EE511 Assignment1

Zechen Ha

January 2020

1 Question1

Statement:

Simulate tossing a fair coin (a Bernoulli trial) 50 times. Count the number of heads. Record the longest run of heads.

Experiment:

To simulate the process of tossing a fair coin and repeat 50 times independently, I create a random number with random.random() in Python, and set the range [0,0.5] as Tail, and set the range (0.5,1] as Head. Repeat the process for 50 times and count the number of heads and the longest run of heads.

Answer:

Number of Heads: 26

The longest run of Heads: 6

Question-a Statement:

Repeat the above experiment 20, 100, 200, and 1000 times. Generate a histogram for each showing the number of heads in 50 flips. Comment on the limit of the histogram.

Answer:

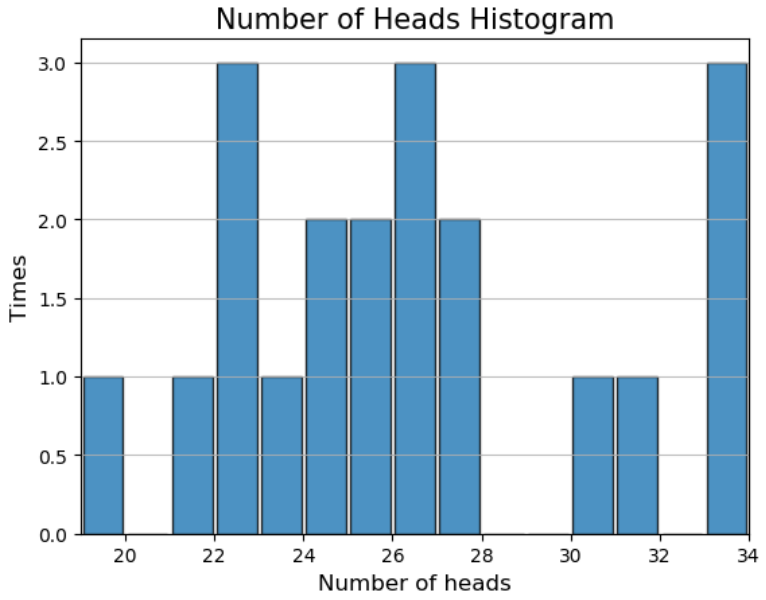

Figure 1: Repeat experiment 20 times

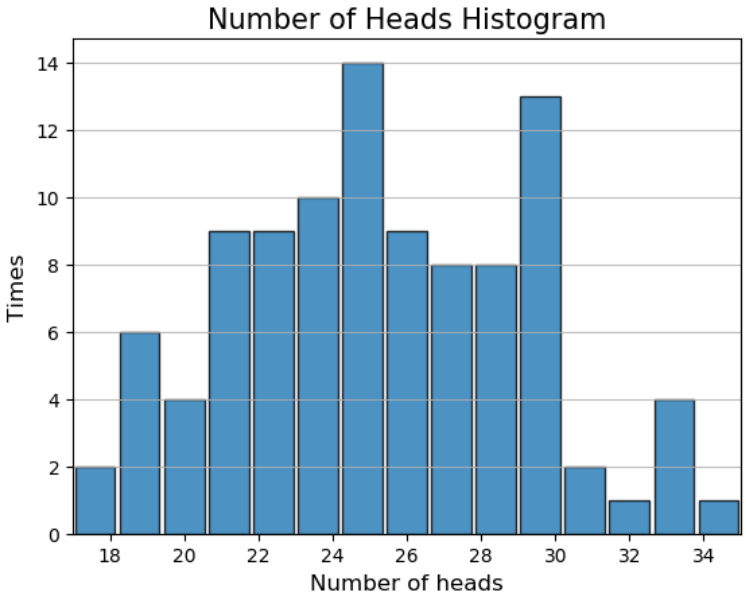

Figure 2: Repeat experiment 100 times

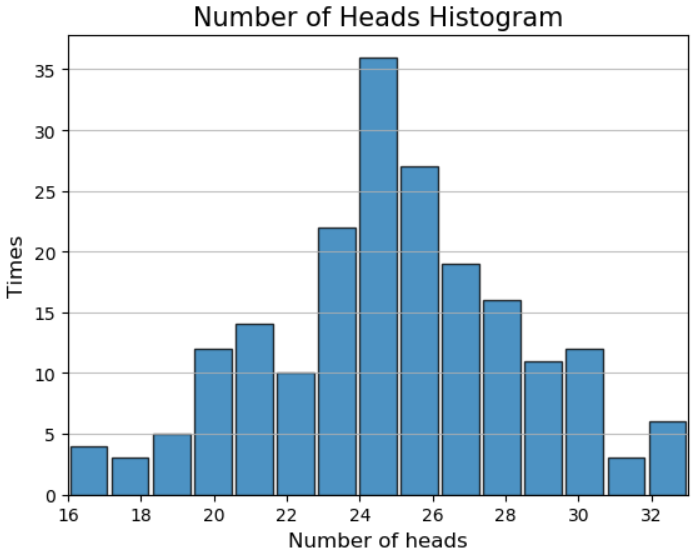

Figure 3: Repeat experiment 200 times

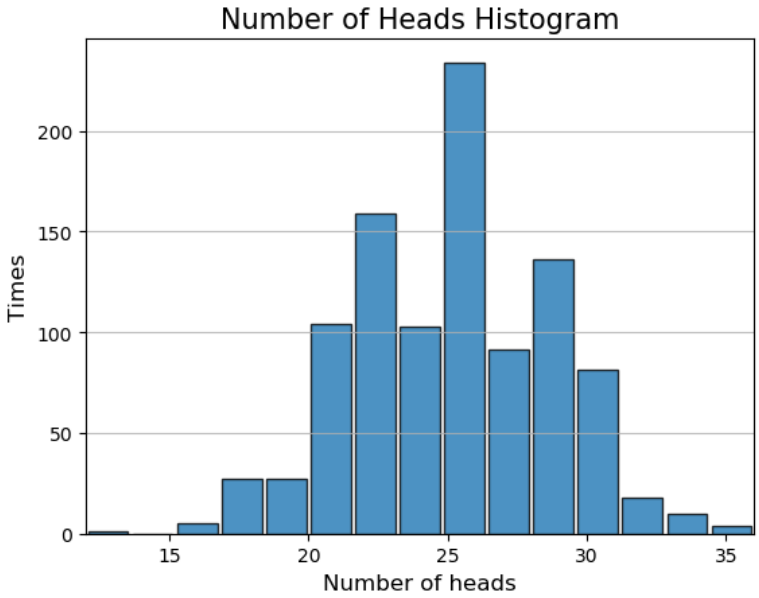

Figure 4: Repeat experiment 1000 times

Comment on the limit of histogram

Each experiment is a Bernoulli trial, which leads to two results with equal probability. Repeat independently and identically in a group of experiments results in binomial trails. In probability theory, the pmf (probability mass function) of binomial hints that probability gets small when n goes far from the mean, and the peak of the distribution is the mean. According to CLT (central limit theorem), the distribution is similar to normal distribution.

We can analyse the limit of the histogram with the help of pmf and CLT. The mean of the binomial distribution is equal to n*p, in which n=50 and p=0.5. So the mean is equal to 25. When the number of heads is far from the mean, there are less times and the times reach peak when number of heads is around 25.

Question1 Code

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

#Initial conditions
fair_coin_boundary = 0.5
count_number_Head = 0
experiment_times = 50
```

```
repeat_times = 1000
run_Head = 0
longest_run_Head = 0
count_number_Head_list = []

#Simulation
for r in range(repeat_times):
    run_Head = 0
    count_number_Head = 0
    for experiment in range(experiment_times):
        if(random.random()>fair_coin_boundary):#Head
            count_number_Head+=1
            run_Head+=1
            if(longest_run_Head < run_Head):
                longest_run_Head = run_Head
        else:#Tail
            run_Head = 0
    count_number_Head_list.append(count_number_Head)

#Output
print(longest_run_Head)
# print(count_number_Head_list)

#Graph
n,bins,patches = plt.hist(count_number_Head_list,bins=15,rwidth
    =0.9,edgecolor='black',alpha=0.8)
plt.xlim(min(count_number_Head_list),max(count_number_Head_list))
plt.grid(axis='y',alpha=0.8)
plt.ylabel('Times',fontsize=12)
plt.xlabel('Number of heads',fontsize=12)
plt.title('Number of Heads Histogram',fontsize=15)
plt.show()
```

2 Question2

Statement:

Simulate tossing a biased coin 200 times where P[HEAD]=0.80. Count the number of heads. Record the longest run of heads. Generate a histogram for the Bernoulli outcomes.

Experiment:

The experiment is similar to the first one. Simulate the Bernoulli trial with Python function random.random(), and set the range of (0.2,1] as heads and set the range of [0,0.2] as tails. Repeat the experiment 200 times. Count the number of heads and the longest run of heads during the process.

Answer:

The number of heads: 165

The longest run of heads: 13

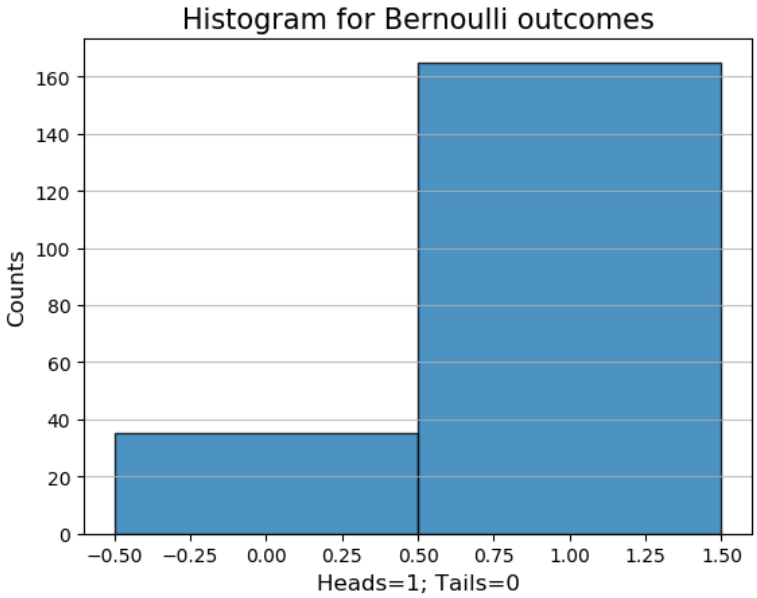

Figure 5: Histogram for Bernoulli outcomes

In probability theory, the mathematical expectation of Bernoulli distribution is n*p, which is equal to 200*0.8=160. The result of the simulation is approximately the same as the expectation.

Question2 Code

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

#Initial conditions
fair_coin_boundary = 0.2
experiment_times = 200
repeat_times = 1
run_Head = 0
longest_run_Head = 0
count_number_Head_list = []
count_number_Head = 0

#Simulation
for r in range(repeat_times):
    run_Head = 0
    for experiment in range(experiment_times):
        if(random.random()>fair_coin_boundary):#Head
            count_number_Head_list.append(1)
            count_number_Head+=1
```

```
            run_Head+=1
            if(longest_run_Head < run_Head):
                longest_run_Head = run_Head
        else:#Tail
            run_Head = 0
            count_number_Head_list.append(0)

#Output
print(longest_run_Head)
print(count_number_Head)

#Graph
n,bins,patches = plt.hist(count_number_Head_list,bins=2,rwidth=1,
    edgecolor='black',alpha=0.8,range=(-0.5,1.5))
# plt.xlim(min(count_number_Head_list),max(count_number_Head_list))
plt.grid(axis='y',alpha=0.8)
plt.ylabel('Counts',fontsize=12)
plt.xlabel('Heads=1; Tails=0',fontsize=12)
plt.title('Histogram for Bernoulli outcomes',fontsize=15)
plt.show()
```

3 Question3

Statement:

Simulate tossing a fair coin 100 times. Generate a histogram showing the heads run lengths.

Experiment:

The experiment is similar and I just count the length of the heads run during the group of experiments.

Answer:

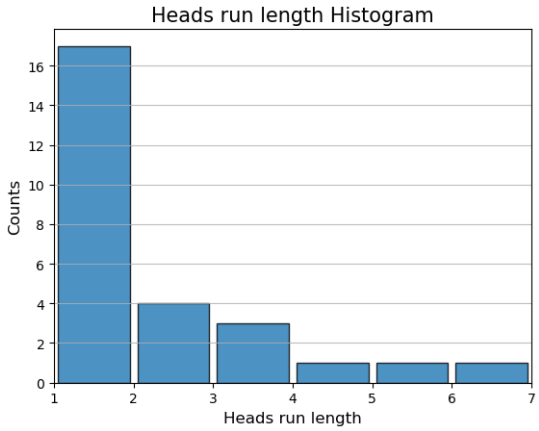

Figure 6: Run heads length histogram

Question3 Code

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

#Initial conditions
fair_coin_boundary = 0.5
experiment_times = 100
repeat_times = 1
run_Head = 0
run_length_list = []

#Simulation
for r in range(repeat_times):
    run_Head = 0
    for experiment in range(experiment_times):
        if(random.random()>fair_coin_boundary):#Head
            run_Head+=1
        else:#Tail
            if(run_Head!=0):
                run_length_list.append(run_Head)
            run_Head = 0

#Output
print(run_length_list)

#Graph
n,bins,patches = plt.hist(run_length_list,bins=6,rwidth=0.9,
    edgecolor='black',alpha=0.8)
plt.xlim(min(run_length_list),max(run_length_list))
plt.grid(axis='y',alpha=0.8)
plt.ylabel('Counts',fontsize=12)
plt.xlabel('Heads run length',fontsize=12)
plt.title('Heads run length Histogram',fontsize=15)
plt.show()
```

4 Question4

Statement:

Simulate tossing a fair coin and count the number of tosses until reaching a user-specified positive number of heads.

Experiment:

Use the input function to get the user-specified positive number of heads. Then check whether it is a valid number. Simulate the Bernoulli trial in a similar way to count the heads number until it reaches the set goal.

Answer:

Number of Heads	Number of Experiment	Heads/Experiment
100	205	0.489
500	985	0.508
1000	2018	0.496
5000	10110	0.495

The number of the needed number of experiment is approximately twice as many as the user-specified number of heads. It meets the expectation of the Bernoulli trial.

Question4 Code

```
import random

#initial conditions
fair_coin_boundry = 0.5
experiment_number = 0
head_number = 0

#Input validation test
try:
    #input number of heads
    goal = int(input("Please input the positive number of heads:"))
    #Simulation
    while(head_number<goal):
        if(random.random()>fair_coin_boundry):#Head
            head_number+=1
        experiment_number+=1

    #Output
    print(experiment_number)
    print(head_number)
except:
    print("Invalid input, please check")
```