STAT3004 Project 1

Question 1

(a)

Equation (4.1.2) gives the transition probabilities in the Greenwood model:

$$p_{(x,y)_t(x,y)_{t+1}} = \mathbb{P}\{(X,Y)_{t+1} = (x,y)_{t+1} | (X,Y)_t = (x,y)_t\} = \binom{x_t}{x_{t+1}} \alpha^{x_{t+1}} (1-\alpha)^{x_t - x_{t+1}}$$

Where $0 \leq \alpha \leq 1$ and $x_{t+1} \in \{0, 1, ..., x_t\}$. Note that these transition probabilities do not explicitly depend on y_t or y_{t+1}. Recognising the binomial form as $\text{Bin}(x_t, \alpha)$, which has well-known expectation, we get:

$$\mathbb{E}[X_{t+1}|X_t = x_t] = x_t\alpha$$

Or, equivalently:

$$\mathbb{E}[X_t|X_{t-1} = x_{t-1}] = x_{t-1}\alpha$$

We note then that in the first step we have:

$$\mathbb{E}[X_t|X_{t-1} = x_{t-1}] = \mathbb{E}[X_1|X_0 = x_0] = \alpha x_0$$

Now to use these two single-step results we notice:

$$\mathbb{E}[X_t|X_0 = x_0] = \mathbb{E}[\mathbb{E}[X_t|X_{t-1}, X_0 = x_0] | X_0 = x_0]$$

Which by the Markov property may be written:

$$\mathbb{E}[X_t|X_0 = x_0] = \mathbb{E}[\mathbb{E}[X_t|X_{t-1}] | X_0 = x_0]$$

Substituting our result for $\mathbb{E}[X_t|X_{t-1}]$:

$$\begin{aligned}\mathbb{E}[X_t|X_0 = x_0] &= \mathbb{E}[\alpha X_{t-1}|X_0 = x_0] \\ &= \alpha\mathbb{E}[X_{t-1}|X_0 = x_0]\end{aligned}$$

So we now have a recursive relation for $\mathbb{E}[X_t|X_0 = x_0]$. Now noting our result for $[X_1|X_0 = x_0]$ we get:

$$\begin{aligned}\mathbb{E}[X_t|X_0 = x_0] &= \alpha\mathbb{E}[X_{t-1}|X_0 = x_0] \\ &= \alpha^{t-1}\mathbb{E}[X_1|X_0 = x_0] \\ &= \alpha^t x_0\end{aligned}$$

Which is the desired expression.

To derive $\mathbb{E}\left[Y_t | X_0 = x_0\right]$ we note that Y_t may by definition be expressed:

$$Y_t = X_{t-1} - X_t$$

And so we get:

$$\mathbb{E}\left[Y_t | X_0 = x_0\right] = \mathbb{E}\left[X_{t-1} - X_t | X_0 = x_0\right] = \mathbb{E}\left[X_{t-1} | X_0 = x_0\right] - \mathbb{E}\left[X_t | X_0 = x_0\right]$$

Substituting in our derived expression for $\mathbb{E}\left[X_t | X_0 = x_0\right]$:

$$\mathbb{E}\left[Y_t | X_0 = x_0\right] = \alpha^{t-1} x_0 - \alpha^t x_0 = \alpha^{t-1}(1-\alpha) x_0$$

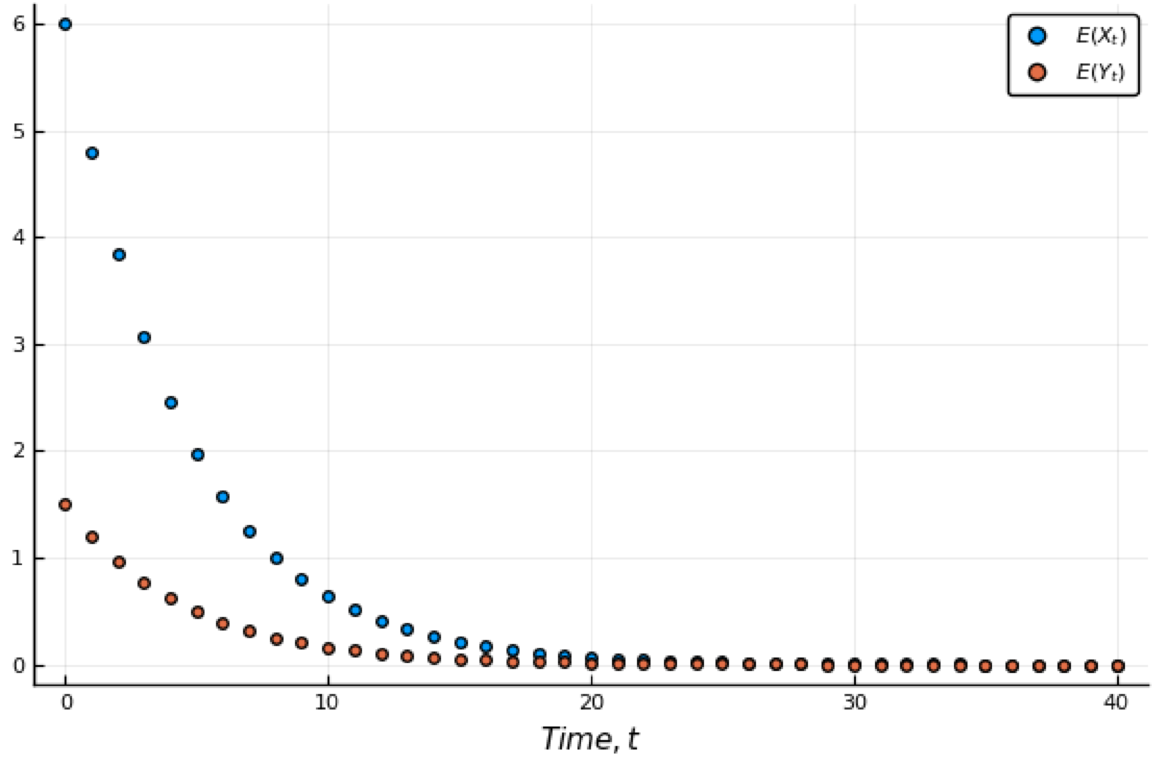
Which is the desired expression.

(b)

The following plot shows the expected values of X_t and Y_t until $t = 40$ for $x_0 = 6$ and $\alpha = 0.8$.

```
In [3]: using LaTeXStrings, LinearAlgebra, Statistics, StatsBase, Plots, Distributions; pyplot()
        x0 = 6; a = 0.8; t = 0:40;#The x_0 and alpha values, and time range
        ex = x0*a.^t; ey = x0*(1-a)*a.^(t-ones(size(t)));
        pyplot()
        scatter(t,ex,label=L"E(X_t)")
        scatter!(t,ey,label=L"E(Y_t)")
        xlabel!(L"Time,t")
```

```
Out[3]:
```



Question 2

(a)

Equation (4.1.3) gives the transition probabilities for the Reed-Frost model:

$$p_{(x,y)_t(x,y)_{t+1}} = \binom{x_t}{x_{t+1}} \left(\alpha^{y_t}\right)^{x_{t+1}} \left(1-\alpha^{y_t}\right)^{x_t - x_{t+1}} \tag{1}$$

Where we have substituted $y_{t+1} = x_t - x_{t+1}$ into the expression. This is easily recognised to be binomially distributed as $\mathrm{Bin}(x_t, \alpha^{y_t})$ which has well-known expectation for X_{t+1}:

$$\mathbb{E}(X_{t+1} | (X,Y)_t = (x,y)_t) = x_t \alpha^{y_t} \tag{2}$$

We can then note that $Y_{t+1} = X_t - X_{t+1}$ and so:

$$\begin{aligned} \mathbb{E}(Y_{t+1} | (X,Y)_t = (x,y)_t) &= \mathbb{E}(X_t - X_{t+1} | (X,Y)_t = (x,y)_t) \qquad (3) \\ &= x_t - \mathbb{E}(X_{t+1} | (X,Y)_t = (x,y)_t) \qquad (4) \\ &= x_t - x_t \alpha^{y_t} = x_t(1-\alpha^{y_t}) \qquad (5) \end{aligned}$$

Combining these two results gives:

$$\mathbb{E}((X,Y)_{t+1} | (X,Y)_t = (x,y)_t) = (x_t \alpha^{y_t}, x_t(1-\alpha^{y_t}))$$

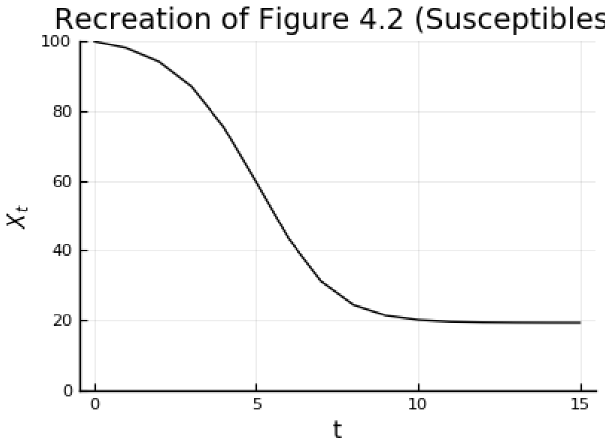
Which is the desired expression.

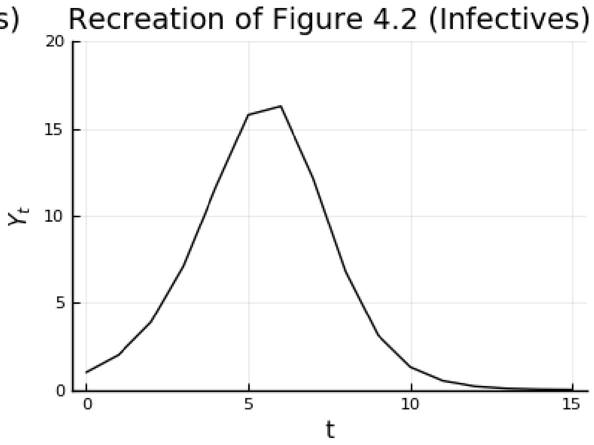
(b)

The following plot recreates Figure 4.2, which shows the expected number of susceptibles, X_t, and infectives, Y_t, for times $0 \le t \le 15$ as calculated recursively from equations (4.2.1) with $x_0 = 100$, $y_0 = 1$, $\alpha = 0.98$.

```
In [17]: x0 = 100; y0 = 1; a=0.98 #initial conditions and alpha
         t = 0:15; #time range
         xy = zeros(length(t), 2); #empty array to store X_t and Y_t values
         xy[1,:] = [x0,y0];
         for j in 2:length(t)
             xt,yt = xy[j-1,:];
             xy[j,:] = [xt*a^yt,xt*(1-a^yt)];
         end
         p1 = plot(t,xy[:,1],yaxis=(L"X_t", (0, 100), 0:20:100))
         p2 = plot(t,xy[:,2],yaxis=(L"Y_t", (0, 20), 0:5:20))
         plot(p1,p2,layout = 2,legend=false, linecolor =:black,xaxis=("t"),size=(800,300),title=["
```

```
Out[17]:
```

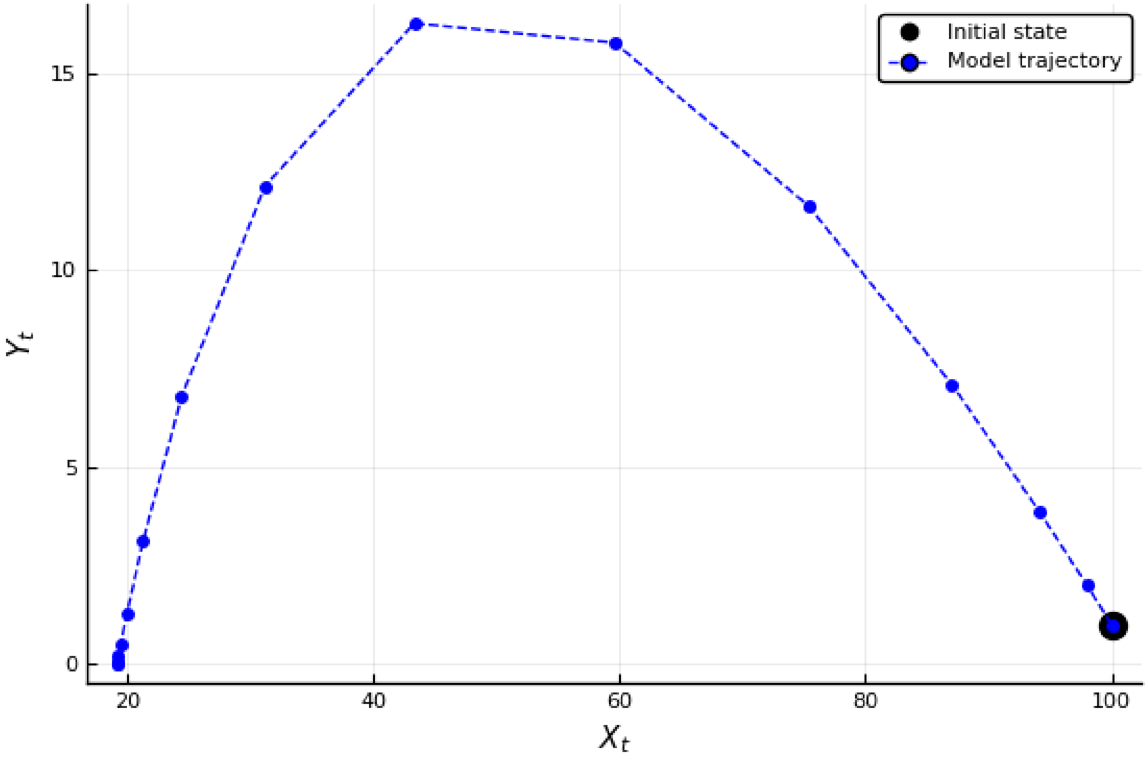




The following plot shows the trajectory of expected values for the Reed-Frost model with $x_0 = 100$, $y_0 = 1$ and $\alpha = 0.98$, as for for the component-wise plots above.

In [4]:
```
#mark the initial state
scatter([xy[1,1]], [xy[1,2]],c=:black, ms=10,label="Initial state")
#plot the trajectory from the values calculated to recreate figure 4.2
plot!(xy[:,1],xy[:,2], c=:blue, ls=:dash, m=(:dot, 5, Plots.stroke(0)),label="Model traject
```

Out[4]:



Question 3

(a)

The transition probability matrix from equation (4.1.5) is:

$$\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1-\alpha & \alpha & 0 & \cdots & 0 \\ (1-\alpha)^2 & 2(1-\alpha)\alpha & \alpha^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (1-\alpha)^{x_0} & x_0(1-\alpha)^{x_0-1}\alpha & \binom{x_0}{2}(1-\alpha)^{x_0-2}\alpha^2 & \cdots & \alpha^{x_0} \end{pmatrix}$$

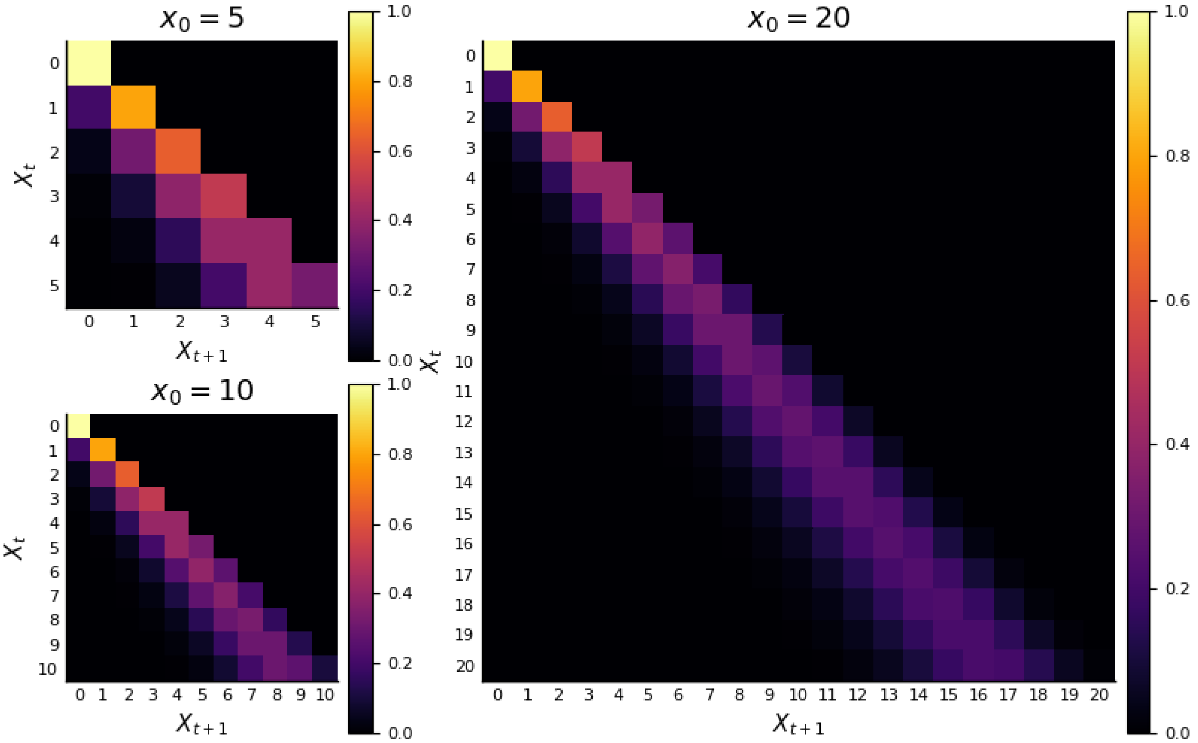
i.e. element P_{ij} of this matrix is given by:

$$P_{ij} = \begin{cases} \binom{i}{j}(1-\alpha)^{i-j}\alpha^j, & j \leq i \\ 0, & j > i \end{cases}$$

Where we have indexing starting from 0. This corresponds to the transition probabilties given in (4.1.2) with $x_t = i$ and $x_{t+1} = j$. The following heatmaps are of this transition probability matrix for $x_0 \in \{5, 10, 20\}$ and $\alpha = 0.8$.

```
In [2]: function makeP(x0,a)
            #Constructs the matrix of transition probabilities for x0 initial susceptibles and alp
            P = zeros(x0+1,x0+1);
            for i in 0:x0
                for j in 0:i
                    P[i+1,j+1] = binomial(i,j)*((1-a)^(i-j))*(a^j);
                end
            end
            return P
        end

        function makeHM(x0,a)
            #Creates a heatmap of the transition matrix
            P = makeP(x0,a)
            hm = heatmap(P,yflip=:true,xticks=(1:x0+1,0:x0),yticks=(1:x0+1,0:x0));
            return hm
        end

        x0s = [5,10,20]; a= 0.8;
        hm1 = makeHM(5,a);
        hm2 = makeHM(10,a);
        hm3 = makeHM(20,a);
        l = @layout [[a; b] c{0.7w}]

        plot(hm1,hm2,hm3, layout=l,xlabel=(L"X_{t+1}"), ylabel=(L"X_t"),size =(800,500),title=[L"x
```

Out[2]:



(b)

Since the number of susceptibles from time t to time $t+1$ can never increase, we have:

$$\mathbb{P}_i(X_n = j \text{ for some } n \geq 0) = 0 \; \forall j > i, \forall i \in [0, x_0]$$

i.e. the state i does not lead to any state $j > i$ and so the state i does not communicate with any state $j > i$. Thus the communicating classes of this Markov chain are the sets $\{i\} \forall i \in [0, x_0]$ - each state only communicates with itself. There are $x_0 + 1$ of these classes in the chain.

However, since for $i > 0$ there is a non-zero probability of the number of susceptibles decreasing, i.e.

$$\mathbb{P}_i(X_n = j \text{ for some } n \geq 0) > 0 \; \forall j < i, \forall i \in [1, x_0]$$

Thus we have that the state i leads to all states $j < i$, but for all those states j does not lead to i (since $i > j$). So the probability of returning to state i from state i, ρ_{ii}, is less than one for $i \in [1, x_0]$, so these states (and the classes they define) are transient. Only for $i = 0$ do we have $\rho_{00} = 1$, since there are no lower states, and so only the class $\{0\}$ is recurrent.

(c)

The following table shows the expected values calculated using equation (4.1.4) to plot the graph in question 1(b) (values in the middle column), as well as the same quantities calculated directly from the formula (values in the right column):

$$\mathbb{E}(X_t | X_0 = x_0) = e_{x_0+1}^T P^t v$$

Where e_{x_0+1} is the $x_0 + 1$ unit vector $[0, 0, ...0]^T$ and v is the vector $[0, 1, 2, ...x_0]^T$. The left column contains the times t at which these expected values are calculated, you can see that these two methods agree numerically.

x_0	From Equation (4.1.4)	From Matrix Formula
0	6.0	6.0
1	4.8	4.8
2	3.84	3.84
3	3.072	3.072
4	2.4576	2.4576
5	1.96608	1.96608
6	1.57286	1.57286
7	1.25829	1.25829
8	1.00663	1.00663
9	0.80531	0.80531
10	0.64425	0.64425

This matrix formula for the expected value holds because P^t is the $t-step$ transition probability matrix. i.e. the elements of P^t give:

$$\mathbb{P}(X_t = j | X_0 = i)$$

Where here we take i, j to be states in $[0, x_0]$. Multiplying this matrix by $e^T_{x_0+1}$ extracts the final row of probabilities - i.e. the row which represents:

$$\mathbb{P}(X_t = j | X_0 = x_0)$$

We do this because we know that $X_0 = x_0$, so the other rows of P^t are unnecessary. We now have a row vector which is the probability distribution of X_t over $[0, x_0]$ given that $X_0 = x_0$ multiplied by the column vector of states $v = [0, 1, ..., x_0]^T$. This product multiplies each state by the probability of being in that state and adds them together, i.e. it performs the expectation calculation:

$$\mathbb{E}(X_t | X_0 = x_0) = \sum_{j=0}^{x_0} j\mathbb{P}(X_t = j | X_0 = x_0) = (e^T_{x_0+1} P^t) v$$

```
In [5]: x0 = 6; a = 0.8; t = 0:10;#x_0 value, alpha value and time range

        #Expected Values from equation (4.1.4)
        ex1 = x0*a.^t;
        #Array to store expected values from matrix formula
        ex2 = zeros(size(ex1));

        P = makeP(x0,a);#The transition probability matrix
        eT = zeros(1,x0+1);
        eT[length(eT)] = 1;
        v = zeros(x0+1,1);
        for j in 1:length(v)
            v[j] = j-1;
        end

        #Expected values from matrix formula
        for tt in t
            ex2[tt+1] = (eT*(P^tt)*v)[1];
        end

        digits = 5
        ex1 = [round(x,digits =digits) for x in ex1];
        ex2 = [round(x,digits =digits) for x in ex2];
        ts = [t for t in t];
        compareEX = hcat(ts,ex1, ex2)
```

```
Out[5]: 11×3 Array{Float64,2}:
          0.0  6.0      6.0
          1.0  4.8      4.8
          2.0  3.84     3.84
          3.0  3.072    3.072
          4.0  2.4576   2.4576
          5.0  1.96608  1.96608
          6.0  1.57286  1.57286
          7.0  1.25829  1.25829
          8.0  1.00663  1.00663
          9.0  0.80531  0.80531
         10.0  0.64425  0.64425
```

(d)

As for question 1 when deriving equations (4.1.4), we make use of the fact that $Y_t = X_{t-1} - X_t$, and so we can use our logic from part 3(c) to get:

$$\begin{align} \mathbb{E}(Y_t|X_0 = x_0) &= \mathbb{E}(X_{t-1} - X_t|X_0 = x_0) \tag{6} \\ &= \mathbb{E}(X_{t-1}|X_0 = x_0) - (X_t|X_0 = x_0) \tag{7} \\ &= e^T_{x_0+1}P^{t-1}v - e^T_{x_0+1}P^t v \tag{8} \\ &= e^T_{x_0+1}(P^{t-1} - P^t)v \tag{9} \end{align}$$

Where $e^T_{x_0+1}$, P and v are defined as before. We can then repeat the comparison process for Y_t, again the left column is the time, the middle column was calculted using (4.1.4) and the right column was calculated using the above matrix formula. It is clear that the two methods agree numerically.

x_0	From Equation (4.1.4)	From Matrix Formula
0	1.5	1.5
1	1.2	1.2
2	0.96	0.96
3	0.768	0.768
4	0.6144	0.6144
5	0.49152	0.49152
6	0.39322	0.39322
7	0.31457	0.31457
8	0.25166	0.25166
9	0.20133	0.20133
10	0.16106	0.16106

```
In [7]: x0 = 6; a = 0.8; t = 0:10;#x_0 value, alpha value and time range

        #Expected Values from equation (4.1.4)
        ey1 = x0*(1-a)*a.^(t-ones(size(t)));
```

```
#Array to store expected values from matrix formula
ey2 = zeros(size(ey1));

P = makeP(x0,a);#The transition probability matrix
eT = zeros(1,x0+1);
eT[length(eT)] = 1;
v = zeros(x0+1,1);
for j in 1:length(v)
    v[j] = j-1;
end

#Expected values from matrix formula
for tt in t
    ey2[tt+1] = (eT*(P^(tt-1) - P^tt)*v)[1];
end

digits = 5
ey1 = [round(x,digits =digits) for x in ey1];
ey2 = [round(x,digits =digits) for x in ey2];
ts = [t for t in t];
compareEX = hcat(ts,ey1, ey2)
```

```
Out[7]: 11×3 Array{Float64,2}:
          0.0  1.5      1.5
          1.0  1.2      1.2
          2.0  0.96     0.96
          3.0  0.768    0.768
          4.0  0.6144   0.6144
          5.0  0.49152  0.49152
          6.0  0.39322  0.39322
          7.0  0.31457  0.31457
          8.0  0.25166  0.25166
          9.0  0.20133  0.20133
         10.0  0.16106  0.16106
```

Question 4

(a)

Equation (4.1.6) is:

$$p_j^t = \mathbb{P}(X_t = j, Y_t > 0) = \sum_{i=j+1}^{x_0-(t-1)} p_i^{t-1} p_{ij}$$

Where $p_i^0 = \delta_{x_0 i}$. Each of the terms in the sum are of the form: $p_i^{t-1} p_{ij}$. p_i^{t-1} may be expressed in words as "the probability that there are i susceptibles in time $t-1$ and there are a non-zero number of infectives", while p_{ij} is "the probability that there will be j susceptibles in timestep t given that there are i susceptibles in timestep $t-1$". Thus each term in the sum is giving the probability that there will be j susceptibles in time t, i susceptibles in time t-1 and a non-zero number of infectives in time $t-1$ (i.e. the infection can still continue in time t). The lower bound on the sum ensures that $Y_t > 0$: starting i from $j+1$ means that i is larger than j and hence $Y_t = X_{t-1} - X_t = i - j > 0$. While the upper bound on the sum ensures that the infection will be ongoing in time $t-1$: in order for the infection to not have ended prior to or during time $t-1$, there must be a decrease of at least one susceptible per timestep, for a total of at least $t-1$ suceptibles removed by timestep $t-1$ and hence the upper bound on j of $x_0-(t-1)$. Thus this equation sums over all the allowed states i the probabilities of being in state i in timestep $t-1$ and then moving from state i to state j in timestep t i.e. it gives the probability of $(X_t = j)$ given the bounds on X_{t-1} which ensure both that the infection is ongoing in timestep $t-1$ and that $Y_t > 0$.

(b)

The following computation uses the recursive relationship $\Gamma(k,n|x_0) = p^{n-1}_{x_0-k}\alpha^{x_0-k}$ to compute $\mathbb{P}(W>4)$ for the Greenwood model with $x_0 = 6$ and $\alpha = 0.8$. It is clear that there are bounds on the values that k, n can take:

$$k \in [0, x_0],\ n \in [1, x_0+1]$$

The bounds on k are clear since we can have at most x_0 infected if we started with x_0 susceptibles. The bounds on n arise from the condition that the infection stops when $Y_t = 0$ for some t, i.e. in order for the infection to continue, at least one susceptible must be infected in each time step, so the infection must stop when there are no more susceptibles to infect which occurs at $t = x_0+1$ at the latest. We can then see:

$$\begin{aligned}
\mathbb{P}(W>4) &= \mathbb{P}(W=5) + \mathbb{P}(W=6) && (10)\\
&= \sum_{k\in\{4,5\}} \sum_{n} \Gamma(k,n|X_0=6) && (11)\\
&= \sum_{k\in\{4,5\}} \sum_{n=1}^{k+1} p^{n-1}_{x_0-k}\alpha^{x_0-k} && (12)
\end{aligned}$$

Where, again, we note that if there are a total of k infected then the longest the infection could last is $k+1$ time steps. The following calculation finds that:

$$\mathbb{P}(W>4) \approx 0.13029836$$

```
In [3]: x0 = 6; a = 0.8;#x_0 value and alpha value
        store = Dict{Tuple{Int64,Int64},Float64}();#dictionary to store previously calculated valu
        P = makeP(x0,a);#the transition probability matrix

        function recursiveGen(j,t)
            #recursive function which generates the p^t_j values
            if !haskey(store, (j,t))
            #check if p^t_j has already been calculated
                if t == 0
                #termination condition for the recursion
                    if j == x0
                        store[(j,t)] = 1
                    else
                        store[(j,t)] = 0
                    end
                else
                #recursive step
                    temp = 0
                    for i in (j+1):(x0-(t-1))
                        temp += recursiveGen(i,t-1)*P[i+1,j+1]
            end
            store[(j,t)] = temp
        end
    end
    return store[(j,t)]
end

prob = sum(sum(recursiveGen(x0-k,n-1)*a^(x0-k) for n in 1:k+1) for k in 5:6);

println("The probability (from recursion) that W > 4 is ", prob)
```

```
The probability (from recursion) that W > 4 is 0.13029835750312502
```

(c)

Now performing the same calculation via Monte-Carlo simulation (using Julia's Distributions.Binomial function), we get

$$\mathbb{P}(W > 4) \approx 0.130687$$

Which is approximately equal to the probability found in part (b).

```
In [4]: x0 = 6; a=0.8;; #x_0 value, alpha value
        N=10^6;#Number of Monte-Carlo simulations to run

        function checkW(x0,a)
            #Runs a Monte-Carlo simulation of a population of x initial susceptibles under the Gre
            #non-zero number of initial infectives (not counted in W, by definition). Returns 1 if
            w = 0
            x = x0
            y = true
            while y>0
                xNew = rand(Binomial(x,a))
                y = x - xNew
                w += y
                x = xNew
            end
            if w > 4
                return 1
            else
                return 0
            end
        end

        MCSim = [checkW(x0,a) for _ in 1:N]
        est = mean(MCSim)
        println("The estimated probability (from Monte-Carlo) that W > 4 is ", est)
```

```
The estimated probability (from Monte-Carlo) that W > 4 is 0.130687
```

(d)

For such a chain with $X_0 = x_0$ and given α, we have transition probabilitiy matrix P with elements given by:

$$P_{ij} = \binom{i}{j}\alpha^j(1-\alpha)^{i-j},\ j \leq i,\ 0 \leq i,j \leq x_0$$

P can be decomposed into its diagonal (Q) and off-diagonal ($\overline{P}$) parts:

$$P = \overline{P} + Q$$

So we get that

$$Q = \text{diag}(1, \alpha, ..., \alpha^{x_0})$$

and

$$\overline{P}_{ij} = \binom{i}{j}\alpha^j(1-\alpha)^{i-j},\ j < i,\ 0 \leq i \leq x_0$$

$\overline{P}$ can be considered the transition probability matrix when we are confined to decreasing-susceptibles transitions, while Q can be considered the transition probability matrix when we are confined to same-state transitions. It is clear that for the epidemic to end at time $T = t$, we must have $t-1$ consecutive decreasing-state transitions followed by a single same-state transition at time t. So the t-step transition matrix given by $\bar{P}^{t-1}Q$ denotes the probabilities of the epidemic ending at time $T = t$ given that $X_0 = i$ and $X_T = X_t = j$:

$$\mathbb{P}(X_0 = i, X_t = j, T = t) = \left(\bar{P}^{t-1}Q\right)_{ij}$$

Since we know that $X_0 = x_0$, we can extract only the final row of this matrix by left-multiplying by $A^T = (0, 0, ..., 1)$. Similarly, we are uninterested in what the final state is so we get:

$$\mathbb{P}(T = t | X_0 = x_0) = \sum_j \mathbb{P}(X_t = j, T = t | X_0 = x_0) = \sum_j \left(\bar{P}^{t-1}Q\right)_{x_0 j} = A'\bar{P}^{t-1}QE$$

Where $E^T = (1, 1, ..., 1)$. Then by definition of the probability generating function:

$$\Psi_T(\theta) = \mathbb{E}\theta^T = \sum_{t=1}^{\infty} \theta^t \mathbb{P}(T = t | X_0 = x_0) = A' \left(\sum_{t=1}^{x_0+1} \overline{P}^{t-1}\theta^{t-1} \right) \theta QE$$

We are now interested in the joint pgf of (W, T):

$$\Psi_{W,T}(\varphi, \theta) = \mathbb{E}(\varphi^W \theta^T) = \sum_{t=1}^{\infty} \sum_{w=0}^{\infty} \mathbb{P}(W = w, T = t | X_0 = x_0)\varphi^w \theta^t$$

But we note that for a given transition i to j, the increase in W is $j - i$. So we can modify $\overline{P}$ to give the pgf of the number of new infectives in a single step by multiplying each element $\overline{P}_{ij}$ by φ^{j-1}. This is equivalent to multiplying the kth subdiagonal of $\bar{P}$ by φ^k. Denote this modified matrix $\overline{P}(\varphi)$. It is clear that the kth subdiagonal of $\overline{P}^t(\varphi)$ will also be equivalent to the kth subdiagonal of $\overline{P}$ multiplied by φ^k, which makes sense since there were $j - i = k$ new infectives in those t steps. Subsituting this modified matrix in our expression for $\Psi_T(\theta)$ gives:

$$\Psi_{W,T}(\varphi, \theta) = A' \left(\sum_{t=1}^{x_0+1} \left(\overline{P}(\varphi)\right)^{t-1} \theta^{t-1} \right) \theta QE$$

It is clear that $\Psi_{W,T}(\varphi, 1) = \Psi_W(\varphi)$ from substituting $\theta = 1$ in our definition of the joint pgf. Substituting this in above gives:

$$\Psi_{W,T}(\varphi, 1) = \Psi_W(\varphi) = A' \left(\sum_{t=1}^{x_0+1} \left(\overline{P}(\varphi)\right)^{t-1} \right) QE$$

We now consider our specific case of $X_0 = x_0 = 6$ and $\alpha = 0.8$. The probability that $W > 4$ is thus equal to:

$$\mathbb{P}(W > 4 | X_0 = 6) = \mathbb{P}(W = 5 | X_0 = 6) + \mathbb{P}(W = 6 | X_0 = 6) = \frac{\Psi_W^{(5)}(0)}{5!} + \frac{\Psi_W^{(6)}(0)}{6!}$$

As previously noted, the elements of $\overline{P}^{t-1}(\varphi)$ are merely the elements of $\overline{P}^{t-1}$ multiplied by φ^k on the kth subdiagonal for all k. Thus we have that:

$$\left(\frac{d^k \left(\overline{P}(\varphi)\right)^{t-1}}{d\varphi^k} \right)_{ij} = \left(\overline{P}^{t-1}\right)_{ij} \frac{d^k \varphi^{j-i}}{d\varphi^k}$$

So clearly when $\varphi = 0$ the only non-zero elements are on the kth subdiagonal and are given by:

$$\left(\frac{d^k \left(\overline{P}(0)\right)^{t-1}}{d\varphi^k} \right)_{ij} = \begin{cases} \left(\overline{P}^{t-1}\right)_{ij} k!, & i - j = k \\ 0, & \text{otherwise} \end{cases}$$

For ease of notation we can define:

$$B_{ij}^{kt} = \begin{cases} \left(\overline{P}^{t-1}\right)_{ij}, & i - j = k \\ 0, & \text{otherwise} \end{cases}$$

Where k, t are superscripts, not powers. So we get:

$$\Psi^{(k)}(0) = A' \left(\sum_{t=1}^{x_0+1} B^{kt} k! \right) QE \tag{13}$$

$$\Rightarrow \mathbb{P}(W = k) = \frac{\Psi^{(k)}(0)}{k!} \tag{14}$$

$$= A' \left(\sum_{t=1}^{x_0+1} B^{kt} \right) QE \tag{15}$$

Combining with out expression for $\mathbb{P}(W > 4)$ gives:

$$\mathbb{P}(W > 4) = \frac{\Psi^{(5)}(0)}{5!} + \frac{\Psi^{(6)}(0)}{6!} \tag{16}$$

$$= A' \left(\sum_{t=1}^{7} B^{5t} \right) QE + A' \left(\sum_{t=1}^{7} B^{6t} \right) QE \tag{17}$$

$$= A' \left(\sum_{t=1}^{7} \left(B^{5t} + B^{6t} \right) \right) QE \tag{18}$$

The following computation calculates this result:

$$\mathbb{P}(W > 4) \approx 0.13029836$$

It is clear that this agrees with the computation from part 4(b).

```
In [5]: x0 = 6; a = 0.8;#x_0 value and alpha values
        P = makeP(x0,a)#transition probability matrix
        Q = Diagonal(diag(P))
        Pbar = P - Q
        AT = zeros(1,x0+1)
        AT[length(AT)] = 1
        E = ones(x0+1,1)

        function makeB(k,t)
            #Creates the matrix (B^{kt}) in the notation used above
            temp = Pbar^(t-1)
            B = zeros(size(Pbar))
            for i in k+1:x0+1
                B[i,i-k] = temp[i,i-k]
            end
            return B
        end

        temp = zeros(size(Pbar))
        for t in 1:x0+1
            temp += makeB(5,t) + makeB(6,t)
        end
        prob = AT*temp*Q*E;
        println("The probability (from PGF) that W > 4 is ", prob[1])
```

```
The probability (from PGF) that W > 4 is 0.13029835750312502
```

Question 5

(a)

The state space is $(X_t, Y_t) \in \{\{0, 1, ..., x_0\} \otimes \{0, 1, ..., x_0\} | x + y \leq x_0\}$ for $0 < t \leq x_0 + 1$. Or, equivalently:

$$(X_{t-1}, X_t) \in \{\{0, 1, ..., x_0\} \otimes \{0, 1, ..., x_0\} | x_t \leq x_{t-1}\} \text{ for } 0 < t \leq x_0 + 1$$

(b)

We note that for a transition $(X, Y)_t = (k, i)$ to $(X, Y)_{t+1} = (l, j)$ we have two constraints on the values for which the transition probability is nonzero:

$$l \leq k \tag{19}$$

$$l + j = k \tag{20}$$

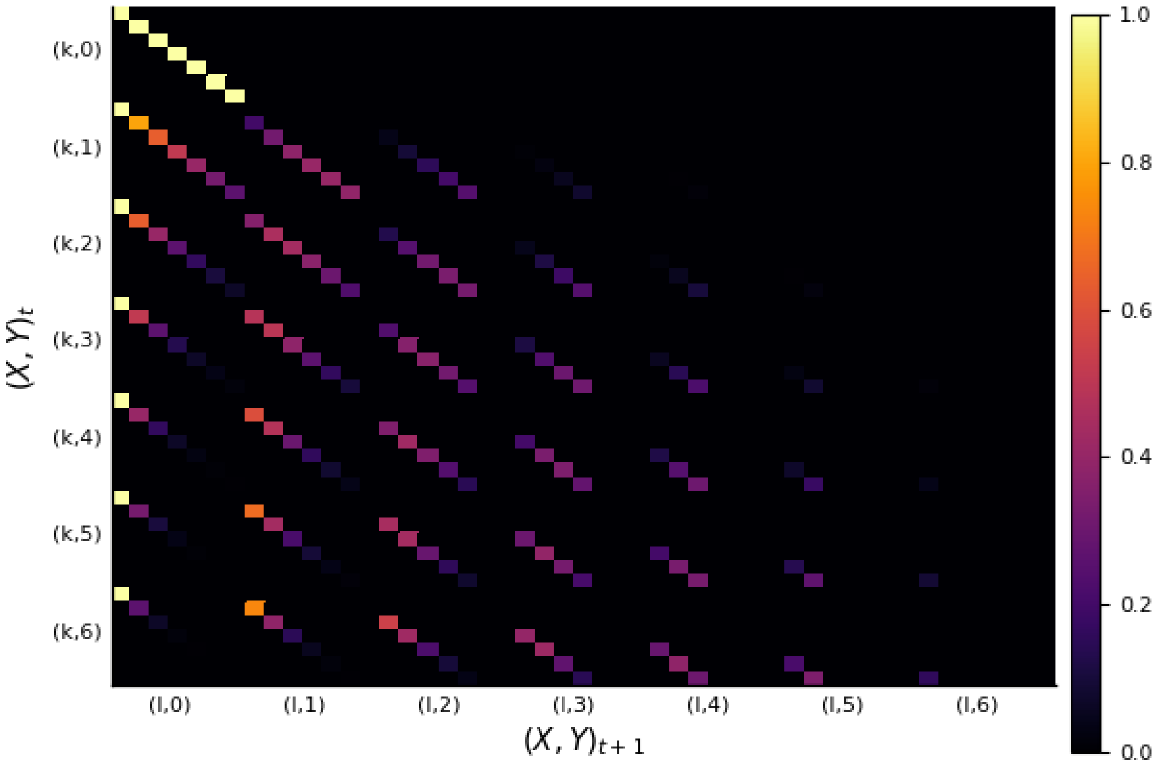
Since the X-component transition from k to l is strictly nonincreasing, we know that a state (k, i) only leads to states (l, j) with $l \leq k$ and a state (l, j) only leads to states (k, i) with $k \leq l$. Thus in order for the state (k, i) to communicate with the state (l, j) we require both $l \leq k$ and $k \leq l$ i.e. we require $l = k$. We now consider such a transition and impose the second condition that $l + j = k$. We now have both $l = k$ and $l + j = k$, which implies that $j = 0$. But a state $(l, 0)$ can only lead to the state $(l, 0)$ since $Y = 0$ and the epidemic is over. So the only communicating classes are the single-element classes $\{(X, Y) = (k, 0)\}$ for each $0 \leq k \leq x_0$. These classes are also all recurrent, since a state $(X, Y)_t = (k, 0)$ leads to itself with unit probability.

(c)

The following snippet generates a heatmap of the transition probability matrix for a Reed Frost model with $x_0 = 6$ and $\alpha = 0.8$. We note that some of these states are only accessible under this model in the initial state, where $x_0 + y_0 > x_0$ by definition.

```
In [39]: x0 = 6; a = 0.8;#x_0 value and alpha value

         using LinearAlgebra, Statistics, StatsBase, Distributions, Plots, Random, SymPy, BlockArra

         function elemRF(k,i,l,j,a)
             #Generates the transition probability under the Reed-Frost model from state (X,Y)_t =
             if j + l != k
                 return 0
             elseif l > k
                 return 0
             end
             return binomial(k,l)*((1-a^i)^j)*(a^(i*l))
         end

         function makeRFPij(x0,a,i,j)
             #Creates the Reed-Frost i,j-th transition probability submatrix for x_0 initial susce
             N = x0+1;
             Pij = zeros(N,N)
             for k in 0:x0
                 for l in 0:x0
                     Pij[k+1,l+1] = elemRF(k,i,l,j,a)
                 end
             end
             return sparse(Pij)
         end

         function makeRFP(x0,a)
             #Creates the Reed-Frost transition probability matrix for x_0 initial susceptibles an
             N = x0+1;
             indList = Int64[N for _ in 1:N];
             RFP = BlockArray{Float64}(undef_blocks,indList,indList)
             for i in 0:x0
                 for j in 0:x0
                     Pij = makeRFPij(x0,a,i,j)
                     setblock!(RFP, Pij, i+1, j+1)
                 end
             end
             return RFP
         end

         P = makeRFP(x0,a)

         heatmap(P,yflip=:true,xlabel=L"(X,Y)_{t+1}",ylabel=L"(X,Y)_t",xticks=(7*(0:x0)+3.5*ones(s
                 ["(l,0)","(l,1)","(l,2)","(l,3)","(l,4)","(l,5)","(l,6)"]),
                 yticks=(7*(0:x0)+3.5*ones(size(0:x0)), ["(k,0)","(k,1)","(k,2)","(k,3)","(k,4)","
```

Out[39]:



(d)

The following snippet runs a Monte-Carlo simulation like in part 4(c) to estimate $\mathbb{P}(W > 4)$, but for the Reed-Frost model rather than Greenwood, with $x_0 = 6$ and $\alpha = 0.8$. Note that it is assumed that the initial number of susceptibles is 1 ($y_0 = 1$), and that this initial number of susceptibles is not included in the value W. The estimated prbability is larger than that for any of the Greenwood model calculations/estimates, which makes sense since the Greenwood model for the same x_0 and α has that each suceptible has a fixed probability α of not being infected in any single time step, whereas in the Reed-Frost model this probability becomes α^{y_t}, where $y_t \geq 0$ is the number of infecteds. Since $|\alpha| < 1$ we have that $\alpha^{y_t} < \alpha$, and so the probability of any one susceptible not being infected in any one time period is smaller under the Reed-Frost model (for y_t>1) than for the Greenwood model. Thus it makes sense that a larger number of overall infected (W) would be expected and hence the probability that $W > 4$ is larger under Reed-Frost than under Greenwood.

$$\mathbb{P}(W > 4) \approx 0.2564017$$

```
In [17]: x0 = 6; a = 0.8; y0 = 1; #initial values of susceptibles and infecteds, alpha = a
         N = 10^7;#number of Monte-Carlo simulations to run

         function checkW(x,y,a)
             #Runs a Monte-Carlo simulation of a population of x initial susceptibles under the Re
             #initial infectives (not counted in W, by definition). Returns 1 if W > 4, returns 0
             w = 0
```

```
        while y>0
            xNew = rand(Binomial(x,a^y))
            y = x - xNew
            w += y
            x = xNew
        end
        if w > 4
            return 1
        else
            return 0
        end
    end

    MCSim = [checkW(x0,y0,a) for _ in 1:N]
    est = mean(MCSim)

    println("The estimated probability (from Monte-Carlo) that W > 4 is ", est)
```

```
The estimated probability (from Monte-Carlo) that W > 4 is 0.2564017
```

Report - Reed Frost Model for a Small Motel (Question 6)

The situation analysed in this report is that of a small motel being used as a quarantine site in a remote town, and how the quarantine protocol affects the number of infecteds coming in to the town. It was found that increasing the size of the motel slightly increased the total rate of infections per person into the town and significantly increased the expected rate of infections out of the motel per day, but halved the expected rate of infections out of the motel per person per day (independently of motel size greater than 2).

Scenario All new arrivals to the town must go through the motel quarantine. The motel has a maximum capcity of x_0 and there is always enough demand (enough new arrivals) on the motel for it to be full. The quarantine protocol is as follows: On day $t_i = 0$, x_0 new arrivals are accepted into the motel and tested for the virus. On day $t = t_i + 1$, the test results come back. If all x_0 tests come back negative, then the x_0 people are released into the town and a new batch of x_0 people are accepted to the motel on the same day ($t = t_i + 1$), begin protocol again with $t_{i,new} = t_{i,old} + 1$. If any of the tests come back positive, the y_i identified infecteds are removed from the motel and the remaining $x_i = x_0 - y_i$ people are tested on the same day ($t = t_i + 1$). This process continues until there are no infecteds in the motel, which can occur by one of two cases: (1) At time $t = t_i + T - 1$, all of the susceptibles have been infected, all of the tests of the remaining people in the motel return positive at time $t = t_i + T$, and all of them are removed from the motel on day $t = t_i + T$, begin protocol again with $t_{i,new} = t_{i,old} + T$; (2) At time $t = t_i + T$, infecteds identified from tests taken at time $t = t_I + T - 1$ are removed from the motel and there are no infecteds left in the motel at time $t = t_i + T$, so all of the tests of the people remaining in the motel return negative in time $t = t_i + T + 1$ and everyone is removed from the motel on that day, begin protocol again with $t_{i,new} = t_{i,old} + T + 1$

Here T is the random variable described by [1] as the duration of an infection. The probability of new arrivals to the town being infected by the virus is $\eta = 0.05$

Model and Assumptions

We assume that the spread of infection within the motel follows the Reed-Frost model, with x_i initial susceptibles, y_i initial infectives ($x_i + y_i = x_0$) and parameters $p = 0.1, \beta = 0.05$ i.e. $\alpha = 1 - p\beta = 0.995$, the probability of a single susceptible avoiding infection from a single infective within the motel. We assume that this probability is correct for all susceptibles and all infectives. The y_i initial infectives follows a $\text{Bin}(x_0, \eta)$ distribution, again this applied for all new arrivals into the town. The assumptions that the parameters α and η apply equally well to all new arrivals and do not change over time are unrealistic since different people have different risk-levels due to e.g. work environment and a pandemic is inherently very time-dependent. For the time-independence assumption to be valid we must assume that the portion of the pandemic being modelled is during an extended peak in cases in the population such that $d\eta/dt \approx 0$. This assumption of a constant η for all people relates to the assumption of zero immunity in the population, which is generally untrue for an ongoing pandemic. Furthermore, reliable calculation/measurment of η during the pandemic is unlikely due to its reliance on testing and reporting of cases and the changing nature of pandemics. The asumption that there will always be x_0 new arrivals available to enter the motel is also a bit unrealistic, since due to the varying times that groups spend in the motel this assumption means that there will be a large buildup of people waiting outside the town to enter the motel quarantine.

We are interested in the long-term expected rate of infections per day per person coming out of the motel (f), as well as the expected rate of infections per person coming out of the motel (g,to explicitly compare with $\eta = 0.05$ infecteds/new arrival without this quarantine protocol) and the expected rate of infections per day coming out of the motel (h). Using the random variables W and T to describe the Reed-Frost model for each group of x_0 accepted into the motel we get that the total number of infected from a group of x_0 is: $W + Y_i$, where $Y_i \sim \text{Bin}(x_0, \eta)$ is the random variable describing the number of infected new arrivals among the x_0 accepted into the motel. The total length of time from letting one group of x_0 into the motel to letting the next group of x_0 into the motel is: $T + I_{\{X_T>0\}}$. Where T is the length of the infection according to the Reed-Frost model, and $I_{\{X_T>0\}}$ is the indicator variable which takes the value 1 if $X_T > 0$ and 0 if $X_T = 0$ e.g. if all of the x_0 people were uninfected when they entered the motel on day 0 then we have $T = 0$ and $X_T = x_0 > 0$ so the next group of x_0 will be let in to the motel on day 1. If instead all of the x_0 people were infected when they entered the motel then we have $T = 1$ and $X_T = 0$ so the next group of x_0 will be let in to the motel on day 1.

The expected rates of interest are then:

$$f = \mathbb{E}\left(\frac{W + Y_i}{x_0\left(T + I_{\{X_T>0\}}\right)}\right),\ g = \mathbb{E}\left(\frac{W + Y_i}{x_0}\right),$$

$$h = \mathbb{E}\left(\frac{W + Y_i}{T + I_{\{X_T>0\}}}\right)$$

The Reed-Frost model of an epidemic with x_0 initial susceptibles, y_0 initial infectives and paramter α (probability that a single susceptible is not infected by a single infected in a single time period) has transition probabilities:

$$\mathbb{P}((X,Y)_{t+1} = (l,j)|(X,Y)_t = (k,i)) = \binom{k}{l}(1-\alpha^i)^{k-l}(\alpha^i)^l$$

Where $j = k - l$ and $l \leq k$ are constraints that result from these probabilities for an allowed transition. It is clear that the conditional distribution of $(X,Y)_{t+1}$ given $(X,Y)_t$ is:

$$X_{t+1} \sim \text{Bin}(x_t, \alpha^{y_t}),\ Y_{t+1} = X_t - X_{t+1}$$

From this conditional distribution, it is simple to use Monte-Carlo simulation to obtain an estimate of the expected values of interest. The Julia script implementing this process is attached at the end of this report for your interest.

We can compare the estimated rate of infections per person coming out of the motel with that generated using p.g.f. methods. Referring to [1] we know that the expected value of W is:

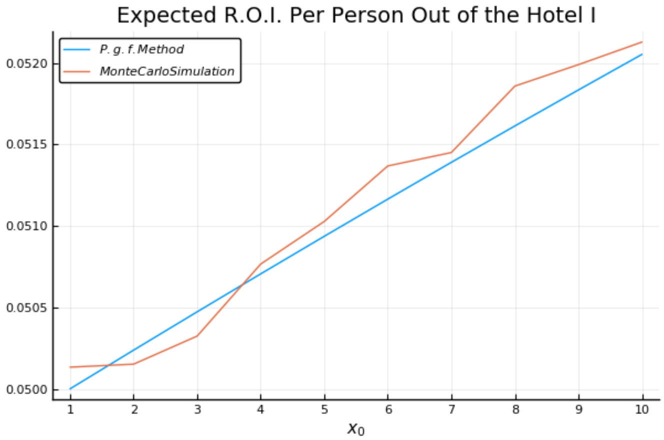
$$\mathbb{E}W = \Psi'_W(1) = A'\left(\sum_{t=1}^{x_0+1} \frac{d}{d\varphi}\left(\overline{P}(\varphi)\right)^{t-1}\Big|_{\varphi=1}\right) QE$$

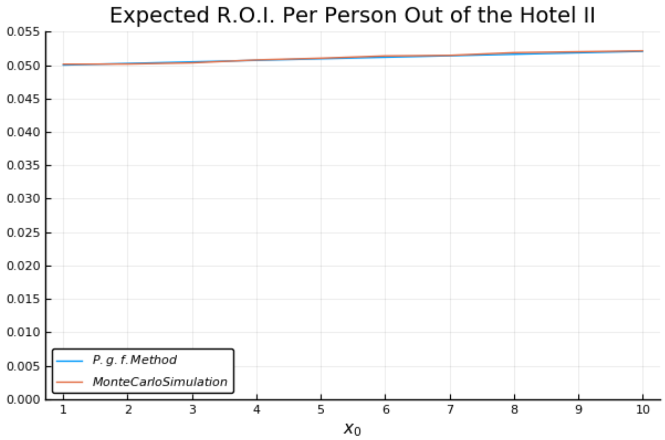
Where A′ is a vector probability distribution for the initial state; E is a vector of ones; $\overline{P} + Q = P$ where Q includes only the probabilities of moving into an absorbing state; and $\overline{P}(\varphi)$ is equal to $\overline{P}$ except that the elements in the j^th subdiagonal of each submatrix of $\overline{P}$, P_{ij} is multiplied by φ^j. We take the initial number of susceptibles for generating the transition probability matrix P as x_0 and then account for the different initial states by making $A_n = \mathbb{P}(Y_i = n)$ as described by the binomial distribution of Y_i.

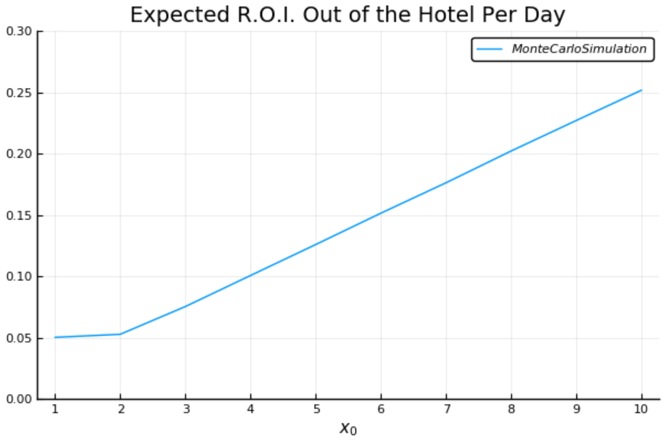
Results The figures on the following pages show the computational results. It can clearly be seen from the first figure that increasing the size of the motel increases the rate of infections per person out of the motel (and hence into the town) approximately linearly i.e. the effective rate is $> \eta$. However, the second figure shows that this increase is relatively small. The third figure shows that the rate of infections out of the hotel per day also increases approximately linearly as the size of the motel increases; however, this increase is significant, with the rate approximately quintupling between $x_0 = 1$ and $x_0 = 10$. The fourth figure indicates that the rate of infections per person out of the hotel per day dramatically drops off by a factor of a half for $x_0 > 1$ to an apparently stable rate of ~ 0.025.

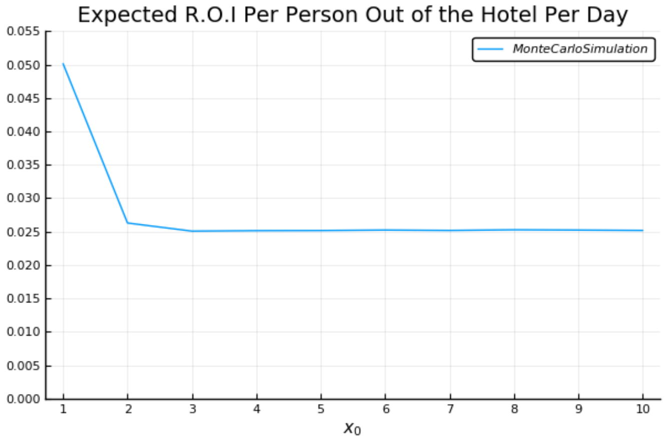
This stationary rate is of interest since it indicates that increasing the size of the motel allows more people to get through the quarantine process in a smaller amount of time. Since more people are leaving the motel per day, the expected rate of infections (fig. 3) leaving the motel per day also increases. However, since the actual rate of infections per person out of the motel only increases slightly (figs. 1 and 2), we can increase the size of the motel and make the quarantine process more efficient without excessively increasing the number of people infected.

Additionally, we note that this quarantine process identifies all infectives coming in to the town and hence removes the possibility of infection spreading within the wider community of the town. Thus despite increasing the expected rate of infections per person out of the motel (and into the town) it decreases the total number of infections occuring within the town's population.









References

[1] D. Daley, J. Gani (1999). Epidemic Modelling: an introduction. Chapter 4. pp. 105-132

Appendix of Code Snippets For Report (Q6)

```
In [5]: using LinearAlgebra, Statistics, StatsBase, Distributions, Plots, Random;
        x0s = 1:10; a = 0.995; eta = 0.05;
        N = 10^6;
        Random.seed!(1234)
        function MotelSim(x0,a)
            w = 0
            t = 0
            yi = rand(Binomial(x0,eta))
            x = x0 - yi
            y = yi
            while y>0
                xNew = rand(Binomial(x,a^y))
                y = x - xNew
                w += y
                t += 1
                x = xNew
            end
            if x > 0
                t += 1
            end
            return [(w+yi)/(x0*t), (w+yi)/x0, (w+yi)/t]
        end

        ests=zeros(3,length(x0s));
        for x0 in x0s
            MCSim = [MotelSim(x0,a) for _ in 1:N]
            est = mean(MCSim);
            print("x_0 = ",x0,":\n")
            print(est,"\n")
            ests[:,x0] = est
        end
```

```
x_0 = 1:
[0.050133, 0.050133, 0.050133]
x_0 = 2:
[0.02629325, 0.0501515, 0.0525865]
x_0 = 3:
[0.025077388888888892, 0.050323666666666673, 0.07523216666666666]
x_0 = 4:
[0.025147208333333337, 0.05076575, 0.10058883333333335]
x_0 = 5:
[0.025166993333333339, 0.05102720000000011, 0.1258349666666667]
x_0 = 6:
[0.0252385805555555, 0.051367833333333224, 0.15143148333333334]
x_0 = 7:
[0.025179780952381044, 0.051450285714285914, 0.17625846666666667]
x_0 = 8:
[0.025276616666666665, 0.051858125, 0.20221293333333332]
```

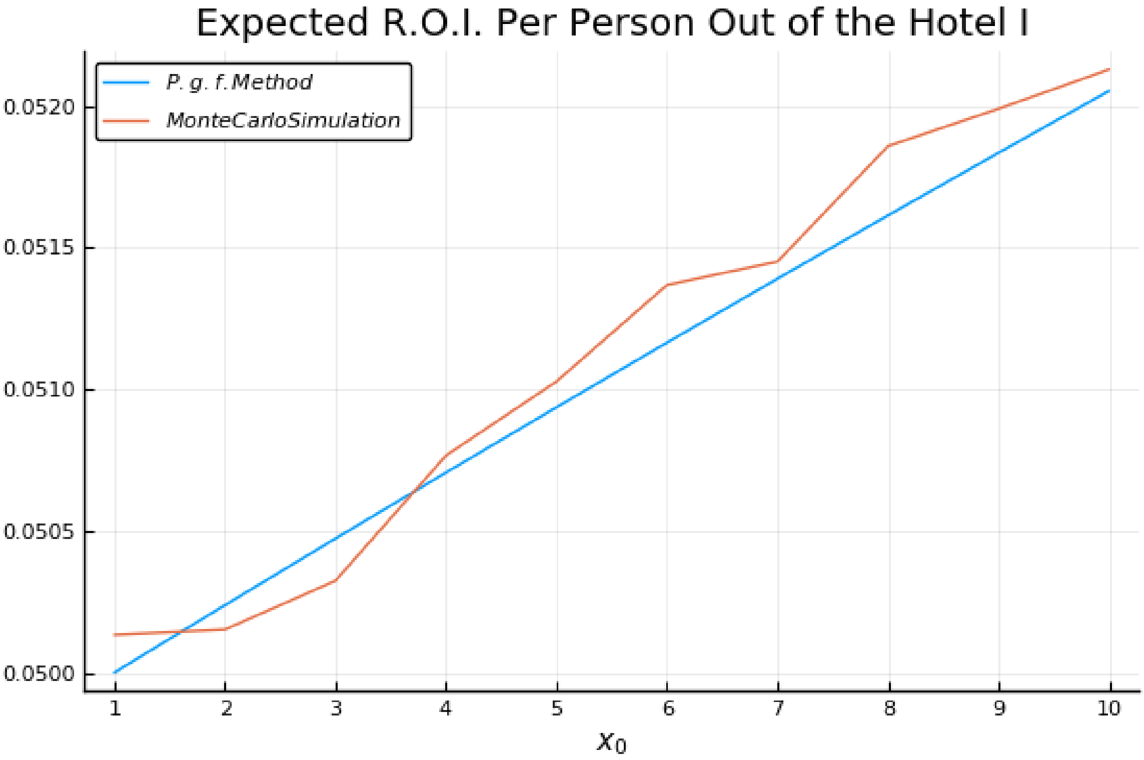
```
x_0 = 9:
[0.025236307407407513, 0.051989666666666906, 0.22712676666666667]
x_0 = 10:
[0.025182381666666816, 0.05212840000000032, 0.25182381666666664]
```

```
In [6]: using LinearAlgebra, Statistics, StatsBase, Distributions, Plots, Random, SymPy,BlockArrays
        x0s = 1:10; a = 0.995; eta = 0.05;

        function elemRF(k,i,l,j,a)
            #Generates the transition probability under the Reed-Frost model from state (X,Y)_t =
            if j + l != k
                return 0
            elseif l > k
                return 0
            end
            return binomial(k,l)*((1-a^i)^j)*(a^(i*l))
        end

        function makeRFPij(x0,a,i,j)
            #Creates the i,j-th Reed-Frost transition probability sub-matrix for x_0 initial susce
            N = x0+1;
            Pij = zeros(N,N)
            for k in 0:x0
                for l in 0:x0
                    Pij[k+1,l+1] = elemRF(k,i,l,j,a)
                end
            end
            return sparse(Pij)
        end

        function makeRFP(x0,a)
            #Creates the complete Reed-Frost transition probability matrix for x_0 initial suscept
            N = x0+1;
            indList = Int64[N for _ in 1:N];
            RFP = BlockArray{Float64}(undef_blocks,indList,indList)
            for i in 0:x0
                for j in 0:x0
                    Pij = makeRFPij(x0,a,i,j)
                    setblock!(RFP, Pij, i+1, j+1)
                end
            end
            return RFP
        end

        function makePBS(PB,x0,s)
            #Creates the P-bar(varphi) matrix with symbolic variable s = varphi
            N = x0+1;
            PBS = zeros(Sym(N^2),N^2)
            for k in 0:x0
        for l in 0:k
            j = k - l
            for i in 0:x0
                PBS[i*N + k + 1, j*N + l + 1] = (s^j)*(PB[i*N + k + 1, j*N + l + 1])
            end
        end
    end
    return PBS
end

function diffPBS(PBS,x0,s)
    #Differentiates the symbolic matric PBS = P-bar(varphi) w.r.t. the symoblic variable s
    #s=varphi=1
    N = x0+1;
    dPBS = zeros(N^2,N^2)
    for k in 0:x0
        for l in 0:k
            j = k - l
            for i in 0:x0
                temp = diff(PBS[i*N + k + 1, j*N + l + 1],s)
                dPBS[i*N + k + 1, j*N + l + 1] = temp(1)
            end
        end
    end
    return dPBS
end

EW = zeros(size(x0s))
for x0 in x0s
    A = zeros((x0+1)^2,1)
    for i = 0:x0
        j = x0-i
        A[j*(x0+1) + i + 1] = binomial(x0,j)*((1-eta)^i)*eta^j
    end
    E = ones((x0+1)^2,1)
    P = makeRFP(x0,a);
    Q = zeros(size(P));
    Q[:,1:x0+1] = P[:,1:x0+1];
    PB = P-Q;
    s = Sym("s")

    PBS = Array(makePBS(PB,x0,s))

    temp = zeros(size(P))
    for t in 1:x0+1
        temp += diffPBS(PBS^(t-1),x0,s)
    end
    EW[x0]=((A')*temp*Q*E)[1]
end
```

```
WARNING: using SymPy.N in module Main conflicts with an existing identifier.
```

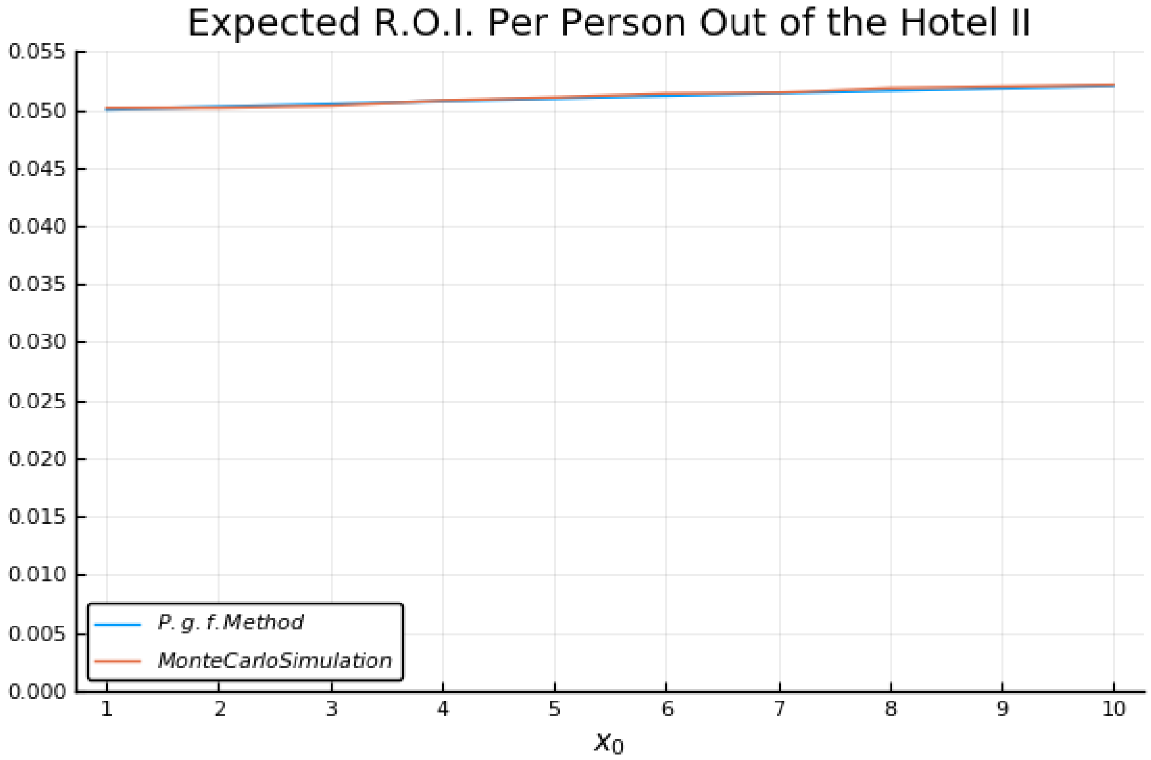
```
In [24]: plot(x0s,EW./x0s + 0.05.*ones(size(EW)), label=L"P.g.f. Method",xticks=(x0s,x0s))
         plot!(x0s,ests[2,:], label=L"Monte Carlo Simulation")
         title!("Expected R.O.I. Per Person Out of the Hotel I")
         xlabel!(L"x_0")
```

Out[24]:

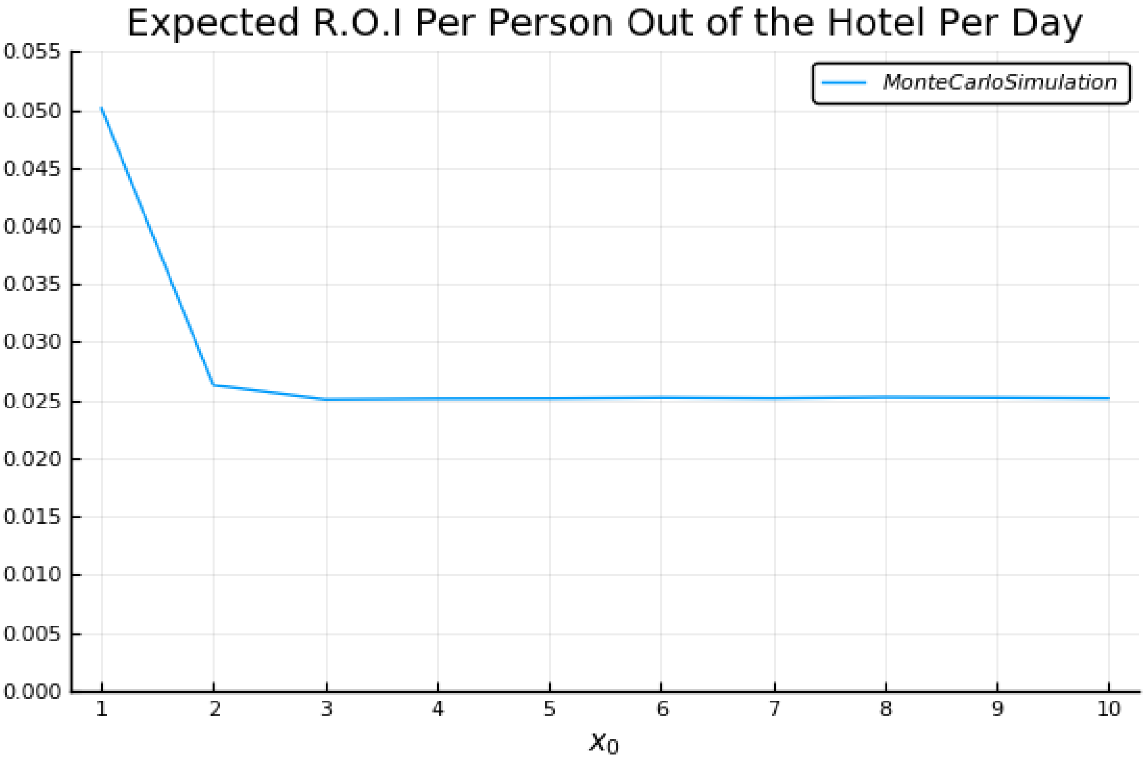


```
In [30]: plot(x0s,EW./x0s + 0.05.*ones(size(EW)), label=L"P.g.f. Method",xticks=(x0s,x0s),yaxis=((
         plot!(x0s,ests[2,:], label=L"Monte Carlo Simulation")
         title!("Expected R.O.I. Per Person Out of the Hotel II")
         xlabel!(L"x_0")
```

Out[30]:

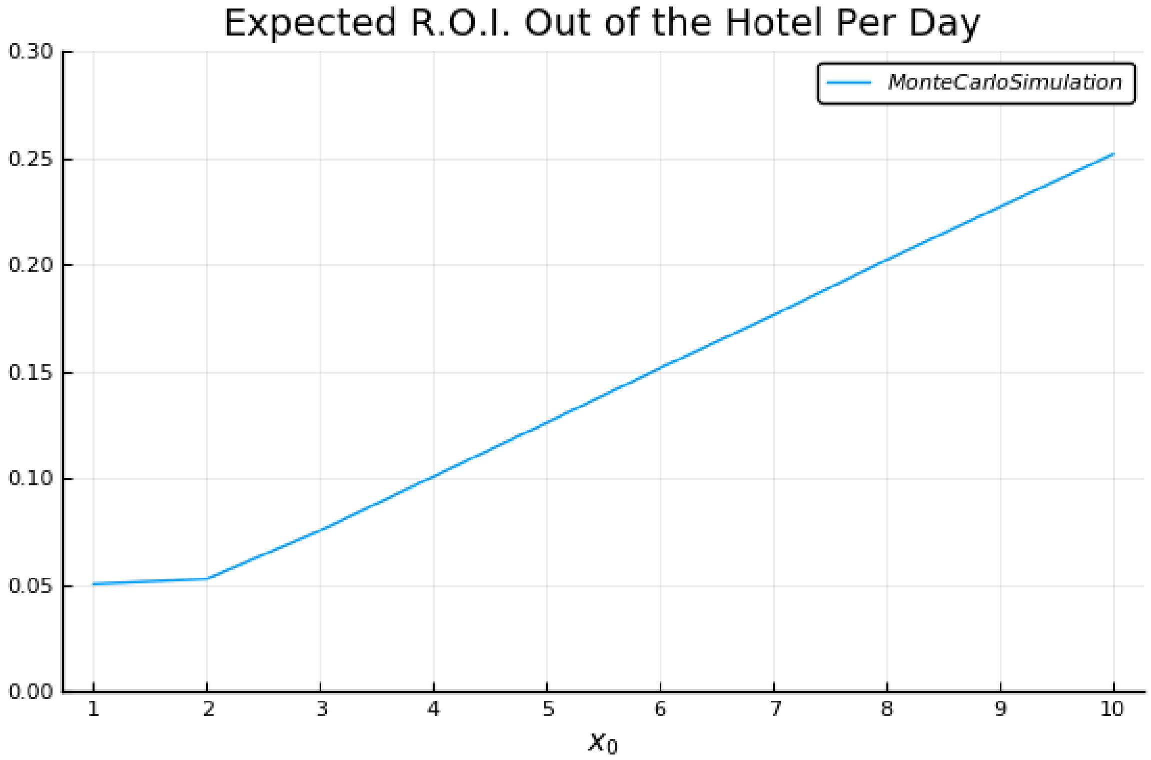


```
In [21]: plot(x0s,ests[1,:], label=L"Monte Carlo Simulation",xticks=(x0s,x0s),yaxis=((0, 0.055), 0
         title!("Expected R.O.I. Per Person Out of the Hotel Per Day")
         xlabel!(L"x_0")

Out[21]:
```



```
In [22]: plot(x0s,ests[3,:], label=L"Monte Carlo Simulation",xticks=(x0s,x0s),yaxis=((0, 0.3), 0:0
         title!("Expected R.O.I. Out of the Hotel Per Day")
         xlabel!(L"x_0")
```

```
Out[22]:
```



```
In [68]:

Out[68]: 0.049986200000212616

In [88]:

Out[88]: 10-element Array{Float64,1}:
          0.05
          0.050237500000000004
          0.050472695656250004
          0.05070557358954895
          0.050936114942218114
          0.05116429560496207
          0.051390086444064155
          0.0516134535293083
          0.051834358362971976
          0.05205275811018773

In [ ]:
```