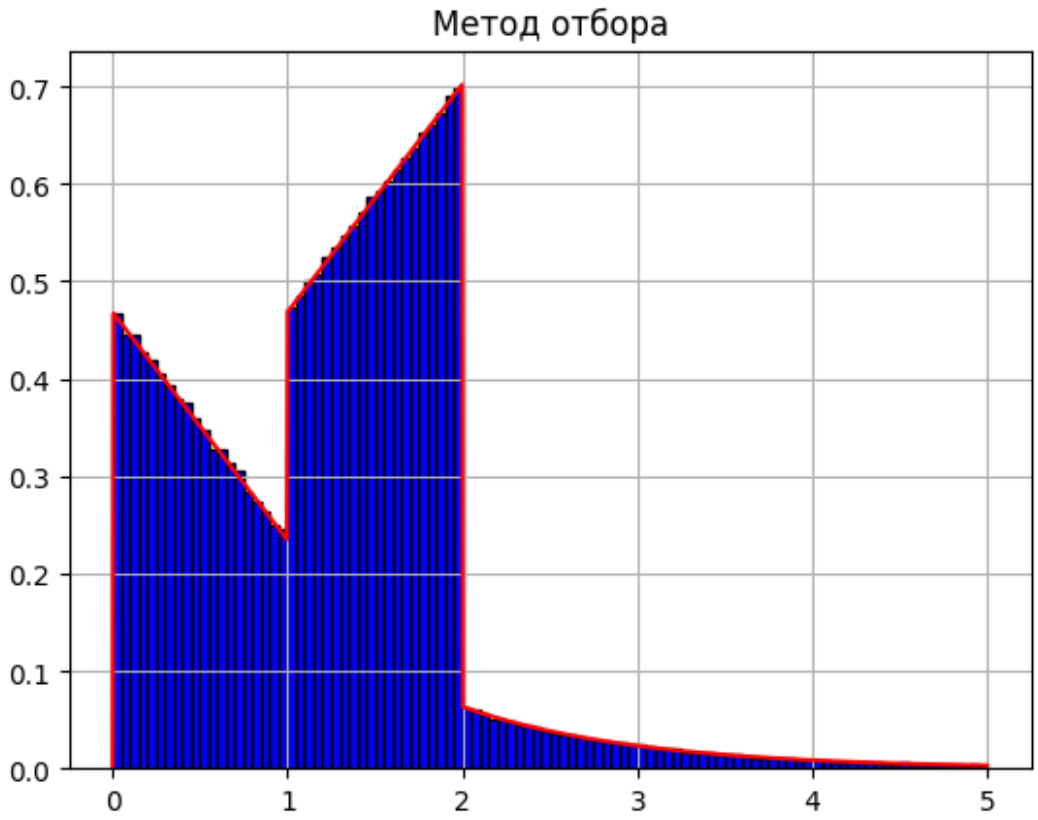
```python
import math
from random import random
import matplotlib.pyplot as plt
import numpy as np


n = 10**6
c = 1/(4+2*math.exp(-2))

def p(x):
    if(0<x and x<=1):
        return c*(2-x)
    elif(1 < x and x <= 2):
        return c*(1+x)
    elif 2<x:
        return 2*c*math.exp(-x)
    else:
        return 0

def Plot_Histogram(data, title):
    plt.hist(data, bins=100, density=True, range=(0,5), color='blue',
edgecolor='black')
    plt.title("Гистограмма")
    x_values = np.linspace(0, 5, 1000)
    y_values = [p(x) for x in x_values]
    plt.plot(x_values, y_values, color = 'red')
    plt.grid(True)
    plt.title(title)
    plt.show()

def Inverse_Transform_Method(n):
    d = [3/2*c, 4*c]
    ci = c**(-1)
    xi = []
    for i in range(n):
        alpha = random()
        if(alpha < d[0]):
            xi.append(2-math.sqrt(4-2*alpha*ci))
        elif(alpha < d[1]):
            xi.append(-1 + math.sqrt(1+2*alpha*ci))
        else:
            xi.append(-math.log((1-alpha)*ci/2))
    return xi

xi = Inverse_Transform_Method(n)
Plot_Histogram(xi, "Метод обратных функций")
```


```
def Decomposition_Method(n):
    q = [3/2*c, 4*c]
    xi = []
    for i in range(n):
        alpha_1, alpha_2 = random(), random()
        if(alpha_1 < q[0]):
            xi.append(2-math.sqrt(4-3*alpha_2))
        elif(alpha_1 < q[1]):
            xi.append(-1 + math.sqrt(4+5*alpha_2))
        else:
            xi.append(2 - math.log(1-alpha_2))
    return xi

xi = Decomposition_Method(n)
Plot_Histogram(xi, "Метод декомпозиции")
```


```
def Selection_Method(n):
    M_1 = 1/(3*math.exp(2))
    M_2 = 2*M_1
    b_1 = math.exp(-1)
    b_2 = b_1*b_1
    xi = []
    for i in range(n):
        while 1:
            alpha_1, alpha_2 = random(), random()
            if (b_1 < alpha_1) and (alpha_2*alpha_1 <
(2+math.log(alpha_1))*M_1):
               xi.append(-math.log(alpha_1))
               break
            if (b_2 < alpha_1 <= b_1) and (alpha_1*alpha_2 < (1-
math.log(alpha_1))*M_1):
                xi.append(-math.log(alpha_1))
                break
            if (alpha_1 <= b_2) and (alpha_2 < M_2):
                xi.append(-math.log(alpha_1))
                break
    return xi

xi = Selection_Method(n)
Plot_Histogram(xi, "Метод отбора")
```


Метод отбора
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
0
1
2
3
4
5