

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Кафедра прикладної математики

Лабораторна робота №1
з кредитного модуля «Випадкові процеси»
Варіант 8

Виконав:

студент групи КМ-81

Донченко Богдан Миколайович

Перевірів викладач:

Пашко Анатолій Олексійович

Київ – 2020

Завдання 1

Умова(1)

1. Згенерувати послідовність з $n = 1000000$ псевдовипадкових чисел, що рівномірно розподілені на інтервалі $(0,1)$ (використати вбудований генератор псевдовипадкових чисел). Побудувати графік.

1.1. Оцінити математичне сподівання та дисперсію отриманої послідовності.

1.2. Побудувати таблицю 1 (кількість L підінтервалів не менше 10), частотну таблицю вивести на екран.

Таблиця 1. Частотна таблиця

Інтервал	Кількість чисел (частота попадань), які випали в даний інтервал	Відносна частота потрапляння
Δ_1	v_1	v_1/n
Δ_2	v_2	v_2/n
...
Δ_L	v_L	v_L/n
	\sum кіл-ть	

1.3. Перевірити гіпотезу про закон розподілу, побудувати гістограму.

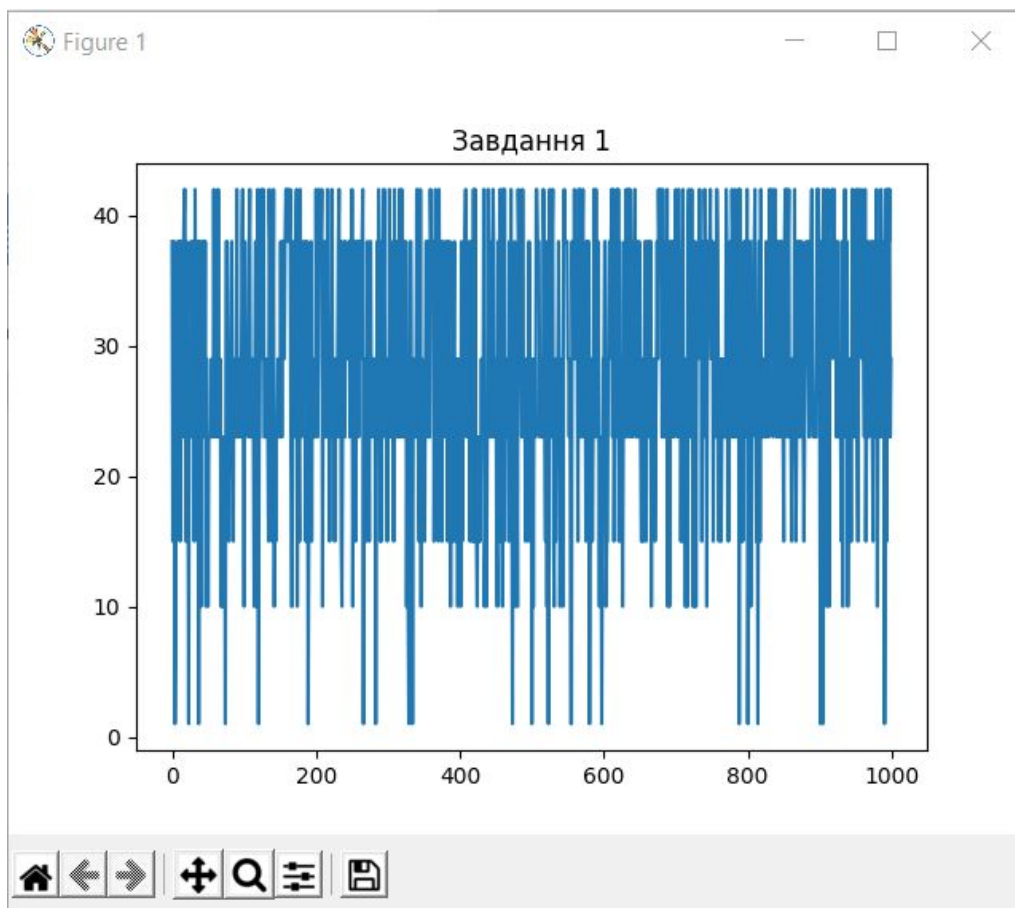
1.4. Побудувати гістограму розподілу випадкової величини $\eta = \max(\xi)$, де ξ - рівномірно розподілена на $(0,1)$ випадкова величина:

а) змодельовати $n = 1000000$ значень величини ξ , знайти максимальне значення $\eta_1 = \max(\xi)$;

б) процедуру а) повторити $m=1000$;

в) для випадкової величини $\eta = \{\eta_i, i = 1, \dots, m\}$ побудувати гістограму.

Графіки та таблиці(1)



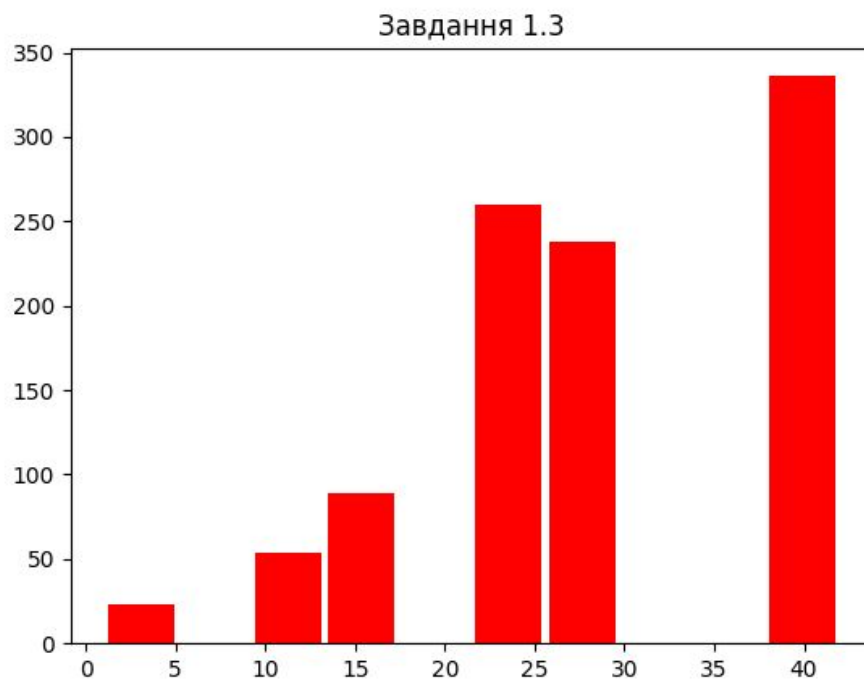
Завдання 1

Мат. ожидание: 28.068

Дисперсия: 80.73138

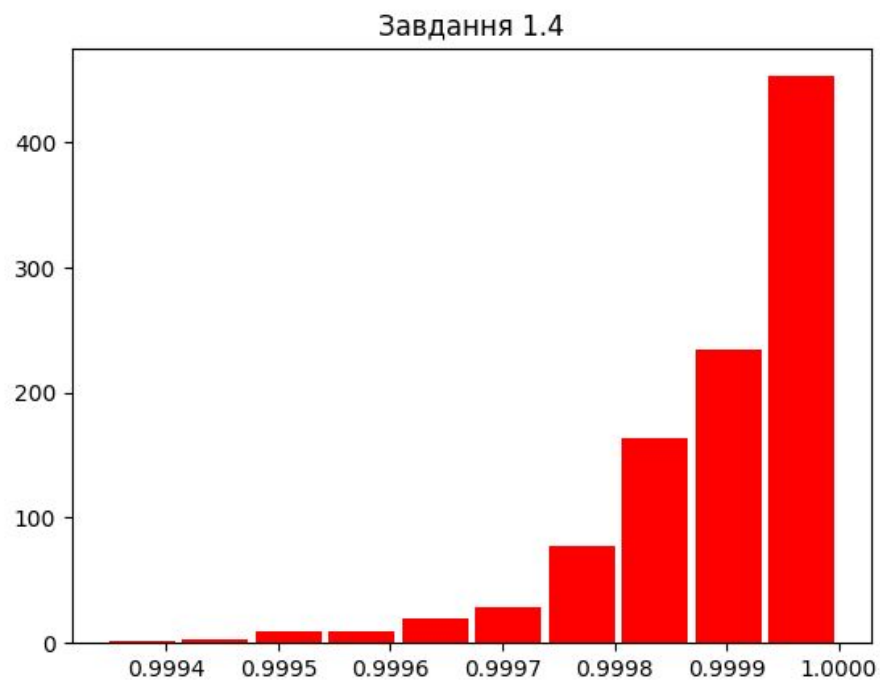
Інтервали	Кількість	Відносна частота
(0.0, 0.1)	100150	0.1002
(0.1, 0.2)	100063	0.1001
(0.2, 0.3)	100412	0.1004
(0.3, 0.4)	100164	0.1002
(0.4, 0.5)	99900	0.0999
(0.5, 0.6)	99760	0.0998
(0.6, 0.7)	100335	0.1003
(0.7, 0.8)	99945	0.0999
(0.8, 0.9)	100025	0.1
(0.9, 1.0)	99246	0.0992
Сума:	1000000	

Figure 1



x=40.67 y=291.3

Figure 1



x=1.000000 y=323.

Код(1)

```
r_arr = np.random.rand(1000000)
head_arr = ["Інтервали", "Кількість", "Відносна частота"]
def task_1():
    graph(r_arr, "Завдання 1")
def task_1_1():
    exp_dis(r_arr)
def task_1_2():
    print_table(get_interval(r_arr, 10), head_arr)
def task_1_3():
    histogramm(r_arr, 10, 0.9, "Завдання 1.3")
def task_1_4():
    m = 1000
    a = 0
    max_arr = []
    while a < m:
        arr = np.random.rand(10000)
        mx = np.max(arr)
        max_arr.append(mx)
        a += 1
    histogramm(np.array(max_arr), 10, 0.9, "Завдання 1.4")

def graph(array, name):
    title = plt.title(name)
    plot = plt.plot(array)
    plt.show()

def exp_dis(array):
    exp_value = np.mean(array)
    print("Мат. ожидание: ", round(exp_value, 5))
    dispersion = np.var(array)
    print("Дисперсия: ", round(dispersion, 5))
    print("")
```

```

def get_interval(r_array, intervals):
    (element, counts) = np.unique(r_array, return_counts=True)
    result = []
    a = np.min(element)
    step = (np.max(element) - np.min(element))/intervals
    b = a+step+0.0001
    n = 0
    for i in range(1, intervals + 1):
        count = counts[(a <= element) & (b >= element)]
        quantity = np.sum(count)
        n += quantity
        frequency = quantity/len(r_array)
        result.append((a, b, quantity, frequency))
        a = b
        b = b+step+0.0001
    return np.array(result)

def print_table(table_arr, table_header):
    table = [table_header]
    table_arr[2] = np.round(table_arr[2], 4)
    for i in table_arr:
        interval = "({}, {})".format(round(i[0], 2), round(i[1], 2))
        table.append([interval, int(i[2]), round(i[3], 4)])
    summa = "Cyma: {}".format(int(np.sum(table_arr.T[2])))
    table.append(["", summa, ""])
    print(tabulate(table))

def histogramm(array, n, w, name):
    title = plt.title(name)
    plt.hist(array, n, rwidth=w, color='red')
    plt.show()

```


Завдання 2

Умова(2)

2. Змодельовати дискретну випадкову величину, задану таблицею 2, побудувати графік.

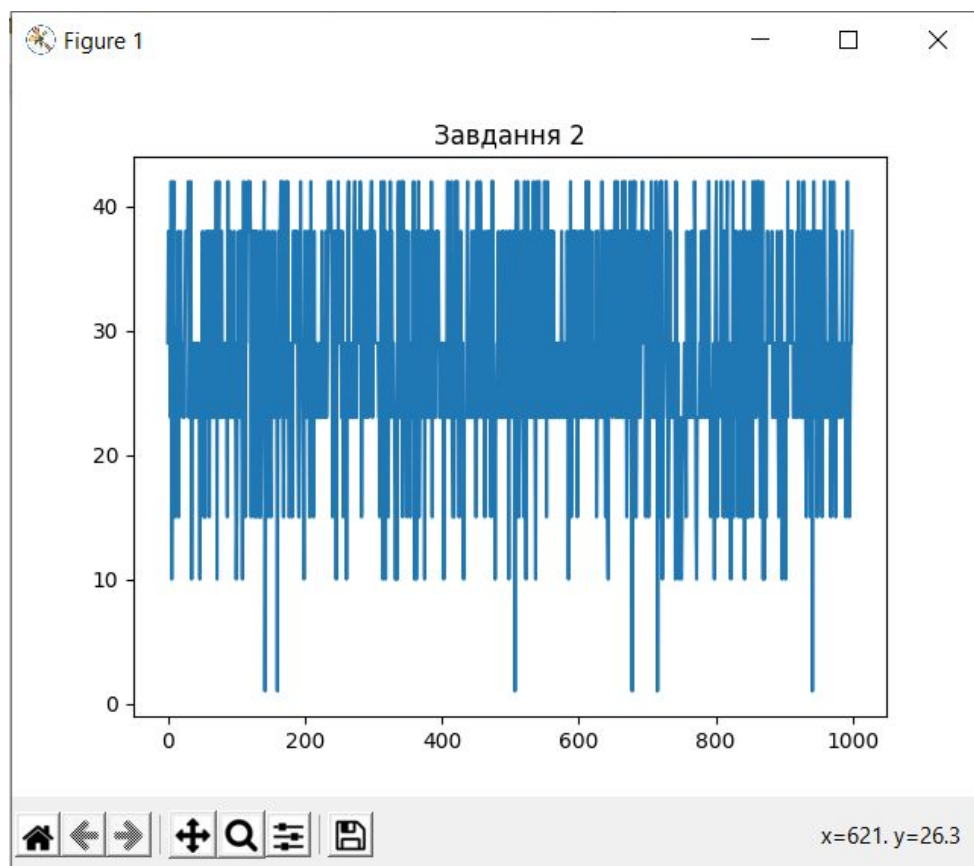
2.1.Оцінити математичне сподівання та дисперсію отриманої дискретної випадкової величини.

2.2. Побудувати частотну таблицю.

2.3. Перевірити гіпотезу про закон розподілу, побудувати гістограму.

8	x_i	1	10	15	23	29	38	42
	p_i	0.02	0.05	0.1	0.28	0.23	0.22	0.1

Графіки та таблиці(2)



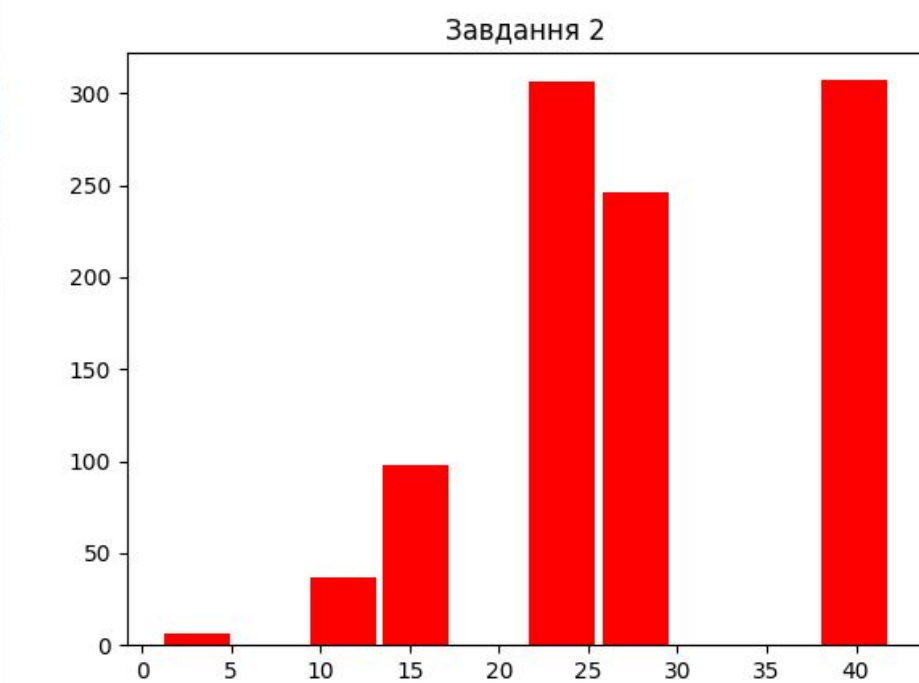
Завдання 2

Мат. ожидание: 28.068

Дисперсия: 80.73138

Интервали	Кількість	Відносна частота
1.0	6.0	0.006
10.0	37.0	0.037
15.0	98.0	0.098
23.0	306.0	0.306
29.0	246.0	0.246
38.0	211.0	0.211
42.0	96.0	0.096
Сума: 1000		

Figure 1



x=28.88 y=97.0

Код (2)

```
x = np.array([1, 10, 15, 23, 29, 38, 42])
p = np.array([0.02, 0.05, 0.1, 0.28, 0.23, 0.22, 0.1])
r_arr = np.random.choice(x, 1000, p=p)
def task_2():
    graph(r_arr, "Завдання 2")
def task_2_1():
    exp_dis(r_arr)
def task_2_2():
    print_dis_table(get_dis_interval(r_arr), head_arr)
def task_2_3():
    histogramm(r_arr, 10, 0.9, "Завдання 2")

def get_dis_interval(r_array):
    (element, counts) = np.unique(r_array, return_counts=True)
    result = []
    for i, j in zip(element, counts):
        result.append([i, j, j / len(r_array)])
    return np.array(result)

def print_dis_table(table_arr, table_header):
    table = [table_header]
    table.extend(table_arr)
    summa = "Сума: {}".format(int(np.sum(table_arr.T[1])))
    table.append(["", summa, ""])
    print(tabulate(table))
```

Завдання 3

Умова(3)

3. Моделювання гауссівського розподілу за алгоритмами:

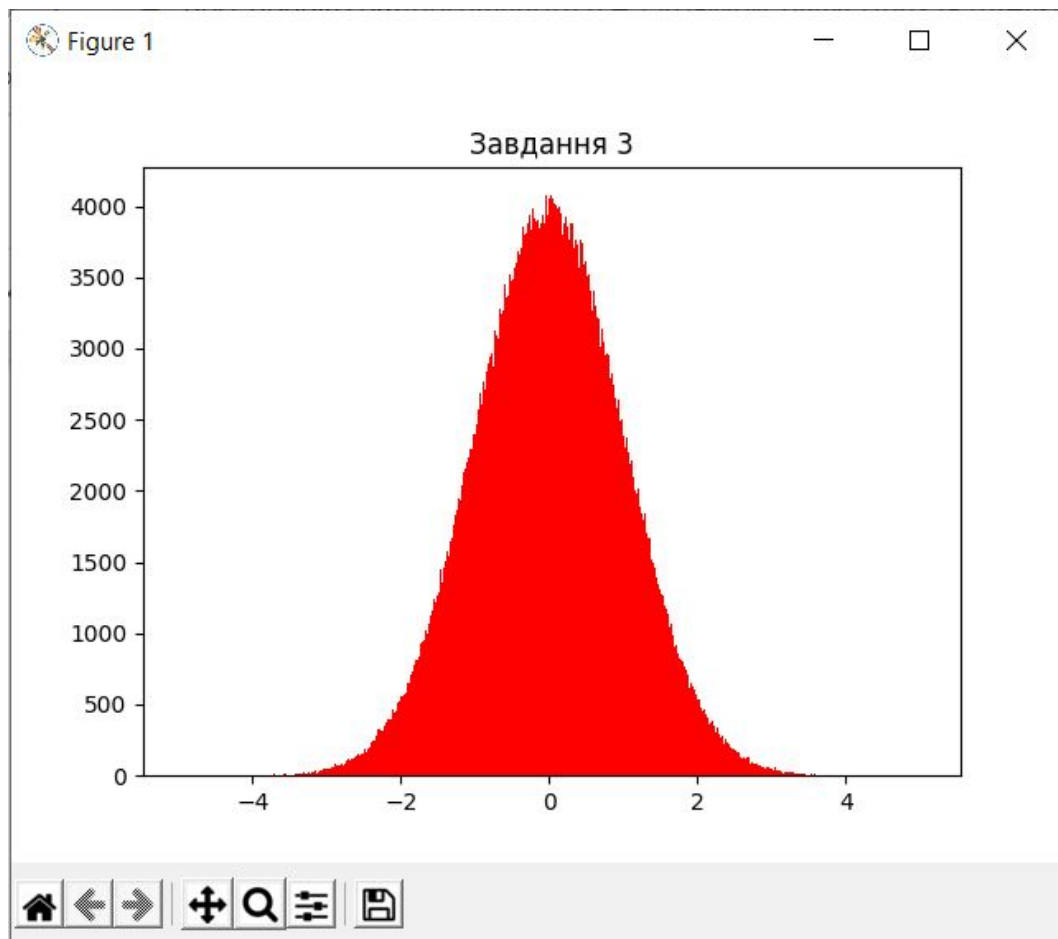
a) $\xi_1 = \sqrt{-2\ln(\alpha_1)} \cos(2\pi\alpha_2), \quad \xi_2 = \sqrt{-2\ln(\alpha_1)} \sin(2\pi\alpha_2),$

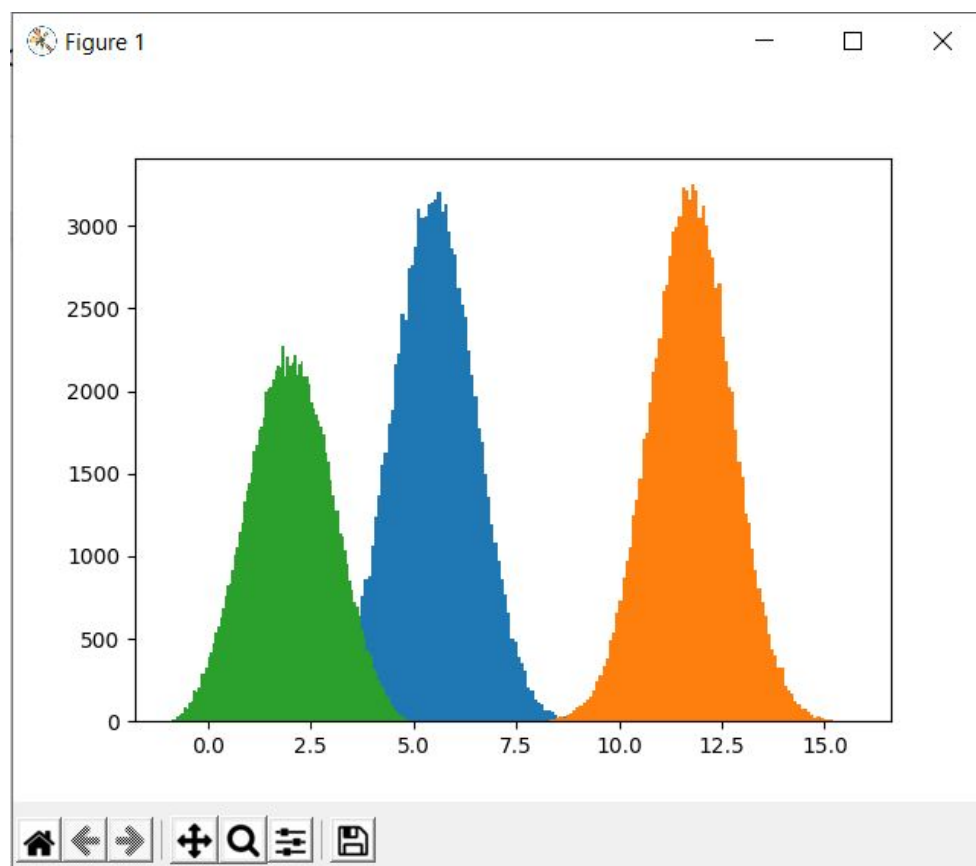
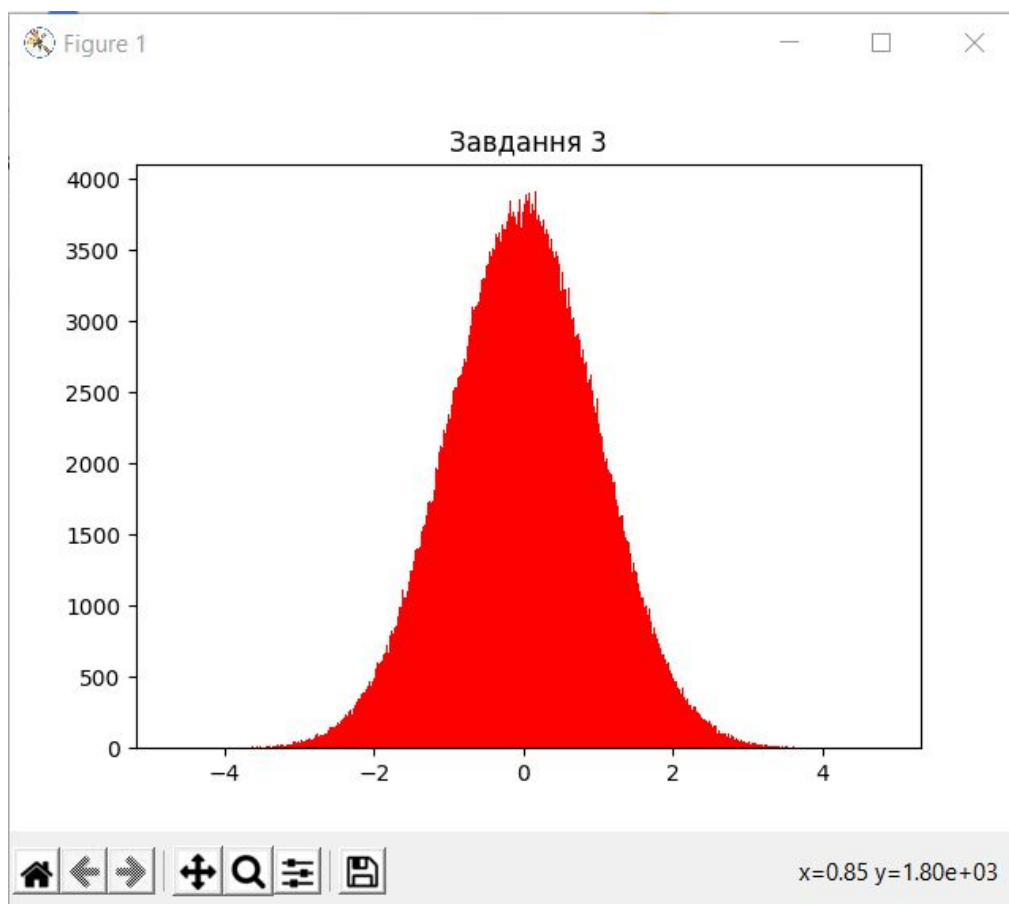
b) $\xi = \sqrt{\frac{12}{n}} \sum_{i=1}^n (\alpha_i - 0.5),$

α_i - рівномірно розподілені в інтервалі (0,1).

Перевірити закон розподілу, в алгоритмі b) перевірити закон розподілу для різних n ($n=12, n=48, n=3$).

Графіки(3)





Код(3)

```
def task_3_a1():
    a1 = np.random.rand(1000000)
    a2 = np.random.rand(1000000)
    formula = np.sqrt(-2*np.log(a1))*np.cos(2*np.pi*a2)
    histogramm(formula, 1000, 1, "Завдання 3")
def task_3_a2():
    a1 = np.random.rand(1000000)
    a2 = np.random.rand(1000000)
    formula = np.sqrt(-2*np.log(a1))*np.sin(2*np.pi*a2)
    histogramm(formula, 1000, 1, "Завдання 3")
def task_3_b():
    n1, n2, n3 = 12, 48, 3
    formula1 = [np.sqrt(12/n1)*(np.sum(np.random.rand(n1))-0.5) for i in range(100000)]
    graf1=np.array(formula1)
    formula2 = [np.sqrt(12/n2)*(np.sum(np.random.rand(n2))-0.5) for i in range(100000)]
    graf2=np.array(formula2)
    formula3 = [np.sqrt(12/n3)*(np.sum(np.random.rand(n3))-0.5) for i in range(100000)]
    graf3=np.array(formula3)
    g1 = plt.hist(graf1, 100)
    g2 = plt.hist(graf2, 100)
    g3 = plt.hist(graf3, 100)
    plt.show()
```

Завдання 4

Умова(4)

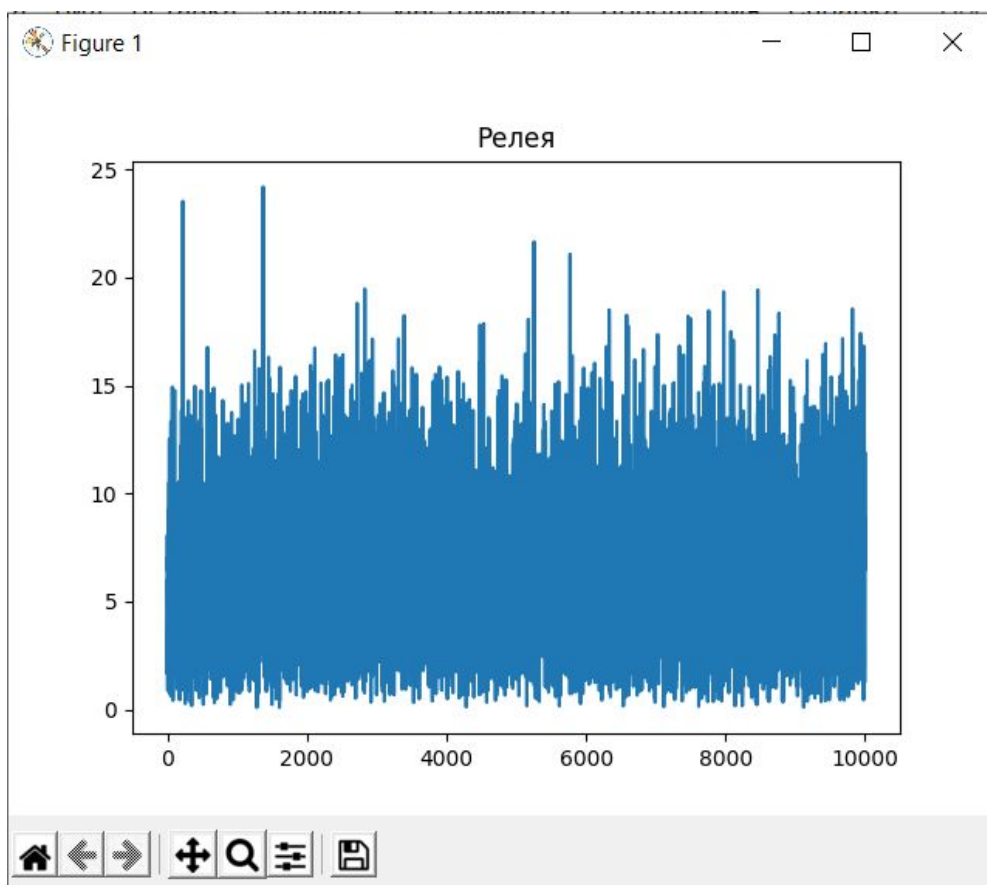
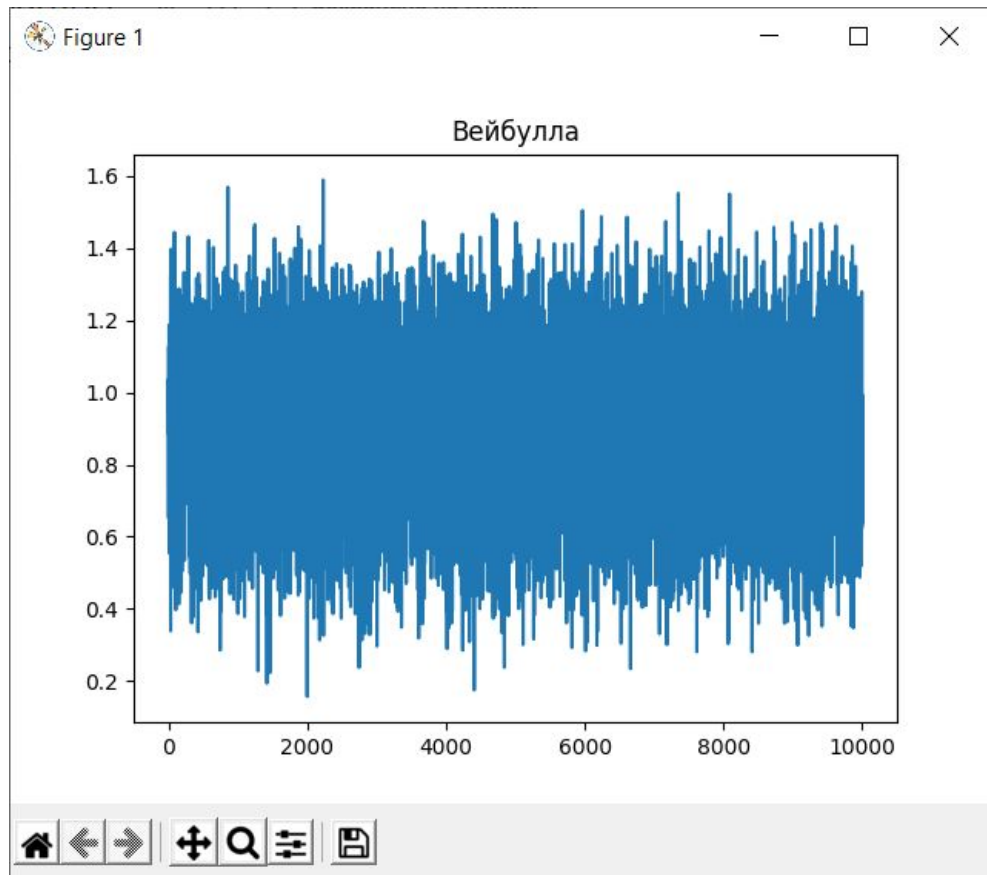
4. Змодельовати неперервні випадкові величини, що мають закони розподілу Вейбулла, Релея, логнормальний, Коші, Накагамі. Побудувати графіки.

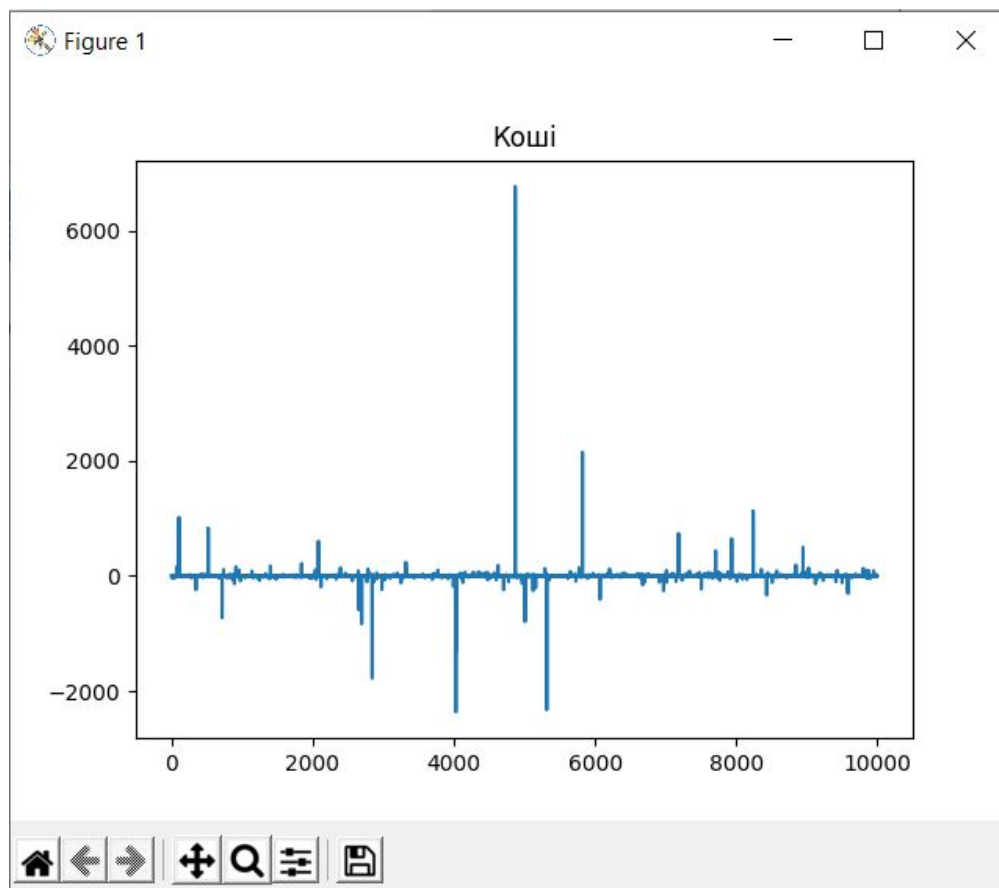
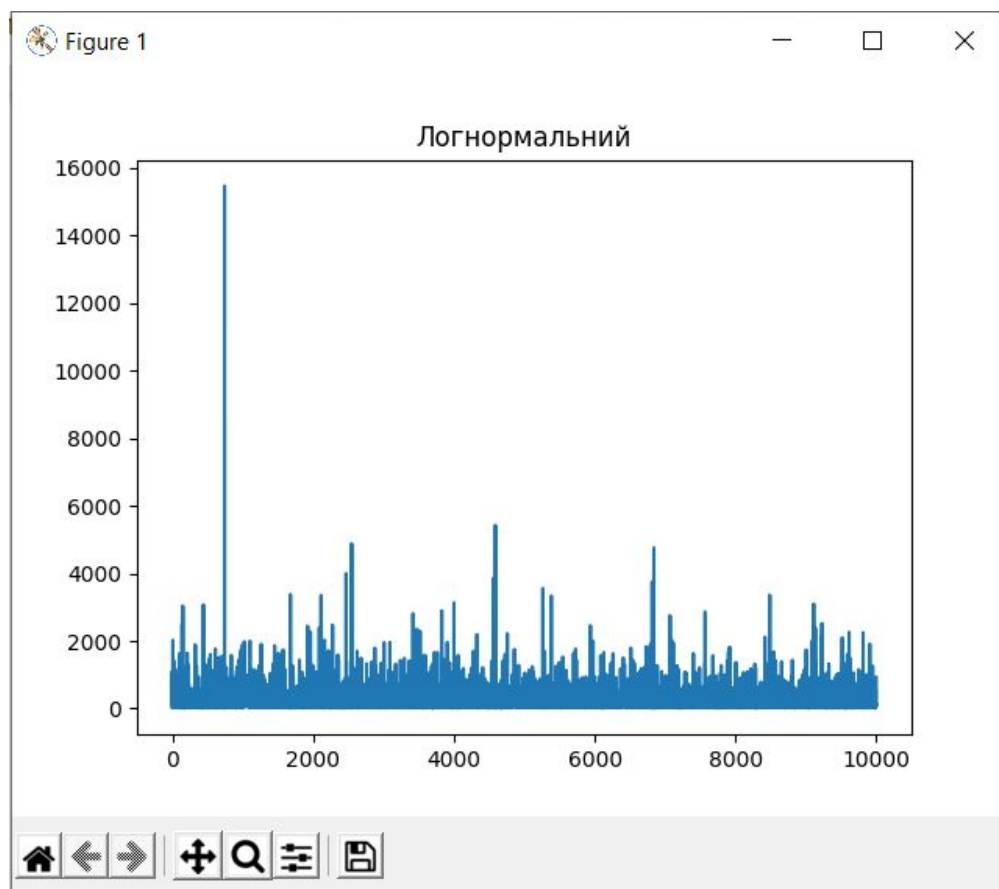
4.1.Оцінити математичне сподівання та дисперсію отриманих випадкових величин.

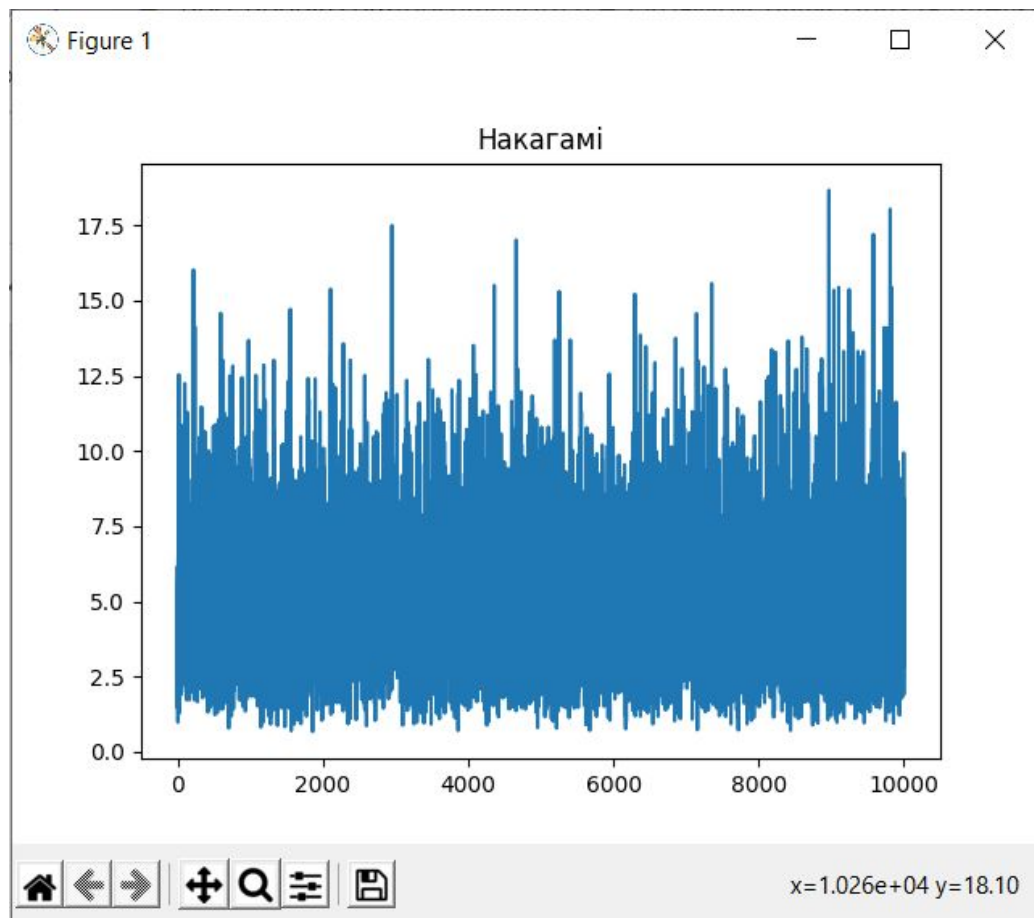
4.2. Побудувати частотну таблицю для кожної реалізації.

4.3. Перевірити гіпотезу про закон розподілу, побудувати гістограму.

Графіки і таблиці(4)







Завдання 4

Вейбулла:

Мат. ожидание: 0.91516

Дисперсия: 0.04387

Релея:

Мат. ожидание: 6.26144

Дисперсия: 10.73738

Логнормальный:

Мат. ожидание: 246.39938

Дисперсия: 119538.67825

Коші:

Мат. ожидание: 0.23038

Дисперсия: 7552.08244

Накагамі:

Мат. ожидание: 4.99465

Дисперсия: 5.026

Вейбулла:

Інтервали	Кількість	Відносна частота
(0.16, 0.31)	26	0.0026
(0.31, 0.45)	152	0.0152
(0.45, 0.6)	564	0.0564
(0.6, 0.75)	1304	0.1304
(0.75, 0.89)	2236	0.2236
(0.89, 1.04)	2738	0.2738
(1.04, 1.18)	1977	0.1977
(1.18, 1.33)	821	0.0821
(1.33, 1.48)	174	0.0174
(1.48, 1.62)	8	0.0008
Сума: 10000		

Релея:

Інтервали	Кількість	Відносна частота
(0.05, 2.1)	876	0.0876
(2.1, 4.15)	2034	0.2034
(4.15, 6.21)	2482	0.2482
(6.21, 8.26)	2049	0.2049
(8.26, 10.31)	1388	0.1388
(10.31, 12.37)	701	0.0701
(12.37, 14.42)	300	0.03
(14.42, 16.47)	125	0.0125
(16.47, 18.53)	35	0.0035
(18.53, 20.58)	10	0.001
Сума: 10000		

Логнормальний:

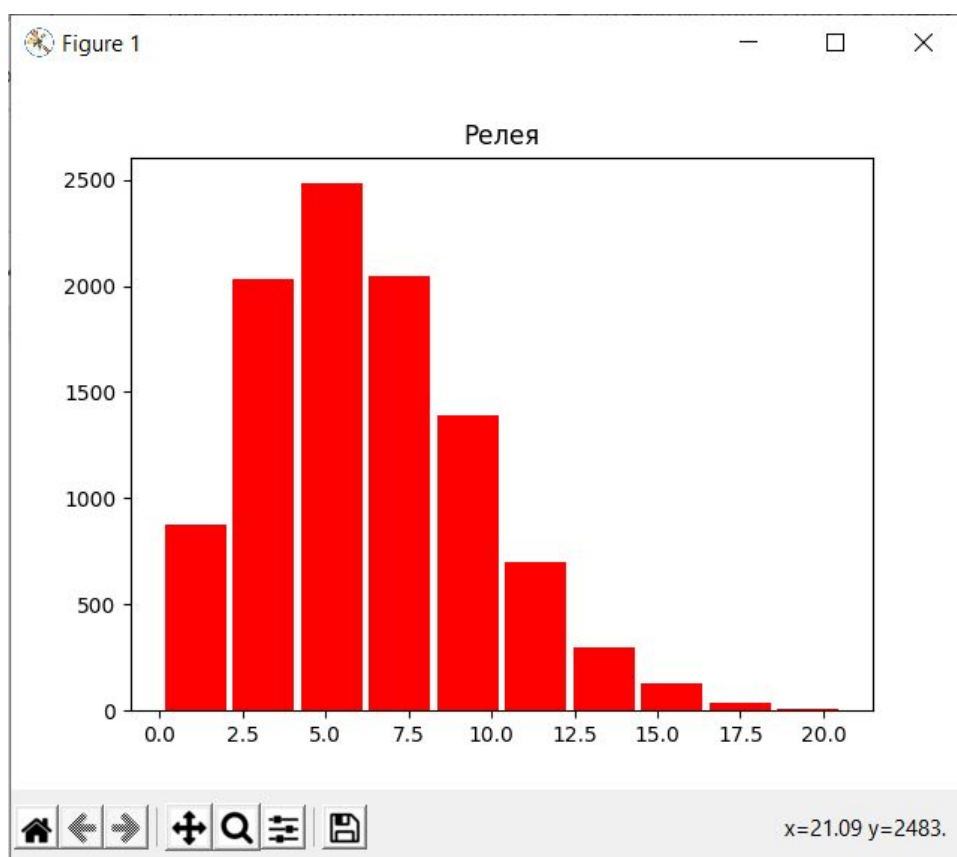
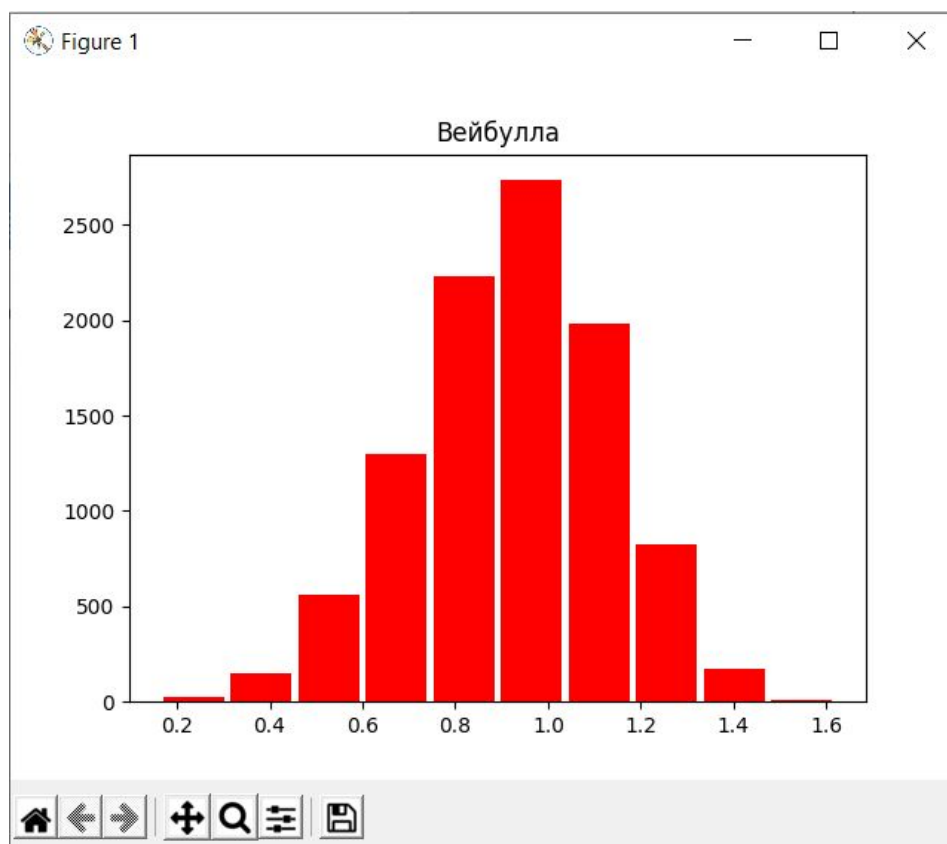
Інтервали	Кількість	Відносна частота
(3.99, 581.3)	9153	0.9153
(581.3, 1158.62)	643	0.0643
(1158.62, 1735.94)	128	0.0128
(1735.94, 2313.26)	44	0.0044
(2313.26, 2890.58)	15	0.0015
(2890.58, 3467.9)	11	0.0011
(3467.9, 4045.22)	2	0.0002
(4045.22, 4622.54)	3	0.0003
(4622.54, 5199.86)	0	0.0
(5199.86, 5777.17)	1	0.0001
Сума: 10000		

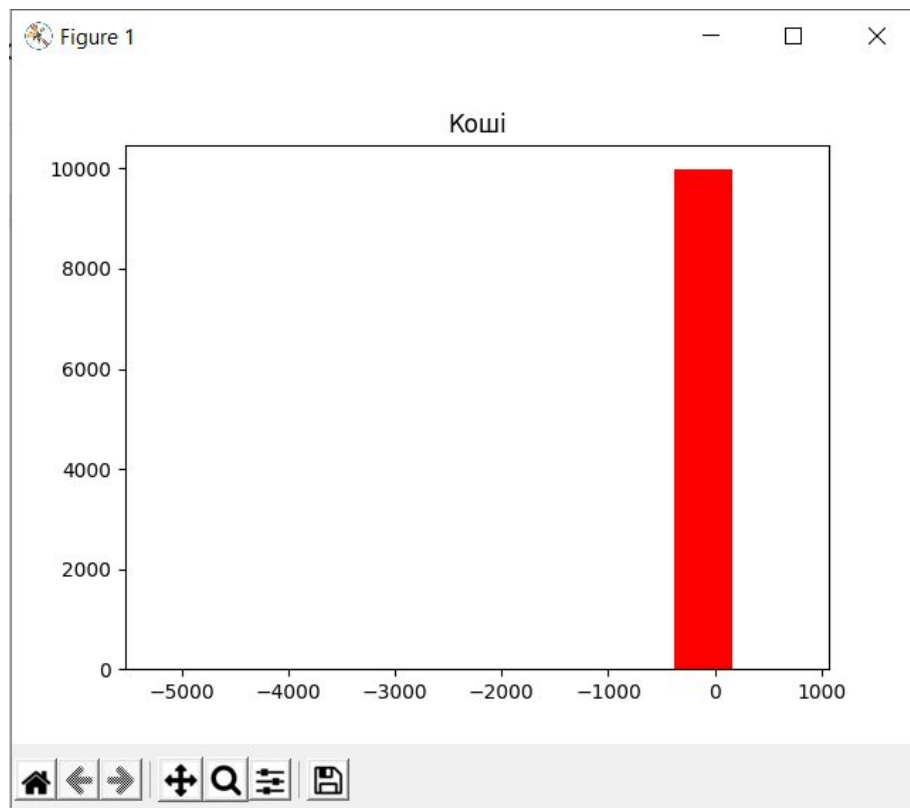
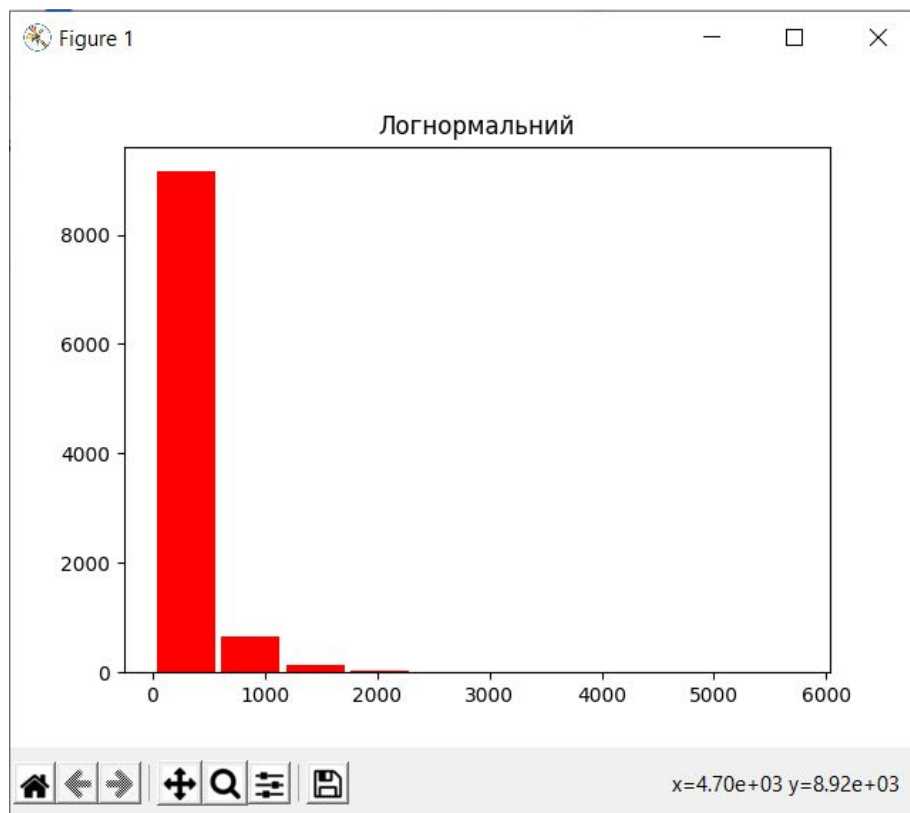
Коші:

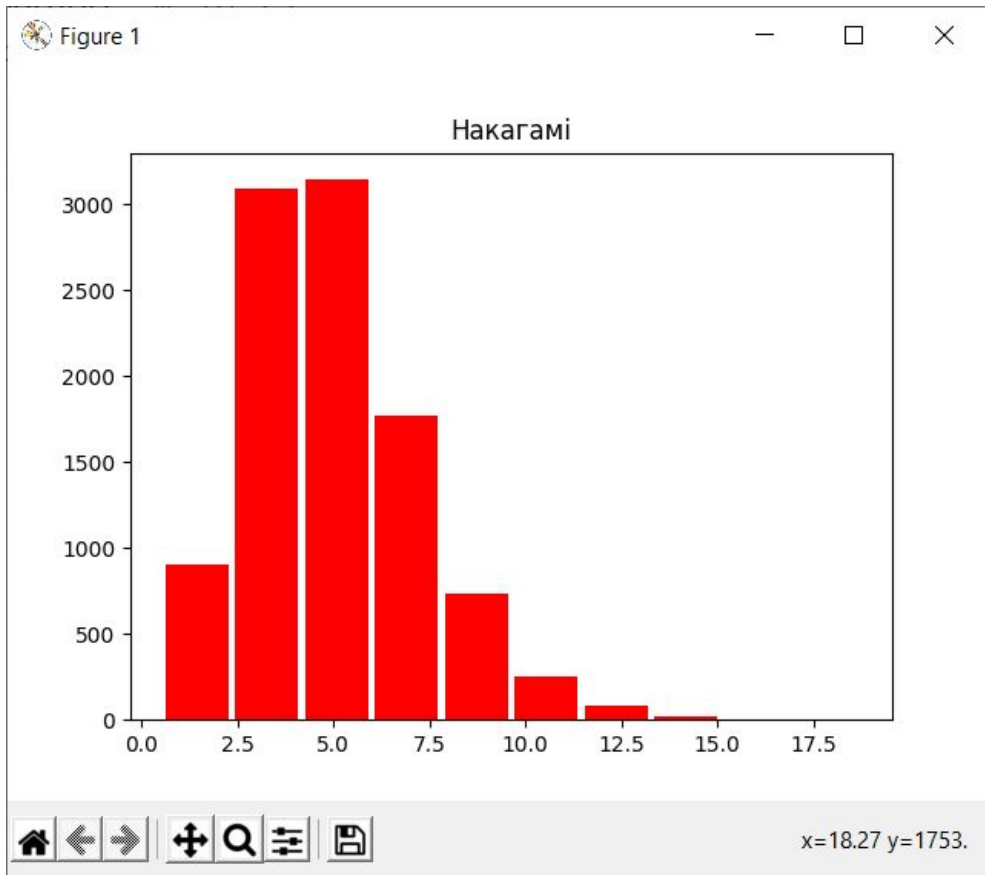
Інтервали	Кількість	Відносна частота
(-5259.62, -4653.64)	1	0.0001
(-4653.64, -4047.65)	0	0.0
(-4047.65, -3441.66)	0	0.0
(-3441.66, -2835.68)	0	0.0
(-2835.68, -2229.69)	0	0.0
(-2229.69, -1623.71)	0	0.0
(-1623.71, -1017.72)	3	0.0003
(-1017.72, -411.74)	4	0.0004
(-411.74, 194.25)	9977	0.9977
(194.25, 800.24)	15	0.0015
Сума: 10000		

Накагамі:

Інтервали	Кількість	Відносна частота
(0.53, 2.35)	904	0.0904
(2.35, 4.17)	3095	0.3095
(4.17, 5.99)	3145	0.3145
(5.99, 7.82)	1776	0.1776
(7.82, 9.64)	730	0.073
(9.64, 11.46)	249	0.0249
(11.46, 13.28)	82	0.0082
(13.28, 15.1)	16	0.0016
(15.1, 16.92)	2	0.0002
(16.92, 18.74)	1	0.0001
Сума: 10000		







Код (4)

```
wei_arr = np.random.weibull(5, 10000)
ray_arr = np.random.rayleigh(5, 10000)
lognorm_arr = np.random.lognormal(5, 1, 10000)
cauchy_arr = np.random.standard_cauchy(10000)
nak_arr = np.random.standard_gamma(5, 10000)
def task_4():
    graph(wei_arr, "Вейбулла")
    graph(ray_arr, "Релея")
    graph(lognorm_arr, "Логнормальный")
    graph(cauchy_arr, "Коші")
    graph(nak_arr, "Накагамі")
def task_4_1():
    print("Вейбулла:")
    exp_dis(wei_arr)
    print("Релея:")
    exp_dis(ray_arr)
    print("Логнормальный:")
    exp_dis(lognorm_arr)
    cauchy = np.array(cauchy_arr)
    print("Коші:")
    exp_dis(cauchy_arr)
    print("Накагамі:")
    exp_dis(nak_arr)
def task_4_2():
    print("Вейбулла:")
    print_table(get_interval(wei_arr, 10), head_arr)
    print("Релея:")
    print_table(get_interval(ray_arr, 10), head_arr)
    print("Логнормальный:")
    print_table(get_interval(lognorm_arr, 10), head_arr)
    print("Коші:")
    print_table(get_interval(cauchy_arr, 10), head_arr)
    print("Накагамі:")
    print_table(get_interval(nak_arr, 10), head_arr)
def task_4_3():
    histogramm(wei_arr, 10, 0.9, "Вейбулла")
    histogramm(ray_arr, 10, 0.9, "Релея")
    histogramm(lognorm_arr, 10, 0.9, "Логнормальный")
    histogramm(cauchy_arr, 10, 0.9, "Коші")
    histogramm(nak_arr, 10, 0.9, "Накагамі")
```

Завдання 5

Умова(5)

5. Обчислити методом Монте-Карло визначені інтеграли

$$\int_0^1 (x^7 + x^5 + x^3) dx, \quad \int_0^{\pi} 2 \sin(3x) dx, \quad \int_0^{\infty} \frac{1}{(x+1)\sqrt{x}} dx.$$

Пояснити отримані результати.

Рішення(5)

```
Завдання 5
Інтеграл 1:
0.5549229136214642

Інтеграл 2:
1.2974858862890033

Інтеграл 3:
0.28892952759754376
```

Код(5)

```
def function_1(x):
    return(x**7 + x**5 + x**3)
def function_2(x):
    return(2*np.sin(3*x))
def function_3(x):
    return(1/((x+1)*np.sqrt(x)))

def task_5():
    print("Інтеграл 1:")
    integral_1 = monte_carlo(function_1, 0, 1, 100)
    print(integral_1)
    print("\nІнтеграл 2:")
    integral_2 = monte_carlo(function_2, 0, np.pi, 100)
    print(integral_2)
    print("\nІнтеграл 3:")
    integral_3 = monte_carlo(function_3, 0, 2000, 100)
    print(integral_3)
```

Завдання 6

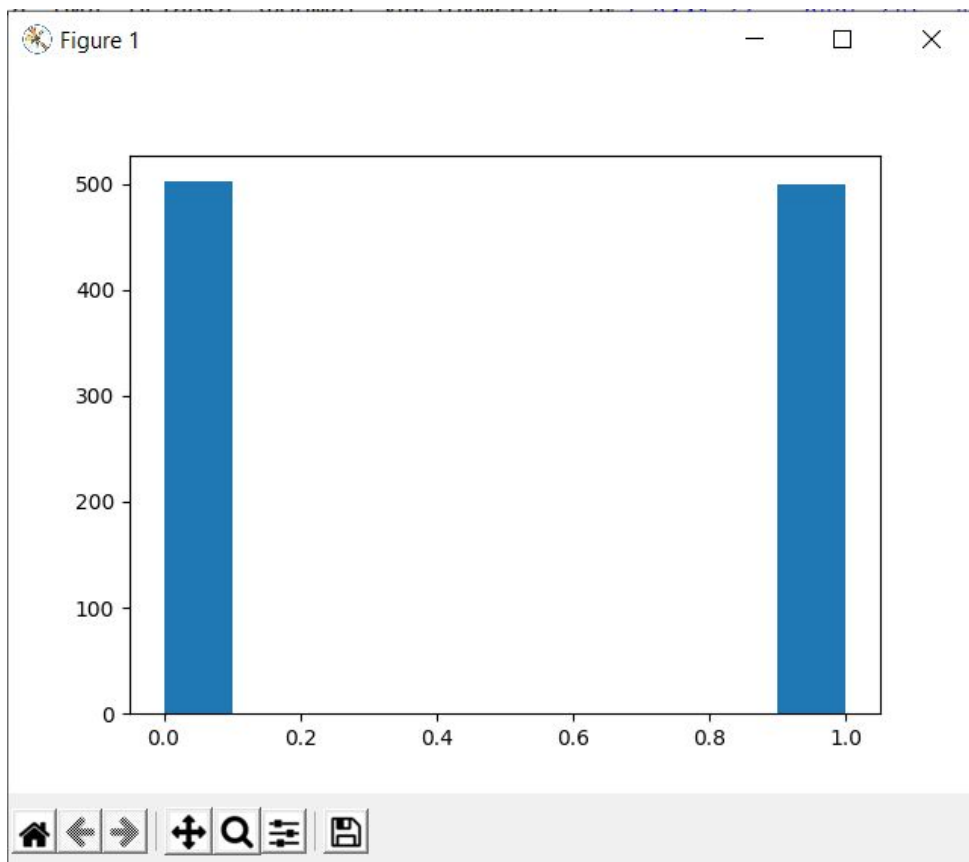
Умова(6)

6. Змодельовати випадкову двійкову послідовність за формулою

$$\beta_i = \begin{cases} 1, & \xi_{i+1} - \xi_i > 0 \\ 0, & \xi_{i+1} - \xi_i \leq 0 \end{cases}$$

Знайти емпіричний закон розподілу двійкової послідовності.

Рішення(6)



Код(6)

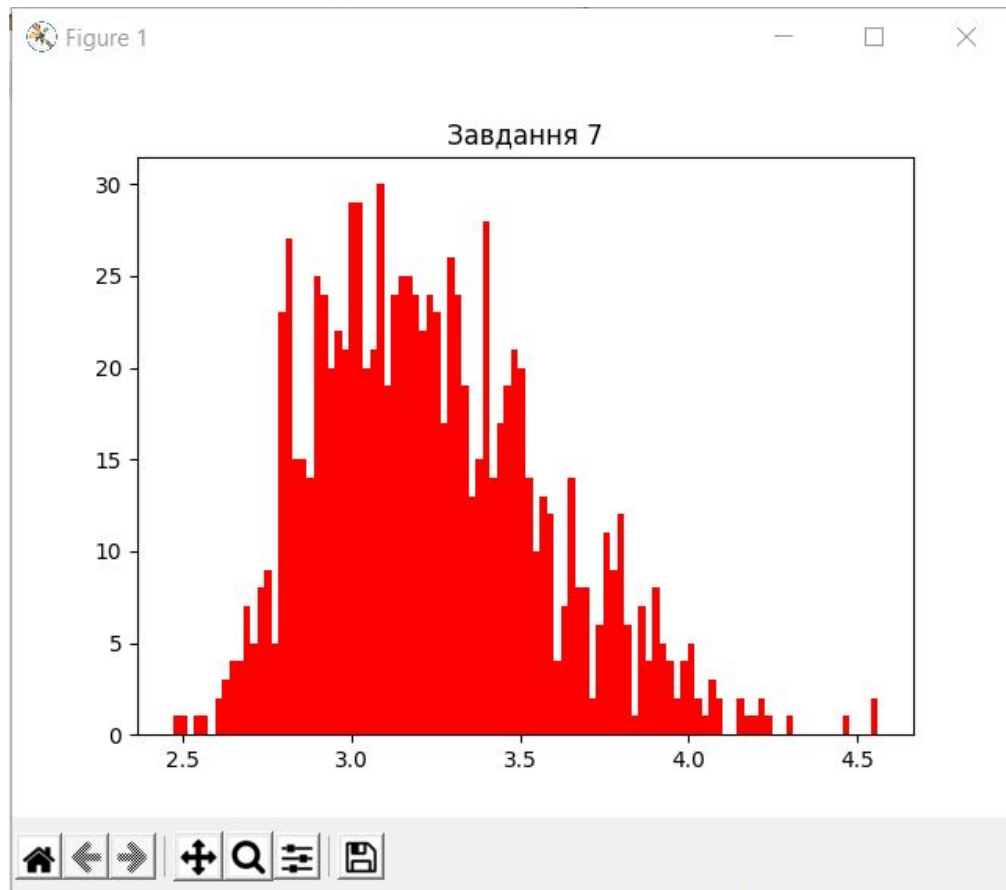
```
def task_6():
    n = 1001
    r_arr = np.random.rand(n)
    new_arr = []
    for i in range(n):
        if r_arr[i-1] < r_arr[i]:
            new_arr.append(1)
        elif r_arr[i-1] >= r_arr[i]:
            new_arr.append(0)
    #arr = new_arr.remove(new_arr[0])
    plt.hist(np.array(new_arr))
    plt.show()
```

Завдання 7

Умова(7)

7. Побудувати гістограму розподілу випадкової величини $\eta = \max(\xi)$, де ξ - випадкова величина, що має нормальний закон розподілу $N(0,1)$.

Рішення(7)



Код(7)

```
def task_7():  
    m = 1000  
    a = 0  
    max_arr = []  
    for i in range(m):  
        arr = np.random.normal(0, 1, 1000)  
        mx = np.max(arr)  
        max_arr.append(mx)  
    histogramm(np.array(max_arr), 100, 1, "Завдання 7")
```