Week 4 - Probability

Nicholas Kortessis

2025-02-05

Probability Distributions

This week, we are going to talk about probability in R. We can do two fundamental things related to probability in R. One is to use R to do random sampling. The other is to use R to do probability modeling.

Remember, probability is all about formalizing the study of uncertainty. There is the basic uncertainty that comes with not knowing

Random Sampling

First, let's think about random sampling. To do that, we will use a recently published data set about the attributes of tree species published in the journal Ecology (Nigro et al. 2024). More information about the data set can be found here.

We will use this data set to illustrate some principles from probability. To do that, assume that this data represents the ENTIRE POPULATION of relevant trees.

First let's load the data and take a look at it using the function `str()`.

```
setwd('/Users/nicholaskortessis/Library/CloudStorage/GoogleDrive-kortessn@wfu.edu/My Drive/Import/Wake I
tree.df <- read.csv(file = 'species_attributes.csv')

str(tree.df)
```

```
## 'data.frame':    104 obs. of  17 variables:
##  $ species_name          : chr  "Abies_amabilis" "Abies_concolor" "Abies_grandis" "Abies_lasiocarpa"
##  $ family                : chr  "Pinaceae" "Pinaceae" "Pinaceae" "Pinaceae" ...
##  $ genus                 : chr  "Abies" "Abies" "Abies" "Abies" ...
##  $ epithet               : chr  "amabilis" "concolor" "grandis" "lasiocarpa" ...
##  $ seed_development_years: int  2 2 2 2 2 2 2 2 2 2 ...
##  $ pollinator_code       : chr  "wind" "wind" "wind" "wind" ...
##  $ mycorrhiza_type       : chr  "EM" "EM" "EM" "EM" ...
##  $ needleleaf_broadleaf  : chr  "needleleaf" "needleleaf" "needleleaf" "needleleaf" ...
##  $ deciduous_evergreen   : chr  "evergreen" "evergreen" "evergreen" "evergreen" ...
##  $ seed_maturation_timing: chr  "late summer" "fall" "late summer" "late summer" ...
##  $ seed_mass_mg          : num  46.2 34.3 21.1 13.7 78.4 ...
##  $ sexual_system         : chr  "monoecious" "monoecious" "monoecious" "monoecious" ...
##  $ shade_tolerance       : chr  "tolerant" "tolerant" "tolerant" "tolerant" ...
##  $ growth_form           : chr  "tree" "tree" "tree" "tree" ...
##  $ seed_bank             : chr  "no" "no" "no" "no" ...
##  $ fleshy_fruit          : chr  "no" "no" "no" "no" ...
##  $ dispersal_syndrome    : chr  "abiotic" "abiotic" "abiotic" "abiotic" ...
```

Checkpoint 1: From the `str()` function, answer the following questions.

- **What are the statistical individuals in this data set?**
- **How many characteristics are measured about each statistical individual?**
- **How many characteristics are categorical and how many are numerical.**

Let's look at some of the data. The data set has a variable called "needleleaf_broadleaf" which indicates the kinds of leaves the individual has. Let's see what possible values this can take.

```
table(tree.df$needleleaf_broadleaf)
```

```
##
##  broadleaf needleleaf
##         88         16
```

Okay, there are two values, either broadleaf (think oak and maple style leaves) or needleleaf (think pine tree needles). Let's sample an individual and see what kind of leaf it has. We can do this with the `sample()` function. All we do is input the object we want to sample from, how many times we want to sample it (using the argument `size`), and whether we want to sample with replacement (using the argument `replace`). The object we want to sample is the leaves.

```
set.seed(1)
sample(tree.df$needleleaf_broadleaf, # sample from the needleleaf_broadleaf column
       size = 1) # Sample 1 individual. No need to worry about replace if we just take 1
```

```
## [1] "broadleaf"
```

Hey, we got a broadleaf. That makes sense. Most of them are broadleaf. What species did we look at? To do that, we need to sample a bit differently. Let's sample *a row from the dataframe.*

```
set.seed(1)
sample.indx <- sample(1:nrow(tree.df), size = 1)
tree.df[sample.indx,]
```

```
##               species_name        family        genus    epithet
## 68 Heteropterys_laurifolia Malpighiaceae Heteropterys laurifolia
##    seed_development_years pollinator_code mycorrhiza_type needleleaf_broadleaf
## 68                      1          animal              AM            broadleaf
##    deciduous_evergreen seed_maturation_timing seed_mass_mg sexual_system
## 68           evergreen                   fall        71.82 hermaphrodite
##    shade_tolerance growth_form seed_bank fleshy_fruit dispersal_syndrome
## 68    intermediate       liana        no           no            abiotic
```

```
tree.df[sample.indx,'species_name']
```

```
## [1] "Heteropterys_laurifolia"
```

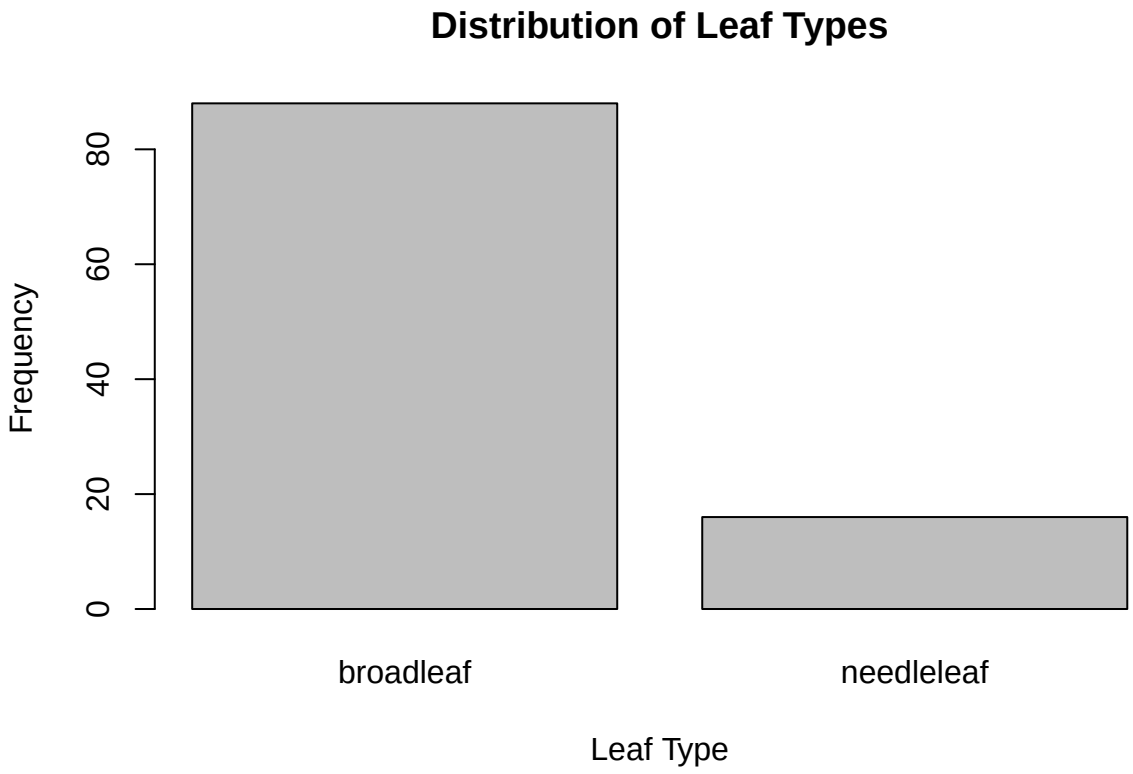
Checkpoint 2: Search where this species is found. Where is it found and what is its common name? Now, you sample your own tree. To do this, we have to undo something done before. We have used the function `set.seed()`, which allows us to control whether the sampling process is repeatable. In essence, R has a whole bunch of random numbers that it goes through when you sample. One way to get the exact same random sample is to use the exact same random numbers. Each set of random numbers is indicated by a 'seed'. Here, we have set the seed to 1. As long as that is the case, the `sample` function will return the same samples. To undo this, we use the following code

```
set.seed(Sys.time())
```

This sets the random number seed to be the same as the internal clock on the computer, which is unique to all times. In that case, you don't get the same species.

Checkpoint 3: Select a random tree and show how it is pollinated and what its sexual system is. We started by looking at how many trees there are with broad leaves versus needle leaves. We can plot this visually with a barplot.

```
leaftype.counts <- table(tree.df$needleleaf_broadleaf)
barplot(leaftype.counts, xlab = 'Leaf Type', ylab = 'Frequency',
        main = 'Distribution of Leaf Types')
```

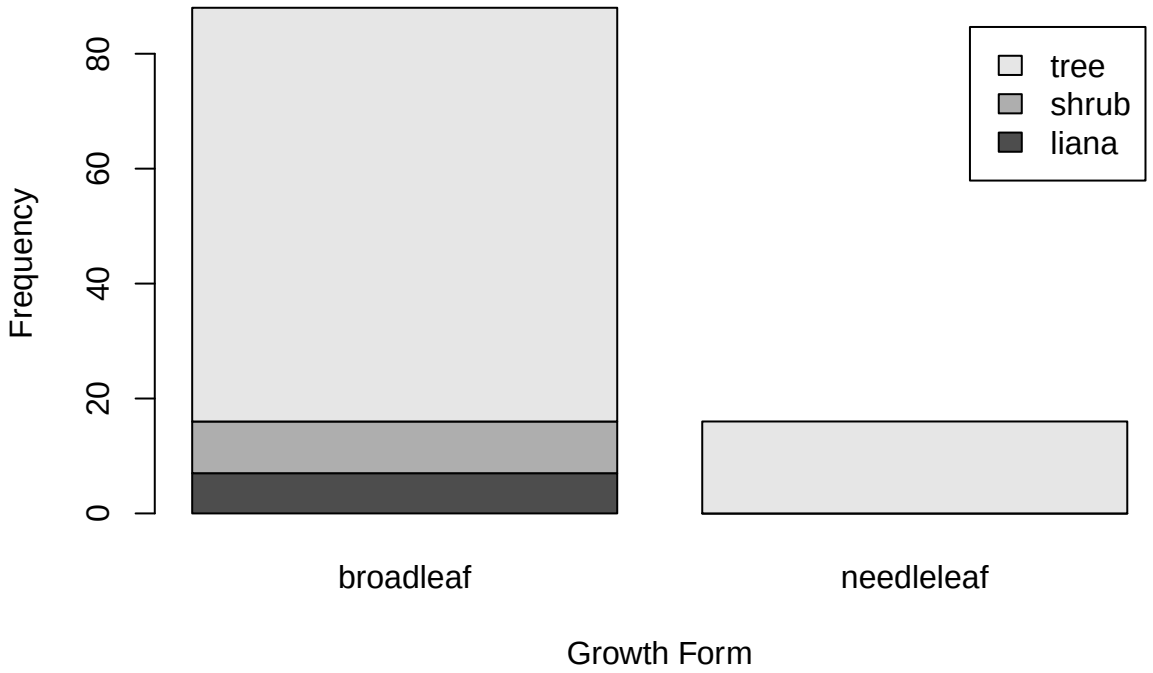


Interestingly, we can also see how leaf type changes for different plant growth forms.

```
(growthform.by.leaftype <- table(tree.df$growth_form,tree.df$needleleaf_broadleaf))
```

```
##
##         broadleaf needleleaf
##   liana         7          0
##   shrub         9          0
##   tree         72         16
```

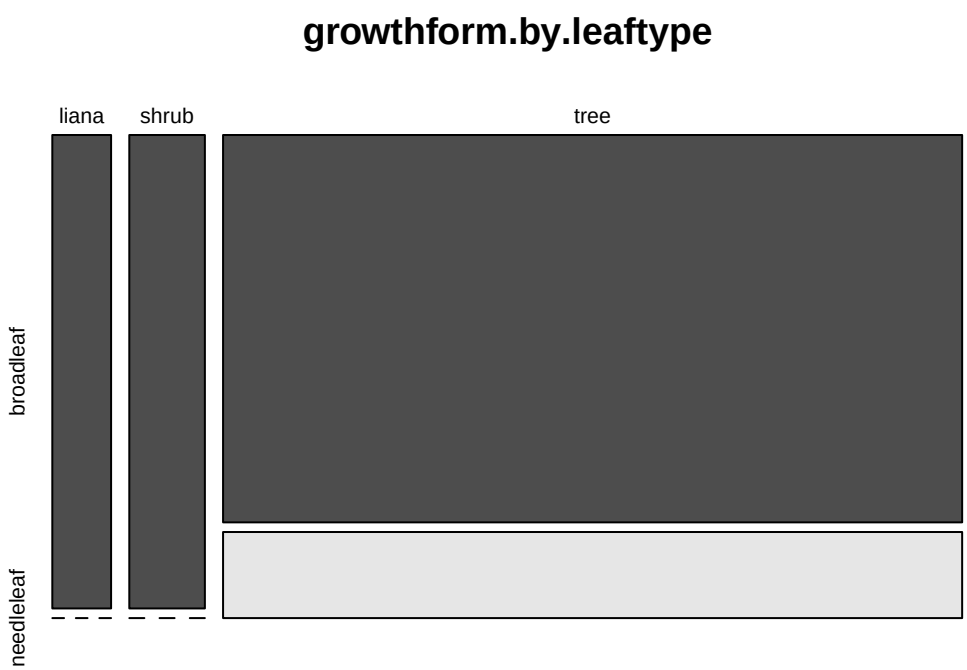
```
barplot(growthform.by.leaftype,
        xlab = 'Growth Form', ylab = 'Frequency',
        legend = T)
```



This does an okay job showing that needle leaves are only found in trees, not on shrubs or lianas.

Another way to look at this as a probability space is to use a **mosaic plot**.

```
mosaicplot(growthform.by.leaftype,
           color = T)
```



These mosaic plots are nice because the area of each rectangle represents the probability of randomly selecting a type in the set. The total black area represents the probability of selecting broad leaf plants. The gray area represents the probability of finding a need leaf tree in the dataset. The left black bar is the probability of selecting a liana (i.e., a woody vine) with broad leaves. Notice that there is no gray box below it, indicating that the probability of selecting a liana with needle leaves is zero.

Checkpoint 4: Is sampling a tree and sampling a needle leaf plan mutually exclusive? Explain why. This is an example of what is called **non-independence** of two variables. Non-independence indicates that the probability of seeing an outcome in one variable depends on outcomes from another variable. Here, we know that needles are not too common. However, if we know they don't occur when looking at lianas or shrubs. This also means the probability of selecting a needle leaf plant is higher if we are only looking at trees.

The `growthform.by.type` table can illustrate this fact if we convert counts into probabilities by dividing by the number of species we have.

```
num.species <- nrow(tree.df)
growthform.by.leaftype/num.species
```

```
##
##          broadleaf needleleaf
##   liana 0.06730769 0.00000000
##   shrub 0.08653846 0.00000000
##   tree  0.69230769 0.15384615
```

If we sum across columns, that gives the probability of each leaf type. If we sum across rows, we get the probability of each shrub type. You can see that the total

```
(leaf.type.prob <- colSums(growthform.by.leaftype/num.species))
```

```
##  broadleaf needleleaf
##  0.8461538  0.1538462
```

```
(growth.form.prob <- rowSums(growthform.by.leaftype/num.species))
```

```
##      liana      shrub       tree
## 0.06730769 0.08653846 0.84615385
```

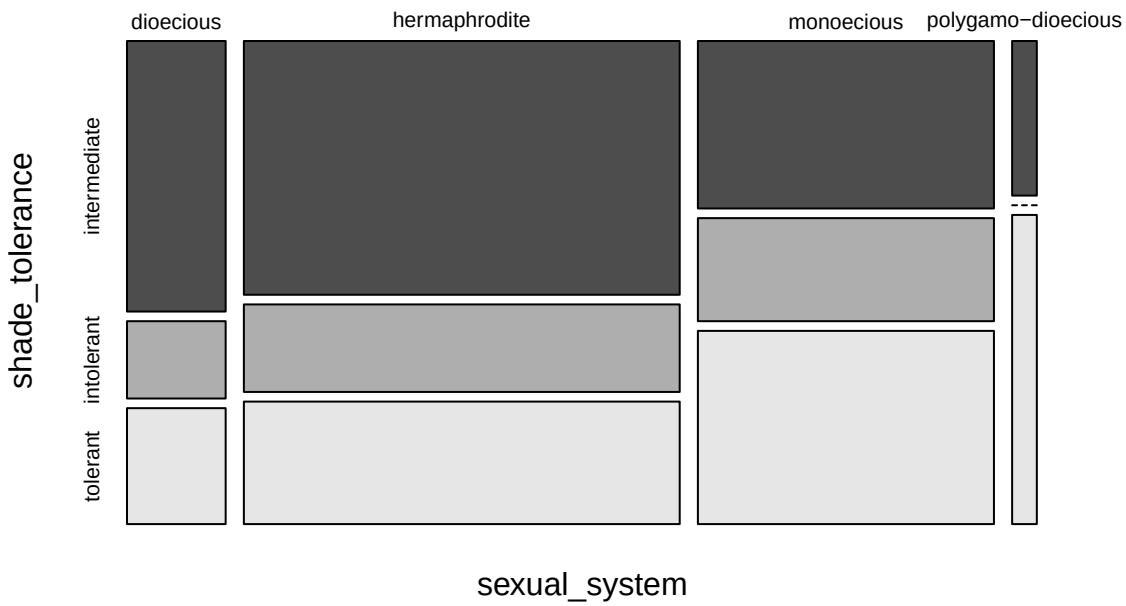
The main importance of non-independence is it tells us if knowing something about one variable changes our expectations about another. This is a measure of information. You can easily identify evidence of non-independence in mosaic plots, by asking if the relative size of blocks in a column change across columns. You interpret this as the probability of different outcomes changes as you have information about other variables.

For our purposes here, we are assuming these species make up the entirety of the species of interest, and so their frequencies in the dataset are taken as true probabilities. In reality, frequencies in a data set are only an ***estimate*** of the true probabilities. We need statistical tools to evaluate the evidence for non-independence. One such tool is a χ^2 test (spelled 'chi-squared' and read like 'pie', but with a hard c) that we will talk about later in the course.

Whether you recognize it or not, all of regression analysis is about identifying and characterizing non-independence! Sketch a linear regression style figure and see if you can figure out why! If you need help figuring out why, ask!

We can make mosaic plots using function notation as well. This just makes the code a bit easier to write and read. Say we want to see the probability of selecting a particular sexual system and shade tolerance. We can write this with function notation by writing `~ variable1 + variable2` in the first argument of `mosaicplot` and the supplying the data in the second argument.

```
mosaicplot(~ sexual_system + shade_tolerance, data = tree.df,
           color = T)
```



Checkpoint 5: Based on this figure, which sexual system has the highest probability of being selected? Which sexual system by shade tolerance combination has the highest probability of being selected?

Checkpoint 6: Make a mosaic plot to visualize whether the presence of fleshy fruits is independent of whether or not the plant species has a seed bank. Based on this figure, present an argument for or against independence of these two characters. If these probabilities bear out, it means that we should see the frequencies of these individuals in a random sample that is very similar to the predicted probabilities. Again, we can use the function `sample` to do this for us. Let's repeat the process of sampling a single individual, but we want to do it *many many* times and ask whether the frequencies in our samples matches the probabilities. For example, let's evaluate the probability of finding a shrub. From our work above, it looks like the probability of sampling a shrub is only $p_{\text{shrub}} = 0.0865$, meaning only 8.65% of individuals should be shrubs if we randomly sample. Let's try it out.

```
num.repeated.samples <- 10000 # That's a lot of individuals to sample
set.seed(1)
sample.indx <- sample(1:nrow(tree.df), # sample the individuals
                      size = num.repeated.samples, # sample an individual many times
                      replace = T) # make sure to put them back before resampling
sample.trees <- tree.df[sample.indx,]
head(sample.trees)
```

```
##               species_name        family        genus       epithet
## 68 Heteropterys_laurifolia Malpighiaceae Heteropterys    laurifolia
## 39    Robinia_pseudoacacia      Fabaceae      Robinia  pseudoacacia
## 1           Abies_amabilis      Pinaceae        Abies      amabilis
## 34   Quercus_ellipsoidalis      Fagaceae      Quercus ellipsoidalis
## 87   Palicourea_croceoides     Rubiaceae   Palicourea    croceoides
## 43     Alchornea_latifolia Euphorbiaceae    Alchornea     latifolia
##    seed_development_years pollinator_code mycorrhiza_type needleleaf_broadleaf
## 68                      1          animal              AM            broadleaf
## 39                      2          animal              AM            broadleaf
```

```
## 1                      2           wind             EM          needleleaf
## 34                     3           wind             EM           broadleaf
## 87                     1         animal             AM           broadleaf
## 43                     1         animal             AM           broadleaf
##    deciduous_evergreen seed_maturation_timing seed_mass_mg sexual_system
## 68           evergreen                  fall     71.82000 hermaphrodite
## 39           deciduous                  fall     19.00557 hermaphrodite
## 1            evergreen           late summer     46.20634    monoecious
## 34           deciduous     late summer - fall   1637.96613    monoecious
## 87           evergreen        spring - summer    178.88889 hermaphrodite
## 43           evergreen                summer     29.01929     dioecious
##    shade_tolerance growth_form seed_bank fleshy_fruit dispersal_syndrome
## 68    intermediate       liana        no           no            abiotic
## 39    intermediate        tree       yes           no            abiotic
## 1         tolerant        tree        no           no            abiotic
## 34      intolerant        tree        no           no        synzoochory
## 87      intolerant       shrub        no          yes       endozoochory
## 43    intermediate        tree        no          yes       endozoochory
```

Alright, we've got our 10,000 individual trees. Let's look at how many are shrubs.

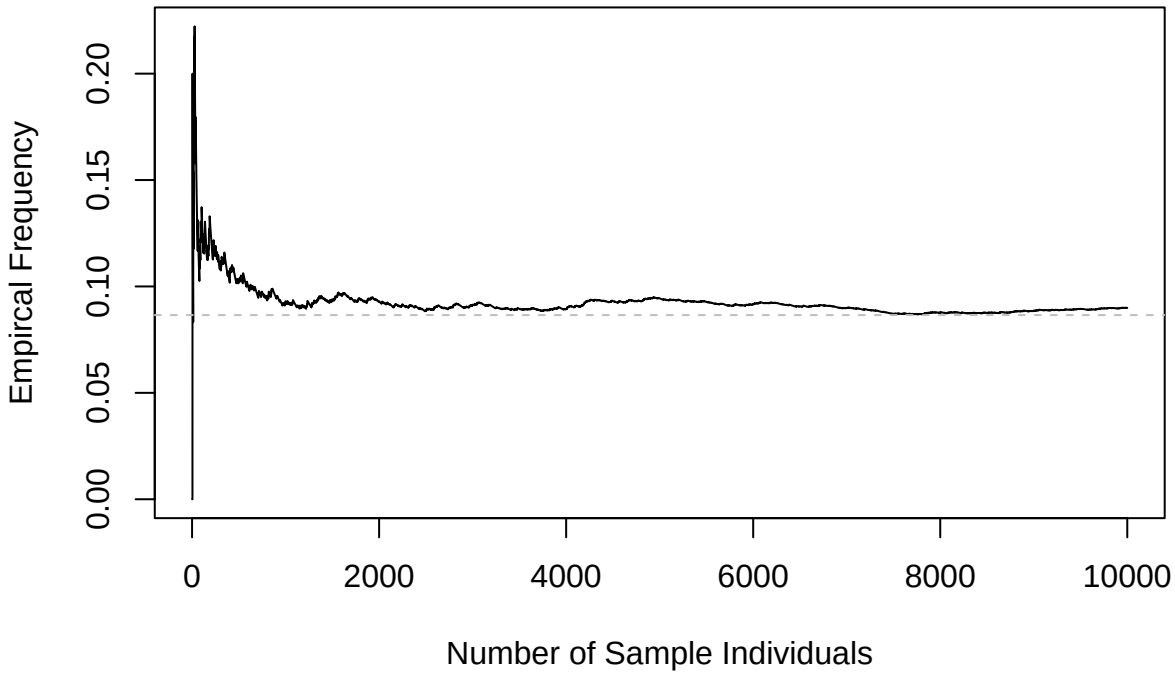
```
table(sample.trees$growth_form)/nrow(sample.trees)
```

```
##
##  liana  shrub   tree
## 0.0689 0.0899 0.8412
```

Look at that. About 8.6% of the samples are shrubs, which is very close to the probability of 8.65%.

We can make a figure to show how this empirical frequencies match the probability as we increase the number of samples. (Don't worry about how this code works right now. I want you to see the figure right now and we will learn about this kind of programming later.)

```
emp.freq <- rep(NA, num.repeated.samples)
for (i in 1:num.repeated.samples){
  emp.freq[i] <- sum(sample.trees[1:i,'growth_form'] == 'shrub')/i
}
plot(emp.freq, typ = 'l',
     xlab = 'Number of Sample Individuals',
     ylab = 'Empircal Frequency',
     main = 'Estimate Convergence in Large Samples'     )
abline(h = sum(tree.df$growth_form == 'shrub')/nrow(tree.df),
       lty = 2, col = 'gray')
```



This is an example of what is called "the law of large numbers". It states that empirical estimates of a quantity eventually converge of the true quantity with sufficient large random samples. You can see that early on, with few samples, we could estimate a probability pretty far from the actual value (shown by the horizontal dashed line). But with enough samples, we do a pretty good job. One question you might have is "how many is enough?" Pretty good question! Hold tight, we'll get there soon.

Probability Modeling

The above exercise was really about exploring the concepts of probability. Probability modeling is all about describing the process of collecting data with simple probability models that match particular assumptions about the biology of the individuals of interest and the process of sampling those individuals.

Probability modeling can be done in many ways, but one of the most helpful is to think in terms of known probability distributions.

We have already talked about a couple of these. One is Bernoulli distribution. The other is the normal distribution. All distributions have **parameters** that determine the shape of the probability distribution. To make this more concrete, let's look at some examples.

Bernoulli Distribution

To get started, remember that the Bernoulli distribution has a single parameter, p, that represents the probability of a 'success'. What counts as a 'success'? Well, a Bernoulli distribution applies to random variables with only two outcomes, so a 'success' is whichever outcome you care about.

Imagine that we have two a lake with three species of fish: a bass, a trout, and a perch. Let's imagine that we know that there are 10,000 fish in the lake and we know exactly how frequent each is in the lake. 30% of all fish are bass, 10% are trout, and 60% are perch. If we are interested in whether we catch a trout, "success" has a probability of 10% (or 0.1). By the same token, a "failure" is any fish that is **not** a trout. Here, that means a perch or a bass. If catching a trout has a 10% chance of occurring, then catching anything that is not a trout has a 90% chance of occurring.

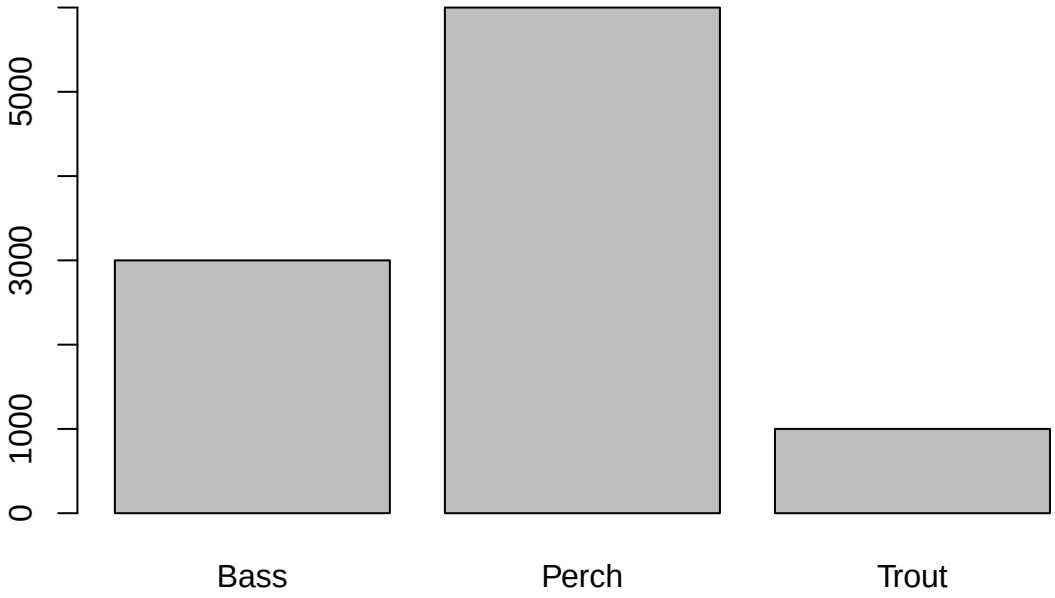
We can model this lake as follows.

```
fish.species <- c('Bass', 'Trout', 'Perch')
frequency <- c(0.3, 0.1, 0.6)
total.fish <- 10000

fish <- rep(fish.species, times = total.fish*frequency)
table(fish)
```

```
## fish
##  Bass Perch Trout
##  3000  6000  1000
```

```
barplot(table(fish))
```



There is our lake.

Now imagine we want to sample a fish from this lake. We can do this with the function `sample`. Let's sample a fish.

```
(sample.fish <- sample(fish, size = 1, replace = T))
```

```
## [1] "Perch"
```

Turns out the function `sample` also has an argument for including a vector of probabilities. This simplifies things. We could sample our lake instead like this

```
(sample.fish <- sample(c("Bass", "Trout", "Perch"),
                       size = 1, replace = T,
                       prob = c(0.3,0.1,0.6)))
```

```
## [1] "Bass"
```

That negates the need for making the whole lake of fish. We can just specify the probabilities right away.

Another example is flipping a coin. We did this once already.

```
coin <- c('Heads', 'Tails')
sample(coin, size = 1)
```

```
## [1] "Heads"
```

Now maybe we want to know what we get if we have a biased coin. Let's bias it way towards tails. Say tails is 5 times as likely to show up as heads. That means that Pr(Tails)/Pr(Heads) = 5. (Note that this way of framing probability is called an odds. Even odds mean a fair coin.)

You can always convert odds to probabilities and back again. First, assume you have the probability of success p. Then the odds of success, $O(\text{success})$, are

$$O(\text{success}) = \frac{Pr(\text{success})}{Pr(\text{failure})} = \frac{Pr(\text{success})}{1 - Pr(\text{success})} = \frac{p}{1-p}.$$

Similarly, if you have the odds, $O(\text{success})$, then the probability of success is

$$O(\text{success}) = p/(1-p) \rightarrow O(\text{success})(1-p) = p \rightarrow p = \frac{O(\text{success})}{1 + O(\text{success})},$$

So if the odds are 9, then the probability is $9/10 = 0.9$.

Another way to write the relationship between probability and odds is

$$p = \frac{1}{1 + \frac{1}{O(\text{success})}} \rightarrow O(\text{success}) = \frac{1}{1 - \frac{1}{p}}.$$

```
odds.tails <- 5
p.tails <- odds.tails/(1 + odds.tails)
p.heads <- 1 - p.tails

# Check that it works
p.tails/p.heads # should be odds.tails
```

```
## [1] 5
```

```
p.tails + p.heads # should be 1
```

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

Now that we have the probability of tails, we can sample our biased coin

```
sample(coin, size = 1, prob = c(p.heads, p.tails))
```

```
## [1] "Tails"
```

Checkpoint 7: Create a Bernoulli random variable that describes whether an individual has a mutant allele. Let the probability of the individual having the allele be 0.01. Sample an individual from this random variable and tell me what it is. Make sure to set the seed in your code so that it is repeatable on my computer.

Binomial Distribution

An extension of the Bernoulli is the Binomial distribution. The Binomial distribution models *how many successes* are in n samples of a Bernoulli random variable with probability of success p. Let's take our fish example and ask how many trout we get if we sample 10 fish with replacement. We could do this sample using the `sample` function.

```
set.seed(1)
(fish.sample <- sample(fish, size = 10, replace = T))
```

```
##  [1] "Bass"  "Perch" "Perch" "Perch" "Perch" "Perch" "Perch" "Bass"  "Perch"
## [10] "Bass"
```

```
sum(fish.sample == 'Trout')
```

```
## [1] 0
```

In this case we got zero.

R has a clever way of doing this same thing. For many probability distributions, it can provide

1. A random sample from the distribution
2. Probabilities of any outcome
3. Cumulative probabilities of any outcome
4. Quantiles of the distribution

For the binomial distribution, these are given by the functions

1. `rbinom`
2. `dbinom`
3. `pbinom`
4. `qbinom`

To see how this work. Let's first put grab a single sample from this distribution. We collect 10 fish where the probability of success is $p = 0.1$.

```
rbinom(n = 1, size = 10, prob = 0.1)
```

```
## [1] 3
```

```
# 1 random value from the distribution with 10 sampled fish
# and probability of catching a trout is 0.1.
```

Ha, we got 1 trout this time.

Now let's imagine that we repeat this sampling process 4 times. That is, we go to the same lake four times, catch 10 fish each time, and then ask how many of the 10 fish are trout. When doing this, we should get 4 numbers (1 for each sample). Each number represents the number of fish in that sample.

```
num.samples <- 4
fish.per.sample <- 10
p.trout <- 0.1
set.seed(1)
fishing.samples <- rbinom(n = num.samples,
                          size = fish.per.sample,
                          prob = p.trout)
fishing.samples
```

```
## [1] 0 1 1 2
```

In the first sample, we caught zero trout. We caught a single trout in the next two samples, and then we caught two trout in the last sample. The estimated frequencies of trout from each of these estimates is then

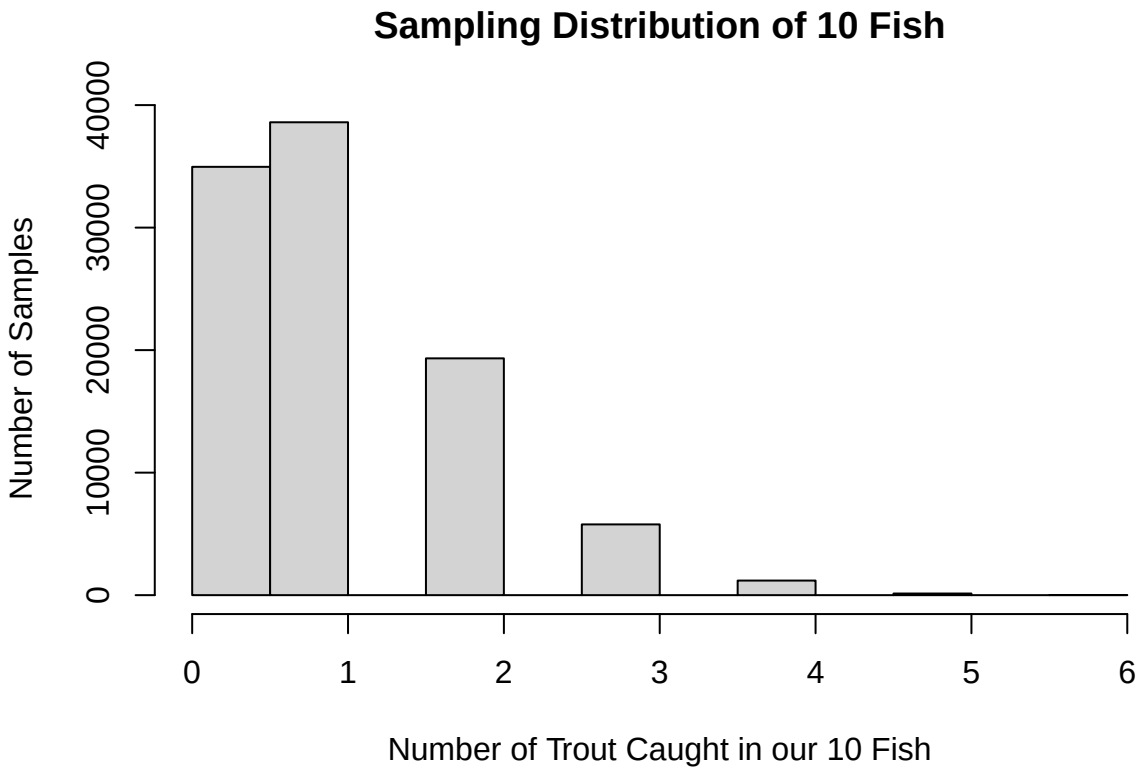
```
(est.trout.freq <- fishing.samples/fish.per.sample)
```

```
## [1] 0.0 0.1 0.1 0.2
```

This gives us a sense of how different our estimate might be if we went out and redid our sampling design catching four fish. The most likely outcome is that we estimate the probability of catching a trout is 0.1. But it's also possible we estimate that 20% of the fish are trout, and it's also possible we estimate there are no trout!

Let's do this a lot of times, which is the hypothetical scenario of re-doing our sampling procedure thousands of times. Under those thousands of times, we should be able to figure any plausible estimate of trout prevalence.

```
num.samples <- 100000 # Repeat our sampling scheme 100,000 times
fishing.samples <- rbinom(n = num.samples,
                          size = fish.per.sample,
                          prob = p.trout)
hist(fishing.samples,
     xlab = 'Number of Trout Caught in our 10 Fish',
     ylab = 'Number of Samples',
     main = 'Sampling Distribution of 10 Fish')
```



This graph shows a probability distribution for the procedure of catching 10 fish and determining how many are trout. Essentially, this says we are most likely to catch 0 or 1 trout. There is a substantial change we catch 2 fish. And dwindling chances we catch 3, 4, and 5. In very rare instances, we could catch 6!

The Binomial distribution tells us these exact probabilities. We don't actually need to randomly sample. To do that, we ask for the *probability density function* using `dbinom()`. The `d` is for 'density' and the `binom` means that it applies to the binomial distribution. We give it parameters (number of individuals sampled and the probability of success) and a set of possible outcomes and it spits back out the probability of each outcome.

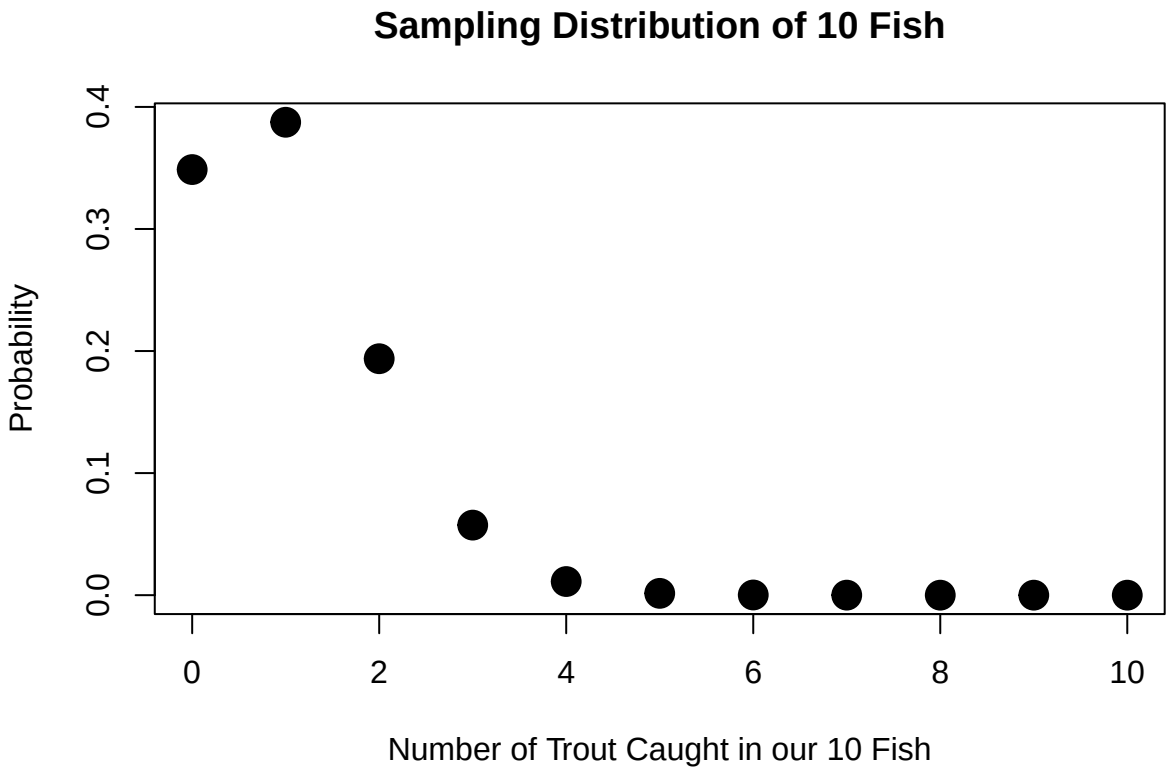
```
num.fish.per.sample <- 10 # Sample size
possible.outcomes <- 0:num.fish.per.sample # Set of possible outcomes (# trout in 10 fish)

(catch.prob <- dbinom(possible.outcomes, # Give it the possible outcomes to evaluate probabilities
       size = num.fish.per.sample, # Tell it the sample size
       prob = p.trout)) # Tell it the probbaility of success
```

```
##  [1] 0.3486784401 0.3874204890 0.1937102445 0.0573956280 0.0111602610
##  [6] 0.0014880348 0.0001377810 0.0000087480 0.0000003645 0.0000000090
## [11] 0.0000000001
```

That's a bunch of numbers. Let's look at a plot of the distribution.

```
plot(possible.outcomes, catch.prob,
     xlab = 'Number of Trout Caught in our 10 Fish',
     ylab = 'Probability',
     main = 'Sampling Distribution of 10 Fish',
     cex = 2, pch = 19)
```



Checkpoint 8: What is the probability of catching 3 trout in this sampling design?

Checkpoint 9: Write code to show the sampling distribution of the number of trout in 20 fish, rather than 10, under the assumption that $p_{trout} = 0.1$. There are other ways to characterize a probability distribution that gives you information about probabilities and how they are apportioned across different outcomes. These are called **cumulative distributions**. Cumulative distributions show how *probability accumulates* as you move from smaller valued to larger valued outcomes. For example, we could ask, what is the probability of finding **at most 5 trout** (i.e., 5 or fewer). A way to think about this is that

$$Pr(5 \text{ or fewer}) = Pr(5 \text{ or } 4 \text{ or } 3 \text{ or } 2 \text{ or } 1 \text{ or } 0).$$

Since each event is mutually exclusive, we can just add their probabilities together to get an answer for $Pr(5 \text{ or fewer})$.

```
sum(dbinom(0:5, # Give it the possible outcomes to evaluate probabilities
       size = num.fish.per.sample, # Tell it the sample size
       prob = p.trout))
```

```
## [1] 0.9998531
```

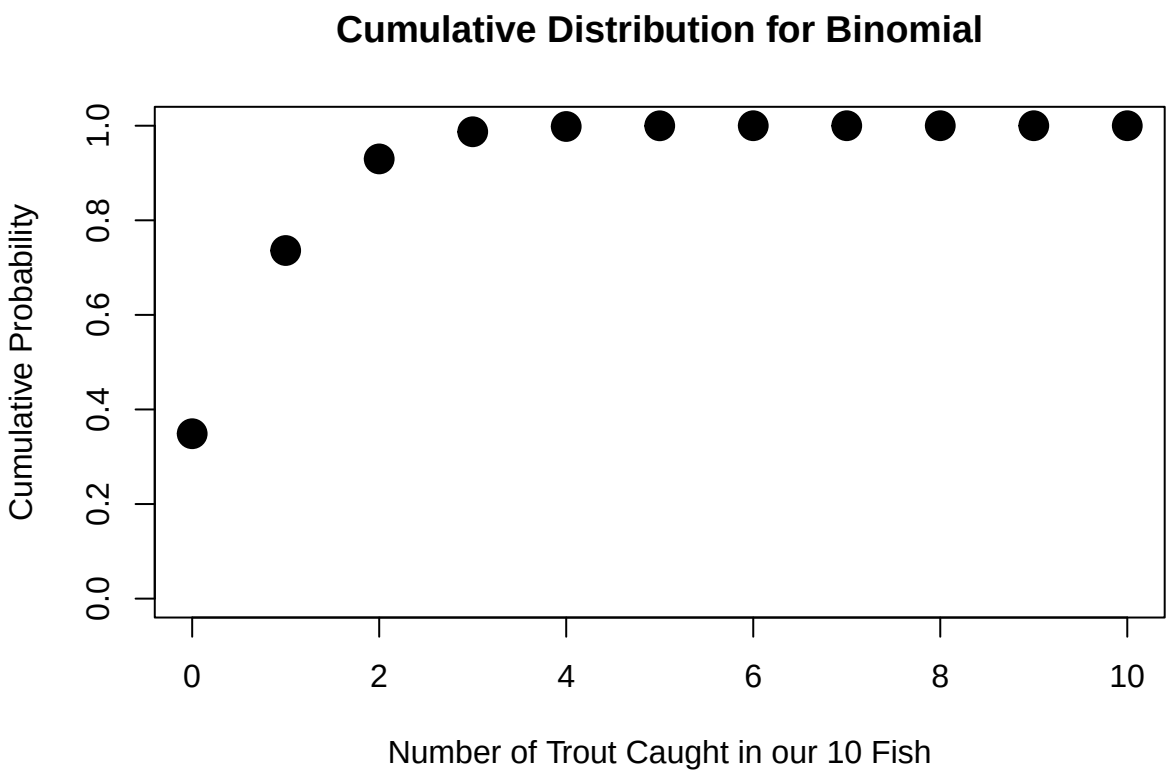
Seems it is almost certain we catch 5 or fewer.

Cumulative distributions do this for us. We use the function `pbinom`. Again, we give it outcomes and parameters (sample size and probability of success) and it will spit out cumulative probabilities.

```
(cumulative.prob <- pbinom(possible.outcomes,
       size = num.fish.per.sample,
       prob = p.trout))
```

```
##  [1] 0.3486784 0.7360989 0.9298092 0.9872048 0.9983651 0.9998531 0.9999909
##  [8] 0.9999996 1.0000000 1.0000000 1.0000000
```

```
plot(possible.outcomes, cumulative.prob,
     xlab = 'Number of Trout Caught in our 10 Fish',
     ylab = 'Cumulative Probability',
     pch = 19, cex = 2,
     main = 'Cumulative Distribution for Binomial',
     ylim = c(0,1))
```



This shows us how probability accumulates as we consider catching few trout to catching many. We see that most the probability of catching 7 or fewer really no different than catching 3 or fewer. That is an indication that most of the likely outcomes are 3 and below. **Probabilities are highest where cumulative probabilities change the fastest.**

Cumulative probabilities end up be very helpful for another purpose: finding quantiles. Remember that quantiles from data analysis represent the observation in the ordered data that is in the "quantile(th)" position. The median is the 50th quantile, meaning it is the data 50% of the way through the ordered data. The 99.9% quantile is the data point is that is 99.9% of the way through the ordered data. Probability distributions have quantiles too, but the quantiles relate to the percentage of the way through probability.

We can find quantiles of probability distributions using the function `qbinom` where the `q` means quantile and the `binom`. Here is an example. Let's find the 25th, 50th, and 75th quantiles of this sampling distribution. We have to give the function the quantiles we want and the distribution parameters (again, the sample size and the probability of success).

```
qbinom(c(0.25, 0.5, 0.75), # quantiles we want
       size = num.fish.per.sample, # number of fish in our sample
       prob = p.trout) # probability of a fish being a trout
```

```
## [1] 0 1 2
```

So there you go. The 25th quantile (or as it is sometimes called, the first quartile) shows that 25% of samples catch 0 trout. The 50th quantile (the median or second quartile) shows that half of samples catch 1 trout or none. And the 75th quantile (the third quartile) says that 75% of samples catch 2 trout or fewer.

Checkpoint 10: Plot the probability distribution and cumulative distribution for the following sample design. You are sampling 50 individuals for infection status where the probability that a single individual is infected is 0.13 and you want to know how many individuals in your sample are infected. Also, find the 30th and 80th quantiles for this distribution.

Normal Distribution

Now let's look into the normal distribution. The normal distribution has a mean (often written as μ) and a variance (often written as σ^2; or as the standard deviation, σ).

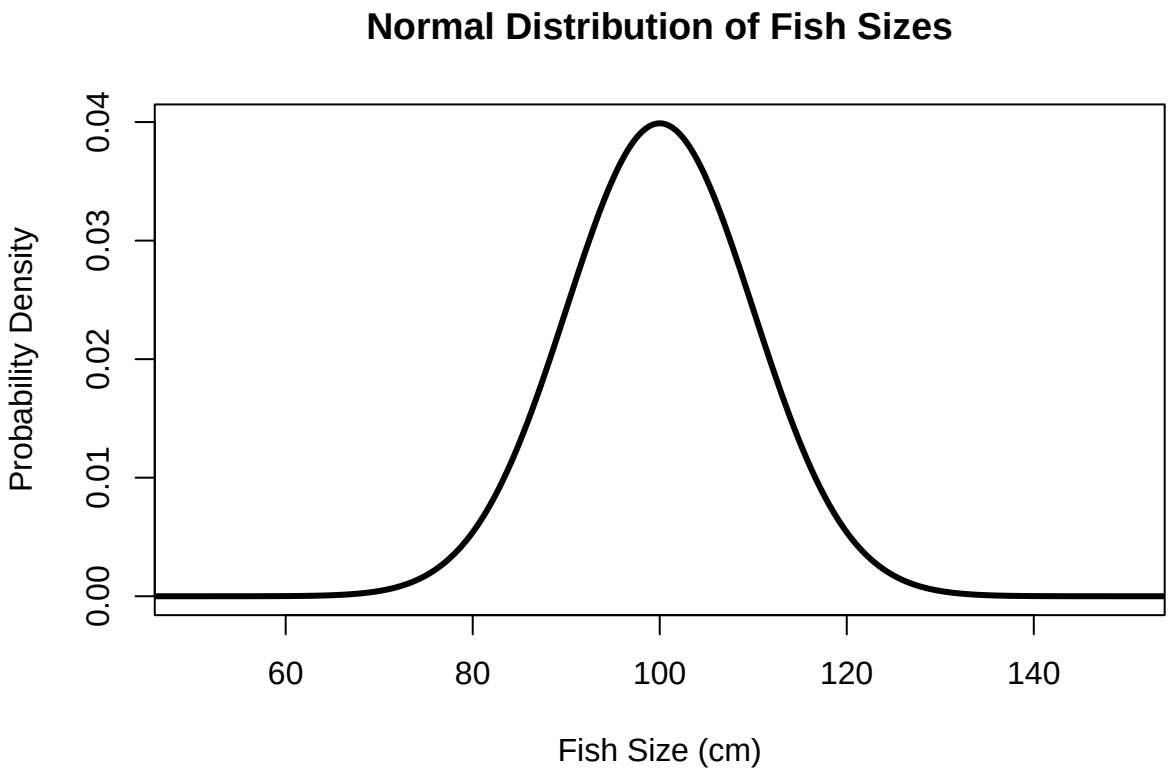
Let's assume fish size is normally distributed with mean $\mu = 100cm$ and standard deviation $\sigma = 10cm$. Let's grab a sample of ten individuals. Here, we use the function `rnorm` where `r` is for a random sample and `norm` is for the normal distribution. Again, we need to say out big our sample size is and give it parameters (mean and standard deviation).

```
sample.size <- 20
fish.size.mean <- 100
fish.size.sd <- 10
(fish.size.sample <- rnorm(n = sample.size,
                           mean = fish.size.mean,
                           sd = fish.size.sd))
```

```
##  [1] 111.38251 112.15134  95.75169  85.49160  92.37075  95.18320 110.49909
##  [8]  92.90284  83.97194 111.34294  93.27538 112.70380 108.53492  96.92831
## [15] 103.45245  98.04674  90.18554 105.89867 108.13525 107.80767
```

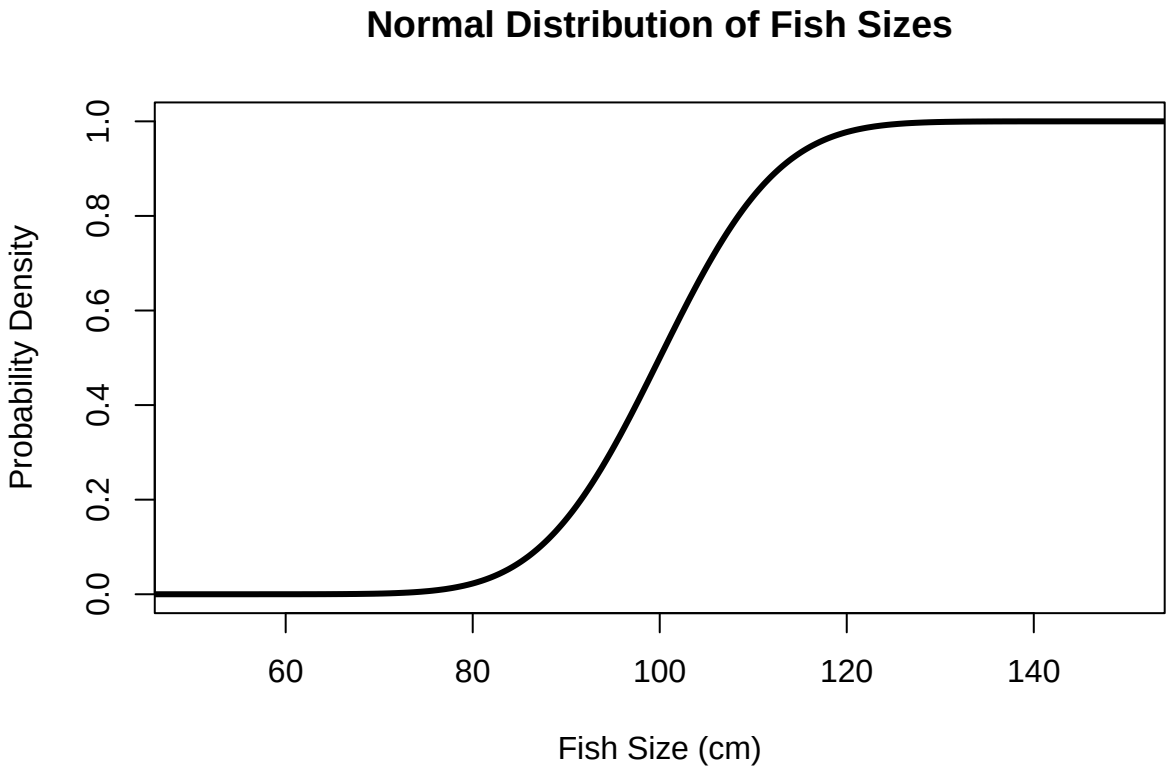
Now let's look at the distribution that gave this random sample.

```
fish.sizes <- seq(from = 0, to = 200, length = 1000) # A bunch of numbers between 0 and 200
size.prob <- dnorm(fish.sizes, # The probabilities associated with each fish size
                   mean = fish.size.mean,
                   sd = fish.size.sd)
plot(fish.sizes, size.prob,
     xlab = 'Fish Size (cm)',
     ylab = 'Probability Density',
     main = 'Normal Distribution of Fish Sizes',
     typ = 'l', lwd = 3,
     xlim = c(50,150))
```



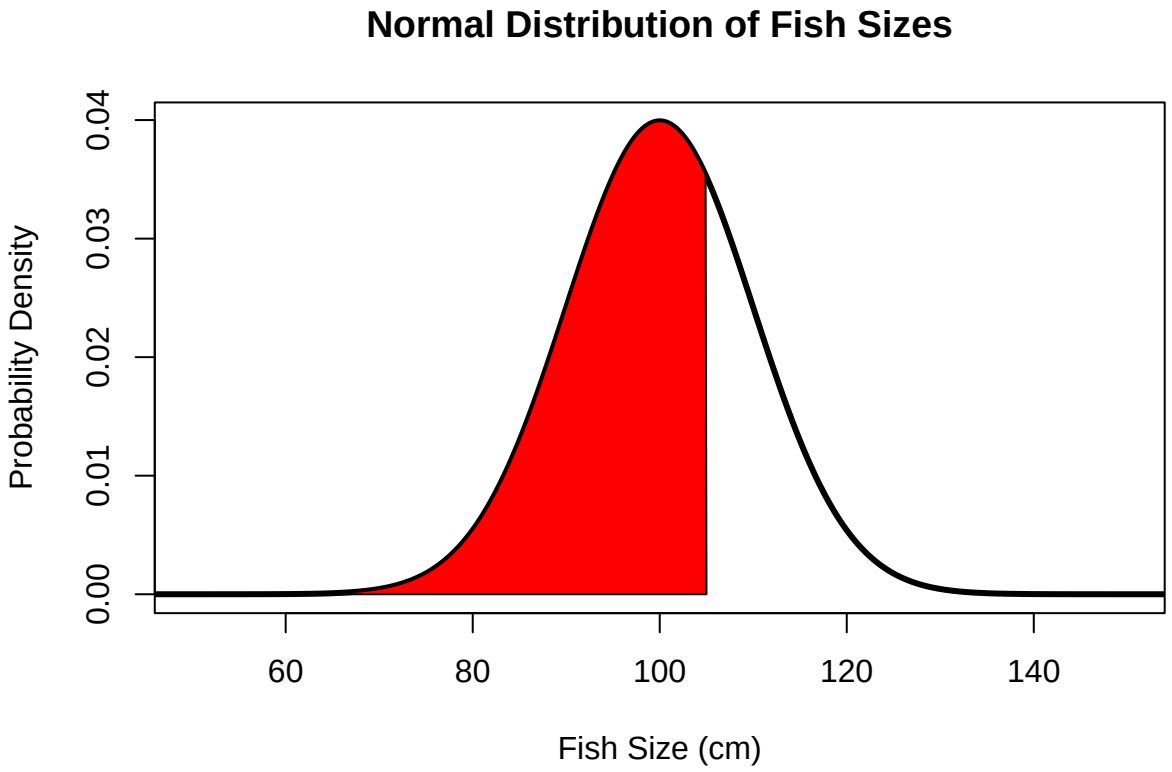
Here is the cumulative distribution for this normal.

```
size.cumulative.prob <- pnorm(fish.sizes,
                              mean = fish.size.mean,
                              sd = fish.size.sd)
plot(fish.sizes, size.cumulative.prob,
     xlab = 'Fish Size (cm)',
     ylab = 'Probability Density',
     main = 'Normal Distribution of Fish Sizes',
     typ = 'l', lwd = 3,
     xlim = c(50,150))
```



The cumulative probability corresponds to the area to the left of the the corresponding fish size on the pdf. Let's take the cumulative probability of fish size of 105cm. The area to the left of this point on the curve is visualized as

```
plot(fish.sizes, size.prob,
     xlab = 'Fish Size (cm)',
     ylab = 'Probability Density',
     main = 'Normal Distribution of Fish Sizes',
     typ = 'l', lwd = 3,
     xlim = c(50,150))
polygon(c(fish.sizes[fish.sizes<=105],105,0), c(size.prob[fish.sizes <= 105],0,0),
        col = 'red')
```

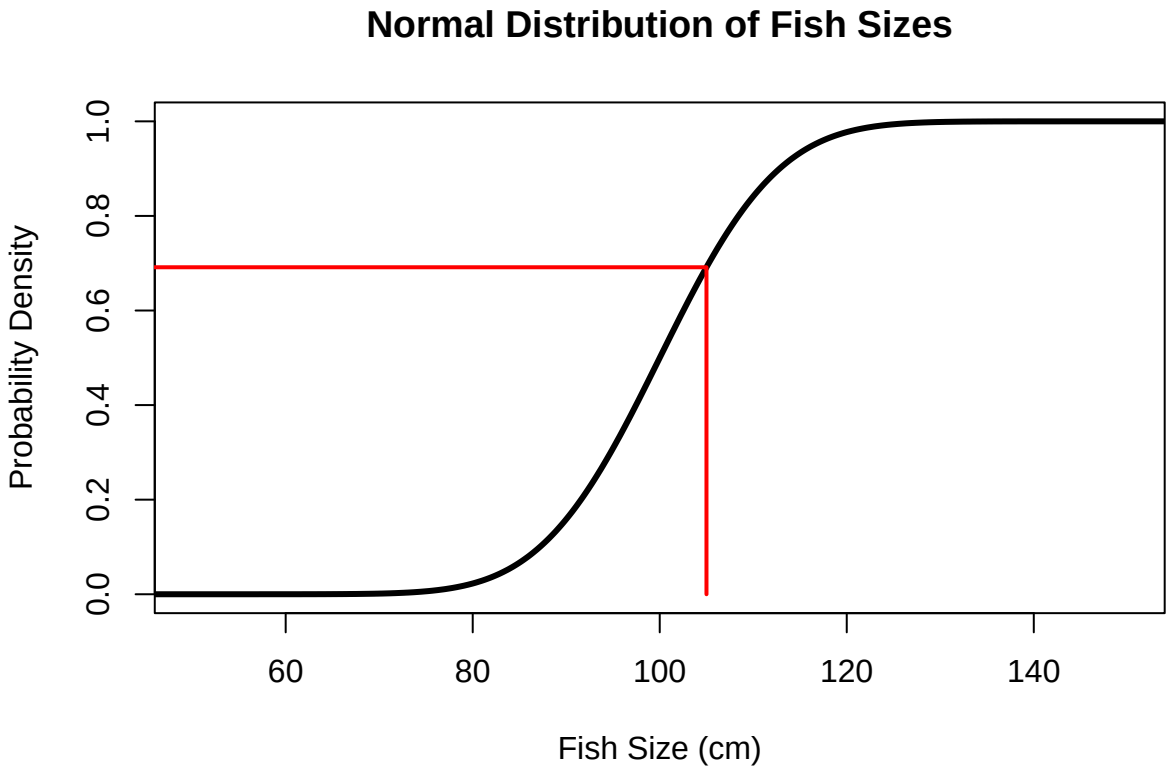


The red area is the total probability of catching a fish smaller than 105 cm. The cdf gives exactly this area!

```
(cumul.prob.105 <- pnorm(105, fish.size.mean, fish.size.sd))
```

```
## [1] 0.6914625
```

```
plot(fish.sizes, size.cumulative.prob,
     xlab = 'Fish Size (cm)',
     ylab = 'Probability Density',
     main = 'Normal Distribution of Fish Sizes',
     typ = 'l', lwd = 3,
     xlim = c(50,150))
lines(c(105, 105, 0), c(0,cumul.prob.105,cumul.prob.105),
      col = 'red', lwd = 2)
```



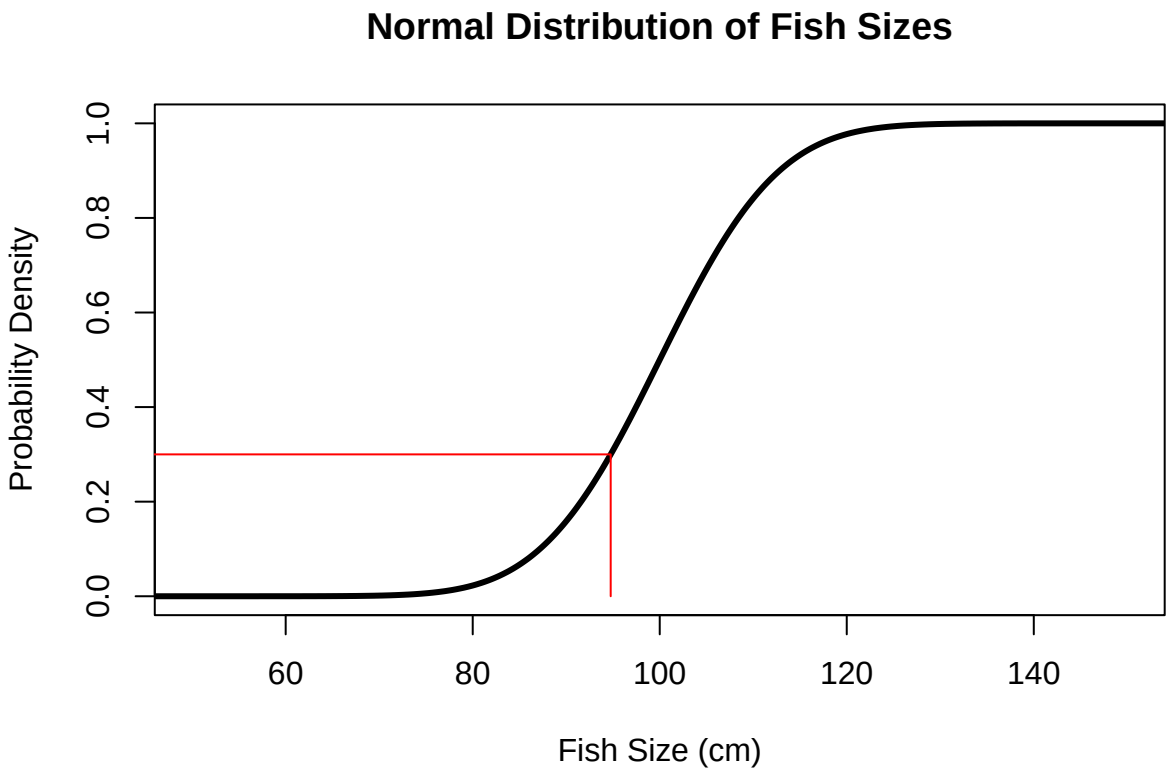
It's about 70%.

Now, let's find the 30th quantile for this distribution. The 30th quantile is the x-value where a line emanating from the y-axis at 0.3 intersects the cumulative probability curve. Here is the value and a visualization.

```
(fish.q.30 <- qnorm(0.3, mean = fish.size.mean, sd = fish.size.sd))
```

```
## [1] 94.75599
```

```
plot(fish.sizes, size.cumulative.prob,
     xlab = 'Fish Size (cm)',
     ylab = 'Probability Density',
     main = 'Normal Distribution of Fish Sizes',
     typ = 'l', lwd = 3,
     xlim = c(50,150))
lines(c(0, fish.q.30, fish.q.30), c(0.3,0.3,0), col = 'red', lwd = 1)
```



What does this 30th quantile mean? It means that 30% of the fish you catch in this population are smaller than about 95cm. That also means 70% of fish are larger than 95cm. That seems pretty useful to know!

Checkpoint 11: Write code that produces a two panel figure. In one, show the probability distribution for a normal with mean 5 and standard deviation of 20. In the other, show that distribution's cumulative density.

Checkpoint 12: Write code to find the 63rd and 12th quantiles of a normal distribution with mean 5 and standard deviation of 20.

References

Nigro, Katherine M., Jessica H. Barton, Diana Macias, V. Bala Chaudhary, Ian S. Pearse, David M. Bell, Angel Chen, et al. 2024. "Co-Mast: Harmonized Seed Production Data for Woody Plants Across US Long-Term Research Sites." *Ecology* 106 (1). https://doi.org/10.1002/ecy.4463.