Лабораторная работа 4, ТВМС

Бочарников Андрей, М3238
Ковешников Глеб, М3238
Шишкин Алексей, М3238

6 апреля 2020 г.

Формулировка

Для случайной величины, распределенной по нормальному закону с параметрами (a,σ^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

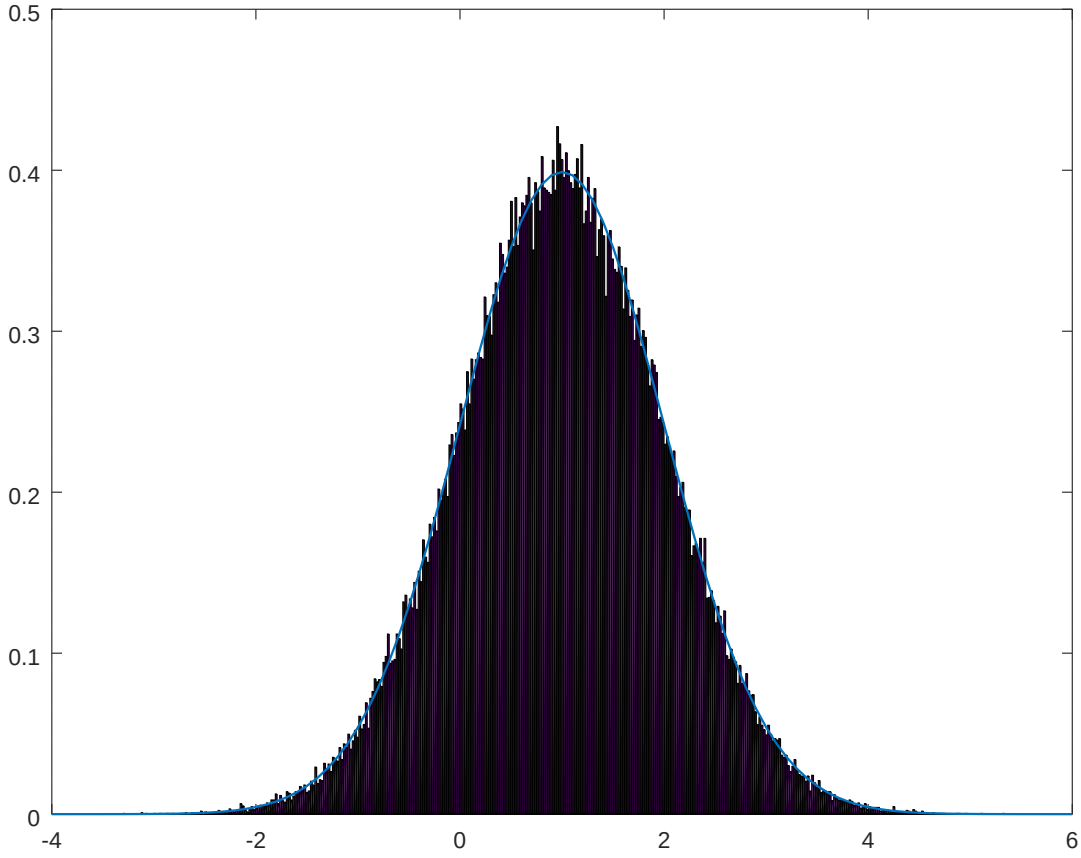
Нормальное распределение.

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 - 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

x_coords_for_normpdf = -4:0.1:6;
bar(x_coords, y_coords);
hold on;
plot(x_coords_for_normpdf, normpdf(x_coords_for_normpdf, mu, sigma), "linewidth", 1);

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;
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


0.5
0.4
0.3
0.2
0.1
0
-4
-2
0
2
4
6

Программа 2

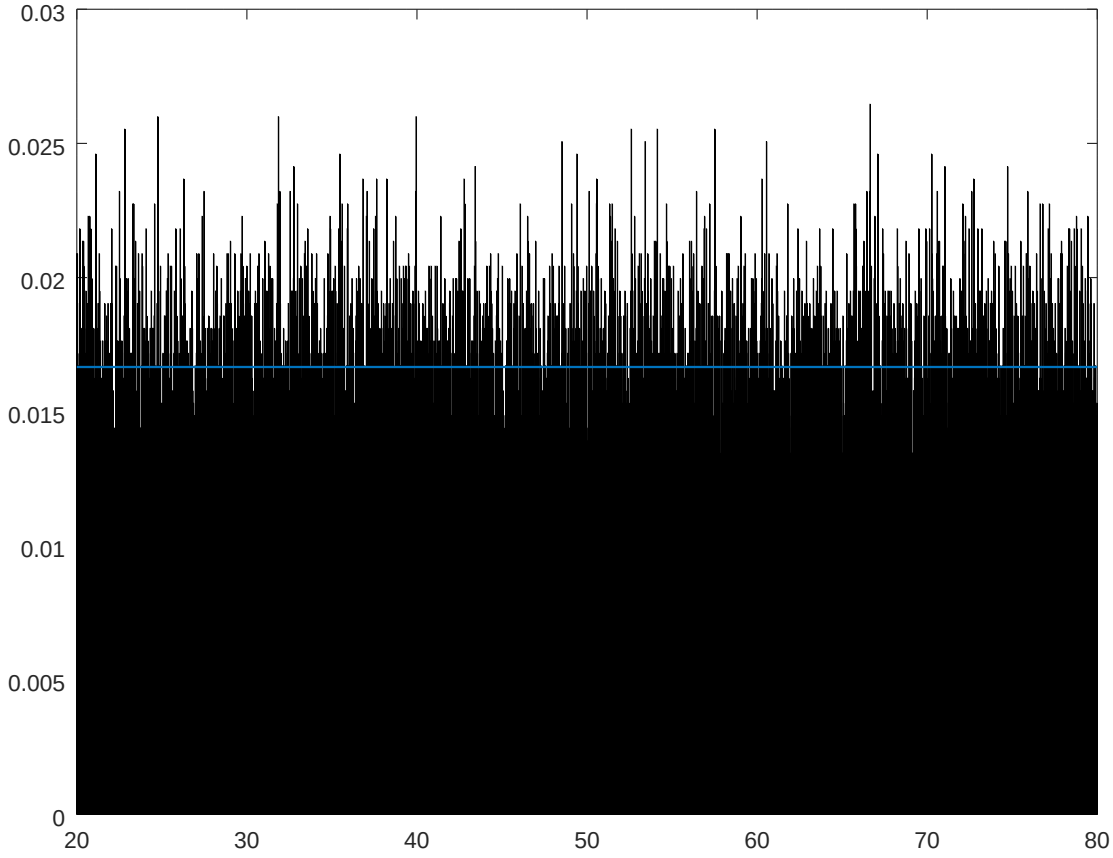
Равномерное распределение.

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

real_y = 1 / (b - a);
bar(x_coords, y_coords);
hold on;
plot([a b], [real_y real_y], "linewidth", 1);

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


0.03
0.025
0.02
0.015
0.01
0.005
0
20
30
40
50
60
70
80

Вывод

TODO.