Национальный Исследовательский Университет
Высшая Школа Экономики
Московский Институт Электроники и Математики

Департамент Прикладной Математики
Кафедра Компьютерной Безопасности

Долгосрочное домашнее задание по математической статистике

Дискретное распределение: *Биномиальное распределение* $n = 87, \theta = 0.6$
Равномерное распределение: *Распределение Максвелла* $\theta = 3$

Выполнил
Смирнов Д. А.

Проверил
Чухно А. Б.

Москва 2022

Оглавление

Домашнее задание 1.

Характеристики вероятностных распределений

Описание основных характеристик распределений

1. Биномиальное распределение

$$P(x) = \binom{n}{x}\theta^x(1-\theta)^{n-x}, x \in \{0, 1, ..., n\}, n \in \mathbb{N}, 0 < \theta < 1$$

1.1.1. Функция Распределения

В общем виде функция распределения Биномиального распределения выглядит следующим образом [***1*** *- Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности МИЭМ НИУ ВШЭ, Москва 2022:* ***стр. 90***]:

$$F(x) = P(\xi < x) = \sum_{k=0}^{x}\binom{n}{x}\theta^k(1-\theta)^{n-k}, x \in \{0, 1, ..., n\}, n \geq 1, 0 < \theta < 1$$

1.1.2. Математическое ожидание

По определению, математическое ожидание дискретной случайной величины вычисляется по формуле [***2*** *- Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 63***]:

$M\xi = \sum\limits_{i \geq 1} x_i \cdot p_i$, где $p_i = P(\xi = x_i)$

Пусть случайная величина ξ имеет биномиальное распределение с параметрами (n, θ), что соответствует числу успехов в n независимых испытаниях Бернулли с вероятностью успеха θ в каждом испытании [***3*** *- Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 67***].

Представим ξ в виде суммы n независимых индикаторов $\chi_1, ..., \chi_n$, где $\chi_i = 1$, если в i-ом испытании Бернулли произошел успех.

Имеем: $P(\chi_i = 1) = \theta, P(\chi_i = 0) = 1 - \theta, i \in \overline{1, n}$

Тогда: $M\xi = M(\sum\limits_{i=1}^{n}\chi_i) = \sum\limits_{i=1}^{n} M\chi_i = n\theta$

1.1.3. Дисперсия

По определению, Дисперсия $D\xi = M(\xi - M\xi)^2 = M\xi^2 - (M\xi)^2$ [***4*** - *Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 71***]
Также по ***Определению 7.5*** [**5** - *Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 71***] знаем, что для дискретной случайной величины
$M(\xi - M\xi)^n = \sum\limits_{k=1}^{n}(x_k - M\xi)^n \cdot P(\xi = x_k)$
Тогда: $D\xi = \sum\limits_{k=1}^{n=2}(\chi_k - M\xi)^2 \cdot p_k$. С учетом функции распределения из пункта **1.1.1**, получаем, что

$$D\xi = \sum_{k=1}^{n=2}(\chi_k - M\xi)^2 \cdot p_k = \sum_{x=0}^{n=2}(x - n\cdot\theta)^2 \cdot \binom{n}{x} \cdot \theta^x \cdot (1-\theta)^{n-x} = n\cdot\theta\cdot(1-\theta)$$

1.1.4. Квантиль уровня γ

γ-квантиль - это такое значение χ случайной величины , для которого $P(\leq \chi\cdot\gamma) = \gamma$. То есть, вероятность того, что случайная величина примет значение меньше или равное x [***6*** - *Источник*]
$\sum\limits_{k=1}^{x}\binom{n}{x}\theta^k \cdot (1-\theta)^{n-k} = \gamma$

1.1.5. Пример интерпритации распределения

Биномиальное распределение - распределение количества «успехов» в последовательности из n независимых случайных экспериментов, таких что вероятность «успеха» в каждом из них равна θ. [***7*** - *Источник*] Такая схема испытаний (экспериментов) называется схемой испытаний Бернулли. [***8*** - *Источник*]
Примеры:

1. **Контроль качества изделий** [***9*** - *Источник*]
   Каждое изделие с вероятностью θ может быть дефектным. Появление как дефектных, так и стандартных изделий происходит независимо друг от друга.
   Пример:
   Дано: $n = 87, \theta = 0.6-$ вероятность того, что изделие "стандартно"
   Решение: Случайная величина ξ распределена по Биномиальному закону с вышеописанными параметрами. Воспользуемся формулой Бернулли:

$P(\xi = x) = P_n(x) = \binom{n}{x}\theta^x(1-\theta)^{n-x} = \binom{87}{x}0.6^x(1-0.6)^{87-x} = \dfrac{87!}{x!(87-x)!} \cdot 0.6^x \cdot 0.4^{87-x}, x = 0, 1, 2, \cdots, 87$

С помощью скрипта на языке Python составим таблицу, описывающую ряд распределения:

	Pi		
0	0.000000000000000	44	0.017591757413457
1	0.000000000000000	45	0.025214852292621
2	0.000000000000000	46	0.034533384661634
3	0.000000000000000	47	0.045187301206180
4	0.000000000000000	48	0.056484126507725
5	0.000000000000000	49	0.067435130626570
6	0.000000000000000	50	0.076876048914290
7	0.000000000000000	51	0.083659229700845
8	0.000000000000000	52	0.086876892381646
9	0.000000000000000	53	0.086057299057291
10	0.000000000000000	54	0.081276337998553
11	0.000000000000000	55	0.073148704198698
12	0.000000000000000	56	0.062698889313169
13	0.000000000000000	57	0.051149093913375
14	0.000000000000000	58	0.039684641829343
15	0.000000000000000	59	0.029259015586041
16	0.000000000000002	60	0.020481310910229
17	0.000000000000012	61	0.013598247407611
18	0.000000000000068	62	0.008553736272529
19	0.000000000000371	63	0.005091509686029
20	0.000000000001891	64	0.002863974198392
21	0.000000000009052	65	0.001520109382223
22	0.000000000040733	66	0.000760054691112
23	0.000000000172674	67	0.000357339145821
24	0.000000000690697	68	0.000157649623156
25	0.000000002610834	69	0.000065116148695
26	0.000000009338753	70	0.000025116228782
27	0.000000031647995	71	0.000009020617380
28	0.000000101725698	72	0.000003006872460
29	0.000000310438769	73	0.000000926775758
30	0.000000900272430	74	0.000000263003931
31	0.000002483009443	75	0.000000068381022
32	0.000006517899787	76	0.000000016195505
33	0.000016294749468	77	0.000000003470465
34	0.000038819844320	78	0.000000000667397
35	0.000088176503527	79	0.000000000114049
36	0.000191049090975	80	0.000000000017107
37	0.000395006904314	81	0.000000000002218
38	0.000779618890092	82	0.000000000000243
39	0.001469281754405	83	0.000000000000022
40	0.002644707157929	84	0.000000000000002
41	0.004547606210586	85	0.000000000000000
42	0.007471067345962	86	0.000000000000000
43	0.011727838275638	87	0.000000000000000

Рис. 1.1:

2. **Телекоммуникации** [***10*** - *Источник*]
   Здесь $(1-\theta)$ - доля необслуженных вызовов.
   Пример для данной интерпритации анологичен примеру, описанному выше.

1.1.6. Соотношения между распределениями

[***11*** - *Источник*]

1. Если $n = 1$, то получаем распределение Бернулли.

2. Если n большое, то в силу центральной предельной теоремы [***12*** - *Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности": **стр. 124***]
   $Bin(n,\theta) \approx N(n\theta, n\theta(1-\theta))$, где $N(n\theta, n\theta(1-\theta))$ - нормальное распределение с $M\xi = n\theta$ и $D\xi = n\theta(1-\theta)$.

3. Если n - большое, а λ - фиксированное число, то $Bin(n, \frac{\lambda}{n}) \approx P(\lambda)$, где $P(\lambda)$ - распределение Пуассона с параметром λ.

1.1.7. Описание способа моделирования выбранных случайных величин

[***13*** - *Источник, **стр. 26***] Биномиальное распределение $Bin(n,\theta)$ с параметрами $n \in \mathbb{N}$и$\theta \in (0,1)$ можно задать с помощью **таблицы распределения:**
$P : \begin{pmatrix} 0 & 1 & \cdots & k & \cdots & n \\ \theta_0 & \theta_1 & \cdots & \theta_k & \cdots & \theta_n \end{pmatrix}$, где $\theta_k = \binom{n}{k}\theta^k(1-\theta)^{n-k}$.
Воспользуемся последовательным методом набором обратных функций в нетабличном варианте. Накопленные вероятности s_k здесь вычисляются в том же цикле, где проверяются неравенства $\alpha \leq s_k$. То есть вероятности θ_k рекуррентно пересчитываются одна через другую, а накопленные вероятности s_k последовательно вычисляются через s_{k-1}иθ_k
Для биномиального распределения $\theta_0 = (1-\theta)^n$ и

$$\frac{\theta_k}{\theta_{k-1}} = \frac{n-k+1}{k}\frac{\theta}{1-\theta}$$

при $k = 1, 2, \cdots, n,$ то мы приходим к алгоритму BIS (Binomial Inverse Sequential).
Однако, воспользуемся мы алгоритмом BISM: заметим, что если $\xi \in Bin(n,\theta)$, то $\eta = n - \xi \in Bin(n, 1-\theta)$. Поэтому, если $\theta > 0.5$, то можно применить алгоритм BIS к моделированию числа неудач в n испытаниях Бернулли с вероятностью успеха θ, а потом перейти к числу успехов, что равносильно применению последовательного метода обратных функций «справа налево», а не «слева направо».

Далее описан **Алгоритм BISM (Binomial Inverse Sequential Modified)**:

Моделирование $Bin(n, \theta)$ модифицированным последовательным методом обратных функций.

Входные данные: n, θ
Результат: ξ.
1. Инициализация

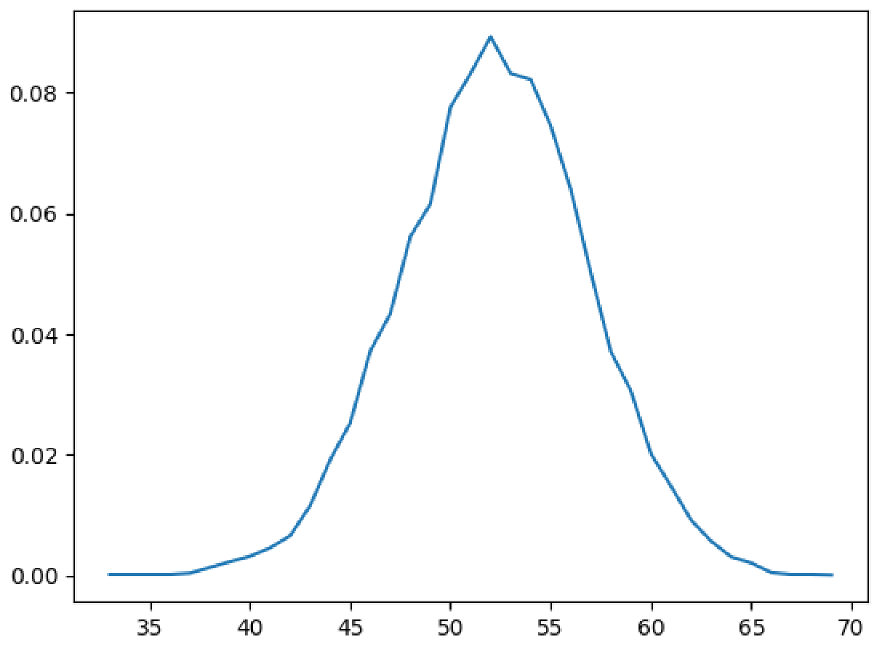
$\triangleright$ If $\theta \leq 0.5$ then $t \longleftarrow \theta$ else $t \longleftarrow 1 - \theta$;

$\triangleright$ $c \longleftarrow t/(1-t); s \longleftarrow r \longleftarrow (1-t)^n; k \longleftarrow 0; Get(\alpha)$;

2. Пересчет вероятностей и поиск окна

$\triangleright$ $k \longleftarrow k+1; r \longleftarrow r \cdot c \cdot (n-k+1)/k; s \longleftarrow s + r$;

2. Завершение: If $p \leq 0.5$ then $\xi \longleftarrow k$ else $\xi \longleftarrow n - k$; STOP.
Комментария требует лишь введение переменной t, которая необходима при завершении алгоритма, чтобы (если нужно) перейти от моделирования числа неудач в испытаниях Бернулли к моделированию числа успехов.
В процессе выполнения Домашнего Задания мною была разработана программа на языке Python, моделирующая 10000 случайных величин с Биномиальным распределением и строющая график плотности вероятности. Параметры: $n = 87, \theta = 0.6$
График:



Листинг кода:

```
import random
import matplotlib.pyplot as plt

def binomial(n, theta):

    if theta <= 0.5:
        t = theta
    else:
        t = 1 - theta

    c = t / (1 - t)
    r = (1 - t) ** n
    s = r
    k = 0
    alpha = random.uniform(0, 1)

    while alpha > s:
        k += 1
        r *= c * (n - k + 1)/k
        s += r

    if theta <= 0.5:
        return k
    else:
        return n - k

binomial_nums = []
for i in range(10000):
    number = binomial(87, 0.6)
    binomial_nums.append(number)

array = dict((i, binomial_nums.count(i)
                /len(binomial_nums)) for i in binomial_nums)
dictionary = dict(sorted(array.items(), key=lambda x: x[0]))
print(dictionary, dictionary.values())
plt.plot(dictionary.keys(), dictionary.values())
plt.show()
```

2. Распределение Максвелла

$$f(x) = \sqrt{\frac{2}{\pi}} \frac{x^2}{\theta^3} e^{-\frac{x^2}{2\theta^2}}, x, \theta \in \mathbb{R}^+$$

1.2.1. Функция Распределения

По определению [***14*** - *Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности": **стр. 52***], функция распределения непрывного распределения вычисляется по формуле: $F(x) = \int\limits_{-\infty}^{x} f(t)\mathrm{d}t$. Тогда:

$$F_\xi(x) = \int\limits_{-\infty}^{x} f(t)\,\mathrm{d}t = \int\limits_{-\infty}^{x} \sqrt{\frac{2}{\pi}} \frac{t^2}{\theta^3} e^{-\frac{t^2}{2\theta^2}}\,\mathrm{d}t = \sqrt{\frac{2}{\pi}} \frac{1}{\theta^3} \int\limits_{-\infty}^{x} t^2 e^{-\frac{t^2}{2\theta^2}}\,\mathrm{d}t =$$

опираясь на то, что $d(e^{-\frac{t^2}{2\theta^2}}) = -\frac{te^{-\frac{t^2}{2\theta^2}}}{\theta^2}\,\mathrm{d}t$, получим $= -\sqrt{\frac{2}{\pi}} \frac{1}{\theta^3} \theta^2 \int\limits_{-\infty}^{x} \frac{t^2}{t} d(e^{-\frac{t^2}{2\theta^2}}) =$ $-\sqrt{\frac{2}{\pi}} \frac{1}{\theta} \int\limits_{-\infty}^{x} t\, d(e^{-\frac{t^2}{2\theta^2}})$ = возьмем интеграл по частям = $-\sqrt{\frac{2}{\pi}} \frac{1}{\theta} (te^{-\frac{t^2}{2\theta^2}} \Big|_{-\infty}^{x} -$ $- \int\limits_{-\infty}^{x} e^{-\frac{t^2}{2\theta^2}}\,\mathrm{d}t) = -\sqrt{\frac{2}{\pi}} \frac{1}{\theta} x e^{-\frac{x^2}{2\theta^2}} + \sqrt{\frac{2}{\pi}} \frac{1}{\theta} \theta\sqrt{2\pi} \int\limits_{-\infty}^{x} \frac{1}{\theta\sqrt{2\pi}}^{-\frac{t^2}{2\theta^2}}\,\mathrm{d}t) =$

так как $x, \theta \in \mathbb{R}^+ = -\sqrt{\frac{2}{\pi}} \frac{1}{\theta} x e^{-\frac{x^2}{2\theta^2}} + \sqrt{\frac{2}{\pi}} \frac{1}{\theta} \theta\sqrt{2\pi} \int\limits_{0}^{x} \frac{1}{\theta\sqrt{2\pi}}^{-\frac{t^2}{2\theta^2}}\,\mathrm{d}t) =$

видно, что правое слагаемое есть функция плотности стандартного нормального распределения $= -\sqrt{\frac{2}{\pi}} \frac{1}{\theta} x e^{-\frac{x^2}{2\theta^2}} + 2 \int\limits_{0}^{x} f_N(t)\,\mathrm{d}t = -\sqrt{\frac{2}{\pi}} \frac{1}{\theta} x e^{-\frac{x^2}{2\theta^2}} + 2F_N(x)$

Здесь $F_N(x)$ - функция распределения N(0,1).

1.2.2. Математическое ожидание

По определению [***15*** - *Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности": **стр. 68***], Математическое ожидание непрерывной случайной величины ξ вычисляется по формуле:

$M\xi = \int_{-\infty}^{+\infty} x \mathrm{d}F_{\xi}(x)$. Тогда:

$$M\xi = \int\limits_{\mathbb{R}} xf\xi(x)\mathrm{d}x = \int\limits_{-\infty}^{+\infty} x\sqrt{\frac{2}{\pi}}\frac{x^2}{\theta^3}\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x = \sqrt{\frac{2}{\pi}}\frac{1}{\theta^3}\int\limits_{-\infty}^{+\infty} x^3\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x =$$

$$= \text{так как } x, \theta \in \mathbb{R}^+, \text{ то} = \sqrt{\frac{2}{\pi}}\frac{1}{\theta^3}\int\limits_{0}^{+\infty} x^3\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x$$

Для упрощения работы, сначала вычислим неопределённый интеграл:

$$\int x^3\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x = \text{Произведём замену вида } u = x^2x^2, \text{ тогда } \mathrm{d}x = \frac{1}{2x}\mathrm{d}u =$$

$$= \frac{1}{2}\int u\exp(-\frac{u}{2\theta^2})\mathrm{d}u = \text{Произведём замену вида } v = -\frac{u}{2\theta^2}, \text{ тогда } \mathrm{d}u = -2\theta^2 dv =$$

$$= 2\theta^4\int ve^v\mathrm{d}v = \text{Возьмём интеграл по частям} = 2\theta^4(ve^v - \int e^v\mathrm{d}v) = 2\theta^4(ve^v - e^v) =$$

$$= \text{Выполним обратную замену} = -(\theta^2x^2 + 2\theta^4)\exp(-\frac{x^2}{2\theta^2}) + \mathbb{C}$$

Таким образом:

$$-(\theta^2x^2 + 2\theta^4)\exp(-\frac{x^2}{2\theta^2})\Big|_0^{+\infty} = 2\theta^4$$

Следовательно, математическое ожидание равно: $\sqrt{\frac{2}{\pi}}\frac{1}{\theta^3}2\theta^4 = \sqrt{\frac{2}{\pi}}2\theta$

1.2.3. Дисперсия

По определению, Дисперсия $D\xi = M(\xi - M\xi)^2 = M\xi^2 - (M\xi)^2$ [***16*** *- Фомин Д. Б, Чухно А. Б., "Теория вероятности. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 71***]

В пункте **1.2.2** было выведено $M\xi = \sqrt{\frac{2}{\pi}}2\theta$, тогда $(M\xi)^2 = \frac{8}{\pi}\theta^2$.

Выведем $M\xi^2$:

$$M\xi^2 = \int\limits_{\mathbb{R}} x^2f\xi(x)\mathrm{d}x = \int\limits_{-\infty}^{+\infty} x^2\sqrt{\frac{2}{\pi}}\frac{x^2}{\theta^3}\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x = \sqrt{\frac{2}{\pi}}\frac{1}{\theta^3}\int\limits_{-\infty}^{+\infty} x^4\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x =$$

$$= \text{так как } x, \theta \in \mathbb{R}^+, \text{ то} = \sqrt{\frac{2}{\pi}}\frac{1}{\theta^3}\int\limits_{0}^{+\infty} x^4\exp(-\frac{x^2}{2\theta^2})\mathrm{d}x =$$

= это следующий интеграл

$$\int_0^{+\infty} x^n \exp(-\alpha x^2)\mathrm{d}x = \begin{cases} \dfrac{(2k-1)!!}{2^{k+1}\alpha^k}\sqrt{\dfrac{\pi}{\alpha}} & \text{если } n = 2k, k \in \mathbb{N}, \alpha > 0 \\ \dfrac{k!}{2\alpha^{k+1}} & \text{если } n = 2k+1, k \in \mathbb{N}, \alpha > 0 \end{cases} =$$

$$= \sqrt{\frac{2}{\pi}}\frac{1}{\theta^3}\frac{3}{8}\sqrt{\pi}(2\theta^2)^{2.5} = 3\theta^2$$

Таким образом, получаем дисперсию:

$$D\xi = M(\xi - M\xi)^2 = M\xi^2 - (M\xi)^2 = 3\theta^2 - \frac{8}{\pi}\theta^2 = \theta^2(3 - \frac{8}{\pi})$$

1.2.4. Квантиль уровня γ

По определению: $F\xi(x_\gamma) = \gamma$, значит квантиль распределения - решение уравнения

$$-\sqrt{\frac{2}{\pi}}\frac{1}{\theta}x_\gamma \exp(-\frac{x_\gamma^2}{2\theta^2}) + 2F_N(x_\gamma) = \gamma$$

Здесь $F_N(x)$ - функция распределения N(0,1)

1.2.5. Пример интерпритации распределения

[***17*** - *Источник*] Распределение Максвелла (Максвелла-Больцмана) лежит в основе кинетической теории газов, объясняющей многие фундаментальные свойства газов, включая давление и диффузию. Распределение Максвелла применимо к множеству свойств индивидуальных молекул в газе. О нём обычно думают как о распределении энергий молекул в газе, но оно может также применяться к распределению скоростей, импульсов, и модуля импульсов молекул.

Ниже приведены некоторые примеры интепритации распределения Максвелла в виде функций плотности распределения:

Распределение по вектору скорости: [***18*** - *Источник*]

$$p = mV$$

$$f_V(v_x, v_y, v_z) = \sqrt{\left(\frac{m}{2\pi kT}\right)^3} \cdot exp\left[\frac{-m(v_x^2, v_y^2, v_z^2)}{2kT}\right]$$

Распределение по проекции скорости:

$$f(v_i) = \sqrt{\frac{m}{2\pi kT}} \cdot exp\left[\frac{-mv_i^2}{2kT}\right]$$

Распределение по модулю импульса:

$$f_p = \int_{\theta=0}^{\pi}\int_{\phi=0}^{2\pi} f_P p^2 sin(\theta) d\theta d\phi = 4\pi\sqrt{\left(\frac{1}{2\pi mkT}\right)^3} \cdot p^2 \cdot exp\left[\frac{-p^2}{2mkT}\right]$$

Распределение по энергии:

$$p^2 = 2mE \text{ и } f_E dE = f_p dp$$
$$f_E = f_p\frac{dp}{dE} = \frac{2\pi}{\sqrt{(\pi kT)^3}} \cdot \sqrt{E} \cdot exp\left[\frac{-E}{kT}\right]$$

[***19*** - *Источник*] А также распределение Максвелла можно записать как дискретное распределение по множеству состояний молекулы, нумеруемых символом i:

$$\frac{N_i}{N} = \frac{exp(-E_i/kT)}{\sum_j exp(-E_j/kT)}$$

Через E_i и N_i обозначены энергия молекулы в i-м состоянии и число таких молекул соответственно, T — температура системы, N — общее число молекул в системе и k — постоянная Больцмана.

1.2.6. Соотношения между распределениями

[***20*** - *Источник*] Распределение Максвелла представляет собой величину 3-мерного вектора, компоненты которого независимы и нормально распределены со средним значением 0 и стандартным отклонением a. Если каждая случайная величина ξ_i распределена как:

$$\xi \sim N(0, a^2)$$

Тогда:

$$Maxwell = \sqrt{\xi_1^2 + \xi_2^2 + \xi_3^2}$$

Тогда распределение Максвелла можно связать с распределением χ^2 с 3мя степенями свободы:

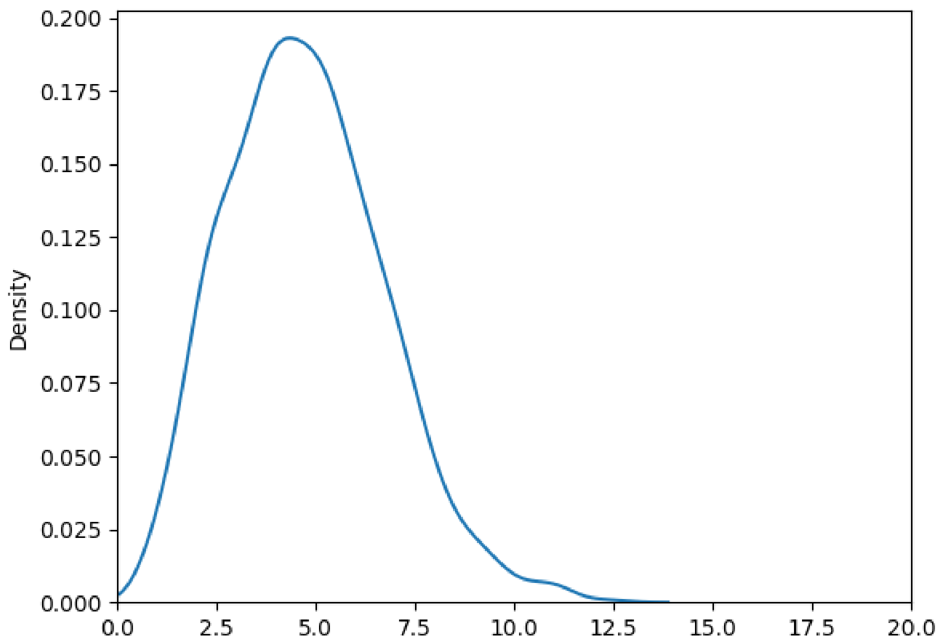
$$Maxwell = a\chi^2(3)$$

1.2.7. Описание способа моделирования выбранных случайных величин

[*21 - Источник*] Для моделирования случайных величин, распределенных по Максвеллу, будем использовать Алгоритм Джонка (Алгоритм Принятия-Отмены) - *en: Johnk's Algorithm (Acceptance-Rejection Algorithm).*
Описание алгоритма:
Входные данные: $\xi_1, \xi_2, \xi_3, \xi_4$ - случайные числа из интервала $(0, 1)$
Выходные данные: ξ

1. $r = -ln\xi_1$ - случайное число из экспоненциального распределения.
2. $w_1 = \xi_2^2, w_2 = \xi_3^2$
3. If $w = w_1 + w_2 > 1$: возвращаемся на шаг **2**
4. $r = r - \frac{w_1}{w} ln\xi_4$
5. $\xi = \theta\sqrt{2r}$

График:



Листинг кода:

```
import matplotlib.pyplot as plt
import numpy as np
import random
import seaborn as sns
```

```
theta = 3.0
def JohnksAlgorithm(xi1, xi2, xi3, xi4):
    r = -np.log(xi1)
    w1 = xi2 ** 2
    w2 = xi3 ** 2
    w = w1 + w2
    if w <= 1:
        r = r - (np.log(xi4) * w1/w)
        return theta * np.sqrt(2 * r)

maxwell_nums = []
while len(maxwell_nums) < 1000:
    xi1 = random.uniform(0, 1)
    xi4 = random.uniform(0, 1)
    xi2 = random.uniform(0, 1)
    xi3 = random.uniform(0, 1)
    number = JohnksAlgorithm(xi1, xi2, xi3, xi4)
    if number is not None:
        maxwell_nums.append(number)

sns.kdeplot(maxwell_nums)
plt.xlim(0, 20)
plt.show()
```

Домашнее задание 2.

Основные понятия математической статистики

1. Биномиальное распределение

2.1.1. Генерация выборок

В данном разделе производится генерация выборок объёмов 5, 10, 100, 200, 400, 600, 800, 1000 биномиального распределения с параметрами $n = 87, \theta = 0.6$

5

53	59	60	49	51

Таблица 1.1: выборка из Биномиального распределения, n = 5

10

58	50	52	54	48	53	51	55	51	48

Таблица 1.2: выборка из Биномиального распределения, n = 10

100

46	53	50	58	57	55	60	52	50	53
49	53	49	50	53	52	50	61	57	55
46	49	58	55	57	51	50	64	46	50
57	45	48	44	49	60	51	57	57	54
48	56	53	50	53	48	53	45	52	46
53	54	52	51	52	53	49	50	52	56
58	54	59	51	54	46	46	54	60	55
51	49	39	61	44	55	43	48	58	56
46	49	53	48	54	55	49	49	54	60
65	55	57	53	53	51	53	51	50	50

Рис. 2.1: Таблица 1.3: выборка из Биномиального распределения, n = 100

200

48	50	53	49	53	48	49	52	52	52
51	48	52	53	52	57	55	49	47	50
48	51	59	59	57	60	49	48	49	53
50	50	51	51	55	57	55	57	52	48
50	52	52	51	57	57	49	52	49	52
50	57	56	52	48	43	49	53	55	51
44	51	53	47	45	47	58	56	58	60
52	53	62	51	55	40	61	49	45	48
46	50	58	50	51	54	54	53	55	53
50	41	54	50	52	57	48	57	52	51
48	49	48	57	56	50	46	55	57	50
45	52	52	54	54	54	48	53	55	52
47	56	43	64	48	49	57	58	53	43
51	43	50	52	55	63	54	51	51	42
47	60	61	47	47	50	60	52	56	50
53	51	53	54	50	52	51	50	53	52
49	47	50	54	62	50	50	48	49	55
49	56	50	50	56	51	55	46	42	48
48	44	50	55	42	56	52	60	60	57
57	53	57	59	59	57	58	58	49	55

Рис. 2.2: Таблица 1.4: выборка из Биномиального распределения, n = 200

400

50	58	48	50	50	60	54	55	42	50
50	57	45	50	48	52	45	49	46	52
51	53	57	58	52	56	62	46	56	60
53	45	52	52	55	53	57	49	55	49
52	56	52	59	47	46	52	52	54	52
54	59	65	56	60	48	63	48	53	55
55	51	45	57	47	46	61	55	43	52
48	46	57	47	57	45	44	58	42	62
57	50	58	68	52	53	48	50	47	43
54	59	55	48	46	49	45	55	48	50
61	62	55	57	52	52	47	45	58	52
56	50	48	53	50	51	52	51	61	48
52	52	54	53	58	56	49	54	46	62
55	56	54	57	61	52	57	57	57	64
50	47	48	50	47	58	51	55	50	59
49	43	43	54	48	52	45	58	47	60
56	54	50	44	39	58	41	57	47	48
52	51	55	48	54	57	59	50	57	57
49	54	57	44	58	50	57	52	57	57
57	55	48	52	47	62	55	52	46	54
53	51	46	41	55	56	45	48	59	55
48	55	59	47	53	55	48	45	53	55
49	47	54	57	56	47	53	48	52	48
53	52	47	54	56	45	46	53	51	53
46	58	55	55	49	48	56	56	61	55
55	47	55	58	53	54	54	57	51	53
49	58	50	59	50	49	53	52	55	59
53	57	58	55	49	58	45	54	59	50
49	50	58	51	53	55	53	57	43	49
50	59	43	61	48	53	52	56	47	56
57	49	53	50	52	48	51	43	50	48
49	46	52	51	54	61	57	55	55	56
58	53	51	51	58	53	52	48	57	51
48	51	56	46	58	55	44	37	62	53
58	54	51	47	58	44	60	51	54	52
51	56	58	51	50	53	50	48	48	50
61	49	56	54	52	50	47	46	61	55
54	57	55	57	54	51	47	58	52	46
46	42	58	48	56	54	60	50	55	50
44	48	54	49	48	51	57	48	44	41

Рис. 2.3: Таблица 1.5: выборка из Биномиального распределения, n = 400

600

43	53	47	68	55	47	56	56	49	56
52	52	55	51	47	52	52	44	51	49
52	58	51	57	51	61	63	46	47	58
61	58	52	56	53	56	44	51	52	50
55	57	47	58	45	44	50	55	51	55
55	58	53	54	56	48	43	62	59	43
53	54	55	46	43	56	57	54	59	53
64	50	56	49	56	53	54	53	55	54
48	46	49	56	58	52	56	55	51	49
49	56	54	53	57	49	49	53	51	58
52	50	53	51	45	61	48	55	53	47
53	46	56	64	61	51	48	51	61	54
55	48	49	51	47	49	54	43	54	52
50	48	52	59	52	46	46	56	48	44
60	59	58	54	45	54	50	59	53	46
56	53	47	56	58	49	54	57	55	51
53	48	50	53	49	55	62	55	52	49
51	48	53	49	50	49	47	50	54	51
40	51	52	50	51	52	46	53	47	54
57	48	55	60	48	49	59	55	56	58
55	58	47	59	49	54	52	52	51	54
57	54	60	49	57	55	47	56	59	59
54	46	49	51	58	52	49	50	46	60
59	54	39	52	43	59	49	61	51	51
55	52	44	44	52	49	54	49	57	62
53	54	54	51	53	57	51	48	55	47
46	53	51	49	52	57	53	56	51	62
55	43	49	60	55	60	59	53	59	47
51	55	50	56	48	45	50	50	53	49
49	52	44	48	52	54	51	61	51	51
45	56	55	55	48	47	46	50	55	51
52	54	60	50	47	52	51	55	47	48
54	46	47	51	51	61	51	56	50	50
49	50	56	51	42	49	50	51	55	46
50	49	49	54	50	55	57	47	48	51
49	49	55	53	55	48	56	53	40	54
51	56	60	49	44	57	52	59	55	46
51	51	52	59	55	55	51	53	53	53
43	44	45	53	53	53	58	57	53	45
43	60	55	43	56	51	52	53	45	55
46	55	53	50	51	57	53	50	53	56
50	53	57	54	53	57	50	58	49	51
53	62	52	53	50	47	49	52	44	55
51	52	47	54	58	54	49	48	54	52
59	52	51	40	51	41	52	55	46	40

45	60	54	57	50	54	54	58	52	51
48	54	55	63	50	57	56	55	52	52
57	51	59	50	57	52	50	49	50	50
55	56	50	53	57	50	56	57	59	50
51	51	50	58	51	54	45	53	56	45
54	51	56	48	55	54	59	50	51	58
44	47	54	52	59	58	48	56	49	56
62	54	48	51	47	59	51	54	50	56
51	49	47	52	51	48	54	45	50	50
55	54	52	52	47	52	58	45	48	60
51	59	51	49	64	49	48	50	54	55
53	52	49	50	46	58	61	53	52	52
50	51	58	54	54	50	60	49	56	51
48	54	51	49	54	50	60	55	50	57
47	60	61	57	58	47	57	57	56	52

Рис. 2.4: Таблица 1.6: выборка из Биномиального распределения, n = 600

800

48	54	46	46	47	59	55	53	46	52
56	52	47	52	49	51	51	50	50	52
49	48	52	59	50	49	53	58	53	52
51	48	47	45	54	54	55	45	57	56
46	53	52	52	57	57	44	60	55	50
54	52	49	54	50	56	51	48	44	51
54	48	58	51	55	50	51	46	52	62
54	48	49	46	53	47	53	47	50	53
52	60	53	55	49	46	49	58	58	51
54	52	55	58	36	56	52	56	52	54
63	55	55	45	53	52	50	57	58	51
48	36	49	47	59	56	44	42	52	52
51	54	50	51	53	52	56	53	52	51
54	54	47	56	55	49	53	48	44	48
43	51	46	46	54	52	50	56	54	53
47	54	56	71	61	61	51	55	55	50
55	51	52	56	50	46	59	59	50	50
50	52	58	51	58	62	50	53	59	45
53	55	49	53	51	49	51	51	50	45
60	62	43	52	56	57	54	56	54	56
55	48	46	51	55	50	52	49	50	54
46	58	51	46	53	54	52	61	55	55
46	60	54	56	52	51	56	57	52	51
54	53	52	53	52	48	52	47	50	55
55	51	53	50	52	55	58	62	55	62
58	47	51	50	46	62	49	53	51	55
55	51	56	55	51	59	57	51	42	47
52	53	56	55	57	49	50	59	39	44
50	50	52	41	49	61	53	54	57	54
47	51	53	51	52	51	56	50	61	52
52	50	56	46	51	57	59	56	48	42
48	52	58	55	47	51	53	56	52	50
56	56	50	45	54	58	47	47	51	46
50	49	54	56	53	53	50	55	56	61
57	50	45	54	52	51	55	62	53	49
49	50	57	53	52	50	46	50	53	54
59	59	52	55	48	48	50	48	47	53
52	54	54	50	57	50	47	52	44	60
55	58	57	54	50	53	50	52	61	61
51	53	54	52	58	45	51	56	55	51
52	50	51	56	46	50	54	51	55	51
55	56	43	55	49	45	49	55	49	58
47	50	55	51	52	53	45	52	56	53
51	39	44	54	47	47	45	51	49	51

51	46	42	52	51	50	54	56	55	51
62	45	52	55	54	52	49	50	56	45
55	51	53	50	47	50	59	54	56	46
49	51	52	52	56	54	48	53	49	53
59	52	56	62	60	47	46	51	61	48
57	51	55	42	58	50	48	46	46	52
44	50	53	51	53	43	58	50	43	53
48	50	51	49	55	50	53	49	52	51
48	56	49	49	59	54	50	59	45	50
47	50	49	53	47	51	48	53	55	52
57	53	48	52	54	56	53	48	51	52
49	50	53	49	56	55	49	60	51	51
49	51	57	50	54	48	52	55	45	46
59	53	49	55	52	56	55	59	54	49
51	54	52	50	55	60	51	64	60	51
43	58	50	51	52	53	54	51	52	57
51	55	49	50	48	53	47	51	49	53
48	56	49	56	51	51	46	47	50	49
52	61	58	48	50	50	43	54	54	49
55	49	53	50	56	55	56	55	49	57
55	56	57	46	57	44	52	54	54	57
53	48	53	58	57	55	54	46	55	57
51	52	51	51	50	45	48	45	56	47
59	54	52	53	54	44	55	43	50	60
53	43	54	54	47	56	51	60	51	55
49	50	59	48	58	53	53	52	60	55
50	52	46	48	45	52	50	49	44	51
41	47	57	56	48	58	49	53	55	53
53	44	57	54	49	54	57	49	58	53
52	62	51	47	51	50	53	59	55	50
51	56	53	54	50	50	57	59	49	57
59	55	58	50	46	46	58	58	58	45
52	51	46	53	45	52	51	56	58	52
46	46	50	48	57	55	56	50	55	52
59	55	49	48	49	45	52	54	51	51
50	49	51	53	47	49	52	53	53	57

Рис. 2.5: Таблица 1.7: выборка из Биномиального распределения, n = 800

1000

47	55	57	59	60	52	52	56	58	55
56	48	59	52	53	56	52	51	54	53
49	61	60	54	51	44	59	52	49	53
49	50	44	52	50	52	60	54	61	62
62	45	54	51	59	55	55	53	43	52
54	44	52	57	59	59	54	50	52	47
51	58	58	50	48	47	50	51	47	51
53	57	46	51	56	56	60	43	44	53
47	52	54	48	60	54	59	51	46	54
48	47	46	54	51	57	49	59	48	54
56	52	50	63	51	55	51	60	58	50
50	54	52	49	50	54	56	52	52	54
54	57	46	53	52	55	62	46	52	61
49	53	54	54	45	56	54	54	55	53
57	49	54	51	52	64	53	58	57	54
63	57	53	48	51	48	52	59	54	65
52	50	53	53	42	57	51	59	50	58
48	56	60	55	49	55	56	47	49	51
56	57	55	55	46	47	54	53	55	47
61	50	56	52	67	53	44	58	57	53
52	49	47	53	46	46	61	55	44	58
52	50	58	48	46	55	48	46	44	52
51	53	54	49	55	51	50	55	46	60
58	58	59	46	44	57	54	54	54	54
50	52	53	56	56	54	55	54	47	50
56	54	49	57	45	55	59	52	50	47
40	44	55	50	49	50	48	44	49	49
64	54	51	51	50	49	61	52	58	53
56	51	49	53	39	58	56	50	47	53
55	55	57	52	43	56	51	51	53	56
60	47	46	53	47	56	59	54	57	52
52	43	55	60	53	51	45	49	57	45
62	53	57	64	50	53	56	47	57	56
58	51	57	57	55	60	52	51	60	45
46	60	44	59	58	48	50	54	54	56
56	60	57	55	60	50	41	59	57	58
48	51	48	51	48	50	60	45	54	51
51	50	45	52	53	54	56	58	45	50
49	45	49	62	53	51	43	57	51	52
50	57	47	54	55	53	53	57	55	50
51	49	56	50	57	49	55	60	56	56
49	59	48	51	52	50	57	55	53	45
51	53	55	55	44	49	44	55	44	56
54	55	49	43	58	49	53	47	50	63
37	47	59	48	51	45	52	43	61	54

54	53	53	59	51	52	51	45	55	47
52	52	54	48	44	59	51	55	44	48
57	49	50	49	49	51	50	46	61	57
51	53	55	51	46	51	48	53	46	46
44	50	52	51	52	52	43	57	49	56
44	52	53	47	49	50	42	48	49	49
53	57	56	55	47	52	55	46	48	56
52	55	50	51	58	50	61	51	55	53
58	53	55	47	53	49	54	51	49	57
52	41	55	49	44	49	60	49	54	57
44	48	52	59	50	61	51	48	53	55
58	51	49	58	53	49	53	54	52	55
52	50	45	60	56	48	57	58	54	47
60	48	52	53	59	54	56	44	46	57
51	55	54	56	55	66	63	49	53	59
56	49	49	53	56	45	52	65	51	54
53	47	45	49	50	48	45	44	53	52
52	56	58	51	55	44	52	55	50	56
51	57	54	52	63	53	54	45	53	43
44	56	52	53	53	54	53	56	58	42
51	54	54	56	53	55	50	48	54	47
51	50	58	46	56	60	58	44	55	43
51	48	52	54	52	45	57	45	50	52
53	52	49	49	58	56	55	60	56	52
62	54	51	51	44	50	53	51	49	49
50	54	61	49	57	51	51	57	60	53
46	59	55	50	54	50	60	54	57	50
53	47	53	51	51	53	55	52	52	55
52	53	44	55	57	55	58	58	52	57
54	50	56	50	58	49	54	62	55	52
55	54	55	51	53	59	44	54	50	51
48	56	51	45	55	49	51	55	53	58
51	54	58	52	56	62	55	51	49	52
47	56	46	56	49	55	53	50	51	60
51	51	64	50	67	53	48	51	56	56
56	50	50	54	59	55	54	53	47	46
56	52	49	59	52	51	45	55	57	51
57	49	51	57	57	43	54	50	43	59
62	53	45	50	51	49	47	48	48	51
56	57	52	54	59	51	54	54	52	53
54	51	48	51	48	59	54	45	51	55
51	57	46	59	53	55	56	53	52	46
46	47	51	48	49	56	51	51	48	53
56	49	54	49	49	47	51	54	50	57

51	53	53	53	50	50	51	49	51	49
47	48	54	50	52	47	49	57	54	58
52	54	57	56	52	52	45	52	61	48
51	57	48	44	62	51	50	56	56	59
58	50	60	45	56	52	58	50	50	49
51	57	48	58	54	63	56	54	55	47
53	50	45	55	50	49	58	50	59	48
43	48	56	45	45	57	50	52	60	54
56	50	48	56	52	47	54	47	55	49
60	50	50	45	57	53	47	51	56	57
50	51	43	55	48	50	49	55	53	50

Рис. 2.6: Таблица 1.8: выборка из Биномиального распределения, n = 1000

2.1.2. Построение эмпирической функции распределения

Из ***Определения 1.11***[*22 - Фомин Д. Б, Чухно А. Б., "Математическая статистика. Курс лекций для студентов кафедры компьютерной безопасности": **стр. 7***] можем видеть, что:
Пусть X - некоторая выборка
$\mu_n(y) = \sum_{i=1}^{n} Ind(X_i \leq y), y \in \mathbb{R}$ - случайная величина, равная числу элементов выборки X меньших или равных y. Тогда функцию:

$$\widehat{F}_n(y) = \frac{\mu_n(y)}{n}$$

Будем называть Эмпирической Функцией Распределения, соответствующей выборке X.
Эмпирическая кумулятивная функция распределения, возвращающая на основе выборки и числа t долю значений в выборке, меньших t, представлена в листинге ниже (под именем CDF).
Листинг кода:

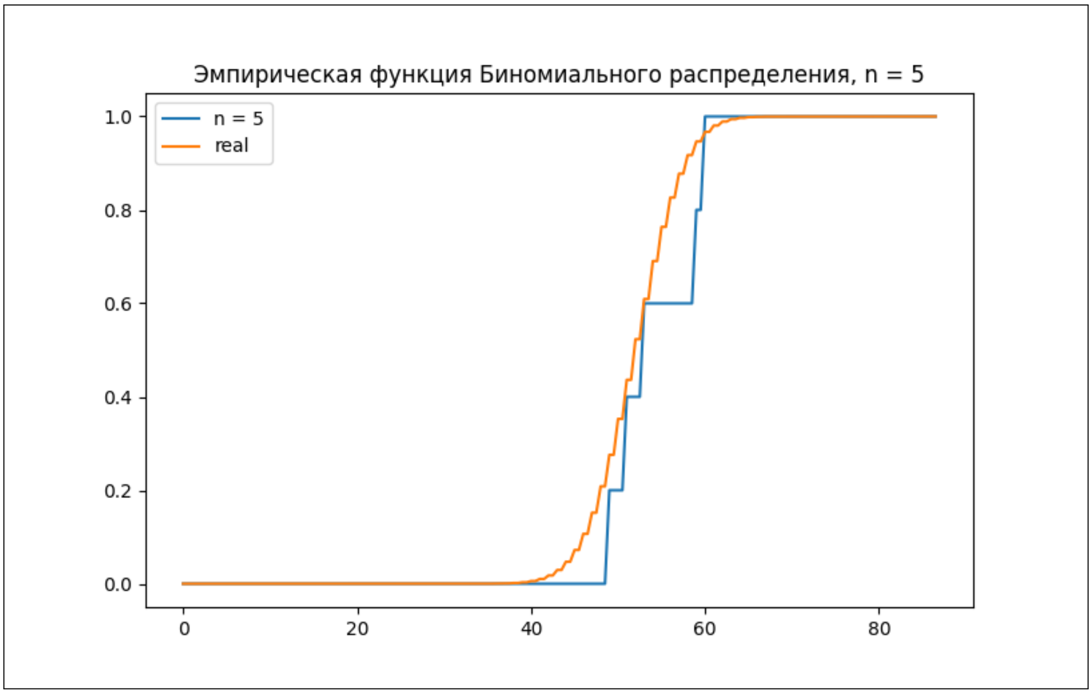
```
import matplotlib.pyplot as plt
import numpy
from scipy.stats import binom

def CDF(selection, t):
    sum = 0
    for element in selection:
        sum += int(element <= t)
    return sum / len(selection)

volumes = [5, 10, 100, 200, 400, 600, 800, 1000]
for volume in volumes:
    selection = []
    tmp_1 = open(f'{volume}.txt', 'r').read()
    tmp_2 = tmp_1.split('\n')
    tmp_3 = []
    for i in tmp_2:
        tmp_3.append(i.split(' '))
    for i2 in tmp_3:
        for j in range(len(i2)):
            selection.append(int(i2[j]))

    x = numpy.arange(0, 87, 0.5)
    y_true = [binom.cdf(t, 87, 0.6) for t in x]
```

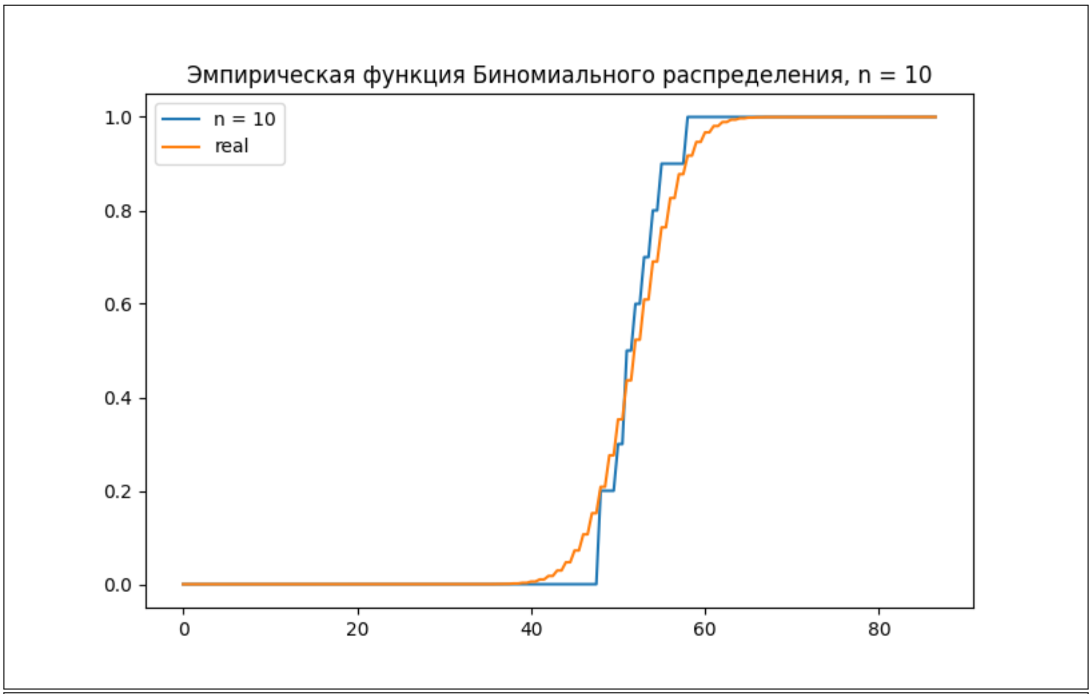
```
y = [CDF(selection , t) for t in x]
fig , ax = plt.subplots()
plt.title(f'                                            ')
ax.plot(x, y, label = f'n = {volume}')
ax.plot(x, y_true, label = 'real')
ax.legend()

fig.set_figheight(5)
fig.set_figwidth(8)
plt.show()
```

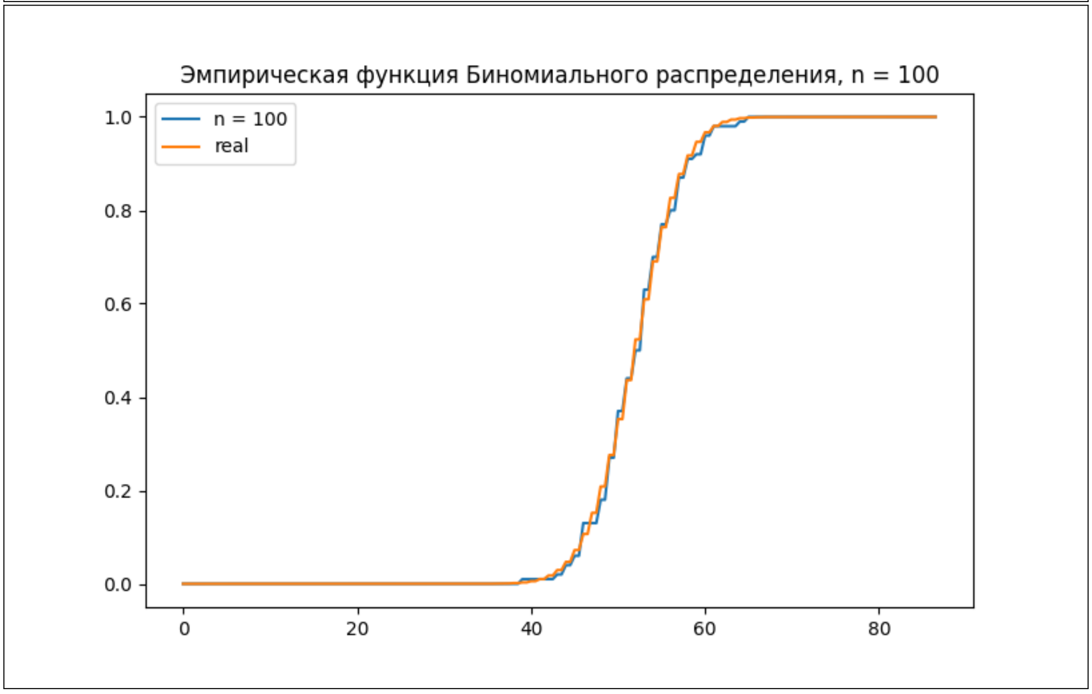
Ниже представлены графики Эмпирической Функции Распределения для каждой выборки с графиками функции распределения случайной величины: Также, проанилизировав приведённые графики, можно сделать вывод, что при увеличении объёма выборки график кумулятивной эмпирической функции распределения всё больше "стремится"к графику функции распределения случайной величины. Также это подтверждается ***Теоремой*** [***23*** - *Фомин Д. Б, Чухно А. Б., "Математическая статистика. Курс лекций для студентов кафедры компьютерной безопасности": **стр. 8***], гласящей о том, что:

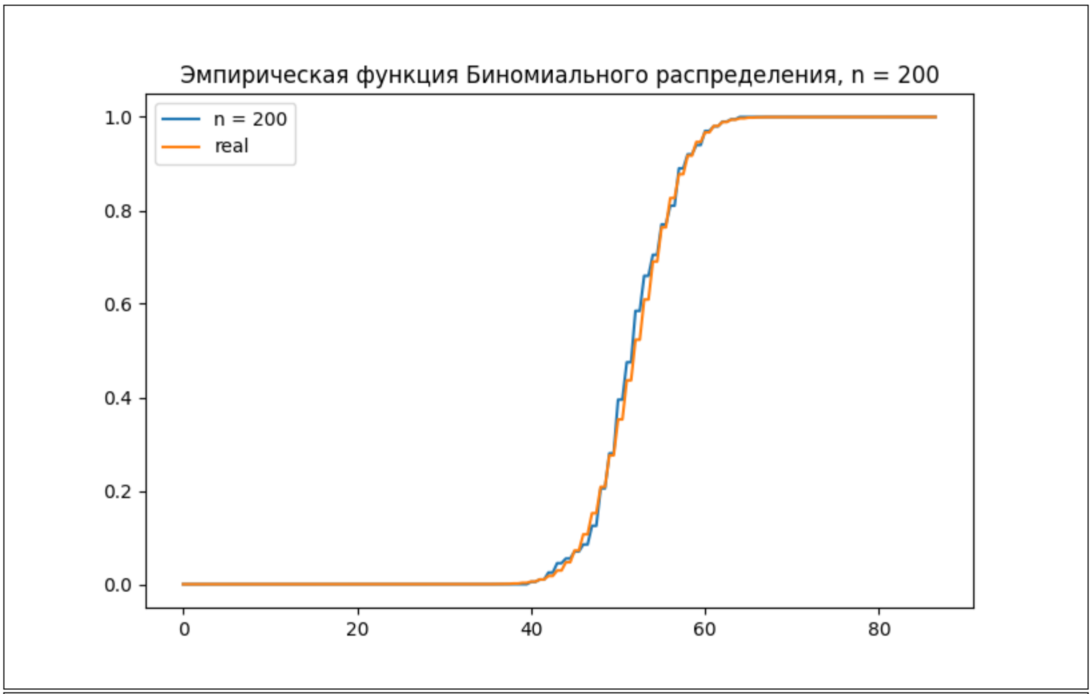
$$\text{Для } \forall x \in \mathbb{R} \text{ и для } \forall \epsilon > 0 \text{ при } n \longrightarrow \infty$$
$$P\Big(\Big|\widehat{F}_n(x) - F(x)\Big| < \epsilon\Big) \longrightarrow 1$$

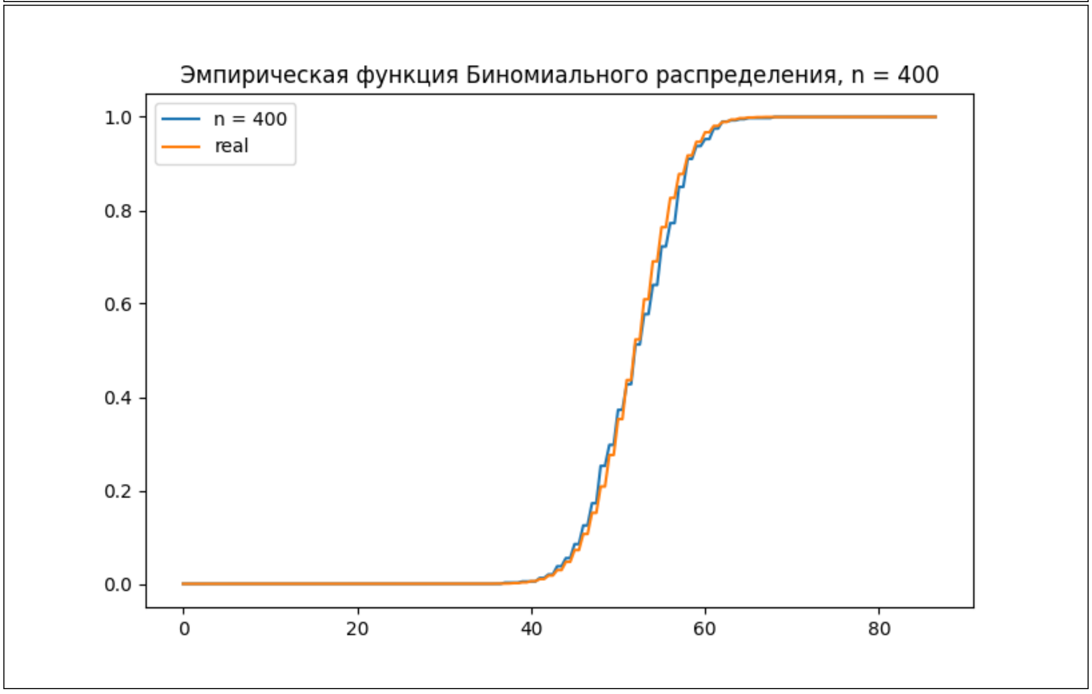
Другими словами: для произвольного фиксированного $y \in \mathbb{R}$ э.ф.р. $\widehat{F}_n(y)$ с увеличением объема выборки n стремится к значению функции распределения $F(y)$.

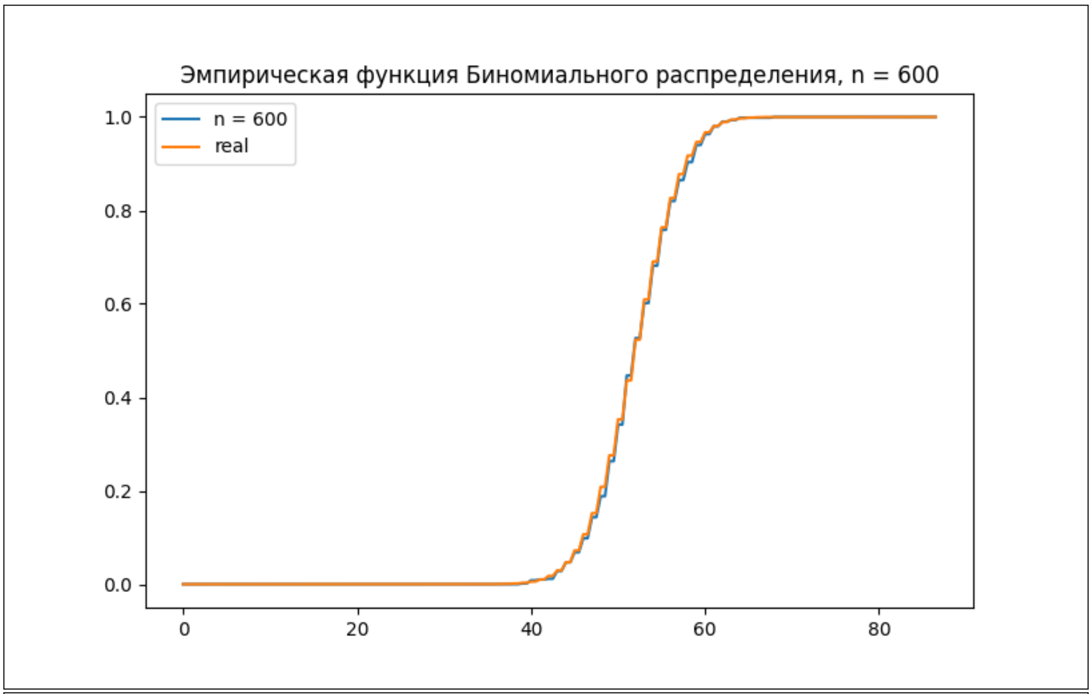


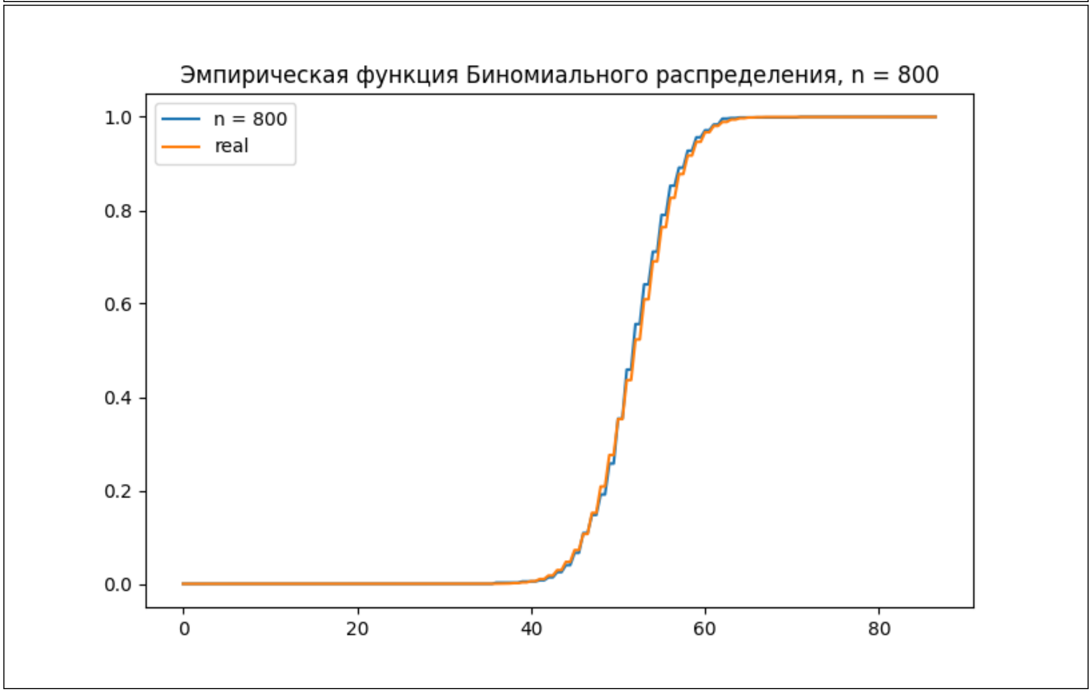

Эмпирическая функция Биномиального распределения, n = 10
n = 10
real
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80

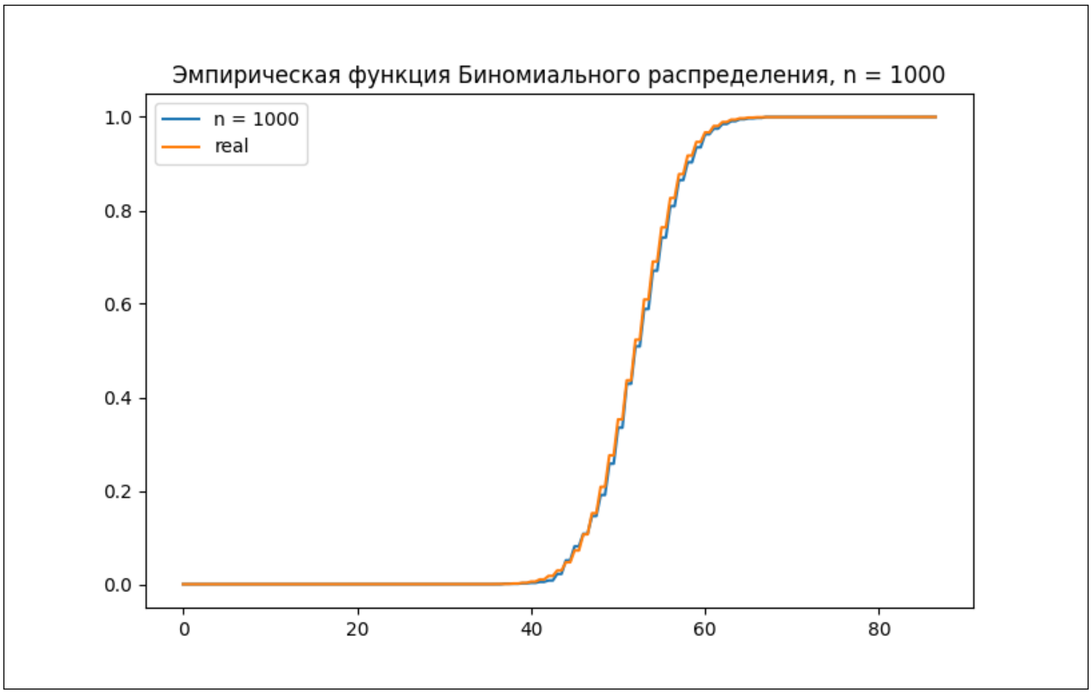

Эмпирическая функция Биномиального распределения, n = 100
n = 100
real
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80


Эмпирическая функция Биномиального распределения, n = 200
n = 200
real
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80


Эмпирическая функция Биномиального распределения, n = 400
n = 400
real
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80


Эмпирическая функция Биномиального распределения, n = 600
n = 600
real
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80


Эмпирическая функция Биномиального распределения, n = 800
n = 800
real
1.0
0.8
0.6
0.4
0.2
0.0
0
20
40
60
80



Для посчета $D_{n,m}$ необходимо описать функцию супремума, которая описана ниже:

```
def sup(selection1, selection2):
    x = np.arange(0, 87, 0.5)
    f1 = [eCDF(selection1, t) for t in x]
    f2 = [eCDF(selection2, t) for t in x]
    s = 0
    for i in range(len(x)):
        s = max(s, abs(f1[i] - f2[i]))
    return s
```

Описав функцию супремума, опишем и саму функцию подсчета $D_{n,m}$:

```
def D(selection1, selection2):
    m = len(selection1)
    n = len(selection2)
    return np.sqrt(m * n/(m + n)) * sup(selection1, selection2)
```

Произведём расчёты функции $D_{n,m}$ для каждой пары выборок с помощью скрипта на языке Python. Получим матрицу значений, которая, что примечательно, симметрична относительно главной диагонали.

	5	10	100	200	400	600	800	1000
5	0.0000	0.7303	0.6765	0.7068	0.6889	0.6755	0.7300	0.6758
10	0.7303	0.0000	0.3920	0.4012	0.5544	0.4495	0.4635	0.4972
100	0.6765	0.3920	0.0000	0.6940	0.6485	0.2932	0.5303	0.3909
200	0.7068	0.4012	0.6940	0.0000	0.9526	0.7144	0.5376	0.9812
400	0.6889	0.5544	0.6485	0.9526	0.0000	0.9941	1.3064	1.0395
600	0.6755	0.4495	0.2932	0.7144	0.9941	0.0000	0.7329	0.3421
800	0.7300	0.4635	0.5303	0.5376	1.3064	0.7329	0.0000	1.1015
1000	0.6758	0.4972	0.3909	0.9812	1.0395	0.3421	1.1015	0.0000

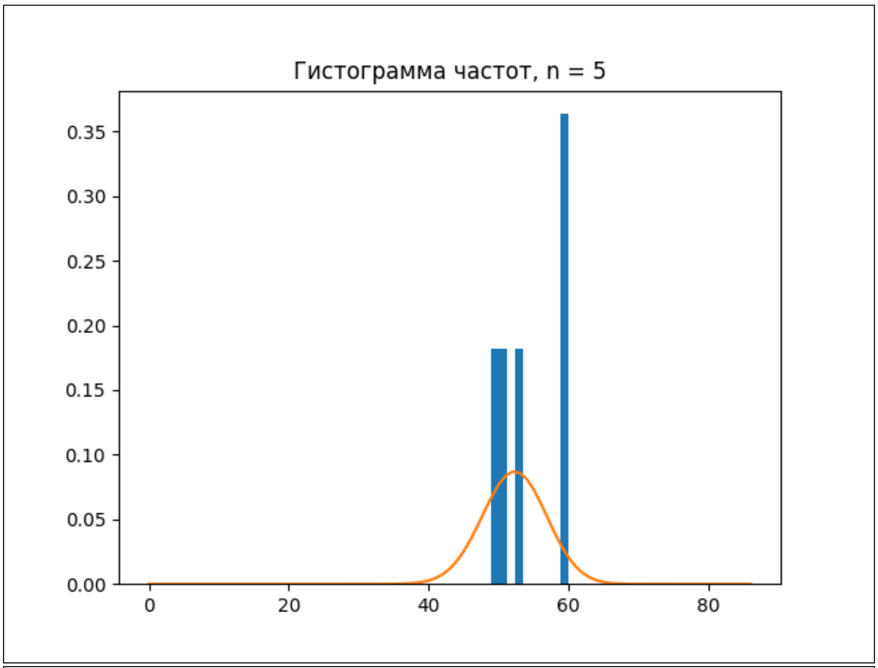
2.1.3. Построение гистограммы частот

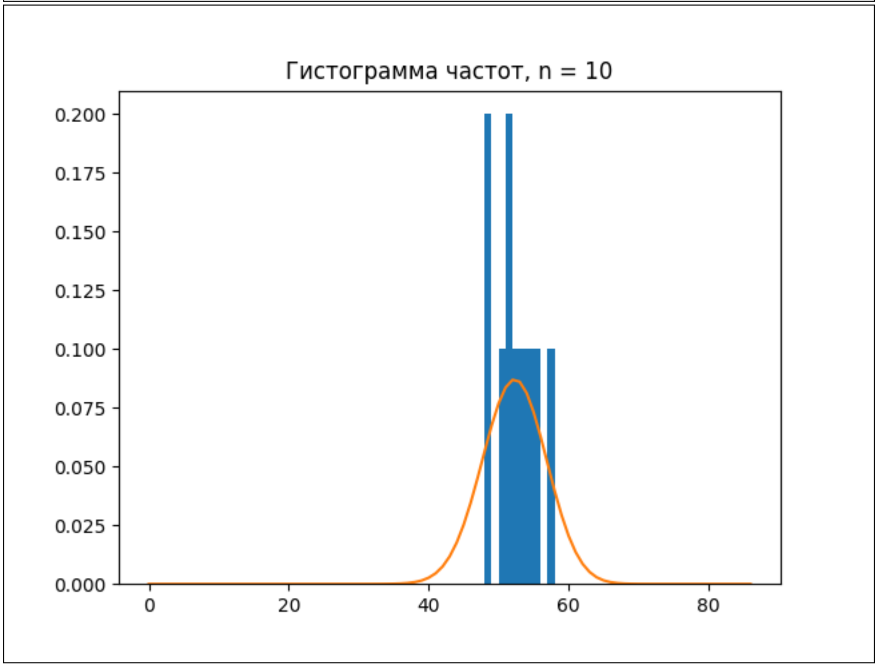
Ниже представлены гистограммы частот для каждой выборки с графиками функции вероятности:
Из приведенных ниже гистограмм частот можно сделать вывод, что с увеличением объёма выборки - её гистограмма частот "стремится"к функции вероятности. Это иллюстрирует следующую теорему:

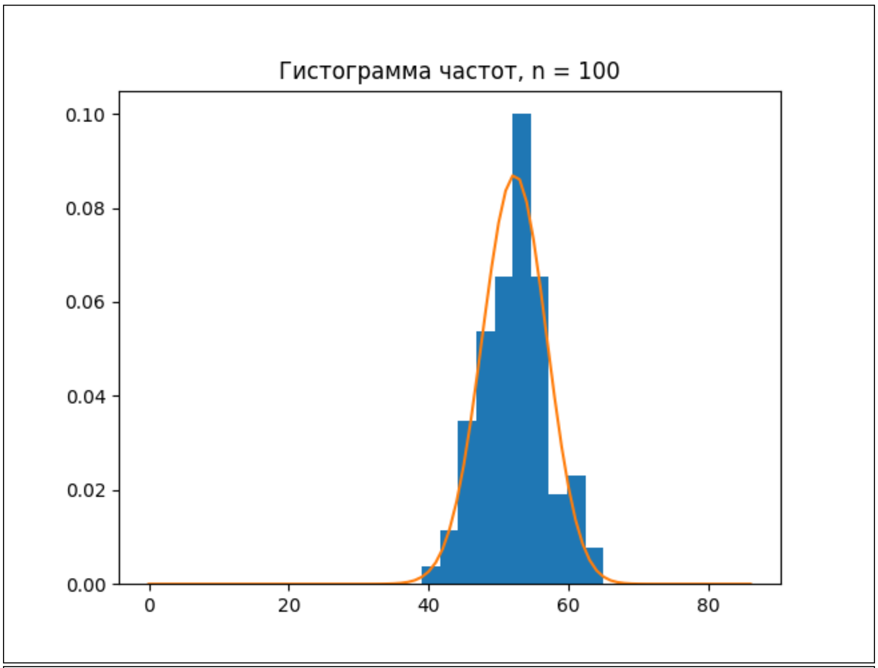
[***24*** *- Н. И. Чернова, "Лекции по математической статистике Нижегородский Государственный Университет:* ***стр. 12, стр. 20***] Пусть распределение F абсолютно непрерывно, f - его истинная плотность. Пусть, кроме того, число k интервалов группировки не зависит от n. Тогда справедлива Теорема:

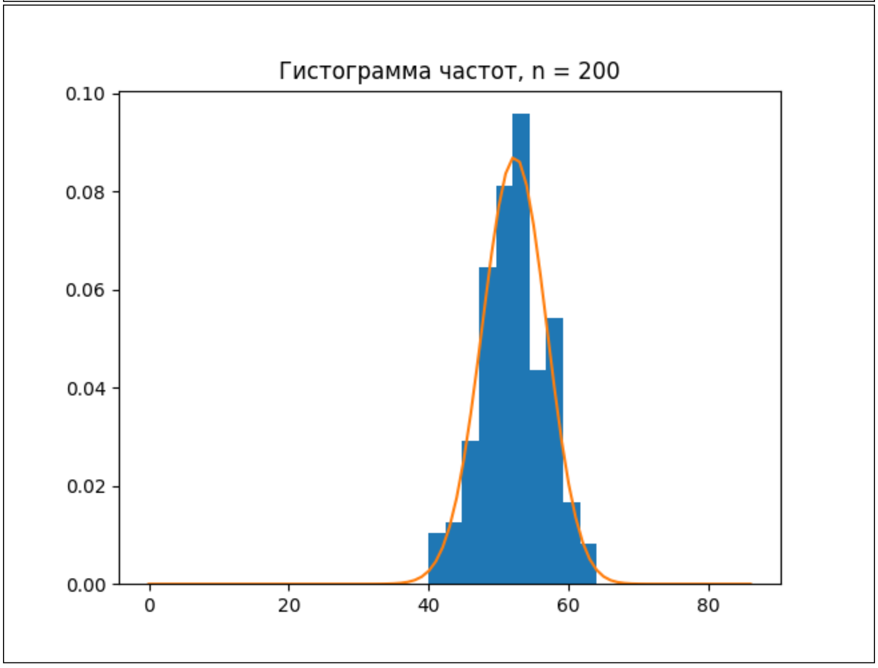
$$\text{При } n \longrightarrow \infty \ \forall j = 1, \cdots, k$$
$$l_j \cdot f_j = \frac{vi}{n} \longrightarrow^p P(X_1 \in A_j) = \int_{A_j} f(\chi) d\chi$$

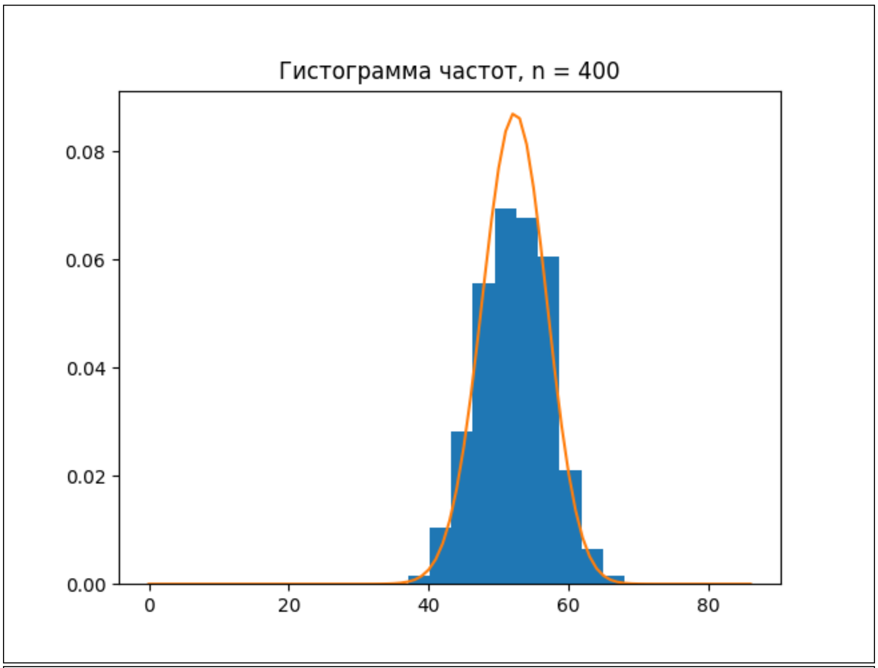
Предполагаемую область значений случайной величины ξ делят независимо от выборки на некоторое количество интервалов (не обязательно одинаковых). Пусть $A_1, \cdots, A_k$ - интервалы на прямой, называемые интервалами группировки. Обозначим для $j = 1, \cdots, k$ через v_j число элементов выборки, попавших в интервал A_j. l_j - длина интервала A_j
Данная теорема утверждает, что площадь столбца гистограммы, построенного над интервалом группировки, с ростом объема выборки сближается с площадью области под графиком плотности над этим же интервалом.

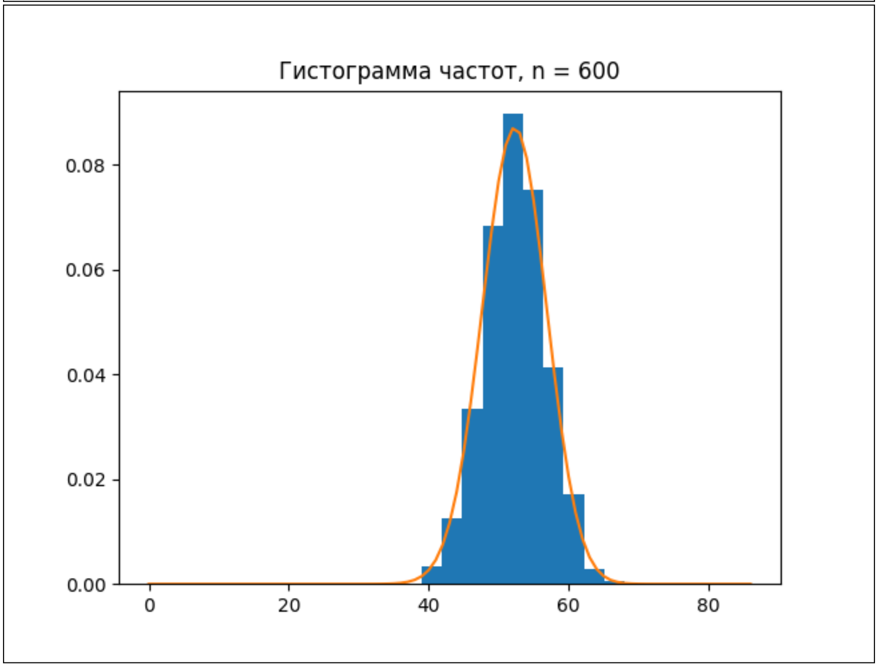

Гистограмма частот, n = 5
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80

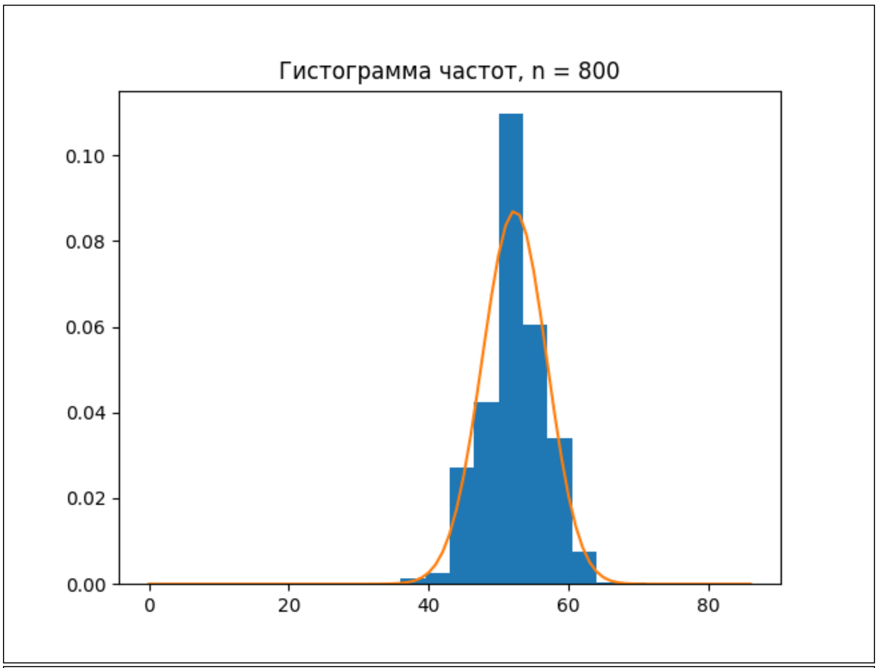

Гистограмма частот, n = 10
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
20
40
60
80

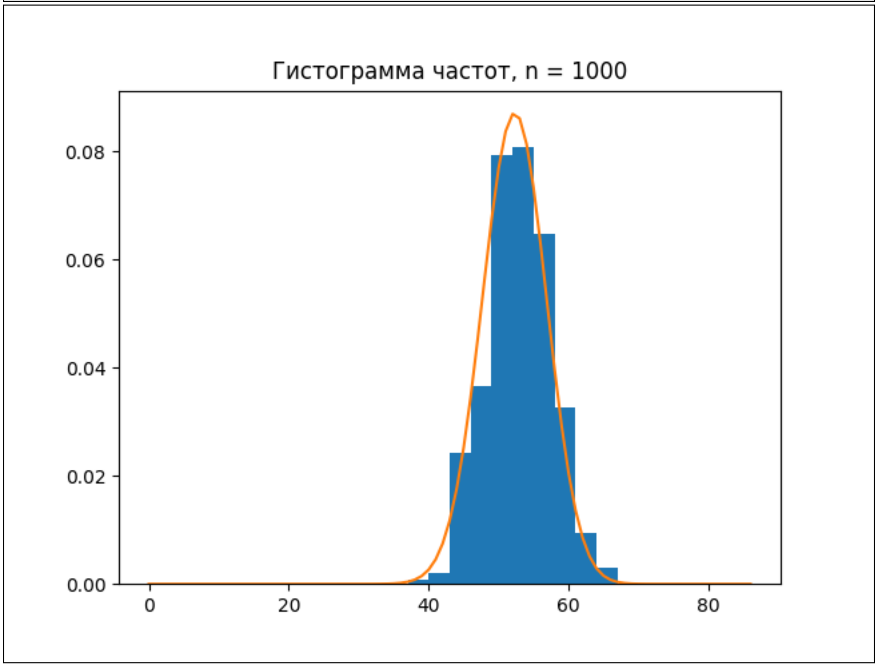

Гистограмма частот, n = 100
0.10
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80


Гистограмма частот, n = 200
0.10
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80


Гистограмма частот, n = 400
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80


Гистограмма частот, n = 600
0.08
0.06
0.04
0.02
0.00
0
20
40
60
80





Листинг кода:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import binom

volumes = [5, 10, 100, 200, 400, 600, 800, 1000]
for volume in volumes:
    selection = []
    tmp_1 = open(f'{volume}.txt', 'r').read()
    tmp_2 = tmp_1.split('\n')
    tmp_3 = []
    for i in tmp_2:
```

```
    tmp_3.append(i.split(' '))
for i2 in tmp_3:
    for j in range(len(i2)):
        selection.append(int(i2[j]))
x = np.arange(0, 87, 1)
y_true = [binom.pmf(t, 87, 0.6) for t in x]
plt.title(f'                 , n = {volume}'
plt.hist(selection, density=True)
plt.plot(x, y_true)
plt.show()
```

2.1.4. Вычисление выборочных моментов

Оценка математического ожидания является **несмещенной** и **состоятельной**
Оценка дисперсии является **смещенной** и **состоятельной**
Докажем эти факты:
Доказательство несмещённости оценки математического ожидания
По ***Определению 2.6*** [**25** - *Фомин Д. Б, Чухно А. Б., "Математическая статистика. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 12***] знаем, что, если смещение $b(\theta)$ равно нулю, то оценка является **несмещённой**, где

$$b(\theta) = M_\theta T(X) - \tau(\theta)$$

Таким образом, в случае выборочного среднего, имеем:

$$\tau(\theta) = MX_i, i = 1, \cdots, n$$
$$\overline{X} = T(X) = \frac{1}{n}\sum_{i=1}^{n} X_i, n \text{ - объём выборки.}$$

Найдем математическое ожидание выборочного среднего:
$MT(X) = \frac{1}{n}M\sum_{i=1}^{n} X_i = \frac{1}{n}\sum_{i=1}^{n} MX_i = \frac{1}{n}nMX_i = MX_i$
Из приведенного доказательства видно, что
$b(\theta) = M_\theta T(X) - \tau(\theta) = MX_i - MX_i = 0$
Следовательно, оценка является **несмещённой**.

Доказательство состоятельности оценки математического ожидания
Воспользуемся утверждением [**26** - *Фомин Д. Б, Чухно А. Б., "Математическая статистика. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 15***] о том, что для проверки состоятельности несмещенной оценки достаточно убедиться в том, что ее дисперсия стремится к 0 при n, стремящемуся к ∞.
Также воспользуемся ***Свойством 4*** [**27** - *Фомин Д. Б, Чухно А. Б., "Теория Вероятности. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 72***], которое гласит, что:

$$D(c \cdot \xi) = x^2 \cdot D\xi$$

Таким образом:
$DT(X) = D\Big(\frac{1}{n} \cdot \sum_{i=1}^{n} MX_i\Big) = \frac{1}{n^2}\Big(\sum_{i=1}^{n} DX_i\Big) = \frac{1}{n^2}nDX_i = \frac{1}{n}DX_i$
Тогда:

$\lim\limits_{n\longrightarrow\infty} DT(X) = \dfrac{1}{n} DX_i = 0$

т.к. $\dfrac{1}{n^2} \longrightarrow 0$ при $n \longrightarrow \infty$

Тогда, можем сделать вывод, что оценка выборочного среднего **несмещённая** и **состоятельная**.

Доказательство смещенности оценки дисперсии

Воспользуемся тем же утверждением, что и в доказательстве несмещенности оценки выборочного среднего. А также воспользуемся утверждениями с уже упомянтой ***стр. 12*** лекций курса Математической статистики. Тогда:

$$\text{Пусть } Y_i = X_i - a_1, a_1 = M\xi (1)$$
$$\text{Следовательно: } \overline{Y} = \overline{X} - M\overline{X} (2)$$

Тогда:

$MT(X) = \dfrac{1}{n} M \sum\limits_{i=1}^{n} (Y_i - \overline{Y})^2 =$ по определению: $\overline{Y} = \sum_{j=1}^{n} Y_j$, поэтому $=$

$= \dfrac{1}{n} M \sum\limits_{i=1}^{n} (Y_i^2 + \overline{Y}^2 - 2\Big(\dfrac{1}{n}\sum\limits_{j=1}^{n} Y_j\Big) Y_i) = \dfrac{1}{n} M \Big(\sum\limits_{i=1}^{n} Y_i^2 + \sum\limits_{i=1}^{n} \overline{Y}^2 - 2\Big(\dfrac{1}{n}\sum\limits_{i,j=1}^{n} Y_j Y_i\Big)\Big) =$

$= \dfrac{1}{n}\Big(\sum\limits_{i=1}^{n} MY_i^2 + \sum\limits_{i=1}^{n} M\overline{Y}^2 - \dfrac{2}{n}\sum\limits_{i,j=1}^{n} M(Y_j Y_i)\Big) =$

Установим следующие факты:

1. $MY_i =$ опираясь на утверждение $(1) = MX_i - M(MX_i) =$ Мат ожидание - константа. Мат ожидание константы равна этой же константе, следовательно $= MX_i - MX_i = 0$

2. $MY_i^2 =$ вычтем из Мат. Ожидания нулевую величину $=$
$= MY_i^2 - (MY_i)^2 = DY_i$

3. Если $i \neq j$, то верно следующее:
$M(Y_i Y_j) = MY_i MY_j$
Учитывая пункт **1** ($MY_i = 0$), получим:
$M(Y_i Y_j) = MY_i MY_j = 0$

4. $M\overline{Y}^2 = \dfrac{1}{n^2}\sum\limits_{i,j=1}^{n} M(Y_iY_j) =$ имеем сумму сумм $= \dfrac{1}{n^2}\Big(\sum\limits_{i\neq j} M(Y_iY_j) +$ $+ \sum\limits_{i=1}^{n} MY_i^2\Big) =$ опираясь на пункт **3**, значем, что $M(Y_iY_j) = 0$, тогда $= \frac{1}{n^2}\sum_{i=1}^{n} MY_i^2 =$ опираясь на пункт **2**, знаем, что $MY_i^2 = DY_i$, тогда $= \dfrac{1}{n^2}\sum\limits_{i=1}^{n} DY_i = \dfrac{1}{n^2}nDY_i = \dfrac{1}{n}DY_i$

$= \dfrac{1}{n}\Big(\Big(1 - \dfrac{2}{n}\Big)\sum\limits_{i=1}^{n} MY_i^2 + \sum\limits_{i=1}^{n} M\overline{Y}^2\Big) = \dfrac{1}{n}\Big((1 - \dfrac{2}{n}\Big)nDY_i + DY_i\Big) =$
$= \dfrac{1}{n}\Big(DY_i\Big(n + 1 - 2\Big)\Big) = \dfrac{n-1}{n}DY_i = \dfrac{n-1}{n}D(X_i - MX_i) =$ Опираясь на то, что $D(\xi + c) = D\xi$, получаем $= \dfrac{n-1}{n}DX_i$

Таким образом, $b(\theta) = DX_i - \dfrac{n-1}{n}DX_i \neq 0$

Следовательно, оценка дисперсии **смещённая**.

Доказательство состоятельности оценки дисперсии

По определению: $\overline{S}^2 = \dfrac{1}{n}\sum\limits_{i=1}^{n}(X_i+\overline{X})^2 = \dfrac{1}{n}\sum\limits_{i=1}^{n}(X_i^2+(\overline{X})^2-2X_i\overline{X}) = \dfrac{1}{n}\sum\limits_{i=1}^{n} X_i^2-$ $- \dfrac{2}{n}\overline{X}\sum\limits_{i=1}^{n} X_i + \dfrac{1}{n}(\overline{X})^2\sum\limits_{i=1}^{n} = \overline{X^2} - 2\overline{X}\cdot\overline{X} + (\overline{X})^2 = \overline{X^2} - (\overline{X})^2$

Тогда, опираясь на вышеописанные записи и Закон Больших Чисел в форме Хинчина [***28*** - *Источник, **стр. 77***], получим, что:
$\overline{X}^2 - (\overline{X})^2 \longrightarrow^{p}_{n\longrightarrow\infty} MX_i^2 - (MX_i)^2 = DX_i$
Таким образом, оценка дисперсии **состоятельна**.

Истинное выборочное среднее при параметрах $n = 87, \theta = 0.6$:
$n\cdot\theta = 87\cdot 0.6 = 52.2$
Истинное выборочная дисперсия при параметрах $n = 87, \theta = 0.6$:
$n\cdot\theta\cdot(1-\theta) = 87\cdot 0.6\cdot 0.4 = 20.88$

	Выборочное среднее	Выборочная дисперсия
5	54.400000	19.040000
10	52.000000	8.800000
100	52.330000	21.761100
200	52.010000	20.449900
400	52.307500	24.532944
600	52.326667	20.873289
800	52.046250	19.191611
1000	52.432000	21.505376

Вычислим погрешность выборочного среднего и выборочной дисперсии, зная истинные значения данных величин:

	Погрешность выборочного среднего	Погрешность выборочной дисперсии
5	4.21%	-8.81%
10	-0.38%	-57.85%
100	0.25%	4.22%
200	-0.36%	-2.06%
400	0.21%	17.49%
600	0.24%	-0.03%
800	-0.29%	-8.09%
1000	0.44%	3.0%

Листинг кода:

```
from numpy import average, var
import pandas as pd
import random


volumes = [5, 10, 100, 200, 400, 600, 800, 1000]
results = []
for volume in volumes:
    selection = []
    tmp_1 = open(f'{volume}.txt', 'r').read()
    tmp_2 = tmp_1.split('\n')
    tmp_3 = []
    for i in tmp_2:
        tmp_3.append(i.split(' '))
    for i2 in tmp_3:
        for j in range(len(i2)):
```

```
            selection.append(int(i2[j]))
    results.append(selection)

means = [average(element) for element in results]
variances = [var(element) for element in results]
dict1 = {'                         ␣                    ':means, '
table = pd.DataFrame(data=dict1)
table.index = volumes
print(table)

means_2 = [f'{round((((s␣-␣52.2)␣/␣52.2)␣*␣100),␣2)}%' for s in me
variances_2 = [f'{round((((s␣-␣20.88)␣/␣20.88)␣*␣100),␣2)}%' for s
dict2 = {'                          ␣                              ␣
table2 = pd.DataFrame(data=dict2)
table2.index = volumes
print(table2)
```

2. Распределение Максвелла

2.2.1. Генерация выборок

В данном разделе производится генерация выборок объёмов 5, 10, 100, 200, 400, 600, 800, 1000 распределения Максвелла с параметром $\theta = 3$

5

5.978659	3.882685	5.332642	3.792912	2.248517

Выборка из распределения Максвелла, n = 5

10

2.437915	2.313844	4.983702	5.628353	3.314616	4.39135	2.641108	4.972171	4.316611	5.1504

Выборка из распределения Максвелла, n = 10

100

```
3.743929 5.985225 2.926826 4.453783 4.341105 6.526389 3.552428 6.954771 4.673033 6.471674
5.876565 4.883488 8.812439 5.243528 1.177563 3.490297 6.25796 10.695482 3.901811 2.918588
3.007704 4.47443 6.119129 0.743467 6.057575 5.675307 5.204846 3.294535 2.528951 2.914533
9.724539 6.30171 7.924079 7.078067 3.142683 5.149782 6.753857 1.014912 2.97325 4.636576
7.116546 5.736846 6.252237 4.072266 3.383452 5.172211 5.030629 6.659726 3.868418 7.084601
1.961672 9.916628 3.285024 5.491316 0.813439 6.125489 5.172604 2.65214 4.490726 6.788681
2.794284 8.230175 3.811409 2.82454 2.095384 6.480868 3.400357 4.950035 2.81351 4.01505
3.588684 6.453623 5.42178 4.25277 10.263959 8.401268 5.6353 5.996109 4.923808 6.655679
1.973557 5.737838 2.072984 2.057768 1.804127 6.361308 6.304689 3.972015 2.9857 4.513968
4.39285 5.750001 7.662104 9.604398 3.311574 6.729593 4.731556 2.790493 3.312852 3.568546
```

Выборка из распределения Максвелла, n = 100

200

```
4.163464 8.281889 4.346958 2.915387 3.621093 2.488622 5.432468 3.938306 2.496168 3.13166
5.022749 6.791328 2.250487 2.243167 6.207998 3.62662 5.868993 4.824561 8.089079 2.897955
7.934098 3.132094 3.92822 4.620692 10.293878 2.825236 3.766273 5.934309 5.022085 7.069286
5.477013 2.611119 5.123878 7.941912 3.61394 6.02347 3.71796 10.112785 2.252721 4.942372
3.938082 1.780738 4.541448 5.507187 7.418386 3.124036 3.85891 4.924271 5.476499 6.651391
3.625395 5.191589 4.157694 6.146099 2.639635 5.493403 3.68225 2.638331 3.984981 2.721442
2.341414 3.320608 4.93472 6.661868 7.127528 4.899688 6.297488 3.097076 3.238655 3.499899
4.803326 3.848491 7.305537 5.110893 3.134719 4.958308 4.690773 5.083176 5.174697 13.63153
4.955428 9.294255 4.740942 2.592964 9.97537 2.525156 2.113076 2.209332 6.20228 1.95286
2.466321 7.388283 5.475265 10.058142 3.072409 9.442314 3.545937 5.383646 4.481288 2.182793
5.212758 4.635863 1.306065 4.726436 6.695163 2.212387 3.627135 5.72037 6.225224 4.269051
4.594351 2.234496 6.740541 3.29108 9.246985 7.596288 6.928925 3.869056 3.43034 8.771654
6.20725 5.194098 4.606409 2.424896 7.141816 4.675675 3.95177 5.567589 3.370954 2.863342
5.1529 6.602498 4.350133 3.76478 4.191208 1.583009 5.459648 5.099775 3.130594 5.387404
3.097768 6.410503 5.793387 3.796698 2.063538 6.368327 3.038426 6.501752 3.739865 3.833632
5.200871 2.840246 2.335008 9.043127 7.382922 2.984322 1.486085 3.374738 7.026923 0.930176
5.41595 3.445524 9.844983 2.239749 9.040646 4.814995 4.284045 3.304887 4.885444 4.012325
5.050713 5.891153 8.822521 2.858886 3.470661 3.032715 5.598996 5.218481 8.817538 3.815099
4.375093 2.761266 4.851612 1.80923 4.541322 2.645631 4.487259 3.529743 4.337325 4.535549
4.977758 7.638169 7.67972 6.38841 2.048959 3.9431 4.792588 0.890109 5.894473 4.728456
```

Выборка из распределения Максвелла, n = 200

400

```
3.776917 6.958315 3.772173 8.359763 5.481413 2.089668 7.641308 7.065061 2.8645 6.460801
4.85396 8.185096 3.443301 4.340883 2.354132 3.821524 4.091768 6.1969 6.75437 5.649847
3.28423 5.467876 6.848927 6.226043 3.701626 1.115328 8.148034 8.107147 3.622003 5.497207
4.874186 3.11056 2.108266 5.465625 6.426856 3.07137 1.850071 7.324902 1.885201 4.72969
7.150973 7.722592 2.974091 3.554373 7.197303 4.148108 4.842163 4.567644 4.662612 4.949355
5.110947 3.59525 4.813358 2.221649 7.323724 6.143892 8.879189 9.87512 1.964412 5.100178
2.292385 2.157791 3.067337 3.580646 4.632528 4.521387 3.463692 4.157183 6.770431 6.511405
4.733868 2.60968 5.549726 7.376608 4.556398 7.026266 5.074277 7.959496 1.256035 6.986711
5.069405 2.477998 5.673337 4.24136 7.434022 6.098614 7.694762 5.401015 5.395085 6.447
4.535953 6.606571 4.135421 5.591579 4.509269 4.735184 2.163232 5.813894 3.897215 3.584435
1.847571 7.153872 4.948098 4.190793 4.580417 2.429442 6.181422 3.589429 5.208721 2.424739
6.040924 2.762293 4.522208 4.168891 10.514337 2.64458 2.918733 3.900396 3.187448 2.278122
4.845926 6.288816 6.956376 5.179668 8.068036 5.037852 5.057931 4.595967 6.826913 6.170748
10.179538 6.584159 7.063099 5.671375 3.603095 3.47526 7.790506 7.988453 6.546195 3.522159
4.574213 5.953324 6.351856 6.930458 7.889354 0.586758 6.713262 5.478398 4.927555 7.548892
3.3581 1.870787 5.400036 2.868282 4.957364 2.647077 3.774696 3.340545 5.814119 3.619053
5.237794 7.560569 5.049634 3.92395 4.416331 5.331689 7.904682 5.023288 5.792399 2.833783
1.173727 6.37802 3.246332 4.035683 3.194984 3.234478 3.541144 8.625274 2.841037 3.395584
4.92973 3.948759 5.617251 6.667465 2.609655 7.617175 2.775822 4.495214 4.224207 3.958122
10.228298 5.338508 6.000432 3.183142 4.358983 8.759116 5.637558 2.343852 4.264341 7.409137
4.457913 3.821588 5.829812 3.139303 3.286095 5.89697 4.012956 2.339075 4.142861 1.800292
5.477308 6.276241 5.38073 6.628801 4.931564 5.344622 4.649822 4.476595 3.313955 4.028891
4.479243 8.63962 3.74368 4.189988 6.464684 5.644882 3.357324 5.572735 3.701676 4.975991
7.265035 1.916665 4.159749 2.523905 4.902916 3.143754 6.871177 3.574917 5.481052 2.069482
4.327596 3.859185 7.074433 7.628188 5.062173 4.692123 0.550708 7.444999 4.01996 3.425001
4.415816 4.087895 4.214403 1.392396 4.510447 4.228354 3.655946 4.277358 5.183163 3.689028
7.824183 3.429858 6.762408 6.319635 7.439142 6.427209 3.441995 3.464079 4.630121 1.617112
3.470511 6.091774 1.711554 2.401684 8.522305 5.713525 4.08504 2.263522 2.030143 3.197291
4.899993 7.015653 4.530009 1.357197 3.886404 7.776212 2.69654 9.287689 5.831228 7.981977
5.496911 4.746666 1.305837 3.405889 3.044436 1.754998 4.979257 4.775211 1.066318 4.281096
1.80072 4.319798 4.33421 4.984958 2.916286 2.1886 2.568419 3.132739 5.331083 5.731949
2.137085 4.379762 3.232485 1.937291 4.19484 1.613766 4.454657 8.383662 4.399131 4.084033
3.290918 4.350329 4.546325 4.443608 2.847722 5.649771 2.798528 7.072764 4.285136 4.75029
6.780231 7.653213 9.933275 5.893088 3.417068 5.614469 5.094857 6.952214 5.224389 7.107797
1.964986 1.909114 7.994521 7.304482 3.512199 8.663458 3.907023 7.705314 5.389866 6.600201
3.286788 6.059271 2.928786 5.880037 2.414404 5.878435 4.856067 3.547098 4.217814 4.416145
3.40402 7.092535 5.333765 2.649198 6.673943 2.032467 6.880346 3.713133 4.600913 5.071385
2.426118 5.986722 3.071188 9.090494 7.603013 7.631258 6.08866 3.349545 4.559601 1.458348
9.195483 1.061365 7.3214 3.166634 2.532388 6.974116 4.023984 2.86516 2.909187 4.220504
3.002751 7.208897 7.513001 4.119757 6.752491 2.411805 5.975628 5.017125 4.459699 2.533649
```

Выборка из распределения Максвелла, n = 400

600

```
5.061416 4.882332 3.912207 4.181802 3.301085 6.319638 4.347718 4.274292 7.471754 2.4036
5.461324 2.789824 2.217251 4.533203 7.422335 7.294195 3.437221 6.06264 5.189411 7.181458
5.150599 5.438119 3.289236 5.423181 5.236302 3.19141 5.349263 3.989748 2.830032 8.636496
4.868978 8.816998 7.132446 3.950771 4.141702 1.007006 2.478733 4.748524 2.724897 3.96016
2.811798 3.005714 5.283518 1.594666 5.144758 3.926501 6.158947 6.144069 6.450032 3.344905
4.707143 1.900209 3.668272 3.153291 5.739294 5.028115 9.341649 4.464943 4.440304 2.5212
6.24512 3.616279 6.442358 3.87293 2.305699 4.575298 4.989394 10.317191 3.87555 7.650634
6.438352 2.398165 5.442296 2.369172 4.87262 2.367833 4.423824 5.16149 2.944764 1.803055
4.13756 3.577352 3.879682 5.993284 4.993318 2.460173 3.309851 4.883632 10.081313 3.417654
5.836967 4.213467 6.091242 5.818177 3.630374 2.803353 3.47463 5.206942 8.077985 9.373814
4.49261 2.349592 3.819781 5.362941 2.585361 4.353973 1.791723 5.969646 3.415229 6.405717
3.657213 5.849514 3.939566 4.570641 0.748869 7.69833 3.108813 5.583124 5.791647 5.879173
5.505198 3.521625 5.37728 8.106184 4.790151 5.569303 6.816087 3.11577 4.35026 3.920286
5.7473 3.10666 4.586752 2.837911 6.545446 3.360672 7.278631 5.482849 5.255767 3.98858
7.114459 3.259477 2.451399 7.02649 2.356987 4.94552 2.300712 8.628625 5.02199 4.918121
6.761225 4.415894 1.766597 3.049868 2.646354 5.857904 5.415275 3.606836 7.204934 2.930211
3.68333 7.560606 3.727815 4.041816 4.036268 4.50236 3.066306 4.397979 2.123369 3.6508
6.154693 2.986366 2.883484 4.217684 4.53297 6.14466 3.699963 3.422814 5.240527 3.886923
3.280748 7.41801 3.415226 2.283739 1.988332 3.359815 3.451087 3.654288 2.813468 6.479301
4.793653 7.747145 3.822243 2.7818 2.393895 6.623358 5.570403 2.144239 3.636316 5.656981
2.558499 4.105308 5.325361 5.668274 7.658885 2.693133 6.843641 7.921616 4.610832 6.996999
5.90041 3.124521 5.126303 5.893294 3.998687 1.824214 4.918997 2.726485 5.995574 7.398376
3.427968 4.145796 4.181976 2.537424 4.701599 6.92239 3.073489 6.640258 3.60121 3.345104
3.996408 5.402394 2.810712 4.976854 5.840185 7.824732 5.141388 6.781703 4.859883 5.529501
7.632582 5.366487 4.989315 4.517233 4.147943 3.558954 6.210423 8.206518 4.964048 7.233523
1.88697 4.32315 3.468355 6.374386 3.678871 6.302066 6.952685 0.72133 6.274352 4.822251
5.773528 5.197064 3.830331 8.412084 8.211417 4.173574 5.023484 1.668932 3.025083 1.925318
1.946488 5.96605 5.815076 5.41467 5.964869 4.504554 7.507203 7.218372 4.948556 4.602901
2.865543 5.45614 3.206578 5.99543 10.578777 4.117617 3.627889 4.435276 1.958924 5.429656
7.287589 3.6454 3.89197 6.706588 3.209973 4.502658 4.413237 2.618649 3.78888 2.257637
4.94776 4.168455 4.498686 6.295945 4.228628 5.209727 5.89263 3.102987 4.812078 4.302369
2.015071 4.921818 5.292456 0.279167 1.398984 3.787828 2.828896 5.606239 6.546879 4.722382
5.533615 6.150162 3.285801 2.835635 4.098189 5.204852 4.680998 3.38924 6.605929 1.815117
3.126459 5.532529 2.890921 2.722636 6.222446 3.313016 1.867724 2.650951 3.523293 4.051961
1.113226 6.73408 5.79159 3.519628 3.085916 8.011318 4.483 3.840466 1.449971 6.225193
8.762854 5.633691 1.492916 4.832575 5.243113 3.902156 6.456079 5.186275 3.438716 5.266621
6.473639 4.574604 7.246956 5.665125 6.062608 7.222708 5.234118 3.156636 7.394885 2.124448
3.930468 7.048137 5.389841 5.413147 5.665329 4.806846 4.033763 5.256146 6.912381 3.902277
8.929271 7.156721 5.833123 5.876008 6.000651 5.304029 5.256796 3.619226 6.535437 5.736013
```

```
7.176891 6.18691 4.992407 2.247415 3.08024 4.758851 3.336557 1.667183 4.468542 1.815571
5.017935 3.015861 7.510548 4.324292 2.463092 5.06945 4.640386 3.566869 4.585277 6.334102
6.327455 8.209358 5.559412 3.740514 3.645189 7.811718 4.167114 3.715088 3.926512 6.174275
5.591913 4.334397 4.905598 4.30793 3.843074 2.859362 2.709063 4.945361 7.447728 4.658335
4.312745 3.481477 3.010259 7.757464 5.26275 3.676211 4.919421 3.941788 3.354463 3.931874
4.17715 3.454763 2.830399 1.545217 2.270638 5.340349 7.842598 5.568035 4.098024 9.463085
4.175002 2.118377 3.332668 3.189403 1.768011 6.826156 4.302221 3.745546 1.523979 3.580423
10.419625 3.066189 4.829626 6.503275 7.820828 3.746122 4.69685 7.361343 4.738473 8.101198
6.651538 3.733125 4.630453 10.130821 2.875091 6.400557 4.453064 8.318552 6.082165 6.105496
2.532748 3.223148 6.278481 1.120307 2.469205 3.905496 3.873247 6.248985 2.593933 3.342106
3.216449 3.470084 1.72454 8.31918 4.798583 8.110346 4.308294 3.835996 7.692004 3.313351
3.814296 6.314529 6.187164 5.245177 3.93131 6.225619 3.806478 8.689573 6.55144 2.039802
1.280372 2.163406 7.49654 3.563709 4.400472 5.757024 3.863079 3.87251 4.248678 5.249299
4.557954 3.793062 7.607625 4.26234 6.493167 7.843516 2.531672 5.658428 4.331216 5.403498
2.32469 5.873036 4.955146 10.280648 5.461348 1.741459 4.979855 3.30714 3.429942 3.642995
3.340144 2.018182 3.518081 3.884849 7.624648 4.72792 3.71174 6.96879 5.979986 3.481308
6.334274 5.009114 4.252564 4.59556 4.28277 3.773017 5.529999 3.102911 7.16988 4.748758
5.286643 9.361536 2.543247 3.89872 4.783506 4.828876 5.798164 4.613802 5.376776 2.773596
3.915419 5.839008 5.969814 4.780935 7.146395 5.419471 5.254007 10.460688 3.264649 5.022777
5.118208 4.369596 3.490503 3.056565 4.111746 7.314587 6.98774 1.980807 8.895519 9.795395
4.739094 3.395745 5.72168 4.63727 5.4218 1.117444 9.546361 3.499012 3.155976 2.021423
```

Выборка из распределения Максвелла, n = 600

800

```
8.79586 6.662023 4.220939 5.013085 3.985878 2.234072 4.484574 8.004695 8.793991 5.003493
3.625391 4.34854 2.762274 3.602267 4.026232 5.294448 4.447796 3.735027 2.823943 6.63729
1.805488 7.707142 3.794824 4.943531 4.372124 5.284159 2.773297 5.02206 5.015371 5.797837
6.096525 3.446629 4.211384 4.581355 6.872422 4.686202 5.57527 4.114533 1.771668 2.334193
3.679262 4.342544 1.916843 5.832384 8.523795 1.950464 1.452541 1.696632 4.618505 2.895523
8.401933 4.292967 2.727127 4.972628 3.430145 2.676287 3.269431 5.014027 4.023453 5.832528
6.11332 6.05789 2.50585 2.647524 3.850686 4.640208 3.963826 2.686111 6.912903 3.369617
6.101465 6.428848 6.765644 5.967813 5.178672 7.061492 3.222531 3.337346 2.773454 6.064492
5.790786 3.312923 5.677023 5.267724 3.643873 2.865705 4.497238 6.053964 8.938678 2.513566
4.910568 5.586425 4.085058 6.379538 6.389852 6.118674 12.226566 2.947681 8.432385 5.731924
8.258966 5.669387 4.851825 0.992904 7.739486 3.883642 5.343534 6.032549 6.771801 4.92362
3.555944 3.628207 6.697406 8.061773 5.791744 3.127673 3.724386 4.295026 7.664162 5.293704
1.006806 5.797511 2.179695 4.710904 3.924924 7.208902 3.027245 4.927069 8.314516 6.29774
7.048303 3.86483 4.875864 7.650727 7.393093 6.19814 2.603556 9.727301 5.043592 5.587762
4.953306 3.907367 3.783708 3.491941 7.545066 5.28998 2.810871 5.971717 2.764893 6.424847
5.578239 5.525046 5.434679 7.824588 4.405162 7.102114 5.995527 6.326882 5.753418 10.155512
2.774452 3.709173 7.238168 5.714807 3.216247 4.726272 2.202969 4.188756 1.238204 4.282607
8.161264 2.789567 4.640637 3.742643 7.400521 5.872102 9.236839 3.595292 5.717502 6.571262
5.01924 4.637841 2.990743 3.227801 5.981333 0.934845 3.233535 9.996956 6.57773 6.171273
5.191913 9.843999 3.13424 2.73142 3.598147 1.497585 8.677321 2.628872 9.583622 11.352554
6.446955 4.45339 5.353904 5.825414 6.504492 2.54368 7.133605 5.14174 4.211686 4.460159
6.355743 4.244396 3.582002 6.988755 2.310438 4.902877 2.667767 1.493012 3.812286 4.543773
2.667127 8.225592 3.864209 4.645441 1.10036 3.276748 2.397163 4.603349 6.3205 6.56598
4.094367 4.92388 3.11384 8.487794 2.136667 4.449644 2.582107 4.909049 3.140717 9.705318
2.018188 5.809258 2.638004 2.918582 4.730359 4.943555 6.171906 5.480388 4.167014 3.901142
4.199155 6.05733 3.592326 7.4075 1.651604 4.676765 5.340161 5.55982 7.869256 4.280367
8.263027 2.428126 4.582362 8.14737 3.981915 2.841861 5.158195 7.916808 4.379079 3.254358
7.093504 4.28904 5.025974 4.132974 2.934711 5.337525 8.859026 3.597628 6.555653 7.024999
4.37445 1.605581 2.095226 5.886283 2.523723 4.213376 2.269591 6.477978 8.076632 1.926741
2.480061 5.574602 3.957629 1.328142 5.156558 4.835381 1.49794 4.066672 8.166101 7.576024
3.900795 7.907398 6.599944 4.101718 3.543435 4.433136 4.47256 9.694111 5.17133 6.276255
3.99717 1.51127 4.224738 3.772451 6.964439 8.617374 4.811535 6.727225 5.679844 8.6381
5.581367 2.801665 7.017632 5.440551 3.090834 7.79973 8.196316 0.910908 2.656761 8.062402
4.678525 3.641742 1.616157 4.28422 3.164443 3.640265 2.003677 8.677652 5.334082 7.450524
7.516873 3.456795 7.342774 4.98054 1.529578 3.694918 4.440409 3.687724 7.400726 5.083169
4.193945 2.151017 6.586089 2.524477 5.069442 2.453599 3.422987 2.192205 6.394798 4.214876
3.821333 5.861345 9.483918 3.849247 1.713737 5.539775 1.891365 3.220337 6.315553 2.345703
1.013882 2.451163 4.26082 2.526393 5.182359 4.161688 2.340583 7.306318 3.859536 2.837092
8.732181 8.281231 3.92164 6.096516 5.763418 5.391219 7.169491 8.139633 7.18399 7.228293
6.27701 5.947827 2.942411 5.119216 4.441732 3.0232 8.462862 3.040369 5.113275 6.384291
2.113129 4.86366 7.584299 9.118146 6.049033 4.211586 3.912169 5.122673 4.153139 7.480872
5.946875 2.253702 5.999867 4.044428 3.973488 1.060981 1.458528 3.194546 4.203685 8.523833
4.12745 5.718501 3.306877 5.884491 5.027122 3.93308 3.726915 5.849012 5.800216 3.128487
2.078646 5.992683 8.588268 2.670049 6.612856 4.810049 4.851878 4.773758 4.377538 1.833163
3.978269 3.936937 3.304833 0.586562 7.488085 3.407015 5.65726 5.700215 5.907202 8.376526
4.503525 5.30232 5.427778 5.732256 5.659488 6.174589 3.331066 2.415049 3.507092 7.913741
6.065485 6.137601 2.516574 4.659108 4.687957 3.730985 1.850539 6.507878 4.42876 5.619123
2.252762 5.359517 5.390731 8.171958 8.078497 3.987851 4.744931 1.264775 7.725816 7.985462
4.551082 5.175352 5.636996 1.433373 4.697521 3.528694 6.540138 8.096179 3.698216 5.070293
```

1.796103	4.588842	4.195297	6.00425	7.374321	5.84383	6.197652	3.618643	3.097477	3.203863
6.76676	6.128105	4.062626	4.980754	2.819874	2.280398	6.11945	5.674738	2.712506	4.340468
5.264099	1.757279	5.984033	4.280313	3.618782	6.032669	5.369041	3.382666	6.667651	1.37003
5.671741	5.382013	7.847173	1.889026	6.990459	2.42141	3.924761	5.665246	5.905965	7.236234
3.716709	2.809569	4.861652	7.79346	4.446165	5.844927	4.787579	8.818245	6.005963	9.178279
4.924528	3.679767	5.994775	1.694971	6.513029	3.984985	9.821942	1.030291	6.026923	8.301225
2.800267	2.752483	8.034616	4.796094	5.525713	4.500977	2.876071	6.598168	5.374342	1.433048
3.253062	5.595658	2.025383	3.557751	4.280931	3.455423	3.72885	3.991707	1.823885	2.692941
5.296471	2.773387	3.415089	4.420254	5.733215	10.418485	3.483535	4.627641	2.917766	2.439837
3.026293	4.288768	1.639722	3.661501	4.065505	6.975154	4.611714	7.859684	2.255196	6.95602
3.176858	3.277602	1.163816	3.789518	5.432328	6.739851	6.360989	3.094278	5.71087	4.666662
3.34015	4.557016	3.505717	5.619111	2.960072	3.899169	4.119087	4.643236	2.622483	3.34337
3.599805	4.156263	9.314435	4.76015	6.852356	6.629933	4.406008	11.416069	5.543388	4.960519
9.441463	1.821192	7.160223	5.995763	2.991739	5.012894	10.230201	4.108959	2.636339	3.213185
5.950111	3.241142	4.209822	6.130777	6.121562	5.054229	4.041456	4.326833	5.164148	1.949277
2.424895	3.741937	3.970443	5.653697	7.283926	3.202356	4.365879	5.692018	4.996688	8.451901
8.065226	3.841664	7.637742	6.123365	4.853782	5.300476	5.833045	4.85558	5.057015	5.417167
10.090256	4.383192	7.578014	3.774445	3.927006	4.63526	4.890004	3.276767	2.366308	5.356246
6.730456	4.782298	9.703042	1.729895	4.848148	2.981035	1.190496	6.746741	3.89087	4.782856
7.616495	7.874905	7.197702	7.393556	3.019809	4.302979	6.845265	2.79147	3.387408	7.660456
4.679707	3.474217	4.451759	4.643444	1.760492	4.527687	6.18858	3.261948	3.014833	5.607326
2.920906	5.690385	4.055519	3.305327	5.234141	3.471044	2.765806	5.114367	4.395211	6.565402
4.304651	6.352627	4.14975	2.274429	4.459298	7.442351	5.6396	3.742101	1.818442	9.033613
2.836887	3.784172	2.133249	2.977579	5.413477	1.756644	2.864162	6.348278	1.734645	4.818984
1.169358	5.799325	5.029078	5.917638	3.142014	4.183774	4.283995	3.77249	5.143962	5.051265
4.04482	5.114879	4.380022	4.972956	8.016101	3.630274	6.121418	7.363847	3.882722	10.865902
4.482168	4.821826	3.083971	4.38426	2.305557	3.940333	5.842252	4.526799	6.541909	3.766276
5.817845	9.289835	6.395415	10.484127	5.827502	6.479491	3.930052	9.252163	2.909929	5.866258
2.51574	8.534111	3.053681	8.211085	4.129348	2.96578	6.696778	6.925277	4.56608	5.397154
7.500266	4.403511	6.875161	2.368737	6.558391	2.992182	3.394009	5.659839	8.092047	6.729075
1.470791	4.374633	5.729951	5.254413	8.273105	5.045516	3.775871	3.479422	3.607381	4.957352

Выборка из распределения Максвелла, n = 800

1000

```
3.499653 1.477128 4.568255 5.336489 2.166396 6.26331 3.8477 10.439971 5.510086 8.134369
6.170561 5.927339 5.915895 7.011581 3.719269 4.766332 3.517709 4.185292 1.57614 6.155039
2.485065 2.219213 4.848931 2.202568 7.261905 4.5473 5.765557 2.326084 4.315635 1.024354
8.267419 4.212315 2.726419 4.737086 5.107034 5.575744 4.569741 6.171957 5.476283 8.247953
6.138316 5.588944 4.601143 4.444233 2.657124 10.55656 1.60868 3.698007 5.6783 6.491624
2.234939 3.277785 2.767834 1.870142 2.435527 6.677671 2.772971 6.328256 3.236066 0.981819
5.014434 6.849111 3.256183 9.45016 5.871001 4.155851 4.991211 4.757728 9.19267 1.277986
2.769376 1.103002 6.522046 7.328961 2.701995 2.473139 7.49492 7.1583 7.072329 4.46371
6.847596 7.464532 6.533642 3.492632 5.321282 8.680665 2.644371 3.586542 9.554414 4.000043
3.691909 2.336473 5.255478 5.662209 4.171855 2.725883 4.979463 2.534656 1.738487 10.002428
4.037831 3.809702 6.179419 9.282866 6.577752 1.765598 2.620056 5.801435 4.79285 6.346087
7.828791 3.252785 3.372118 6.3658 3.844943 9.481955 3.020545 5.120622 4.240292 2.31406
2.280232 6.417779 3.594048 8.301554 2.975194 4.946947 3.649967 4.173429 9.290501 1.188502
8.991834 2.736681 6.99472 6.441588 1.377289 8.444215 2.887 2.515027 6.887885 3.262545
4.256073 4.367372 6.781959 4.110839 5.242083 1.945759 6.031028 1.5053 4.26897 5.013388
5.938033 6.817966 0.563224 5.289043 6.360485 9.198504 4.595242 4.389531 10.359264 5.35562
4.839739 7.536868 3.733968 3.862285 2.996322 4.975633 2.397226 4.268429 3.733497 5.043523
1.88884 4.840525 5.83676 4.092651 6.42122 2.809479 5.721711 3.186446 5.564182 2.930388
7.723964 5.277271 4.887644 3.642561 7.049554 6.079062 5.875207 2.993825 5.386347 4.172309
11.575464 2.490372 3.254549 4.25457 6.615974 6.828652 3.910027 5.596743 5.628633 6.873701
4.814183 6.155232 6.534484 4.720397 2.904609 3.30491 4.045343 3.952053 3.789159 4.263307
4.759324 2.353184 2.870954 2.302321 6.166962 3.626227 2.942716 2.420425 6.650276 5.489135
3.560108 4.041884 4.945207 3.933782 10.213988 6.306195 3.95678 3.828785 4.032602 4.465063
3.626622 5.059106 2.877187 2.718474 4.967771 2.019084 6.737092 7.780721 5.618063 4.631442
3.131239 8.517338 5.91211 7.274651 3.666455 5.956814 2.675215 4.084991 4.650164 2.406024
3.101433 1.824544 4.327885 2.126415 5.068315 5.350694 6.072783 2.714011 6.488409 4.325353
4.094103 3.9477 7.44429 7.164175 5.832352 4.344374 6.043571 1.367093 4.201445 2.676168
1.785253 3.239593 6.561884 5.643876 3.83905 3.629085 2.194222 2.005424 4.108656 6.626679
6.992088 5.548359 2.892116 5.142409 7.624831 3.006096 7.88197 6.904463 5.741383 2.198722
4.920558 6.083345 5.23124 2.407182 7.016883 2.754815 7.578985 5.36957 5.418922 8.934045
5.386454 4.679776 2.921503 3.768579 3.003448 4.844327 6.855677 4.107097 7.732021 3.924529
8.443169 9.911545 3.481838 4.631584 5.893996 1.859658 5.096739 3.189677 4.295698 4.408684
7.245854 2.46472 6.513379 4.810341 4.440959 4.337154 6.658241 4.066306 3.255843 4.092131
5.161509 7.638453 6.416214 2.2613 4.053208 4.984948 3.950657 4.547 2.214907 0.639122
3.970622 2.712702 5.996083 3.680188 7.342515 8.260081 2.418146 4.096128 7.441699 5.729759
3.77537 4.757572 4.402373 0.903924 5.326952 4.671893 4.902297 4.270468 6.093635 7.290702
5.99371 8.170184 6.057584 5.006556 5.692901 3.875907 5.670595 6.639712 3.824252 7.582311
5.071729 3.095206 7.059571 8.035687 7.463707 1.27355 3.000708 3.284586 6.090252 5.976479
6.862248 1.164626 3.366987 2.966571 3.239442 5.113505 3.864661 4.443924 3.636662 8.612806
2.454884 4.46112 6.578096 2.692368 3.820151 8.064883 4.494706 2.653483 5.251923 1.807386
3.268051 7.523858 3.594618 5.290678 4.368644 2.522279 3.120291 7.295306 5.621461 5.343205
2.697101 4.516327 3.587215 2.705778 9.388659 7.891972 5.31875 4.860299 6.934203 4.483264
3.732855 7.59911 1.840256 4.305623 3.808503 5.989378 6.578569 1.761413 10.014783 4.965525
4.157876 6.275735 2.565145 1.914241 2.358519 6.155413 3.870906 4.160361 4.750009 4.089239
3.782217 0.956823 6.461397 1.622033 2.944158 4.089566 4.093222 1.289773 3.399538 3.824188
3.524273 5.644818 5.084473 7.553252 6.587813 3.584785 4.309272 1.854301 4.54006 5.941266
```

4.54125	1.944399	6.119431	4.346149	7.21035	4.652964	4.085278	4.599012	6.22877	2.344079
4.216534	7.607488	3.285029	4.532767	7.221152	3.324232	2.697281	5.879698	2.022291	4.374762
5.311087	1.686313	3.535125	4.345978	1.465712	7.438652	3.339191	1.48682	6.233615	4.908448
1.937106	3.955442	6.747417	3.690776	2.588832	4.652763	11.875929	7.546979	6.533185	7.563211
1.949452	3.012849	6.867713	3.358134	2.709739	2.914573	4.443212	3.949661	5.046777	6.499583
2.180692	4.693216	5.996684	5.763656	3.182614	2.310086	2.971135	9.018024	4.783431	4.546006
7.44589	5.416306	4.170012	5.085613	6.394022	3.675864	4.169601	5.828676	2.977224	8.599496
4.812133	7.514206	4.258223	4.36543	5.18918	6.406311	2.974807	3.630991	3.308214	5.123156
3.818677	2.092654	4.257202	2.532631	2.292004	4.294049	8.493359	4.234591	4.410943	5.898012
1.935975	5.388417	10.606632	4.088427	2.266276	7.240456	5.447934	5.72148	3.012751	7.290307
4.605633	3.161235	3.657039	5.188488	3.606641	3.944434	2.040963	4.958029	4.059597	5.878092
8.078674	6.349588	3.516679	4.457955	6.626871	4.048692	3.260949	5.881597	6.299658	4.968661
3.950845	4.024648	4.047865	4.757695	4.882248	4.805547	4.967763	5.196727	6.435197	5.370157
4.095083	0.911247	7.56235	6.846233	2.153204	5.150889	3.801726	7.679555	3.63939	8.36837
2.980395	5.210527	5.852333	7.374261	5.629985	10.124962	3.248711	3.736896	3.54921	2.599695
4.21672	4.820714	1.744245	9.218746	2.868069	5.612867	4.301308	3.63081	5.507686	3.011469
2.734074	2.673831	5.476139	1.746357	7.402151	7.609691	3.837047	3.598181	11.586922	3.328518
6.957482	4.257566	3.364324	5.150166	4.298561	9.742079	4.47064	6.636982	3.702012	1.797339
6.214357	3.490552	4.637603	8.889174	1.971447	3.231009	2.516258	6.252646	4.029256	4.778304
4.034452	6.702186	2.977673	7.277759	3.765688	6.670596	5.640704	2.8088	4.1892	9.684258
6.53389	3.227983	6.637699	4.253876	6.163148	7.842581	7.014663	3.68521	4.713713	1.882006
3.130667	4.113598	6.677949	7.085313	3.068059	6.723111	7.309703	2.811773	5.620105	3.475844
5.841399	7.460356	4.952559	6.447719	4.841917	2.569614	3.730757	1.177965	1.512455	1.741799
5.819497	1.600871	4.140809	4.594465	6.446527	6.378305	4.345328	6.640948	4.94475	1.741137
3.859103	5.120862	6.02082	6.444698	5.06664	2.715686	5.158985	3.680253	6.675818	6.64896
7.083923	3.163774	6.578399	3.141075	4.559855	4.198702	2.822253	3.931598	10.117534	1.677775
4.467829	3.636398	7.380956	2.037311	7.005847	2.439551	7.775669	2.411262	6.72908	7.411604
1.705545	10.678095	3.422666	5.81938	3.156825	5.103974	2.746526	2.685259	5.619641	3.146493
5.712669	3.848941	3.770966	7.006608	3.287068	0.709524	6.154126	6.567705	5.906152	2.63243
1.870833	4.409025	4.193829	3.290382	7.325259	4.002716	4.880749	4.149124	3.854015	3.947274
4.395495	3.446163	3.821235	2.375071	3.415398	2.669786	1.771365	6.695511	6.289227	6.498787
7.203458	3.66997	7.343665	4.128408	3.882703	5.149755	3.834989	8.484731	4.150211	4.263062
0.726491	4.909341	4.573164	4.789114	3.378079	5.916091	5.802967	6.796915	2.948484	4.035629
2.227719	4.68603	6.966399	8.544476	6.080928	5.04833	2.568933	3.303448	3.241151	5.193198
4.501849	3.756814	4.832896	3.391234	5.115501	5.119315	6.347194	2.688365	4.228142	4.828844
7.334086	4.388925	9.120164	5.074614	1.456649	5.041677	6.067506	7.031034	6.57899	4.993737
7.209998	4.677824	3.416181	4.904292	7.204093	3.17643	6.319581	2.731698	2.18084	4.82378
1.155741	6.417757	4.57868	4.012056	5.992698	2.915107	4.293474	3.919707	3.462898	6.929063
2.966795	6.178192	4.381311	7.7611	6.959701	5.091151	4.798957	4.645705	4.466657	3.944388
5.978913	6.003632	1.621619	6.988135	4.178234	2.739049	3.765689	5.403355	5.439912	5.341857
6.792279	7.134889	2.42139	3.330532	3.55603	3.672975	3.960047	4.631055	6.829631	6.25393
3.445012	5.16046	6.304468	3.337951	5.659947	3.804673	5.042998	5.777338	5.958256	3.621321
3.046247	4.656204	4.770397	5.716468	5.893847	2.917068	2.713936	1.580596	2.70755	2.23974
3.365989	2.465138	6.440714	1.865186	1.597072	5.125562	5.771748	4.852165	3.133196	2.777964
7.156232	1.554392	6.412557	5.281639	3.708711	5.237367	7.248857	6.124798	5.580002	4.160395
6.679968	5.009287	7.386216	6.324586	6.280149	6.080202	3.30165	7.695203	8.130411	5.854252
2.197487	2.580184	7.521049	0.196011	3.711428	6.450143	11.528391	4.421968	6.535393	3.184683
5.570034	5.629892	3.207643	6.436756	4.464529	6.357115	5.320271	7.007097	3.548776	7.620029
6.121661	2.053348	3.977271	6.273099	0.54645	4.43155	4.525706	5.687972	3.190659	5.497928
3.071362	4.228432	1.917353	6.704779	5.192228	4.025878	5.843825	6.784509	8.612941	4.595166
6.407314	5.407114	7.340102	3.694387	5.000013	2.648222	2.568859	2.040521	3.62977	4.418521
2.393188	4.113742	3.409048	3.946656	5.294355	4.221921	6.073485	7.972198	4.506334	10.430841
7.489839	3.939916	7.020677	5.713666	7.155438	6.355261	4.867949	1.858758	3.571529	3.409232
2.829955	5.194716	5.442708	4.850288	2.560345	3.366481	5.164106	5.788603	8.553069	5.396935

Выборка из распределения Максвелла, n = 1000

2.2.2. Построение эмпирической функции распределения

Опираясь на утверждения об Эмпирической Функции Распределения из пункта **2.1.2**, построим графики ЭФР. Эмпирическая кумулятивная функция распределения, возвращающая на основе выборки и числа t долю значений в выборке, меньших t, представлена в листинге ниже (под именем *CDF*).

Листинг кода:

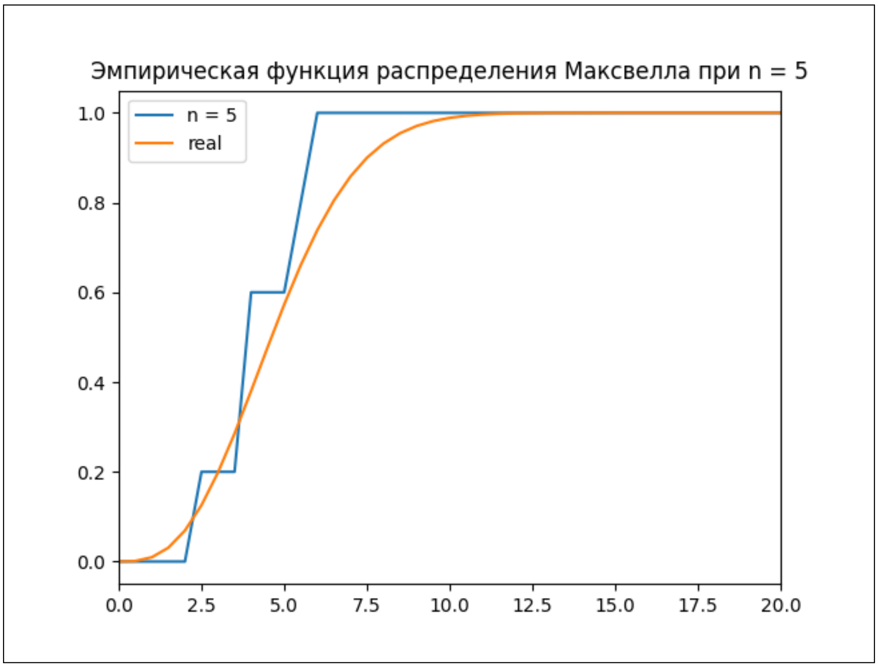
```
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import maxwell

def CDF(select, t):
    sum = 0
    for element in select:
        sum += int(element <= t)
    return sum / len(select)

volumes = [5, 10, 100, 200, 400, 600, 800, 1000]
for volume in volumes:

    selection = []
    tmp_1 = open(f'{volume}.txt', 'r').read()
    tmp_2 = tmp_1.split('\n')
    tmp_3 = []
    for i in tmp_2:
        tmp_3.append(i.split('␣'))
    for i2 in tmp_3:
        for j in range(len(i2)):
            selection.append(float(i2[j]))

    x = np.arange(0, 100, 0.5)
    y_true = [maxwell.cdf(t, loc = 0, scale = 3) for t in x]
    y = [CDF(selection, t) for t in x]
    plt.title(f'                    ␣                    ␣
    plt.xlim(0, 20)
    plt.plot(x, y, label = f'n␣=␣{volume}')
    plt.plot(x, y_true, label = 'real')
    plt.legend()

    plt.show()
```

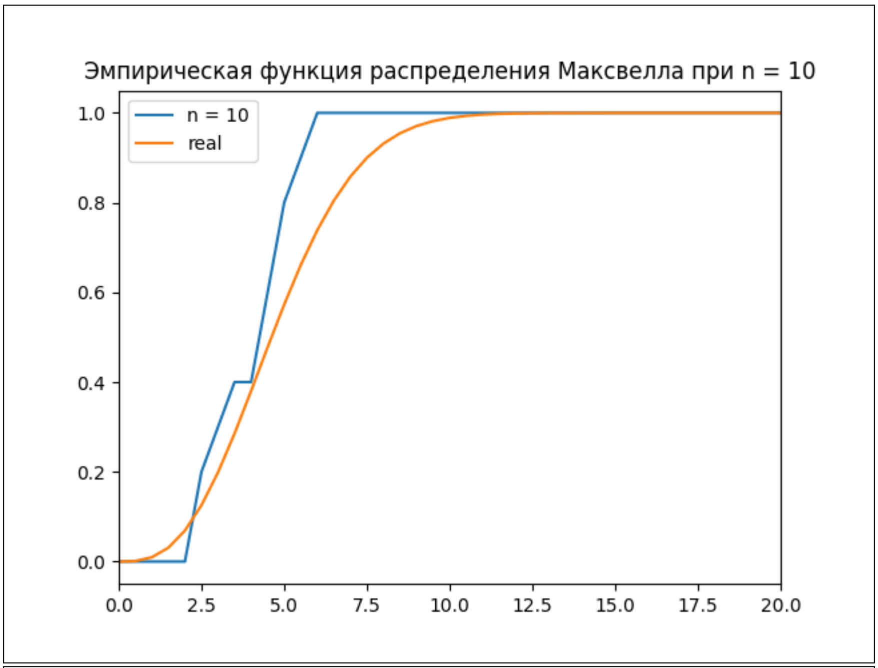
Ниже представлены графики Эмпирической Функции Распределения для каждой выборки с графиками функции распределения случайной величины:

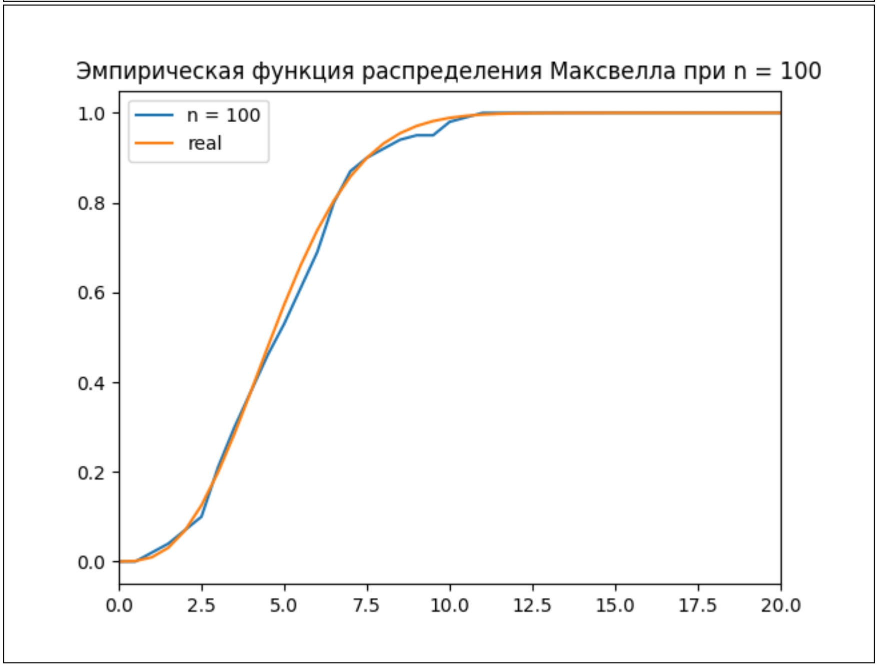
Также, проанилизировав приведённые графики, можно сделать вывод, что при увеличении объёма выборки график кумулятивной эмпирической функции распределения всё больше "стремится"к графику функции распределения случайной величины. Также это подтверждается ***Теоремой*** [***29*** *- Фомин Д. Б, Чухно А. Б., "Математическая статистика. Курс лекций для студентов кафедры компьютерной безопасности":* ***стр. 8***], гласящей о том, что:

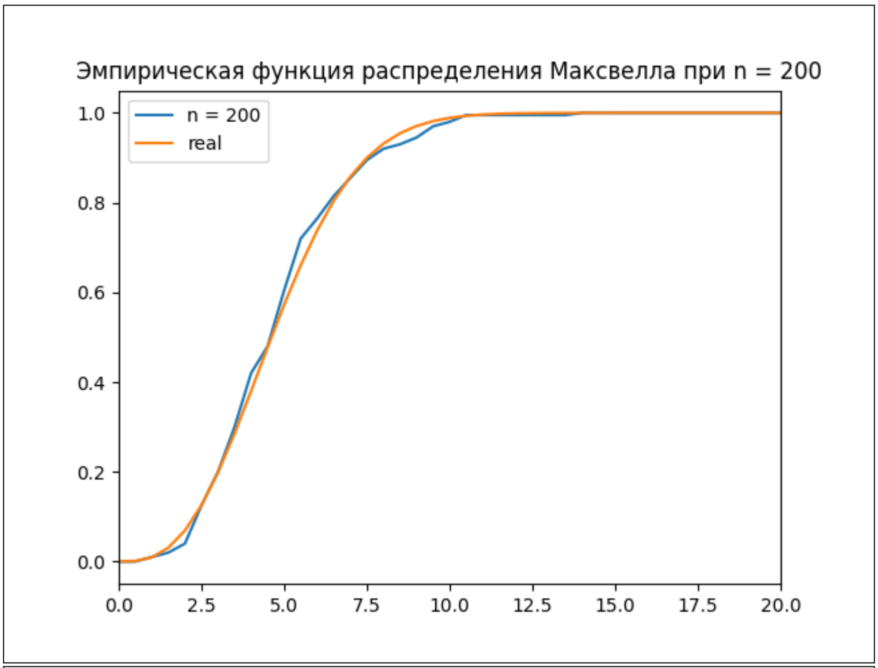
$$\text{Для } \forall x \in \mathbb{R} \text{ и для } \forall \epsilon > 0 \text{ при } n \longrightarrow \infty$$
$$P\left(\left|\widehat{F}_n(x) - F(x)\right| < \epsilon\right) \longrightarrow 1$$

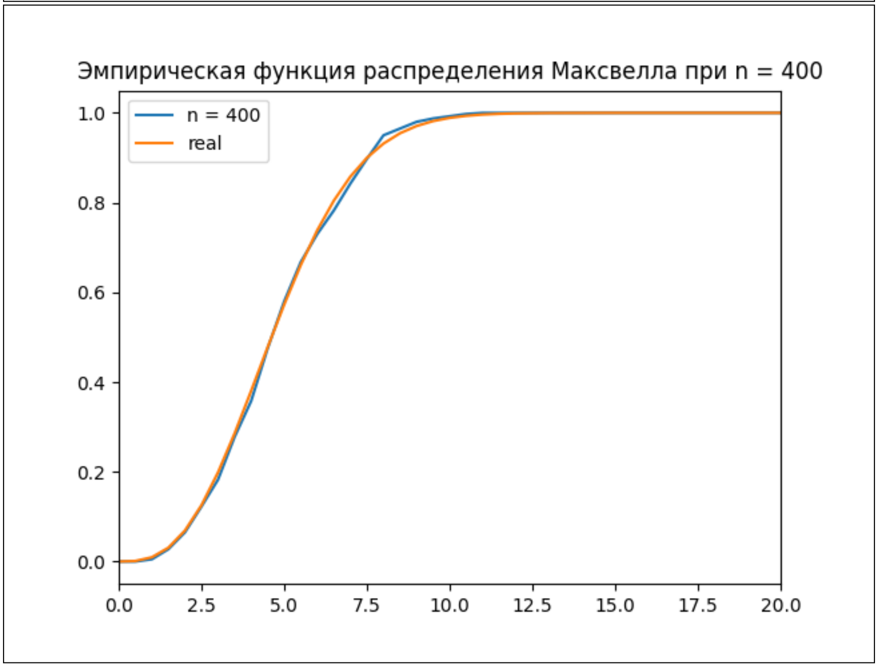
Другими словами: для произвольного фиксированного $y \in \mathbb{R}$ э.ф.р. $\widehat{F}_n(y)$ с увеличением объема выборки n стремится к значению функции распределения $F(y)$.

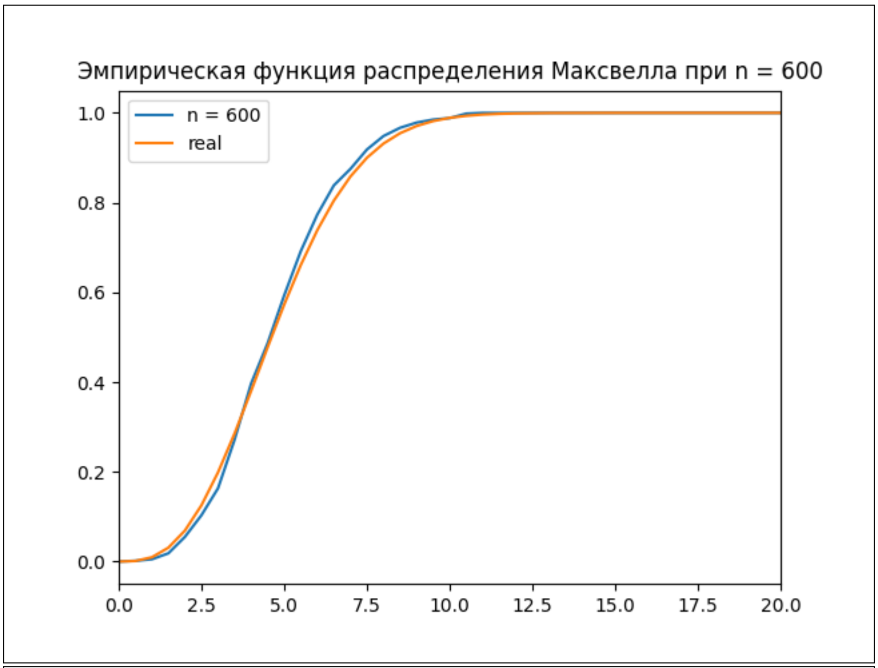


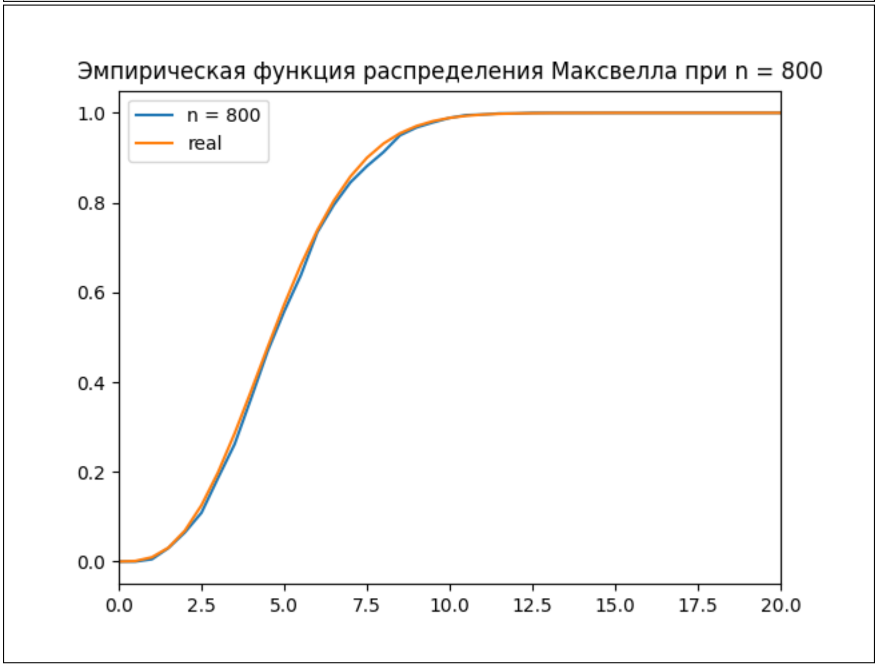

Эмпирическая функция распределения Максвелла при n = 10
n = 10
real
0.0
0.2
0.4
0.6
0.8
1.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

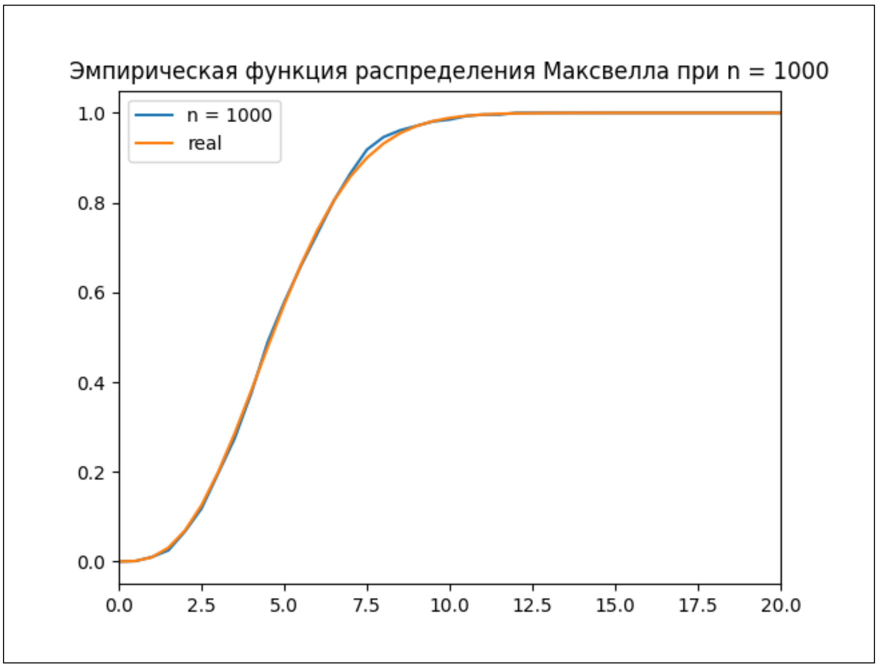

Эмпирическая функция распределения Максвелла при n = 100
n = 100
real
0.0
0.2
0.4
0.6
0.8
1.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Эмпирическая функция распределения Максвелла при n = 200
n = 200
real
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Эмпирическая функция распределения Максвелла при n = 400
n = 400
real
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Эмпирическая функция распределения Максвелла при n = 600
n = 600
real
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


Эмпирическая функция распределения Максвелла при n = 800
n = 800
real
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0



Для посчета $D_{n,m}$ необходимо описать функцию супремума, которая описана ниже:

```
def sup(selection1, selection2):
    x = np.arange(0, 87, 0.5)
    f1 = [eCDF(selection1, t) for t in x]
    f2 = [eCDF(selection2, t) for t in x]
    s = 0
    for i in range(len(x)):
        s = max(s, abs(f1[i] - f2[i]))
    return s
```

Описав функцию супремума, опишем и саму функцию подсчета $D_{n,m}$:

```
def D(selection1, selection2):
    m = len(selection1)
    n = len(selection2)
    return np.sqrt(m * n/(m + n)) * sup(selection1, selection2)
```

Произведём расчёты функции $D_{n,m}$ для каждой пары выборок с помощью скрипта на языке Python. Получим матрицу значений, которая, что примечательно, симметрична относительно главной диагонали.

	5	10	100	200	400	600	800	1000
5	0.0000	0.3651	0.6765	0.5190	0.6000	0.5047	0.5935	0.6000
10	0.3651	0.0000	0.9347	0.7252	0.8433	0.7109	0.8367	0.8464
100	0.6765	0.9347	0.0000	0.8981	0.5143	0.7715	0.4125	0.4577
200	0.5190	0.7252	0.8981	0.0000	0.7217	0.4491	1.0436	0.8004
400	0.6000	0.8433	0.5143	0.7217	0.0000	0.8650	0.6124	0.3803
600	0.5047	0.7109	0.7715	0.4491	0.8650	0.0000	1.0030	0.8198
800	0.5935	0.8367	0.4125	1.0436	0.6124	1.0030	0.0000	0.7748
1000	0.6000	0.8464	0.4577	0.8004	0.3803	0.8198	0.7748	0.0000

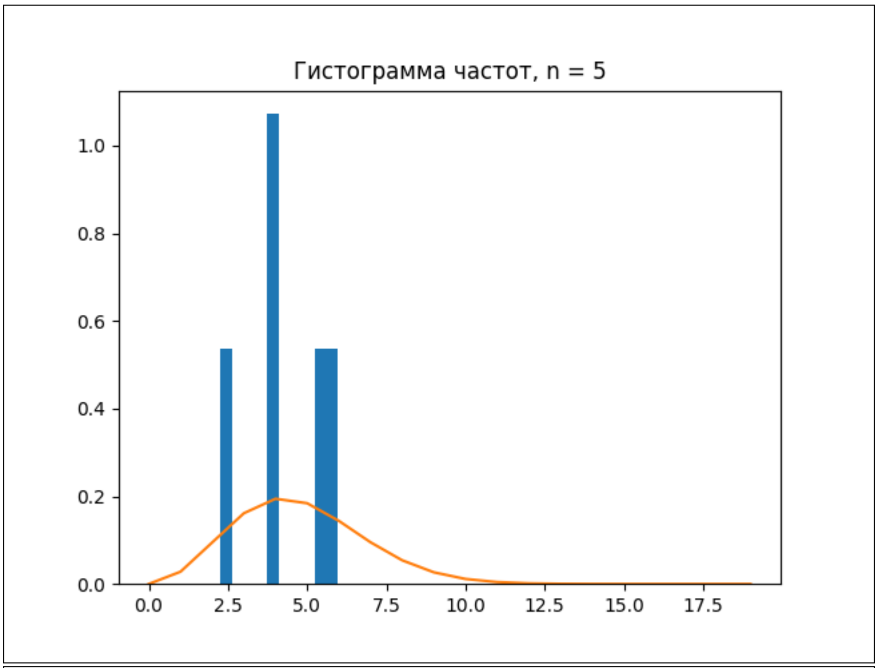
2.2.3. Построение гистограммы частот

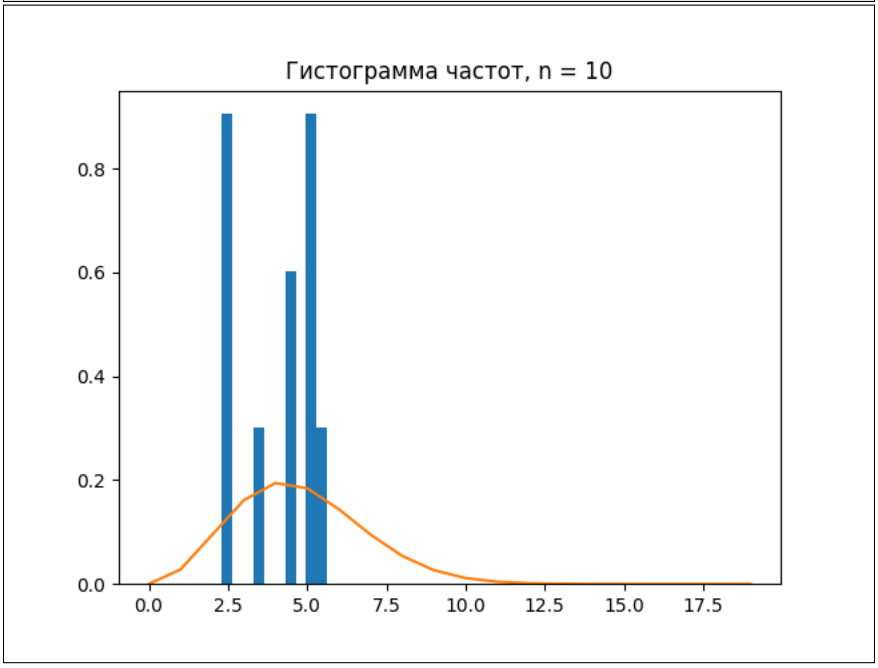
Ниже представлены гистограммы частот для каждой выборки с графиками функции плотности распределения:
Из приведенных ниже гистограмм частот можно сделать вывод, что с увеличением объёма выборки - её гистограмма частот "стремится"к функции плотности распределения. Это иллюстрирует следующую теорему:

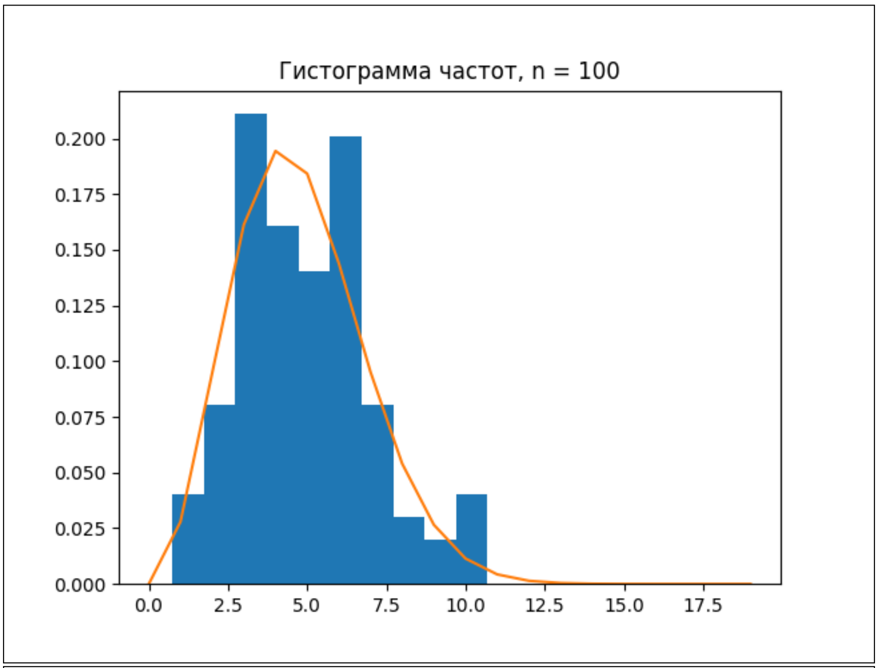
[***30*** *- Н. И. Чернова, "Лекции по математической статистике Нижегородский Государственный Университет -* ***стр. 12, стр. 20***] Пусть распределение F абсолютно непрерывно, f - его истинная плотность. Пусть, кроме того, число k интервалов группировки не зависит от n. Тогда справедлива Теорема:

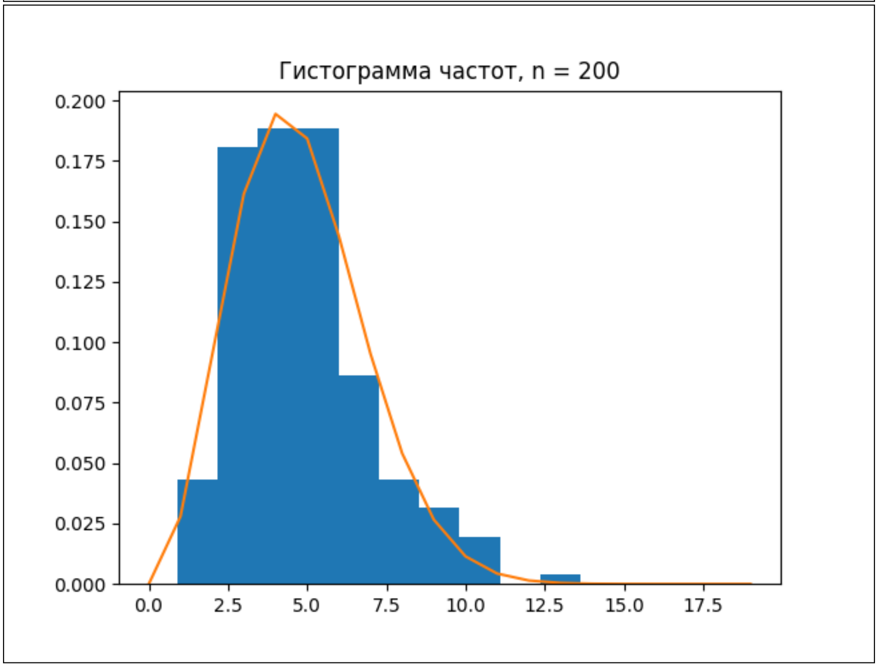
$$\text{При } n \longrightarrow \infty \ \forall j = 1, \cdots, k$$
$$l_j \cdot f_j = \frac{vi}{n} \longrightarrow^p P(X_1 \in A_j) = \int_{A_j} f(\chi) d\chi$$

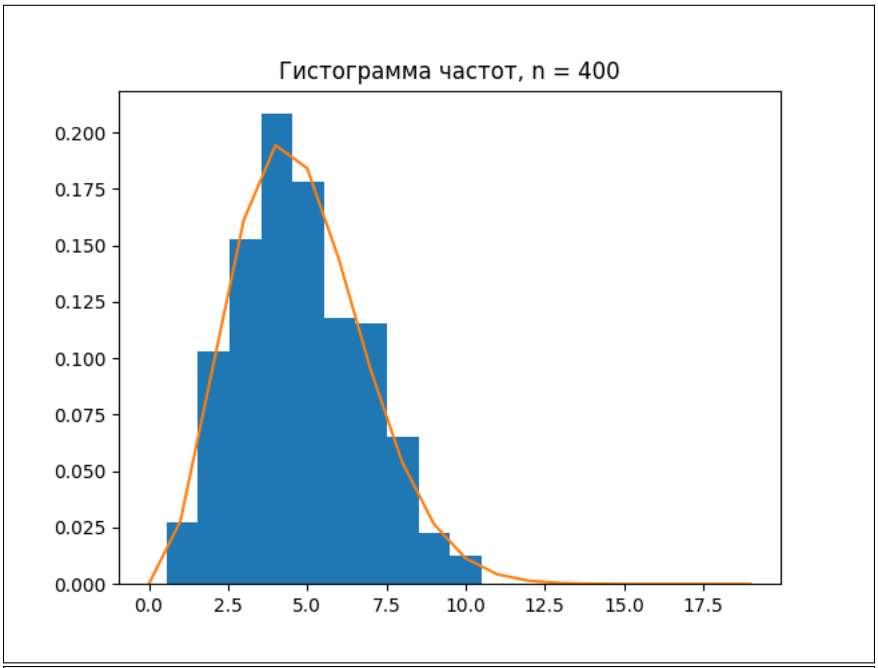
Предполагаемую область значений случайной величины ξ делят независимо от выборки на некоторое количество интервалов (не обязательно одинаковых). Пусть $A_1, \cdots, A_k$ - интервалы на прямой, называемые интервалами группировки. Обозначим для $j = 1, \cdots, k$ через v_j число элементов выборки, попавших в интервал A_j. l_j - длина интервала A_j
Данная теорема утверждает, что площадь столбца гистограммы, построенного над интервалом группировки, с ростом объема выборки сближается с площадью области под графиком плотности над этим же интервалом.

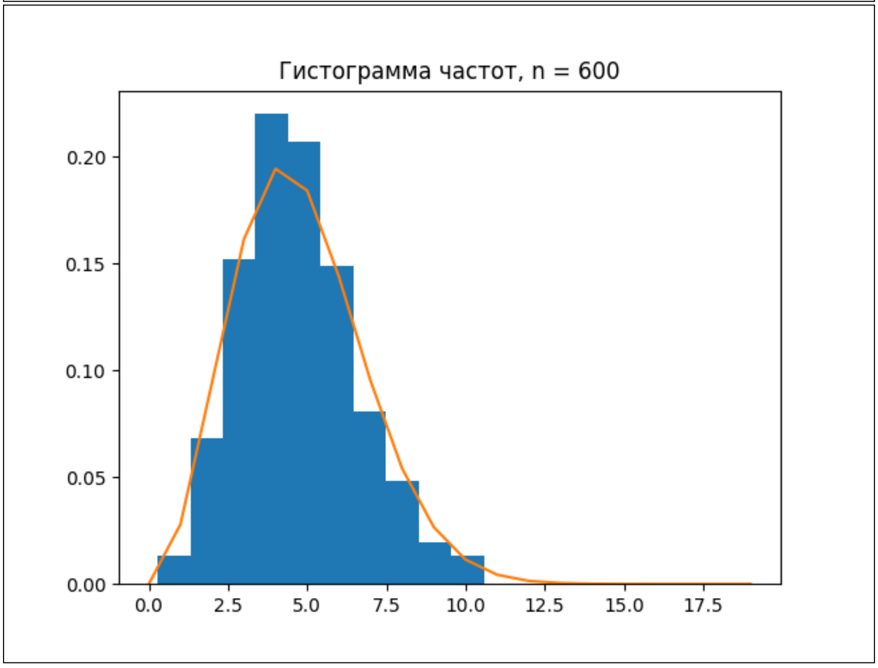

Гистограмма частот, n = 5
1.0
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

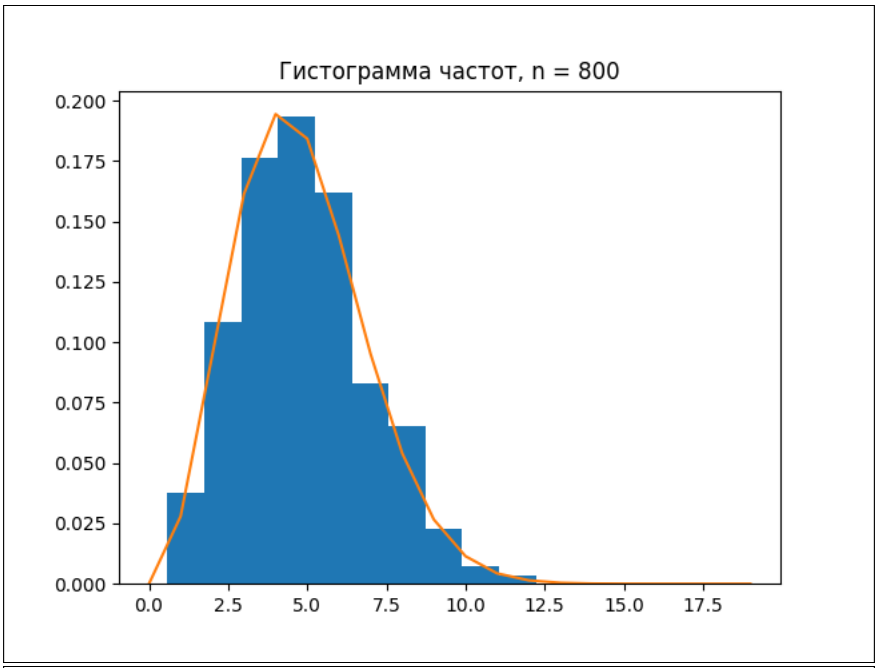

Гистограмма частот, n = 10
0.8
0.6
0.4
0.2
0.0
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5

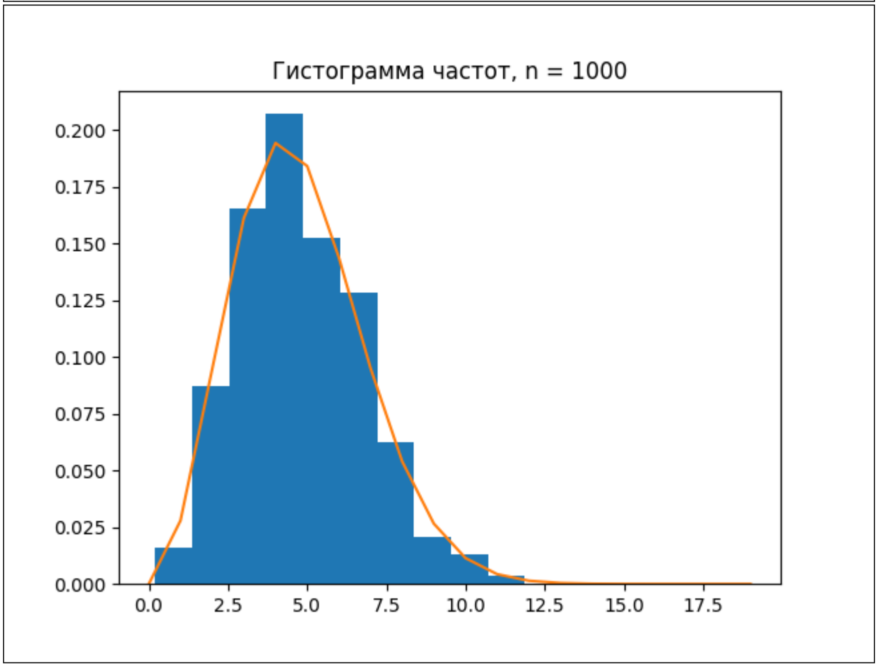

Гистограмма частот, n = 100
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Гистограмма частот, n = 200
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Гистограмма частот, n = 400
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5


Гистограмма частот, n = 600
0.20
0.15
0.10
0.05
0.00
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5





Листинг кода:

```
import random
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import maxwell

volumes = [5, 10, 100, 200, 400, 600, 800, 1000]

for volume in volumes:
    selection = []
    tmp_1 = open(f'{volume}.txt', 'r').read()
    tmp_2 = tmp_1.split('\n')
```

```
tmp_3 = []
for i in tmp_2:
    tmp_3.append(i.split(' '))
for i2 in tmp_3:
    for j in range(len(i2)):
        selection.append(float(i2[j]))
x = np.arange(0, 20, 1)
y_true = [maxwell.pdf(t, loc = 0, scale = 3) for t in x]
plt.title(f'                              , n = {volume}'
plt.hist(selection, density=True)
plt.plot(x, y_true)
plt.show()
```

2.2.4. Вычисление выборочных моментов

Оценка математического ожидания является несмещенной и состоятельной (факты, доказанные в пункте **2.1.4**)
Оценка дисперсии является смещенной и состоятельной (факты, доказанные в пункте **2.1.4**)

Истинное выборочное среднее при параметрах $\theta = 3$:
$2 \cdot \theta \cdot \sqrt{\dfrac{2}{\pi}} = 2 \cdot 3 \cdot \sqrt{\dfrac{2}{\pi}} = 4.787$
Истинное выборочная дисперсия при параметрах $\theta = 3$:
$\theta^2 \cdot \left(3 - \dfrac{8}{\pi}\right) = 9 \cdot \left(3 - \dfrac{8}{\pi}\right) = 4.081$

	Выборочное среднее	Выборочная дисперсия
5	4.247083	1.702023
10	4.015007	1.373846
100	4.913280	4.537851
200	4.778659	4.404227
400	4.807834	3.795886
600	4.735343	3.436854
800	4.889886	4.117263
1000	4.780937	3.903749

Вычислим погрешность выборочного среднего и выборочной дисперсии, зная истинные значения данных величин:

	Погрешность выборочного среднего	Погрешность выборочной дисперсии
5	-11.28%	-58.3%
10	-16.13%	-66.34%
100	2.63%	11.18%
200	-0.18%	7.9%
400	0.43%	-7.0%
600	-1.09%	-15.8%
800	2.14%	0.87%
1000	-0.13%	-4.36%

Листинг кода:

```
import numpy as np
import pandas as pd
```

```
volumes = [5, 10, 100, 200, 400, 600, 800, 1000]
theta = 3.0

results = []
for volume in volumes:
    selection = []
    tmp_1 = open(f'{volume}.txt', 'r').read()
    tmp_2 = tmp_1.split('\n')
    tmp_3 = []
    for i in tmp_2:
        tmp_3.append(i.split(' '))
    for i2 in tmp_3:
        for j in range(len(i2)):
            selection.append(float(i2[j]))
    results.append(selection)

aver_mine = np.sqrt(2/np.pi) * 2 * theta
var_mine = (theta ** 2) * (3 - (8/np.pi))
'''print('                              : ', aver_mine)
print('                          : ', var_mine)'''


means = [np.average(element) for element in results]
variances = [np.var(element) for element in results]
dict1 = {'                             ':means, '
table = pd.DataFrame(data=dict1)
table.index = volumes
print(table)


means_2 = [f'{round((((s - aver_mine) / aver_mine) * 100), 2)}%' f
variances_2 = [f'{round((((s - var_mine) / var_mine) * 100), 2)}%'
dict2 = {'
table2 = pd.DataFrame(data=dict2)
table2.index = volumes
print(table2)
```