1. (a) It is given that $\mathbb{E}[X_{t+1}|X_t] = \alpha X_t$, so it is true that $\mathbb{E}[X_t|X_{t-1}] = \alpha X_{t-1}$. So,

$$\begin{aligned}
\mathbb{E}[X_t] &= \mathbb{E}[\mathbb{E}[X_t|X_{t-1}]] \\
&= \alpha\mathbb{E}[X_{t-1}] \\
&= \alpha(\alpha\mathbb{E}[X_{t-2}] \\
&= \alpha \cdots \alpha\mathbb{E}[X_0] \qquad (\alpha \ t \text{ times}) \\
&= \alpha^t\mathbb{E}[X_0]
\end{aligned}$$

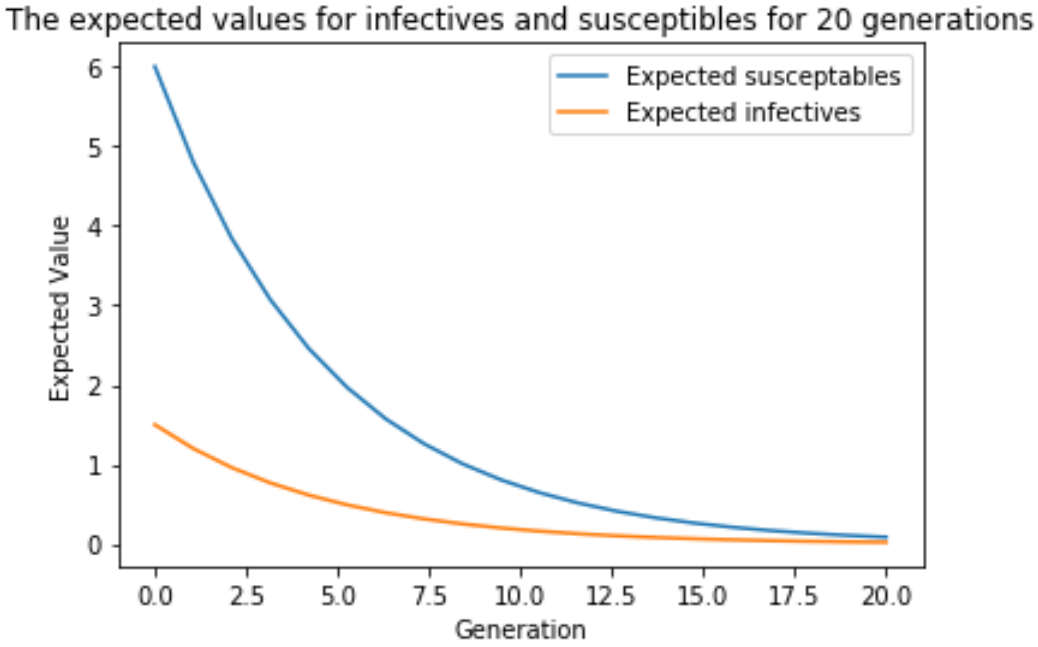
Therefore, $\mathbb{E}[X_t|X_0 = x_0] = \alpha^t x_0$, since $\mathbb{E}[X_0|X_0 = x_0] = x_0$

And because $Y_t = X_{t-1} - X_t$,

$$\begin{aligned}
\mathbb{E}[Y_t] &= \mathbb{E}[X_{t-1} - X_t] \\
&= \mathbb{E}[X_{t-1}] - \mathbb{E}[X_t] \\
&= \alpha^{t-1}x_0 - \alpha^t x_0 \\
&= (\alpha^{t-1} - \alpha^t)x_0 \\
&= \alpha^{t-1}(1-\alpha)x_0
\end{aligned}$$

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

(b)

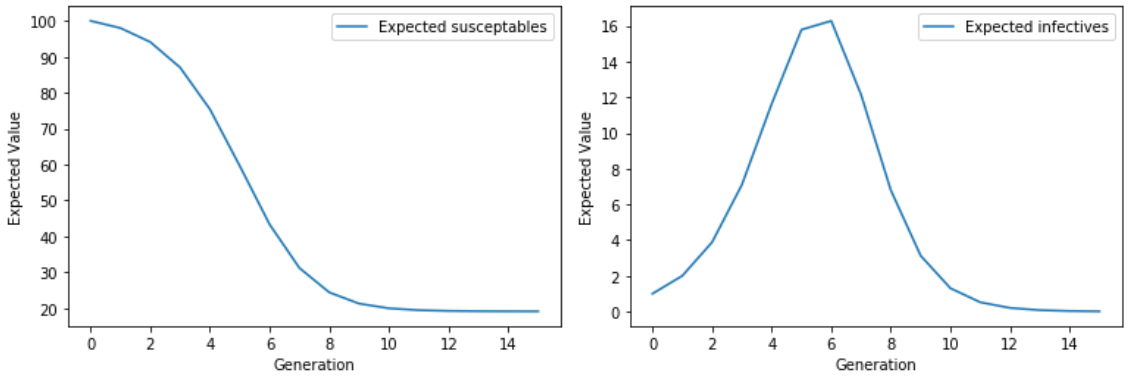


2. (a) We are given that $X_{t+1} \sim \text{Bin}(X_t, \alpha^{Y_t})$, so $\mathbb{E}[X_{t+1} \,|\, (X,Y)_t = (x,y)_t] = x_t\alpha^{y_t}$. (Expectation of a binomial variable)

Since a susceptible will either remain a susceptible or become an infective between times $t$ and $t+1$, it follows that $Y_{t+1} \sim \text{Bin}(X_t, 1-\alpha^{y_t})$.
Therefore, $\mathbb{E}[Y_{t+1} \,|\, (X,Y)_t = (x,y)_t] = x_t(1-\alpha^{y_t})$.

So we get that $\mathbb{E}[(X,Y)_{t+1} \,|\, (X,Y)_t = (x,y)_t] = (x_t\alpha^{y_t}, x_t(1-\alpha^{y_t}))$.

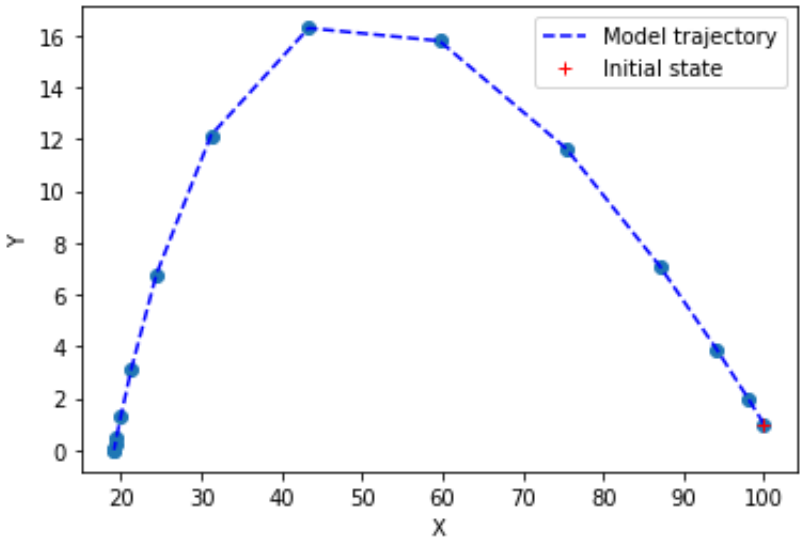
(b)




These figures were generated using the following python script:

```
import matplotlib.pyplot as plt
import numpy as np

a = 0.98

X = np.zeros(16)
X[0] = 100

Y = np.zeros(16)
Y[0] = 1

t = range(0,16)

for i in range(1,16):
    X[i] = X[i-1]*(a**(Y[i-1]))
    Y[i] = X[i-1]*(1-a**(Y[i-1]))

plt.figure(1)
plt.plot(t, X, label = "Expected susceptables")
```

```
plt.xlabel('Generation')
plt.ylabel('Expected Value')
plt.legend()

plt.figure(2)
plt.plot(t, Y, label = "Expected infectives")
plt.xlabel('Generation')
plt.ylabel('Expected Value')
plt.legend()
```

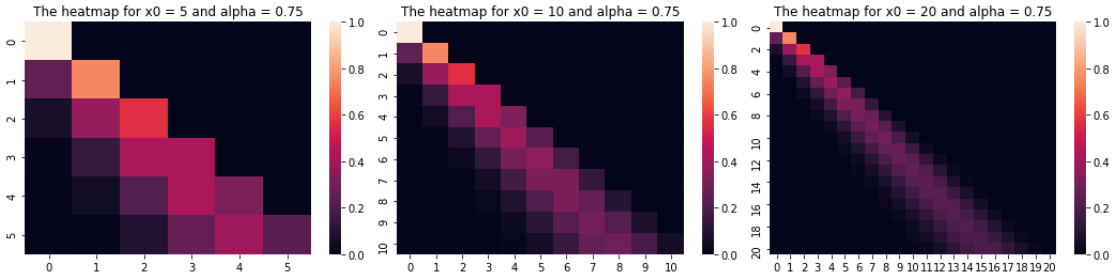



This figure was generated using the following python script:

```
import matplotlib.pyplot as plt
import numpy as np

a = 0.98

X = np.zeros(16)
X[0] = 100

Y = np.zeros(16)
Y[0] = 1

t = range(0,16)

init = [100,1]

for i in range(1,16):
```

```
        X[i] = X[i-1]*(a**(Y[i-1]))
        Y[i] = X[i-1]*(1-a**(Y[i-1]))

plt.scatter(X, Y)
plt.plot(X,Y,'b--',label = 'Model trajectory')
plt.plot(init[0],init[1],'r+',label = 'Initial state')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
```

3. (a)



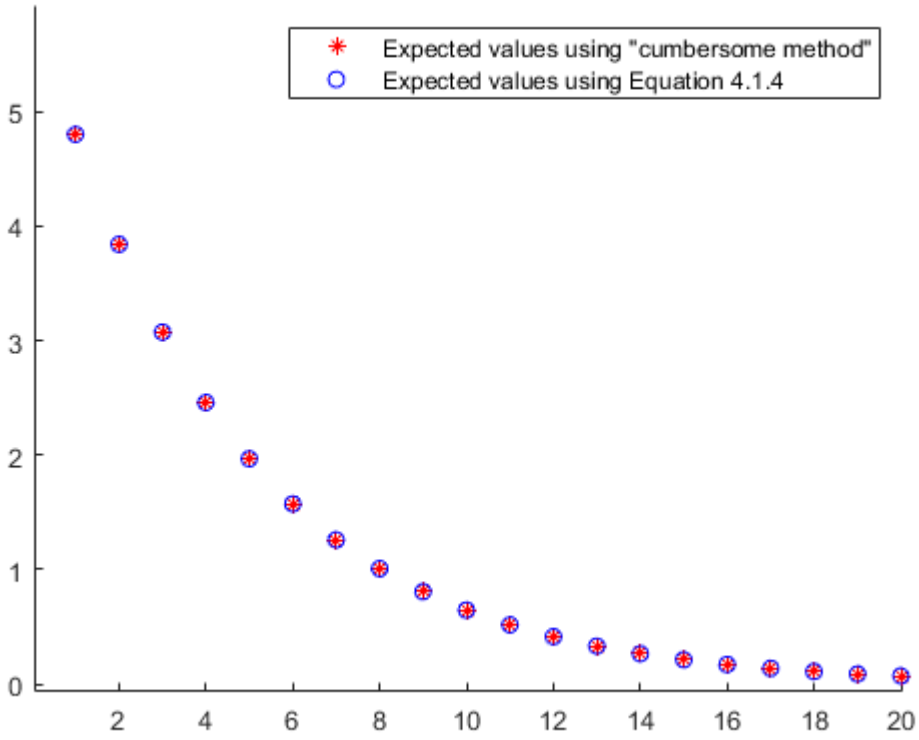

These were generated with $\alpha = 0.75$ using the following python script and changing the value for x0:

```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from scipy.special import comb

a = 0.75
x0 = 5

X = np.zeros(x0 + 1)
X[0] = x0

Y = np.zeros(x0 + 1)
Y[0] = 1

P = np.zeros([x0+1,x0+1])

for i in range(0,x0+1):
    for j in range(0,x0+1):
        if i>=j:
            P[i][j] = comb(i,j)*((1-a)**(i-j))*a**(j)

ax = sns.heatmap(P)
plt.title('The heatmap for x0 = %d and alpha = 0.75' % (x0))
plt.show()
```

(b) The communicating classes are; $\{0\}$, $\{1\}$, $\{2\}$, $\ldots$, $\{x_0\}$. There are $x_0 + 1$ communicating classes, all of which are transient except $\{0\}$, which is recurrent.

(c) The following Matlab script was used to numerically compute the expected values and compare to the values found in 1b:

```
a = 0.8;
x0 = 6;
P = zeros(x0+1);
n = 21;
t = 0:n-1;
e = [zeros(1,x0),1];
v = (0:x0).';
EX = zeros(1,20);

for i=1:(x0+1)
    for j=1:(x0+1)
        if i>=j
            P(i,j) = nchoosek(i-1,j-1)*(1-a)^(i-j)*a^(j-1);
        end
    end
end

for i=1:n
    Pt = P^(i-1);
    EX(i) = e*Pt*v;
end

for i=1:n
    EX1(i) = a^(i-1)*x0;
end

hold on
scatter(t,EX,'r*')
scatter(t,EX1,'b')
legend('Expected values using "cumbersome method"','Expected
   values using Equation 4.1.4')
hold off
```




This holds because it is essentially the expected value formula, multiplying each probability by the value that it would take, and summing them.

(d) The following lines were added to the script from 3(c) to compute the expectations for $Y_t$. It uses that

$$Y_{t+1} = X_t - X_{t+1}$$

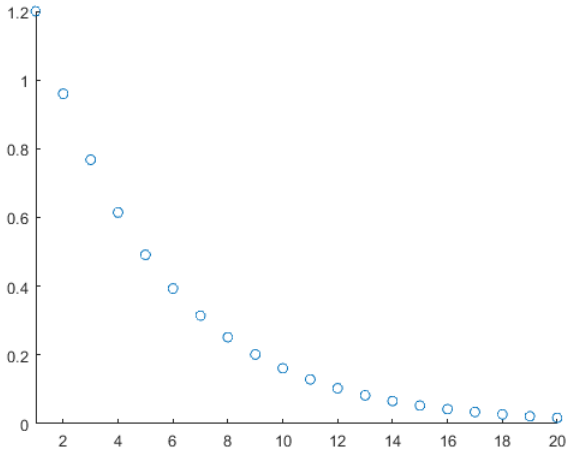
which gives

$$\mathbb{E}[Y_{t+1}] = \mathbb{E}[X_t] - \mathbb{E}[X_{t+1}]$$

and combining this with the methods for calculating $\mathbb{E}[X_t]$ in 3(c).

```
for i=2:n
    EY(i) = EX(i)-EX(i-1);
end
scatter(t,EY)
```

The following scatter plot was created:




4. (a) Equation (4.1.6) says that the probability of being at state $j$ at time $t$ is equal to the sum of the probabilities of all the different ways of getting there from time $t-1$. This is recursively defined. The number $p_i^{t-1}$ is the probability of being at state $i$ at time $t-1$ and $p_{ij}$ is the probability of moving from that state to state $j$. These are added over every possible state $i$, and hence give the probability $p_j^t$.

   (b) The following Matlab script uses the recursive relationship given to numerically compute $\mathbb{P}(W > 4)$

```
a = 0.8;
x0 = 6;
P = zeros(x0+1);
t = x0+2;
delta = 1;
p = zeros((x0+1),t+1);
p(x0+1,1) = delta;
gamma = zeros(x0+1,t+1);

for i=1:(x0+1)
    for j=1:(x0+1)
        if i>=j
            P(i,j) = nchoosek(i-1,j-1)*(1-a)^(i-j)*a^(j
                -1);
        end
    end
end

for g=2:t+1
    for j=x0:-1:1
        p(j,g) = p(j+1:(x0+1),g-1)'*P(j+1:(x0+1),j);
```

```
    end
end

for k=1:(x0+1)
    for n=2:(t+1)
        gamma(k,n) = p(x0+2-k,n-1)*a^(x0+1-k);
    end
end

prob = sum(gamma(6,:)) + sum(gamma(7,:))
```

This gives $\mathbb{P}(W > 4) = 0.1303$.

(c) The following Matlab script was used to generate $10^6$ repeated trajectories and estimate the probability that more than four susceptibles were infected by the time there are no infectives:

```
x0 = 6;
a=0.8;
n = 10^6;
X = zeros(n,x0);
Y = zeros(n,x0);
X(:,1) = x0;
Y(:,1) = 1;
W = zeros(n,1);

for i=1:n
    for j=2:(x0+1)
        X(i,j)=binornd(X(i,j-1),a);  % Generates X_t+1
            from a binomial distribution
        Y(i,j)=X(i,j-1)-X(i,j);
        if Y(i,j) == 0
            for t=(j+1):(x0+1)
                X(i,t)=NaN;          % Stops the pandemic
                    when the number of infectives reaches
                    zero.
            end
            break
        end
    end
    W(i,1) = min(X(i,:));            % Finds the number of
        susceptibles remaining when pandemic ends
end
```

```
prob = sum(W<2)/n                    % Finds proportion of time that W > 4
```

This gave the result that $\mathbb{P}(W > 4) = 0.1305$

(d) The following Matlab script was used to reproduce the PGF computations to then calculate $\mathbb{P}(W > 4)$:

```
a = 0.8;
x0 = 6;
P = zeros(x0+1);
Pbar = zeros(x0+1);
A = [zeros(1,x0),1];
E = ones(1,x0+1).';
I = eye(x0+1);
syms phi theta

for i=1:(x0+1)
    for j=1:(x0+1)
        if i>=j
            Pbar(i,j) = nchoosek(i-1,j-1)*(1-a)^(i-j)*a^(j-1); % Generate transition matrix Pbar
        end
        if i==j
            Q(i,j) = nchoosek(i-1,j-1)*(1-a)^(i-j)*a^(j-1);  % Generates the matrix Q
        end
        if i>j
            T(i,j) = phi^(i-j);
        end
    end
end


T(:,x0+1)=0;
Pbar2 = Pbar.*T;     % Creates the matrix Pbar(phi)

syms psi(phi)
psi(phi) = A*(inv((I-Pbar2)))*Q*E;  % Generates the PGF of W (theta = 1 since only finding size of epidemic)

syms dpsi(phi,n)

for i=1:(x0+1)
```

```
34      dpsi(phi) = diff(psi, phi, i-1);
35      prob(i) = dpsi(0)/factorial(i-1);    % Calculates
           vector of probabilities for W
36  end
37
38  p = sum(prob(6:7))
```

This gave the result $\mathbb{P}(W > 4) = 0.1303$.

As shown, all three methods for $\mathbb{P}(W > 4)$ agree.

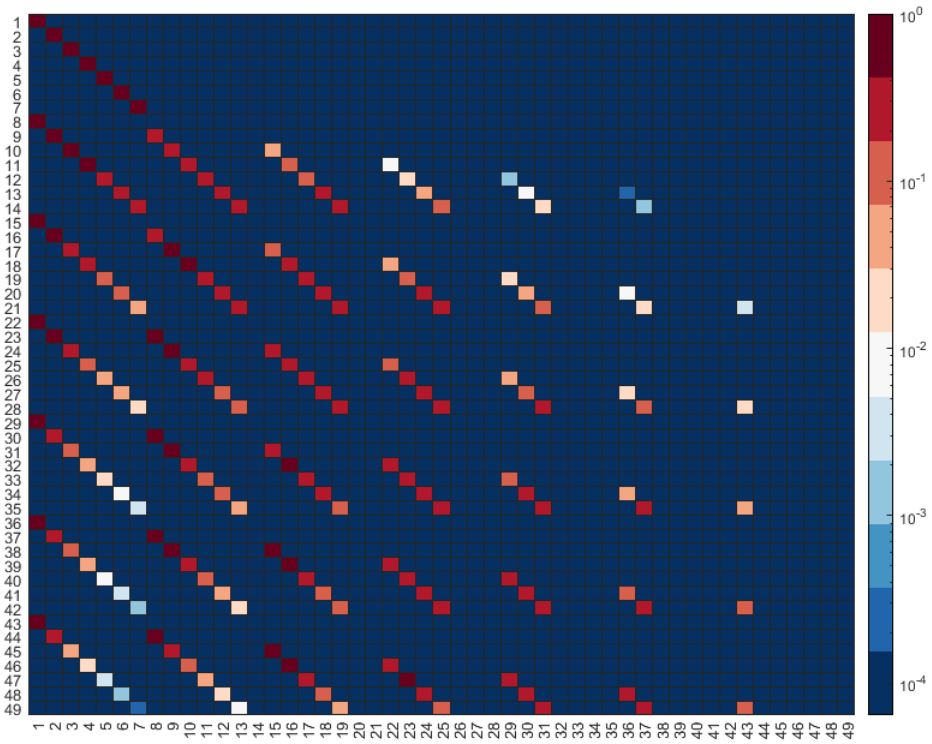
5. (a) $\{(X,Y) : X+Y \leq x_0, \quad X \geq 0, \quad Y \geq 1\}$

   (b) Every element $(X,Y)$ is it's own communicating class since the values for $X$ can only decrease or stay the same, so it is impossible to go from one state to another and the back to the original value if X decreases, and if it stays the same from one time period to the next, then Y has changed to 0 which means that (X,Y) cannot go back to the original state.

   (c) The following Matlab script was used to plot the heat-map of the transition probabilities:

```
a = 0.8;
x0 = 6;
A=zeros(x0+1);

for i=1:(x0+1)
    for j=1:(x0+1)
        A=zeros(x0+1);
        for r=1:(x0+1)
            for s=1:(x0+1)
                if r==s+j-1      % Generates the
                    submatrices
                    A(r,s) = nchoosek(s+j-2,j-1)*((1-a^(i
                        -1))^(j-1))*(a^((i-1)*(s-1)));

                end
            end
        end
        P{i,j} = A;
    end
end

P=cell2mat(P);
heatmap(P,'Colormap',redbluecmap,'ColorScaling','log')
```

   This generates the heat-map below:



(d) The following Matlab script was used to run a Monte-Carlo simulation with $10^6$ repeated trajectories to estimate $P(W > 4)$.

```
x0 = 6;
a=0.8;
n = 10^6;
X = zeros(n,x0);
Y = zeros(n,x0);
X(:,1) = x0;
Y(:,1) = 1;
W = zeros(n,1);

for i=1:n
    for j=2:(x0+1)
        X(i,j)=binornd(X(i,j-1),a^(Y(i,j-1)));    % Generates the 10^6 trajectories using binomial distribution
        Y(i,j)=X(i,j-1)-X(i,j);
    end
    W(i,1) = min(X(i,:));     % Finds the number of susceptibles remaining when pandemic ends
```

```
16 end
17
18 prob = sum(W < 2)/n    % Finds proportion of time that W
       >4
```

This gave the result that $\mathbb{P}(W > 4) = 0.2560$.
This result is much higher compared to 4c, since the probability of being infected is higher due to the value for $\alpha$ being raised to the power of $Y_t$. This model takes into account the total amount of people infected and makes it more likely for susceptibles to be infected when there are more infectives, which the Greenwood model does not account for.

6. This report considers the impact on infections of having a motel to quarantine new arrivals into a regional town. The basic layout is that a number $x_0$ people will be accepted into the motel before being allowed to enter the town, where they will be tested for the disease. The reuslts for these test are released the following day, and if there are no individuals infected then they are allowed to leave the motel and enter the town. However, if there are of the $x_0$ infected, the infected are removed and taken to health care elsewhere, and all others are kept in the motel until there are no more individuals infected. Because of the delay in test results, it is possible that those infected infect others before test results are available. When there are no more people infected, everyone is released and a new batch of $x_0$ people are brought in. This process is then repeated for a certain time period.

   In this analysis, the impact of the size of $x_0$ will be investigated and discussed. Specifically, $x_0 = 1, \ldots, 10$ will be tested, over a long time period to find long-term effects. The main goal is to find the expected rate of infection per day coming out of the motel as a function of $x_0$. The method used for this is Monte-Carlo simulation in Matlab to simulate this scenario, using the Reed-Frost Model as the basis for the infection model (see Appendix 1.1 for Matlab script). In this case, we are assuming that $p = 0.1$ and $\beta = 0.05$. That is, $\alpha = 0.995$.

   In this simulation, a time period of $10^5$ days was used so that sample size was large enough. The table below shows the number of individuals infected on a given day for the first 9 days of the simulation. The rows represent the value for $x_0$, so that the first row is $x_0 = 1$ and so on up to $x_0 = 10$.

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 0 | 2 | 0 | 0 | 1 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

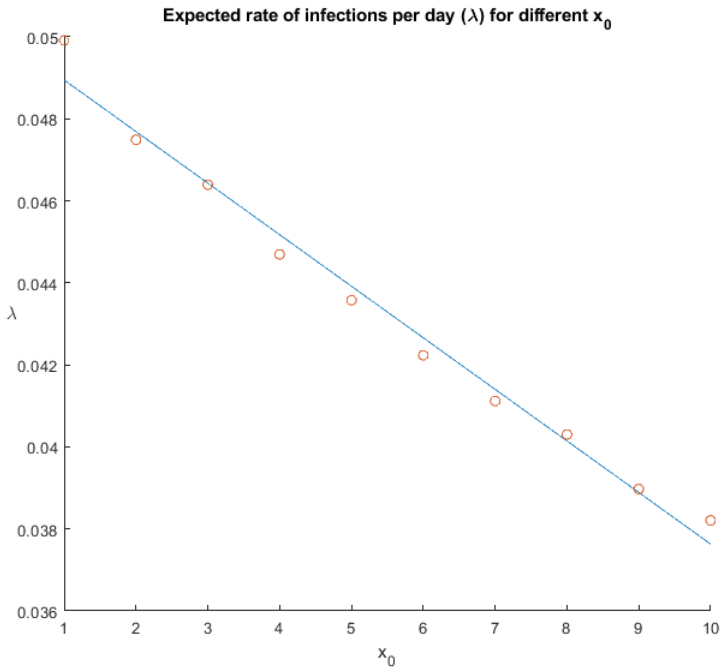
   The table below shows the rate of infections coming out of the hotel, from here on denoted by $\lambda$, given the different values of $x_0$:

| $x_0$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\lambda$ | 0.0499 | 0.0475 | 0.0464 | 0.0447 | 0.0436 | 0.0422 | 0.0411 | 0.0403 | 0.0390 | 0.0382 |

   This data is then used to create a polynomial fit to get $\lambda$ as a function of $x_0$. The resulting relationship is

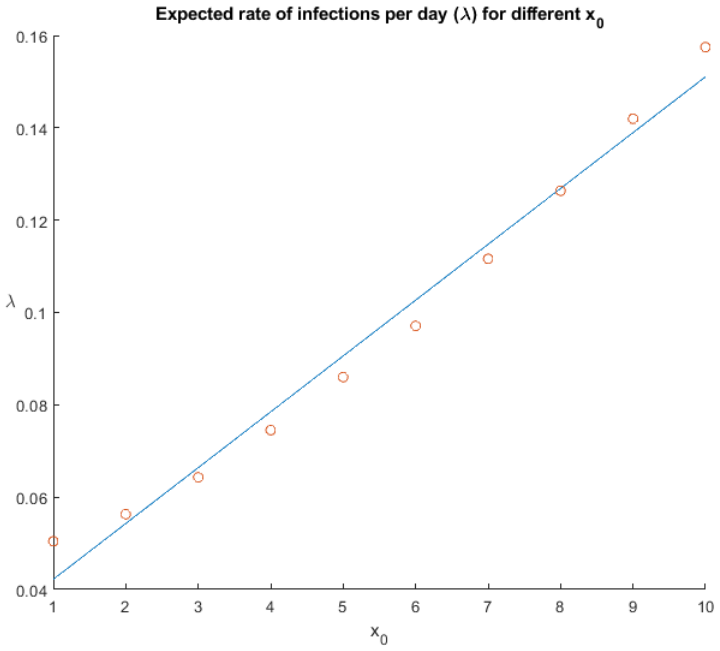
$$\lambda = 0.0503 - 0.0013x_0$$

These values are plotted along with the line of best fit below:




It can be seen here that $\lambda$ decreases with increasing values for $x_0$. This is because the rate of infection between the individuals in the motel $1 - \alpha = 0.005$ is lower than the probability of a new arrival having the disease $\eta = 0.05$. This means that keeping people in the motel until there are no infections, which is likely to last longer with more people, results in a lower amount of people infected than bringing in $x_0$ people straight into the town each with infection rate $\eta$.

Conversely, we see a positive relationship when $\alpha < 0.95$. Take for example $\alpha = 0.8$, the plotted points and line of best fit are shown below:




This indicates that the relationship is strongly dependent on the values for $p$ and $\beta$. We can conclude from this analysis that if $\alpha > \eta$, then keeping new arrivals into the town in the motel until no more of them are infected is worthwhile as it decreases the rate of infections $\lambda$. There are also the added benefits of not releasing any infected people at all into the town's society, which obviously reduces the community transmissions in that town. There could possibly be an argument for keeping the motel policy in place if $\alpha < \eta$, due to these added benefits that are not discussed in this model. These effects would have to further analysed to discover the potential advantages or disadvantages. There is also the fairly important assumption that there is no immunity to the virus. This is of course unlikely and could impact the results significantly. Obviously some people are more likely (immunosuppressed) or less likely (some sort of immunity) to be infected by a disease by others, so using a constant value for $\alpha$ could be considered unrealistic. This however makes the model significantly more challenging. Regardless, from the perspective of the people in the town, the motel will always be the best option in terms of their health.

# Appendix

## 1.1

```
a = 0.995;
eta = 0.05;
n = 10^5;
x = [1:10];
Y = zeros(10,n);
X = zeros(10,n);

for x0=1:10
    for t=1:n
        if t==1
            Y(x0,t) = binornd(x0,eta);
            X(x0,t) = x0 - Y(x0,t);
        elseif Y(x0,t-1)==0
            Y(x0,t) = binornd(x0,eta);
            X(x0,t) = x0 - Y(x0,t);
        else
            X(x0,t)=binornd(X(x0,t-1),a^(Y(x0,t-1)));
            Y(x0,t)=X(x0,t-1)-X(x0,t);
        end
    end
end

totalpeople = sum(X+Y,2);
Infected =  sum(Y');
Proportion = Infected./totalpeople';
coef = polyfit(x, Proportion,1);
x1 = 1:0.01:10;
y = coef(1)*x1+coef(2);
hold on
plot(x1,y)
scatter(x, Proportion)
xlabel('x_0')
ylabel('\lambda')
title('Expected rate of infections per day (\lambda) for different x_0
    ')
set(get(gca,'ylabel'),'rotation',0)
hold off

syms x0
sympref('FloatingPointOutput',true);
rate = coef(1)*x0 + coef(2)
```