



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Дальневосточный федеральный университет»
(ДФУ)**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ**

ЛАБОРАТОРНАЯ РАБОТА №4

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 X_1, X_2, \dots, X_n из X .

Для вычисления доверительного интервала параметра θ с уровнем доверия γ можно использовать алгоритм:

- Сгенерировать k новых выборок (размером n) из исходной выборки;
- Для каждой выборки вычислить значение статистики;
- Отсортировать полученные значения статистики, доверительный интервал определяется как перцентили:

$$\left(\theta^*_{\left(\left\lfloor \frac{1-\gamma}{2} k \right\rfloor\right)}, \theta^*_{\left(\left\lceil \frac{1+\gamma}{2} k \right\rceil\right)} \right).$$

Реализация в python без использования scipy.stats:

```
1. def bootstrap(sample, estimator, confidence_level=0.95, n_resamples=1000):
2.     n = len(sample)
3.     estimates = []
4.     for _ in range(n_resamples):
5.         resample = np.random.choice(sample, size=n, replace=True)
6.         estimates.append(estimator(resample))
7.
8.     estimates = np.array(estimates)
9.     estimate = np.mean(estimates)
10.    lower = np.percentile(estimates, (1 - confidence_level) / 2 * 100)
11.    upper = np.percentile(estimates, (1 + confidence_level) / 2 * 100)
12.
13.    return estimate, lower, upper
14.
```

Реализация в python с использованием scipy.stats:

```
1. from scipy.stats import bootstrap
2. confidence_level = 0.95
3. n_resamples = 1000
4.
5. BOOTSTRAP = bootstrap((sample,), estimator,
6.                        confidence_level=confidence_level, n_resamples=n_resamples)
```

2. Таблица результатов

Распределение	Параметр	Истинное значение	Bootstrap нижняя граница	Bootstrap верхняя граница	Bootstrap scipy нижняя граница	Bootstrap scipy верхняя граница
uniform_100	Среднее	9.751311	9.162131	10.341601	9.131361	10.286040
uniform_100	Дисперсия	9.241274	7.705924	10.824683	7.955956	11.151979

uniform_100	Стандартное отклонение	3.039946	2.765170	3.290793	2.823316	3.299272
uniform_1000	Среднее	9.930836	9.749813	10.106592	9.745119	10.104801
uniform_1000	Дисперсия	8.580982	8.127596	9.071502	8.148909	9.062116
uniform_1000	Стандартное отклонение	2.929331	2.854304	3.011807	2.860067	3.010114
bernoulli_100	Среднее	0.800000	0.720000	0.870000	0.710000	0.870000
bernoulli_100	Дисперсия	0.160000	0.106667	0.203636	0.114242	0.203636
bernoulli_100	Стандартное отклонение	0.400000	0.337713	0.451261	0.337998	0.451261
bernoulli_1000	Среднее	0.709000	0.682000	0.735025	0.681000	0.737000
bernoulli_1000	Дисперсия	0.206319	0.194013	0.218177	0.195403	0.218163
bernoulli_1000	Стандартное отклонение	0.454224	0.441020	0.467094	0.439799	0.465924
binom_100	Среднее	7.870000	7.460000	8.280000	7.500000	8.300000
binom_100	Дисперсия	4.313100	3.307442	5.431677	3.442834	5.667919
binom_100	Стандартное отклонение	2.076800	1.786413	2.355561	1.849788	2.394824
binom_1000	Среднее	8.125000	8.002000	8.264075	7.994453	8.256143
binom_1000	Дисперсия	4.605375	4.207558	5.017957	4.225998	5.022738
binom_1000	Стандартное отклонение	2.146014	2.050164	2.235741	2.063720	2.247607
normal_100	Среднее	14.675226	14.093500	15.306378	14.089059	15.262458
normal_100	Дисперсия	9.239656	6.814280	11.816597	7.237905	12.249116
normal_100	Стандартное отклонение	3.039680	2.631744	3.433234	2.688797	3.543607
normal_1000	Среднее	14.967388	14.784930	15.164646	14.801034	15.151561
normal_1000	Дисперсия	8.857275	8.022858	9.704136	8.121335	9.794533
normal_1000	Стандартное отклонение	2.976117	2.845396	3.116302	2.855076	3.122332

https://colab.research.google.com/drive/1aC_rc2R2jCckEdEZ5G4pi1F5ivB9QUHF?usp=sharing

– полный код на colab