

Теория вероятностей и математическая статистика

Задание 1.

Вероятность того, что стрелок попадет в мишень, выстрелив один раз, равна 0.8. Стрелок выстрелил 100 раз. Найдите вероятность того, что стрелок попадет в цель ровно 85 раз.

Подсказка № 1

Определите тип распределения. Поскольку задача связана с вероятностью успеха (попадания) при фиксированном числе независимых испытаний (выстрелов), используйте биномиальное распределение. Биномиальное распределение описывает вероятность получения точно k успехов в n независимых испытаниях при вероятности успеха p в каждом испытании.

Подсказка № 2

Используйте функцию для биномиального распределения. В Python для расчета вероятности по биномиальному распределению можно использовать функцию `pmf` из библиотеки `scipy.stats`. Эта функция рассчитывает вероятность получения точного числа успехов k при заданных параметрах n и p .

Подсказка № 3

Проверьте параметры задачи. Перед выполнением расчетов убедитесь, что вы правильно задали параметры: $n=100$ — количество выстрелов, $p=0.8$ — вероятность попадания, и $k=85$ — количество попаданий, которое необходимо рассчитать.

Подсказка № 4

Рассчитайте вероятность для ровно 85 попаданий. Используйте функцию `binom.pmf(k, n, p)`, чтобы рассчитать вероятность того, что стрелок попадет ровно 85 раз из 100. Эта функция автоматически учитывает все необходимые вычисления для биномиального распределения.

Эталонное решение:

```
from scipy.stats import binom

# Данные задачи
n = 100 # количество выстрелов
```

```
p = 0.8 # вероятность попадания
k = 85  # количество попаданий

# Вероятность того, что стрелок попадет ровно 85 раз
probability_85_hits = binom.pmf(k, n, p)

print(f"Вероятность того, что стрелок попадет ровно 85 раз:
{probability_85_hits:.5f}")
```

Задача 2.

Вероятность того, что лампочка перегорит в течение первого дня эксплуатации, равна 0.0004. В жилом комплексе после ремонта в один день включили 5000 новых лампочек. Какова вероятность, что ни одна из них не перегорит в первый день? Какова вероятность, что перегорят ровно две?

Подсказка № 1

Определите тип распределения. Поскольку вероятность перегорания лампочки мала, а количество лампочек большое, используйте распределение Пуассона. Это распределение хорошо подходит для расчета вероятностей редких событий на большом количестве испытаний.

Подсказка № 2

Рассчитайте параметр λ . В распределении Пуассона параметр λ (среднее число событий) равен произведению количества испытаний n и вероятности события p . В данном случае $\lambda = n \times p$, где $n=5000$ и $p=0.0004$.

Подсказка № 3

Используйте функцию для распределения Пуассона. В Python для расчета вероятностей по распределению Пуассона используйте функцию `pmf` из библиотеки `scipy.stats`. Эта функция рассчитывает вероятность получения точного числа событий (например, перегорания лампочек) при заданном параметре λ .

Подсказка № 4

Рассчитайте вероятность для 0 и 2 перегораний. Для расчета вероятности того, что ни одна лампочка не перегорит, используйте `poisson.pmf(0, λ)`. Для расчета вероятности того, что перегорят ровно две лампочки, используйте `poisson.pmf(2, λ)`.

Эталонное решение:

```
from scipy.stats import poisson

# Данные задачи

n = 5000 # количество лампочек

p = 0.0004 # вероятность перегорания

λ = n * p # параметр λ (среднее число событий)

# Вероятность, что ни одна лампочка не перегорит

probability_no_failures = poisson.pmf(0, λ)

print(f"Вероятность того, что ни одна лампочка не перегорит:
{probability_no_failures:.5f}")

# Вероятность, что перегорят ровно две лампочки

probability_two_failures = poisson.pmf(2, λ)

print(f"Вероятность того, что перегорят ровно две лампочки:
{probability_two_failures:.5f}")
```

Задача 3.

Монету подбросили 144 раза. Какова вероятность, что орел выпадет ровно 70 раз?

Подсказка № 1

Определите тип распределения. Когда вы имеете дело с подсчетом числа успехов (выпадение орла) в фиксированном количестве независимых испытаний (подбрасывание монеты), используйте биномиальное распределение. Оно описывает вероятность получения определенного числа успехов при известной вероятности успеха в каждом испытании.

Подсказка № 2

Задайте параметры распределения. В данном случае число испытаний $n=144$, вероятность успеха (выпадения орла) $p=0.5$ и вероятность того, что орел выпадет ровно $k=70$. Эти параметры необходимо передать в функцию биномиального распределения.

Подсказка № 3

Используйте функцию `binom.pmf`. В Python для вычисления вероятности по биномиальному распределению используйте функцию `pmf` из библиотеки `scipy.stats`. Эта функция принимает три аргумента: k — число успехов, n — количество испытаний и p — вероятность успеха, и возвращает вероятность получения ровно k успехов.

Подсказка № 4

Рассчитайте вероятность. Для того чтобы найти вероятность того, что орел выпадет ровно 70 раз, используйте следующую запись: `binom.pmf(k, n, p)`. Это обеспечит правильный расчет вероятности для данного количества подбрасываний и выпадений орла.

Эталонное решение:

```
# Данные задачи

n = 144 # количество подбрасываний

p = 0.5 # вероятность выпадения орла

k = 70  # количество выпадений орла

# Вероятность того, что орел выпадет ровно 70 раз

probability_70_heads = binom.pmf(k, n, p)

print(f"Вероятность того, что орел выпадет ровно 70 раз:
{probability_70_heads:.5f}")
```

Задача 4.

В первом ящике находится 10 мячей, из которых 7 - белые. Во втором ящике - 11 мячей, из которых 9 белых. Из каждого ящика вытаскивают случайным образом по два мяча. Какова вероятность того, что все мячи белые? Какова вероятность того, что ровно два мяча белые? Какова вероятность того, что хотя бы один мяч белый?

Подсказка № 1

Определите ключевые события. В задаче рассматриваются три ситуации: все мячи белые, ровно два мяча белые, хотя бы один мяч белый. Начните с анализа того, какие комбинации мячей могут привести к каждой из этих ситуаций. Это поможет вам правильно составить формулы для расчетов.

Подсказка № 2

Используйте комбинации для расчета вероятностей. В Python для вычисления числа комбинаций используйте функцию `comb` из модуля `math`. Комбинации помогают вычислить, сколько различных пар мячей можно выбрать из определенного количества белых и не белых мячей.

Подсказка № 3

Рассчитайте вероятность извлечения двух белых мячей из каждого ящика. Для каждого ящика сначала вычислите вероятность того, что оба мяча будут белыми. Это делается путем деления числа комбинаций белых мячей на общее число комбинаций всех возможных мячей в ящике.

Подсказка № 4

Вычислите вероятность того, что все мячи белые. Для этого умножьте вероятность извлечения двух белых мячей из первого ящика на вероятность извлечения двух белых мячей из второго ящика. Это даст вам общую вероятность того, что все четыре мяча белые.

Подсказка № 5

Рассчитайте вероятность того, что ровно два мяча белые. Здесь вам нужно рассмотреть два сценария: в первом ящике два белых, во втором — нет, и наоборот. Найдите вероятность каждого сценария, затем сложите их.

Подсказка № 6

Используйте правило дополнения. Для вычисления вероятности того, что хотя бы один мяч будет белым, сначала найдите вероятность того, что ни один мяч не будет белым (оба мяча в каждом ящике не белые). Затем вычтите эту вероятность из единицы.

Эталонное решение:

```
from math import comb

# Данные задачи
total_balls_box1 = 10
```

```
white_balls_box1 = 7

total_balls_box2 = 11

white_balls_box2 = 9

# Вероятности извлечения двух белых мячей из первого ящика

prob_two_white_box1 = comb(white_balls_box1, 2) /
comb(total_balls_box1, 2)

# Вероятности извлечения двух белых мячей из второго ящика

prob_two_white_box2 = comb(white_balls_box2, 2) /
comb(total_balls_box2, 2)

# Вероятность, что все мячи белые

prob_all_white = prob_two_white_box1 * prob_two_white_box2

print(f"Вероятность того, что все мячи белые: {prob_all_white:.5f}")

# Вероятность того, что ровно два мяча белые

prob_one_white_box1 = (comb(white_balls_box1, 1) *
comb(total_balls_box1 - white_balls_box1, 1)) /
comb(total_balls_box1, 2)

prob_one_white_box2 = (comb(white_balls_box2, 1) *
comb(total_balls_box2 - white_balls_box2, 1)) /
comb(total_balls_box2, 2)

prob_two_white_exactly = (prob_two_white_box1 * prob_one_white_box2)
+ (prob_one_white_box1 * prob_two_white_box2)

print(f"Вероятность того, что ровно два мяча белые:
{prob_two_white_exactly:.5f}")
```

```
# Вероятность того, что хотя бы один мяч белый

prob_no_white_box1 = comb(total_balls_box1 - white_balls_box1, 2) /
comb(total_balls_box1, 2)

prob_no_white_box2 = comb(total_balls_box2 - white_balls_box2, 2) /
comb(total_balls_box2, 2)

prob_at_least_one_white = 1 - (prob_no_white_box1 *
prob_no_white_box2)

print(f"Вероятность того, что хотя бы один мяч белый:
{prob_at_least_one_white:.5f}")
```