



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.....	3
Краткие теоретические сведения .....	5
Результаты расчётов.....	13
Список литературы .....	23
Приложение .....	24

## Задание

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

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

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

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

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 теоретическими значениями.

## Краткие теоретические сведения

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

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

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

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

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  $(x_1^*, w_1), (x_2^*, w_2), \dots, (x_m^*, w_m)$ .

Эмпирическая функция распределения:

$$F_N^{\exists}(x) = \sum_{x_i^* \leq x} w_i = \begin{cases} 0, & x < x_1^*, \\ w_1, & x_1^* \leq x < x_2^*, \\ w_1 + w_2, & x_2^* \leq x < x_3^*, \\ w_1 + w_2 + w_2, & x_3^* \leq x < x_4^*, \\ \dots\dots\dots, & \\ 1, & x \geq x_m^* \end{cases}$$

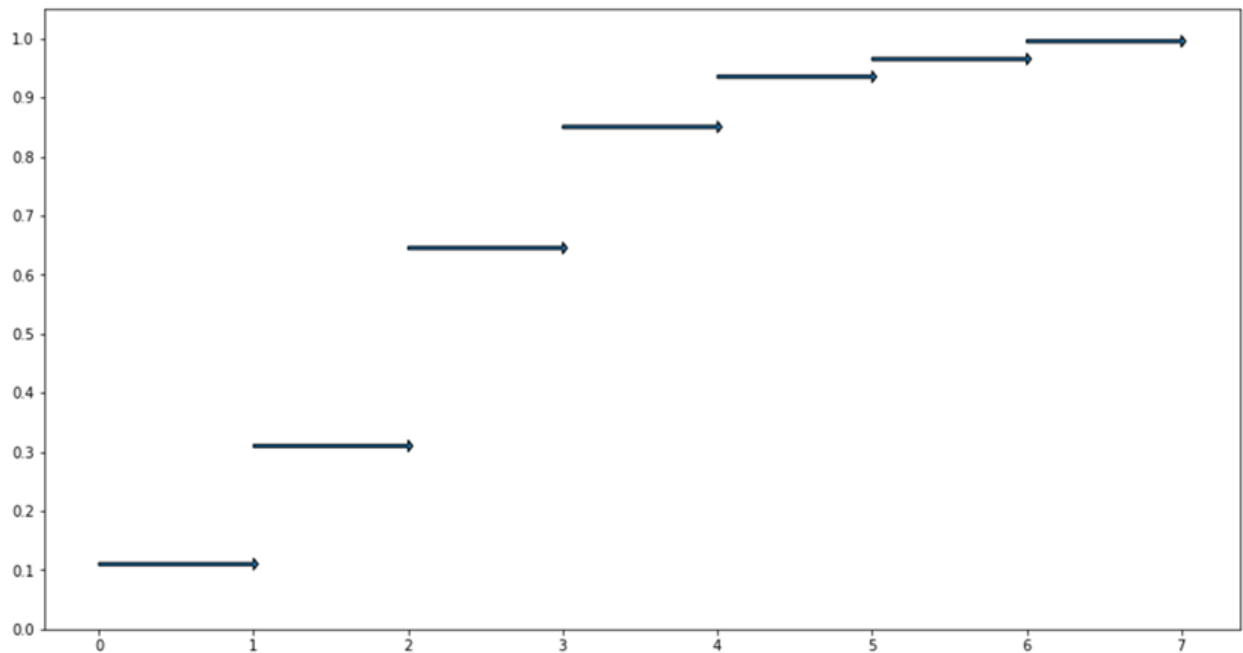


Рисунок 1. Образец графика эмпирической функции распределения.

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):

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

Выборочная дисперсия:

$$D_B = \sum_{i=1}^m (x_i^* - \bar{x})^2 * w_i = \bar{\mu}_2 - (\bar{\mu}_1)^2$$

Выборочная центральный момент k-ого порядка:

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

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

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

Выборочное среднее квадратическое отклонение:

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

Выборочный коэффициент асимметрии:

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

Выборочный коэффициент эксцесса:

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

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

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

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

Выборочная медиана:

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

<b>Биномиальное распределение</b>	
Вероятность	$p_k = C_n^k * p^k * q^{n-k}, k = 0, 1, \dots, n, q = 1 - p$
Математическое ожидание	$np$
Дисперсия	$npq$
Среднее квадратическое отклонение	$\sqrt{npq}$
Мода	$[(n+1)p]$ , если $(n+1)p$ - дробное; $[(n+1)p - \frac{1}{2}]$ , 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	$p_k = q^k * p, k = 0, 1, \dots, q = 1 - p$
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  $[a_0, a_1], (a_1, a_2], \dots, (a_{m-1}, a_m]$ ; 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:

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

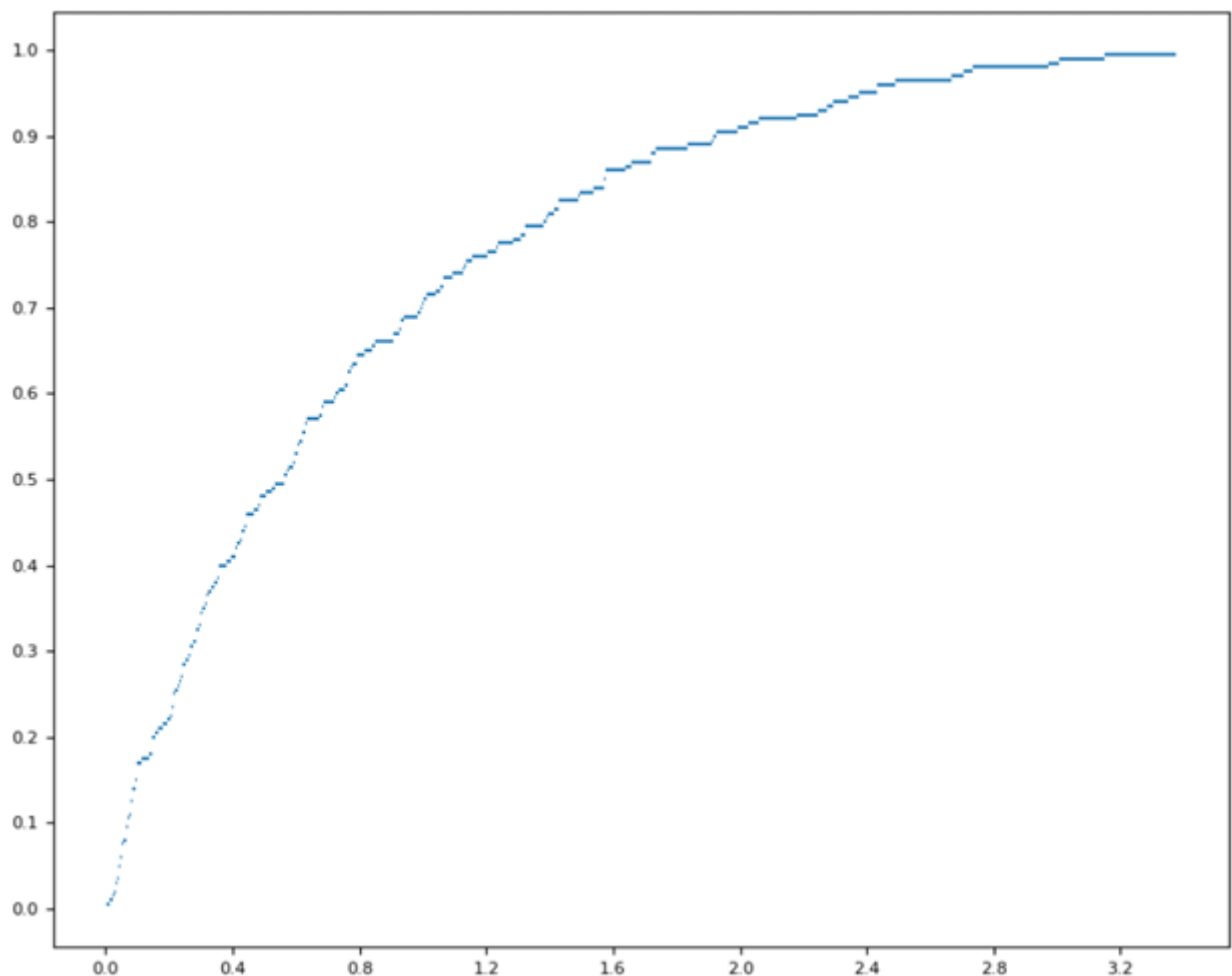


Рисунок 2. Образец графика эмпирической функции распределения.

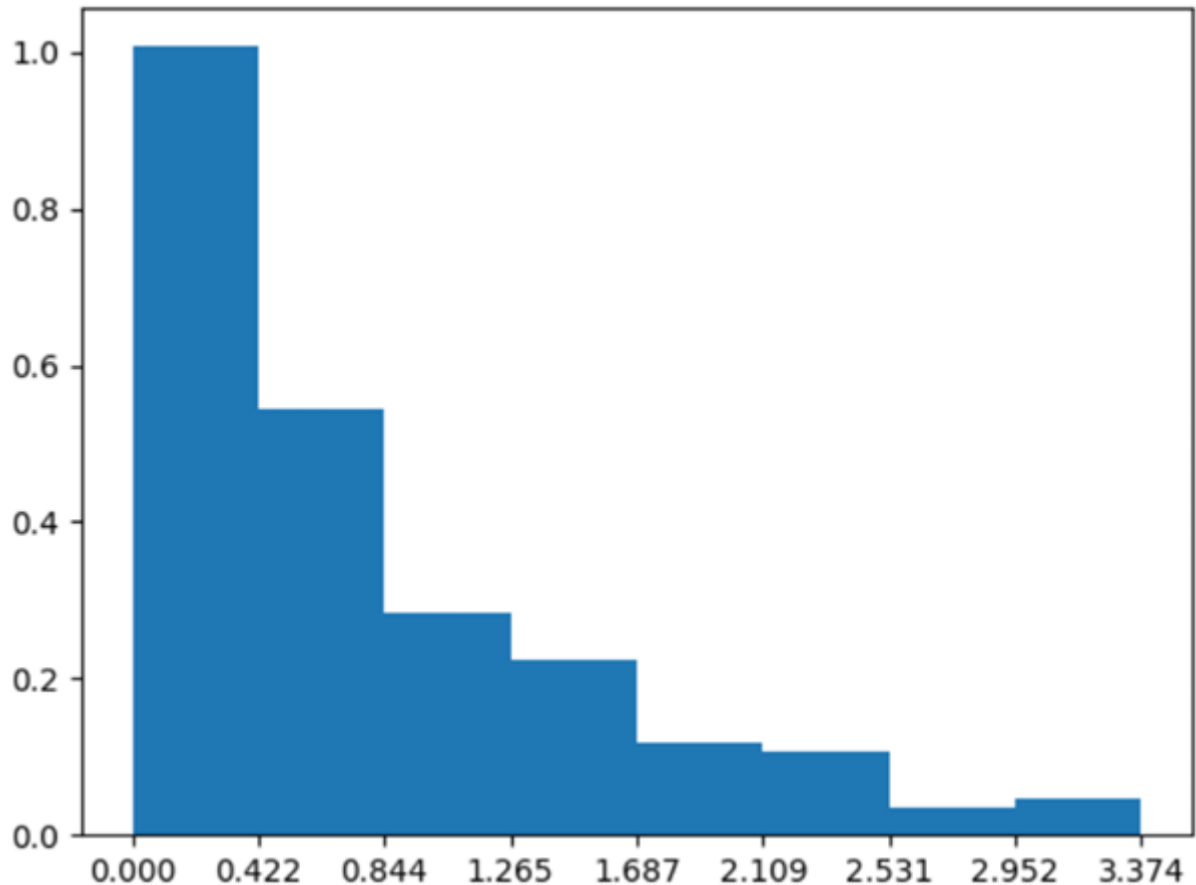


Рисунок 3. Образец гистограммы относительных частот.

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

Выборочное среднее:

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

Выборочная дисперсия с поправкой Шеппарда:

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

Выборочное среднее квадратическое отклонение:

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

Выборочная мода:

Если модальный интервал, на котором высота гистограммы максимальна, один, то  $\bar{M}_0 = a_k + h \frac{w_{k+1} - w_k}{2w_{k+1} - w_k - w_{k+2}}$ , 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  $(a_k, a_{k+1}), \dots, (a_{k+l-1}, a_{k+l})$  - 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:

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

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

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	$\lambda^{-1}$
Дисперсия	$\lambda^{-2}$
Среднее квадратическое отклонение	$\lambda^{-1}$
Мода	0
Медиана	$\lambda^{-1} \ln 2$
Коэффициент асимметрии	2
Коэффициент эксцесса	6

## Результаты расчётов

### Задание 1: Биномиальное Распределение

Параметры Бинамиольного Распределения:

$$V = 165$$

$$N = 200$$

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

$$p = 0.2 + 0.003 * 165 = 0.695$$

2	2	3	3	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	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	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	8	8	8	8
8	9	9	9	9	9	9	9	9	9
9	9	9	9	9	9	9	9	9	9
9	9	9	9	9	10	10	10	10	10

Таблица 1: 200 выборок биномиального распределения

$X$	$n_k$	$w_k$	$s_k$
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	

Таблица 2: Статический ряд бинамиольного распределения

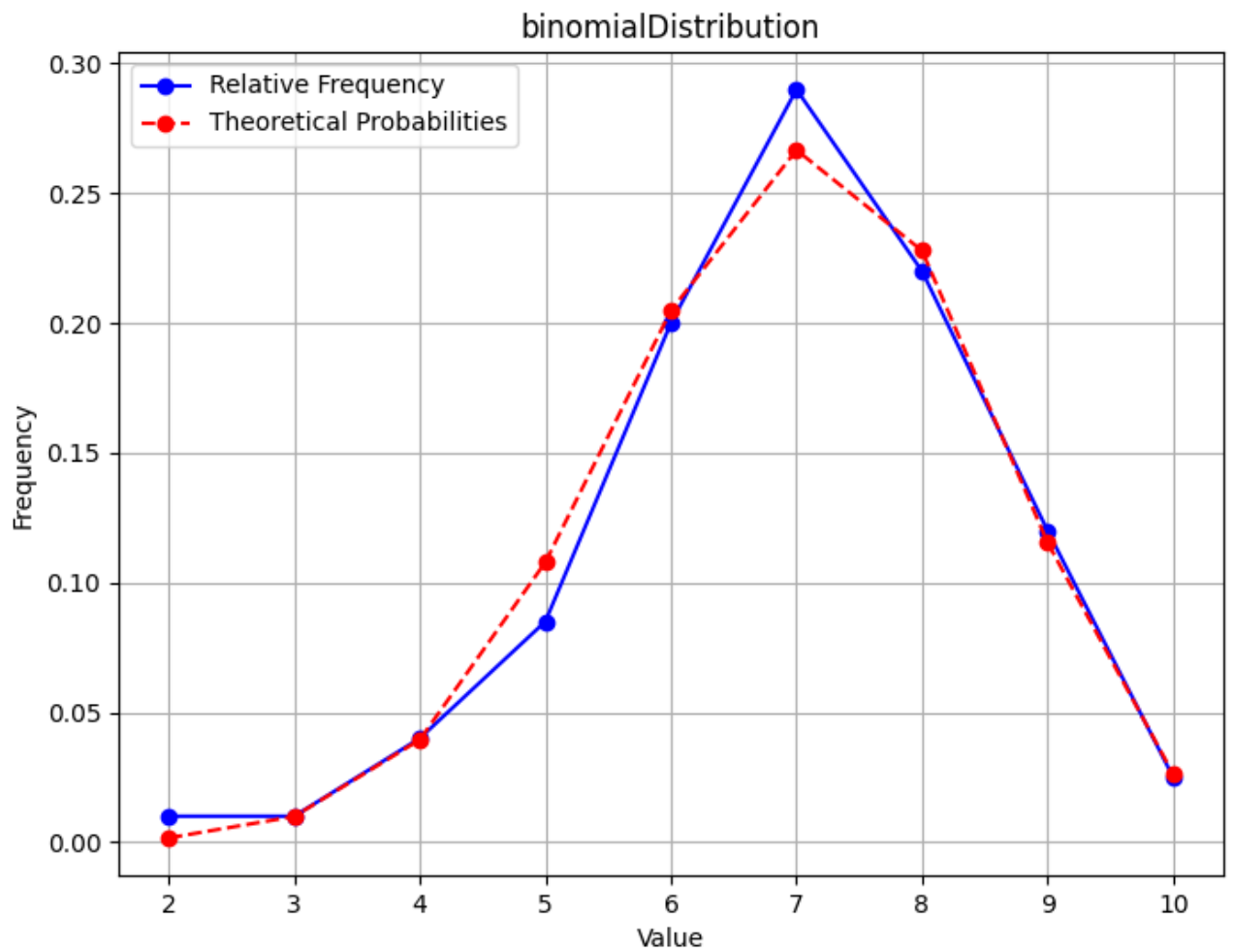


Рисунок 1: Полигон относительных частот

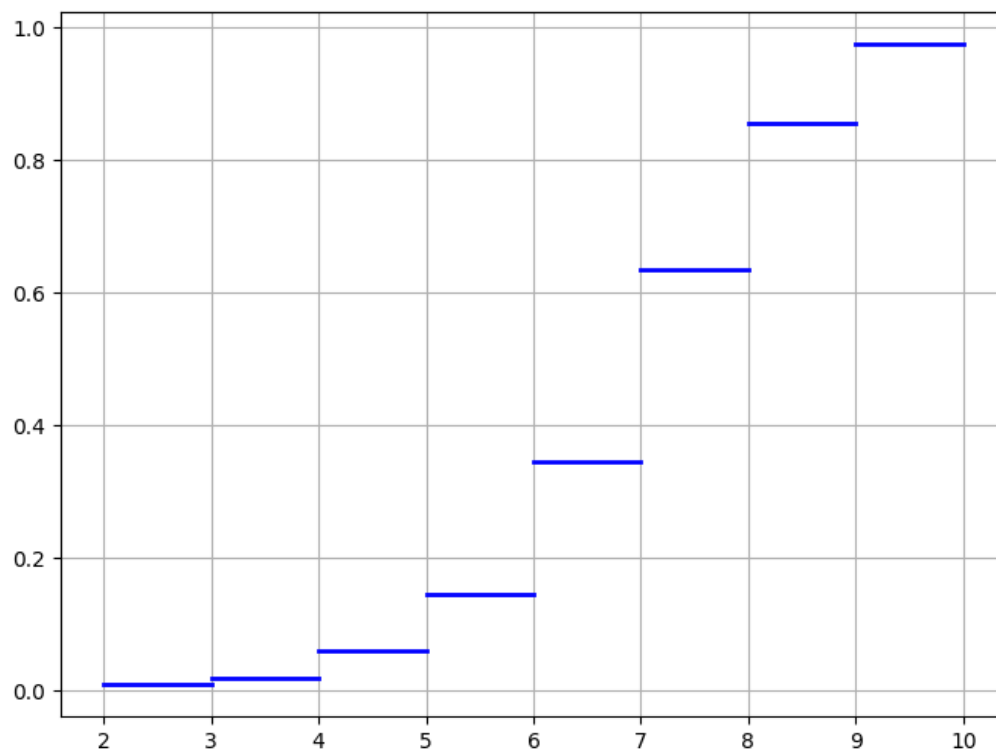


Рисунок 2: График эмпирической функции распределения

$x_k^*$	$w_k$	$\bar{p}_k$	$ w_k - \bar{p}_k $
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
Total	1.00000	1.00000	0.02333

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	0.07194%
Дисперсия	2.23298	2.11975	0.11323	5.34167%
Среднее квадратичное отклонение	1.49431	1.45594	0.03838	2.63610%
Мода	7.00000	7.00000	0.00000	0.00000%
Медиана	7.00000	6.00000	1.00000	16.66667%
Коэффициент асимметрии	-0.50751	-0.26787	0.23965	-89.46504%
Коэффициент эксцесса	0.49828	-0.12825	0.62653	-488.52242%

Таблица 4: Таблица сравнения рассчитанных характеристик с теоретическими значениями

## Задание 2: Геометрическое Распределение

Параметры геометрического распределения:

$$V = 165$$

$$N = 200$$

$$p = 0.2 + 0.003 * 165 = 0.695$$

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	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	2
2	2	2	2	2	2	2	2	2	2
2	2	2	2	3	3	4	4	5	5

Таблица 5: 200 выборок геометрического распределения

$x_k^*$	$n_k$	$w_k$	$s_k$
0	139.00000	0.69500	0.69500
1	40.00000	0.20000	0.89500
2	15.00000	0.07500	0.97000
3	2.00000	0.01000	0.98000
4	2.00000	0.01000	0.99000
5	2.00000	0.01000	1.00000
Total	200	1	

Таблица 6: Статистический ряд геометрического распределения



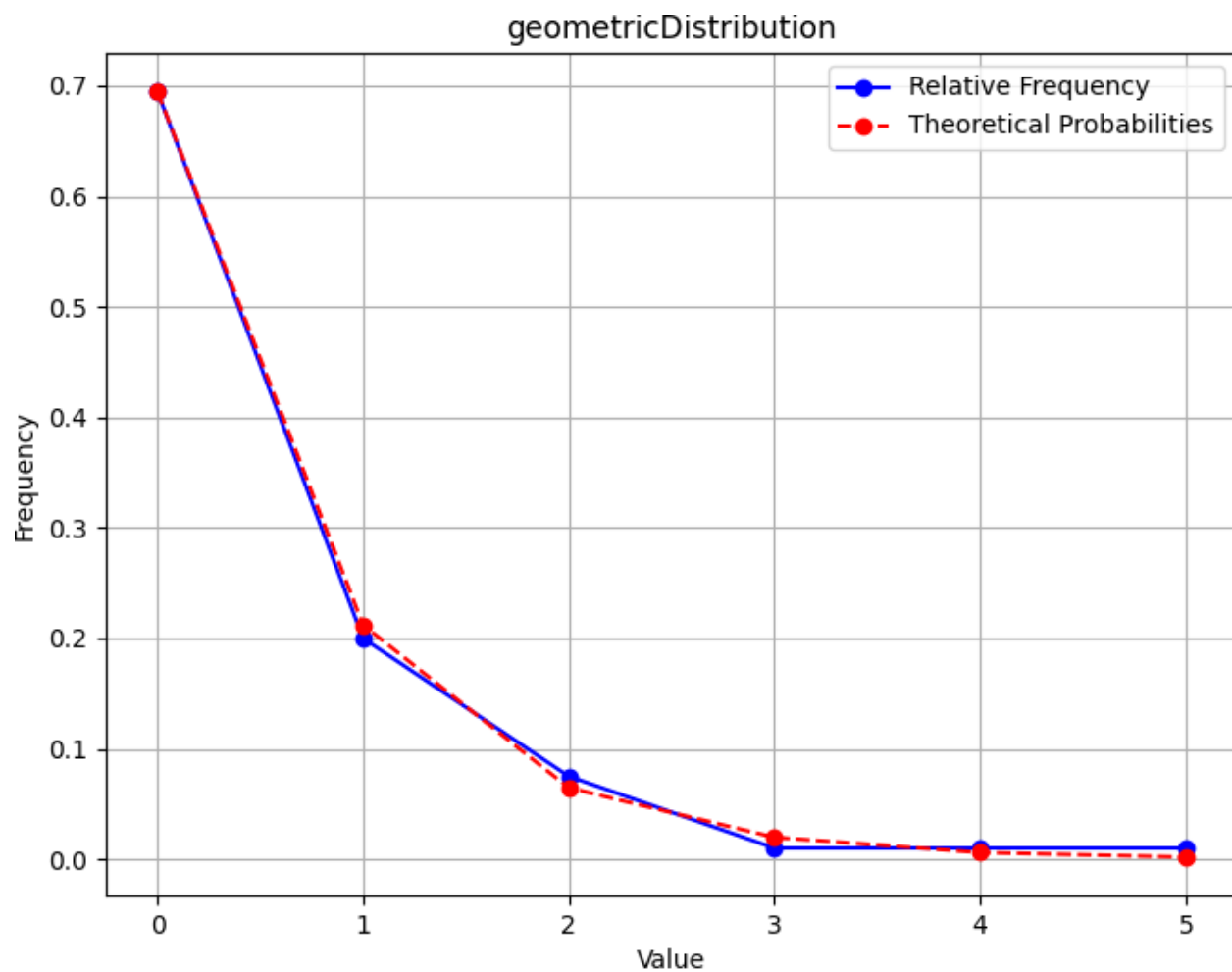


Рисунок 3: Полигон относительных частот

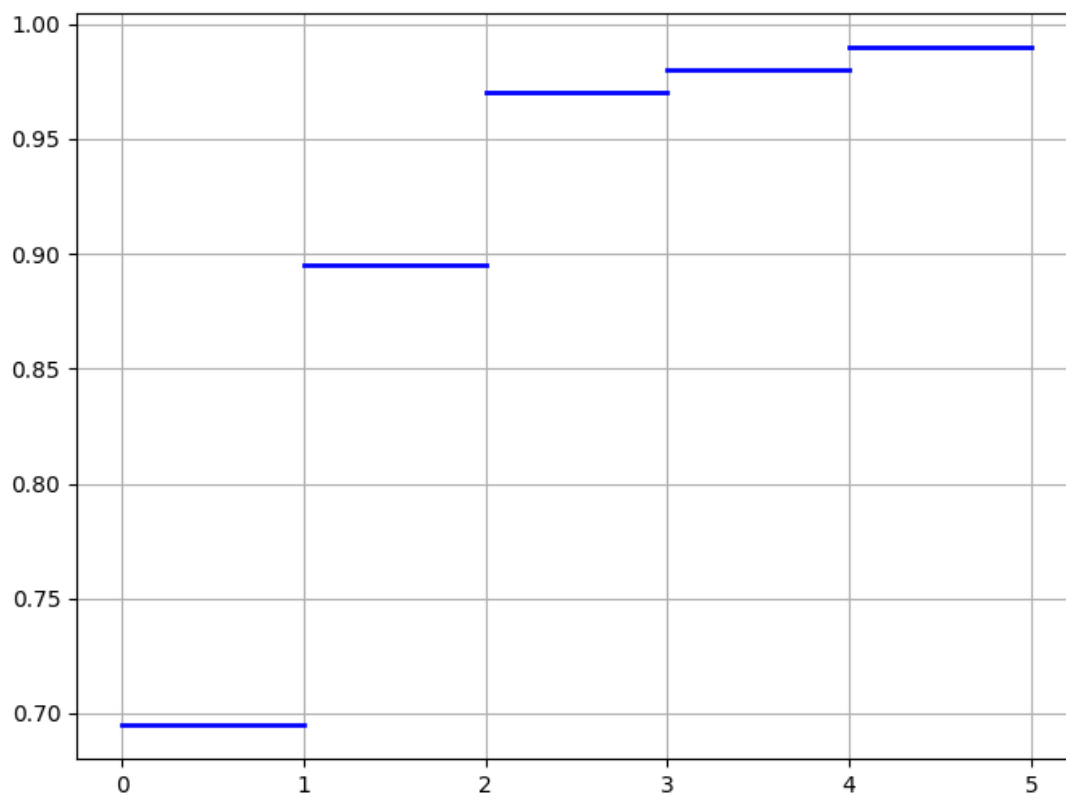


Рисунок 4: График эмпирической функции распределения

$x_k^*$	$w_k$	$\bar{p}_k$	$ w_k - \bar{p}_k $
0	0.69500	0.69500	0.00000
1	0.20000	0.21197	0.01197
2	0.07500	0.06465	0.01035
3	0.01000	0.01972	0.00972
4	0.01000	0.00601	0.00399
5	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	0.47	0.43885	0.03115	7.0981%
Дисперсия	0.7791	0.63144	0.14766	23.38464%
Среднее квадратичное отклонение	0.882666	0.79463	0.08804	11.07937%
Мода	0	0	0	
Медиана	0	0	0	
Коэффициент асимметрии	2.555887	2.36298	0.19291	8.16384%
Коэффициент эксцесса	7.929562	7.58369	0.34587	4.56071%

Таблица 8: Таблица сравнения рассчитанных характеристик с теоретическими значениями

### Задание 3: Показательное Распределение

$$N = 200$$

$$V = 165$$

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

0.00306	0.00371	0.04183	0.0442	0.04783	0.066	0.07368	0.08025	0.08563	0.09244
0.10731	0.10801	0.11337	0.12014	0.12885	0.13818	0.13895	0.15989	0.21774	0.22329
0.22422	0.23575	0.24321	0.24516	0.26399	0.31737	0.34497	0.34604	0.35145	0.3673
0.37426	0.37859	0.41151	0.44657	0.46871	0.48668	0.48888	0.49093	0.49241	0.4998
0.54084	0.56362	0.56835	0.57651	0.58269	0.59036	0.62898	0.65897	0.66393	0.68246
0.68725	0.68919	0.70492	0.71196	0.74379	0.75689	0.75781	0.78059	0.8002	0.80272
0.82656	0.83362	0.84316	0.8641	0.88088	0.89439	0.92116	0.92767	0.94034	0.9498
0.95345	0.95414	0.97444	0.9802	0.98827	0.98992	1.00121	1.00238	1.01547	1.01792
1.0259	1.03556	1.07177	1.09257	1.13872	1.1495	1.20082	1.2015	1.2026	1.2149
1.25272	1.25649	1.25966	1.26166	1.2647	1.29719	1.30317	1.32714	1.34519	1.39762
1.39909	1.40847	1.40987	1.41305	1.41438	1.41876	1.54192	1.55343	1.55429	1.61373
1.64199	1.66582	1.66981	1.68855	1.69111	1.71109	1.72252	1.7414	1.74438	1.7654
1.78518	1.79209	1.79461	1.80649	1.85393	1.86697	1.8713	2.00102	2.02318	2.02787
2.06355	2.08105	2.09078	2.17723	2.22549	2.23326	2.30447	2.31632	2.42838	2.43942
2.45323	2.48679	2.63999	2.64311	2.68339	2.71357	2.72038	2.73557	2.7577	2.76445
2.84827	2.85959	2.86203	2.87557	2.8776	2.88841	2.90409	2.95716	2.96552	2.985
2.98602	3.15051	3.17777	3.37107	3.51239	3.83728	3.97097	3.99558	4.20866	4.24013
4.3667	4.37453	4.3756	4.38595	4.51129	4.51745	4.5287	4.62674	4.67219	4.69204
4.70677	4.84025	4.88461	5.00635	5.15679	5.19494	5.28251	5.50308	5.72359	6.06887
6.432	6.61762	6.90661	7.29179	8.13319	8.20789	8.95765	9.73263	9.7493	10.7548
									1

Таблица 9: 200 выборок показательного распределения

### Показательное Распределение

Интервалы	$n_k$	$w_k$
(0.00306, 1.24692)	90.00000	0.45000
(1.24692, 2.49078)	52.00000	0.26000
(2.49078, 3.73464)	23.00000	0.11500
(3.73464, 4.97850)	18.00000	0.09000
(4.97850, 6.22236)	7.00000	0.03500
(6.22236, 7.46622)	4.00000	0.02000
(7.46622, 8.71008)	2.00000	0.01000
(8.71008, 10.75481)	4.00000	0.02000
	200.00000	1.00000

Таблица 10: Интервальный Ряд

$x_k^*$	$n_k$	$w_k$
0.62499	90	0.45
1.86885	52	0.26
3.11271	23	0.115
4.35657	18	0.09
5.60043	7	0.035
6.84429	4	0.02
8.08815	2	0.01
9.73244	4	0.02
	200.00000	1.00000

Таблица 11: Ассоциированный статистический ряд

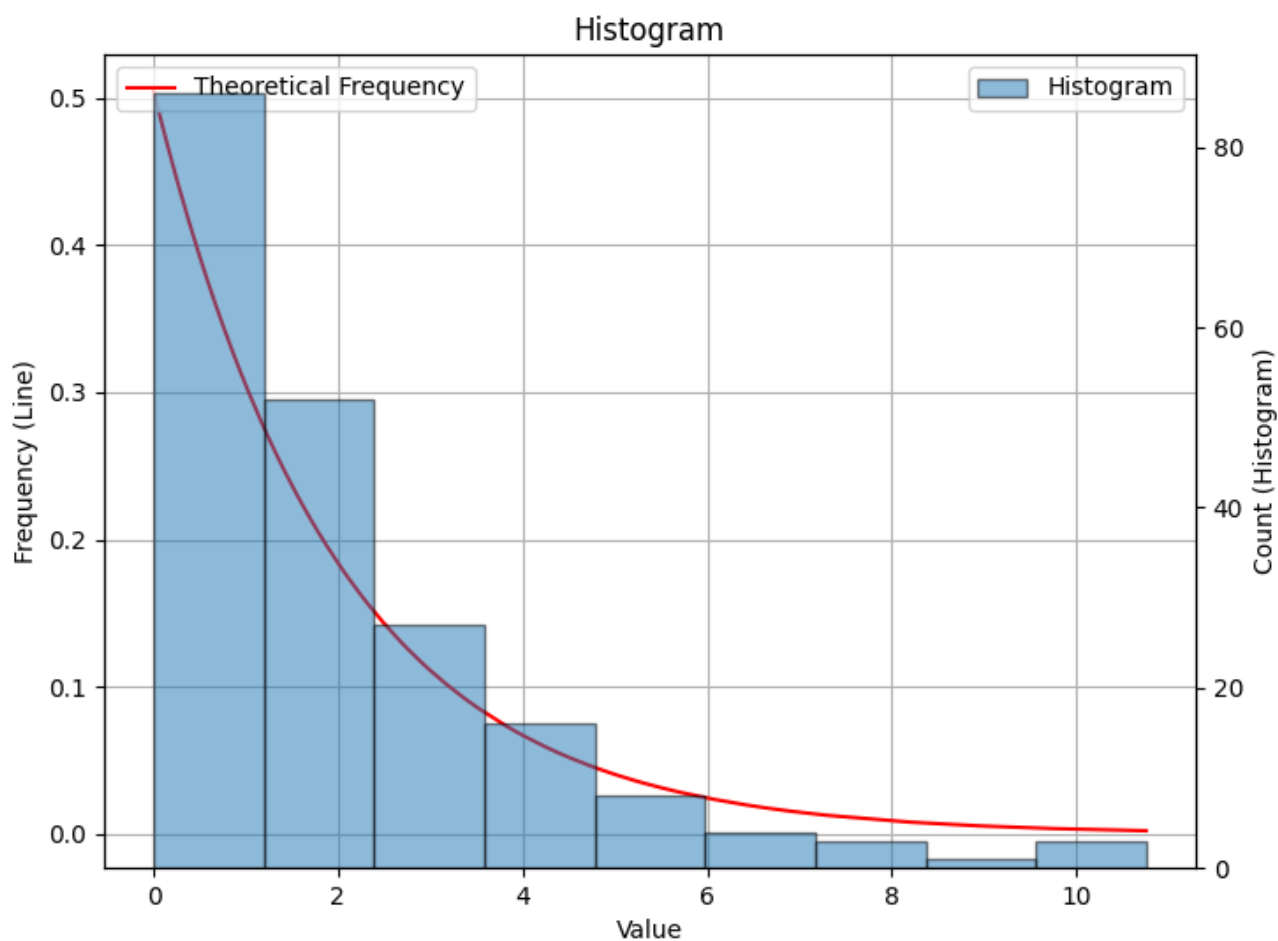


Рисунок 5: Гистограмма относительных частот с наложенным на нее и выделенным красным цветом графиком плотности показательного распределения

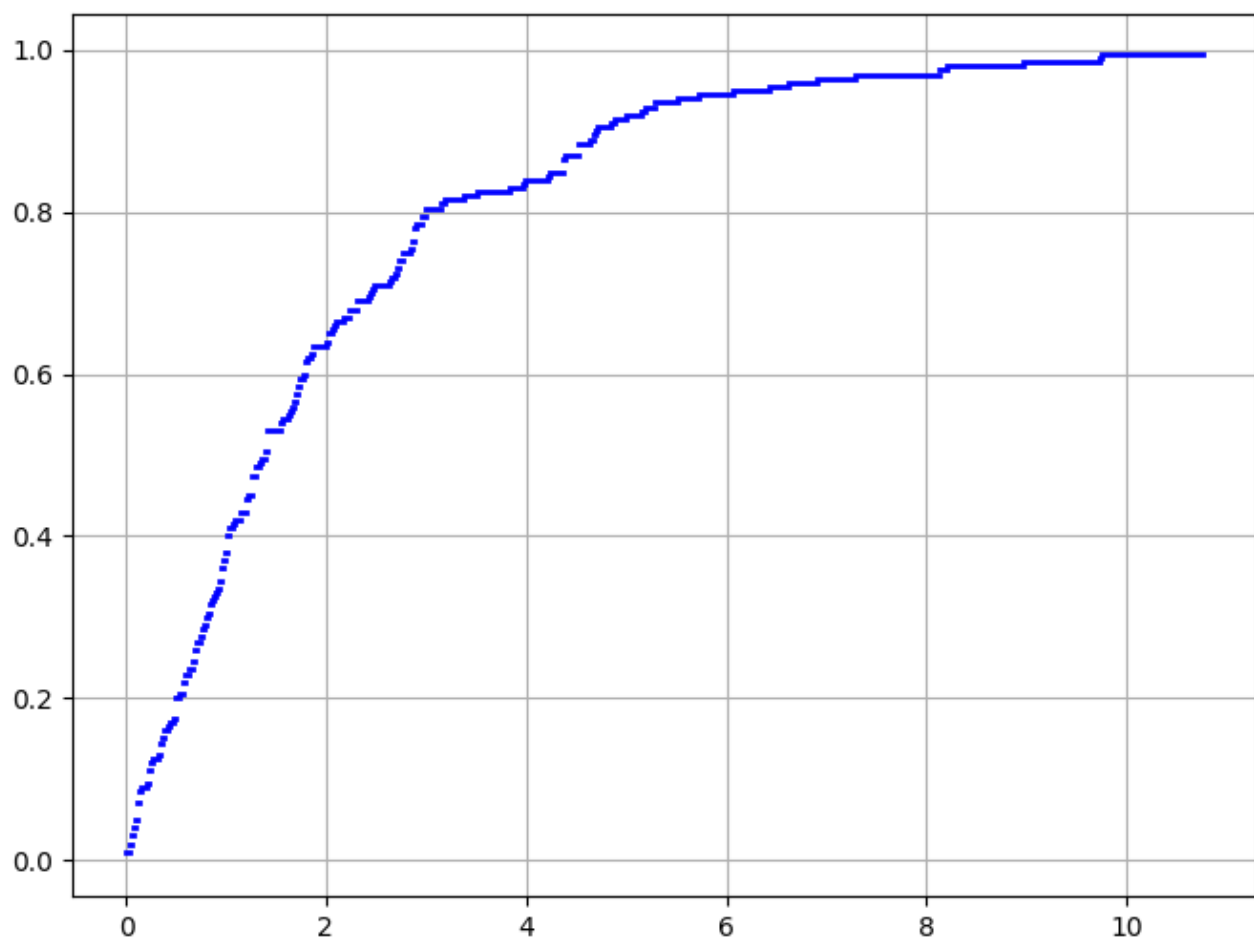


Рисунок 6: График эмпирической функции распределения

Интервалы:	$w_k$	$\bar{p}_k$	$ w_k - \bar{p}_k $
(0.00306, 1.24692)	0.45000	0.46570	0.01570
(1.24692, 2.49078)	0.26000	0.24849	0.01151
(2.49078, 3.73464)	0.11500	0.13259	0.01759
(3.73464, 4.97850)	0.09000	0.07075	0.01925
(4.97850, 6.22236)	0.03500	0.03775	0.00275
(6.22236, 7.46622)	0.02000	0.02014	0.00014
(7.46622, 8.71008)	0.01000	0.01075	0.00075
(8.71008, 10.75481)	0.02000	0.00792	0.01208
	1.00000	1.00000	0.01925

Таблица 12: Таблица сравнения относительных частот и теоретических вероятностей

<b>Название показателя</b>	<b>Выборочное значение</b>	<b>Теоретическое значение</b>	<b>Абсолютное отклонение</b>	<b>Относительное отклонение</b>
Среднее значение	2.12563	1.98020	0.14543	7.34421
Дисперсия	3.97095	3.92118	0.04977	1.26926
Среднее квадратичное отклонение	1.99273	1.98020	0.01253	0.63276
Мода	0.04086	0.00000	0.04086	inf
Медиана	1.25726	1.37257	0.11531	8.40103
Коэффициент асимметрии	1.77903	2.00000	0.22097	11.04850
Коэффициент эксцесса	3.29373	6.00000	2.70627	45.10450

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

MS\_1.py (Биномиальное Распределение):

```
from TR_1 import *
import numpy as np
import matplotlib.pyplot as plt
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)

print("200 случайных значений из биномиального распределения с
параметрами n = ", n, f"p = {p: .5f}")
x_binom = rng.binomial(n, p, N)
print(x_binom)
x_bi_save = x_binom.copy().reshape((20, 10))
x_bi_save.sort()
df = pd.DataFrame(x_bi_save)
df.to_excel('binomial.xlsx')

task(x_binom, 'binomial')
```

MS\_2.py(Геометрическое Распределение):

```
from TR_1 import *
import numpy as np
import matplotlib.pyplot as plt
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)

print("200 случайных значений из геометрического распределения с
параметром ", f"p = {p: .5f}")
x_geom = rng.geometric(p, N) - 1
```



```

print(x_geom)
X_geo_save = x_geom.copy().reshape((20, 10))
X_geo_save.sort()
df = pd.DataFrame(X_geo_save)
df.to_excel('geometric.xlsx')

task(x_geom, 'geometric')

```

MS\_3.py(Показательное Распределение):

```

from TR_1 import *
import numpy as np
import matplotlib.pyplot as plt
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)

print("200 случайных значений из экспонентального распределения с
параметром ", f"lambda = {lamda: .5f}")
x_exponential = rng.exponential(1/lamda, N)
x_exponential.sort()
print(x_exponential)
x_po_save = x_exponential.copy().reshape((20, 10))
df = pd.DataFrame(x_po_save)
df.to_excel('exponential.xlsx')

task(x_exponential, 'exponential')

```

TR\_1.py(Код для выполнения распрелений):

```

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 == 'exponential':
        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 == 'exponential':
        step = np.round(1 + np.log2(N), 5)
        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 i == 0:
                    if intervals[i] <= X[j] <= intervals[i+1]:
                        intervals_freq[i] += freq[j]
                        intervals_relative_freq[i] += rel_freq[j]
                else:
                    if intervals[i] < X[j] <= intervals[i+1]:

```

```

        intervals_freq[i] += freq[j]
        intervals_relavie_freq[i] += rel_freq[j]

intervals = np.round(intervals, 5)

total_row = {
    'Frequency': f"{total_freq: .5f}",
    'Relative Frequency': f"{total_rel_freq: .5f}"
}
df_freq_intervals = pd.DataFrame({
    'Interval: ': list(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_relavie_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)])
    middle = np.round(middle, 5)
    df_freq_middle = pd.DataFrame({'Middle': middle, 'Frequency':
intervals_freq, 'Relative Frequency': intervals_relavie_freq})
    df_freq_middle.loc[len(df_freq_intervals)] = total_row
    print(df_freq_middle)
    df_freq_middle.to_excel(name_distribution + '_middle.xlsx')
    cumulative_freq_intervals = np.cumsum(intervals_relavie_freq)

#2
if name_distribution == 'binomial':
    theo_freq = np.array([(math.comb(n, k) * (p ** k) * ((1-p) ** (n-k)))
for k in counts[0]])
    elif name_distribution == 'geometric':
        theo_freq = np.array([p * ((1-p) ** k) for k in counts[0]])
    elif name_distribution == 'exponential':
        theo_freq = np.array([lamda * np.exp(-lamda * k) for k in
counts[0]])
        intervals_theoretical_relavie_freq = np.zeros(len(intervals)-1)
        for i in range(len(intervals)-1):
            intervals_theoretical_relavie_freq[i] = -np.exp(-lamda *
intervals[i+1]) + np.exp(-lamda * intervals[i])
            intervals_theoretical_relavie_freq =
np.round(intervals_theoretical_relavie_freq, 5)

#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)

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

    ax1.plot(counts[0], theo_freq, 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()
if name_distribution == 'binomial' or name_distribution == 'geometric':
    #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.argmax(freq == np.amax(freq))].flatten().tolist()
    mode = (modes[0] + modes[len(modes) - 1]) / 2
    print("Mode = ", f"{mode: .5f}")

    #8
    idx = np.where(cum_freq >= 0.5)[0][0]
    if cum_freq[idx] == 0.5 and idx < len(X) - 1:

```

```

        median = (X[idx] + X[idx + 1]) / 2
    else:
        median = X[idx]
    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}")

if name_distribution == 'exponential':
    mean = np.array([middle[i] * intervals_relavie_freq[i] for i in
range(len(middle))]).sum()

    h = ((X[len(X)-1] - X[0]) / len(X))
    variance = np.array([(middle[i] - mean) ** 2) *
intervals_relavie_freq[i] for i in range(len(middle))]).sum() - (h ** 2)
/ 12

    deviation = np.sqrt(variance)

    mode = intervals[0] + h * (intervals_relavie_freq[0]) / (2 *
intervals_relavie_freq[0] - intervals_relavie_freq[1])

    if 0.5 in cumulative_freq_intervals.tolist():
        median = middle[cumulative_freq_intervals.tolist().index(0.5)+1]
    else:
        for i in range(len(cumulative_freq_intervals)):
            if cumulative_freq_intervals[i] > 0.5:
                pivot = i-1
                break
        median = intervals[pivot+1] + h * ((0.5 -
cumulative_freq_intervals[pivot]) / intervals_relavie_freq[pivot+1])

    def sample_k_moment_around_mean(k, mean):
        return np.array([(middle[i] - mean) ** k *
intervals_relavie_freq[i] for i in range(len(middle))]).sum()

    sample_skeness = sample_k_moment_around_mean(3, mean) / deviation **
3
    sample_kurtosis = sample_k_moment_around_mean(4, mean) / deviation **
4 - 3

if name_distribution == 'binomial' or name_distribution == 'geometric':
    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]
    })
    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')

    if name_distribution == 'exponential':
        abs_diff_freq_intervals = np.abs(intervals_theoretical_relative_freq
- intervals_relative_freq)
        freq_compare_intervals = pd.DataFrame({
            'Interval': ': list(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 = {
            'X': 'Total',
            'Relative Frequency': f"{total_rel_freq: .5f}",
            'Theoretical Frequency': f"{1: .5f}",
            'Absolute Difference': f"{np.max(abs_diff_freq_intervals): .5f}"
        }
        freq_compare_intervals.loc[len(freq_compare_intervals)] =
total_row_compare_intervals
        freq_compare_intervals.to_excel(name_distribution +
'_intervals_compare.xlsx')

#10
char_combine = None
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_kurtosis = (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_kurtosis, theo_mode, theo_median])
    theo_values = np.round(theo_values, 5)
    real_values = np.array([mean, variance, deviation, sample_skewness,
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)])
        abs_differences = np.round(abs_differences, 5)
        rel_differences = np.array([abs_differences[i] / theo_values[i] for i
in range(len(theo_values))])
        rel_differences *= 100
        rel_differences = np.round(rel_differences, 5)
        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':
        theoretical_mean = (1-p) / 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 = 0
        theo_median = np.round((-1 / np.log2(1 - p)) - 1)
        theo_values = np.array([theoretical_mean, theo_var, theo_deviation,
theo_skewness, theo_kurtois, theo_mode, theo_median])
        theo_values = np.round(theo_values, 5)
        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))])
        abs_differences = np.round(abs_differences, 5)
        rel_differences = np.array([abs_differences[i] / theo_values[i] for i
in range(len(theo_values))])
        rel_differences *= 100
        rel_differences = np.round(rel_differences, 5)
        char_combine = pd.DataFrame({'Characteristic': ['Mean', 'Variance',
'Deviation', 'Skewness', 'Kurtosis', 'Mode', 'Median'],
                                'Sample': real_values,
                                'Theoretical': theo_values,
                                'Absolute Difference': abs_differences,
                                'Relative Difference': rel_differences})
        char_combine.to_excel(name_distribution + '_char.xlsx')
    elif name_distribution == 'exponential':
        theo_mean = 1/lamda
        theo_var = 1/(lamda**2)
        theo_deviation = 1/lamda
        theo_skewness = 2

```

```

theo_kurtois = 6
theo_mode = 0
theo_median = np.log(2) / lamda
theo_values = np.array([theo_mean, theo_var, theo_deviation,
theo_skewness, theo_kurtois, theo_mode, theo_median])
theo_values = np.round(theo_values, 5)
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))])
abs_differences = np.round(abs_differences, 5)
rel_differences = np.array([abs_differences[i] / theo_values[i] for i
in range(len(theo_values))])
rel_differences *= 100
rel_differences = np.round(rel_differences, 5)
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)

```