EE 214 Machine Problem 1.1

Paul Vincent S. Nonat
Electrical and Electronics Engineering Institute
University of the Philippines Diliman
paul.vincent.nonat@eee.upd.edu.ph

1. The Problem

Consider the example on the experiment of coin tossing. A fair coin is tossed 100 times with the relative frequency of heads tabulated as S_0,S_1,...,S_{100} based on the number of heads. Simulate 1,000 repetitions of the experiment and obtain the data on the number of trials $N_{1000}(S_k)$ where S_k is the number of heads. For example $N_{1000}(S_{50}) = 75$, means 75 trials yielded 50 heads.

1.1) Obtain the following data on the sum of the no. of trials for each group (of 3) with the specified no. of heads S_k :

$N_{1000}(S_{33})+N_{1000}(S_{34})+N_{1000}(S_{35}) =;$
$N_{1000}(S_{36})+N_{1000}(S_{37})+N_{1000}(S_{38}) =;$
$N_{1000}(S_{39})+N_{1000}(S_{40})+N_{1000}(S_{41}) =;$
$N_{1000}(S_{42})+N_{1000}(S_{43})+N_{1000}(S_{44}) =;$
$N_{1000}(S_{45})+N_{1000}(S_{46})+N_{1000}(S_{47}) =;$
$N_{1000}(S_{48})+N_{1000}(S_{49})+N_{1000}(S_{50}) =;$
$N_{1000}(S_{51})+N_{1000}(S_{52})+N_{1000}(S_{53}) =;$
$N_{1000}(S_{54})+N_{1000}(S_{55})+N_{1000}(S_{56}) =;$
$N_{1000}(S_{57})+N_{1000}(S_{58})+N_{1000}(S_{59}) =;$
$N_{1000}(S_{60})+N_{1000}(S_{61})+N_{1000}(S_{62}) =;$
$N_{1000}(S_{63})+N_{1000}(S_{64})+N_{1000}(S_{65}) =;$
$N_{1000}(S_{66})+N_{1000}(S_{67})+N_{1000}(S_{68}) =;$

1.2) Using the data in 1.1 by using a bar graph, plot the average number of heads per group vs. k, where k is the number of trials per group. Does the graph resemble a bell curve? Why and why not?

1.3) Using the result of the experiment, compute the probability, expected value and standard deviation of X - the no. of heads per trial (100 tosses). Perform 20 runs and print the results.

2. Methodology

A fair coin tossed in the air has two probable outcomes, either heads or tail. Throwing it 100 times and recording its output can be modeled as independent with one another. Hence, we can model this experiment using a binomial random variable.

To simulate the 100-time coin toss for 1,000 repetitions, The following functions will be implemented in a python program

```
def toss_coin():
    coin=np.random.randint(2)
    return coin;

def experiment(x,no_toss):
    no_heads=0
    no_tails=0
    result=np.zeros(no_toss)
    for x in range(0,no_toss):
        result[x]=toss_coin()
    for i in result:
    #count the number of heads and tails
        if i ==1:
            no_heads = no_heads+1
        if i==0:
            no_tails = no_tails+1
    return no_heads,no_tails;
```

The *toss coin* function emulates a coin flip, which returns a value of 1 for heads, and 0 for tails. To emulate the trials describe in the problem, a parametric *experiment* function will be implemented, which requires the trial number x and the number of coin toss per repetition.

To Compute for probability p, expected value $E[X]$, and standard deviation $\delta(X)$. The simulation will use the following equations

$$p = \frac{h}{n} \tag{1}$$

$$E[X] = np \tag{2}$$

$$\delta(X) = \sqrt{(Var[X])} \tag{3}$$

Where h is the total number of heads per iteration and n for the number of coin toss per trial. Since the coin toss problem can be modeled as a binomial random variable, The formula for calculating $Var[X]$ is equivalent to

$$Var[X] = np(1-p) \tag{4}$$

Thus, we use eqn. 1-4 in the python code to calculate for these values in 20 trials.

```
print("p    E(X)    STD")
for i in range(20):
    p[i]=heads_pr[i]/no_toss
```

```
    E_X[i]=no_toss*p[i]
    std_X[i]=np.sqrt(n*p[i]*(1-p[i]))
    print(str(p[i])+"    "+str(E_X[i])+"    "+str(std_X[i]))
```

To be able to count the number of experiment repetition $N_{1000}(S_k)$,which yields an equal number of heads S_k, the program was configured to scan for all the 1000 repetitions, and count the experiment that returns S_0 heads up to S_100.

```
for x in range(0,no_repetition):
      heads_pr[x],tails_pr[x] =experiment(x,no_toss)
S_heads=np.zeros(no_toss)
for S_k in range(0,no_toss):
    for n in range(0,no_repetition) :
        if heads_pr[n]==S_k+1:
            S_heads[S_k] = S_heads[S_k]+1
```

The above line of code performs this counting method by first recording the number of heads S_k per iteration. After recording, the counting starts until the program was able to record the total number of trials which yield the same S_k.

3. Simulation Result

Problem 1.1 wishes to determine the sum of the no. of trials for each group of 3. It was done by counting the number of trials that yield with the equal S_k and then sum them up according to the prescribed groupings. The result of the simulation was presented in Table 1.

Table 1. Sum of the no. of trials for each S_k

Group of 3 S_k	Sum
$\mathrm{N}_{1000}(S_{33}) + N_{1000}(S_{34}) + N_{1000}(S_{35})$	1
$\mathrm{N}_{1000}(S_{36}) + N_{1000}(S_{37}) + N_{1000}(S_{38})$	5
$\mathrm{N}_{1000}(S_{39}) + N_{1000}(S_{40}) + N_{1000}(S_{41})$	32
$\mathrm{N}_{1000}(S_{42}) + N_{1000}(S_{43}) + N_{1000}(S_{44})$	93
$\mathrm{N}_{1000}(S_{45}) + N_{1000}(S_{46}) + N_{1000}(S_{47})$	168
$\mathrm{N}_{1000}(S_{48}) + N_{1000}(S_{49}) + N_{1000}(S_{50})$	231
$\mathrm{N}_{1000}(S_{51}) + N_{1000}(S_{52}) + N_{1000}(S_{53})$	205
$\mathrm{N}_{1000}(S_{54}) + N_{1000}(S_{55}) + N_{1000}(S_{56})$	178
$\mathrm{N}_{1000}(S_{57}) + N_{1000}(S_{58}) + N_{1000}(S_{59})$	64
$\mathrm{N}_{1000}(S_{60}) + N_{1000}(S_{61}) + N_{1000}(S_{62})$	19
$\mathrm{N}_{1000}(S_{63}) + N_{1000}(S_{64}) + N_{1000}(S_{65})$	4
$\mathrm{N}_{1000}(S_{66}) + N_{1000}(S_{67}) + N_{1000}(S_{68})$	0

Note: this values randomized whenever the prob1.py script is executed.

Upon doing the simulation, it can be observed that the output resembles a bell curve. We confirmed this by plotting the average number of S_k per group vs the number of trials per group. This graph was presented in Fig. 1

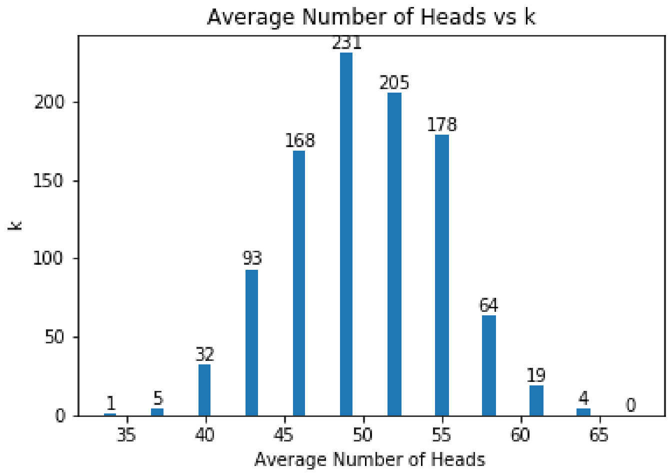

Fig. 1. Characteristic of c(n) function.

The probability p, expected value $E[X]$, and standard deviation $\delta(X)$ of the problem was also calculated using eqn. 1-4. The result for 20 trials was tabulated in Table 2.

4. Conclusion

A python program that simulates a fair coin toss of 100 times for 1000 repetition was successfully implemented. Also, the sum of S_K for groups of three resembles a bell curve. Hence this indicates that the probability for S_k depicts a normal distribution.

Table 2. Problem 1.3 Result

Trial	p	$E[X]$	$\delta(X)$
1	0.51	51	15.80
2	0.50	50	15.80
3	0.58	58	15.60
4	0.50	50	15.80
5	0.55	55	15.72
6	0.45	45	15.72
7	0.49	49	15.80
8	0.45	45	15.72
9	0.53	53	15.77
10	0.55	55	15.72
11	0.54	54	15.75
12	0.48	48	15.79
13	0.51	51	15.80
14	0.47	47	15.77
15	0.54	54	15.75
16	0.50	50	15.80
17	0.49	49	15.80
18	0.50	50	15.80
19	0.48	48	15.79
20	0.55	55	15.72

Note: this values randomized whenever the prob1.py script is executed.