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

Отчет

по лабораторной работе №1 Получение базовой последовательности псевдослучайных чисел и тестовые проверки его работы

по дисциплине «Введение в машинное обучение»

Выполнила Студентка гр. 5130904/10101

Никифорова Е. А.

Руководитель

Чуркин В. В.

Оглавление

Цель работы	3
Ход работы	3
Результаты	4
Выборка объемом 10	4
Выборка объемом 100	5
Выборка объемом 1000	6
Выборка объемом 10000	7
Таблица точечных оценок	7
Вывод	8
Приложение	c

Цель работы

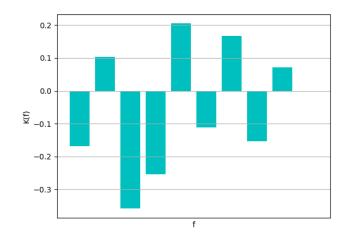
- 1. Получение на ЭВМ с помощью программного датчика базовой последовательности псевдослучайных чисел, имеющих равномерное распределение.
- 2. Освоение методов статистической оценки полученного распределения: вычисление эмпирических значений для математического ожидания и дисперсии.
- 3. Освоение методов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

Ход работы

- 1. Получение псевдослучайной последовательности, имеющей равномерный характер на интервале [0,1], с помощью программного датчика случайных чисел.
- 2. Вычисление эмпирических значений математического ожидания и дисперсии полученной последовательности псевдослучайных чисел; сравнение полученных результатов с соответствующими теоретическими значениями.
- 3. Вычисление значений автокорреляционной функции и построение коррелограммы.
- 4. Графическое представление законов распределения: построение эмпирической функции плотности распределения и эмпирической интегральной функции распределения; сравнение с соответствующими теоретическими кривыми.

Результаты

Выборка объемом 10



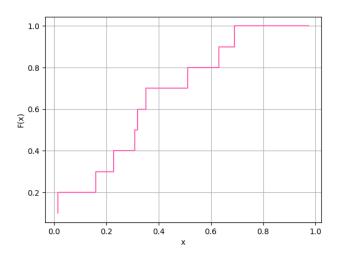


Рисунок 1 Коррелограмма 10

Рисунок 2 Функция распределения 10

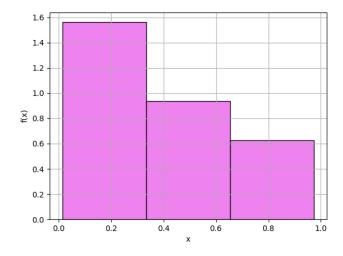
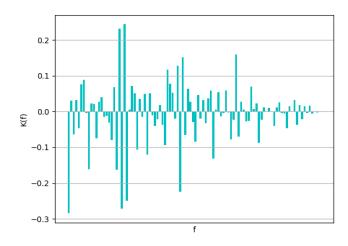


Рисунок ЗФункции плотности распределения 10

Выборка объемом 100



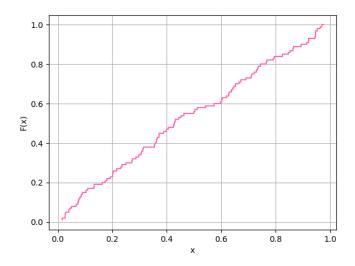


Рисунок 4 Коррелограмма 100



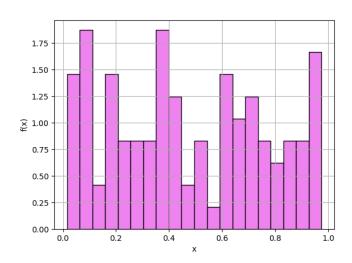


Рисунок 6Функции плотности распределения 100

Выборка объемом 1000

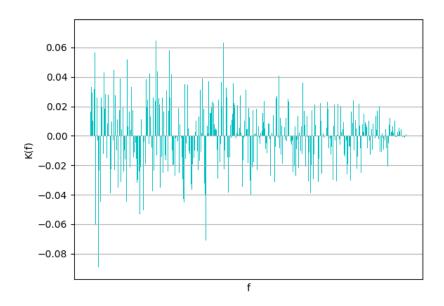


Рисунок 7 Коррелограмма 1000

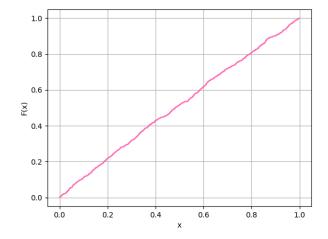


Рисунок 8 Функция распределения 1000

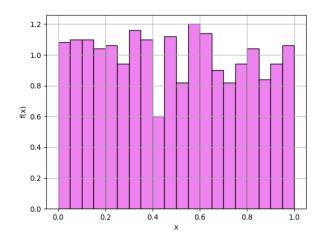


Рисунок 9 Функции плотности распределения 1000

Выборка объемом 10000

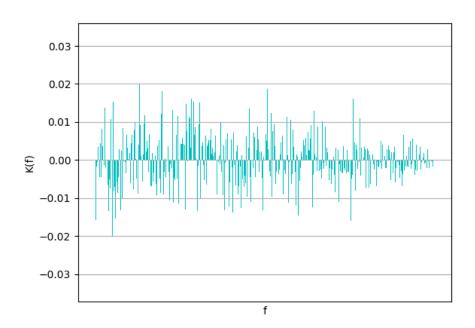


Рисунок 10 Коррелограмма 10000

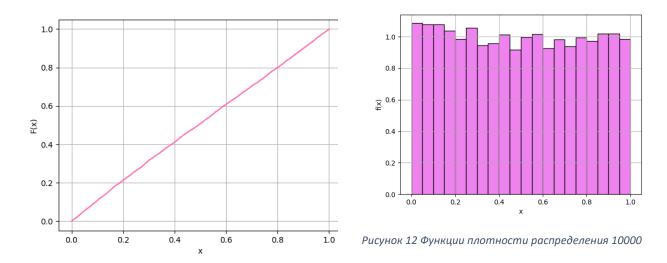


Рисунок 11 Функция распределения 10000

Таблица точечных оценок

n	M	M (Theoretical value)	M (Delta)	D	D (Theoretical value)	D (Delta)
10	0.4191388406159298	0.5	-0.08086115938407018	0.07297748784150068	0.08333	-0.010352512158499319
100	0.4736054960092702	0.5	-0.02639450399072979	0.0867580799330651	0.08333	0.003428079933065098
1000	0.4873694432150475	0.5	-0.012630556784952507	0.08434912852743026	0.08333	0.0010191285274302603
10000	0.493468624074553	0.5	-0.006531375925447003	0.08530482511234833	0.08333	0.0019748251123483246

Вывод

По результатам лабораторной работы, можно сделать вывод, что при малом количестве данных мы не можем утверждать, что функция распределена равномерно. При увеличении количества случайных величин уменьшается delta, следовательно, функция принимает равномерный закон распределения.

Приложение

```
import matplotlib.pyplot as plt
import numpy as np
import random
from statsmodels.distributions.empirical distribution import ECDF
def get_info_for_n(n: int) -> dict:
    arr = [random.random() for _ in range(n)]
    mat_og = sum(arr) / n
    dispersion = sum([(value - mat_og) ** 2 for value in arr]) / n
    sred_kvadr = np.sqrt(dispersion)
    korrelyation = []
    for f in range(1, n + 1):
        up\_value = sum((arr[j - 1] - mat\_og) * (arr[j + f - 1] - mat\_og) for j in
range(1, n - f + 1))
        down_value = sum((arr[i - 1] - mat_og) ** 2 for i in range(1, n + 1))
        korrelyation.append(up value / down value)
    return {'Математическое ожидание': mat og,
            'Эмпирическая дисперсия': dispersion,
            'Эмпирическое среднее квадратическое': sred_kvadr,
            'Коэффициенты корреляции': korrelyation,
            'Значения массива': arr}
def show_korrelation_diagram(arr: []) -> None:
    plt.bar(range(len(arr)), arr, color='c')
    plt.xlabel("f")
    plt.ylabel("K(f)")
    plt.xticks([])
    plt.grid(True)
    plt.show()
def show_density_distribution_function(arr: []) -> None:
    bins num = 3
    if len(arr) > 10:
        bins_num = 20
    plt.hist(arr, bins=bins num,
             color='violet', edgecolor='black', density=True)
    plt.xlabel('x')
    plt.ylabel('f(x)')
    plt.grid(True)
    plt.show()
def show_distribution_function(arr: []) -> None:
    ecdf = ECDF(arr)
    plt.step(ecdf.x, ecdf.y, color="hotpink")
    plt.xlabel("x")
```

```
plt.ylabel("F(x)")
    plt.grid(True)
    plt.show()
def show_table(arr: []):
    col_names = ["n", "M", "M (Theoretical value)", "M (Delta)", "D", " D (Theoretical
value)", "D (Delta)"]
    table = plt.table(colLabels=col_names, cellText=arr, loc='center', cellLoc='center')
    table.auto_set_font_size(False)
    table.set fontsize(7)
    plt.axis('off')
    plt.grid(True)
    plt.show()
if __name__ == "__main__":
    arr_of_nums = [10, 100, 1000, 10000]
    table = []
    for n in arr_of_nums:
        result = get_info_for_n(n)
        values = result.get('Значения массива')
        mat_og = result.get('Математическое ожидание')
        dispersion = result.get('Эмпирическая дисперсия')
        table.append([n, mat_og, 0.5, mat_og - 0.5, dispersion, 0.08333, dispersion -
0.08333])
        show_korrelation_diagram(result.get('Коэффициенты корреляции'))
        show_distribution_function(values)
        show_density_distribution_function(values)
    show_table(table)
```