

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Дальневосточный федеральный университет»**

**(ДВФУ)**

|  |  |
| --- | --- |
|  | **ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ** |

**ЛАБОРАТОРНАЯ РАБОТА №3**

По курсу «Теория вероятностей и математическая статистика»

Студент группы Б9123-01.03.02ии

Моттуева Уруйдана Михайловна

г. Владивосток

2025

Ход работы:



1. Краткое математическое описание

Имеется выборка из .

Эмпирическая функция распределения определяется как:

где – индикаторная функция, равная 1, если , и 0 в противном случае.

Доверительный интервал для ЭФР с уровнем доверия можно оценить как:

где

Для 95%-го доверительного интервала используется приближение:

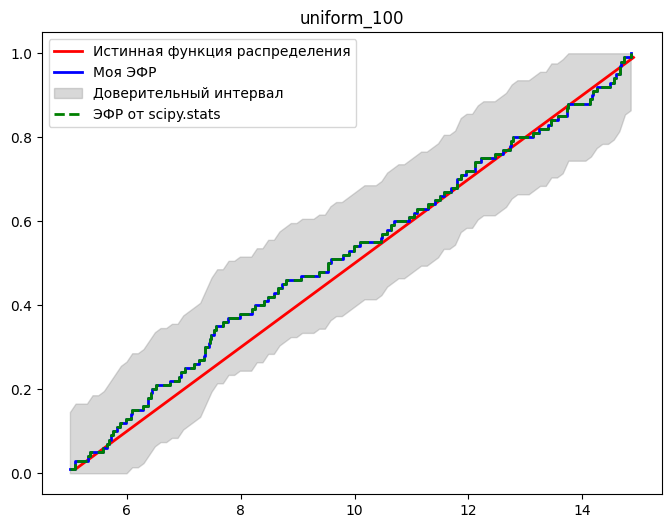
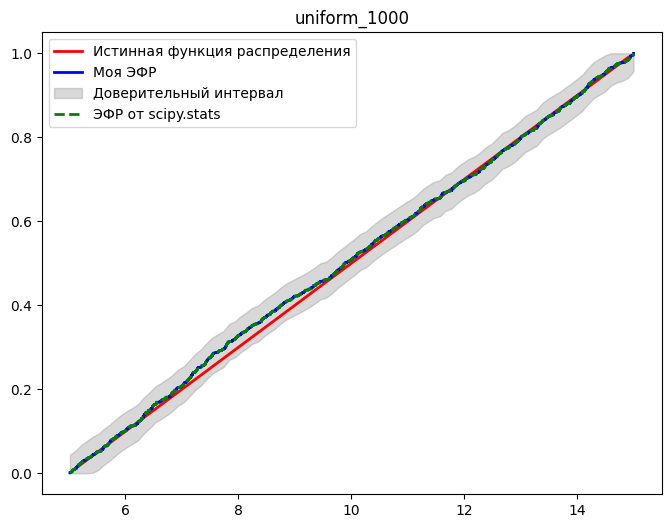
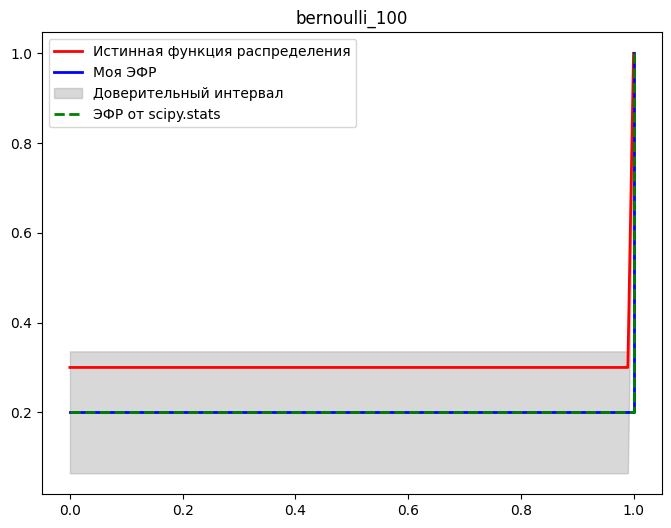
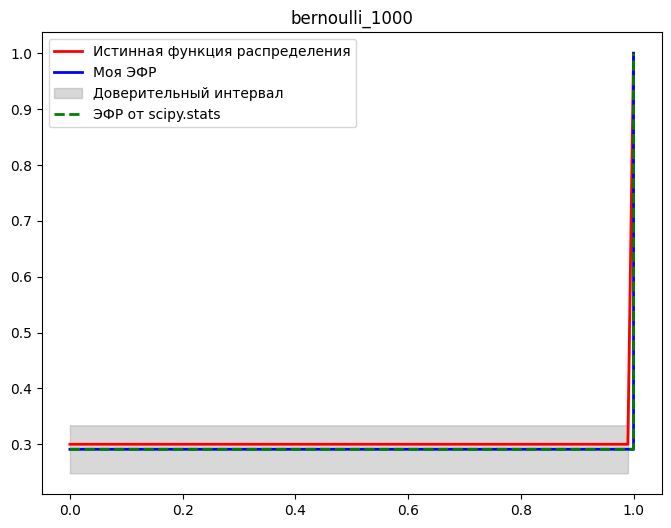
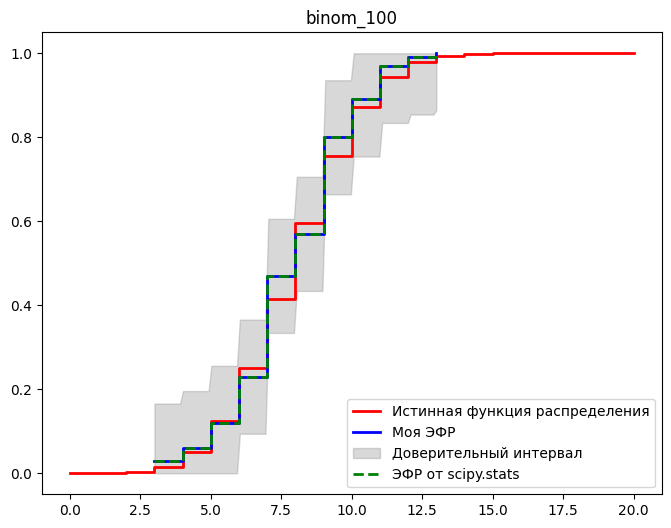
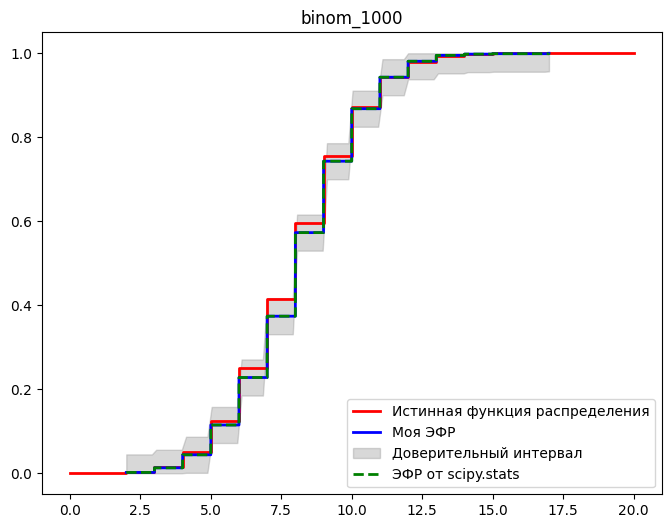
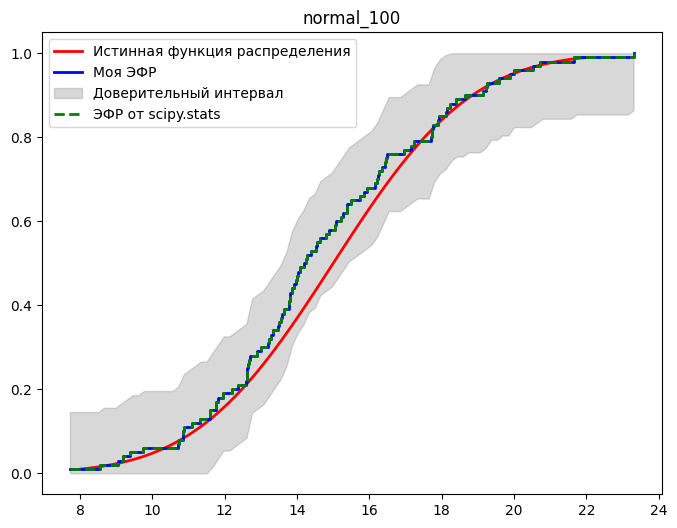
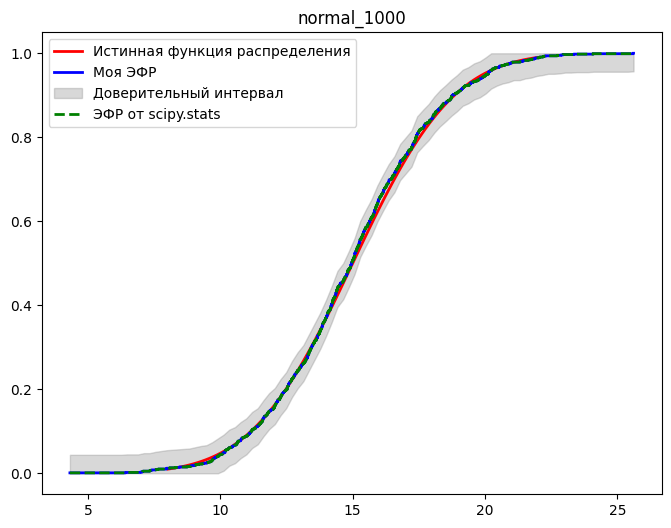
В Python реализация без использования scipy.stats:

1. def emperical\_distrib\_func(data, x):
2. n = len(data)
3. count\_less\_or\_equal = sum(1 for value in data if value <= x)
4. EFR = count\_less\_or\_equal / n
5. epsilon = np.sqrt(np.log(40) / (2 \* n))
6. lower\_bound = max(0, EFR - epsilon)
7. upper\_bound = min(1, EFR + epsilon)
8. return EFR, lower\_bound, upper\_bound

где строки 2-4 это определение ЭФР, 6-8 – поиск доверительного интервала.

В Python реализация ЭФР с использованием scipy.stats:

1. def built\_in\_ecdf(data, x):
2. ecdf\_obj = ECDF(data)
3. EFR = ecdf\_obj(x)
4. n = len(data)
5. std\_error = np.sqrt(EFR \* (1 - EFR) / n)
6. trust\_int = t.interval(1 - 0.05, df=n-1, loc=EFR, scale=std\_error)
7. lower\_bound = max(0, trust\_int[0])
8. upper\_bound = min(1, trust\_int[1])
9. return EFR, lower\_bound, upper\_bound
10. Выходные данные



<https://colab.research.google.com/drive/1EPjb1T1TRLe_F5U0u8RtHi7Ay_GgEkPE?usp=sharing> – ссылка на колаб с полным кодом программы