|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | |  |  |  | | 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](#_Toc2897)

[Краткие теоретические сведения 5](#_Toc28770)

[Результаты расчётов 1](#_Toc2879)3

[Список литературы 23](#_Toc29997)

[Приложение 24](#_Приложение)

# Задание

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

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

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 .

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 ***n*** и ***p***:

Выборка объёмом *N=*200 с сгенерированными псевдослучайными числами,

распределённые по геометрическому закону с параметром ***p***:

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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| … | … | … | … |
|  |  |  |  |
|  |  |  | - |

Таблица 1. Статистический ряд.

Полигон относительных частот - ломаная линия, соединяющая последовательно точки с координатами .

Эмпирическая функция распределения:

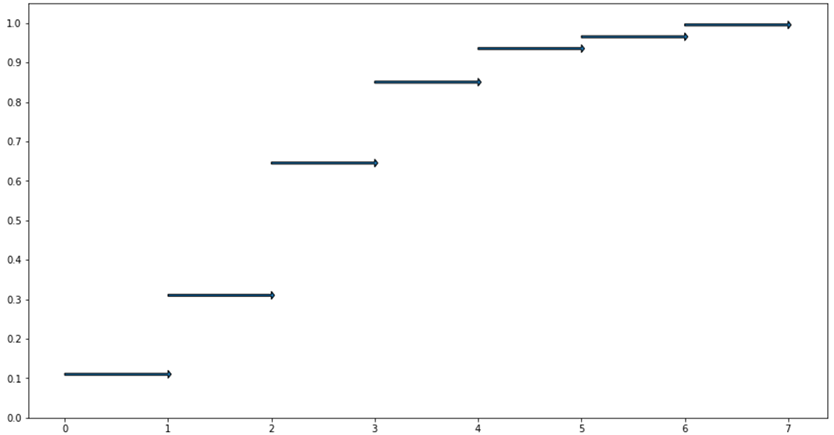


Рисунок 1. Образец графика эмпирической функции распределения.

Выборочное среднее:

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

Выборочная дисперсия:

Выборочная центральный момент k-ого порядка:

Выборочное среднее квадратическое отклонение:

Выборочный коэффициент асимметрии:

Выборочный коэффициент эксцесса:

Выборочная мода ;

если то ,

если , то *i < k < j*, то - не существует.

Выборочная медиана:

|  |  |
| --- | --- |
| **Биномиальное распределение** | |
| Вероятность |  |
| Математическое ожидание |  |
| Дисперсия |  |
| Среднее квадратическое отклонение |  |
| Мода | , если - дробное;  , если - целое; |
| Медиана |  |
| Коэффициент асимметрии |  |
| Коэффициент эксцесса |  |

Таблица 2. Характеристики биномиального распределения.

|  |  |
| --- | --- |
| **Геометрическое распределение** | |
| Вероятность |  |
| Математическое ожидание |  |
| Дисперсия |  |
| Среднее квадратическое отклонение |  |
| Мода | 0 |
| Медиана |  |
| Коэффициент асимметрии |  |
| Коэффициент эксцесса |  |

Таблица 3. Характеристики геометрического распределения.

В задании 3 полученную выборку псевдослучайных чисел, распределённые по показательному закону, упорядочить по возрастанию, определить интервалы ; число интервалов находится по формуле Стерджеса .

|  |  |  |
| --- | --- | --- |
| Интервалы |  |  |
|  |  |  |
|  |  |  |
| ... | ... | ... |
|  |  |  |
|  |  |  |

Таблица 4. Интервальный ряд.

- число значений, попавших в *i-*ый интервал; - относительная частота попадания в *i-*ый интервал, .

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
| ... | ... | ... |
|  |  |  |
|  |  |  |

Таблица 5. Ассоциированный статический ряд, где - середина интервала.

Эмпирическая функция распределения:

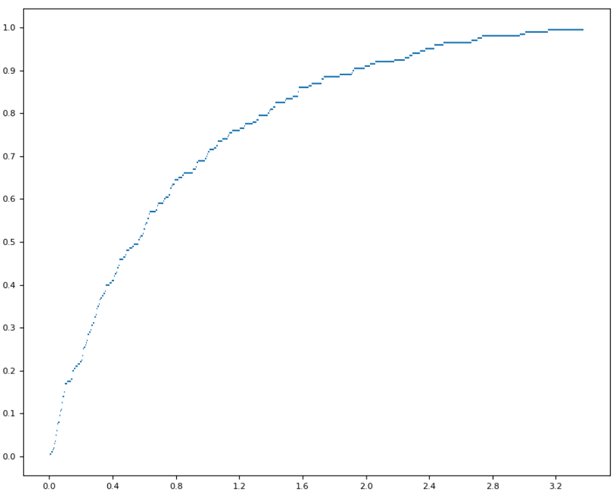


Рисунок 2. Образец графика эмпирической функции распределения.

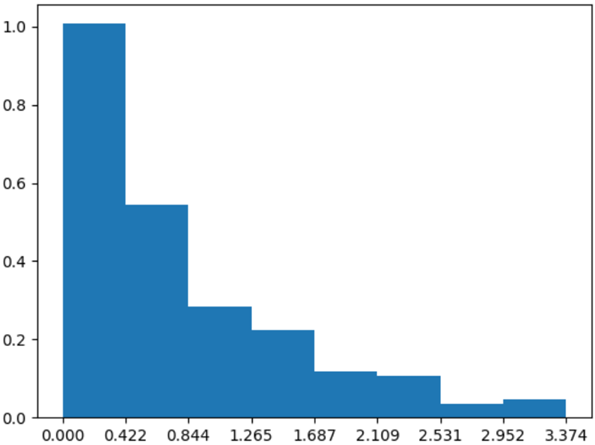


Рисунок 3. Образец гистограммы относительных частот.

Площадь *i*-ого столбца гистограммы равна , а высота .

Выборочное среднее:

Выборочная дисперсия с поправкой Шеппарда:

Выборочное среднее квадратическое отклонение:

Выборочная мода:

Если модальный интервал, на котором высота гистограммы максимальна, один, то , где - левая граница модального интервала ; - правая граница модального интервала ; - относительная частота на модальном интервале; , - относительные частоты интервалов слева и справа от модального интервала.

Если модальных интервалов несколько, и все они идут подряд (т. е. интервалы , ..., - все модальные), то

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

Выборочная медиана:

Выборочный момент *k*-ого порядка:

Выборочный центральный момент *k*-ого порядка:

Выборочный коэффициент асимметрии:

Выборочный коэффициент эксцесса:

|  |  |
| --- | --- |
| **Показательное распределение:** ) | |
| Математическое ожидание |  |
| Дисперсия |  |
| Среднее квадратическое отклонение |  |
| Мода | 0 |
| Медиана |  |
| Коэффициент асимметрии |  |
| Коэффициент эксцесса |  |

# Результаты расчётов

**Задание 1: Биномиальное Распределение**

Параметры Бинамиольного Распределения:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 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 |
| 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*** |  |  |  |
| 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

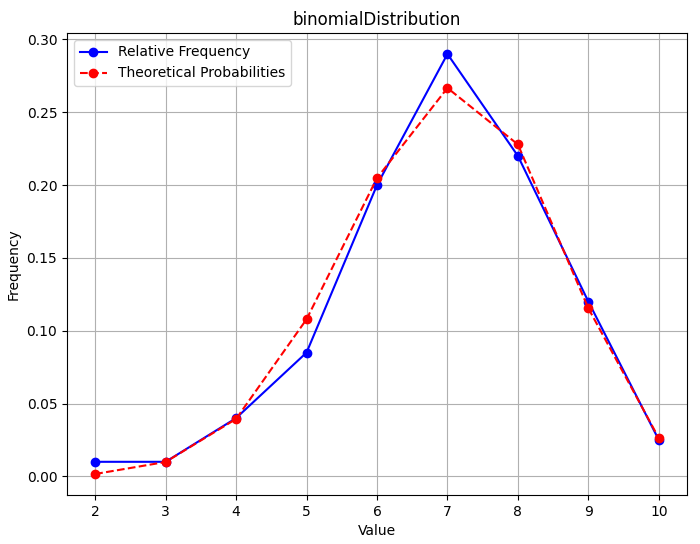


Рисунок 1: Полигон относительных частот

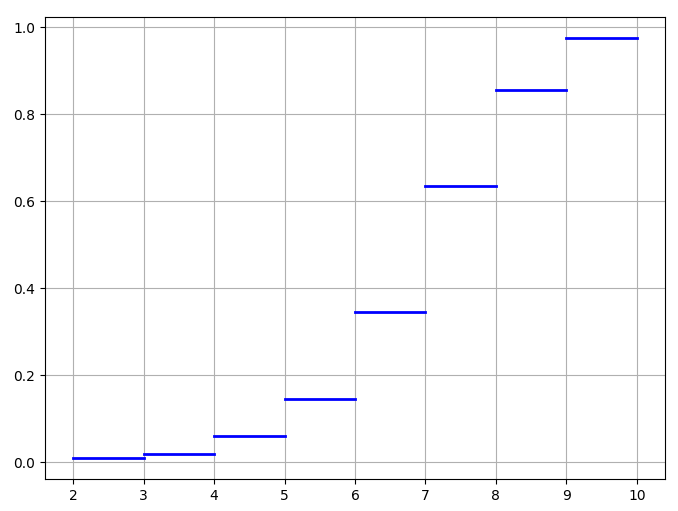


Рисунок 2: График эмпирической функции распеределении

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 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 сранения рассчитанных характеристик с теоретическими значениями

**Задание 2: Геометрическое Распределение**

Параметры геометрического распределения:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 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

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 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

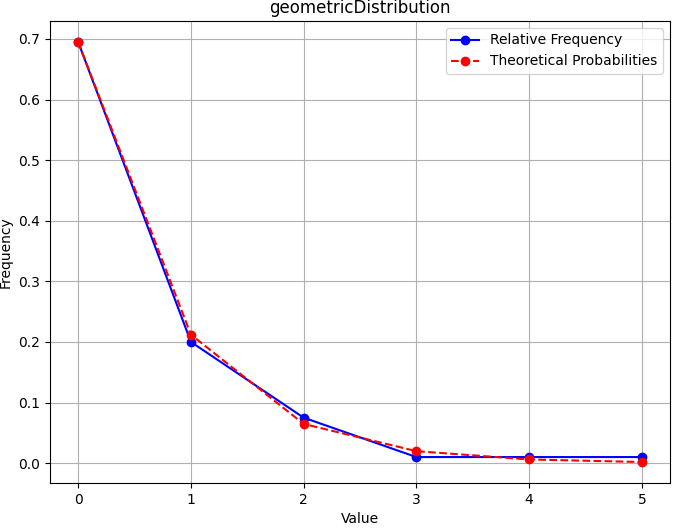


Рисунок 3: Полигон относительных частот

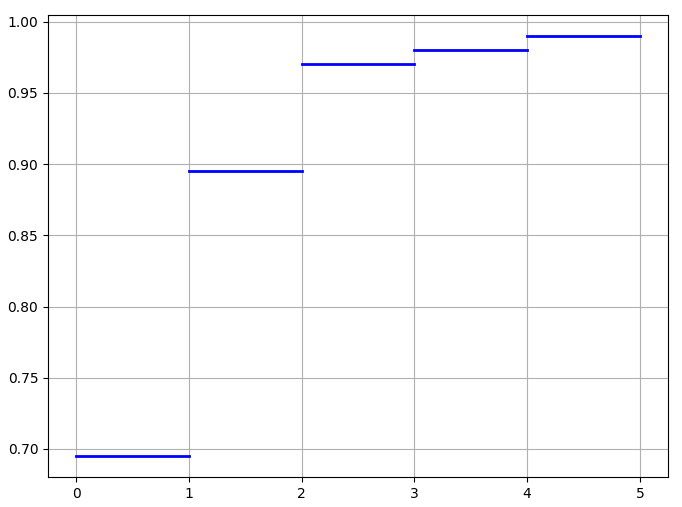


Рисунок 4: График эмпирической функции распределения

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 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**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 10.75481 |

Таблица 9: 200 выборок показательного распределения

Показательное Распределение

|  |  |  |
| --- | --- | --- |
| **Интервалы** |  |  |
| (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

|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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

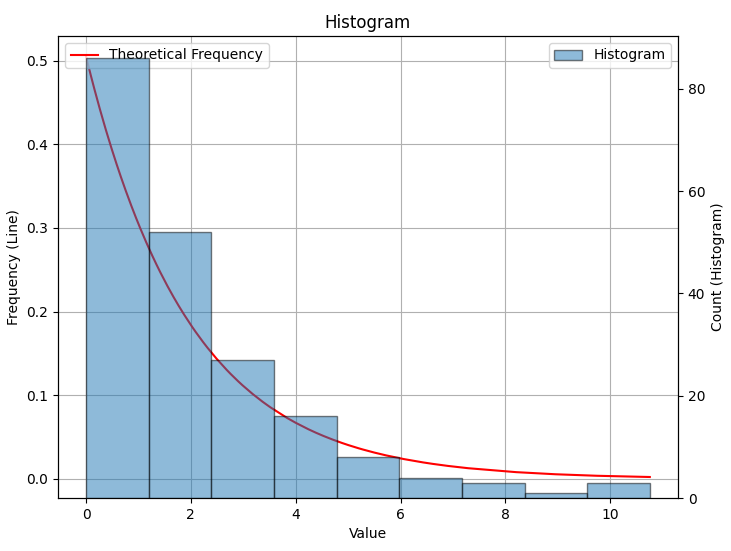


Рисунок 5: Гистограмма относительных частот с наложенным на нее и выделенным красным цветом графиком плотности показательного распределения

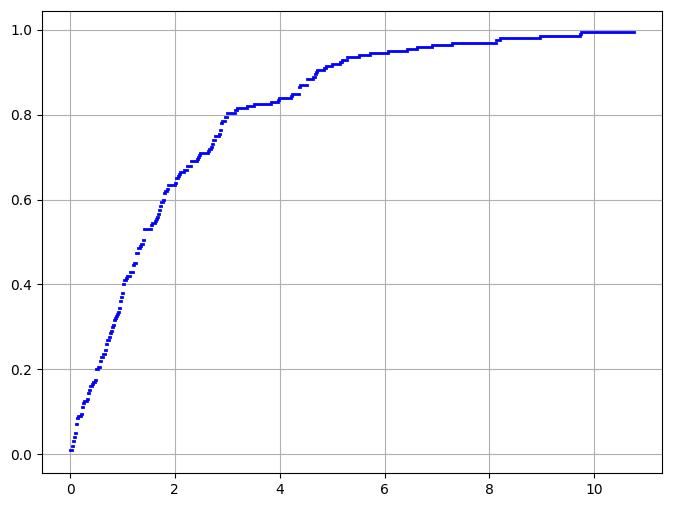


Рисунок 6: График эмфирической функции распределения

|  |  |  |  |
| --- | --- | --- | --- |
| **Интервалы:** |  |  |  |
| (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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Название показателя** | **Выборочное значение** | **Теоретическое значение** | **Абсолютное отклонение** | **Относительное отклонение** |
| 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 экспонентиального 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\_relatvie\_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\_relatvie\_freq[i] += rel\_freq[j]

else:

if intervals[i] < X[j] <= intervals[i+1]:

intervals\_freq[i] += freq[j]

intervals\_relatvie\_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\_relatvie\_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\_relatvie\_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\_relatvie\_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\_relatvie\_freq = np.zeros(len(intervals)-1)

for i in range(len(intervals)-1):

intervals\_theoretical\_relatvie\_freq[i] = -np.exp(-lamda \* intervals[i+1]) + np.exp(-lamda \* intervals[i])

intervals\_theoretical\_relatvie\_freq = np.round(intervals\_theoretical\_relatvie\_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.argwhere(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\_relatvie\_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\_relatvie\_freq[i] for i in range(len(middle))]).sum() - (h \*\* 2) / 12

deviation = np.sqrt(variance)

mode = intervals[0] + h \* (intervals\_relatvie\_freq[0]) / (2 \* intervals\_relatvie\_freq[0] - intervals\_relatvie\_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\_relatvie\_freq[pivot+1])

def sample\_k\_moment\_around\_mean(k, mean):

return np.array([(middle[i] - mean) \*\* k \* intervals\_relatvie\_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\_relatvie\_freq - intervals\_relatvie\_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\_relatvie\_freq],

'Theoretical Frequency': [f"{val:.5f}" for val in intervals\_theoretical\_relatvie\_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\_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])

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(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)