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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ «09.03.04 Программная инженерия»

ОТЧЕТ
ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Название: Интервальные оценки

Дисциплина: Математическая статистика

Студент ИУ7-64Б ____________ С. Д. Параскун

Группа — Подпись, дата — И. О. Фамилия

Преподаватель ____________ М. А. Велищанский

Подпись, дата — И. О. Фамилия

Москва, 2022 г.

1. Задание на лабораторную

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

Содержание работы:

1. Для выборки объема n из генеральной совокупности X реализовать в виде программы на ЭВМ:
   - вычисление точечных оценок $\hat{\mu}(\vec{x}_n)$ и $S^2(\vec{x}_n)$ математического ожидания MX и дисперсии DX соответственно;
   - вычисление нижней и верхней границ $\underline{\mu}(\vec{x}_n)$, $\overline{\mu}(\vec{x}_n)$ для γ-доверительного интервала для математического ожидания MX;
   - вычисление нижней и верхней границ $\underline{\sigma}^2(\vec{x}_n)$, $\overline{\sigma}^2(\vec{x}_n)$ для γ-доверительного интервала для дисперсии DX;
2. Вычислить $\hat{\mu}$ и S^2 для выборки из индивидуального варианта;
3. Для заданного пользователем уровня доверия γ и N – объема выборки из индивидуального варианта:
   - на координатной плоскости Oyn построить прямую $y = \hat{\mu}(\vec{x}_N)$, также графики функций $y = \hat{\mu}(\vec{x}_n)$, $y = \underline{\mu}(\vec{x}_n)$, $y = \overline{\mu}(\vec{x}_n)$ как функций объема n выборки, где n изменяется от 1 до N;
   - на другой координатной плоскости Ozn построить прямую $z = S^2(\vec{x}_N)$, также графики функций $z = S^2(\vec{x}_n)$, $z = \underline{\sigma}^2(\vec{x}_n)$, $z = \overline{\sigma}^2(\vec{x}_n)$ как функций объема n выборки, где n изменяется от 1 до N.

В работе использовалась выборка по 13 варианту.

2. γ-доверительный интервал

2.1 Определение

Пусть дана случайная величина X, закон распределения которой известен с точностью до неизвестного параметра θ, и для параметра θ построен интервал $(\underline{\theta}(\vec{X}_n), \overline{\theta}(\vec{X}_n))$, где $\underline{\theta}(\vec{x}_n)$ и $\overline{\theta}(\vec{X}_n)$ – пара статистик случайной выборки $\vec{X}_n$, такой что:

$$P\left\{\underline{\theta}(\vec{X}_n) < \theta < \overline{\theta}(\vec{X}_n)\right\} = \gamma. \tag{2.1}$$

Тогда $(\underline{\theta}(\vec{X}_n), \overline{\theta}(\vec{X}_n))$ называется интервальной оценкой параметра θ с коэффициентом доверия γ (γ-доверительный интервал).

Для удобства также можно ввести величину $\alpha = \dfrac{1-\gamma}{2}$ – вероятность отклонения результата в определенном направлении.

2.2 Формулы для вычисления границ γ-доверительного интервала

Пусть $\vec{X}_n$ – случайная выборка объема n из генеральной совокупности X, распределенной по нормальному закону с параметрами μ и σ^2.

1. Для мат. ожидания:

$$\underline{\mu}(\vec{X}_n) = \overline{X} + \frac{S(\vec{X}_n)}{\sqrt{n}} t_{\alpha}(n-1), \tag{2.2}$$

$$\overline{\mu}(\vec{X}_n) = \overline{X} + \frac{S(\vec{X}_n)}{\sqrt{n}} t_{1-\alpha}(n-1), \tag{2.3}$$

   где $\overline{X}$ – точечная оценка мат. ожидания, $S(\vec{X}_n)$ – точечная оценка дисперсии, $t_{1-\alpha}(n-1)$ – квантиль уровня α распределения Стьюдента со степенями свободы $n-1$;

2. Для дисперсии:

$$\underline{\sigma}^2(\vec{X}_n) = \frac{(n-1) \cdot S^2(\vec{X}_n)}{\chi^2_{1-\alpha}(n-1)}, \tag{2.4}$$

$$\overline{\sigma}^2(\vec{X}_n) = \frac{(n-1) \cdot S^2(\vec{X}_n)}{\chi^2_{\alpha}(n-1)}, \tag{2.5}$$

где $S(\vec{X}_n)$ – точечная оценка дисперсии, $\chi^2_\alpha(n-1)$ – квантиль уровня α для распределения χ^2 с $n-1$ степенями свободы.

3. Код программы

```
function main()
    f = fopen("X.txt", "r");
    buf = textscan(f, "%f", 'Delimiter', ' ');
    X = buf{1,1};
    X = X';
    fclose(f);

    gamma = 0.9;
    alpha = (1 - gamma) / 2;

    MU = MX(X);
    S2 = DX(X);
    fprintf("MU = %f, S2 = %f\n", MU, S2);

    MU_low = get_low_MU(X, alpha);
    MU_high = get_high_MU(X, alpha);
    fprintf("MU_low = %f, MU_high = %f\n", MU_low, MU_high);

    S2_low = get_low_S2(X, alpha);
    S2_high = get_high_S2(X, alpha);
    fprintf("S2_low = %f, S2_high = %f\n", S2_low, S2_high);

    gamma = 0.9;
    alpha = (1 - gamma) / 2;
    N = length(X);

    figure;
    hold on;
    plot([1, N], [MU, MU], 'r');
    plot((1 : N), get_MU_array(X, N), 'g');
    plot((1 : N), get_MU_low_array(X, N, alpha), 'b');
    plot((1 : N), get_MU_high_array(X, N, alpha), 'm');
    l = legend('\mu\^(x_N)', '\mu\^(x_n)', '_{--}\mu\^(x_n)',
        '^{--}\mu\^(x_n)');
    set(l, 'fontsize', 14);
    grid on;

    figure;
    hold on;
    plot([2, N], [S2, S2], 'r');
    plot((1 : N), get_S2_array(X, N), 'g');
    plot((1 : N), get_S2_low_array(X, N, alpha), 'b');
    S2_high_array = get_S2_high_array(X, N, alpha);
    plot((4 : N), S2_high_array(4 : N), 'm');
    l = legend('S^2(x_N)', 'S^2(x_n)', '_{--}\sigma^2(x_n)',
```

```
        '^{--}\sigma^2(x_n)');
    set(1, 'fontsize', 14);
    grid on;
end

function mx = MX(X)
    n = length(X);
    sum = sum(X);
    mx = sum / n;
endfunction

function dx = DX(X)
    n = length(X);
    MX = MX(X);
    if (n == 1)
        dx = 0;
    else
        dx = sum((X - MX).^2) / (n - 1);
    endif
endfunction

function mu_low = get_low_MU(X, alpha)
    n = length(X);
    MU = MX(X);
    S = sqrt(DX(X));
    mu_low = MU + S / sqrt(n) * tinv(alpha, n - 1);
endfunction

function mu_high = get_high_MU(X, alpha)
    n = length(X);
    MU = MX(X);
    S = sqrt(DX(X));
    mu_high = MU + S / sqrt(n) * tinv(1 - alpha, n - 1);
endfunction

function s2_low = get_low_S2(X, alpha)
    n = length(X);
    S2 = DX(X);
    s2_low = (n - 1) * S2 / chi2inv(1 - alpha, n -1);
endfunction

function s2_high = get_high_S2(X, alpha)
    n = length(X);
    S2 = DX(X);
    s2_high = (n - 1) * S2 / chi2inv(alpha, n -1);
endfunction

function MU_array = get_MU_array(X, N)
```

```
    MU_array = zeros(1, N);
    for i = 1 : N
        MU_array(i) = MX(X(1 : i));
    endfor
endfunction

function MU_low_array = get_MU_low_array(X, N, alpha)
    MU_low_array = zeros(1, N);
    for i = 1 : N
        cur_X = X(1 : i);
        MU_low_array(i) = get_low_MU(cur_X, alpha);
    endfor
endfunction

function MU_high_array = get_MU_high_array(X, N, alpha)
    MU_high_array = zeros(1, N);
    for i = 1 : N
        cur_X = X(1 : i);
        MU_high_array(i) = get_high_MU(cur_X, alpha);
    endfor
endfunction

function S2_array = get_S2_array(X, N)
    S2_array = zeros(1, N);
    for i = 1 : N
        S2_array(i) = DX(X(1 : i));
    endfor
endfunction

function S2_low_array = get_S2_low_array(X, N, alpha)
    S2_low_array = zeros(1, N);
    for i = 1 : N
        cur_X = X(1 : i);
        S2_low_array(i) = get_low_S2(cur_X, alpha);
    endfor
endfunction

function S2_high_array = get_S2_high_array(X, N, alpha)
    S2_high_array = zeros(1, N);
    for i = 1 : N
        cur_X = X(1 : i);
        S2_high_array(i) = get_high_S2(cur_X, alpha);
    endfor
endfunction
```

4. Результаты расчетов для выборки по варианту

При вычислении результатов параметр $\gamma = 0.9$.

$\hat{\mu}(\vec{x}_n)$	$\underline{\mu}(\vec{x}_n)$	$\overline{\mu}(\vec{x}_n)$
-10.131750	-10.270946	-9.992554

$S^2(\vec{x}_n)$	$\underline{S}^2(\vec{x}_n)$	$\overline{S}^2(\vec{x}_n)$
0.846041	0.692138	1.061888

При построении графиков функций $n = 1..N$, причем $N = 120$ – объем исходной выборки.

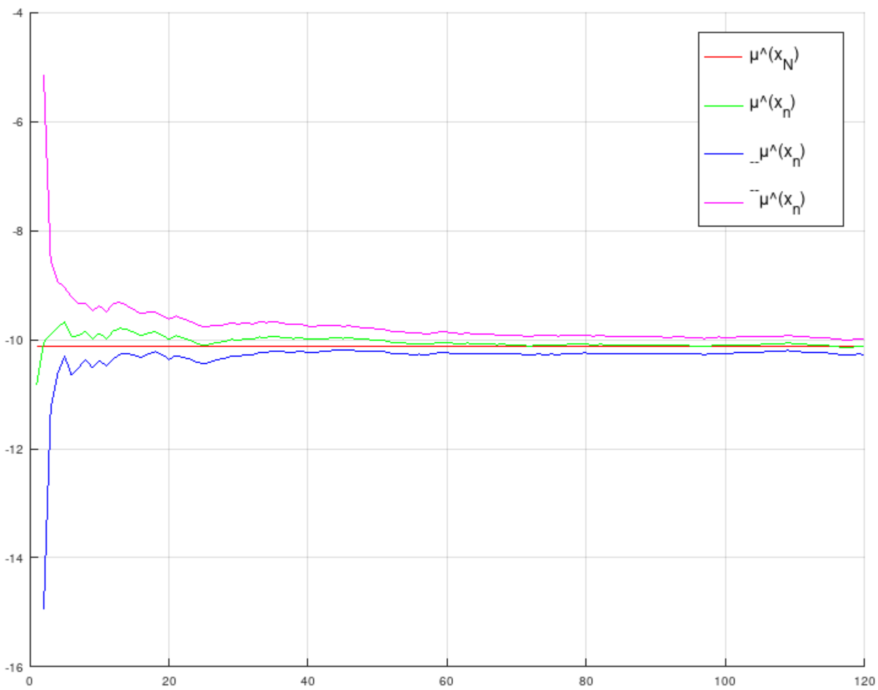

Рисунок 4.1 – Оценка для мат. ожидания

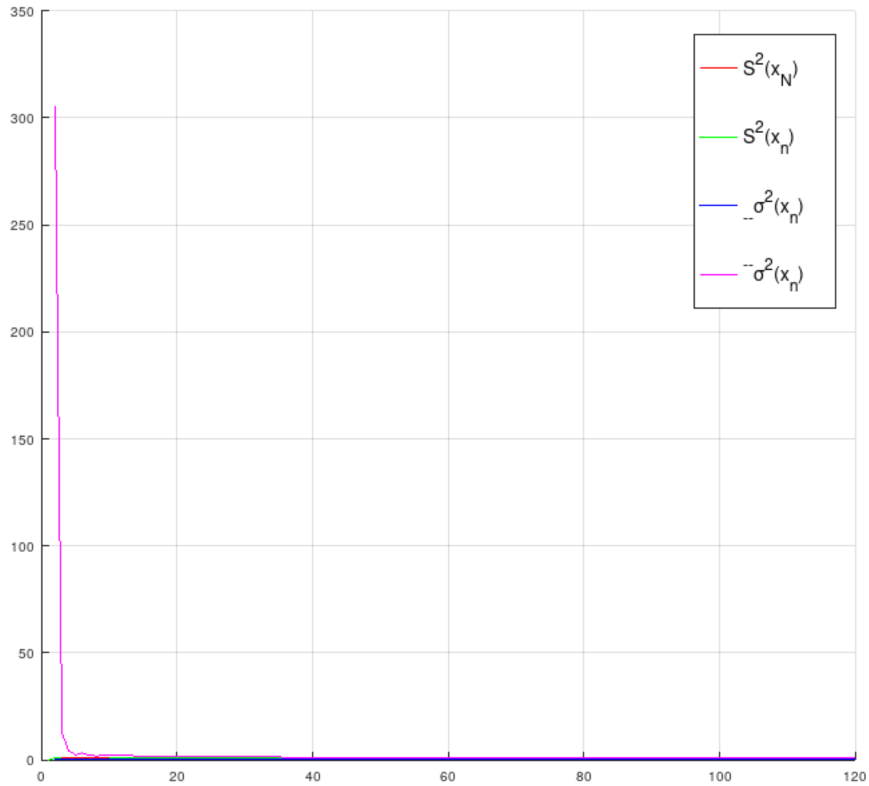

Рисунок 4.2 – Оценка для дисперсии

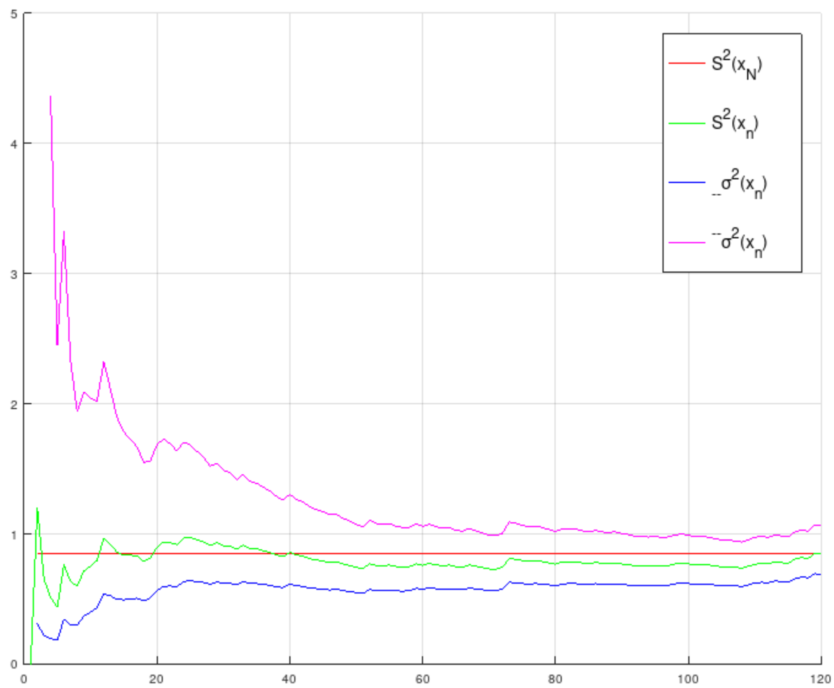

Рисунок 4.3 – Оценка для дисперсии (график $\overline{\sigma}^2(\vec{x}_n)$ начинается с $n = 4$)