ГУАП

КАФЕДРА № 41

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| старший преподаватель |  |  |  | Е. К. Григорьев |
| должн., уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 2. |
| МОДЕЛИРОВАНИЕ ГЕНЕРАТОРОВ РАВНОМЕРНО РАСПРЕДЕЛЕННЫХ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ. |
| по курсу: МОДЕЛИРОВАНИЕ. |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4217 |  |  |  | | У. А. Мазориев |
|  |  |  | подпись, дата | |  | инициалы, фамилия |

**Цель работы:**

получить навыки моделирования нормально распределенных псевдослучайных чисел в программной среде Pyton, а также первичной оценки качества полученных псевдослучайных чисел

**Результат выполнения работы**

Программный код:<https://github.com/Mrx112426/Modelirovanie/tree/main>

**Индивидуальный вариант №10**:

**Ход выполнения работы**

На отдельном листке бумаге рисуется график плотности распределения, для этого аналитически рассчитываются необходимые значения функции плотности вероятности. Результат показан на рисунке 1.

Изображение выглядит как текст, рукописный текст, бумага, Бумажное изделие

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 1 – График плотности распределения

На отдельном листе строится график функции распределения нормального закона с параметрами по варианту (рис. 2). Для этого используется метод стандартизации нормального распределения, т.е. преобразование нормального распределения с параметрами (4, 3) в стандартное нормальное распределение (0, 1) путем вычитания мат. ожидания из каждого значения и деления на стандартное отклонение:

Для вычисления значения *F(x)* используется таблица *z* значений.

Изображение выглядит как текст, рукописный текст, бумага, чернила

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 2 – График функции распределения

Далее графики визуализируются с помощью ЯП Python (рис. 3).

Изображение выглядит как График, диаграмма, линия

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 3 – Визуализация с помощью Python

Графики совпадают с теми, что построены вручную.

После генерируются 12 набор псевдослучайных чисел, всего реализуется 4 метода в каждом из которых три выборки N = [1000, 5000, 10000], на рисунке 4 показан пример сгенерированных выборок.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – Сгенерированные числа при N = 1000

После строятся гистограммы для каждого набора псевдослучайных чисел (рис. 5–7), эмпирические функции распределения (рис. 8 – 10) и диаграммы рассеивания (рис. 11- 13).

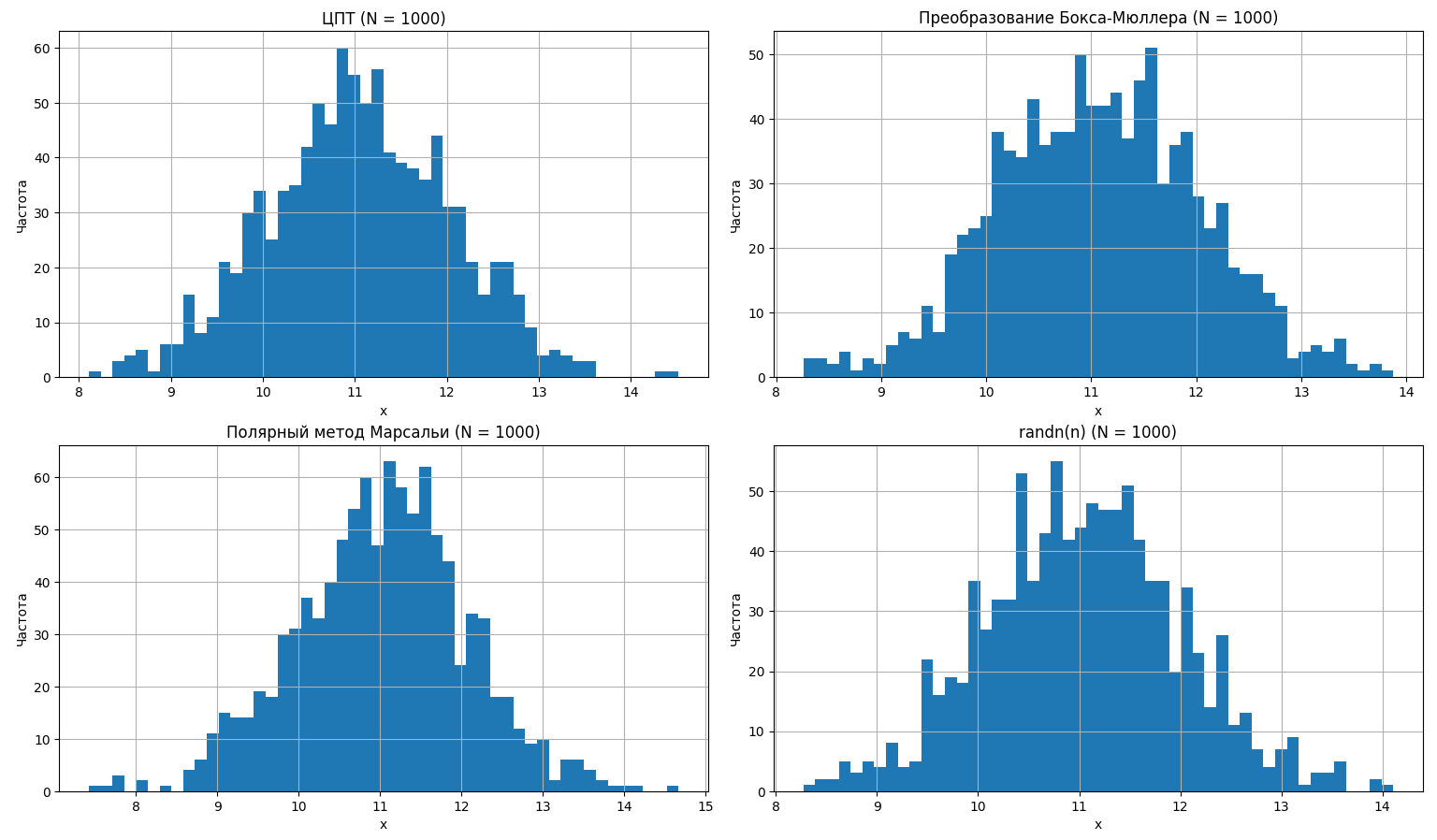


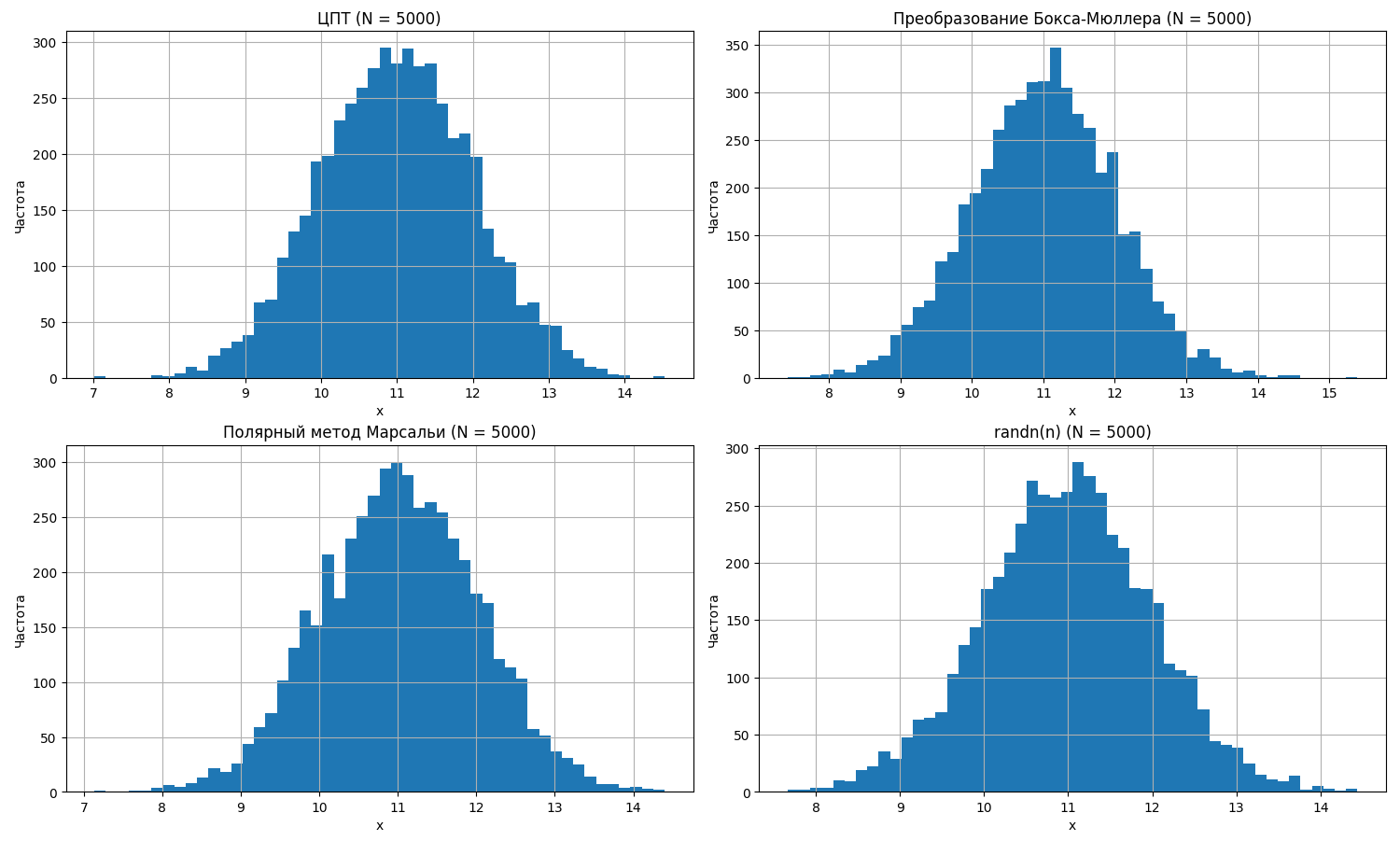
Рисунок 5 – Гистограммы для N = 1000

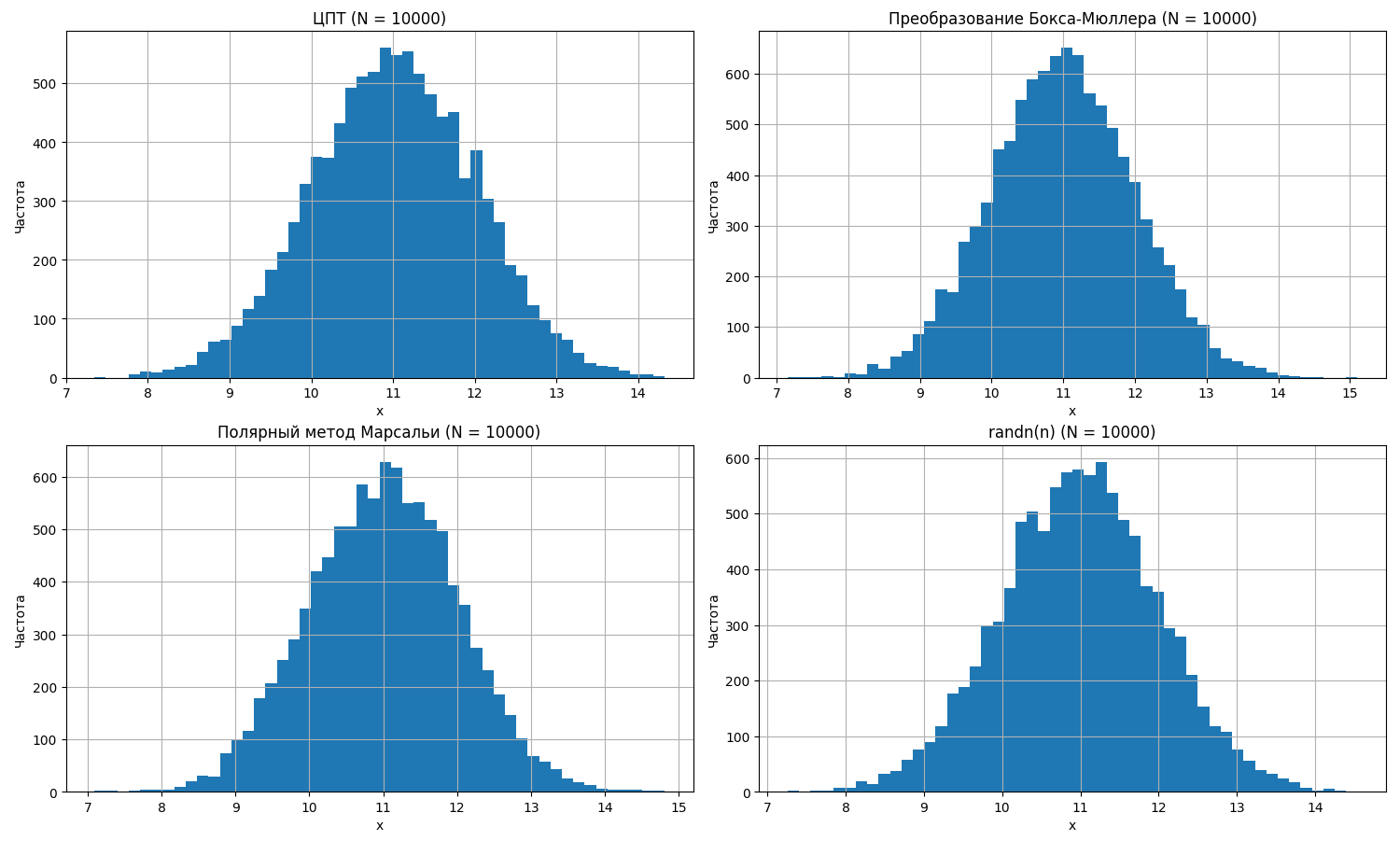
Рисунок 6 – Гистограммы для N = 5000

Рисунок 7 – Гистограммы для N = 10000

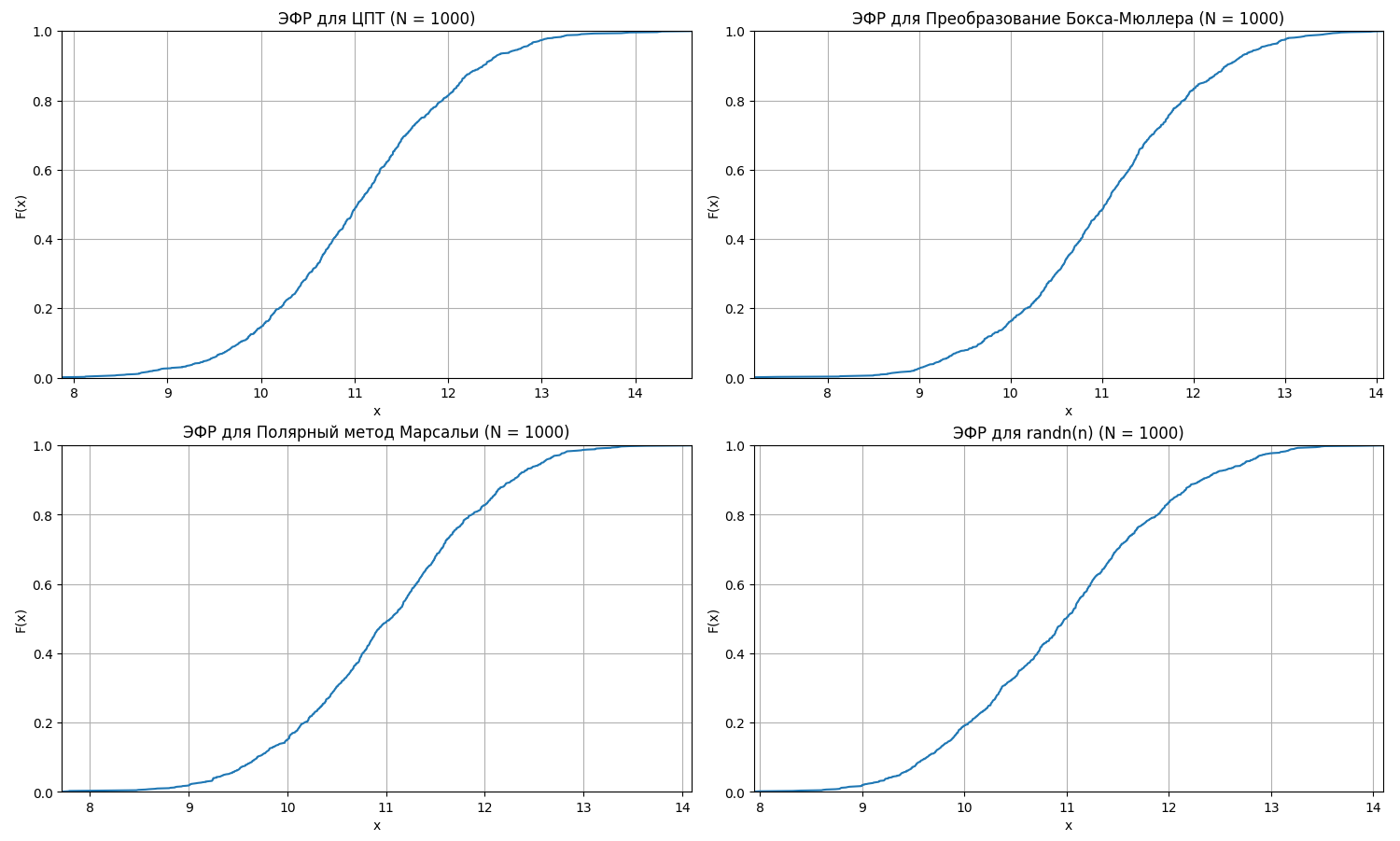


Рисунок 8 – ЭФР для N = 1000

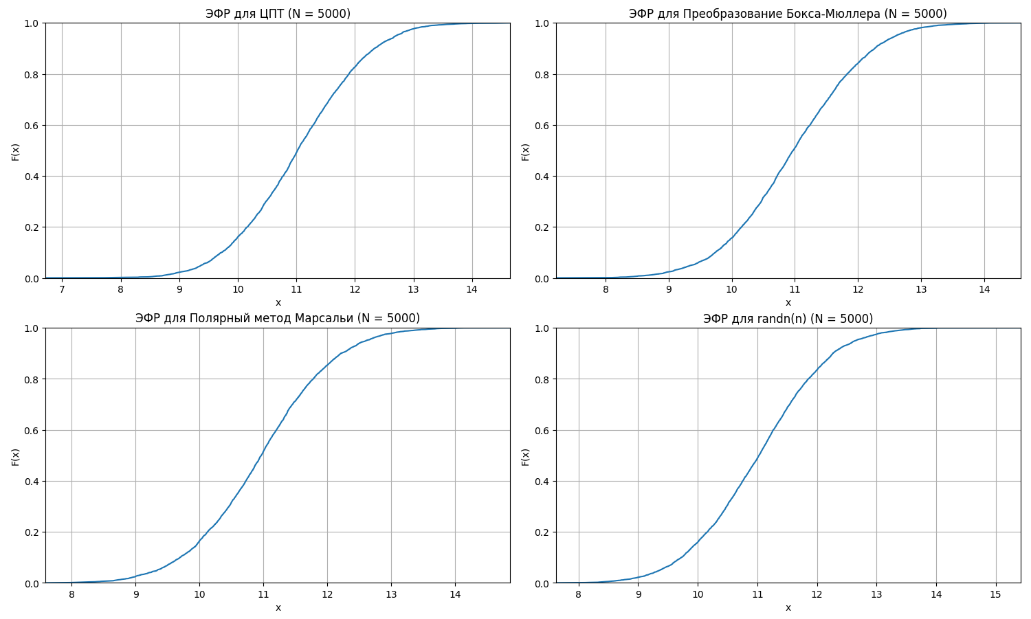


Рисунок 9 – ЭФР для N = 5000

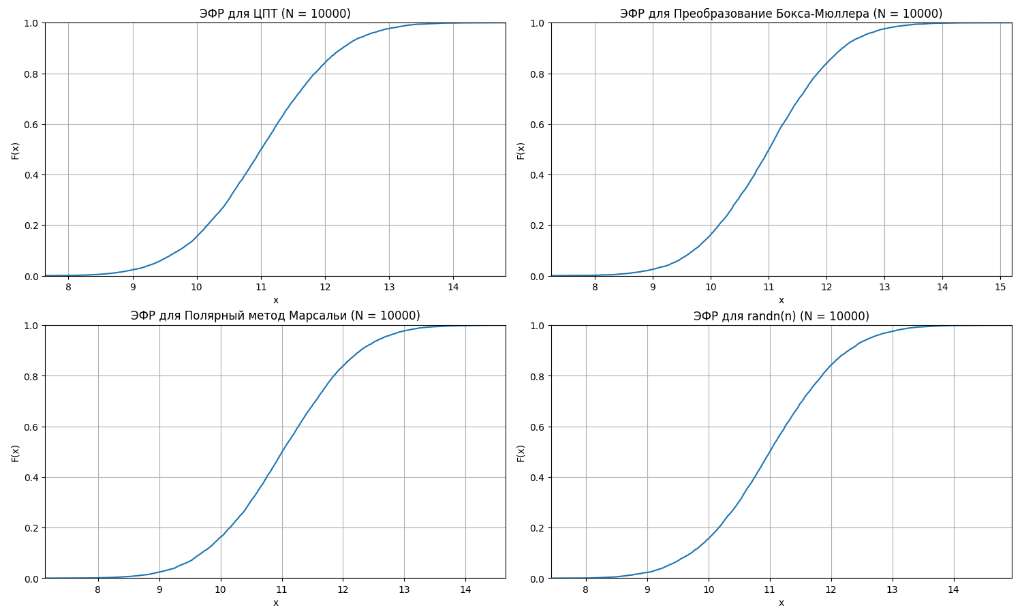


Рисунок 10 – ЭФР для N = 10000

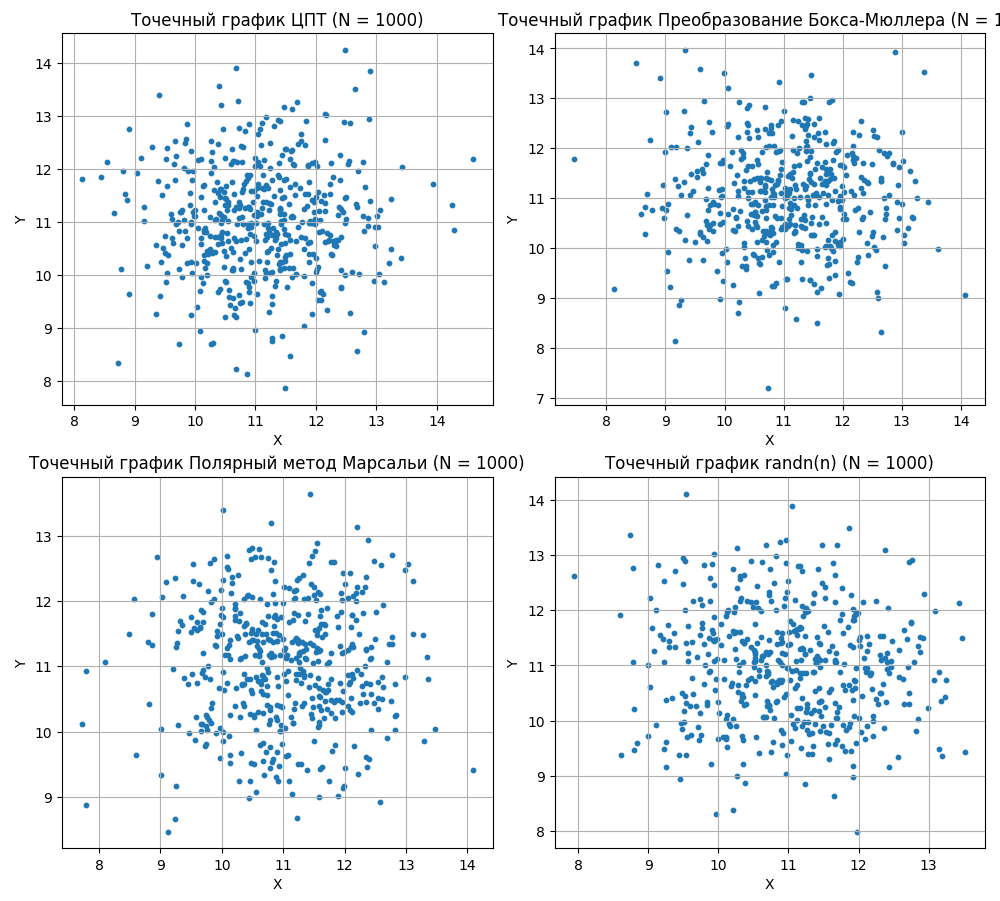


Рисунок 10 – Точечный график для N = 1000

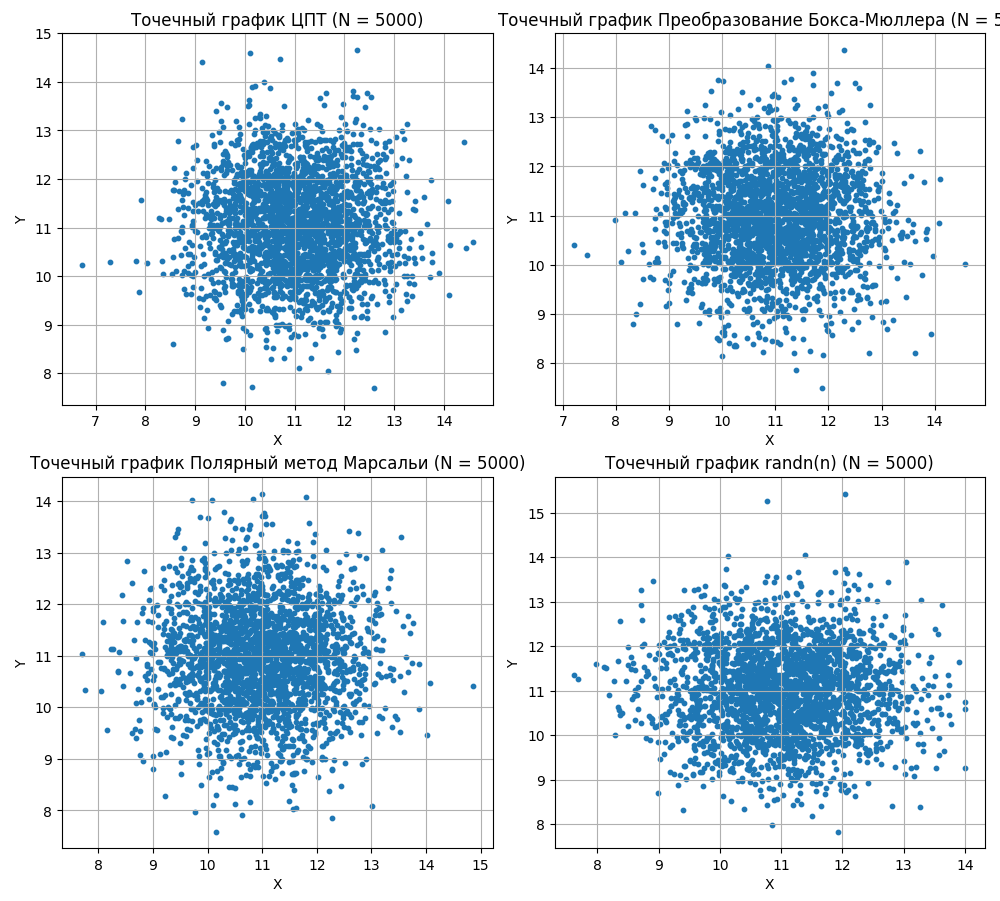


Рисунок 11 – Точечный график для N = 5000

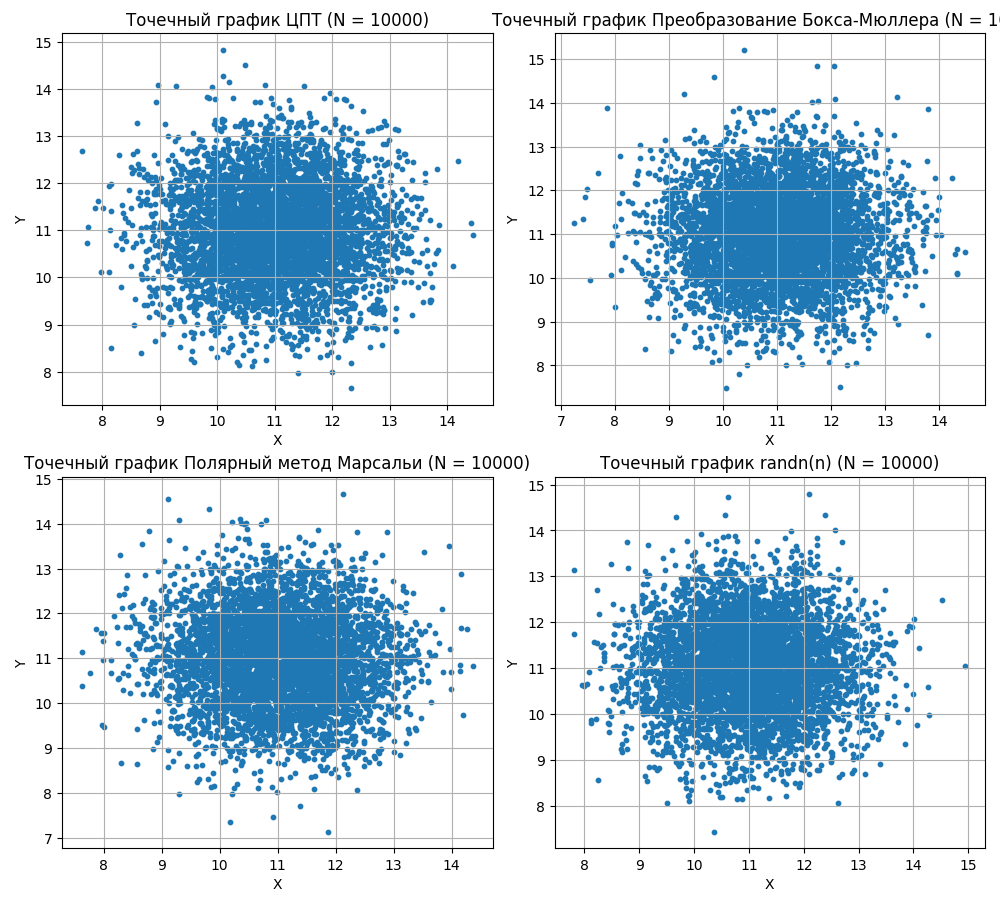


Рисунок 12 – Точечный график для N = 10000

Эмпирические функции распределения для всех сгенерированных наборов данных демонстрируют, что выбранные методы генерации адекватно воспроизводят нормально распределенные псевдослучайные числа.

Гистограммы подтверждают, что распределения стремятся к нормальному, что указывает на корректность работы генераторов. В целом результаты всех методов схожи.

Отмечена тенденция: с увеличением объема выборки распределение становится более нормальным, что вполне ожидаемо.

При объеме выборки в 10 000 значений метод ЦПТ показывает несколько более стабильные результаты. Все генераторы успешно создают равномерное распределение псевдослучайных чисел, что подтверждается диаграммами рассеяния, не показывающими явных закономерностей.

Далее строятся графики QQ-plot (рис. 13–15).

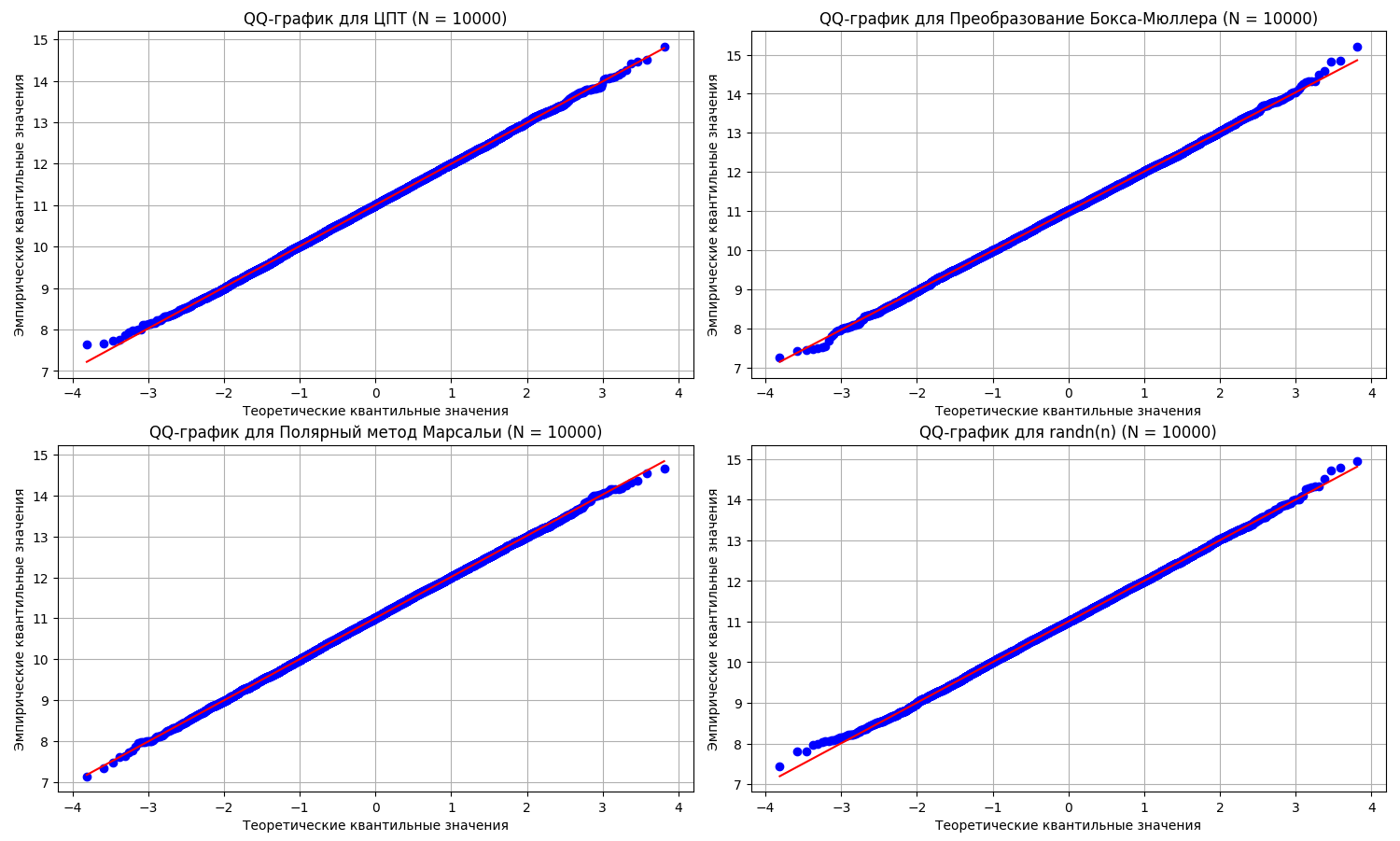


Рисунок 13 – График «квантиль - квантиль» для N = 1000

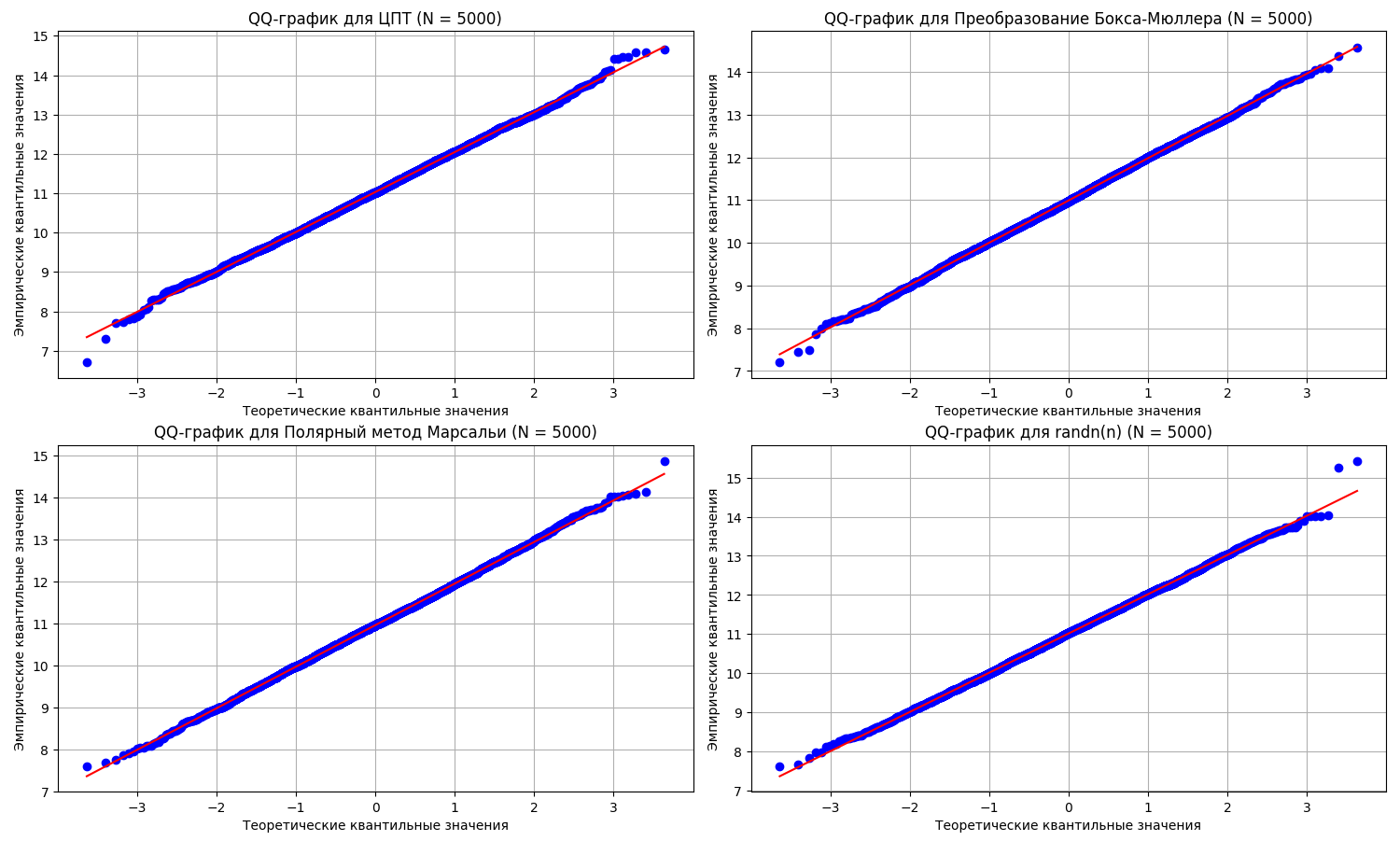


Рисунок 14 – График «квантиль - квантиль» для N = 5000

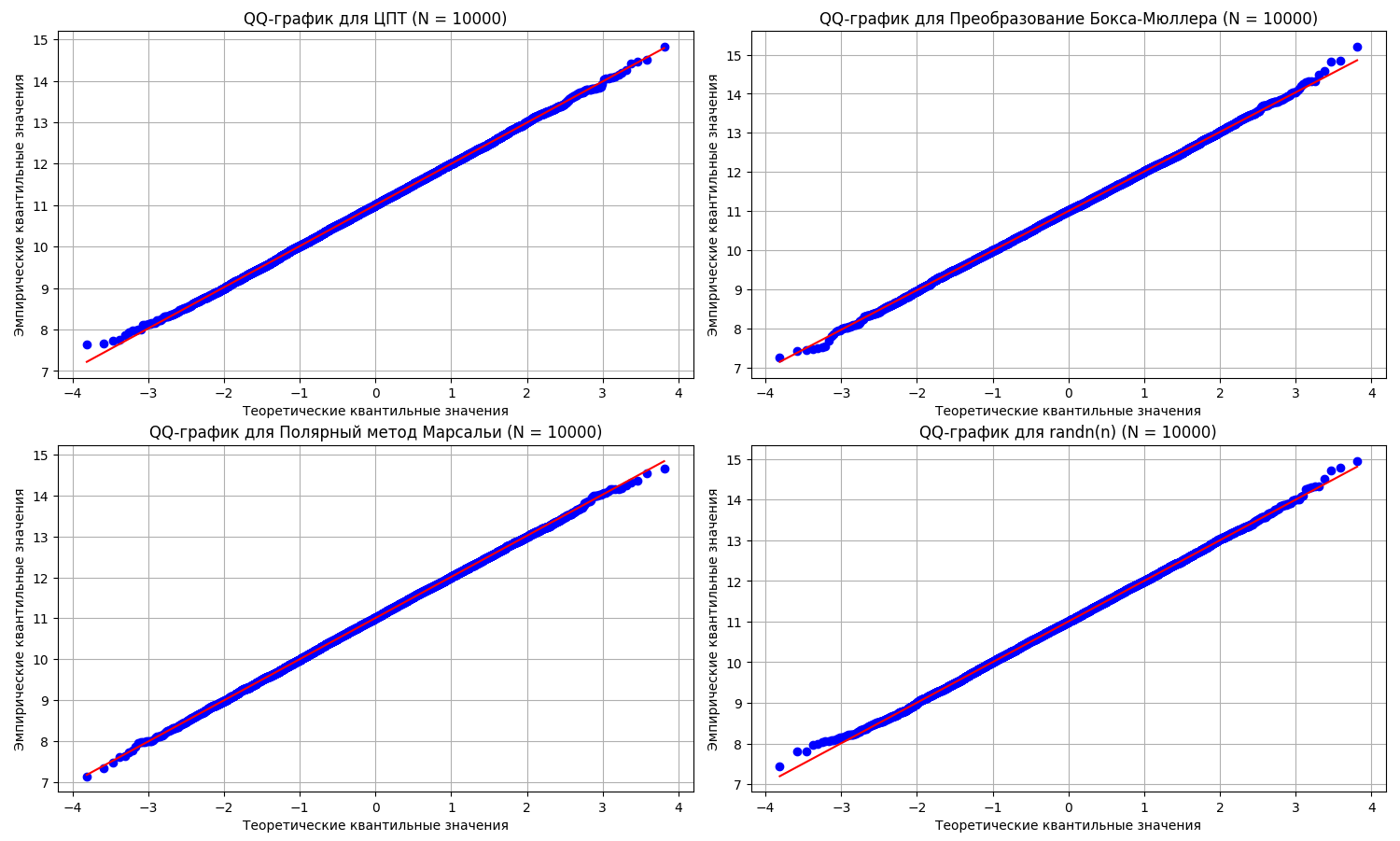


Рисунок 15 – График «квантиль - квантиль» для N = 10000

Все графики QQ-plot подтверждают, что примененные методы генерируют нормально распределенные псевдослучайные числа с высокой степенью точности. Плотное распределение точек вдоль диагональной линии указывает на соответствие с теоретическим нормальным распределением

После подсчитываются оценки математического ожидания, дисперсии и среднеквадратического отклонения (СКО) (рис. 16).

Изображение выглядит как текст, снимок экрана, меню, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 16 – Показатели по методам

Математическое ожидание для всех методов генерации близко к 11, что говорит о корректности генерации чисел.

Каждый метод демонстрирует стабильные и достаточно точные результаты, приближенные к нормальному распределению. При малых выборках наблюдаются незначительные отклонения: метод центральной предельной теоремы слегка занижает дисперсию, а преобразование Бокса-Мюллера иногда её завышает. Однако по мере увеличения объема выборки все методы сходятся к ожидаемым значениям, что подтверждает их корректность.

**Вывод**

В ходе выполнения лабораторной работы были приобретены навыки генерации нормально распределенных псевдослучайных чисел в среде Python, а также их первичной статистической оценки.

Анализ полученных данных, включая гистограммы, эмпирические функции распределения, графики рассеяния и QQ-plot, подтвердил, что использованные методы адекватно воспроизводят нормальное распределение. С увеличением объема выборки распределения становятся более симметричными и соответствуют теоретическим ожиданиям.

Исследование математического ожидания, дисперсии и стандартного отклонения показало, что все методы демонстрируют стабильные и точные результаты. Встроенный генератор randn(n) ожидаемо работает наиболее точно, а исправленный полярный метод Марсальи теперь корректно воспроизводит нормальное распределение.