



Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИУК «Информатика и управление»

КАФЕДРА ИУК4 «Программное обеспечение ЭВМ,

информационные технологии»

Домашняя работа №1

«Первичная обработка данных»

ДИСЦИПЛИНА: «Методы обработки информации»

Выполнил: студент гр. ИУК4-72Б _____ (_____)
(подпись) (Ф.И.О.)

Проверил: _____ (_____)
(подпись) (Ф.И.О.)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:

- Оценка:

Калуга, 2023

Цель: формирование у студентов практических навыков обработки статистических данных.

Задачи: моделирование непрерывной СВ, анализ исходных данных, построение оценок плотности вероятности, нахождение точечных и интервальных оценок параметров распределения.

Задание 1

Постановка задачи

Вариант №6

6	Накагами	$m = 0.5, \sigma = 1$
---	----------	-----------------------

1. Выполнить статистическое моделирование случайной величины с заданным законом распределения (табл. 1) путем генерации отсчетов $\alpha 1i$, $i = 1, \dots, N$ случайных величин с 6 равномерным распределением в интервале $[0, 1]$ (или, при необходимости нескольких СВ ($\alpha 1, \alpha 2, \dots, \alpha k$); $N=10000$. Сформировать соответствующий script-файл в среде MATLAB.

2. Получить гистограмму для закона распределения в соответствии с вариантом задания. Гистограмма может быть получена в среде MATLAB с помощью оператора `hist(X1,N)`, $X1$ — анализируемая случайная величина, N — число интервалов на гистограмме, которое должно составлять от 100 до 500. Сравнить полученную гистограмму с соответствующим графиком плотности вероятности $f(x)$ в соответствии с заданием.

3. Вычислить: — выборочное среднее значение, — медиану, — нижний и верхний квартиль, — выборочную дисперсию и СКО, смоделированной случайной величины и сравнить их с теоретическими значениями (мат. ожиданием и дисперсией, медианой, нижним и верхним квартилем).

4. Сделать выводы.

Листинг программы:

```
% Вариант 6
% Накагами
% sigma=1, m=0.5
```

```

sigma = 1;
m = 0.5;
func = @(x) ((2*m^m)/(gamma(m)*sigma^m))*x^(2*m-1)*exp(-(m/sigma)*x^2) * 200;

size = 5000;
N = 100;
xs = randn(1, size);
x_range = linspace(0, 1, N);
y_range = linspace(0, 1, N);
for i=1:N
    y_range(1, i) = func(x_range(1, i));
end
alpha = nakagami_dist(m, sigma, size);
histogram(alpha, N);
hold on;
plot(x_range, y_range);
meanValue = mean(alpha); % Выборочное среднее
medmnValue = median(alpha); % Медиана
lowerQuantile = quantile(alpha, 0.25); % Верхний квартиль
upperQuantile = quantile(alpha, 0.75); % Верхний квартиль
varmnce = var(alpha); % Выборочная дисперсия
stdDevmtion = std(alpha); % Стандартное отклонение
dist = makedist('Nakagami', m, sigma);
fprintf('-----МОДЕЛЬ-----\n');
fprintf('Выборочное среднее значение: %.4f\n', meanValue);
fprintf('Медиана: %.4f\n', medmnValue);
fprintf('Нижний квартиль: %.4f\n', lowerQuantile);
fprintf('Верхний квартиль: %.4f\n', upperQuantile);
fprintf('Выборочная дисперсия: %.4f\n', varmnce);
fprintf('Стандартное отклонение: %.4f\n', stdDevmtion);
fprintf('-----ТЕОРИЯ-----\n');
fprintf('Выборочное среднее значение: %.4f\n', mean(dist));
fprintf('Медиана: %.4f\n', median(dist));
fprintf('Нижний квартиль: %.4f\n', icdf(dist, 0.25));
fprintf('Верхний квартиль: %.4f\n', icdf(dist, 0.75));
fprintf('Дисперсия: %.4f\n', var(dist));
fprintf('Стандартное отклонение: %.4f\n', std(dist));

function x = gamma_dist(alpha, beta, n)

x = zeros(n,1);

for i = 1:n
    while true
        u1 = rand();
        u2 = rand();
        v1 = log(u1)/alpha;
        v2 = log(u2);
        if alpha < 1
            w = u2^(1/alpha);
        else
            w = exp(-v2*(alpha-1));
        end
        if w <= (1 - alpha*v1)
            x(i) = w^(1/alpha)*beta;
            break;
        end
    end
end

end
function x = nakagami_dist(m, sigma, n)

```

```

x = zeros(n,1);
for i = 1:n
    x(i) = sqrt(gamrnd(m, sigma/m, 1)/(2*m)*sigma);
end

end

```

Результат:

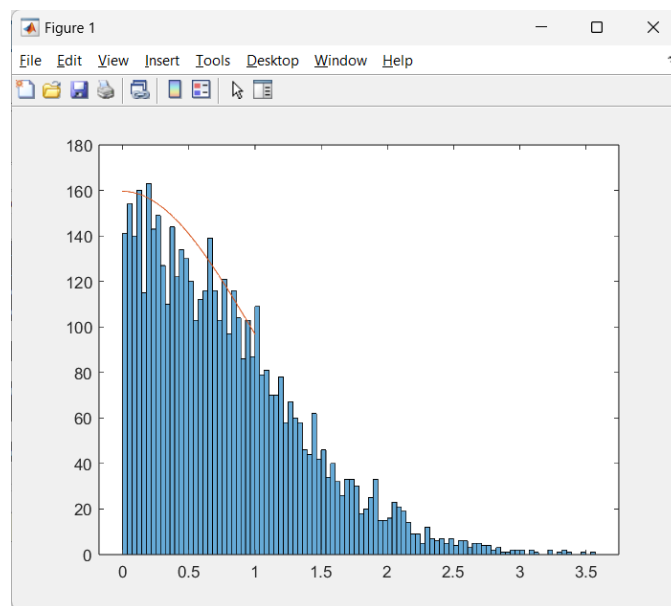


Рисунок 1 – Графики теоретической и экспериментальной

```

-----МОДЕЛЬ-----
Выборочное среднее значение: 0.7853
Медиана: 0.6741
Нижний квартиль: 0.3096
Верхний квартиль: 1.1182
Выборочная дисперсия: 0.3517
Стандартное отклонение: 0.5931
-----ТЕОРИЯ-----
Выборочное среднее значение: 0.7979
Медиана: 0.6745
Нижний квартиль: 0.3186
Верхний квартиль: 1.1503
Дисперсия: 0.3634
Стандартное отклонение: 0.6028

```

Рисунок 2 – Полученные характеристики теоретического и экспериментального распределения

Задание 2

Вариант 14

Постановка задачи

Для обработки преподавателем выдается случайных чисел.

Эти числа хранятся в файле TestNN.csv.

1. Выборка подвергается обработке и оформляется в виде таблицы.
2. Графические характеристики выборки – строим гистограмму и полигон приведенных частот. Выдвигаем гипотезу о виде плотности вероятности генерального распределения.
3. Находим выборочные характеристики положения и рассеивания.
4. Для сравнения с гистограммой и полигоном приведенных частот на одном чертеже постройте графики гистограммной оценки плотности вероятности $f_{\Gamma}(x)$ параметрической оценки плотности вероятности $f_{\Pi}(x)$, и усредненную ядерную оценку плотности вероятности $f_{\text{УЯ}}(x)$.
5. Значения оценок плотности вероятности в средних точках промежутков группированного статистического ряда оформите в виде таблицы.
6. Проанализируйте близость оценок по средним квадратическим отклонениям $f_{\text{УЯ}}(x)$ и $f_{\Pi}(x)$ от $f_{\Gamma}(x)$.

Листинг программы

```
import argparse
import csv

import numpy as np
import prettytable

import matplotlib.pyplot as plt
import statistics as st
from scipy.stats import gaussian_kde

if __name__ == "__main__":
    parser = argparse.ArgumentParser()
    parser.add_argument("-file") or "./data/Test14.csv"

    args = parser.parse_args()
    file = args.file

    points = []
```

```

with open(file, newline='') as csvfile:
    reader = csv.reader(csvfile, delimiter=' ', quotechar='|')
    for row in reader:
        points.append(float("".join(row)))
points.sort()

min_point = points[0]
max_point = points[-1]
points_range = max_point - min_point

print(f"Размах выборки: {points_range:.2f}")

num_bins = 1 + int(np.ceil(np.log2(len(points))))
print(f"Количество интервалов: {num_bins}")

step = points_range / num_bins
print(f"Длина интервала: {step:.2f}")

_bins = []
for i in range(num_bins):
    current_min = min_point + i * step
    current_max = min_point + (i + 1) * step
    current_range = (current_min, current_max)
    count = len(
        list(
            filter(
                lambda x: current_min <= x <= current_max, points
            )
        )
    )
    current_average = (current_max + current_min) / 2.
    _bins.append(
        {
            "average": round(current_average, 4),
            "minimum": round(current_min, 4),
            "maximum": round(current_max, 4),
            "count": round(count, 4)
        }
    )

table = prettytable.PrettyTable()
table.field_names = [
    "Номер промежутка", "a_{i-1}", "a_i", "n_i",
    "Средняя точка промежутка"
]

index = 1
for bin in _bins:
    table.add_row([index := index + 1, bin["minimum"], bin["maximum"],
        bin["count"], bin["average"]])

print(table)

print(f"Выборочное среднее: {np.mean(points):.2f}")
print(f"Медиана: {np.median(points):.2f}")
print(f"Мода: {st.mode(points):.2f}")
print(f"Размах выборки: {max(points) - min(points):.2f}")
print(f"Выборочная дисперсия: {np.var(points):.2f}")
print(f"Стандартное отклонение выборки: {np.sqrt(np.var(points)):.2f}")

```

```

print(f"Коэффициент вариации: "
      f"{np.sqrt(np.var(points)) / np.mean(points) / 100:.2f}")

plt.hist(
    points, color="grey", edgecolor="black", bins=num_bins, range=(
        min_point,
        max_point), alpha=0.5, density=True, label="Гистограмма"
)
centers = [bin["average"] for bin in _bins]
_bins_plot = [bin["count"] / 49 for bin in _bins]
plt.plot(centers, _bins_plot, color="black")

plt.show()

def normal_distribution(x):
    return 1 / np.sqrt(2 * np.pi) / np.sqrt(np.var(points)) * (
        np.exp(-1 / 2 * (
            (x - np.mean(points)) / np.sqrt(np.var(points))) ** 2
        )
    )

kde = gaussian_kde(points)
xs = np.linspace(min_point, max_point, 100)
plt.hist(
    points, color="grey", edgecolor="black", bins=num_bins, range=(
        min_point,
        max_point), alpha=0.5, density=True, label="Гистограмма"
)
plt.plot(xs, [kde(x) for x in xs], color="red",
         label="Усреднённая ядерная оценка")
plt.plot(xs, [normal_distribution(x) for x in xs], color="green",
         label="Параметрическая оценка")
plt.legend()

plt.show()

table = prettytable.PrettyTable()
table.field_names = [
    "z_i", "n_i", "f_Г(x)", "f_УЯ(x)", "f_П(x)", "(f_УЯ(x)-f_Г(x))^2",
    "(f_П(x)-f_Г(x))^2",
]

index = 1
for bin in _bins:
    current_average = bin["average"]
    current_count = bin["count"]
    current_histogram = round(current_count / len(points) / 0.5, 4)
    current_kde = round(float(kde(current_average)), 4)
    current_parametric = round(normal_distribution(current_average), 4)
    diff_kde = round((current_kde - current_histogram) ** 2, 4)
    diff_parametric = round((current_parametric - current_histogram) **
2, 4)
    table.add_row([
        current_average, current_count, current_histogram,
        current_kde, current_parametric, diff_kde, diff_parametric
    ])

print(table)

```

Результаты выполнения программы

```
Размах выборки: 3.25  
Количество интервалов: 9  
Длина интервала: 0.36
```

Рисунок 3 – Параметры построения гистограммы

Номер промежутка	a_{i-1}	a_i	n_i	Средняя точка промежутка
2	-1.538	-1.1768	10	-1.3574
3	-1.1768	-0.8157	16	-0.9963
4	-0.8157	-0.4546	13	-0.6352
5	-0.4546	-0.0935	20	-0.274
6	-0.0935	0.2677	22	0.0871
7	0.2677	0.6288	20	0.4482
8	0.6288	0.9899	5	0.8093
9	0.9899	1.351	19	1.1705
10	1.351	1.7121	10	1.5316

Рисунок 4 – Результат обработки выборки

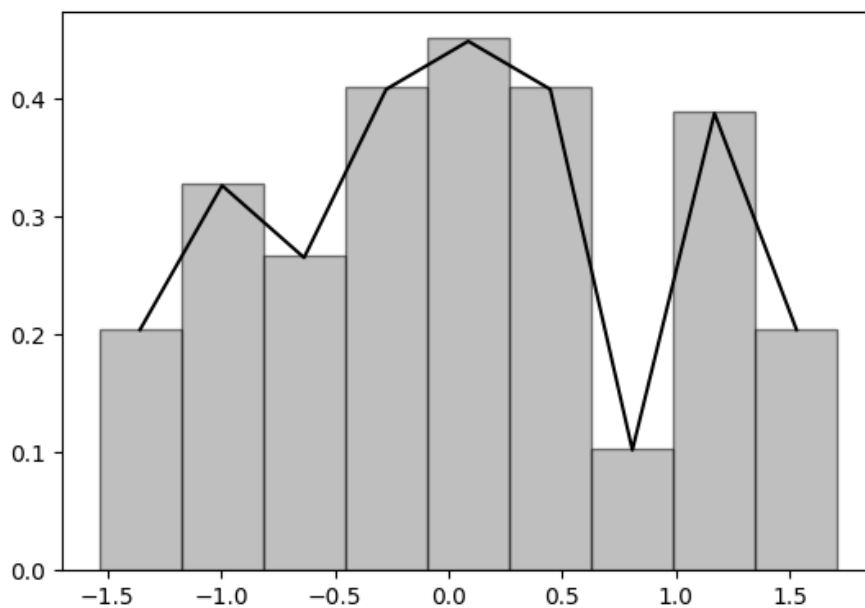


Рисунок 5 – Построенные гистограмма и полигон частот

```
Выборочное среднее: 0.07  
Медиана: 0.05  
Мода: -1.54  
Размах выборки: 3.25  
Выборочная дисперсия: 0.74  
Стандартное отклонение выборки: 0.86  
Коэффициент вариации: 0.12
```


Рисунок 6 – Выборочные характеристики положения и рассеяния

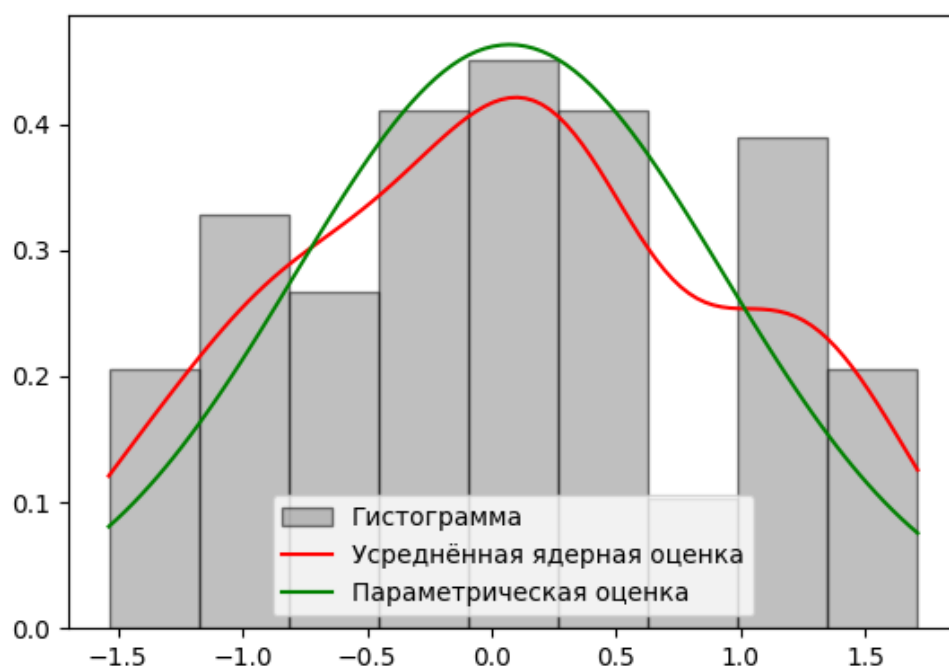


Рисунок 7 – Параметрическая и усреднённая ядерная оценки выборки

z_i	n_i	$f_{\Gamma}(x)$	$f_{УЯ}(x)$	$f_{П}(x)$	$(f_{УЯ}(x) - f_{\Gamma}(x))^2$	$(f_{П}(x) - f_{\Gamma}(x))^2$
-1.3574	10	0.1481	0.1685	0.1169	0.0004	0.001
-0.9963	16	0.237	0.2556	0.2146	0.0003	0.0005
-0.6352	13	0.1926	0.3149	0.3306	0.015	0.019
-0.274	20	0.2963	0.3752	0.4272	0.0062	0.0171
0.0871	22	0.3259	0.4212	0.463	0.0091	0.0188
0.4482	20	0.2963	0.3592	0.421	0.004	0.0156
0.8093	5	0.0741	0.2637	0.3211	0.0359	0.061
1.1705	19	0.2815	0.2508	0.2054	0.0009	0.0058
1.5316	10	0.1481	0.184	0.1102	0.0013	0.0014

Рисунок 8 – Значения плотностей вероятности в средних точках интервалов

Усреднённая ядерная оценка находится ближе к гистограммной оценке плотности, поскольку в отличие от параметрической она является асимметричной.

Задание 3

Постановка задачи

Сгенерировать выборку из 100 элементов, имеющих указанное в вашем варианте распределение. Считая один из параметров распределения неизвестным, найти его точечную оценку:

- а) методом моментов (с помощью указанных в задании моментов);
- б) методом максимального правдоподобия.

Построить график функции правдоподобия и убедиться, что найденная с помощью метода максимального правдоподобия оценка действительно является точкой максимума функции правдоподобия. Сравнить полученные точечные оценки с истинным значением параметра распределения.

Вариант 14

X - выборка из распределения χ_k^2 , где $k = 3$. Найти оценку параметра k , считая его неизвестным. Метод моментов реализовать с помощью моментов 1-го и 2-го порядков.

Ход выполнения практического задания

Выпишем формулы для нахождения математического ожидания и дисперсии для распределения χ^2 :

$$E\chi^2 = k$$

$$D\chi^2 = 2k$$

Получаем следующие точечные оценки для k :

Для момента 1-го порядка:

$$k^* = \bar{X}$$

Для момента 2-го порядка:

$$k^* = \frac{s^2}{2}$$

Найдём выборочные характеристики распределения:

```
Выборочные показатели
Выборочное среднее: 3.263741116279519
Выборочная дисперсия: 6.751452867399329

Теоретические показатели
Математическое ожидание chi(k=3.00): 3.0
Дисперсия chi(k=3.00): 6.0
```

Рисунок 8 – Выборочные и теоретические показатели распределения

Воспользовавшись методом моментов, найдём точечную оценку параметра k^* :

```
Метод моментов

Точечная оценка по 2-му моменту:
k=3.38
Математическое ожидание chi(k=3.38): 3.3757264336996644
Дисперсия chi(k=3.38): 6.751452867399329

Точечная оценка по 1-му моменту
k=3.26
Математическое ожидание chi(k=3.26): 3.263741116279519
Дисперсия chi(k=3.26): 6.527482232559038
```

Рисунок 10 – Точечные оценки параметра, полученные методом моментов

Построим графики, соответствующие полученным значениям параметра:

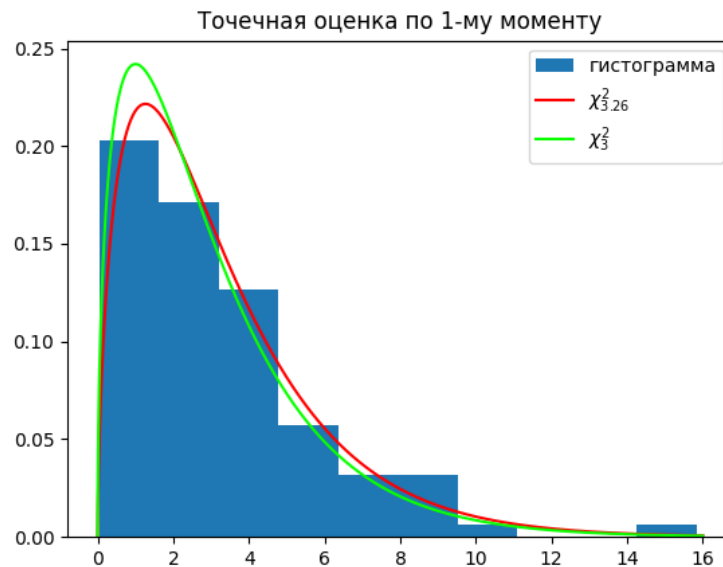


Рисунок 11 – График функции при точечной оценке, полученной по первому моменту

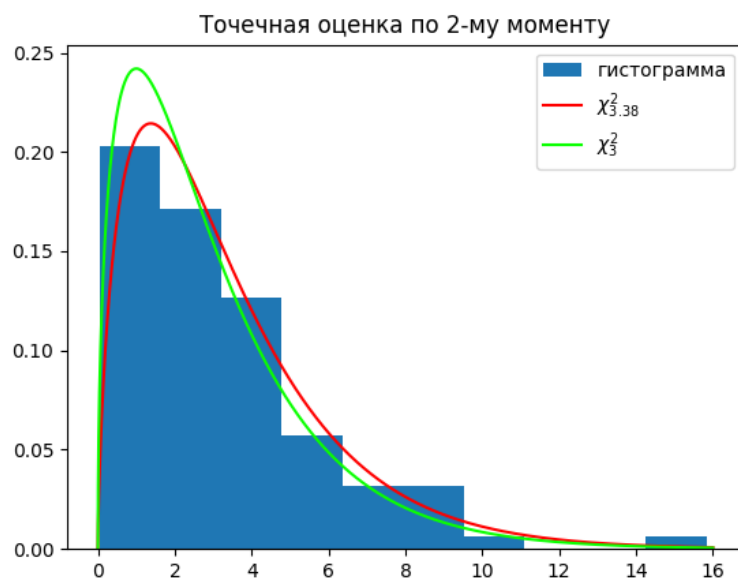


Рисунок 12 – График функции при точечной оценке, полученной по второму моменту

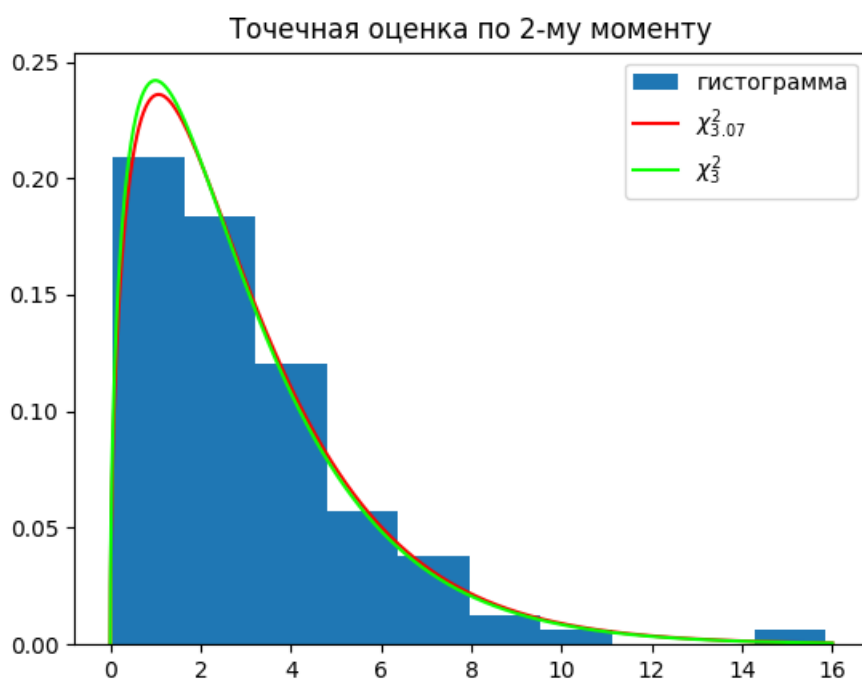


Рисунок 13 – Точечная оценка параметра k^* , вычисленная методом моментов 2-го порядка

Воспользуемся методом максимального правдоподобия.

Построим логарифмическую функцию правдоподобия для заданного распределения:

$$L(k) = \log(f(x; k)) = \log\left(\prod_{i=1}^n f(x_i; k)\right) = \sum_{i=1}^n \log\left(\frac{x_i^{\frac{k}{2}-1} e^{-\frac{x_i}{2}}}{\Gamma\left(\frac{k}{2}\right) 2^{\frac{k}{2}}}\right) =$$

$$= \left(\frac{k}{2} - 1\right) \sum_{i=1}^n \log x_i - \frac{1}{2} \sum_{i=1}^n x_i - n \log\left(\Gamma\left(\frac{k}{2}\right)\right) - \frac{nk}{2} \log 2$$

Построим график зависимости логарифмической функции правдоподобия на заданном промежутке значений k при заданных значениях выборки. Найдём максимальное значение функции и точку, соответствующую ему.

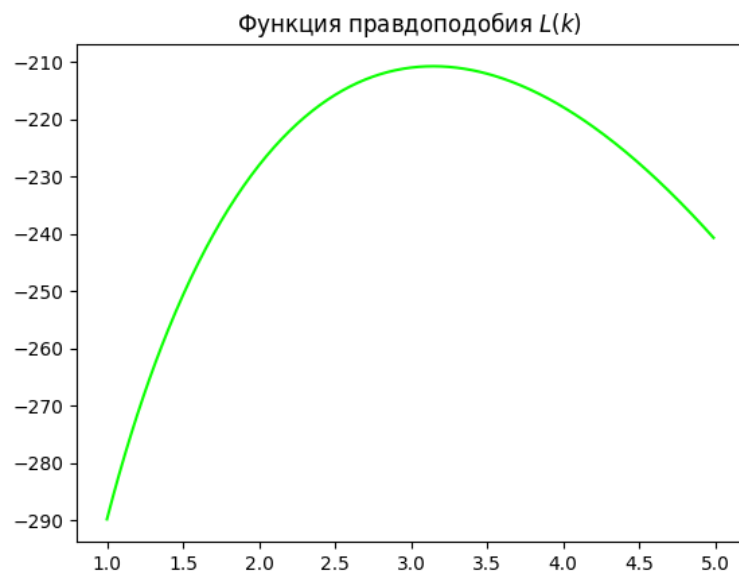


Рисунок 14 – Функция правдоподобия

```
Метод наибольшего правдоподобия
Наиболее правдоподобное значение параметра: k=3.15
Математическое ожидание chi(k=3.15): 3.150000000000002
Дисперсия chi(k=3.15): 6.300000000000004
```

Рисунок 15 – Точечная оценка параметра, полученная методом максимального правдоподобия

Построим график, соответствующий полученному значению параметра:

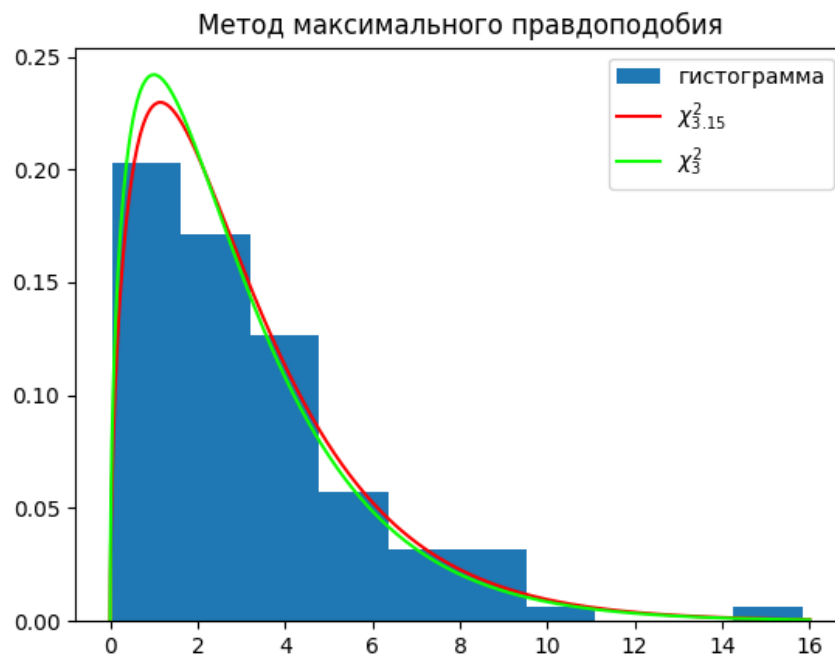


Рисунок 16 – График функции при точечной оценке, полученной методом максимального правдоподобия

Таким образом, получаем, что наиболее точной оказалась оценка, полученная методом моментов по второму моменту.

Задание 4

Постановка задачи

1. Для обеих выборок построить точный доверительный интервал уровня доверия q_0 для параметра σ^2 , считая:

- а) а неизвестным,
- б) а известным и равным a_0 .

2. В одной системе координат построить графики зависимости длины доверительного интервала от уровня доверия q для всех четырех случаев (объем выборки равен n_1 , а неизвестно; объем выборки равен n_1 , а известно; объем выборки равен n_2 , а неизвестно; объем выборки равен n_2 , а

известно). При этом q придать минимум 50 разных значений через равные промежутки.

Вариант 14

$$a_0 = 4, q_0 = 0.8$$

Ход выполнения практического задания

Примем размеры малой и большой выборок $n_1 = 15, n_2 = 70 \cdot n_1 = 1050$, соответственно.

Формула доверительного интервала для σ^2 при известном a :

$$P_{a, \sigma^2} \left(\frac{n \cdot s_1^2}{g_2} < \sigma^2 < \frac{n \cdot s_1^2}{g_1} \right) = 1 - \varepsilon, \text{ где}$$

s_1^2 – выборочная дисперсия,

g_1 и g_2 –

квантили распределения χ^2 с n степенями свободы уровня $\alpha = \frac{\varepsilon}{2}$ и $\alpha = 1 - \frac{\varepsilon}{2}$.

Формула доверительного интервала для σ^2 при неизвестном a :

$$P_{a, \sigma^2} \left(\frac{(n-1) \cdot s_0^2}{g_2} < \sigma^2 < \frac{(n-1) \cdot s_0^2}{g_1} \right) = 1 - \varepsilon, \text{ где}$$

s_0^2 – несмещенная выборочная дисперсия,

g_1 и g_2 –

квантили распределения χ^2 с n степенями свободы уровня $\alpha = \frac{\varepsilon}{2}$ и $\alpha = 1 - \frac{\varepsilon}{2}$.

```
Математическое ожидание неизвестно:
Доверительный интервал для  $\sigma^2$  при малой выборке и при уровне доверия 0.8: (0.35, 0.95)
Доверительный интервал для  $\sigma^2$  при большой выборке и при уровне доверия 0.8: (0.88, 0.99)

Математическое ожидание равно  $a_0 = 4$ :
Доверительный интервал для  $\sigma^2$  при малой выборке и при уровне доверия 0.8: (0.33, 0.87)
Доверительный интервал для  $\sigma^2$  при малой выборке и при уровне доверия 0.8: (0.88, 0.99)
```

Рисунок 17 – Полученные доверительные интервалы

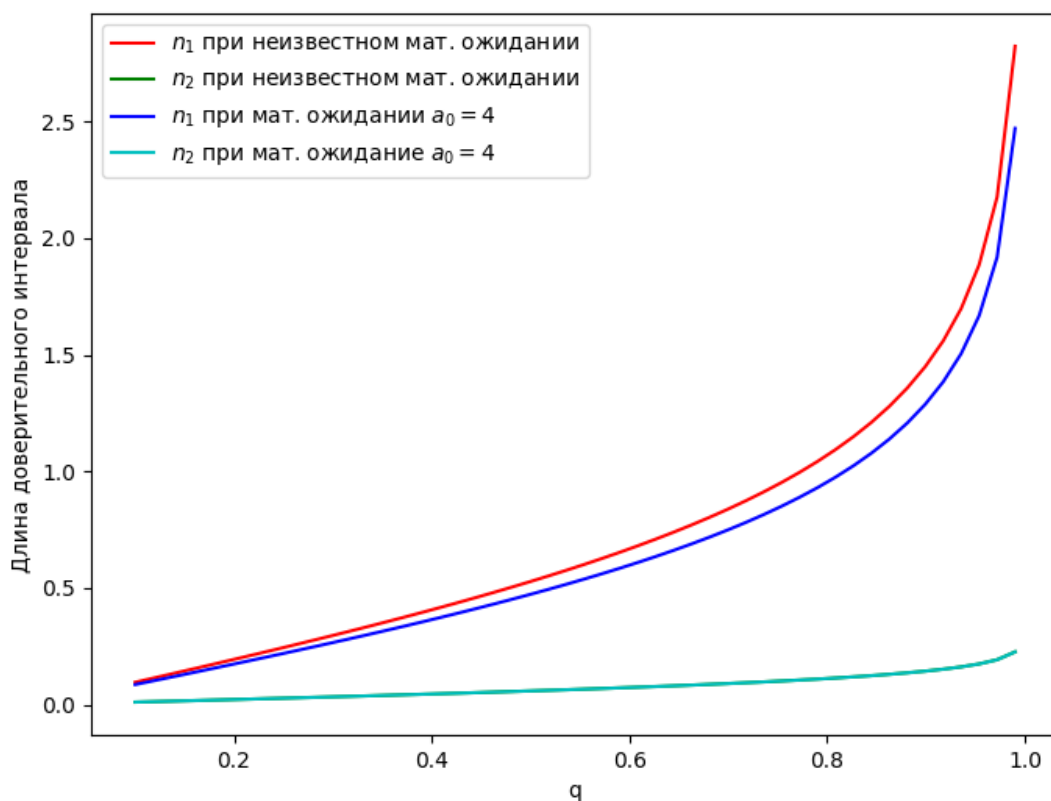


Рисунок 18 - Графики зависимости длины доверительного интервала от уровня доверия q

Длина доверительного интервала, характеризующая точность интервального оценивания, зависит от объема выборки n и уровня доверия: при увеличении объема выборки длина доверительного интервала уменьшается, а при приближении уровня доверия к единице – увеличивается. Также при известном значении a и в случае большой выборки, длина доверительного интервала стабильна независимо от уровня доверия.

Листинг программы

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

def trust_interval_with_unknown_mean(
    sample: np.array,
    q: float
) -> tuple[float, float]:

    alpha = 1 - q
    data = np.array(sample)
    n = len(sample)
```



```

sample_variance = np.var(data, ddof=1)

chi2_lower = stats.chi2.ppf(alpha / 2, df=n - 1)
chi2_upper = stats.chi2.ppf(1 - alpha / 2, df=n - 1)

lower_bound = (n - 1) * sample_variance / chi2_upper
upper_bound = (n - 1) * sample_variance / chi2_lower

return lower_bound, upper_bound

def trust_interval_with_known_a(
    sample: np.array,
    q: float
) -> tuple[float, float]:

    alpha = 1 - q
    sample_variance = np.var(sample, ddof=0)
    degrees_of_freedom = n = len(sample)

    chi2_lower = stats.chi2.ppf(alpha / 2, df=degrees_of_freedom)
    chi2_upper = stats.chi2.ppf(1 - alpha / 2, df=degrees_of_freedom)

    lower_bound = (n * sample_variance) / chi2_upper
    upper_bound = (n * sample_variance) / chi2_lower

    return lower_bound, upper_bound

if __name__ == '__main__':

    mean, sigma = 4, 1
    q = 0.8

    first_count = 15
    second_count = first_count * 70

    first_sample = np.random.normal(mean, sigma, first_count)
    second_sample = np.random.normal(mean, sigma, second_count)

    print('Математическое ожидание неизвестно:')

    lower_bound_SPA, upper_bound_SPA = trust_interval_with_unknown_mean(
        first_sample, q
    )
    print(
        "\tДоверительный интервал для  $\sigma^2$  при малой выборке и при уровне "
        f"доверия {q}: ({lower_bound_SPA:.2f}, {upper_bound_SPA:.2f})"
    )

    lower_bound_BPA, upper_bound_BPA = trust_interval_with_unknown_mean(
        second_sample, q
    )
    print(
        "\tДоверительный интервал для  $\sigma^2$  при большой выборке и при уровне "
        f"доверия {q}: ({lower_bound_BPA:.2f}, {upper_bound_BPA:.2f})"
    )

    print(f"\nМатематическое ожидание равно a_0 = {mean}:")
    lower_bound_SPB, upper_bound_SPB = trust_interval_with_known_a(
        first_sample, q
    )
    print(
        "\tДоверительный интервал для  $\sigma^2$  при малой выборке и при уровне

```

```

доверия "
    f"{q}: ({lower_bound_SPB:.2f}, {upper_bound_SPB:.2f})"
)

lower_bound_BPB, upper_bound_BPB = trust_interval_with_known_a(
    second_sample, q
)
print(
    "\tДоверительный интервал для  $\sigma^2$  при малой выборке и при уровне
доверия "
    f"{q}: ({lower_bound_BPB:.2f}, {upper_bound_BPB:.2f})"
)

new_q = np.linspace(0.1, 0.99, 50)
first_y = []
second_y = []
third_y = []
fourth_y = []

for i in range(50):
    left, right = trust_interval_with_unknown_mean(first_sample,
new_q[i])
    first_y.append(right - left)
    left, right = trust_interval_with_unknown_mean(second_sample,
new_q[i])
    second_y.append(right - left)
    left, right = trust_interval_with_known_a(first_sample, new_q[i])
    third_y.append(right - left)
    left, right = trust_interval_with_known_a(second_sample, new_q[i])
    fourth_y.append(right - left)

plt.figure(figsize=(8, 6))

plt.plot(
    new_q, first_y, 'r', label='$n_1$, мат. ожидание '
                                'неизвестно'
)
plt.plot(
    new_q, second_y, 'g', label='$n_2$, мат. ожидание '
                                'неизвестно'
)
plt.plot(
    new_q, third_y, 'b', label='$n_1$, мат. ожидание '
                                'известно ($a_0=4$)'
)
plt.plot(
    new_q, fourth_y, 'c', label='$n_2$, мат. ожидание '
                                'известно ($a_0=4$)'
)

plt.ylabel('Длина доверительного интервала')
plt.xlabel('q')
plt.legend()
plt.show()

```

Вывод: в ходе выполнения домашней работы были получены практические навыки обработки статистических данных.

РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Гмурман, В.Е. Теория вероятностей и математическая статистика: учеб. пособие для вузов/ В.Е. Гмурман. - М.: Юрайт, 2014. – 479 с. 21
2. Гринь, А.Г. Вероятность и статистика [Электронный ресурс]: учебное пособие/ А.Г. Гринь.— Омск: Омский государственный университет им. Ф.М. Достоевского, 2013.— 304 с.— Режим доступа: <http://www.iprbookshop.ru/24879.html>
3. Кельберт М.Я. Вероятность и статистика в примерах и задачах [Электронный ресурс]/ Кельберт М.Я. Сухов Ю.М.. - М.: МЦНМО, 2010. - Т. 1. Основные понятия теории вероятностей и математической статистики. - 486 с. - URL: [//biblioclub.ru/index.php?page=book&id=69109](http://biblioclub.ru/index.php?page=book&id=69109)