



1. **function** lab1 ()
2. **function** Mu = MathExpect(MyMassive)
3. Mu = sum(MyMassive)/**size**(MyMassive, 2);
4. **end**
6. **function** Sigma = Dispersion(MyMassive)
7. tmp = MathExpect(MyMassive);
8. Sigma = sum((MyMassive - tmp) .\* (MyMassive - tmp))/**size**(MyMassive, 2);
9. **end**
11. **function** countinterv = subIntervals(**size**)
12. countinterv = floor(log2(**size**)) + 2;
13. **end**
15. **function** Intervals(X, m)
16. sortX = sort(X);
17. n = **size**(sortX,2);
18. delta = (sortX(**end**) - sortX(1)) / m;
19. J = sortX(1):delta:sortX(**end**);
20. numElem = zeros(1, m);
22. **for** i = 1:n
23. **for** j = 1:(**size**(J,2) - 1)
24. if (sortX(i) >= J(j) && sortX(i) < J(j+1))
25. numElem(j) = numElem(j) + 1;
26. break;
27. **end**
28. **end**
29. **end**
30. numElem(**end**) = numElem(**end**) + 1;
32. **for** i = 1:**size**(numElem,2)
33. numElem(i) = numElem(i)/(n \* delta);
34. **end**
35. J = [J(1) J];
36. numElem = [0 numElem 0];
38. stairs(J, numElem), grid;
39. **end**
41. **function** f(X, M, D, m, R)
42. delta = R/m;
43. Sigma = sqrt(D);
44. Xn = **min**(X):delta/20:**max**(X);
45. Y = normpdf(Xn, M, Sigma);
46. plot(Xn, Y, '-\*');
47. **end**
48. **function** F(X, MX, DX, m, R)
49. delta = R/m;
50. Xn = **min**(X):delta/20:**max**(X);
51. Y = 1/2 \* (1 + erf((Xn - MX) / sqrt(2\*DX)));
53. plot(Xn, Y, '--');
54. **end**
55. **function** emprf(Mas)
56. [eY, eX] = ecdf(sort(Mas));
57. stairs(eX, eY)
58. **end**
60. clear all;
62. MyMassive = [11.89,9.60,9.29,10.06,9.50,8.93,9.58,6.81,8.69,9.62,9.01,10.59,10.50,11.53,9.94,8.84,8.91,6.90,9.76,7.09,11.29,11.25,10.84,10.76,7.42,8.49,10.10,8.79,11.87,8.77,9.43,12.41,9.75,8.53,9.72,9.45,7.20,9.23,8.93,9.15,10.19,9.57,11.09,9.97,8.81,10.73,9.57,8.53,9.21,10.08,9.10,11.03,10.10,9.47,9.72,9.60,8.21,7.78,10.21,8.99,9.14,8.60,9.14,10.95,9.33,9.98,9.09,10.35,8.61,9.35,10.04,7.85,9.64,9.99,9.65,10.89,9.08,8.60,7.56,9.27,10.33,10.09,8.51,9.86,9.24,9.63,8.67,8.85,11.57,9.85,9.27,9.69,10.90,8.84,11.10,8.19,9.26,9.93,10.15,8.42,9.36,9.93,9.11,9.07,7.21,8.22,9.08,8.88,8.71,9.93,12.04,10.41,10.80,7.17,9.00,9.46,10.42,10.43,8.38,9.01];
64. % Задание 1
65. %% поиск **Min** и **Max**
66. MinMas = **min**(MyMassive);
67. MaxMas = **max**(MyMassive);
69. %% размах R
71. R = MaxMas - MinMas;
73. %% оценки мат. ожидания и дисперсии
75. Mu = MathExpect(MyMassive);
76. Sigma = Dispersion(MyMassive);
78. %% Интервалы
80. countinterv = subIntervals(**size**(MyMassive, 2));
82. %% график 1
83. %%% f(X, MX, DX, m, R) - Нормальная функция плотности вероятности
84. Intervals(MyMassive, countinterv);
85. hold **on**;
86. f(MyMassive, Mu, (**size**(MyMassive,2)/(**size**(MyMassive,2) \* Sigma)), countinterv, R);
87. legend('Гистограмма','Функция плотности распределения нормальной случайной величины');
88. hold **off**;
90. %% график 2
91. %%% F(X, MX, DX, m, R) -
92. figure;
93. emprf(sort(MyMassive));
94. hold **on**;
95. F(sort(MyMassive), Mu, (**size**(MyMassive,2)/(**size**(MyMassive,2) \* Sigma)), countinterv, R);
96. grid **on**;
97. legend('Эмпирическая функция распределения','Функция распределения нормальной случайной величины');
98. hold **off**;
99. **end**