DELHI TECHNOLOGICAL UNIVERSITY

Stochastic Process Lab File

MC – 303

Anish Sachdeva
DTU/2K16/MC/13

SNO.	EXPERIMENT	DATE	SIGNATURE
1.	Discrete State Space: No. of cars washed on n^{th} day in a car wash given minimum 20 cars are washed and maximum 30 cars are washed.		
2.	Continuous State Space: Average time taken for a car to be worked on n^{th} day of month given time required is 2 minutes and maximum time taken is 4 minutes.		
3.	Continuous State Space: Variation in temperature in a time period of 5 minutes given that temperature can vary between 22 and 26 C.		
4.	Discrete State Space: No. of times temperature changed within 5 minutes and when it changed given temperature can change a maximum of 10 times.		
5.	It has been observed that a fuse designed by a company follows bernoulli distribution when being manufactured. If the company sells fuse in a box of 20 and gives guarantee that box will be replaced if no. of defective fuses is greater than 2. Find expected no. of replacements in a lot of 1000 boxes.		
6.	Assuming that a circuit has an IC whose time to failure is exponentially distributed with expected lifetime of 3 months. If there are 10 spare IC's and time from failure to replacement is negligible. What is the probability that circuit is operational for a year? For non homogenous case consider expected lifetime to be 3/n and for renewal		

	consider time to failure is uniformly distributed with b = 3 and a = 0.		
7.	Simple unrestricted random walk		
8.	To implement a transition probability matrix (TPM)		
9.	Plotting a Normal Curve		
10.	Consider an M/M/1 model and for steady state write a program to find the mean and variance of queue length, the mean and variance of waiting time and mean duration of busy period.		

QUESTION 1

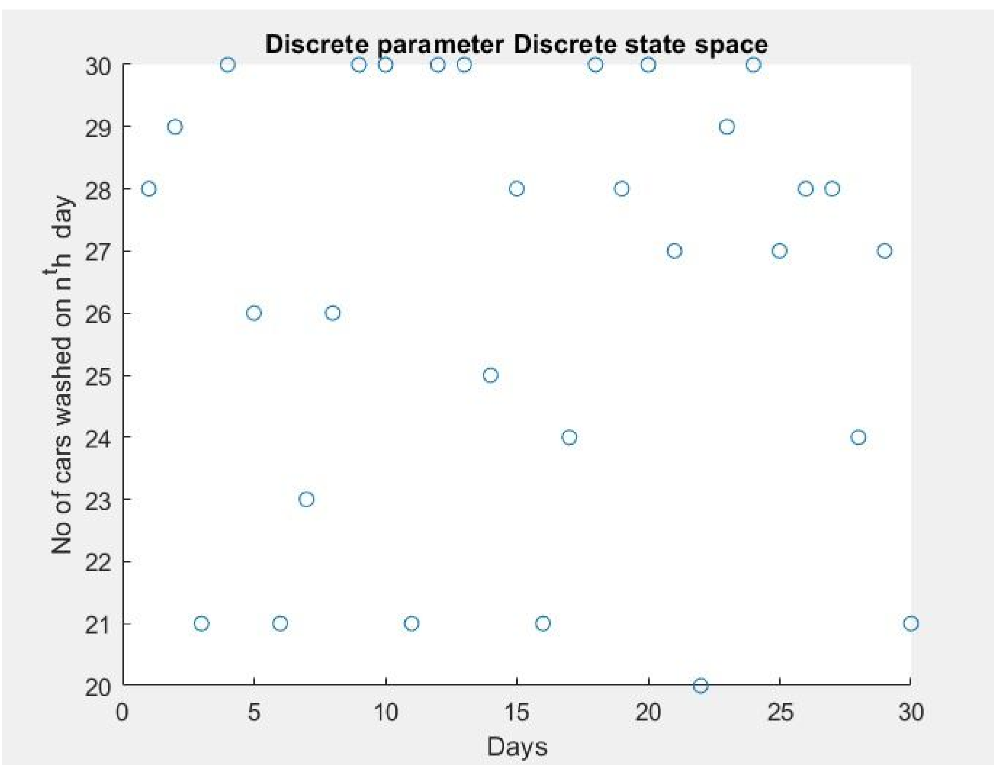
Simulate the following discrete parameter stochastic processes.

Discrete State Space: No. of cars washed on n^{th} day in a car wash given minimum 20 cars are washed and maximum 30 cars are washed.

CODE:

```
x = [1 : 1 : 30]
y = 20 + rand([0, 10], 30, 1);
p = scatter(x, y);
xlabel("Days");
ylabel("No of cars washed on nth day");
title("Discrete parameter Discrete state space");
```

OUTPUT:



QUESTION 2

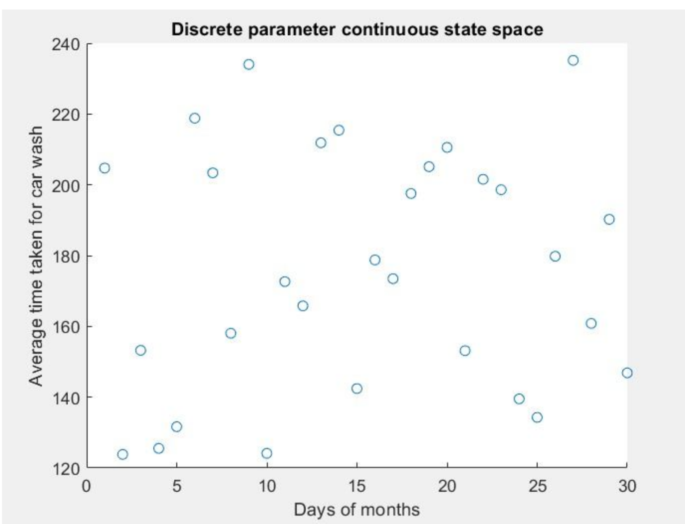
Simulate the following discrete parameter stochastic processes.

Continuous State Space: Average time taken for a car to be worked on n^{th} day of month given time required is 2 minutes and maximum time taken is 4 minutes.

CODE

```
x = [1 : 1 : 30]
y = 120 + 120.*rand(30, 1);
p = scatter(x, y);
xlabel("Days of months");
ylabel("Average time taken for car wash");
title("Discrete parameter continuous state space");
```

OUTPUT



QUESTION 3

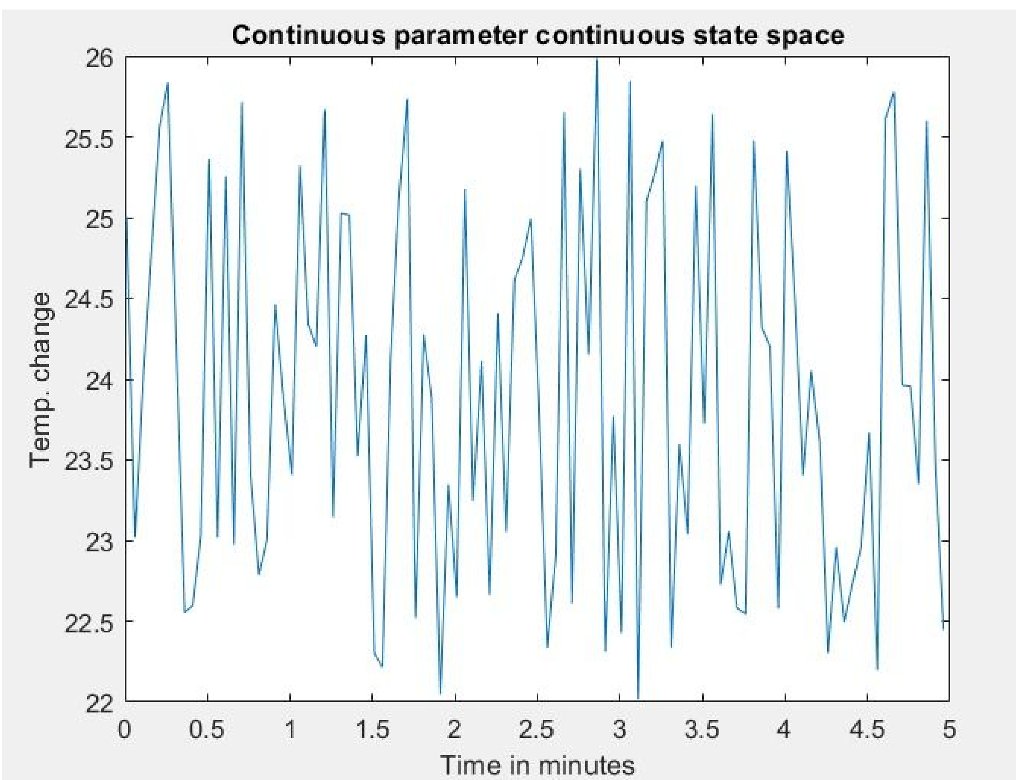
Simulate the following continuous parameter stochastic processes.

Continuous State Space: Variation in temperature in a time period of 5 minutes given that temperature can vary between 22 and 26 C.

CODE

```
x = [0.01 : 0.05 : 5]
y = 22 + 4.*rand(100, 1);
p = plot(x, y);
xlabel("Time in minutes");
ylabel("Temp. change");
title("Continuous parameter continuous state space");
```

OUTPUT



QUESTION 4

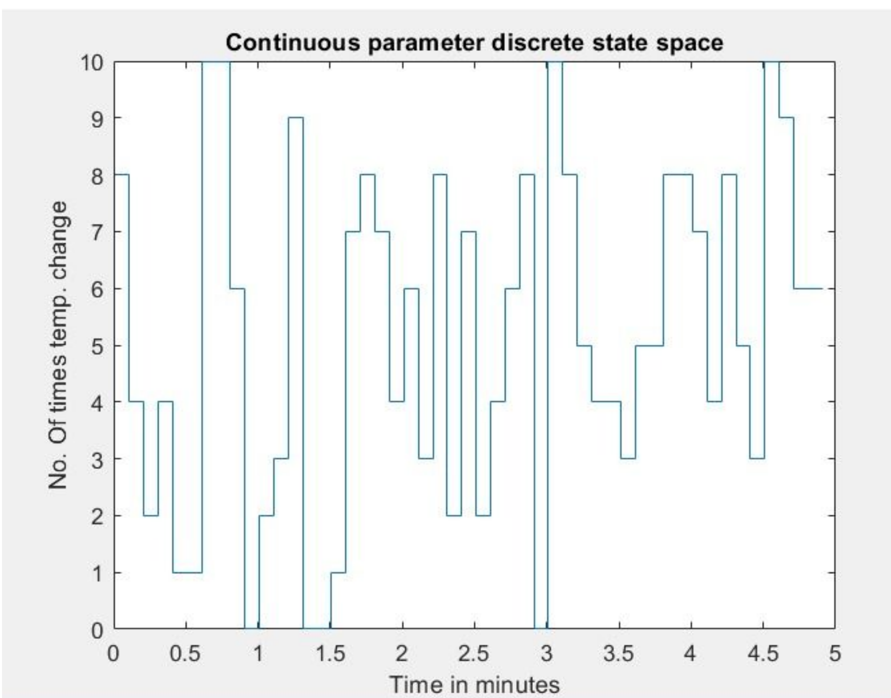
Simulate the following continuous parameter stochastic processes.

Discrete State Space: No. of times temperature changed within 5 minutes and when it changed given temperature can change a maximum of 10 times.

CODE

```
x = [0.01 : 0.5 : 5]
y = rand([0 : 10], 50, 1);
p = stairs(x, y);
xlabel("Time in minutes");
ylabel("No. Of times temp. change");
title("Continuous parameter discrete state space");
```

OUTPUT



QUESTION 5

It has been observed that a fuse designed by a company follows bernoulli distribution when being manufactured. If the company sells fuse in a box of 20 and gives guarantee that box will be replaced if no. of defective fuses is greater than 2. Find expected no. of replacements in a lot of 1000 boxes. Given -

1) Pr[fuse is defective] = 0.01 = p
2) Pr[n^{th} fuse is defective] = 0.01n = pn.

CODE

```
function[] = bernoulli()
p = 0.01;
q = 1 - p;
pr = 1 - ((nchoosek(20, 0) * (p ^ 0) * (q ^ 20)) +
(nchoosek(20, 1) * (p ^ 1) * (q ^ 19))
                       + (nchoosek(20, 2) * (p ^ 2) * (q
^ 18)))
no_replace = 1000 * p;
disp(no_replace)
end
```

OUTPUT

```
ans = 15.1903
```

QUESTION 6

Assuming that a circuit has an IC whose time to failure is exponentially distributed with expected lifetime of 3 months. If there are 10 spare IC's and time from failure to replacement is negligible. What is the probability that circuit is operational for a year? For non homogenous case consider expected lifetime to be 3/n and for renewal consider time to failure is uniformly distributed with b = 3 and a = 0.

CODE

```
function[] = poisson_homo(parameter, n)
    p = 0;
    for i = 0:n
                    pr = pr + ((exp(-parameter / i) *
        (parameter / i) ^ i) / factorial(i));
    end
end
```

OUTPUT

```
>> poisson(1, 10)

    ans = 0.4482
```

CODE

```
function[] = poisson_non_homo(parameter, n)
p = 0;
for i = 0:n
     pr = pr + ((exp(-parameter) * (parameter) ^ i) /
factorial(i));
end
end
```

OUTPUT

```
>>poisson(4, 10)

ans = 0.9972
```

CODE

```
function[] = uniform_renewal(n, f, a, b)
    C = 1 /(b - a)
    p = 0;
    for i = 0:n
                    pr = pr + ((c^n i * t^i * (i + 1 -
        e*t))/ factorial(i + 1));
    end
end
```

OUTPUT

```
>>uniform_renewal(10, 12, 0, 3)

ans = 0.09492
```

QUESTION 7

A simple unrestricted random walk with

1. p = 0.4, q = 0.6
2. p = 0.4, q = 0.5

Find the probability that after 100 steps at n = 100 the particle lies between -15 and 20 in both cases. Find the probability that particle is away from 25 that is position at n = 100 >= 25.

CODE

```
n=input('enter n - ');
p=input('enter p - ');
q=input('enter q - ');
if (p+q)<1
     r=1-p-q;
else
     r=0;
end
x1=input('\n enter Required points for
Probability[x1<X<x2]\n x1 - ');
x2=input('x2 - ');
x1=x1-0.5;
x2=x2+0.5;
m = p-q;
v = p+q-(p-q)^2;
mu = n*m;
var = n*v;
sigma=sqrt(var);
z1 = (x1-mu)/sigma;
z2 = (x2-mu)/sigma;
p = normcdf(z2)-normcdf(z1);
fprintf('\n P[% f < x < % f] = % f \n',x1,x2,p);
```

OUTPUT

```
>>random_walk

        enter p       0.4

        enter q       0.6

        P[-16 < x < 21] = 0.3415

        P[24 < = x] = 0.000004

>>random_walk

        enter p       0.4

        enter q       0.5

        P[-15.5 < x < 20.5] = 0.7194

        P[24.5 < = x] = 0.000128
```

QUESTION 8

To implement a transition probability matrix(TPM).

Find out the transition probability matrix for a random walk with barriers at 1 & 5 where :-

P[Zi = 1] = 0.5, P[Zi = -1] = 0.4, P[Zi = 0] = 0.1. For all 4 cases:

1. Both side absorbing barriers
2. Left side absorbing and right side reflecting barriers
3. Both side reflecting barriers
4. Left side reflecting and right side absorbing barriers

THEORY

If Xi = i, we say that the process is in state i at time n. Further we say Pij is the probability that if at time n the process is in state i then at time n + 1 the process will be in state j.

If there are n states in the process then there are n*n transition probability states. The one step transition probabilities are completely specified in the form of a transition probability matrix where:-

1. Pij >= 0
2. Sum of Pij of all independent rows = 1

CODE

```
function[answer] = markov_chain(p, q, r, n, c)
tpm = zeros(n, n);
for i = 1:n
     for j = 1:n
          If i - j == -1 && i ~= 1 && i~= n
               tpm(i, j) = p;
          else If i - j == 0 && i ~= 1 && i~= n
               tpm(i, j) = r;
          else if i - j == 1 && i ~= 1 && i ~= n
               tpm(i, j) = q;
          end
     end
end
```

```
Switch c
      Case 1:
            tpm(1, 1) = 1;  tpm(n, n) = 1;
      Case 2:
            tpm(1, 1) = 1;
            tpm(n, n) = 1 - q;
            tpm(n, n - 1) = q;
      Case 3:
            tpm(1, 1) = 1 - p;
            tpm(1, 2) = p;
            tpm(n, n) = 1 - q;
            tpm(n, n - 1) = q;
      Case 4:
            tpm(1, 1) = 1 - p;
            tpm(1, 2) = p;
            tpm(n, n) = 1;
end
answer = tpm;
for i = 1:n
      answer = answer * tpm
end
end
```

OUTPUT

>>markov_chain(0.5, 0.4, 0.1, 5, 1)

answer =

1.0000	**0**	**0**	**0**	**0**
0.4444	**0.00001**	**0**	**0**	**0**
0.1975	**0.000024**	**0.00001**	**0**	**0**
0.0876	**0.000024**	**0.000024**	**0.00009**	**0**
0	**0**	**0**	**0**	**1**

>>markov_chain(0.5, 0.4, 0.1, 5, 2)

answer =

1.0000	**0**	**0**	**0**	**0**

0.4444	0.0001	0	0	0
0.1975	0.00024	0.00001	0	0
0.0876	0.00024	0.000024	0.00001	0
0.0640	0.02368	0.02984	0.0373	0.04

```
>>markov_chain(0.5, 0.4, 0.1, 5, 3)
```

answer =

0.171045	0.12393	0	0	0
0.09914	0.071901	0	0	0
0.05752	0.041624	0.00001	0	0
0.03328	0.02424	0.000024	0.00001	0
0.0384	0.04288	0.02984	0.037324	0.046

```
>>markov_chain(0.5, 0.4, 0.1, 5, 4)
```

answer =

0.171045	0.12393	0	0	0
0.09914	0.071901	0	0	0
0.05752	0.041624	0.00001	0	0
0.03328	0.02424	0.000024	0.00001	0
0	0	0	0	1

Experiment 9

Question

Write a Program to plot a normal curve

Theory

A normal curve is a bell-shaped curve which shows the probability distribution of a continuous random variable. Moreover, the normal curve represents a normal distribution. The total area under the normal curve logically represents the sum of all probabilities for a random variable. Hence, the area under the normal curve is one. Also, the standard normal curve represents a normal curve with mean 0 and standard deviation 1. Thus, the parameters involved in a normal distribution is mean (μ) and standard deviation (σ).

Characteristics of a normal curve:

- The values of mean, median and mode are same

- It represents a unimodal distribution as it has only one peak.

- It shows a symmetric distribution as 50% of the data set lies on the left side of the mean and 50% of the data set lies on the right side of the mean.

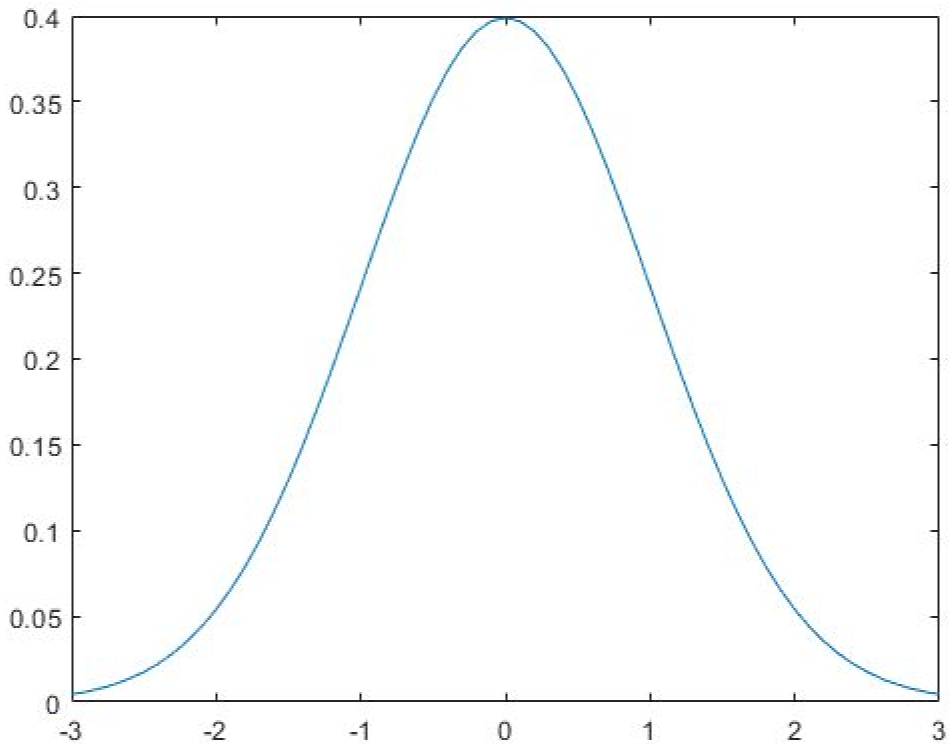
- Empirical rule: 68% of the data fall within $\mu \pm \sigma$, 95% of the data fall within $\mu \pm 2\sigma$ and 99.7% of the data fall within $\mu \pm 3\sigma$

Code

```
x = linspace(-10, 10, 5000);
for i = 0:1:5
    for j = 1:1:6
                    n = 1 / (j - 2*sqrt(2 * pi) *
        exp(-(x-1)^2 / (2*j^2));

        hold on;
        plot(n,n)
    end
end
```

Output



Question 10

Aim

Consider an M/M/1 model and for steady state write a program to find the mean and variance of queue length, the mean and variance of waiting time and mean duration of busy period.

Code

```
lambda = input('Enter Arrival Rate - ');
mu = input('Enter Service Rate - ');
rho = lambda/mu;
fprintf('Mean length of queue is:')
lq = (rho.^2)/(1-rho)
fprintf('Variance of queue length')
varlq = ((rho.^2)*(1+rho-rho.^2))/(1-rho).^2
fprintf('Mean Waiting time:');
wq = lq/lambda
fprintf('Variance of waiting time;')
varwq = (rho.*(2-rho))/((1 - rho).^2*mu.^2)
fprintf('Mean duration of busy period')
w = wq + 1/mu
```

Output

```
Enter Arrival Rate - 3
Enter Service Rate - 5
Mean length of queue is:
lq = 0.9000

Variance of queue length
var_ql = 2.7900

Mean Waiting time:
mean_wt = 0.3000

Variance of waiting time;
var_wt = 0.2100

Mean duration of busy period
mean_busy_period = 0.5000
```