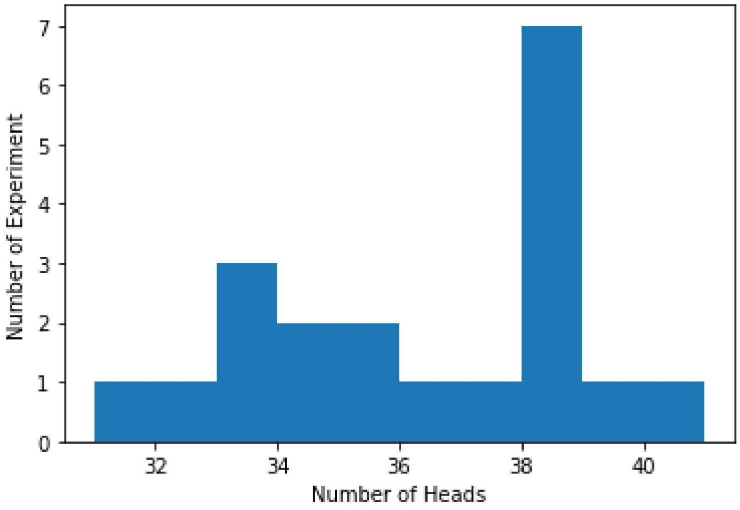
Name: Venkata Meghana Achanta

USC ID: 2578990261

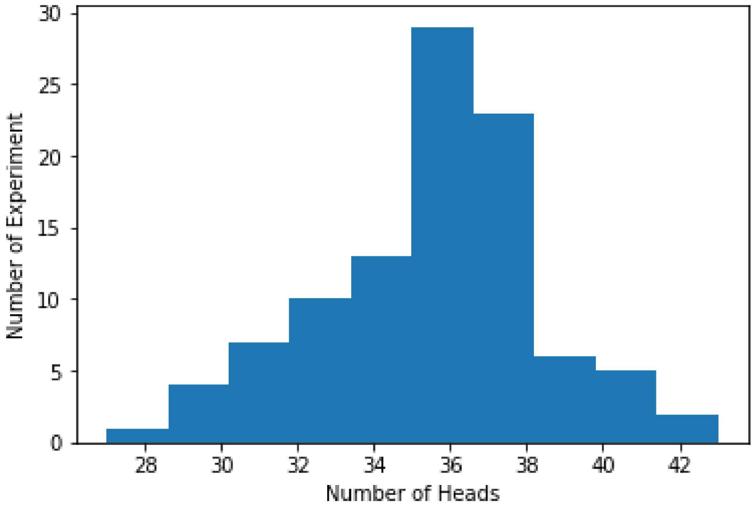
Q1.a) The total number of heads recorded for a biased coin with P[H]= 0.7, when tossed 50 times is 35 and the longest run of heads obtained is 12.

Q1.b) As the number of times the experiment is repeated for 50 trials, the histogram takes the shape of a normal distribution(results in a bell curve). The probability of obtaining a head is more in the middle of the total experiments performed and reduces at the end.

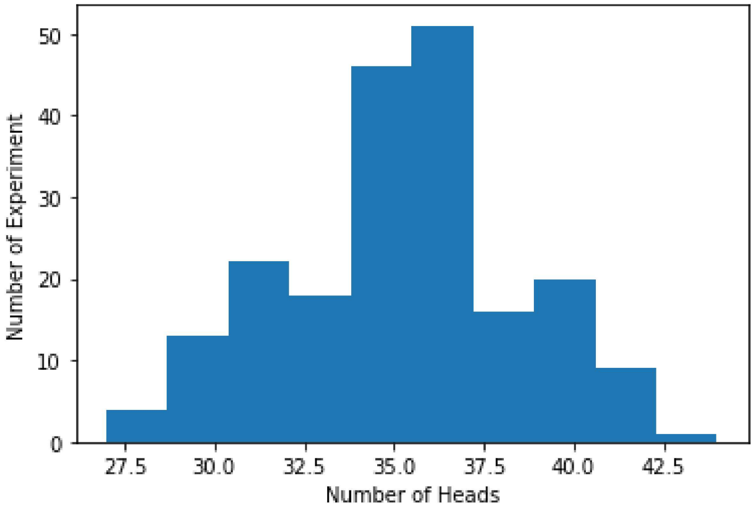
i. Histogram for n = 20



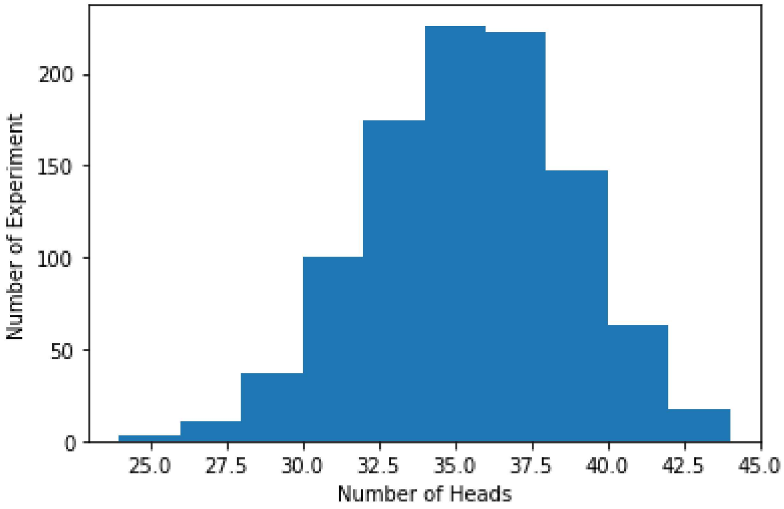
ii.Histogram for n = 100



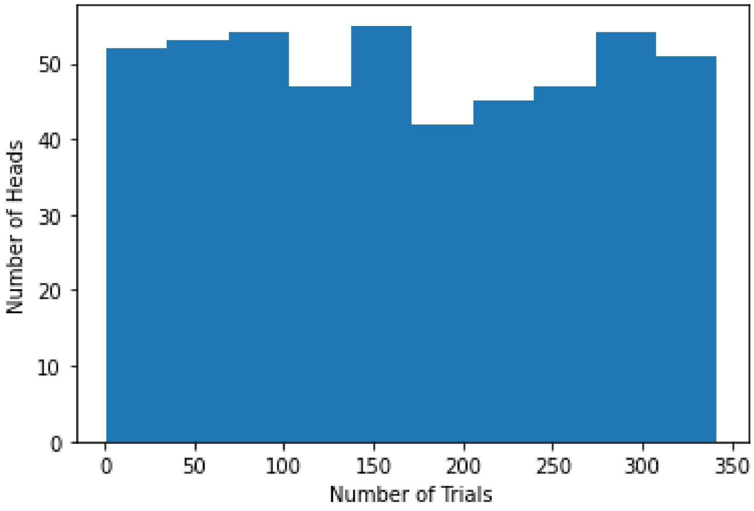
iii. Histogram for n = 200



iv. Histogram for n = 1000

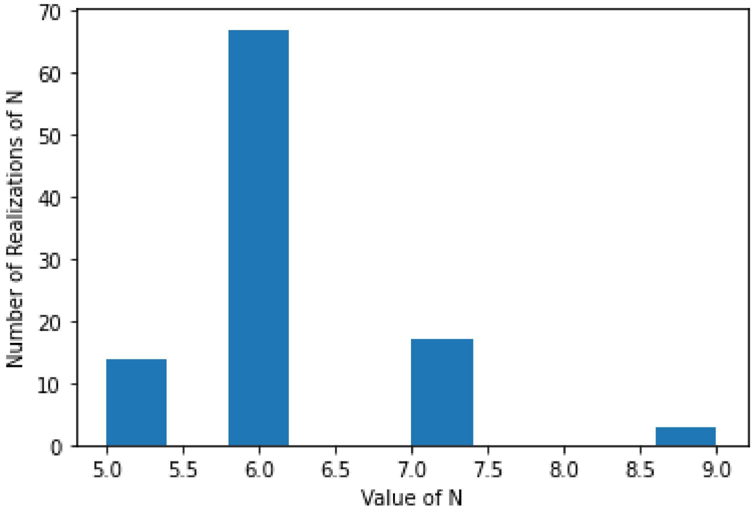


Q1.c) The histogram depicting the heads run length when the coin is tossed 500 times is as follows:

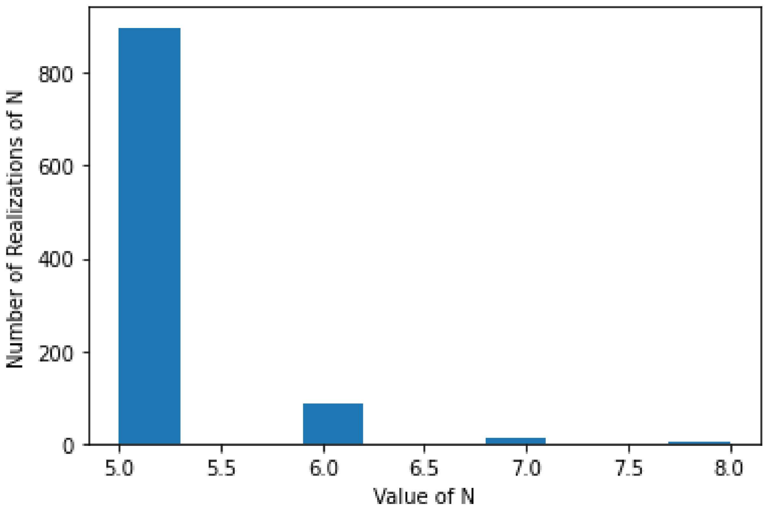


Q2. As the realizations of N increases, the expected value of N decreases.

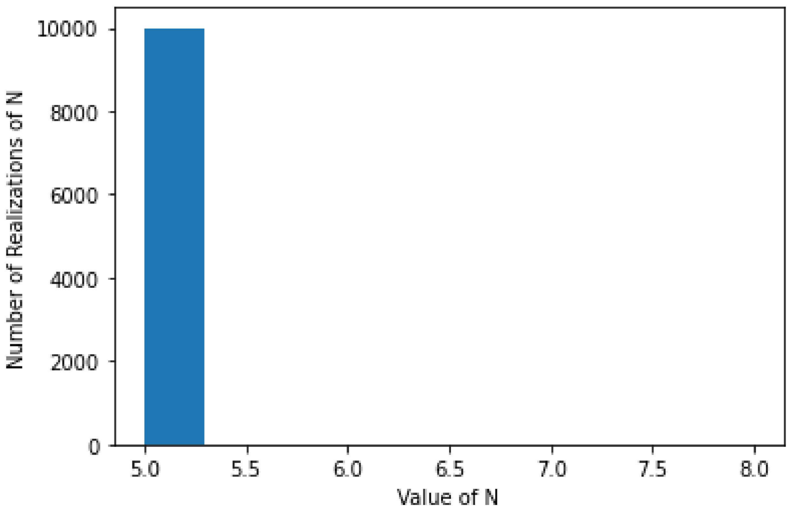
i. Histogram for N = 100



ii. Histogram for N = 1000



iii. Histogram for N = 10000



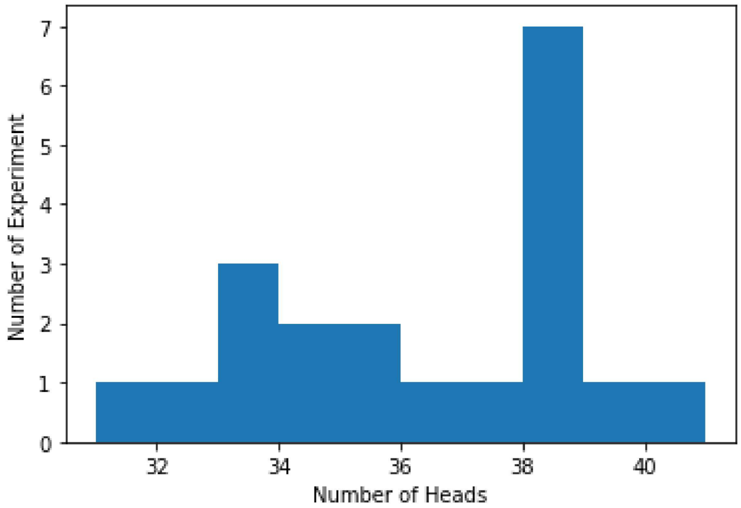
Appendix

```
import sys
import random
import matplotlib.pyplot as plt
import math
```

```
#Question 1a
N = 50
p = 0.7
number_of_heads = 0
number_of_cont_heads = 0
n = 100
freq = []
c = 0

for i in range(1, N+1):
  toss = random.random()
  if toss < p:
    number_of_heads += 1
    c += 1
  else:
    number_of_heads = 0
  number_of_cont_heads = max(number_of_cont_heads, number_of_heads)
  freq.append(number_of_cont_heads)
  num_of_heads = 0
  num_cont_of_heads = 0
print('The longest run of obtaining heads is:', max(freq))
print('The total number of times head was obtained when the biased coin was tossed is:',c)
```

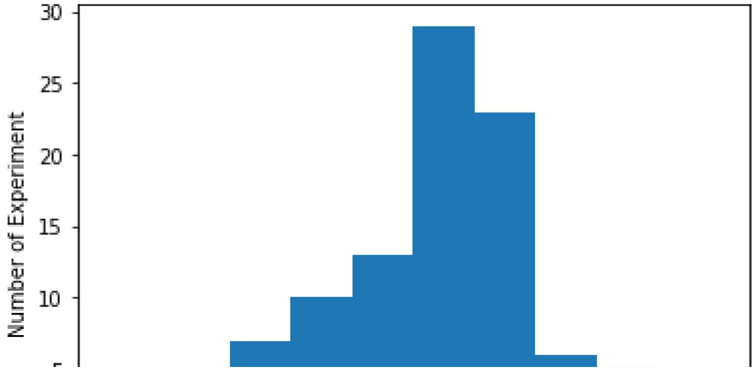
```
The longest run of obtaining heads is: 12
The total number of times head was obtained when the biased coin was tossed is: 35
```

```
#Question 1b for n =20
N = 50
p = 0.7
number_of_heads = 0
number_of_cont_heads = 0
n = 20
freq = []

for j in range(n):
  for i in range(1, N+1):
    toss = random.random()
    if toss < p:
      number_of_heads += 1
  freq.append(number_of_heads)
  number_of_heads = 0

plt.hist(freq)

plt.ylabel('Number of Experiment')
plt.xlabel('Number of Heads');
```

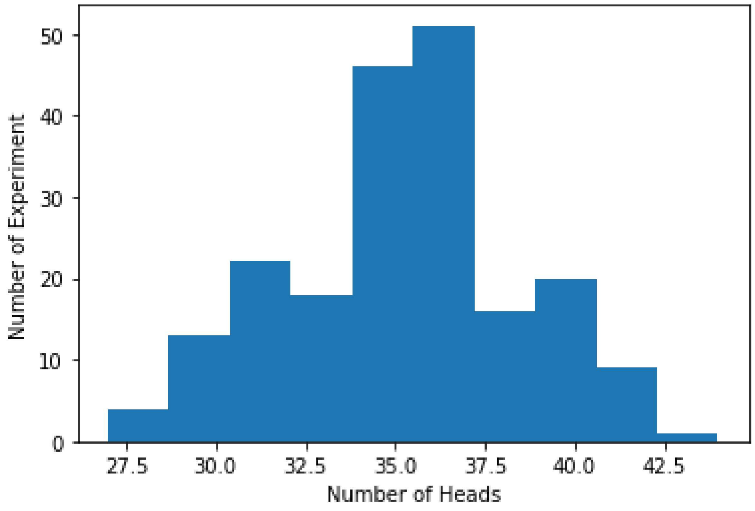


```
#Question 1b for n = 100

N = 50
p = 0.7
number_of_heads = 0
number_of_cont_heads = 0
n = 100
freq = []

for j in range(n):
  for i in range(1, N+1):
    toss = random.random()
    if toss < p:
      number_of_heads += 1
  freq.append(number_of_heads)
  number_of_heads = 0
print(freq)
plt.hist(freq)

plt.ylabel('Number of Experiment')
plt.xlabel('Number of Heads');
```

```
[36, 37, 30, 34, 37, 36, 35, 32, 34, 32, 31, 36, 35, 39, 29, 35, 32, 35, 30, 35, 37, 31,
```

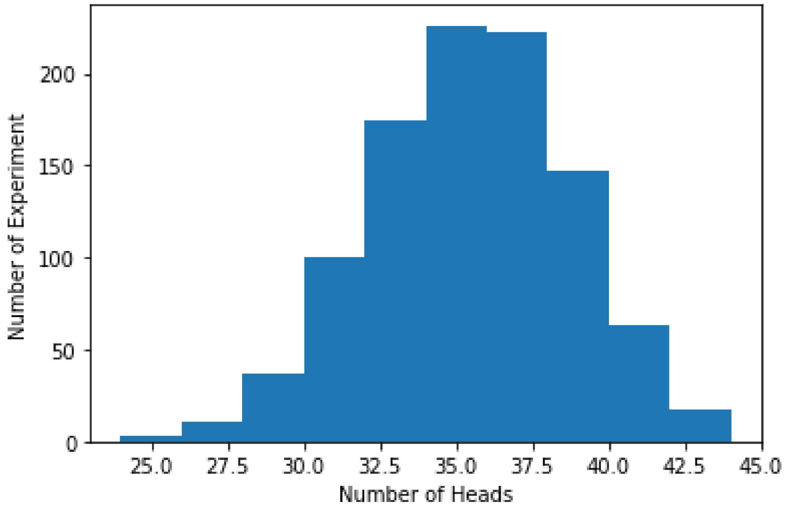


```
#Question 1b for n = 200
N = 50
p = 0.7
number_of_heads = 0
number_of_cont_heads = 0
n = 200
freq = []

for j in range(n):
  for i in range(1, N+1):
    toss = random.random()
    if toss < p:
      number_of_heads += 1
  freq.append(number_of_heads)
  number_of_heads = 0

plt.hist(freq)

plt.ylabel('Number of Experiment')
plt.xlabel('Number of Heads');
```

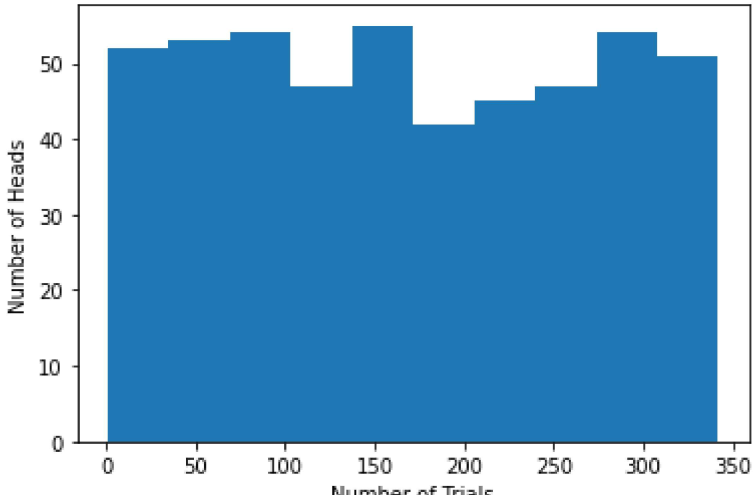


```
#Question 1b for n = 1000
N = 50
p = 0.7
```

```
number_of_heads = 0
number_of_cont_heads = 0
n = 1000
freq = []

for j in range(n):
  for i in range(1, N+1):
    toss = random.random()
    if toss < p:
      number_of_heads += 1
  freq.append(number_of_heads)
  number_of_heads = 0

plt.hist(freq)

plt.ylabel('Number of Experiment')
plt.xlabel('Number of Heads');
```

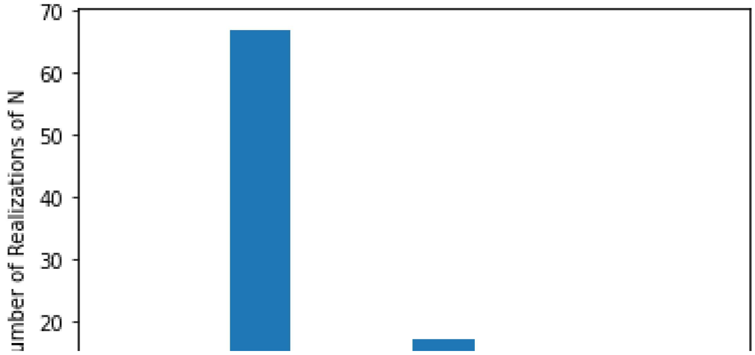


```
#Question 1c

N = 500
p = 0.7
number_of_heads = 0
number_of_cont_heads = 0
freq = []

for i in range(1, N+1):
  toss = random.random()
  if toss < p:
    number_of_heads += 1
  freq.append(number_of_heads)


plt.hist(freq)
plt.ylabel('Number of Heads')
plt.xlabel('Number of Trials');
```



```
import sys
import random
from statistics import mean
import matplotlib.pyplot as plt


#For 100 realizations of N
sum = 0
count = 0
l = []
N = []
E = []

while (len(N)<=100):
  samp = random.uniform(0,1)
  sum = sum + samp
  count = count + 1
  if (sum >= 4):
    l.append(count)
    N.append(min(l))
    sum = 0
    count = 0
E = mean(N)

print('The expected value of N is', E)

plt.hist(N)
plt.xlabel('Value of N')
plt.ylabel('Number of Realizations of N')
```

```
The expected value of N is 6.118811881188119
Text(0, 0.5, 'Number of Realizations of N')
```

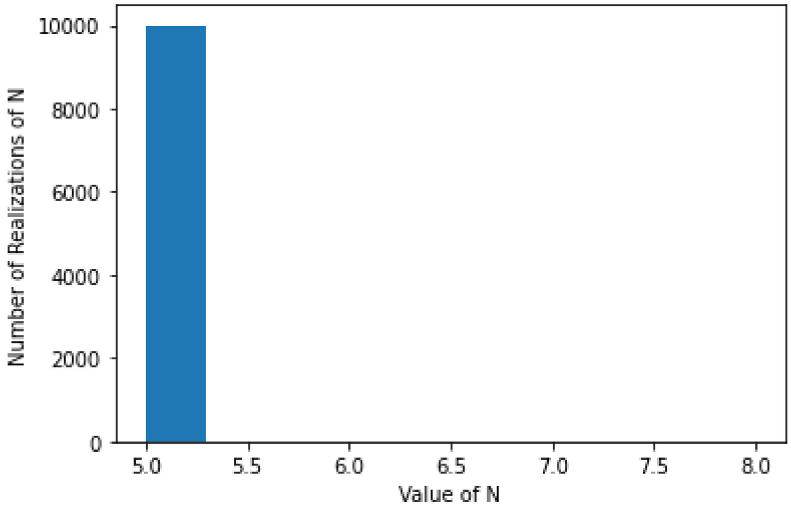


```
#For 1000 realizations of N
sum = 0
count = 0
l = []
N = []
E = []

while (len(N)<=1000):
  samp = random.uniform(0,1)
  sum = sum + samp
  count = count + 1
  if (sum >= 4):
    l.append(count)
    N.append(min(l))
    sum = 0
    count = 0
E = mean(N)

print('The expected value of N is', E)

plt.hist(N)
plt.xlabel('Value of N')
plt.ylabel('Number of Realizations of N')
```

```
The expected value of N is 5.11988011988012
```

```
#For 10000 realizations of N
sum = 0
count = 0
l = []
N = []
E = []

while (len(N)<=10000):
  samp = random.uniform(0,1)
  sum = sum + samp
  count = count + 1
  if (sum >= 4):
    l.append(count)
    N.append(min(l))
    sum = 0
    count = 0
E = mean(N)

print('The expected value of N is', E)

plt.hist(N)
plt.xlabel('Value of N')
plt.ylabel('Number of Realizations of N')
```

```
The expected value of N is 5.002899710028997
Text(0, 0.5, 'Number of Realizations of N')
```


Number of Realizations of N
Value of N