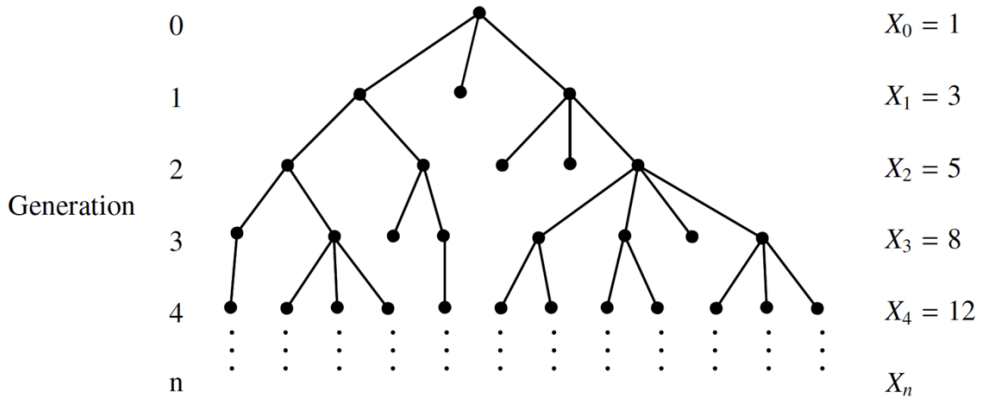
BIM 105
Fall 2024

Lab 5: Branching Processes

Instructions: Read the prompts and use MATLAB to answer the questions. Submit your solution as a single script. A template script is available on Canvas.

Part 1. Background

Characterizing the spread of a communicable disease through a population can often be useful in understanding the disease and controlling epidemics. At a basic level, disease spread can be modeled as a branching process. Starting with a single infected individual, the disease can transmit to a number of other individuals in the population (secondary infections). Then, each of those infected individuals may transmit the disease to others in the population, and so on. The result of this process is a "family tree" of disease progression from the initial case, as illustrated below.



(image from Bob Pego[1], "Coagulation dynamics in branching processes", CMU)

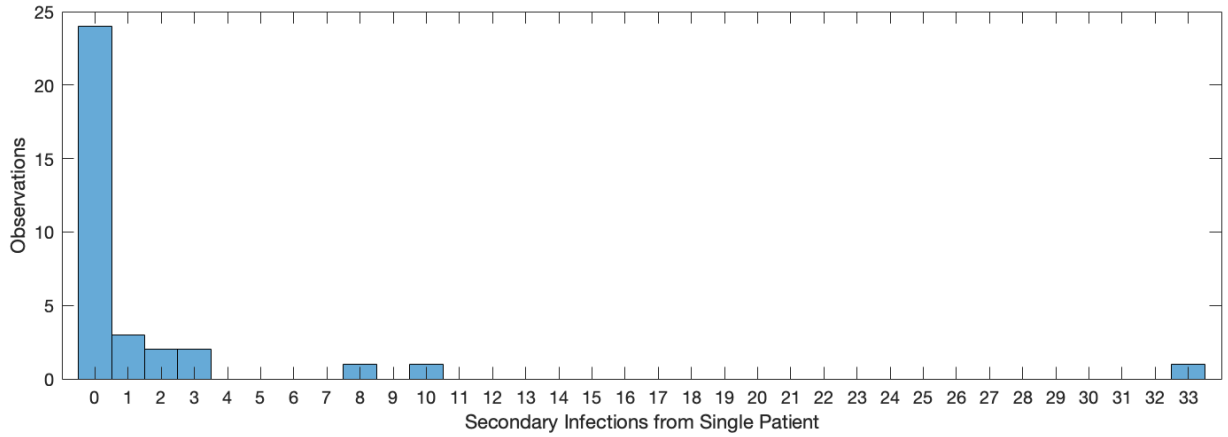
The number of secondary infections stemming from a single individual can be considered as a random variable and is referred to as the *offspring distribution*. A single person may infect a large number of people, or a single person may not infect anyone else at all. It's the properties of the offspring distribution that dictate the behavior of disease spread in a branching process. If the disease transmits very scarcely, that single infection will not likely lead to a large epidemic. However, if the disease tends to transmit to at least one other person from each infection, the population could be in trouble.

[1] https://math.psu.edu/events/16205

Part 2. Case Study: SARS-CoV-1 in Singapore

The Singapore Ministry of Health and World Health Organization (WHO) conducted an investigation into the 2003 Singapore SARS (severe acute respiratory syndrome) outbreak[2] which infected 238 people. As a part of their study, they compiled statistics on the number of secondary infections from individual patients to glean at the offspring distribution of the outbreak.

The number of secondary SARS infections from 34 individuals was tabulated, and the distribution of secondary infections is below:



The largest number of secondary cases linked to one patient was 33, but most patients did not transmit to anyone.

One model for this offspring distribution could be a $\text{Poisson}(\lambda)$ distribution. Fitting a Poisson model to data is simple: the best estimate of the parameter λ is just the sample average (recall that a $\text{Poisson}(\lambda)$ distribution has mean λ). The sample mean is $\lambda = 1.88$, meaning that the best Poisson fit of the data is a $\text{Poisson}(1.88)$ distribution.

In the next section, we'll utilize this fitted Poisson as our offspring distribution in simulations of a branching process and characterization of outbreak behaviors.

[2] Centers for Disease Control and Prevention (CDC). (2003). Severe acute respiratory syndrome--Singapore, 2003. MMWR. Morbidity and mortality weekly report, 52(18), 405.

Starting with a single infected person, simulating an outbreak in MATLAB amounts to sampling secondary infections from each infected person and repeating this process to a desired time point. For instance, a function **`sim_branching_poisson(lambda, n_max)`** is shown below which implements this resampling procedure and returns the chain of secondary infections from the initial case until **`n_max`** generations are realized.

```
function X = sim_branching_poisson(lambda, n_max)

X = zeros(1, n_max+1); % Initialize infection counts to zero
X(1) = 1; % Single infection at X0

for n = 1:n_max

    counts = 0;

    for i = 1:X(n)
        counts = counts + poissrnd(lambda);
    end

    X(n+1) = counts; % Total number of secondary infections

end
end
```

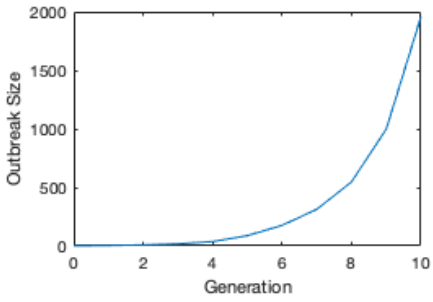
We can perform a single simulation of 10 generations with our Poisson fit by calling **`sim_branching_poisson(1.88, 10)`**. The output of the function is a vector containing the number of cases at each time point, and it might look something like this:

```
output =
     1   5   9   16   23   29   55   121 187   366   706
```

For this specific simulation, the outbreak steadily grows over the 10 iterations of the branching process.

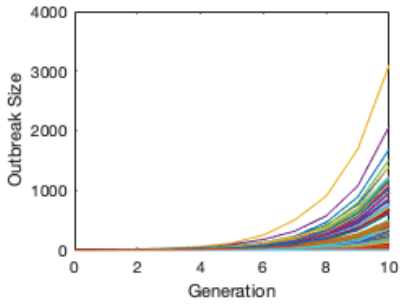
Question 1: Use MATLAB to perform a simulation of a branching process for a disease that spreads from an individual to others according to a $\text{Poisson}(1.88)$ distribution. Simulate to the 10^{th} generation. Plot the outbreak size over time.

```
X = sim_branching_poisson(1.88, 10);
figure;
p = plot(0:1:10, X);
xlabel('Generation')
ylabel('Outbreak Size')
```



Question 2: Use MATLAB to produce 100 independent simulations of this system. Plot all the trajectories of outbreak size in a single figure.

```
Xs = NaN(11, 100);
for i = 1:100
    Xs(:, i) = sim_branching_poisson(1.88, 10);
end
figure;
p = plot(0:1:10, Xs);
xlabel('Generation')
ylabel('Outbreak Size')
```



Question 3: What is the mean outbreak size at the 10th generation?

```
answers.q3 = mean(Xs(11, :)); -> ~500-650
```

Question 4: In what proportion of simulations did the outbreak stop by the 10th generation?

```
answers.q4 = sum(Xs(11,:)==0)/100; -> 20-25%
```

At each generation, we expect (on average) to see λ times *more* cases than the previous generation. In other words, the size of an outbreak at generation n would be λ^n. The total number of infections over time, then, would be

$$\sum_{n}^{t} \lambda^n = 1 + \lambda + \lambda^2 + \cdots + \lambda^t$$

Question 5: What is the expected total number of infections by the 10th generation given that $\lambda = 1.88$?

```
lambda = 1.88;
predicted_sum = sum(lambda.^(0:1:10)); -> 1177
```

Question 6: Use `cumsum` to compute the total number of infections by the 10th generation in your simulations. How does the distribution of total infections compare to your prediction in Question 5?

```
Xs_cum = cumsum(Xs);
mean_Xs_cum10 = mean(Xs_cum(end, :)); -> 1100-1200 (agrees with prediction)
```

The total number of infections in our simulations agrees with the prediction from Question 5.

Part 3. A Negative Binomial Approach

As we saw in the RNA-Seq lab, a Negative Binomial distribution generalizes the Poisson distribution (by compounding it with the Gamma distribution) to allow for flexibility in the variance. Let's try fitting the observed offspring data with this distribution instead of a Poisson.

The counts from the offspring distribution are available on Canvas in a file called **`secondary_infections.mat`**. Download this file and use the **`load`** function in MATLAB to load the **`counts`** vector into your script.

Question 7: Use the MATLAB function **`nbinfit`** to estimate r and p for these data.

7a: What values of r and p does the fit give you?

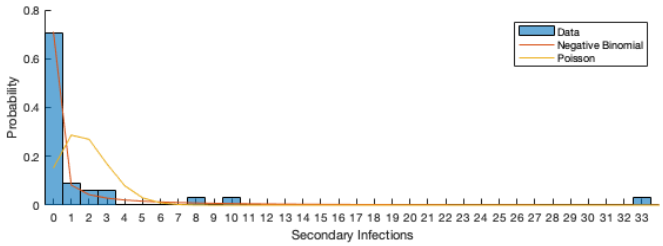
```
load('secondary_infections.mat');
params = nbinfit(counts);
answers.q9a = params; -> r=0.1126, p=0.0612
```

7b: Plot a histogram of the data. Set the **`'Normalization'`** property of the **`histogram`** function to **`'probability'`** (this normalizes the histogram such that its values sum to 1). Then, overlay a plot of a Negative Binomial probability distribution with the fitted parameters r and p.

```
f = figure;
hold on
histogram(counts, 'Normalization', 'probability');
plot(0:1:35, nbinpdf(0:1:35, params(1), params(2)));
xlim([-0.5 34])
xticks(0:1:33)
xlabel('Secondary Infections')
ylabel('Probability')
```

7c: Now overlay the Poisson distribution from earlier where our best fit was $\lambda = 1.88$. Of the two distributions (Poisson and Negative Binomial), which one do you think better captures the offspring distribution?

```
plot(0:1:35, poisspdf(0:1:35, 1.88))
legend('Data', 'Negative Binomial', 'Poisson');
NB appears to be a better fit.
```



A Negative Binomial seems to be a better model.

We now have two models of the offspring distribution. In Part 2, we investigated the behavior of outbreaks assuming that the offspring distribution was Poisson. To investigate the consequences of a Negative Binomial, we can again use the simulations and analytical tools we applied to the Poisson process. It's again convenient to define another function to simulate the branching process according to the new model.

```
function X = sim_branching_nbin(r, p, n_max)

X = zeros(1, n_max+1); % Initialize infection counts to zero
X(1) = 1; % Single infection at X0

for n = 1:n_max

    counts = 0;

    for i = 1:X(n)
        counts = counts + nbinrnd(r, p);
    end

    X(n+1) = counts; % Total number of secondary infections

end
end
```

Question 8: Repeat the simulations from Part 3, but this time use the fitted Negative Binomial distribution from Question 10. Generate 100 simulations and answer the following questions:

8a: What is the mean outbreak size at the 10th generation?

```
Xs_NB = NaN(11, 100);
for i = 1:100
    Xs_NB(:,i) = sim_branching_nbin(params(1), params(2), 10);
end
figure;
p = plot(0:1:10, Xs_NB);
xlabel('Generation')
ylabel('Outbreak Size')
answers.q8a = mean(Xs_NB(11,:)); -> 400-700, should be similar to
Poisson, but larger variance
```

8b: In what proportion of simulations did the outbreak stop by the 10th generation?

```
answers.q8b = sum(Xs_NB(11,:)==0)/100; -> ~90%
```

8c: Use **`cumsum`** to compute the total number of infections by the 10th generation in your simulations.

```
Xs_NB_cum = cumsum(Xs_NB);
mean_Xs_NB_cum10 = mean(Xs_NB_cum(end,:));
answers.q8c = mean_Xs_NB_cum10; -> 900-1300
```

8d: How does the total number of infections compare to results from simulations that used the Poisson offspring distribution? What change is observed in the proportion of outbreaks that end by the 10th generation?

```
answers.q8d = ['The total outbreak sizes are comparable, but the proportion ' ...
    'of outbreaks that end by the 10th generation is much higher in the NB model.'];
```

The total outbreak sizes are comparable, but the proportion of outbreaks that end by the 10th generation is much higher in the NB model'