Inventory Problem

Part A

```
%Set Initial Values
%Time:
T=30;                                   %limit time
t=[];t(1)=0;t(2)=0;                     %current time
L=2/24;                                 %Delivery time
t0=-(1/8)*log(rand(1));
%The arrival time of one customer (initial value)
t1=t0+2/24;%The arrival time of product(initial value)

%Quantity:
x=[];x(1)=0;x(2)=0;                   %Current inventory level
D=[];D(1)=0;                          %Demanding/ customer
s=10;S=120;                           %Minimum & Maximum inventory level
y=0;                                  %Quantity of goods ordered but not arrived

%Cost price
r=12;                                 %sell price/unit
h=0.5;                                %store price/unit
c=5;                                  %cost price/unit
Dv=10;                                %Delivery cost/ride

%Cost series
H=[];H(1)=0;                         %Storing cost
C=[];C(1)=0;                         %Cost price
Loss=[];                             %Money that should have been earned is not earned
R=[];R(1)=0;                         %Revenue
Profit=[];Profit(1)=0;               %Net Profit

%Counters
i=2;                                 %Counter of event
d=0;                                 %Counter of delivery
Event_type=['t0'];
```

```
while t(i)<=T
    if t0<t1                    %Customer arrives earlier than product
        H(i)=H(i-1)+(t0-t(i))*x(i)*h;
        t(i+1)=t0;
        Event_type=[Event_type;'t0'];
        u1=rand(1);
        if u1<0.7               %Generate demanding
            D(i)=1;
        else if u1<0.9
```

```
            D(i)=2;
            else if u1<0.98
                    D(i)=3;
                else D(i)=4;
                end
            end
        end
        if D(i)<x(i)        %Compare demand and current inventory
            w=D(i);
            x(i+1)=x(i)-w;
            Loss(i)=0;
        else
            w=x(i);
            Loss(i)=(D(i)-x(i))*2;
            x(i+1)=0;
        end

        R(i)=R(i-1)+w*r-Loss(i);

        if x(i+1)<s&&y==0 %Situation that needs replenishment
            y=S-x(i+1);
            t1=t(i+1)+L;
            d=d+1;
        end
        C(i)=C(i-1);
        u=rand(1);
        t0=-(1/8)*log(u)+t(i+1);

    else %if t0>t1:            Product arrives earlier than customer

        H(i)=H(i-1)+(t1-t(i))*x(i-1)*h;
        t(i+1)=t1;
        Event_type=[Event_type;'t1'];
        R(i)=R(i-1);
        C(i)=C(i-1)+c*y;
        x(i+1)=x(i)+y;
        y=0;
        t1=Inf;
        D(i)=0;
    end
    Profit(i)=R(i)-C(i)-H(i)-d*Dv;
    i=i+1;
end
```

```
%adjust the length of serieses
D(i)=0;
R(i)=R(i-1);
Profit(i)=Profit(i-1);
Loss(i)=0;
Event_type=[Event_type;'t0'];

t=t';
R=R';
D=D';
Loss=Loss';
x=x';
Profit=Profit';

results=table(t,D,x,R,Profit,Loss,Event_type);
results.Properties.VariableNames =
{'Time','Demand','Inventory_Level','Revenue','Profit','Loss','Event_type'}
```

```
%Output:
%Net Profit:
fprintf('Net Profit = %f\n',Profit(i));
%average net profit:
fprintf('Average Net Profit = %f\n',Profit(i)/t(i-1));
%inventory:
subplot(2,2,1)
plot(t,x)
title 'Inventory'
%demand:
subplot(2,2,2)
plot(t,D)
title 'Demand'
%revenue:
subplot(2,2,3)
plot(t,R)
title 'Revenue'
%profit:
subplot(2,2,4)
plot(t,Profit)
title 'Profit'
```

Part B

```
%Set Initial Values
%Time:
T=30;                                   %limit time
t = [];t(1)=0;                          %current time
L1=0.1;L2=2;                            %Delivery time
t0 = -(1/8)*log(rand(1));
%The arrival time of one customer (initial value)
t1=t0+L1;                               %The arrival time of product(initial value)

%Quantity:
x1=0;x2=0;
X1 = [x1];                              % Remaining Amount of Products in Store
X2 = [x2];                              % Remaining Amount of Products at Depot
d = [0];                                % Demand of Each Customer
s1=20; S1=100;
% Minimum & Maximum inventory level for store
s2=50; S2=300;
% Minimum & Maximum inventory level for depot

%Cost price
r=200;                                  % sell price/unit
c=50;                                   % cost price/unit
K1=5;K2=5;                              % delivery cost
h1=2;h2=0.5;                            % store price/unit

%Cost series
R = [0];                                % Net Profit
cost=0;                                 % Cost Price
Loss=[];
% Money that should have been earned is not earned
y = 0;                                  % Quantity ordered but not delivered

%Counters
i=1;j=0;
Event_type=['t0'];
```

```
while t(i) <= T
% T is the Length of Time Period

    if X2(i)> s2
    % Depot can deliver product to store; SAME AS Problem 1

        if t0<t1    % Customers arrived before products
            t(i+1)=t0;
            Event_type=[Event_type;'t0'];
            d(i) = randsample(4,1,true,[.7,.2,.08,.02]);
            %demand of customer
            X2(i+1)=X2(i);
            if d(i)<X1(i)
            % Demand Compared with the Storage;
            % There is no loss
                w=d(i);
            % Amount of products that sell to customers
                X1(i+1)=X1(i)-w; % Update the storage of store
                Loss(i)=0;
            else                        % There exists loss
                w=X1(i);
                Loss(i)=(d(i)-X1(i))*2;
                X1(i+1)=0;
            end
            R(i+1) = R(i) + w * r - (X1(i) * h1 - X2(i) * h2)*(t(i+1)-t(i)); %Total revenue

            if X1(i)<s1 && y==0
            y=S1-X1(i);                 % Amount that product make up
            X2(i+1) = X2(i)-y;      % Update the storage of depot
            t1=t(i+1)+L1;               % Time when products arrived at store
            end

        else
        % if t0>t1; Customers arrived after products arrived
            if X1(i)<s1 && y==0
                y=S1-X1(i);         % Amount that product make up
                X2(i+1) = X2(i)-y;
                % Update the storage of depot
            else
                X2(i+1)=X2(i);
            end
            t(i+1)=t0+L1;
            Event_type=[Event_type;'t1'];
            R(i+1)=R(i);
            X1(i+1)=S1;
            cost = c*y;
            y=0;
            t1=Inf;
            d(i)=0;
            R(i+1) = R(i) - (X1(i) * h1 - X2(i) * h2)*(t(i+1)-t(i))-K1-cost;
        end

    else
    % If x2<s2, depot cannot satisfy store
        j = j+1;
        if j ==1
            t_wait = t(i);
        % Record the time when there are lack of product
        end
        if t(i) < t_wait+L2          % Still no product in depot
            t(i+1) = t0;
            Event_type=[Event_type;'t0'];
            % Update the time, since store will not demand product from depot,
            % There will be no t1
            X2(i+1)=X2(i);
            d(i) = randsample(4,1,true,[.7,.2,.08,.02]);
            if X1(i)>d(i)
            % Storage in store satisfy customers' demand
                X1(i+1) = X1(i) - d(i);
            % Remaining Amount of Products in Store
                R(i+1) = R(i) + d(i) * r - (X1(i) * h1 - X2(i) * h2)*(t(i+1)-t0);
            else
                X1(i+1)=0;
                Loss(i) = (d(i)-X1(i))*2;
                R(i+1) = R(i) + X1(i) * r - (X1(i) * h1 - X2(i) * h2)*(t(i+1)-t0)-
Loss(i);
            end
        else        % Product have already arrived at depot
            t(i+1)=t(i);
            Event_type=[Event_type;'t2'];
            X2(i+1) = S2;            % Update the storage of depot
            j = 0;
            X1(i+1)=X1(i);
            R(i+1) = R(i)-K2-c*(S2-X2(i));
            t1=t(i)+L1;
        end

    end
    t0 = -(1/8)*log(rand(1))+t(i+1);
    i = i+1;
end
```

```
d(i)=0;
Loss(i)=0;
t=t';
R=R';
Loss=Loss';
X1=X1';
X2=X2';
d=d';
T=table(t,d,Event_type,X1,X2,R,Loss);
T.Properties.VariableNames =
{'Time','Demand','Event_type','Inventory_Store','Inventory_Depot','Profit','Loss'}
```

```
% Demand
subplot(2,2,1)
plot(t,d)
title 'Demand'
% Remaining Amount of Products in Store
subplot(2,2,2)
plot(t,X1)
title 'Remaining Amount of Products in Store'
% Remaining Amount of Products at Depot
subplot(2,2,3)
plot(t,X2)
title 'Remaining Amount of Products at Depot'
% Profit
subplot(2,2,4)
plot(t,R)
title 'Profit'
```