# Probability

INFO 640 | Pratt Institute | McSweeney

## Overview

In this lab, we are going to model some distributions and use R to calculate the likelihood of picking a sample from within a certain range.

We'll start with coin flips. Coin flips can be represented by a binomial distribution. That is, there's only 2 outcomes: heads or tails.

R has a direct way to calculate this, with rbinom()

Let's start with just one fair coin flipped one time. Just for fun, let's call 1 heads and 0 tails.

```
rbinom(1, 1, .5)
```

Run that again, did you get the same answer? Run it until you do. This is the building blocks for almost all mathematical models. From election predictions to financial forcasts, weather, disease spreading etc., simulating outcomes (usually binomial) is the most basic unit. Increasing the samples, changing the weights, layering outcomes to make them interdependent. All of these things make the models more complex and representative of the world we think we know. But ultimately, it comes down to this one little unit.

Let's try another. This time, flip ten coins unique one time each

```
rbinom(10, 1, .5)
```

You get a vector of responses. You can obviously sum these together, but can also do things like randomly choose a winner based on position in a line.

```
d1 <- rbinom(10, 1, .5)
sum(d1)
```

What if, instead we wanted to flip one coin 10 times?

```
rbinom(1, 10, .5)
```

We don't get a vector here - we get the result of how many times this one coin came up heads. This is equvalent to summing your vector of 10 coins.

What if, instead, we wanted to flip 10 coins 10 times each. Take a second to think: will you get a vector or one number, and what will it look like?

```
rbinom(10, 10, .5)
```

Great! We got a vector of the outputs. Now let's try changing the probability, let's make heads (1) more likely. In most situations, this will be your Relevant class, or the target category in your data.

```
rbinom(10, 10, .8)
```

Just for fun, change it the other way

```
rbinom(10, 10, .2)
```

Ok, enough of that. Let's visualize the distribution to prove to ourselves that the likelihood of any given outcome is normally distributed.

First assign a variable, flips to a vector representing

```
flips <- rbinom(100000, 10, .5)
hist(flips)
```

Great! It converges to a normal distribution - just like we were expecting!

Now let's see what the liklihood is of getting exactly 5 heads. Remember that flips is a vector of outcomes. We want to know where that vector is equal to exactly 5.

```
flips == 5
```

We got another vector. Now we're going to use a bit of a mathematical trick. In R, TRUE is equivaent to 1, and FALSE is equivalent to 0. If we take the mean of this vector, it essentially returns a probability.

```
mean(flips == 5)
```

A slightly easier way to calculate this is with pbinom(). pbinom() gives us just the probability rather than the vector. Here we have 5 coins with 10 flips each - equally likely.

```
pbinom(5, 10, .5)
```

So the probability of getting exactly 5 heads in 10 flips is .62

That's all we will do for binomial distributions. Let's move on to normal distributions.

Normal distributions are useful because they often describe naturally occurring phenomena.

We'll use the classic example of heights. We know that heights are normally distributed and the mean height of American women in 65 inches.

Let's take a sample of 10,000 American women and model the distribution.

```
rnorm(10000, 65, 3.5)
```

Just like before, we get a vector of the heights of 10,000 women. Let's use it to make a histogram.

```
heights <- rnorm(10000, 65, 3.5)
hist(heights)
```

That looks fine, but you may want to represent the actual normal distribution and find out how likely it is to randomly select an American woman from the population who is over 5 foot 10 (70 inches) (or some other height). This requires slightly more code. We're going to write our first function.

**This function is optional** We'll define a function that takes any random observation and plots it along a normal distribution. Then we will use the integrate() function, which takes the intergral of the function to calculate the area under the normal curve from the point we defined as it approaches infinity. If you didn't understand that, don't worry - it's not essential to the concepts here or to this lab

```
f <- function(x){ dnorm(x, mean=65, sd=3.5) }
integrate(f, 70, Inf)
```

Looks like the liklihood is 7.66% that a randomly selected woman would be over 5'10".

**back to required**

If you haven't taken calculus (ever or in a few years) and don't feel up to writing a function to take the integral every time, R has a function built in - just like for the binomial distribution.

```
pnorm(70, 65, 3.5)
```

Note that pnorm calculates from the left side of the curve - always. So the result you get is always the answer to: What is the probability of selecting an observation less than X from the population? It's obviously easier to subtract this from 1 than to keep writing functions and takeing the integral.

So if you want the answer to "what is the probability of selecting an observation above X from the population", use:

```
1-pnorm(70, 65, 3.5)
```

Congratulations! You now have enough knowledge about probabilities to calculate the outcome of nearly anything – so long as you know the mean and the standard deviation of the population. It's a narrow range of things that you can calculate, but still incredibly powerful.