Weining Bao weiningb@usc.edu 8287723653

Project 2

1. Generate a large sequence of random numbers using the rand function in Matlab.

i) Compute basic statistics to test uniformity
- Calculate mean and variance; compare to expected.
- Can you make a statistical statement about these results?

ii) Independence Tests
- Compute $\frac{2}{N}\sum_{i=1}^{N/2} X_{2i-1}X_{2i}$. N is even. If adjacent numbers in the sequence are independent, what should this be?

iii) Uniform Distribution
Use the data to generate a uniformly distributed discrete random variable that take on values from 0,1,2,…,M and generate a histogram of the results for some number of runs. How well does this compare to what you expect? Can you make a statistical statement about this?

iv) N coins are flipped. Let H (a random variable) be the number of heads. H has a binomial distribution, i.e., $P(H=h)=\begin{pmatrix} N \\ h \end{pmatrix}(\frac{1}{2})^N$. Use the data to generate a random variable that has a binomial distribution, for N = 10. Show the distribution for H is a close match to the analytical result.

Matlab Simulation

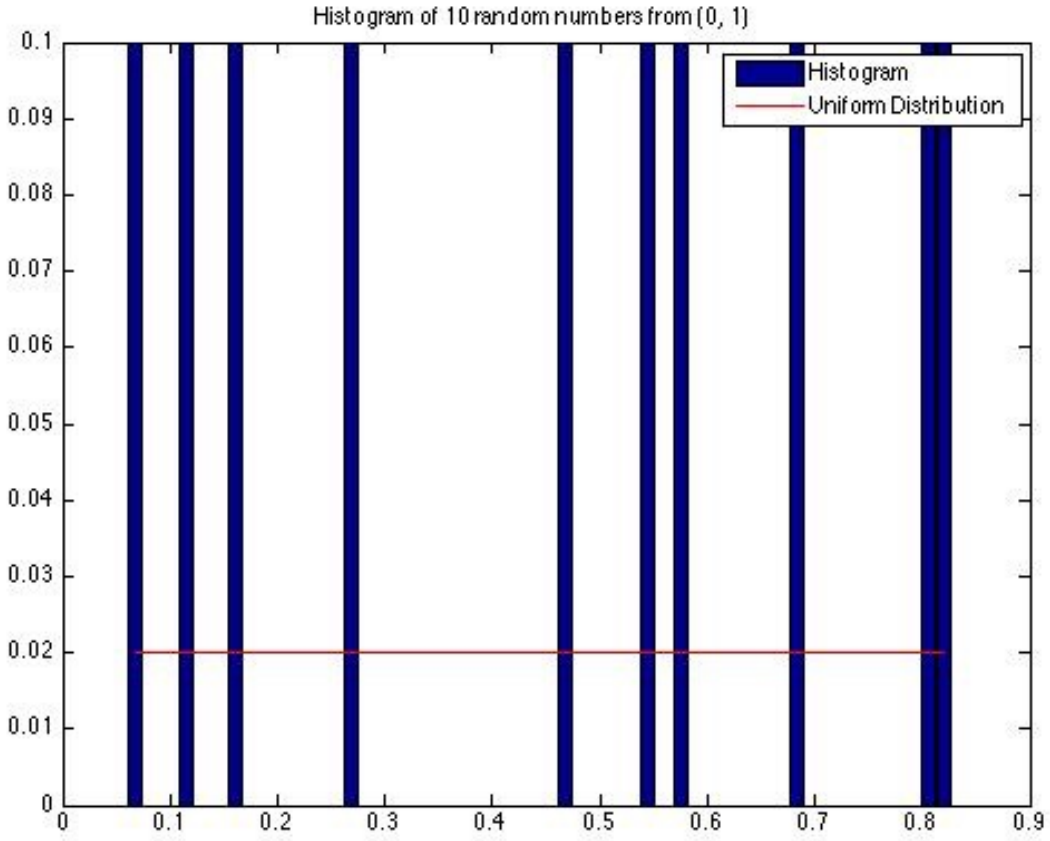
Code at last

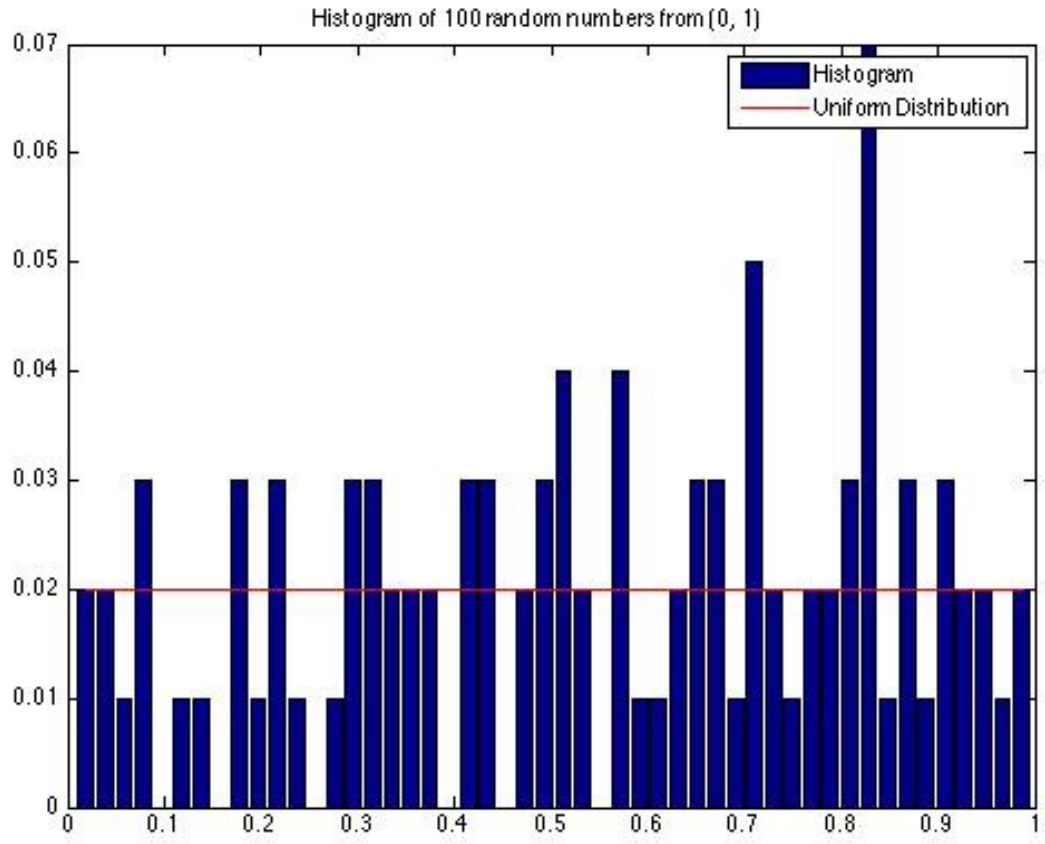
i)

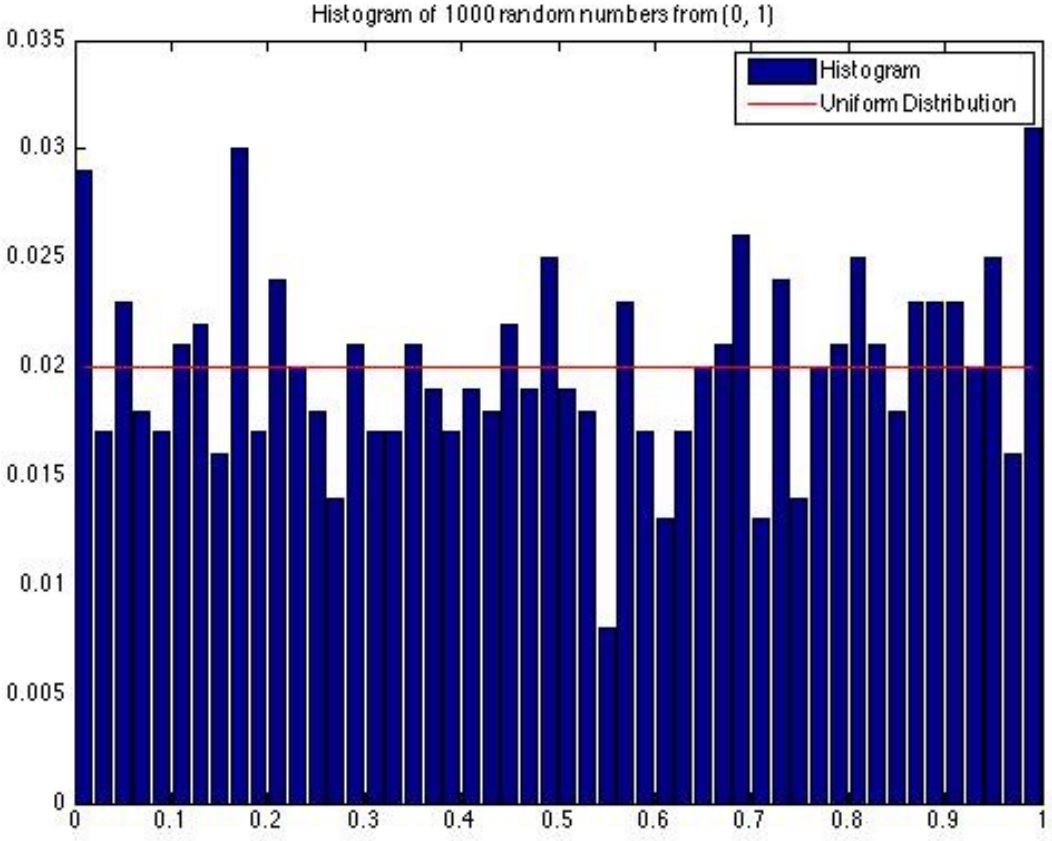
Ideally, random number from (0, 1) should have uniform distribution between 0 and 1. If the output random number is considered a random variable X, then it is continuous one with probability distribution function: $f_X(x)=\begin{cases} 1 & 0<x<1 \\ 0 & otherwise \end{cases}$ and cumulative probability function: $F_X(x)=\begin{cases} 1 & x\geq 1 \\ x & 0<x<1 \\ 0 & x\leq 0 \end{cases}$. So the mean or expectation of this random variable is $E(X)=\int_{-\infty}^{\infty} x f_X(x)dx=\int_0^1 x\,dx=\frac{1}{2}$, variance is $\sigma^2=E(X^2)-[E(X)]^2=\frac{1}{12}$ where $E(X^2)=\int_{-\infty}^{\infty} x^2 f_X(x)dx=\int_0^1 x^2\,dx=\frac{1}{3}$.

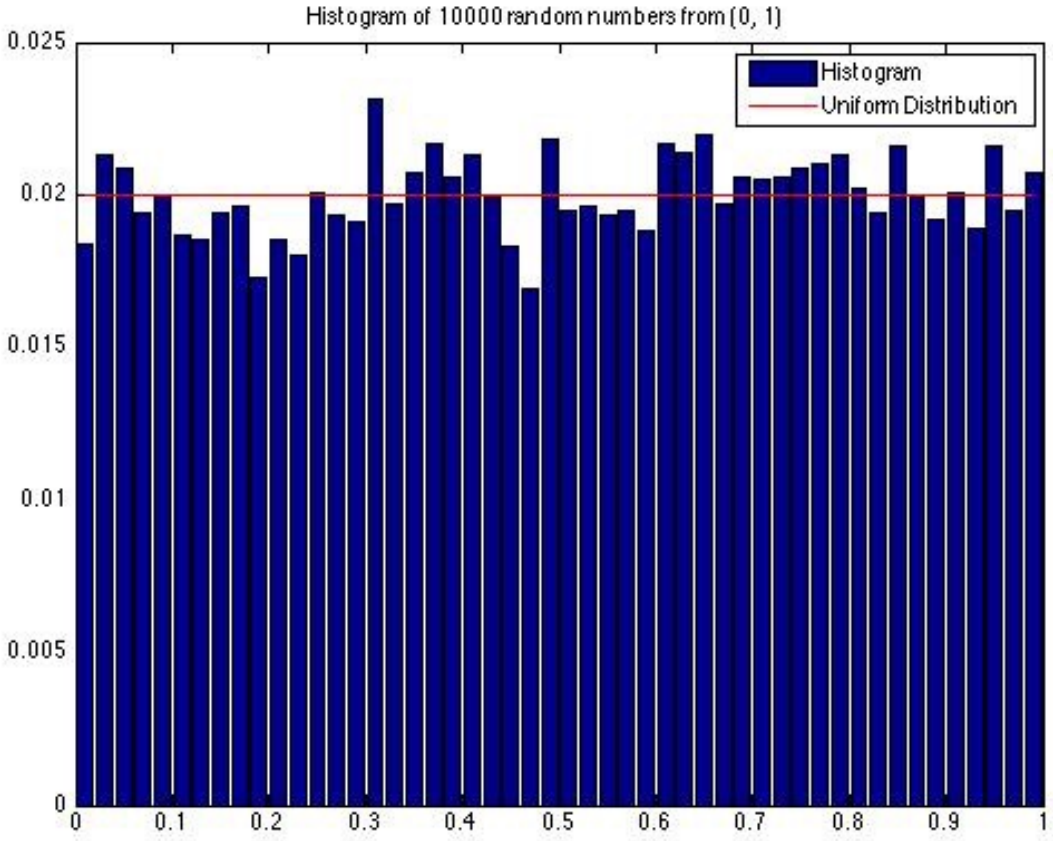
So in this part, the Matlab implementation generates 10, 100, 1000, 10000 and 100000 random numbers sequence and calculates the means and variances of these 5 sequences. At last, histograms of these sequences are plotted with 50 bins. Because the relative frequency of events' occurrence will approach their probability as the number of experiments increases, the means, variances are closer to the theoretical value and histograms tend to be uniform. Here are the results and normalized histograms:

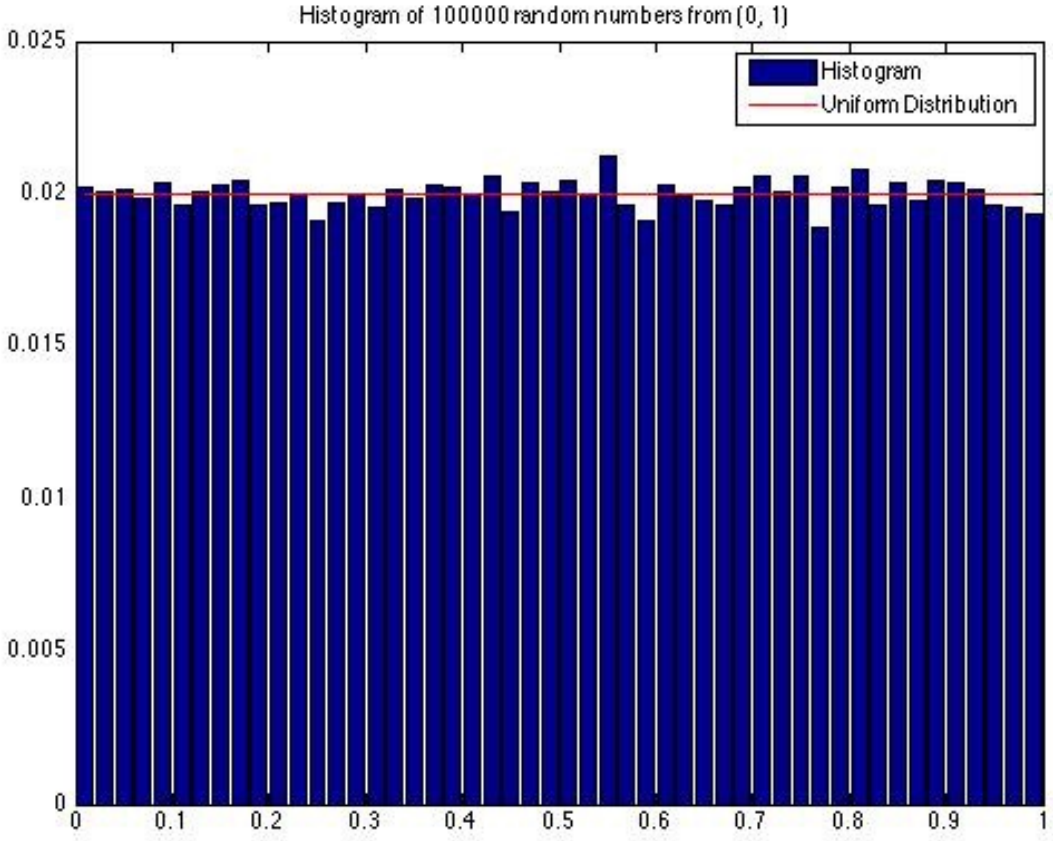
Sequence of 10 random numbers from (0, 1).
Mean: 0.450294. Variance: 0.082146.
Sequence of 100 random numbers from (0, 1).
Mean: 0.548621. Variance: 0.075848.
Sequence of 1000 random numbers from (0, 1).
Mean: 0.505861. Variance: 0.088878.
Sequence of 10000 random numbers from (0, 1).
Mean: 0.504763. Variance: 0.082909.
Sequence of 100000 random numbers from (0, 1).
Mean: 0.499948. Variance: 0.083201.




Histogram of 100 random numbers from (0, 1)
Histogram
Uniform Distribution
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Histogram of 1000 random numbers from (0, 1)
Histogram
Uniform Distribution
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Histogram of 10000 random numbers from (0, 1)
Histogram
Uniform Distribution
0.025
0.02
0.015
0.01
0.005
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Histogram of 100000 random numbers from (0, 1)
Histogram
Uniform Distribution
0.025
0.02
0.015
0.01
0.005
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

So it is reasonable to say that the random numbers from generator rand() in Matlab are in uniform distribution, which is proved by the above simulations.

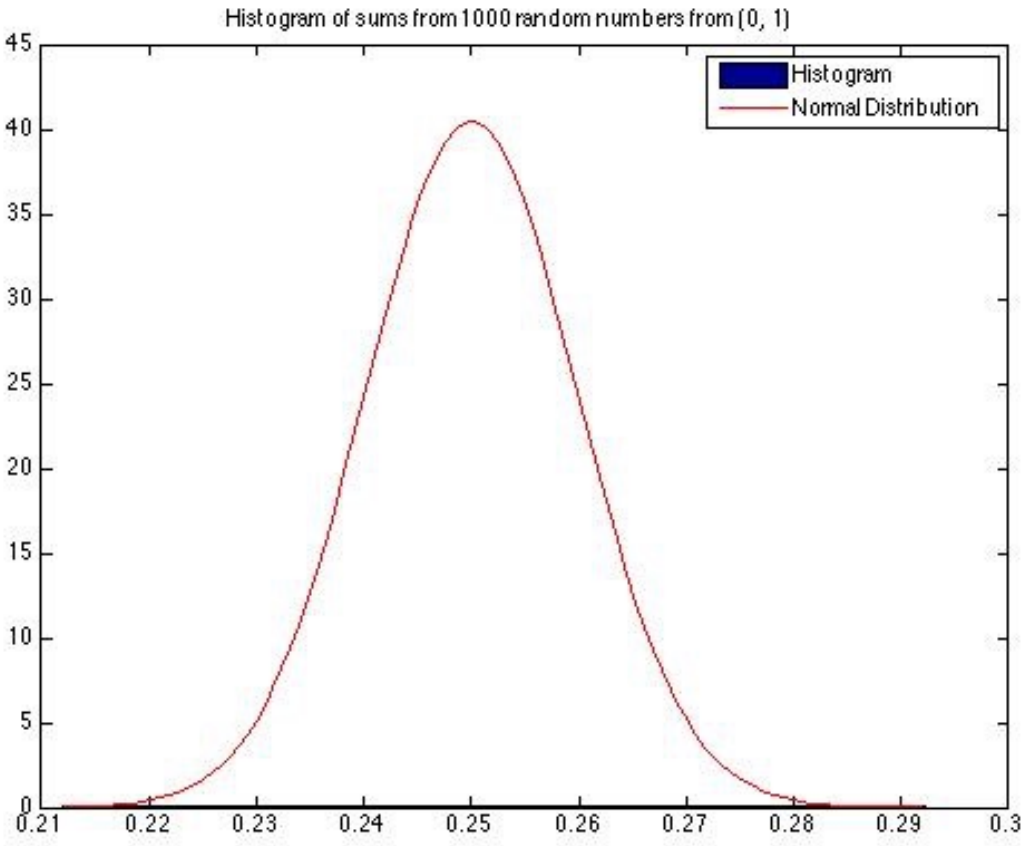
ii)

$\frac{2}{N}\sum_{i=1}^{N/2} X_{2i-1}X_{2i} = \frac{2}{N}(X_1X_2 + X_3X_4 + ...)$. Because adjacent numbers in the sequence are independent, so each $X_{2i-1}X_{2i}$ is independent too. If X_{2i-1}, X_{2i} are represented by two independent identically distributed (IID) random variables X, Y, they both have uniform distribution $f_X(x) = \begin{cases} 1 & 0 < x < 1 \\ 0 & otherwise \end{cases}$ among (0, 1). If their product is represented by a random variable Z, the cumulative probability function $F_Z(z) = \int_{z=xy} f_{XY}(x,y)\,dx\,dy = \int_{z=xy} f_X(x) f_Y(y)\,dx\,dy$ because X, Y are independent too. After further calculation, $F_Z(z) = \begin{cases} 0 & z \le 0 \\ 1 & z \ge 1 \end{cases}$, when $0 < z < 1$, $F_Z(z) = \int_{x=0}^{x=z}\int_{y=0}^{y=1} dx\,dy + \int_{x=z}^{x=1}\int_{y=0}^{y=\frac{z}{x}} dx\,dy = z(1 - \ln z)$, so $f_Z(z) = \frac{dF_Z(z)}{dz} = \begin{cases} -\ln z & 0 < z < 1 \\ 0 & otherwise \end{cases}$. So the expectation of Z, $E(Z) = -\int_0^1 z \ln z\,dz = \frac{1}{4}$, variance is $\sigma^2 = E(Z^2) - [E(Z)]^2 = \frac{1}{9} - \frac{1}{16} = \frac{7}{144}$.

In this way, $\frac{2}{N}\sum_{i=1}^{N/2} X_{2i-1}X_{2i} = \frac{2}{N}\sum_{i=1}^{N/2} Z_i$ while Z_i are all IID random variables with mean = 1/4 and variance = 7/144. According to Central Limit Theorem, $\frac{\sum_{i=1}^{n} Z_i}{n} \sim N(\mu, \frac{\sigma^2}{n})$ if n is large enough. In the Matlab implementation, 100000 random numbers are generated for this part, so n = 100000 / 2 = 50000. In this case, $\frac{\sum_{i=1}^{50000} Z_i}{50000} \sim N(\frac{1}{4}, \frac{7/144}{50000})$. Similarly, 10000 and 1000 random numbers sequences are also tried. The mean is unchanged while the variances are $\frac{7}{144 * 5000}$ and $\frac{7}{144 * 500}$, respectively. 10000 runs are tried for each sequence and here are the results:

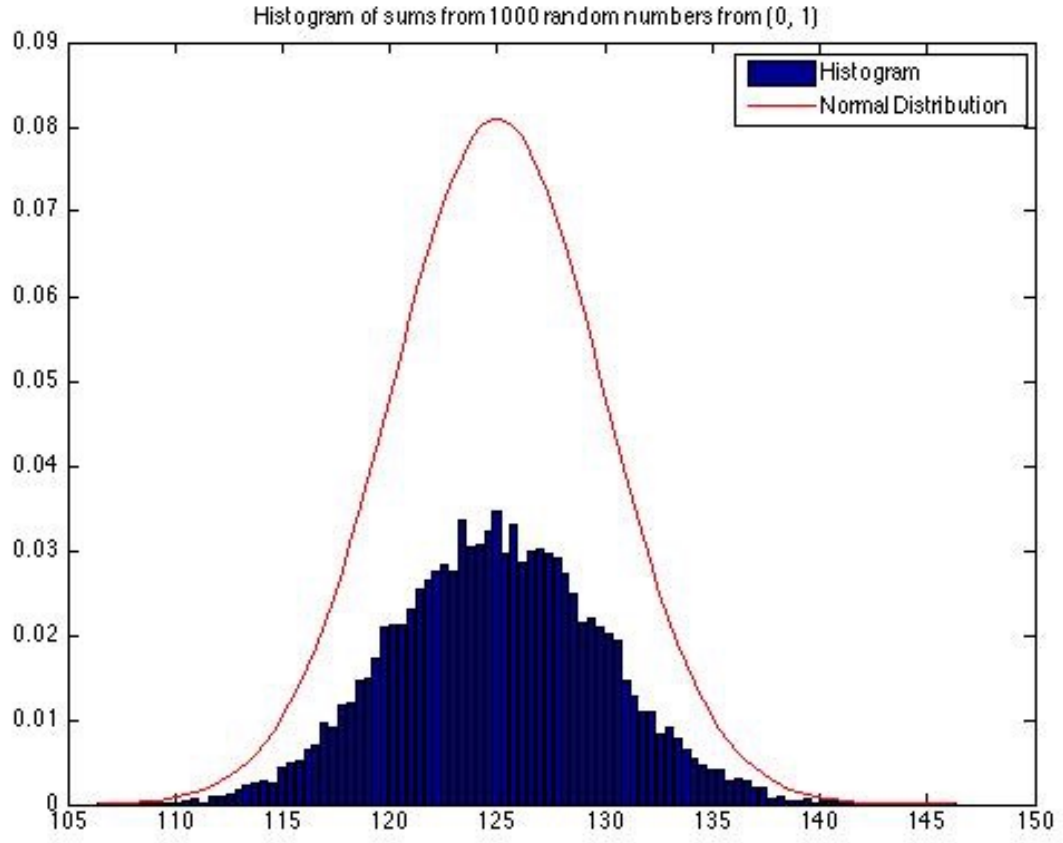
Sums from sequence of 1000 random numbers from (0, 1).
Mean: 0.250082. Variance: 0.000098.
Sums from sequence of 10000 random numbers from (0, 1).
Mean: 0.250018. Variance: 0.000010.

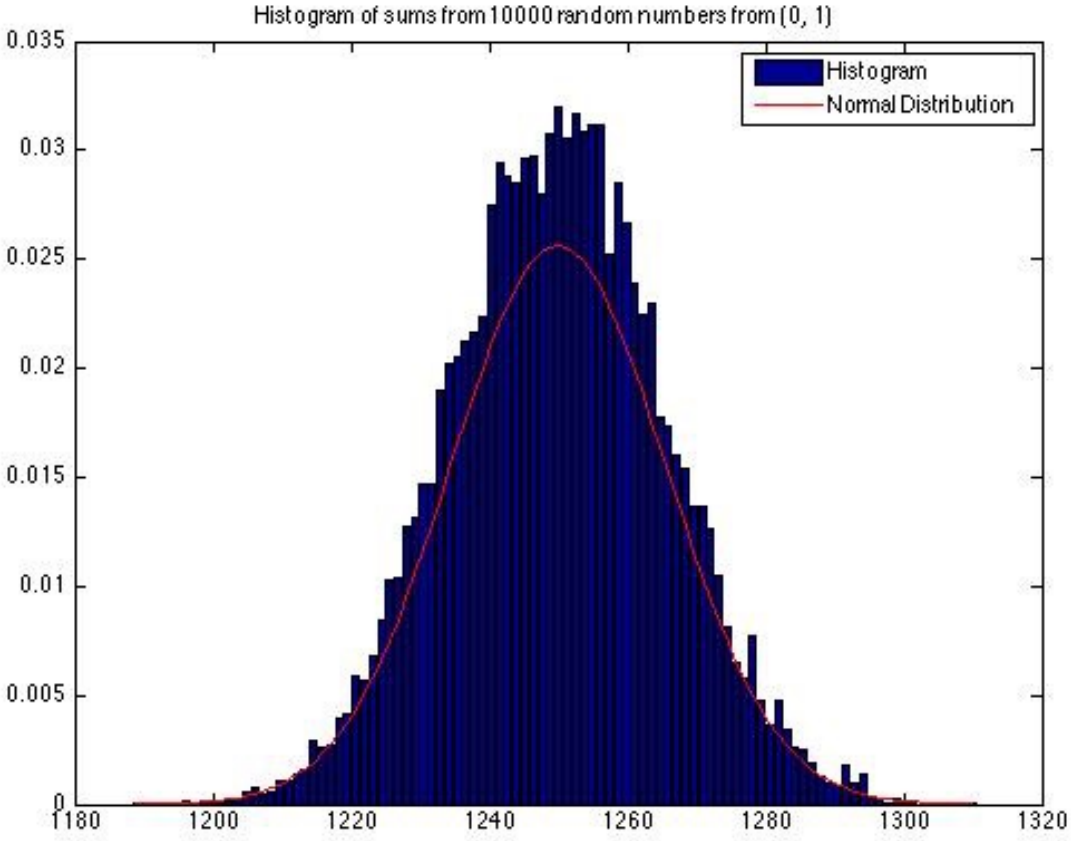
Sums from sequence of 100000 random numbers from (0, 1).
Mean: 0.249993. Variance: 0.000001.
The means and variances are very close to the theoretical values. However the histograms and normal distribution curves are not close, for example:

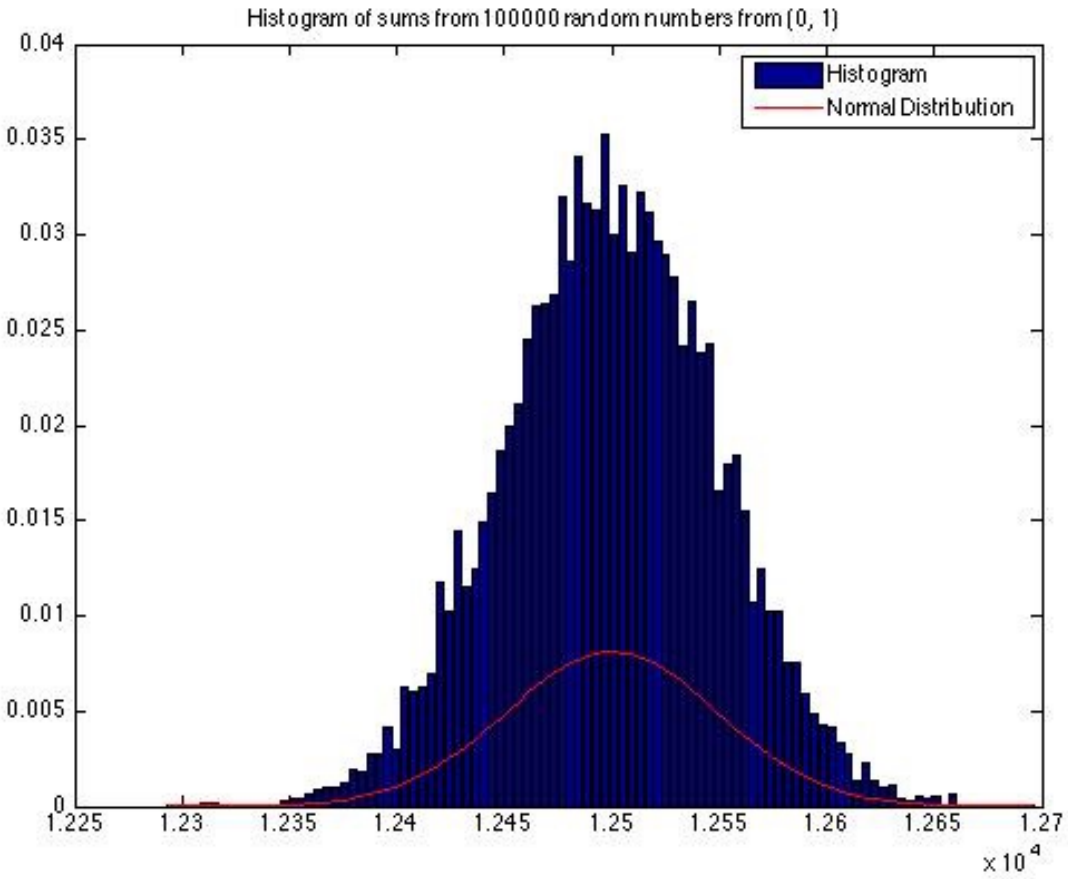


There is some problems with normal distribution curve whose values even surpass 1. I think it is because that the variance or standard deviation is too small so the variable type error in the exponential function in Matlab brings this kind of giant error. So another sum is calculated instead: $\sum_{i=1}^{n} Z_i \sim N(n\mu, n\sigma^2)$. For the 1000, 10000 and 100000 random numbers sequences, the normal distributions have mean 0.125 * 1000 = 125, 1250 and 12500 respectively. Similarly, variance of 7 * 1000 / 288 = 24.31, 243.1 and 2431 should be observed respectively. These are all got from the results:

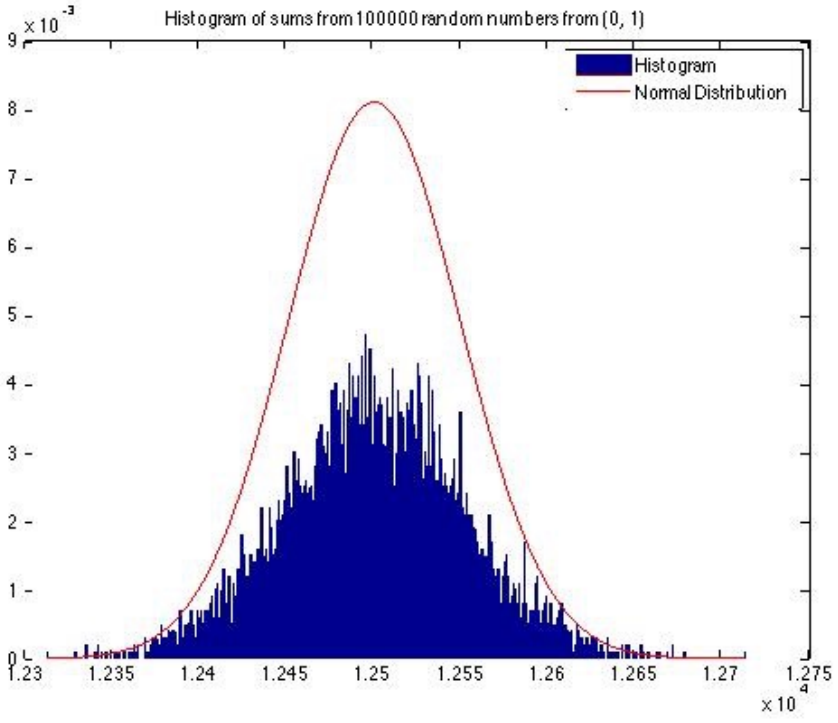
Sums from sequence of 1000 random numbers from (0, 1).
Mean: 125.008256. Variance: 24.502938.
Sums from sequence of 10000 random numbers from (0, 1).
Mean: 1249.949014. Variance: 241.547905.
Sums from sequence of 100000 random numbers from (0, 1).
Mean: 12499.505210. Variance: 2443.008077.


Histogram of sums from 1000 random numbers from (0, 1)
Histogram
Normal Distribution
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
105
110
115
120
125
130
135
140
145
150


Histogram of sums from 10000 random numbers from (0, 1)
Histogram
Normal Distribution
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
1180
1200
1220
1240
1260
1280
1300
1320



The histograms and corresponding normal distribution curves follow:



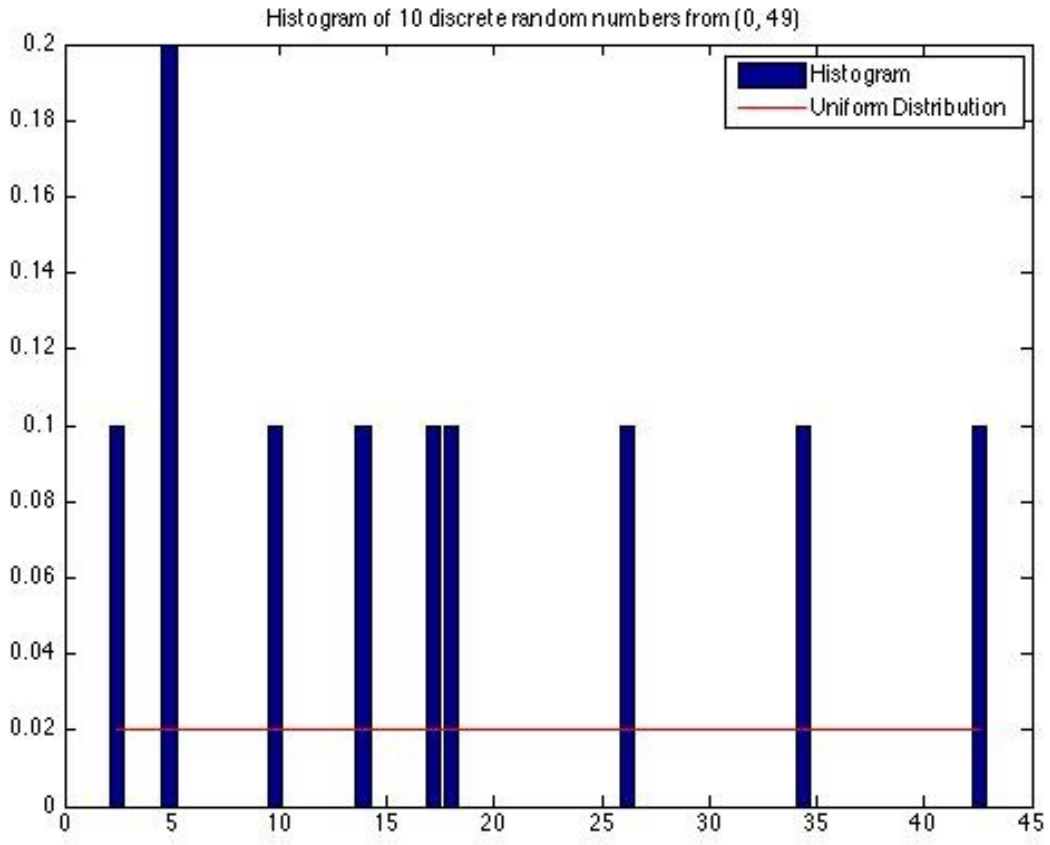
So the histograms and normal distribution curves fit well. Here 100 bins are used. Different number of bins will lead to different distance between histogram and curve. For example, on the left is the plot with 1000 bins:
With larger number of bins, the normalized values of histogram become smaller which

increases the distance. Anyway, the shape of histograms and corresponding normal distribution curves are close. It is reasonable to say that the sum as a random variable can be estimated by a normal distribution according to Central Limit Theorem. This proves the assumption that adjacent numbers in the sequence are independent. So the part one and two show that the random number generator rand() in Matlab is good which gives independent random number with uniform distribution.

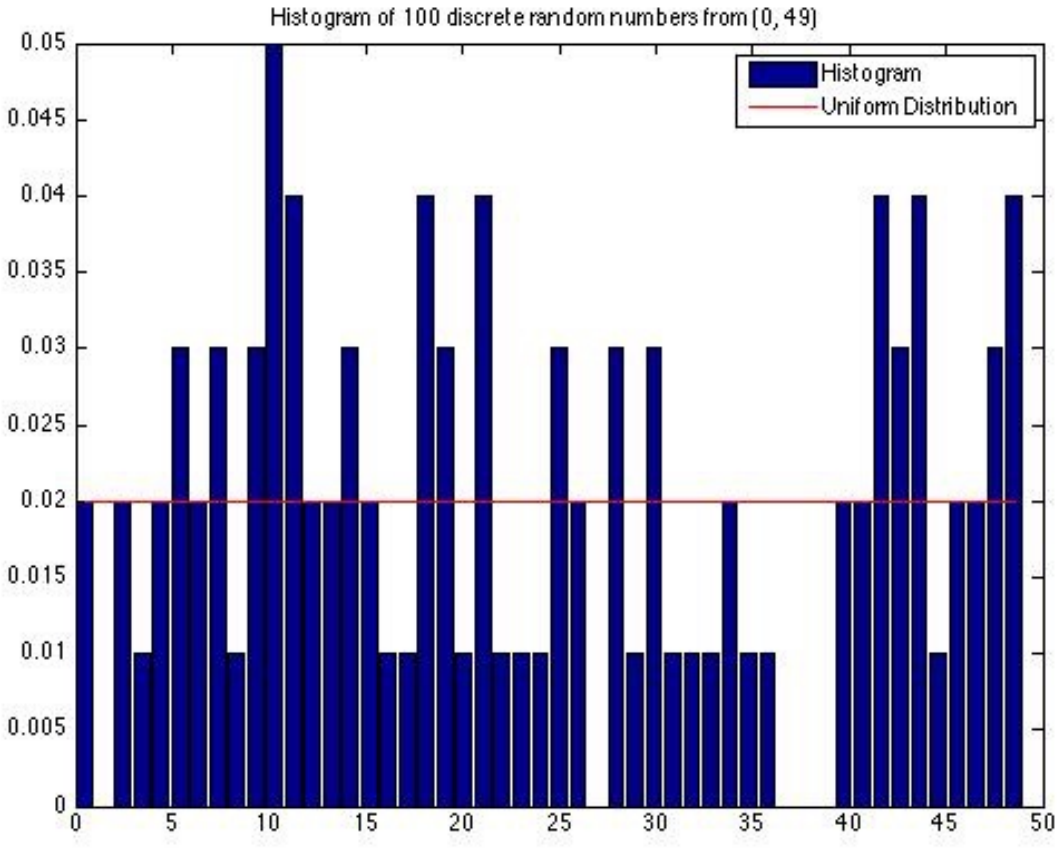
iii)

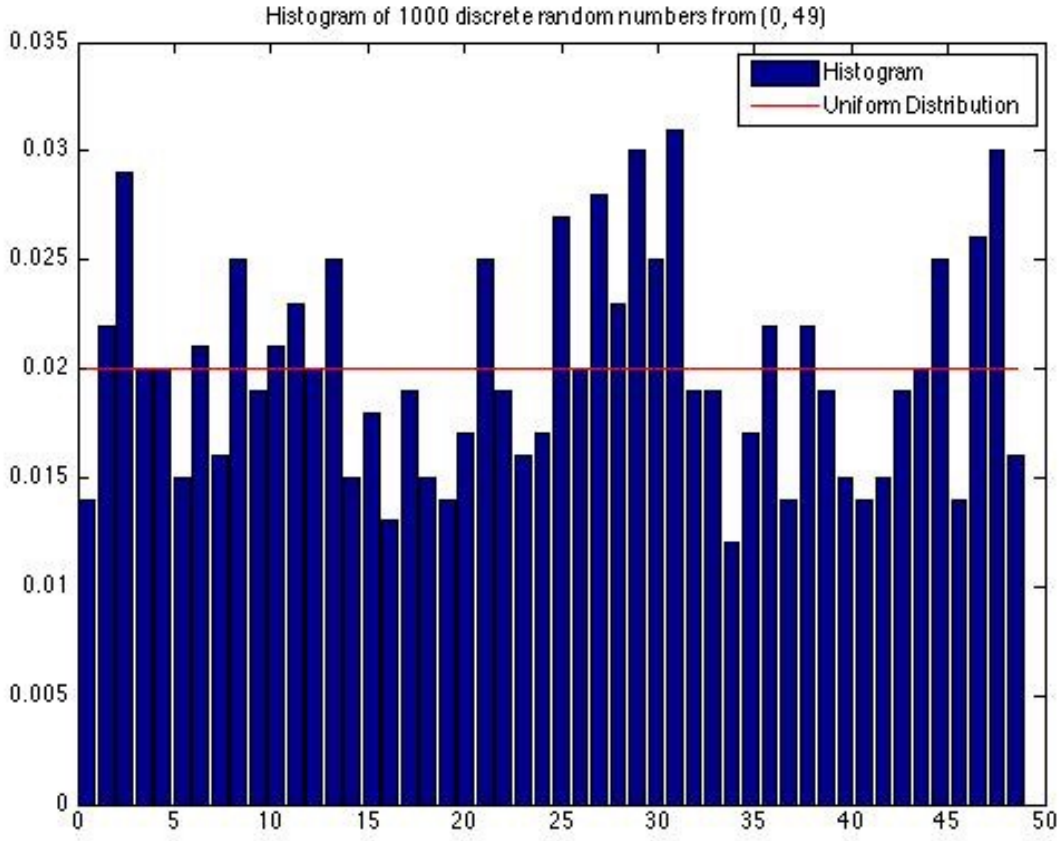
Use the data to generate a uniformly distributed discrete random variable that take on values from 0,1,2,…,M and generate a histogram of the results for some number of runs. How well does this compare to what you expect? Can you make a statistical statement about this?

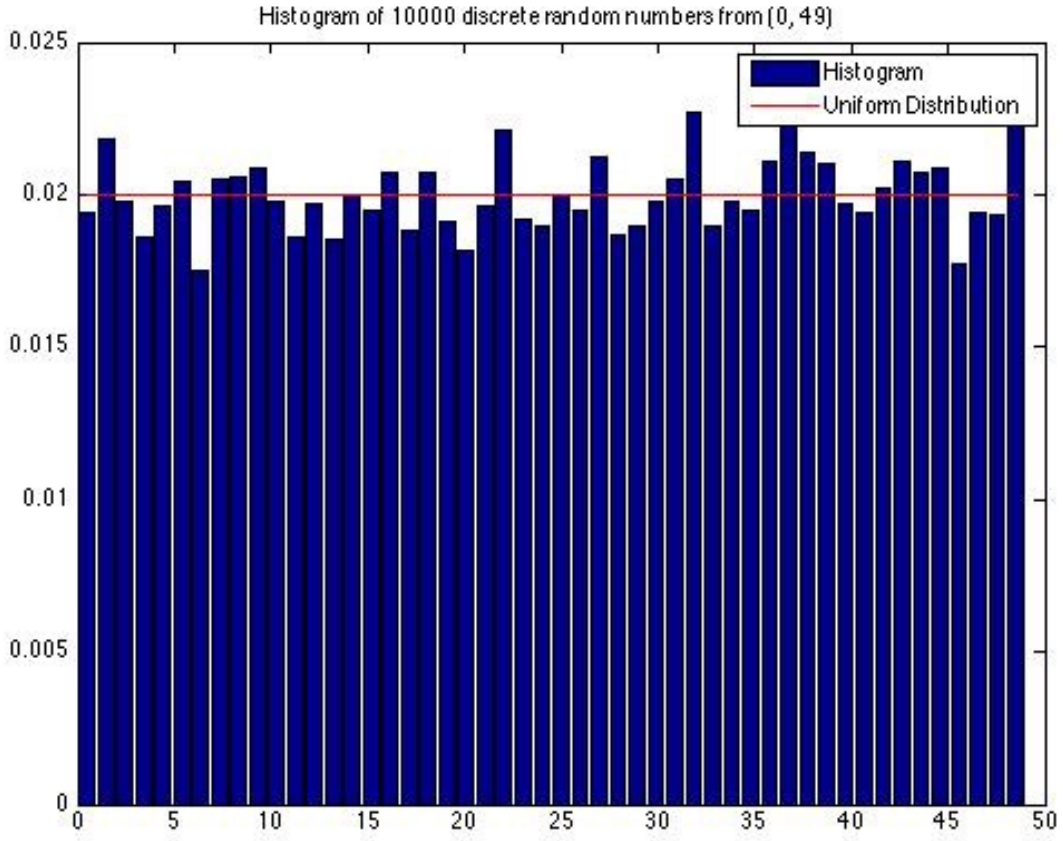
Discrete random variable X is generated from the random numbers sequence produced in part 1. The random variable X takes value 0, 1, 2… 49 because the histogram uses 50 bins. It was proved that the random number values among (0, 1) generated is of uniform distribution, so ideally X should have probability mass function $p_X(x) = \begin{cases} 1/50 & x = 0,1,2...49 \\ 0 & otherwise \end{cases}$. In this way, the expectation is 49 / 2 = 24.5, variance is 49 * 50 * 99 / (6 *50) - 24.5^2 = 208.25. 10, 100, 1000, 10000 and 100000 random numbers sequences are still generated here and here are the results:

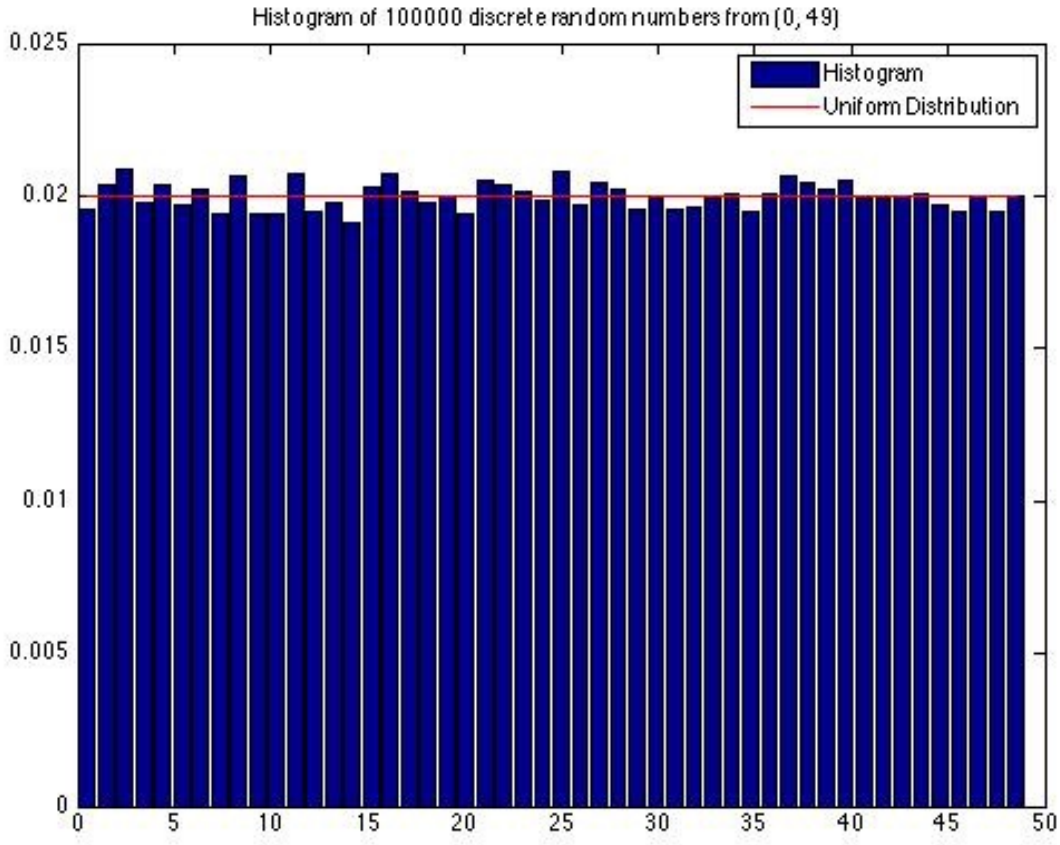


Random number (0, 49) from sequence of 10 random numbers.
Mean: 17.400000. Variance: 179.600000.
Random number (0, 49) from sequence of 100 random numbers.
Mean: 24.200000. Variance: 227.656566.
Random number (0, 49) from sequence of 1000 random numbers.
Mean: 24.555000. Variance: 208.553529.
Random number (0, 49) from sequence of 10000 random numbers.
Mean: 24.687300. Variance: 208.817600.
Random number (0, 49) from sequence of 100000 random numbers.
Mean: 24.481610. Variance: 208.014082.




Histogram of 1000 discrete random numbers from (0, 49)
Histogram
Uniform Distribution
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
20
25
30
35
40
45
50


Histogram of 10000 discrete random numbers from (0, 49)
Histogram
Uniform Distribution
0.025
0.02
0.015
0.01
0.005
0
0
5
10
15
20
25
30
35
40
45
50



Similar to part 1, the relative frequency of events' occurrence will approach their probability as the number of experiments increases, the means, variances are closer to the theoretical value and histograms tend to be uniform. So it is reasonable to say that the random variable X generated from rand() sequence in Matlab are in uniform distribution, which is proved by the above simulations.

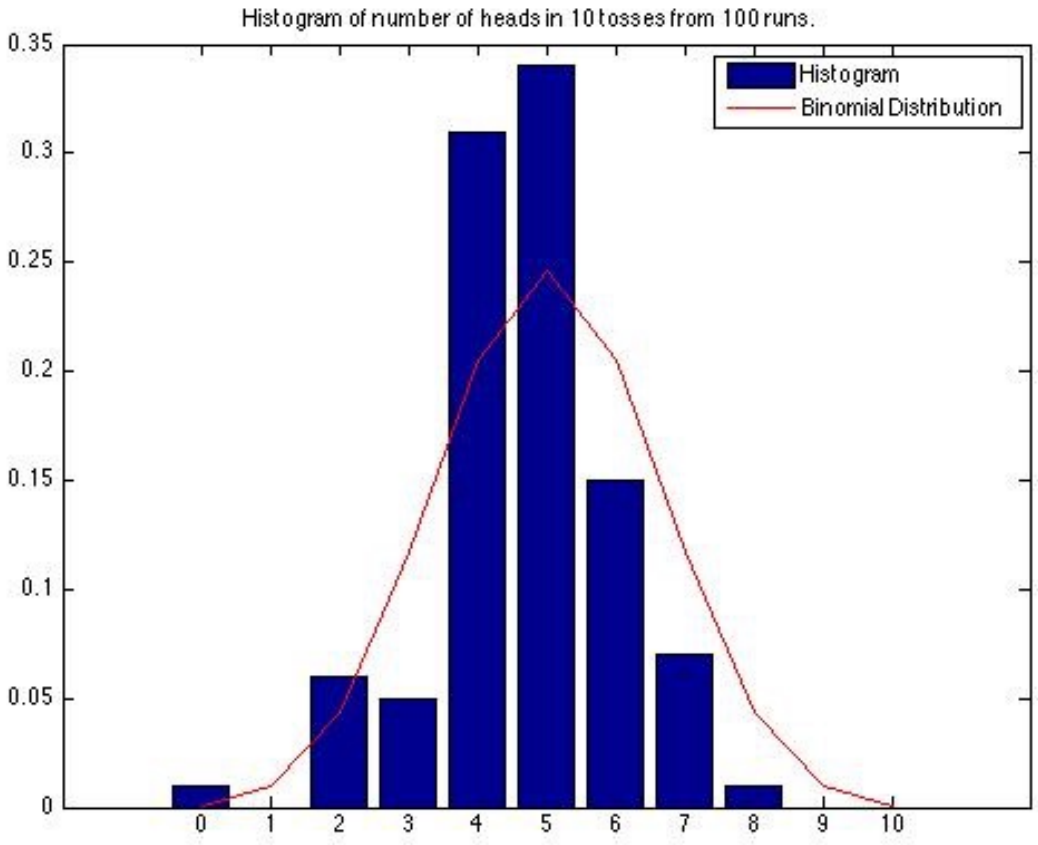
iv)

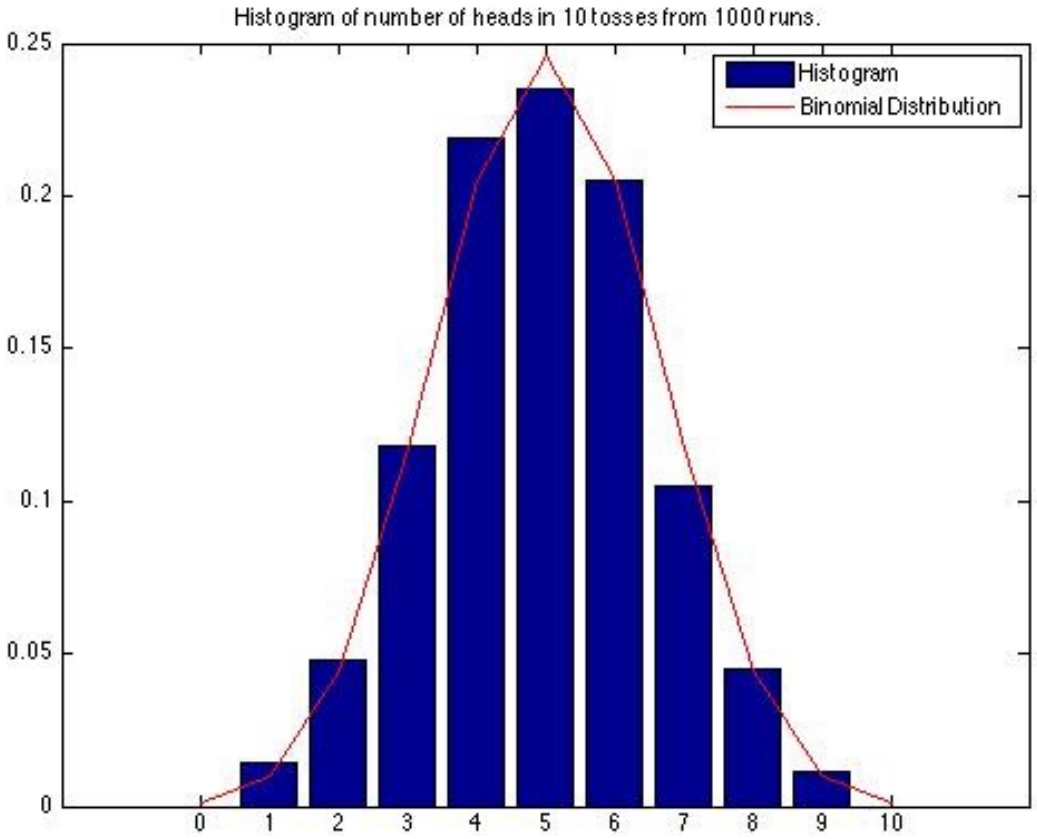
N coins are flipped. Let H (a random variable) be the number of heads. H has a binomial distribution, i.e., $P(H=h)=\begin{pmatrix} N \\ h \end{pmatrix}(\frac{1}{2})^N$. Use the data to generate a random variable that has a binomial distribution, for N = 10. Show the distribution for H is a close match to the analytical result.

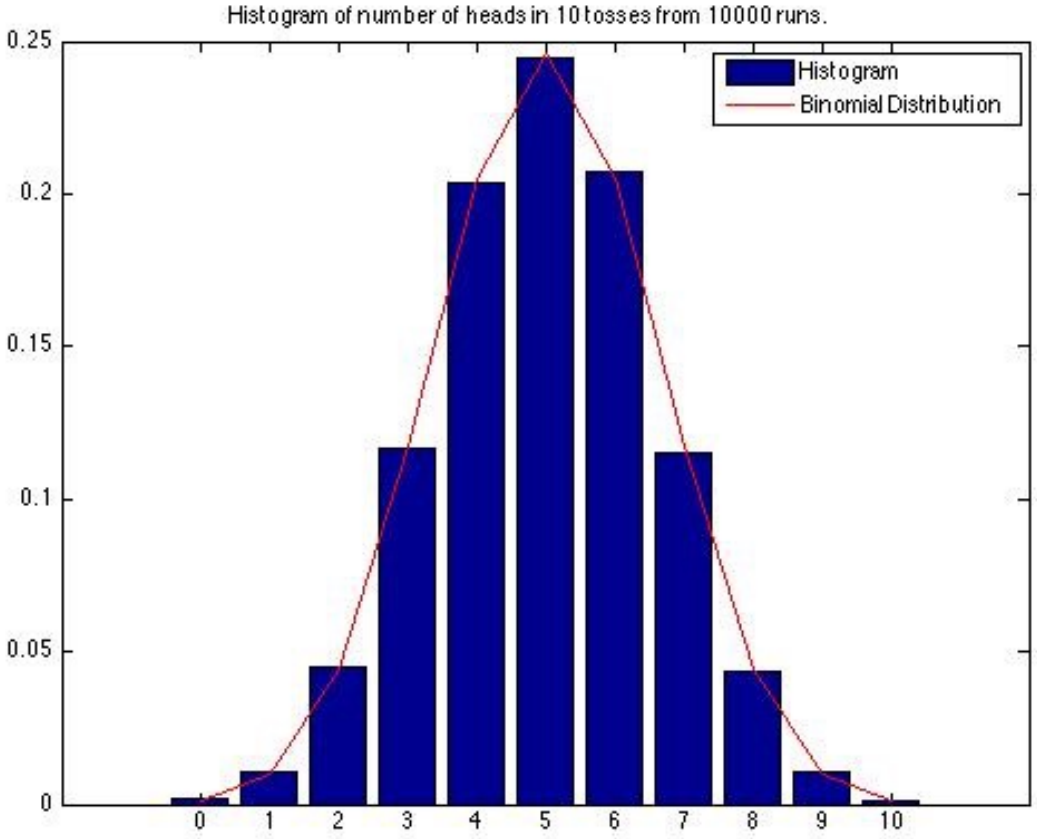
10 coin tosses are simulated with random number generator. Random numbers are generated between (0, 1). If it is larger than 0.5, the toss is considered to give head. Otherwise, toss gives tail. The number of heads in 10 tosses H is a random variable with binomial distribution. So it has mean of np = 10 * 0.5 = 5, variance of np(1-p) = 2.5. The simulation tries 100, 1000 and 10000 runs. Also with the increase of the number of runs, the result histograms tend to get closer to theoretical values. Here are the results and histogram with theoretical binomial distribution curve. Histograms are plotted with 11 bins:

Number of heads in 10 tosses from 100 runs.
Mean: 4.680000. Variance: 1.734949.
Number of heads in 10 tosses from 1000 runs.
Mean: 4.939000. Variance: 2.551831.
Number of heads in 10 tosses from 10000 runs.
Mean: 4.994600. Variance: 2.534224.

So the distribution for H is a close match to the analytical binomial distribution with n = 10 and p = 0.5.




Histogram of number of heads in 10 tosses from 1000 runs.
Histogram
Binomial Distribution
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10


Histogram of number of heads in 10 tosses from 10000 runs.
Histogram
Binomial Distribution
0.25
0.2
0.15
0.1
0.05
0
0 1 2 3 4 5 6 7 8 9 10

Code:

```
% part 1
fprintf('Part 1: \n');
u = ones(50, 1) / 50; % uniform distribution curve
for count = 1:5 % test 10, 100, 1000, 10000, 100000 random numbers sequence
   fprintf('Sequence of %d random numbers from (0, 1).\n', 10^count);
   rands = rand(10^count, 1); % generate random numbers sequence
   fprintf('Mean: %f. Variance: %f.\n', mean(rands), var(rands));
   figure; % plot the histogram and uniform distribution curve
   [nelements, centers] = hist(rands, 50); % generate histogram
   bar(centers, nelements / 10^count); hold on % plot histogram
   plot(centers, u, 'r'); hold off % plot uniform distribution curve
   title(['Histogram of ', num2str(10^count), ' random numbers from (0, 1)']);
   legend('Histogram', 'Uniform Distribution');
end

% part 2
fprintf('\nPart 2: \n');
for count = 3:5 % test 1000, 10000, 100000 random numbers sequence
   sums = zeros(10000, 1); % stores the sums of 100 runs
   for i = 1:10000
      rands = rand(10^count, 1); % generate random numbers sequence
      % calculate the sum
      sums(i) = sum(rands(1:2:(10^count - 1)) .* rands(2:2:10^count));
   end
   fprintf('Sums from sequence of %d random numbers from (0, 1).\n', 10^count);
   fprintf('Mean: %f. Variance: %f.\n', mean(sums), var(sums));
   figure; % plot the histogram and normal distribution curve
   [nelements, centers] = hist(sums, 100); % generate histogram with 100 bins
   bar(centers, nelements / 10000); hold on % plot histogram
   g = normpdf(centers, 0.125 * 10^count, sqrt(7 * 10^count / 288));
   plot(centers, g, 'r'); hold off % plot normal distribution curve
   title(['Histogram of sums from', num2str(10^count), ...
        ' random numbers from (0, 1)']);
   legend('Histogram', 'Normal Distribution');
end

% part 3
fprintf('\nPart 3: \n');
u = ones(50 , 1) / 50; % uniform distribution curve
for count = 1:5 % test 10, 100, 1000, 10000, 100000 random numbers sequence
   fprintf('Random number (0, 49) from sequence of %d random numbers.\n',...
    10^count);
   rands = rand(10^count, 1); % generate random numbers sequence
   zero249 = zeros(10^count, 1); % mapping from rands
   for i = 1:49
      zero249 = zero249 + (rands >= (i/50));
   end
```

```
    fprintf('Mean: %f. Variance: %f.\n', mean(zero249), var(zero249));
    figure; % plot the histogram and uniform distribution curve
    [nelements, centers] = hist(zero249, 50); % generate histogram
    bar(centers, nelements / 10^count); hold on % plot histogram
    plot(centers, u, 'r'); hold off % plot uniform distribution curve
    title(['Histogram of ', num2str(10^count),...
     ' discrete random numbers from (0, 49)']);
    legend('Histogram', 'Uniform Distribution');
end

% part 4
fprintf('\nPart 4: \n');
toss = 10; % number of tosses per run
for run = [100, 1000, 10000] % number of runs
    heads = zeros(run, 1); % number of heads per run
    for i = 1:run % run 100 times
        head = 0; % the number of heads per run
        for j = 1:toss % toss 10 times per run
            c = round(rand()); % random number 1 or 0, 1 as head, 0 as tail
            if c == 1
                head = head + 1;
            end
        end
        heads(i) = head;
    end
    fprintf('Number of heads in 10 tosses from %d runs.\n', run);
    fprintf('Mean: %f. Variance: %f.\n', mean(heads), var(heads));
    b = binopdf(0:10, 10, 0.5); % binomial distribution curve
    figure; % plot the histogram and binomial distribution curve
    [nelements, centers] = hist(heads, 0:10); % generate histogram
    bar(centers, nelements / run); hold on % plot histogram
    plot(centers, b, 'r'); hold off % plot binomial distribution curve
    title(['Histogram of number of heads in 10 tosses from ', num2str(run),...
     ' runs.']);
    legend('Histogram', 'Binomial Distribution');
end
```