

# Math 318 Homework 6

Xander Naumenko

10/03/23

**Question 1a.** Since  $X$  is the sum of normal variables, we have that  $\mu_X = 200$  and  $(9\sigma_X)^2 = 9\mu^2 \implies \sigma_X = \frac{1}{3}\sigma$  and  $X$  is normal. Let  $Y = \frac{X-\mu_X}{\sigma_X} = 3\frac{X-200}{\sigma} = N(0, 1)$ . Then we have that:

$$P(|X - 200| > 5|\sigma = 5) = 2P(N(0, 1) < -3\frac{5}{5}) = 2\Phi(-3) = 0.0027.$$

$$P(|X - 200| > 5|\sigma = 5) = 2P(N(0, 1) < -3\frac{10}{5}) = 2\Phi(-1.5) = 0.1336.$$

$$P(|X - 200| > 5|\sigma = 5) = 2P(N(0, 1) < -3\frac{15}{5}) = 2\Phi(-1) = 0.3173.$$

**Question 1b.** From the probabilities above, case  $i$  would be rejected at 5% level of confidence since it is the only one with  $p < 0.05$ .

**Question 1c.** Again from the probabilities above, only case  $i$  would be rejected at 1% level of confidence since it is the only one with  $p < 0.01$ .

**Question 2a.** This is equivalent to a series of Bernoulli trials, each with mean  $\frac{1}{5}$  and variance  $pq = \frac{4}{25} \implies \sigma = \frac{2}{5}$ . Then the sum  $S$  approximates a normal distribution with mean  $\frac{n}{5}$  and variance  $\frac{4n}{25}$ . As such we have that

$$\frac{S - \mu n}{\sqrt{n}\sigma} = \frac{S - \frac{n}{5}}{\frac{2}{5}\sqrt{n}} \approx N(0, 1).$$

Using tables we know that  $\Phi(-3.09) = 0.001$  which is what we want to achieve, so to find the required  $n$  we must solve the equation:

$$3.09 = \frac{\frac{n}{3} - \frac{n}{5}}{\frac{2}{5}\sqrt{n}} \implies n = 85.93.$$

Thus we would need at least 86 cards to be sufficiently certain.

**Question 2b.** If the claim is true, then we instead have approximately a normal distribution centered about  $\frac{n}{2}$  with variance  $\frac{n}{4}$ . By the central limit theorem we have

$$\begin{aligned} \frac{S - \mu n}{\sqrt{n}\sigma} &= \frac{S - \frac{n}{2}}{\sqrt{n}\frac{1}{2}} = N(0, 1) \\ \implies P\left(S < \frac{n}{3}\right) &= P\left(\frac{1}{4}\sqrt{n}N(0, 1) + \frac{n}{2} < \frac{n}{3}\right) = P\left(N(0, 1) < -\frac{2}{3}\sqrt{n}\right) \\ &= P\left(N(0, 1) < -\frac{2}{3}\sqrt{86}\right) = 3.156 \times 10^{-10}. \end{aligned}$$

**Question 3a.** The following python was used to calculate the standard deviation/mean:

---

```

1 import numpy as np
2
3 bread = [ ... ]
4
5 print(np.mean(bread))
6 print(np.std(bread, ddof=1))

```

---

From these we know that the 95% confidence interval is within  $\frac{1.96\sigma^2}{\sqrt{n}}$  from the mean, so the confidence interval

$$[0.99838, 0.99962].$$

**Question 3b.** Yes we can since it's outside the 95% interval, although the upper bound is quite close to 1kg.

**Question 3c.** Given that we're using the 95% confidence interval to reject the null hypothesis, the chance of us incorrectly reaching the wrong decision is 5%.

**Question 3d.** Using the above logic, on average  $50 * 0.05 = 2.5$  of them would reject the bakery's claim.

**Question 4a.** Integrating over  $x$  and  $y$  respectively:

$$f_Y(y) = \int_0^2 \frac{xy+1}{3} dy = \frac{1}{6}4x + \frac{2}{3} = \frac{2(y+1)}{3}$$

$$f_X(x) = \int_0^1 \frac{xy+1}{3} dy = \frac{1}{6}x + \frac{1}{3} = \frac{x+2}{6}$$

$$f_{X|Y}(x|y) = \frac{f_{XY}(x,y)}{f_Y(y)} = \frac{xy+1}{2(y+1)}$$

$$f_{Y|X}(y|x) = \frac{f_{XY}(x,y)}{f_X(x)} = \frac{2(xy+1)}{x+2}.$$

**Question 4b.**

$$E[X|Y] = \int_0^1 x \frac{xy+1}{2(y+1)} dx = \frac{4y+3}{3y+3}$$

$$E[Y|X] = \int_0^1 y \frac{2(xy+1)}{x+2} dy = \frac{2x+3}{3x+6}.$$

**Question 5a.** For a given  $x$  as the hint suggests the expected value of a single die role is  $\frac{x+1}{2}$ , so since the expected value of a sum is the sum of expected values we have that

$$E[Y|X] = x \frac{x+1}{2}.$$

**Question 5b.** Each of the possible values for  $X$  are equally likely, so we can just sum over their conditional probability times  $\frac{1}{4}$ :

$$E[Y] = \frac{1}{4} \left( \frac{4^2+4}{2} + \frac{6^2+6}{2} + \frac{8^2+8}{2} + \frac{12^2+12}{2} \right) = 36.25.$$

**Question 6a.** The mean is 0.3414 and the variance is  $2.469 \times 10^{-7}$ . The code used was:

---

```

1 import numpy as np
2
3 def f(x):
4     return 1/(2*np.pi)**0.5*np.exp(-x**2/2)
5
6 n = 10000
7 I = [sum([f(x) for x in np.random.uniform(size=n)]) / n for _ in range(100)]
8
9 print(np.mean(I))
10 print(np.std(I, ddof=1)**2)

```

---

**Question 6b.** The integral represents the area under the curve, and we're effectively calculating the fraction of the unit square that is taken up by the function's area. However the square's area is just 1, multiplying the number we calculate by 1 would give us the area under the curve.

**Question 6c.** The mean was 0.341 and the variance was  $2 \times 10^{-5}$ . The code used:

---

```

1 S = [sum([1 if y < f(x) else 0 for x, y in zip(np.random.uniform(size=n), np.random.
2         uniform(size=n))]) / n for _ in range(100)]
3
4 print(np.mean(S))
5 print(np.std(S, ddof=1)**2)

```

---

**Question 6d.** The first method seems to be better since the variance was lower, so it takes less iterations of the first method to converge on a solution. The variance of the second solutions was two orders of magnitude larger, so for a given number of iterations the solution will be less precise.