Group assignment 1

November 2025

Group Members

Baraa Magdy

Celia Medina Giménez

Dino Keylas

Maria Naz

Oskar Johannes Piibar

Problem 1.

Suppose that A and B are independent events, show that A^c and B^c are independent.

Solution

If A^c and B^c are independent then:

$$P(A^c \cap B^c) = P(A^c)P(B^c).$$

Now, we have (by De Morgan's law):

$$P(A^c \cap B^c) = P((A \cup B)^c).$$

By the complement rule:

$$P(A^c \cap B^c) = 1 - P(A \cup B).$$

By the inclusion–exclusion formula:

$$= 1 - \big(P(A) + P(B) - P(A \cap B)\big).$$

Since A and B are independent:

$$= 1 - \big(P(A) + P(B) - P(A)P(B)\big).$$

Now to factor:

$$= 1 - P(A) - P(B) + P(A)P(B).$$

Group the first two terms together:

$$= \big(1 - P(A)\big) - P(B) + P(A)P(B).$$

Now group the last two terms:

$$= \big(1 - P(A)\big) - \big(P(B) - P(A)P(B)\big).$$

Factor $P(B)$ from that difference:

$$= \big(1 - P(A)\big) - P(B)\big(1 - P(A)\big).$$

Now factor out the common term $\big(1 - P(A)\big)$:

$$= \big(1 - P(A)\big)\big(1 - P(B)\big).$$

Finally, as $1 - P(A) = P(A^c)$ and $1 - P(B) = P(B^c)$:

$$= P(A^c)P(B^c).$$

Therefore, A^c and B^c are independent if A and B are.

Problem 2.

The probability that a child has brown hair is 1/4. Assume independence between children and assume there are three childrens.

Problem 2A.

If it is known that at least one child has brown hair, what is the probability that at least two children have brown hair?

Solution

Let $p =$ "a child has brown hair" $= 1/4$.

We want:

$$P(X \geq 2 \mid X \geq 1) = \frac{P(X \geq 2)}{P(X \geq 1)} \qquad (*)$$

as $\{X \geq 2\} \subseteq \{X \geq 1\}$

where $X \sim \text{Binomial}(n = 3, p = 1/4)$.

$$P(X = 2) = \binom{3}{2} \left(\tfrac{1}{4}\right)^2 \left(\tfrac{3}{4}\right)^1 = \tfrac{9}{64},$$

$$P(X = 3) = \binom{3}{3} \left(\tfrac{1}{4}\right)^3 \left(\tfrac{3}{4}\right)^0 = \tfrac{1}{64}.$$

So,

$$P(X \geq 2) = \tfrac{9}{64} + \tfrac{1}{64} = \tfrac{10}{64} = \tfrac{5}{32}.$$

Also,

$$P(X = 0) = \binom{3}{0} \left(\tfrac{1}{4}\right)^0 \left(\tfrac{3}{4}\right)^3 = \tfrac{27}{64}.$$

$$P(X \geq 1) = 1 - P(X = 0) = 1 - \tfrac{27}{64} = \tfrac{37}{64}.$$

Now coming back to (*):

$$P(X \geq 2 \mid X \geq 1) = \frac{P(X \geq 2)}{P(X \geq 1)} = \frac{5/32}{37/64} = \frac{10}{37}.$$

Problem 2B.

If it known that the oldest child has brown hair, what is the probability that at least two children have brown hair?

Solution

We know that one child has brown hair, so we need to calculate the probability of at least one of the other 2 having brown hair.

$$\left(\tfrac{1}{4} \cdot \tfrac{3}{4}\right) + \left(\tfrac{3}{4} \cdot \tfrac{1}{4}\right) + \left(\tfrac{1}{4} \cdot \tfrac{1}{4}\right).$$

Here:

- First term = one of them has brown hair (case 1).
- Second term = one of them has brown hair (case 2).
- Third term = both have brown hair.

$$= \tfrac{3}{16} + \tfrac{3}{16} + \tfrac{1}{16} = \tfrac{7}{16}.$$

Problem 3.

Let (X, Y) be uniform on the unit disc, $\{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq 1\}$. Set $R = \sqrt{X^2 + Y^2}$. What is the CDF and PDF of R?

Solution

CDF: Given the uniform distribution of the unit disc, then selecting a point in there has a flat density over the circle.

$$F_R(r) = P(R \leq r)$$

Therefore the probability of being in a certain area equals to the area of the chosen point divided with the area of the unit disc.

$$F_R(r) = \frac{A(r)}{A(r=1)} = \frac{\pi r^2}{\pi 1^2} = r^2.$$

Thus

$$F_R(r) = \begin{cases} 0, & r < 0, \\ r^2, & 0 \leq r \leq 1, \\ 1, & r > 1. \end{cases}$$

PDF: From the calculated CDF the PDF can be easily derived:

$$f_R(r) = \frac{\mathrm{d}}{\mathrm{d}r} F_R(r)$$

Given that in the uniform distribution r is continuous for $0 \leq r \leq 1$.

$$f_R(r) = \frac{\mathrm{d}}{\mathrm{d}r} r^2 = \begin{cases} 2r, & 0 \leq r \leq 1, \\ 0, & \text{otherwise.} \end{cases}$$

Problem 4.

A fair coin is tossed until a head appears. Let X be the number of tosses required. Find $\mathbb{E}[X]$.

Solution.

The probability of getting first Head in the k-th trial, (getting Tail in the k-1 trials):

$$P(X = k) = P(T_1 \cap T_2 \cap \cdots \cap T_{k-1} \cap H_k)$$

Given independent Bernoulli trials,

$$P(X = k) = P(T_1)P(T_2)\cdots P(H_k) = (P(T))^{k-1}P(H)$$

Let $P(H) = p,\ P(T) = q.$ then

$$P(X = k) = q^{k-1}p$$

Given fair coin P(T) = P(H) = $\frac{1}{2}$

$$\mathbb{E}[X] = \sum_{k=1}^{\infty} k\left(q^{k-1}p\right) = \sum_{k=1}^{\infty} k\left(\tfrac{1}{2}\right)^k$$

From geometric series:

$$\sum_{n=0}^{\infty} r^n = \frac{1}{1-r}$$

Differentiate w.r.t r:

$$\sum_{n=0}^{\infty} nr^{n-1} = \frac{1}{(1-r)^2}$$

Multiply both sides by r:

$$\sum_{n=0}^{\infty} nr^n = \frac{r}{(1-r)^2}$$

From this:

$$\sum_{k=1}^{\infty} k\left(\tfrac{1}{2}\right)^k = \frac{\frac{1}{2}}{(1-\frac{1}{2})^2} = 2$$

$$\therefore\ \mathbb{E}[X] = 2$$

Problem 5.

Let $X1, ..., Xn$ be IID from Bernoulli (p). **(a)** Let $\alpha > 0$ be fixed and define

$$\varepsilon_n = \sqrt{\frac{1}{2n} \log \frac{2}{\alpha}}.$$

Let

$$\hat{p}_n = \frac{1}{n} \sum_{i=1}^{n} X_i$$

and define the confidence interval

$$I_n = \big[\hat{p}_n - \varepsilon_n,\ \hat{p}_n + \varepsilon_n\big].$$

Use Hoeffding's inequality to show that

$$P\big(p \in I_n\big) \geq 1 - \alpha.$$

Problem 5A.

Solution

Let $X_1, \ldots, X_n \overset{\text{iid}}{\sim} \text{Bernoulli}(p)$, and define the sample mean

$$\hat{p}_n = \frac{1}{n} \sum_{i=1}^{n} X_i.$$

We want to construct a confidence interval for p using Hoeffding's inequality.

Step 1: Hoeffding's inequality

For i.i.d. random variables $X_i \in [0, 1]$, Hoeffding's inequality states that for any $t > 0$,

$$P\big(\big|\hat{p}_n - \mathbb{E}[X_i]\big| \geq t\big) \leq 2e^{-2nt^2}.$$

Since $\mathbb{E}[X_i] = p$, this becomes

$$P\big(|\hat{p}_n - p| \geq t\big) \leq 2e^{-2nt^2}.$$

Step 2: Choice of t

Define

$$\varepsilon_n = \sqrt{\frac{1}{2n} \log \frac{2}{\alpha}}.$$

Plugging $t = \varepsilon_n$ into Hoeffding's bound gives

$$P\big(|\hat{p}_n - p| \geq \varepsilon_n\big) \leq 2 \exp\left(-2n \cdot \frac{1}{2n} \log \frac{2}{\alpha}\right).$$

Simplifying:

$$= 2 \exp\left(-\log \frac{2}{\alpha}\right) = 2 \cdot \frac{\alpha}{2} = \alpha.$$

Step 3: Conclusion

Thus,

$$P\big(|\hat{p}_n - p| < \varepsilon_n\big) \geq 1 - \alpha.$$

But the event $\{|\hat{p}_n - p| < \varepsilon_n\}$ is equivalent to

$$p \in [\hat{p}_n - \varepsilon_n, \hat{p}_n + \varepsilon_n].$$

Therefore,

$$P(p \in I_n) \geq 1 - \alpha,$$

where

$$I_n = \big[\hat{p}_n - \varepsilon_n, \hat{p}_n + \varepsilon_n\big].$$

Hence, I_n is a $(1 - \alpha)$ confidence interval for p based on Hoeffding's inequality.

Problem 5B.

Let $\alpha = 0.05$ and $p = 0.4$. Conduct a simulation study to see how often the confidence interval I_n contains p (called coverage). Do this for n = 10, 100, 1000, 10000. Plot the coverage as a function of n.

Solution

We use code below to simulate how often the confidence interval I_n contains p (called coverage) for n = 10, 100, 1000, 10000.

```python
import numpy as np
import matplotlib.pyplot as plt

def calculate_epsilon(n, alpha):
    return np.sqrt((1 / (2 * n)) * np.log(2 / alpha))

def generate_bernoulli_samples(p, size):
    return np.random.binomial(1, p, size)

def calculate_coverage(n, p_true, alpha, simulations=10000):
    epsilon = calculate_epsilon(n, alpha)
    coverage_count = 0

    for _ in range(simulations):
        X = generate_bernoulli_samples(p_true, n)
        X_ = np.mean(X)

        ci_lower_bound = max(0, X_ - epsilon)  # Ensure non-negative
        ci_upper_bound = min(1, X_ + epsilon)  # Ensure not greater than 1

        # Check if true parameter is in interval
        if ci_lower_bound <= p_true <= ci_upper_bound:
            coverage_count += 1

    # Returns: coverage: fraction of intervals containing true p
    return coverage_count / simulations

def simulate(n_values, p_true, alpha):
    for n in n_values:
        epsilon = calculate_epsilon(n, alpha)
        coverage = calculate_coverage(n, p_true, alpha, simulations=10000)

        coverage_results.append(coverage)
        epsilon_values.append(epsilon)

    # Plotting
    plt.figure(figsize=(10, 6))
    plt.plot(n_values, coverage_results, 'bo-', linewidth=2, markersize=15, label='Coverage')
    plt.axhline(y=1-alpha, color='r', linestyle='--', linewidth=1, label=f'Theoretical minimum: {1-alpha}')

    # Set annotations for each point
    for _, (n, coverage) in enumerate(zip(n_values, coverage_results)):
        plt.annotate(f'{coverage:.4f}',
                     xy=(n, coverage),
                     xytext=(0, 15),
                     textcoords='offset points',
                     ha='center',
                     fontsize=10)

    plt.title('Coverage vs Sample Size\n(α = 0.05, p = 0.4)', fontsize=14)
    plt.xlabel('Sample Size (n)', fontsize=10)
    plt.ylabel('Coverage', fontsize=10)
    plt.xscale('log') # for visualization proportion
    plt.grid(True, alpha=0.3)
    plt.legend(fontsize=10)
    plt.ylim(0.9, 1.01)
    plt.tight_layout()
    plt.show()

alpha = 0.05
p_true = 0.4
n_values = [10, 100, 1000, 10000]

```

```
coverage_results = []
epsilon_values = []

simulate(n_values, p_true, alpha)


print("Coverage Analysis")
print("α = 0.05, p = 0.4")
print("-" * 40)

print(f"\nTheoretical guarantee: P(p ∈ In)     {1-alpha} = {1-alpha}")
print(f"All simulated coverages meet guarantee: {all(c >= 1-alpha for c in
    coverage_results)}")

print(f"\nSummary:")
for i in range(0, len(epsilon_values)):
    interval_length = 2 * epsilon_values[i]
    print(f"n = {n_values[i]:5d}: ϵ = {epsilon_values[i]:.4f}, Coverage = {
    coverage_results[i]:.4f}, Interval length = {interval_length:.4f}")
```

The following are the diagrams:

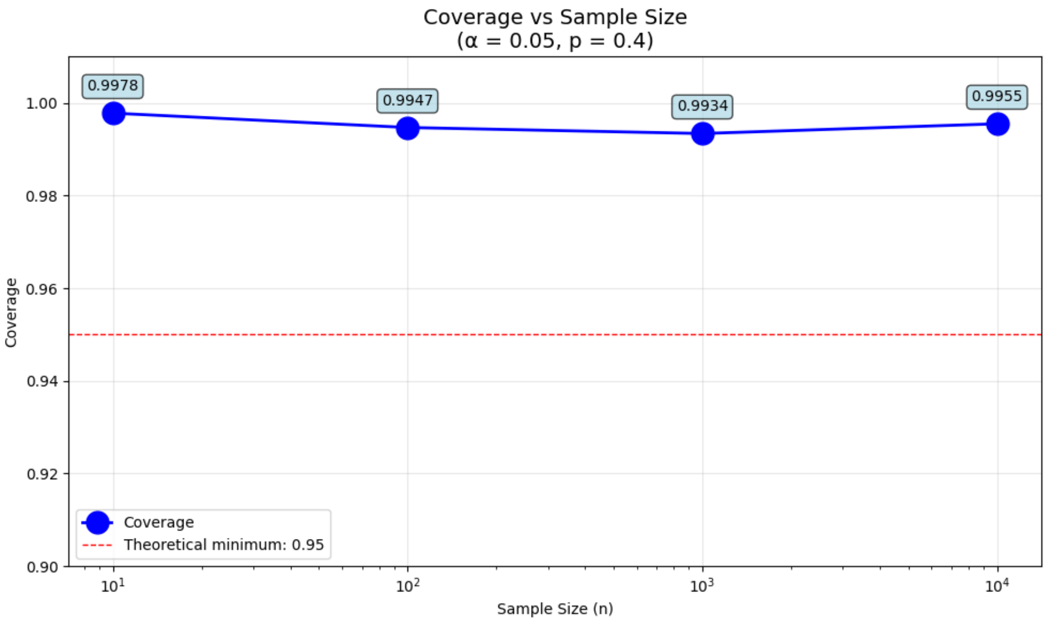

Figure 1: Coverage Analysis Diagram

Analysis

The table below shows the details of the experiment.

n	ε	Coverage(p)	Interval length
10	0.4295	0.9975	0.8589
100	0.1358	0.9922	0.2716
1000	0.0429	0.9935	0.0859
10000	0.0136	0.9946	0.0272

Table 1: Simulation results: ε, coverage probability(p), and interval length for different n.

$$\text{Theoretical guarantee: } P\big(p \in I_n\big) \geq 0.95$$

Conclusion: All simulated coverages meet the guarantee

Problem 5C.

Plot the length of the confidence interval as a function of n

Solution

The following code produces a diagram of the confidence interval.

```python
import numpy as np
import matplotlib.pyplot as plt

alpha = 0.05
n_values = [10, 100, 1000, 10000]
ci_lengths = []

def calculate_epsilon(n, alpha):
    return np.sqrt((1 / (2 * n)) * np.log(2 / alpha))

def draw_ci_length_diagram():
    for n in n_values:
        epsilon = calculate_epsilon(n, alpha)
        ci_length = 2*epsilon
        ci_lengths.append(ci_length)

    # Plotting
    plt.figure(figsize=(10, 6))
    plt.plot(n_values, ci_lengths, 'bo-', linewidth=2, markersize=15, label='CI Length'
    )
    for _, (n, cil) in enumerate(zip(n_values, ci_lengths)):
        plt.annotate(f'n={n}\nlen={cil:.4f}',
                     xy=(n, cil),
                     xytext=(0, 15),
                     textcoords='offset points',
                     fontsize=10)
    plt.title('Confidence Interval Length vs Sample Size (α = 0.05)', fontsize=14)
    plt.xlabel('Sample Size (n)', fontsize=10)
    plt.ylabel('Confidence Interval Length', fontsize=10)
    plt.xscale('log') # for visualization proportion
    plt.grid(True, alpha=0.3)
    plt.legend(fontsize=10)
    plt.ylim(-0.2, 1)
    plt.tight_layout()
    plt.show()

draw_ci_length_diagram()
```

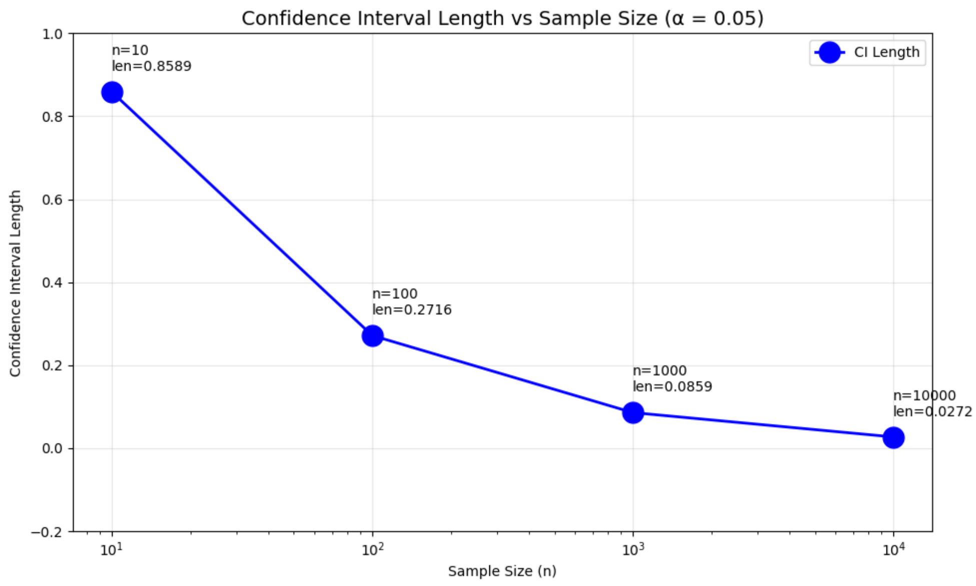

Figure 2: Confidence Interval Length as function of n

From the diagram, **we conclude that by increasing the sample size (n) the length of the confidence interval shrinks because our confidence about the estimate increases**.

Problem 5D.

Say that X_1, ..., X_n represents if a person has a disease or not. Let us assume that unbeknownst to us the true proportion of people with the disease has changed from p = 0.4 to p = 0.5. We use the confidence interval to make a decision, that is when presented with evidence (samples) we calculate In and our decision is that the true proportion of people with the disease is in I_n. Conduct simulation study to answer the following question: Given that the true proportion has changed, what is the probability that our decision is correct? Again using n = 10, 100, 1000, 10000.

Solution

The code used to solve this exercise is provided below. It uses the same $\alpha = 0.05$ value as exercise 5B, however, now the true population proportion is $p = 0.5$, instead of the previous 0.4. The experiment is run 5000 times for each of the n-values. The code checks and counts the times where the p-value is in the confidence interval and returns a probability in the end.

```python
import numpy as np
import matplotlib . pyplot as plt

def calculate_epsilon(n, alpha):
    return np.sqrt((1 / (2 * n)) * np.log(2 / alpha))

def generate_bernoulli_samples(p, size):
    return np.random.binomial(1, p, size)

def exercise_5d(n):
    p = 0.5
    alpha = 0.05
    in_the_interval = 0
    experiments = 5000

    epsilon = calculate_epsilon(n, alpha)

    for _ in range(experiments):
        random_samples = generate_bernoulli_samples(p, size=n)
        mean = np.mean(random_samples)
        ci_upper_bound = mean + epsilon
        ci_lower_bound = mean - epsilon
        if ci_lower_bound <= p <= ci_upper_bound:
            in_the_interval += 1

    probability = in_the_interval/experiments
    probabilities.append(probability)

    return probability


list_of_n = [10, 100, 1000, 10000]
probabilities = []

for n in list_of_n:
    print(f"The probability for {n} is: {exercise_5d(n)}")

print(probabilities)
plt.figure()
plt.plot(list_of_n, probabilities, marker='s', label='Correct decision with p=0.5')
plt.axhline(1 - 0.05, linestyle='--', label='Theoretical minimum 1  alpha  ', color='r')
for x, y in zip(list_of_n, probabilities):
    plt.annotate(f"{y:.4f}", (x, y), textcoords="offset points", xytext=(0, -14), ha='
    center', fontsize=9)

plt.xscale('log')
plt.ylim(0.94, 1.0)
plt.xlabel('n')
plt.ylabel('Probability')
plt.title('Hoeffding CI: Probability vs n')
plt.legend()
plt.tight_layout()
plt.show()
```

All of the found probabilities are higher than the theoretical minimum displayed in Figure 3. From the figure, we can see that the probability decreases a bit when n gets bigger, however the difference in small when n is 100 and more.

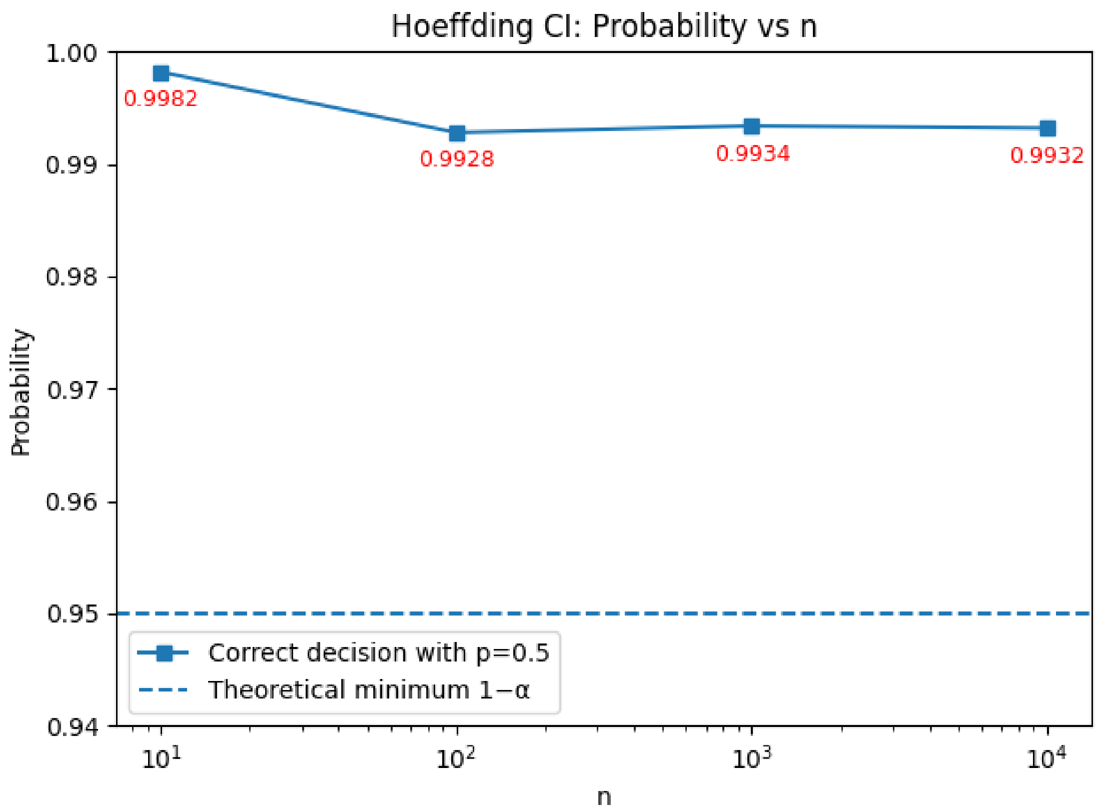

Figure 3: Probability given the true population proportion has changed, over n = (10, 100, 1000, 10000)