

Министерство цифрового развития, связи и
массовых коммуникаций Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего образования «Сибирский государственный университет
телекоммуникаций и информатики» (СибГУТИ)

ЛАБОРАТОРНАЯ РАБОТА №4
по дисциплине «Моделирование»

Выполнил:
студент гр. ИС-142
«__» мая 2025 г.

_____ /Григорьев Ю.В./

Проверил:
преподаватель
«__» мая 2025 г.

_____ /Уженцева А.В./

Оценка « _____ »

Новосибирск 2025

ВВЕДЕНИЕ

В данной работе разработана программа для анализа выборок из массива чисел и построения **частотных матриц включения**. Основная цель исследования — изучить, как различные критерии выборки (например, числа, кратные 5, простые числа и случайная выборка) **влияют на структуру выборки** и ее представление в виде частотной матрицы.

Ключевые особенности реализации:

- Рассматриваются два вида выборки:
 - **С возвратом** (один и тот же элемент может быть выбран несколько раз).
 - **Без возврата** (каждый элемент выбирается не более одного раза).
- Вместо бинарных матриц (0 и 1) строятся **частотные матрицы**, которые показывают, **сколько раз** каждый элемент был выбран.
- Для наглядности строятся **графики частотных матриц**, которые сохраняются в файлы.

ВЫПОЛНЕНИЕ РАБОТЫ

1. Генерация исходного массива

Программа создает массив целых чисел от 1 до size. В рамках эксперимента выбраны три размера массива:

- 10 элементов (малый размер, удобный для просмотра).
- 100 элементов (средний размер).
- 1000 элементов (крупный размер, приближен к реальным данным).

Создание массива:

```
def generate_numbers(size):  
    return np.arange(1, size + 1)
```

2. Выборка элементов по критериям

Из массива выбираются элементы, соответствующие заданным критериям:

1. Числа, кратные 5 (делятся на 5 без остатка).
2. Простые числа (имеют ровно два делителя: 1 и само число).
3. Случайная выборка (каждый элемент выбирается с вероятностью 50%).

Реализация критериев:

```
sample_criteria = [  
    ("Кратные пяти", lambda x: x % 5 == 0),  
    ("Простые числа", lambda x: all(x % i != 0 for i in range(2, int(np.sqrt(x)) + 1)) and x >  
1),  
    ("Случайная выборка", lambda x: random.random() > 0.5)  
]
```

3. Проведение выборки

После применения критерия создаются два типа выборок:

- С возвращением: элементы могут повторяться.
- Без возвращения: каждый элемент выбирается только один раз.

```
def sample_with_replacement(numbers, sample_size, seed=None):
```

```
    if seed is not None:
```

```
        random.seed(seed)
```

```
    return [int(random.choice(numbers)) for _ in range(sample_size)]
```

```
def sample_without_replacement(numbers, sample_size, seed=None):
```

```
    if seed is not None:
```

```
        random.seed(seed)
```

```
    return list(map(int, random.sample(list(numbers), sample_size)))
```

(seed варьируется случайно, чтобы результаты различались при каждом запуске)

4. Формирование частотной матрицы

Теперь матрица не бинарная, а показывает частоту выбора элементов.

Формат матрицы:

- Столбцы – индексы элементов исходного массива.
- Строки – номера элементов выборки.

- Числа в матрице – сколько раз элемент был выбран в разных проходах.

```
def build_frequency_matrix(numbers, sample, passes=3, use_seed=True):  
    n = len(sample)  
    m = len(numbers)  
    matrix = np.zeros((n, m), dtype=int)  
    for pass_idx in range(passes):  
        seed = random.randint(0, 10000) if use_seed else None # Варьируем seed  
        sample_wr = sample_with_replacement(sample, len(sample), seed)  
        for i, value in enumerate(sample_wr):  
            idx = np.where(numbers == value)[0][0]  
            matrix[i, idx] += 1  
    return matrix
```

5. Визуализация матриц

Чтобы понять, какие элементы выбираются чаще, строятся графики частотных матриц.

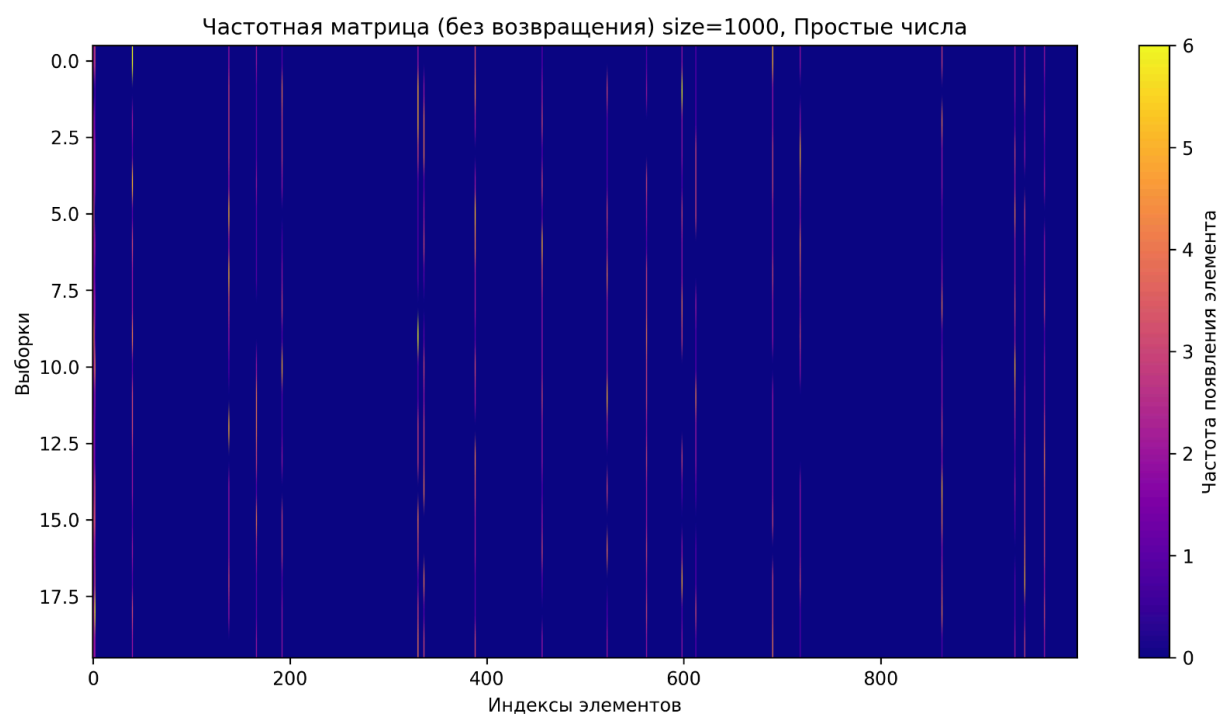
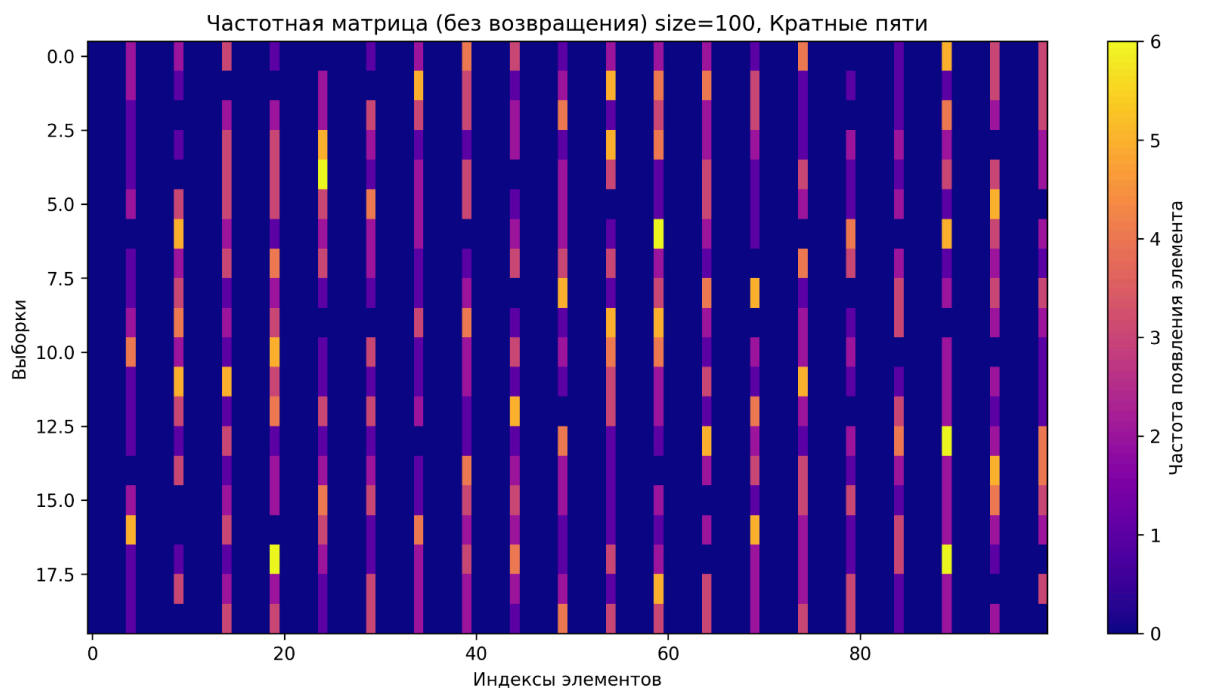
Ось X – индексы элементов исходного массива.

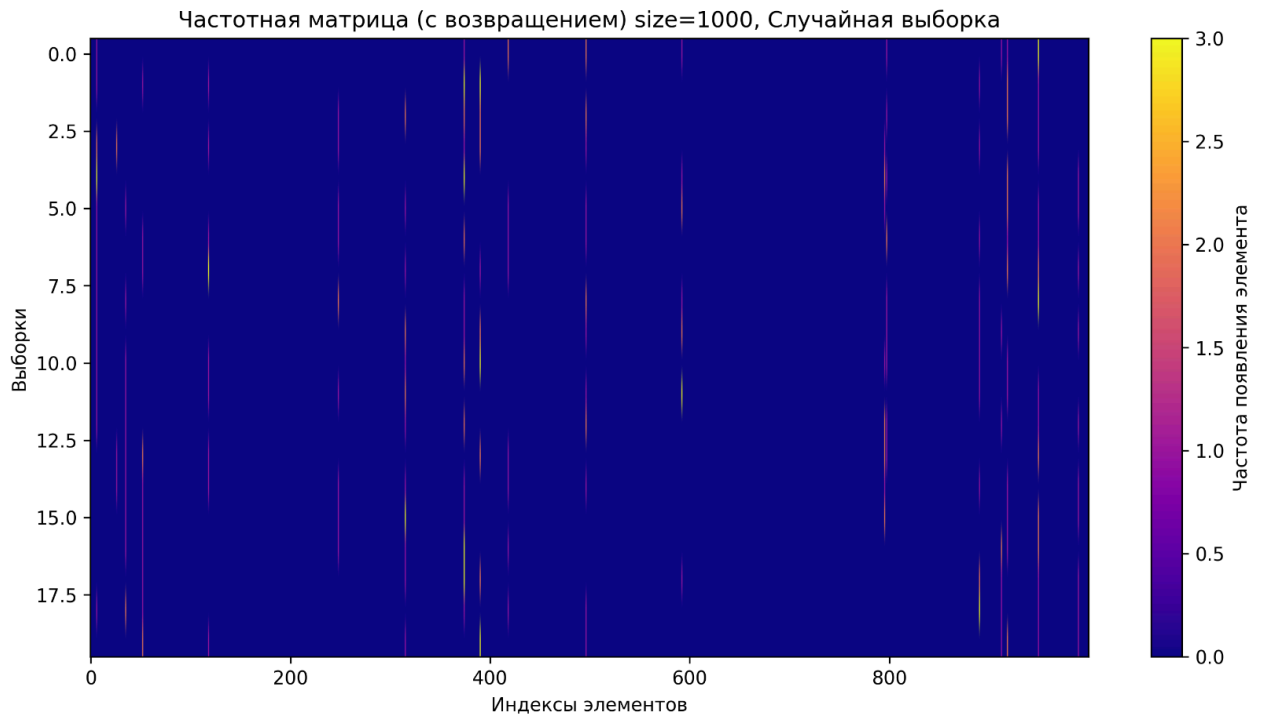
Ось Y – номера элементов выборки.

Цвет – чем ярче, тем чаще элемент был выбран.

```
def save_matrix_plot(matrix, title, filename):  
    plt.figure(figsize=(12, 6))  
    plt.imshow(matrix, cmap="plasma", aspect="auto")  
    plt.xlabel("Индексы элементов")  
    plt.ylabel("Выборки")  
    plt.title(title)  
    plt.colorbar(label="Частота появления элемента")  
    plt.savefig(filename, dpi=300)  
    plt.close()
```

Примеры построенных графиков (часть от всех сгенерированных):





Выводы в консоль:

```
Исходный массив (size=10): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
Выборка (2 элементов) из 10 по критерию: Кратные пяти  
[5, 10]
```

```
Пример размещений элементов выборки:
```

```
Размещение с возвращением: [10, 10]
```

```
Размещение без возвращения: [5, 10]
```

```
Матрица частот с возвращением, 13 проходов:
```

```
0 0 0 0 0 0 0 0 0 13
```

```
0 0 0 0 0 0 0 0 0 13
```

```
Матрица частот без возвращения, 42 прохода:
```

```
0 0 0 0 16 0 0 0 0 26
```

```
0 0 0 0 19 0 0 0 0 23
```

```
Графики сохранены: plots\freq_matrix_wR_10_Кратные_пяти.png, plots\freq_matrix_wOR_10_Кратные_пяти.png
```

Пример размещений элементов выборки:
Размещение с возвращением: [4, 4, 10, 5, 1]
Размещение без возвращения: [1, 10, 4, 5, 8]

2	0	0	4	2	0	0	0	0	5
2	0	0	8	3	0	0	0	0	0
2	0	0	5	3	0	0	0	0	3
4	0	0	2	2	0	0	0	0	5
0	0	0	8	5	0	0	0	0	0

11	0	0	13	6	0	0	9	0	3
11	0	0	8	7	0	0	5	0	11
14	0	0	7	8	0	0	12	0	1
9	0	0	13	4	0	0	13	0	3
16	0	0	6	10	0	0	6	0	4

Графики сохранены: `plots\freq_matrix_WR_100_Простые_числа.png`, `plots\freq_matrix_WOR_100_Простые_числа.png`

В результате работы разработана **программа для анализа выборок из числовых массивов** и построения **частотных матриц включения**.

1. Реализованы три критерия выборки (**кратные 5, простые числа, случайная выборка**).
2. Проведен анализ выборки **с возвращением и без него**.
3. Вместо бинарных матриц теперь строятся **частотные матрицы**, показывающие, **какие числа выпадают чаще**.

4. Добавлена **вариативность seed**, чтобы результаты различались при каждом запуске.
5. Построены графики, наглядно демонстрирующие распределение частот.

Практическое значение работы:

1. Аналогичные методы используются в **статистике и машинном обучении** для изучения распределений выборок.
2. **Частотные матрицы** позволяют анализировать, какие элементы выпадают чаще в выборках.
3. Возможные расширения работы:
 - Анализ различных распределений вероятностей.
 - Генерация зависимых выборок.
 - Автоматизированное сравнение разных критериев выборки.

Приложение.

Листинг программы

```
import numpy as np
import random
import matplotlib.pyplot as plt
import os

def generate_numbers(size):
    return np.arange(1, size + 1)

def sample_with_replacement(numbers, sample_size, seed=None):
    if seed is not None:
        random.seed(seed)
    return [int(random.choice(numbers)) for _ in range(sample_size)]

def sample_without_replacement(numbers, sample_size, seed=None):
    if seed is not None:
        random.seed(seed)
    return list(map(int, random.sample(list(numbers), sample_size)))

def build_frequency_matrix(numbers, sample, passes=3, use_seed=True):
    n = len(sample)
    m = len(numbers)
    matrix = np.zeros((n, m), dtype=int)

    for pass_idx in range(passes):
        seed = random.randint(0, 10000) if use_seed else None
        sample_wr = sample_with_replacement(sample, len(sample), seed)

        for i, value in enumerate(sample_wr):
            idx = np.where(numbers == value)[0][0]
            matrix[i, idx] += 1

    return matrix

def save_matrix_plot(matrix, title, filename):
```



```

plt.figure(figsize=(12, 6))
plt.imshow(matrix, cmap="plasma", aspect="auto")
plt.xlabel("Индексы элементов")
plt.ylabel("Выборки")
plt.title(title)
plt.colorbar(label="Частота появления элемента")
plt.savefig(filename, dpi=300)
plt.close()

def print_matrix(matrix):
    for row in matrix:
        print(" ".join(f"{x:2}" for x in row))

output_dir = "plots"
os.makedirs(output_dir, exist_ok=True)

sizes = [10, 100]
sample_criteria = [
    ("Кратные пяти", lambda x: x % 5 == 0),
    ("Простые числа", lambda x: all(x % i != 0 for i in range(2, int(np.sqrt(x)) + 1))
and x > 1),
    ("Случайная выборка", lambda x: random.random() > 0.5)
]

for size in sizes:
    numbers = generate_numbers(size)
    print(f"\nИсходный массив (size={size}): {numbers.tolist()}")

    for criterion_name, criterion in sample_criteria:
        sample = [int(x) for x in numbers if criterion(x)]
        print(f"\nВыборка ({len(sample)} элементов) из {size} по критерию:
{criterion_name}")
        print(sample)

        if not sample:
            print(f"Для критерия {criterion_name} выборка пуста")
            continue

        sample_wr = sample_with_replacement(sample, min(len(sample), 20))
        sample_wor = sample_without_replacement(sample, min(len(sample), 20))

        print("\nПример размещений элементов выборки:")
        print(f"    Размещение с возвращением: {sample_wr}")
        print(f"    Размещение без возвращения: {sample_wor}")

        matrix_wr = build_frequency_matrix(numbers, sample_wr, passes=13, use_seed=True)
        matrix_wor = build_frequency_matrix(numbers, sample_wor, passes=42,
use_seed=True)

        print("\nМатрица частот с возвращением, 13 проходов:")
        print_matrix(matrix_wr)

        print("\nМатрица частот без возвращения, 42 прохода:")
        print_matrix(matrix_wor)

        filename_wr = os.path.join(output_dir,
f"freq_matrix_WR_{size}_{criterion_name.replace(' ', '_')}.png")
        filename_wor = os.path.join(output_dir,
f"freq_matrix_WOR_{size}_{criterion_name.replace(' ', '_')}.png")

        save_matrix_plot(matrix_wr, f"Частотная матрица (с возвращением) size={size},
{criterion_name}", filename_wr)
        save_matrix_plot(matrix_wor, f"Частотная матрица (без возвращения) size={size},
{criterion_name}", filename_wor)

        print(f"Графики сохранены: {filename_wr}, {filename_wor}")

```