commentstyle

Министерство образования Российской Федерации
Московский государственный технический
университет им. Н.Э. Баумана

Отчет по лабораторной работе №2
по курсу «Математическая статистика»

Тема: «Интервальные оценки»

Вариант 5

Студент: **Горохова И.Б.**
Группа: **ИУ7-61**

Преподаватель: **Власов П.А.**

Москва, 2018

Содержание

Определения

Доверительный интервал уровня γ для параметра θ - пара статистик $\underline{\theta}(\vec{X})$ и $\overline{\theta}(\vec{X})$ таких, что $P\{\theta \in (\underline{\theta}(\vec{X}), \overline{\theta}(\vec{X}))\} = \gamma$

Формулы для вычисления

Для вычисления границ γ-доверительного интервала для параметров нормальной случайной величины используют три центральные статистики:

<table>
<tr><th>Параметры:</th><th>Оценить:</th><th>Центральная статистика:</th><th>Границы:</th></tr>
<tr><td>μ - неизвестно,
σ - известно</td><td>μ</td><td>$\frac{\mu-\overline{X}}{\sigma}\sqrt{n} \sim N(0,1)$</td><td>$\underline{\mu} = \overline{X} - \frac{u_{1-\alpha}\sigma}{\sqrt{n}}$
$\overline{\mu} = \overline{X} + \frac{u_{1-\alpha}\sigma}{\sqrt{n}}$</td></tr>
<tr><td>μ - неизвестно,
σ - неизвестно</td><td>μ</td><td>$\frac{\mu-\overline{X}}{S(\vec{X_n})}\sqrt{n} \sim St(n-1)$</td><td>$\underline{\mu} = \overline{X} - \frac{t_{1-\alpha}S(\vec{X_n})}{\sqrt{n}}$
$\overline{\mu} = \overline{X} + \frac{t_{1-\alpha}S(\vec{x_n})}{\sqrt{n}}$</td></tr>
<tr><td>μ - известно,
σ - неизвестно</td><td>σ</td><td rowspan="2">$\frac{S^2(\vec{X_n})}{\sigma^2}(n-1) \sim \chi^2(n-1)$</td><td rowspan="2">$\underline{\sigma} = \frac{S^2(\vec{X_n})(n-1)}{h_{1-\alpha}}$
$\overline{\sigma} = \frac{S^2(\vec{X_n})(n-1)}{h_{\alpha}}$</td></tr>
<tr><td>μ -неизвестно,
σ -неизвестно</td><td>σ</td></tr>
</table>

где $\alpha = \frac{1-\gamma}{2}$;

$u_\alpha, t_\alpha, h_\alpha$ - квантили уровня α нормального распределения, распределения Стьюдента и распределения Хи-квадрат соответственно

Текст программы

Листинг 1: Функция Main

```
function main()
    sample = importdata('data(var5).txt');
    N = length(sample);

    [mu] = GetMuExpectedValue(sample);
    [s2] = GetS2DispersionValue(sample);

    fprintf("mu: %.4f\n",mu);
    fprintf("s2: %.4f\n",s2);

    gam = 0.9;
    [lM, hM] = GetBordersForExpectedValue(gam, s2, mu, N);
    fprintf("MX borders with gamma=%.2f: (%.4f .. %.4f)\n",
        gam, lM, hM);

    [lD, hD] = GetBordersForDispersionValue(gam, s2, N);
    fprintf("DX borders with gamma=%.2f: (%.4f .. %.4f)\n",
        gam, lD, hD);

    figure(1);
    hold on;
    GetGraphMX(sample, N, gam);

    figure(2);
    hold on;
    GetGraphDX(sample, N, gam);
end
```

Листинг 2: Точечные оценки MX и DX

```
function [Mu] = GetMuExpectedValue(sample)
    n = length(sample);
    Mu = sum(sample)/n;
end

function [s2] = GetS2DispersionValue(sample)
    n = length(sample);
    m = GetMuExpectedValue(sample);
    if n > 1
        s2 = sum((sample-m).^2)/(n-1);
```

```
    else
        s2 = 0;
    end
end
```

Листинг 3: Границы γ-доверительного интервала для MX

```
function [lM, hM] = GetBordersForExpectedValue(gam, s2, mu,
    n)
    alpha = (1 + gam)/2;
    quantile = tinv(alpha, n-1);
    border = quantile * sqrt(s2) / sqrt(n);

    lM = mu - border;
    hM = mu + border;
end
```

Листинг 4: Границы γ-доверительного интервала для DX

```
function [lD, hD] =  GetBordersForDispersionValue(gam, s2,
   n)
    alpha1 = (1 - gam)/2;
    alpha2 = (1 + gam)/2;
    quantile1 = chi2inv(alpha1, n-1);
    quantile2 = chi2inv(alpha2, n-1);

    lD = s2*(n-1)/quantile2;
    hD = s2*(n-1)/quantile1;
end
```

Листинг 5: Графики для оценок MX

```
function GetGraphMX(sample, n, gam)
    mu = zeros(n,1);
    s2 = zeros(n,1);
    lMu = zeros(n,1);
    hMu = zeros(n,1);
    line = zeros(n,1);

    line(1:n) = mu(n);
    for i = 1:n
        part = sample(1:i);
        [mu(i)] = GetMuExpectedValue(part);
        [s2(i)] = GetS2DispersionValue(part);
```

```
        [lMu(i), hMu(i)] = GetBordersForExpectedValue(gam,
            s2(i), mu(i), i);
    end

    plot(line, 'g');
    plot(lMu, 'r');
    plot(hMu, 'b');
    plot(mu, 'k');
    grid on;
    xlabel('n');
    ylabel('\mu');
    legend('\mu\^(x_N)','_{--}\mu^(x_n)', '^{--}\mu^(x_n)',
        '\mu\^(x_n)');
end
```

Листинг 6: Графики для оценок DX

```
function GetGraphDX(sample, n, gam)
    s2 = zeros(n,1);
    lSigma = zeros(n,1);
    hSigma = zeros(n,1);
    line = zeros(n,1);
    startl = 3;

    line(startl:n) = s2(n);
    for i = startl:n
        part = sample(1:i);
        [s2(i)] = GetS2DispersionValue(part);
        [lSigma(i), hSigma(i)] =
            GetBordersForDispersionValue(gam, s2(i), i);
    end

    plot((startl:n), line(startl:n), 'g');
    plot((startl:n), lSigma(startl:n), 'r');
    plot((startl:n), hSigma(startl:n), 'b');
    plot((startl:n), s2(startl:n), 'k');
    grid on;
    xlabel('n');
    ylabel('\sigma');
    legend('S^2(x_N)','_{--}\sigma^2(x_n)', '^{--}\sigma^2(
      x_n)', 'S^2(x_n)');
end
```

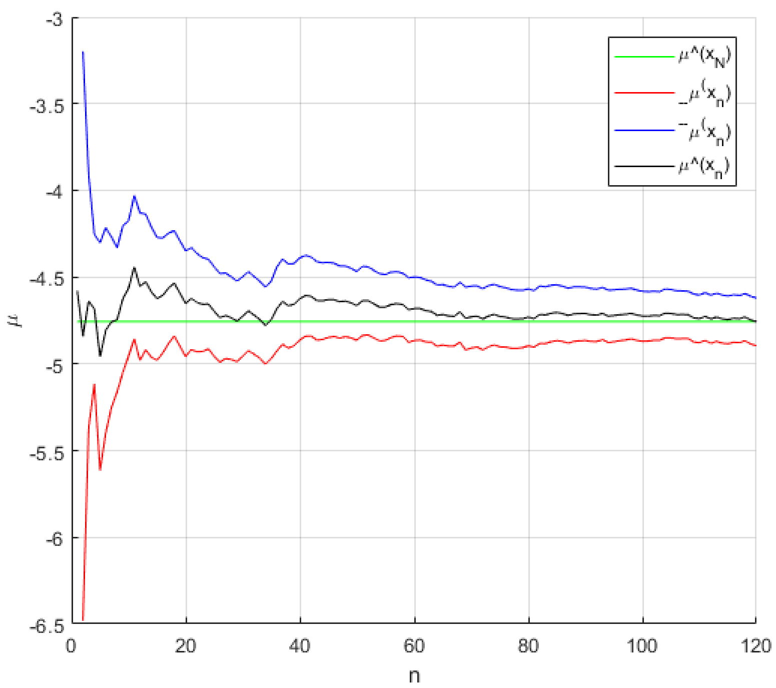
Результаты расчетов и графики

```
>> MSlab2
mu: -4.7579
s2: 0.8115
MX borders with gamma=0.90: (-4.8942 .. -4.6216)
DX borders with gamma=0.90: (0.6639 .. 1.0185)
```

$\hat{\mu}(\vec{x_n})$	-4.7579
$S^2(\vec{x_n})$	0.8115

	Нижняя граница	Верхняя граница
μ	-4.8942	-4.6216
σ^2	0.6639	1.0185

На координатной плоскости OYn построить прямую $y = \hat{\mu}(\vec{x_N})$, также графики функций $y = \hat{\mu}(\vec{x_n})$, $y = \underline{\mu}(\vec{x_N})$ и $y = \overline{\mu}(\vec{x_N})$ 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 $z = S^2(\vec{x_n})$, $z = \underline{S^2}(\vec{x_N})$ и $z = \overline{S^2}(\vec{x_N})$ как функций объема n выборки, где n изменяется от 1 до N: