

	<p>Министерство образования и науки Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (МГТУ им. Н.Э. Баумана)</p>
---	--

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе № 1

По курсу: Моделирование

На тему: Изучение функции распределения и функции плотности распределения случайной величины

Студент:

Турсунов Жасурбек Рустамович

Группа: ИУ7-76Б

Преподаватель:

Рудаков Игорь Владимирович

Москва, 2021 г.

Содержание

1	Задание	2
2	Теоритическая часть	2
2.1	Равномерное распределение	2
2.2	Нормальное распределение	2
3	Результаты	3
3.1	Равномерное распределение	3
3.2	Нормальное распределение	5
4	Листинг кода	6

1 Задание

Реализовать программу для построения графиков функции и плотности для следующих распределений:

- равномерное распределение;
- нормальное распределение (вариант 15).

2 Теоритическая часть

2.1 Равномерное распределение

Непрерывное равномерное распределение - распределение случайной вещественной величины, принимающей значения, принадлежащие некоторому промежутку конечной длины, характеризующееся тем, что плотность вероятности на этом промежутке почти всюду постоянна.

Плотность распределения представлена в формуле 1.

$$f_X(x) = \begin{cases} \frac{1}{b-a}, x \in [a, b] \\ 0, x \notin [a, b] \end{cases} \quad (1)$$

Функция распределения представлена в формуле 2.

$$F_X(x) = \begin{cases} 0, x < a \\ \frac{x-a}{b-a}, a \leq x < b \\ 1, x \geq b \end{cases} \quad (2)$$

2.2 Нормальное распределение

Нормальное распределение — распределение вероятностей, которое в одномерном случае задаётся функцией плотности вероятности, совпадающей с функцией Гаусса.

Плотность распределения представлена в формуле 3.

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3)$$

Функция распределения представлена в формуле 4.

$$F(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^x e^{-\frac{(t-\mu)^2}{2\sigma^2}} dt \quad (4)$$

3 Результаты

3.1 Равномерное распределение

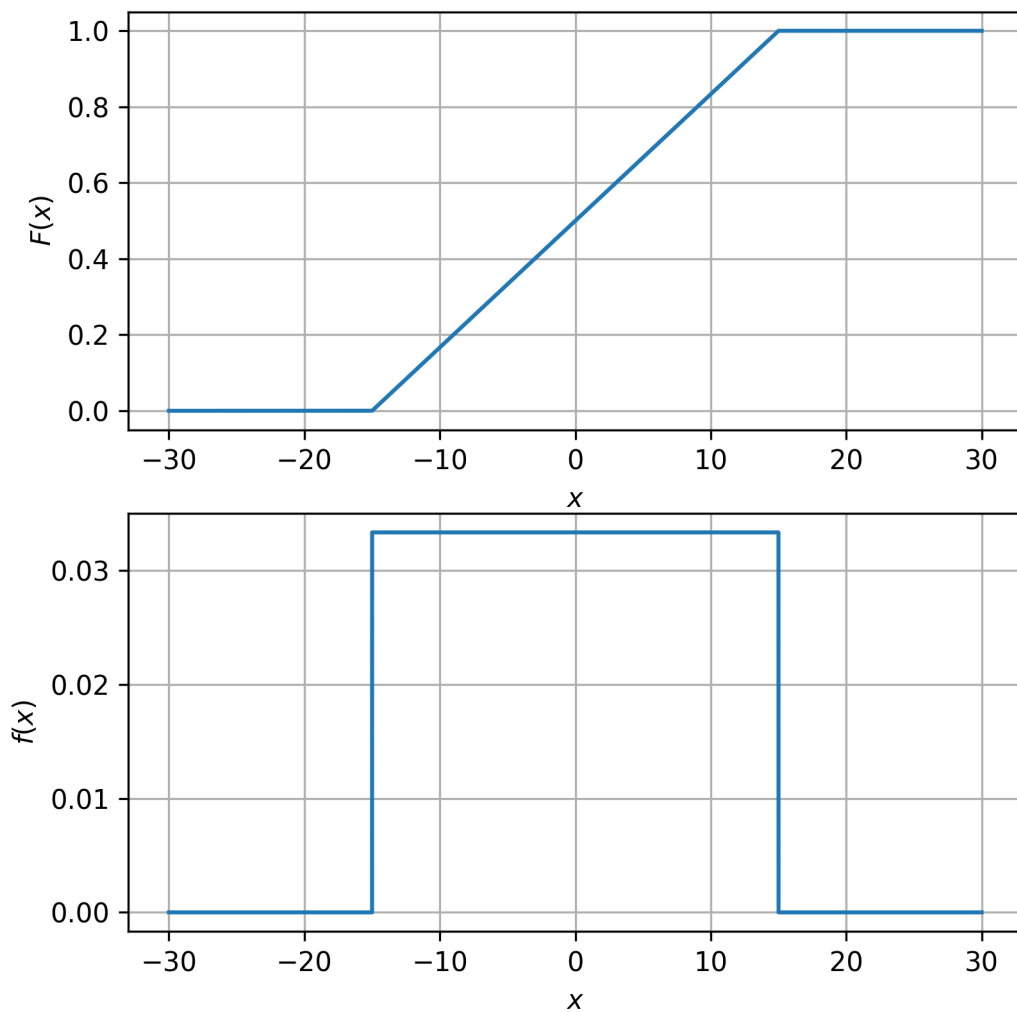


Рис. 1: Равномерное распределение при $a = -15$, $b = 15$

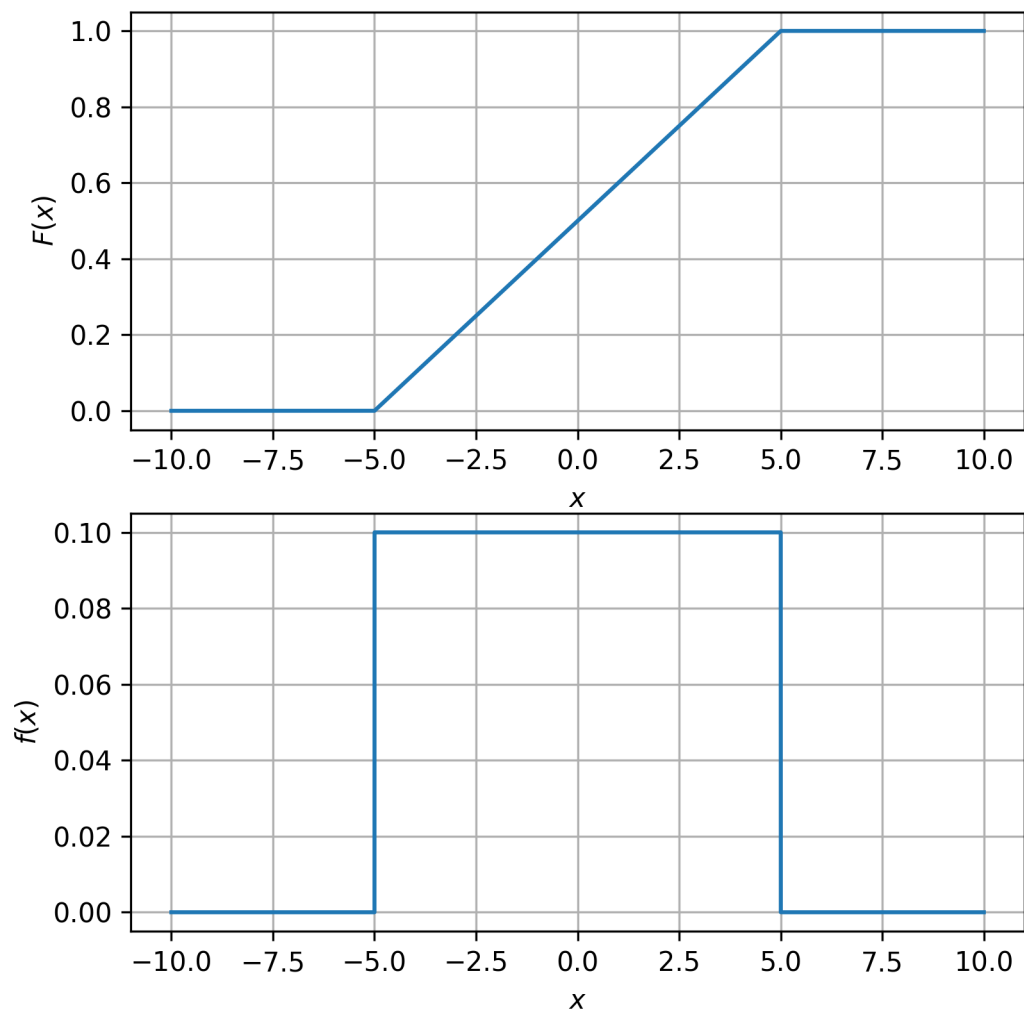


Рис. 2: Равномерное распределение при $a = -5$, $b = 5$

3.2 Нормальное распределение

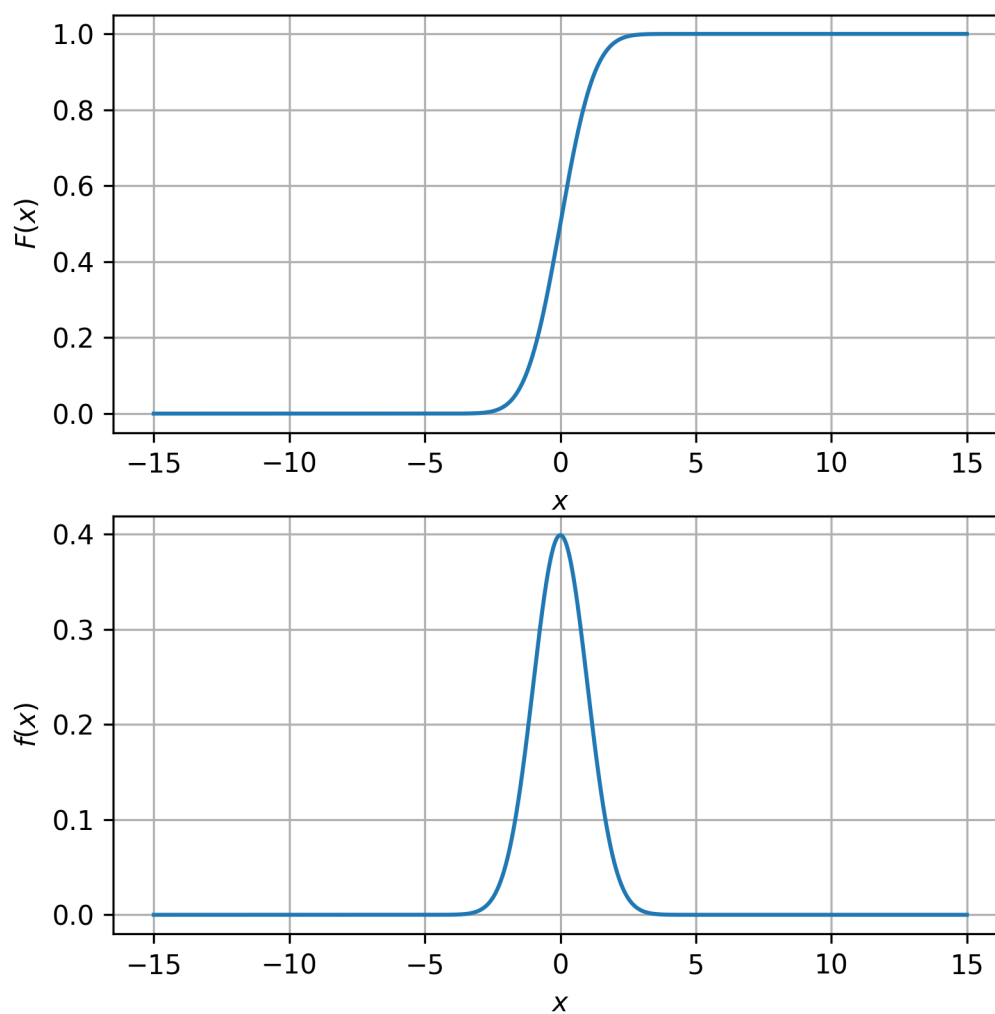


Рис. 3: Нормальное распределение при $\mu = 0$, $\sigma = 1$

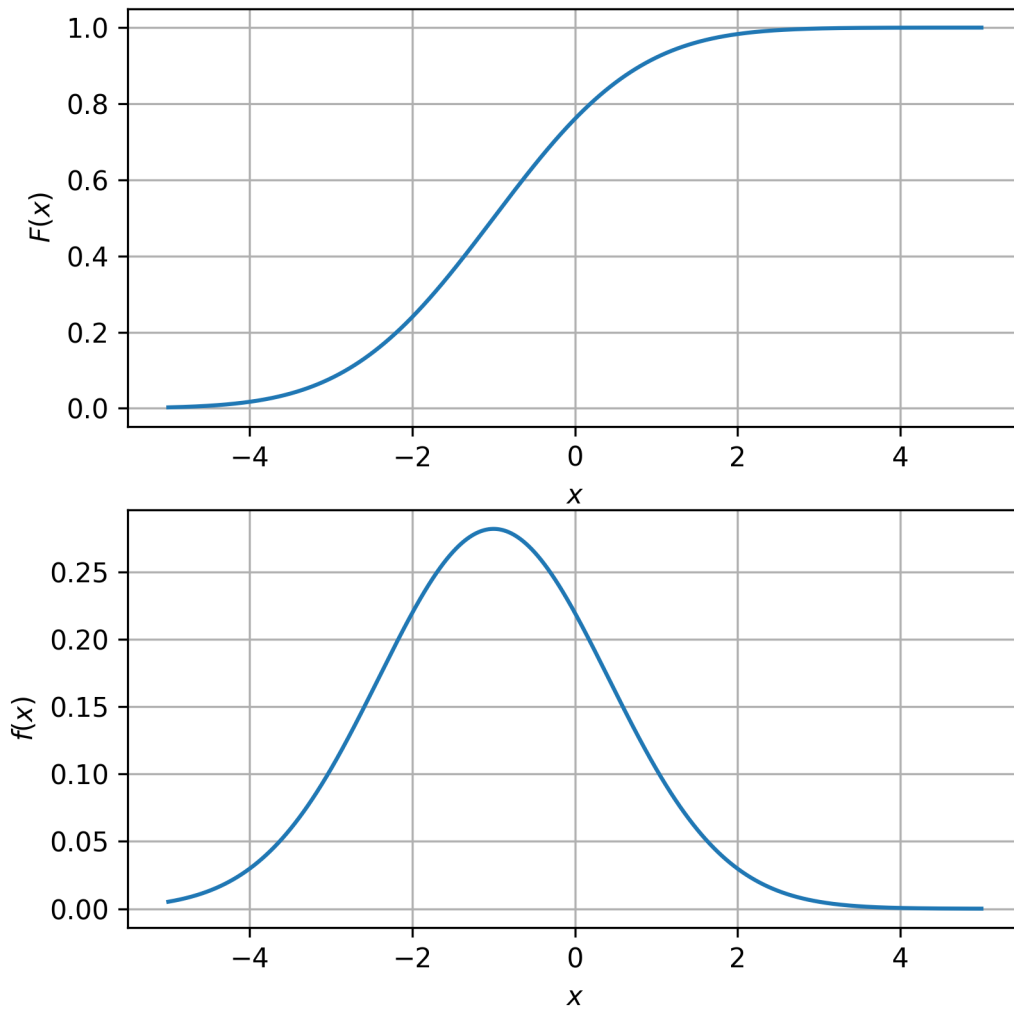


Рис. 4: Нормальное распределение при $\mu = -1$, $\sigma = 2$

4 Листинг кода

```

1 def main():
2     a = float(input("Enter start point a: "))
3     b = float(input("Enter end point b: "))
4     mu = float(input("Enter mu for normal distribution: "))
5     sigma = float(input("Enter sigma for normal distribution: "))
6
7     delta = b - a
8
9     x_uniform = np.arange(a - delta / 2, b + delta / 2, 0.001)
10    x_normal = np.arange(a, b, 0.001)
11
12

```

```

13     y_uniform_cdf = [uniform_distribution_cdf(a, b, value)
14                       for value in x_uniform]
15     y_uniform_pdf = [uniform_distribution_pdf(a, b, value)
16                      for value in x_uniform]
17
18     y_normal_cdf = normal_distribution_cdf(x_normal, mu, sigma)
19     y_normal_pdf = normal_distribution_pdf(x_normal, mu, sigma)
20
21     draw_plots(x_uniform, y_uniform_cdf, y_uniform_pdf)
22     draw_plots(x_normal, y_normal_cdf, y_normal_pdf)
23
24
25 def draw_plots(x, y_cdf, y_pdf):
26     fig, axs = plt.subplots(2, figsize=(6,7))
27
28     axs[0].plot(x, y_cdf)
29     axs[1].plot(x, y_pdf)
30
31     axs[0].set_xlabel('$x$')
32     axs[0].set_ylabel('$F(x)$')
33
34     axs[1].set_xlabel('$x$')
35     axs[1].set_ylabel('$f(x)$')
36
37     axs[0].grid(True)
38     axs[1].grid(True)
39
40     plt.show()
41
42
43 def uniform_distribution_cdf(a, b, x):
44     return (x - a) / (b - a) if (a <= x < b) else 0 if x < a else 1
45
46 def uniform_distribution_pdf(a, b, x):
47     return 1 / (b - a) if (a <= x <= b) else 0
48
49 def normal_distribution_cdf(x, mu, sigma):
50     return norm.cdf(x, mu, sqrt(sigma))
51

```



```
52 def normal_distribution_pdf(x, mu, sigma):  
53     return norm.pdf(x, mu, sqrt(sigma))
```

Листинг 1: программная реализация равномерного распределения и нормального распределения