Lab 3: Part 1

Melissa Pulley;

2024-02-07

You will need to submit the CSV file, and your code. Please answer the questions by annotating your answers in the code (using the pound # symbol)

These boxes will inform you of things you need to submit or questions you need to answer!

These boxes will be used to respond to questions. :)

1 Introduction

The past lectures might have been pretty complicated for some of you. That's OK! Hopefully, this lab will help you.

Before we start, lets looks at some useful definitions

Let's start working on defining probability distributions.

A **distribution** describes the spread of the data which shows all possible values or intervals of the data and how they occur. A population has a distribution, and so does a sample. There are also theoretical distributions. They are also called **probability distributions**. Remember that we talked about random processes, random events and random variables? They all talk about situations (or variables) in which the outcome can not be determined in advance. We can't solve for them, because it can take various values. For example, let's roll a dice.

To roll a 6 side dice, all sides with the same probability of appearing, run the following code:

```
sample(1:6,1)
```

Run it multiple times. Each time you a rolling a dice, and getting a different value! You don't know what value you're going to get. But you do know that each time, you have a 1/6 probability of obtaining each particular number. There is a probability distribution that is reigning this.

Run the following code:

```
x<-1:6
P<- rep(1/6,6)
df<-data.frame(x,P)
df
```

P represents P(X=x). And what you just did is a theoretical or probability distribution. It's not based on real data, but based on a function that describes the probability of different possible values of a random variable. In this case 1 to 6.

In case you are curious, when talking about a dice roll, we are working with a multinomial distribution (because there are more than 2 possible results).

2. Defining a discrete probability distribution

During Monday's lecture, we looked at defining a discrete probability distribution. Remember? slide 15 has the steps to define a discrete probability distribution. The four steps were:

1. Define what is the random variable X (uppercase!)
2. List all possible random variable values x (lowercase!). This is the sample space
3. List all probability of occurrence for each x.

So, let's work through this in R. This is the same example used in slide 15! Let's call this scenario 1.

First, let's define the random variable X. In this case, X is the number of heads out of 3 coin tosses (s)

Second, list all possible random variables x. This is the sample space. If we toss a coin 3 times, we can get zero, one, two, or three heads, so, x should be a vector with those 4 values. So, let's write some code that lists all these four potential values. Let's also define n, which is the number of tosses (3). Finally, let's also define p. This is a fair coin, so the probability is 0.

```
x<-0:3
n<-3
p<-0.5
```

You will need to fill the right side of the code. Think about how to do this. As we move forward I will be providing less of the code.

Third, we will list the probability of occurrence for each x.

For this, we use the following:

```
pmf1<-dbinom(x,n,p)
```

Remember, PMF stands for probability mass function: a probability measure that gives us probabilities of the possible values for a random variable

Let's make a dataframe!

```
df1<-data.frame(x,p=pmf1)
```

Let's look at our probability distribution!

```
print(df1)
```

This should be identical to the table we saw in class (slide 15).

Let's obtain the expected value!

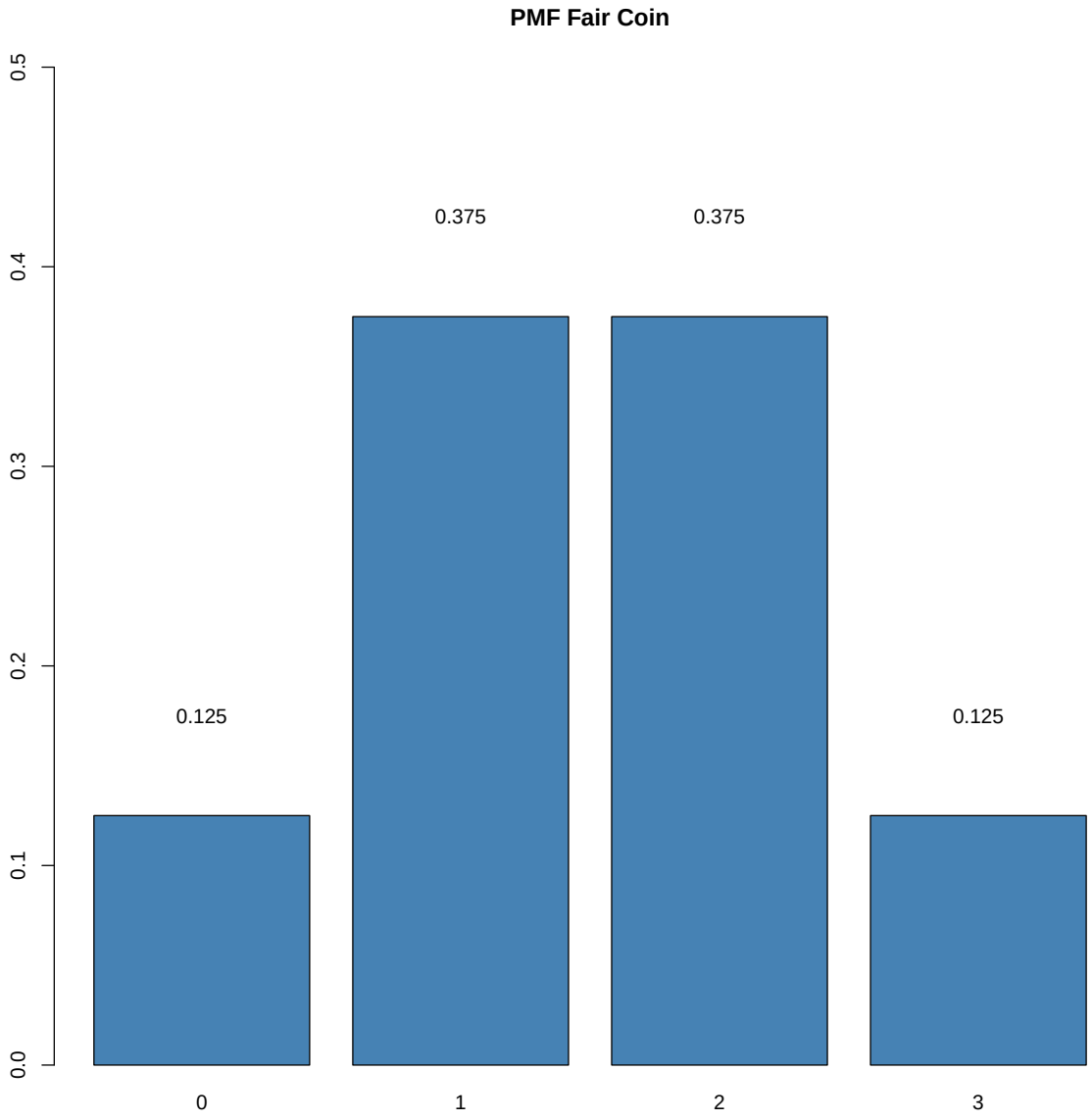
```
EV<-sum(pmf1*x)
EV
```

[1] 1.5

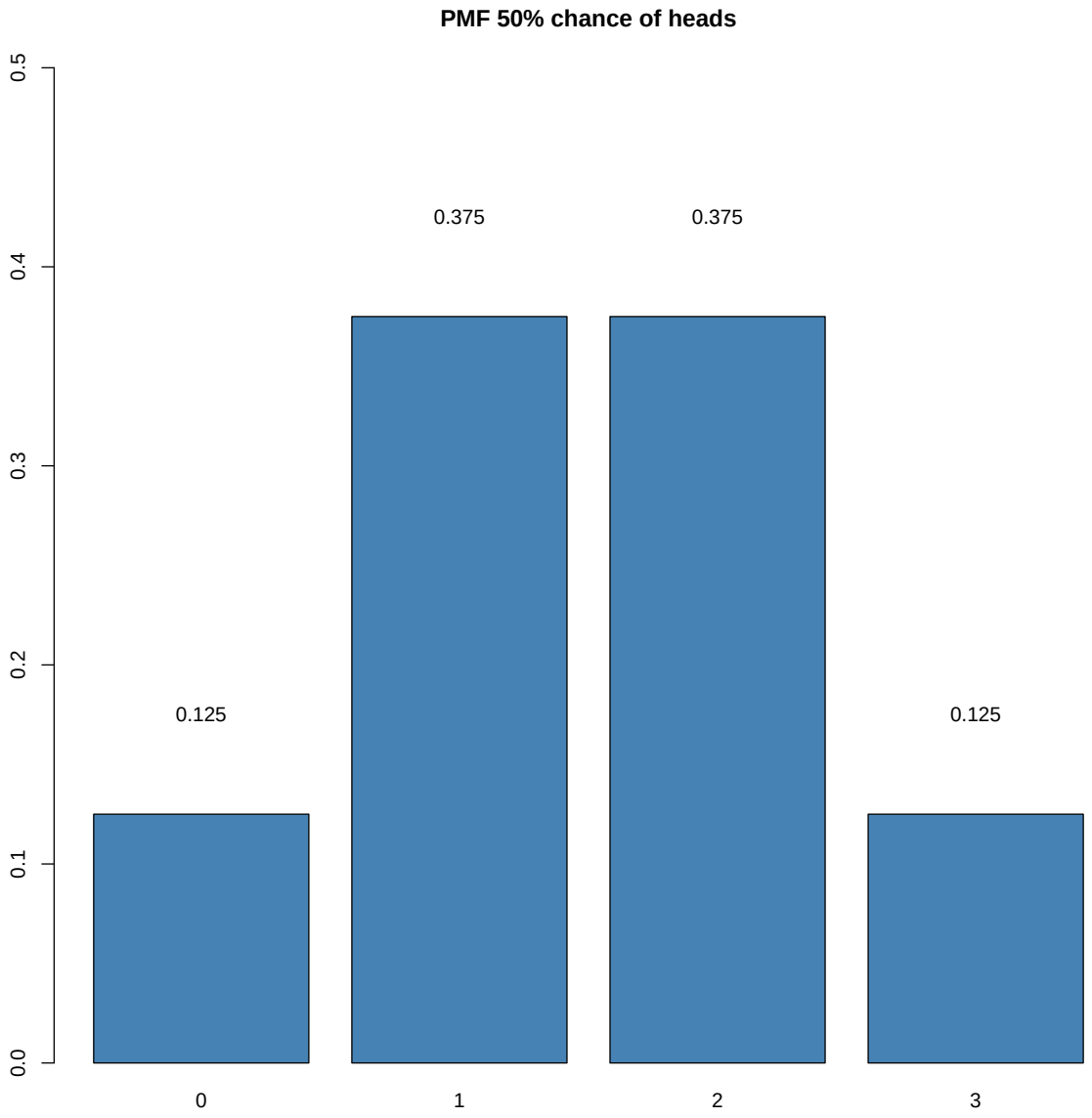
The expected value is the sum of the multiplication of each x by their corresponding probability of occurrence,

Finally, let's plot this.

```
newplot<-barplot(pmf1, col = "steelblue",names=0:3, ylim=c(0,0.5),main="PMF Fair Coin")
text(x=newplot, y=pmf1+0.05,labels = pmf1)
```



If you did everything correct, you should get something that looks like:



If you got something different, trace back and see where you might have stumbled.

Some information to dive a little deeper in the binomial probability distribution

Feel free to skip the information between the breaklines. However, there is an opportunity for an extra credit. I recommend you don't attempt the extra credit unless you're done with the rest of the lab. It might be a better way to manage your time.

Remember that the probability distribution of a binomial random variable is defined as (equation 1):

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

and (equation 2):

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

which is giving us the combination of ways that a certain event might happen. So, in the scenario in which we are tossing 3 coins, the ways we can obtain 2 heads are: HHT, HTH, THH. So, 3. Therefore, if we solved the combinations equation we would get:

$$\binom{3}{2} = 3$$

This is called a combination. As you can tell, the order didn't matter. HHT, HTH, and THH were all counted as "two successes" independently of the order of the elements.

When we care about the order, then it's called a permutation.

EXTRA CREDIT (1 pt) Obtain the probability distribution for scenario 1. But instead of using the dbinom() function, use equation 1 and write the equation (in r) for each of the four possible outcomes. Again, I recommend you come back to this question at the end of the lab

Consider the following code:

```
x<-0:3
n<-3
p<-0.5
PX = rep(NA, n+1)
# 3 choose 0 is 1
for (i in 0:3){
PX[i+1] = choose(n,i)*p^{i}*(1-p)^{n-i}
}
PX
```

[1] 0.125 0.375 0.375 0.125

The probability $P(X = x)$ is $\{0.125, 0.375, 0.375, 0.125\}$ for each x in between 0 and 3.

Now, imagine the following scenario (scenario 2):

1- 5 coin tosses

2- probability of a head is 75%

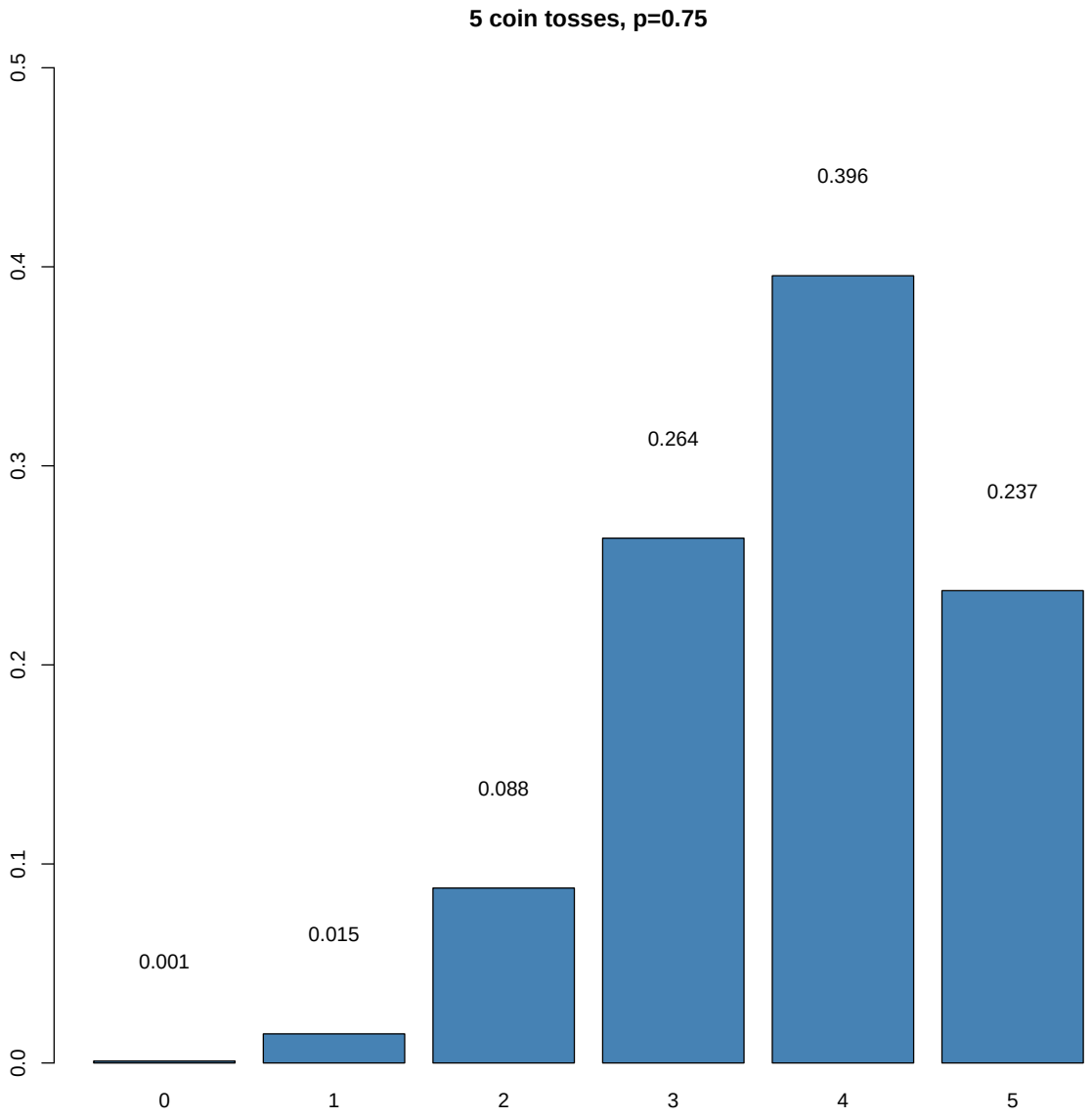
Q1. 2 pts. Write code to obtain the PMF, expected value and plot it (barplot) the same way we did it with the 3 coin tosses example. No need to upload the plot, but include the code. Look at the PMF, does it make sense? Why or why not?

The PMF below makes sense. Since we have a higher probability of heads, it makes sense that we should have a a higher probability of drawing a larger number of heads for a given number of coin tosses.

With the peak of the bargraph at 4, the single most likely outcome is drawing 4 out of 5 heads, which is landing on heads 80% of the time, which is the closest possible option to our set probability of 75% heads, given this number of coin tosses.

```
x<-0:5
n<-5
p<-0.75

pmf2<-dbinom(x,n,p)
newplot<-barplot(pmf2, col = "steelblue",names=0:5, ylim=c(0,0.5),main="5 coin tosses, p=0.75")
text(x=newplot, y=pmf2+0.05,labels = round(pmf2,3))
```



3. Defining a discrete probability distribution with your own example

As part of your homework, you were tasked with coming up with a biological example (check Canvas announcement for the parameters of the example). You should have a value for n, a value for p, and x should be 0:n. While you are free to make up these numbers, hopefully they make some biological sense (maybe something you know or something you read, maybe you even found the values in a paper)

We are going to work with the example.

First off, let's define the define the discrete probability function for your example. Follow the instructions that we used for scenario and scenario 2.

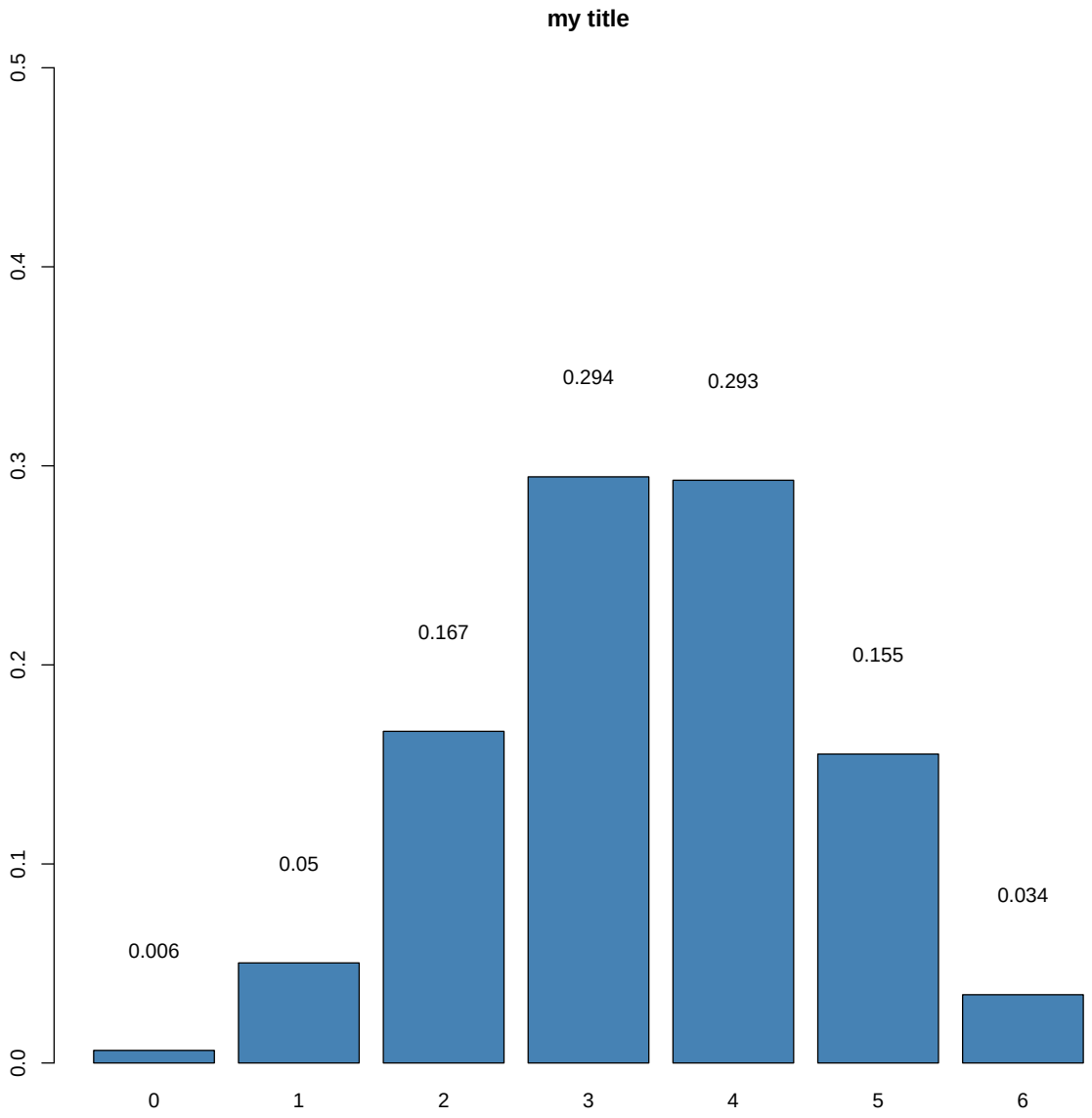
Q2. 3 pts. Describe your biological scenario. Write code to obtain the PMF of your scenario, expected value, and plot it with a barplot. Depending on the size of your plot, displaying the values over the barplot might not be possible or look good. You need to come up with a barplot that you think represents the PMF of your scenario the best. After looking at the barplot, describe the biological importance of that theoretical distribution. Essentially, imagine someone looked at that plot and asked: "but what does it mean?", and they don't have much statistical knowledge. How would you explain that to them?

In my scenario, I am considering whether a human single-child birth results in a male or female child in a family. I will consider that the probability of success, that is, having a female child is 0.57, and that each family has 6 children (single birth only - no twins, etc.).

The bar plot below describes the distribution of families that x number of female children ranging from 0 to 6. So for example, there is approximately a 16.7% chance that a family with 6 children in this population will have 2 female children, and the other 4 are male.

```
x<-0:6
n<-6
p<-0.57

my_pmf<-dbinom(x,n,p)
newplot<-barplot(my_pmf, col = "steelblue",names=0:6, ylim=c(0,0.5),main="my title")
text(x=newplot, y=my_pmf+0.05,labels = round(my_pmf,3))
```



A good way to know whether your PMF might be ok is by running:

```
sum(dbinom(x,n,p))
```

[1] 1

If it equals 1, then we are good!

Now, that PMF shows us the probability of having x successes. Where x = 0, x= 1, x = 2 . . . x =n.

Now, let's simulate a single study.

This is based on your scenario.

```
sims1 = rbinom(n,1,p)
```

What you should be getting is a vector of 1's and 0's. Where 1 is a success and 0 is a failure (I prefer to call that a non-success). Run that code multiple times, and see how the results are different every time (again, a

random process!)

Let's save that simulation in an object called sim1.

To obtain the total number of successes of your simulated data, you can run:

```
sum(sims1)
```

[1] 4

4. Reviewing Bernoulli PMF and Bernoulli Distributions

Once again, I will be defining some important concepts. Hopefully they are clear by now. If by the end of the lab you don't understand them, come talk to me please, so that you won't fall behind.

PMF (again): The PMF or probability mass function tells us the probability of getting a specific value of a discrete random variable. It is written as P(X=x), which reads, the probability of a discrete random variable named 'X', getting the value 'x'. The possible values x would depend on the distribution type: Bernoulli would be different from Binomial, and both would be different from Poisson and Negative Binomial. For example, x can only take on possible values of 0 or 1 for a Bernoulli distributed X. By contrast, for Binomial, x can take on possible integer values from 0,1,2...N where N is the # of trials.

Bernoulli distribution

A random variable X generated from a Bernoulli distribution with probability of success p can be expressed as:

$$X \sim \mathsf{Bernoulli}(x)$$

The PMF of a Bernoulli random variable X is:

$$P(X = x) = p^x(1-p)^{1-x}$$

where x can be 0 (failure) or 1 (success). If you substitute x with 1, then:

$$P(X = 1) = p$$
$$P(X = 0) = 1 - p$$

Estimating Maximum Likelihood

I hope you read about MLE (Maximum Likelihood estimates). The "Mark: A gentle introduction" book is a great resource! For this simulated example, the MLE of p is simply the sum of X (or sum of sims1) divided by the total number of observations of X.

For example, $X \sim \mathsf{Bernoulli}(x)$ and 10 observations of X generated from a probability distribution are: 0 1 1 0 1 0 1 0 1 1

The maximum likelihood estimate (mle) of p (let's call this $\hat{p}$) is $\frac{\sum(X_i)}{n}$, so, in my example:

$$\hat{p} = \frac{0+1+1+0+1+0+1+0+1+1}{10} = 0.6$$

Does this make any sense? Is it intuitive?

Q3. 2 pts. Based on your research on maximum likelihood, does it make sense that the mle is 0.6? Explain why you think mle in this case matches with the presented equation

Maximum likelihood is a tool to estimate parameters. If we are estimating the probability of success, it makes sense that since our data contains 6 successes in 10 trials, our MLE is 0.6 $=\frac{6}{10}$

Q4. 2 pts. Based on your simulated data, estimate the mle for your scenario

```
p = sum(sims1)/length(sims1)
```

'r toString(p)'

For my simulation, I received the data 0 1 1 0 1 1. The MLE is found by taking the sum and dividing by the number of simulations. The MLE for this data is

$$p = \frac{2}{3}$$

5 Poisson distribution

Download the insects csv file and save it in a new object called dat.

```
dat=read.csv("insects.csv")
```

This is data on counts of the number of distinct species found in conifer trees in Nicaragua. Each observation is one tree that was sampled for one hour, and the count is the number of species.

Look at the data. Do you think this is Poisson, or negative binomial (where there is oversidpersion)?

Q5. 1 pts. Estimate the mean, variance and IQR for this count data

```
count_mean = mean(dat$count)
count_mean
```

[1] 3.62

```
count_var = var(dat$count)
count_var
```

[1] 3.732929

```
Q=quantile(dat$count)
IQR = as.numeric(Q[4]-Q[2])
IQR
```

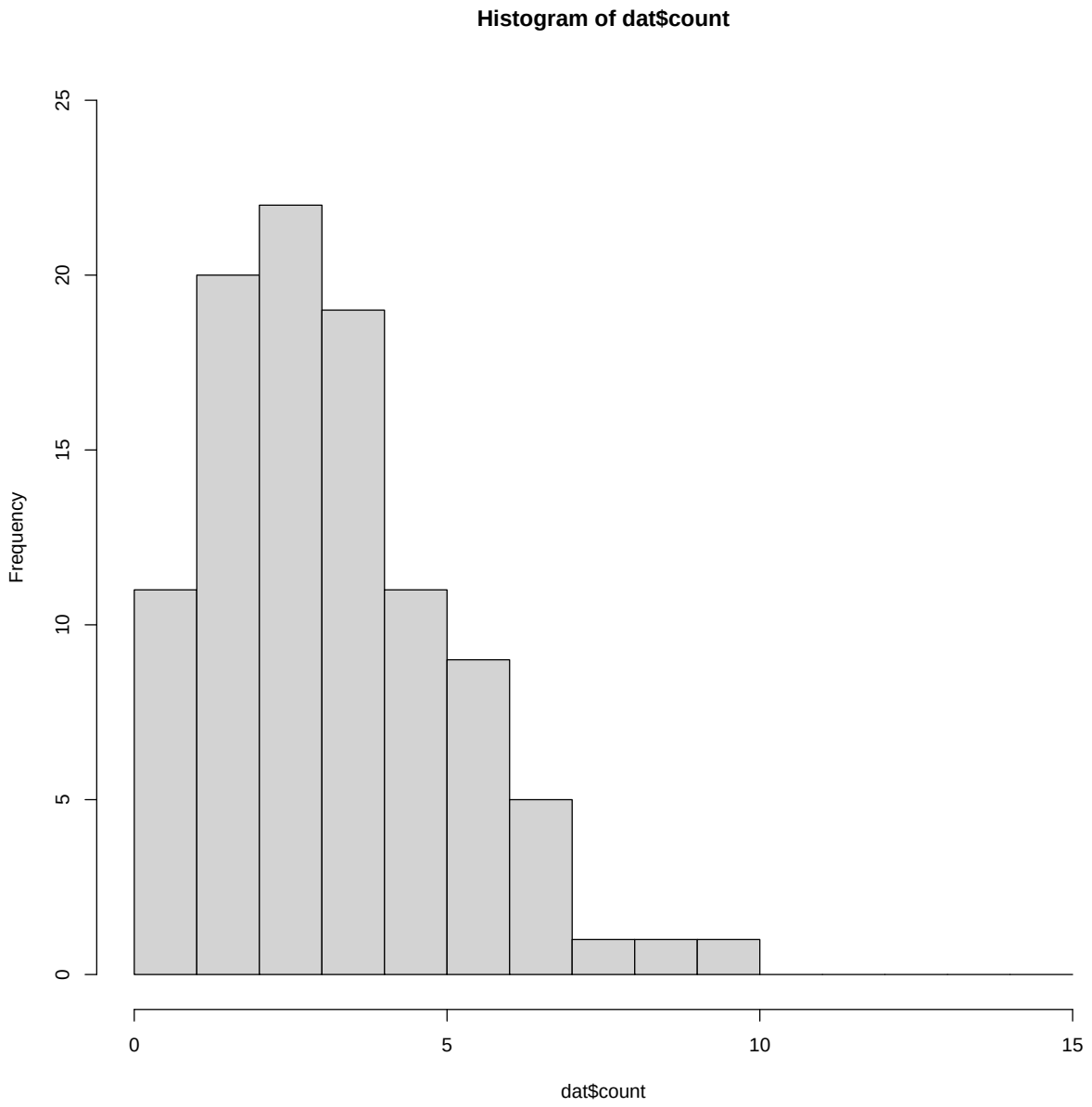
[1] 3

The mean is 3.62, the variance is 3.733, and the Inner quartile range is 3.

Based on the data (overdispersion?) Think whether it is Poisson, or Negative binomial.

We can explore the distribution of number of species by plotting a histogram:

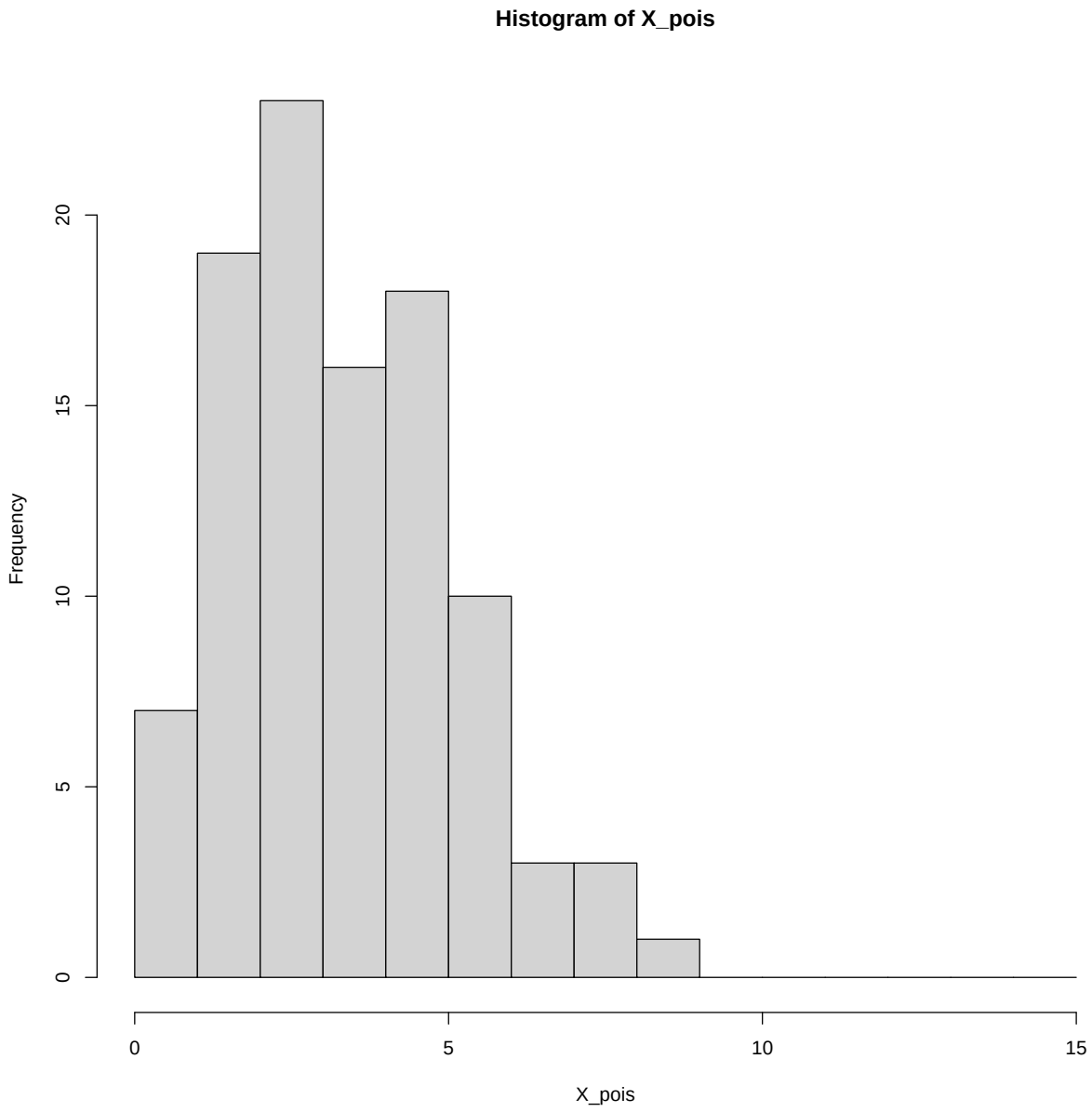
```
hist(dat$count,breaks = seq(0,max(dat$count)+5),xlim=c(0,max(dat$count)+5),ylim=c(0,25))
```



What if we generate a random sample of quantities generated from a Poisson distribution with lambda=mean(dat$count)? If the observed species abundance is drawn from a Poisson distribution with lambda equals to its mean, the histogram of random Poisson quantities should look similar to the histogram of observed species abundance.

```
X_pois <- rpois(n=length(dat$count), lambda=mean(dat$count))

hist(X_pois,breaks = seq(0,max(dat$count)+5),xlim=c(0,max(dat$count)+5))
```

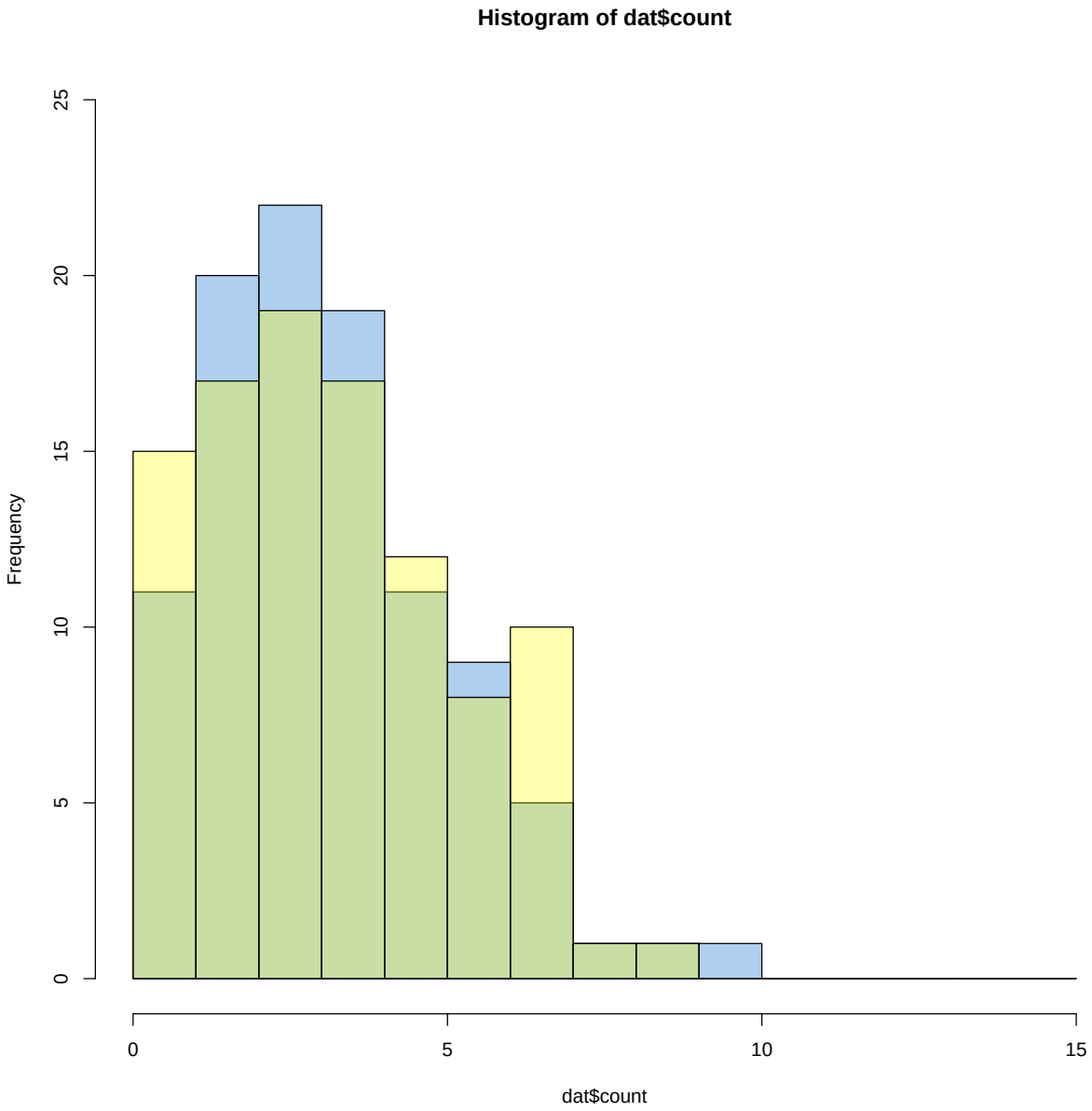


And if we want to plot the two together:

```
c1 <- rgb(0,102,204,max = 255, alpha = 80, names = "lt.blue")
c2 <- rgb(255,255,0, max = 255, alpha = 80, names = "lt.yellow")

hist(dat$count,breaks = seq(0,max(dat$count)+5),xlim=c(0,max(dat$count)+5),col=c1,ylim=c(0,25))


X_pois <- rpois(n=length(dat$count), lambda=mean(dat$count))
hist(X_pois,breaks = seq(0,max(dat$count)+5),xlim=c(0,max(dat$count)+5),ylim=c(0,25),col=c2,add=T)
```



It is good practice to do it multiple times, each time running the rpois code line.

Q6. 2 pts. Save an overlapping histogram and upload it to Canvas. Explain why running the code multiple times is good practice, and whether you think the Poisson distribution fits based on the overlapping histograms

Running the code multiple times is good practice because there is some variation in the data each time. Because the code using to generate the poison data is random, generating the code multiple times will allow us to get a better sense of how the data distribution is shaped.

Referring to Figure 1, while the histograms do not match up exactly, they do have peaks at the same approximate location. The actual data in below, has a slightly smaller peak value and is spread slightly more. Overall, the poisson distribution still appears to make a decent visual approximation for the data, but may need more data and more rigorous computations to make an accurate conclusion.

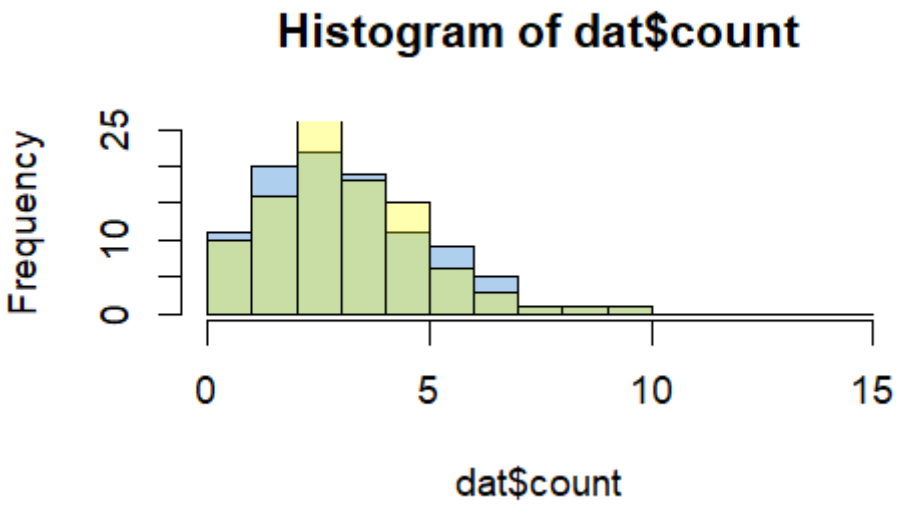


Figure 1: Overlapping histogram for Q6

Resources

These are hyperlinks

The base R cheat-sheet

Dr. Fordyce Poisson

An introduction to R

Mark: a Gentel Introduction

End of document