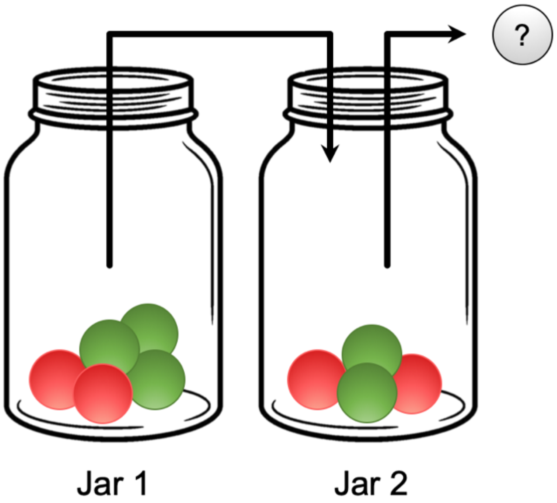
MAT/BIS 107
Spring 2021

Lab 2: Probabilities, Bayes' Formula, Estimating Population Size

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

Part 1. Conditional Probability and Bayes' Formula[1]

Consider the possible outcomes of the following scenario. You have two jars – one jar contains 2 red balls and 3 green balls, and the other jar contains 2 red balls and 2 green balls. One ball from Jar 1 is selected at random and moved to Jar 2, but its color is not known. One ball from Jar 2 is then selected at random.



Suppose you wanted to compute the probability that the ball selected from Jar 2 is green. In our scenario, we don't know for sure the color of the ball that was moved from Jar 1 to Jar 2. In any case, we know that the ball that was transferred must have been either red or green. We can compute the probability of selecting a green ball from Jar 2, *conditioned* on each of these possibilities. In other words, if the ball transferred to Jar 2 was green, the probability of selecting a green ball from Jar 2 would be:

$$\mathrm{P(green \mid green\ ball\ transferred)} = \frac{\#\ \text{green balls in Jar 2 after transfer}}{\#\ \text{total balls in Jar 2 after transfer}} = \frac{3}{5}$$

Similarly, if the ball transferred to Jar 2 was red, the probability of selecting a green ball from Jar 2 would be:

[1] Example from D.V. Sarwate, University of Illinois at Urbana-Champaign. ECE 313.

$$\mathrm{P(green \mid red\ ball\ transferred)} = \frac{\#\ \mathrm{green\ balls\ in\ Jar\ 2\ after\ transfer}}{\#\ \mathrm{total\ balls\ in\ Jar\ 2\ after\ transfer}} = \frac{2}{5}$$

These are *conditional probabilities* in that they are *conditioned* on a particular event. In reality, we do not know which color ball was transferred into Jar 2. However, if we wanted to compute the probability that the ball drawn from Jar 2 is green in the absence of knowing which ball was transferred, we can use the **theorem of total probability**. More formally, for our example:

$$\begin{aligned}\mathrm{P(green)} = \mathrm{P(green \mid green\ ball\ transferred)} \cdot \mathrm{P(green\ ball\ transferred)} \\ + \mathrm{P(green \mid red\ ball\ transferred)} \cdot \mathrm{P(red\ ball\ transferred)}\end{aligned}$$

We can see that we are weighting each of the conditional probabilities by the probability that the conditioned-on event occurred. This theorem is what allows us to find unconditional probabilities from conditional probabilities. Because there are 3 green and 2 red balls in Jar 1, the total probability of drawing a green ball from Jar 2 can then be computed as:

$$\mathrm{P(green)} = \frac{3}{5} \cdot \frac{3}{5} + \frac{2}{5} \cdot \frac{2}{5} = \frac{13}{25}$$

Now, what if we draw a green ball from Jar 2 and are curious which color ball was transferred from Jar 1. We don't know the color of the ball that was transferred from Jar 1, but we'd like to compute the probability of each color having been transferred. In other words, we'd like to compute $\mathrm{P(green\ ball\ transferred \mid green)}$ and $\mathrm{P(red\ ball\ transferred \mid green)}$.

In order to do this, we can leverage **Bayes' formula** (or Bayes' theorem, or Bayes' rule) which states

$$\mathrm{P(A \mid B)} = \frac{\mathrm{P(B \mid A)P(A)}}{\mathrm{P(B)}}$$

For our particular problem, we like to compute

$$\begin{aligned}\mathrm{P(green\ ball\ transferred \mid green)} &= \frac{\mathrm{P(green \mid green\ ball\ transferred)P(green\ ball\ transferred)}}{\mathrm{P(green)}} \\ &= \frac{(3/5)(3/5)}{(13/25)} = 9/13\end{aligned}$$

In other words, if we drew a green ball from Jar 2, there's about a 69.2% chance a green ball was transferred from Jar 1 to Jar 2.

<u>Question 1</u>: What is the probability of a red ball having been transferred from Jar 1 to Jar 2 if the color of the ball drawn from Jar 2 is red?

This example with two jars is relatively simple, but how do we know that our application of the laws of probability are correct? Let's use MATLAB to simulate this system a large number of times and confirm that our assessment is correct. For instance, we computed the probability of drawing a green ball from Jar 2 to be 13/25, which is 0.52. Do simulations of the system result in a green ball being drawn 52% of the time?

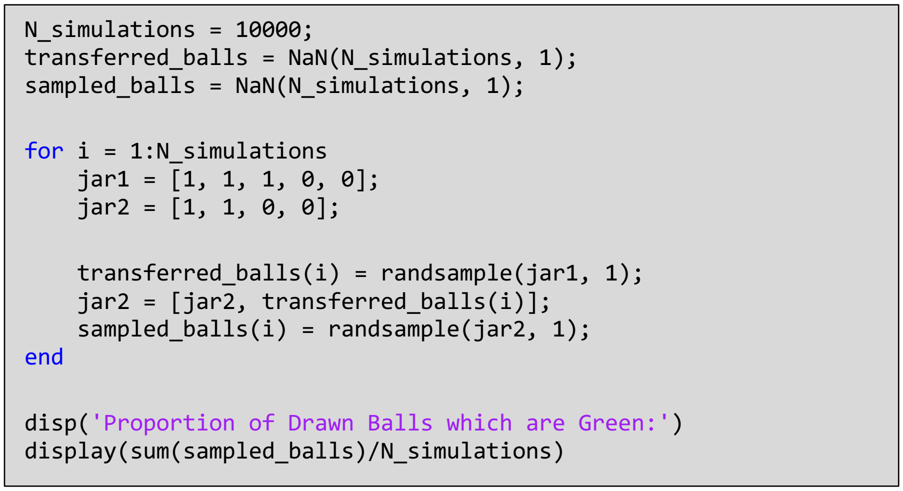
To set up a simulation, let's first decide how many simulations to perform. The larger the number the better, but there are computational limits, of course. Let's use 10,000 simulations to start. Next, we need to use MATLAB to set up the two jars, and we need to figure out how to sample from one and transfer the result to the other.

There are many ways to represent the two jars in MATLAB. A simple implementation is to define each jar as a vector, and use zeros and ones to represent colors. For example, if we decide that a "1" represents green and a "0" represents red, we could set up the jars as:

```
jar1 = [1, 1, 1, 0, 0];
jar2 = [1, 1, 0, 0];
```

From there, we need to randomly select an element from **jar1**. MATLAB has a built-in function called **randsample** which can do this for us. To sample one element from **jar1** (i.e., sample one ball from Jar 1), we would call **randsample(jar1, 1)**. The second argument tells the function how many elements we'd like to draw.

Bringing these methods together, the following code performs 10,000 simulations of our system. At the end, it displays the proportion of balls from Jar 2 that were green.



```
N_simulations = 10000;
transferred_balls = NaN(N_simulations, 1);
sampled_balls = NaN(N_simulations, 1);

for i = 1:N_simulations
    jar1 = [1, 1, 1, 0, 0];
    jar2 = [1, 1, 0, 0];

    transferred_balls(i) = randsample(jar1, 1);
    jar2 = [jar2, transferred_balls(i)];
    sampled_balls(i) = randsample(jar2, 1);
end

disp('Proportion of Drawn Balls which are Green:')
display(sum(sampled_balls)/N_simulations)
```

<u>Question 2</u>: Run the code in MATLAB. What proportion of balls drawn from Jar 2 were green? Does this agree with our computed probability earlier?

Question 3: We used Bayes' formula to compute the probability of the transferred ball being green having sampled a green ball from Jar 2. The following command computes the proportion of green draws from Jar 2 that coincide with a green ball being transferred from Jar 1. Add the command to the end of the code and run it – does the proportion agree with our probability calculated earlier?

```
prob_green_transferred_given_green_draw =
      sum(sampled_balls(transferred_balls == 1))/sum(sampled_balls)
```

Part 2. Estimating Population Size with Animal Tagging

One method of estimating animal populations sizes is called “mark-recapture,” wherein a group of animals are caught, marked in some manner (ear tags, RFID chip, etc.), and then returned to their environment. After some time has passed (long enough that the tagged animals can sufficiently “mix” back into the population, yet short enough that the overall population level has not changed drastically), a number of animals are then captured from the population. One might claim that the proportion of captured animals (from the second capture) which are tagged is approximately the same as the proportion of tagged animals in the entire system. That is to say, with high probability:

$$\frac{\text{\# of tagged animals in second capture}}{\text{\# total animals in second capture}} \approx \frac{\text{\# of tagged animals in total system}}{\text{\# total animals in total system}}$$

Given that we know the number of tagged animals from the first capture, performing a second capture allows us to estimate the overall population size by solving this equality. For instance, suppose an ecologist is assessing the size of a population of fish in a lake. They catch and tag 15 fish, return them to the population, then resample 15 fish again, 5 of which are found to be tagged. The total population size could then be estimated by:

$$\frac{\text{5 tagged fish in second capture}}{\text{15 total fish in second capture}} \approx \frac{\text{15 tagged fish in total population}}{x \text{ fish in total population}} \Rightarrow x = 45$$

Because a third of fish in the second capture are tagged, we assume a third the total population is also tagged, which leads us to a population size of 45. This is the general schematic for estimating population size from a mark-recapture experiment.

Of course, there is some variability in that the number of animals you capture will not necessarily be representative of the entire population. In other words, there is some probability that these ratios will not be identical.

<u>Question 4</u>: Suppose that in a mark-recapture experiment, 10 animals are captured, tagged, and returned to the population. In a second capture again with 10 animals, two animals are found to be tagged. What is the estimated population size from this experiment?

<u>Question 5</u>: Suppose that in a mark-recapture experiment, 10 animals are captured, tagged, and returned to the population. In the second capture, zero animals are found to be tagged. What does this tell us about the size of the population?

Now that we understand how mark-recapture can give us an estimated population size, let’s look at a specific example. Suppose again that an ecologist is assessing a population of fish within a lake. Let there be 100 fish in the lake; 20 of them are caught, marked, and returned by the ecologist. At a later point in time, 15 fish are caught, and the number that are tagged are found to be 5.

There are a lot of ways to collect 15 fish from a population of 100 (assuming the fish are all distinguishable). The exact number is $\binom{100}{15}$. Then, however, consider how many ways there are to collect 15 fish from the lake where exactly 5 of them are tagged (as in the example). This number could be computed as $(\text{#of ways to sample 5 tagged fish}) * (\text{#of ways to sample 10 untagged fish})$. In our example there are 20 tagged fish and 80 untagged fish, so the number would be $\binom{20}{5}\binom{80}{10}$.

Given the number of ways to sample 5 tagged fish and the total number of ways to sample any 15 fish, we can now compute the probability of sampling 5 tagged fish in the second capture. This is because each possible sampling is equally-likely. The probability is therefore just the ratio between the number of ways to sample our observation and the total number of ways to sample, which is:

$$\text{Prob}(\text{sample 5 tagged fish}) = \frac{\binom{20}{5}\binom{80}{10}}{\binom{100}{15}} \approx 0.10$$

<u>Question 6</u>: What is the probability of collecting exactly 4 tagged fish in the second capture?

<u>Question 7</u>: What is the probability of collecting exactly 0 tagged fish in the second capture?

To generalize, first consider a lake with a population of N fish. Absent any tagging, consider how many ways there are to sample n during a capture. We can count the number of ways as $N * (N-1) * \ldots * (N-n+1)$, which is just $\binom{N}{n}$.

Now, in a mark-recapture experiment, B fish are caught and tagged in the first capture. In a second capture, n fish are sampled. The number of ways to sample n fish from the lake <u>where k of them have been tagged</u> can then be written as $\binom{N-B}{n-k}\binom{B}{k}$. The probability of sampling exactly k tagged fish in the second recapture is then the number of ways to collect k tagged fish in a sample of k fish divided by the total number of ways to collect n fish. Thus:

$$\text{Prob}(\text{sample } k \text{ tagged fish}) = \frac{\binom{N-B}{n-k}\binom{B}{k}}{\binom{N}{n}}$$

This is known as the <u>hypergeometric distribution</u>.

Having a general formula for the probability of an outcome enables us to explore the experiment computationally. In MATLAB, it'll be handy to define a function that returns this probability for any inputs N, B, n, and k. For example, you could use the following function:

```
function result = mark_recapture_prob(N, B, n, k)

result = nchoosek(N-B, n-k)*nchoosek(B, k)/nchoosek(N, n);

end
```

Alternatively, you could call MATLAB's built-in function **`hygepdf`**, which will provide the same result. Note that **`hygepdf`** function uses a different order of inputs than the function above, so be careful that you use the correct syntax. Type the command "**`help hygepdf`**" for more information on the usage of the function.

Having this formula allows us to explore the statistical properties of the mark-recapture experiment. For instance, for our example above where $N = 100$, $B = 20$, and $n = 15$, we could compute and plot the probability distribution over k. The code below does just that:

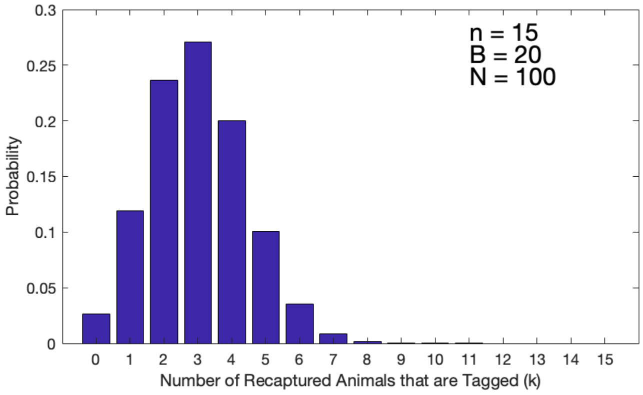
```
N = 100;
B = 20;
n = 15;
k = 0:15;

probs_k = zeros(16, 1);

for i = 0:15
    probs_k(i+1) = mark_recapture_prob(N, B, n, k(i+1));
end

figure(3)
bar(k, probs_k)
xticks(k)
xlim([-1 n+1])
xlabel('Number of Recaptured Animals that are Tagged (k)')
ylabel('Probability')
```

The code produces the following plot:



The mostly likely outcome is $k = 3$ for the system parameters provided. This makes some intuitive sense, as one-fifth of the population is tagged after the first capture and there are 15 total samples in the second round. The probability of recapturing zero tagged animals ($k = 0$) is small, but not insignificant. The distribution is mostly comprised by the outcomes where $k < 9$.

<u>Question 8</u>: Use MATLAB to investigate the probability of sampling 0 tagged animals in the second capture across different population sizes (*N*). Assume that 10 animals are tagged (*B*=10), and the second recapture always samples the same number of animals that were tagged (*n*=*B*). What is the overall trend you observe? What type of problem does this present?

Part 3. Sampling with Replacement

In Part 1, animals are captured, tagged, and released as a group. A second group is resampled from the population to estimate population size. Each of these groups (the tagged group and the resampled group of the second capture) are *sampled without replacement*. In other words, you cannot capture and tag the same animal more than once in the first round; and you cannot sample the same animal more than once in the second capture.

Suppose instead that a population of N fish are characterized in a slightly different manner. Imagine that B fish are caught, tagged, and released. Then, instead of performing a second capture all at once, fish are checked for tags one at a time, where each fish is checked and released back into the lake before continuing. Assume that at each sample, the fish are evenly dispersed. This process is performed until n fish are sampled.

Because the sampled fish are returned to the population and the population is evenly dispersed, each recaptured fish has the same probability to be tagged, which is B/N. Conversely, the probability of recapturing a single untagged fish is $(N-B)/N$.

If the sampling with replacement is repeated until n fish have been inspected, then the probability of observing exactly k tagged fish is given by the binomial distribution, which is

$$\text{Prob}(\text{sample } k \text{ tagged animals}) = \binom{n}{k}\left(\frac{B}{N}\right)^k\left(\frac{N-B}{N}\right)^{n-k}$$

The first term, $\binom{n}{k}$, is the number of ways to pick k tagged animals with n samples. The remaining part of the formula, $\left(\frac{B}{N}\right)^k\left(\frac{N-B}{N}\right)^{n-k}$, is the probability of observing k tagged animals in one specific configuration. Thus, the product of these quantities gives us the probability of observing k tagged animals in any ordering.

In MATLAB, computing the probability of an outcome using this distribution is as simple as calling **binopdf(k, n, B/N)**. As an example, consider the probability of recapturing 3 OR 4 tagged animals when $n = 10$, $B = 10$, and $N = 100$.

```
B = 10;
n = 10;
N = 100;

prob_3_or_4 = binopdf(3, n, B/N) + binopdf(4, n, B/N);
```

This computation gives us $\text{Prob}(k = 3 \text{ or } 4) = 0.0686$.

Consider a “perfect” outcome of k; in other words, the k which when divided by n produces the *exact* ratio of tagged/untagged animals in the population (and subsequently leads to a correct population size estimate). For example, if $N = 20$, $B = 5$, and $n = 4$, a “perfect” k would be $k = 1$. This is because a measurement of $k = 1$ would correspond to a proportion of tagged animals $k/n = 0.25$, which exactly equals the proportion in the entire population, B/N.

Question 9: For both sampling methods (without replacement: hypergeometric, with replacement: binomial), calculate the probability that the number of tagged fish recaptured is within 1 of a “perfect” measurement.

a. $N = 20$, $B = 5$, $n = 4$ (i.e., calculate the probability of k in {0, 1, 2})

b. $N = 1000$, $B = 50$, $n = 40$ (i.e., calculated the probability of k in {1, 2, 3})

c. $N = 1000$, $B = 200$, $n = 40$ (i.e., calculate the probability of k in {7, 8, 9})

Question 10: Does one method seem to be more reliable than the other when compared through this metric? If so, speculate as to why that may be the case.

Part 4. Simulations

Let’s set up a mock population in MATLAB and simulate the two sampling methods to observe how well they work as a predictor of population size. To initialize a population, we can use a vector of zeros. For example, using a population of *N* = 100:

```
N = 100;
population = zeros(N, 1);
```

To simulate the action of capturing and tagging 20 fish, we can sample positions in our population vector and assign the positions to be equal to 1. As such, each 1 in the population vector corresponds to a tagged fish.

```
tags = randsample(1:100, 20, false);
population(tags) = 1;
```

If we wanted to simulate one round of sampling *with* replacement, we could pick 15 random indices of the population vector. The function **`randsample`** can do this for us (note the third argument indicates if we want to sample with replacement, and is set to **`true`** below). We then count how many of these indices corresponded to a value of 1 to determine the number of tagged fish in the recapture. For example, the code below performs this sampling and computes the estimated total population size from the observed proportion of tagged fish:

```
B = 20; n = 15;
samples = randsample(1:N, n, true);
k_tagged = sum(population(samples));
r_estimated_wr = k_tagged/n;
N_estimated_wr = B/r_estimated_wr;
```

To simulate one round of sampling *without* replacement, we pick 15 random indices of the population vector as above; however, this time we specify in our call to **`randsample`** that the sampling should be performed without replacement by setting the third argument to **`false`**.

```
B = 20; n = 15;
samples = randsample(1:N, n, false);
k_tagged = sum(population(samples));
r_estimated_wor = k_tagged/n;
N_estimated_wor = B/r_estimated_wor;
```

Having implemented the process to simulate *one* experiment, we can then scale the code up to repeat the simulation multiple times. Writing the code to scale to more simulations would take too much time away from this lab, so here we provide a block of code that does it for you. The number of simulations is a parameter you can play with. Initially, we have set the number of simulations at 100.

```
N = 100;
B = 20;
n = 15;
population = zeros(N, 1);
tags = randsample(1:N, B, false); % Randomly sample animals to tag
population(tags) = 1; % Mark tagged animals

N_sims = 100; % Number of simulations

r_estimated_wr_i = NaN(N_sims, 1); % Estimates for ratio of tagged animals with replacement
r_estimated_wor_i = NaN(N_sims, 1); % Estimates for ratio without replacement

for i = 1:N_sims

    samples = randsample(1:N, n, true);
    k_tagged = sum(population(samples));
    r_estimated_wr_i(i) = k_tagged/n;

    samples = randsample(1:N, n, false);
    k_tagged = sum(population(samples));
    r_estimated_wor_i(i) = k_tagged/n;

end

N_estimated_wr = B./r_estimated_wr_i; % Population size estimates with replacement
N_estimated_wor = B./r_estimated_wor_i; % Population size estimates without replacement

figure()
histogram(N_estimated_wr, 0:5:5*N)
hold on
histogram(N_estimated_wor, 0:5:5*N)
plot([100, 100],[0, N_sims/2], '--r')
xlabel('Estimated Population Size')
ylabel('Number of Simulations')
legend('Sampling With Replacement','Sampling Without Replacement')
```

Question 11: Use the provided code to simulate the experiment with and without replacement. Scale the number of simulations up to a large number and observe the behavior the two sampling methods. Which sampling method gives better predictions of the population size?