|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | |  |  |  | | --- | --- | --- | |  |  |  | | Министерство образования и науки РОССИЙСКОЙ ФЕДЕРАЦИИ | | |   Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»** | |
|  | |
|  | |
|  |  |

ИНСТИТУТ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

КАФЕДРА ВЫСШЕЙ МАТЕМАТИКИ

Лабораторная работа 1

 по курсу «**Теория вероятностей и математическая статистика, часть 2**»

**ВАРИАНТ 14**

Тема: \_\_\_\_\_\_\_ **Первичная обработка выборки**\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнил:

Студент 4-го курса

Малов И.М.

Группа: КМБО-01-20

МОСКВА – 2023

**Содержание**

[**Задание 3**](#_Toc148893755)

[**Краткие теоретические сведения 5**](#_Toc148893756)

[**Результаты расчетов 17**](#_Toc148893757)

[**Задание 1 17**](#_Toc148893758)

[**Задание 2 21**](#_Toc148893759)

[**Задание 3 25**](#_Toc148893760)

[**Задание 4 29**](#_Toc148893761)

[**Задание 5 35**](#_Toc148893762)

[**Задание 6 37**](#_Toc148893763)

[**Задание 7 43**](#_Toc148893764)

[**Задание 8 47**](#_Toc148893765)

[**Список литературы 51**](#_Toc148893766)

[**Приложение 52**](#_Toc148893767)

# Задание

**Задание 1.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по биномиальному закону с параметрами ***n*** и ***p***:



**Задание 2.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по геометрическому закону с параметром ***p***:



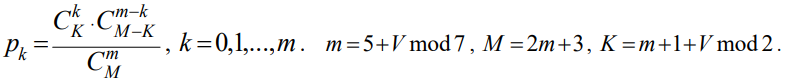
**Задание 3.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по закону Пуассона с параметром ***λ***:



**Задание 4.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных равномерно на множестве {0, …, *n*-1}:



**Задание 5.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по гипергеометрическому закону с параметрами ***M***, ***K***, ***m***:

 Следуя **Указаниям** для всех выборок в Заданиях 1-5 построить:

1) статистический ряд;

2) полигон относительных частот;

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 6.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по нормальному закону с параметрами *a* = (-1)*V*•0,005•*V* и σ2, где σ = 0,005•*V*+1.

**Задание 7.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по показательному закону с параметром *λ* = 2 +(-1)*V*•0,01•*V*.

**Задание 8.** Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных равномерно на отрезке [*a*, *b*], где *a* = (-1)*V*•0,02•*V*, *b* = *a* + 6.

Следуя **Указаниям** для всех выборок в Заданиях 6-8 построить:

1) интервальный ряд и ассоциированный статистический ряд;

2) гистограмму относительных частот;

3) график эмпирической функции распределения;

найти:

1) выборочное среднее;

2) выборочную дисперсию 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.

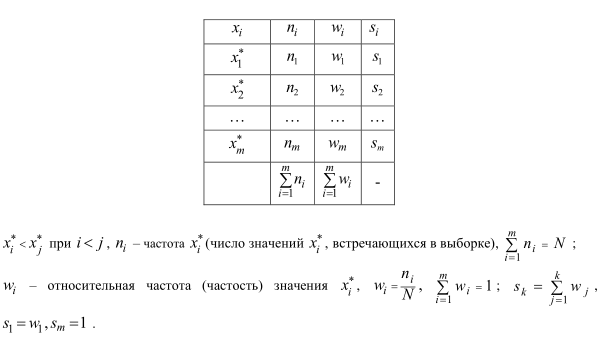
V − номер варианта.

Результаты вычислений приводить в отчете с точностью до **0,00001**.

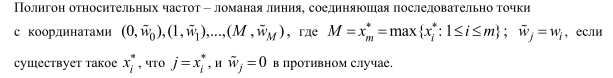
# Краткие теоретические сведения

**Информация о дискретных случайных величинах**

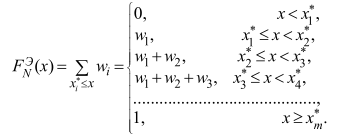
**Определение статистического ряда**



**Определение полигона относительных частот**



**Определение эмпирической функции распределения**



**Формула для расчета выборочного среднего**



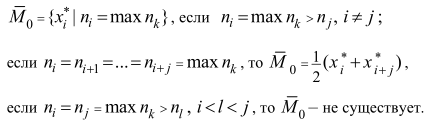
**Формула для расчета выборочной дисперсии**



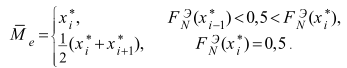
**Формула для расчета выборочного среднего квадратичного отклонения**

****

**Формула для расчета выборочной моды**



**Формула для расчета выборочной медианы**



**Формула для расчета выборочного центрального k-ого момента**



**Формула для расчета выборочного коэффициента асимметрии**



**Формула для расчета выборочного коэффициента эксцесса**



**Биноминальное распределение**

**Выражение для вероятностей ряда распределения**

****

**Выражение для математического ожидания**

****

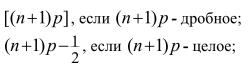
**Выражение для дисперсии**

****

**Выражение для среднего квадратичного отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициента эксцесса**

****

**Геометрическое распределение**

**Выражение для вероятностей ряда распределения**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

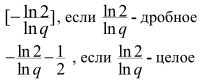
**Выражение для среднего квадратичного отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициента эксцесса**

****

**Распределение Пуассона**

**Выражение для вероятностей ряда распределения**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

**Выражение для среднего квадратичного отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициента эксцесса**

****

**Равномерное распределение на множестве**

**Выражение для вероятностей ряда распределения**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

**Выражение для среднего квадратичного отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициента эксцесса**

****

**Гипергеометрическое распределение**

**Выражение для вероятностей ряда распределения**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

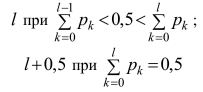
**Выражение для среднего квадратичного отклонения**

****

**Выражение для моды**

****

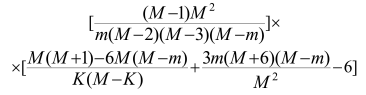
**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициента эксцесса**

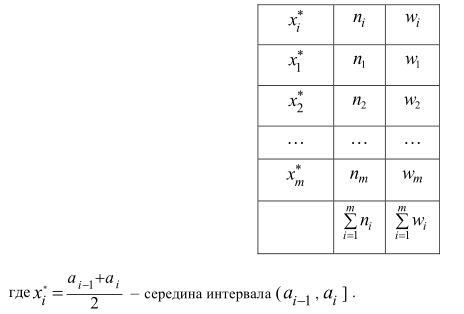
****

**Информация о непрерывных случайных величинах**

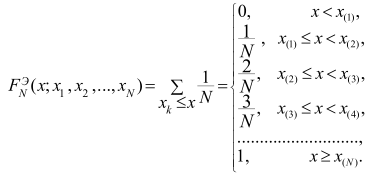
**Интервальный ряд** — это способ представления данных в виде диапазонов значений, называемых интервалами. Каждый интервал включает в себя наблюдения, попадающие в данный диапазон.

****

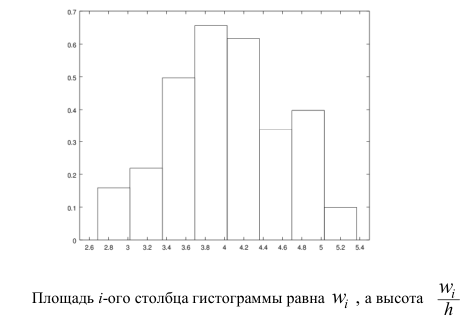
**Определение ассоциированного статистического ряда**

****

**Определение эмпирической функции распределения**

****

**Определение гистограммы относительных частот**

****

**Выборочное среднее**:

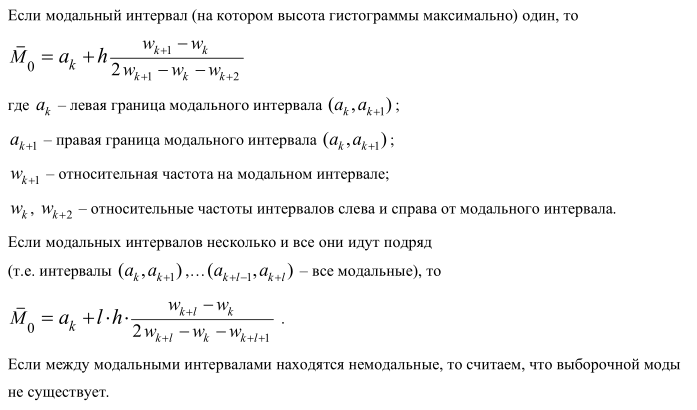


**Выборочная дисперсия с поправкой Шеппарда**:

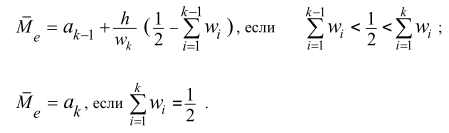


**Выборочное среднее квадратичное отклонение**: σ =

**Формула выборочной моды**

****

**Формула выборочной медианы**

****

**Формула выборочного центрального момента k-ого порядка**

****

**Формула выборочного коэффициента асимметрии**

****

**Формула выборочного коэффициента эксцесса**

****

**Нормальное распределение**

**Выражение для функции плотности вероятности**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

**Выражение для среднего квадратического отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициент эксцесса**

****

**Показательное распределение**

**Выражение для функции плотности вероятности**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

**Выражение для среднего квадратического отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

; 2

**Выражение для коэффициент эксцесса**

; 6

**Равномерное распределение**

**Выражение для функции плотности вероятности**

****

**Выражение для математического ожидания**

****

**Выражение для дисперсии**

****

**Выражение для среднего квадратического отклонения**

****

**Выражение для моды**

****

**Выражение для медианы**

****

**Выражение для коэффициента асимметрии**

****

**Выражение для коэффициент эксцесса**

****

**Средства языка программирования**

Программа написана на скриптовом интерпретируемом языке программирования - Python.

Для расчетов использовались:

1. numPy - для работы с многомерными массивами, возведения в степень, округления до 5 знаков после запятой, взятия целой части, сохранения данных в файл и загрузки их из него, получения различных статистических выборок по заданным параметрам.
2. pandas - предоставляет специальные структуры данных и операции для манипулирования таблицами.
3. matplotlib - визуализация таблиц и графиков.
4. math - использование функций для подсчета биномиальных коэффициентов и логарифмов.
5. statistics - нахождение моды и медиан

Для получения выборок использовались функции:

* ***binom.rvs(n, p, size)*** – возвращает *N* целых чисел распределённых по биномиальному закону с параметрами *n, p;*
* ***geom.rvs(p, size****)* – возвращает *N* целых чисел распределённых по геометрическому закону с параметром *p;*
* ***poisson.rvs(l, size)*** – возвращает N целых чисел, распределённых по Пуассоновскому закону с параметром *l;*
* ***randint.rvs(0, n, size)*** – возвращает *N* целых чисел распределённых равномерно на множестве *[0, n-1];*
* ***hypergeom.rvs(M, K,m size)*** – возвращает *N* целых чисел распределённых по гипергеометрическому закону с параметрами *M, K, m;*
* ***norm.rvs(a, sigma, size)*** – возвращает *N* дробных чисел распределённых по нормальному закону с параметрами *a,* *;*
* ***expon.rvs(1/l, size)*** – возвращает *N* дробных чисел распределённых по показательному закону с параметром *l;*
* ***uniform.rvs(size, A, B–A)*** – возвращает *N* дробных чисел распределённых равномерно на отрезке *[A, B].*
* ***np.random.normal(a,σ,200)*** - генерация выборки, распределенной по нормальному закону, a-среднее “центр” распределения, σ -стандартное отклонение(разброс или “ширина”).
* ***np.random.exponential(1/lbd,200)*** - генерация выборки, распределенной по экспоненциальному закону,1/lbd - параметр масштаба, обратный параметру скорости λ.
* ***np.random.uniform(a,b,200)*** - генерация выборки, распределенной равномерно, a-нижняя граница выходного интервала, все сгенерированные значения будут больше или равны этому значению, b - верхняя граница выходного интервала, все сгенерированные значения будут меньше или равны этому значению.

# Результаты расчетов

## Задание 1

*n* = 19, *p* = 0.27

Выборка неупорядоченная:

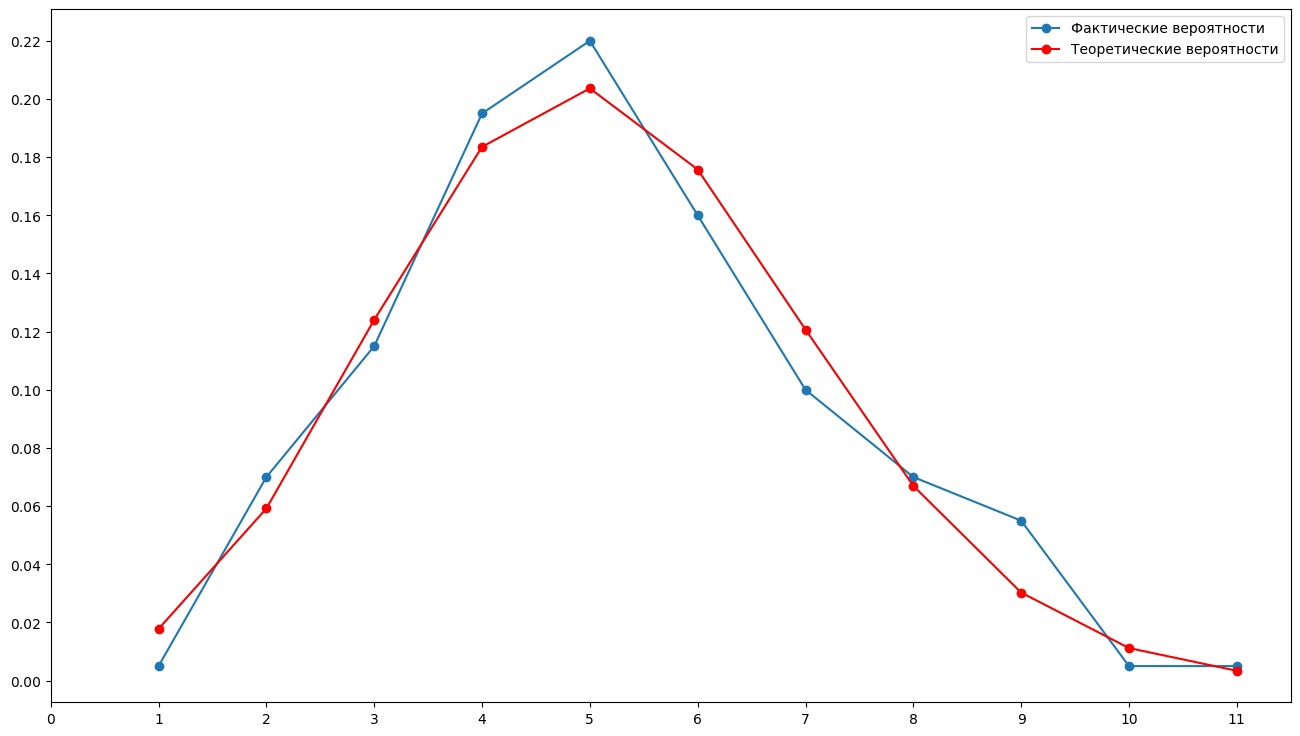
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 3 | 4 | 3 | 3 | 5 | 4 | 9 | 5 | 5 | 6 |
| 3 | 7 | 4 | 4 | 5 | 6 | 4 | 5 | 4 | 2 |
| 7 | 5 | 4 | 7 | 9 | 5 | 4 | 7 | 9 | 2 |
| 5 | 6 | 5 | 4 | 8 | 3 | 9 | 6 | 7 | 4 |
| 2 | 5 | 4 | 3 | 4 | 5 | 2 | 5 | 3 | 6 |
| 5 | 9 | 6 | 3 | 4 | 5 | 7 | 6 | 5 | 8 |
| 6 | 3 | 4 | 6 | 6 | 8 | 6 | 5 | 3 | 5 |
| 2 | 4 | 2 | 8 | 4 | 4 | 5 | 4 | 4 | 6 |
| 4 | 8 | 8 | 2 | 7 | 6 | 6 | 2 | 10 | 7 |
| 6 | 8 | 2 | 6 | 2 | 8 | 4 | 4 | 5 | 8 |
| 5 | 4 | 5 | 5 | 5 | 5 | 3 | 7 | 5 | 6 |
| 5 | 6 | 5 | 1 | 8 | 5 | 6 | 5 | 6 | 5 |
| 4 | 4 | 4 | 2 | 3 | 8 | 3 | 3 | 9 | 4 |
| 8 | 9 | 3 | 4 | 7 | 9 | 4 | 8 | 6 | 4 |
| 6 | 4 | 6 | 3 | 5 | 4 | 5 | 3 | 4 | 4 |
| 6 | 5 | 6 | 2 | 6 | 9 | 5 | 7 | 3 | 7 |
| 7 | 5 | 2 | 6 | 9 | 11 | 5 | 7 | 5 | 7 |
| 7 | 7 | 6 | 5 | 6 | 7 | 8 | 7 | 4 | 3 |
| 6 | 3 | 2 | 4 | 3 | 5 | 5 | 6 | 5 | 5 |
| 6 | 4 | 3 | 5 | 5 | 3 | 7 | 4 | 9 | 4 |

Выборка упорядоченная:

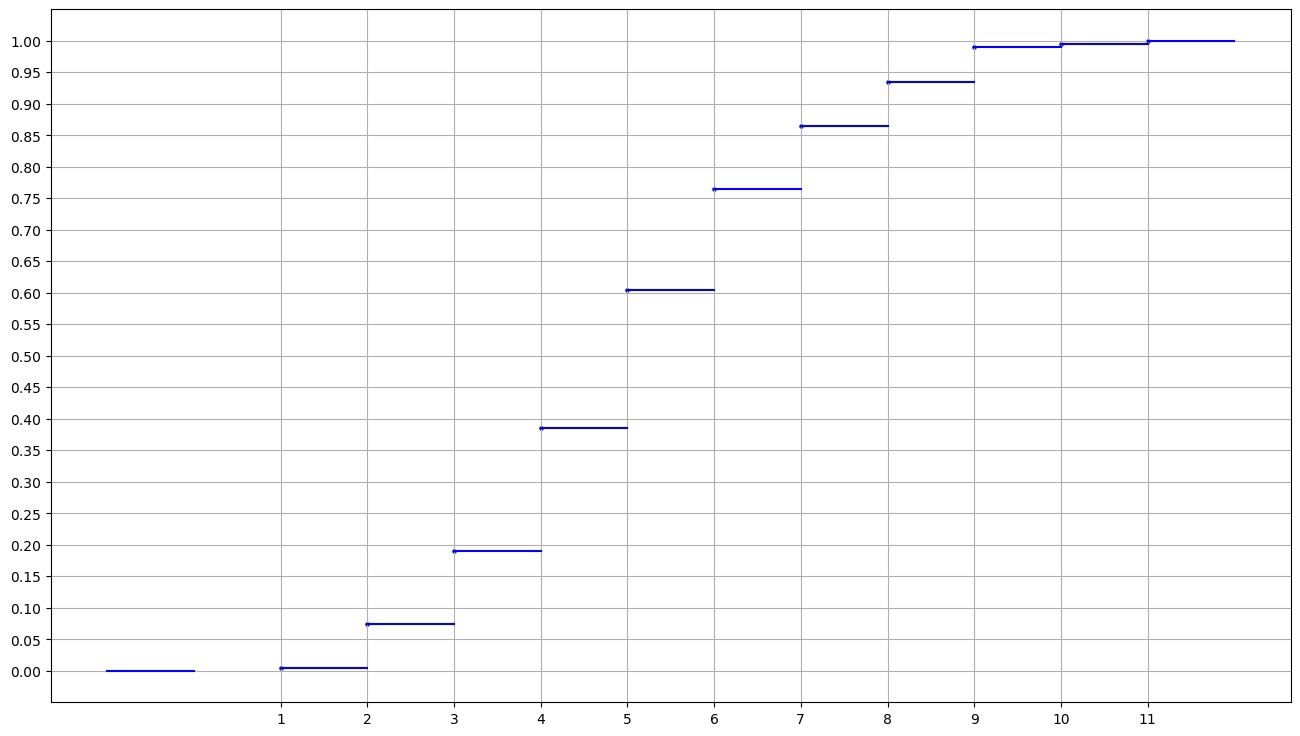
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 5 | 5 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 5 | 5 | 5 | 5 | 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 | 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 | 9 | 9 | 9 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 10 | 11 |

Статистический ряд:

|  |  |  |  |
| --- | --- | --- | --- |
| *xi* | *ni* | *wi* | *si* |
| 1 | 1 | 0.00500 | 0.00500 |
| 2 | 14 | 0.07000 | 0.07500 |
| 3 | 23 | 0.11500 | 0.19000 |
| 4 | 39 | 0.19500 | 0.38500 |
| 5 | 44 | 0.22000 | 0.60500 |
| 6 | 32 | 0.16000 | 0.76500 |
| 7 | 20 | 0.10000 | 0.86500 |
| 8 | 14 | 0.07000 | 0.93500 |
| 9 | 11 | 0.05500 | 0.99000 |
| 10 | 1 | 0.00500 | 0.99500 |
| 11 | 1 | 0.00500 | 1.00000 |



**Рисунок 1.** Полигон относительных частот задания 1



**Рисунок 2**. Эмпирическая функция распределения задания 1

1. Выборочное среднее: 5.19000
2. Выборочная дисперсия: 3.72251
3. Выборочное среднее квадратичное отклонение: 1.92938
4. Выборочная мода: 5.00000
5. Выборочная медиана: 5.00000
6. Выборочный коэффициент асимметрии: 0.37201
7. Выборочный коэффициент эксцесса: -0.25795

**Биномиальное распределение**

|  |  |  |  |
| --- | --- | --- | --- |
| *j* | *wj* | *pj* | *|wj-pj|* |
| 1 | 0.00500 | 0.01778 | 0.01278 |
| 2 | 0.07000 | 0.05919 | 0.01081 |
| 3 | 0.11500 | 0.12405 | 0.00905 |
| 4 | 0.19500 | 0.18352 | 0.01148 |
| 5 | 0.22000 | 0.20363 | 0.01637 |
| 6 | 0.16000 | 0.17574 | 0.01574 |
| 7 | 0.10000 | 0.12071 | 0.02071 |
| 8 | 0.07000 | 0.06697 | 0.00303 |
| 9 | 0.05500 | 0.03027 | 0.02473 |
| 10 | 0.00500 | 0.01120 | 0.00062 |
| 11 | 0.00500 | 0.00339 | 0.00161 |
|  | 1.00000 | 0.99644 | 0.02473 |

**Биномиальное распределение**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название показателя | Экспериментальное значение | Теоретическое значение | Абсолютное отклонение | Относительное отклонение |
| Выборочное среднее | 5.19000 | 5.13000 | 0.06000 | 1.16279 |
| Выборочная дисперсия | 3.72251 | 3.74490 | 0.02239 | 0.59967 |
| Выборочное среднее квадратическое отклонение | 1.92938 | 1.93517 | 0.00579 | 0.29965 |
| Выборочная мода | 5.00000 | 5.00000 | 0.00000 | 0.00000 |
| Выборочная медиана | 5.00000 | 5.00000 | 0.00000 | 0.00000 |
| Выборочный коэффициент асимметрии | 0.37201 | 0.23770 | 0.13431 | 44.05701 |
| Выборочный коэффициент эксцесса | -0.25795 | -0.04876 | 0.30671 | 293.23581 |

## Задание 2

*p* = 0.27

Неупорядоченная выборка:

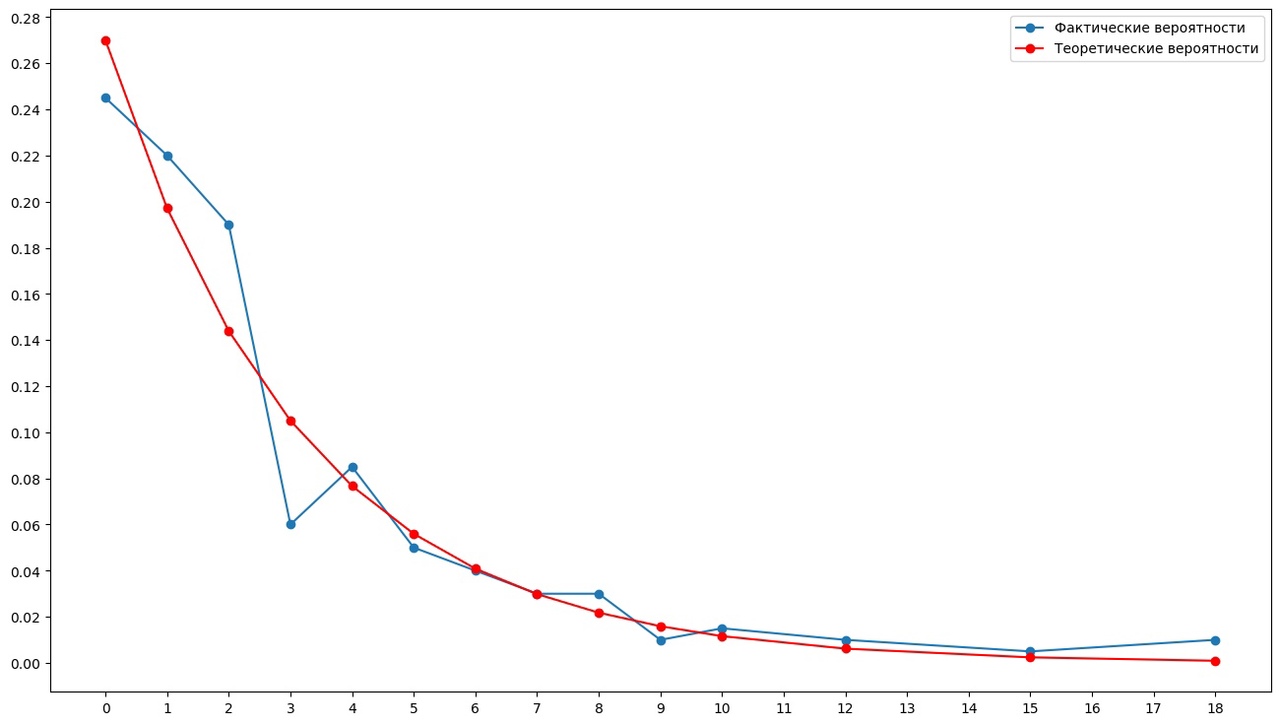
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 4 | 2 | 0 | 10 | 15 | 0 | 4 | 0 | 3 |
| 2 | 3 | 12 | 2 | 3 | 3 | 2 | 2 | 5 | 1 |
| 0 | 8 | 1 | 4 | 2 | 1 | 0 | 1 | 1 | 5 |
| 2 | 3 | 7 | 2 | 0 | 8 | 7 | 0 | 6 | 1 |
| 0 | 2 | 0 | 1 | 4 | 1 | 6 | 0 | 4 | 1 |
| 0 | 2 | 2 | 7 | 8 | 4 | 1 | 1 | 1 | 2 |
| 0 | 1 | 2 | 7 | 18 | 7 | 0 | 5 | 6 | 0 |
| 5 | 0 | 5 | 4 | 4 | 4 | 2 | 0 | 10 | 1 |
| 0 | 2 | 3 | 4 | 0 | 1 | 3 | 2 | 0 | 1 |
| 1 | 0 | 2 | 0 | 0 | 2 | 12 | 9 | 2 | 6 |
| 5 | 5 | 3 | 1 | 2 | 1 | 1 | 0 | 2 | 4 |
| 0 | 2 | 0 | 1 | 0 | 2 | 5 | 4 | 2 | 2 |
| 3 | 4 | 0 | 2 | 2 | 1 | 1 | 6 | 0 | 4 |
| 2 | 2 | 6 | 1 | 0 | 1 | 1 | 0 | 0 | 5 |
| 1 | 0 | 5 | 1 | 2 | 8 | 7 | 1 | 1 | 2 |
| 8 | 0 | 0 | 3 | 1 | 0 | 1 | 1 | 1 | 0 |
| 4 | 1 | 18 | 2 | 0 | 0 | 6 | 3 | 2 | 1 |
| 0 | 1 | 3 | 1 | 0 | 0 | 0 | 1 | 2 | 2 |
| 0 | 0 | 2 | 2 | 2 | 1 | 10 | 4 | 9 | 0 |
| 1 | 4 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 8 |

Упорядоченная выборка:

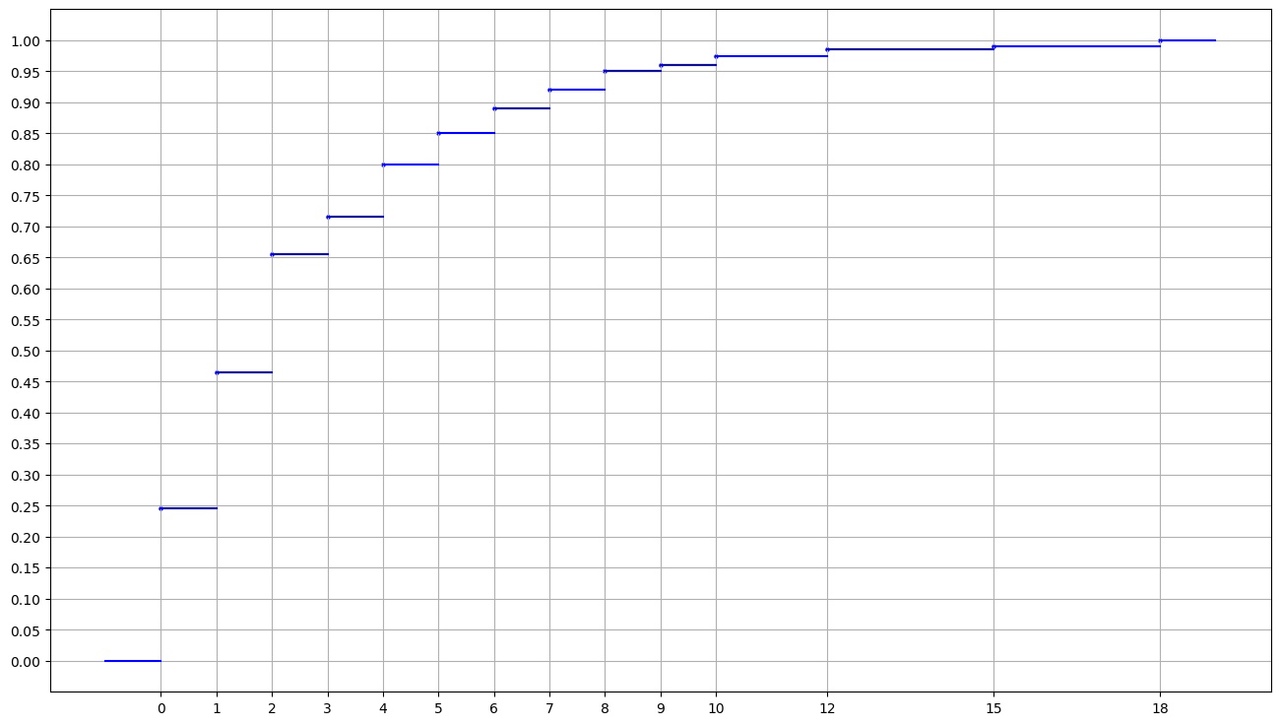
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 7 | 7 |
| 7 | 7 | 7 | 7 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 10 | 10 | 10 | 12 | 12 | 15 | 18 | 18 |

Статистический ряд:

|  |  |  |  |
| --- | --- | --- | --- |
| xi | ni | wi | si |
| 0 | 49 | 0.24500 | 0.24500 |
| 1 | 44 | 0.22000 | 0.46500 |
| 2 | 38 | 0.19000 | 0.65500 |
| 3 | 12 | 0.06000 | 0.71500 |
| 4 | 17 | 0.08500 | 0.80000 |
| 5 | 10 | 0.05000 | 0.85000 |
| 6 | 8 | 0.04000 | 0.89000 |
| 7 | 6 | 0.03000 | 0.92000 |
| 8 | 6 | 0.03000 | 0.95000 |
| 9 | 2 | 0.01000 | 0.96000 |
| 10 | 3 | 0.01500 | 0.97500 |
| 12 | 2 | 0.01000 | 0.98500 |
| 15 | 1 | 0.00500 | 0.99000 |
| 18 | 2 | 0.01000 | 1.00000 |



**Рисунок 3**. Полигон относительных частот задания 2



**Рисунок 4**. Эмпирическая функция распределения задания 2

1. Выборочное среднее: 2.67500
2. Выборочная дисперсия: 9.96922
3. Выборочное среднее квадратичное отклонение: 3.15741
4. Выборочная мода: 0.00000
5. Выборочная медиана: 2.00000
6. Выборочный коэффициент асимметрии: 2.10798
7. Выборочный коэффициент эксцесса: 5.88894

**Геометрическое распределение**

|  |  |  |  |
| --- | --- | --- | --- |
| *j* | *wj* | *pj* | *|wj-pj|* |
| 0 | 0.24500 | 0.27000 | 0.02500 |
| 1 | 0.22000 | 0.19710 | 0.02290 |
| 2 | 0.19000 | 0.14388 | 0.04612 |
| 3 | 0.06000 | 0.10503 | 0.04503 |
| 4 | 0.08500 | 0.07668 | 0.00832 |
| 5 | 0.05000 | 0.05597 | 0.00597 |
| 6 | 0.04000 | 0.04086 | 0.00086 |
| 7 | 0.03000 | 0.02983 | 0.00017 |
| 8 | 0.03000 | 0.02177 | 0.00823 |
| 9 | 0.01000 | 0.01590 | 0.00590 |
| 10 | 0.01500 | 0.01160 | 0.00340 |
| 12 | 0.01000 | 0.00618 | 0.00382 |
| 15 | 0.00500 | 0.00241 | 0.00259 |
| 18 | 0.01000 | 0.00094 | 0.00906 |
|  | 1.00000 | 0.97815 | 0.04612 |

**Геометрическое распределение**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название показателя | Экспериментальное значение | Теоретическое значение | Абсолютное отклонение | Относительное отклонение |
| Выборочное среднее | 2.67500 | 2.70370 | 0.02870 | 1.06717 |
| Выборочная дисперсия | 9.96922 | 10.01372 | 0.0445 | 0.44538 |
| Выборочное среднее квадратическое  отклонение | 3.15741 | 3.16445 | 0.00704 | 0.22272 |
| Выборочная мода | 0.00000 | 1.00000 | 1.00000 | 200.00000 |
| Выборочная медиана | 2.00000 | 2.00000 | 0.00000 | 0.00000 |
| Выборочный коэффициент асимметрии | 2.10798 | 2.02481 | 0.08317 | 4.02488 |
| Выборочный коэффициент эксцесса | 5.88894 | 6.09986 | 0.21092 | 3.51862 |

## Задание 3

*λ* = 0.64

Неупорядоченная выборка:

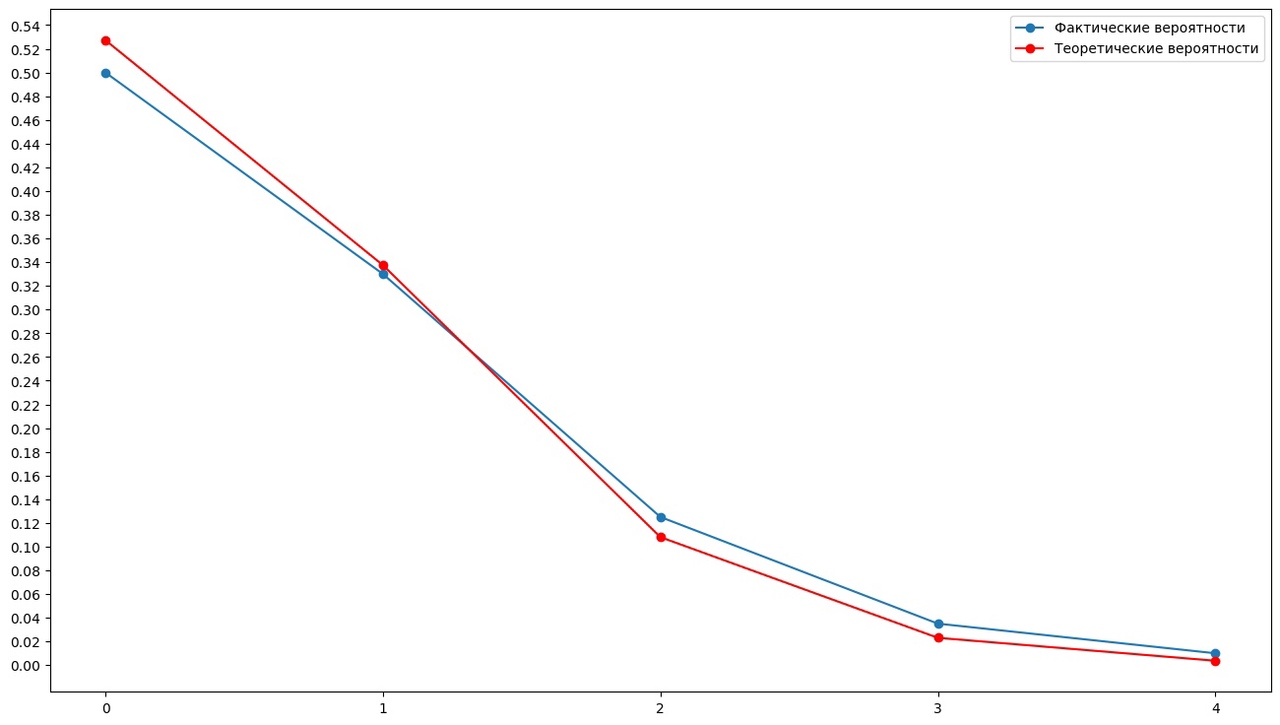
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 2 |
| 0 | 1 | 2 | 1 | 0 | 0 | 0 | 0 | 1 | 3 |
| 0 | 0 | 1 | 0 | 0 | 0 | 3 | 1 | 0 | 2 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 4 | 0 | 0 | 1 | 0 | 0 | 3 | 0 | 0 | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 3 | 0 | 1 |
| 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 1 | 1 | 2 | 1 | 1 | 1 |
| 1 | 0 | 2 | 0 | 0 | 0 | 2 | 0 | 1 | 0 |
| 0 | 2 | 1 | 2 | 2 | 1 | 1 | 0 | 2 | 0 |
| 1 | 0 | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 4 | 0 | 0 | 0 | 1 | 0 |
| 1 | 0 | 0 | 0 | 2 | 1 | 0 | 0 | 2 | 1 |
| 1 | 0 | 1 | 1 | 1 | 3 | 0 | 0 | 2 | 2 |
| 1 | 1 | 0 | 2 | 1 | 3 | 0 | 0 | 0 | 1 |
| 2 | 2 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 2 |
| 1 | 1 | 1 | 2 | 0 | 0 | 0 | 0 | 0 | 1 |
| 3 | 0 | 1 | 0 | 1 | 0 | 1 | 2 | 0 | 2 |
| 2 | 1 | 0 | 0 | 0 | 0 | 1 | 2 | 2 | 0 |
| 0 | 1 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 1 |

Упорядоченная выборка:

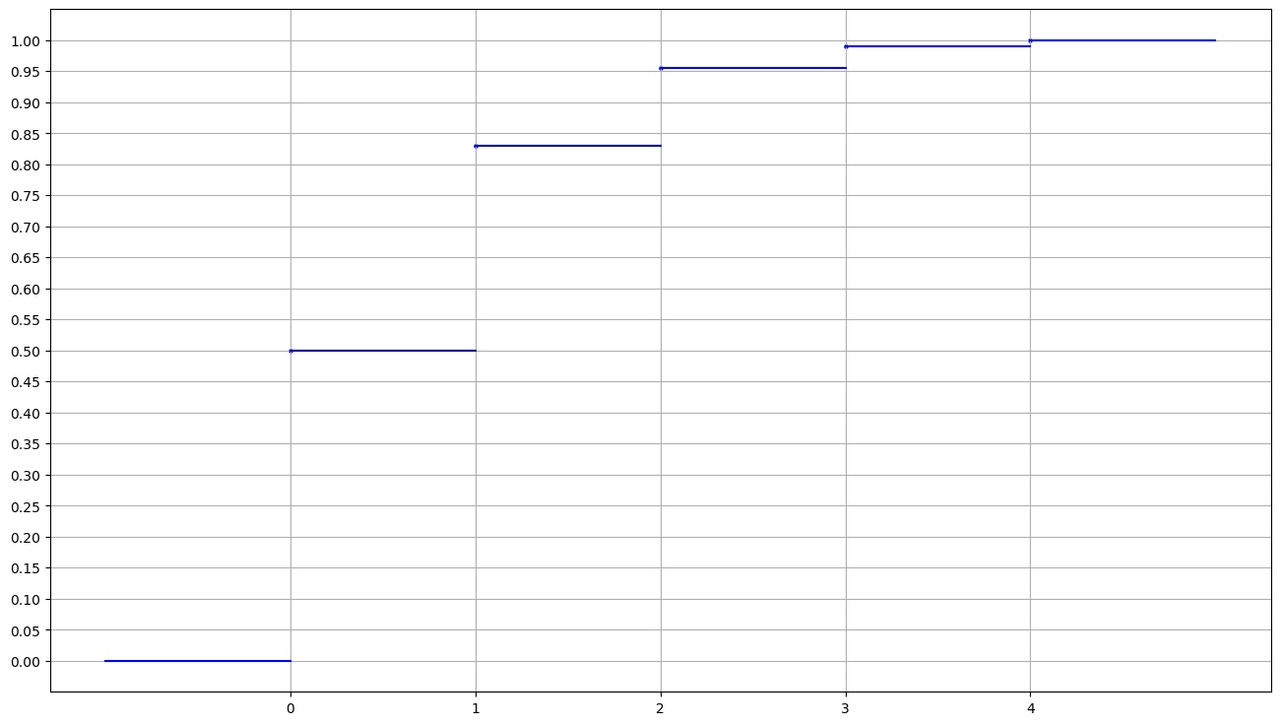
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 | 4 |

Статистический ряд:

|  |  |  |  |
| --- | --- | --- | --- |
| xi | ni | wi | si |
| 0 | 100 | 0.50000 | 0.50000 |
| 1 | 66 | 0.33000 | 0.83000 |
| 2 | 25 | 0.12500 | 0.95500 |
| 3 | 7 | 0.03500 | 0.99000 |
| 4 | 2 | 0.01000 | 1.00000 |



**Рисунок 5.** Полигон относительных частот задания 3



**Рисунок 6**. Эмпирическая функция распределения задания 3

1. Выборочное среднее: 0.72500
2. Выборочная дисперсия: 0.78329
3. Выборочное среднее квадратичное отклонение: 0.88504
4. Выборочная мода: 0.00000
5. Выборочная медиана: 0.50000
6. Выборочный коэффициент асимметрии: 1.22830
7. Выборочный коэффициент эксцесса: 1.27317

**Распределение по закону Пуассона**

|  |  |  |  |
| --- | --- | --- | --- |
| *j* | *wj* | *pj* | *|wj-pj|* |
| 0 | 0.50000 | 0.52729 | 0.02729 |
| 1 | 0.33000 | 0.33747 | 0.00747 |
| 2 | 0.12500 | 0.10799 | 0.01701 |
| 3 | 0.03500 | 0.02304 | 0.01196 |
| 4 | 0.01000 | 0.00369 | 0.00631 |
|  | 1.00000 | 0.99947 | 0.02729 |

**Распределение по закону Пуассона**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название показателя | Экспериментальное значение | Теоретическое значение | Абсолютное отклонение | Относительное отклонение |
| Выборочное среднее | 0.72500 | 0.64000 | 0.08500 | 12.45421 |
| Выборочная дисперсия | 0.78329 | 0.64000 | 0.14329 | 20.13504 |
| Выборочное среднее  квадратическое  отклонение | 0.88504 | 2.96296 | 2.07792 | 108.00000 |
| Выборочная мода | 0.00000 | 0.00000 | 0.00000 | 0.00000 |
| Выборочная медиана | 0.50000 | 0.00000 | 0.50000 | 200.00000 |
| Выборочный коэффициент асимметрии | 1.22830 | 1.25000 | 0.02170 | 1.75120 |
| Выборочный коэффициент эксцесса | 1.27317 | 1.56250 | 0.28933 | 20.40646 |

## Задание 4

*n* = 19

Неупорядоченная выборка:

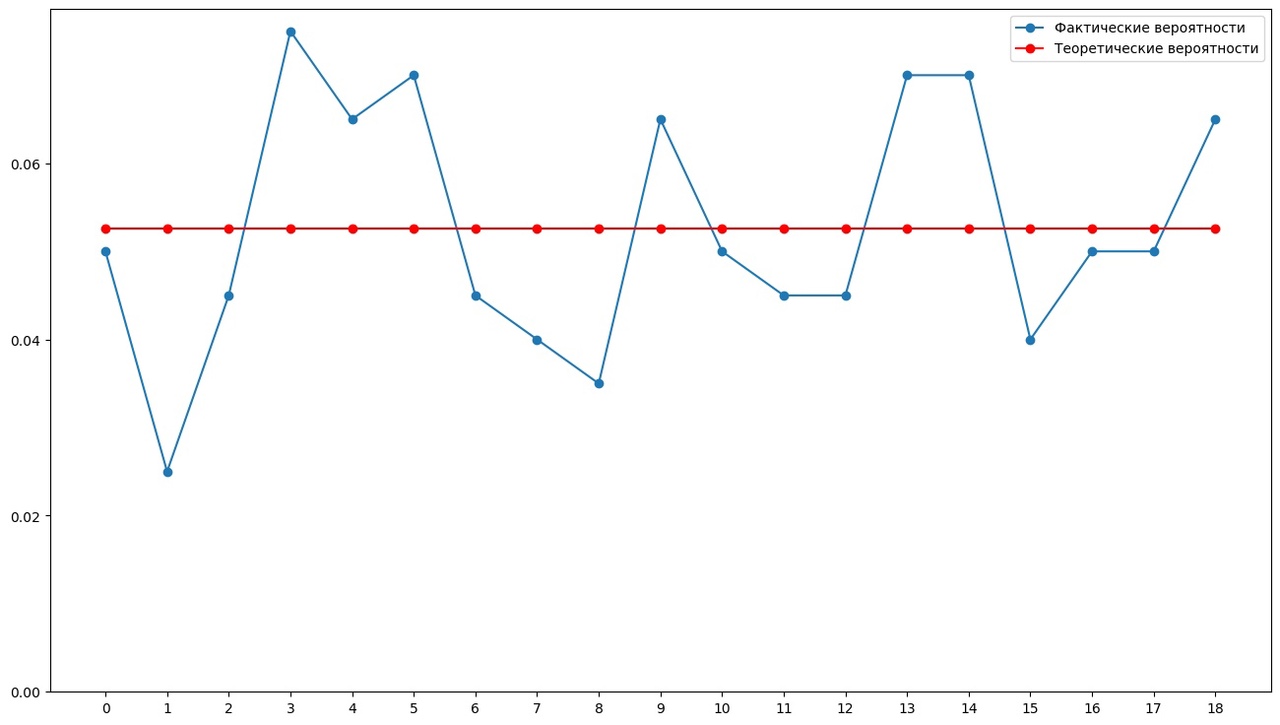
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 2 | 12 | 18 | 12 | 13 | 18 | 12 | 5 | 3 |
| 5 | 6 | 8 | 18 | 6 | 13 | 0 | 7 | 14 | 13 |
| 4 | 10 | 12 | 4 | 6 | 5 | 5 | 9 | 2 | 9 |
| 14 | 7 | 4 | 16 | 10 | 14 | 18 | 16 | 10 | 14 |
| 5 | 14 | 3 | 15 | 15 | 14 | 6 | 14 | 9 | 13 |
| 4 | 6 | 11 | 18 | 1 | 14 | 3 | 4 | 15 | 15 |
| 4 | 18 | 11 | 14 | 17 | 3 | 9 | 7 | 1 | 5 |
| 8 | 12 | 5 | 6 | 2 | 12 | 13 | 18 | 5 | 14 |
| 2 | 3 | 13 | 15 | 15 | 2 | 0 | 4 | 1 | 4 |
| 3 | 6 | 17 | 3 | 5 | 0 | 7 | 7 | 14 | 11 |
| 12 | 4 | 9 | 7 | 0 | 5 | 18 | 8 | 9 | 9 |
| 8 | 0 | 17 | 10 | 0 | 11 | 13 | 10 | 3 | 10 |
| 11 | 8 | 0 | 2 | 9 | 6 | 3 | 15 | 16 | 10 |
| 16 | 17 | 17 | 16 | 4 | 9 | 13 | 13 | 17 | 5 |
| 3 | 6 | 12 | 11 | 10 | 1 | 7 | 0 | 2 | 3 |
| 0 | 14 | 3 | 11 | 9 | 15 | 1 | 13 | 14 | 13 |
| 2 | 7 | 3 | 11 | 2 | 4 | 18 | 13 | 17 | 14 |
| 16 | 16 | 12 | 8 | 17 | 4 | 16 | 16 | 10 | 18 |
| 4 | 8 | 18 | 11 | 13 | 10 | 3 | 0 | 17 | 9 |
| 9 | 18 | 9 | 16 | 18 | 3 | 5 | 17 | 5 | 13 |

Упорядоченная выборка:

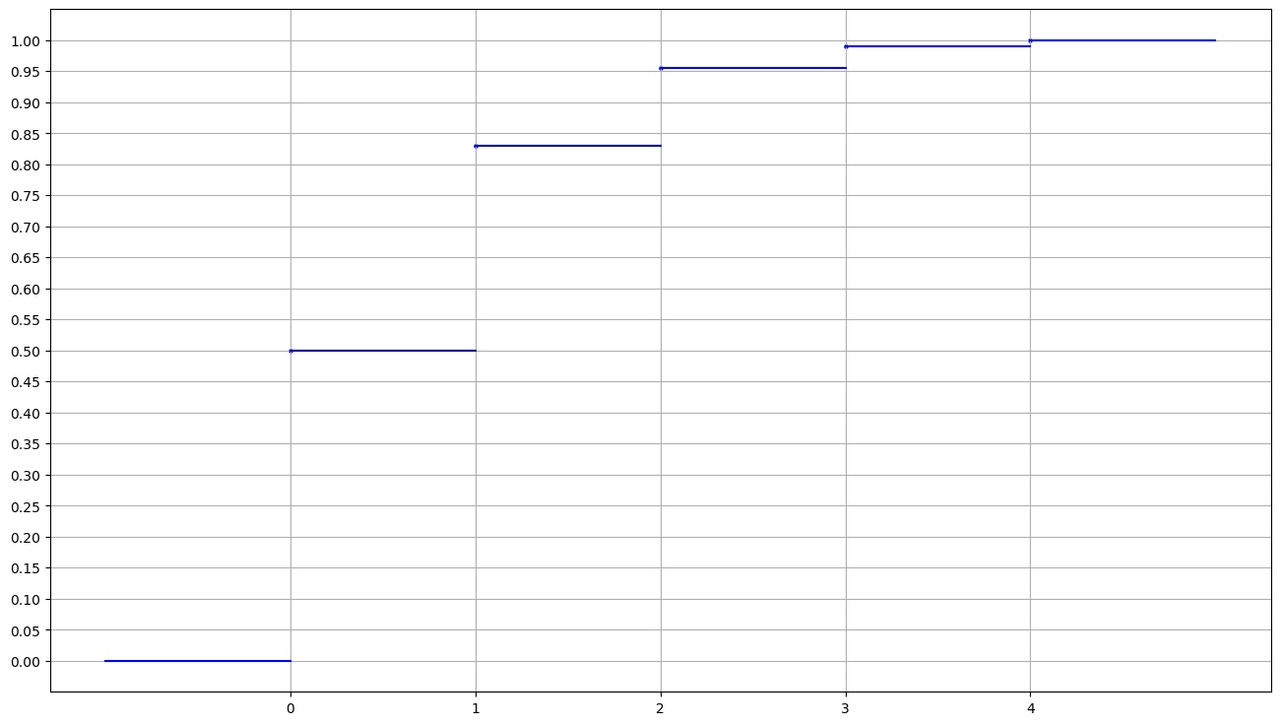
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 5 | 5 | 5 | 5 | 5 | 5 | 6 | 6 | 6 | 6 |
| 6 | 6 | 6 | 6 | 6 | 7 | 7 | 7 | 7 | 7 |
| 7 | 7 | 7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 |
| 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 |
| 9 | 9 | 9 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |
| 10 | 10 | 10 | 11 | 11 | 11 | 11 | 11 | 11 | 11 |
| 11 | 11 | 12 | 12 | 12 | 12 | 12 | 12 | 12 | 12 |
| 12 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 | 13 |
| 13 | 13 | 13 | 13 | 13 | 14 | 14 | 14 | 14 | 14 |
| 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 14 | 15 |
| 15 | 15 | 15 | 15 | 15 | 15 | 15 | 16 | 16 | 16 |
| 16 | 16 | 16 | 16 | 16 | 16 | 16 | 17 | 17 | 17 |
| 17 | 17 | 17 | 17 | 17 | 17 | 17 | 18 | 18 | 18 |
| 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 | 18 |

Статистический ряд:

|  |  |  |  |
| --- | --- | --- | --- |
| xi | ni | wi | si |
| 0 | 10 | 0.05000 | 0.05000 |
| 1 | 5 | 0.02500 | 0.07500 |
| 2 | 9 | 0.04500 | 0.12000 |
| 3 | 15 | 0.07500 | 0.19500 |
| 4 | 13 | 0.06500 | 0.26000 |
| 5 | 14 | 0.07000 | 0.33000 |
| 6 | 9 | 0.04500 | 0.37500 |
| 7 | 8 | 0.04000 | 0.41500 |
| 8 | 7 | 0.03500 | 0.45000 |
| 9 | 13 | 0.06500 | 0.51500 |
| 10 | 10 | 0.05000 | 0.56500 |
| 11 | 9 | 0.04500 | 0.61000 |
| 12 | 9 | 0.04500 | 0.65500 |
| 13 | 14 | 0.07000 | 0.72500 |
| 14 | 14 | 0.07000 | 0.79500 |
| 15 | 8 | 0.04000 | 0.83500 |
| 16 | 10 | 0.05000 | 0.88500 |
| 17 | 10 | 0.05000 | 0.93500 |
| 18 | 13 | 0.06500 | 1.00000 |



**Рисунок 7.** Полигон относительных частот задания 4



**Рисунок 8**. Эмпирическая функция распределения задания 4

1. Выборочное среднее: 9.21000
2. Выборочная дисперсия: 29.86523
3. Выборочное среднее квадратичное отклонение: 5.46491
4. Выборочная мода: 3.00000
5. Выборочная медиана: 9.00000
6. Выборочный коэффициент асимметрии: -0.001
7. Выборочный коэффициент эксцесса: -1.23407

**Равномерное распределение на множестве**

|  |  |  |  |
| --- | --- | --- | --- |
| *j* | *wj* | *pj* | *|wj-pj|* |
| 0 | 0.05000 | 0.05263 | 0.00263 |
| 1 | 0.02500 | 0.05263 | 0.02763 |
| 2 | 0.04500 | 0.05263 | 0.00763 |
| 3 | 0.07500 | 0.05263 | 0.02237 |
| 4 | 0.06500 | 0.05263 | 0.01237 |
| 5 | 0.07000 | 0.05263 | 0.01737 |
| 6 | 0.04500 | 0.05263 | 0.00763 |
| 7 | 0.04000 | 0.05263 | 0.01263 |
| 8 | 0.03500 | 0.05263 | 0.01763 |
| 9 | 0.06500 | 0.05263 | 0.01237 |
| 10 | 0.05000 | 0.05263 | 0.00263 |
| 11 | 0.04500 | 0.05263 | 0.00763 |
| 12 | 0.04500 | 0.05263 | 0.00763 |
| 13 | 0.07000 | 0.05263 | 0.01737 |
| 14 | 0.07000 | 0.05263 | 0.01737 |
| 15 | 0.04000 | 0.05263 | 0.01263 |
| 16 | 0.05000 | 0.05263 | 0.00263 |
| 17 | 0.05000 | 0.05263 | 0.00263 |
| 18 | 0.06500 | 0.05263 | 0.01237 |
|  | 1.00000 | 1.00000 | 0.02763 |

**Равномерное распределение на множестве**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название показателя | Экспериментальное значение | Теоретическое значение | Абсолютное отклонение | Относительное отклонение |
| Выборочное среднее | 9.21000 | 9.00000 | 0.21000 | 2.30643 |
| Выборочная дисперсия | 29.86523 | 30.00000 | 0.13477 | 0.45024 |
| Выборочное среднее квадратичное отклонение | 5.46491 | 5.47723 | 0.01232 | 0.22518 |
| Выборочная мода | 3.00000 | 9.00000 | 6.00000 | 100.00000 |
| Выборочная медиана | 9.00000 | 9.00000 | 0.00000 | 0.00000 |
| Выборочный коэффициент асимметрии | -0.00100 | 0.00000 | 0.00100 | 200.00000 |
| Выборочный коэффициент эксцесса | -1.23407 | -1.20667 | 0.02740 | 2.24522 |

## Задание 5

*m* = 5, *M* = 13, *K* =6

Неупорядоченная выборка:

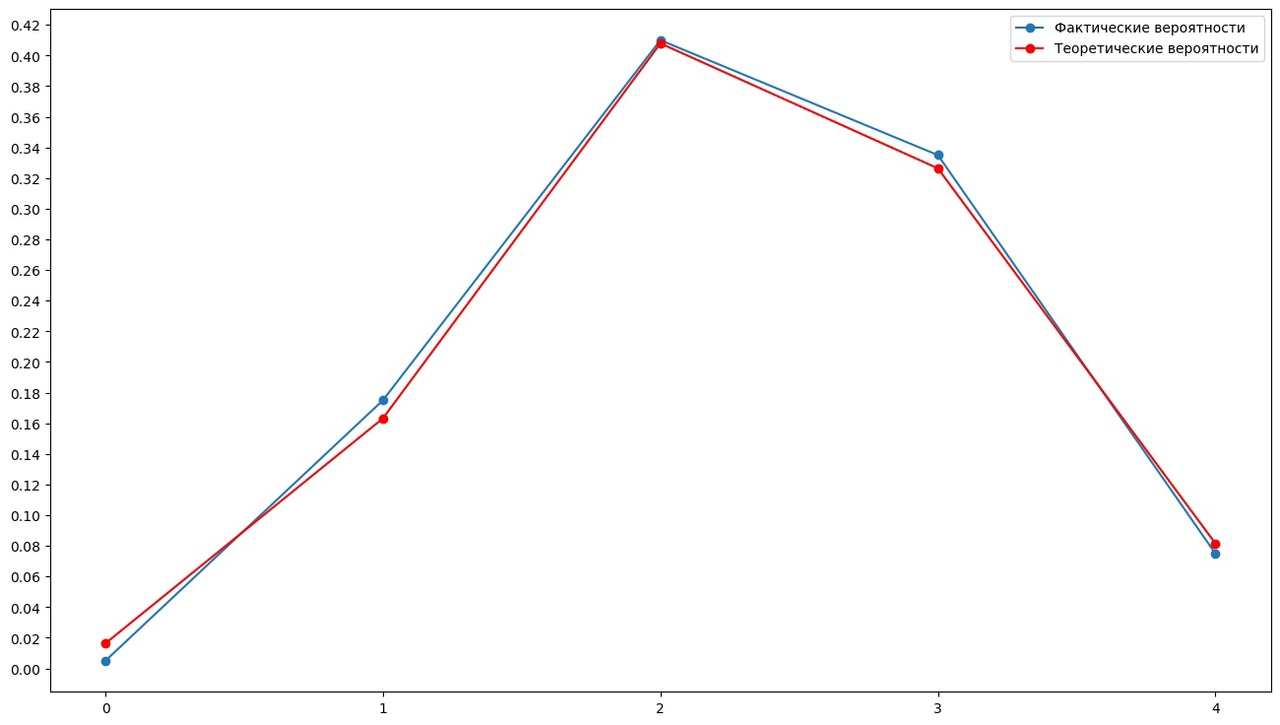
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 2 | 1 | 2 | 2 | 2 | 2 | 3 | 1 | 2 | 3 |
| 3 | 0 | 3 | 2 | 3 | 1 | 2 | 2 | 3 | 1 |
| 2 | 2 | 2 | 1 | 2 | 2 | 3 | 3 | 2 | 2 |
| 2 | 3 | 2 | 1 | 3 | 3 | 3 | 2 | 3 | 1 |
| 1 | 2 | 1 | 2 | 2 | 1 | 2 | 2 | 1 | 3 |
| 2 | 2 | 2 | 1 | 1 | 2 | 1 | 2 | 3 | 2 |
| 3 | 1 | 3 | 3 | 4 | 4 | 2 | 3 | 3 | 3 |
| 2 | 3 | 2 | 2 | 3 | 2 | 1 | 2 | 3 | 3 |
| 2 | 3 | 2 | 3 | 3 | 1 | 1 | 3 | 3 | 2 |
| 3 | 2 | 3 | 1 | 3 | 1 | 3 | 2 | 3 | 3 |
| 1 | 2 | 1 | 2 | 3 | 3 | 1 | 3 | 2 | 4 |
| 2 | 2 | 2 | 2 | 3 | 2 | 3 | 2 | 3 | 3 |
| 4 | 2 | 4 | 3 | 3 | 2 | 2 | 2 | 2 | 3 |
| 3 | 3 | 3 | 1 | 2 | 3 | 3 | 1 | 2 | 3 |
| 4 | 3 | 4 | 2 | 2 | 2 | 3 | 4 | 4 | 1 |
| 3 | 3 | 3 | 2 | 4 | 3 | 1 | 4 | 4 | 2 |
| 3 | 2 | 3 | 3 | 4 | 2 | 2 | 3 | 3 | 3 |
| 2 | 2 | 2 | 1 | 2 | 3 | 3 | 3 | 2 | 2 |
| 3 | 1 | 3 | 3 | 2 | 1 | 1 | 1 | 4 | 3 |
| 1 | 2 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 3 |

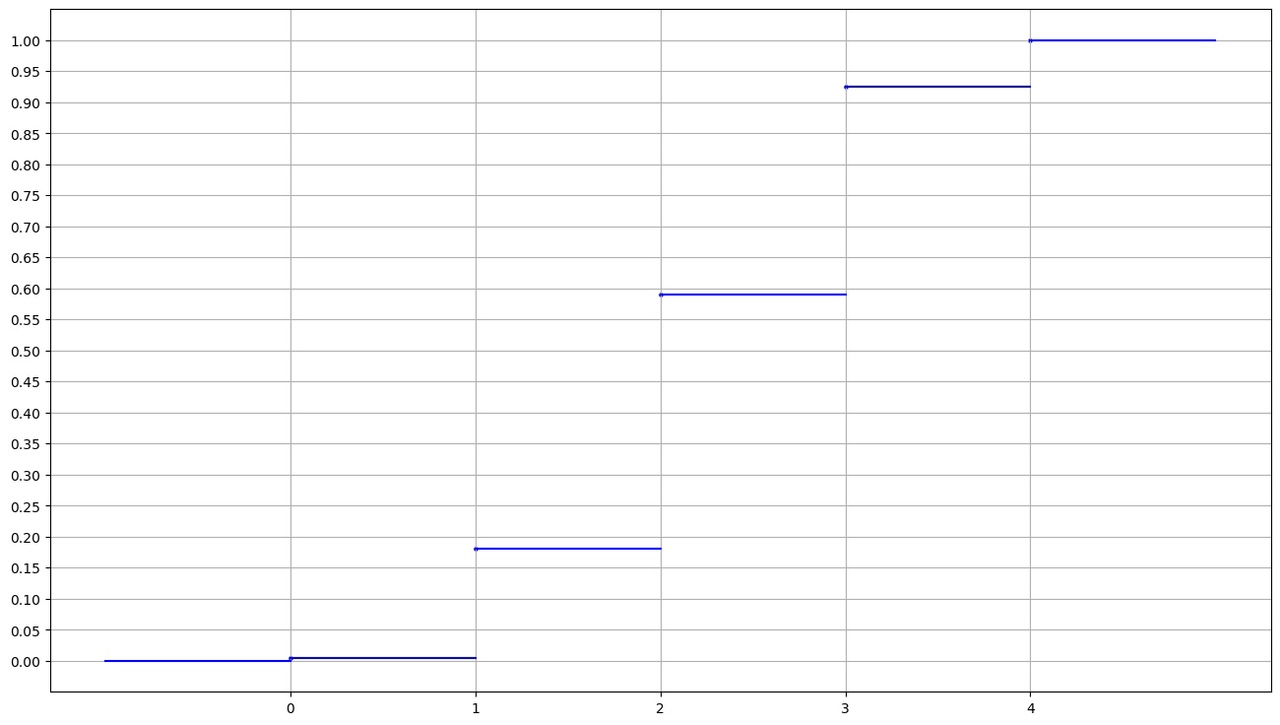
Упорядоченная выборка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 3 | 3 | 3 | 3 | 3 | 4 | 4 | 4 | 4 | 4 |
| 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |

Статистический ряд:

|  |  |  |  |
| --- | --- | --- | --- |
| xi | ni | wi | si |
| 0 | 1 | 0.00500 | 0.00500 |
| 1 | 35 | 0.17500 | 0.18000 |
| 2 | 82 | 0.41000 | 0.59000 |
| 3 | 67 | 0.33500 | 0.92500 |
| 4 | 15 | 0.07500 | 1.00000 |

**Рисунок 9.** Полигон относительных частот задания 5



**Рисунок 10.** Эмпирическая функция распределения задания 5

1. Выборочное среднее: 2.30000
2. Выборочная дисперсия с поправкой Шеппарда: 0.74372
3. Выборочное среднее квадратичное отклонение: 0.86239
4. Выборочная мода: 2.00000
5. Выборочная медиана: 2.00000
6. Выборочный коэффициент асимметрии: 0.04274
7. Выборочный коэффициент эксцесса: -0.51789

**Гипергеометрическое распределение**

|  |  |  |  |
| --- | --- | --- | --- |
| *j* | *wj* | *pj* | *|wj-pj|* |
| 0 | 0.00500 | 0.01632 | 0.01132 |
| 1 | 0.17500 | 0.16317 | 0.01183 |
| 2 | 0.41000 | 0.40793 | 0.00207 |
| 3 | 0.33500 | 0.32634 | 0.00866 |
| 4 | 0.34500 | 0.08159 | 0.00659 |
|  | 1 | 0.99814 | 0.01183 |

**Гипергеометрическое распределение**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название показателя | Экспериментальное значение | Теоретическое значение | Абсолютное отклонение | Относительное отклонение |
| Выборочное среднее | 2.30000 | 2.30769 | 0.00769 | 0.33379 |
| Выборочная дисперсия | 0.74372 | 0.8284 | 0.08468 | 10.77271 |
| Выборочное среднее квадратическое отклонение | 0.86239 | 0.91017 | 0.04778 | 5.39107 |
| Выборочная мода | 2.00000 | 3.00000 | 1.00000 | 40.00000 |
| Выборочная медиана | 2.00000 | 3.00000 | 1.00000 | 40.00000 |
| Выборочный коэффициент асимметрии | 0.04274 | 0.02305 | 0.01969 | 59.85712 |
| Выборочный коэффициент эксцесса | -0.51789 | 0.00000 | 0.51789 | 200.00000 |

## Задание 6

*a* = 0.7, *σ* = 1.07

Неупорядоченная выборка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.98697 | 0.2574 | 1.66759 | 0.85293 | 3.32035 | 1.12325 | 0.49074 | 1.59119 | -0.25468 | 1.65632 |
| 0.0129 | 0.15728 | -2.33536 | 2.59483 | 0.50619 | 0.79187 | 1.93871 | 1.69123 | 0.73056 | -0.71495 |
| -0.02446 | 1.78826 | 1.72136 | -0.57907 | 1.45402 | 2.34465 | -0.89014 | -0.12409 | 2.13068 | 1.26005 |
| 0.81653 | 1.6494 | -1.15915 | 0.10952 | 0.36803 | 1.85491 | 0.76883 | 0.25624 | 0.85037 | 0.25642 |
| 2.39921 | 0.7909 | -0.61306 | -0.19682 | 1.21724 | 0.05721 | 0.17613 | 0.23835 | 1.6097 | 2.30574 |
| 2.04282 | -0.37165 | 1.12699 | 0.66567 | -1.24357 | -0.03316 | 1.29239 | 0.3549 | 0.68548 | -1.18335 |
| 0.53952 | 0.83745 | 1.22408 | 2.1632 | -0.09351 | 0.11671 | 0.10773 | 2.32947 | 0.37308 | -0.26419 |
| -1.43348 | -0.95199 | 0.44123 | 1.35158 | 0.63966 | -0.60054 | 2.51774 | 1.97106 | -0.36591 | -0.55112 |
| 0.08786 | -0.40938 | 0.32616 | 1.11886 | -0.6871 | 0.47099 | -0.39396 | 2.31883 | 0.66268 | 0.15935 |
| 1.49633 | 1.06918 | 0.73513 | 0.70759 | 2.0405 | 1.73659 | -1.20957 | 0.80574 | 1.50458 | -0.2419 |
| 1.44817 | -0.68251 | -0.55741 | 1.35811 | 0.02875 | 0.08007 | 2.3271 | -1.01107 | 1.87644 | 0.69548 |
| -1.0169 | 1.56568 | 1.78893 | 0.50923 | -0.55239 | -0.4678 | 0.99259 | 0.82317 | -0.13346 | 0.81421 |
| -0.21281 | -0.97659 | 0.21171 | 0.71814 | 1.226 | 0.36386 | -0.40422 | 1.65762 | 0.2203 | 1.90077 |
| 0.85715 | 1.77768 | 0.8855 | -0.93908 | 2.93434 | 1.43228 | -0.66474 | 0.21736 | 1.37739 | 1.05477 |
| 1.33834 | 0.33059 | 1.20745 | 0.18506 | 0.66957 | -0.8251 | 0.95238 | -0.61135 | 1.37343 | 0.26155 |
| 1.39288 | 0.36123 | 2.16016 | 0.32999 | 1.18765 | -0.43535 | 0.14793 | 1.75382 | 1.75012 | 0.05092 |
| 1.3489 | -0.5904 | -0.14204 | 1.43565 | -0.91272 | -1.09161 | -0.50353 | 1.46643 | -1.18118 | 1.75573 |
| 1.34175 | 3.09509 | 0.82979 | -0.00975 | 0.95938 | 1.62747 | 1.193 | 0.42263 | 2.42152 | 0.47459 |
| 0.98842 | 0.78026 | 1.01693 | 0.60571 | 0.67302 | 1.14982 | 0.62876 | 0.85523 | 0.97501 | 0.83066 |
| 1.01929 | 0.93999 | 2.76005 | 0.22414 | 1.06242 | 2.83877 | 0.48937 | 1.05917 | 2.33863 | -1.09579 |

Упорядоченная выборка:

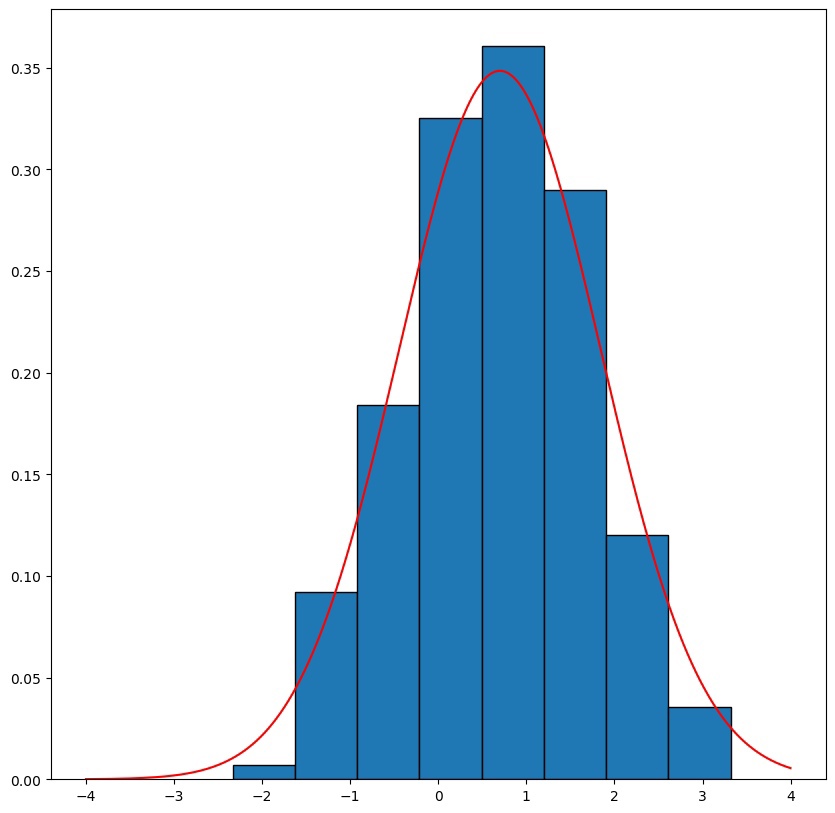
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -2.33536 | -1.43348 | -1.24357 | -1.20957 | -1.18335 | -1.18118 | -1.15915 | -1.09579 | -1.09161 | -1.0169 |
| -1.01107 | -0.97659 | -0.95199 | -0.93908 | -0.91272 | -0.89014 | -0.8251 | -0.71495 | -0.6871 | -0.68251 |
| -0.66474 | -0.61306 | -0.61135 | -0.60054 | -0.5904 | -0.57907 | -0.55741 | -0.55239 | -0.55112 | -0.50353 |
| -0.4678 | -0.43535 | -0.40938 | -0.40422 | -0.39396 | -0.37165 | -0.36591 | -0.26419 | -0.25468 | -0.2419 |
| -0.21281 | -0.19682 | -0.14204 | -0.13346 | -0.12409 | -0.09351 | -0.03316 | -0.02446 | -0.00975 | 0.0129 |
| 0.02875 | 0.05092 | 0.05721 | 0.08007 | 0.08786 | 0.10773 | 0.10952 | 0.11671 | 0.14793 | 0.15728 |
| 0.15935 | 0.17613 | 0.18506 | 0.21171 | 0.21736 | 0.2203 | 0.22414 | 0.23835 | 0.25624 | 0.25642 |
| 0.2574 | 0.26155 | 0.32616 | 0.32999 | 0.33059 | 0.3549 | 0.36123 | 0.36386 | 0.36803 | 0.37308 |
| 0.42263 | 0.44123 | 0.47099 | 0.47459 | 0.48937 | 0.49074 | 0.50619 | 0.50923 | 0.53952 | 0.60571 |
| 0.62876 | 0.63966 | 0.66268 | 0.66567 | 0.66957 | 0.67302 | 0.68548 | 0.69548 | 0.70759 | 0.71814 |
| 0.73056 | 0.73513 | 0.76883 | 0.78026 | 0.7909 | 0.79187 | 0.80574 | 0.81421 | 0.81653 | 0.82317 |
| 0.82979 | 0.83066 | 0.83745 | 0.85037 | 0.85293 | 0.85523 | 0.85715 | 0.8855 | 0.93999 | 0.95238 |
| 0.95938 | 0.97501 | 0.98697 | 0.98842 | 0.99259 | 1.01693 | 1.01929 | 1.05477 | 1.05917 | 1.06242 |
| 1.06918 | 1.11886 | 1.12325 | 1.12699 | 1.14982 | 1.18765 | 1.193 | 1.20745 | 1.21724 | 1.22408 |
| 1.226 | 1.26005 | 1.29239 | 1.33834 | 1.34175 | 1.3489 | 1.35158 | 1.35811 | 1.37343 | 1.37739 |
| 1.39288 | 1.43228 | 1.43565 | 1.44817 | 1.45402 | 1.46643 | 1.49633 | 1.50458 | 1.56568 | 1.59119 |
| 1.6097 | 1.62747 | 1.6494 | 1.65632 | 1.65762 | 1.66759 | 1.69123 | 1.72136 | 1.73659 | 1.75012 |
| 1.75382 | 1.75573 | 1.77768 | 1.78826 | 1.78893 | 1.85491 | 1.87644 | 1.90077 | 1.93871 | 1.97106 |
| 2.0405 | 2.04282 | 2.13068 | 2.16016 | 2.1632 | 2.30574 | 2.31883 | 2.3271 | 2.32947 | 2.33863 |
| 2.34465 | 2.39921 | 2.42152 | 2.51774 | 2.59483 | 2.76005 | 2.83877 | 2.93434 | 3.09509 | 3.32035 |

Интервальный ряд:

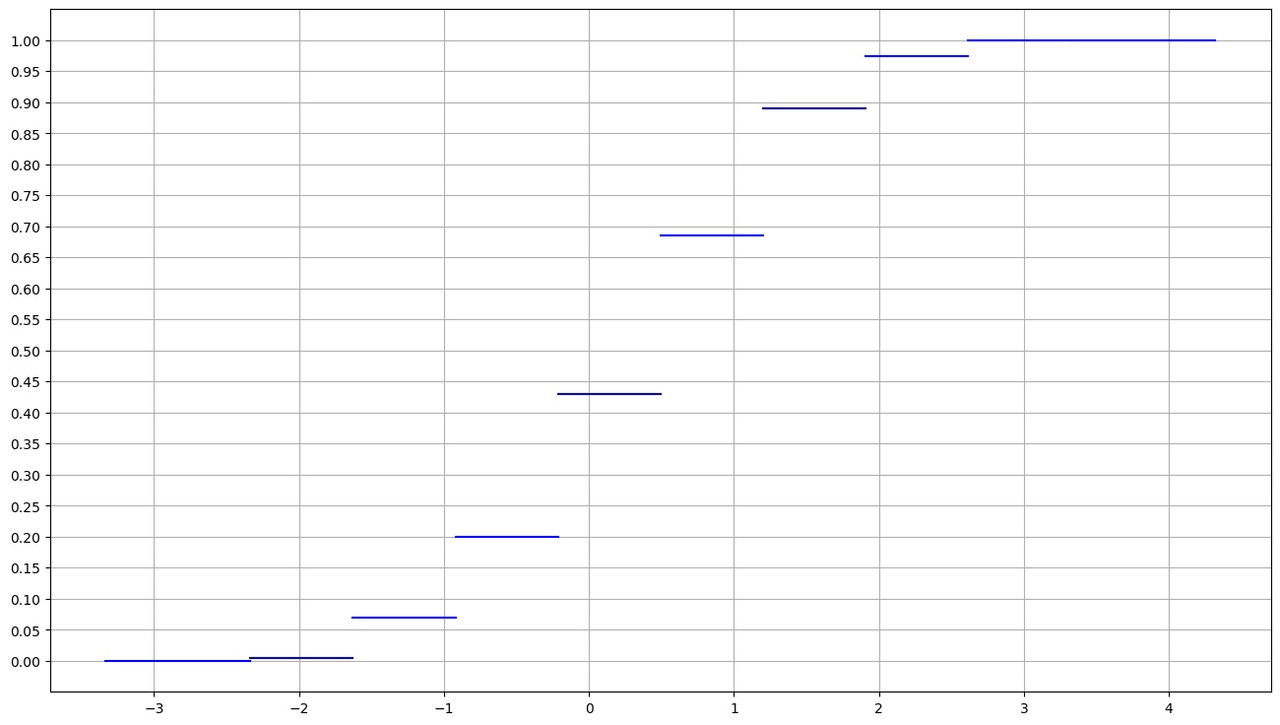
|  |  |  |
| --- | --- | --- |
| Интервалы | *ni* | *wi* |
| [-2.33536, -1.62840] | 1 | 0.00500 |
| (-1.62840, -0.92143] | 13 | 0.06500 |
| (-0.92143, -0.21447] | 26 | 0.13000 |
| (-0.21447, 0.49249] | 46 | 0.23000 |
| (0.49249, 1.19946] | 51 | 0.25500 |
| (1.19946, 1.90642] | 41 | 0.20500 |
| (1.90642, 2.61339] | 17 | 0.08500 |
| (2.61339, 3.32035] | 5 | 0.02500 |
|  | 200 | 1.00000 |

Ассоциированный статистический ряд:

|  |  |  |
| --- | --- | --- |
| *xi\** | *ni* | *wi* |
| -1.98188 | 1 | 0.00500 |
| -1.27492 | 13 | 0.06500 |
| -0.56795 | 26 | 0.13000 |
| 0.13901 | 46 | 0.23000 |
| 0.845975 | 51 | 0.25500 |
| 1.55294 | 41 | 0.20500 |
| 2.259905 | 17 | 0.08500 |
| 2.96687 | 5 | 0.02500 |
|  | 200 | 1.00000 |



**Рисунок 11.** Гистограмма относительных частот задания 6



**Рисунок 12.** График эмпирической функции распределения задания 6

1. Выборочное среднее: 0.68537
2. Выборочная дисперсия с поправкой Шеппарда: 1.01698
3. Выборочное среднее квадратичное отклонение: 1.00845
4. Выборочная мода: 0.54231
5. Выборочная медиана: 0.72435
6. Выборочный коэффициент асимметрии: -0.01412
7. Выборочный коэффициент эксцесса: 0.28606

**Распределение по нормальному закону**

|  |  |  |  |
| --- | --- | --- | --- |
| Интервал | *wi* | *pi* | |*wi-pi*| |
| [-2.33536, -1.62840] | 0.00500 | 0.01250 | 0.00750 |
| (-1.62840, -0.92143] | 0.06500 | 0.05007 | 0.01493 |
| (-0.92143, -0.21447] | 0.13000 | 0.13153 | 0.00153 |
| (-0.21447, 0.49249] | 0.23000 | 0.22674 | 0.00326 |
| (0.49249, 1.19946] | 0.25500 | 0.25656 | 0.00156 |
| (1.19946, 1.90642] | 0.20500 | 0.19056 | 0.01444 |
| (1.90642, 2.61339] | 0.08500 | 0.09290 | 0.00079 |
| (2.61339, 3.32035] | 0.02500 | 0.02971 | 0.00471 |
|  | 1.00000 | 0.99057 | 0.01493 |

**Распределение по нормальному закону**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название  показателя | Экспериментальное значение | Теоретическое значение | Абсолютное  отклонение | Относительное отклонение |
| Выборочное среднее | 0.68537 | 0.70000 | 0.01463 | 2.11207 |
| Выборочная дисперсия | 1.01698 | 1.14490 | 0.12792 | 11.83414 |
| Выборочное среднее квадратичное отклонение | 1.00845 | 1.07000 | 0.06155 | 5.92268 |
| Выборочная мода | 0.54231 | 0.70000 | 0.15769 | 35.38658 |
| Выборочная медиана | 0.72435 | 0.70000 | 0.02435 | 3.41910 |
| Выборочный коэффициент асимметрии | -0.01412 | 0.00000 | 0.01412 | - |
| Выборочный коэффициент эксцесса | -0.28606 | 0.00000 | 0.28606 | - |

## Задание 7

λ = 2.14

Неупорядоченная выборка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1.58505 | 0.71414 | 0.24766 | 0.85571 | 0.01687 | 0.43679 | 0.94881 | 0.2524 | 0.31764 | 0.04354 |
| 0.56375 | 0.57853 | 0.31187 | 1.31242 | 0.00592 | 0.18868 | 0.57159 | 0.59534 | 0.08569 | 0.10332 |
| 0.5626 | 0.20095 | 0.0362 | 0.44006 | 0.51485 | 0.97486 | 0.23925 | 0.07965 | 0.14684 | 0.86903 |
| 0.08603 | 0.57425 | 0.80117 | 0.18362 | 0.00478 | 0.06364 | 0.11039 | 0.4568 | 0.07485 | 2.46183 |
| 0.57305 | 0.78218 | 0.08871 | 0.6097 | 0.04526 | 0.68527 | 0.26473 | 0.10919 | 0.00772 | 1.3151 |
| 0.62259 | 0.46606 | 0.3953 | 0.26164 | 0.17999 | 0.88624 | 1.40704 | 0.01355 | 0.35757 | 0.39344 |
| 0.05733 | 0.80602 | 0.16224 | 0.12523 | 0.90213 | 0.93052 | 1.22004 | 0.15813 | 0.16784 | 0.54336 |
| 0.4584 | 1.02906 | 1.09223 | 0.18519 | 0.25021 | 0.17435 | 0.26158 | 0.02514 | 0.44221 | 0.32141 |
| 1.96432 | 0.57469 | 0.8587 | 0.6299 | 0.41165 | 0.08963 | 0.14839 | 0.2441 | 0.10676 | 0.4349 |
| 0.21585 | 0.9095 | 0.32526 | 0.21272 | 0.11334 | 0.12498 | 0.74667 | 0.15309 | 0.47806 | 0.30614 |
| 0.47391 | 0.4564 | 0.73006 | 0.96805 | 0.9117 | 0.02396 | 0.38055 | 0.49626 | 0.40379 | 0.4755 |
| 1.52318 | 0.45795 | 0.44244 | 0.1623 | 1.06213 | 0.68562 | 0.37681 | 0.47321 | 0.40945 | 0.3798 |
| 1.38226 | 0.33944 | 2.19111 | 0.05395 | 0.75676 | 0.93484 | 0.1982 | 0.24654 | 0.34657 | 0.70378 |
| 0.44281 | 0.23832 | 1.8109 | 0.85019 | 1.30773 | 0.55026 | 0.36593 | 0.5416 | 0.67609 | 0.2163 |
| 0.40682 | 0.80178 | 0.66478 | 0.21004 | 2.0699 | 0.10536 | 0.529 | 0.46802 | 0.56963 | 0.60706 |
| 0.2508 | 0.22465 | 0.37869 | 1.0225 | 0.02161 | 0.45624 | 0.15721 | 0.33036 | 0.56594 | 0.60386 |
| 0.14936 | 0.9708 | 0.55468 | 0.29384 | 0.32782 | 0.2946 | 0.61464 | 0.47274 | 0.30418 | 0.44419 |
| 0.20028 | 0.07023 | 0.2851, | 0.00928 | 0.51861 | 0.76508 | 0.13429 | 0.8152, | 0.95521 | 0.48052 |
| 0.62227 | 0.34717 | 0.62301 | 1.0895 | 0.14686 | 0.11586 | 0.6323 | 0.10109 | 0.43013 | 0.99823 |
| 0.11181 | 0.35232 | 0.44474 | 0.62914 | 0.47597 | 0.4775 | 0.19323 | 0.07592 | 0.19759 | 0.2249 |

Упорядоченная выборка:

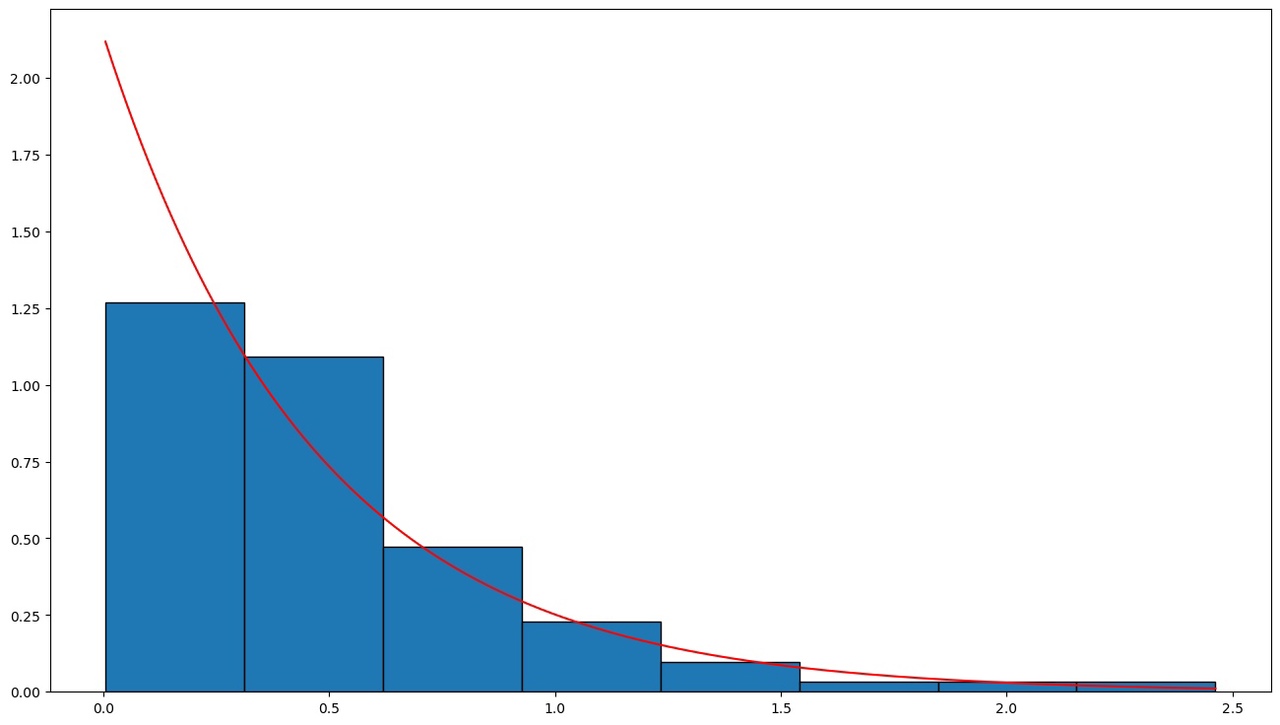
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.00478 | 0.00592 | 0.00772 | 0.00928 | 0.01355 | 0.01687 | 0.02161 | 0.02396 | 0.02514 | 0.0362 |
| 0.04354 | 0.04526 | 0.05395 | 0.05733 | 0.06364 | 0.07023 | 0.07485 | 0.07592 | 0.07965 | 0.08569 |
| 0.08603 | 0.08871 | 0.08963 | 0.10109 | 0.10332 | 0.10536 | 0.10676 | 0.10919 | 0.11039 | 0.11181 |
| 0.11334 | 0.11586 | 0.12498 | 0.12523 | 0.13429 | 0.14684 | 0.14686 | 0.14839 | 0.14936 | 0.15309 |
| 0.15721 | 0.15813 | 0.16224 | 0.1623 | 0.16784 | 0.17435 | 0.17999 | 0.18362 | 0.18519 | 0.18868 |
| 0.19323 | 0.19759 | 0.1982 | 0.20028 | 0.20095 | 0.21004 | 0.21272 | 0.21585 | 0.2163 | 0.22465 |
| 0.2249 | 0.23832 | 0.23925 | 0.2441 | 0.24654 | 0.24766 | 0.25021 | 0.2508 | 0.2524 | 0.26158 |
| 0.26164 | 0.26473 | 0.2851 | 0.29384 | 0.2946 | 0.30418 | 0.30614 | 0.31187 | 0.31764 | 0.32141 |
| 0.32526 | 0.32782 | 0.33036 | 0.33944 | 0.34657 | 0.34717 | 0.35232 | 0.35757 | 0.36593 | 0.37681 |
| 0.37869 | 0.3798 | 0.38055 | 0.39344 | 0.3953 | 0.40379 | 0.40682 | 0.40945 | 0.41165 | 0.43013 |
| 0.4349 | 0.43679 | 0.44006 | 0.44221 | 0.44244 | 0.44281 | 0.44419 | 0.44474 | 0.45624 | 0.4564 |
| 0.4568 | 0.45795 | 0.4584 | 0.46606 | 0.46802 | 0.47274 | 0.47321 | 0.47391 | 0.4755 | 0.47597 |
| 0.4775 | 0.47806 | 0.48052 | 0.49626 | 0.51485 | 0.51861 | 0.529 | 0.5416 | 0.54336 | 0.55026 |
| 0.55468 | 0.5626 | 0.56375 | 0.56594 | 0.56963 | 0.57159 | 0.57305 | 0.57425 | 0.57469 | 0.57853 |
| 0.59534 | 0.60386 | 0.60706 | 0.6097 | 0.61464 | 0.62227 | 0.62259 | 0.62301 | 0.62914 | 0.6299 |
| 0.6323 | 0.66478 | 0.67609 | 0.68527 | 0.68562 | 0.70378 | 0.71414 | 0.73006 | 0.74667 | 0.75676 |
| 0.76508 | 0.78218 | 0.80117 | 0.80178 | 0.80602 | 0.8152 | 0.85019 | 0.85571 | 0.8587 | 0.86903 |
| 0.88624 | 0.90213 | 0.9095 | 0.9117 | 0.93052 | 0.93484 | 0.94881 | 0.95521 | 0.96805 | 0.9708 |
| 0.97486 | 0.99823 | 1.0225 | 1.02906 | 1.06213 | 1.0895 | 1.09223 | 1.22004 | 1.30773 | 1.31242 |
| 1.3151 | 1.38226 | 1.40704 | 1.52318 | 1.58505 | 1.8109 | 1.96432 | 2.0699 | 2.19111 | 2.46183 |

Интервальный ряд:

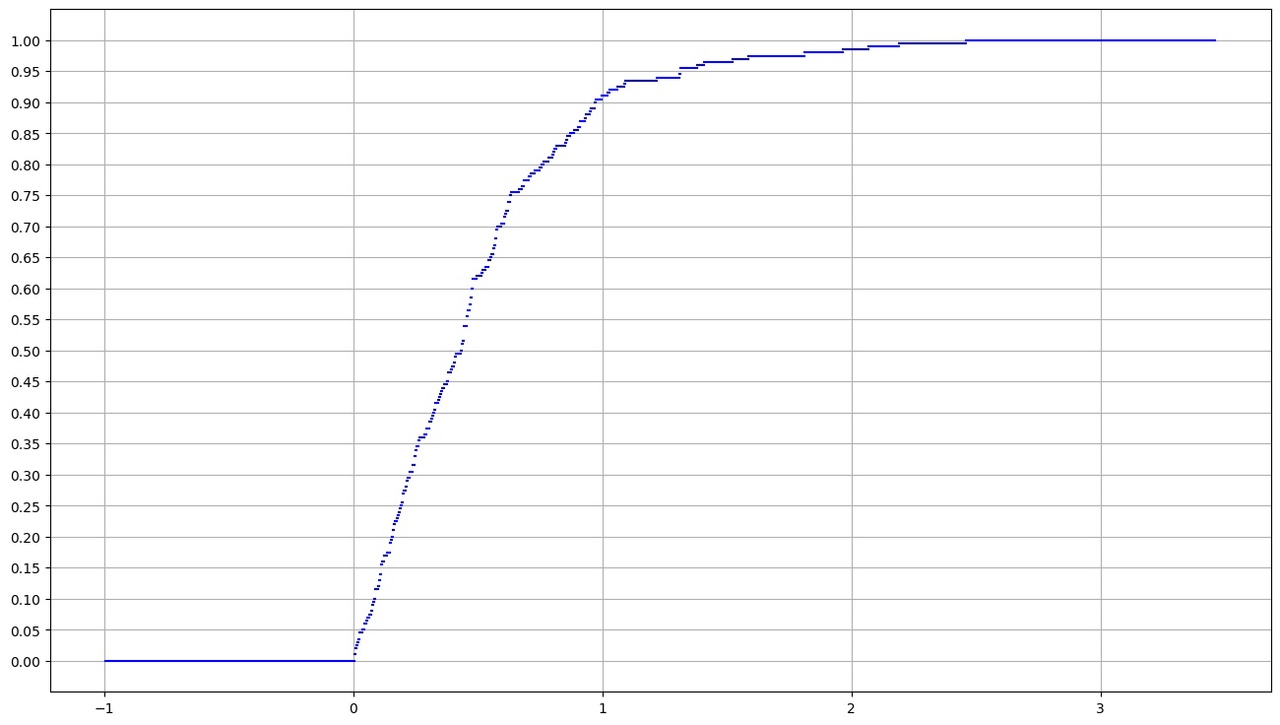
|  |  |  |
| --- | --- | --- |
| Интервалы | *ni* | *wi* |
| [0.00000, 0.30773] | 77 | 0.38500 |
| (0.30773, 0.61546] | 68 | 0.34000 |
| (0.61546, 0.92319] | 29 | 0.14500 |
| (0.92319, 1.230915] | 14 | 0.07000 |
| (1.230915, 1.53864] | 6 | 0.03000 |
| (1.53864, 1.84637] | 2 | 0.01000 |
| (1.84637, 2.15410] | 2 | 0.01000 |
| (2.15410, 2.46183] | 2 | 0.01000 |
|  | 200 | 1.00000 |

Ассоциированный статистический ряд:

|  |  |  |
| --- | --- | --- |
| *xi\** | *ni* | *wi* |
| 0.153865 | 77 | 0.38500 |
| 0.461595 | 68 | 0.34000 |
| 0.769325 | 29 | 0.14500 |
| 1.077055 | 14 | 0.07000 |
| 1.384785 | 6 | 0.03000 |
| 1.692515 | 2 | 0.01000 |
| 2.000245 | 2 | 0.01000 |
| 2.307975 | 2 | 0.01000 |
|  | 200 | 1.00000 |



**Рисунок 13.** Гистограмма относительных частот задания 7



**Рисунок 14.** График эмпирической функции распределения задания 7

1. Выборочное среднее: 0.49542
2. Выборочная дисперсия с поправкой Шеппарда: 0.17212
3. Выборочное среднее квадратичное отклонение: 0.41487
4. Выборочная мода: -
5. Выборочная медиана: 0.432515
6. Выборочный коэффициент асимметрии: 1.76714
7. Выборочный коэффициент эксцесса: 4.30184

**Распределение по показательному закону**

|  |  |  |  |
| --- | --- | --- | --- |
| Интервал | *wi* | *pi* | |*wi-pi*| |
| [0.00000, 0.30773] | 0.38500 | 0.48239 | 0.09739 |
| (0.30773, 0.61546] | 0.34000 | 0.24969 | 0.09031 |
| (0.61546, 0.92319] | 0.14500 | 0.12924 | 0.01576 |
| (0.92319, 1.230915] | 0.07000 | 0.06690 | 0.00310 |
| (1.230915, 1.53864] | 0.03000 | 0.03463 | 0.00463 |
| (1.53864, 1.84637] | 0.01000 | 0.01792 | 0.00792 |
| (1.84637, 2.15410] | 0.01000 | 0.00928 | 0.00072 |
| (2.15410, 2.46183] | 0.01000 | 0.00480 | 0.00520 |
|  | 1.00000 | 0.99485 | 0.09739 |

**Распределение по показательному закону**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название  показателя | Экспериментальное  значение | Теоретическое значение | Абсолютное отклонение | Относительное  отклонение |
| Выборочное среднее | 0.49642 | 0.46729 | 0.02913 | 6.04539 |
| Выборочная дисперсия | 0.17212 | 0.21836 | 0.04624 | 23.68367 |
| Выборочное среднее  квадратичное отклонение | 0.41487 | 0.46729 | 0.05242 | 11.88447 |
| Выборочная мода | - | 0.00000 | - | - |
| Выборочная медиана | 0.432515 | 0.32390 | 0.10861 | 28.71723 |
| Выборочный коэффициент асимметрии | 1.76714 | 2.00000 | 0..23286 | 12.36269 |
| Выборочный коэффициент эксцесса | 4.30184 | 6.00000 | 1.69816 | 32.96809 |

## Задание 8

*a* = 0.28 , *b* = 6.28

Неупорядоченная выборка:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6.10346 | 2.85261 | 3.98383 | 6.18185 | 3.10023 | 6.0174 | 6.04172 | 1.54215 | 5.31046 | 2.62264 |
| 5.71363 | 3.47689 | 3.49528 | 2.22075 | 0.80054 | 3.57324 | 0.55363 | 1.28617 | 5.52488 | 3.66873 |
| 2.80727 | 5.88372 | 4.65758 | 3.90008 | 4.02744 | 0.42278 | 3.97555 | 2.20399 | 2.76297 | 3.93868 |
| 3.05056 | 3.11922 | 4.32179 | 3.24335 | 1.37261 | 2.69627 | 4.19823 | 2.986 | 1.66048 | 5.85716 |
| 1.50158 | 3.35958 | 2.41449 | 4.75537 | 0.62745 | 5.9915 | 6.06525 | 1.70151 | 3.75902 | 1.69652 |
| 3.33251 | 1.37559 | 2.34994 | 4.29508 | 5.17899 | 0.84803 | 0.71742 | 0.46363 | 4.08641 | 0.61406 |
| 4.1895 | 2.62425 | 2.81607 | 4.97473 | 5.21953 | 4.24738 | 4.33001 | 4.58254 | 4.01568 | 0.85423 |
| 4.1735 | 2.47386 | 2.5332 | 3.55828 | 0.78018 | 4.9003 | 1.53462 | 0.71697 | 3.03379 | 4.62012 |
| 1.14 | 0.43581 | 3.40881 | 0.38693 | 4.13967 | 4.09273 | 5.57586 | 1.29068 | 6.01856 | 1.22761 |
| 0.41946 | 5.96469 | 5.63625 | 3.6945 | 5.96018 | 3.56707 | 4.13283 | 4.05319 | 1.62083 | 0.91023 |
| 2.50118 | 4.11607 | 5.0721 | 4.36815 | 4.68962 | 3.45495 | 3.13936 | 4.65301 | 3.02097 | 1.81675 |
| 5.48651 | 1.60709 | 0.58829 | 0.43869 | 0.44424 | 1.20489 | 6.21098 | 3.85058 | 4.94848 | 0.62691 |
| 0.9863 | 3.50592 | 4.89359 | 1.12228 | 5.89191 | 0.95598 | 0.48674 | 5.65524 | 3.34509 | 4.93511 |
| 4.66987 | 4.05727 | 6.15904 | 4.58448 | 1.45184 | 3.13357 | 3.96013 | 1.2157 | 1.8508 | 2.1225 |
| 4.11592 | 3.50328 | 5.676 | 1.17439 | 4.32113 | 5.66935 | 2.95471 | 5.86504 | 6.05439 | 2.74151 |
| 2.49408 | 2.68061 | 4.9712 | 3.67192 | 0.81677 | 1.44042 | 1.92615 | 2.54887 | 2.41564 | 6.21233 |
| 0.6882 | 1.37967 | 3.21685 | 2.20513 | 4.685 | 6.07272 | 5.23675 | 4.1068 | 3.15107 | 4.32698 |
| 1.45518 | 1.75494 | 4.97853 | 6.10433 | 4.52124 | 3.73423 | 6.07565 | 5.13112 | 3.1957 | 5.05439 |
| 2.74221 | 1.17625 | 1.22427 | 4.15154 | 4.23741 | 1.25853 | 3.46957 | 4.48617 | 2.69029 | 5.03893 |
| 4.57078 | 4.23236 | 3.30191 | 3.91917 | 5.84449 | 1.25724 | 2.35328 | 4.19634 | 2.14601 | 5.89708 |

Упорядоченная выборка:

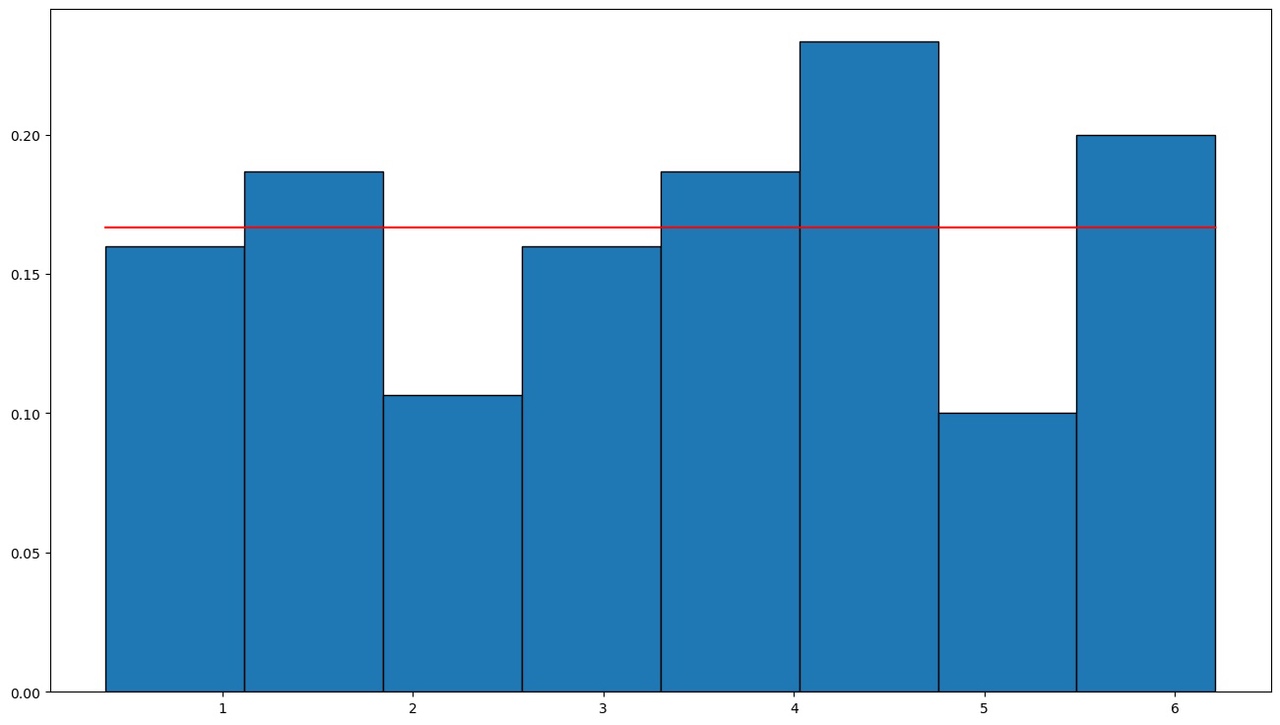
|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0.38693 | 0.41946 | 0.42278 | 0.43581 | 0.43869 | 0.44424 | 0.46363 | 0.48674 | 0.55363 | 0.58829 |
| 0.61406 | 0.62691 | 0.62745 | 0.6882 | 0.71697 | 0.71742 | 0.78018 | 0.80054 | 0.81677 | 0.84803 |
| 0.85423 | 0.91023 | 0.95598 | 0.9863 | 1.12228 | 1.2014 | 1.17439 | 1.17625 | 1.20489 | 1.2157 |
| 1.22427 | 1.22761 | 1.25724 | 1.25853 | 1.28617 | 1.29068 | 1.37261 | 1.37559 | 1.37967 | 1.44042 |
| 1.45184 | 1.45518 | 1.50158 | 1.53462 | 1.54215 | 1.60709 | 1.62083 | 1.66048 | 1.69652 | 1.70151 |
| 1.75494 | 1.81675 | 1.8508 | 1.92615 | 2.1225 | 2.14601 | 2.20399 | 2.20513 | 2.22075 | 2.34994 |
| 2.35328 | 2.41449 | 2.41564 | 2.47386 | 2.49408 | 2.50118 | 2.5332 | 2.54887 | 2.62264 | 2.62425 |
| 2.68061 | 2.69029 | 2.69627 | 2.74151 | 2.74221 | 2.76297 | 2.80727 | 2.81607 | 2.85261 | 2.95471 |
| 2.986 | 3.02097 | 3.03379 | 3.05056 | 3.10023 | 3.11922 | 3.13357 | 3.13936 | 3.15107 | 3.1957 |
| 3.21685 | 3.24335 | 3.30191 | 3.33251 | 3.34509 | 3.35958 | 3.40881 | 3.45495 | 3.46957 | 3.47689 |
| 3.49528 | 3.50328 | 3.50592 | 3.55828 | 3.56707 | 3.57324 | 3.66873 | 3.67192 | 3.6945 | 3.73423 |
| 3.75902 | 3.85058 | 3.90008 | 3.91917 | 3.93868 | 3.96013 | 3.97555 | 3.98383 | 4.01568 | 4.02744 |
| 4.05319 | 4.05727 | 4.08641 | 4.09273 | 4.1068 | 4.11592 | 4.11607 | 4.13283 | 4.13967 | 4.15154 |
| 4.1735 | 4.1895 | 4.19634 | 4.19823 | 4.23236 | 4.23741 | 4.24738 | 4.29508 | 4.32113 | 4.32179 |
| 4.32698 | 4.33001 | 4.36815 | 4.48617 | 4.52124 | 4.57078 | 4.58254 | 4.58448 | 4.62012 | 4.65301 |
| 4.65758 | 4.66987 | 4.685 | 4.68962 | 4.75537 | 4.89359 | 4.9003 | 4.93511 | 4.94848 | 4.9712 |
| 4.97473 | 4.97853 | 5.03893 | 5.05439 | 5.0721 | 5.13112 | 5.17899 | 5.21953 | 5.23675 | 5.31046 |
| 5.48651 | 5.52488 | 5.57586 | 5.63625 | 5.65524 | 5.66935 | 5.676 | 5.71363 | 5.84449 | 5.85716 |
| 5.86504 | 5.88372 | 5.89191 | 5.89708 | 5.96018 | 5.96469 | 5.9915 | 6.0174 | 6.01856 | 6.04172 |
| 6.05439 | 6.06525 | 6.07272 | 6.07565 | 6.10346 | 6.10433 | 6.15904 | 6.18185 | 6.21098 | 6.21233 |

Интервальный ряд:

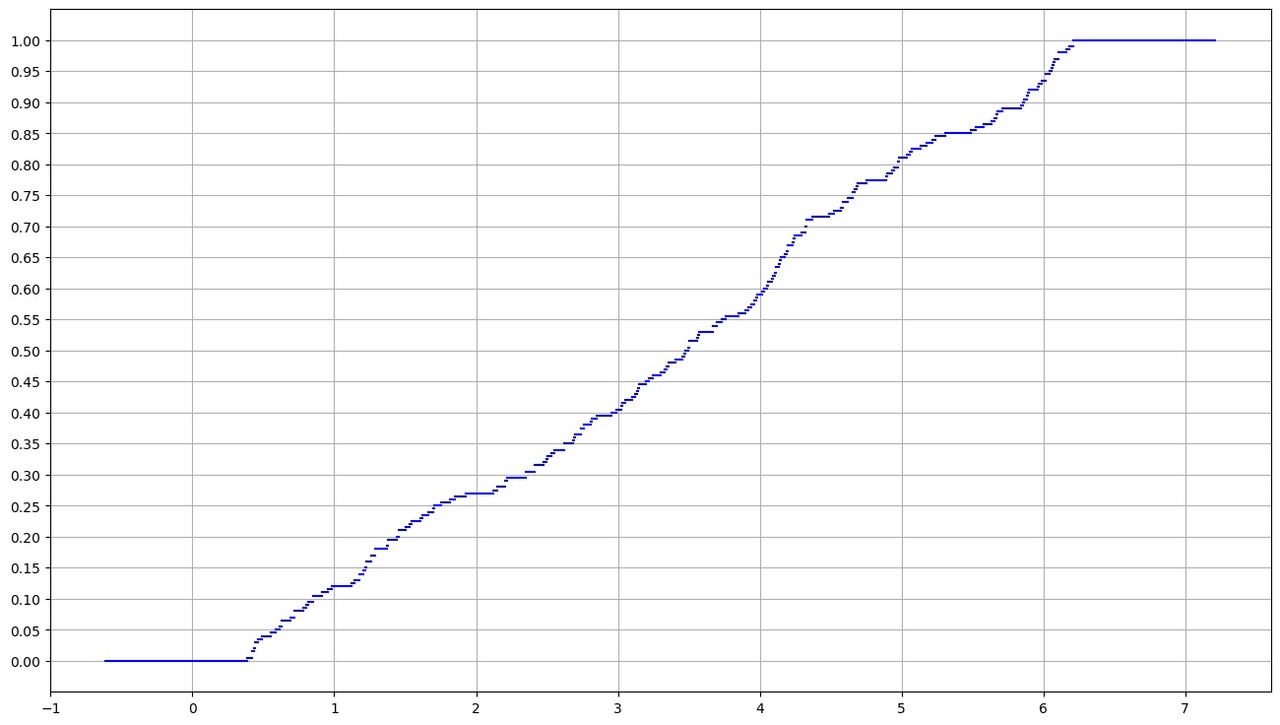
|  |  |  |
| --- | --- | --- |
| Интервалы | *ni* | *wi* |
| [0.28000, 1.03000] | 24 | 0.12000 |
| (1.03000, 1.78000] | 27 | 0.14000 |
| (1.78000, 2.53000] | 15 | 0.08000 |
| (2.53000, 3.28000] | 26 | 0.12000 |
| (3.28000, 4.03000] | 28 | 0.14000 |
| (4.03000, 4.78000] | 35 | 0.17500 |
| (4.78000, 5.53000] | 17 | 0.07500 |
| (5.53000, 6.28000] | 28 | 0.15000 |
|  | 200 | 1.00000 |

Ассоциированный статистический ряд:

|  |  |  |
| --- | --- | --- |
| *xi\** | *ni* | *wi* |
| 0.65500 | 24 | 0.12000 |
| 1.40500 | 27 | 0.14000 |
| 2.15500 | 15 | 0.08000 |
| 2.90500 | 26 | 0.12000 |
| 3.65500 | 28 | 0.14000 |
| 4.40500 | 35 | 0.17500 |
| 5.15500 | 17 | 0.07500 |
| 5.90500 | 28 | 0.15000 |
|  | 200 | 1.00000 |



**Рисунок 15**. Гистограмма относительных частот задания 8



**Рисунок 16.** График эмпирической функции распределения задания 8

1. Выборочное среднее: 3.34944
2. Выборочная дисперсия с поправкой Шеппарда: 2.922
3. Выборочное среднее квадратичное отклонение: 1.70939
4. Выборочная мода: 4.34371
5. Выборочная медиана: 3.48609
6. Выборочный коэффициент асимметрии: -0.08018
7. Выборочный коэффициент эксцесса: -1.10246

**Равномерное распределение на отрезке**

|  |  |  |  |
| --- | --- | --- | --- |
| Интервал | *wi* | *pi* | |*wi-pi*| |
| [0.28000, 1.03000] | 0.12000 | 0.12500 | 0.00500 |
| (1.03000, 1.78000] | 0.14000 | 0.12500 | 0.01500 |
| (1.78000, 2.53000] | 0.08000 | 0.12500 | 0.04500 |
| (2.53000, 3.28000] | 0.12000 | 0.12500 | 0.00500 |
| (3.28000, 4.03000] | 0.14000 | 0.12500 | 0.01500 |
| (4.03000, 4.78000] | 0.17500 | 0.12500 | 0.05000 |
| (4.78000, 5.53000] | 0.07500 | 0.12500 | 0.05000 |
| (5.53000, 6.28000] | 0.15000 | 0.12500 | 0.02500 |
|  | 1.00000 | 1.00000 | 0.05000 |

**Распределение по показательному закону**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название  показателя | Экспериментальное значение | Теоретическое значение | Абсолютное отклонение | Относительное отклонение |
| Выборочное среднее | 3.34944 | 3.28000 | 0.06944 | 2.09490 |
| Выборочная дисперсия | 2.92200 | 3.00000 | 0.07800 | 2.63425 |
| Выборочное среднее  квадратичное отклонение | 1.70939 | 1.73205 | 0.02266 | 1.31689 |
| Выборочная мода | 4.34371 | 3.28000 | 1.06371 | 27.90531 |
| Выборочная медиана | 3.48609 | 3.28000 | 0.20609 | 6.09185 |
| Выборочный коэффициент асимметрии | -0.8018 | 0.00000 | 0.08018 | - |
| Выборочный коэффициент эксцесса | -1.10246 | -1.20000 | 0.09754 | 8.47268 |

# 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, 2021.
3. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Юрайт, 2020.
4. Гмурман В.Е. Руководство к решению задач по теории вероятностей и математической статистике. – М.: Юрайт, 2020.
5. Емельянов Г.В., Скитович В.П. Задачник по теории вероятностей и математической статистике. – СПб.: Лань, 2019.
6. Ивченко Г.И., Медведев Ю.И. Математическая статистика. – М.: URSS, 2020.
7. Кибзун А.И., Горяинова Е.Р., Наумов А.В. Теория вероятностей и математическая статистика. Базовый курс с примерами и задачам. Учебное пособие – М.: ФИЗМАТЛИТ, 2005.
8. Кобзарь А.И. Прикладная математическая статистика: для инженеров и научных работников – М.: ФИЗМАТЛИТ, 2006.
9. Письменный Д.Т. Конспект лекций по теории вероятностей,

математической статистике и случайным процессам: учеб. пособие для вузов. – М.: Айрис-пресс, 2020.

# Приложение

Код выполнения заданий:

from numpy import random

import numpy as np

import random

from scipy.stats import binom

from scipy.stats import geom

from scipy.stats import poisson

from scipy.stats import randint

from scipy.stats import hypergeom

from scipy.stats import norm

from scipy.stats import expon

from scipy.stats import uniform

import random

binominal\_array = binom.rvs(n=19, p=0.27, size=200)

print(binominal\_array)

binominal\_array = sorted(binominal\_array)

print(binominal\_array)

binominal\_array = [1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 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, 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, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 9, 10, 11]

print(binominal\_array)

geom\_array = geom.rvs(p=0.27, size=200)

for i in range(len(geom\_array)):

geom\_array[i]-=1

print(geom\_array)

geom\_array = sorted(geom\_array)

print(geom\_array)

poisson\_array = poisson.rvs(0.64, size=200)

print(poisson\_array)

poisson\_array = sorted(poisson\_array)

print(poisson\_array)

n = 5 + 14%28

a = 0

b = n

print(n)

randint\_array = randint.rvs(a, b, size = 200)

print(randint\_array)

randint\_array = sorted(randint\_array)

print(randint\_array)

m = 5 + 14%7 #

M = 2\*m + 3 #

K = m + 1 + 14%2 #

print(m, M, K)

hyper\_array = hypergeom.rvs(M, K, m, size = 200)

print(hyper\_array)

hyper\_array = sorted(hyper\_array)

print(hyper\_array)

#Следуя Указаниям для всех выборок в Заданиях 1-5, построить:

#1) статистический ряд;

#2)полигон относительных частот;

#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

from collections import Counter

import matplotlib.pyplot as plt

from scipy.stats import binom

from scipy.stats import geom

from scipy.stats import poisson

from scipy.stats import hypergeom

import numpy as np

def frequency(a):

t = a[0]

count = []

unique = []

otn = []

k = 0

for i in a:

if i == t:

k+=1

else:

unique.append(t)

count.append(k)

otn.append(round(k/len(a),5))

k = 1

t = i

unique.append(t)

count.append(k)

otn.append(round(k/len(a),5))

return unique, count, otn

def own\_emp(a,s):

plt.figure(figsize = (16,9))

n = len(a) -1

plt.plot([-1,0], [0,0], color = 'blue')

for i in range(n):

pair = (a[i], a[i+1])

plt.plot(pair, [s[i],s[i]], color = 'blue')

plt.plot((a[n],a[n]+1),[1.0, 1.0], color = 'blue')

plt.scatter(a, s, color='blue', s=5, marker='o')

plt.grid()

plt.xticks(a)

plt.yticks(np.arange(21) \* 0.05)

plt.show()

def polygon(unique, otn, t):

x = unique

y = otn

plt.figure(figsize = (16,9))

plt.plot(x,y, '-o', label='Фактические вероятности')

z = []

if t == 1:

for i in x:

t = binom.pmf(i, 19, 0.27)

z.append(t)

elif t ==2:

for i in x:

t = geom.pmf(i+1, 0.27)

z.append(t)

elif t ==3:

for i in x:

t = poisson.pmf(i, 0.64)

z.append(t)

elif t ==4:

for i in x:

t = 1/19

z.append(t)

else:

for i in x:

t = hypergeom.pmf(i, 13, 6, 5)

z.append(t)

#plt.plot(x,z, '-o')

plt.plot(x,z, '-o',color='red', label='Теоретические вероятности')

plt.legend()

max\_value = max(unique)

max\_probability = max(z)

plt.xticks(np.arange(0, max\_value + 1, 1))

plt.yticks(np.arange(0, max\_probability + 0.02, 0.02))

plt.show()

#Ищем выборочное среднее, выборочную дисперсию, выборочное среднее квадратическое отклонение, выборочную моду, выборочную медиану, выборочный коэффициент асимметрии, выборочный коэффициент эксцесса

import statistics

import pandas as pd

def osn\_stats(a):

unique, count, otn = frequency(a)

#выборочное среднее

v\_midle = np.mean(a)

#выборочную дисперсию

v\_disp = np.var(a, ddof = 1)

#factor = np.sqrt(len(a)-1)/(len(a)-2.7)

#v\_disp = v\_disp\*factor

#выборочное среднее квадратическое отклонение

vsk = np.sqrt(v\_disp)

#выборочную моду

moda = statistics.mode(a)

#выборочную медиану

med = statistics.median(a)

#выборочный коэффициент асимметрии

ap = pd.Series(a)

vka = ap.skew()

#выборочный коэффициент эксцесса

vkx = ap.kurt()

return round(v\_midle,5), round(v\_disp,5), round(vsk,5), moda, round(med,5), round(vka,5), round(vkx,5)

a, b, c = frequency(binominal\_array)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

polygon(a,c,1)

own\_emp(a,sum)

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats(binominal\_array)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная дисперсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",temp4,

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

probability = []

a, b, c = frequency(binominal\_array)

for i in a:

temp = binom.pmf(i, 19, 0.27)

probability.append(temp)

​

print("\n","j w p\_j w-p\_j")

sum = 0

sum

for i in range(len(a)):

print("\n", a[i]," ",c[i], " ", round(probability[i],5), " ", round(abs(c[i] - probability[i]),5))

sum+=probability[i]

print(sum)

def ab(x,y):

return round(abs(x-y),5)

def ot(x,y):

return round(abs((x - y) / ((x + y) / 2)) \* 100, 5)

p = 0.27

n = 19

print((n+1)\*p)

print('\nВыборочное среднее, ',5.19, ",", round(n\*p,5), ",", round(abs(5.19 - round(n\*p,5)),5),",",

round(abs((5.19 - round(n\*p,5)) / ((5.19 + round(n\*p,5)) / 2)) \* 100,5))

print('\nВыборочная дисперсия,',3.72251 , ",",round(n\*p\*(1-p),5),",", round(abs(3.72251 - round(n\*p\*(1-p),5)),5),",",

round(abs((3.72251 - round(n\*p\*(1-p),5)) / ((3.72251 + round(n\*p\*(1-p),5)) / 2)) \* 100,5))

print('\nВыборочное среднее квадратичное отклонение,',1.92938 , ",",round((n\*p\*(1-p))\*\*0.5,5),",",

round(abs(1.92938 - round((n\*p\*(1-p))\*\*0.5,5)),5),",",round(abs((1.92938 - round((n\*p\*(1-p))\*\*0.5,5)) / ((1.92938 + round((n\*p\*(1-p))\*\*0.5,5)) / 2)) \* 100,5))

print('\nВыборочная мода,',5 ,",",5 ,",",0,", 0")

print('\nВыборочная медиана,',5,",",5,",",0,", 0")

print('\nВыборочный коэффициент асимметрии,',0.37201 , ",",round((1-2\*p)/((n\*p\*(1-p))\*\*0.5),5),",",

round(abs(0.37201 - round((1-2\*p)/((n\*p\*(1-p))\*\*0.5),5)),5),",",round(abs((0.37201 -round((1-2\*p)/((n\*p\*(1-p))\*\*0.5),5)) / ((0.37201 + round((1-2\*p)/((n\*p\*(1-p))\*\*0.5),5)) / 2)) \* 100,5))

print('\nВыборочный коэффициент эксцесса,',-0.25795,",", round((1-6\*p\*(1-p))/(n\*p\*(1-p)),5),",",

round(abs(-0.25795 + round((1-6\*p\*(1-p))/(n\*p\*(1-p)),5)),5),",", round(abs((-0.25795 +round((1-6\*p\*(1-p))/(n\*p\*(1-p)),5)) / ((-0.25795 - round((1-6\*p\*(1-p))/(n\*p\*(1-p)),5)) / 2)) \* 100,5))

#Геометрическое распределение

a, b, c = frequency(geom\_array)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

polygon(a,c,2)

own\_emp(a,sum)

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats(geom\_array)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",temp4,

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

from scipy.stats import geom

probability = []

a, b, c = frequency(geom\_array)

for i in a:

temp = geom.pmf(i+1, 0.27)

probability.append(temp)

print("\n","j w p\_j w-p\_j")

sum = 0

for i in range(len(a)):

print("\n", a[i]," ",c[i], " ", round(probability[i],5), " ", round(abs(c[i] - probability[i]),5))

sum+=probability[i]

print(sum)

import math

p = 0.27

q = 1-p

print(math.log(2)/math.log(q))

print('\nВыборочное среднее: ', round(q/p,5))

print('\nВыборочная дисперсия: ',round(q/(p\*\*2),5))

print('\nВыборочное среднее квадратичное отклонение: ',round((q\*\*0.5)/p,5))

print('\nВыборочная мода: ',1)

print('\nВыборочная медиана: ',2)

print('\nВыборочный к. асимметрии: ',round((2-p)/math.sqrt(q),5),)

print('\nВыборочный к. эксцесса: ', round(6+(p\*\*2)/q,5))

import math

p = 0.27

q = 1-p

print('\nВыборочное среднее, ',2.675, ",", 2.7037, ",", ab(2.675, 2.7037),",",

ot(2.675,2.7037))

print('\nВыборочная дисперсия,',9.96922 , ",",10.01372,",", ab(9.96922 ,10.01372),",",

ot(9.96922 ,10.01372))

print('\nВыборочное среднее квадратичное отклонение,',3.15741, ",",3.16445,",",

ab(3.15741, 3.16445),",",ot(3.15741, 3.16445))

print('\nВыборочная мода,',0,",",1,",",ab(0, 1),",",ot(0,1))

print('\nВыборочная медиана,',2,",",2,",",0,",0")

print('\nВыборочный коэффициент асимметрии,',2.10798 , ",",2.02481,",",

ab(2.10798,2.02481),",",ot(2.10798,2.02481))

print('\nВыборочный коэффициент эксцесса,',5.88894,",", 6.09986,",",

ab(5.88894,6.09986),",",ot(5.88894,6.09986))

a, b, c = frequency(poisson\_array)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

polygon(a,c,3)

own\_emp(a,sum)

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats(poisson\_array)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",temp4,

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

from scipy.stats import poisson

probability = []

a, b, c = frequency(poisson\_array)

for i in a:

temp = poisson.pmf(i, 0.64)

probability.append(temp)

print("\n","j w p\_j w-p\_j")

sum = 0

for i in range(len(a)):

print("\n", a[i]," ",c[i], " ", round(probability[i],5), " ", round(abs(c[i] - probability[i]),5))

sum+=probability[i]

print(sum)

import math

l = 0.64

print(l+1/3-0.02/l)

print('\nВыборочное среднее: ', l)

print('\nВыборочная дисперсия: ',l)

print('\nВыборочное среднее квадратичное отклонение: ',round((l\*\*0.5)/p,5))

print('\nВыборочная мода: ',0)

print('\nВыборочная медиана: ',0)

print('\nВыборочный к. асимметрии: ',round(1/math.sqrt(l),5))

print('\nВыборочный к. эксцесса: ', round(1/l,5))

print('\nВыборочное среднее, ', 0.725, ",", 0.64, ",", ab(0.725, 0.64), ",", ot(0.725, 0.64))

print('\nВыборочная дисперсия,', 0.78329, ",", 0.64, ",", ab( 0.78329, 0.64), ",", ot( 0.78329, 0.64))

print('\nВыборочное среднее квадратичное отклонение,', 0.88504 , ",", 2.96296, ",", ab( 0.88504 , 2.96296), ",", ot( 0.88504 , 2.96296))

print('\nВыборочная мода,', 0, ",", 0, ",", 0, ",", 0)

print('\nВыборочная медиана,', 0.5, ",", 0, ",", 0.5, ",", ot( 0.5 , 0))

print('\nВыборочный коэффициент асимметрии,', 1.2283, ",", 1.25, ",", ab(1.2283, 1.25), ",", ot(1.2283, 1.25))

print('\nВыборочный коэффициент эксцесса,', 1.27317, ",", 1.5625, ",", ab(1.27317, 1.5625), ",", ot(1.27317, 1.5625))

a, b, c = frequency(randint\_array)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

polygon(a,c,4)

own\_emp(a,sum)

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats(randint\_array)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",temp4,

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

probability = []

a, b, c = frequency(randint\_array)

for i in a:

temp = 1/19

probability.append(temp)

print("\n","j w p\_j w-p\_j")

sum = 0

for i in range(len(a)):

print("\n", a[i]," ",c[i], " ", round(probability[i],5), " ", round(abs(c[i] - probability[i]),5))

sum+= probability[i]

print(sum)

import math

n = 19

print('\nВыборочное среднее: ', (n-1)/2)

print('\nВыборочная дисперсия: ',round((n\*\*2-1)/12,5))

print('\nВыборочное среднее квадратичное отклонение: ',round(0.5\*(((n\*\*2-1)/3)\*\*0.5),5))

print('\nВыборочная мода: ',(n-1)/2)

print('\nВыборочная медиана: ',(n-1)/2)

print('\nВыборочный к. асимметрии: ',0)

print('\nВыборочный к. эксцесса: ', round(-(6\*(n\*\*2+1)/(5\*(n\*\*2-1))),5))

print('\nВыборочное среднее, ',9.21, ",", 9, ",", ab(9.21, 9), ",", ot(9.21, 9))

print('\nВыборочная дисперсия,', 29.86523 , ",", 30, ",", ab(29.86523 , 30), ",", ot(29.86523 , 30))

print('\nВыборочное среднее квадратичное отклонение,', 5.46491 , ",", 5.47723, ",", ab(5.46491 , 5.47723), ",", ot(5.46491 , 5.47723))

print('\nВыборочная мода,', 3 , ",", 9, ",", ab(3 , 9), ",", ot(3 , 9))

print('\nВыборочная медиана,', 9, ",", 9, ",", ab(9, 9), ",", ot(9, 9))

print('\nВыборочный коэффициент асимметрии,', -0.001, ",", 0, ",", ab(-0.001, 0), ",", ot(-0.001, 0 ))

print('\nВыборочный коэффициент эксцесса,', -1.23407, ",",-1.20667, ",", ab(-1.23407, -1.20667), ",", ot(-1.23407, -1.20667))

a, b, c = frequency(hyper\_array)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

polygon(a,c,5)

own\_emp(a,sum)

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats(hyper\_array)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",temp4,

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

from scipy.stats import hypergeom

probability = []

a, b, c = frequency(hyper\_array)

for i in a:

temp = hypergeom.pmf(i, 13, 6, 5)

probability.append(temp)

print("\n","j w p\_j w-p\_j")

sum = 0

for i in range(len(a)):

print("\n", a[i]," ",c[i], " ", round(probability[i],5), " ", round(abs(c[i] - probability[i]),5))

sum +=probability[i]

print(sum)

import math

m = 5

M = 13

K = 6

print((M-1)\*(M\*\*2)/(m\*(M-2)\*(M-3)\*(M-m)))

print('\nВыборочное среднее: ', round(m\*K/M,5))

print('\nВыборочная дисперсия: ',round(m\*K\*(M-K)\*(M-m)/((M-1)\*(M\*\*2)),5))

print('\nВыборочное среднее квадратичное отклонение: ',round(((m\*K\*(M-K)\*(M-m)/(M-1))\*\*0.5)/M,5))

print('\nВыборочная мода: ',3)

print('\nВыборочная медиана: ',3)

print('\nВыборочный к. асимметрии: ',round(((M-2\*K)\*(M-2\*m)/(M-2))\*math.sqrt((M-1)/(m\*K\*(M-K)\*(M-m))),5))

print('\nВыборочный к. эксцесса: ', 0)

print('\nВыборочное среднее, ', 2.3 , ",", 2.30769, ",", ab(2.3 , 2.30769), ",", ot(2.3 , 2.30769))

print('\nВыборочная дисперсия,', 0.74372 , ",", 0.8284, ",", ab(0.74372 , 0.8284), ",", ot(0.74372 , 0.8284))

print('\nВыборочное среднее квадратичное отклонение,', 0.86239 , ",", 0.91017, ",", ab(0.86239 , 0.91017), ",", ot(0.86239 , 0.91017))

print('\nВыборочная мода,', 2, ",", 3, ",", ab(2, 3), ",", ot(2, 3))

print('\nВыборочная медиана,', 2, ",", 3, ",", ab(2, 3), ",", ot(2, 3))

print('\nВыборочный коэффициент асимметрии,', 0.04274 , ",", 0.02305, ",", ab(0.04274 , 0.02305), ",", ot(0.04274 , 0.02305))

print('\nВыборочный коэффициент эксцесса,', -0.51789, ",", 0, ",", ab(-0.51789, 0), ",", ot(-0.51789, 0))

# Задания 6-8

#Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных по нормальному закону

mu, sigma = 0.7, 1.07 # mean and standard deviation

norm\_r = np.random.normal(mu, sigma, 200)

norm\_r=[0.98696775, 0.01290478, -0.02446487, 0.81653421, 2.3992132, 2.04281783,

0.53951583, -1.43347883, 0.08785767, 1.4963278 , 1.44817026, -1.01690341,

-0.21280541, 0.85714779, 1.33833754, 1.39287846 , 1.34889942, 1.34174647,

0.98842293, 1.01928685 , 0.25739691 , 0.15727801 , 1.7882599 , 1.64939617,

0.79090123, -0.37165065, 0.8374534 , -0.95198516, -0.40937812, 1.06917679,

-0.68251469, 1.56568327, -0.97658981, 1.77767923, 0.33059275, 0.36122551,

-0.59040087, 3.09508953, 0.78025914, 0.93998975, 1.66759126 ,-2.33536192,

1.72135719, -1.15914536, -0.61305951, 1.12699296, 1.22408331 , 0.44123201,

0.32615578, 0.73512675, -0.55740655, 1.78893025, 0.21171127 , 0.88550187,

1.20744646, 2.16016147, -0.14204157, 0.82979242, 1.01693491, 2.76005472,

0.85292561, 2.59483077, -0.57906704, 0.10952454, -0.19682447, 0.66567394,

2.16320456, 1.3515773 , 1.11886089, 0.70758872, 1.35810849, 0.50922658,

0.71813521, -0.93908125, 0.18506375, 0.32998521, 1.43565329, -0.0097507,

0.60570962, 0.22413903, 3.32035375, 0.50618537, 1.45402171, 0.36802897,

1.21723852, -1.24357058, -0.09350686, 0.63966391, -0.68709817, 2.04050409,

0.02875, -0.55238818, 1.22600104, 2.93434109, 0.66956762, 1.18764582,

-0.91272491, 0.95938008, 0.67301727, 1.06241504, 1.12324941, 0.79187164,

2.34465065, 1.85491056, 0.05720817, -0.0331572 , 0.11671447, -0.6005448,

0.47099021, 1.73659372, 0.08007006, -0.46780245, 0.36385671, 1.43228456,

-0.82510102, -0.43535083, -1.09160882, 1.62747004, 1.14982249, 2.83876994,

0.4907401, 1.93871017, -0.89014022, 0.76882676, 0.1761311 , 1.29238867,

0.10772599, 2.51773521, -0.39396245, -1.20957271, 2.3270992 , 0.99258523,

-0.40421515, -0.66474081, 0.95237945, 0.14792594, -0.50352991, 1.19300245,

0.62876348, 0.4893717 , 1.591195 , 1.69122616, -0.12408866, 0.2562376,

0.23834853, 0.35489792, 2.32946823, 1.97105677, 2.31882545, 0.80573956,

-1.01106893, 0.82317234, 1.65761547, 0.21736128, -0.61135179, 1.75381879,

1.46642759, 0.42263054, 0.85523056, 1.05916513, -0.25467939, 0.73056125,

2.13067539, 0.8503713 , 1.60970342, 0.68548456, 0.37307842, -0.36590724,

0.66267531, 1.50457705, 1.87643855, -0.13346186, 0.22030328, 1.37739164,

1.37343124, 1.75012053, -1.18118449, 2.42151537, 0.97500757, 2.33862511,

1.65632148, -0.71495013, 1.26004984, 0.25641972, 2.30574119, -1.18334875,

-0.26418892, -0.55112357, 0.15934814, -0.24190262, 0.69548179, 0.81420698,

1.90077452, 1.05477255, 0.26155026, 0.05091513, 1.75572534, 0.47458592,

0.8306628, -1.09579 ]

temp = []

for i in norm\_r:

temp.append(round(i,5))

print(temp)

def create\_emp\_function(accumulated\_relative\_value\_frequencies, max\_value):

plt.rcParams.update({'font.size': 6})

plt.axhline(y=0, xmin=0, xmax=0.045, color='blue')

for i in range(0, max\_value):

x = [i, i + 1]

s = accumulated\_relative\_value\_frequencies[i]

y = [s, s]

plt.plot(x, y, color='blue')

plt.axhline(y=1, xmin=0.955, xmax=1, color='blue')

plt.yticks(numpy.arange(0, 1 + 0.02, 0.02))

plt.xticks(numpy.arange(0, max\_value + 1, 1))

plt.grid()

plt.show

def own\_emp(a,s):

plt.figure(figsize = (16,9))

n = len(a) -1

plt.plot([-1,0], [0,0], color = 'blue')

for i in range(n):

pair = (a[i], a[i+1])

plt.plot(pair, [s[i],s[i]], color = 'blue')

plt.plot((a[n],a[n]+1),[1.0, 1.0], color = 'blue')

plt.scatter(a, s, color='blue', s=5, marker='o')

plt.grid()

plt.xticks(a)

plt.yticks(np.arange(21) \* 0.05)

plt.show()

norm\_r =[0.98696775, 0.01290478, -0.02446487, 0.81653421, 2.3992132, 2.04281783,

0.53951583, -1.43347883, 0.08785767, 1.4963278 , 1.44817026, -1.01690341,

-0.21280541, 0.85714779, 1.33833754, 1.39287846 , 1.34889942, 1.34174647,

0.98842293, 1.01928685 , 0.25739691 , 0.15727801 , 1.7882599 , 1.64939617,

0.79090123, -0.37165065, 0.8374534 , -0.95198516, -0.40937812, 1.06917679,

-0.68251469, 1.56568327, -0.97658981, 1.77767923, 0.33059275, 0.36122551,

-0.59040087, 3.09508953, 0.78025914, 0.93998975, 1.66759126 ,-2.33536192,

1.72135719, -1.15914536, -0.61305951, 1.12699296, 1.22408331 , 0.44123201,

0.32615578, 0.73512675, -0.55740655, 1.78893025, 0.21171127 , 0.88550187,

1.20744646, 2.16016147, -0.14204157, 0.82979242, 1.01693491, 2.76005472,

0.85292561, 2.59483077, -0.57906704, 0.10952454, -0.19682447, 0.66567394,

2.16320456, 1.3515773 , 1.11886089, 0.70758872, 1.35810849, 0.50922658,

0.71813521, -0.93908125, 0.18506375, 0.32998521, 1.43565329, -0.0097507,

0.60570962, 0.22413903, 3.32035375, 0.50618537, 1.45402171, 0.36802897,

1.21723852, -1.24357058, -0.09350686, 0.63966391, -0.68709817, 2.04050409,

0.02875, -0.55238818, 1.22600104, 2.93434109, 0.66956762, 1.18764582,

-0.91272491, 0.95938008, 0.67301727, 1.06241504, 1.12324941, 0.79187164,

2.34465065, 1.85491056, 0.05720817, -0.0331572 , 0.11671447, -0.6005448,

0.47099021, 1.73659372, 0.08007006, -0.46780245, 0.36385671, 1.43228456,

-0.82510102, -0.43535083, -1.09160882, 1.62747004, 1.14982249, 2.83876994,

0.4907401, 1.93871017, -0.89014022, 0.76882676, 0.1761311 , 1.29238867,

0.10772599, 2.51773521, -0.39396245, -1.20957271, 2.3270992 , 0.99258523,

-0.40421515, -0.66474081, 0.95237945, 0.14792594, -0.50352991, 1.19300245,

0.62876348, 0.4893717 , 1.591195 , 1.69122616, -0.12408866, 0.2562376,

0.23834853, 0.35489792, 2.32946823, 1.97105677, 2.31882545, 0.80573956,

-1.01106893, 0.82317234, 1.65761547, 0.21736128, -0.61135179, 1.75381879,

1.46642759, 0.42263054, 0.85523056, 1.05916513, -0.25467939, 0.73056125,

2.13067539, 0.8503713 , 1.60970342, 0.68548456, 0.37307842, -0.36590724,

0.66267531, 1.50457705, 1.87643855, -0.13346186, 0.22030328, 1.37739164,

1.37343124, 1.75012053, -1.18118449, 2.42151537, 0.97500757, 2.33862511,

1.65632148, -0.71495013, 1.26004984, 0.25641972, 2.30574119, -1.18334875,

-0.26418892, -0.55112357, 0.15934814, -0.24190262, 0.69548179, 0.81420698,

1.90077452, 1.05477255, 0.26155026, 0.05091513, 1.75572534, 0.47458592,

0.8306628, -1.09579 ]

temp = []

for i in norm\_r:

temp.append(round(i,5))

print(temp)

norm\_r = temp

norm\_r = sorted(norm\_r)

print(norm\_r)

norm\_r = [-2.33536, -1.43348, -1.24357, -1.20957, -1.18335, -1.18118, -1.15915, -1.09579, -1.09161, -1.0169, -1.01107, -0.97659, -0.95199, -0.93908, -0.91272, -0.89014, -0.8251, -0.71495, -0.6871, -0.68251, -0.66474, -0.61306, -0.61135, -0.60054, -0.5904, -0.57907, -0.55741, -0.55239, -0.55112, -0.50353, -0.4678, -0.43535, -0.40938, -0.40422, -0.39396, -0.37165, -0.36591, -0.26419, -0.25468, -0.2419, -0.21281, -0.19682, -0.14204, -0.13346, -0.12409, -0.09351, -0.03316, -0.02446, -0.00975, 0.0129, 0.02875, 0.05092, 0.05721, 0.08007, 0.08786, 0.10773, 0.10952, 0.11671, 0.14793, 0.15728, 0.15935, 0.17613, 0.18506, 0.21171, 0.21736, 0.2203, 0.22414, 0.23835, 0.25624, 0.25642, 0.2574, 0.26155, 0.32616, 0.32999, 0.33059, 0.3549, 0.36123, 0.36386, 0.36803, 0.37308, 0.42263, 0.44123, 0.47099, 0.47459, 0.48937, 0.49074, 0.50619, 0.50923, 0.53952, 0.60571, 0.62876, 0.63966, 0.66268, 0.66567, 0.66957, 0.67302, 0.68548, 0.69548, 0.70759, 0.71814, 0.73056, 0.73513, 0.76883, 0.78026, 0.7909, 0.79187, 0.80574, 0.81421, 0.81653, 0.82317, 0.82979, 0.83066, 0.83745, 0.85037, 0.85293, 0.85523, 0.85715, 0.8855, 0.93999, 0.95238, 0.95938, 0.97501, 0.98697, 0.98842, 0.99259, 1.01693, 1.01929, 1.05477, 1.05917, 1.06242, 1.06918, 1.11886, 1.12325, 1.12699, 1.14982, 1.18765, 1.193, 1.20745, 1.21724, 1.22408, 1.226, 1.26005, 1.29239, 1.33834, 1.34175, 1.3489, 1.35158, 1.35811, 1.37343, 1.37739, 1.39288, 1.43228, 1.43565, 1.44817, 1.45402, 1.46643, 1.49633, 1.50458, 1.56568, 1.59119, 1.6097, 1.62747, 1.6494, 1.65632, 1.65762, 1.66759, 1.69123, 1.72136, 1.73659, 1.75012, 1.75382, 1.75573, 1.77768, 1.78826, 1.78893, 1.85491, 1.87644, 1.90077, 1.93871, 1.97106, 2.0405, 2.04282, 2.13068, 2.16016, 2.1632, 2.30574, 2.31883, 2.3271, 2.32947, 2.33863, 2.34465, 2.39921, 2.42152, 2.51774, 2.59483, 2.76005, 2.83877, 2.93434, 3.09509, 3.32035]

from numpy import random

lbd = 2.14

#p\_raspr = random.exponential(scale=1/lbd, size=200)

p\_raspr=[1.58504809, 0.5637526, 0.56259649, 0.08603452, 0.573049, 0.62259265,

0.05733476, 0.45840145, 1.96432019, 0.2158457, 0.4739142 , 1.52318452,

1.3822641 , 0.44280702, 0.406825 , 0.25079618, 0.14935659, 0.20027992,

0.62227167, 0.11180698, 0.71413961, 0.57852813, 0.20095035, 0.5742529,

0.78217963, 0.46606379, 0.80602334, 1.02906051, 0.57468666, 0.9095001,

0.45639952, 0.45795272, 0.33943739, 0.23831746, 0.80178179, 0.2246473,

0.97080169, 0.07023156, 0.34716861, 0.35232249, 0.24766207, 0.31187309,

0.03619662, 0.801175 , 0.08871078, 0.39529685, 0.16223517, 1.09222563,

0.8587036 , 0.32525925, 0.73006286, 0.44244176, 2.19110835, 1.81089794,

0.66478307, 0.37868767, 0.554676 , 0.28509838, 0.62300715, 0.44473957,

0.85571077, 1.3124239 , 0.44005705, 0.18362434, 0.60969715, 0.26164105,

0.12523331, 0.18518654, 0.62989759, 0.21271848, 0.9680501 , 0.16230212,

0.05394759, 0.85018814, 0.21004205, 1.02249559, 0.29383772, 0.00928326,

1.08950187, 0.62913992, 0.0168727 , 0.00591628, 0.51485101, 0.00477915,

0.04525808, 0.17998507, 0.90213078, 0.25021375, 0.41165161, 0.1133422,

0.91169986, 1.0621319 , 0.75675775, 1.30773229, 2.06989711, 0.02161452,

0.32781875, 0.51860848, 0.14686399, 0.47596534, 0.43678888, 0.18868065,

0.9748618 , 0.06363732, 0.68527032, 0.88623789, 0.93052427, 0.17435455,

0.0896337 , 0.12498122, 0.02396228, 0.68562436, 0.93484371, 0.55025782,

0.10536131, 0.45623569, 0.29459532, 0.76508085, 0.11585695, 0.47749568,

0.94881409, 0.57158662, 0.23924747, 0.11038608, 0.26473497, 1.40704165,

1.22003571, 0.26157862, 0.14838886, 0.74666624, 0.38054897, 0.37681042,

0.19819965, 0.36592625, 0.52899609, 0.15721381, 0.61463901, 0.13429461,

0.63230227, 0.19323377, 0.25240266, 0.5953445 , 0.07964568, 0.45679998,

0.10919336, 0.01355375, 0.15813469, 0.02513919, 0.24409787, 0.15308825,

0.4962611 , 0.47321454, 0.2465434 , 0.54159687, 0.46801651, 0.33036106,

0.47273961, 0.81519883, 0.10109317, 0.07591779, 0.31763603, 0.08568659,

0.14684223, 0.07485197, 0.0077217 , 0.35756611, 0.1678389 , 0.44220878,

0.10675774, 0.47806128, 0.40378859, 0.40945328, 0.34656513, 0.67609289,

0.56963221, 0.56594272, 0.30418276, 0.95520615, 0.43012568, 0.19758999,

0.04354109, 0.10332367, 0.86902978, 2.4618303 , 1.31509856, 0.39344319,

0.54336201, 0.32141118, 0.43490291, 0.30613519, 0.47549743, 0.37980466,

0.70378113, 0.21629673, 0.60705731, 0.6038563 , 0.44418912, 0.48051624,

0.99822641, 0.22489852]

t = []

for i in p\_raspr:

t.append(round(i,5))

p\_raspr = t

print(p\_raspr)

p\_raspr = sorted(p\_raspr)

print(p\_raspr)

p\_raspr = [0.00478, 0.00592, 0.00772, 0.00928, 0.01355, 0.01687, 0.02161, 0.02396, 0.02514, 0.0362, 0.04354, 0.04526, 0.05395, 0.05733, 0.06364, 0.07023, 0.07485, 0.07592, 0.07965, 0.08569, 0.08603, 0.08871, 0.08963, 0.10109, 0.10332, 0.10536, 0.10676, 0.10919, 0.11039, 0.11181, 0.11334, 0.11586, 0.12498, 0.12523, 0.13429, 0.14684, 0.14686, 0.14839, 0.14936, 0.15309, 0.15721, 0.15813, 0.16224, 0.1623, 0.16784, 0.17435, 0.17999, 0.18362, 0.18519, 0.18868, 0.19323, 0.19759, 0.1982, 0.20028, 0.20095, 0.21004, 0.21272, 0.21585, 0.2163, 0.22465, 0.2249, 0.23832, 0.23925, 0.2441, 0.24654, 0.24766, 0.25021, 0.2508, 0.2524, 0.26158, 0.26164, 0.26473, 0.2851, 0.29384, 0.2946, 0.30418, 0.30614, 0.31187, 0.31764, 0.32141, 0.32526, 0.32782, 0.33036, 0.33944, 0.34657, 0.34717, 0.35232, 0.35757, 0.36593, 0.37681, 0.37869, 0.3798, 0.38055, 0.39344, 0.3953, 0.40379, 0.40682, 0.40945, 0.41165, 0.43013, 0.4349, 0.43679, 0.44006, 0.44221, 0.44244, 0.44281, 0.44419, 0.44474, 0.45624, 0.4564, 0.4568, 0.45795, 0.4584, 0.46606, 0.46802, 0.47274, 0.47321, 0.47391, 0.4755, 0.47597, 0.4775, 0.47806, 0.48052, 0.49626, 0.51485, 0.51861, 0.529, 0.5416, 0.54336, 0.55026, 0.55468, 0.5626, 0.56375, 0.56594, 0.56963, 0.57159, 0.57305, 0.57425, 0.57469, 0.57853, 0.59534, 0.60386, 0.60706, 0.6097, 0.61464, 0.62227, 0.62259, 0.62301, 0.62914, 0.6299, 0.6323, 0.66478, 0.67609, 0.68527, 0.68562, 0.70378, 0.71414, 0.73006, 0.74667, 0.75676, 0.76508, 0.78218, 0.80117, 0.80178, 0.80602, 0.8152, 0.85019, 0.85571, 0.8587, 0.86903, 0.88624, 0.90213, 0.9095, 0.9117, 0.93052, 0.93484, 0.94881, 0.95521, 0.96805, 0.9708, 0.97486, 0.99823, 1.0225, 1.02906, 1.06213, 1.0895, 1.09223, 1.22004, 1.30773, 1.31242, 1.3151, 1.38226, 1.40704, 1.52318, 1.58505, 1.8109, 1.96432, 2.0699, 2.19111, 2.46183]

#Получить выборку, сгенерировав 200 псевдослучайных чисел, распределенных равномерно на отрезке

a = 0.28 # начало отрезка

b = a+6 # конец отрезка

n = 200 # количество чисел

#random\_numbers = [round(random.uniform(a, b),5) for \_ in range(n)]

random\_numbers=[6.10346, 5.71363, 2.80727, 3.05056, 1.50158, 3.33251, 4.1895, 4.1735, 1.14, 0.41946, 2.50118, 5.48651, 0.9863, 4.66987, 4.11592, 2.49408, 0.6882, 1.45518, 2.74221, 4.57078, 2.85261, 3.47689, 5.88372, 3.11922, 3.35958, 1.37559, 2.62425, 2.47386, 0.43581, 5.96469, 4.11607, 1.60709, 3.50592, 4.05727, 3.50328, 2.68061, 1.37967, 1.75494, 1.17625, 4.23236, 3.98383, 3.49528, 4.65758, 4.32179, 2.41449, 2.34994, 2.81607, 2.5332, 3.40881, 5.63625, 5.0721, 0.58829, 4.89359, 6.15904, 5.676, 4.9712, 3.21685, 4.97853, 1.22427, 3.30191, 5.31046, 5.52488, 2.76297, 1.66048, 3.75902, 4.08641, 4.01568, 3.03379, 6.01856, 1.62083, 3.02097, 4.94848, 3.34509, 1.8508, 6.05439, 2.41564, 3.15107, 3.1957, 2.69029, 2.14601, 6.18185, 2.22075, 3.90008, 3.24335, 4.75537, 4.29508, 4.97473, 3.55828, 0.38693, 3.6945, 4.36815, 0.43869, 1.12228, 4.58448, 1.17439, 3.67192, 2.20513, 6.10433, 4.15154, 3.91917, 3.10023, 0.80054, 4.02744, 1.37261, 0.62745, 5.17899, 5.21953, 0.78018, 4.13967, 5.96018, 4.68962, 0.44424, 5.89191, 1.45184, 4.32113, 0.81677, 4.685, 4.52124, 4.23741, 5.84449, 6.0174, 3.57324, 0.42278, 2.69627, 5.9915, 0.84803, 4.24738, 4.9003, 4.09273, 3.56707, 3.45495, 1.20489, 0.95598, 3.13357, 5.66935, 1.44042, 6.07272, 3.73423, 1.25853, 1.25724, 6.04172, 0.55363, 3.97555, 4.19823, 6.06525, 0.71742, 4.33001, 1.53462, 5.57586, 4.13283, 3.13936, 6.21098, 0.48674, 3.96013, 2.95471, 1.92615, 5.23675, 6.07565, 3.46957, 2.35328, 1.54215, 1.28617, 2.20399, 2.986, 1.70151, 0.46363, 4.58254, 0.71697, 1.29068, 4.05319, 4.65301, 3.85058, 5.65524, 1.2157, 5.86504, 2.54887, 4.1068, 5.13112, 4.48617, 4.19634, 2.62264, 3.66873, 3.93868, 5.85716, 1.69652, 0.61406, 0.85423, 4.62012, 1.22761, 0.91023, 1.81675, 0.62691, 4.93511, 2.1225, 2.74151, 6.21233, 4.32698, 5.05439, 5.03893, 5.89708]

print(random\_numbers)

random\_numbers = sorted(random\_numbers)

print(random\_numbers)

random\_numbers = [0.38693, 0.41946, 0.42278, 0.43581, 0.43869, 0.44424, 0.46363, 0.48674, 0.55363, 0.58829, 0.61406, 0.62691, 0.62745, 0.6882, 0.71697, 0.71742, 0.78018, 0.80054, 0.81677, 0.84803, 0.85423, 0.91023, 0.95598, 0.9863, 1.12228, 1.14, 1.17439, 1.17625, 1.20489, 1.2157, 1.22427, 1.22761, 1.25724, 1.25853, 1.28617, 1.29068, 1.37261, 1.37559, 1.37967, 1.44042, 1.45184, 1.45518, 1.50158, 1.53462, 1.54215, 1.60709, 1.62083, 1.66048, 1.69652, 1.70151, 1.75494, 1.81675, 1.8508, 1.92615, 2.1225, 2.14601, 2.20399, 2.20513, 2.22075, 2.34994, 2.35328, 2.41449, 2.41564, 2.47386, 2.49408, 2.50118, 2.5332, 2.54887, 2.62264, 2.62425, 2.68061, 2.69029, 2.69627, 2.74151, 2.74221, 2.76297, 2.80727, 2.81607, 2.85261, 2.95471, 2.986, 3.02097, 3.03379, 3.05056, 3.10023, 3.11922, 3.13357, 3.13936, 3.15107, 3.1957, 3.21685, 3.24335, 3.30191, 3.33251, 3.34509, 3.35958, 3.40881, 3.45495, 3.46957, 3.47689, 3.49528, 3.50328, 3.50592, 3.55828, 3.56707, 3.57324, 3.66873, 3.67192, 3.6945, 3.73423, 3.75902, 3.85058, 3.90008, 3.91917, 3.93868, 3.96013, 3.97555, 3.98383, 4.01568, 4.02744, 4.05319, 4.05727, 4.08641, 4.09273, 4.1068, 4.11592, 4.11607, 4.13283, 4.13967, 4.15154, 4.1735, 4.1895, 4.19634, 4.19823, 4.23236, 4.23741, 4.24738, 4.29508, 4.32113, 4.32179, 4.32698, 4.33001, 4.36815, 4.48617, 4.52124, 4.57078, 4.58254, 4.58448, 4.62012, 4.65301, 4.65758, 4.66987, 4.685, 4.68962, 4.75537, 4.89359, 4.9003, 4.93511, 4.94848, 4.9712, 4.97473, 4.97853, 5.03893, 5.05439, 5.0721, 5.13112, 5.17899, 5.21953, 5.23675, 5.31046, 5.48651, 5.52488, 5.57586, 5.63625, 5.65524, 5.66935, 5.676, 5.71363, 5.84449, 5.85716, 5.86504, 5.88372, 5.89191, 5.89708, 5.96018, 5.96469, 5.9915, 6.0174, 6.01856, 6.04172, 6.05439, 6.06525, 6.07272, 6.07565, 6.10346, 6.10433, 6.15904, 6.18185, 6.21098, 6.21233]

#Следуя Указаниям для всех выборок в Заданиях 6-8, построить:

#1)интервальный ряд и ассоциированный статистический ряд;

#2)гистограмму относительных частот;

#3)график эмпирической функции распределения; найти:

#4)выборочное среднее;

#5)выборочную дисперсию с поправкой Шеппарда;

#6)выборочное среднее квадратическое отклонение;

#7)выборочную моду;

#8)выборочную медиану;

#9)выборочный коэффициент асимметрии;

#10)выборочный коэффициент эксцесса

import math

m = 1 + math.floor(math.log2(200))

print(m)

def interval(data):

# Вычисляем минимальное и максимальное значения в массиве

min\_val = np.min(data)

max\_val = np.max(data)

# Вычисляем размах значений (диапазон)

range\_val = max\_val - min\_val

# Определяем количество интервалов (можно выбрать любое число)

num\_intervals = 8

# Вычисляем длину интервала

interval\_length = range\_val / num\_intervals

# Создаем список границ интервалов

interval\_boundaries = [min\_val + i \* interval\_length for i in range(num\_intervals)]

interval\_boundaries.append(max\_val)

# Создаем список счетчиков для каждого интервала

interval\_counts = [0] \* num\_intervals

# Проходим по всем значениям в массиве и увеличиваем счетчик соответствующего интервала

for value in data:

for i in range(num\_intervals):

if value >= interval\_boundaries[i] and value <= interval\_boundaries[i+1]:

interval\_counts[i] += 1

break

# Выводим результаты

print("Интервалы: ", interval\_boundaries)

print("Частоты: ", interval\_counts)

s = []

s.append(interval\_counts[0]/200)

for i in range(1,5):

t = s[i-1]+interval\_counts[i]/200

s.append(t)

return s

def own\_emp2(a,s):

plt.figure(figsize = (16,9))

n = len(a) -1

plt.plot([-1,0], [0,0], color = 'blue')

for i in range(n):

pair = (a[i], a[i+1])

plt.plot(pair, [s[i],s[i]], color = 'blue')

plt.plot((a[n],a[n]+1),[1.0, 1.0], color = 'blue')

plt.scatter(a, s, color='blue', s=5, marker='o')

plt.grid()

plt.xticks(a)

plt.yticks(np.arange(21) \* 0.05)

plt.show()

#Нормальный закон

s = interval(norm\_r)

import matplotlib.pyplot as plt

import math

import numpy as np

from scipy.stats import norm

import scipy.stats

m = 1 + math.floor(math.log2(200))

h = (3.32035 + 2.33536) / m

x = np.arange(-4.0, 4.0, 0.001)

data = norm\_r

fig = plt.figure(figsize=(10, 10))

ax = fig.add\_subplot(111)

heights, bins, \_ = ax.hist(data, edgecolor='black', weights=(np.ones\_like(data) / len(data)) / h, bins=m)

# Печать высот столбиков

print("Высоты столбиков:")

print(heights)

# Печать весов столбиков

weights = heights \* h

print("Веса столбиков:")

print(weights)

# Создание массива частичных сумм весов столбиков

partial\_sums = np.cumsum(weights)

print("Частичные суммы весов столбиков:")

print(partial\_sums)

plt.plot(x, norm.pdf(x, 0.7, 1.07\*\*2), color='red')

plt.show()

# Печать значения h

print("Значение h:")

print(h)

def own\_emp2(a,s):

plt.figure(figsize = (16,9))

n = len(a) - 1

plt.plot([a[0]-1,a[0]], [0,0], color = 'blue')

for i in range(n):

pair = (a[i], a[i+1])

plt.plot(pair, [s[i],s[i]], color = 'blue')

plt.plot((a[n],a[n]+1.0),[1.0, 1.0], color = 'blue')

#for i in range(n):

#plt.scatter(a[i], s[i], color='blue', s=5, marker='o')

plt.grid()

#plt.xticks(a)

plt.yticks(np.arange(21) \* 0.05)

plt.show()

def sheppard\_correction2(data, h):

v\_disp = np.var(data, ddof = 1)

v\_disp = v\_disp - (h\*\*2)/12

return v\_disp

def interval2(data):

# Вычисляем минимальное и максимальное значения в массиве

min\_val = np.min(data)

max\_val = np.max(data)

# Вычисляем размах значений (диапазон)

range\_val = max\_val - min\_val

# Определяем количество интервалов (можно выбрать любое число)

num\_intervals = 5

# Вычисляем длину интервала

interval\_length = range\_val / num\_intervals

# Создаем список границ интервалов

interval\_boundaries = [min\_val + i \* interval\_length for i in range(num\_intervals)]

interval\_boundaries.append(max\_val)

# Создаем список счетчиков для каждого интервала

interval\_counts = [0] \* num\_intervals

# Проходим по всем значениям в массиве и увеличиваем счетчик соответствующего интервала

for value in data:

for i in range(num\_intervals):

if value >= interval\_boundaries[i] and value <= interval\_boundaries[i+1]:

interval\_counts[i] += 1

break

return interval\_boundaries, interval\_counts

def moda(a,h):

interval\_boundaries, interval\_counts = interval2(a)

m = -1

t = 0

for i in range(5):

if interval\_counts[i] > m:

m = interval\_counts[i]

t = i

ch = []

for i in range(5):

ch.append(interval\_counts[i]/200)

if t ==4:

moda = interval\_boundaries[t]+(ch[t]-ch[t-1])/(2\*ch[t]-ch[t-1])

else:

moda = interval\_boundaries[t]+(ch[t]-ch[t-1])/(2\*ch[t]-ch[t-1]-ch[t+1])

return moda

def osn\_stats2(a,h):

#unique, count, otn = frequency(a)

#выборочное среднее

v\_midle = np.mean(a)

#выборочную дисперсию

v\_disp = sheppard\_correction2(a,h)

#выборочное среднее квадратическое отклонение

vsk = np.sqrt(v\_disp)

#выборочную моду

m = moda(a,h)

#выборочную медиану

med = statistics.median(a)

#выборочный коэффициент асимметрии

ap = pd.Series(a)

vka = ap.skew()

#выборочный коэффициент эксцесса

vkx = ap.kurt()

return round(v\_midle,5), round(v\_disp,5), round(vsk,5), m, round(med,5), round(vka,5), round(vkx,5)

temp = [-2.33536, -1.6283962500000002, -0.9214325000000001, -0.21446874999999999, 0.4924949999999999, 1.1994587499999998, 1.9064225000000001, 2.61338625, 3.32035]

ch = [0.005, 0.065, 0.13, 0.23, 0.255, 0.205, 0.085, 0.025]

s = [0.005, 0.07 , 0.2 , 0.43 , 0.685, 0.89 , 0.975, 1]

own\_emp2(temp, s)

a, b, c = frequency(norm\_r)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

own\_emp2(a,sum)

h = (temp[7]-temp[0])/8

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats2(norm\_r,h)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",round(temp4,5),

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

sum = 0

for i in range(8):

p = round(norm.cdf(temp[i+1], 0.7, 1.07) - norm.cdf(temp[i], 0.7, 1.07),5)

print("(",temp[i],",",temp[i+1],") ",ch[i]," ",p," ", round(abs(ch[i]-p),5))

sum+=p

print(sum," ",1-sum)

a = 0.7

s = 1.07

print('\nВыборочное среднее: ', a)

print('\nВыборочная дисперсия: ',round(s\*\*2,5))

print('\nВыборочное среднее квадратичное отклонение: ',s)

print('\nВыборочная мода: ',a)

print('\nВыборочная медиана: ',a)

print('\nВыборочный к. асимметрии: ',0)

print('\nВыборочный к. эксцесса: ', 0)

print('\nВыборочное среднее, ', 0.68537 , ",", 0.7, ",", ab(0.68537 , 0.7), ",", ot(0.68537 , 0.7))

print('\nВыборочная дисперсия,', 1.01698, ",",1.1449, ",", ab(1.01698, 1.1449), ",", ot(1.01698, 1.1449))

print('\nВыборочное среднее квадратичное отклонение,', 1.00845, ",", 1.07, ",", ab(1.00845, 1.07), ",", ot(1.00845, 1.07))

print('\nВыборочная мода,', 0.54231, ",", 0.7, ",", ab(0.54231, 0.7), ",", ot(0.54231, 0.7))

print('\nВыборочная медиана,', 0.72435 , ",", 0.7, ",", ab( 0.72435 , 0.7), ",", ot( 0.72435 , 0.7))

print('\nВыборочный коэффициент асимметрии,', -0.01412 , ",", 0, ",", ab(-0.01412 , 0), ",", " - ")

print('\nВыборочный коэффициент эксцесса,', -0.28606, ",", 0, ",", ab(-0.28606, 0), ",", " - ")

#Показательный закон

def interval(data):

# Вычисляем минимальное и максимальное значения в массиве

min\_val = 0.00478

max\_val = 2.46183

# Вычисляем размах значений (диапазон)

range\_val = max\_val - min\_val

# Определяем количество интервалов (можно выбрать любое число)

num\_intervals = 8

# Вычисляем длину интервала

interval\_length = range\_val / num\_intervals

# Создаем список границ интервалов

interval\_boundaries = [min\_val + i \* interval\_length for i in range(num\_intervals)]

interval\_boundaries.append(max\_val)

# Создаем список счетчиков для каждого интервала

interval\_counts = [0] \* num\_intervals

# Проходим по всем значениям в массиве и увеличиваем счетчик соответствующего интервала

for value in data:

for i in range(num\_intervals):

if value >= interval\_boundaries[i] and value <= interval\_boundaries[i+1]:

interval\_counts[i] += 1

break

# Выводим результаты

print("Интервалы: ", interval\_boundaries)

print("Частоты: ", interval\_counts)

s = []

s.append(interval\_counts[0]/200)

for i in range(1,5):

t = s[i-1]+interval\_counts[i]/200

s.append(t)

return s

s = interval(p\_raspr)

import matplotlib.pyplot as plt

import math

import numpy as np

from scipy.stats import expon

import scipy.stats

m = 1 + math.floor(math.log2(200))

h = (2.46183 - 0.00478) / m

x = np.arange(0.00478, 2.46183, 0.001)

data = p\_raspr

fig = plt.figure(figsize=(16, 9))

ax = fig.add\_subplot(111)

heights, bins, \_ = ax.hist(data, edgecolor='black', weights=(np.ones\_like(data) / len(data)) / h, bins=m)

# Печать высот столбиков

print("Высоты столбиков:")

print(heights)

# Печать весов столбиков

weights = heights \* h

print("Веса столбиков:")

print(weights)

# Создание массива частичных сумм весов столбиков

partial\_sums = np.cumsum(weights)

print("Частичные суммы весов столбиков:")

print(partial\_sums)

plt.plot(x, expon.pdf(x, scale=1/lbd), color='red')

plt.show()

# Печать значения h

print("Значение h:")

print(h)

temp = [0.00478, 0.31191125000000003, 0.6190425, 0.92617375, 1.233305, 1.5404362500000002, 1.8475675, 2.15469875, 2.46183]

s = [0.39 , 0.725, 0.87 , 0.94 , 0.97 , 0.98 , 0.99 , 1. ]

ch =[0.39 , 0.335 , 0.145 , 0.07 , 0.03 , 0.01 , 0.01 , 0.01 ]

own\_emp2(temp, s)

a, b, c = frequency(p\_raspr)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

own\_emp2(a,sum)

h = (temp[7]-temp[0])/8

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats2(p\_raspr,h)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",round(temp4,5),

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

sum = 0

for i in range(8):

p = round(expon.cdf(temp[i+1], scale=1/2.14) - expon.cdf(temp[i], scale=1/2.14),5)

print("(",temp[i],",",temp[i+1],") ",ch[i]," ",p," ", round(abs(ch[i]-p),5))

sum+=p

print(sum," ",1-sum)

lbd = 2.14

print('\nВыборочное среднее: ', round(1/lbd,5))

print('\nВыборочная дисперсия: ',round(1/(lbd\*\*2),5))

print('\nВыборочное среднее квадратичное отклонение: ',round(1/lbd,5))

print('\nВыборочная мода: ',0)

print('\nВыборочная медиана: ',round(math.log(2)/lbd,5))

print('\nВыборочный к. асимметрии: ',2)

print('\nВыборочный к. эксцесса: ', 6)

print('\nВыборочное среднее, ', 0.49642 , ",", 0.46729, ",", ab(0.49642 , 0.46729), ",", ot(0.49642 , 0.46729))

print('\nВыборочная дисперсия,', 0.17212, ",", 0.21836, ",", ab(0.17212, 0.21836), ",", ot(0.17212, 0.21836))

print('\nВыборочное среднее квадратичное отклонение,',0.41487, ",", 0.46729, ",", ab(0.41487, 0.46729), ",", ot(0.41487, 0.46729))

print('\nВыборочная мода,',0.65343 , ",", 0, ",", ab(0.65343 , 0), ", -")

print('\nВыборочная медиана,', 0.43251, ",", 0.3239, ",", ab(0.43251, 0.3239), ",", ot(0.43251, 0.3239))

print('\nВыборочный коэффициент асимметрии,', 1.76714, ",", 2, ",", ab(1.76714, 2), ",", ot(1.76714, 2))

print('\nВыборочный коэффициент эксцесса,', 4.30184, ",", 6, ",", ab(4.30184, 6), ",", ot(4.30184, 6))

#Равномерное распределение

s = interval(random\_numbers)

def interval(data):

# Вычисляем минимальное и максимальное значения в массиве

min\_val = 0.28

max\_val = 6.28

# Вычисляем размах значений (диапазон)

range\_val = max\_val - min\_val

# Определяем количество интервалов (можно выбрать любое число)

num\_intervals = 8

# Вычисляем длину интервала

interval\_length = 0.75

# Создаем список границ интервалов

interval\_boundaries = [min\_val + i \* interval\_length for i in range(num\_intervals)]

interval\_boundaries.append(max\_val)

# Создаем список счетчиков для каждого интервала

interval\_counts = [0] \* num\_intervals

# Проходим по всем значениям в массиве и увеличиваем счетчик соответствующего интервала

for value in data:

for i in range(num\_intervals):

if value >= interval\_boundaries[i] and value <= interval\_boundaries[i+1]:

interval\_counts[i] += 1

break

# Выводим результаты

print("Интервалы: ", interval\_boundaries)

print("Частоты: ", interval\_counts)

s = []

s.append(interval\_counts[0]/200)

for i in range(1,8):

t = s[i-1]+interval\_counts[i]/200

s.append(t)

return s

import matplotlib.pyplot as plt

import math

import numpy as np

from scipy.stats import uniform

import scipy.stats

m = 1 + math.floor(math.log2(200))

h = (6.28 - 0.28) / m

x = np.arange(min(random\_numbers), max(random\_numbers), 0.001)

data = random\_numbers

fig = plt.figure(figsize=(16, 9))

ax = fig.add\_subplot(111)

heights, bins, \_ = ax.hist(data, edgecolor='black', weights=(np.ones\_like(data) / len(data)) / h, bins=m)

# Печать высот столбиков

print("Высоты столбиков:")

print(heights)

# Печать весов столбиков

weights = heights \* h

print("Веса столбиков:")

print(weights)

# Создание массива частичных сумм весов столбиков

partial\_sums = np.cumsum(weights)

print("Частичные суммы весов столбиков:")

print(partial\_sums)

plt.plot(x, uniform.pdf(x, loc=0.28, scale=6), color='red')

plt.show()

# Печать значения h

print("Значение h:")

print(h)

a, b, c = frequency(random\_numbers)

print("\n","x n w s")

sum = []

for i in range(len(a)):

if i == 0:

sum.append(c[i])

else:

sum.append(c[i]+sum[i-1])

print("\n", a[i]," ",b[i], " ", c[i], " ", round(sum[i],5))

own\_emp2(a,sum)

temp = [0.28, 1.03, 1.78, 2.5300000000000002, 3.2800000000000002, 4.03, 4.78, 5.53, 6.28]

s = [0.12 , 0.26, 0.34, 0.46, 0.6, 0.775, 0.85, 1. ]

ch = [0.12 , 0.14 , 0.08 ,0.12 , 0.14 , 0.175 ,0.075, 0.15 ]

own\_emp2(temp, s)

h = (temp[7]-temp[0])/8

temp1, temp2, temp3, temp4, temp5, temp6, temp7 = osn\_stats2(random\_numbers,h)

print("\n","Выборочное среднее: ", temp1,

"\n","Выборочная диспрерсия с поправкой Шеппарда: ", temp2,

"\n","Выборочное среднее квадратичное отклонение: ", temp3,

"\n","Выборочная мода: ",round(temp4,5),

"\n","Выборочная медиана: ",temp5,

"\n","Выборочный коэффициент асимметрии: ",temp6,

"\n","Выборочный коэффициент эксцесса: ",temp7)

sum = 0

a = 0.28 # начало отрезка

b = a+6 # конец отрезка

for i in range(8):

p = round((temp[i+1] - temp[i]) / (b - a),5)

print("(",temp[i],",",temp[i+1],") ",ch[i]," ",p," ", round(abs(ch[i]-p),5))

sum+=p

print(sum," ",1-sum)

a = 0.28 # начало отрезка

b = a+6 # конец отрезка

print('\nВыборочное среднее: ', round((a+b)/2,5))

print('\nВыборочная дисперсия: ',round(((b-a)\*\*2)/12,5))

print('\nВыборочное среднее квадратичное отклонение: ',round((b-a)/(2\*math.sqrt(3)),5))

print('\nВыборочная мода: ',(a+b)/2)

print('\nВыборочная медиана: ',round((a+b)/2,5))

print('\nВыборочный к. асимметрии: ',0)

print('\nВыборочный к. эксцесса: ', -6/5)

print('\nВыборочное среднее, ', 3.34944 , ",", 3.28, ",", ab(3.34944 , 3.28), ",", ot(3.34944 , 3.28))

print('\nВыборочная дисперсия,', 2.922 , ",", 3.0, ",", ab(2.922 , 3.0), ",", ot(2.922 , 3.0))

print('\nВыборочное среднее квадратичное отклонение,', 1.70939, ",", 1.73205, ",", ab(1.70939, 1.73205), ",", ot(1.70939, 1.73205))

print('\nВыборочная мода,', 4.34371, ",",3.2800000000000002, ",", ab(4.34371, 3.2800000000000002), ",", ot(4.34371, 3.2800000000000002))

print('\nВыборочная медиана,', 3.48609 , ",", 3.28, ",", ab(3.48609 , 3.28), ",", ot(3.48609 , 3.28))

print('\nВыборочный коэффициент асимметрии,', -0.08018 , ",", 0, ",", ab(-0.08018 , 0), ", -")

print('\nВыборочный коэффициент эксцесса,', -1.10246, ",", -1.2, ",", ab(-1.10246, -1.2), ",", ot(-1.10246, -1.2))