# Group Assignment 1

Daniel Cada, Albert Bargalló i Sales, Erik Schahine,
Michel Messo, and Moaaz Tameer Islam

September 27, 2024

## 1

If A and B are independent events, the following applies:

$$A^c \cap B^c = (A \cup B)^c = 1 - P(A \cup B) = 1 - (P(A) + P(B) - P(A \cap B)) = \tag{1}$$
$$1 - (P(A) + P(B) - (P(A) \cdot P(B))) = 1 - P(A) - (P(B) \cdot (1 - P(A)) = \tag{2}$$
$$(1 - P(A) \cdot (1 - P(B)) = P(A^c) \cdot P(B^c) \tag{3}$$

### 2.a

Event A: At least two kids have Brown Hair
Event B: At least one has brown hair

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{P(A)}{P(B)} \tag{4}$$

Due to the fact that if two kids have Brown hair, one kid also has to have brown hair. As having brown hair is an independent event, we can write the distribution for having brown hair as the following:

$$X \sim Bin(n = 3, p = \frac{1}{4}) \tag{5}$$

Since the probability that at least one has is the same as saying that not 0 has, we can write the probability as:

$$\frac{P(X = 2) + P(X = 3)}{1 - P(X = 0)} \tag{6}$$

Calculating that

$$P(X = 3) = \frac{1}{64} \tag{7}$$
$$P(X = 2) = \frac{9}{64} \tag{8}$$
$$P(X = 0) = \frac{27}{64} \tag{9}$$

Plugging that into our original equation we get that:

$$P(A) = \frac{\frac{9}{64} + \frac{1}{61}}{\frac{37}{64}} = \frac{10}{37} \tag{10}$$

### 2.b

Let's say that A=Youngest Kid, B=Middle Kid, C=Oldest Kid. Then the probability that at least two kids have brown hair, if we know that C has brown hair can be described as the probability that either A or B has brown hair.

$$P(A) = \frac{1}{4} \tag{11}$$
$$P(B) = \frac{1}{4} \tag{12}$$
$$P(A \cup B) = P(A) + P(B) - P(A \cap B) = \frac{1}{4} + \frac{1}{4} - \frac{1}{16} = \frac{7}{16} \tag{13}$$

# 3

Let's begin by using the fact that the integral over the region $A = x^2 + y^2 \leq 1$ is equal to 1 (since every possible $r$ will be in this region).

$$\int_A \int f \; dxdy = 1 \tag{14}$$

Since we know that $f$ is uniform, it's a constant. Then we can call if "$c$" and move it out of the integral and put it on the right-hand side of the equation. Resulting in the following equation. The area A can be described as:

$$\int_A \int 1 \; dxdy = \frac{1}{c} \tag{15}$$

Since we are working on the unit circle, we know that the area of the circle is equal to $\pi$. That is $\frac{1}{c} = \pi \rightarrow f = c = \frac{1}{\pi}$. The distribution function $F(r)$ can be described using polar coordinates where we are left with an extra $r$ due to the determinant of the Jacobian.

$$F(r) = \int_{R \leq r} \int f(r) \; dxdy = \int_0^{2\pi} \int_0^r f(r) \cdot r \; drd\theta \xrightarrow{f=\frac{1}{\pi}} \int_0^{2\pi} \int_0^r \frac{1}{\pi} \cdot r \; drd\theta = \int_0^{2\pi} \frac{r^2}{2\pi} \; d\theta = \frac{r^2}{2\pi} \cdot [\theta]_0^{2\pi} = r^2 \tag{16}$$

Thus we can say that the Probability Density Function $f(r)$ is $\frac{1}{\pi}$. And the Cumulative Distribution Function is $F(r) = r^2$.

Note for next time, the two Fs we are trying to find are:

$$f_{X,Y}(x,y) = \frac{1}{\pi}, \quad \text{for } x^2 + y^2 \leq 1 f_R(r) = \frac{d}{dr} F_R(r) = 2r, \quad \text{for } 0 \leq r \leq 1 \tag{17}$$

# 4

This is confusing,y

The flipping of a fair coin can be described as the following geometric series.

$$(1-p)^{n-1} \cdot p = (\frac{1}{2})^{n-1} \cdot \frac{1}{2} = \frac{1}{2^n} \tag{18}$$

Using the expected value formula for a geometric series from the lecture notes, we can calculate the expected value below:

$$E(X) = \frac{1}{p} \tag{19}$$

Where $p = \frac{1}{2}$.

$$E(X) = \frac{1}{\frac{1}{2}} = 2 \tag{20}$$

# 5

## 5.a

We will be using Hoeffding's (primarily the second equation):

$$P\left(\overline{X}_n - E\left[\overline{X}_n\right] \leq -\epsilon\right) \leq e^{-\frac{2n\epsilon^2}{(b-a)^2}} \tag{21}$$

$$P\left(\left|\overline{X}_n - E\left[\overline{X}_n\right]\right| \geq \epsilon\right) \leq 2e^{-\frac{2n\epsilon^2}{(b-a)^2}} \tag{22}$$

In this setting, we have a=0, b=1 since X is Bernoulli. We begin by setting $\alpha = 2e^{-2n\epsilon^2}$, rearranging we get that $\epsilon_n = \sqrt{\frac{1}{2n} ln(\frac{2}{\alpha})}$

$$P(p \in I_n) = P\left(\hat{p}_n - \sqrt{\frac{1}{2n}\ln\left(\frac{2}{\alpha}\right)} \leq p \leq \hat{p}_n + \sqrt{\frac{1}{2n}\ln\left(\frac{2}{\alpha}\right)}\right) \tag{23}$$

Using Hoeffding's inequality:

$$= P\left(|\hat{p}_n - p| > \epsilon_n\right) \leq \alpha \tag{24}$$

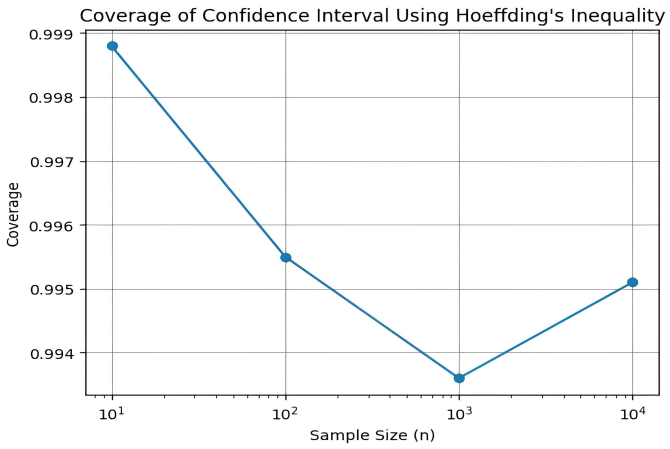
$$= 1 - P\left(|\hat{p}_n - p| > \sqrt{\frac{1}{2n}\ln\left(\frac{2}{\alpha}\right)}\right) \geq 1 - \alpha. \tag{25}$$

### 5.b

We ran the following code in our Jupyter notebook:

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

# Parameters
alpha = 0.05
p = 0.4
n_values = [10, 100, 1000, 10000]
num_experiments = 10000
coverages = []

# Simulation
for n in n_values:
    coverage_count = 0
    for _ in range(num_experiments):
        samples = np.random.binomial(1, p, n)
        p_hat = np.mean(samples)
        epsilon = np.sqrt(np.log(2/alpha) / (2*n))
        lower_bound = p_hat - epsilon
        upper_bound = p_hat + epsilon
        if lower_bound <= p <= upper_bound:
            coverage_count += 1
    coverage = coverage_count / num_experiments
    coverages.append(coverage)

# Plotting
plt.plot(n_values, coverages, marker='o', linestyle='-')
plt.xscale('log')
plt.xlabel('Sample Size (n)')
plt.ylabel('Coverage')
plt.title('Coverage of Confidence Interval Using Hoeffding\'s Inequality'
    )
plt.grid(True)
plt.show()

print(np.array(coverages))
```

Giving us the following graph:

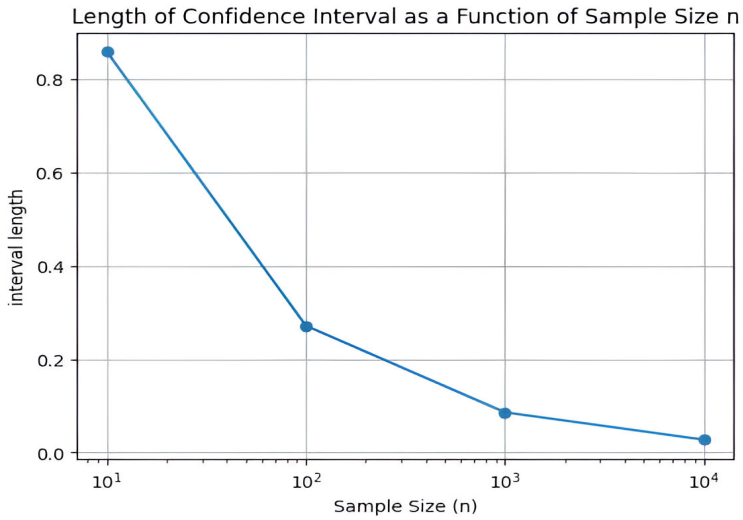


### 5.c

We ran the following code in our Jupyter notebook:

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

# Parameters
alpha = 0.05
p = 0.4
n_values = [10, 100, 1000, 10000]
num_experiments = 10000
intervals = []

# Simulation
for n in n_values:
    coverage_count = 0
    for _ in range(num_experiments):
        samples = np.random.binomial(1, p, n)
        p_hat = np.mean(samples)
        epsilon = np.sqrt(np.log(2/alpha) / (2*n))
        lower_bound = p_hat - epsilon
        upper_bound = p_hat + epsilon
    interval_length = upper_bound - lower_bound
    intervals.append(interval_length)

# Plotting
plt.plot(n_values, intervals, marker='o', linestyle='-')
plt.xscale('log')
plt.xlabel('Sample Size (n)')
plt.ylabel('interval length')
plt.title('Length of Confidence Interval as a Function of Sample Size n')
plt.grid(True)
plt.show()
```

Giving us the following graph:



**5.d**

We ran the following code in our Jupyter notebook:

```
import numpy as np

# Parameters
p_true = 0.5
p_init = 0.4
alpha = 0.05
num_simulations = 1000  # Number of simulations for each n
n_values = [10, 100, 1000, 10000]

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

results = {}
for n in n_values:
    correct_decisions = []
    epsilon_n = calculate_epsilon_n(n, alpha)
    for _ in range(num_simulations):
        samples = np.random.binomial(1, p_true, n)
        p_hat = np.mean(samples)

        # Calculating the confidence interval using epsilon_n
        ci_lower = p_hat - epsilon_n
        ci_upper = p_hat + epsilon_n

        '''
        Checking if the initial proportion p_init = 0.4 falls within
        the confidence interval [ci_lower, ci_upper]. This comparison
        is crucial because it evaluates how well our estimated
        confidence interval based on the true proportion p_true = 0.5
        can capture the previously assumed proportion p_init = 0.4.

        In real-world scenarios, the true proportion of a disease
        might change over time due to various factors, but our
        existing medical knowledge or historical data might still
        reflect the older proportion. Here, p_init = 0.4 represents
        this outdated knowledge, while p_true = 0.5 is the current,
        true proportion.
        '''
        is_correct = (ci_lower <= p_init <= ci_upper)
        correct_decisions.append(is_correct)

    # Calculating the probability that p_init falls within the
       constructed confidence interval
    correct_probability = np.mean(correct_decisions)
    results[n] = correct_probability

# Results
The results obtained are:
```

```
    For n = 10, P = 0.992
    For n = 100, P = 0.756
    For n = 1000, P = 0.001
    For n = 10000, P = 0.000
```