CS109 Lecture 16 Worksheet: ANSWER KEY

Bootstrapping

Summer 2026

Lecture 16 Worksheet: Answer Key

Instructor / TA copy. Problem-set solutions are hidden in the standard build; run `./convert-to-pdf.sh 16 solutions` for the full instructor/TA edition.

Problem 1 (Review): Checkout Totals

(a) S is a sum of 100 i.i.d. RVs with finite mean and variance, so by the CLT

$$S \approx N(\mu = 100 \cdot 3 = 300, \quad \sigma^2 = 100 \cdot 4 = 400), \qquad \sigma = 20.$$

(b) With continuity correction (integer-valued sum):

$$P(S \geq 320) \approx P(\text{Normal} \geq 319.5) = 1 - \Phi\left(\frac{319.5 - 300}{20}\right) = 1 - \Phi(0.975) \approx 0.165.$$

Problem 2: Ping Times

(a) $\bar{X} = \frac{12+15+11+14+13+19+12+16}{8} = \frac{112}{8} = 14$ ms.

(b) Deviations from 14: $-2, 1, -3, 0, -1, 5, -2, 2$; squares sum to 48. So

$$S^2 = \frac{48}{8-1} = \frac{48}{7} \approx 6.86 \text{ ms}^2.$$

We divide by $n-1$ because the sample mean is the value that *minimizes* the sum of squared deviations from the data, so using $\bar{X}$ in place of μ systematically underestimates the spread; dividing by $n-1$ exactly corrects the bias.

(c) $\text{SE} = \sqrt{S^2/n} = \sqrt{6.86/8} \approx 0.93$ ms. It estimates the standard deviation of the *sample mean itself* — how much $\bar{X}$ would bounce around if we redid the 8-ping experiment many times.

(d) $14 \pm 2(0.93) = [12.1,\ 15.9]$ ms (approximately). By the CLT, $\bar{X}$ is approximately normal, so an interval of ± 2 standard errors captures the true mean latency for roughly 95% of repeated experiments.

Problem 3: A Bootstrap You Can Enumerate

(a) Each of the 3 draws has 3 equally likely outcomes: $3^3 = 27$ ordered resamples.

(b) $P(\max = 9) = 1 - P(\text{no draw is } 9) = 1 - \left(\frac{2}{3}\right)^3 = 1 - \frac{8}{27} = \frac{19}{27} \approx 0.70$.

(c) The mean is 5 iff the resample sums to 15, which (with values from $\{2, 4, 9\}$) happens only for the multiset $\{2, 4, 9\}$, i.e., the $3! = 6$ orderings. $P = \frac{6}{27} = \frac{2}{9} \approx 0.22$.

(d) Sampling n from n without replacement always returns the entire original sample (in some order), so every resample has *exactly* the same statistic and the bootstrap distribution collapses to a single point — no variability, no error bars. (Also, after each removal the draw probabilities no longer match the sample PMF.)

Problem 4: Bootstrapping the Median

(a)

```
bootstrap_median_std(ratings):            # len(ratings) = 50
    medians = []
    repeat 10,000 times:
        resample = np.random.choice(ratings, 50, replace=True)
        medians.append(median(resample))
    return std(medians)
```

(b) The bootstrap assumption: the sample distribution (the histogram of your 50 ratings) is a good estimate of the underlying true distribution, so drawing from the sample with replacement mimics drawing fresh samples from the population.

(c) It fails when the samples are not i.i.d., or when the underlying distribution has a long tail that the sample is unlikely to have captured (rare, extreme values matter but never made it into your sample).

Problem 5: Is the New Compiler Flag Faster?

(a) **Null hypothesis:** the flag makes no difference — all 85 runtimes are drawn from one and the same distribution, and the observed 6 ms difference is due to sampling error alone.

(b)

```
p_value(A_times, B_times):                 # 40 and 45 measurements
    universal = concat(A_times, B_times)   # pool: the null's one distribution
    count = 0
    repeat 10,000 times:
        resample_A = np.random.choice(universal, 40, replace=True)
        resample_B = np.random.choice(universal, 45, replace=True)
        diff = mean(resample_A) - mean(resample_B)
        if abs(diff) >= 6: count += 1
    return count / 10,000
```

(c) $p = 0.03$: if the flag truly made no difference, only about 3% of experiments like ours would show a gap of 6 ms or more, so the data provide reasonably strong evidence that the flag matters. With $p = 0.4$, a 6 ms gap would be entirely unremarkable under the null, so we could not claim any real difference.

Problem 6: Estimating Course Size (Winter 2025 Final)

(a) Estimate $E[R]$ with the sample mean of the ratios, and scale:

$$E[T] = 300 \cdot \bar{R} = 300 \cdot \frac{1}{10} \sum_{i=1}^{10} r_i.$$

(b) Estimate $\mathrm{Var}(R)$ with the (unbiased) sample variance, and scale by 300^2 (constants come out of variance squared):

$$\mathrm{Var}(T) = 300^2 \cdot S_R^2 = 300^2 \cdot \frac{1}{9} \sum_{i=1}^{10} (r_i - \bar{R})^2.$$

(c) Bootstrap the statistic "$\mathrm{Var}(T)$ estimate":

```
vars = []
repeat 10,000 times:
    resample = np.random.choice([r_1..r_10], 10, replace=True)
    vars.append(300^2 * sample_variance(resample))
return std(vars)
```

The standard deviation of the 10,000 recomputed variance estimates is the bootstrap estimate of the sampling variability of $\widehat{\mathrm{Var}}(T)$.

Challenge Problem: Two Schools of Thought

(a) Beta is conjugate to the coin flips: 5 heads and 10 tails update $\mathrm{Beta}(2, 2)$ to

$$p \mid \text{data} \sim \mathrm{Beta}(2 + 5,\ 2 + 10) = \mathrm{Beta}(7, 12), \qquad E[p \mid \text{data}] = \frac{7}{19} \approx 0.37.$$

(b) Treat the 15 flips (5 ones, 10 zeros) as the sample. Repeat 10,000 times: resample 15 flips with replacement, record the fraction of heads in the resample. The histogram of the 10,000 fractions is the bootstrap distribution of $\hat{p}$ (centered at $\frac{5}{15} = \frac{1}{3}$).

(c) The Bayesian posterior expresses *belief about the parameter* (probability that p lies in a range, given prior + data); the bootstrap distribution expresses *sampling variability of the estimator* (how $\hat{p}$ would vary across repeated experiments) — no prior, and p is treated as a fixed unknown, not a random variable.