Теория вероятностей и математическая статистика

Задание 1.

Даны значения зарплат из выборки выпускников: 100, 80, 75, 77, 89, 33, 45, 25, 65, 17, 30, 24, 57, 55, 70, 75, 65, 84, 90, 150. Посчитать (желательно без использования статистических методов наподобие std, var, mean) среднее арифметическое, среднее квадратичное отклонение, смещенную и несмещенную оценки дисперсий для данной выборки.

Подсказка № 1

Среднее арифметическое (mean) вычисляется как сумма всех значений выборки, деленная на количество значений. Убедитесь, что вы правильно используете сумму и количество элементов. В Python для суммы используйте функцию `sum()`, а для количества элементов - функцию `len()`.

Подсказка № 2

Для расчета смещенной дисперсии вам нужно вычислить квадрат разности каждого значения выборки и среднего арифметического, затем сложить все эти квадраты и разделить на общее количество элементов. Убедитесь, что делите на `n` (общее количество элементов), а не на `n-1`.

Подсказка № 3

Среднее квадратичное отклонение (standard deviation) является квадратным корнем из смещенной дисперсии. В Python для нахождения квадратного корня используйте функцию `math.sqrt()`. Убедитесь, что вы применяете квадратный корень к смещенной дисперсии, чтобы получить стандартное отклонение.

Подсказка № 4

При выполнении вычислений убедитесь, что сначала производится возведение в степень (в данном случае - квадрат разности), затем сложение и деление. Порядок операций в Python соответствует математическому порядку, так что сначала выполняются возведение в степень, затем умножение и деление, и только потом сложение и вычитание.

Эталонное решение:

```
import math
```

```
data = [100, 80, 75, 77, 89, 33, 45, 25, 65, 17, 30, 24, 57, 55, 70,
75, 65, 84, 90, 150]

n = len(data)


# Среднее арифметическое

mean = sum(data) / n


# Смещенная дисперсия

variance_biased = sum((x - mean) ** 2 for x in data) / n


# Несмещенная дисперсия

variance_unbiased = sum((x - mean) ** 2 for x in data) / (n - 1)


# Среднее квадратичное отклонение

stddev = math.sqrt(variance_biased)


print(f"Среднее арифметическое: {mean:.2f}")

print(f"Среднее квадратичное отклонение: {stddev:.2f}")

print(f"Смещенная дисперсия: {variance_biased:.2f}")

print(f"Несмещенная дисперсия: {variance_unbiased:.2f}")
```

Задача 2.

В первом ящике находится 8 мячей, из которых 5 - белые. Во втором ящике - 12 мячей, из которых 5 белых. Из первого ящика вытаскивают случайным образом два мяча, из второго - 4. Какова вероятность того, что 3 мяча белые?

Подсказка № 1

Определите вероятность того, что из первого ящика будут извлечены два белых мяча. Для этого вычислите комбинации из белых мячей и общее количество комбинаций для двух мячей из первого ящика. Используйте функцию `comb()` для вычисления комбинаций в Python. Форма для вычисления вероятности: $\frac{comb(5,2)}{comb(8,2)}$

Подсказка № 2

Определите вероятность того, что из второго ящика будет извлечён один белый мяч и три небелых мяча. Для этого вычислите комбинации белых и небелых мячей, а также общее количество комбинаций для четырёх мячей из второго ящика. Используйте `comb()` для вычисления комбинаций:$\frac{comb(5,1)\times comb(7,3)}{comb(12,2)}$

Подсказка № 3

Вероятности событий из независимых ящиков перемножаются для нахождения общей вероятности. В этом случае, перемножьте вероятность извлечения двух белых мячей из первого ящика и вероятность извлечения одного белого мяча из второго ящика.

Подсказка № 4

Убедитесь, что вы правильно определяете комбинации для каждого случая. Например, для вычисления вероятности двух белых мячей из первого ящика, вам нужно учитывать выбор двух мячей из пяти белых из общего количества мячей.

Эталонное решение:

```
from math import comb


# Данные задачи

total_balls_box1 = 8

white_balls_box1 = 5

total_balls_box2 = 12

white_balls_box2 = 5


# Вероятности

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

```
prob_one_white_box2 = comb(white_balls_box2, 1) *
comb(total_balls_box2 - white_balls_box2, 3) /
comb(total_balls_box2, 4)


# Вероятность того, что 3 мяча белые

prob_three_white = prob_two_white_box1 * prob_one_white_box2

print(f"Вероятность того, что 3 мяча белые: {prob_three_white:.5f}")
```

Задача 3.

На соревновании по биатлону один из трех спортсменов стреляет и попадает в мишень. Вероятность попадания для первого спортсмена равна 0.9, для второго — 0.8, для третьего — 0.6. Найти вероятность того, что выстрел произведен:
а). первым спортсменом
б). вторым спортсменом
в). третьим спортсменом.

Подсказка № 1

В задаче нужно найти вероятность того, что выстрел был произведён каждым из трёх спортсменов, зная их вероятности попадания. Эта задача предполагает использование теоремы Байеса. Однако, в данном случае предполагается, что выстрел попал в цель и вероятность попадания для каждого спортсмена используется для вычисления вероятности, что выстрел был произведён именно этим спортсменом.

Подсказка № 2

Чтобы найти вероятность того, что выстрел был произведён каждым спортсменом, нужно сначала найти общее количество вероятностей попадания всех спортсменов. Это сумма вероятностей попадания каждого спортсмена: ptotal=p1+p2+p3.

Подсказка № 3

Для каждого спортсмена вероятность того, что выстрел был произведён этим спортсменом, вычисляется как вероятность попадания этого спортсмена, делённая на общую сумму вероятностей попадания всех спортсменов:

- Для первого спортсмена: prob1=p1/p1+p2+p3
- Для второго спортсмена: prob2=p2,p1+p2+p3
- Для третьего спортсмена: prob3=p3/p1+p2+p3

Подсказка № 4

После вычисления вероятностей, проверьте, что сумма всех вероятностей равна 1. Это проверит, что расчёты были выполнены корректно.

Эталонное решение:

```
# Вероятности попадания для спортсменов

p1 = 0.9

p2 = 0.8

p3 = 0.6


# Вероятности того, что выстрел произведен каждым спортсменом

prob1 = p1 / (p1 + p2 + p3)

prob2 = p2 / (p1 + p2 + p3)

prob3 = p3 / (p1 + p2 + p3)


print(f"Вероятность того, что выстрел произведен первым спортсменом: {prob1:.5f}")

print(f"Вероятность того, что выстрел произведен вторым спортсменом: {prob2:.5f}")

print(f"Вероятность того, что выстрел произведен третьим спортсменом: {prob3:.5f}")
```

Задача 4.

В университет на факультеты А и В поступило равное количество студентов, а на факультет С студентов поступило столько же, сколько на А и В вместе. Вероятность того, что студент факультета А сдаст первую сессию, равна 0.8. Для студента факультета В эта вероятность равна 0.7, а для студента факультета С - 0.9. Студент сдал первую сессию. Какова вероятность, что он учится:

а). на факультете А

б). на факультете В

в). на факультете С?

Подсказка № 1

В задаче указано, что факультеты A и B имеют равное количество студентов, а факультет C имеет в два раза больше студентов, чем факультеты A и B вместе. Обозначьте количество студентов на факультетах A и B как x. Тогда на факультете C будет 2(x+x)=4x2(x + x) = 4x студентов. Сложите количество студентов на всех факультетах, чтобы получить общее количество студентов.

Подсказка № 2

Используйте вероятность сдачи сессии для студентов каждого факультета и умножьте её на долю студентов этого факультета от общего числа студентов. Это даст вероятность того, что случайно выбранный студент сдаст сессию и учится на данном факультете.

- Для факультета A: prob_pass_A=prob_A×pA
- Для факультета B: prob_pass_B=prob_B×pB
- Для факультета C: prob_pass_C=prob_C×pC

Подсказка № 3

Сложите вероятности сдачи сессии для всех факультетов, чтобы получить общую вероятность, что случайно выбранный студент сдаст сессию.

Подсказка № 4

Используйте теорему Байеса для вычисления вероятностей того, что студент учится на конкретном факультете, при условии, что он сдал сессию:

- Для факультета A: prob_A_given_pass=prob_pass_A/total_prob
- Для факультета B: prob_B_given_pass=prob_pass_B/total_prob
- Для факультета C: prob_C_given_pass=prob_pass_C/total_prob

Эталонное решение:

```
# Вероятности сдачи сессии

pA = 0.8

pB = 0.7

pC = 0.9



# Пропорции студентов

total_students = 2 + 2 + 4  # факультет C имеет в два раза больше
студентов, чем A и B вместе

prob_A = 2 / total_students
```

```
prob_B = 2 / total_students

prob_C = 4 / total_students


# Вероятности сдачи сессии для факультетов

prob_pass_A = prob_A * pA

prob_pass_B = prob_B * pB

prob_pass_C = prob_C * pC


# Общая вероятность сдачи сессии

total_prob = prob_pass_A + prob_pass_B + prob_pass_C


# Вероятности для каждого факультета

prob_A_given_pass = prob_pass_A / total_prob

prob_B_given_pass = prob_pass_B / total_prob

prob_C_given_pass = prob_pass_C / total_prob


print(f"Вероятность, что студент учится на факультете A:
{prob_A_given_pass:.5f}")

print(f"Вероятность, что студент учится на факультете B:
{prob_B_given_pass:.5f}")

print(f"Вероятность, что студент учится на факультете C:
{prob_C_given_pass:.5f}")
```

Задача 5.

Устройство состоит из трех деталей. Для первой детали вероятность выйти из строя в первый месяц равна 0.1, для второй - 0.2, для третьей - 0.25. Какова вероятность того, что в первый месяц выйдут из строя:
а). все детали
б). только две детали

в). хотя бы одна деталь
г). от одной до двух деталей?

Подсказка № 1

Чтобы найти вероятность того, что все детали выйдут из строя, перемножьте вероятности выхода каждой детали из строя. Убедитесь, что используете правильный порядок операций в Python, чтобы получить точный результат:

- Формула: `prob_all_fail = p1 * p2 * p3`

Подсказка № 2

Вычислите вероятность того, что ровно две детали выйдут из строя. Для этого вам нужно рассчитать вероятность того, что выходят из строя любые две детали, а третья деталь работает. Используйте следующую формулу:

- Формула: `prob_two_fail = (p1 * p2 * (1 - p3) + p1 * (1 - p2) * p3 + (1 - p1) * p2 * p3)`

Подсказка № 3

Расчитайте вероятность того, что хотя бы одна деталь выйдет из строя. Это можно найти, вычитая вероятность того, что все детали будут работать из 1. Используйте формулу для вероятности того, что ни одна деталь не выйдет из строя:

- Формула: `prob_at_least_one_fail = 1 - ((1 - p1) * (1 - p2) * (1 - p3))`

Подсказка № 4

Определите вероятность того, что от одной до двух деталей выйдут из строя. Для этого суммируйте вероятность того, что ровно одна деталь выйдет из строя и вероятность того, что ровно две детали выйдут из строя:

- Формула: `prob_one_or_two_fail = prob_two_fail + (p1 * (1 - p2) * (1 - p3) + (1 - p1) * p2 * (1 - p3) + (1 - p1) * (1 - p2) * p3)`

Эталонное решение:

```
from scipy.stats import binom
```

```
# Вероятности выхода из строя деталей
p1 = 0.1
p2 = 0.2
p3 = 0.25

# Вероятность того, что все детали выйдут из строя
prob_all_fail = p1 * p2 * p3

# Вероятность того, что ровно две детали выйдут из строя
prob_two_fail = (p1 * p2 * (1 - p3) + p1 * (1 - p2) * p3 + (1 - p1) * p2 * p3)

# Вероятность того, что хотя бы одна деталь выйдет из строя
prob_at_least_one_fail = 1 - ((1 - p1) * (1 - p2) * (1 - p3))

# Вероятность того, что от одной до двух деталей выйдут из строя
prob_one_or_two_fail = prob_two_fail + (p1 * (1 - p2) * (1 - p3) + (1 - p1) * p2 * (1 - p3) + (1 - p1) * (1 - p2) * p3)

print(f"Вероятность того, что все детали выйдут из строя: {prob_all_fail:.5f}")
print(f"Вероятность того, что только две детали выйдут из строя: {prob_two_fail:.5f}")
print(f"Вероятность того, что хотя бы одна деталь выйдет из строя: {prob_at_least_one_fail:.5f}")
print(f"Вероятность того, что от одной до двух деталей выйдут из строя: {prob_one_or_two_fail:.5f}")
```