Практическая работа № 4

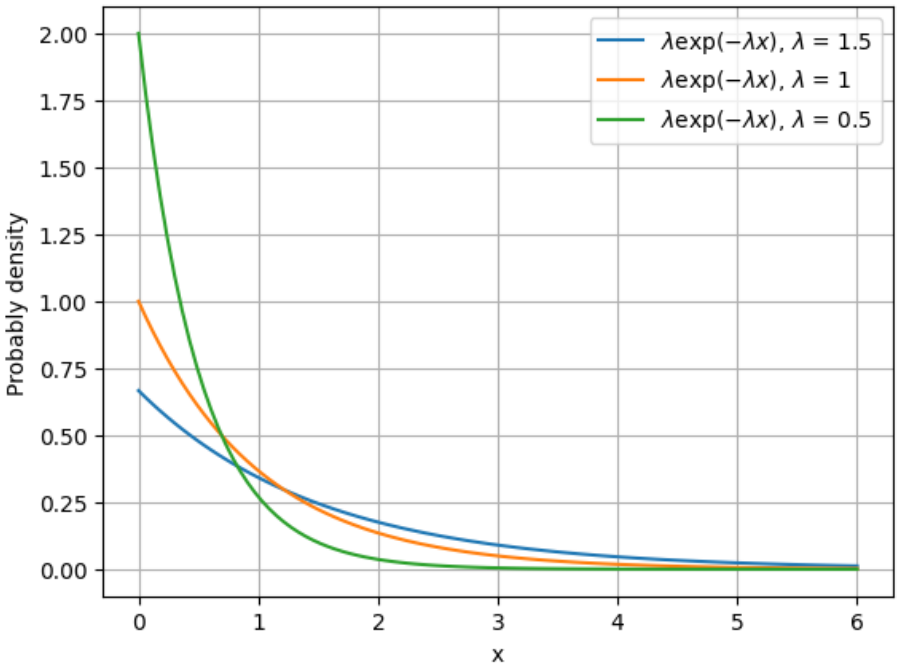
Валерий Сергеевич Верхотуров БСБО-05-20

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import expon
import random
from IPython.core.display_functions import display
```

Задание 1

Создать функцию, описывающую распределение плотности вероятности для экспоненциального закона распределения случайной величины. Построить графики плотности вероятности экспоненциального закона с заданными значениями параметра масштаба λ : $\lambda = 1.5$, $\lambda = 1$ и $\lambda = \frac{1}{2}$.

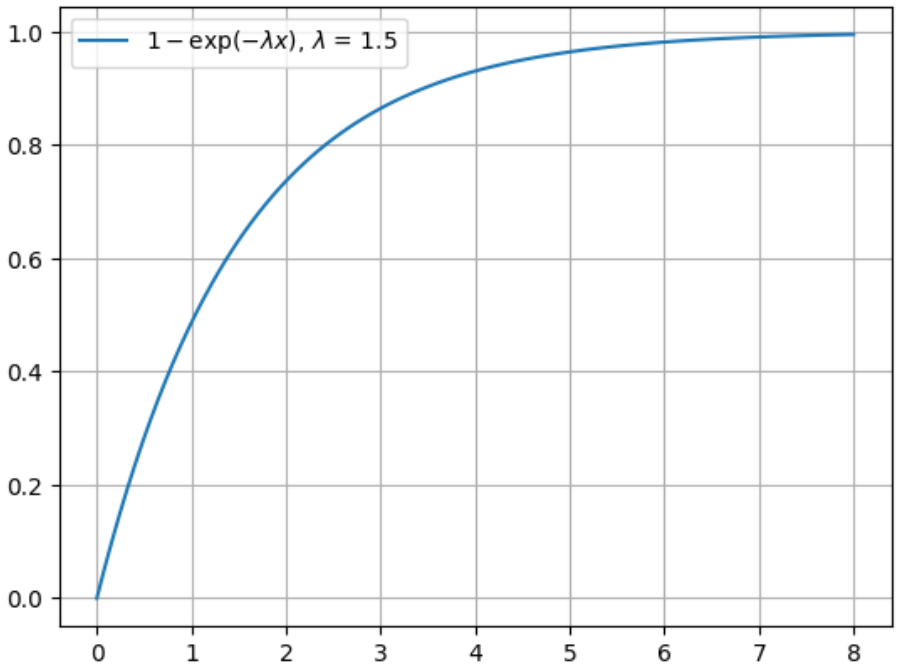
```
def task_1():
    scales = [1.5, 1, 0.5]
    x = np.linspace(0, 6, 100)
    for scale in scales:
        plt.plot(x, expon.pdf(x, scale=scale), label=f'$\lambda \exp(-\lambda x)$, $\lambda$
    plt.legend()
    plt.grid(True)
    plt.ylabel('Probably density')
    plt.xlabel('x')
    plt.show()

task_1()
```



Задание 2

Описать функцию обратного преобразования для экспоненциального закона с параметром $\lambda = 1.5$ и построить её график.

```
def task_2():
    scale = 1.5

    x = np.linspace(0, 8, 100)

    # Cumulative distribution function
    plt.plot(x, expon.cdf(x, scale=scale), label=f'$1-\exp(-\lambda x)$, $\lambda$ = {scale}
    plt.legend()
    plt.grid(True)

task_2()
```



Задания 3, 4

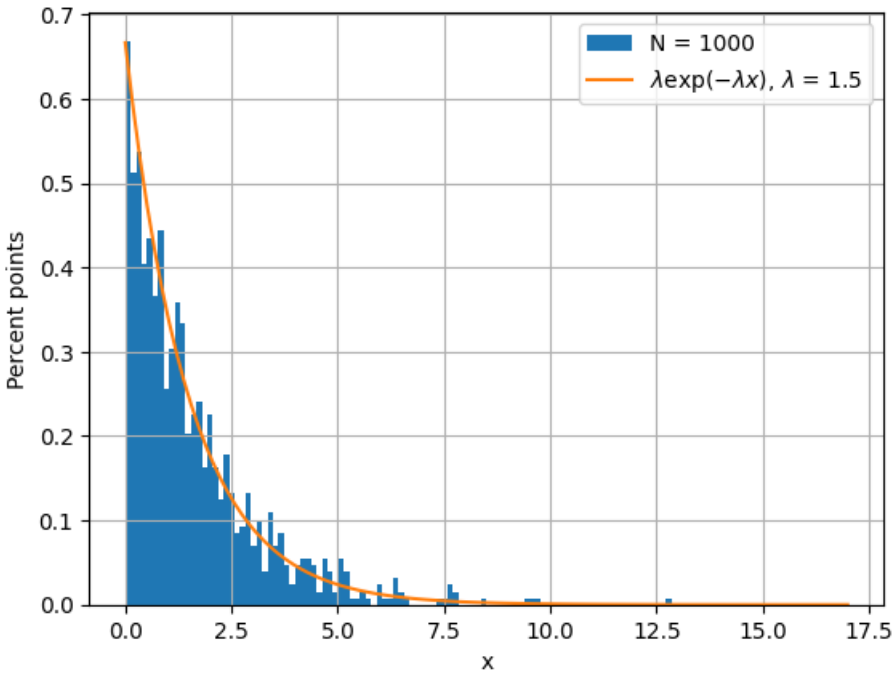
С помощью известной функции обратного преобразования (см. задание 2) провести моделирование экспоненциального закона распределения с помощью метода обратной функции. Для моделирования равномерно распределенной случайной величины использовать встроенный генератор случайных чисел `np.random.rand()`. Число экспериментов для моделирования принять равными $N = 10^3, 10^4\, 10^5\, 10^6$.

Построить гистограммы относительных частот для полученных последовательностей случайных чисел (см. задание 3) на 100 интервалах и рассчитать для них значения среднеквадратического отклонения от функции плотности вероятности экспоненциального закона распределения.

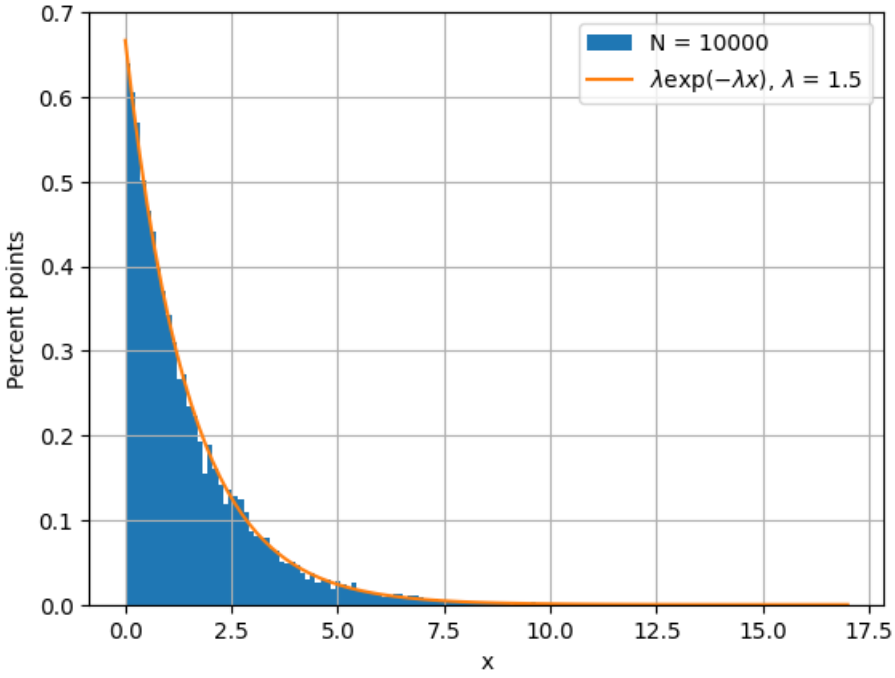
```
def task_3_4():
    scale = 1.5
    experiments = [10 ** 3, 10 ** 4, 10 ** 5, 10 ** 6]
    parts = 100

    x = np.linspace(0, 17, 100)

    for experiment in experiments:
        y = expon.ppf(
```

```
            [random.random() for _ in range(experiment)],
            scale=scale)
        print("First 10 elements of y:")
        display(y[:10])

        plt.hist(y, bins = parts, density = True, label = f'N = {experiment}')
        plt.plot(x, expon.pdf(x, scale=scale), label=f'$\lambda \exp(-\lambda x)$, $\lambda$
        plt.legend()
        plt.ylabel('Percent points')
        plt.xlabel('x')
        plt.grid(True)
        plt.show()

task_3_4()
```

```
First 10 elements of y:
array([0.26108809, 0.60051306, 1.79795743, 4.12244978, 1.36752087,
       1.46626561, 1.95951586, 1.11985539, 1.35231707, 1.36351018])
```

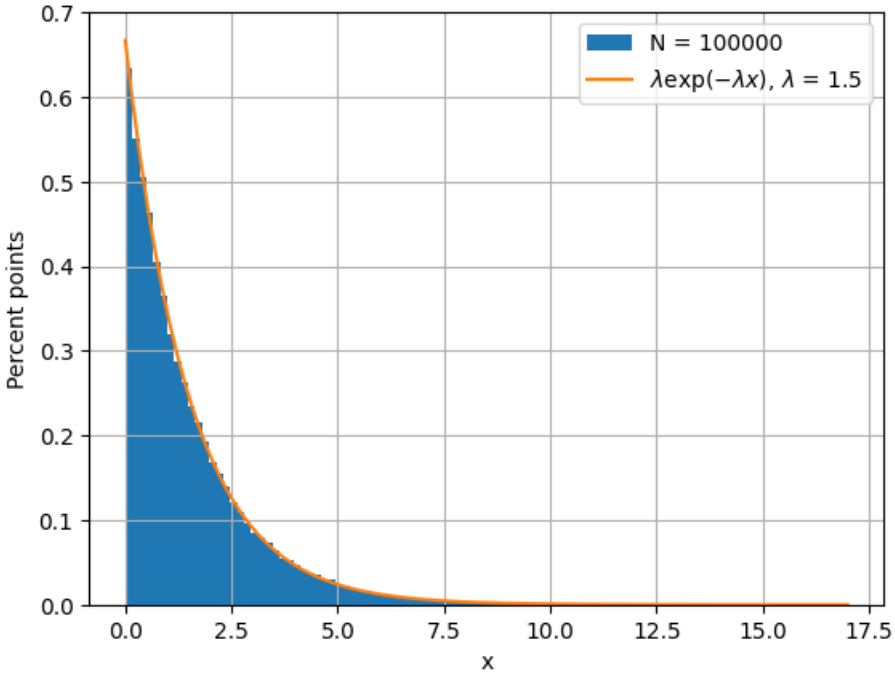



```
First 10 elements of y:
array([0.81159231, 2.07905584, 0.61256452, 2.42650309, 0.03075629,
       0.74296061, 0.78388096, 4.29866036, 3.08655963, 1.16727436])
```

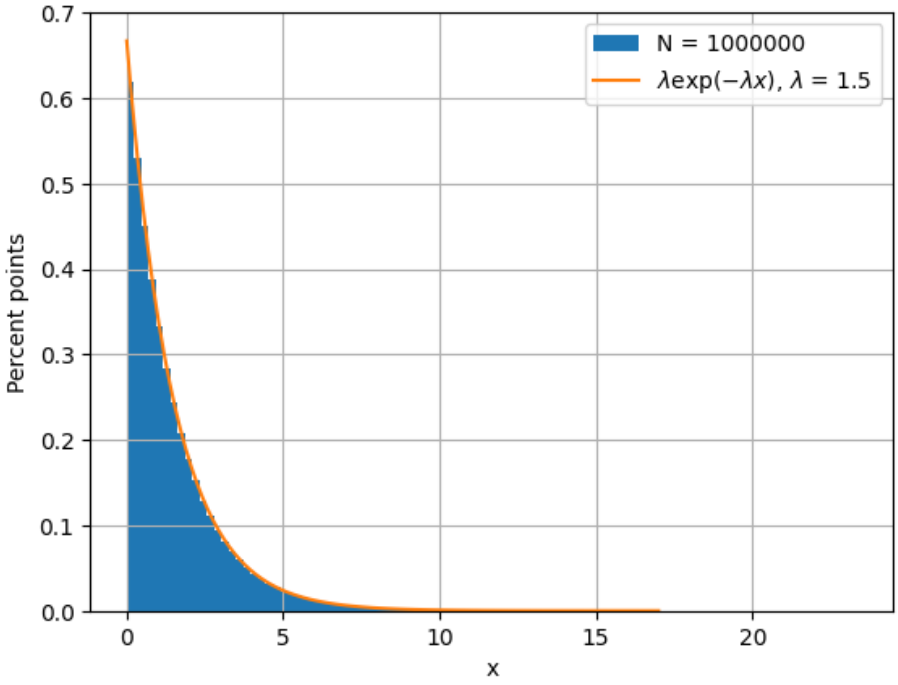



```
First 10 elements of y:
```

```
array([0.31101965, 1.58430649, 2.49273732, 1.62339486, 2.43480166,
       0.27990393, 5.20720268, 0.83669411, 3.39111316, 0.47951797])
```




```
First 10 elements of y:
array([1.62604451, 4.95935121, 0.76203212, 0.5022636 , 1.17191308,
       0.9612542 , 0.06307622, 4.81881513, 0.15361942, 0.03288727])
```


N = 1000000
$\lambda \exp(-\lambda x)$, λ = 1.5
Percent points
x
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0
5
10
15
20