * w_i, \overline{\mu}_1^0 = 0, \overline{\mu}_2^0 = D_B = \overline{\mu}_2 - (\overline{\mu}_1)^2$$

Выборочный коэффициент асимметрии:

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

Выборочный коэффициент эксцесса:

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

Показательное распределение: $f(x) = \lambda e^{-\lambda x}, x \in [0, +\infty)$	
Математическое ожидание	λ^{-1}
Дисперсия	λ^{-2}
Среднее квадратическое отклонение	λ^{-1}
Мода	0
Медиана	$\lambda^{-1} ln2$
Коэффициент асимметрии	2
Коэффициент эксцесса	6

Результаты расчётов

$V = 165$

$N = 200$

$n = 5 + 165 \% 20 = 10$

$p = 0.2 + 0.003 * 165 = 0.695$

$\lambda = 1 + (-1)^{165} * (165 * 0.003) = 0.505$

Задание 1:

7	7	9	9	9	8	6	8	8	8
6	6	5	5	5	7	8	5	8	6
8	8	7	6	9	9	8	8	7	8
9	6	7	9	7	6	7	7	6	4
6	7	9	6	9	8	8	6	7	6
6	8	7	7	8	7	7	7	5	7
7	8	7	6	9	8	6	7	8	7
9	8	8	6	4	9	4	7	7	4
10	6	9	6	8	8	9	7	7	8
8	6	8	8	9	5	9	8	9	8
7	7	7	6	7	8	8	5	5	6
6	7	7	9	9	9	4	7	5	6
7	5	7	6	7	6	6	6	7	5
8	6	7	9	9	7	5	6	7	8
2	3	5	8	6	7	7	8	5	7
7	2	7	7	8	7	7	8	5	8
6	6	7	10	6	7	6	8	7	6
4	6	7	9	7	7	6	10	7	7
9	8	8	4	8	6	7	6	6	8
10	5	7	7	10	8	4	8	3	5

Таблица 1: 200 выборок биномиального распределения

Биномиальное распределение

X	n_k	w_k	s_k
2	2.00000	0.01000	0.01000
3	2.00000	0.01000	0.02000
4	8.00000	0.04000	0.06000
5	17.00000	0.08500	0.14500
6	40.00000	0.20000	0.34500
7	58.00000	0.29000	0.63500
8	44.00000	0.22000	0.85500
9	24.00000	0.12000	0.97500
10	5.00000	0.02500	1.00000
Total	200	1	0

Таблица 2: Статический ряд

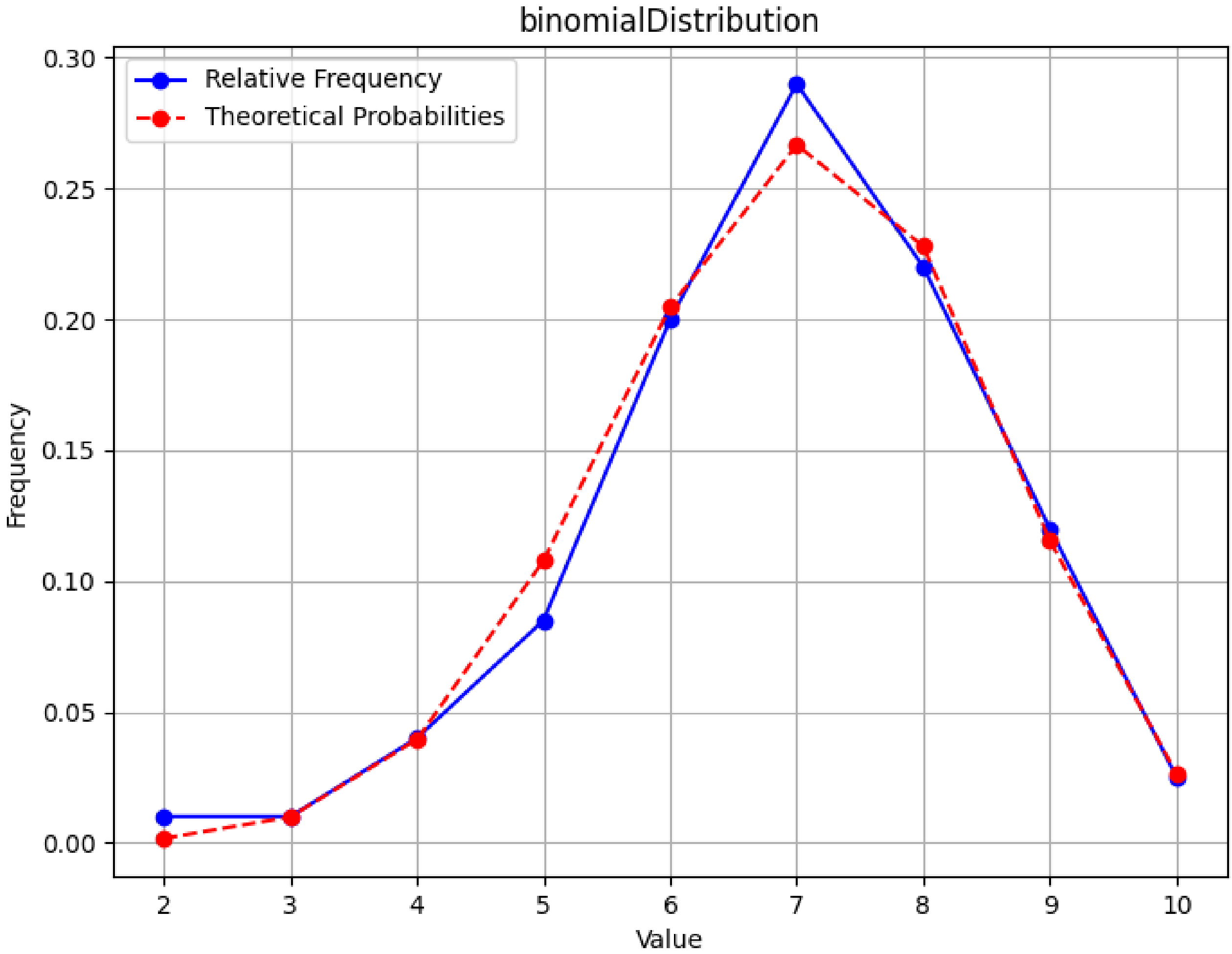

Рисунок 1: Полигон относительных частот

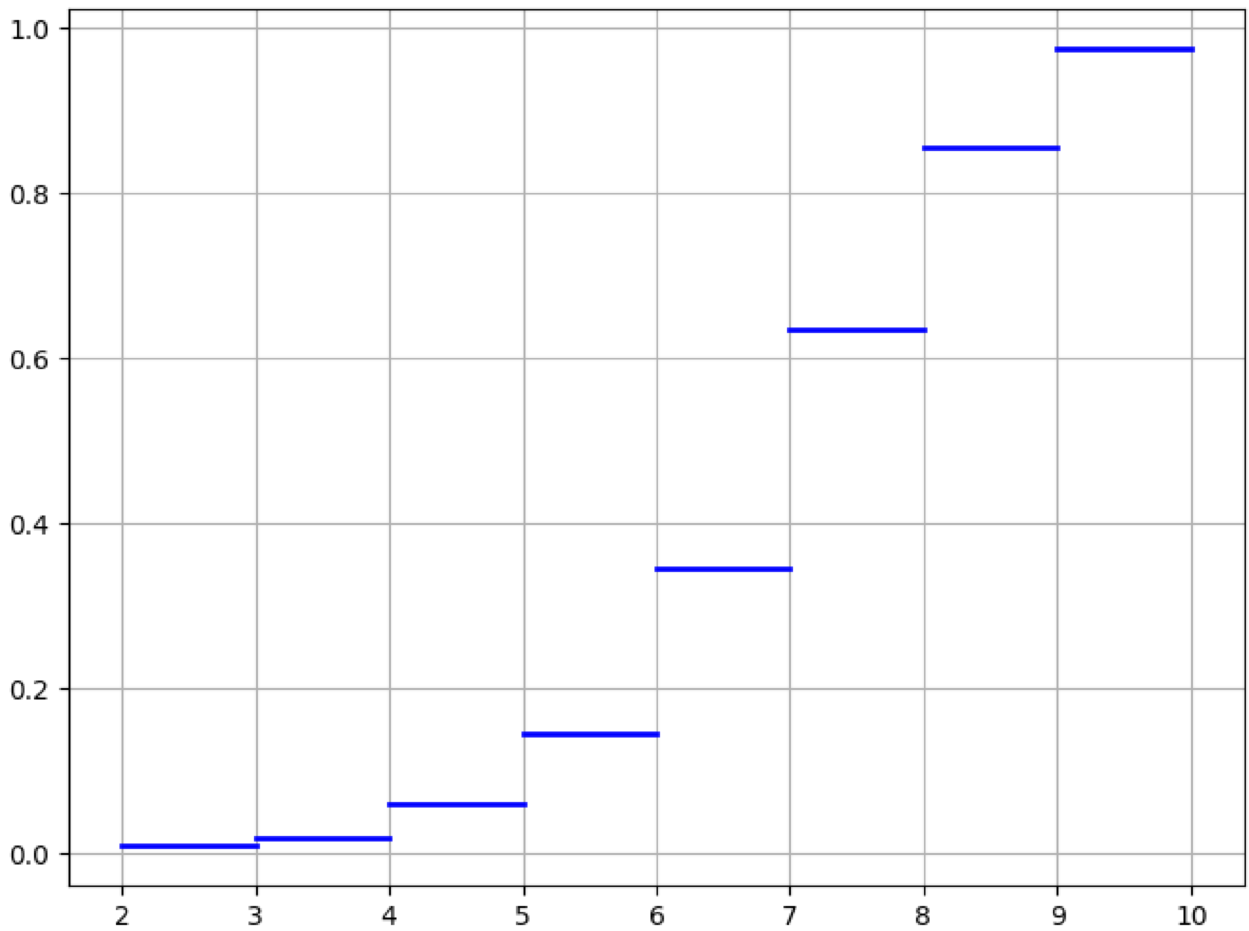

Рисунок 2: График эмпирической функции распеределении

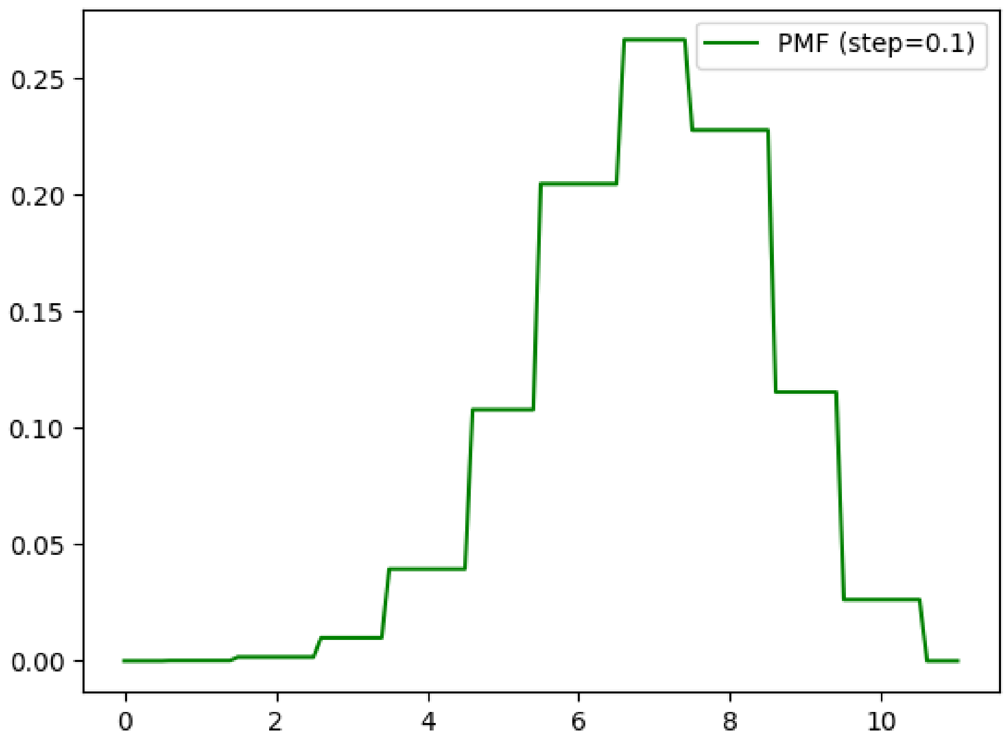

Рисунок 3: График плотности распределения

x_k^*	w_k	$\bar{p}_k$	$\lvert w_k - \bar{p}_k \rvert$
2	0.01000	0.00163	0.00837
3	0.01000	0.00989	0.00011
4	0.04000	0.03944	0.00056
5	0.08500	0.10785	0.02285
6	0.20000	0.20480	0.00480
7	0.29000	0.26667	0.02333
8	0.22000	0.22787	0.00787
9	0.12000	0.11539	0.00461
10	0.02500	0.02629	0.00129

Таблица 3: Таблица сравнения относительных частот и теоретических вероятностей

Название показателя	**Выборочное значение**	**Теоретическое значение**	**Абсолютное отклонение**	**Относительное отклонение**
Среднее значение	6.95500	6.95000	0.00500	0.00072
Дисперсия	2.23298	2.11975	0.11323	0.05341
Среднее квадратичное отклонение	1.49431	1.45594	0.03838	0.02636
Мода	7.00000	7.00000	0.00000	0.00000
Медиана	6.50000	6.00000	0.50000	0.08333
Коэффициент асимметрии	-0.50751	-0.26787	0.23965	-0.89464
Коэффициент эксцесса	0.49828	-0.12825	0.62653	-4.88534

Таблица 4: Таблица сранения рассчитанных характеристик с теоретическими значениями

Задание 2:

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

Таблица 5: 200 выборок геометрического распределения

Геометрическое распределение

x_k^*	n_k	w_k	s_k
1	139.00000	0.69500	0.69500
2	40.00000	0.20000	0.89500
3	15.00000	0.07500	0.97000
4	2.00000	0.01000	0.98000
5	2.00000	0.01000	0.99000
6	2.00000	0.01000	1.00000
Total	200	1	

Таблица 6: Статистический ряд

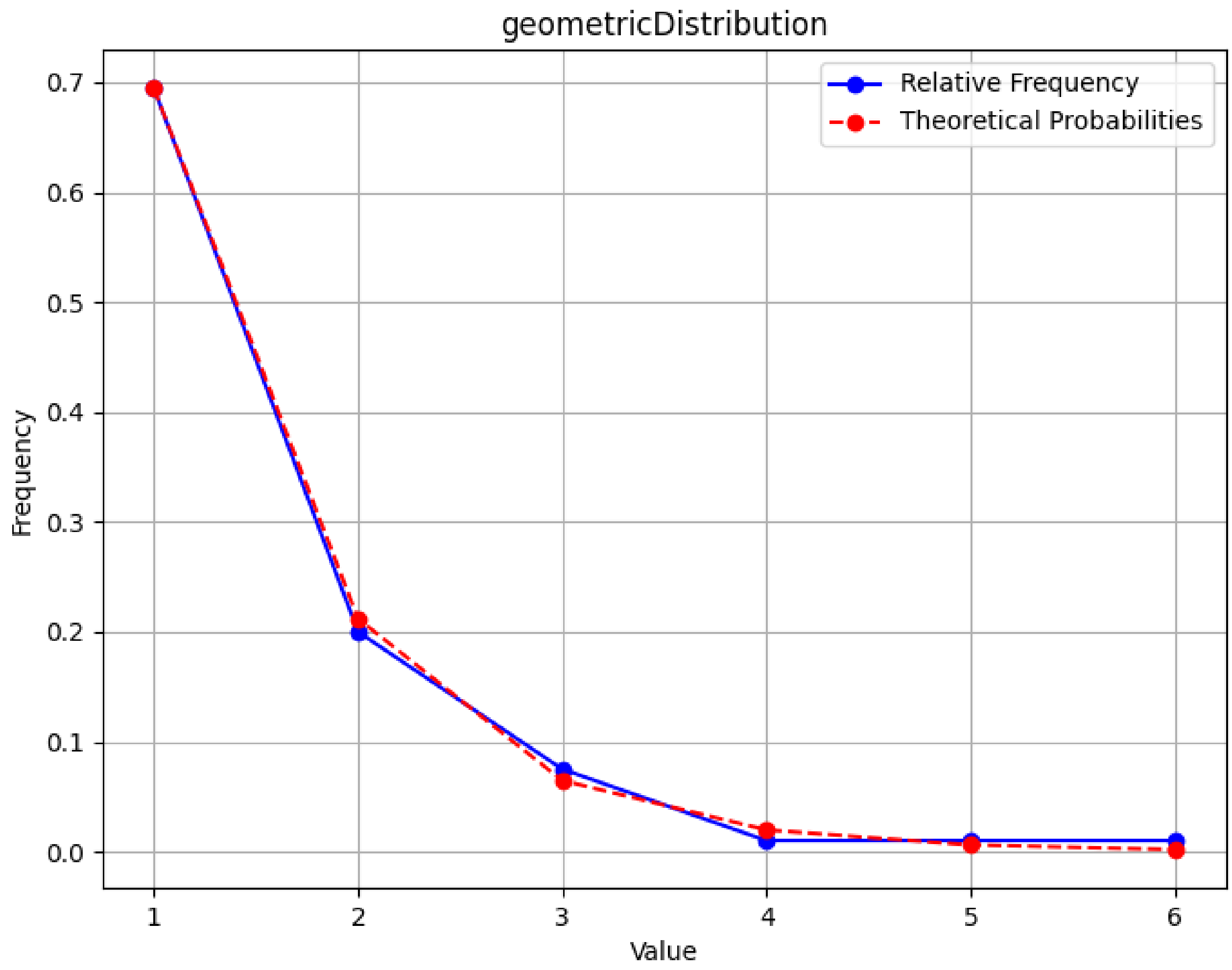

Рисунок 4: Полигон относительных частот

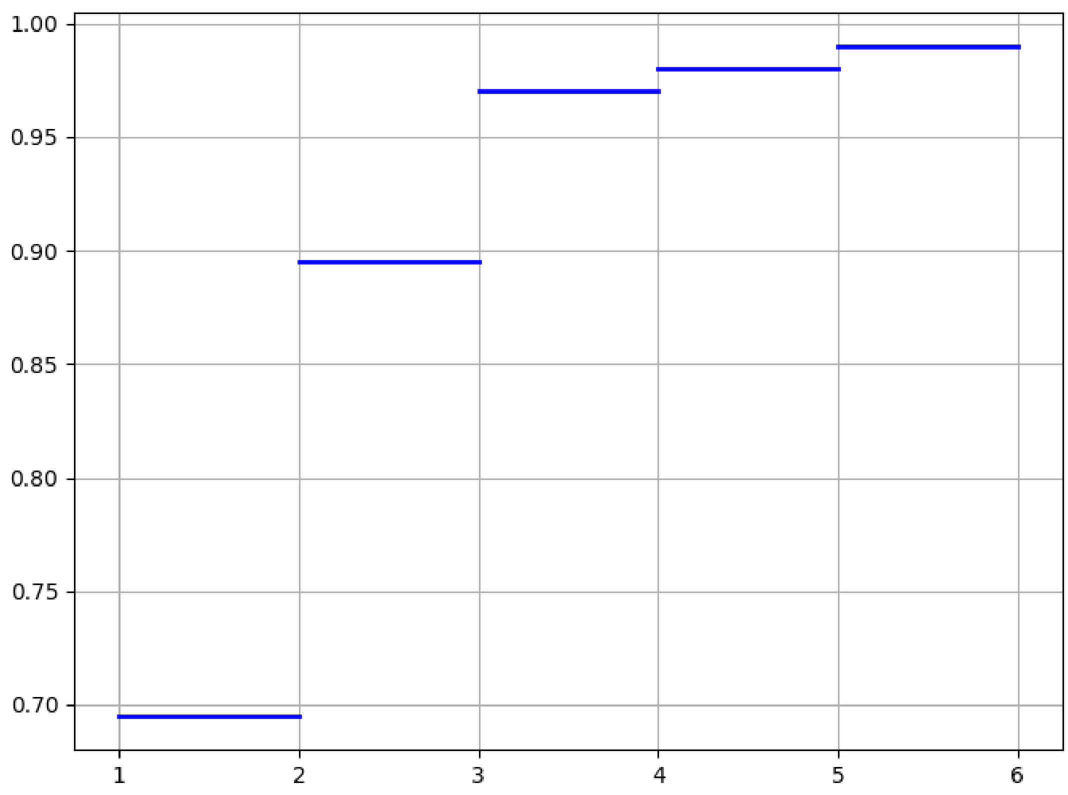

Рисунок 5: График эмпирической функции распределения

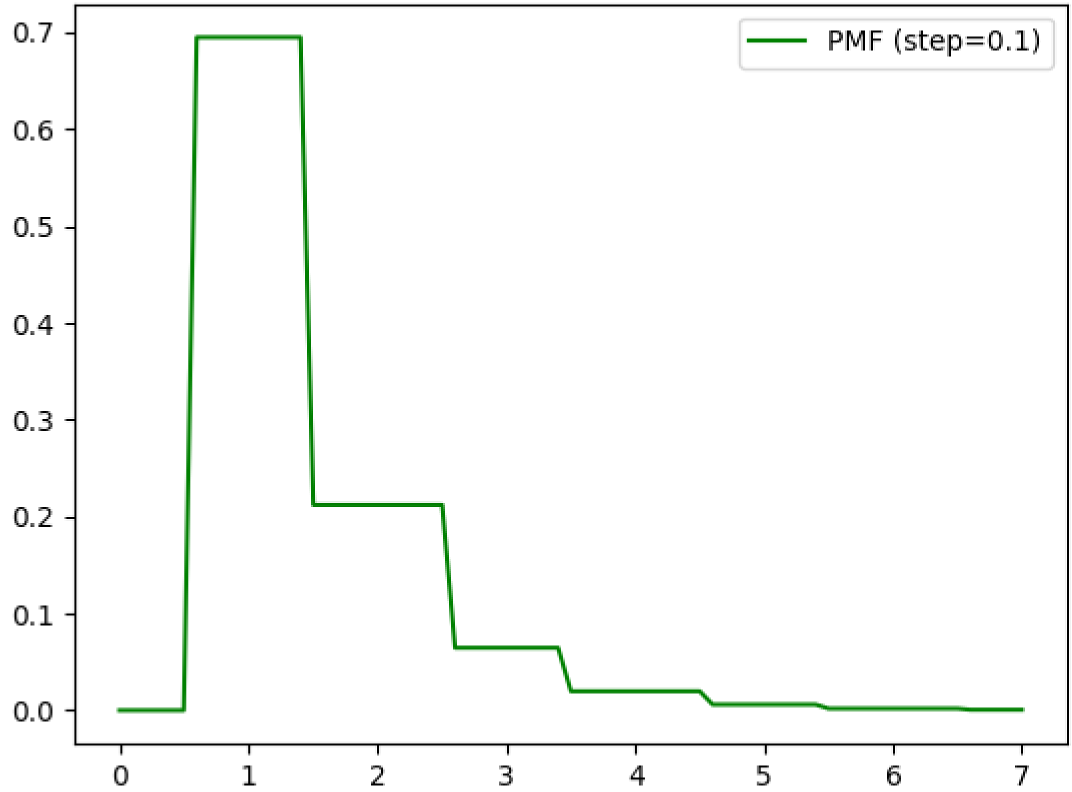

Рисунок 6: График плотности распределения (с шагом 0.1)

x_k^*	w_k	$\overline{p}_k$	$\lvert w_k - \overline{p}_k \rvert$
1	0.69500	0.69500	0.00000
2	0.20000	0.21197	0.01197
3	0.07500	0.06465	0.01035
4	0.01000	0.01972	0.00972
5	0.01000	0.00601	0.00399
6	0.01000	0.00183	0.00817
Total	1.00000	1.00000	0.01197

Таблица 7: Таблица сравнения относительных частот и теоретических вероятностей

Название показателя	**Выборочное значение**	**Теоретическое значение**	**Абсолютное отклонение**	**Относительное отклонение**
Среднее значение	1.47000	1.43885	0.03115	0.02165
Дисперсия	0.77910	0.63144	0.14766	0.23385
Среднее квадратичное отклонение	0.88267	0.79463	0.08804	0.11079
Мода	1.00000	1.00000	0.00000	0.00000

Медиана	3.50000	1.00000	2.50000	2.50000
Коэффициент асимметрии	2.55589	2.36298	0.19290	0.08164
Коэффициент эксцесса	7.92956	7.58369	0.34587	0.04561

Таблица 8: Таблица сравнения рассчитанных характеристик с теоретическими значениями

Задание 3:

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

Таблица 9: 200 выборок распределения Пуассона

Распределение Пуассона

Интервалы	n_k	w_k
(0.00000, 0.34707)	130.00000	0.65000
(0.34707, 0.69413)	0.00000	0.00000
(0.69413, 1.04120)	56.00000	0.28000
(1.04120, 1.38827)	0.00000	0.00000
(1.38827, 1.73534)	0.00000	0.00000
(1.73534, 2.08240)	11.00000	0.05500
(2.08240, 2.42947)	0.00000	0.00000
(2.42947, 3.00000)	0.00000	0.00000
	200	1

Таблица 10: Интервальный Ряд

x_k^*	n_k	w_k
0.17353	130.00000	0.65000
0.52060	0.00000	0.00000
0.86767	56.00000	0.28000
1.21474	0.00000	0.00000
1.56180	0.00000	0.00000
1.90887	11.00000	0.05500
2.25594	0.00000	0.00000
2.71474	0.00000	0.00000
	200	1

Таблица 11: Ассоциированный статистический ряд

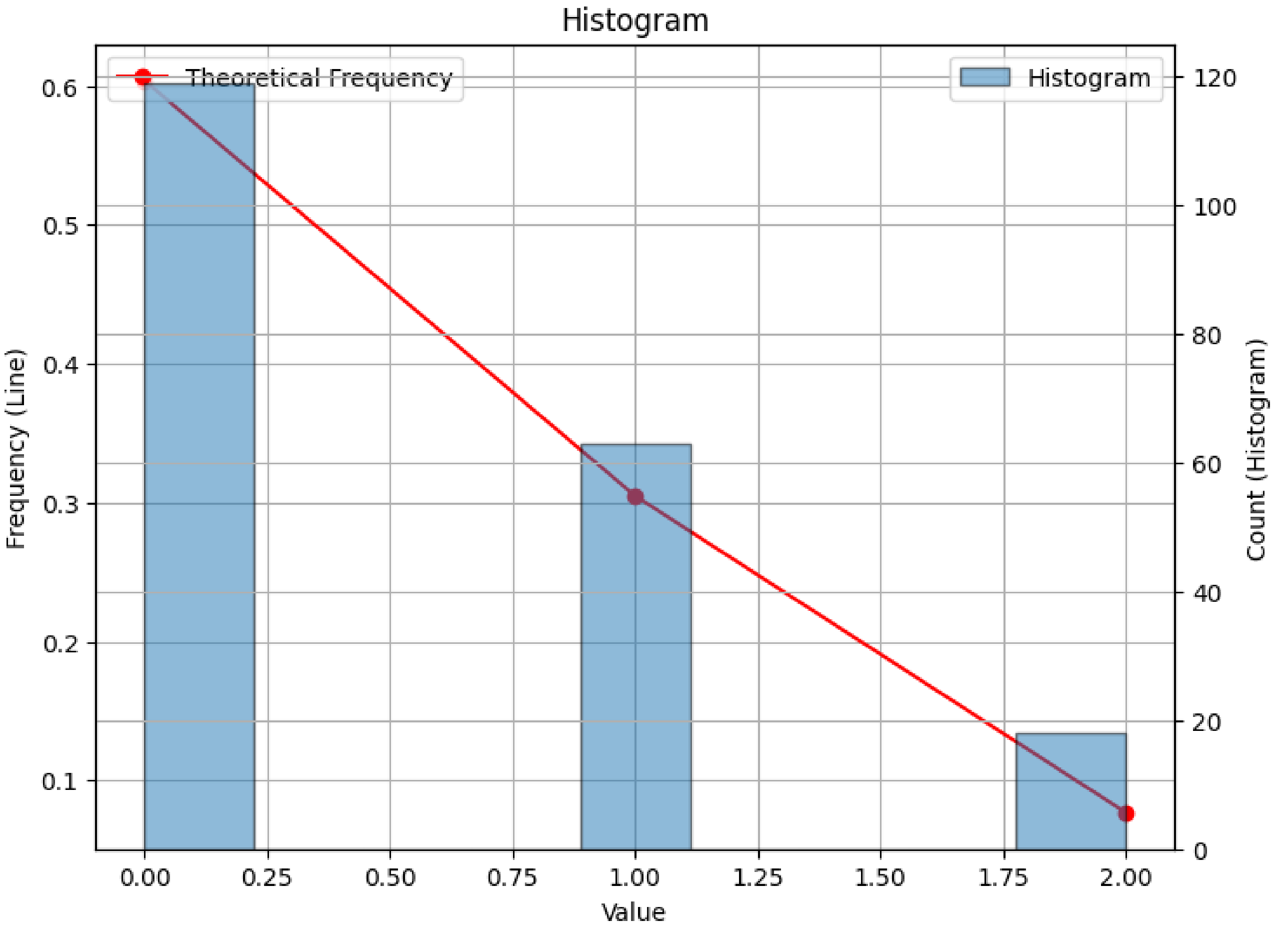

Рисунок 7: Гистограмма относительных частот с наложенным на нее и выделенным красным цветом графиком плотности показательного распределения

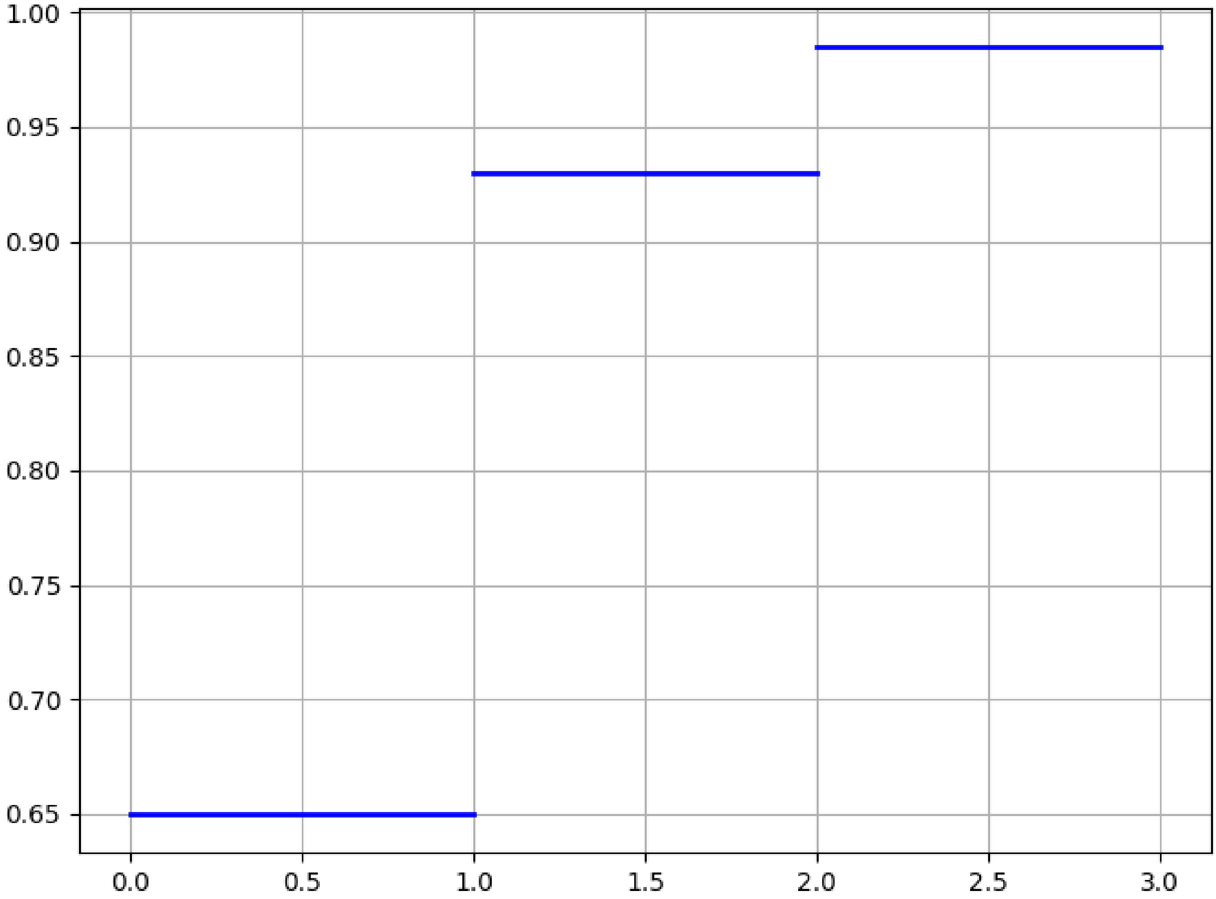

Рисунок 8: График эмфирической функции распределения

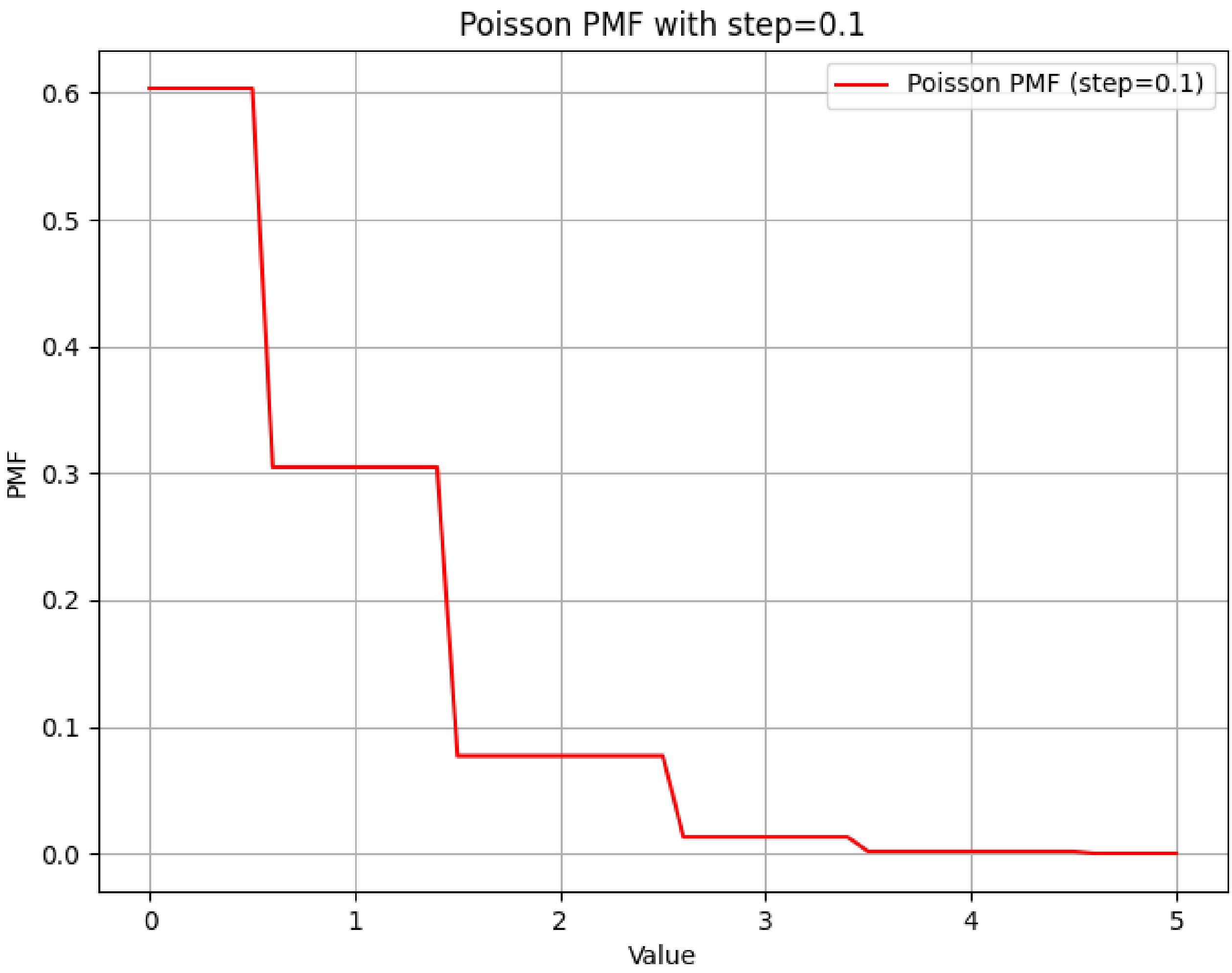

Рисунок 9: График плотности распределения(с шагом 0.1)

Интервалы:	w_k	$\overline{p}_k$	$\lvert w_k - \overline{p}_k \rvert$
(0.00000, 0.34707)	0.65000	0.60351	0.04649
(0.34707, 0.69413)	0.00000	0.00000	0.00000
(0.69413, 1.04120)	0.28000	0.30477	0.02477
(1.04120, 1.38827)	0.00000	0.00000	0.00000
(1.38827, 1.73534)	0.00000	0.00000	0.00000
(1.73534, 2.08240)	0.05500	0.07695	0.02195
(2.08240, 2.42947)	0.00000	0.00000	0.00000
(2.42947, 3.00000)	0.00000	0.00000	0.00000
	1	1	0.046494

Таблица 12: Таблица сравнения относительных частот и теоретических вероятностей

Название показателя	**Выборочное значение**	**Теоретическое значение**	**Абсолютное отклонение**	**Относительное отклонение**
Среднее значение	0.43500	0.50500	0.07000	0.13861
Дисперсия	0.44577	0.50500	0.05923	0.11728
Среднее квадратичное отклонение	0.66766	0.71063	0.04297	0.06047
Мода	0.00000	0.00000	0.00000	nan
Медиана	1.50000	0.00000	1.50000	nan
Коэффициент асимметрии	1.54875	1.40720	0.14155	0.10059
Коэффициент эксцесса	2.18847	1.98020	0.20827	0.10518

Таблица 13: Таблица сравнения рассчитанных характеристик с теоретическими значениями

Список 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.-704с.
3. Гмурман В. Е. Теория вероятностей и математическая статистика.-М.: Юрайт, 2013.-479с.
4. Гмурман В. Е. Руководство к решению задач по теории вероятностей и математической статистике. - М.:Юрайт,2013.-404с.
5. Емельянов Г.В. Скитович В.П. Задачник по теории вероятностей и математической статистике.-СПб.: Лань, 2007.-336с.
6. Кибзун А.И., Горяинова Е.Р., Наумов А.В. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачами. Учебное пособие - М.: ФИЗМАТЛИТ, 2005.-232с.
7. Кобзарь А.И. Прикладная математическая статистика: для инженеров и научных работников - М.: ФИЗМАТЛИТ, 2006.-816с.
8. Монсик В.Б., Скрынников А.А. Вероятность и статистика. - М.: БИНОМ, 2015-384с.
9. Сборник задач по теории вероятностей, математической статистике и теории случайных функций: Учеб. пособие для вузов / Под ред. А. А. Свешникова. - СПб.: Лань, 2012. - 472с.
10. Письменный Д.Т. Конспект лекций по теории вероятностей, математической статистике и случайным процессам: учеб. пособие для вузов.- М.: Айрис-пресс,2013.-288с.

Приложение

```
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
import pandas as pd
import math

N = 200 #Sample
V = 165 #Variant

n = 5 + V % 20
p = 0.2 + 0.003 * V
lamda = 1 + ((-1) ** V) * (V * 0.003)

seed = 10 + V
rng = np.random.default_rng(seed=seed)

def task(X_distribution, name_distribution):
  #1
  counts = np.unique(X_distribution, return_counts=True)
  X = counts[0]
  freq = counts[1]
  rel_freq = counts[1] / N
  cum_freq = np.cumsum(rel_freq)

  total_freq = freq.sum(axis=0)
  total_rel_freq = rel_freq.sum(axis=0)
  intervals = None

  if name_distribution == 'binomial' or name_distribution ==
'geometric' or name_distribution == 'poisson':
    total_row = {
      'X': 'Total',
      'Frequency': total_freq,
      'Relative Frequency': total_rel_freq,
      'Cumulative Frequency': ''
    }

    df_freq = pd.DataFrame({
        'X': counts[0],
        'Frequency': [f"{val:.5f}" for val in freq],
        'Relative Frequency': [f"{val:.5f}" for val in
rel_freq],
        'Cumulative Frequency': [f"{val:.5f}" for val in
cum_freq]
        })
```

```
    df_freq.loc[len(df_freq)] = total_row

    print(df_freq)
    df_freq.to_excel(name_distribution + '_stat.xlsx')

  if name_distribution == 'poisson':
        step = 1 + np.log2(N)
        distance_between_step = (X.max() - X.min()) / step
        intervals = np.array([X.min() + i *
distance_between_step for i in range(int(step))])
        intervals = np.append(intervals, X.max())
        intervals_freq = np.zeros(len(intervals)-1)
        intervals_relative_freq = np.zeros(len(intervals)-1)
        for i in range(len(intervals)-1):
            for j in range(len(X)):
                if intervals[i] <= X[j] < intervals[i+1]:
                    intervals_freq[i] += freq[j]
                    intervals_relative_freq[i] += rel_freq[j]

        total_row = {'Frequency': total_freq, 'Relative
Frequency': total_rel_freq}
        df_freq_intervals = pd.DataFrame({'Interval: ':
zip([f"{val:.5f}" for val in intervals[:-1]], [f"{val:.5f}"
for val in intervals[1:]]), 'Frequency': [f"{val:.5f}" for val
in intervals_freq], 'Relative Frequency': [f"{val:.5f}" for
val in intervals_relative_freq]})
        df_freq_intervals.loc[len(df_freq_intervals)] =
total_row
        print(df_freq_intervals)
        df_freq_intervals.to_excel(name_distribution +
'_intervals.xlsx')

        middle = np.array([(intervals[i] + intervals[i+1]) / 2
for i in range(len(intervals)-1)])
        df_freq_middle = pd.DataFrame({'Middle': [f"{val:.5f}"
for val in middle], 'Frequency': [f"{val:.5f}" for val in
intervals_freq], 'Relative Frequency': [f"{val:.5f}" for val
in intervals_relative_freq]})
        df_freq_middle.loc[len(df_freq_intervals)] = total_row
        print(df_freq_middle)
        df_freq_middle.to_excel(name_distribution +
'_middle.xlsx')

  #2
  if name_distribution == 'binomial':
```

```
    theo_freq = np.array([(math.comb(n, k) * (p ** k) * ((1-p)
** (n-k))) for k in X])
  elif name_distribution == 'geometric':
    theo_freq = np.array([(p * ((1-p) ** (k-1))) for k in
counts[0]])
  elif name_distribution == 'poisson':
    theo_freq = np.array([(lamda ** k) * np.exp(-lamda) /
math.factorial(k) for k in counts[0]])
    intervals_theoretical_relative_freq =
np.zeros(len(intervals)-1)
    for i in range(len(intervals)-1):
        for j in range(len(X)):
            if intervals[i] <= X[j] < intervals[i+1]:
                intervals_theoretical_relative_freq[i] +=
theo_freq[j]

  #plot
  if name_distribution == 'binomial' or name_distribution ==
'geometric':
    plt.figure(figsize = (8,6))
    plt.plot(X, rel_freq, marker = 'o', linestyle = '-', color
= 'b', label = 'Relative Frequency')
    plt.plot(X, theo_freq, marker = 'o', linestyle = '--',
color = 'red', label = 'Theoretical Probabilities')
    plt.xlabel("Value")
    plt.ylabel("Frequency")
    plt.title(name_distribution + "Distribution")
    plt.legend()
    plt.grid(True)
    plt.show()
    #3
    plt.figure(figsize = (8,6))
    for i in range(0, len(counts[0])-1):
      plt.plot([counts[0][i], counts[0][i+1]], [cum_freq[i],
cum_freq[i]], color='b', linestyle='-', linewidth=2)
    plt.grid(True)
    plt.show()

    x_min = 0
    x_max = X.max() + 1
    x_vals = np.arange(x_min, x_max + 0.1, 0.1)
    if name_distribution == 'binomial':
        pmf_vals = []
        for xv in x_vals:
            k_int = int(round(xv))
            if 0 <= k_int <= n:
```

```
                val = math.comb(n, k_int) * (p**k_int) * ((1 -
p)**(n - k_int))
            else:
                val = 0
            pmf_vals.append(val)

    else:  # geometric
        pmf_vals = []
        for xv in x_vals:
            k_int = int(round(xv))
            if k_int >= 1:
                val = p * ((1 - p)**(k_int - 1))
            else:
                val = 0
            pmf_vals.append(val)

    plt.plot(x_vals, pmf_vals, 'g-', label='PMF (step=0.1)')
    plt.legend()
    plt.show()

  if name_distribution == 'poisson':
    fig, ax1 = plt.subplots(figsize=(8, 6))

    ax1.plot(counts[0], theo_freq, marker='o', linestyle='-',
color='r', label='Theoretical Frequency')
    ax1.set_xlabel('Value')
    ax1.set_ylabel('Frequency (Line)')
    ax1.set_title('Histogram')
    ax1.legend(loc='upper left')
    ax1.grid(True)
    ax2 = ax1.twinx()
    ax2.hist(X_distribution, bins='sturges',
edgecolor='black', alpha=0.5, label='Histogram')
    ax2.set_ylabel('Count (Histogram)')
    ax2.legend(loc='upper right')
    plt.show()

    x_min = 0
    x_max = X.max() + 2
    x_vals = np.arange(x_min, x_max + 0.1, 0.1)

    pmf_vals = []
    for xv in x_vals:
        k_int = int(round(xv))
        if k_int >= 0:
```

```
            val = (lamda**k_int) * np.exp(-lamda) /
math.factorial(k_int)
        else:
            val = 0
        pmf_vals.append(val)

    plt.figure(figsize=(8, 6))
    plt.plot(x_vals, pmf_vals, 'r-', label='Poisson PMF
(step=0.1)')
    plt.title("Poisson PMF with step=0.1")
    plt.xlabel("Value")
    plt.ylabel("PMF")
    plt.grid(True)
    plt.legend()
    plt.show()

    plt.figure(figsize = (8,6))
    for i in range(0, len(counts[0])-1):
      plt.plot([counts[0][i], counts[0][i+1]], [cum_freq[i],
cum_freq[i]], color='b', linestyle='-', linewidth=2)
    plt.grid(True)
    plt.show()

  #4
  mean = np.array([X[i] * rel_freq[i] for i in
range(len(X))]).sum()
  print("Sample mean = ", f"{mean: .5f}")

  #5
  variance = np.array([((X[i] - mean) ** 2) * rel_freq[i] for
i in range(len(X))]).sum()
  print("Sample variance = ", f"{variance: .5f}")

  #6
  deviation = np.sqrt(variance)
  print("Sample standard deviation = ", f"{deviation: .5f}")

  #7
  modes = X[np.argwhere(freq ==
np.amax(freq))].flatten().tolist()
  mode = (modes[0] + modes[len(modes) - 1]) / 2
  print("Mode = ", f"{mode: .5f}")

  #8
  if len(X) % 2 == 0:
      median = (X[int(len(X)/2)] + X[int(len(X)/2) - 1]) / 2
```

```
  else:
      if 0.5 in X.tolist():
          median = X[cum_freq.tolist().index(0.5)]
      else:
          left = -1
          for element in cum_freq:
              if element < 0.5:
                  left += 1
              else:
                  right = left + 1
                  break
          median = (X[left] + X[right]) / 2
  print("Median = ", f"{median: .5f}")

  #9
  def sample_k_moment_around_mean(k, mean):
      return np.array([(X[i] - mean) ** k * rel_freq[i] for i
in range(len(X))]).sum()
  sample_skeness = sample_k_moment_around_mean(3, mean) /
deviation ** 3
  print("Sample skewness = ", f"{sample_skeness: .5f}")
  sample_kurtosis = sample_k_moment_around_mean(4, mean) /
deviation ** 4 - 3
  print("Sample kurtosis = ", f"{sample_kurtosis: .5f}")

  abs_diff_freq = np.abs(theo_freq - rel_freq)
  freq_compare = pd.DataFrame({
      'X': X,
      'Relative Frequency': [f"{val:.5f}" for val in
rel_freq],
      'Theoretical Frequency': [f"{val:.5f}" for val in
theo_freq],
      'Absolute Difference': [f"{val:.5f}" for val in
abs_diff_freq]
      })

  if name_distribution == 'poisson':
    abs_diff_freq_intervals =
np.abs(intervals_theoretical_relative_freq -
intervals_relative_freq)
    freq_compare_intervals = pd.DataFrame({
        'Interval: ': zip([f"{val:.5f}" for val in
intervals[:-1]], [f"{val:.5f}" for val in intervals[1:]]),
        'Relative Frequency': [f"{val:.5f}" for val in
intervals_relative_freq],
```

```
        'Theoretical Frequency': [f"{val:.5f}" for val in
intervals_theoretical_relative_freq],
        'Absolute Difference': [f"{val:.5f}" for val in
abs_diff_freq_intervals]
        })
    total_row_compare_intervals = {
        'Relative Frequency': total_rel_freq,
        'Theoretical Frequency': 1,
        'Absolute Difference': np.max(abs_diff_freq)
    }
    freq_compare_intervals.loc[len(freq_compare_intervals)] =
total_row_compare_intervals
    freq_compare_intervals.to_excel(name_distribution +
'_intervals_compare.xlsx')

  total_row_compare = {
      'X': 'Total',
      'Relative Frequency': f"{total_rel_freq: .5f}",
      'Theoretical Frequency': f"{1: .5f}",
      'Absolute Difference': f"{np.max(abs_diff_freq): .5f}"
      }
  freq_compare.loc[len(freq_compare)] = total_row_compare
  print(freq_compare)
  freq_compare.to_excel(name_distribution + '_compare.xlsx')

  #10
  if name_distribution == 'binomial':
    theo_mean = n * p
    theo_var = n * p * (1 - p)
    theo_deviation = np.sqrt(n * p * (1 - p))
    theo_skewness = ((1-p)-p) / np.sqrt(n * p * (1 - p))
    theo_kurtois = (1 - 6 * p * (1 - p)) / (n * p * (1 - p))
    theo_mode = np.floor((n + 1) * p)
    theo_median = np.floor(n * p)
    theo_values = np.array([theo_mean, theo_var,
theo_deviation, theo_skewness, theo_kurtois, theo_mode,
theo_median])
    real_values = np.array([mean, variance, deviation,
sample_skeness, sample_kurtosis, mode, median])
    abs_differences = np.array([abs(mean - theo_mean),
abs(variance - theo_var), abs(deviation - theo_deviation),
abs(sample_skeness - theo_skewness), abs(sample_kurtosis -
theo_kurtois), abs(mode - theo_mode), abs(median -
theo_median)])
    rel_differences = np.array([abs_differences[i] /
theo_values[i] for i in range(len(theo_values))])
```

```
    char_combine = pd.DataFrame({'Characteristic': ['Mean',
'Variance', 'Deviation', 'Skewness', 'Kurtosis', 'Mode',
'Median'],
                                'Sample': [f"{val:.5f}" for
val in real_values],
                                'Theoretical': [f"{val:.5f}"
for val in theo_values],
                                'Absolute Difference':
[f"{val:.5f}" for val in abs_differences],
                                'Relative Difference':
[f"{val:.5f}" for val in rel_differences]
                                })
    char_combine.to_excel(name_distribution + '_char.xlsx')
  elif name_distribution == 'geometric':
    theo_mean = 1 / p
    theo_var = (1 - p) / (p ** 2)
    theo_deviation = np.sqrt((1 - p) / (p ** 2))
    theo_skewness = (2 - p) / np.sqrt(1 - p)
    theo_kurtois = 6 + p ** 2 / (1 - p)
    theo_mode = 1
    theo_median = np.round((-1 / np.log2(1 - p)))
    theo_values = np.array([theo_mean, theo_var,
theo_deviation, theo_skewness, theo_kurtois, theo_mode,
theo_median])
    real_values = np.array([mean, variance, deviation,
sample_skeness, sample_kurtosis, mode, median])
    abs_differences = np.array([abs(real_values[i] -
theo_values[i]) for i in range(len(theo_values))])
    rel_differences = np.array([abs_differences[i] /
theo_values[i] for i in range(len(theo_values))])
    char_combine = pd.DataFrame({'Characteristic': ['Mean',
'Variance', 'Deviation', 'Skewness', 'Kurtosis', 'Mode',
'Median'],
                                'Sample': [f"{val:.5f}" for
val in real_values],
                                'Theoretical': [f"{val:.5f}"
for val in theo_values],
                                'Absolute Difference':
[f"{val:.5f}" for val in abs_differences],
                                'Relative Difference':
[f"{val:.5f}" for val in rel_differences]
                                })
    char_combine.to_excel(name_distribution + '_char.xlsx')
  elif name_distribution == 'poisson':
    theo_mean = lamda
    theo_var= lamda
```

```
    theo_deviation = np.sqrt(lamda)
    theo_skewness = 1 / np.sqrt(lamda)
    theo_kurtois = 1 / lamda
    theo_mode = np.floor(lamda)
    theo_median = np.floor(lamda + 1/3 - 0.02/lamda)
    theo_values = np.array([theo_mean, theo_var,
theo_deviation, theo_skewness, theo_kurtois, theo_mode,
theo_median])
    real_values = np.array([mean, variance, deviation,
sample_skeness, sample_kurtosis, mode, median])
    abs_differences = np.array([abs(real_values[i] -
theo_values[i]) for i in range(len(theo_values))])
    rel_differences = []
    for i in range(len(theo_values)):
        if theo_values[i] == 0:
            rel_differences.append(np.nan)  # or "N/A"
        else:
            rel_differences.append(abs_differences[i] /
theo_values[i])
    rel_differences = np.array(rel_differences)
    char_combine = pd.DataFrame({'Characteristic': ['Mean',
'Variance', 'Deviation', 'Skewness', 'Kurtosis', 'Mode',
'Median'],
                                'Sample': [f"{val:.5f}" for
val in real_values],
                                'Theoretical': [f"{val:.5f}"
for val in theo_values],
                                'Absolute Difference':
[f"{val:.5f}" for val in abs_differences],
                                'Relative Difference':
[f"{val:.5f}" for val in rel_differences]
                                })
    char_combine.to_excel(name_distribution + '_char.xlsx')


  print(char_combine)
```