|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

**ОТЧЕТ**

*к лабораторной работе №2*

*По курсу: «Математическая статистика»*

*На тему: «Интервальные оценки»*

Студентка ИУ7-65Б

Оберган Т.М

14 вариант

Преподаватели

Власов П.А.

Волков И.К.

*Москва, 2020 г.*

# Введение

**Цель** **работы:** построение доверительных интервалов для математичского ожидания и дисперсии нормальной случайной величины.

## Содержание работы:

1. Для выборки объема *n* из нормальной генеральной совокупности *X* реализовать в виде программы на ЭВМ

a. вычисление точечных оценок и математического ожидания *MX* и дисперсии *DX* соответственно;

b. вычисление нижней и верхней границ , для *γ*-доверительного интервала для математического ожидания *MX*;

c. вычисление нижней и верхней границ для *γ*-доверительного интервала для математического ожидания *DX*;

2. вычислить *μ* и *S*2 для выборки из индивидуального варианта;

3. для заданного пользователем уровня доверия *γ* и *N* – объема выборки из индивидуального варианта:

a. на координатной плоскости построить прямую *y* = *μ*( *xN* ), также графики функций *y* = *μ*(*xn*), *y* = *μ*(*xn*) и *y* = *μ*(*xn*) как функций объема *n* выборки, где *n* изменяется от 1 до *N*;

b. на другой координатной плоскости *Ozn* построить прямую *z* = *S*2( *xN* ), также графики функций *z* = *S*2(*xn*), *z* = *σ*2(*xn*) и   
*z* = *σ*2(*xn*) как функций объема *n* выборки, где *n* изменяется от 1 до *N*.

# Теоретическая часть

В данной части рассмотрены формулы для вычисления величин, границ *γ*-доверительного интервала, а также определение *γ*-доверительного интервала.

## 1.1 Определения

Интервальной оценкой с коэффициентом доверия *γ* (*γ*-доверительной

интервальной оценкой) параметра *θ* называют пару статистик *θ*() и *θ*() таких, что

(1)

## 1.2 Формулы для вычисления величин

**Оценка математического ожидания:**

**Несмещённая оценка дисперсии:**

**Выборочная дисперсия:**

## 1.3 Формулы для вычисления границ *γ*-доверительного интервала

Пусть *Xn* — случайная выборка объема *n* из генеральной совокупности *X*, распределенной по нормальному закону с параметрами *μ* и *σ*2.

**Оценка** **для** **математического** **ожидания** **при** **известной** **дисперсии**

Где *Xn* - оценка математического ожидания, n – объем выборки

*S*(*Xn*) — точечная оценка дисперсии случайной выборки *Xn*,

*hα*(*n* −1) — квантиль уровня *α* распределения Стьюдента с *n* − 1 степенями свободы.

*t*

**Оценка для дисперсии**

Где n — объем выборки,*S*(*Xn*) — точечная оценка дисперсии случайной выборки *Xn*,

*χα*(*n* − 1) — квантиль уровня *α* для распределения *χ*2 с *n*− 1 степями свободы.

2

# Листинг

**function** lab2()

X = [3.38,1.21,1.85,2.24,4.17,2.99,4.81,2.71,2.70,4.41,3.21,3.15,2.77,4.05,3.89,1.56,2.78,2.04,2.82,3.28,2.63,1.89,3.57,3.15,3.80,5.40,3.25,2.04,2.61,5.06,2.87,2.66,4.80,3.86,0.09,2.45,2.40,2.14,1.69,2.36,5.44,2.77,1.94,2.55,3.97,1.88,3.01,4.21,4.74,2.02,2.38,2.46,3.51,2.89,1.57,3.53,0.77,3.31,3.58,2.77,3.61,3.71,2.38,3.06,4.29,4.76,1.69,1.59,3.21,2.74,3.99,3.53,3.52,2.84,1.21,2.82,4.34,3.65,2.22,2.87,3.14,3.58,1.96,3.41,3.85,1.96,3.02,4.22,3.10,2.68,3.67,1.70,5.47,5.02,2.52,3.09,2.19,4.44,2.33,2.27,3.34,3.05,4.35,3.58,3.43,4.49,3.57,3.20,1.53,3.53,3.53,1.27,3.40,4.53,2.21,3.28,3.50,2.01,3.30,1.86];

N = length(X);

gamma = 0.9;

alpha = (1 - gamma) / 2;

mu = get\_mu(X);

Ssqr = get\_Ssqr(X);

array\_of\_mu = get\_array\_of\_mu(X, N);

array\_of\_var = get\_array\_of\_var(X, N);

mu\_higher = get\_mu\_higher(array\_of\_mu, array\_of\_var, alpha, N);

mu\_lower = get\_mu\_lower(array\_of\_mu, array\_of\_var, alpha, N);

sigma\_higher = get\_sigma\_higher(array\_of\_var, alpha, N);

sigma\_lower = get\_sigma\_lower(array\_of\_var, alpha, N);

fprintf('mu = %.3f\n', mu);

fprintf('S^2 = %.3f\n', Ssqr);

fprintf('mu\_lower = %.3f\n', mu\_lower(**end**));

fprintf('mu\_higher = %.3f\n', mu\_higher(**end**));

fprintf('sigma^2\_lower = %.3f\n', sigma\_lower(**end**));

fprintf('sigma^2\_higher = %.3f\n', sigma\_higher(**end**));

figure

hold on;

plot([1, N], [mu, mu], 'g');

plot((1 : N), array\_of\_mu, 'r');

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

plot((1 : N), mu\_higher, 'm');

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

grid on;

hold off;

figure

hold on;

plot([1, N], [Ssqr, Ssqr], 'g');

plot((1 : N), array\_of\_var, 'r');

plot((1 : N), sigma\_lower, 'b');

plot((4 : N), sigma\_higher(4 : length(sigma\_higher)), 'm');

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

grid on;

hold off;

**function** mu = get\_mu(X)

mu = sum(X)/size(X, 2);

**end**

**function** Ssqr = get\_Ssqr(X)

n = size(X, 2);

Ssqr = n / (n - 1) \* get\_sigma\_sqr(X);

**end**

**function** array\_of\_mu = get\_array\_of\_mu(X, N)

array\_of\_mu = zeros(1, N);

**for** i = 1 : N

array\_of\_mu(i) = get\_mu(X(1 : i));

**end**

**end**

**function** array\_of\_var = get\_array\_of\_var(X, N)

array\_of\_var = zeros(1, N);

**for** i = 1 : N

array\_of\_var(i) = get\_Ssqr(X(1 : i));

**end**

**end**

**function** mu\_higher = get\_mu\_higher(array\_of\_mu, array\_of\_var, alpha, N)

mu\_higher = zeros(1, N);

**for** i = 1 : N

mu\_higher(i) = array\_of\_mu(i) + sqrt(array\_of\_var(i) ./ i) .\* tinv(1 - alpha, i - 1);

**end**

**end**

**function** mu\_lower = get\_mu\_lower(array\_of\_mu, array\_of\_var, alpha, N)

mu\_lower = zeros(1, N);

**for** i = 1 : N

mu\_lower(i) = array\_of\_mu(i) + sqrt(array\_of\_var(i) ./ i) .\* tinv(alpha, i - 1);

**end**

**end**

**function** sigma\_higher = get\_sigma\_higher(array\_of\_var, alpha, N)

sigma\_higher = zeros(1, N);

**for** i = 1:N

sigma\_higher(i) = array\_of\_var(i) .\* (i - 1) ./ chi2inv(alpha, i - 1);

**end**

**end**

**function** sigma\_lower = get\_sigma\_lower(array\_of\_var, alpha, N)

sigma\_lower = zeros(1, N);

**for** i = 1 : N

sigma\_lower(i) = array\_of\_var(i) .\* (i - 1) ./ chi2inv(1 - alpha, i - 1);

**end**

**end**

**function** sigma = get\_sigma\_sqr(X)

tempMu = get\_mu(X);

sigma = sum((X - tempMu) .\* (X - tempMu)) / size(X,2);

**end**

**end**

# Результат работы программы

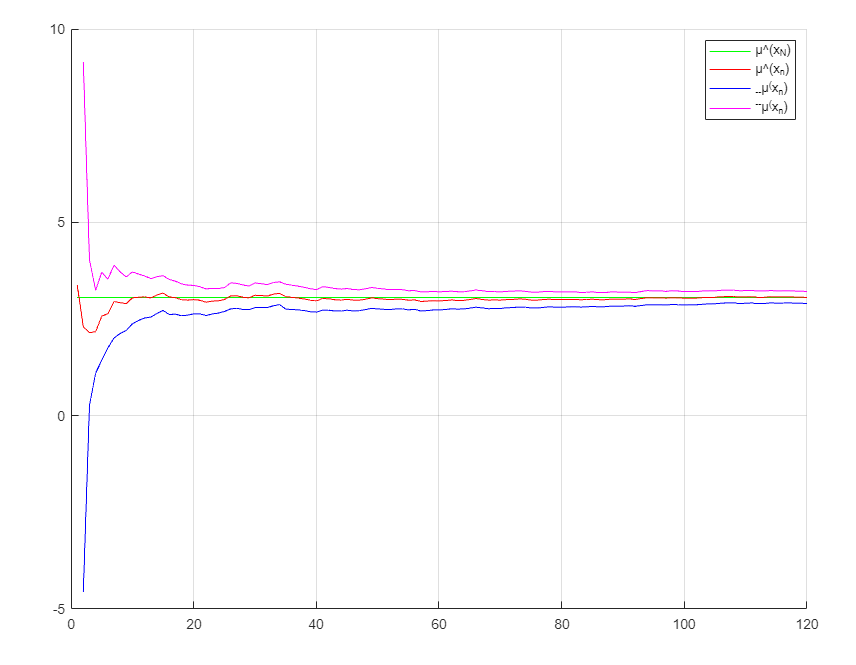
**

Рис. 1 - оценка для математического ожидания

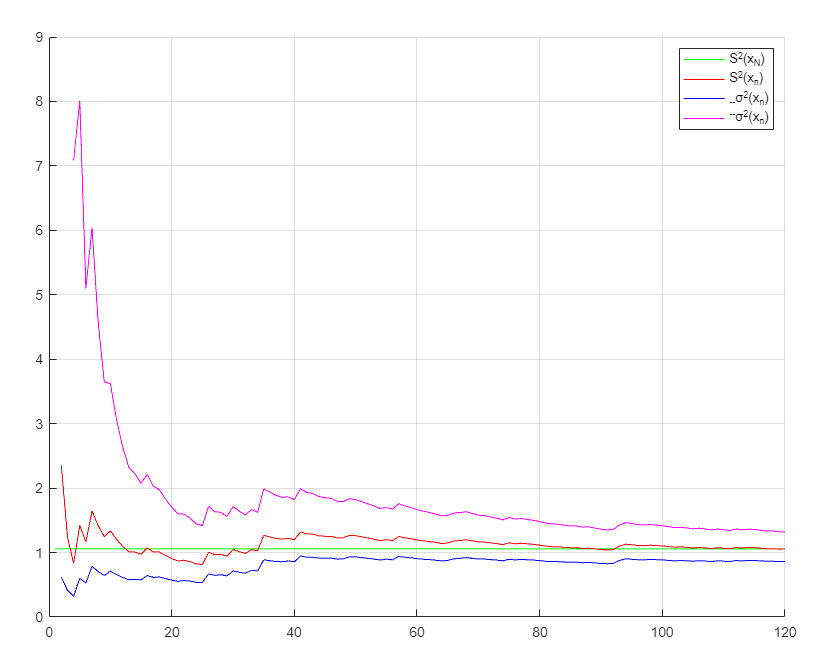
**

Рис.2 – оценка для дисперсии