*Государственное образовательное учреждение высшего профессионального образования*

*«Московский государственный технический университет имени Н. Э. Баумана»*

ОТЧЕТ

По лабораторной работе №2

По курсу «Математическая статистика»

Тема: «Оценка математического ожидания и дисперсии»

Вариант 21

|  |  |
| --- | --- |
| Студент: | Янова Д.Ю. |
| Группа: | ИУ7-63 |
| Преподаватель: | Саркисян П. С. |

Москва, 2019

**1. Определения**

**γ-доверительный интервал** – пара статистик (называемых нижней и верхней границами) таких, что для неизвестного параметра . Другими словами, γ-доверительный интервал – интервал, который покрывает теоретическое значение параметра с вероятностью .

**2. Формулы вычисления**

При вычислении границ γ-доверительного интервала для параметров нормальной случайной величины используются три центральных статистики:

|  |  |  |
| --- | --- | --- |
| параметры | центральная статистика: | границы: |
| – неизв., – изв.; оценить |  | *,* |
| – неизв., – неизв., оценить |  | *,* |
| – неизв., – неизв., оценить |  |  |
| – изв., – неизв.; оценить |

где – квантили уровня нормального распределения, распределения Стьюдента и распределения хи-квадрат соответственно.

**3. Листинг программы**

function Lab2()

sample = importdata('data.txt');

[exp, s2] = CalcExpDisp(sample);

fprintf('mat. expectation: %.6f\ndispersion: %.6f\n', exp, s2);

gamma = input('Input gamma: ');

if isempty(gamma) gamma = 0.9; disp(gamma); end

N = input('Input N: ');

if isempty(N) N = length(sample); disp(N); end

[lowM, highM] = CalcBordersExp(exp, s2, gamma, N);

[lowD, highD] = CalcBordersDisp(s2, gamma, N);

fprintf('mat.exp. borders: (%.6f .. %.6f)\n', lowM, highM);

fprintf('dispersion borders: (%.6f .. %.6f)\n', lowD, highD);

figure(1);

hold on;

PlotMathExps(sample, gamma, N);

figure(2);

hold on;

PlotDispersions(sample, gamma, N);

end

function [exp, s2] = CalcExpDisp(sample)

%% вычисление точечных оценок математического ожидания и дисперсии

n = length(sample);

exp = sum(sample) / n;

if n > 1

s2 = sum((sample - exp).^2) / (n-1);

else

s2 = 0;

end

end

function [lowM, highM] = CalcBordersExp(exp, s2, gamma, N)

%% вычисление нижней и верхней границ матожидания

%неизвестны матожидание и дисперсия, оцениваем матожидание;

%статистика ~St(n-1): P{|(m - mu^)/sqrt(s2)\*sqrt(n)| < q\_alpha} = gamma

alpha = 1 - (1 - gamma) / 2;

quantile = tinv(alpha, N-1);

border = quantile \* sqrt(s2) / sqrt(N);

lowM = exp - border;

highM = exp + border;

end

function [lowD, highD] = CalcBordersDisp(s2, gamma, N)

%% вычисление нижней и верхней границ дисперсии

%неизвестны матожидание и дисперсия, оцениваем дисперсию;

%статистика ~chi2(n-1): P{ q\_low < s2\*(n-1)/disp < q\_high } = gamma

low = (1 - gamma) / 2;

a\_quantile = chi2inv(low, N-1);

highD = s2\*(N-1) / a\_quantile;

high = 1 - low;

a\_quantile = chi2inv(high, N-1);

lowD = s2\*(N-1) / a\_quantile;

end

function PlotMathExps(sample, gamma, N)

%% на координатной плоскости Oyn построить прямую y=mu^(x\_N), а также

%графики функций mu^(x\_n), mu\_down(x\_n), mu\_up(x\_n) как функций от объема n

%выборки, где n изменяется от 1 до N

%определяем матожидания и дисперсии для разных n

mu = zeros(N,1);

s2 = zeros(N,1);

for i = 1:N

part = sample(1:i);

[mu(i), s2(i)] = CalcExpDisp(part);

end

%заполняем массив значений для прямой

mu\_line = zeros(N,1);

mu\_line(1:N) = mu(N);

%заполняем массивы значений для границ

mu\_down = zeros(N,1);

mu\_up = zeros(N,1);

for i = 1:N

[mu\_down(i), mu\_up(i)] = CalcBordersExp(mu(i), s2(i), gamma, i);

end

plot((1:N), mu\_line, 'g');

plot((1:N), mu, 'k');

plot((1:N), mu\_up, 'b');

plot((1:N), mu\_down, 'r');

grid on;

xlabel('n');

ylabel('\mu');

legend('\mu\^(x\_N)', '\mu\^(x\_n)', '\mu^{up}(x\_n)', '\mu\_{down}(x\_n)');

end

function PlotDispersions(sample, gamma, N)

%% на координатной плоскости Ozn построить прямую y=S2(x\_N), а также

%графики функций S2(x\_n), sigma\_down(x\_n), sigma\_up(x\_n) как функций от

%объема n выборки, где n изменяется от 1 до N

%на малых n дисперсия прыгает до 300, мелкие значения не разглядеть

start = 5;

%определяем матожидания и дисперсии для разных n

mu = zeros(N,1);

s2 = zeros(N,1);

for i = start:N

part = sample(1:i);

[mu(i), s2(i)] = CalcExpDisp(part);

end

%заполняем массив значений для прямой

s2\_line = zeros(N,1);

s2\_line(1:N) = s2(N);

%заполняем массивы значений для границ

sigma\_down = zeros(N,1);

sigma\_up = zeros(N,1);

for i = start:N

[sigma\_down(i), sigma\_up(i)] = CalcBordersDisp(s2(i), gamma, i);

end

nvalues = (start:N);

plot(nvalues, s2\_line(nvalues), 'g');

plot(nvalues, s2(nvalues), 'k');

plot(nvalues, sigma\_up(nvalues), 'b');

plot(nvalues, sigma\_down(nvalues), 'r');

grid on;

xlabel('n');

ylabel('\sigma');

legend('S^2(x\_N)', 'S^2(x\_n)', '\sigma^{up}(x\_n)', '\sigma\_{down}(x\_n)');

end

**4. Результаты работы для индивидуальной выборки**

Lab\_2

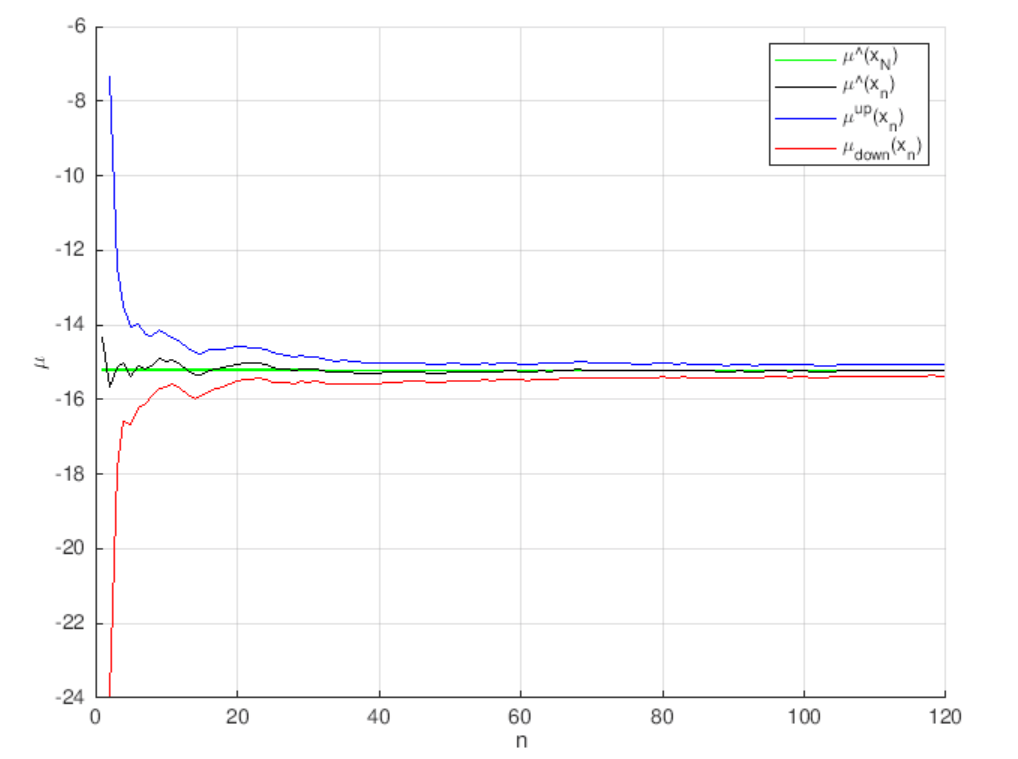
mat. expectation: -15.220917  
dispersion: 0.968029  
Input gamma:

0.9

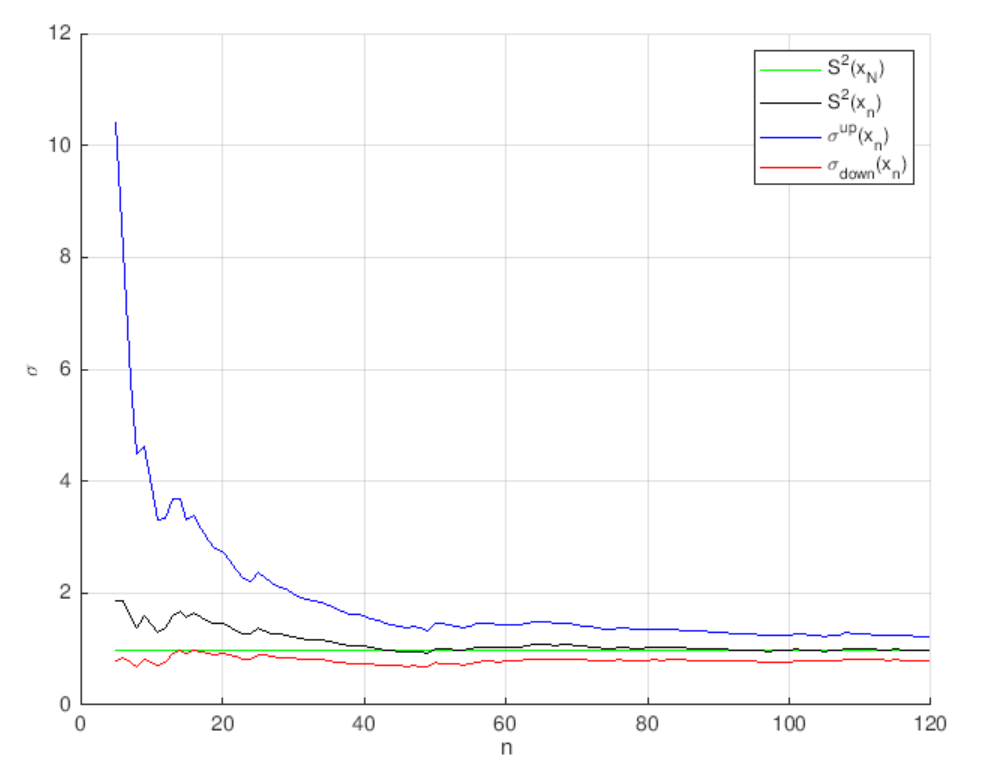
Input N:

120

mat.exp. borders: (-15.369810 .. -15.072023)  
dispersion borders: (0.791935 .. 1.214997)



**Рис. 1: Графики для математического ожидания**



**Рис.2: Графики для дисперсии**