

Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
Высшая школа интеллектуальных систем и суперкомпьютерных технологий

### **Расчетное задание №1**

Тема: Вытягивание шаров из случайно выбранной корзины. Угадывание  
корзины.

Дисциплина: Теория вероятностей и математическая статистика

Выполнил студент гр. 3530901/10003 \_\_\_\_\_ Сергиенко К. А.  
(подпись)

Руководитель \_\_\_\_\_ Никитин К. В.  
(подпись)

“ \_\_\_\_ ” \_\_\_\_\_ 2023 г.

Санкт-Петербург  
2023

## Оглавление

Описание .....	3
Задание .....	3
Анализ задания .....	4
Часть 1 .....	4
Часть 2 .....	4
Практическое решение .....	5
Результат .....	5
Часть 1 .....	5
Часть 2 .....	6
Вывод.....	9
Приложение .....	10
Листинг .....	10

## Описание

Имеется  $N$  корзин, в каждой из которой находится известная комбинация шаров различного цвета. Игрок вначале случайно выбирает одну из корзин и затем последовательно  $K$  раз вынимает из нее по несколько  $d = 2 \div 3$  шаров с возвращением. Перед каждым подходом игрока к корзине ведущий с заданной вероятностью может случайно заменить ее на какую-то другую.

По истечении опытов игроку необходимо определить, к какой из  $N$  корзин он скорее всего подходил

## Задание

1а. После каждого опыта необходимо вычислить ряд распределения апостериорных вероятностей гипотез – к какой из корзин подошел игрок. Представить соответствующие результаты визуально на графике в форме изменения с течением опытов диаграмм распределений вероятностей гипотез. Зам. В данной задаче всего  $N$  гипотез.

1б. Определять после каждого извлечения, какие корзины имеют наибольшую вероятность. Визуализировать эволюцию изменения наиболее вероятной корзины.

1с. Построить зависимость числа превалирующих гипотез от числа проведенных опытов.

2а. Определить приближенно частоту вынимания шаров каждого цвета из корзины (экспериментальный профиль). Рассчитать теоретические вероятности вынимания шаров каждого цвета из каждой корзины –получится теоретических профилей для каждой корзины.

2б. Сравнить теоретический профиль с каждым из полученных экспериментальными найти наиболее похожий. Сравнить с полученным результатов в п. 1

2с. Построить графики изменения экспериментальных профилей для различного количества опытов

### Анализ задания

#### Часть 1

Вероятность выбора  $i$  корзины:

$$P(H_i) = \frac{1}{N}$$

Вероятность вытащить комбинацию из  $i$  корзины:

$$P(A|H_i) = \frac{\prod C^s}{T^d}$$

Вероятность достать комбинацию из любой корзины:

$$P(A) = \sum_{i=1}^N P(H_i) * P(A|H_i) = \frac{1}{N} \sum_{i=1}^N P(A|H_i)$$

Апостериорная вероятность без учета вероятности подмены корзины

$$P(H_i|A) = \frac{P(H_i) * P(A|H_i)}{P(A)} = \frac{P(A|H_i)}{\sum_{j=1}^N P(A|H_j)}$$

С учетом вероятности подмены корзины  $p$ :

$$P(H_i|A) = \frac{P(A|H_i) * (1 - p) + \frac{p}{N-1} \sum_{k=1}^N P(A|H_k)_{k \neq i}}{\sum_{j=1}^N P(A|H_j)}$$

#### Часть 2

Математический профиль представляет из себя  $\frac{color\_count(m)}{T}$  для каждого из  $m$  цветов.

## Практическое решение

Обрабатывать данные, полученные в ходе эксперимента, будем при помощи программы на языке python с использованием следующих библиотек:

- `linesache` для считывания файла построчно
- `matplotlib` для построения графиков
- `numpy` для работы с данными
- `re` для обработки входных данных

Код программы с комментариями приведен в листинге.

## Результат

### Часть 1

Согласно графикам (рисунки 1 – 3) приблизительно к 30 шагу опыта, остается одна превалирующая гипотеза, а к 50 вероятность соответствующей гипотезы становится равна 1. Согласно этому, шары вытаскивали из 6 корзины.

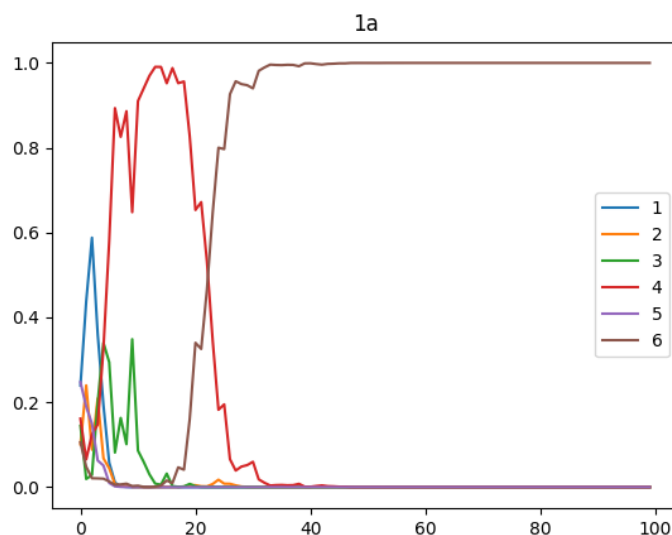


Рис. 1. График изменения вероятностей гипотез с течением опыта

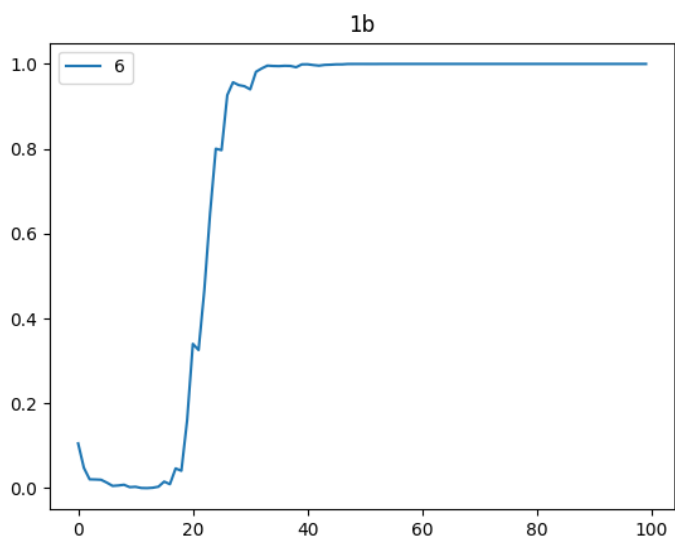


Рис. 2. Наиболее вероятная гипотеза

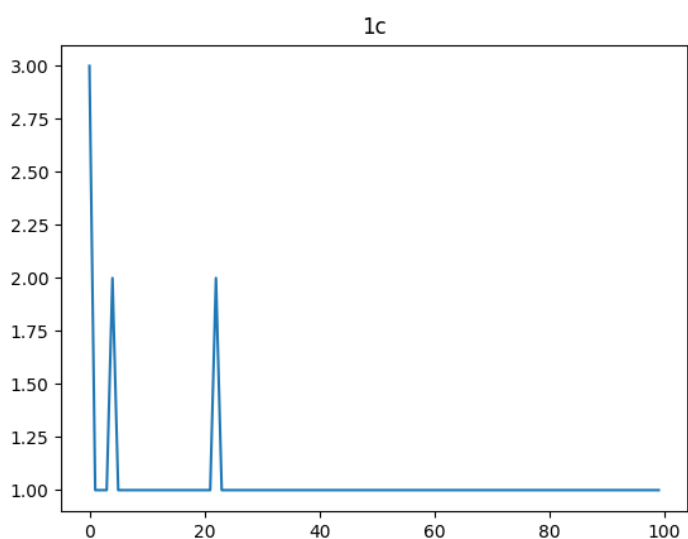


Рис. 3. Зависимость числа превалирующих гипотез

## Часть 2

Сравнивая полученный математический профиль (рисунок 5) с профилями корзин (рисунок 4), видно, что он схож с последней 6 корзиной. Согласно графику (рисунок 6), такой математический профиль сформировался приблизительно к 4000 шагу.

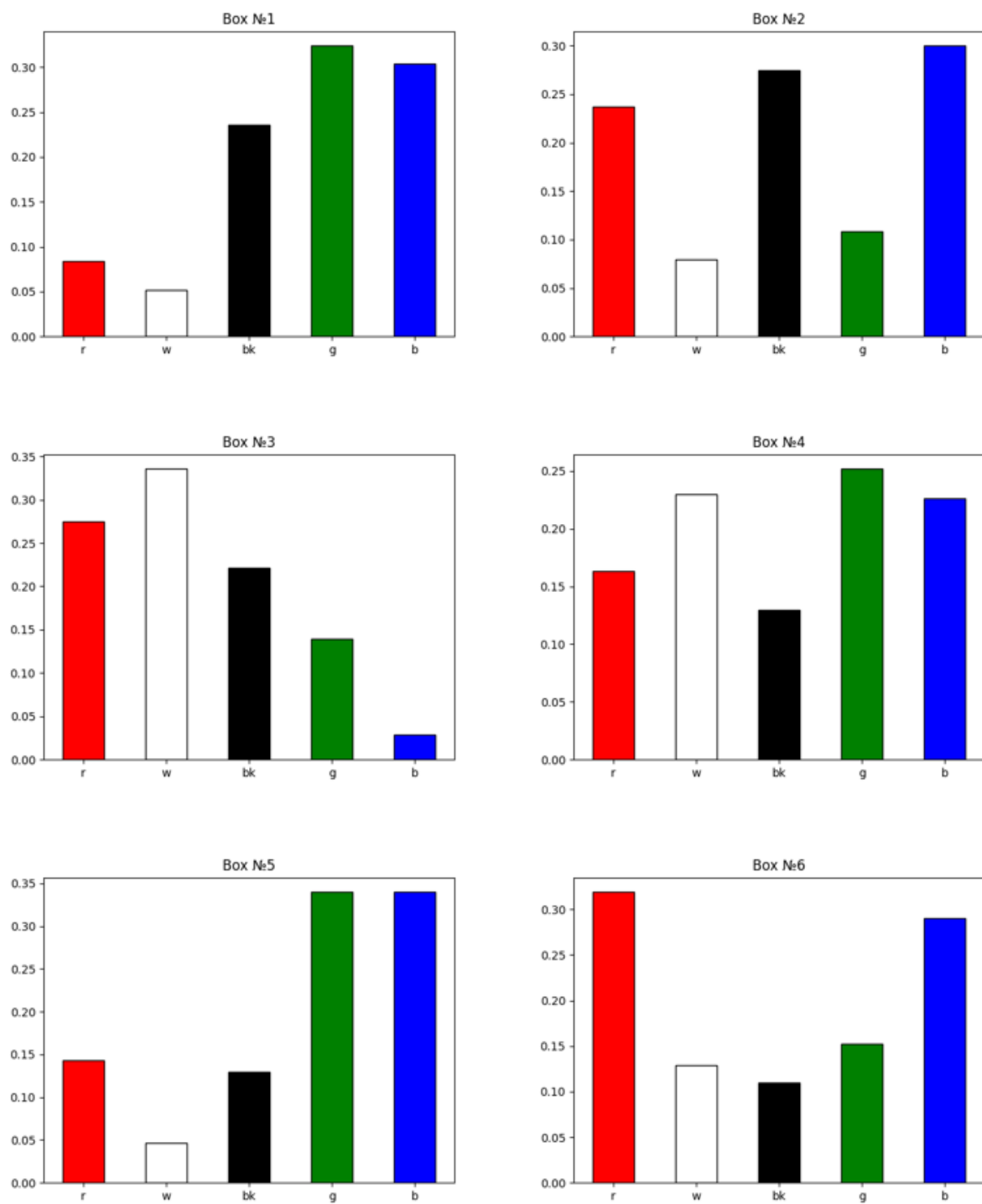


Рис. 4. Математические профили корзин

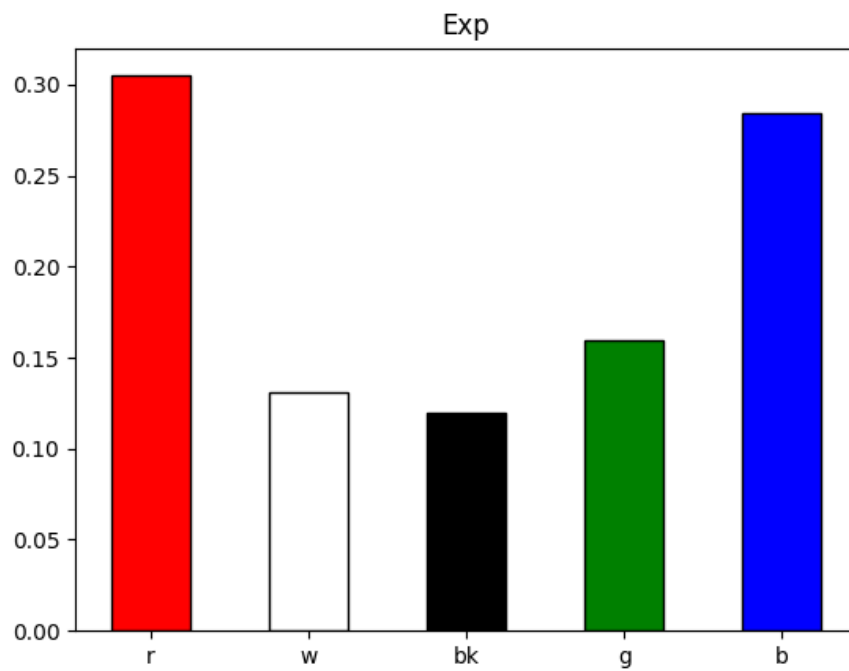


Рис. 5. Экспериментально полученный математический профиль

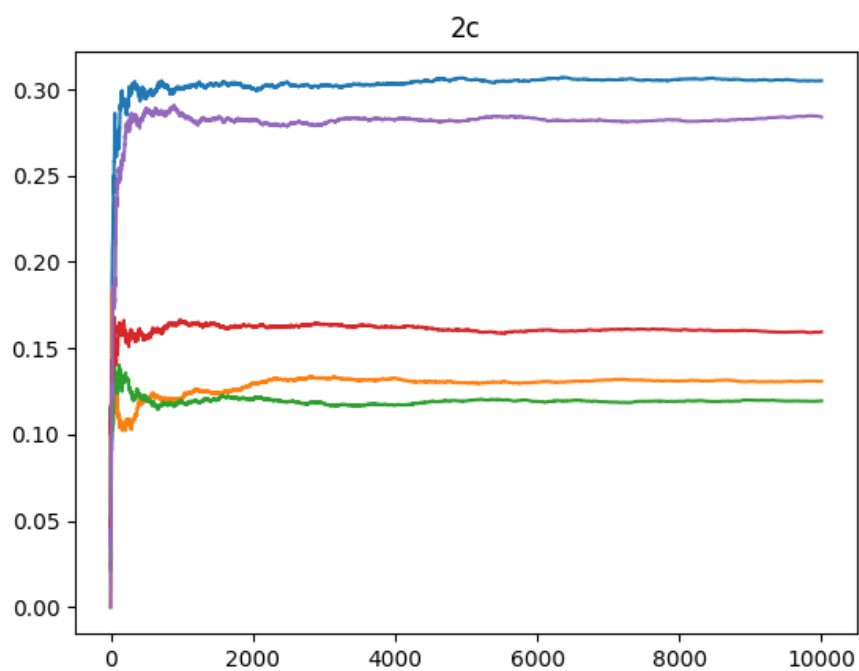


Рис. 6. Изменение математического профиля с течением опыта



## **Вывод**

В результате исследования опыта была написана программа, строящая графики, указанные в задании. Исходя из полученных результатов, скорее всего игрок подходил к 6 корзине. Это можно определить как по полученным вероятностям, подсчитанным при помощи формулы Байеса, так и по математическим профилям.

# Приложение

## Листинг

```
import linecache
from matplotlib import pyplot
import numpy as np
import re

# класс корзины, который содержит
# номер - 1,
# общее количество шаров,
# количество шаров соответствующего цвета,
# математический профиль корзины
class Box:
    def __init__(self, n, t, balls):
        math_profile = []
        for c in balls:
            math_profile.append(c / t)

        self.n = int(n)
        self.T = int(t)
        self.balls = balls
        self.P = math_profile

# Построение графиков
def plot_graphs():
    pyplot.title('1a')
    pyplot.plot(plot_res)
    pyplot.legend(np.arange(1, N + 1, 1))
    pyplot.show()

    pyplot.title('1b')
    pyplot.plot(plot_mod)
    pyplot.legend([mod + 1])
    pyplot.show()

    pyplot.title('1c')
    pyplot.plot(plot_dom)
    pyplot.show()

    pyplot.title('2c')
    pyplot.plot(exp_prof)
    pyplot.show()

# Визуализация математических профилей
def plot_hists():
    c = ['r', 'w', 'bk', 'g', 'b']
    for box in Boxes:
        pyplot.title(("Box №" + str(box.n + 1)))
        pyplot.bar(c, box.P, width=0.5, color=['red', 'white', 'black',
'green', 'blue'],
                    edgecolor='black')
        pyplot.show()

    pyplot.title("Exp")
    pyplot.bar(c, exp_prof[-1], width=0.5, color=['red', 'white', 'black',
'green', 'blue'],
                edgecolor='black')
    pyplot.show()
```

```

# Получение и обработка входных параметров
def getInitData():
    global N, m, d, p_change, nExp
    init_data = linecache.getline('task_1_ball_boxes.txt', 2).split(',')

    N = int(re.sub("\D", "", init_data[0]))
    m = int(re.sub("\D", "", init_data[1]))
    d = int(re.sub("\D", "", init_data[2]))
    p_change = float(re.sub("[^\d.]", "", init_data[3]))
    nExp = int(re.sub("\D", "", init_data[4]))

    boxes_data = []
    for i in range(0, N):
        cur_box = linecache.getline('task_1_ball_boxes.txt', 3 + i).split('
')

        n = int(re.sub("\D", "", cur_box[1]))
        T = int(re.sub("\D", "", cur_box[3]))
        r = int(re.sub("\D", "", cur_box[5]))
        w = int(re.sub("\D", "", cur_box[7]))
        bk = int(re.sub("\D", "", cur_box[9]))
        g = int(re.sub("\D", "", cur_box[11]))
        b = int(re.sub("\D", "", cur_box[13]))
        colors = [r, w, bk, g, b]

        boxes_data.append(Box(n - 1, T, colors))

    return boxes_data

# Подсчет вероятности достать комбинацию шагов exp_balls на k шаге
эксперимента для корзины box
def get_prob(box, exp_balls):
    c = 0
    res = 1
    for color in box.balls:
        res *= color ** exp_balls[c]
        c += 1
    res /= box.T ** 4
    return res

# Получение превалирующих гипотез на k шаге эксперимента
def get_dom():
    cur_dominant = np.zeros(N)
    max_prob = max(PAHi)
    diff = 0.1

    for i in range(0, len(PAHi)):
        p = PAHi[i]
        if max_prob - p < diff:
            cur_dominant[i] = 1
    return cur_dominant

# Получение результатов k-ого шага эксперимента
def get_k_res(x, balls):
    global p_change, PAHi, total_balls

    # Обработка полученных данных
    k = linecache.getline('task_1_ball_boxes.txt', x).split(' ')
    for j in range(1, 5):
        if k[len(k) - j].__contains__('Red'):
            balls[0] += 1
        elif k[len(k) - j].__contains__('White'):
            balls[1] += 1
        elif k[len(k) - j].__contains__('Black'):

```

```

        balls[2] += 1
    elif k[len(k) - j].__contains__('Green'):
        balls[3] += 1
    elif k[len(k) - j].__contains__('Blue'):
        balls[4] += 1

# Подсчет вероятностей гипотез
for box in Boxes:
    t = get_prob(box, balls) * (1 - p_change)
    inner_t = 0
    for inner_box in Boxes:
        if inner_box.n != box.n:
            inner_t += get_prob(inner_box, balls)
    inner_t *= p_change / (N - 1)
    t += inner_t
    PAHi[box.n] *= t

s = sum(PAHi)

res = []
for i in range(0, len(Boxes)):
    PAHi[i] /= s
    res.append(PAHi[i])
return res

# Подсчет математического профиля эксперимента на k шаге
def get_prof(x, cur_balls, cur_prof):
    res = list(cur_prof)
    for i in range(0, len(cur_balls)):
        res[i] = cur_balls[i] / (x * d)
    return res

# Получение полных результатов эксперимента
def get_exp_res():
    total_balls, counter_balls, cur_prof = np.zeros(5), np.zeros(5),
np.zeros(5)
    prob, dominant, prof = [], [], []
    for i in range(10, nExp + 10):
        prob.append(get_k_res(i, counter_balls))
        total_balls += counter_balls
        prof.append(get_prof(i, total_balls, cur_prof))
        dominant.append(get_dom())
        counter_balls = np.zeros(5)
    return prob, dominant, prof

if __name__ == "__main__":

    Boxes = getInitData()

    PAHi = np.ones(N)

    exp_prob, exp_dominant, exp_prof = get_exp_res()
    mod = int(np.argmax(sum(exp_dominant)))

    # Создание списков с меньшим количеством данных для лучшей визуализации
    изменений
    plot_res, plot_dom, plot_mod = [], [], []
    for i in range(0, 100):
        plot_res.append(exp_prob[i])
        plot_dom.append(sum(exp_dominant[i]))
        plot_mod.append(exp_prob[i][mod])

```

```
plot_graphs()  
plot_hists()
```