Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчет

по лабораторной работе №1

по дисциплине «**Теория вероятностей**»

Выполнил:

Чечеватов Роман

Факультет: ФИТиП

Группа: M32381

Преподаватель: Мурзина А. А.

Санкт-Петербург 2023

Задача 1, Вариант 2

Условие: Уходя из детского сада каждый ребенок случайно берет один левый и один правый ботинок. Найти вероятность того, что все они уйдут не в своих парах ботинок.

Аналитическое решение:

Обозначим как n количество детей в саду. Заметим, что процесс выбора всеми детьми ботинок можно охарактеризовать перестановкой (обозначим её как π), у которой элемент π_i – номер ребенка, чей ботинок выбрал ребенок с номером i. Тогда обозначим как φ, ψ две перестановки, соответствующие правым и левым ботинкам. Искомая вероятность это $1 - P(\exists i \text{ т.ч.} \varphi_i = \psi_i = i)$. Вторую вероятность посчитать уже проще: пусть k – количество таких подходящих i, s_1, s_2 – другие различные неподвижные точки перестановок φ, ψ. Также обозначим как $d(n)$ число беспорядков – перестановок без неподвижных точек. Тогда число подходящих перестановок выражается формулой:

$$S = \sum_{k=1}^{n} \sum_{s_1=0}^{n-k} \sum_{s_2=0}^{n-k} \binom{n}{k} \binom{n-k}{s_1} \binom{n-k-s_1}{s_2} d(n-k-s_1)\, d(n-k-s_2)$$

А искомая вероятность:

$$1 - \frac{S}{n!^2}$$

Для вычисления количества беспорядков воспользуемся следующей рекуррентной формулой: $\mathrm{d}(n) = (n-1)\big(d(n-1) + d(n-2)\big)$.

Симуляция:

Сгенерируем 2 случайных перестановки φ, ψ – одну для левых ботинок, одну для правых, и проверим существование такого элемента, что $\varphi_i = \psi_i = i$ – если такой элемент есть, то есть и ребенок, который ушел в своей паре ботинок. Реализация алгоритмов как вычисления точного ответа представлена на листинге 1, эксперимента – на листинге 2.

```python
# Листинг 1
def derangement(n: int):
    if n == 1:
        return 0
    if n == 0 or n == 2:
        return 1
    return (n - 1) * (derangement(n - 2) + derangement(n - 1))


def get_expected(n: int):
    result = 0
    for k in range(1, n + 1):
        for s1 in range(0, n + 1 - k):
            for s2 in range(0, n + 1 - k):
                result += math.comb(n, k) * math.comb(n - k, s1) *
math.comb(n - k - s1, s2) * derangement(n - k - s1) * derangement(n - k -
s2)
    return 1 - result / (math.factorial(n) ** 2)
```

```python
# Листинг 2
def get_random_permutation(n: int):
    perm = list(range(n))
    random.shuffle(perm)
    return perm


def run_single_experiment(n: int):
    # Успех: у каждого ребенка хотя бы один чужой ботинок
    # Неудача: существует ребенок, у которого оба свои
    shoes_l = get_random_permutation(n)
    shoes_r = get_random_permutation(n)
    for i in range(n):
        if shoes_l[i] == i and shoes_r[i] == i:
            return False
    return True
```

На рисунке 1 представлены графики $\frac{\text{количество успехов}}{\text{количество экспериментов}}$ для различных параметров эксперимента. При количестве экспериментов, стремящемся к бесконечности, значение дроби стремится к вероятности успеха.

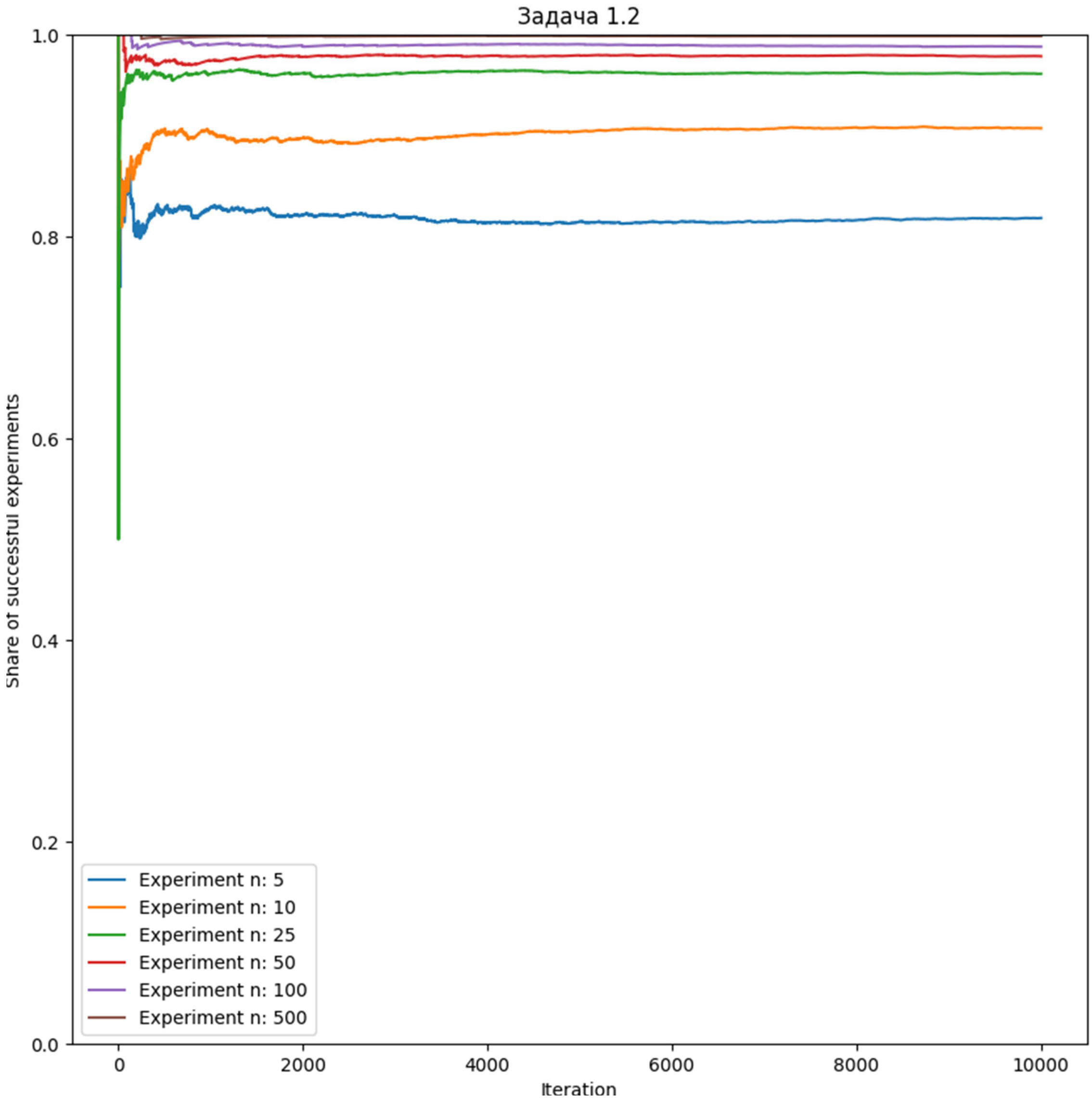

Рисунок 1

Вычисленные значения вероятности по полученной аналитически формуле для представленных на рисунке n (округлены до 3 знаков после запятой): 0.823,

0.905, 0.961, 0.980, 0.990 – сходится с полученными в ходе эксперимента результатами.

Задача 2, Вариант 1

Условие: На части параболы $y = x^2, x \in (0, 2)$ случайно выбирается точка. Как можно интерпретировать "случайность" в данной задаче? С какой вероятностью угол, образованный радиус-вектором выбранной точки с положительным направлением оси абсцисс, не превосходит $\pi/3$?

Аналитическое решение:

1. «Интерпретировать случайность» означает построить вероятностное пространство – $\langle \Omega,\ \mathbb{A},\ P: \mathbb{A} \to \mathbb{R}_+ \rangle$, где $\mathbb{A}$ – некоторая σ-алгебра.

   a. Одним из примеров подходящих к задаче вероятностных пространств является $\langle (0, 2), \mathbb{B}, P \sim U(0, 2) \rangle$, где $\mathbb{B}$ – борелевская σ-алгебра. В этом вероятностном пространстве x равновероятно выбирается из интервала $(0, 2)$.

   b. Заметим, что на интервале $(0, 2)$ функция $y = x^2$ биективна, а значит можно выбирать y: $\langle (0, 4), \mathbb{B}, P \sim U(0, 4) \rangle$ – y равновероятно выбирается из интервала $(0, 4)$.

   c. Заметим, что на промежутке $[0,\ {}^{\pi}/_{2})$ функция tg тоже биективна. Тогда, обозначив как α искомый угол, можем выбирать значение этого угла. Тогда $tg(\alpha) = {}^{y}/_{x} = {}^{x^2}/_{x} = x$. Получив x можем получить и y. Из предыдущего равенства получаем $\alpha \in (arctg(0), arctg(2))$, и вероятностное пространство $\langle (0, arctg(2)), \mathbb{B}, P \sim U(0, arctg(2)) \rangle$.

   d. В рассмотренных выше вероятностных пространствах можно выбирать значения переменных не только равновероятно, но и иными способами.

2. В качестве вероятностного пространства выберу рассмотренное в случае 1a: равновероятно выбирается $x \in (0,2)$. Воспользуемся полученным ранее равенством $tg(\alpha) = x$ и преобразуем:

$$\alpha \leq {}^{\pi}/_{3}$$

$$tg(\alpha) \leq \sqrt{3}$$

$$x \leq \sqrt{3}$$

$$P(x \le \sqrt{3}) = \frac{\mu(0,\sqrt{3})}{\mu(0,2)} = \frac{\sqrt{3}}{2}$$

Ответ: $\frac{\sqrt{3}}{2}$

Симуляция:

В каждом эксперименте равновероятно выберем $x \in (0,2)$, получим значение y, используя функцию арктангенса получим значение угла α, которое и сравним с $\pi/3$. Код представлен на листинге 3, на рисунке 2 – результат эксперимента.

```
# Листинг 3
def run_single_experiment():
    # Успех: угол не превосходит pi/3
    x = random.uniform(0, 2)
    while x == 0 or x == 2:
        x = random.uniform(0, 2)
    y = x ** 2
    angle = math.atan2(y, x)
    return angle <= math.pi / 3
```

Эксперимент подтвердил полученный аналитически результат.

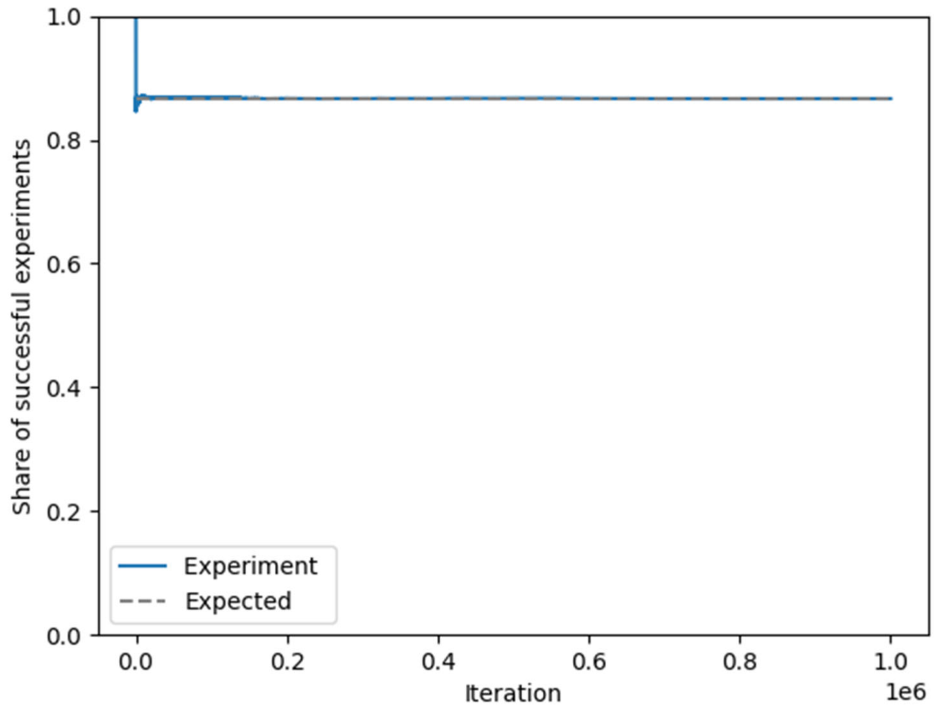

Рисунок 2 – результаты эксперимента задачи 2

Задача 3, вариант 3

Условие: Введем события $A_i = \{X = i\}, B_i = \{Y = i\}, i \geq 0$. Известно, что для любых $i \geq 0$ и $j \geq 0$ события A_iи B_i – независимы, при этом

$$P(X = i) = P(Y = i) = (1 - p)^i p, i \geq 0,$$

где $p \in (0, 1)$. Найти $P(X = i\,|X + Y = j)$.

Аналитическое решение:

Обозначим $q = 1 - p$ и преобразуем по определению условной вероятности:

$$P(X = i\,|X + Y = j) = \frac{P(X = i\,\&\,Y = j - i)}{P(X + Y = j)} =$$

$$= \frac{P(X = i) * P(Y = j - i)}{\sum_{k=0}^{j} P(X = k\,\&\,Y = j - k)} = \frac{q^i p * q^{j-i} p}{\sum_{k=0}^{j} q^k p * q^{j-k} p} =$$

$$= \frac{p^2 q^j}{\sum_{k=0}^{j} p^2 q^j} = \frac{1}{j + 1}$$

Ответ: ${}^{1}\!/_{j + 1}$

Симуляция:

Я не нашел в стандартной библиотеке языка Python 3 функции симуляции представленного в задаче распределения случайной величины, поэтому пришлось реализовать самому: равновероятно выбирается точка $x \in [0, 1]$, отрезок $[0, 1]$ разбивается на отрезки длин $p, pq, pq^2, \dots$, после этого ищу номер того отрезка, в который попала точка x – это и есть искомое значение случайной величины.

В рамках эксперимента строю значения X, Y, если $X + Y \neq j$, повторяю эксперимент снова, выбирая новые значения X, Y. Как только условие выполнено в качестве результата эксперимента возвращаю результат проверки $X = i$.

Исходный код функции, проводящей один эксперимент, представлен на листинге 4.

```python
# Листинг 4
def run_single_experiment(p: float, i: int, j: int):
    # Успех: X = i
    # Предусловие: X + Y = j
    X = get_x(p)
    Y = get_x(p)
    while X + Y != j:
        X = get_x(p)
        Y = get_x(p)
    return X == i
```

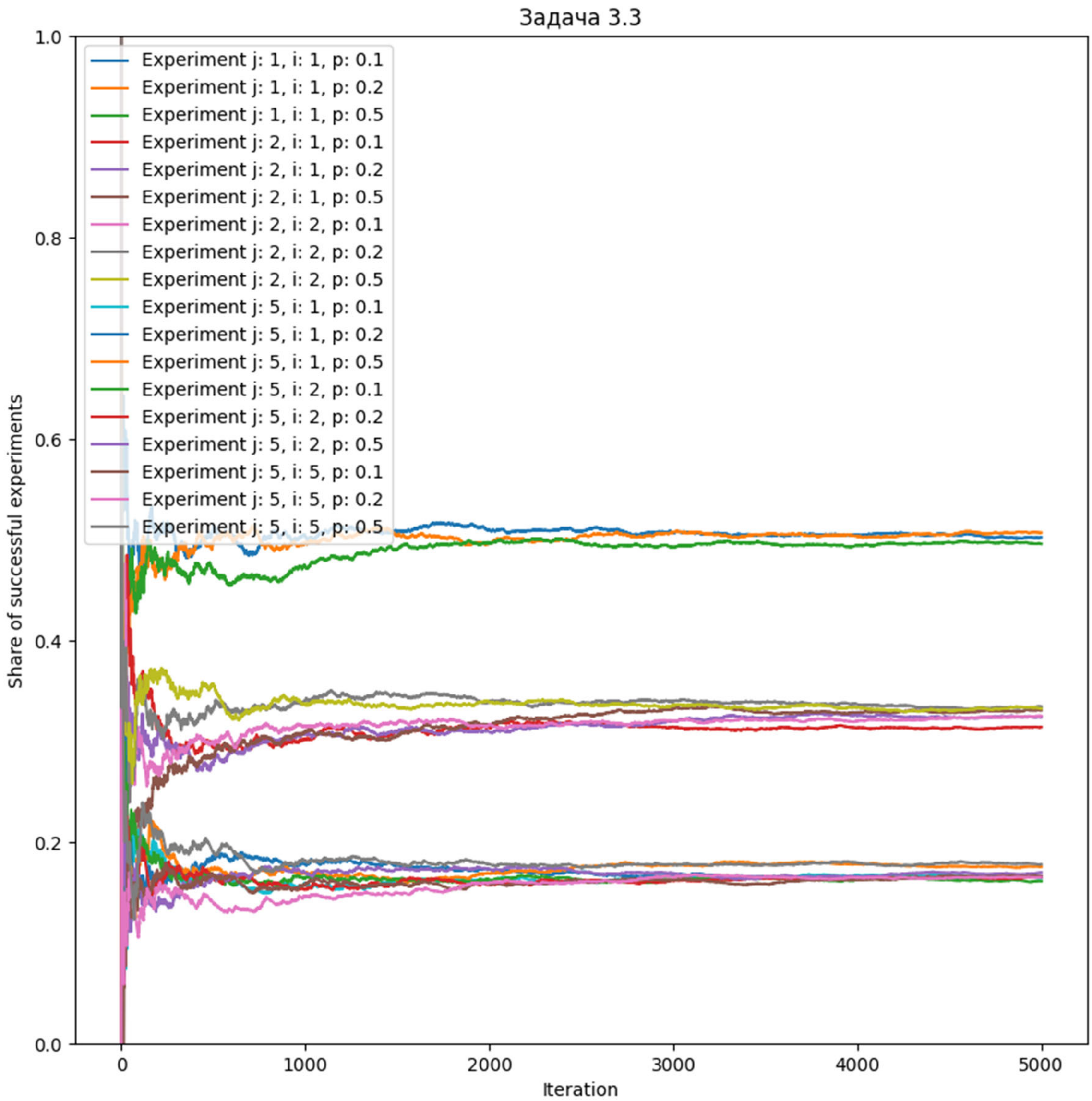

Рисунок 3

Для анализа выберу несколько значений: $p \in \{0.1, 0.2, 0.5\}$, $i, j \in \{1,2,5\}, i \leq j$: случай $i > j$ проверил отдельно – результат совпал с ожидаемым, вероятность оказалась равна 0, так как в таком случае величина Y должна иметь отрицательное значение. Для остальных значений графики сходимости представлены на рисунке 3.

С одной стороны, хотелось бы проверить поведение для большего числа значений параметров, но и при всего трех значениях каждого параметра цвета на графике уже начали повторяться. Тем не менее видно, что для $j = 1$ (первые 3 графика – синий, оранжевый, зеленый) вероятность оказывается около 0.5, что совпадает с аналитически полученным результатом. То же верно и для $j = 2$, чьи графики сходятся около значения $1/_3 \approx 0.33$, и для $j = 5$ – около $1/_6 \approx 0.166$.

Задача 4

Условие: Рассмотрите схемы Бернулли при $n \in \{10, 100, 1000, 10000\}$ и $p \in \{0.001, 0.01, 0.1, 0.25, 0.5\}$ и рассчитайте точные вероятности (где это возможно) $P\left(S_n \in \left[{}^{n}/_{2} - \sqrt{npq}, {}^{n}/_{2} + \sqrt{npq}\right]\right)$, где S_n – количество успехов в n испытаниях, и приближенную с помощью одной из предельных теорем. Сравните точные и приближенные вероятности. Объясните результаты.

Точное решение:

Для каждого n, p посчитаем (в `float64`) границы диапазона для S_n, пересечем с $\mathbb{Z}$, для каждого i из полученного множества посчитаем вероятность $P(S_n = i)$, сложим по всем i, так как события дизъюнктны. Вычислить вероятность можно точно в рациональных числах, так как $P(S_n = i) = \binom{n}{i} p^i (1-p)^{n-i}$, а операции умножения и возведения в целую степень рациональных чисел дают рациональный результат, факториал целых чисел тоже целый.

Здесь приведу только некоторые результаты, так как числа получаются длинные:

- `n=10, p=1/2: range is (3.419, 6.581) or [4, 6],`

  `Exact result: 21/32`
- `n=10, p=1/4: range is (3.631, 6.369) or [4, 6],`

  `Exact result: 28917/131072`
- `n=10, p=1/10: range is (4.051, 5.949) or [5, 5]`

  `Exact result: 3720087/2500000000`

Полный вывод представлен в репозитории в файле `4-exact-output.txt`

Приближенное решение:

Воспользуемся интегральной теоремой Муавра-Лапласа:

$$\mathrm{P}\left(x_1 \le \frac{S_n - np}{\sqrt{npq}} \le x_2\right) \approx \frac{1}{\sqrt{2\pi}} \int\limits_{x_1}^{x_2} e^{-t^2/2} dt$$

Для начала необходимо определить, чему равны x_1, x_2:

$$x_1\sqrt{npq} + np = {}^{n}/_{2} - \sqrt{npq}$$

$$x_1 = \frac{n}{2\sqrt{npq}} - \frac{np}{\sqrt{npq}} - 1$$

$$x_1 = \frac{\sqrt{n}}{2\sqrt{pq}} - \frac{\sqrt{np}}{\sqrt{q}} - 1$$

$$x_1 = \frac{\sqrt{n} - 2p\sqrt{n}}{2\sqrt{pq}} - 1$$

$$x_1 = \frac{(1-2p)}{2\sqrt{pq}} - 1,\ x_2 = \frac{(1-2p)}{2\sqrt{pq}} + 1$$

Реализация функции $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$ доступна в Python 3 в модуле `statistics`, как `statistics.NormalDist().cpf(x)`. В программе вычисляю значения x_1, x_2 по приведенной формуле выше, вызываю и возвращаю `cpf(x2)-cpf(x1)`.

Разница (по абсолютной величине) между точным (преобразованным в `float64`) и приближенным значениями представлена в таблице 1. Замечу, что 0 не означает на самом деле 0, но число настолько малое, что оно меньше самого маленького `float64`.

	p				
n	**0.5**	**0.25**	**0.1**	**0.01**	**0.01**
10	0.026439	0.018502	0.000839	2.40E-08	2.51E-13
100	0.046057	3.35E-06	3.61E-21	6.10E-72	9.60E-122
1000	0.009626	1.50E-58	1.33E-215	0	0
10000	0.004815	0	0	0	0

Таблица 1

Примечания по реализации

Исходный код программ написан на языке Python 3 и доступен по ссылке https://github.com/ChechevatovR/teorever-1. В файле `runner.py` расположен общий для нескольких задач код: запуска множественных экспериментов с различными параметрами, и построения графиков.