

# Лабораторная работа 5, ТВМС

Бочарников Андрей, М3238

Ковешников Глеб, М3238

Шишкин Алексей, М3238

30 апреля 2020 г.

## Формулировка

Для трёх распределений  $X \sim N(a, \sigma^2)$ ,  $X \sim N(!, !)$  и распределения Лапласа, сравнить следующие оценки параметра  $a$ :

- 1.
- 2.
- 3.

!! Сравнивать оценки нужно с точки зрения квадратичного риска. сравнить их выборочные среднеквадратичные отклонения. Сравнить с теоретическими среднеквадратичными отклонениями.

## Входные данные

- Объем выборки:  $n_1 = 100, n_2 = 10000$
- $\alpha = 1$
- ?? Параметры нормального распределения:
- ?? Параметры равномерного распределения:
- ?? Параметры распределения Лапласа:
- Количество выборок:  $m = 100$

## Программа 1

Нормальное распределение.

### 3.1 Исходный код

```
pkg load statistics

function count_risks(a, sigma, n, m)
    C1 = 0.4;
    std1 = std2 = std3 = [];
    for i = 1 : n
        X = sort(normrnd(a, sigma, m, n));
        med = median(X);
        std1(i) = std(mean(X));
        std2(i) = std(med);
        std3(i) = std((X(1, :) + X(m, :)) / 2);
    endfor
    risk1 = mean((std1 - sigma / sqrt(m)) .^ 2);
    risk2 = mean((std2 - sqrt(pi) * sigma / sqrt(2 * m)) .^ 2);
    risk3 = mean((std3 - sqrt(C1) * sigma / sqrt(log(m))) .^ 2);
    printf("Risk1=%d, risk2=%d, risk3=%d\n", risk1, risk2, risk3)
endfunction
```

```
count_risks(1, 3, 100, 100);
count_risks(1, 3, 100, 10000);
```

## 3.2 Выходные данные

### Программа 2

Равномерное распределение.

#### 4.1 Исходный код

```
pkg load statistics

function count_risks(a, delta, n, m)
    std1 = std2 = std3 = [];
    for i = 1 : n
        X = sort(normrnd(a - delta / 2, a + delta / 2, m, n));
        med = median(X);
        std1(i) = std(mean(X));
        std2(i) = std(med);
        std3(i) = std((X(1, :) + X(m, :)) / 2);
    endfor
    risk1 = mean((std1 - delta / sqrt(12 * m)) .^ 2);
    risk2 = mean((std2 - delta / sqrt(4 * m)) .^ 2);
    risk3 = mean((std3 - delta / sqrt(2 * m * m)) .^ 2);
    printf("Risk1=%d, risk2=%d, risk3=%d\n", risk1, risk2, risk3)
endfunction

count_risks(1, 3, 100, 100);
count_risks(1, 3, 100, 10000);
```

## 4.2 Выходные данные

### Программа 3

Распределение Лапласа.

#### 5.1 Исходный код

```
pkg load statistics

function count_risks(a, u, n, m)
    C2 = 0.9;
    std1 = std2 = std3 = [];
    for i = 1 : n
        X = sort(laplace_rnd(m, n));
        % X += u ???
        med = median(X);
        std1(i) = std(mean(X));
        std2(i) = std(med);
        std3(i) = std((X(1, :) + X(m, :)) / 2);
    endfor
    risk1 = mean((std1 - u * sqrt(2) / sqrt(m)) .^ 2);
    risk2 = mean((std2 - u / sqrt(m)) .^ 2);
    risk3 = mean((std3 - sqrt(C2) * u) .^ 2);
```

```
printf("Risk1_=%d,_risk2_=%d,_risk3_=%d\n", risk1 , risk2 , risk3)
endfunction

count_risks(1, 3, 100, 100);
count_risks(1, 3, 100, 10000);
```

## 5.2 Выходные данные

TODO: вывод о том какая из оценок с точки зрения квадратичного риска является наилучшей.

## Вывод