STAT7304 Project 1

By Ashley Gibson

Question 1

Part (a)

For the Greenwood model,

$$\mathbb{E}(X_t \,|\, X_{t-1}) = \alpha X_{t-1}.$$

Taking the conditional expectation of both sides for a fixed $X_0 = x_0$ gives

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

from which the law of total expectation yields

$$\mathbb{E}(X_t \,|\, X_0 = x_0) = \alpha \mathbb{E}(X_{t-1} \,|\, X_0 = x_0).$$

Recursively evaluating this gives

$$\begin{aligned} \mathbb{E}(X_t \,|\, X_0 = x_0) &= \alpha \mathbb{E}(X_{t-1} \,|\, X_0 = x_0) \\ &= \alpha^2 \mathbb{E}(X_{t-2} \,|\, X_0 = x_0) \\ &= \alpha^3 \mathbb{E}(X_{t-3} \,|\, X_0 = x_0) \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \\ &= \alpha^{t-1} \mathbb{E}(X_1 \,|\, X_0 = x_0) \\ &= \alpha^t \mathbb{E}(X_0 \,|\, X_0 = x_0) \\ &= \alpha^t x_0. \end{aligned}$$

Also for the Greenwood model,

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

Taking the conditional expectation of both sides for a fixed $X_0 = x_0$ gives

$$\begin{aligned} \mathbb{E}(Y_t \,|\, X_0 = x_0) &= \mathbb{E}(X_{t-1} \,|\, X_0 = x_0) - \mathbb{E}(X_t \,|\, X_0 = x_0) \\ &= \alpha^{t-1} x_0 - \alpha^t x_0 \\ &= \alpha^{t-1}(1-\alpha) x_0. \end{aligned}$$

Part (b)

Assume that $x_0 = 6$ and $\alpha = 0.8$. Substituting these into the results from part (a) gives

$$\mathbb{E}(X_t \,|\, X_0 = 6) = 6 \times 0.8^t,$$

$$\begin{aligned} \mathbb{E}(Y_t \,|\, X_0 = 6) &= 6 \times 0.8^{t-1}(1 - 0.8) \\ &= 6 \times \frac{0.8^t}{0.8} \times 0.2 \\ &= 1.5 \times 0.8^t. \end{aligned}$$

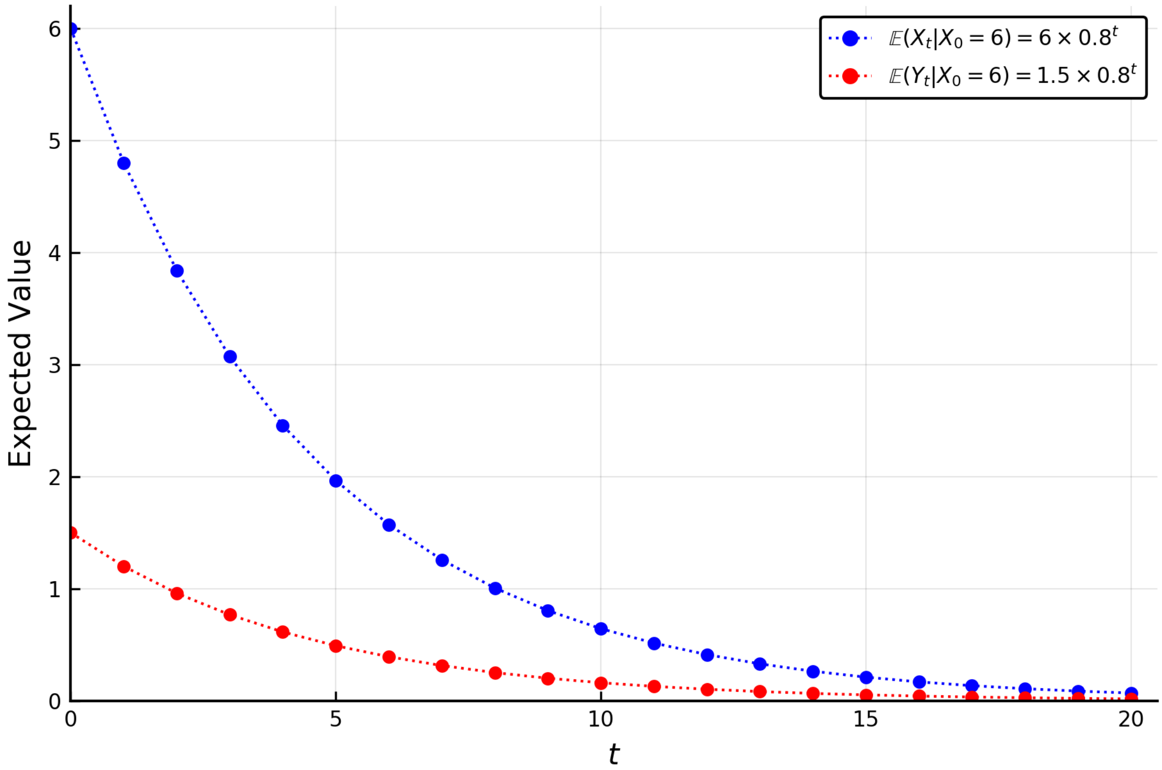

Figure 1: Plot of the expected values of X_t and Y_t given $X_0 = 6$ for the Greenwood model with $\alpha = 0.8$.

Julia Code

```
using Plots, LaTeXStrings; pyplot(dpi=300)

tmax = 20
t = [0:tmax;]
f = zeros(tmax + 1)
g = zeros(tmax + 1)

for t in 0:tmax
    f[t+1] = 6 * 0.8 ^ t
    g[t+1] = 1.5 * 0.8 ^ t
end

plot(t, f, label = L"\mathbb{E}(X_t | X_0=6) = 6 \times 0.8^t",
    xlims = (0, tmax+0.5),  ylims = (0, 6.2),
    xlabel = L"t", ylabel = "Expected Value",
    c = :blue, ls = :dot, m = (:dot, 5, Plots.stroke(0)))

plot!(t, g, label = L"\mathbb{E}(Y_t | X_0=6) = 1.5 \times 0.8^t",
    c = :red, ls = :dot, m = (:dot, 5, Plots.stroke(0)))
```

Question 2

Part (a)

For the Reed–Frost model,

$$X_{t+1} \sim \text{Bin}\left(X_t, \alpha^{Y_t}\right),$$

so the expected value of X_{t+1} for a fixed pair $(X, Y)_t = (x, y)_t$ is given by

$$\begin{aligned}\mathbb{E}[X_{t+1} \mid (X,Y)_t = (x,y)_t] &= \left[\mathbb{E}[X_{t+1}]\right]_{(X,Y)_t=(x,y)_t} \\ &= \left[X_t \alpha^{Y_t}\right]_{(X,Y)_t=(x,y)_t} \\ &= x_t \alpha^{y_t}.\end{aligned}$$

Also for the Reed–Frost model, $Y_{t+1} = X_t - X_{t+1}$, so the expected value of Y_{t+1} is given by

$$\begin{aligned}\mathbb{E}[Y_{t+1} \mid (X,Y)_t = (x,y)_t] &= \mathbb{E}[X_t \mid (X,Y)_t = (x,y)_t] - \mathbb{E}[X_{t+1} \mid (X,Y)_t = (x,y)_t] \\ &= x_t - x_t \alpha^{y_t} \\ &= x_t(1 - \alpha^{y_t}).\end{aligned}$$

Combining these results yields

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

Part (b)

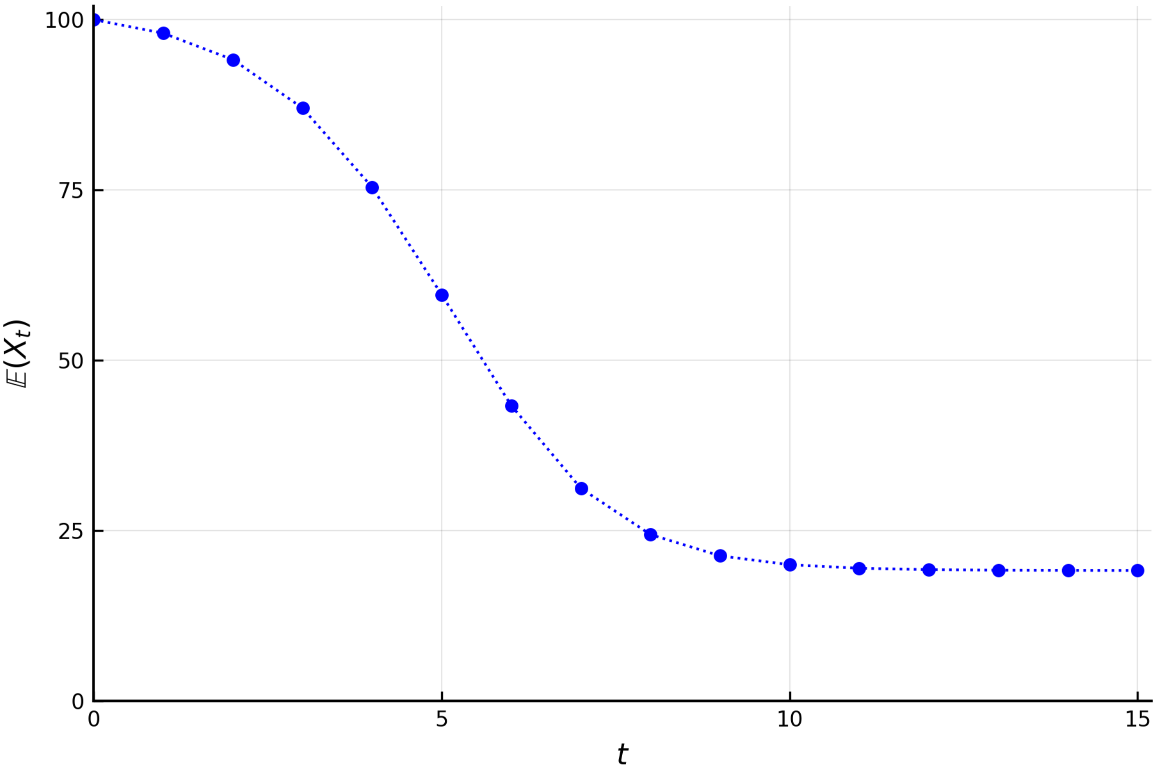

Figure 2: Plot of the expected value of X_t given $X_0 = 100$ and $Y_0 = 1$ for the Reed–Frost model with $\alpha = 0.98$.

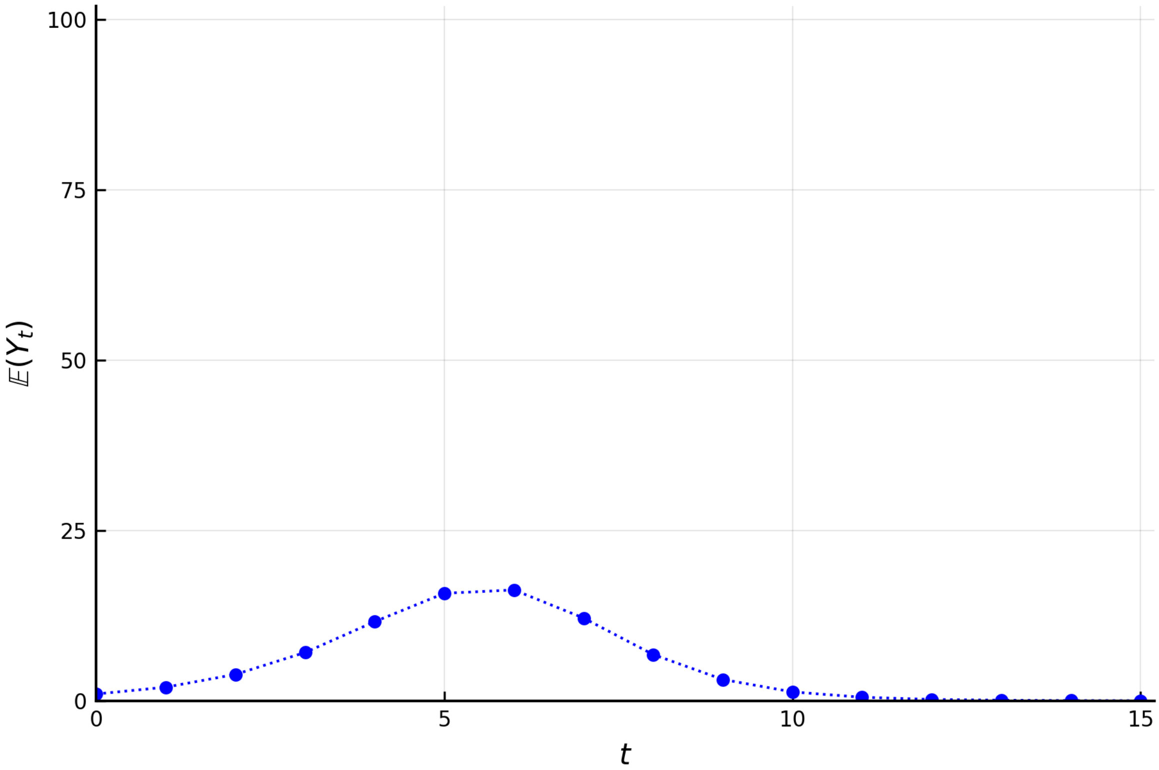

Figure 3: Plot of the expected value of Y_t given $X_0 = 100$ and $Y_0 = 1$ for the Reed–Frost model with $\alpha = 0.98$.

Julia Code

```
using Plots, LaTeXStrings; pyplot(dpi=300)

alph = 0.98
tmax = 15
t = [0:tmax;]
x = zeros(tmax+1)
y = zeros(tmax+1)
x[1] = 100
y[1] = 1

for i = 1:tmax
    x[i+1] = x[i] * (alph ^ y[i])
    y[i+1] = x[i] * (1 - alph ^ y[i])
end

plot(t, x, legend = false,
    xlim = (0, tmax+0.2), ylim = (0, max(x[1], y[1]) + 2),
    xlabel = L"t", ylabel = L"\mathbb{E}(X_t)",
    c = :blue, ls = :dot, m = (:dot, 5, Plots.stroke(0)))

plot(t, y, legend = false,
    xlim = (0, tmax+0.2), ylim = (0, max(x[1], y[1]) + 2),
    xlabel = L"t", ylabel = L"\mathbb{E}(Y_t)",
    c = :blue, ls = :dot, m = (:dot, 5, Plots.stroke(0)))
```

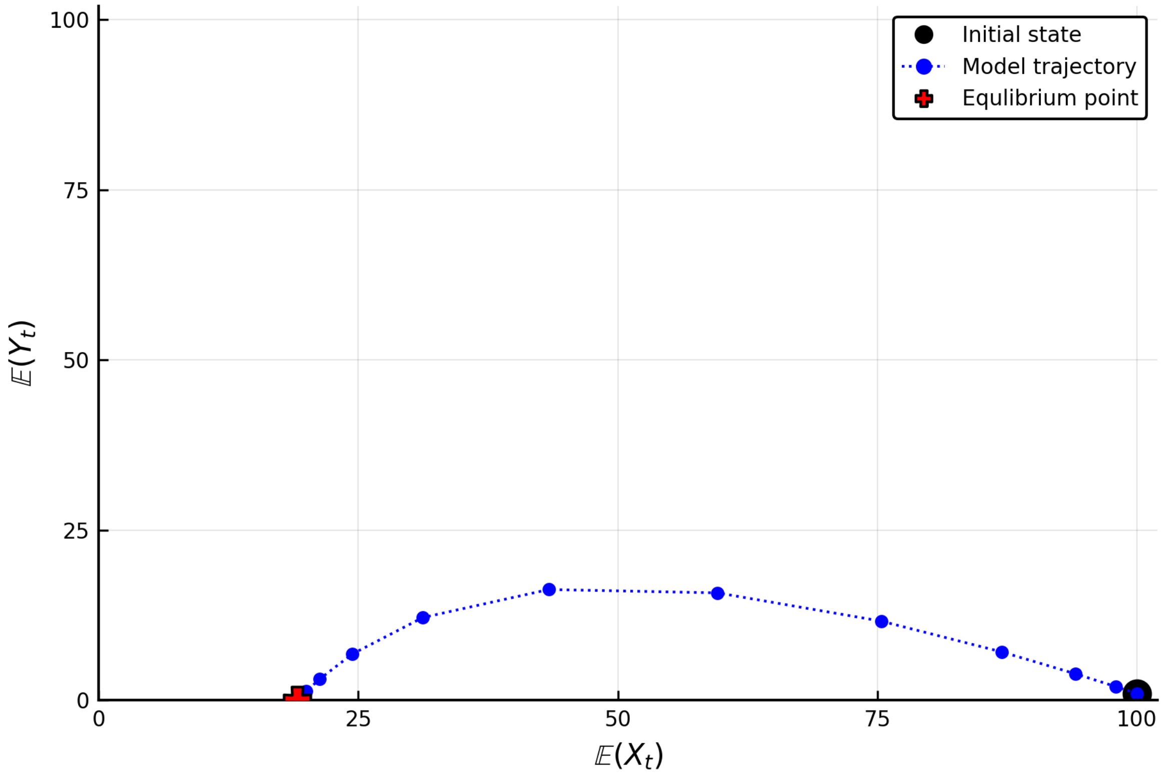

Figure 4: Plot of the expected values of X_t and Y_t given $X_0 = 100$ and $Y_0 = 1$ for the Reed–Frost model with $\alpha = 0.98$.

Julia Code

```
using Plots, LaTeXStrings; pyplot(dpi=300)

alph = 0.98
next(x,y) = [x * alph ^ y, x * (1 - alph ^ y)]
equibPoint = [19.12740463953112, 0]

initX = [100, 1]
tEnd = 100

traj = [[] for _ in 1:tEnd]
traj[1] = initX

for t in 2:tEnd
    traj[t] = next(traj[t-1]...)
end

scatter([traj[1][1]], [traj[1][2]], label = "Initial state",
    c = :black, ms = 10)
plot!(first.(traj),last.(traj), label = "Model trajectory",
    xlims = (0, tEnd+2), ylims = (0, tEnd+2),
    c = :blue, ls = :dot, m = (:dot, 5, Plots.stroke(0)))
scatter!([equibPoint[1]], [equibPoint[2]], label = "Equlibrium point",
    xlabel = L"\mathbb{E}(X_t)", ylabel = L"\mathbb{E}(Y_t)",
    c = :red, shape = :cross, ms = 10)
```

Question 3

Part (a)

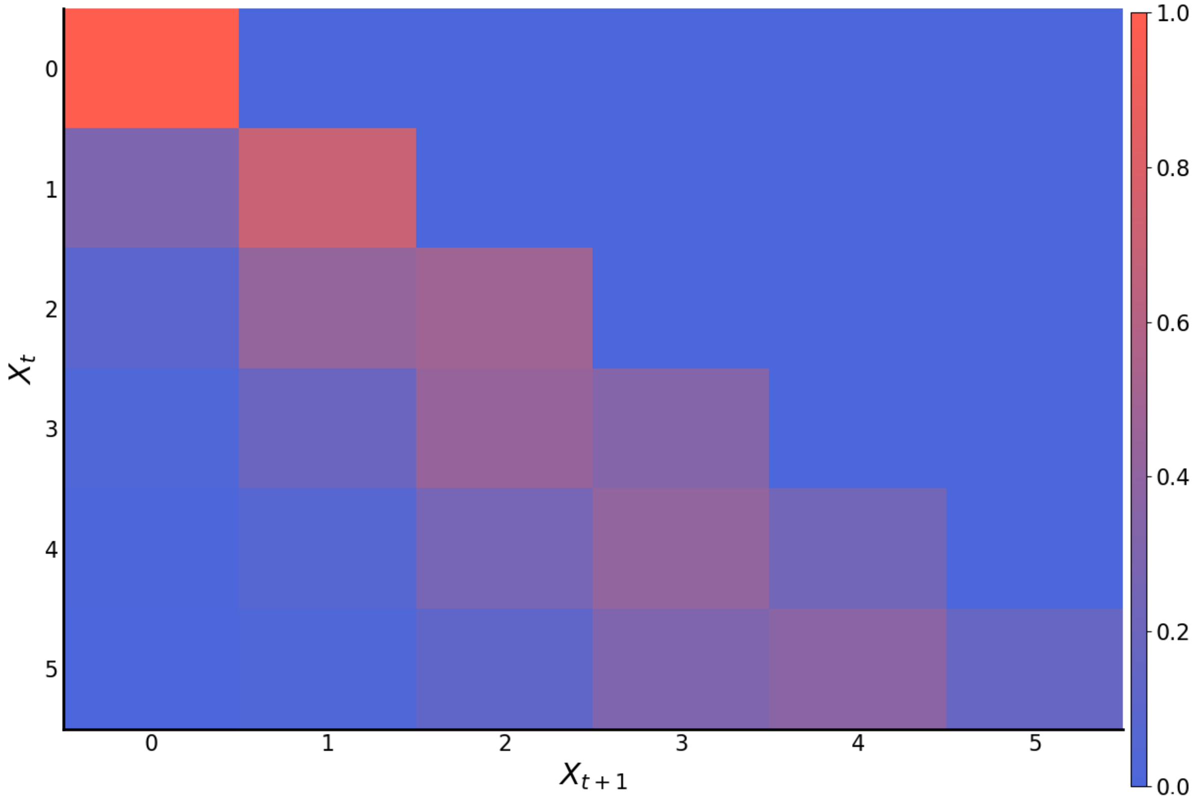

Figure 5: Heat map of the transition matrix for the Greenwood model with $\alpha = 0.7$ and $x_0 = 5$.

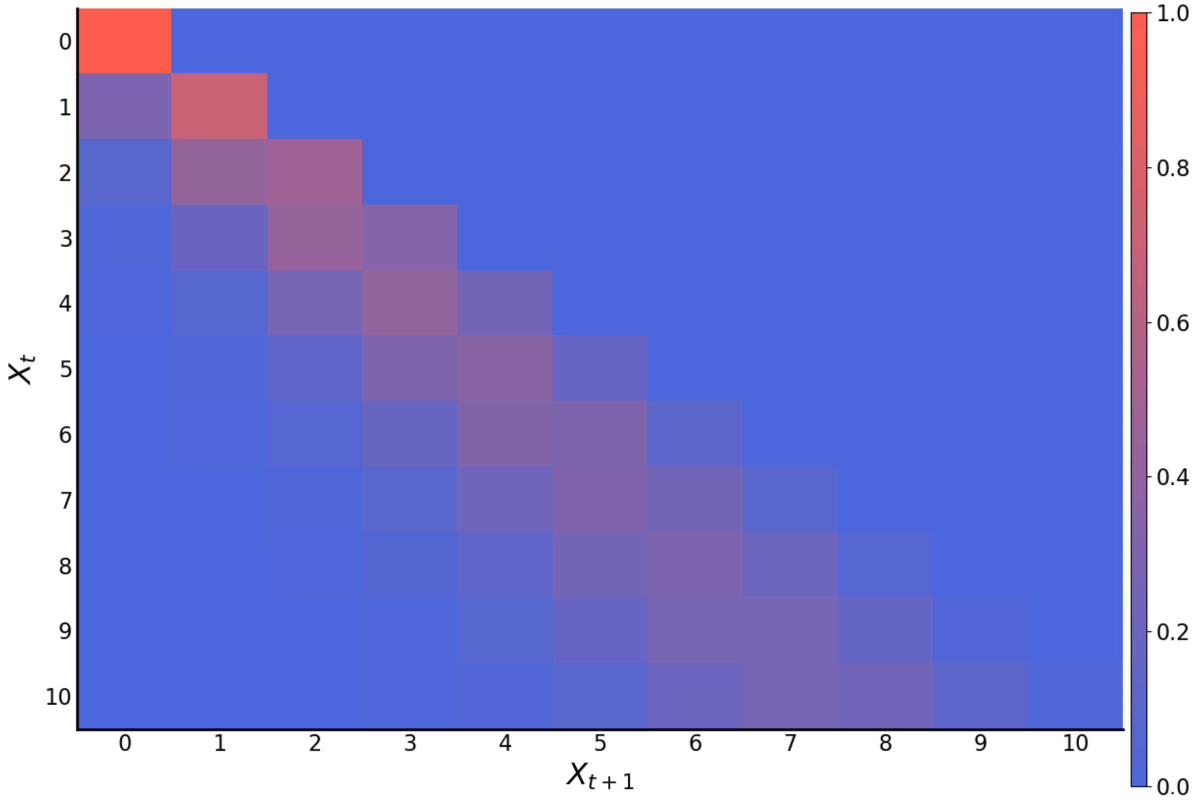

Figure 6: Heat map of the transition matrix for the Greenwood model with $\alpha = 0.7$ and $x_0 = 10$.

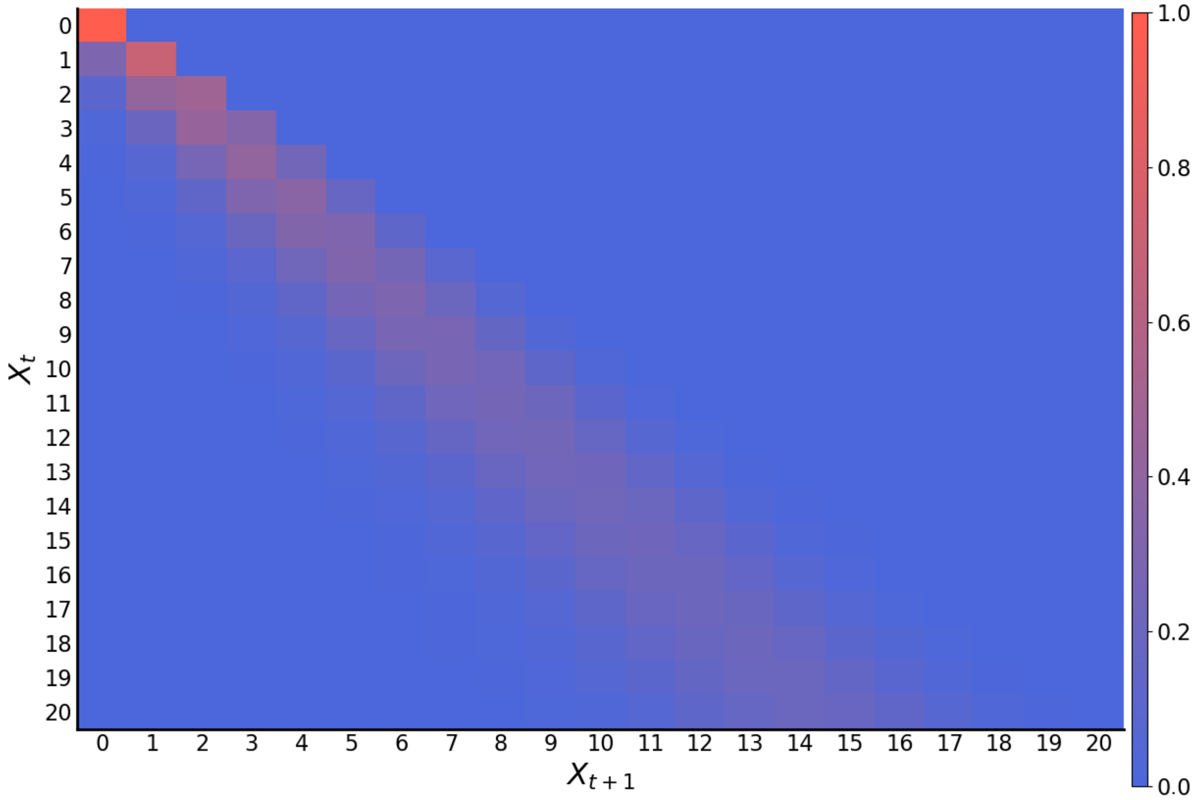

Figure 7: Heat map of the transition matrix for the Greenwood model with $\alpha = 0.7$ and $x_0 = 20$.

Julia Code

```
using Plots, LaTeXStrings; pyplot(dpi=300)

alph = 0.7
x0 = 20

P = zeros(x0+1, x0+1)
for i = 1:x0+1
    for j = 1:x0+1
        P[i, j] = binomial(i-1,j-1) * (1-alph)^((i-1)-(j-1)) * alph^(j-1)
    end
end

heatmap(P, c = cgrad([:royalblue, :tomato]), yflip = true,
    xlabel = L"X_{t+1}", ylabel = L"X_t")
xticks!([1:x0+1;], string.([0:x0;]))
yticks!([1:x0+1;], string.([0:x0;]))
```

Part (b)

For the Greenwood model, the transition matrix shows that

$$
\begin{array}{cccccc}
0 \leftrightarrow 0 & 0 \nrightarrow 1 & 0 \nrightarrow 2 & 0 \nrightarrow 3 & \cdots & 0 \nrightarrow x_0 \\
1 \rightarrow 0 & 1 \leftrightarrow 1 & 1 \nrightarrow 2 & 1 \nrightarrow 3 & \cdots & 1 \nrightarrow x_0 \\
2 \rightarrow 0 & 2 \rightarrow 1 & 2 \leftrightarrow 2 & 2 \nrightarrow 3 & \cdots & 2 \nrightarrow x_0 \\
3 \rightarrow 0 & 3 \rightarrow 1 & 3 \rightarrow 2 & 3 \leftrightarrow 3 & \cdots & 3 \nrightarrow x_0 \\
\vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
x_0 \rightarrow 0 & x_0 \rightarrow 1 & x_0 \rightarrow 2 & x_0 \rightarrow 3 & \cdots & x_0 \leftrightarrow x_0
\end{array}
$$

Hence, there are $x_0 + 1$ communicating classes, given by $\{0\}, \{1\}, \{2\}, \{3\}, \ldots, \{x_0\}$, which correspond to the diagonal entries of the transition matrix. Because $0 < \alpha < 1$, only $\{0\}$ is recurrent, while all the other communicating classes are transient.

Parts (c) and (d)

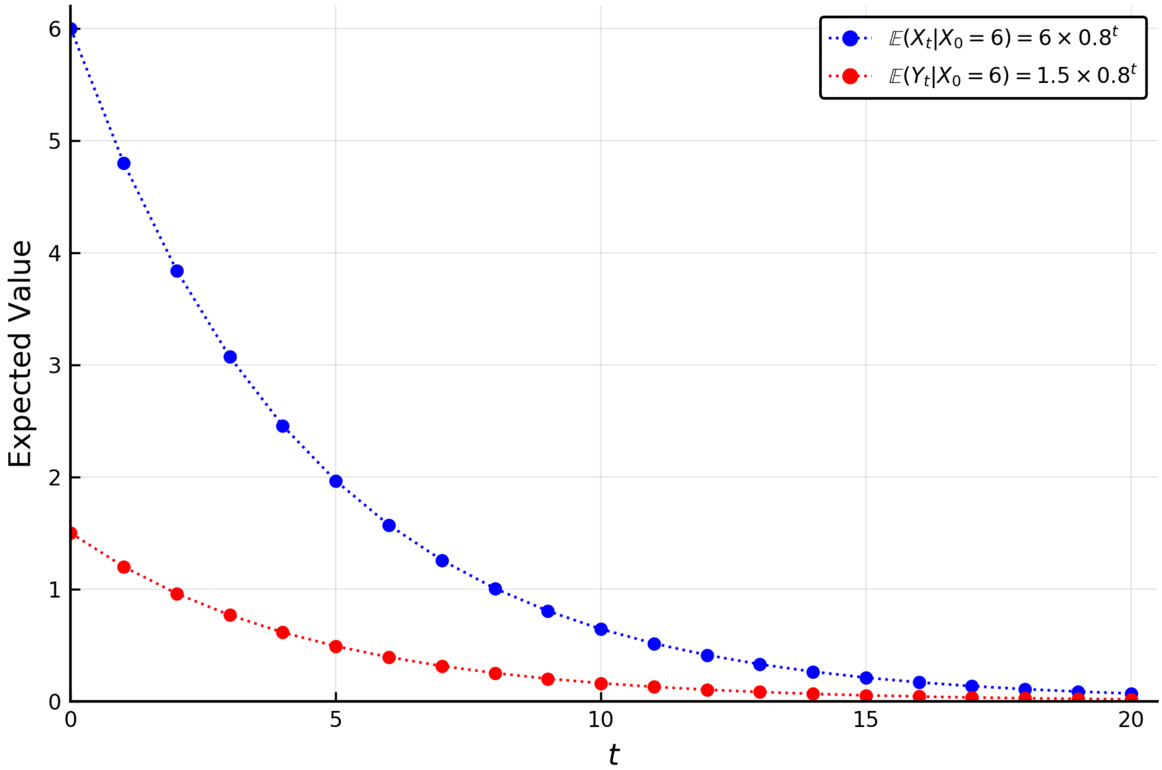

Figure 8: Plot of the expected values of X_t and Y_t given $X_0 = 6$ for the Greenwood model with $\alpha = 0.8$, obtained by computing $e_{x_0+1}^{\top} P^t v$. Compare this with Figure 1, which was generated differently.

Analysis for Part (c)

These results are identical to those of Question 1, because P^t applies the transition matrix P to the state vector v a total of t times, while the unit vector $e_{x_0+1}^{\top}$ selects the desired state. Because the state vector contains all possible numbers of susceptibles, and the transition matrix contains the probability of a transition between each pair of states, the expectation $\mathbb{E}(X_t)$ is merely the product of the transition matrix with the state vector t times, with all the unused states filtered out.

Analysis for Part (d)

The expected value of Y_t was calculated directly from the expected value of X_t by using the formula $Y_t = X_{t-1} - X_t$, and taking the conditional expectation of both sides with respect to $X_0 = 6$. This gives

$$\mathbb{E}(Y_t \,|\, X_0 = 6) = \mathbb{E}(X_{t-1} \,|\, X_0 = 6) - \mathbb{E}(X_t \,|\, X_0 = 6).$$

Unfortunately, there's no way to determine $\mathbb{E}(Y_0 \,|\, X_0 = 6)$ using this method, because $\mathbb{E}(X_{-1} \,|\, X_0 = 6)$ is undefined. However, the Greenwood model is unaffected by the choice of $\mathbb{E}(Y_0 \,|\, X_0 = 6)$ for $t \geq 1$, so $\mathbb{E}(Y_0 \,|\, X_0 = 6)$ was given the same value as in Question 1, where the equation for $\mathbb{E}(Y_0 \,|\, X_0 = 6)$ stipulated that it must be equal to 1.5.

Julia Code

```
using Plots, LaTeXStrings; pyplot(dpi=300)

alph = 0.8
x0 = 6
tmax = 20
t = [0:tmax;]

# Part (c)

e = zeros(Int64, x0 + 1)
e[x0 + 1] = 1

v = [0:x0;]

P = zeros(x0+1, x0+1)
for i = 1:x0+1
    for j = 1:x0+1
        P[i, j] = binomial(i-1,j-1) * (1-alph)^((i-1)-(j-1)) * alph^(j-1)
    end
end

EX = zeros(tmax + 1)
for i in 0:tmax
    EX[i + 1] = transpose(e) * P ^ i * v
end

plot(t, EX, label = L"\mathbb{E}(X_t | X_0=6) = 6 \times 0.8^t",
    xlims = (0, tmax+0.5),  ylims = (0, 6.2),
    xlabel = L"t", ylabel = "Expected Value",
    c = :blue, ls = :dot, m = (:dot, 5, Plots.stroke(0)))

# Part (d)

EY = zeros(tmax + 1)
EY[1] = 1.5
for i in 1:tmax
    EY[i + 1] = EX[i] - EX[i + 1]
end

plot!(t, EY, label = L"\mathbb{E}(Y_t | X_0=6) = 1.5 \times 0.8^t",
    c = :red, ls = :dot, m = (:dot, 5, Plots.stroke(0)))
```

Question 4

Part (a)

For the Greenwood model, $X_t \sim \mathrm{Bin}(X_{t-1}, \alpha)$. Hence, the entries of the transition matrix are given by

$$\begin{aligned} p_{ij} &\equiv \mathbb{P}(X_t = j \mid X_{t-1} = i, Y_{t-1} > 0) \\ &= \binom{i}{j}(1-\alpha)^{i-j}\alpha^j, \end{aligned}$$

where $p_i^0 = \mathbb{P}(X_0 = i) = \delta_{x_0 i}$ is the initial condition, and the constraint that $Y_{t-1} > 0$ ensures that the epidemic hasn't ended, which occurs precisely when there are no more infectives (i.e., when $Y_{t-1} = 0$). By the law of total probability,

$$\begin{aligned} p_j^t &\equiv \mathbb{P}(X_t = j, Y_t > 0) \\ &= \sum_{i=0}^{x_0} \mathbb{P}(X_t = j, Y_t > 0 \mid X_{t-1} = i, Y_{t-1} > 0)\, \mathbb{P}(X_{t-1} = i, Y_{t-1} > 0). \end{aligned}$$

When $i < j$, there are more susceptibles at time t than at time $t-1$, which is only possible if people are added to the model. This violates the assumptions of the model, so all of the probabilities with $i < j$ are equal to zero. Similarly, when $i = j$, there are no new infectives at time t, so $Y_t = 0$ and the epidemic ends. Because the first probability in the summation requires that $Y_t > 0$, it must be equal to zero when $i = j$. Hence, the lower boundary of the summation is $i = j + 1$. Because $Y_t > 0$, there is at least one new infective in each iteration, which means that there is at least one fewer susceptible in each iteration. This means that the probabilities with $x_0 - (t-2) \le i \le x_0$ are zero, so the upper boundary can be reduced by the number of iterations past $t = 1$. This implies that $j + 1 \le i \le x_0 - (t-1)$, so the equation becomes

$$\begin{aligned} p_j^t &\equiv \mathbb{P}(X_t = j, Y_t > 0) \\ &= \sum_{i=j+1}^{x_0-(t-1)} \mathbb{P}(X_t = j, Y_t > 0 \mid X_{t-1} = i, Y_{t-1} > 0)\, \mathbb{P}(X_{t-1} = i, Y_{t-1} > 0). \end{aligned}$$

Because $i \neq j$ in the summation, the stipulation that $Y_t > 0$ in the first probability is redundant, and can be dropped. Then, the first probability is equal to p_{ij}, while the second probability is equal to p_i^{t-1}. This gives

$$\begin{aligned} p_j^t &\equiv \mathbb{P}(X_t = j, Y_t > 0) \\ &= \sum_{i=j+1}^{x_0-(t-1)} \mathbb{P}(X_t = j \mid X_{t-1} = i, Y_{t-1} > 0)\, \mathbb{P}(X_{t-1} = i, Y_{t-1} > 0) \\ &= \sum_{i=j+1}^{x_0-(t-1)} p_{ij} p_i^{t-1} \\ &= \sum_{i=j+1}^{x_0-(t-1)} p_i^{t-1} p_{ij}. \end{aligned}$$

Part (b)

Using the recursive relationship $\Gamma(k, n \mid x_0) = p_{x_0-k}^{n-1}\alpha^{x_0-k}$ gives $\mathbb{P}(W > 4) \approx 0.1303$.

Julia Code

```
alph = 0.8
x0 = 6

P = zeros(x0+1, x0+1)
for i = 1:x0+1
    for j = 1:x0+1
        P[i, j] = binomial(i-1,j-1) * (1-alph)^((i-1)-(j-1)) * alph^(j-1)
    end
end

function p(j, t)
    if t == 0
        return j == x0 ? 1 : 0
    elseif j + 1 > x0 - (t - 1)
        return 0
    else
        sum(p(i, t-1) * P[i+1,j+1] for i in j+1:x0-(t-1))
    end
end

function Gamma(k, n)
    return p(x0-k, n-1) * alph ^ (x0 - k)
end

prob1a = 1 - sum(Gamma(k, n) for k in 0:4, n in 1:x0+1)
prob1b = sum(Gamma(k, n) for k in 5:6, n in 1:x0+1)

print(prob1a)
print(prob1b)
```

Part (c)

Using a Monte Carlo simulation with 10^6 trajectories gives $\mathbb{P}(W > 4) \approx 0.1307$, which is within 0.34% of the result from Part (b).

Julia Code

```
using Distributions, Random;

Random.seed!(1)

alph = 0.8
x0 = 6
n = 10^6

results = zeros(Int64, 1, n)

for i = 1:n
    x = x0
    y = 1

    while x > 0 && y > 0
        x_new = rand(Binomial(x, alph))
```

```
        y_new = x - x_new
        x = x_new
        y = y_new
    end
    results[i] = x0 - x
end

prob2 = mean(results .> 4)

print(prob2)
```

Part (d)

Using the PGF computations from subsection 4.1.1 gives

$$\Psi_W(\varphi) \approx 0.03602\varphi^6 + 0.09428\varphi^5 + 0.14358\varphi^4 + 0.16849\varphi^3 + 0.16663\varphi^2 + 0.12885\varphi + 0.26214.$$

By adding the coefficients of φ^5 and φ^6, it can be seen that $\mathbb{P}(W > 4) \approx 0.1303$, which is within $10^{-12}\%$ of the result from Part (b) and within 0.34% of the result from Part (c). Hence, all three results are identical to two decimal places.

Julia Code

```
using LinearAlgebra, SymPy;

alph = 0.8
x0 = 6

A = zeros(Int64, x0 + 1)
A[7] = 1

QE = zeros(x0+1)

for i in 0:x0
    QE[i+1] = alph ^ i
end

phi = Sym("φ")

Pbarφ = sympy.zeros(x0+1, x0+1)
for i = 1:x0+1
    for j = 1:x0+1
        if i > j
            Pbarφ[i, j] = binomial(i-1,j-1)*((1-alph)*phi)^((i-1)-(j-1))*alph^(j-1)
        end
    end
end

ΨWφ = transpose(A) * (sympy.eye(x0 + 1) - Pbarφ)^-1 * QE

prob3 = 0.0360175726193908 + 0.0942807848837342

print(prob3)
```

Question 5

Part (a)

When $t = 0$, the state space is

$$S_0 = \{(x_0, y_0)\}.$$

The infectives are then removed, so only x_0 people are left. Hence, when $t = 1$, the state space is

$$S_1 = \bigcup_{i=0}^{x_0} \{(i, x_0 - i)\}.$$

The infectives are then removed again, and if there are no new infectives, the epidemic ends. Similarly, if everyone gets infected, there are no susceptibles left to catch the infection. Hence, when $t = 2$, the state space is

$$S_2 = \bigcup_{i=1}^{x_0-1} \bigcup_{j=0}^{i} \{(j, i - j)\}.$$

The infectives are removed again, so when $t = 3$, the state space is

$$S_3 = \bigcup_{i=1}^{x_0-2} \bigcup_{j=0}^{i} \{(j, i - j)\}.$$

Iterating this through to $t = x_0 + 1$ shows that $S_{x_0+1} = \emptyset$, and there are no new items in the state space. Hence, the complete state space for all t is given by

$$S = \bigcup_{t=0}^{x_0} S_t = \left(\bigcup_{i=1}^{x_0} \bigcup_{j=0}^{i} \{(j, i - j)\} \right) \cup \{(x_0, y_0)\}.$$

Part (b)

The Reed–Frost model has $x_0 + 1$ communicating classes, which are given by

$$\{(0, 0)\}, \{(1, 0)\}, \{(2, 0)\}, \{(3, 0)\}, \ldots, \{(x_0, 0)\}.$$

These classes correspond to the diagonal entries of the submatrix P_{00}, where $Y_{t+1} = Y_t = 0$ and $X_{t+1} = X_t = m$ for some $m \in \{0, 1, \ldots, x_0\}$. All of these classes are recurrent, because there's no way to leave them after entering them.

Part (c)

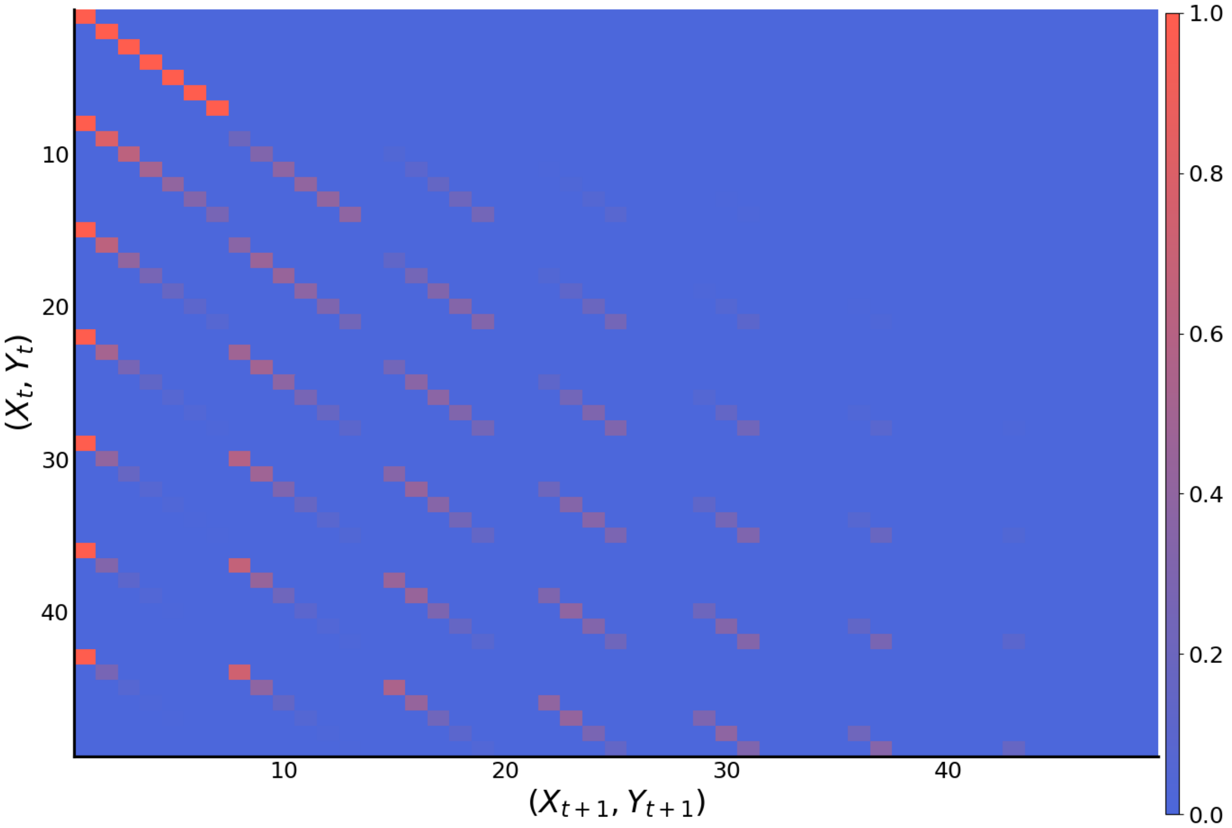

Figure 9: Heat map of the transition matrix for the Reed–Frost model with $\alpha = 0.8$ and $x_0 = 6$.

Julia Code

```
using Plots, LaTeXStrings, BlockArrays; pyplot(dpi=300)

alph = 0.8
x0 = 6

x0plusones = (x0+1)*ones(Int64,x0+1)

P = BlockArray(zeros((x0+1)^2, (x0+1)^2), x0plusones, x0plusones)
for i = 0:x0
    for j = 0:x0
        Pij = zeros(x0+1, x0+1)
        for k = 0:x0
            for l = 0:x0
                if j + l == k
                    Pij[k+1,l+1] = binomial(k, l) * (1 - alph^i)^j * alph^(i*l)
                end
            end
        end
        setblock!(P, Pij, i+1, j+1)
    end
end

heatmap(P, c = cgrad([:royalblue, :tomato]), yflip = true,
    xlabel = L"X_{t+1}", ylabel = L"X_t")
```

Part (d)

Using a Monte Carlo simulation with 10^6 trajectories gives $\mathbb{P}(W > 4) \approx 0.2566$, which is almost double the result for Question 4(c). This is because the Reed–Frost model, used by this question, is based on a different set of assumptions to the Greenwood model, used by Question 4(c). The Greenwood model assumes that the cause of infection is independent of the number of infectives, whereas the Reed–Frost model assumes that they are exponentially related. Consequently, the number of people infected grows exponentially faster in the Reed–Frost model than in the Greenwood model, so the total number of people infected during the outbreak is significantly larger on average.

Julia Code

```
using Distributions, Random;

Random.seed!(1)

alph = 0.8
x0 = 6
n = 10^6

results = zeros(Int64, 1, n)

for i = 1:n
    x = x0
    y = 1

    while x > 0 && y > 0
        x_new = rand(Binomial(x, alph^y))
        y_new = x - x_new
        x = x_new
        y = y_new
    end
    results[i] = x0 - x
end

prob4 = mean(results .> 4)

print(prob4)
```

Question 6

Scenario

During a COVID-19 pandemic involving travel restrictions and other security measures, all new arrivals to a regional town must be quarantined at a small motel. Because there are dozens of new arrivals to the town each day, there is never a shortage of people entering the motel. The probability of any new arrivals being infected is 0.05, and no one has any immunity to COVID-19.

Whenever the motel is empty, it accepts enough new individuals to fill its capacity, and these new individuals are tested on the spot for COVID-19. The next day, the test results are returned, and if anyone tests positive to COVID-19, they are removed from the motel and sent to a health centre. Because the remaining individuals may have caught COVID-19 from the infected individuals while the test results were being analysed, they are given another test for COVID-19, and the process repeats itself until all individuals in the motel test negative to COVID-19. When this happens, all the remaining individuals are released from the motel and sent back into the community, and a new batch of people arrives to take their place. Hence, the motel is at full capacity again, and the process repeats itself.

Model

Both the Greenwood model and the Reed–Frost model were used to model the spread of COVID-19 throughout the motel, and to determine the long-term rate of infection. In both cases, the parameter $\eta = 0.05$ was used for the initial rate of infection among new people in the motel, while $\alpha = 1 - p\beta = 1 - 0.1 \times 0.05 = 0.995$ was used to calculate subsequent rates. For the Greenwood model, the random variables were given by $X_{t+1} \sim \mathrm{Bin}(X_t, \alpha)$ and $Y_{t+1} = X_t - X_{t+1}$, while for the Reed–Frost model, they were given by $X_{t+1} \sim \mathrm{Bin}(X_t, \alpha^{Y_t})$ and $Y_{t+1} = X_t - X_{t+1}$, where X_t is the number of susceptibles at time t and Y_t is the number of infectives at time t.

Both models were simulated in Julia by using the Monte Carlo method with $n = 1000$ iterations and different capacities x_0 of the motel. Each iteration was performed over $m = 1000$ generations of people, but because a generation would last for multiple days if a person tested positive to COVID-19, the final value of t was always slightly bigger than m. The values of m and n were chosen to be 1000 because this is the square root of a million, which is the maximum number of iterations that can be performed on the simulation computer without overloading it.

Results

Despite the differences between the Greenwood model and the Reed–Frost model, the results for both of them were almost identical. This is probably because the rate of infection was very low, which made it difficult for the virus to spread quicker by infecting multiple people. The mean number of infected people at the motel on any given day is shown in the following table:

Motel Capacity (x_0)	1	2	3	4	5	6	7	8	9	10
Greenwood Model	0.050	0.092	0.133	0.171	0.207	0.242	0.274	0.306	0.336	0.365
Reed–Frost Model	0.050	0.092	0.133	0.171	0.207	0.242	0.275	0.307	0.337	0.367

This shows that as the maximum number of people staying at the motel increases, the mean number of people infected by COVID-19 also increases, which isn't surprising given that none of them have any immunity to COVID-19. Accounting for the number of people who were at the motel on any given day, the rate at which any individual could expect to be infected is shown in the following table:

Motel Capacity (x_0)	1	2	3	4	5	6	7	8	9	10
Greenwood Model	0.050	0.046	0.044	0.043	0.042	0.040	0.039	0.038	0.038	0.037
Reed–Frost Model	0.050	0.046	0.044	0.043	0.042	0.041	0.040	0.039	0.038	0.037

This means that as the capacity of the motel increases, the rate at which people get infected decreases, which seems counterintuitive at first, because infections usually spread faster when there are more people, instead of slower. However, this decrease is due to the fact that although the rate of infection amongst new arrivals is $\eta = 0.05$, the rate of infection amongst people staying at the motel is $1 - \alpha = 0.005$ for the Greenwood model and $1 - \alpha^{Y_t}$ for the Reed–Frost model. The latter equates to 0.005 when $Y_t = 1$, 0.010 when $Y_t = 2$, 0.015 when $Y_t = 3$ and 0.049 when $Y_t = 10$, which are all less than $\eta = 0.05$. Hence, it is mathematically impossible for an individual's risk of infection to increase as x_0 increases, or even to stay the same, unless a different model is adopted.

Discussion

Many of the assumptions used in this analysis are unrealistic. For example, it assumes that no one has immunity to COVID-19, even though there's a good chance that at least some people would be immune to it. It also assumes that the rate of infection amongst new arrivals is $\eta = 0.05$, and either $1-\alpha = 0.005$ or $1-\alpha^{Y_t}$ for people staying in the motel, even though the rates of infection for COVID-19 are known to be much higher than this. The Greenwood model also assumes that the rate of infection is constant, even though the rate of infection of COVID-19 is known to grow exponentially. Hence, the Reed–Frost model is more accurate than the Greenwood model, and would differ significantly from the Greenwood model if η, p and β were more realistic.

Another unrealistic assumption is that there are dozens of people arriving in the town, but that they can only stay in one particular motel, which doesn't have enough room for them all. Hence, it is unclear where all the other people go, and if they were staying in other motels, this motel wouldn't necessarily be filled whenever it takes in new arrivals. There's also the possibility that some visitors will visit the town multiple times, and will be required to stay in the motel multiple times, so their infection rate might be different from those who are visiting the town for the first time. Finally, this model assumes that COVID-19 has an incubation period of one day, when it is known to have an incubation period of at least 14 days. Hence, there's a high probability that some people who have COVID-19 will test negative to it, and will inadvertently pass it on to other people when they leave the motel.

Julia Code

```
using Plots, LaTeXStrings, Distributions, Random; pyplot()

Random.seed!(1)

x0 = 20
eta = 0.05
n = 10^3
m = 10^3

pp = 0.1
beta = 0.05
alph = 1 - pp * beta

x_tot = zeros(1, n)
y_tot = zeros(1, n)
total = zeros(1, n)
prop1 = zeros(1, n)
```

```
prop2 = zeros(1, n)

for i = 1:n
    x_res = Int64[]
    y_res = Int64[]
    tot_res = Int64[]

    for j = 1:m
        x = x0
        x_new = rand(Binomial(x, 1 - eta))
        y_new = x - x_new
        append!(x_res, x_new)
        append!(y_res, y_new)
        append!(tot_res, x)
        x = x_new
        y = y_new
        if (x == x0)
            continue
        end

        while x > 0
            # Leave as is for the Greenwood model
            # Replace "alph" with "alph^y" for the Reed-Frost model
            x_new = rand(Binomial(x, alph))
            y_new = x - x_new
            append!(x_res, x_new)
            append!(y_res, y_new)
            append!(tot_res, x)
            if (x_new == x)
                break
            end
            x = x_new
            y = y_new
        end
    end

    x_tot[i] = mean(x_res)
    y_tot[i] = mean(y_res)
    total[i] = mean(tot_res)

    # Number of infectives/day (flow of the infection)
    prop1[i] = mean(y_res)

    # Number of infectives/totalpeople/day (infection rate)
    prop2[i] = mean(y_res ./ tot_res)
end

print(mean(x_tot))
print(mean(y_tot))
print(mean(total))
print(mean(prop1))
print(mean(prop2))
```