

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
**«Рязанский государственный радиотехнический университет
Имени В. Ф. Уткина»**

Факультет вычислительной техники
Кафедра вычислительной и прикладной математики

Отчёт по практической работе №8

по дисциплине:
“Моделирование”

по теме:

“ Тактическое планирование эксперимента”

Выполнил: студент гр. 242
Фокин А.М.

Проверил: Анастасьев А. А.

Рязань 2025

Цель работы:

Каждое задание предполагает разработку программной имитационной модели случайного блуждания, с помощью которой могут быть получены необходимые результаты по двум частям лабораторной работы. В первой части работы выполняется планирование и получение результатов эксперимента по оценке среднего значения указанного параметра с заданной точностью при заданной достоверности . Во второй части выполняется планирование и получение результатов эксперимента для оценки с заданной точностью при заданной достоверности выборочной дисперсии указанного параметра. В процессе проведения пробного эксперимента при планировании строится статистическое распределение исследуемого параметра и определяется целесообразность аппроксимации этого распределения нормальным законом. В зависимости от этого выбирается подход для определения объема эксперимента. Варианты заданий берутся из практического занятия №7.

Модель падения дождевой капли. При воздействии случайных порывов легкого ветра падение дождевой капли можно моделировать случайным блужданием на квадратной решетке. Движение начинается с узла, расположенного на расстоянии h над горизонтальной линией (поверхностью земли). Вероятность $p\downarrow$ шага «вниз» больше вероятности $p\uparrow$ шага «вверх». Вероятности скачков целесообразно выбирать равными $p\downarrow=0,5$; $p\uparrow=0,1$; $p\leftarrow=p\rightarrow=0,2$. Определите время τ , за которое капля достигает горизонтальной прямой, и функциональную зависимость τ от h (4..6 значений)

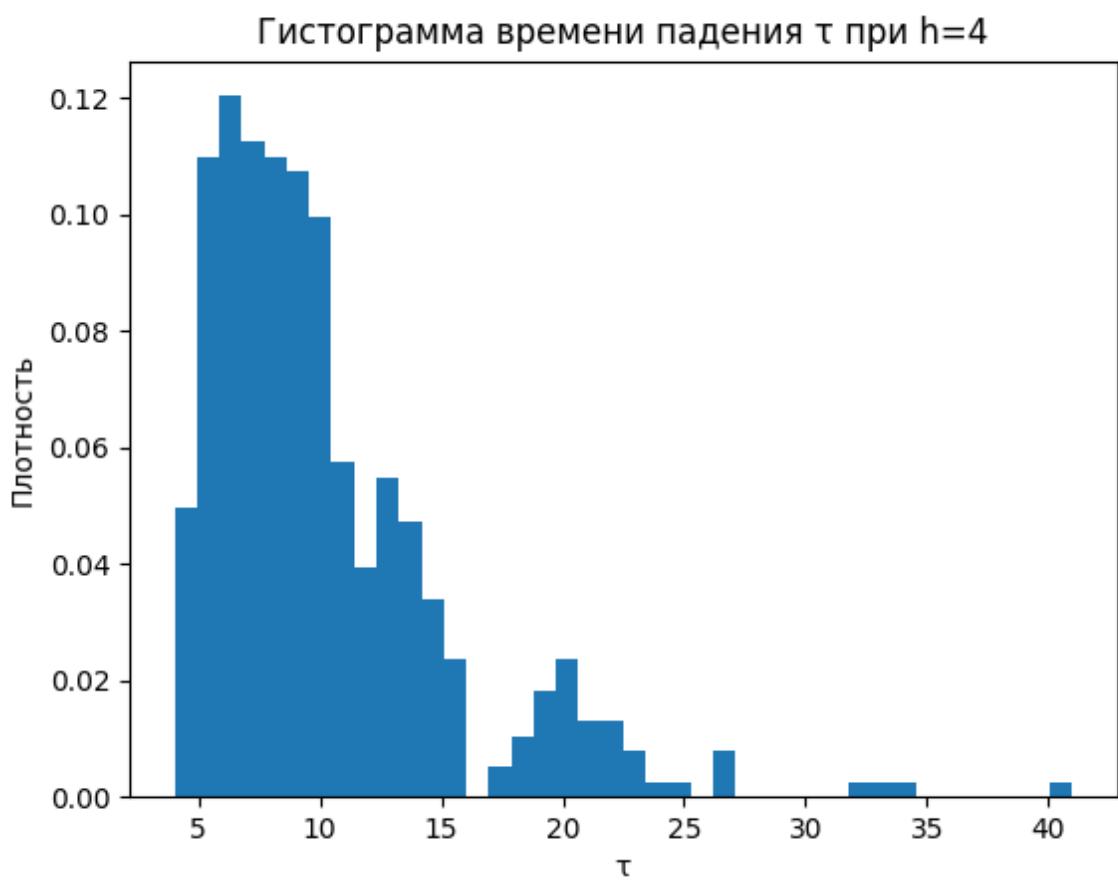
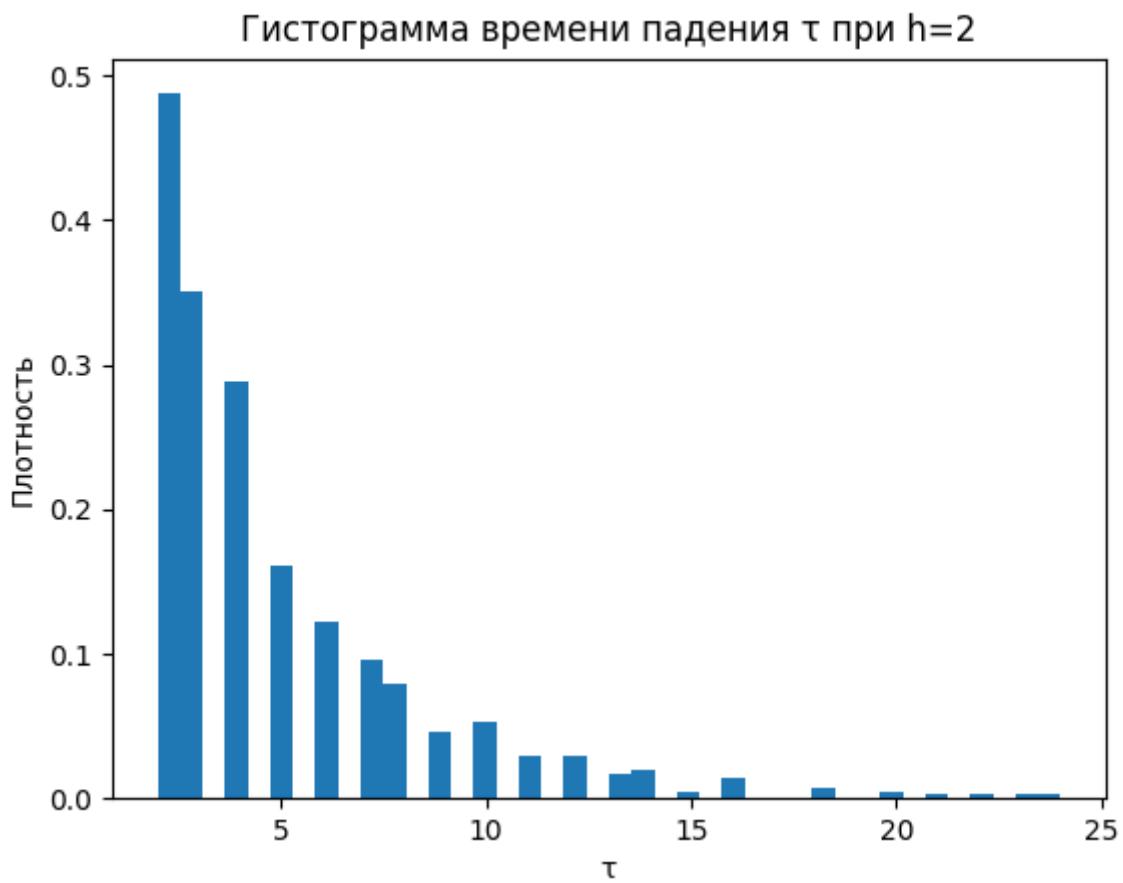
Программа реализует следующие операции:

- **Моделирование случайного блуждания дождевой капли:** каждый шаг выбирается случайно с вероятностями p_down , p_up , p_left , p_right в функции `one_walk(h)`; подсчитывается число шагов до касания земли `steps` += 1.
- **Планирование объема эксперимента для оценки среднего и дисперсии:** по пробной выборке рассчитываются среднее, стандартное отклонение и доверительный интервал (`conf_interval_mean`, `conf_interval_var`), далее вычисляется необходимое количество опытов для заданной относительной точности и доверительной вероятности.
- **Формирование статистического распределения времени падения:** массив `data_mean` или `data_var` собирается из результатов многократных симуляций; затем строится **гистограмма**: `plt.hist(data_mean, bins=40, density=True)`.

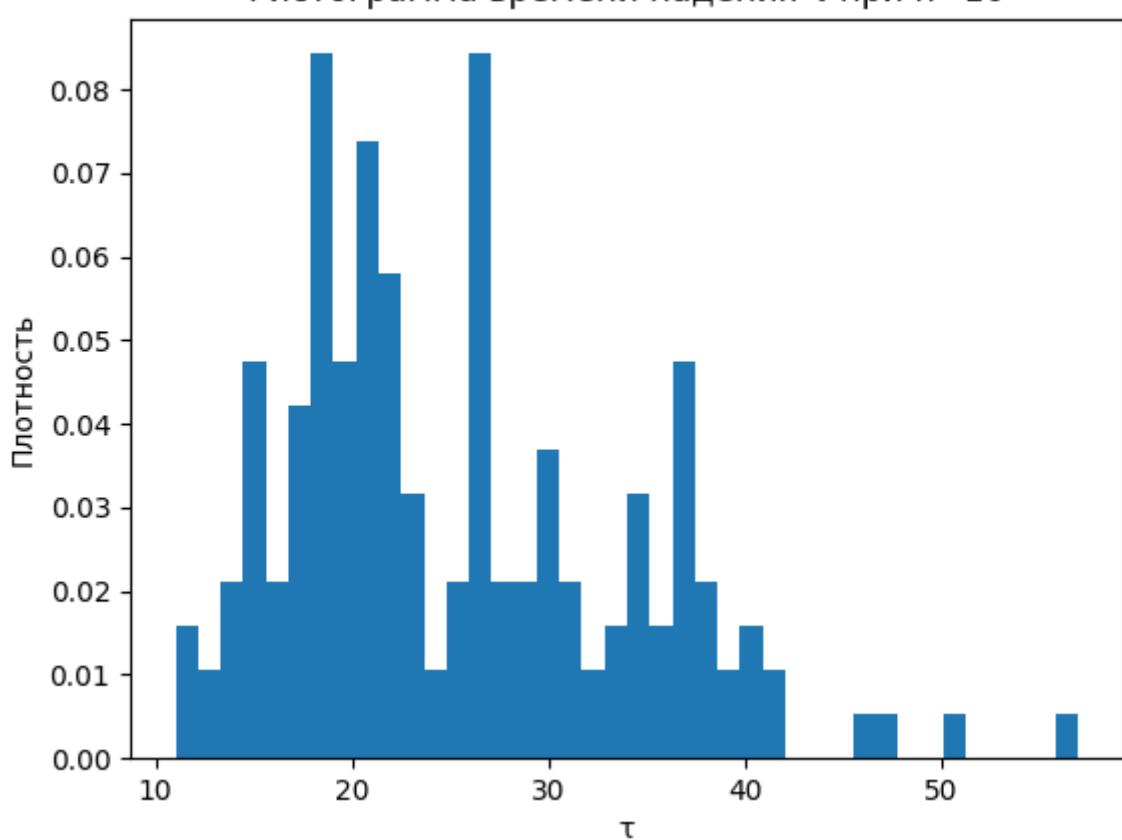
– **Наложение теоретических плотностей и эмпирической функции распределения:** для проверки аппроксимации нормальным законом можно строить нормальную плотность и эмпирическую ФР:
norm_pdf = $(1/(std*sqrt(2*pi))) * exp(-0.5*((xs-mean)/std)^2)$
plt.step(s, y, where='post') — эмпирическая ФР.

полный код программы приведен в приложении 1

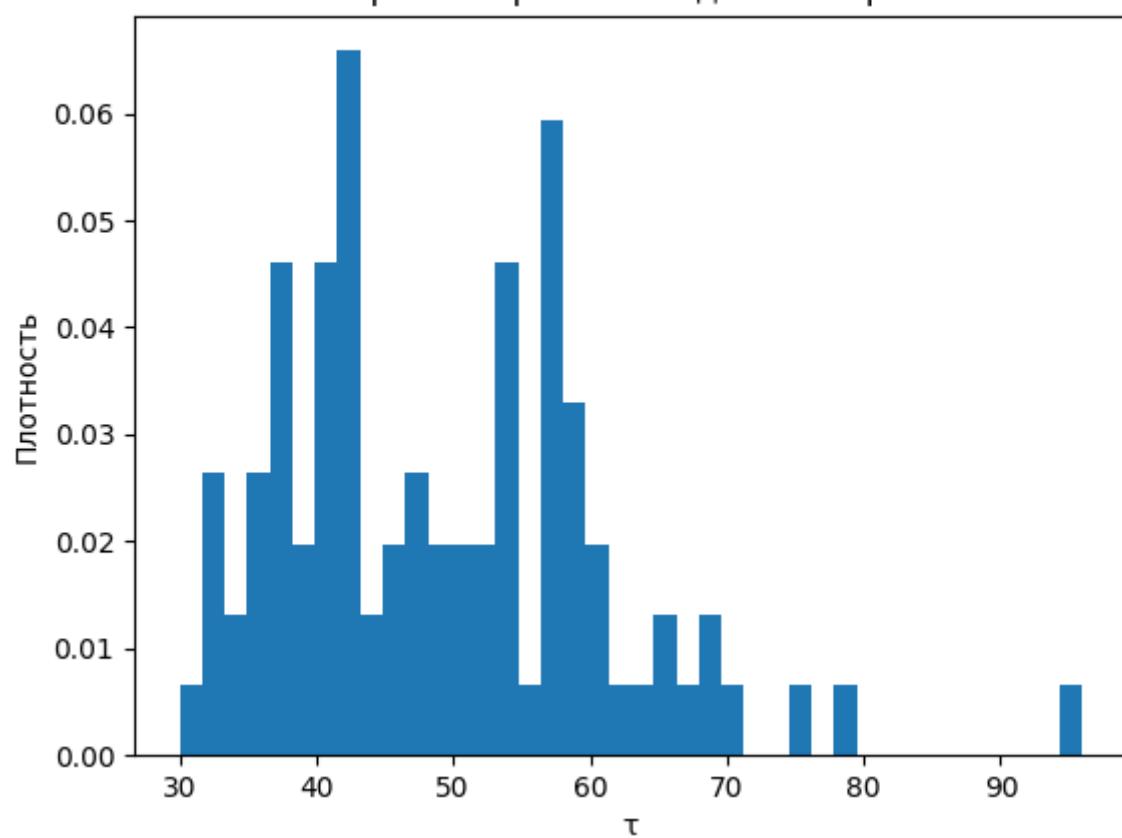
Результат работы программы:



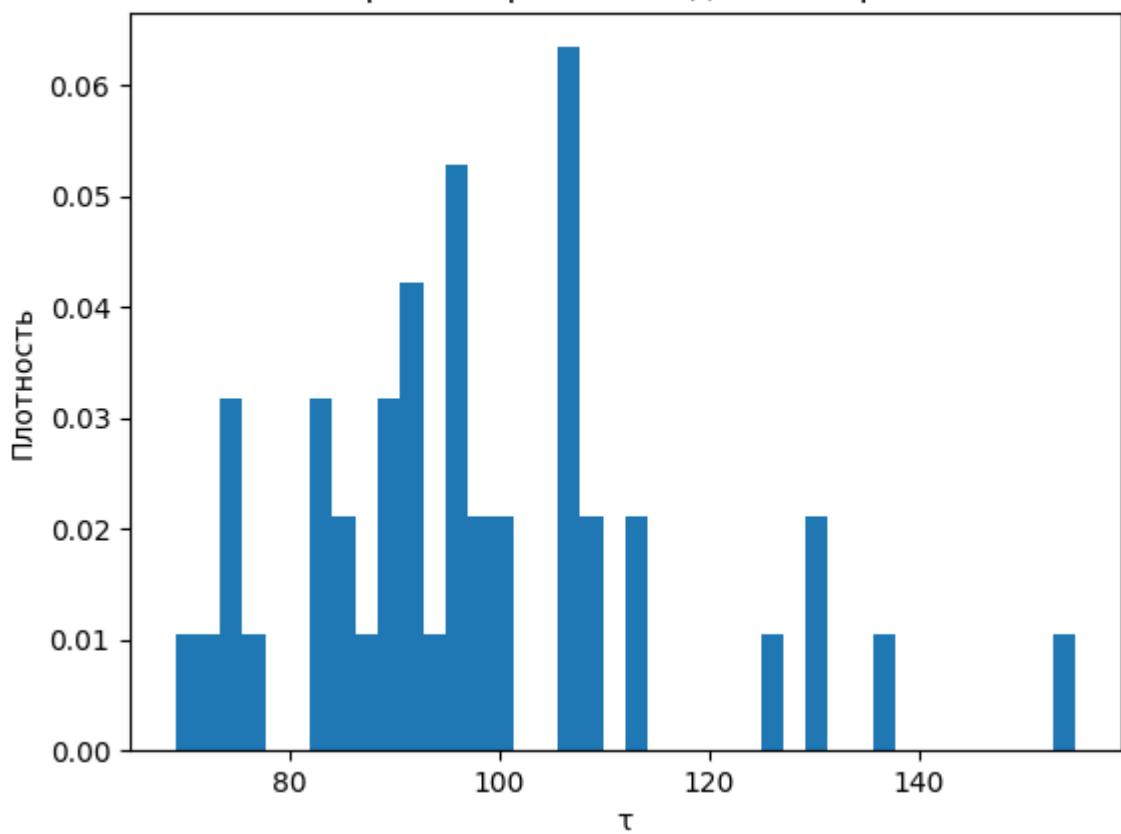
Гистограмма времени падения τ при $h=10$



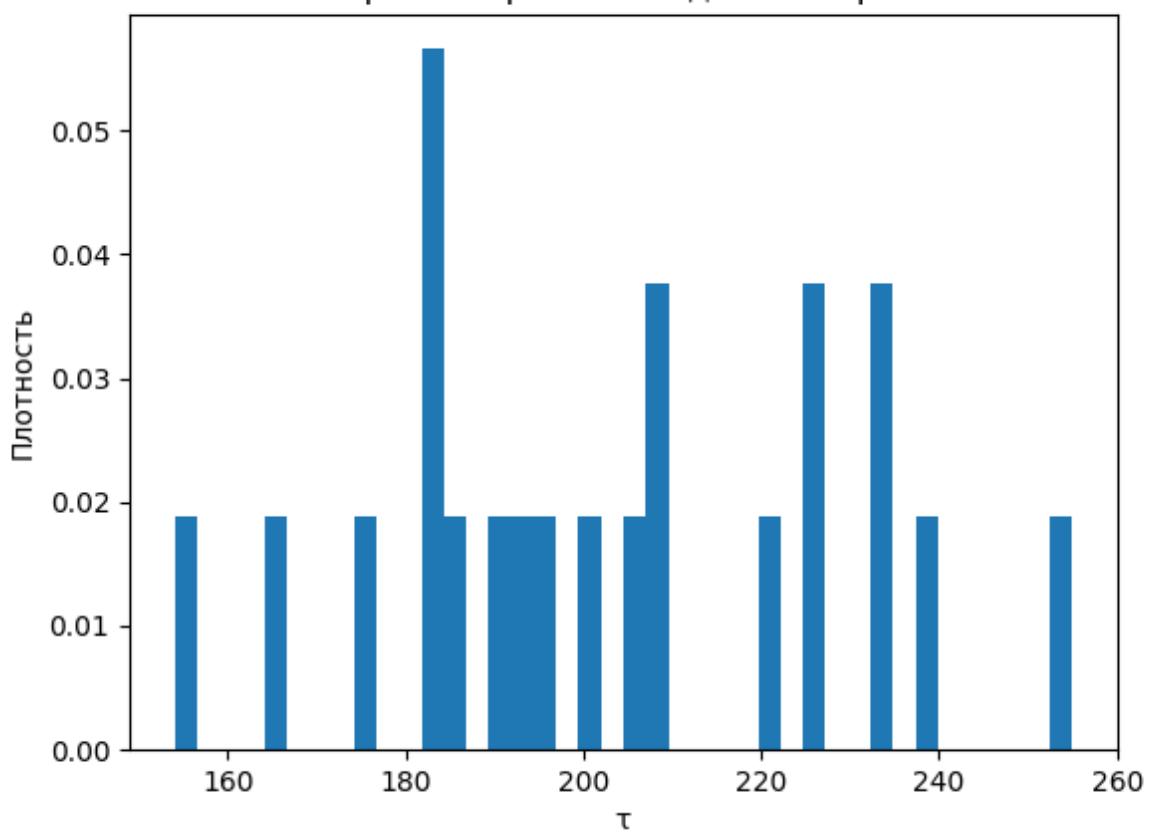
Гистограмма времени падения τ при $h=20$



Гистограмма времени падения τ при $h=40$



Гистограмма времени падения τ при $h=80$



```
===== RESTART: D:\мусорка\учеба\моделирования\lab8-Fokin-242.py =====
h = 2
Планируемое число опытов для оценки среднего: 757
Среднее τ = 4.87, доверительный интервал: [4.63, 5.12]
Планируемое число опытов для оценки дисперсии: 769
Дисперсия τ = 12.50

h = 4
Планируемое число опытов для оценки среднего: 413
Среднее τ = 10.19, доверительный интервал: [9.67, 10.70]
Планируемое число опытов для оценки дисперсии: 769
Дисперсия τ = 29.37

h = 10
Планируемое число опытов для оценки среднего: 165
Среднее τ = 25.35, доверительный интервал: [24.03, 26.66]
Планируемое число опытов для оценки дисперсии: 769
Дисперсия τ = 69.20

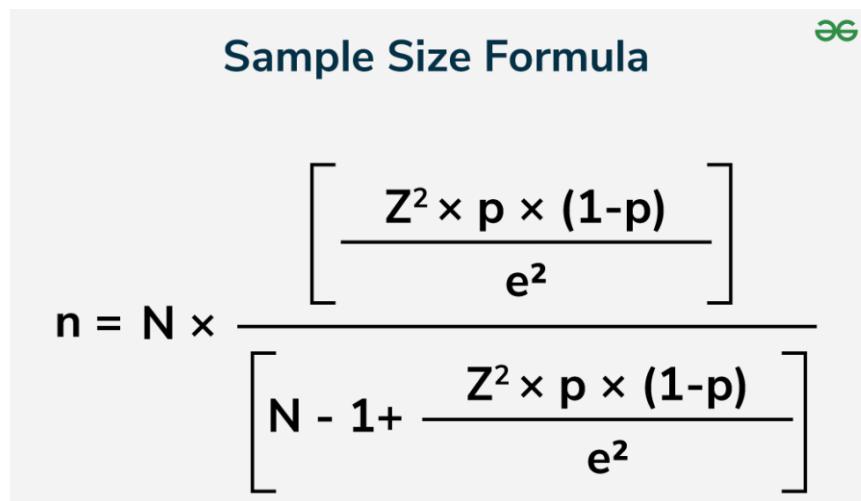
h = 20
Планируемое число опытов для оценки среднего: 92
Среднее τ = 49.28, доверительный интервал: [46.85, 51.71]
Планируемое число опытов для оценки дисперсии: 769
Дисперсия τ = 130.07

h = 40
Планируемое число опытов для оценки среднего: 44
Среднее τ = 98.05, доверительный интервал: [92.75, 103.35]
Планируемое число опытов для оценки дисперсии: 769
Дисперсия τ = 261.09

h = 80
Планируемое число опытов для оценки среднего: 21
Среднее τ = 203.05, доверительный интервал: [191.88, 214.22]
Планируемое число опытов для оценки дисперсии: 769
Дисперсия τ = 577.87
```

Ответ на вопрос

7. Каким образом можно определить объем выборки при оценивании вероятности наступления события с заданной точностью?



The slide has a green header "Sample Size Formula" and a green logo in the top right corner. The formula is displayed in a large, clear font:

$$n = N \times \frac{\left[\frac{Z^2 \times p \times (1-p)}{e^2} \right]}{\left[N - 1 + \frac{Z^2 \times p \times (1-p)}{e^2} \right]}$$

Для выборки меньшего размера... формула для оценки среднего значения генеральной совокупности с определенным уровнем достоверности:

$$n = [(Z \cdot \sigma) / E]^2$$

- n — объём выборки, который нужно определить;
- Z — квантиль нормального распределения, соответствующий доверительной вероятности;
- σ — стандартное отклонение (или для вероятности события $\sqrt{p(1 - p)}$);
- E — допустимая погрешность.

То есть, сначала задаётся доверительная вероятность и допустимая ошибка, затем с помощью формулы вычисляется необходимое количество испытаний n .

Источник:

<https://www.geeksforgeeks.org/mathssample-size-formula/>

Приложение 1 - код программы

```
import numpy as np
import math
import matplotlib.pyplot as plt

p_down = 0.5
p_up = 0.1
p_left = 0.2
p_right = 0.2
N = 10000
beta = 0.95
z = 1.96
h_values = [2, 4, 10, 20, 40, 80]
np.random.seed(0)

def one_walk(h):
    x = 0
    y = h
    steps = 0
    while y > 0:
        r = np.random.rand()
        if r < p_down:
            y -= 1
        elif r < p_down + p_up:
            y += 1
        elif r < p_down + p_up + p_left:
            x -= 1
        else:
            x += 1
        steps += 1
    return steps

def conf_interval_mean(data):
    mean = np.mean(data)
    std = np.std(data, ddof=1)
    d = z * std / math.sqrt(len(data))
    return mean, std, mean - d, mean + d

def conf_interval_var(data):
    var = np.var(data, ddof=1)
    n = len(data)
    chi2_low = (n - 1) * var / (0.5 * (1 + beta))
    chi2_high = (n - 1) * var / (0.5 * (1 - beta))
```

```

    return var, chi2_low, chi2_high

def plan_experiment_mean(h, eps_rel):
    sample = np.array([one_walk(h) for _ in range(1000)])
    mean, std, _, _ = conf_interval_mean(sample)
    n = math.ceil((z * std / (eps_rel * mean)) ** 2)
    return n

def plan_experiment_var(h, eps_rel):
    sample = np.array([one_walk(h) for _ in range(1000)])
    var = np.var(sample, ddof=1)
    n = math.ceil(2 * (z / eps_rel) ** 2)
    return n

for h in h_values:
    n_mean = plan_experiment_mean(h, 0.05)
    n_var = plan_experiment_var(h, 0.1)
    data_mean = np.array([one_walk(h) for _ in range(n_mean)])
    data_var = np.array([one_walk(h) for _ in range(n_var)])
    mean, std, low_m, high_m = conf_interval_mean(data_mean)
    var, low_v, high_v = conf_interval_var(data_var)
    print(f"h = {h}")
    print(f"Планируемое число опытов для оценки среднего: {n_mean}")
    print(f"Среднее τ = {mean:.2f}, доверительный интервал: [{low_m:.2f}, {high_m:.2f}]")
    print(f"Планируемое число опытов для оценки дисперсии: {n_var}")
    print(f"Дисперсия τ = {var:.2f}")
    print()
    plt.hist(data_mean, bins=40, density=True)
    plt.title(f"Гистограмма времени падения τ при h={h}")
    plt.xlabel("τ")
    plt.ylabel("Плотность")
    plt.show()

```