Министерство науки и высшего образования 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)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Отчет по лабораторной работе №1 по курсу "Моделирование"

Тема Расследование функций плотности и распределения случайной величины

Студент Тэмуужин Янжинлхам

Группа ИУ7И-77б (ИУ7-73Б)

Преподаватель Рудаков И. В.

Москва — 2022 г.

1 Задание

Реализовать программу для построения графиков функции распределения и функции плотности вероятности для случайных величин, имеющих следующие распределения:

- равномерное распределение;
- нормальное распределение.

2 Теоретические сведения

2.1 Равномерное распределение

Случайная величина X имеет равномерное распределение $X \sim R(a, b)$ на отрезке $[a, b]$, где $a, b \in R$, если ее функция плотности $f(x)$ имеет следующий вид:

$$f(x) = \begin{cases} \frac{1}{b-a}, & x \in [a, b] \\ 0, & \text{иначе.} \end{cases} \tag{2.1}$$

Функция распределения $F(x) = \int_{-\infty}^{x} f(t)dt$ принимает вид:

$$F(x) = \begin{cases} \frac{x-a}{b-a}, & x \in [a, b] \\ 1, & \text{x > b} \\ 0, & \text{x < a} \end{cases} \tag{2.2}$$

2.2 Нормальное распределение

Случайная величина X имеет нормальное распределение $X \sim N(m, \sigma^2)$ с параметрами m и σ^2 ($\sigma > 0$), если её функция плотности имеет следующий вид:

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-m)^2}{2\sigma^2}}, x \in R. \tag{2.3}$$

Функция распределения имеет вид:

$$F(x) = \frac{1}{\sqrt{2\pi}\sigma} \int_{-\infty}^{x} e^{-\frac{(t-m)^2}{2\sigma^2}} dt \tag{2.4}$$

Функция плотности нормального распределения имеет характерную колоколообразную форму; m является координатой x «центра» этого колокола (центра симметрии), а σ характеризует разброс значений случайной величины; чем меньше σ, тем выше экстремум функции плотности.

3 Результаты работы программы

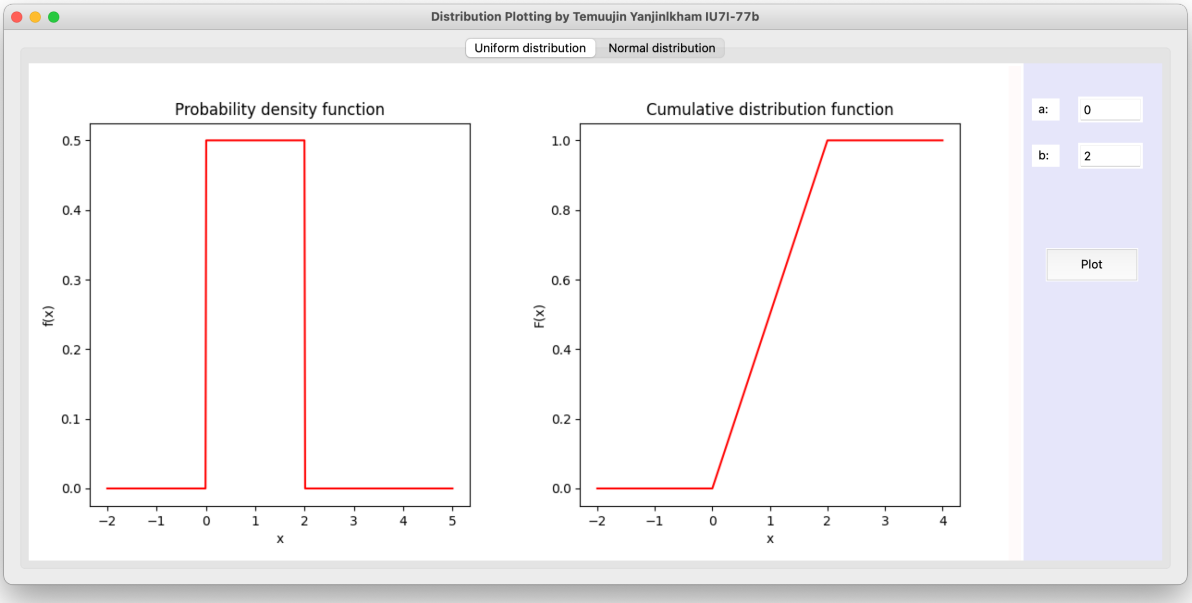

Рис. 3.1: Графики распределения $F(x)$ и плотности распределения $f(x)$ для равномерного распределения при $a = 0, b = 2$

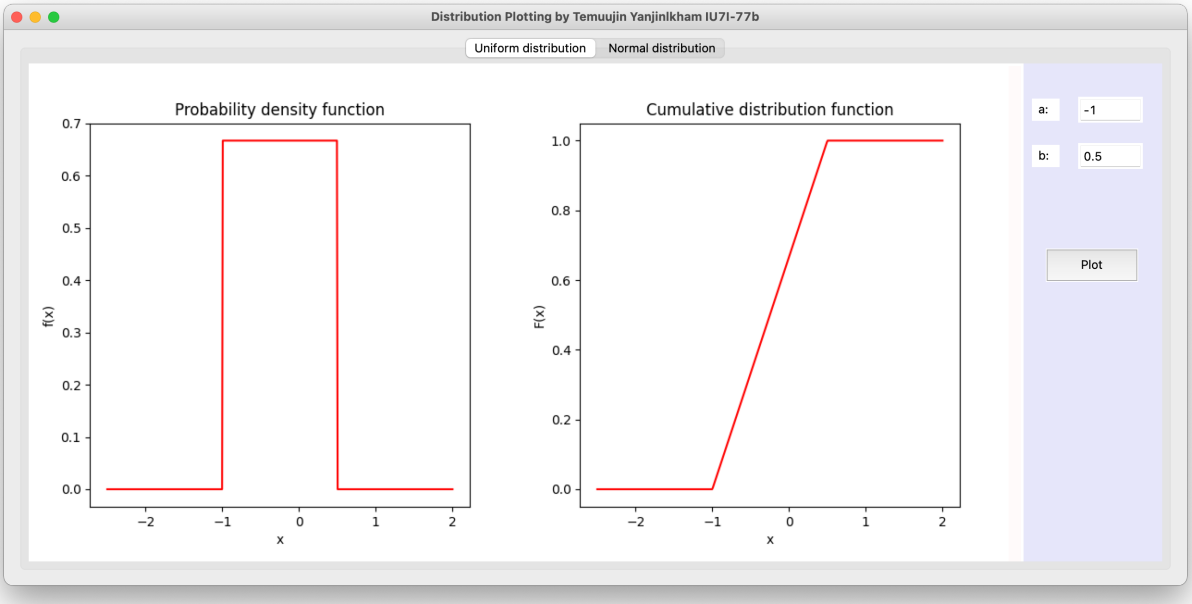

Рис. 3.2: Графики распределения $F(x)$ и плотности распределения $f(x)$ для равномерного распределения при $a = -1, b = 0.5$

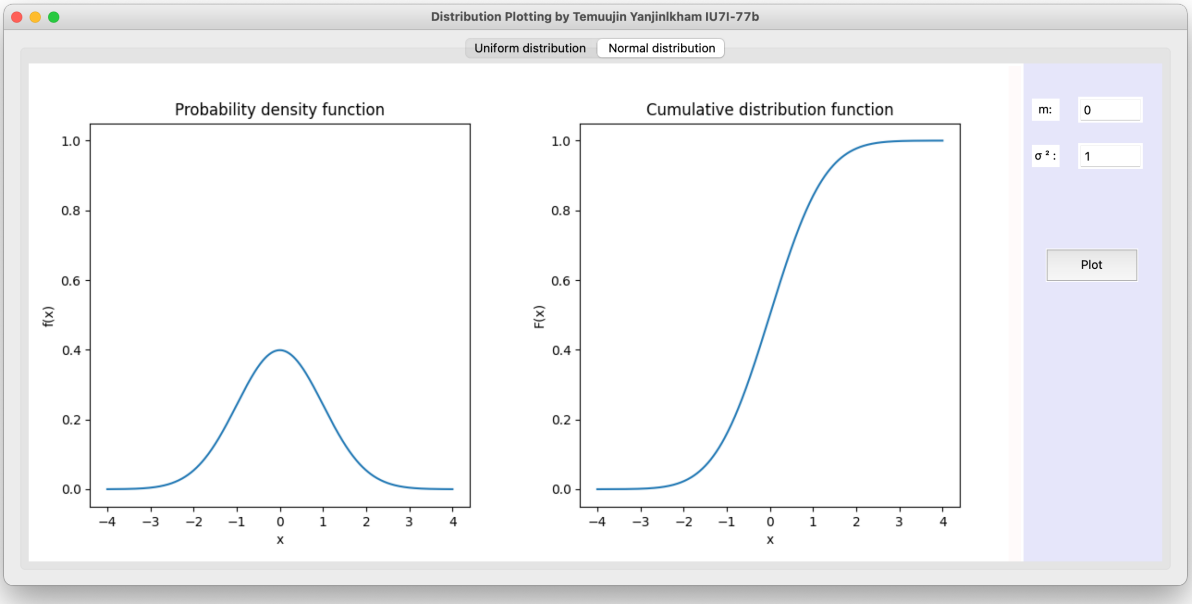

Рис. 3.3: Графики распределения $F(x)$ и плотности распределения $f(x)$ для нормального распределения при $m = 0, \sigma^2 = 1$

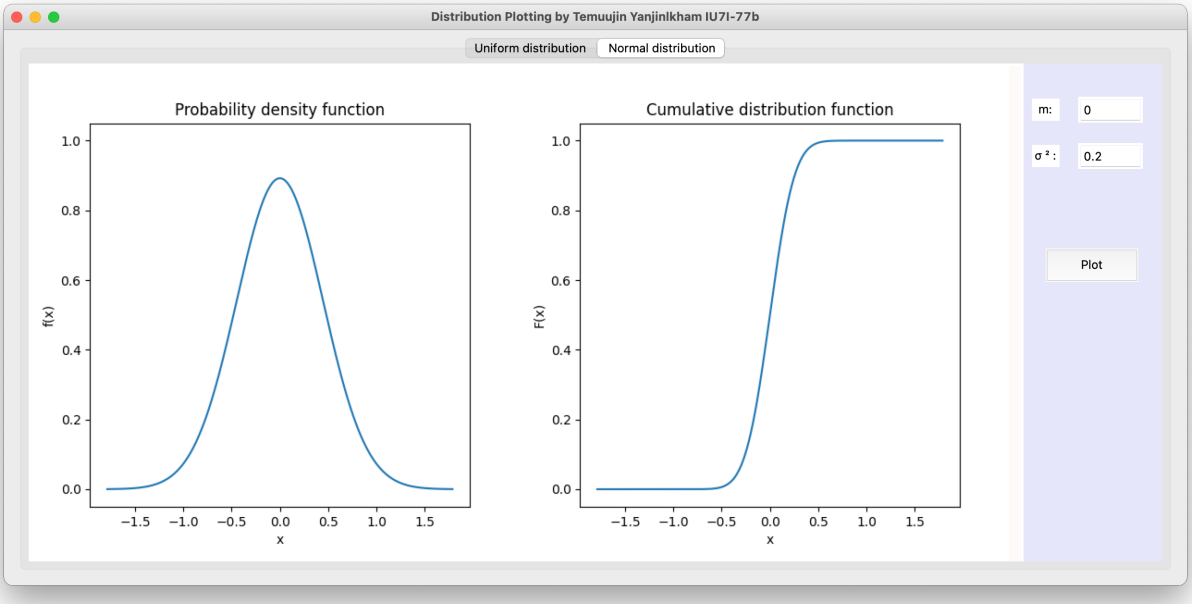

Рис. 3.4: Графики распределения $F(x)$ и плотности распределения $f(x)$ для нормального распределения при $m = 0, \sigma^2 = 0.2$

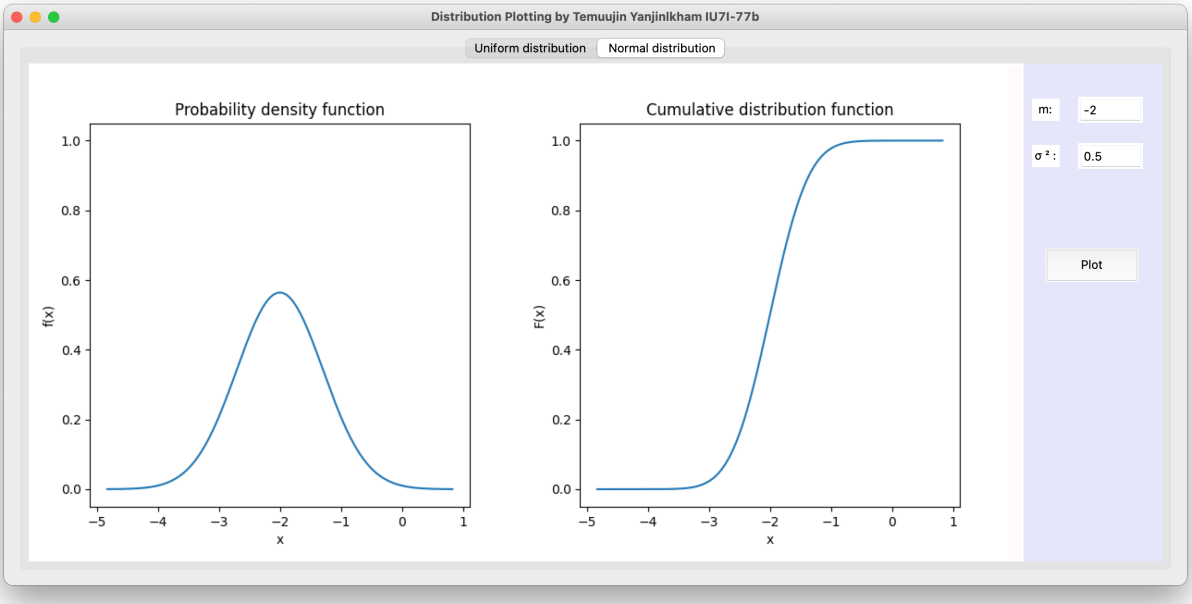

Рис. 3.5: Графики распределения $F(x)$ и плотности распределения $f(x)$ для нормального распределения при $m = -2, \sigma^2 = 0.5$

4 Код программы

```
from tkinter import *
from tkinter import ttk
from statistics import NormalDist
from math import exp, sqrt, pi, pow
import numpy
import matplotlib.pyplot as plt
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg


def uniform_density(x, a, b):
    if x >= a and x <= b:
        return 1 / (b - a)
    return 0

def uniform_dist(x, a, b):
    if x < a:
        return 0
    elif x > b:
        return 1
    else:
        return (x - a) / (b - a)

def norm_density(x, m, sigma):
    return exp(- pow((x - m)/sigma, 2) / 2) / (sigma * sqrt(2 * pi))

def norm_dist(x, m, sigma):
    return NormalDist(mu=m, sigma=sigma).cdf(x)

def plot_uniform():
    a = float(a_ent.get())
    b = float(b_ent.get())
    start = 2 * a - b
    stop = 2 * b - a

    # Density graph
    figure1 = plt.Figure(figsize=(5.3,5), dpi=100)
    ax1 = figure1.add_subplot(111)
    density = FigureCanvasTkAgg(figure1, tab1)
    density.get_tk_widget().pack(side=LEFT, fill=BOTH)

    density_x = numpy.linspace(start, stop, num=500, endpoint=True, retstep=False,
        dtype=None, axis=0)
    density_y = [uniform_density(x, a, b) for x in density_x]

```

```
    ax1.plot(density_x, density_y, color="red")
    ax1.set_title("Probability␣density␣function")
    ax1.set_xlabel("x")
    ax1.set_ylabel("f(x)")

    # Distribution graph
    figure2 = plt.Figure(figsize=(5.3,5), dpi=100)
    ax2 = figure2.add_subplot(111)
    dist = FigureCanvasTkAgg(figure2, tab1)
    dist.get_tk_widget().pack(side=LEFT, fill=BOTH)

    dist_x = numpy.linspace(start, stop, num=500, endpoint=True, retstep=False,
        dtype=None, axis=0)
    dist_y = [uniform_dist(x, a, b) for x in dist_x]

    ax2.plot(dist_x, dist_y, color="red")
    ax2.set_title("Cumulative␣distribution␣function")
    ax2.set_xlabel("x")
    ax2.set_ylabel("F(x)")

def plot_normal():
    m = float(m_ent.get())
    sigma = sqrt(float(sigma_ent.get()))

    start = m - 4*sigma
    stop = m + 4*sigma

    # Density graph
    figure1 = plt.Figure(figsize=(5.3,5), dpi=100)
    ax1 = figure1.add_subplot(111)
    figure1.delaxes(ax1)
    ax1 = figure1.add_subplot(111)
    density = FigureCanvasTkAgg(figure1, tab2)
    density.get_tk_widget().pack(side=LEFT, fill=BOTH)

    arr_x = numpy.linspace(start, stop, num=500, endpoint=True, retstep=False,
        dtype=None, axis=0)
    density_y = [norm_density(x, m, sigma) for x in arr_x]

    ax1.plot(arr_x, density_y)
    ax1.set_title("Probability␣density␣function")
    ax1.set_xlabel("x")
    ax1.set_ylabel("f(x)")
    ax1.set_ylim([-0.05, 1.05])

    # Distribution graph
    figure2 = plt.Figure(figsize=(5.3,5), dpi=100)
    ax2 = figure2.add_subplot(111)
```

```
    figure2.delaxes(ax2)
    ax2 = figure2.add_subplot(111)
    dist = FigureCanvasTkAgg(figure2, tab2)
    dist.get_tk_widget().pack(side=LEFT, fill=BOTH)

    dist_y = [norm_dist(x, m, pow(sigma, 2)) for x in arr_x]

    ax2.plot(arr_x, dist_y)
    ax2.set_title("Cumulative␣distribution␣function")
    ax2.set_xlabel("x")
    ax2.set_ylabel("F(x)")
    ax2.set_ylim([-0.05, 1.05])


if __name__ == "__main__":
    root = Tk()
    root.configure(bg="lavender")
    root.title('Distribution␣Plotting␣by␣Temuujin␣Yanjinlkham␣IU7I-77b')
    root.geometry("1350x600")

    tabControl = ttk.Notebook(root)
    tab1 = Frame(tabControl, borderwidth=0, background="lavender", border=0)
    tab2 = Frame(tabControl, borderwidth=0, background="lavender", border=0)

    tabControl.add(tab1, text ='Uniform␣distribution')
    tabControl.add(tab2, text ='Normal␣distribution')
    tabControl.pack(expand = 1, fill ="both")

    canv1 = Canvas(tab1, width=1070, height=800, bg="snow")
    canv2 = Canvas(tab2, width=1070, height=800, bg="snow")
    canv1.place(x=0, y=0)
    canv2.place(x=0, y=0)

    # Uniform tab
    Label(tab1, text="a:␣").place(x=1100, y=50, anchor="center", width=30)
    a_ent = Entry(tab1, width="50")
    a_ent.place(x=1170, y=50, anchor="center", width=70)

    Label(tab1, text="b:␣").place(x=1100, y=100, anchor="center", width=30)
    b_ent = Entry(tab1, width="50")
    b_ent.place(x=1170, y=100, anchor="center", width=70)

    plot_button = Button(master = tab1,
                         command = plot_uniform,
                         height = 2,
                         width = 10,
                         text = "Plot")
    plot_button.place(x=1100, y=200, width=100)
```

```
138
139     # Normal tab
140     Label(tab2, text="m:").place(x=1100, y=50, anchor="center", width=30)
141     m_ent = Entry(tab2, width="50")
142     m_ent.place(x=1170, y=50, anchor="center", width=70)
143
144     Label(tab2, text=u"\u03C3␣\u00B2␣\u003A").place(x=1100, y=100, anchor="center",
            width=30)
145     sigma_ent = Entry(tab2, width="50")
146     sigma_ent.place(x=1170, y=100, anchor="center", width=70)
147
148     plot_button = Button(master = tab2,
149                          command = plot_normal,
150                          height = 2,
151                          width = 10,
152                          text = "Plot")
153     plot_button.place(x=1100, y=200, width=100)
154
155     root.mainloop()
```