

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и технологий
Высшая школа программной инженерии

Лабораторная работа
«ПОЛУЧЕНИЕ БАЗОВОЙ ПОСЛЕДОВАТЕЛЬНОСТИ
ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ И ТЕСТОВЫЕ ПРОВЕРКИ ЕГО
РАБОТЫ»

Выполнил
студент гр. 3530904/00103

Плетнева А. Д.

Руководитель

Чуркин В. В.

Санкт-Петербург
2023

Оглавление

Элементы оглавления не найдены.

Цель работы

1. Получение на ЭВМ с помощью программного датчика базовой последовательности псевдослучайных чисел, имеющих равномерное распределение.
2. Освоение методов статистической оценки полученного распределения: вычисление эмпирических значений для математического ожидания и дисперсии.
3. Освоение методов оценки статистики связи: вычисление значений автокорреляционной функции и построение коррелограммы.
4. Освоение методов графического представления законов распределения: построение функции плотности распределения и интегральной функции распределения.

Ход работы

Была получена псевдослучайная последовательность, имеющая равномерный характер на интервале $[0,1]$, с помощью программного датчика случайных чисел - `np.random.random(10000)`

Следующим шагом было вычисление эмпирических значений математического ожидания и дисперсии полученной последовательности псевдослучайных чисел по формулам:

$$\bar{M} = (u[1] + u[2] + \dots + u[n])/n$$

$$D = \{(u[1] - \bar{M})^2 + (u[2] - \bar{M})^2 + \dots + (u[n] - \bar{M})^2\}/n$$

Сравним полученные результаты с соответствующими теоретическими значениями:

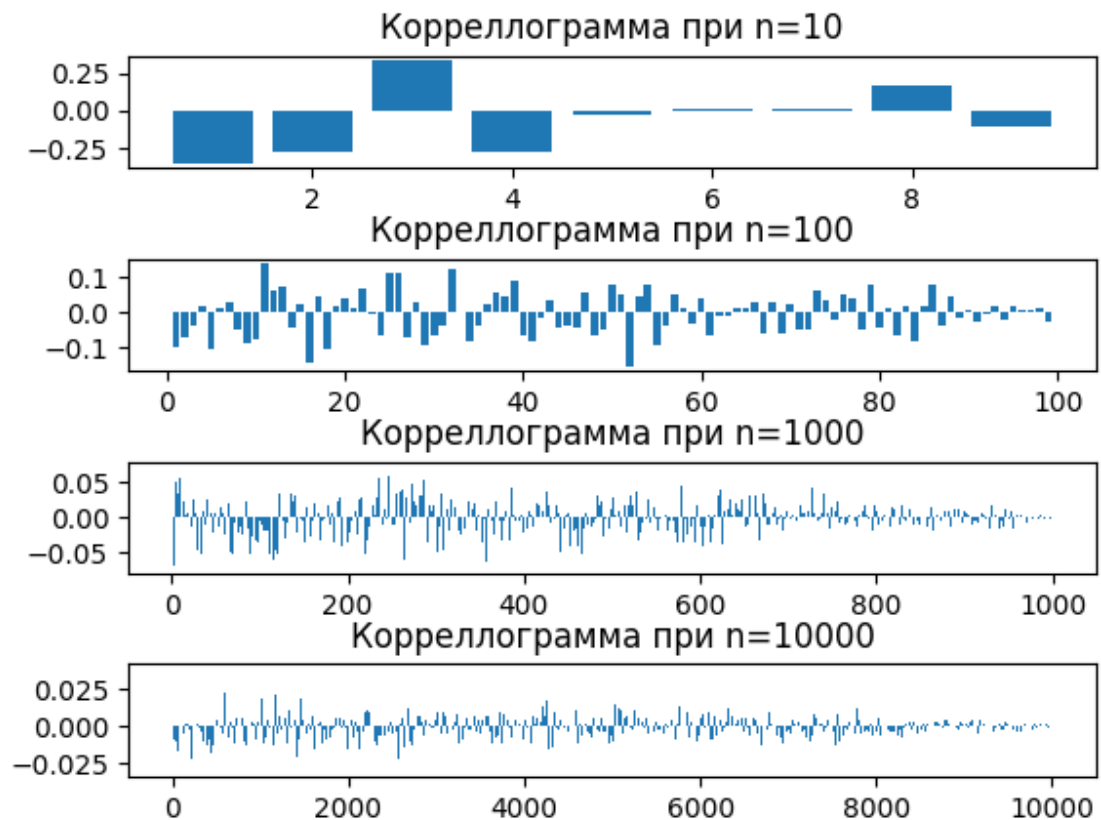
n	Оценка распределений	RAND (эксперимент)	Теоретическое значение	Отклонение
10	M	0.41697303444509926	0.5	0.08302696555490074
10	D	0.11358929065900594	0.08333	0.03025929065900594
100	M	0.515041703133988	0.5	0.015041703133988005
100	D	0.08341519997046651	0.08333	8.519997046650685e-05
1000	M	0.5118952642236532	0.5	0.011895264223653168
1000	D	0.08082059371964843	0.08333	0.002509406280351567
10000	M	0.5002374141144181	0.5	0.00023741411441813298
10000	D	0.08298090412052792	0.08333	0.0003490958794720861

Видим, что с увеличением размера массива отклонение уменьшается, математическое ожидание и дисперсия стремятся к своим теоретическим значениям.

Для оценки степени связанности псевдослучайных чисел воспользуемся корреляционной (или "автокорреляционной") функцией $K(f)$, которая представляет собой последовательность коэффициентов корреляции, зависящих от величины сдвига f , как от аргумента:

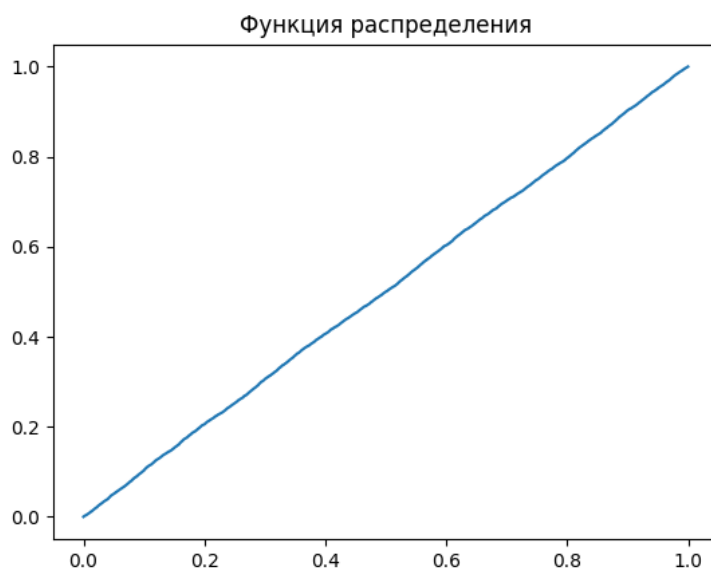
$$K[f] = \frac{\sum_{i=1}^{n-f} (u[i] - \bar{M})(u[i+f] - \bar{M})}{\sum_{i=1}^n (u[i] - \bar{M})^2}$$

Строим для всех n коррелограммы:



В нуле значение $K[0]$ будет равно единице, не будем это отображать, чтобы улучшить масштаб. При увеличении количества n уменьшаются максимальные значения K .

Графическое представление законов распределения:





Сравнивая полученные графики с соответствующими теоретическими кривыми, можно заметить, что полученное нами распределение очень близко к равномерному.

Вывод

Программный датчик случайных чисел - `np.random.random()` можно использовать в качестве базового для получения случайных величин с равномерным законом распределения.

Текст программы

```
import numpy as np
import scipy
from prettytable import PrettyTable
import matplotlib.pyplot as plt

def calc_m_d(n_array, rand_array, table):
    fig, axes = plt.subplots(nrows=4)
    plt.subplots_adjust(hspace=0.8)
    plt.figure(1)
    for index, n in enumerate(n_array):
        M = 0
        D = 0
        K = [0] * n
        for i in range(n):
            M += rand_array[i]
        M /= n
        for i in range(n):
            D += (rand_array[i] - M) ** 2
        D /= n
        for f in range(n):
            K[f] = 0
            for i in range(n - f):
                K[f] += (rand_array[i] - M) * (rand_array[i + f] - M)
            K[f] /= D * n

        axes[index].bar(np.arange(1, n),
                        np.array(K[1:n])
                        )
        axes[index].set_title(f'Корреллограмма при n={n}')
        table.add_row([n, 'M', M, 0.5, abs(0.5 - M)])
        table.add_row([n, 'D', D, 0.08333, abs(0.08333 - D)])

table = PrettyTable(['n', 'Оценка распределений', 'RAND (эксперимент)',
                    'Теоретическое значение', 'Отклонение'])
n = [10, 100, 1000, 10000]
rand_array = np.random.random(10000)
calc_m_d(n, rand_array, table)
data_sorted = np.sort(rand_array)
p = 1. * np.arange(len(rand_array)) / (len(rand_array) - 1)
plt.figure(2)
fig, axes = plt.subplots()
axes.plot(data_sorted, p)
axes.set_title(f'Функция распределения')
plt.figure(3)
figure, ax = plt.subplots()
ax.hist(rand_array, density=True)
ax.set_title(f'Плотность распределения')
plt.show()
print(table)
```