

Санкт-Петербургский политехнический университет Петра Великого
Институт компьютерных наук и кибербезопасности
Высшая школа компьютерных технологий и информационных систем

Отчёт по расчётному заданию №1

Дисциплина: Теория вероятности и математическая статистика

Выполнил студент гр. 5130901/20003 _____ Гусев М.М.
(подпись)

Доцент, к.т.н. _____ Никитин К.В.
(подпись)

“ ” _____ 2024 г.

Санкт-Петербург
2024

Задание

Ведущий загадывает число в диапазоне от 1 до N . На каждом шаге он выбирает случайную операцию (сложение, умножение, деление без остатка) и число, далее он выполняет арифметическую операцию и сообщает результат в виде одной из цифр получившегося числа, т.е. цифру и ее порядок. Также ведущий периодически может сбрасывать все, т.е. как бы начинать игру с самого начала. Нужно угадать число.

1а. После каждого k опыта необходимо вычислить ряд распределения апостериорных вероятностей гипотез по каждому варианту возможного значения числа. Представить соответствующие результаты визуально на графике в форме изменения с течением опытов диаграмм распределений вероятностей гипотез.

1b. Определять после каждой попытки, какие гипотезы имеют наибольшую вероятность. Визуализировать эволюцию изменения наиболее вероятных гипотез.

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

2. Попытаться придумать частотную статистику, на основе которой можно выделить полезную информацию о значении числа.

Прим. Пока не совсем понимаю как анализировать полученные данные с теоретической точки зрения, указал только, что распределение нормальное. То есть как вывести апостериорную вероятность по формуле Байеса? Предполагаю, что вероятность события A (результат который сообщает ведущий) не поддается оценке, т.к. он может дать любое целое число и также сообщить любую позицию одной из 10 цифр, а количество этих вариантов бесконечно.

Решение

Основной код

```
data = open('task_1_numbers.txt', 'r')

options = re.findall(r'\d+', data.readline().strip())
Nmin = int(options[0])
Nmax = int(options[1])
Nexp = int(options[2])

prior = 1 / (Nmax - Nmin + 1)
hypotheses = {i: [prior, i] for i in range(Nmin, Nmax + 1)}

_ = data.readline()
del _

new_hypotheses = {}
last_possibles = Nmax - Nmin + 1

hyps_for_draw = {i: [prior] for i in range(Nmin, Nmax + 1)}

number_of_hyps = []
for i in range(Nexp - 2):
    start = time.time()
    """ читаем строку в словарь """
    s = data.readline()
    step = re.search(r'\d+', s).group()
    start_idx = s.find("{")
    end_idx = s.find("}") + 1
    json_str = s[start_idx:end_idx].replace("'", "\'")
    exp = json.loads(json_str)
    del s
    del start_idx
    del end_idx
    del json_str

    if exp['op'] == '-':

        for hyp in hypotheses.keys():
            if hypotheses[hyp][0] != 0:
                hypotheses[hyp][1] = abs(hypotheses[hyp][1] - int(exp['num']))
                check(hypotheses, hyp, exp, new_hypotheses)

    elif exp['op'] == '+':

        for hyp in hypotheses.keys():
            if hypotheses[hyp][0] != 0:
                hypotheses[hyp][1] += int(exp['num'])
                check(hypotheses, hyp, exp, new_hypotheses)

    elif exp['op'] == '*':

        for hyp in hypotheses.keys():
            if hypotheses[hyp][0] != 0:
                hypotheses[hyp][1] *= int(exp['num'])
                check(hypotheses, hyp, exp, new_hypotheses)

    elif exp['op'] == '/':

        for hyp in hypotheses.keys():
            if hypotheses[hyp][0] != 0:
                hypotheses[hyp][1] /= int(exp['num'])
                check(hypotheses, hyp, exp, new_hypotheses)
```

```

elif exp['op'] == 'r':
    for hyp in hypotheses.keys():
        if hypotheses[hyp][0] != 0:
            hypotheses[hyp][1] = hyp

# probs = []
for hyp in hypotheses.keys():

    if hypotheses[hyp][0] != 0:
        hypotheses[hyp][0] = 1 / len(new_hypotheses)
        new_hypotheses[hyp][0] = 1 / len(new_hypotheses)

        hyps_for_draw[hyp].append(hypotheses[hyp][0])
    # probs.append(hypotheses[hyp][0])

if last_possibles != len(new_hypotheses):
    print(f'On step #{step} after "{exp["op"]}" operation number of possible
numbers: {len(new_hypotheses)}')
    last_possibles = len(new_hypotheses)

number_of_hyps.append(len(new_hypotheses))

if len(new_hypotheses) == 1:
    print(new_hypotheses)
    break
data.close()

```

Первоначально открывается файл с данными от ведущего, с первой строки читаются переменные Nmin, Nmax и Nexр, далее создается словарь (он же hash map) hypotheses, в котором ключами являются все числа заданного промежутка, а значениями – массив, на первом месте содержащий вероятность этой гипотезы, на втором изначально само число, далее оно будет изменяться в зависимости от операции, которую сообщает ведущий. Словарь new_hypotheses создан для добавления в него на каждом шаге гипотез, которые подходят по результатам опыта. В основном цикле осуществляется чтение открытого файла построчно, в словарь exp извлекаются данные, полученные от ведущего, далее в зависимости от операции осуществляется проход по всем гипотезам, если вероятность гипотезы ещё не нулевая, тогда с данным числом совершается переданная операция и проверяется соответствие условию с помощью функции check()

```

def check(hypotheses, hyp, exp, new_hypotheses):
    if exp['outType'] == 'digit':
        if not digit_on_place(str(hypotheses[hyp][1]), str(exp['outVal']),
int(exp['extraParams'])):
            hypotheses[hyp][0] = 0
            if hyp in new_hypotheses:
                del new_hypotheses[hyp]
        else:
            new_hypotheses[hyp] = hypotheses[hyp]
    elif exp['outType'] == 'sum':
        if sum(map(int, str(hypotheses[hyp][1]))) != exp['outVal']:
            hypotheses[hyp][0] = 0
            if hyp in new_hypotheses:

```

```
del new_hypotheses[hyp]
else:
    new_hypotheses[hyp] = hypotheses[hyp]
```

В зависимости от типа переданной информации проверяется либо положение цифры в получившемся числе, либо его сумма. Если число не прошло проверку, вероятность этой гипотезы приравнивается к 0, также эта гипотеза удаляется из словаря верных гипотез. Если прошло, то эта гипотеза, наоборот, добавляется в новый словарь.

В случае операции 'r' число, стоящее на втором месте в массиве-значении каждой гипотезы, приравнивается своему изначальному значению.

Далее для каждой гипотезы с ненулевой вероятностью определяется нынешняя вероятность.

Цикл завершается, как только остаётся одна верная гипотеза.

1а. Распределения гипотез

Число гипотез конечное и вероятность каждой гипотезы на каждом шаге одинаковая, значит они должны описываться равномерным распределением. Исходя из этого, вероятности гипотез на каждом шаге определяются по формуле

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

где N – общее число возможных вариантов.

График распределения до получения какой-либо информации от ведущего выглядит следующим образом:

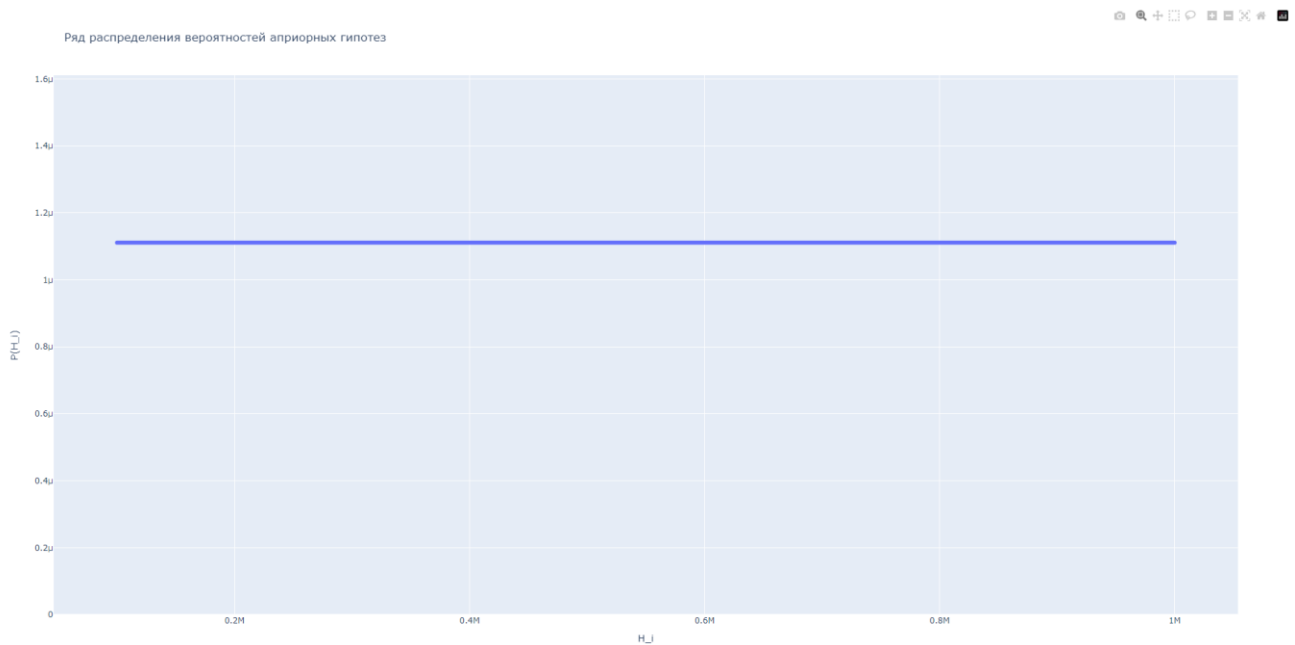


Рис. 1 Априорные вероятности каждой гипотезы

По ходу получения информации от ведущего вероятности гипотез будут меняться так: те числа, что не прошли проверку просто «выбывают», их вероятность становится равной 0, у тех же чисел, что прошли проверку, вероятность будет равной и больше, чем ранее, т.к. количество возможных вариантов уменьшилось.

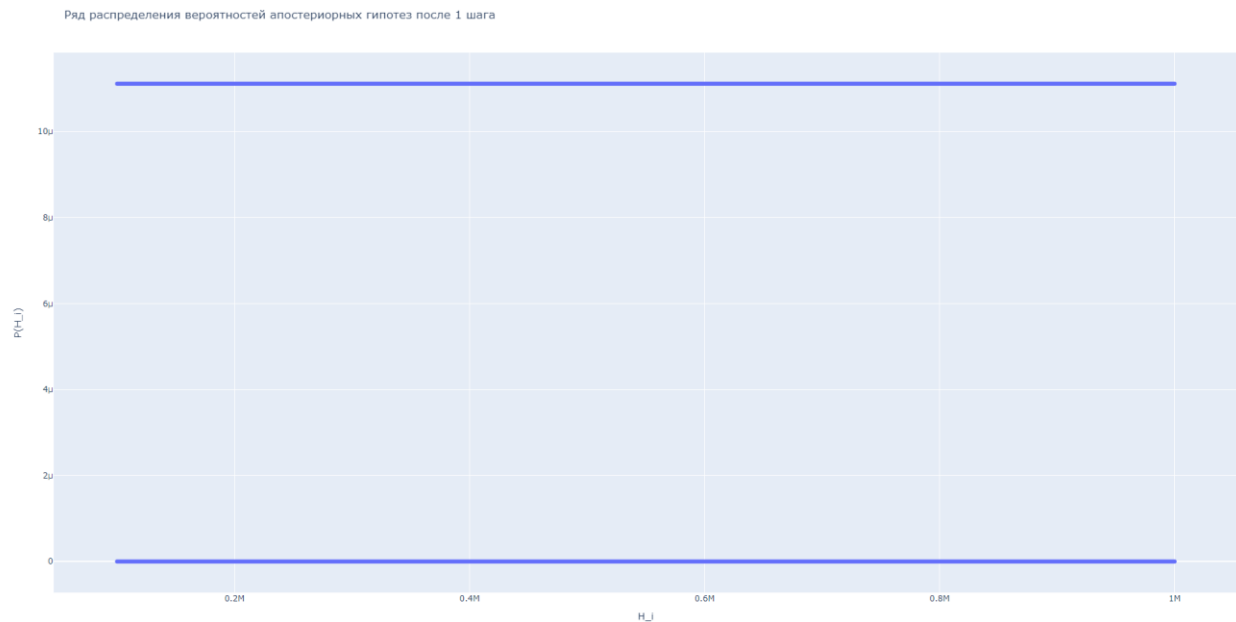


Рис. 2 Распределение гипотез после 1 опыта

На первом же шаге количество возможных чисел резко сократилось с 900001 до 89990, вероятность каждой гипотезы стала соответственно равной $1/89990$.

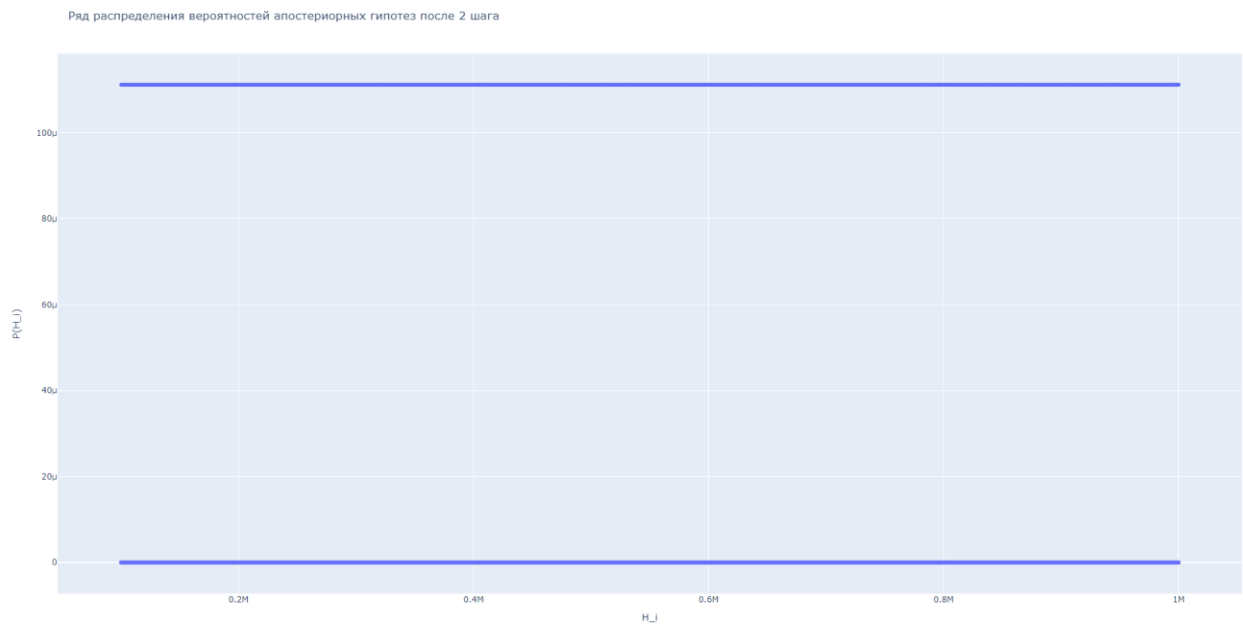


Рис. 3 Распределение гипотез после 2 опыта

Количество возможных вариантов стало равным 8997.

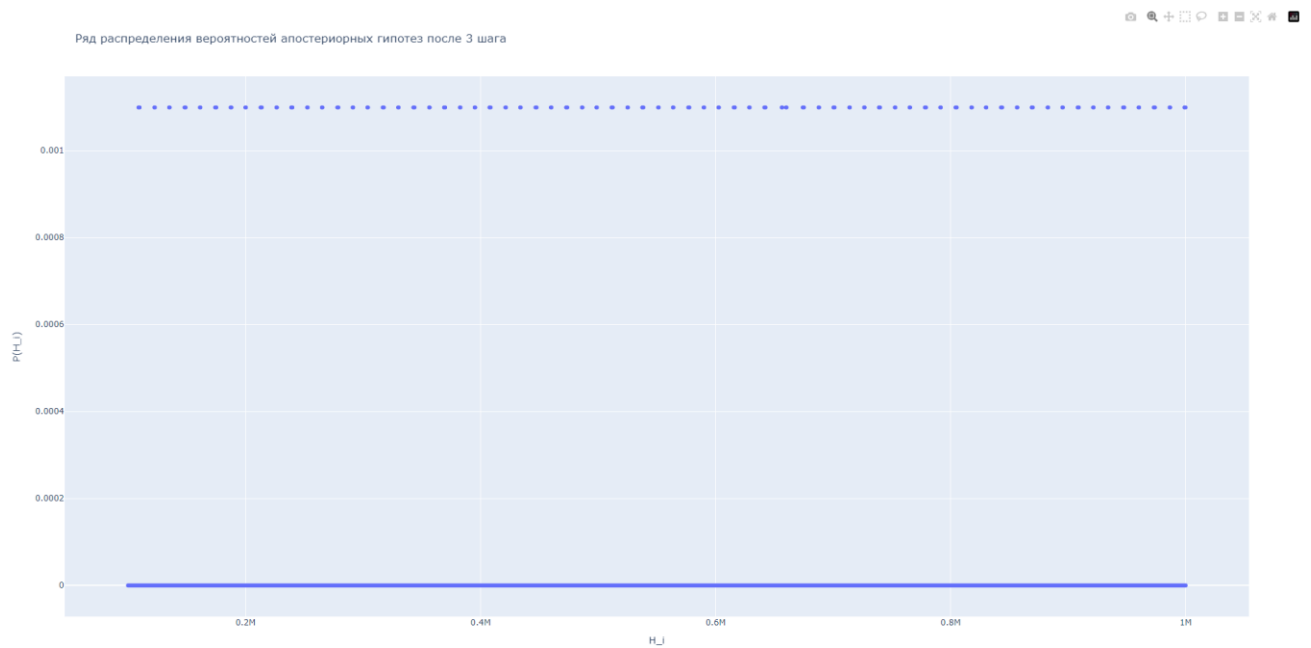


Рис. 4 Распределение гипотез после 3 опыта

Количество возможных чисел стало равным 910. Уже на 3 шаге количество чисел сократилось в 1000 раз по сравнению с изначальным количеством.

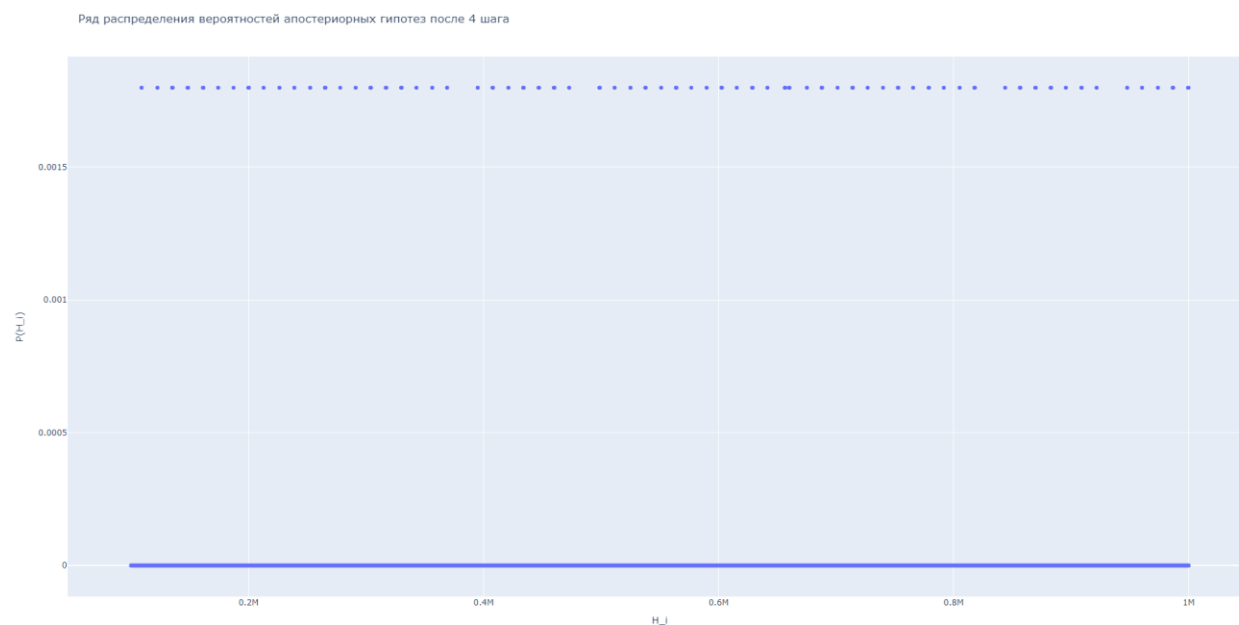


Рис. 5 Распределение гипотез после 4 опыта

Количество возможных чисел 556.

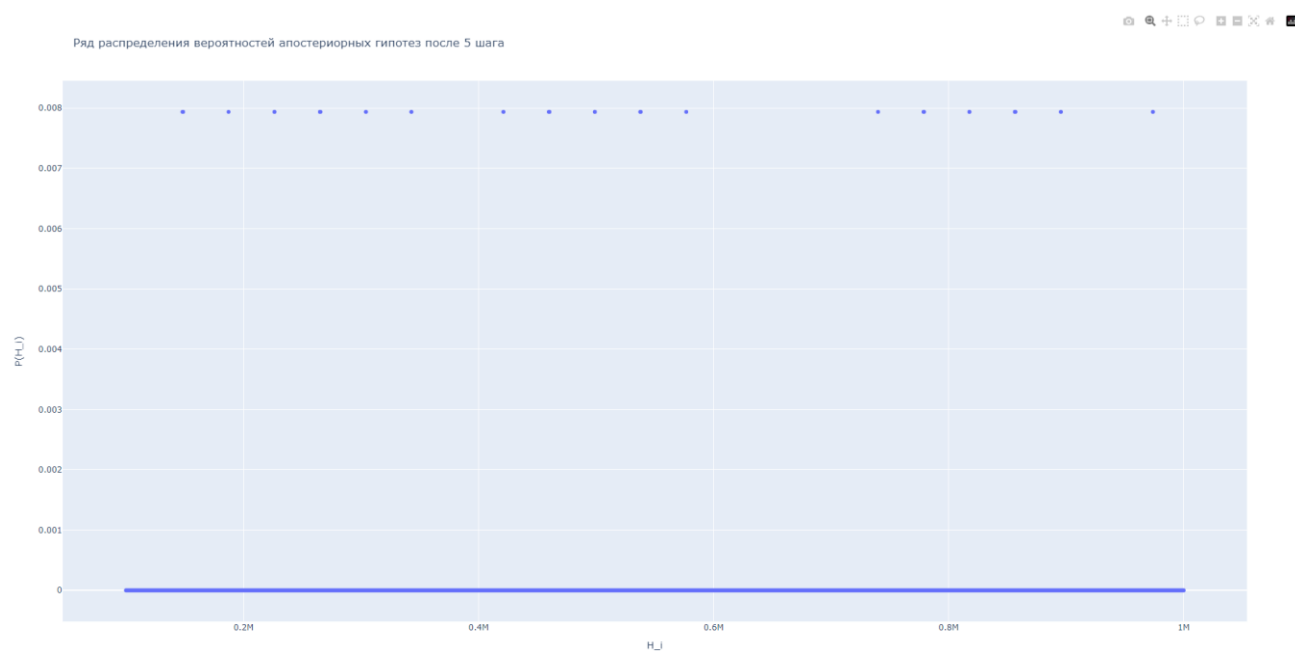


Рис. 6 Распределение гипотез после 5 опыта

Количество возможных чисел 126.

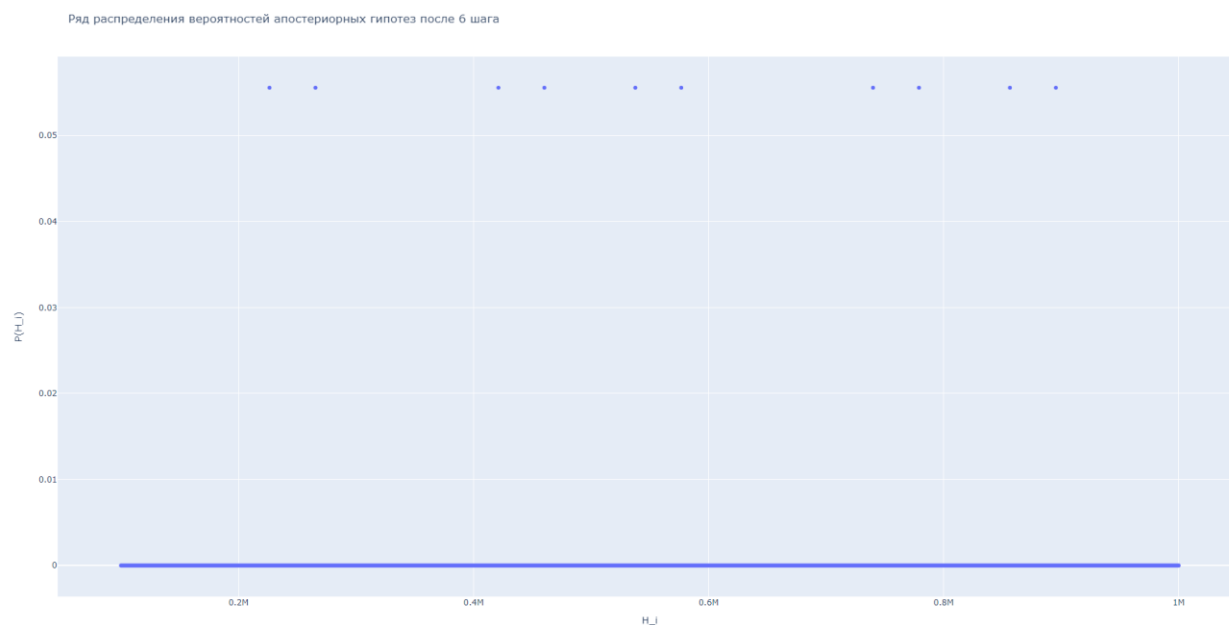


Рис. 7 Распределение гипотез после 6 опыта

Количество возможных чисел 18.

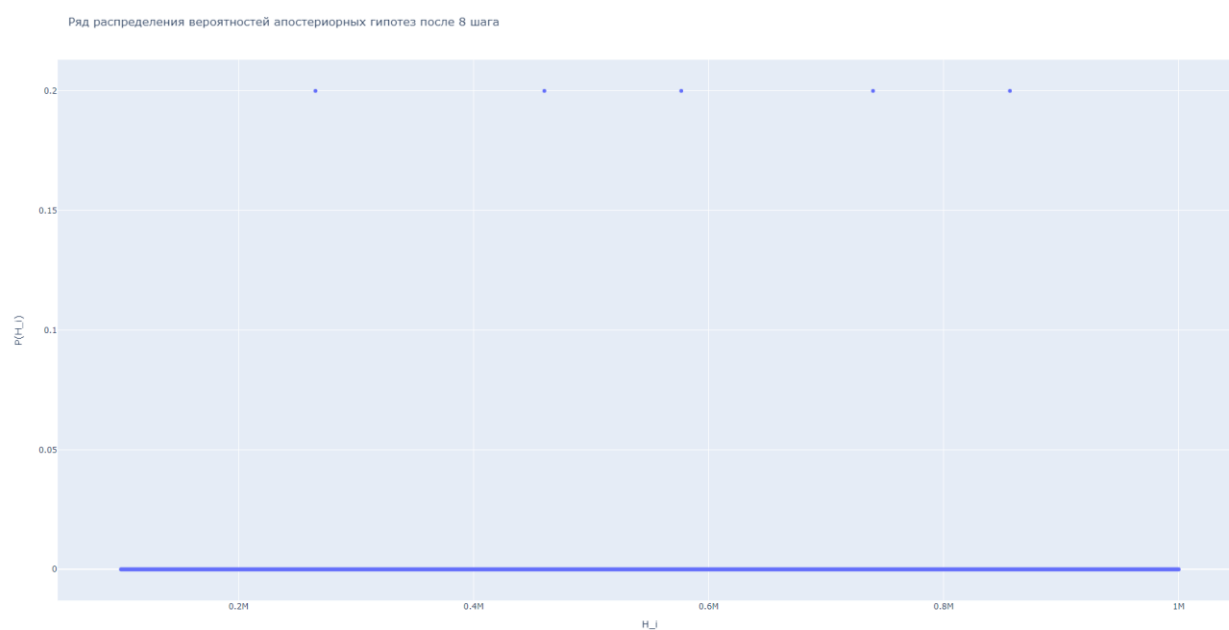


Рис. 8 Распределение гипотез после 8 опыта

Количество возможных чисел 5.

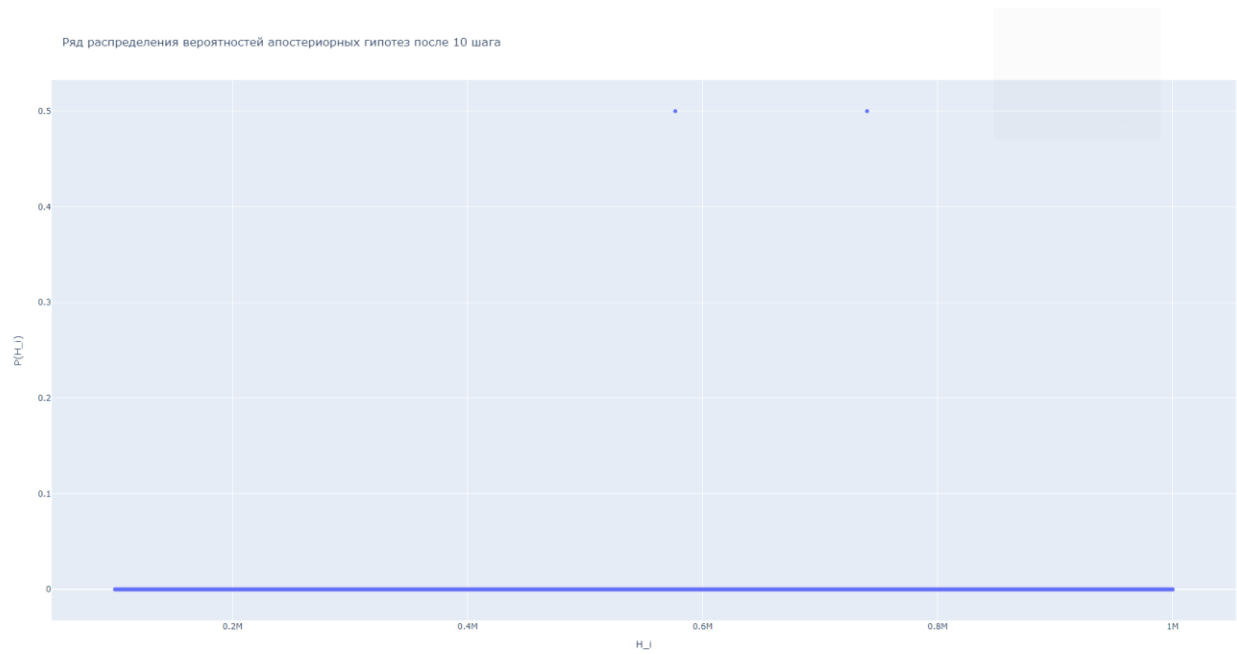


Рис. 9 Распределение гипотез после 10 опыта

Количество возможных чисел 2.

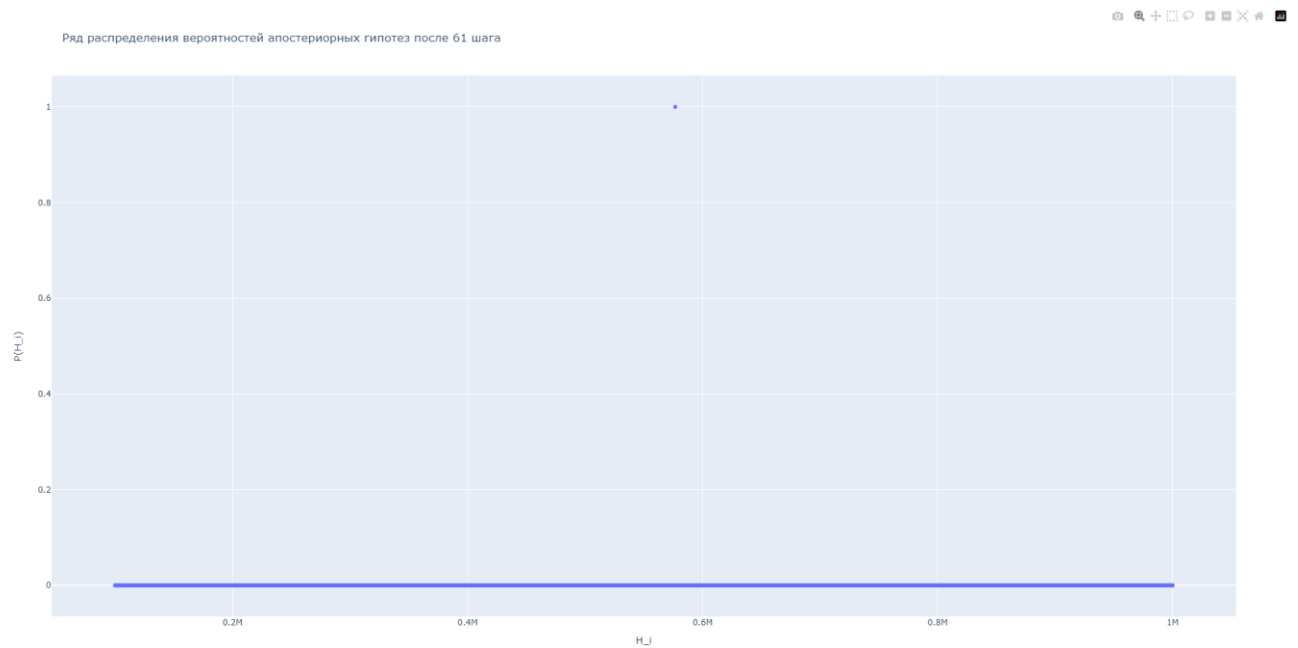


Рис. 10 Распределение гипотез после 61 опыта

Число отгадано, это 576739.

1b. Эволюцию изменения наиболее вероятных гипотез

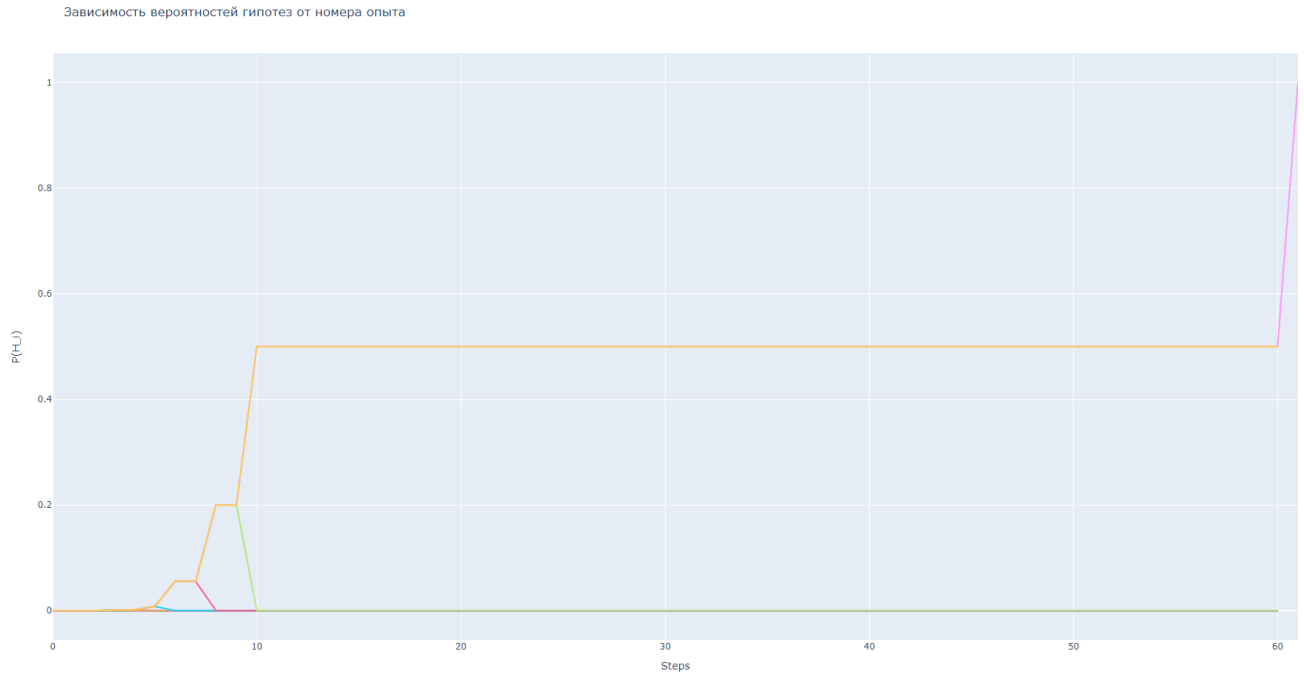


Рис. 11 Зависимость вероятностей гипотез от полученной информации

Первые 5 опытов вероятности гипотез достаточно близки к нулю, из-за большого их количества, высокий рост наблюдается после 6 опыта, когда число вариантов стало равным 18, после 8 опыта число вариантов сократилось до 5 и теперь, в связи с небольшим их количеством, можно привести эти варианты, это числа 265440, 460245, 576739, 739935 и 856429. После 10 опыта осталось всего 2 варианта: 576739, 739935. Далее на протяжении 50 опытов вероятности этих гипотез не изменялись, после 61 опыта вариант 739935 не подошёл и осталось одно число – 576739, его вероятность стала равной 1 и цикл завершился.

1с. Зависимость числа превалирующих гипотез от числа проведенных опытов

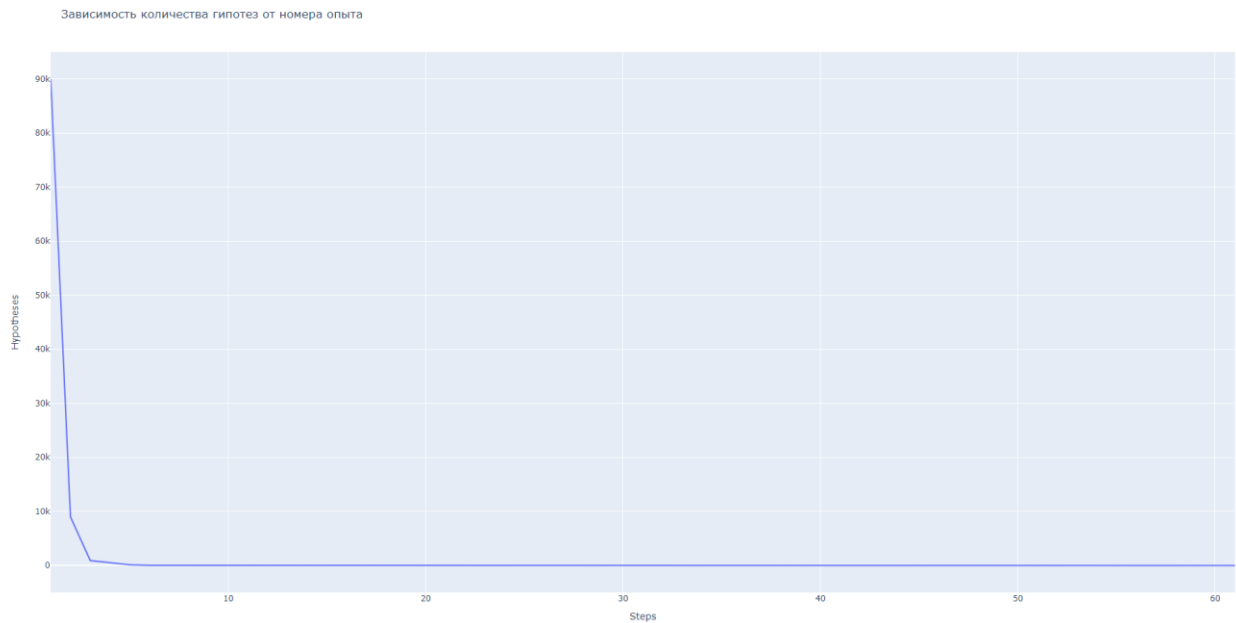


Рис. 12 Зависимость числа гипотез от числа проведенных опытов

На данном графике наглядно показано как количество возможных вариантов резко сократилось в первые 10 опытов, после чего долго оставалось равным 2.

Приложение

```
import re
import json
import plotly.graph_objects as go
import plotly.express as px
import pandas as pd

def digit_on_place(num: str, d: str, i: int):
    num = num[::-1]
    return len(num) >= i and num[i-1] == d

def check(hypotheses, hyp, exp, new_hypotheses):
    if exp['outType'] == 'digit':
        if not digit_on_place(str(hypotheses[hyp][1]), str(exp['outVal']),
int(exp['extraParams']))):
            hypotheses[hyp][0] = 0
            if hyp in new_hypotheses:
                del new_hypotheses[hyp]
            else:
                new_hypotheses[hyp] = hypotheses[hyp]
    elif exp['outType'] == 'sum':
        if sum(map(int, str(hypotheses[hyp][1]))) != exp['outVal']:
            hypotheses[hyp][0] = 0
            if hyp in new_hypotheses:
                del new_hypotheses[hyp]
            else:
                new_hypotheses[hyp] = hypotheses[hyp]

def draw_hyp_plot(x, y, k):
    data = {'x': x, 'y': y}
    df = pd.DataFrame(data)

    fig = px.scatter(df, x='x', y='y', labels={'x': 'Hi', 'y': 'P(Hi)'},
        title=f'Ряд распределения вероятностей апостериорных гипотез
после {k} шага')

    fig.show()

data = open('task_1_numbers.txt', 'r')

options = re.findall(r'\d+', data.readline().strip())
Nmin = int(options[0])
Nmax = int(options[1])
Nexp = int(options[2])

prior = 1 / (Nmax - Nmin + 1)
hypotheses = {i: [prior, i] for i in range(Nmin, Nmax + 1)}

_ = data.readline()
del _

new_hypotheses = {}
last_possibles = Nmax - Nmin + 1

hyps_for_draw = {i: [prior] for i in range(Nmin, Nmax + 1)}

number_of_hyps = []

for i in range(Nexp - 2):
```

```

""" читаем строку в словарь """
s = data.readline()
step = re.search(r'\d+', s).group()
start_idx = s.find("{")
end_idx = s.find("}") + 1
json_str = s[start_idx:end_idx].replace("'", "\"")
exp = json.loads(json_str)
del s
del start_idx
del end_idx
del json_str

if exp['op'] == '-':

    for hyp in hypotheses.keys():
        if hypotheses[hyp][0] != 0:
            hypotheses[hyp][1] = abs(hypotheses[hyp][1] - int(exp['num']))
            check(hypotheses, hyp, exp, new_hypotheses)

elif exp['op'] == '+':

    for hyp in hypotheses.keys():
        if hypotheses[hyp][0] != 0:
            hypotheses[hyp][1] += int(exp['num'])
            check(hypotheses, hyp, exp, new_hypotheses)

elif exp['op'] == '*':

    for hyp in hypotheses.keys():
        if hypotheses[hyp][0] != 0:
            hypotheses[hyp][1] *= int(exp['num'])
            check(hypotheses, hyp, exp, new_hypotheses)

elif exp['op'] == '/':

    for hyp in hypotheses.keys():
        if hypotheses[hyp][0] != 0:
            hypotheses[hyp][1] /= int(exp['num'])
            check(hypotheses, hyp, exp, new_hypotheses)

elif exp['op'] == 'r':
    for hyp in hypotheses.keys():
        if hypotheses[hyp][0] != 0:
            hypotheses[hyp][1] = hyp

# probs = []
for hyp in hypotheses.keys():

    if hypotheses[hyp][0] != 0:
        hypotheses[hyp][0] = 1 / len(new_hypotheses)
        new_hypotheses[hyp][0] = 1 / len(new_hypotheses)

        hyps_for_draw[hyp].append(hypotheses[hyp][0])
        # probs.append(hypotheses[hyp][0])

if last_possibles != len(new_hypotheses):
    print(f'On step #{step} after "{exp['op']}" operation number of possible
numbers: {len(new_hypotheses)}')
last_possibles = len(new_hypotheses)

number_of_hyps.append(len(new_hypotheses))

# if step == '61':
#     draw_hyp_plot(list(hypotheses.keys()), probs, step)

```

```

        if len(new_hypotheses) == 1:
            print(new_hypotheses)
            break

data.close()

number_of_exps = int(step)
steps = [i for i in range(0, number_of_exps+1)]

lengths = {}

for hyp in hyps_for_draw.keys():
    lengths[len(hyps_for_draw[hyp])] = hyp

fig = go.Figure()
for hyp in lengths.values():
    y = hyps_for_draw[hyp]

    for i in range(len(y), int(step)):
        y.append(0)

    fig.add_trace(go.Scatter(x=steps, y=y, mode='lines', name=str(hyp)))

fig.update_layout(title_text='Зависимость вероятностей гипотез от номера опыта')
fig.update_xaxes(title_text='Steps')
fig.update_yaxes(title_text='P(H_i)')
fig.show()

fig = go.Figure(go.Scatter(x=steps, y=number_of_hyps, mode='lines'))
fig.update_layout(title_text='Зависимость количества гипотез от номера опыта')
fig.update_xaxes(title_text='Steps')
fig.update_yaxes(title_text='Hypotheses')
fig.show()

```