

# Лабораторная работа 4, ТВМС

Бочарников Андрей, М3238

Ковешников Глеб, М3238

Шишкин Алексей, М3238

29 марта 2020 г.

## Формулировка

Для случайной величины, распределенной по нормальному закону с параметрами  $(a, \sigma^2)$ , выполнить следующие действия:

1. Задать параметры распределения  $X \sim N(a, \sigma^2)$ .
  2. Построить выборку генеральной совокупности  $X$ .
  3. Построить график гистограммы.
  4. Проверить гипотезу о виде распределения по критерию хи-квадрат.
- Аналогично для  $X \sim U(a, b)$  - равномерно распределенной на  $[a, b]$  случайной величины.

## Входные данные

- Выборка генеральной совокупности:  $n = 10^5$
- TODO.

## Программа 1

Нормальное распределение.

### 3.1 Исходный код

```
pkg load statistics

clc;
clear all;

function res = f(x)
    res = exp(- x ^ 2 / 2);
endfunction

function res = laplace(x)
    res = quad(@f, 0, x) * 1 / sqrt(2 * pi);
endfunction

n = 10 ^ 5;
mu = 1;
sigma = 1;
alpha = 0.95;
real_hi2 = 100734.7;

X = sort(normrnd(mu, sigma, n, 1));

l = min(X);
r = max(X);
```

```

buckets = ceil((r - 1) * n ^ (1 / 3));
delta = (r - 1) / buckets;
x_coords = zeros(buckets, 1);
cnt_in_bucket = zeros(buckets, 1);
y_coords = zeros(buckets, 1);
walls = zeros(buckets, 1);

for i = 1 : buckets
    cur_l = (i - 1) * delta + 1;
    cur_r = cur_l + delta;
    walls(i) = cur_r;
    x_coords(i) = (cur_r + cur_l) / 2;
    cnt_in_bucket(i) = sum(X <= cur_r) - sum(X < cur_l);
    y_coords(i) = cnt_in_bucket(i) / (n * delta);
endfor

bar(x_coords, y_coords);

printf("Размер_выборки_=_%d\n", n);
printf("Границы_=_[%d;_%d]\n", 1, r);
printf("Выбранная_длина_интервалов_=_%d\n", delta);
printf("Количество_интервалов_=_%d\n", buckets);

printf("\\n");

# PART 2

E = sum(x_coords .* cnt_in_bucket) / n;
D = sum((x_coords - E) .^ 2 .* cnt_in_bucket) / n;
FIXED_D = n / (n - 1) * D;
SQ_D = sqrt(FIXED_D);

printf("Предполагаемое_матожидание_=_%d\n", E);
printf("Предполагаемая_дисперсия_=_%d\n", FIXED_D);

# suffix P means Предполагаемое""

z = (walls - E) / SQ_D;

P = zeros(buckets, 1);

P(1) = laplace(z(1)) - (-0.5);
for i = 2 : buckets
    P(i) = laplace(z(i)) - laplace(z(i - 1));
endfor

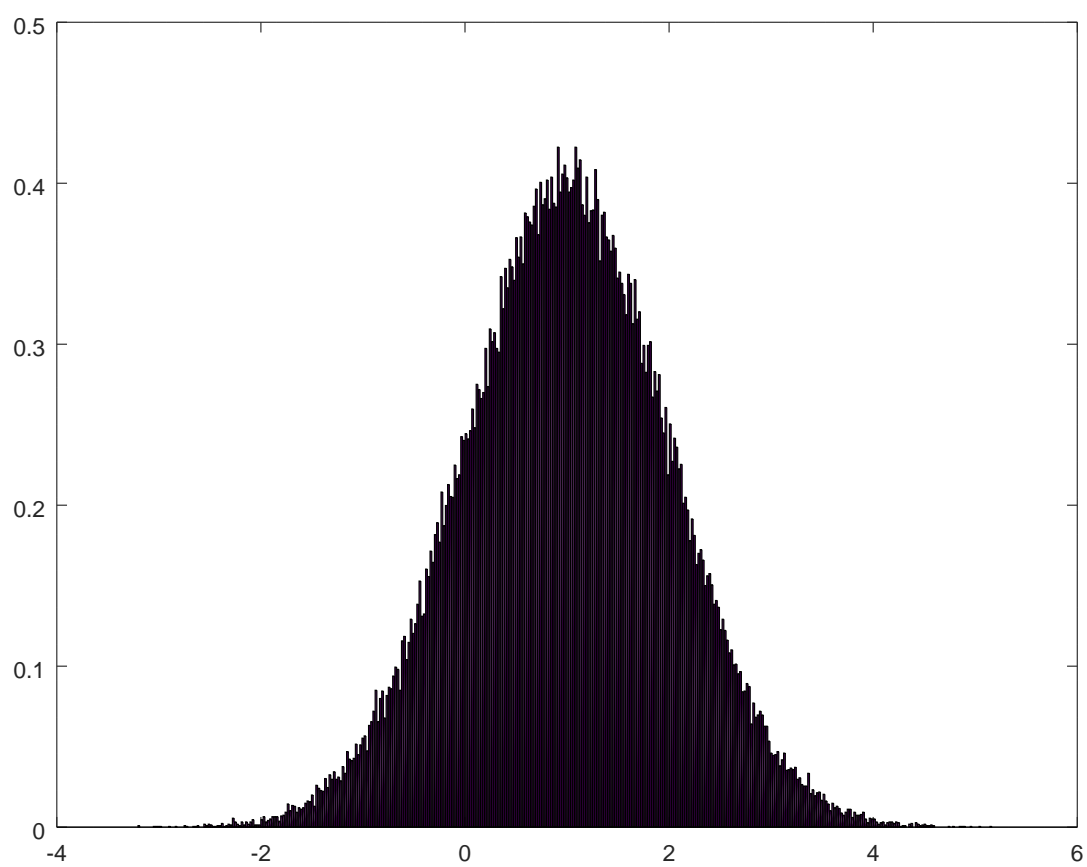
nP = P * n;

hi2 = sum((cnt_in_bucket - nP) .^ 2) ./ nP);

printf("Полученное_значение_Хи_квадрат_=_%d\n", hi2);
printf("Теоретическое_значение_Хи_квадрат_=_%d\n", real_hi2);
printf("Заданное_распределение_есть_равномерное_=_%d\n", hi2 < real_hi2);

```

## 3.2 График



## Программа 2

Равномерное распределение.

### 4.1 Исходный код

```
pkg load statistics

clc;
clear all;

n = 10 ^ 5;
a = 20;
b = 80;
alpha = 0.95;
real_hi2 = 100734.7;

X = sort(unifrnd(a, b, 1, n));

# PART 1

l = min(X);
r = max(X);
buckets = ceil((r - l) * n ^ (1 / 3));
delta = (r - l) / buckets;
x_coords = zeros(buckets, 1);
cnt_in_bucket = zeros(buckets, 1);
y_coords = zeros(buckets, 1);
walls = zeros(buckets, 1);

for i = 1 : buckets
    cur_l = (i - 1) * delta + l;
    cur_r = cur_l + delta;
    walls(i) = cur_r;
    x_coords(i) = (cur_r + cur_l) / 2;
    cnt_in_bucket(i) = sum(X <= cur_r) - sum(X < cur_l);
    y_coords(i) = cnt_in_bucket(i) / (n * delta);
endfor

bar(x_coords, y_coords);

printf("Размер_выборки_=%d\n", n);
printf("Границы_=[%d; %d]\n", l, r);
printf("Выбранная_длина_интервалов_=%d\n", delta);
printf("Количество_интервалов_=%d\n", buckets);

printf("\n");

# PART 2

E = sum(x_coords .* cnt_in_bucket) / n;
D = sum((x_coords - E) .^ 2 .* cnt_in_bucket) / n;
SQ_D = sqrt(D);

printf("Предполагаемое_матожидание_=%d\n", E);
printf("Предполагаемая_дисперсия_=%d\n", D);

# suffix P means Предполагаемое"
aP = E - sqrt(3) * SQ_D;
bP = E + sqrt(3) * SQ_D;
fP = 1 / (bP - aP);
```

```

printf( "Предполагаемые_границы_=[%d;_%d]\n", aP, bP);
printf( "Предполагаемое_значение_функции_=%d\n", fP );

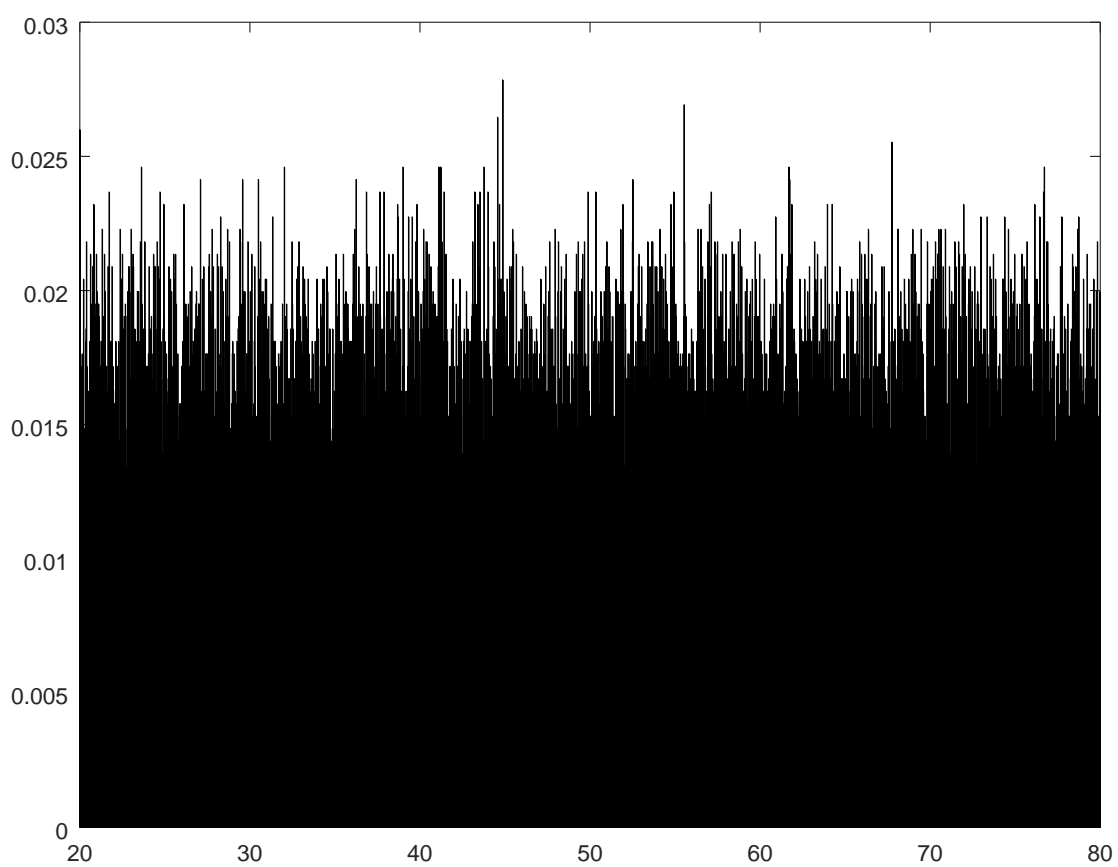
nP = zeros(buckets, 1);
nP(1) = n * fP * (walls(1) - aP);
for i = 1 : buckets - 2
    nP(i + 1) = n * fP * (walls(i + 1) - walls(i));
endfor
nP(buckets) = n * fP * (bP - walls(buckets - 1));

hi2 = sum((cnt_in_bucket - nP) .^ 2) ./ nP);

printf( "Полученное_значение_Хи_квадрат_=%d\n", hi2);
printf( "Теоретическое_значение_Хи_квадрат_=%d\n", real_hi2);
printf( "Заданное_распределение_есть_равномерное_=%d\n", hi2 < real_hi2);

```

## 4.2 График



## Вывод

TODO.