EE 511

PROJECT #1

BY

PRANAV GUNDEWAR

USC ID: 4463612994

EMAIL: gundewar@usc.edu

Q1 Part A – Routine to simulate fair Bernoulli Trial

SUMMARY:

- This program uses random.uniform function which generates series of random number between zeros and ones with uniform distribution. Experiment consists of 100 trials to have fair idea about Bernoulli trial.
- Using basic iterative count, we get the number of heads in each experiment.
- Histogram has been displayed which shows analysis of fair trials.
- Success can be denoted as 1 and failure as 0. Probability of both is 0.5 for fair trial.

$$\begin{array}{ll} If\ Value > 0.5, & a = 1 \\ else & a = 0 \end{array}$$

RESULT AND ANALYSIS:

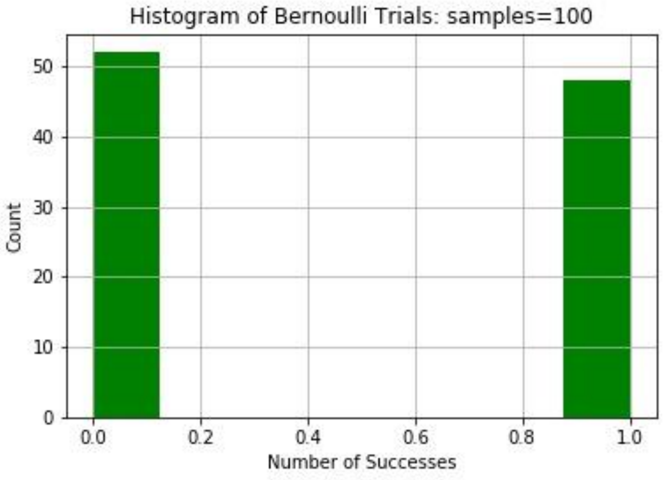
- Histogram of 100 trials has been plotted and 2 bins of values 0 or 1 can be seen. 1 represents success and 0 failure. Hence, we can get number of successes of histogram.
- Success is almost equal to failure as probability is 0.5 which can be inferred from histogram.

CODE:

```
"""

@author: Pranav Gundewar

@course; EE 511 Simulation Methods of Stochastic Systems

Project 1

Q1 Part A

Simulation of fair Bernoulli trial

"""

# Importing Libraries

import numpy as np

import random

import matplotlib.pyplot as plt


# Initializing Variables

b = np.empty([100]);
```

```
h = 0
for i in range(100):
    a = random.uniform(0,1)                    # Generating random numbers between 0 to 1 with uniform districution
    if (a > 0.5):                         # Chechking for success
        h = h + 1
        b[i] = 1
    else:
        b[i] = 0
print("Number of heads: ",h)
print("Number of tails: ",100-h)
# Plotting the histogram
plt.hist(b, bins = 'auto', facecolor='green')
plt.xlabel('Bernoulli Trials')
plt.ylabel('Count')
plt.title('Histogram of Bernoulli Trials: samples=100')
plt.grid(True)
plt.savefig('Q1a.jpeg')
plt.show()
```

OUTPUT:



Q1 Part B – Routine to count number of successes

SUMMARY:

- This program uses random.uniform function which generates series of random number between zeros and ones with uniform distribution. Experiment consists of 100 samples of 7 trials each.
- Using basic iterative count, we get the number of successes in each sample.
- Histogram has been displayed which shows analysis of number of successes.
- Success can be denoted as 1 and failure as 0. Probability of both is 0.5 for fair trial.

RESULT AND ANALYSIS:

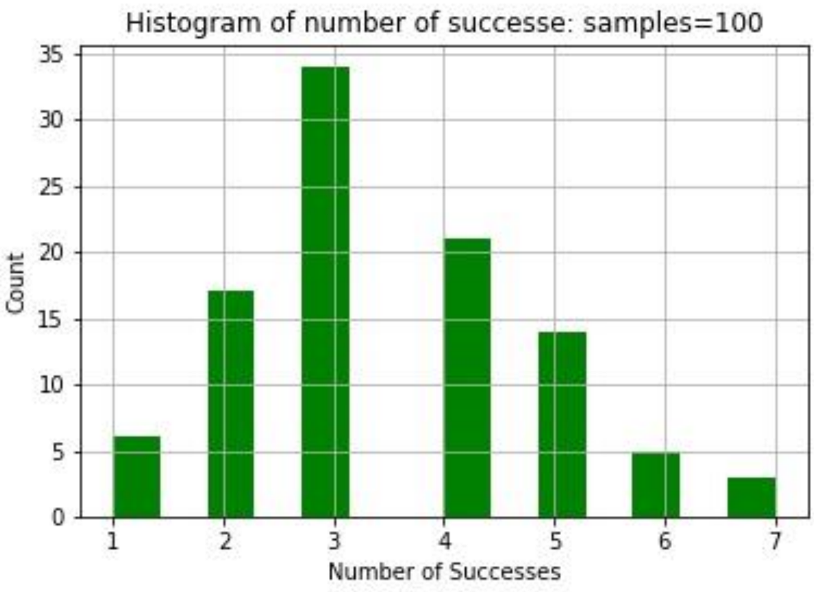
- Histogram of 100 trials has been plotted and number of successes each sample can be seen. We can get information about how many times success occurred.
- Histograms shows random variable has Gaussian distribution since it has bell shaped curve. Histogram has '3' successes as centre point.
- Histogram is sum of IID samples hence according to Central Limit Theorem(CLT), it has normal distribution.

CODE:

```
# -*- coding: utf-8 -*-

"""

@author: Pranav Gundewar

@course; EE 511 Simulation Methods of Stochastic Systems

Project 1

Q1 Part B

Routine to count number of successes

"""


# Importing Libraries

import numpy as np

import random

import matplotlib.pyplot as plt
```

```
# Initializing Variables
h = np.zeros([100])
for i in range(100):
    for j in range(7):
        a = random.uniform(0,1)                  # Generating random numbers between 0 to 1 with uniform districution
        if (a > 0.5):                      # Chechking for success
            h[i] = h[i]+1
# Plotting the histogram
plt.hist(h, bins = 'auto', facecolor='green')
plt.xlabel('Number of Successes')
plt.ylabel('Count')
plt.title('Histogram of number of successe: samples=100')
plt.grid(True)
plt.savefig('Q1b.jpeg')
plt.show()
```

OUTPUT:



Q1 Part C – Routine to count longest run of heads

SUMMARY:

- This program uses random.uniform function which generates series of random number between zeros and ones with uniform distribution. Experiment consists of 100 samples of 7 trials each. For each trial, longest run of heads in counted and stored in variable.
- Using basic iterative count, we get the longest run of heads per sample. Hence, a variable will store longest run of heads for 100 samples.
- Histogram has been displayed which shows analysis of longest run of heads.

RESULT AND ANALYSIS:

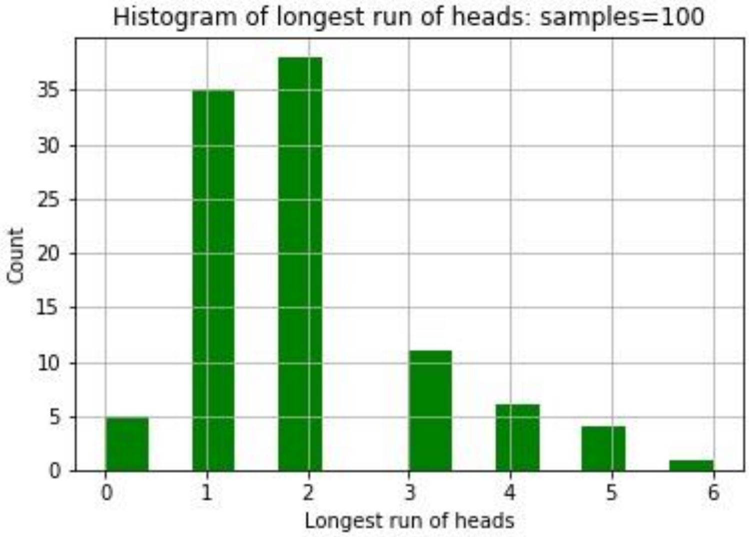
- Histogram of 100 samples has been plotted and the longest run of heads per sample can be seen..
- Histograms indicates that longest run of 2 heads has been occurred more times than other run of heads.
- Histogram is sum of IID samples and shows Poisson distribution.

CODE:

```
# -*- coding: utf-8 -*-
"""
@author: Pranav Gundewar
@course; EE 511 Simulation Methods of Stochastic Systems
Project 1
Q1 Part C
Routine to count longest run of heads
"""


# Importing Libraries
import numpy as np
import random
import matplotlib.pyplot as plt
```

```
# Initializing Variables

lh = np.zeros([100])                    # Longest run of heads
h = np.zeros([100])

# Checking to count longest run of heads
for i in range(100):
    for j in range(7):
        a = random.uniform(0,1)              # Generating random numbers between 0 to 1 with
uniform distribution
        if (a > 0.5):                   # Checking for success
            h[i] = h[i]+1
        else:
            if (lh[i]<h[i]):
                lh[i] = h[i]
            h[i] = 0

# Plotting the histogram
plt.hist(lh, bins = 'auto', facecolor='green')
plt.xlabel('Longest run of heads')
plt.ylabel('Count')
plt.title('Histogram of longest run of heads: samples=100')
plt.grid(True)
plt.savefig('Q1c.jpeg')
plt.show()
```

OUTPUT:



Q2 – Simulate Bernoulli random variable for 300 samples of k trials each

SUMMARY:

- This program uses random.uniform function which generates series of random number between zeros and ones with uniform distribution. Experiment consists of 300 sums of 7 trials each. For each trial, we get number of successes in each sample.
- Using basic iterative count, we get the number of successes per sample. Hence, a variable will store number of successes for 300 samples.
- Histogram has been displayed which shows analysis of number of successes

RESULT AND ANALYSIS:

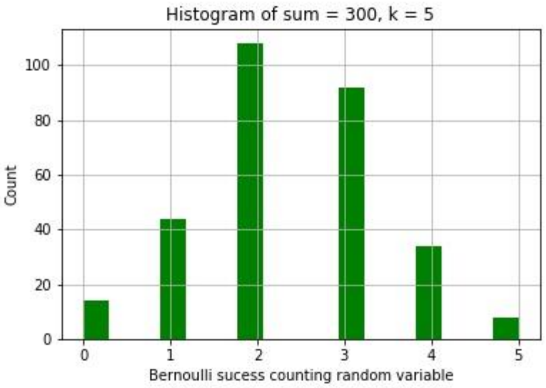
- Histogram of 100 samples has been plotted and the number of successes per sample can be seen.
- For different values k trials, histograms have been plotted. As number of trials are increased, we can see bell shape curve more prominently.
- Histograms shows random variable has Gaussian distribution since it has bell shaped curve. Hence Bernoulli success random variable has normal distribution with more prominent bell shape as samples increases.

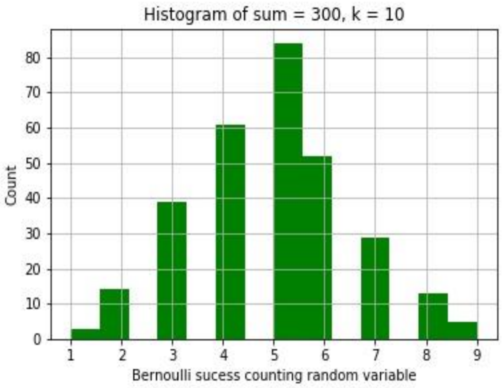
CODE:

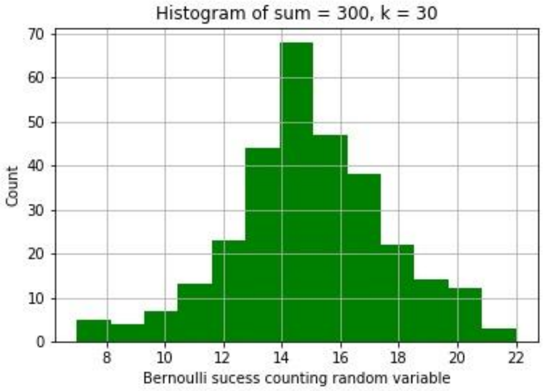
```
# -*- coding: utf-8 -*-

"""

@author: Pranav Gundewar
```

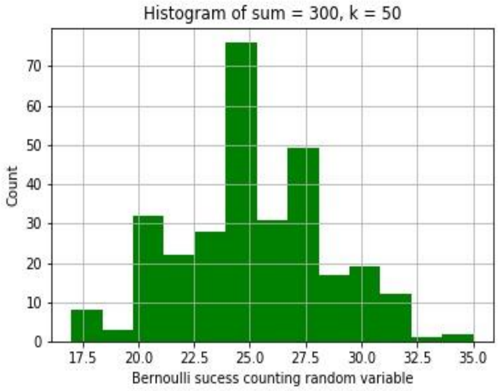
```
@course; EE 511 Simulation Methods of Stochastic Systems

Project 1

Q2- Counting Successes

Routine to find random number of edges selected

"""

# Importing Libraries

import numpy as np

import random

import matplotlib.pyplot as plt


# Initializing Variables

sum = 300                                   # Number of sum

h = np.zeros([sum])

k = 5                                       # Number of samples


# Checking sucess counting random variable

for i in range(sum):

    for j in range(k):

        a = random.uniform(0,1)             # Generating random numbers between 0 to 1
with uniform districution

        if (a > 0.5):                       # Chechking for success

            h[i] = h[i] + 1;


# Plotting the histogram

plt.hist(h, bins = 'auto', facecolor='green')

plt.xlabel('Bernoulli sucess counting random variable')

plt.ylabel('Count')
plt.title('Histogram of sum = 300, k = 5')

plt.grid(True)

plt.savefig('Q2(k=5).jpeg')

plt.show()
```

OUTPUT:









Q3 – Networking Part 1

SUMMARY:

- 20 people exist in group and number of possible edges are to be calculated. Also, for given probability of 0.05 routine to find possible number of edges is to be written.
- This program uses random.uniform function which generates series of random number between zeros and ones with uniform distribution. Experiment consists of 100 samples of 180 trials each. For each trial, we get number of possible edges.
- Histogram has been displayed which shows analysis of number of possible edges.

RESULT AND ANALYSIS:

- For n=20 people, maximum number of edges are (n*(n-1))/2 which is 190 edges as it is sum of first ‘n’ numbers.
- After finding maximum edges, a routine is built to find possible edges with probability of 0.05.
- Histogram of 100 samples has been plotted and the number of possible edges per sample can be seen.
- We can observe curve has centre about 10 edges per sample.
- Histograms shows random variable has Gaussian distribution since it has bell shaped curve. Hence Bernoulli success random variable has normal distribution with more prominent bell shape as samples increases.

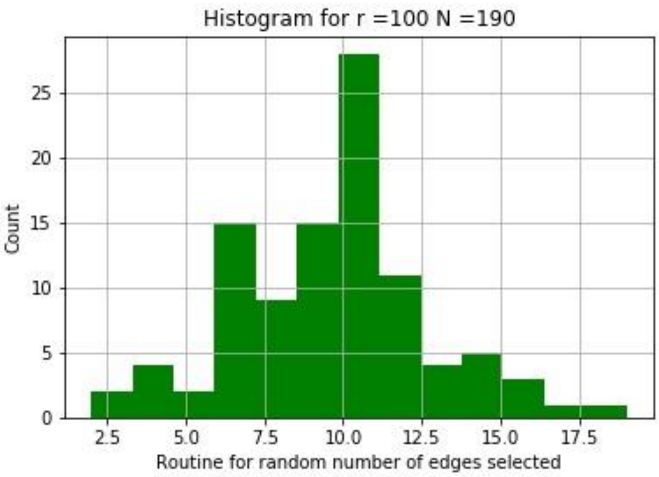
CODE:

```
# -*- coding: utf-8 -*-

"""

@author: Pranav Gundewar

@course; EE 511 Simulation Methods of Stochastic Systems

Project 1

Q3- Networking Part 1

Routine to find random number of edges selected

"""

# Importing Libraries

import numpy as np

import random

import matplotlib.pyplot as plt


# Initializing Variables

n = 20                                  # Number of people in group

N = (int) (n*(n-1)/2)

r = 100                                 # Number of Samples

print('Total number of possible edges: ',N)

h = np.zeros([r])                       # Initializing an array for number of edges
```

```
# Checking number of edges for r samples of N trials
for i in range(r):
    for j in range(N):
        a = random.uniform(0,1)                    # Generating random numbers between 0 to 1 with uniform districution
        if (a < 0.05):                             # Chechking if p<0.05 to select an edge
            h[i] = h[i] + 1;
# Plotting the histogram
plt.hist(h, bins = 'auto', facecolor='green')
plt.xlabel('Routine for random number of edges selected')
plt.ylabel('Count')
plt.title('Histogram for r =100 N =190')
plt.grid(True)
plt.savefig('Q3.jpeg')
plt.show()
```

OUTPUT:

```
Total number of possible edges:  190
```


Histogram for r =100 N =190
Count
Routine for random number of edges selected