Note:

Dear Sir/madam,

I have uploaded all of the code into github.

If you are interested in them, just check

<https://github.com/Vin453095545/EE511/invitations>

You can check the README.txt and to find who I am.

Thanks!

Fan(Vin) Zhang USC ID: 1417-68-5115

EE511 homework 3 Feb. 6

Question 1: A components manufacturer delivers a batch of 125 microchips to a parts distributor. The distributor checks for lot conformance by counting the number of defective chips in a random sampling (without replacement) of the lot. If the distributor finds any defective chips in the sample it rejects the entire lot. Suppose that there are six defective units in the lot of 125 microchips. Simulate the lot sampling to estimate the probability that the distributor will reject the lot if it tests five microchips. What is the fewest number of microchips that the distributor should test to reject this lot 95% of the time?

**Theoretical results:**

* Five tries

0.2212

* Fewest number of microchips that the distributor should test to reject this lot 95% of the time:

As the code result shows, the result is that we need to try 49 times.

%theoretically

clear

TOTAL=125;

DEFECTIVE=6;

FINE=TOTAL-DEFECTIVE;

TRIES=5;i=0;P=0;REJECT=0.95;

while(sum(P)<REJECT)

i=i+1;

TOTAL=125;

DEFECTIVE=6;

FINE=TOTAL-DEFECTIVE;

P(i)=1;

for j=1:i-1

P(i)=P(i)\*FINE/TOTAL;

TOTAL=TOTAL-1;

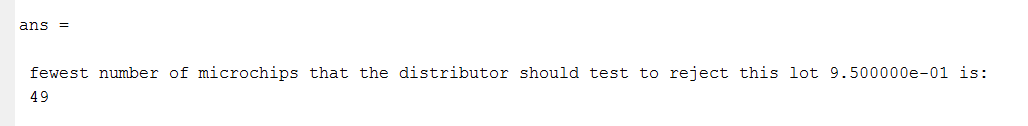
FINE=FINE-1;

end

P(i)=P(i)\*DEFECTIVE/TOTAL;

End

sprintf(' fewest number of microchips that the distributor should test to reject this lot %d is: \n %d', REJECT, i)



**Simulation results:**

* Analysis: Just to for loop until 95% reject it.
* **C**ode

%Q1(2

clear

B=ones(1,125);

B(1:6)=0;

DEFECT\_N=0;

N=10000

EXPERIMENT=0;

while(DEFECT\_N/N<0.95)

if(EXPERIMENT==5)

sprintf('the propability of 5 tires to rejecct: %2f ',DEFECT\_N/N)

end

EXPERIMENT=EXPERIMENT+1;

DEFECT\_N=0;

for z=1:N%N experiments

A=B;

i=0;

while(i<EXPERIMENT)%the number of trials

i=i+1;

j=ceil(125\*rand);

if(A(j)==1)

A(j)=2;%without replacement

elseif(A(j)==0)

DEFECT\_N=DEFECT\_N+1;

break;

else

i=i-1;

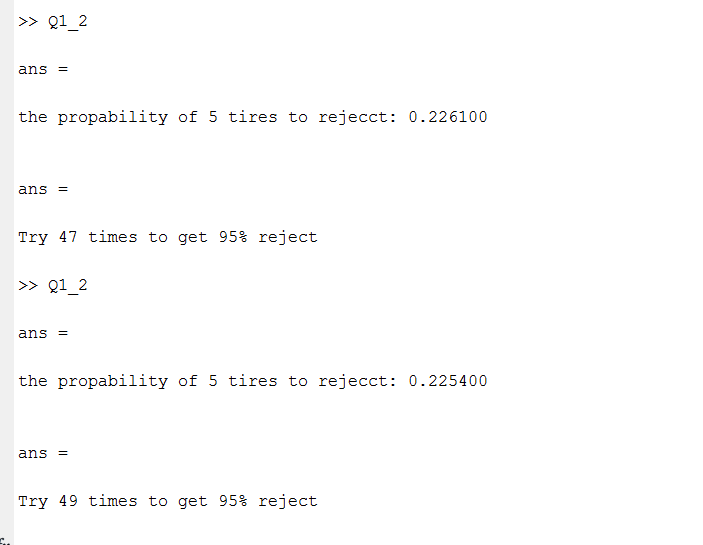
end

end

end

end

sprintf('Try %d times to get 95%% reject', EXPERIMENT)



* Comment:

As we can see, the simulation result is very close the to the theoretical results.

Question2:

Suppose that 120 cars arrive at a freeway onramp per hour on average. Simulate one hour of arrivals to the freeway onramp: (1) subdivide the hour into small time intervals (< 1 second) and then (2) perform a Bernoulli trial to indicate a car arrival within each small time interval. Generate a histogram for the number of arrivals per hour. Repeat the counting experiment by sampling directly from an equivalent Poisson distribution by using the inverse transform method (described in class). Generate a histogram for the number of arrivals per hour using this method. Overlay the theoretical p.m.f. on both histograms. Comment on the results.

**Analysis for title of statement**:

Three tasks should be finished in this question:

a. Generate a histogram for number of arrivals per hour with Bernoulli trail

b. Generate a histogram for number of arrivals per hour with inverse transform methods

c. Overlay the theoretical on both histograms and make comment.

**Solution for task a:**

* Analysis:
  + Divide 1 hour into N(N=10000) parts.
  + Suppose only at most one car arrives in each interval.
  + The theoretical probability that one car comes into is 120/N.
  + Generate a random value, if it smaller than 10/N, than assume a car comes during this interval.
* Code

%Q2

clear

N=10000;

TRIES=[100,1000,10000];

E=120/N;

for z=1:3

for y=1:TRIES(z)

for i=1:N

CarArriving(i)=rand<E;

end

SumCarArriving(y)=sum(CarArriving);

end

subplot(1,3,z);

hist(SumCarArriving);

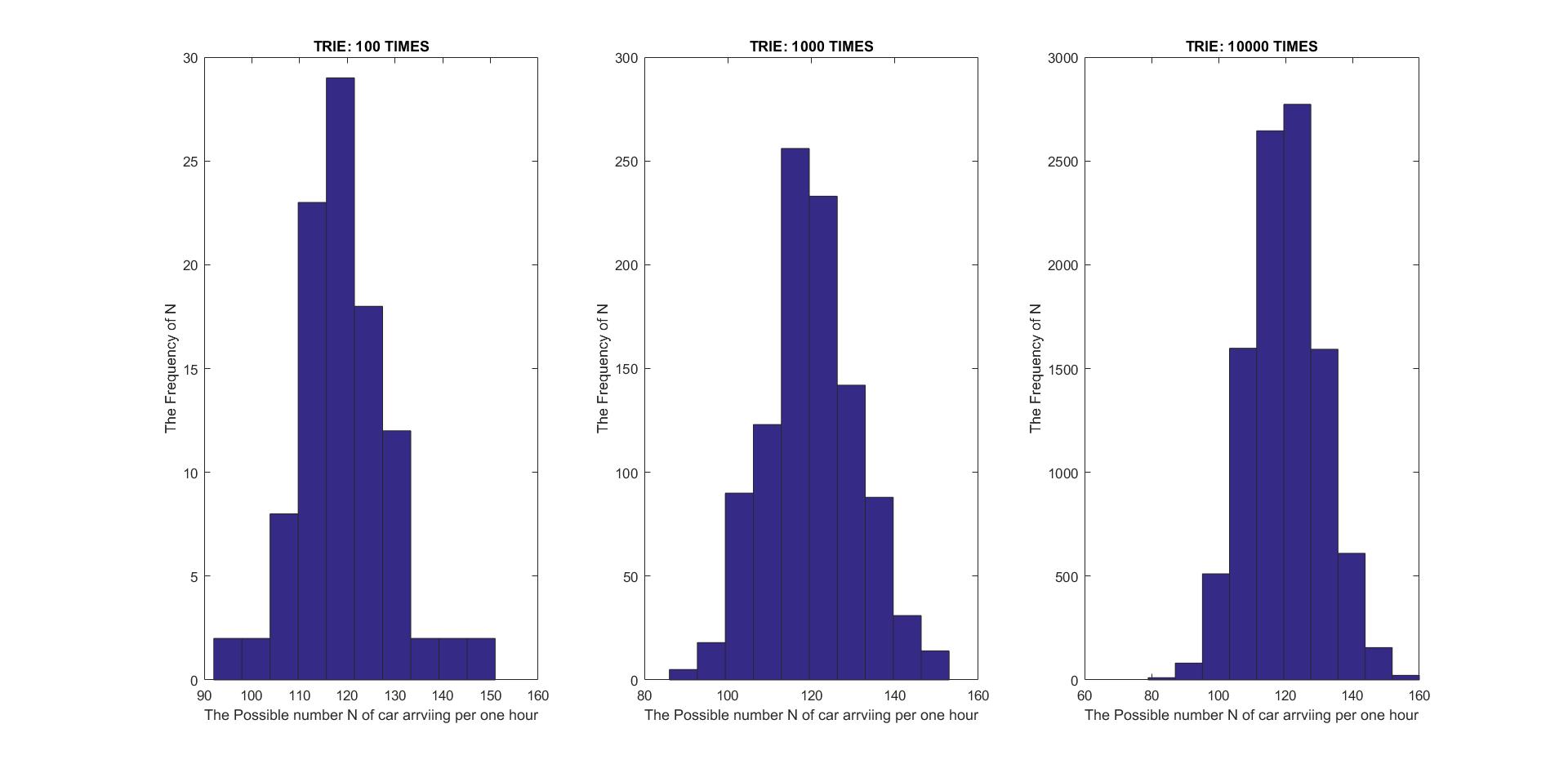
title(['TRIE: ',num2str(TRIES(z)),' TIMES']);

ylabel('The Frequency of N');

xlabel('The Possible number N of car arrviing per one hour');

end

* Results：



**Solution for task b:**

* Analysis**:** inverse transform methods algorithm:
  + Define a passion distribution: In this question,
  + generate a rand to make it larger than , and record the k.
  + repeated above operation.
* Code
* %2\_2
* clear

* TRIES=[100,1000];
* lambda=120;
* %generate a theoretical poisson p.d.f
* Theo\_Poisson=poisspdf(lambda,90:0.1:159)
* %generate a simultaiton histgram with inverse transform method

for z=1:2

for i=1:TRIES(z)

temp=rand;

j(i)=1;

while(temp>sum(poisspdf(lambda,1:j(i))))

%inverse transform methodequal to exp(-lambda).\*lambda.^k(i)/factorial(k(i))+p

j(i)=j(i)+1;

end

end

subplot(1,2,z);

histogram(j,1)

bins= range(j)%rescale the bandwidth of p

histogram(j,bins,'Normalization','probability')

hold on

plot(90:0.1:159,Theo\_Poisson)

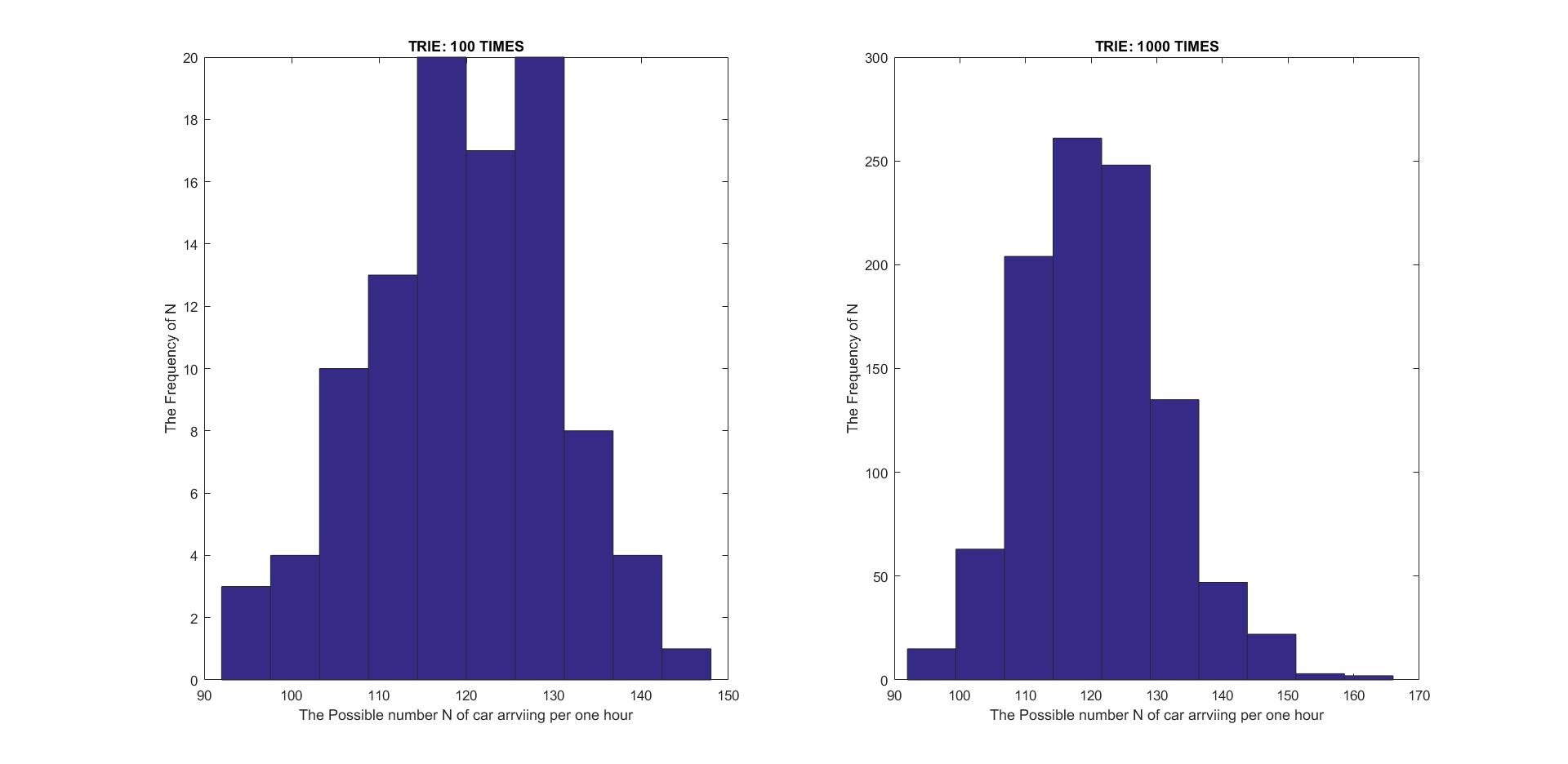
title(['TRIE: ',num2str(TRIES(z)),' TIMES']);

ylabel('The Frequency of N');

xlabel('The Possible number N of car arrviing per one hour');

end

* Results:



**Solution for task b:**

In this task, we need to rescale the histogram of the figure in task a and task b. And the unit width of simulation figure in x-axil to match the theoretical is necessary.

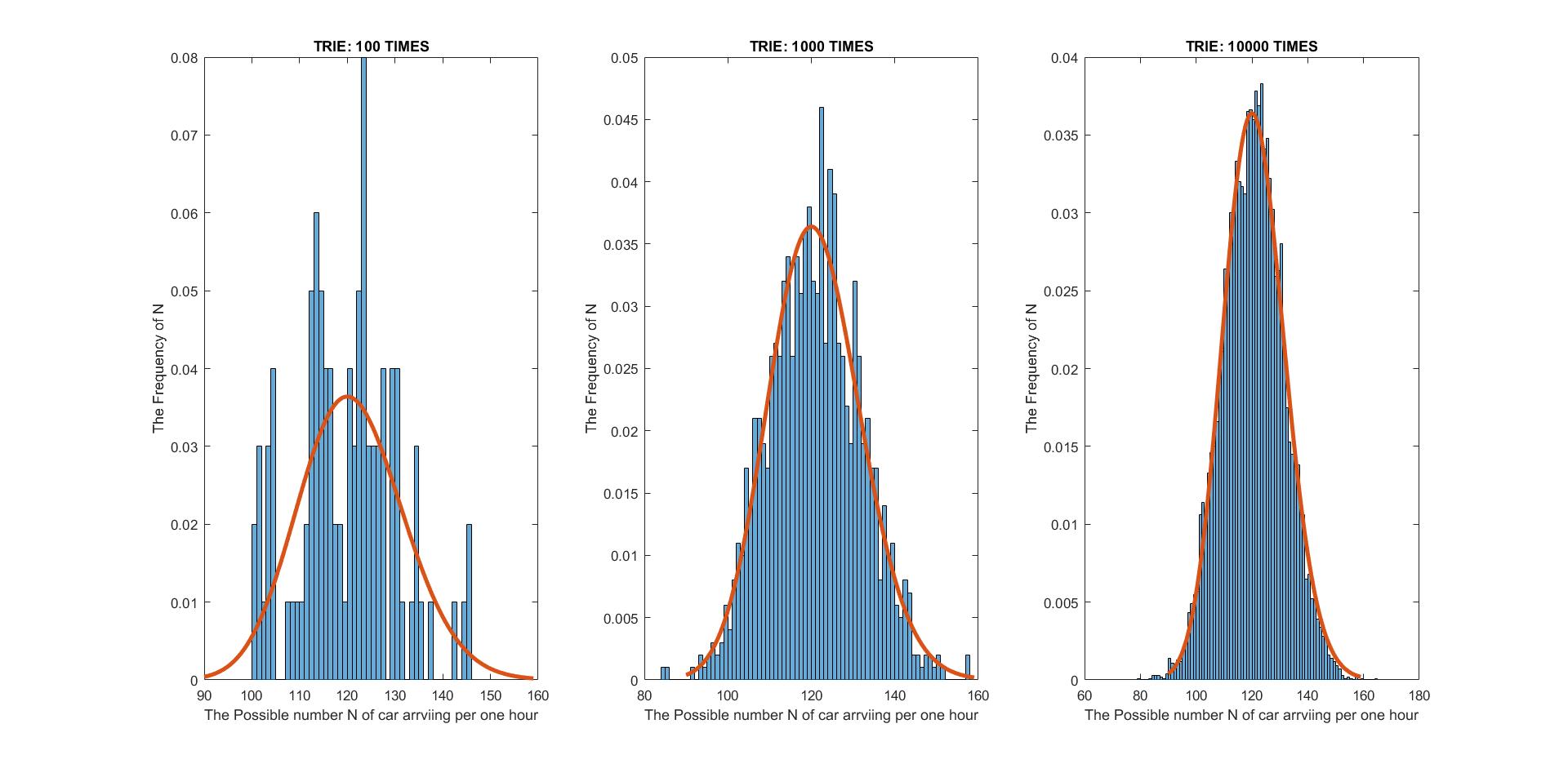


Figure: the overlay between theoretical and simulation with Bernoulli trail

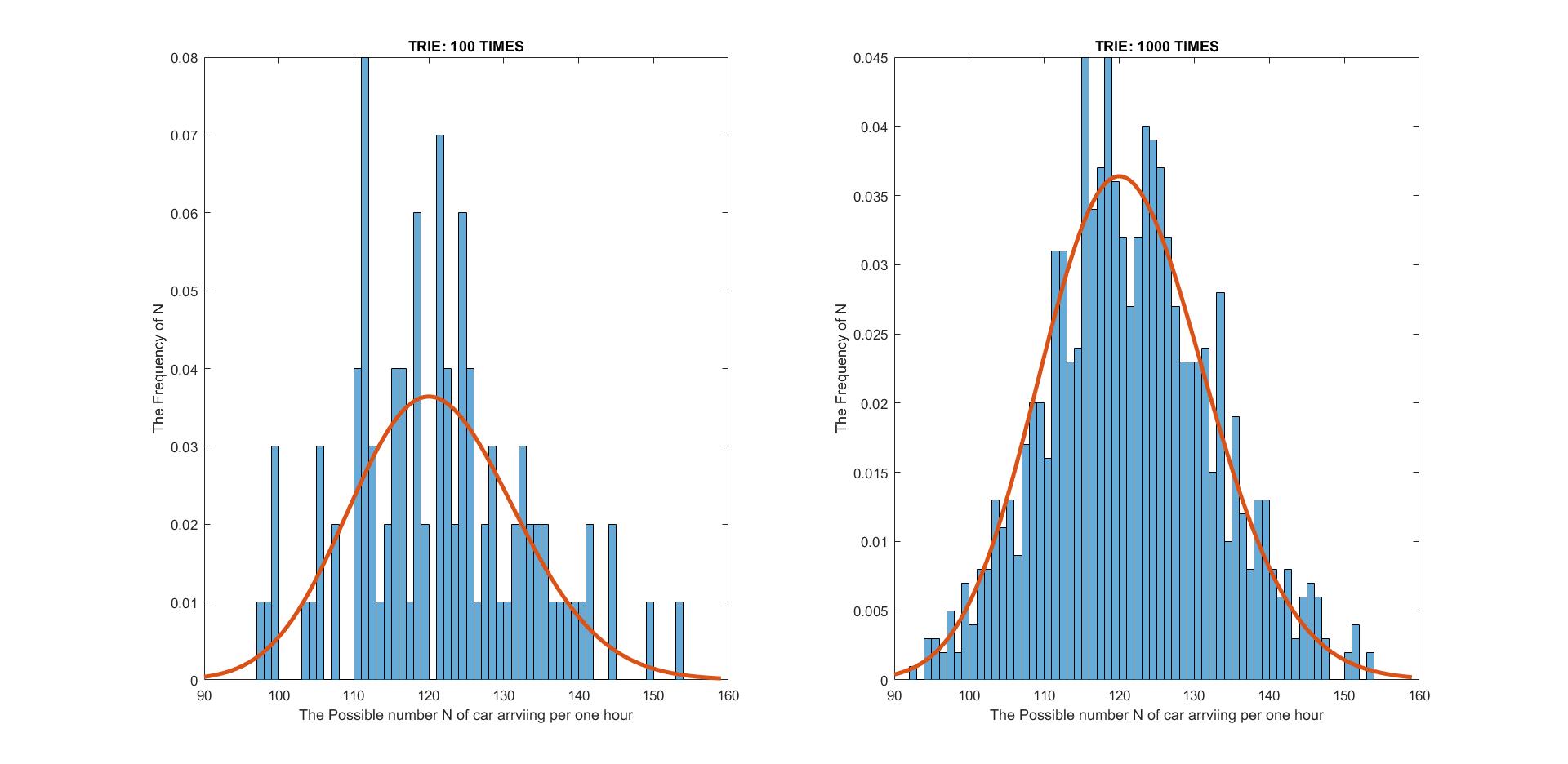
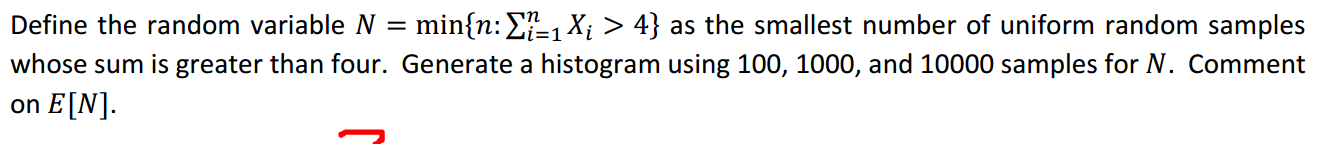


Figure: the overlay between theoretical and simulation with inverse transform method

* Comment:
  + As we can see from the above figure, both two methods fit the theoretical results well.
  + The more we try, the more the simulation will fit the theoretical results.

Question3 :



* **Codes:**

clear

TRIES=[100,1000,10000]

SUM=0;

i=0;

for z=1:3

for j=1:TRIES(z)

SUM=0;

i=0;

while(SUM<4)

SUM=rand+SUM;

i=i+1;

end

x(j)=i;

end

subplot(1,3,z)

hist(x)

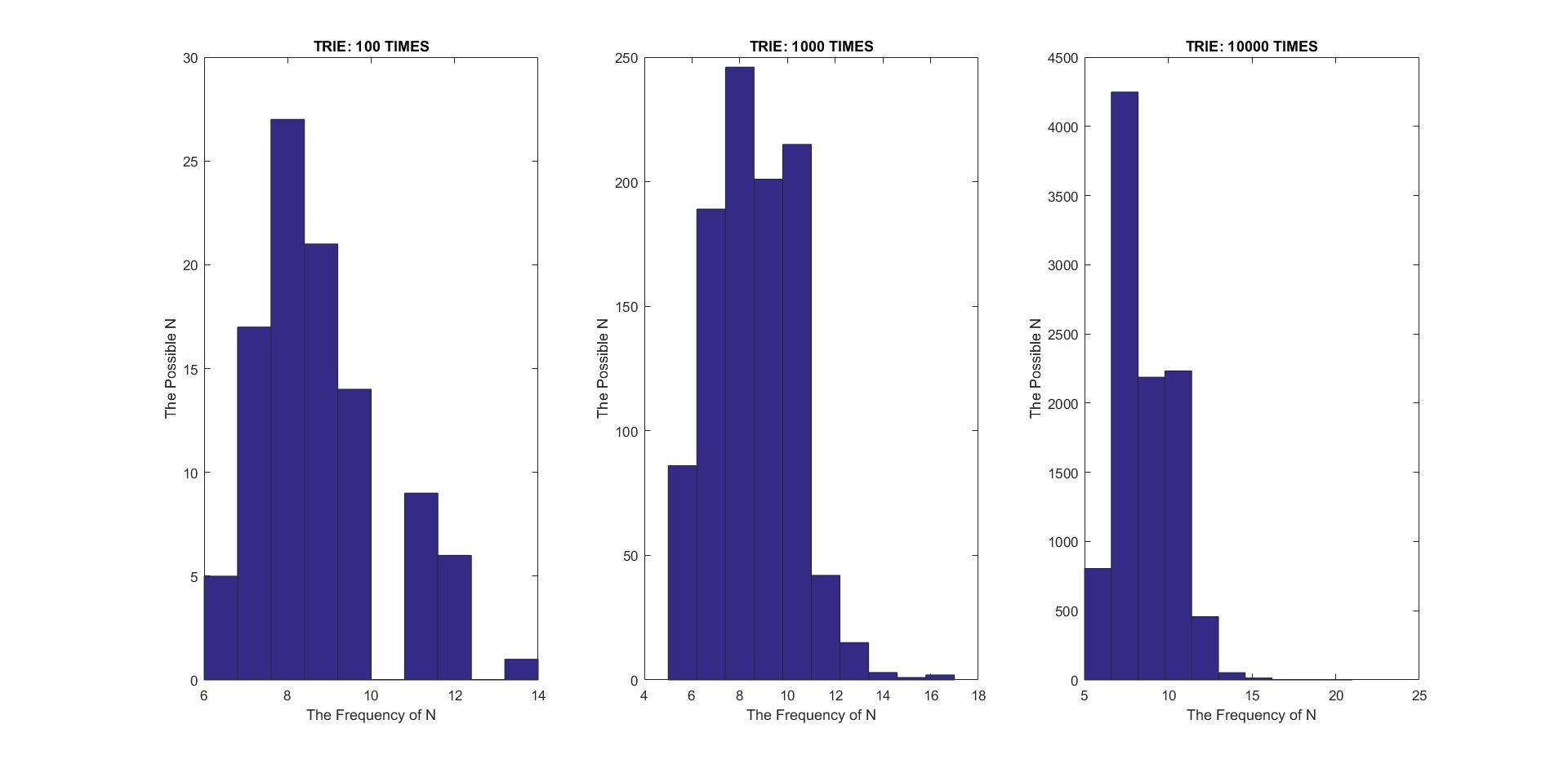
title(['TRIE: ',num2str(TRIES(z)),' TIMES']);

xlabel('The Frequency of N');

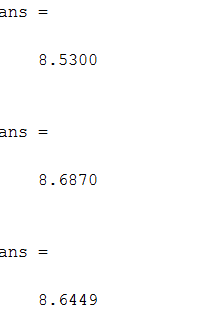
ylabel('The Possible N');

end

* results:

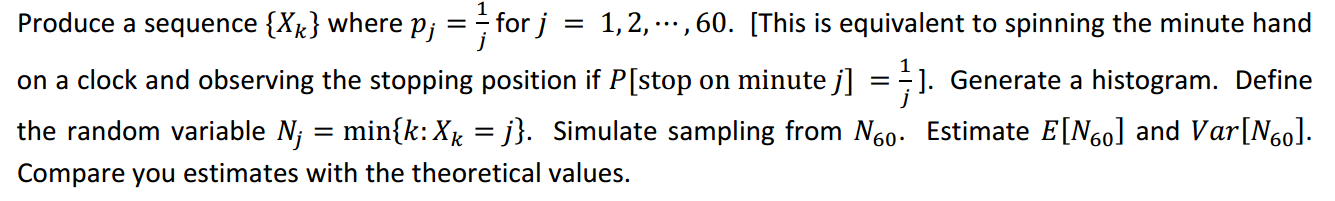


The mean for three different simulation times are:



* Comment:
  + Since the theoretical N is 8.667, so when we do simulation, the frequency of 8.667 will be the highest.
  + The more we try, the higher of the frequency of 8.667.

Question4:



* Analysis:
  + Generate a sequence {}
  + generate N rand, calculate the number that the rand fall into the target.
  + Do loop until the generated rand number fall into , and record the number of trials to obtain this rand number.
* Code for histogram

%Q4

p=1./(1:60)

p=p./sum(p)

TRIES=[100,1000,10000]

for z=1:3

y=randsrc(TRIES(z),1,[1:60; p])%to use randsrc, MATLAB shoud be updated to 2016b

subplot(1,3,z);

hist(y)

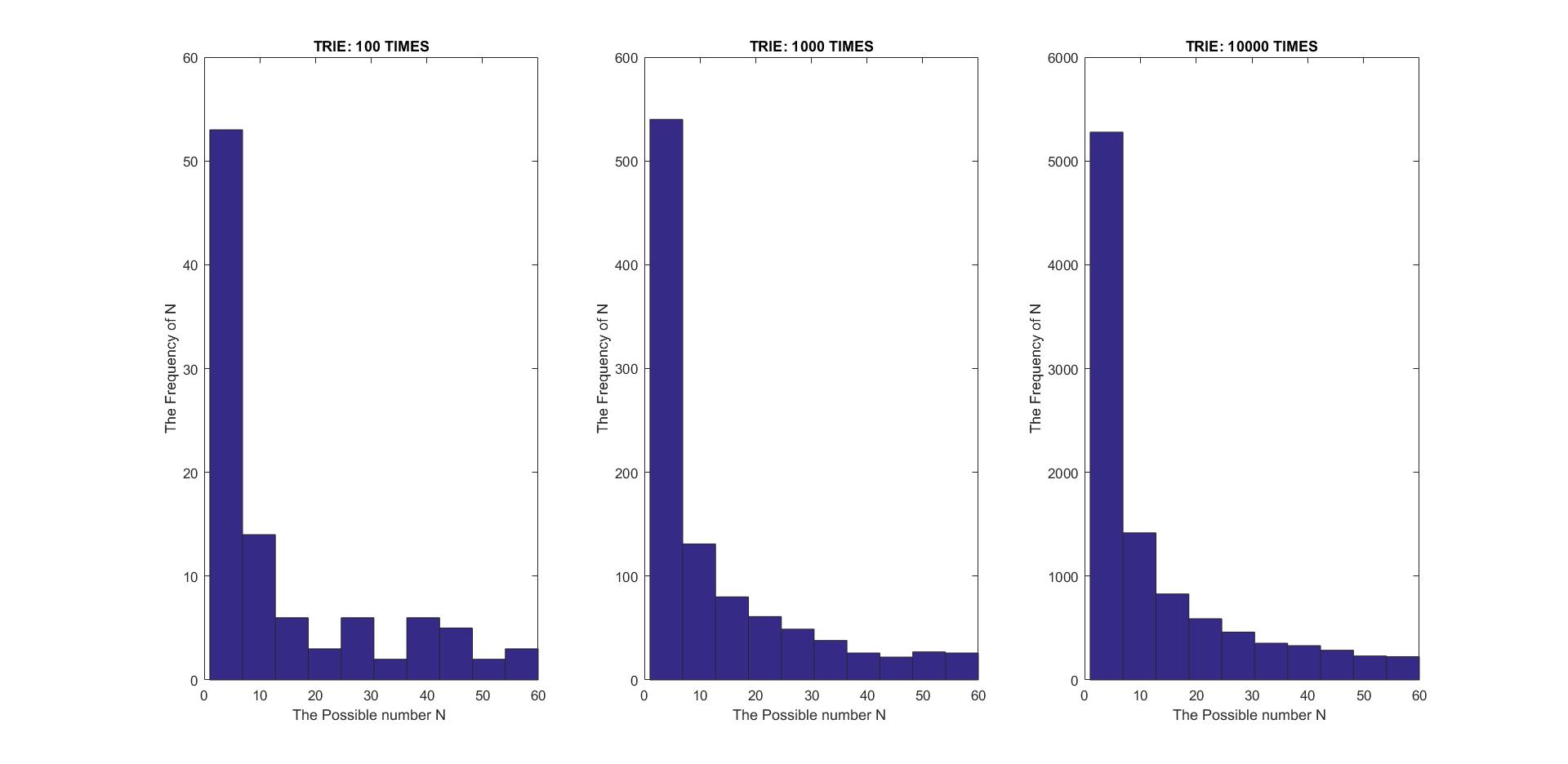
title(['TRIE: ',num2str(TRIES(z)),' TIMES']);

ylabel('The Frequency of N');

xlabel('The Possible number N ');

end

* Results for histogram:



* Code for

%4

clear

p=1./(1:60);

p=p./sum(p);

TRIES=[100,1000,10000];

for z=1:3

for i=1:TRIES(z)

N(i)=0;

while(rand<(1-p(60)))

N(i)=N(i)+1;

end

end

subplot(1,3,z);

hist(N)

title(['TRIE: ',num2str(TRIES(z)),' TIMES']);

ylabel('The Frequency of N');

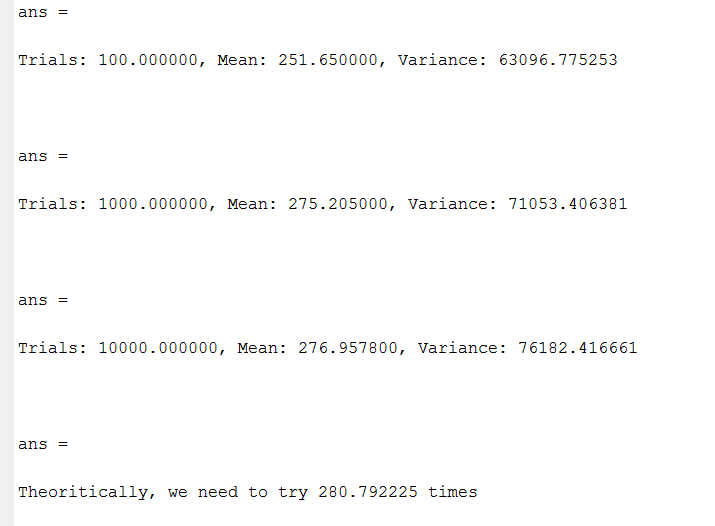
xlabel('The Possible number N of car arrviing per one hour');

sprintf('Trials: %f, Mean: %f, Variance: %f\n', TRIES(z), mean(N),var(N))

end

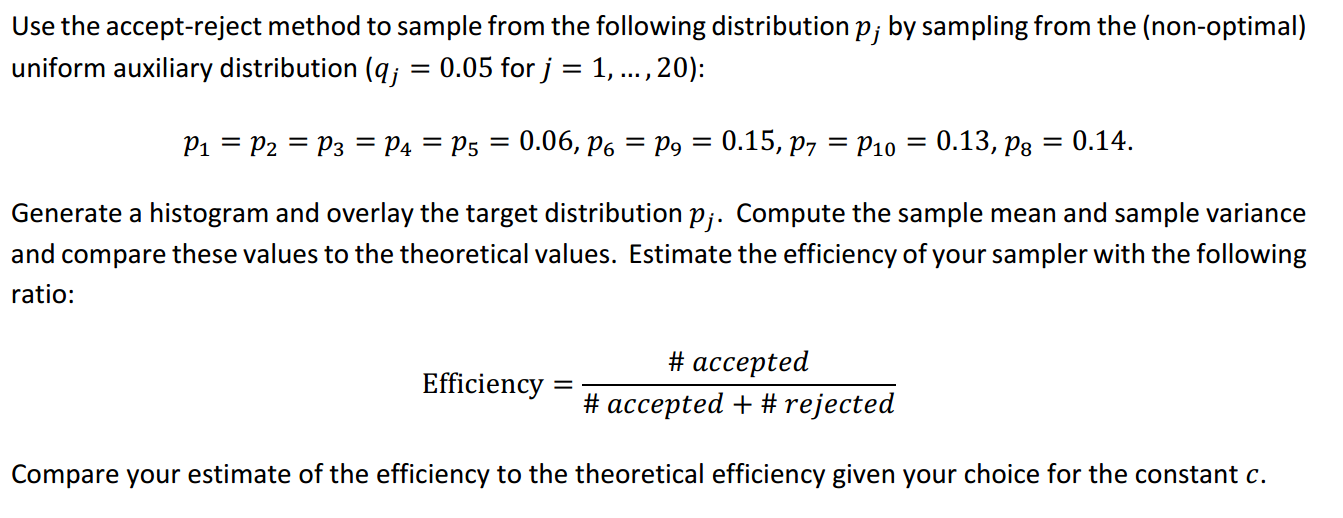
sprintf('Theoritically, we need to try %f times', 1/p(60))

* Results for



* Comment:
  + The more times we simulate, the closer to theoretical results the simulation results are.
  + From the simulation results, we can find that when the simulations times are not enough large, the results have great possibility to be smaller than the theoretical result. May be it is because the probability that the generated rand number fall into is too low. So the results are either much larger than the theoretical results (when it falls in into ) or always smaller than the theoretical results.

Question 5:



* **code for distribution of pj:**

%Q5

clear

N=10000

q=[6 6 6 6 6 15 13 14 15 13]/100

x(1:10)=0;

for j=1:N

temp=rand;

for i=1:10

x(i)=x(i)+(temp<sum(q(1:i)));

end

end

X=[x(1),diff(x)];

sampleX=X/N;

bar(sampleX);

hold on

plot(q,'--gs',...

'LineWidth',2,...

'MarkerSize',10,...

'MarkerEdgeColor','r',...

'MarkerFaceColor',[0.5,0.5,0.5]);

title('Probability')

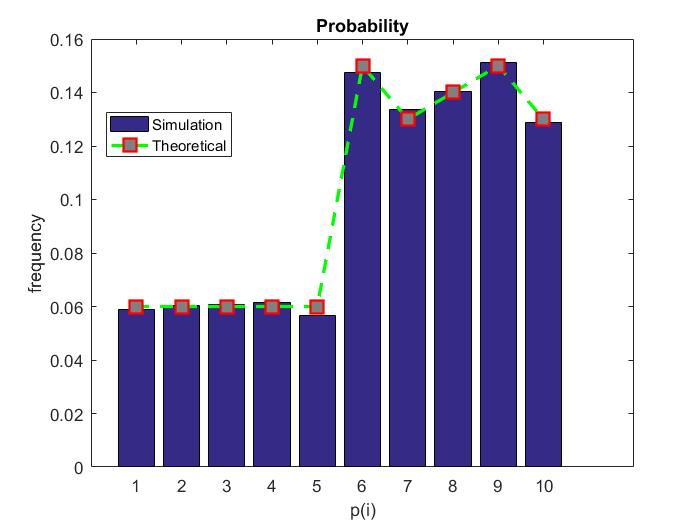
xlabel('p(i)');

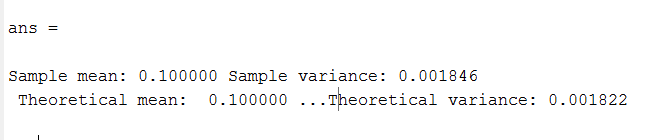
ylabel('frequency')

legend('Simulation','Theoretical')

sprintf('Sample mean: %f Sample variance: %f\n Theoretical mean: %f ...Theoretical variance: %f',mean(sampleX),var(sampleX),mean(q),var(q))

* **Results for distribution of pj:**





* Analysis for the calculating efficiency；
  + Theoretically： the probability of the efficiency is the
* Code for the calculating efficiency：

clear

N=10000;

p=[6 6 6 6 6 15 13 14 15 13]/100

q=0.1;

c=max([p,q])/q

for i = 1:N, k = 0;

while(1)

k = k + 1;

j = 1 + floor(10\*rand); % Get Uniform j

if (c\*rand) < p(j)/0.1 % Accept p(j) if U<p(j)/c, q(j)= 0.1

X(i) = p(j); C(i) = k;

break

end

end

end

sprintf('Simulation efficience: %f\n My estimate: %f', 1/mean(C),sum(p/c))

* Results for the calculating efficiency

