

Теория вероятностей и математическая статистика

Задание 1.

Из колоды в 52 карты извлекаются случайным образом 4 карты.

а) Найти вероятность того, что все карты – крести.

б) Найти вероятность, что среди 4-х карт окажется хотя бы один туз

Подсказка № 1

Для начала важно понять, что из колоды в 52 карты вы извлекаете 4 карты. Важно найти общее количество возможных комбинаций для 4-х карт из 52. Для этого используйте формулу сочетаний $C(n,k)$, которая вычисляется как $\frac{n!}{k!(n-k)!}$. В Python эту функцию можно найти в модуле `math` как `math.comb(total_cards, 4)`.

Подсказка № 2

В колоде 13 крестей, и задача части (а) заключается в том, чтобы найти количество способов выбрать 4 карты, если все они должны быть крести. Для этого снова используйте формулу сочетаний, но на этот раз для крестей: `math.comb(clubs, 4)`.

Подсказка № 3

Для вычисления вероятности того, что все 4 карты будут крести, нужно разделить количество благоприятных исходов (все карты – крести) на общее количество возможных комбинаций. Формула: `probability_all_clubs = clubs_combinations / total_combinations`.

Подсказка № 4

Рассчитайте вероятность отсутствия тузов, чтобы найти вероятность того, что среди 4-х карт нет ни одного туза, сначала найдите количество комбинаций карт, среди которых нет тузов. Для этого из общего числа карт (52) нужно исключить тузов (4). Затем снова используйте формулу сочетаний для выбора 4-х карт из оставшихся 48: `no_aces_combinations = math.comb(total_cards - aces, 4)`.

Подсказка № 5

Найдите вероятность появления хотя бы одного туза. Если известна вероятность того, что среди 4-х карт нет ни одного туза, то вероятность появления хотя бы одного туза можно найти как дополнение: `probability_at_least_one_ace = 1 - probability_no_aces`. Это связано с тем, что сумма вероятностей противоположных событий всегда равна 1.

Эталонное решение:

```
import math
```

```

# Часть а) Вероятность, что все карты крести
total_cards = 52

clubs = 13

total_combinations = math.comb(total_cards, 4)

clubs_combinations = math.comb(clubs, 4)

probability_all_clubs = clubs_combinations / total_combinations

print(f"Вероятность того, что все карты крести:
{probability_all_clubs:.5f}")

# Часть б) Вероятность, что хотя бы один туз
aces = 4

no_aces_combinations = math.comb(total_cards - aces, 4)

probability_no_aces = no_aces_combinations / total_combinations

probability_at_least_one_ace = 1 - probability_no_aces

print(f"Вероятность того, что хотя бы один туз:
{probability_at_least_one_ace:.5f}")

```

Задача 2.

На входной двери подъезда установлен кодовый замок, содержащий десять кнопок с цифрами от 0 до 9. Код содержит три цифры, которые нужно нажать одновременно. Какова вероятность того, что человек, не знающий код, откроет дверь с первой попытки?

Подсказка № 1

Используйте формулу сочетаний. Поскольку кнопки нажимаются одновременно и порядок их нажатия не важен, для расчета количества возможных комбинаций используйте формулу сочетаний: $C(n, k) = \frac{n!}{k!(n-k)!}$, где n — это количество доступных кнопок, а k — количество кнопок, которые нужно нажать. В Python формула вычисляется с помощью функции `math.comb(buttons, code_digits)`.

Подсказка № 2

Рассчитайте общее количество возможных комбинаций. После того как вы определили количество доступных кнопок и количество цифр в коде, рассчитайте общее количество возможных комбинаций для ввода кода. Это количество показывает, сколько разных кодов можно составить из имеющихся кнопок. В Python результат будет выглядеть так: `total_combinations = math.comb(buttons, code_digits)`.

Подсказка № 3

Определите вероятность угадывания кода. Вероятность угадать код с первой попытки определяется как отношение одного благоприятного исхода (угаданный код) к общему числу возможных исходов (все возможные комбинации). Формула:

`probability_first_try = 1 / total_combinations`.

Эталонное решение:

```
import math

# Количество кнопок и количество цифр в коде
buttons = 10
code_digits = 3

total_combinations = math.comb(buttons, code_digits)

probability_first_try = 1 / total_combinations

print(f"Вероятность угадать код с первой попытки:
{probability_first_try:.5f}")
```

Задача 3.

В ящике имеется 15 деталей, из которых 9 окрашены. Рабочий случайным образом извлекает 3 детали. Какова вероятность того, что все извлеченные детали окрашены?

Подсказка № 1

Рассчитайте общее количество комбинаций, чтобы найти вероятность того, что все извлеченные детали будут окрашены, сначала определите общее количество возможных комбинаций для 3-х деталей, которые можно извлечь из 15. Для этого используйте формулу сочетаний: `math.comb(total_parts, 3)`.

Подсказка № 2

Рассчитайте количество благоприятных исходов. Найдите количество возможных комбинаций, когда все 3 извлеченные детали окрашены. Для этого используйте

формулу сочетаний, но на этот раз для окрашенных деталей:

```
math.comb(colored_parts, 3).
```

Подсказка № 3

Найдите вероятность того, что все извлеченные детали окрашены. Вероятность того, что все извлеченные детали будут окрашены, равна отношению количества благоприятных исходов (окрашенные детали) к общему числу возможных исходов.

Формула: $\text{probability_all_colored} = \frac{\text{colored_combinations}}{\text{total_combinations}}$.

Эталонное решение:

```
import math

# Всего деталей и количество окрашенных
total_parts = 15
colored_parts = 9

total_combinations = math.comb(total_parts, 3)
colored_combinations = math.comb(colored_parts, 3)

probability_all_colored = colored_combinations / total_combinations

print(f"Вероятность того, что все детали окрашены: {probability_all_colored:.5f}")
```

Задача 4.

В лотерее 100 билетов. Из них 2 выигрышных. Какова вероятность того, что 2 приобретенных билета окажутся выигрышными?

Подсказка № 1

Рассчитайте общее количество возможных комбинаций. Поскольку покупается 2 билета, нужно рассчитать общее количество возможных пар билетов, которые могут быть выбраны из 100. Это делается с помощью формулы сочетаний:

```
math.comb(total_tickets, 2).
```

Подсказка № 2

Рассчитайте количество благоприятных исходов. Поскольку вам нужно узнать вероятность того, что оба приобретенных билета будут выигрышными, нужно найти

количество способов выбрать 2 выигрышных билета из 2 доступных. Это также вычисляется с помощью формулы сочетаний: `math.comb(winning_tickets, 2)`.

Подсказка № 3

Вычислите вероятность того, что оба билета окажутся выигрышными. Вероятность того, что оба приобретенных билета будут выигрышными, равна отношению числа благоприятных исходов к общему числу возможных исходов. Формула:
`probability_two_winning = winning_combinations / total_combinations`.

Эталонное решение:

```
import math

# Количество билетов и выигрышных
total_tickets = 100
winning_tickets = 2

total_combinations = math.comb(total_tickets, 2)
winning_combinations = math.comb(winning_tickets, 2)

probability_two_winning = winning_combinations / total_combinations

print(f"Вероятность того, что оба билета выигрышные:
{probability_two_winning:.5f}")
```