Binomial Probability

Gaston Sanchez

Learning Objectives

- Function `factorial()`
- Function `choose()`
- Getting to know the function `dbinom()`

Introduction

In this tutorial we cover computational aspects of the topics described in Chapter 15 *The Binomial Formula* of the text book. In particular, this formula allows you to answer questions like:

- A coin is tossed three times. What is the chance of getting exactly one head?
- A die is rolled ten times. What is the chance of getting exactly three aces?
- A box contains one red marble and nine green ones. Five draws are made at random with replacement. What is the chance that exactly two draws will be red?

These problems are similar, and can be solved using what is called the **binomial probability** formula. Before describing the formula, you should become familiar with the conditions that make an experiment or process to fall within a *binomial setting*.

Binomial Experiment

For an experiment or process to be considered under the binomial setting, it needs to fulfill the following conditions:

- There is a fixed number of trials. We denote this number by n.
- The n trials are **independent** and repeated under identical conditions.
- Each trial has only two outcomes, which for convenience we call *success* and *failure*.
- For each individual tria, the probability of a success is the same. We denote the probability of success by p.
- The central problem of a binomial experiment is to find the probability of k successes out of n trials.

A classic binomial example is tossing a coin $n = 5$ times and counting the number of heads. Why is this experiment binomial? Let's go over the checklist.

- There is a fixed number $n = 5$ of tosses.
- The outcome of each toss falls into one of two categories: "heads" or "tails".

- The probability of "heads" in each toss is the same. Assuming it is a fair coin, then the probability of "heads" is 0.50.
- The tosses are independent. This means that the outcome in a given toss, does not affect the outcome in the subsequent toss.

The main consideration in an experiment or process that fits the binomial setting has to do with counting the number of successes (or failures) in n trials or attempts. It turns out that the probability of the number of successees (or failures) follows what is called the **binomial distribution**.

A non-binomial experiment

There are experiments that seem to be binomial but that they are actually not. For example, suppose you flip a coin until you get three tails, and you count how many flips it takes to get there. Is this situation binomial? The possible outcomes are "heads" and "tails". The probability of "tails" is the same in each flip. And the flips are independent. However, the number of flips is not fixed. So this is not a binomial experiment.

Example

Let's go back to one of the motivating questions. What is the chance of getting exactly one head when a coin is flipped three times?

To answer this question, we need to find the number of ways in which one head occurs when a coin is flipped three times. The possible outcomes when flipping a coin three times are:

- heads, tails, tails (HTT)
- tails, heads, tails (THT)
- tails, tails, heads (TTH)
- heads, heads, tails (HHT)
- heads, tails, heads (HTH)
- tails, heads, heads (THH)
- heads, heads, heads (HHH)
- tails, tails, tails (TTT)

Notice that only three of the listed outcomes have excatly one head: HTT, THT, and TTH. Thus, the probability of getting exactly one head is 3/8.

Now, what if we are interested in finding the probability on getting exactly two heads? Looking at the list of outcomes, there are three ways to get 2 heads: HHT, HTH, and THH. Consequently, the probability of two heads is 3/8.

Likewise, the probability of getting exactly 3 heads in 3 flips, is 1/8 because there is only one outcome HHH. This is also the same probability of getting no heads, which corresponds to the outcome TTT.

In this example, "getting heads" is considered a *success*. And assuming that the coin is a fair coin, the probability of heads is 1/2. The number of successes in three flips ranges from 0 (no heads), to 3 (all heads). More precisely, the distribution of the number of heads has the following frequencies, with their corresponding probabilities:

Heads	Ways it can happen	Probability
0	1	1/8
1	3	3/8
2	3	3/8
3	1	1/8

The number of ways in which we can get $k = 0, 1, 2, 3$ heads out of $n = 3$ flips follows a specific pattern. In general, the number of ways in which k successes can happen in n trials, is given by the number of **combinations** in which k objects can be arranged from a collection of n. And the resulting values are known as **binomial coefficients**.

The general formula to find the number of combinations of k objects selected from a collection of n is given by the *binomial coefficient* formula:

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

Another common notation for the binomial coefficient is:

$${}_nC_k = \binom{n}{k}$$

Factorials

To study the binomial formula, we must talk about the **factorial** of a non-negative integer. The factorial of a non-negative integer n, denoted $n!$, is the product of all positive integers less than or equal to n. For example, the factorial of 4 is $4! = 4 \times 3 \times \times 2 \times 1 = 24$.

Here are some examples of factorials for the first four positive integers:

- $1! = 1$
- $2! = 2 \times 1 = 1$
- $3! = 3 \times 2 \times 1 = 6$
- $4! = 4 \times 3 \times \times 2 \times 1 = 24$
- and so on.

By definition $0! = 1$. Also note that there is NO factorial of a negative integer.

You can calculate factorials in R using various approaches. The simplest one is by multiplying all positive integers:

```
# 4!
4 * 3 * 2 * 1
```

```
## [1] 24
```

This way to compute the factorial of a non-negative integer works but it requires a lot of typing. Imagine if you had to type all the products for 10!

```
# 10!
10 * 9 * 8 * 7 * 6 * 5 * 4 * 3 * 2 * 1
```

```
## [1] 3628800
```

or worse: the products for 20!, or a larger integer.

Another option to compute the factorial is to create a sequence vector, and then pass it to the `prod()` function. `prod()` computes the product of all the elements in the provided vector:

```
# another way to calculate 4!
seq4 = c(1, 2, 3, 4)
prod(seq4)
```

```
## [1] 24
```

```
# equivalent to
prod(1:4)
```

```
## [1] 24
```

```
# equivalent to
prod(4:1)
```

```
## [1] 24
```

A third, and better, option is the function `factorial()` which is the dedicated function in R to calculate the factorial of a number:

```
# 4! with factorial()
factorial(4)
```

```
## [1] 24
```

Combinations

Let's go back to the formula of the *binomial coefficients* denoted by ${}_nC_k$ (read "n choose k"), and given by:

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

${}_nC_k$ is the number of ways to choose k things our of n. Put another way, the number of combinations of n things taken k at a time.

R provides the function `choose()` to calculate combinations.

```
n = 4
k = 2

# 4 choose 2
choose(n, k)
```

```
## [1] 6
```

Combinations are called **binomial coefficients**.

Binomial Formula

The formula for the binomial probability is:

$$P(k) = \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k}$$

equivalent to:

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

where:

- n is the number of (fixed) trials
- p is the probability of success on each trial
- $1-p$ is the probability of failure on each trial
- k is a variable that represents the number of successes out of n trials

Example

Let's consider an example. A die is rolled 10 times. What is the percentage of getting exactly 2 aces?

```
n = 10  # 10 rolls
k = 2   # 2 aces
p = 1/6 # probability of ace

choose(n, k) * (p^k) * (1-p)^(n-k)
```

```
## [1] 0.29071
```

Or what about the probability of getting 5 aces?

```
n = 10  # 10 rolls
k = 5   # 5 aces
p = 1/6 # probability of ace

choose(n, k) * (p^k) * (1-p)^(n-k)
```

```
## [1] 0.01302381
```

In general, you can compute the probabilities for all possible number of aces: 0, 1, . . . , 10.

```
n = 10    # 10 rolls
k = 0:10  # number of aces (fromm 0 to 10)
p = 1/6   # probability of ace

choose(n, k) * (p^k) * (1-p)^(n-k)
```

```
##  [1] 1.615056e-01 3.230112e-01 2.907100e-01 1.550454e-01 5.426588e-02
##  [6] 1.302381e-02 2.170635e-03 2.480726e-04 1.860544e-05 8.269086e-07
## [11] 1.653817e-08
```

Function `dbinom()`

It turns out that R provides a dedicated function that allows you to calculate the probability of a binomial process: `dbinom()`.

`dbinom()` takes three main inputs:

- `x`: the number of successes
- `size`: the number of trials
- `prob`: the probability of success

From the previous example, the probability of getting exactly 2 aces when a die is rolled 10 times, can be obtained with `dbinom()` as follows:

```
n = 10   # 10 rolls
k = 2    # 2 aces
p = 1/6  # probability of ace

dbinom(x = k, size = n, prob = p)
```

```
## [1] 0.29071
```

Likewise, you can use `dbinom()` to obtain the probability for all possible number of aces: 0, 1, . . . , 10.

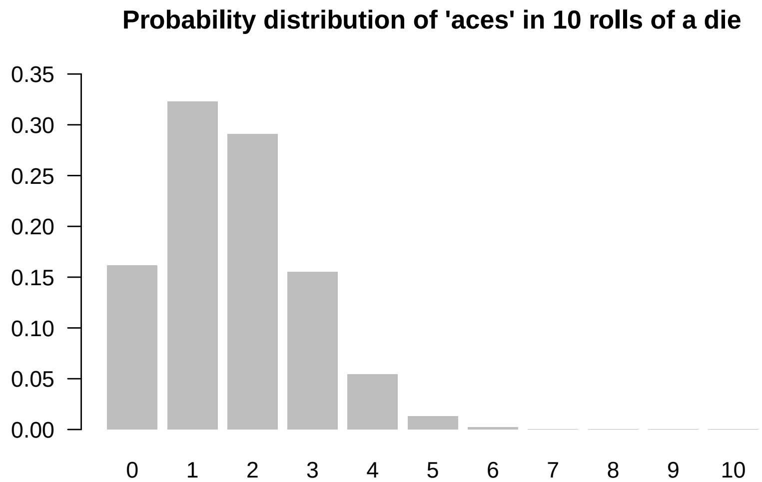
```
n = 10    # 10 rolls
k = 0:10  # number of aces (fromm 0 to 10)
```

```r
p = 1/6   # probability of ace

dbinom(x = k, size = n, prob = p)
```

```
##  [1] 1.615056e-01 3.230112e-01 2.907100e-01 1.550454e-01 5.426588e-02
##  [6] 1.302381e-02 2.170635e-03 2.480726e-04 1.860544e-05 8.269086e-07
## [11] 1.653817e-08
```

You can take the vector of probabilities of aces and plot a *theoretical histogram*. To create the graph in R you can use the `barplot()` function:

```r
prob_aces = dbinom(x = k, size = n, prob = p)

barplot(prob_aces, las = 1, names.arg = 0:10, ylim = c(0, 0.35), border = NA,
        main = "Probability distribution of 'aces' in 10 rolls of a die")
```



One More Example

1. A die is rolled 6 times. What is the chance of obtaining exactly one "five"?

```r
n = 6   # rolls
k = 1   # one 'five'
p = 1/6 # probability of 'five'

choose(n, k) * (p^k) * (1-p)^(n-k)
```

```
## [1] 0.4018776
```

What is the chance of obtaining exactly three "fives"?

```r
n = 6   # rolls
k = 3   # three 'five'
p = 1/6 # probability of 'five'
```

```r
choose(n, k) * (p^k) * (1-p)^(n-k)
```

```
## [1] 0.05358368
```

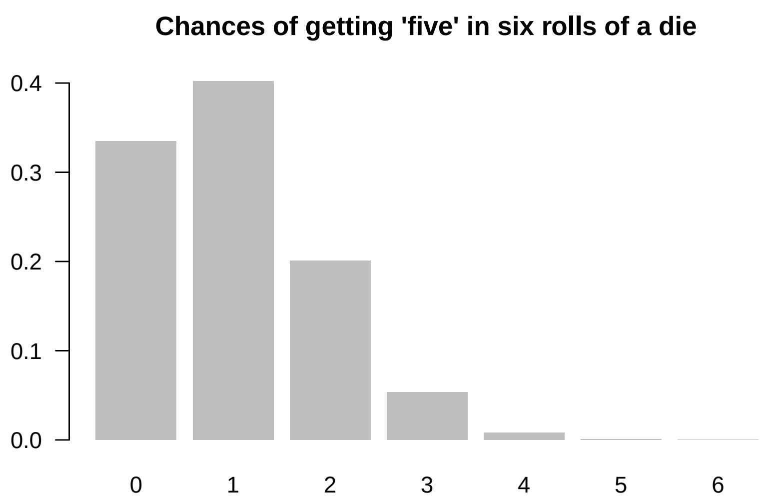
Let's calculate all possible values for the number of "fives", that is, $k = 0, 1, 2, 3, 4, 5, 6$

```r
n = 6    # rolls
k = 0:6  # all possible 'five's
p = 1/6  # probability of 'five'

# probability distribution of 'five's
prob_fives = choose(n, k) * (p^k) * (1-p)^(n-k)

# rounding probabilities to 5 decimal places
round(prob_fives, 5)
```

```
## [1] 0.33490 0.40188 0.20094 0.05358 0.00804 0.00064 0.00002
```

You can take the vector of probabilities and plot a *theoretical histogram*. To create the graph in R you can use the `barplot()` function:

```r
barplot(prob_fives, las = 1, names.arg = 0:6, border = NA,
        main = "Chances of getting 'five' in six rolls of a die")
```



What is the probability of getting at least 3 "fives"?

```r
n = 6    # rolls
k = 3:6  # 3, 4, 5, or 6 'five's
p = 1/6  # probability of 'five'

# probabilities of 3, 4, 5, and 6 'five's
prob_3_or_more = choose(n, k) * (p^k) * (1-p)^(n-k)

# probability of at elast3 'five's
```

```r
sum(prob_3_or_more)
```

```
## [1] 0.06228567
```