Week 9: Discrete Probability Distributions

Contents

In the lecture this week, we covered discrete probability distributions. In this live R, we're going to learn about functions that allow us to explore these concepts further. We'll start by loading the tidyverse:

```
library(tidyverse)
```

As we review this material, let's work with our example from the previous lecture. Imagine you perform an experiment where someone has to select the hand in which a coin is hidden 5 times. Let our random variable, X, be the number of correct guesses. This means that X can take on any value between 0 and 5.

0.1 Describing our Random Variable

We can compute some descriptive data about our random variable, X. One thing that we can do is to calculate the **expected value E(X) and standard deviation SD(X)**. These values give us an idea of the long-term mean and standard deviation of our data. In other words, if we were to do this experiment over and over, we would expect to choose the correct hand an average of $E(X) \pm SD(X)$ times.

To calculate the long-term mean, we use the formula $E(X) = n * p$. To calculate the standard deviation, we use the formula $SD(X) = \sqrt{n * p * (1 - p)}$

Keep in mind that in the case of these equations (and the probability mass function), the parameters n and p mean something different than other times you may encounter them in this course. In this case, $n =$ *number of trials* and $p =$ *probability of success.*

Let's compute $E(X)$ and $SD(X)$ for our random variable.

```
n <- 5
p <- 0.5

n*p
```

```
## [1] 2.5
```

```
sqrt(n*p*(1-p))
```

```
## [1] 1.118034
```

0.2 Probability Density Function

Now, let's calculate the probability of selecting the correct hand exactly 3 out of 5 times. To do this, we will use the formula for the probability mass function. Because we have two possible outcomes (correct/incorrect), we will use the probability mass function associated with binomial data:

$Pr(X = k) = \binom{n}{k} p^k q^{n-k}$

where: $k = number\ of\ successes$

$n = number\ of\ trials$

$p = P(success)$

$q = P(failure)$

We could apply the PMF manually by plugging our values into the equation and using the *factorial()* function. Note that I've broken this down into the separate steps we've discussed in the lecture, but you could do this all in one go if you prefer.

```r
k <- 3
q <- 1-p

step1 <- factorial(n)/(factorial(k)*factorial(n-k))
step2 <- p^k*q^(n-k)

step1*step2
```

```
## [1] 0.3125
```

Or we could save time and effort by using a function to do the work for us. *dbinom()* gives us the value of the probability mass function for a binomial distribution, given k successes, n trials and a probability of p:

```r
dbinom(k, n, p)
```

```
## [1] 0.3125
```

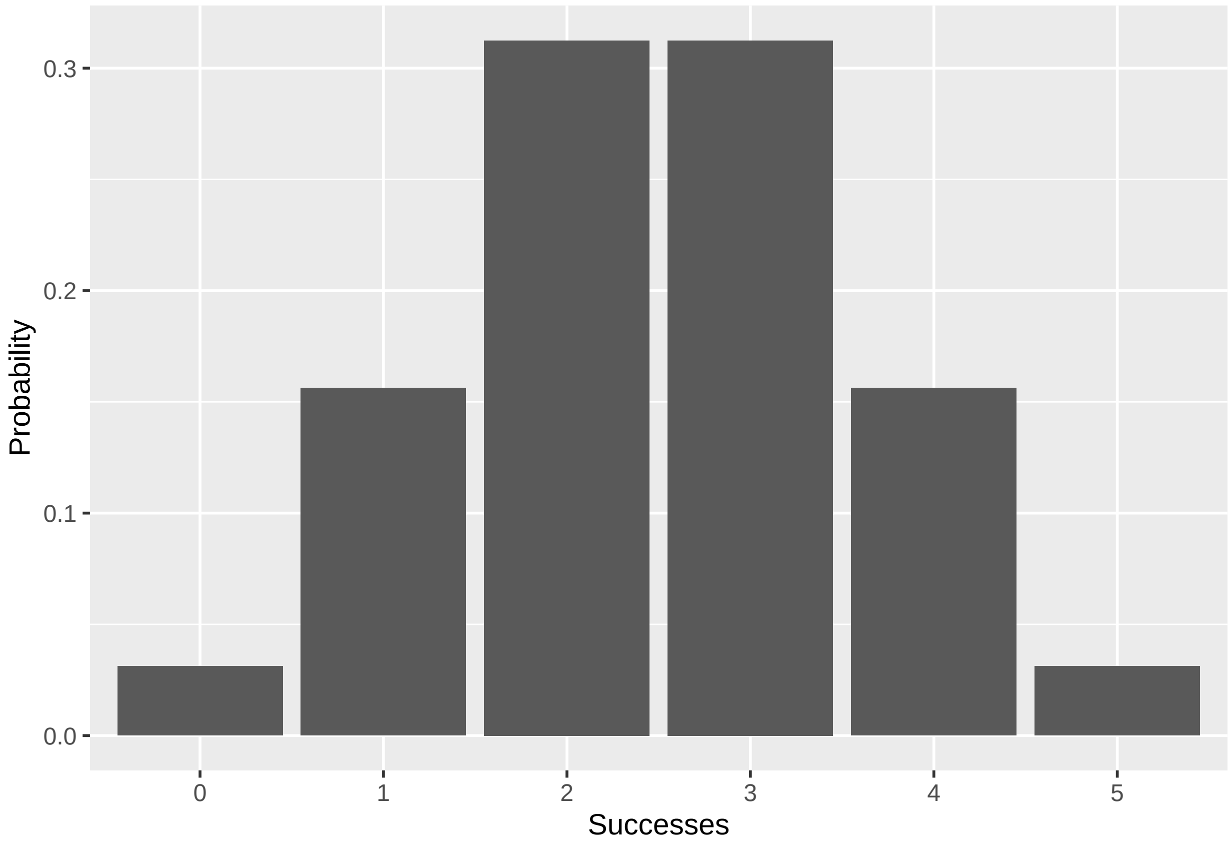
Conveniently, k doesn't have to be a single value. You can compute the PMF for a range of k. Let's say we wanted to compute the values for each possible outcome of our experiment (0 correct guesses:5 correct guesses). We could do that by passing k as a vector that contains the values 0-5:

```r
dbinom(0:5, n, p)
```

```
## [1] 0.03125 0.15625 0.31250 0.31250 0.15625 0.03125
```

We can generate a probability distribution plot by feeding these values into a bar plot.

```r
dat <- tibble(Successes = as.factor(0:5), Probability = dbinom(0:5, n, p))

ggplot(dat, aes(Successes, Probability)) + geom_bar(stat = 'identity')
```



0.3 Cumulative Distribution Function

The PMF allows us to compute the probability that the correct choice is made *exactly* 3 times out of 5. But perhaps we want to calculate the probability that the correct choice is made 3 times or less. In this case, we can use the cumulative distribution function:

$P(X \leq 3) = \sum_{i=0}^{x} \binom{n}{i} p^i (1-p)^{n-i}$

This looks complex, but really, it's just summing the outcome of the PMF across all values up to the value of interest (which, in our case, is 3).

Let's skip doing it the hard way in the interest of time, and instead go straight to the easy way, with the *pbinom()* function. This function applies the cdf of the binomial distribution to k. As with *dbinom()*, this function can also be applied to a vector of values. It can take, as arguments, the same values we previously gave dbinom:

```
pbinom(k, n, p)
```

```
## [1] 0.8125
```

```
pbinom(0:5, n, p)
```

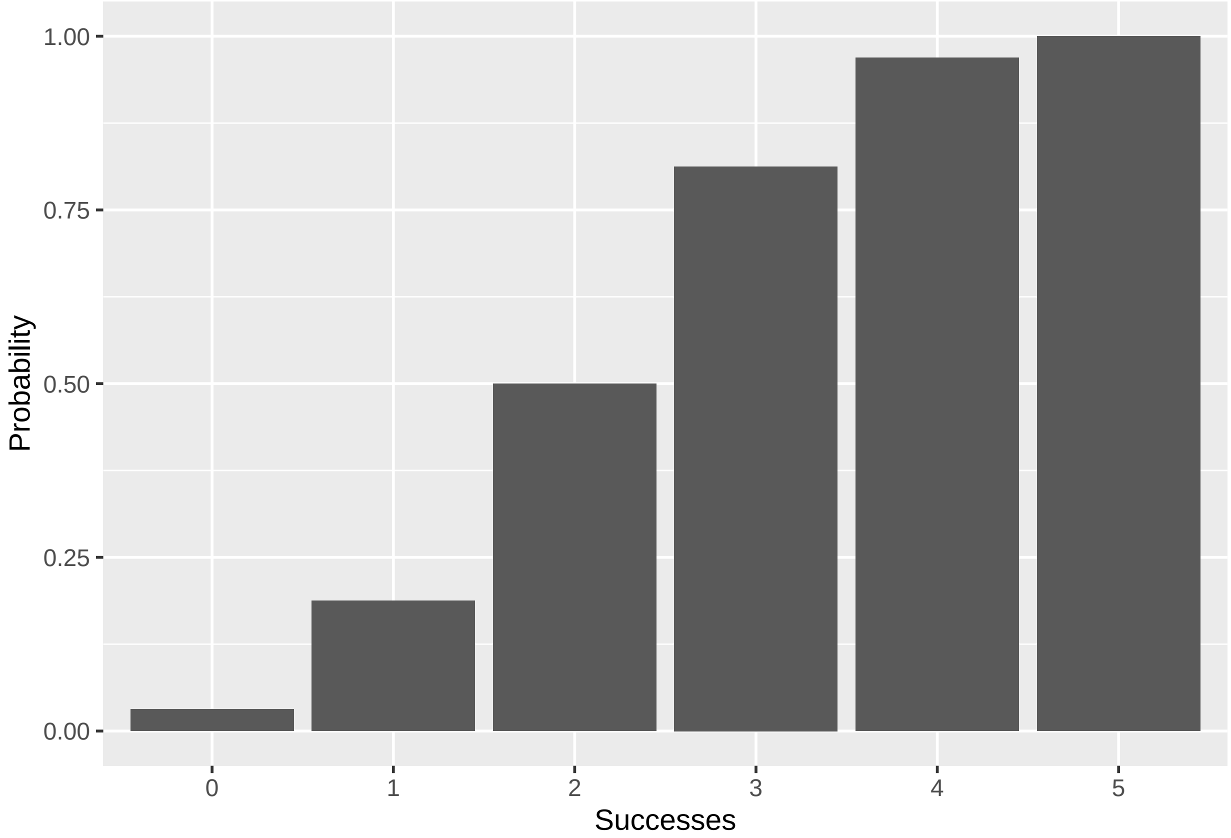
```
## [1] 0.03125 0.18750 0.50000 0.81250 0.96875 1.00000
```

Like before, we can also produce a plot from our results:

```
pDat <- tibble(Successes = as.factor(0:5), Probability = pbinom(0:5, n, p))
pDat
```

```
## # A tibble: 6 x 2
##   Successes Probability
##   <fct>           <dbl>
## 1 0              0.0312
## 2 1              0.187
## 3 2              0.5
## 4 3              0.812
## 5 4              0.969
## 6 5              1
```

```
ggplot(pDat, aes(x = Successes, y = Probability)) + geom_bar(stat = 'identity')
```



0.4 Example experiment

Now let's say we conducted the experiment with 100 different participants and collected the number of correct responses from each. Here, we'll create some data and calculate frequencies. Then, we'll use *dbinom()* to calculate the expected binomial probability for each outcome:

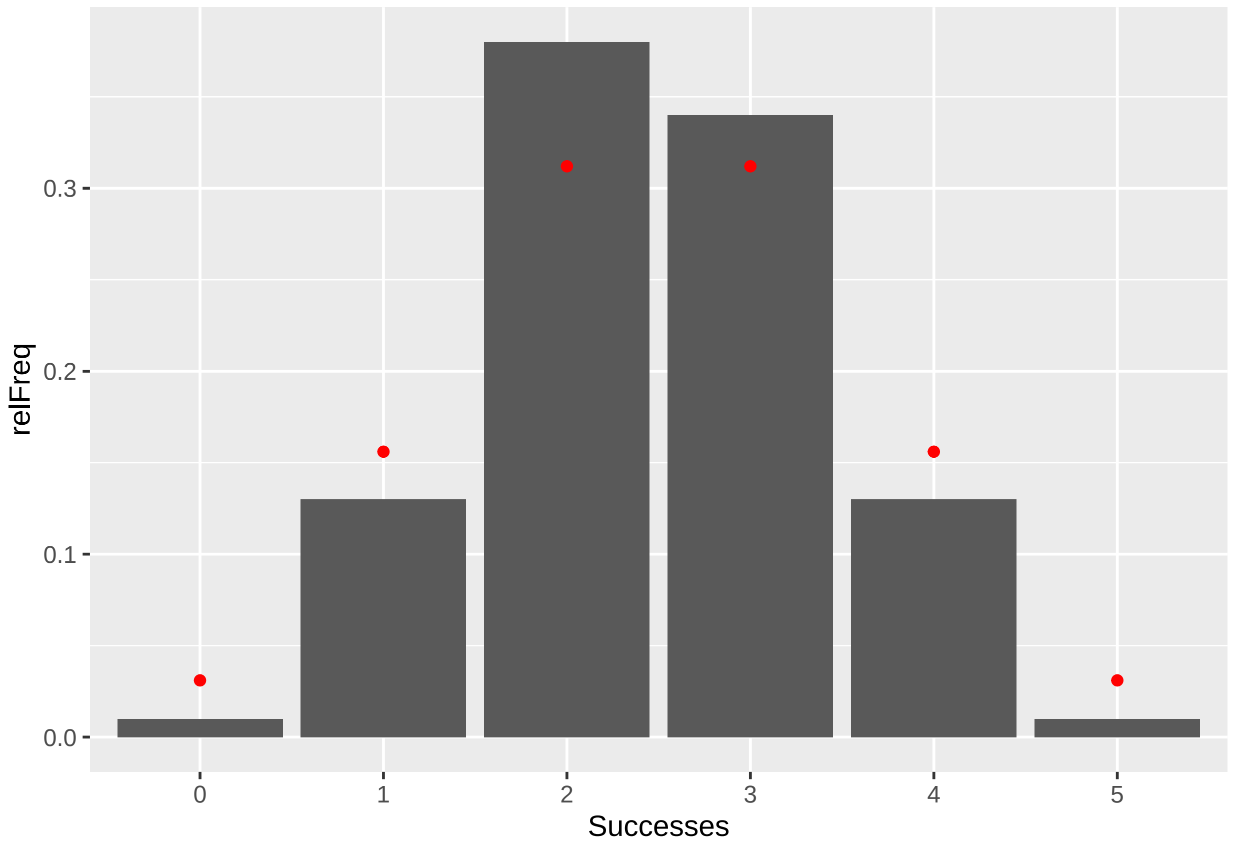
```
set.seed(608)
handDat <- tibble(Successes = as.factor(round(rnorm(100, mean = 2.5, sd = .90), 0)))
```

```r
freqDistr <- handDat %>%
  count(Successes) %>%
  mutate(relFreq = n / sum(n),
         expProb = round(dbinom(0:5, 5, 0.5), 3)) %>%
    rename(rawFreq = n)

freqDistr
```

```
## # A tibble: 6 x 4
##   Successes rawFreq relFreq expProb
##   <fct>       <int>   <dbl>   <dbl>
## 1 0               1    0.01   0.031
## 2 1              13    0.13   0.156
## 3 2              38    0.38   0.312
## 4 3              34    0.34   0.312
## 5 4              13    0.13   0.156
## 6 5               1    0.01   0.031
```

Now, we'll plot a frequency distribution of our sample's data, and see how well it matches up with the expectations, given that the data fits the binomial distribution (the red points).

```r
p1 <- ggplot(freqDistr, aes(x = Successes, y = relFreq)) + geom_bar(stat = 'identity') +
  geom_point(aes(x = Successes, y = expProb), colour = 'red')

p1
```



In this case, it seems that the data fit the binomial distribution relatively well - the probabilities generally align. Participants are most likely to select the correct hand 2 or 3 times, less likely to select the correct hand 1 or 4 times, and unlikely to never select the correct hand, or to select the correct hand all 5 times.

0.5 Example Write-up

In this analysis, we asked 100 participants to select the hand in which a coin was hidden. Participants completed 5 trials each. We measured the total number of trials in which they were able to successfully identify the hand with the coin. See Figure 1 for a visualisation of the frequency of each outcome, with a Binomial fit superimposed as red dots. Participants were most likely to choose the correct hand 2 out of 5 times. Participants were more likely to select the correct hand 2 (0.38) or 3 (0.34) times. Participants were highly unlikely to never select the correct hand (0.01), or select the correct hand all 5 times (0.01).

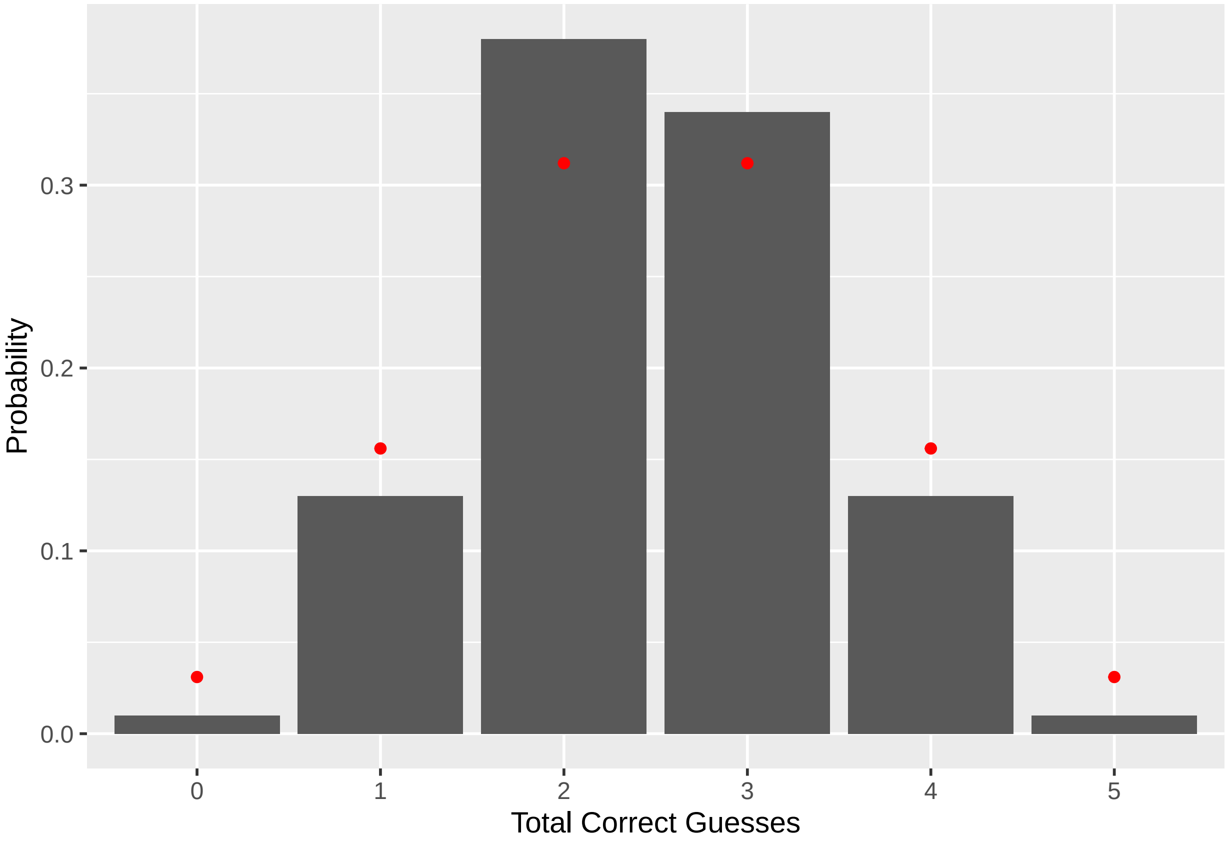


Figure 1: Outcome Frequency