

# EE 230 Spring 2021-2022

## HW 1

**Group Number: 2443075, 2442630**

**Group Members: Abdullah Emir Göğüsdere, İsmail Enes Bülbül**

**1)**

**a)**

$$\Omega = \{x | x \in N, x > 0\}$$

The sample space consists of natural numbers greater than 0. The number represents number of total flipping before someone wins. For example, 2 represents that Ayşe get tail in the first trial, and Bora get head in his first trial. Any number represents that Ayşe wins if the remainder from the division of the number by 3 is 1, and represents Bora wins if the remainder is 2, and represents Ceyda wins otherwise.

**b)**

$$A = \{x | x \in \Omega, x \% 3 = 1\}$$

$$B = \{x | x \in \Omega, x \% 3 = 2\}$$

$$A \cup B = \{x | x \in \Omega, x \% 3 \neq 0\}$$

$$(A \cup B)^c = \{x | x \in \Omega, x \% 3 = 0\}$$

**c)**

Let the events  $E_k$  be defined as follows:

$E_k$ : the event for the case that first  $k-1$  trial is tail and  $k$ th trial is head.

Then we can write events A and B in terms of  $E_k$  as follows:

$$A = E_1 \cup E_4 \cup E_7 \cup \dots$$

$$B = E_2 \cup E_5 \cup E_8 \cup \dots$$

Since these events are disjoint, we can calculate the probabilities as follows:

$$P(A) = P(E_1) + P(E_4) + P(E_7) + \dots$$

$$P(B) = P(E_2) + P(E_5) + P(E_8) + \dots$$

The probability of  $E_k$  can be calculated as:

$$P(E_k) = (1 - p)^{k-1} \cdot p$$

Then we can calculate  $P(A)$  and  $P(B)$  as:

$$P(A) = \sum_{k=1}^{\infty} (1-p)^{3 \cdot (k-1)} \cdot p \\ = p \cdot (((1-p)^3)^0 + ((1-p)^3)^1 + ((1-p)^3)^2 + \dots) = \frac{p}{1-(1-p)^3} = \frac{1}{p^2-3p+3}$$

$$P(B) = \sum_{k=1}^{\infty} (1-p)^{3 \cdot (k-1)+1} \cdot p \\ = p \cdot (1-p) \cdot (((1-p)^3)^0 + ((1-p)^3)^1 + ((1-p)^3)^2 + \dots) = \frac{p \cdot (1-p)}{1-(1-p)^3} = \frac{1-p}{p^2-3p+3}$$

d)

Since events A and B are disjoint, by axiom 2, we have

$$P(A \cup B) = P(A) + P(B) = \frac{1}{p^2-3p+3} + \frac{1-p}{p^2-3p+3} = \frac{2-p}{p^2-3p+3}$$

Since  $(A \cup B)$  and  $(A \cup B)^c$  are disjoint, and  $\Omega = (A \cup B) + (A \cup B)^c$  we have  $P((A \cup B)^c) = P(\Omega) - P(A \cup B)$

$$\text{By axiom 3, } P(\Omega) = 1. \text{ Therefore } P((A \cup B)^c) = \frac{p^2-2p+1}{p^2-3p+3}$$

e)

For A and B:

$$P(A) + P(B) - 1 = \frac{2-p}{p^2-3p+3} - 1 = \frac{-p^2+2p-1}{p^2-3p+3} = \frac{-(p-1)^2}{(p-3/2)^2+3/4} \leq 0 \quad \forall p \quad 0 \leq p \leq 1$$

$P(A \cap B) = 0$  (events are disjoint)

$$0 \geq \frac{-(p-1)^2}{(p-3/2)^2+3/4} \quad (\text{equation holds})$$

For arbitrary two events C and D:

$$P(C \cap D) \geq P(C) + P(D) - 1$$

$$P(C \cap D) \geq P(C \setminus D) + P(D \setminus C) + 2P(C \cap D) - 1$$

$$1 \geq P(C \setminus D) + P(D \setminus C) + P(C \cap D)$$

$$1 \geq P(C \cup D) \quad (\text{equation holds})$$

2)

a)

$B : A_3 \cap A_5$  given  $A_2$

b)

$$P(B) = P(A_3 \cup A_5 | A_2) = \frac{P(A_2 \cap (A_3 \cup A_5))}{P(A_2)} = \frac{P((A_2 \cap A_3) \cup (A_2 \cap A_5))}{P(A_2)} \\ = \frac{P((A_2 \cap A_3)) + P((A_2 \cap A_5)) - P((A_2 \cap A_3 \cap A_5))}{P(A_2)}$$

$A_2 = \{2, 4, 6, \dots, 100\}$  has 50 elements

$A_2 \cap A_3 = \{6, 12, 18, \dots, 96\}$  has 16 elements

$A_2 \cap A_5 = \{10, 20, 30, \dots, 100\}$  has 10 elements

$A_2 \cap A_3 \cap A_5 = \{30, 60, 90\}$  has 3 elements

$$\frac{P((A_2 \cap A_3)) + P((A_2 \cap A_5)) - P((A_2 \cap A_3 \cap A_5))}{P(A_2)} = \frac{0.16 + 0.1 - 0.03}{0.5} = 0.58$$

c)

$$P(A_a \cup A_b | A_2) = \frac{A_2 \cap (A_a \cup A_b)}{P(A_2)} = \frac{P(A_2 \cap A_a) + P(A_2 \cap A_b) - P(A_2 \cap A_a \cap A_b)}{P(A_2)}$$

If we assume that  $a \neq b$ ,  $a \neq 2$ , and  $b \neq 2$ ; than we have

$$P(A_a \cup A_b | A_2) = \frac{P(A_{2a}) + P(A_{2b}) - P(A_{2ab})}{P(A_2)}$$

To obtain a formula for general case (without any assumption like  $a \neq 2$ ) we can divide subscripts in the numerator by the greatest common divisor.

Therefore our formula will be

$$P(A_a \cup A_b | A_2) = \frac{P(A_{\frac{2a}{\gcd(2,a)}}) + P(A_{\frac{2b}{\gcd(2,b)}}) - P(A_{\frac{2ab}{\gcd(2,a) \cdot \gcd(a,b)}})}{P(A_2)}$$

3)

Let the events  $A_k$  and  $B$  be defined as :

$A_k$ : The event that the family has k children

$B$ : The event that the chosen child is the youngest one

$B'$ : The event that the chosen child is the oldest one

By theorem 2:

$$P(B | A_k) = P(B' | A_k) = 1/k$$

$$P(B') = \sum_{k=1}^4 P(B' | A_k) \cdot P(A_k) = \sum_{k=1}^4 (1/k) \cdot p_k$$

$$P(B) = \sum_{k=1}^4 P(B | A_k) \cdot P(A_k) = \sum_{k=1}^4 (1/k) \cdot p_k$$

$$P(B) = P(B') = 0.25 + 0.45/2 + 0.2/3 + 0.1/4 = 0.56$$

$$P(A_k | B) = \frac{P(B | A_k) \cdot P(A_k)}{P(B)} = \frac{(1/k) \cdot p_k}{0.56}$$

a)

$$P(A_1 | B) = \frac{p_k}{0.56k} = 0.25/0.56 = 0.45$$

b)

$$P(A_3 | B) = \frac{p_k}{0.56k} = 0.2/(3 * 0.56) = 0.12$$

c)

$$P(B') = 0.56$$

4)

a)

$$\binom{5}{3} \cdot p^3 \cdot q^2 = 10 \cdot p^3 \cdot q^2$$

b)

Let the index of the first outcome  $\omega_q$  be  $n$ , then we should have outcomes  $\omega_p$  or  $\omega_r$  for the indexes between 1 and  $n-1$ . We can calculate this probability as:

$$\sum_{k=2}^5 (1-q)^{n-1} \cdot q$$

But we have to remove the case that all outcomes before  $n$ th trial is  $\omega_r$ , so our formula will be:

$$\sum_{k=2}^5 ((1-q)^{n-1} - r^{n-1}) \cdot q = q \cdot \sum_{k=1}^4 ((1-q)^n - r^n)$$

c)

We can use the formula with replacing the upper limit of summation with  $\infty$ .

$$q \cdot \sum_{k=1}^{\infty} ((1-q)^n - r^n) = q \cdot \left( \frac{1}{1-(1-q)} - \frac{1}{1-r} \right) = 1 - \frac{q}{1-r} = \frac{1-r-q}{1-r} = \frac{p}{1-r}$$

5)

## EE230 HW1 MATLAB Question: Parameter Estimation

*İsmail Enes Bülbül, Abdullah Emir Göğüşdere*

*2442630, 2443075*

### Table of Contents

Task: Estimating the probability of success from Bernoulli trials. . . .	4
Part I. Sampling . . . . .	4
Part II. Estimation . . . . .	6
Part III. Evaluation . . . . .	7
Appendix A. Function Declarations . . . . .	10
A.1. Your Estimator . . . . .	10

### Task: Estimating the probability of success from Bernoulli trials.

The Bernoulli random variable is a mathematical abstraction for a possibly unfair coin toss, expressed as follows:

$$X \in \{0, 1\} \quad P(X = 1) = p, \quad P(X = 0) = 1 - p, \quad p \in [0, 1]$$

The  $X = 1$  case is called a *success*, and consequently  $X = 0$  is called as *failure*. In this work, you will be trying to estimate the probability of success  $p$  of a Bernoulli random variable from  $N$  independent & identically distributed (iid.) samples.

## Part I. Sampling

You will begin by using `randsample()` to take independent identically distributed (iid.) samples from the random variable described above. Think about discrete probabilistic model given above: What is the sample space, elementary events, and the assigned probabilities to these elementary events? How do they correspond to the arguments of `randsample()`?

- Assign a variable `N` for the number of samples, and adjust its value with a numeric slider that ranges from 10 to 1000 in steps of 10. Have the control run the current section as its value is changing.
- Assign a variable `p` for the probability of success, and use a numeric slider to adjust its value. Have the control run the current section as its value is changing.
- Use the function `randsample()` to generate `N` samples with probability of success `p`. Consult `help randsample` to figure out how the function works, if you need to.

```
clear variables; close all; clc;
N = 1000;
p = 0.7;

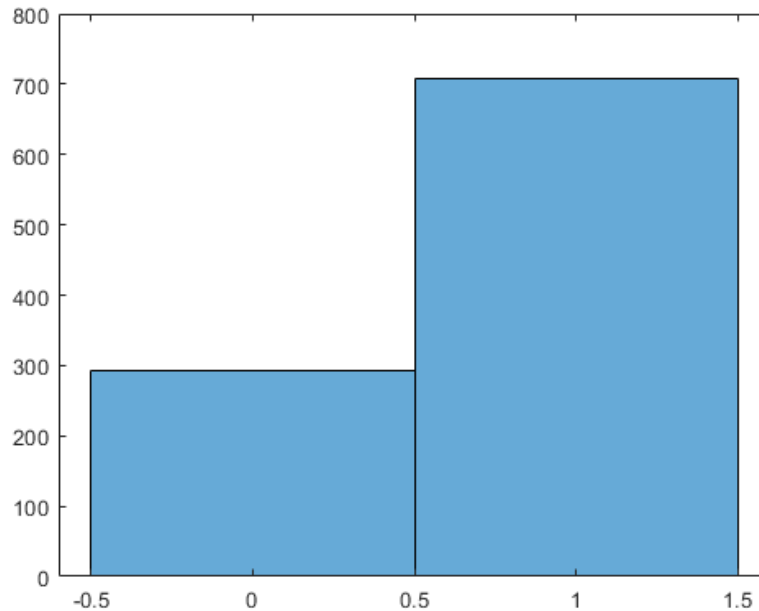
samples = randsample([0 1],N,true,[1-p p]);
```

Now for some visualization.

- Use `histogram()` to plot how many of which outcome has been sampled. Prettify the plot.

```
figure

histogram(samples)
```



## Part II. Estimation

Now that you have your samples, let's do some parameter estimation. You will try to estimate the probability of success via the most intuitive formula: Number of successes divided by the total number of trials.

- Use the previously generated samples and apply this formula to estimate the probability of success.

```
p_est = sum(samples(:) == 1)/N
```

```
p_est = 0.7080
```

What you code above is called an *estimator*: A function that maps the observed samples to the estimate of the quantity in question. You will want to convert your estimator to a stand-alone MATLAB function, so that it may be called when needed.

- Complete the empty function `estimator()` in [Appendix A.1](#). and call it once more below to make sure it works.

```
p_est = estimator(samples)
```

```
p_est = 0.7080
```

### Part III. Evaluation

In this final part, you are asked to evaluate the success of your estimator as a function of number of observations.

- Generate a new, "big" data set of  $N = 1000$  samples, again with  $p$  adjusted with a numeric slider.
- Write a `for` loop to use your estimator with the first  $n$  samples *of the dataset you generate in this section* where  $n$  varies from 1 to  $N$ , and record the estimator output to a vector.
- Plot the estimator output as a function of the number of observed samples. On the same plot, plot a horizontal line that is equal to the value of true  $p$ , so that it is easier to compare the estimated value to the true one.
- Have the final estimate of  $p$  and the true  $p$  printed in the title of the plot. This title must adapt to the final estimate and  $p$  value as set by the numeric slider.
- Add a button control to rerun this section for a new data set of same size, and observe whether different estimated  $p$  vs. nr. of samples used converge to the same final  $p$  estimate. **Bonus:** Plot different realizations of these plot *on the same figure* multiple (5-10 should be plenty). If you do so, you can choose either run's final estimate to show in the title.

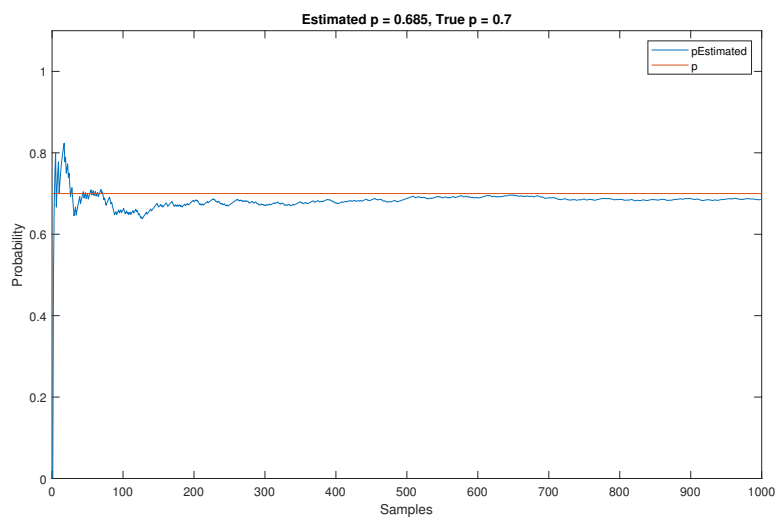
```
f = figure;  
f.Position = [0 0 1000 600];  
  
p = 0.7;  
N = 1000;  
data = randsample([0 1],N,true,[1-p p]);  
  
p_est_ = ones(1,N);  
  
for n = 1:N  
  
    p_est_(n) = estimator(data(1,1:n));  
  
end  
  
plot(p_est_)  
hold on;
```

```

plot(p*ones(1,N))

xlim([0 1000])
ylim([0 1.1])
legend("pEstimated", "p")
title("Estimated p = " + p_est_(1,N) + ", True p = " + p)
ylabel("Probability")
xlabel("Samples")

```



### Bonus:

```

hold off; close all;
f = figure;
f.Position = [0 0 1000 600];

numExp = 5;
pEst = string(zeros(1,numExp));

for repeat = 1:numExp
    data = randsample([0 1],N,true,[1-p p]);

    p_est_ = ones(1,N);

```



```

for n = 1:N

    p_est_(n) = estimator(data(1,1:n));

end

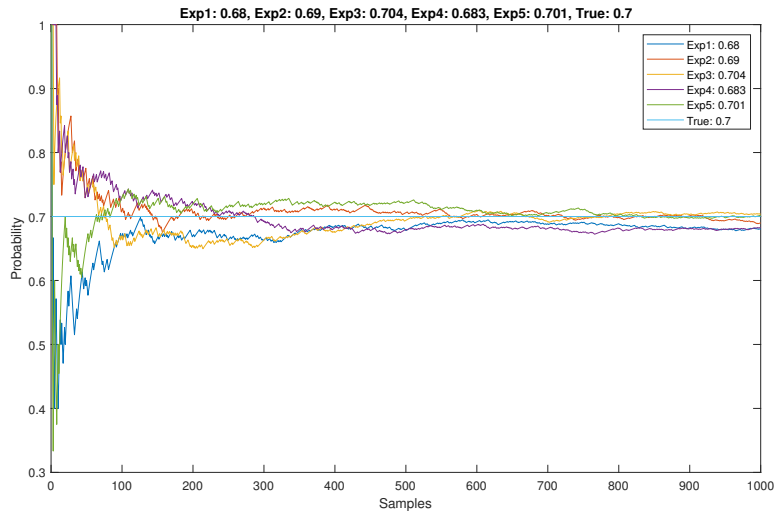
pEst(1,repeat) = "Exp" + ...
string(repeat) + ": " + string(p_est_(1,N));

plot(p_est_)
hold on;

end

pEst = [string(pEst), "True: " + string(p)];
plot(p*ones(1,N))
title(join(pEst,", "))
legend(pEst)
ylabel("Probability")
xlabel("Samples")
hold off;

```



- Comment briefly on the behaviour you observe on the plot you create above. How does your estimator perform? Do the plots vary from run to run? What are the similarities and differences between different runs?

From the above plots, as we increase the portion of the **data** to be calculated in each loop until it reaches the whole portion, it can be seen that the

**pEstimated** converges to the true **p**. Therefore, we can say that the **estimator** performs sufficiently. Moreover, we can see that the plots vary from run to run since the **randsample()** creates **N**-sized samples randomly and closely weighted to true **p**.

The difference between runs is that the **data** sets in each run are created randomly and closely weighted to true **p**, so the data sets in each run are not the same. Although this is the case, their similarity is that their **pEstimated** will eventually converge to the true **p**.

## Appendix A. Function Declarations

### A.1. Your Estimator

```
function [p_est] = estimator(samples)
    [p_est] = sum(samples(:) == 1)/length(samples);
end
```