

Министерство образования Российской Федерации
Московский Государственный Технический
Университет им. Н.Э. Баумана

Отчет по лабораторной работе №1
По курсу «Математическая статистика»

**Тема: «Гистограмма и эмпирическая
функция распределения»**
Вариант 5

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

Преподаватель: **Власов П.А.**

Москва, 2018

Содержание

Формулы для вычисления	2
Определения	2
Текст программы	3
Результаты расчетов для выборки из индивидуального вари- анта	7

Формулы для вычисления

Максимальное значение выборки:

$$M_{max} = \max(x_1, x_2, \dots, x_n)$$

Минимальное значение выборки:

$$M_{min} = \min(x_1, x_2, \dots, x_n)$$

Размах выборки:

$$R = M_{max} - M_{min}$$

Выборочное среднее:

$$\hat{\mu}(\vec{X}) = \bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$$

Выборочная дисперсия:

$$S^2(\vec{X}) = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$$

Определения

Эмпирическая плотность распределения, соответствующая реализации \vec{x}_n случайной выборки \vec{X}_n - функция

$$f_n(x) = \begin{cases} \frac{n_i}{n\Delta} & , \text{ если } x \in J_i \\ 0 & , \text{ иначе} \end{cases}$$

где $\Delta = \frac{x_{(n)} - x_{(1)}}{m}$ - размер интервала,

$$m = [\log_2 n] + 1,$$

$$J_i = [x_{(1)} + (i-1)\Delta; x_{(1)} + i\Delta],$$

$$J_m = [x_{(1)} + (m-1)\Delta; x_{(1)} + m\Delta]$$

n_i = число элементов выборки принадлежащих J_i , $i = 1, \dots, m$

Гистограмма - график функции $f_n(x)$, которая представляет собой кусочно-постоянную функцию

Эмпирическая функция распределения, построенная по выборке - отображение $F_n : R \rightarrow R$, определенное правилом

$F_n(x) = \frac{n(x, \vec{x})}{n}$, где $n(x, \vec{x})$ - количество элементов выборки, которые меньше x , а n - объем этой выборки.

Если все элементы этой выборки попарно различны, то

$$F_n(x) = \begin{cases} 0 & , x \leq x_{(1)} \\ \frac{i}{n} & , x \in (x_{(i)}; x_{(i+1)}] \\ 1 & , x > x_{(n)} \end{cases}$$

Текст программы

Листинг 1: Функция Main

```

1 function Main()
2     sample = importdata('data(var5).txt');
3     %sample = importdata('1.txt');
4     sample = sort(sample);
5
6     [Max] = getMaxValue(sample);
7     [Min] = getMinValue(sample);
8     [Range] = getRange(sample);
9     fprintf("min value: %.2f \n", Min);
10    fprintf("max value: %.2f \n", Max);
11    fprintf("range      : %.2f \n", Range);
12
13    [MX] = getExpectedValue(sample);
14    [DX] = getDispersionValue(sample);
15    fprintf("MX          : %.3f \n", MX);
16    fprintf("DX          : %.3f \n", DX);
17
18    [GroupTable, m] = Group(sample);
19    for i=1:m-1
20        fprintf("[%5.2f;%5.2f) ", GroupTable(1,i),
21                GroupTable(1,i+1));
22    end
23    fprintf("[%5.2f;%5.2f]\n", GroupTable(1,m), Max);
24    for i=1:m
25        fprintf("%13d ", GroupTable(2,i));
26    end
27    fprintf("\n\n\n");

```

```

27
28     figure(1);
29     grid;
30     hold on;
31     HistogramAndDensity(sample)
32
33     figure(2);
34     grid;
35     hold on;
36     EmpiricalAndDensity(sample)
37 end

```

Листинг 2: Функции вычисления максимального и минимального значения

```

1 function [Max] = getMaxValue(sample)
2     Max = max(sample);
3 end
4
5 function [Min] = getMinValue(sample)
6     Min = min(sample);
7 end

```

Листинг 3: Функция вычисления размаха выборки

```

1 function [Range] = getRange(sample)
2     Range = getMaxValue(sample) - getMinValue(sample);
3 end

```

Листинг 4: Функции вычисления оценок

```

1 function [MX] = getExpectedValue(sample)
2     n = length(sample);
3     MX = sum(sample)/n;
4 end
5
6 function [DX] = getDispersionValue(sample)
7     n = length(sample);
8     MX = getExpectedValue(sample);
9     DX = sum((sample - MX).^2) / (n-1);
10 end

```

Листинг 5: Группировка значений

```

1 function [GroupTable, m] = Group(sample)
2     n = length(sample);
3     m = floor(log2(n)) + 2;
4     GroupTable = zeros(2,m);
5     Delta = getRange(sample)/m;
6
7     for k = 0:m-1
8         GroupTable(1, k+1) = sample(1)+Delta*k;
9     end
10
11     count = 0;
12     for i = 1:n
13         for j = 1:m-1
14             if sample(i) >= GroupTable(1,j) && sample(i) <
15                 GroupTable(1, j+1)
16                 GroupTable(2, j) = GroupTable(2, j) + 1;
17                 count = count + 1;
18                 break;
19             end
20         end
21     end
22     GroupTable(2, m) = n - count;
23 end

```

Листинг 6: Гистограмма и график функции плотности

```

1 function [] = HistogramAndDensity(sample)
2     [min] = getMinValue(sample);
3     [max] = getMaxValue(sample);
4     count = 100;
5     Delta = (max-min)/(count-1);
6
7     Graph = zeros(2, count);
8     [MX] = getExpectedValue(sample);
9     [DX] = getDispersionValue(sample);
10
11     for i = 1:count
12         X = min + Delta*(i-1);
13         Graph(1, i) = X;
14         Graph(2, i) = NormalDensityDistribution(X, MX, DX);
15     end
16

```

```

17     [GroupTable, n] = Group(sample);
18     x = zeros(n+1);
19     y = zeros(n+1);
20     znam = length(sample)*getRange(sample)/n;
21     for i = 1:n
22         x(i) = GroupTable(1,i);
23         y(i) = GroupTable(2,i) ./ znam;
24         fprintf("%f %f \n", x(i), y(i));
25     end
26     x(n+1) = max;
27     y(n+1) = y(n);
28
29     %histogram
30     stairs(x, y), grid;
31     %density graph
32     plot(Graph(1,:), Graph(2,:), 'r'), grid;
33 end
34
35 function [y] = NormalDensityDistribution(x, mx, dx)
36     y = exp(-(x-mx).^2/2/dx)/sqrt(2*pi*dx);
37 end

```

Листинг 7: Графики эмпирической функции и функции распределения случайной величины

```

1 function EmpiricalAndDensity(sample)
2     [min] = getMinValue(sample);
3     [max] = getMaxValue(sample);
4     count = 100;
5     Delta = (max-min)/(count-1);
6
7     Graph = zeros(2, count);
8     [MX] = getExpectedValue(sample);
9     [DX] = getDispersionValue(sample);
10
11     for i = 1:count
12         X = min + Delta*(i-1);
13         Graph(1,i) = X;
14         Graph(2,i) = NormalDistribution(X, MX, DX);
15     end
16
17     n = length(sample);
18     F = zeros(n);
19     for i = 1:n

```

```

20     F(i) = EmpiricFunc(sample(i), sample, n);
21 end
22
23 %graph
24 plot(Graph(1,:), Graph(2,:), 'r'), grid;
25 %empirical graph
26 stairs(sample, F), grid;
27 end
28
29 function [y] = NormalDistribution(x, mx, dx)
30     syms t;
31     y = 1/sqrt(2*pi*dx) * int( exp(-((t-mx).^2)/2/dx), t, -
        Inf, x);
32 end
33
34 function [Fi] = EmpiricFunc(x, sample, n)
35     count = 0;
36     for i = 1:n
37         if (sample(i) < x)
38             count = count + 1;
39         end
40     end
41     Fi = count/n;
42 end

```

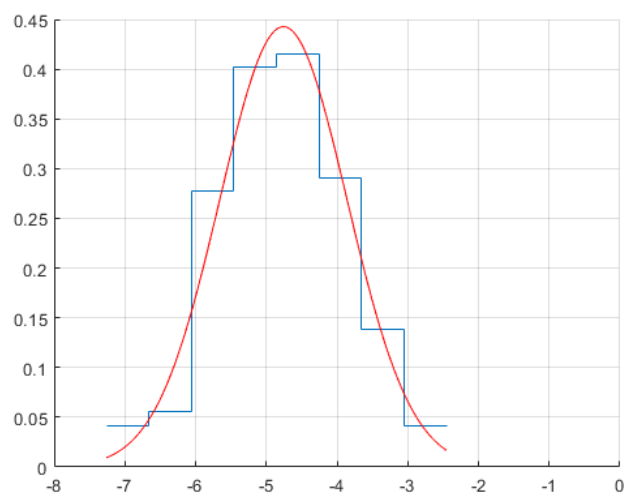
Результаты расчетов для выборки из индивидуального варианта

M_{min}	-7.26
M_{max}	-2.45
R	4.81
MX	-4.758
DX	0.812

Группировка значений выборки в m=8 интервалов

[-7.26; -6.66)	[-6.66; -6.06)	[-6.06; -5.46)	[-5.46; -4.86)
3	4	20	29
[-4.86; -4.25)	[-4.25; -3.65)	[-3.65; -3.05)	[-3.05; -2.45]
30	21	10	3

Построение на одной координатной плоскости гистограммы и графика функции плотности распределения вероятностей нормальной случайной величины с математическим ожиданием $\hat{\mu}$ и дисперсией S^2 : Построение



на другой координатной плоскости графика эмпирической функции распределения и функции распределения нормальной случайной величины с математическим ожиданием $\hat{\mu}$ и дисперсией S^2 :

