Week 5 - Estimation from Sampling Distributions

Nicholas Kortessis

2025-02-12

Review of Sampling Distributions

We did sampling distributions last week as part of probability. Let's get to using them to infer things about the population.

Sampling distributions are **the probability distribution of a sample statistic.** We have talked about how we measure things about individuals, but statisticians are normally interested in the properties of populations. For example, we might be interested in the central tendency of a population. As such, we might measure the mean value across a bunch of individuals. We could be interested in other things. Below is a table of things we could be interested in about populations and what we might measure from our sample to learn something about the population.

Population Property	Sample Statistic
Central Tendency	Mean, median
Variability	Standard deviation, quantiles, minimum, maximum,
Measures of Association	Correlation, covariance, functional relationships

The goal in statistics is to infer things about the population from properties of samples. How do we make this jump? The way to make the jump is by studying and using the properties of ***sampling distributions***, which give the probability distribution for any sample statistic we are interested in.

The Binomial Distribution - The simplest sampling distribution

We have seen a sampling distribution before in the case of the binomial distribution. A binomial distribution gives the sampling distribution for the number of 'successes' in a random sample of n individuals from the same population, with the same probability that an individual has the characteristic denoted by a 'success'.

To see how this might work, let's consider that we are looking at frogs that can be infected with the chytrid fungus. Let's assume that there is a probability $p = 0.6$ that a frog is infected. If we randomly sample frogs, the number of infected frogs in a sample is given by the binomial distribution.

Let's see what the distribution looks like for a sample of $n = 5$ frogs.

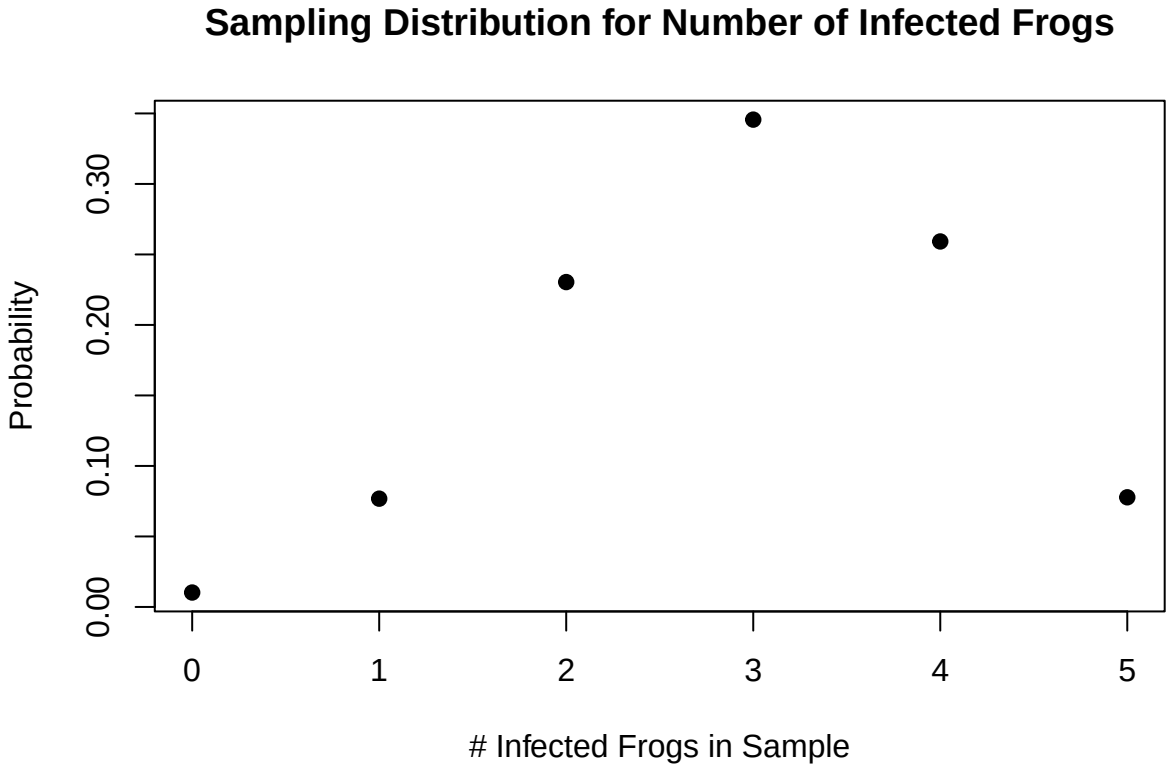
```
num.frogs.sampled <- 5
prob.inf <- 0.6

(poss.outcomes <- 0:num.frogs.sampled)
```

```
## [1] 0 1 2 3 4 5
```

```
(outcomes.probs <- dbinom(poss.outcomes, size = num.frogs.sampled, p = prob.inf))
```

```
## [1] 0.01024 0.07680 0.23040 0.34560 0.25920 0.07776
```

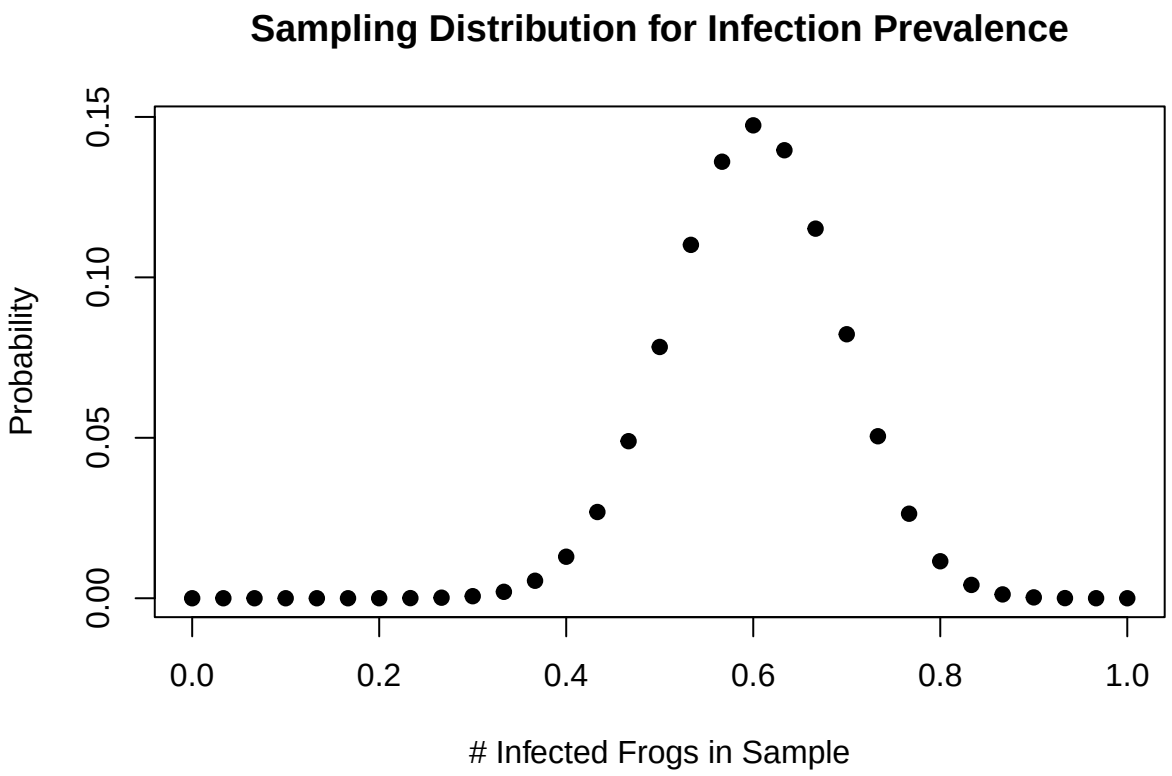
```r
plot(poss.outcomes, outcomes.probs, pch = 19,
     xlab = '# Infected Frogs in Sample', ylab = 'Probability',
     main = 'Sampling Distribution for Number of Infected Frogs')
```



This says that we are most likely to get a sample with three infected frogs out of 5, but we are also likely to get a sample with 2 or 3. Only 1 infected frog or all infected frogs are also less likely. The least likely outcome is that none of the frogs are infected.

Let's do this again, but now envision that we catch $n = 30$ frogs instead of 5. And this time, let's convert the possible outcomes to **infection prevalence**. Infection prevalence is simply the number of frogs infected divided by the total number of frogs. To do this, we take each possible outcome (# infected frogs) and divide by the number of frogs sampled.

```r
num.frogs.sampled <- 30
prob.inf <- 0.6

poss.outcomes <- 0:num.frogs.sampled
outcomes.probs <- dbinom(poss.outcomes, size = num.frogs.sampled, prob = prob.inf)

inf.prevalence <- poss.outcomes/num.frogs.sampled
plot(inf.prevalence, outcomes.probs, pch = 19,
     xlab = '# Infected Frogs in Sample', ylab = 'Probability',
     main = 'Sampling Distribution for Infection Prevalence')
```



Unsurprisingly, we are most likely to see a sample where 60% of the frogs are infected. But we could see as high as 80% infected or as low as 40% infected.

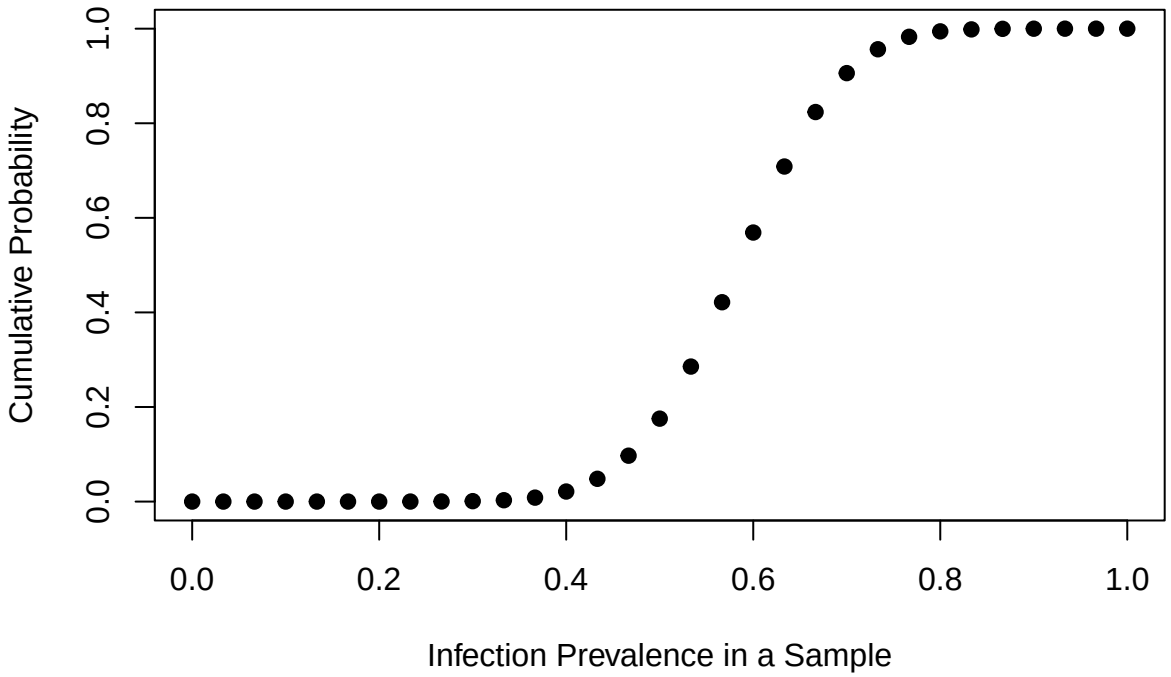
Checkpoint 1: Plot the sampling distribution of infection prevalence when we sample 100 frogs. What are the bounds on reasonable sample prevalence in this case?

Calculating the most likely outcomes

We now have a sampling distribution where we can look and decide what the likely samples are to look like. Let's find first the 50% most likely outcomes. The most likely outcomes are the ones in the middle of the peak near 60% infection prevalence. But where do we draw the line to denote the 50% most likely?

To do that, recognize that the smallest prevalence values are very unlikely and the highest prevalence values are also highly unlikely. Just as we talked about in lecture, we can easily find the 50% of least likely outcomes using cumulative distribution functions and quantile functions. First, let's look at the cumulative distribution function to see where we have something like 25% of the probability accumulate on the left side of the distribution.

```
cumul.prob <- pbinom(poss.outcomes, size = num.frogs.sampled, p = prob.inf)
plot(inf.prevalence, cumul.prob, pch = 19, xlab = 'Infection Prevalence in a Sample',
     ylab = 'Cumulative Probability')
```



It looks 25% of the probability occurs around 0.5 infection prevalence. But we don't have to guess. We can find exactly by finding the 25% quantile using the quantile function for the binomial distribution.

```
qbinom(0.25, size = num.frogs.sampled, prob = prob.inf)
```

```
## [1] 16
```

This says that 16/30 of the frogs (≈ 0.53 prevalence) or fewer represent 25% of the outcomes. The next 25% of the least likely outcomes occurn at the very high end of the distribution. We can find this by looking at the cutoff where 75% of the cumulative density is. Any outcomes above that represent the other 25% of least likely outcomes.

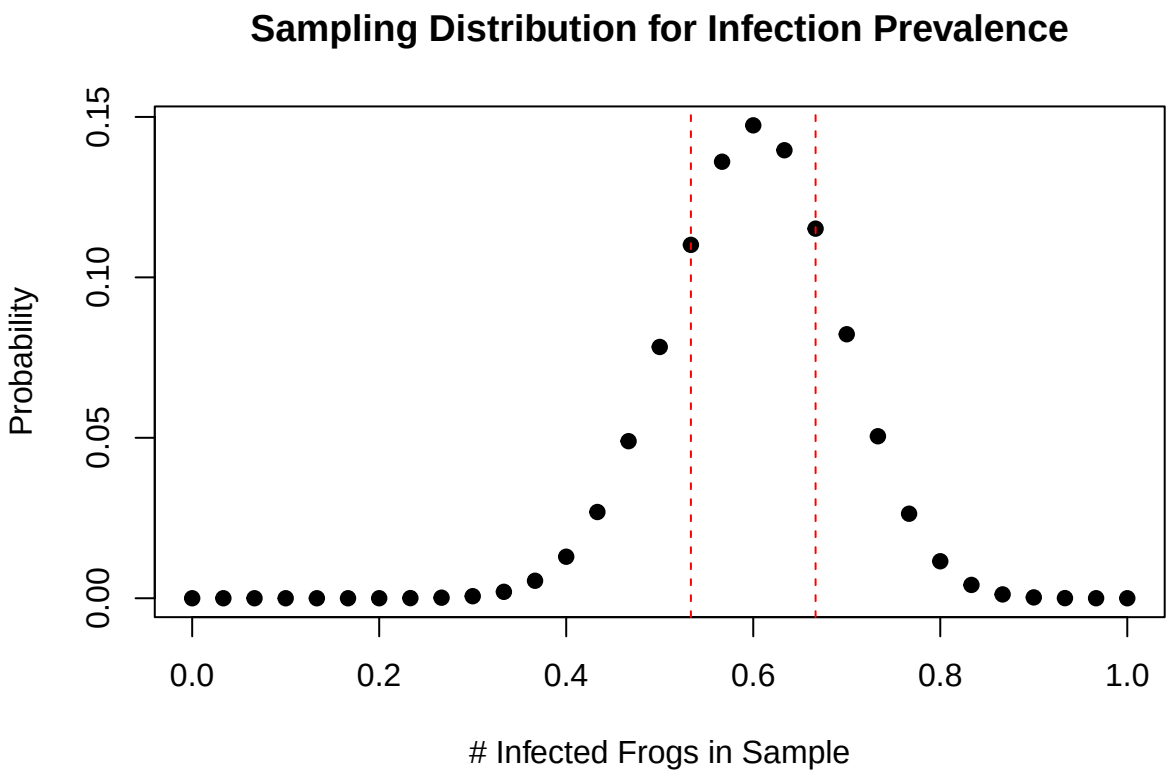
```
qbinom(0.75, size = num.frogs.sampled, prob = prob.inf)
```

```
## [1] 20
```

This says that finding more than 20 infected frogs in our sample of 30 (0.66 prevalence) also has about a 25% chance of happening. Thus, we know that 50% of the most likely outcomes are the values above 16 and below 21 (i.e., 16-20). That says that 50% of our prevalence estimates should be in the range 16/30 (prevalence of 53%) to 20/30 (prevalence of 66%). Let's put these boundaries on the probability plot.

To do these boundaries, we will use the function `abline`. The function `abline` adds a line to a pre-existing plot. You can add a horizontal line using `abline(h = )`, you can add a vertical line using `abline(v = )`, or you can add any straight line by using `abline(a = , b = )` where `a` is the intercept of a line and `b` is the slope. We want to add vertical lines. Let's make them dashed. To do that, we add `typ = 2`. Let's also make the lines red. To do that, we add `col = 'red'`.

```
plot(inf.prevalence, outcomes.probs, pch = 19,
     xlab = '# Infected Frogs in Sample', ylab = 'Probability',
     main = 'Sampling Distribution for Infection Prevalence')
abline(v = c(16,20)/num.frogs.sampled,
       lty = 2,
       col = 'red')
```



Now let's do the same with 95% of the most likely outcomes. Let's first get rid of 2.5% of the least likely to the left. These occur below

```
qbinom(0.025, size = num.frogs.sampled, prob = prob.inf)
```

```
## [1] 13
```

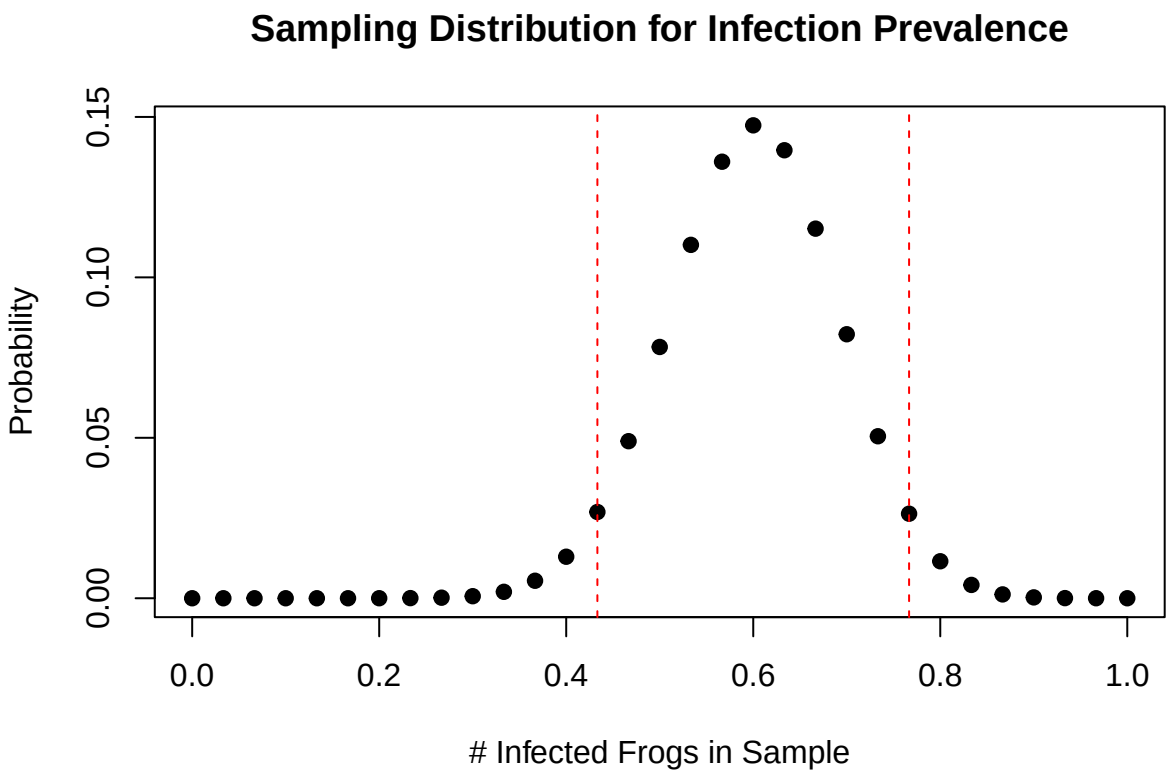
And now let's find the cutoff for the 2.5% least likely high prevalence samples. They are

```
qbinom(0.975, size = num.frogs.sampled, prob = prob.inf)
```

```
## [1] 23
```

This says 95% of the most likely samples we get have anywhere between 13 and 23 infected frogs out of 30. This means prevalence is in the range 43% -76%. Let's plot this one.

```
plot(inf.prevalence, outcomes.probs, pch = 19,
     xlab = '# Infected Frogs in Sample', ylab = 'Probability',
     main = 'Sampling Distribution for Infection Prevalence')
abline(v = c(13,23)/num.frogs.sampled,
       lty = 2,
       col = 'red')
```



Checkpoint 2: Find the region of 60% of the most likely prevalence values for a sample of 100 frogs. Plot the boundaries on your probability distribution from checkpoint 1.

Checkpoint 3: Now do the same for the region of 95% of the most likely sample prevalences.

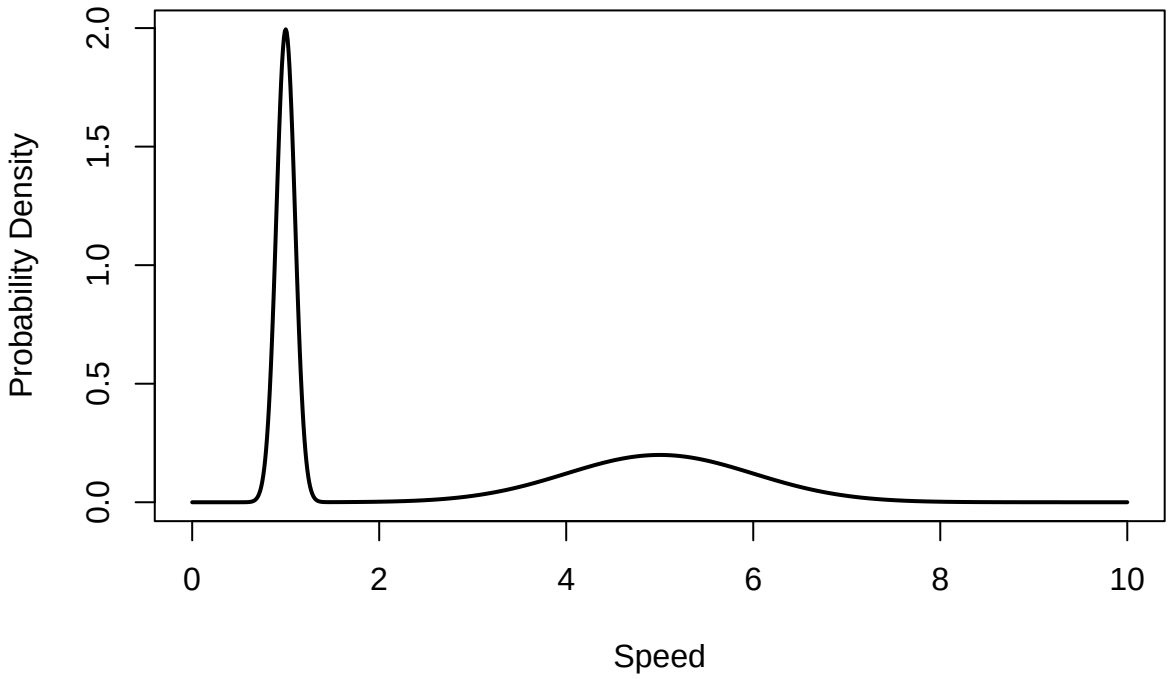
Other Sampling Distributions

We have talked a lot about the binomial, but another distribution turns out to be very important for modeling sampling distributions. That distribution is the normal distribution. The normal distribution turns out to be a good approximation for the sampling distribution of all kinds of sample parameters. But the most important is the mean.

Mathematicians have proved that the normal distribution is a good sampling distribution for the mean. This has been proved by a theorem called The Central Limit Theorem. The Central Limit Theorem (CLT) says that if you add up enough random variables, you eventually get a normal distribution of their sum, regardless of what the probability distribution is for every individual. This is exactly what we do when we calculate a mean. We add up a bunch of individual properties before dividing by the number of individuals in our sample. So the CLT applies to the mean (it turns out it applies to many other sample properties as well).

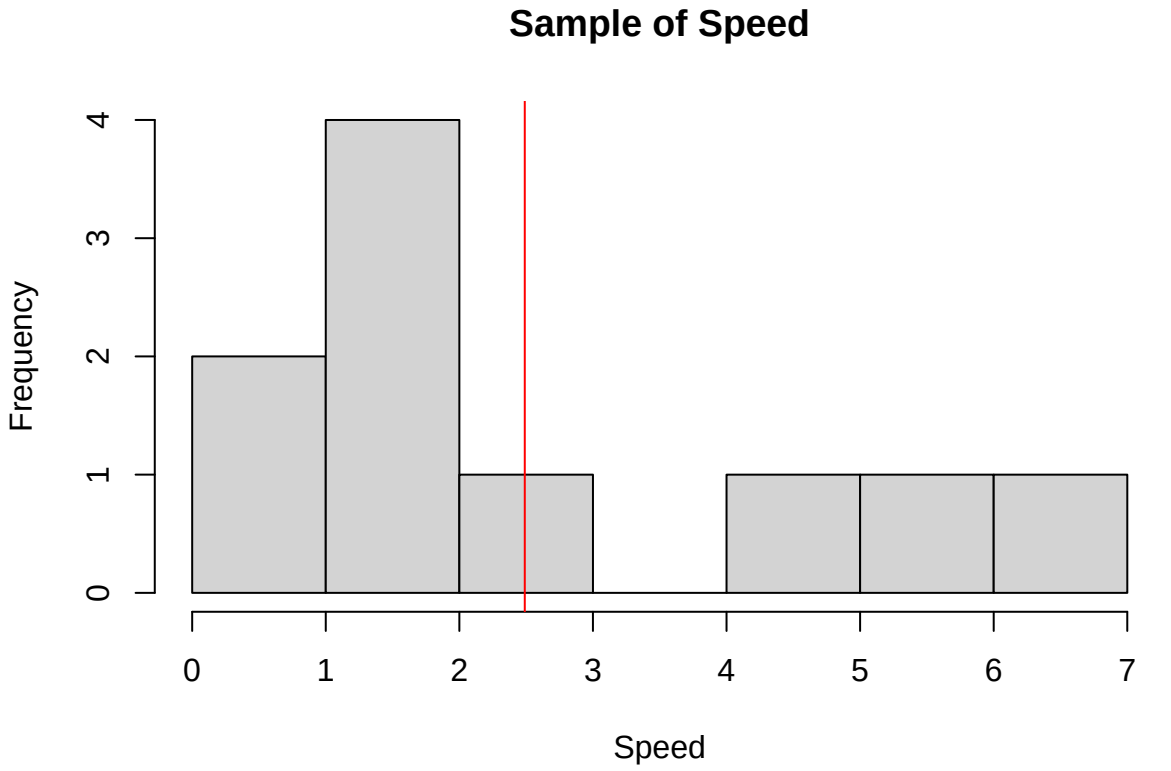
Rather than go through that proof, let's use R to check to see if it works. Let's create some wacky distribution that is very non-normal, sample from it a bunch of times and see how the distribution of the mean looks.

Here is an example with some positive individual value, such as speed. (I've suppressed the code here. It's a bit too complicated at this point in the course. This section is just to illustrate the CLT.)



Most individuals are very slow. Some go kind of fast. This distribution clearly doesn't look like a normal distribution. It's pretty odd. But it turns out the average speed of this distribution is 3.

Let's sample this population and calculate its mean. First, let's sample 10 individuals.

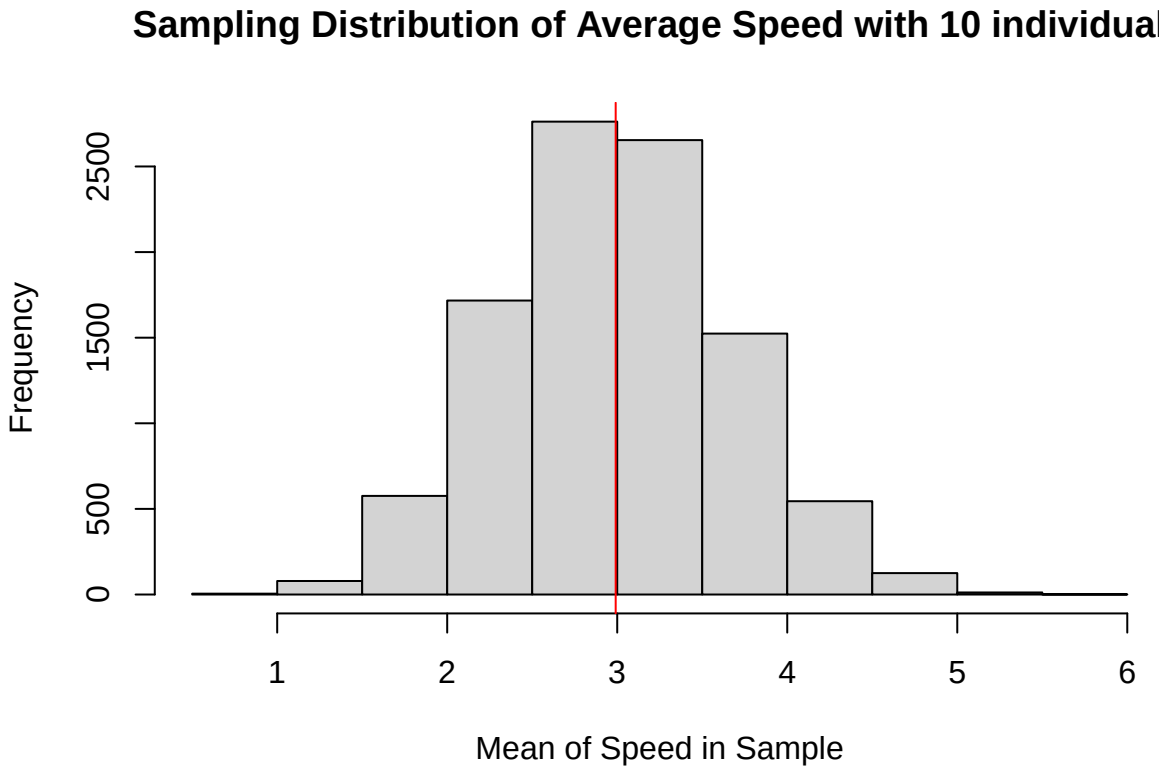



```
## [1] 2.489758
```

So the average here is about 2.5. That's not 3, the actual mean of the distribution, but it is somewhat close. But this is just one sample. Would we always be this close to 3? It's hard to tell. That is what sampling

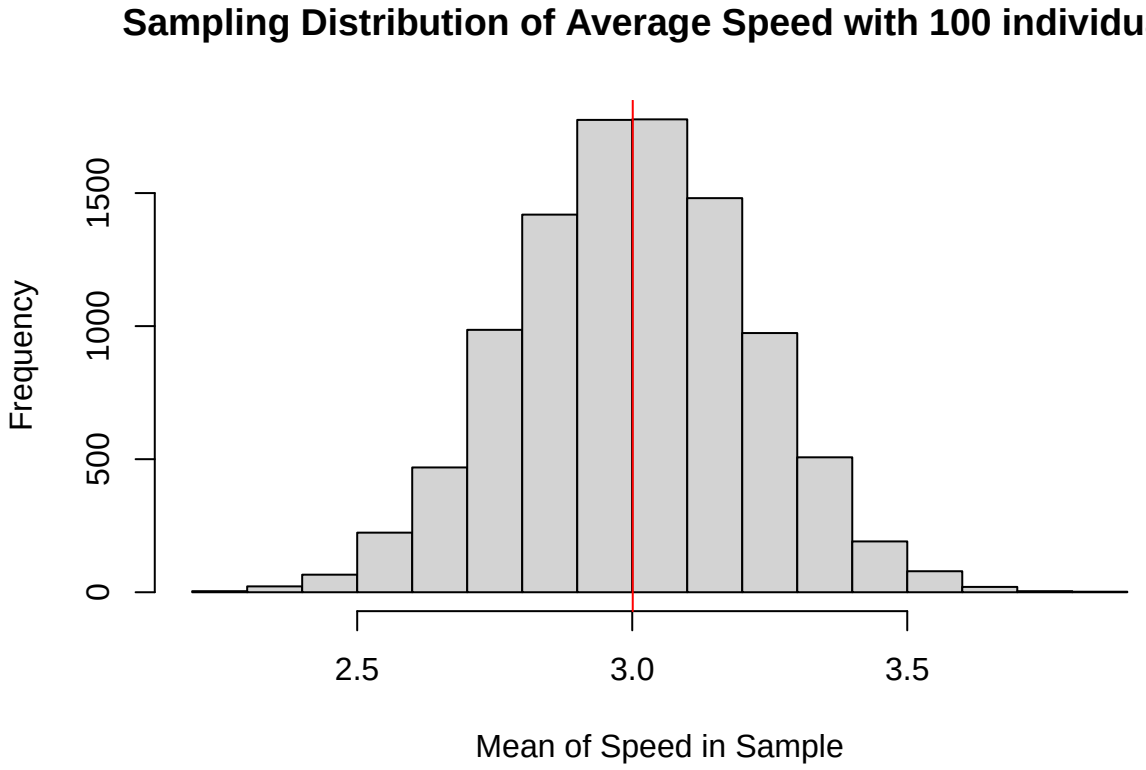
distributions are for. They tell us how likely we are to see any property in our sample.

To get a sense of this sampling distribution, let's repeat our sample again 10,000 times, calculate the mean every time, and see what the distribution of sample means is like (this is the computer way of finding the sampling distribution just like the binomial, but in the case of the binomial we can use math and probability trees to figure it out).

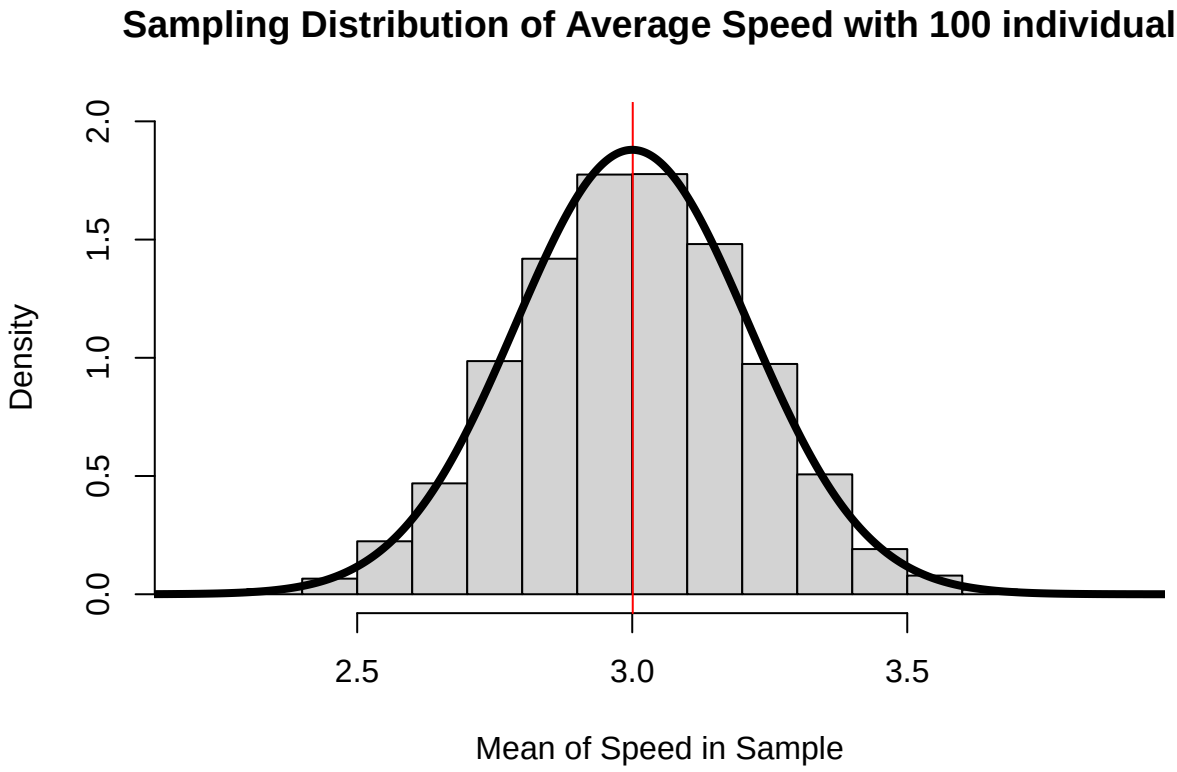


Look at that. It says that most of the time we will estimate a mean between 2 and 4 and the highest probability is right at 3! Pretty cool. Moreover, this sampling distribution of the mean looks an awful lot like a normal distribution, **even though the distribution at the individual level was quite odd and very non-normal.**

Let's do the entire thing again but envision we sample 100 individuals this time and calculate their mean. What does the sampling distribution of the mean look like?



Checkpoint 4: How does this distribution differ from the distribution when we only sampled 10 individuals? The Central Limit Theorem tells us exactly what this distribution should look like. It should be a normal distribution with a mean given by the mean of the individual distribution and a standard deviation given by the standard deviation of the individual distribution divided by the square root of population size. Here is the what the Central Limit Theorem tells us the sampling distribution looks like.



Hey, it matches really well the sampling distribution that we simulated.

This is great because it means that we can use the properties of the normal distribution to find the most likely outcomes. We just need to know the what the properties of the normal distribution are.

The most important thing is the sampling distribution of the mean can almost always be written as a normal distribution.

Standard Errors

One of the most important features of sampling distributions is the variability in estimates from one sample to the next. How do we characterize that variabilty? Well, we measure the variance or standard deviation of the sampling distribution. This feature of sampling distributions is so important that we give it a special name: a standard error.

The standard error is the standard deviation of a sampling distribution. Standard errors relate to any sampling distribution. It could be for the binomial or it could be for the normal, or any other sampling distribution we know about.

The variability in our calculation of means from different samples can be measured by the **standard error of the mean,** which is the standard deviation of the sampling distribution. It has the following expression

$$SE_{\bar{X}} = \frac{\sigma}{\sqrt{n}}.$$

We typically represent the standard error of the mean with the symbol $SE_{\bar{X}}$ where the $\bar{X}$ is the notation for a mean. It has two components: 1) σ, the standard deviation of the random variable applying to an individual, and 2) n the number of individuals in the sample.

This equation says that an estimate of the mean is more variable from sample to sample if

1. the population itself is quite variable (i.e., σ is large), or

2. we sample very few individuals (i.e., n is small).

Here is an example. Let's assume we are catching fish with mean size = 100 cm and standard deviation in fish size in the population of sd = 10cm. If we catch 20 fish and calculate their mean size, how variable is the estimate of mean fish size from sample to sample? To get a sense, let's calculate $SE_{\bar{X}}$.

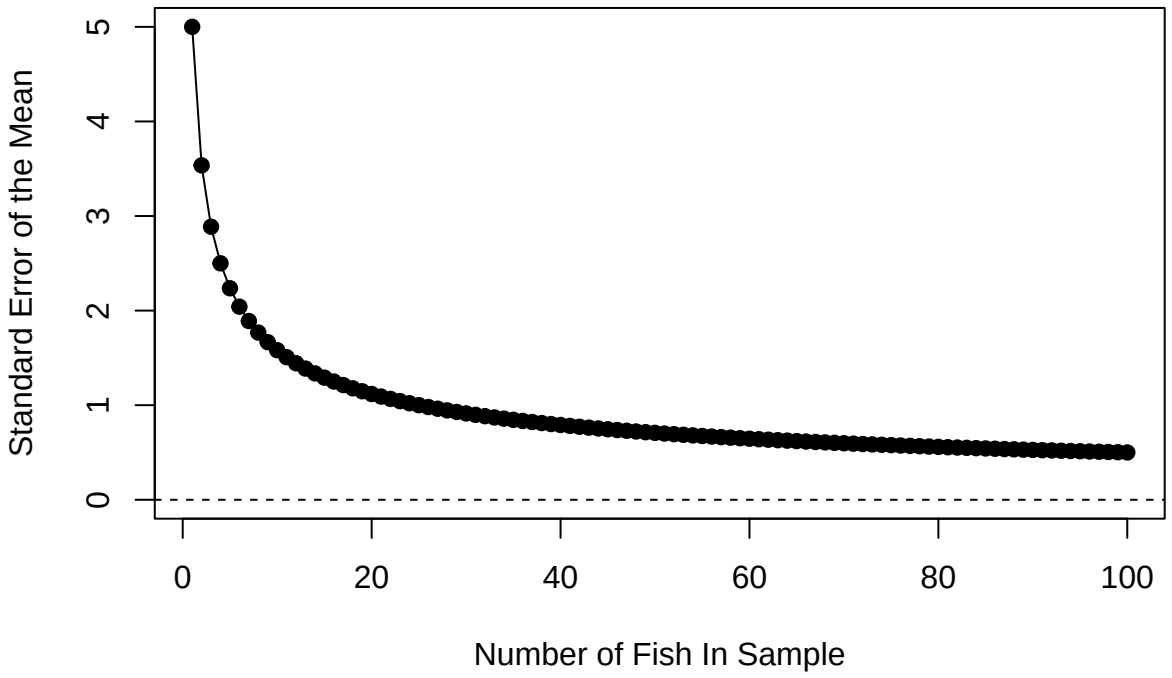
```
fish.size.sd <- 10
sample.size <- 20

SE.mean.fish.size <- fish.size.sd/sqrt(sample.size)
SE.mean.fish.size
```

```
## [1] 2.236068
```

This means that we can expect the mean fish size of a sample of 20 fish to change by about 2.24 cm on average.

Checkpoint 5: What is $SE_{\bar{X}}$ when we sample 30 fish, 50 fish, and 100 fish? One way to view the effect of sample size on the variability in the mean from different samples is to plot how the variability changes as a function of sample size. Let's do this assuming our fish population had a standard deviation of 5cm in fish length, rather than 10 cm.

```
# Let's pick a bunch of sample sizes.
sample.sizes <- seq(from = 1, to = 100, by = 1)
# State the variability in the population in fish length
sd.fish.length <- 5
# Now calculate the SE of the mean for every sample size
SE.mean <- sd.fish.length/sqrt(sample.sizes)

# Let's take a look
plot(sample.sizes, SE.mean, typ = 'o', pch = 19,
     xlab = 'Number of Fish In Sample', ylab = 'Standard Error of the Mean',
     ylim = c(0,5))
abline(h = 0, lty = 2)
```



This figure shows the uncertainty in the estimate of the mean as a function of sample size. It shows something familiar: We are more certain in our estimate of the mean when we have larger samples because the variability in the mean across samples declines with the sample size. However, not all increases in sample sizes have the same effect. For example, going from 2 fish to 20 fish has a huge effect, but going from 20 fish in the sample to 40 fish doesn't have near the same effect.

Confidence Intervals of the Mean

We can use standard errors to create confidence intervals. We'll talk about the theory of how this is done more in class, but for now, let's work on calculating some confidence intervals ourselves.

Confidence intervals give the range of reasonable values where we expect the actual population parameter to be given the sample characteristics we have measured.

Let's take a look at some examples with some data.

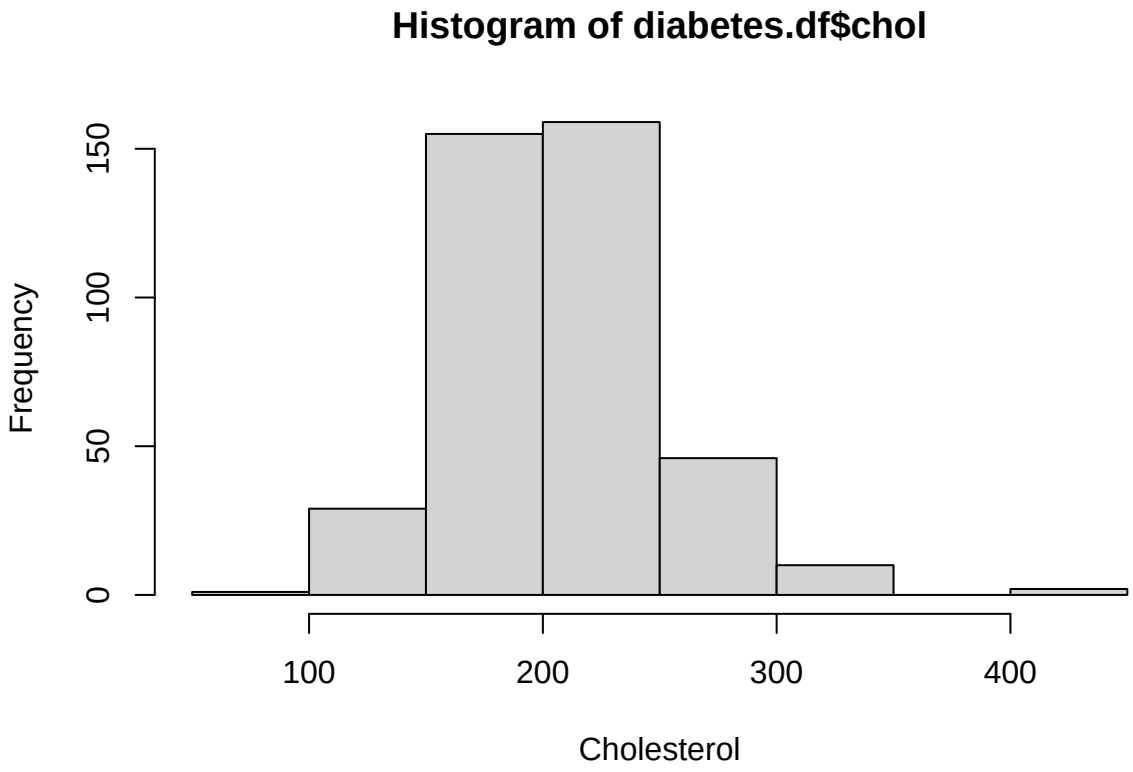
Go to the course canvas page and download the data "diabetes.csv". General information about where the data come from and the study associated with the data is found here and information about each of the variables is found here. Once you download the data, load it into R and name it `diabetes.df`.

```
str(diabetes.df)
```

```
## 'data.frame':    403 obs. of  19 variables:
##  $ id      : int  1000 1001 1002 1003 1005 1008 1011 1015 1016 1022 ...
##  $ chol    : int  203 165 228 78 249 248 195 227 177 263 ...
##  $ stab.glu: int  82 97 92 93 90 94 92 75 87 89 ...
##  $ hdl     : int  56 24 37 12 28 69 41 44 49 40 ...
##  $ ratio   : num  3.6 6.9 6.2 6.5 8.9 ...
##  $ glyhb   : num  4.31 4.44 4.64 4.63 7.72 ...
##  $ location: chr  "Buckingham" "Buckingham" "Buckingham" "Buckingham" ...
##  $ age     : int  46 29 58 67 64 34 30 37 45 55 ...
##  $ gender  : chr  "female" "female" "female" "male" ...
##  $ height  : int  62 64 61 67 68 71 69 59 69 63 ...
##  $ weight  : int  121 218 256 119 183 190 191 170 166 202 ...
##  $ frame   : chr  "medium" "large" "large" "large" ...
##  $ bp.1s   : int  118 112 190 110 138 132 161 NA 160 108 ...
##  $ bp.1d   : int  59 68 92 50 80 86 112 NA 80 72 ...
##  $ bp.2s   : int  NA NA 185 NA NA NA 161 NA 128 NA ...
##  $ bp.2d   : int  NA NA 92 NA NA NA 112 NA 86 NA ...
##  $ waist   : int  29 46 49 33 44 36 46 34 34 45 ...
##  $ hip     : int  38 48 57 38 41 42 49 39 40 50 ...
##  $ time.ppn: int  720 360 180 480 300 195 720 1020 300 240 ...
```

Let's assume that individuals in this data set are a random sample and we want to do something like estimate cholesterol levels in this population. The cholesterol distribution in this particular sample is

```
hist(diabetes.df$chol, xlab = 'Cholesterol')
```




```
summary(diabetes.df$chol)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    78.0   179.0   204.0   207.8   230.0   443.0       1
```

Before going any further, notice that there is an NA in this dataset. NA means that there is no data about cholesterol for one of the individuals. Dealing with missing data is a part of doing statistics because most data sets are missing pieces of information. For now, let's get rid of this individual. If we don't the NA will cause lots of problems. First, we need to find the individual by finding the row that corresponds to an NA in the cholesterol variable and take all other rows. We can do this with `subset`.

```
new.diabetes.df <- subset(diabetes.df, subset = !(is.na(chol)))
summary(new.diabetes.df$chol)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    78.0   179.0   204.0   207.8   230.0   443.0
```

Perfect. None of the summary statistics changed. The only thing that changes is there is no longer an NA listed.

Checkpoint 6: How does the subset command `!is.na(chol)` work? Looks like the mean from this sample is 207.8. Is this the average cholesterol value in the population? Probably not. Is it close? Maybe? How do we make a range of reasonable values where the mean actually is? We use confidence intervals. To make confidence intervals, we need standard errors. Let's calculate the standard error by first calculating the standard deviation in the data and the sample size of the data.

```
chol.mean <- mean(new.diabetes.df$chol)
chol.n <- length(new.diabetes.df$chol)
```

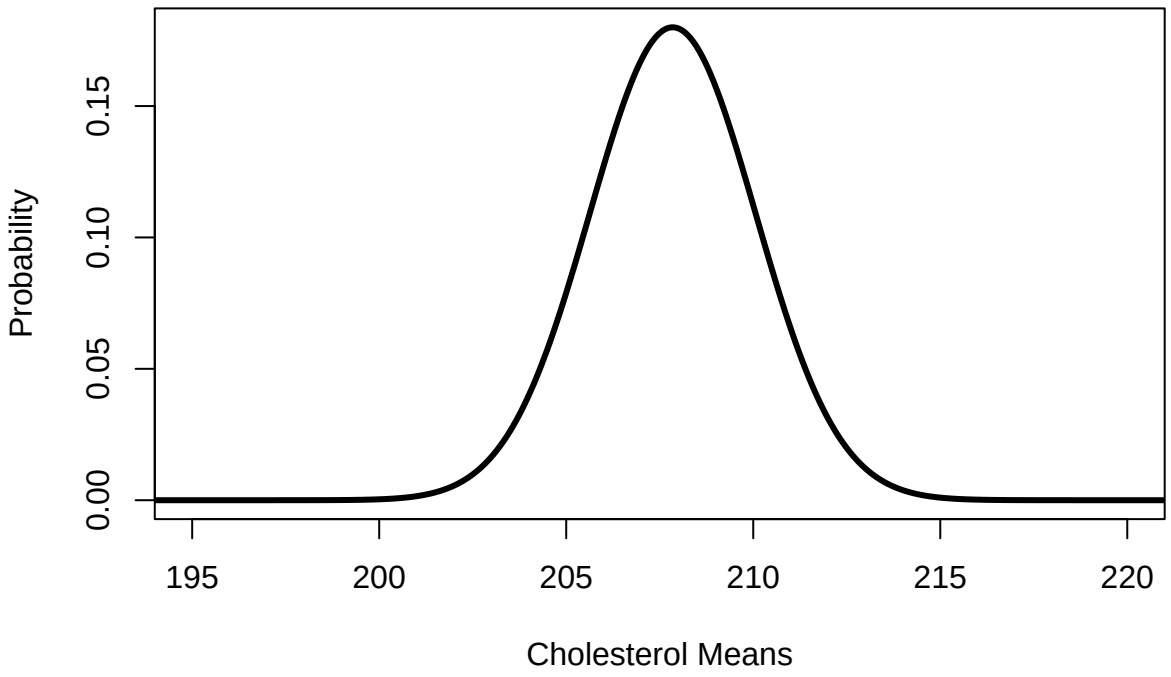
Now we can use these together to get the standard error using the equation from above.

```
(chol.se <- sd(diabetes.df$chol, na.rm = T)/sqrt(chol.n))
```

```
## [1] 2.216743
```

Now that we have the standard error, we can approximate the sampling distribution with a normal distribution. Let's do that and plot it.

```
# Create a bunch of possible means we could calcualte
possible.mean.chol <- seq(from = 150, to = 250, length = 10000)
# Find their associated probabilities of occurrence
poss.mean.prob <- dnorm(possible.mean.chol, mean = chol.mean, sd = chol.se)

# And plot them
plot(possible.mean.chol, poss.mean.prob, typ = 'l',
     xlab = 'Cholesterol Means', lwd = 3,
     ylab = 'Probability', xlim = c(195,220))
```



Now that we have the sampling distribution for mean cholesterol, we can create a confidence interval by looking for the most likely outcomes. Let's find the 95% of the most likely outcomes from this distribution by using the quantile function.

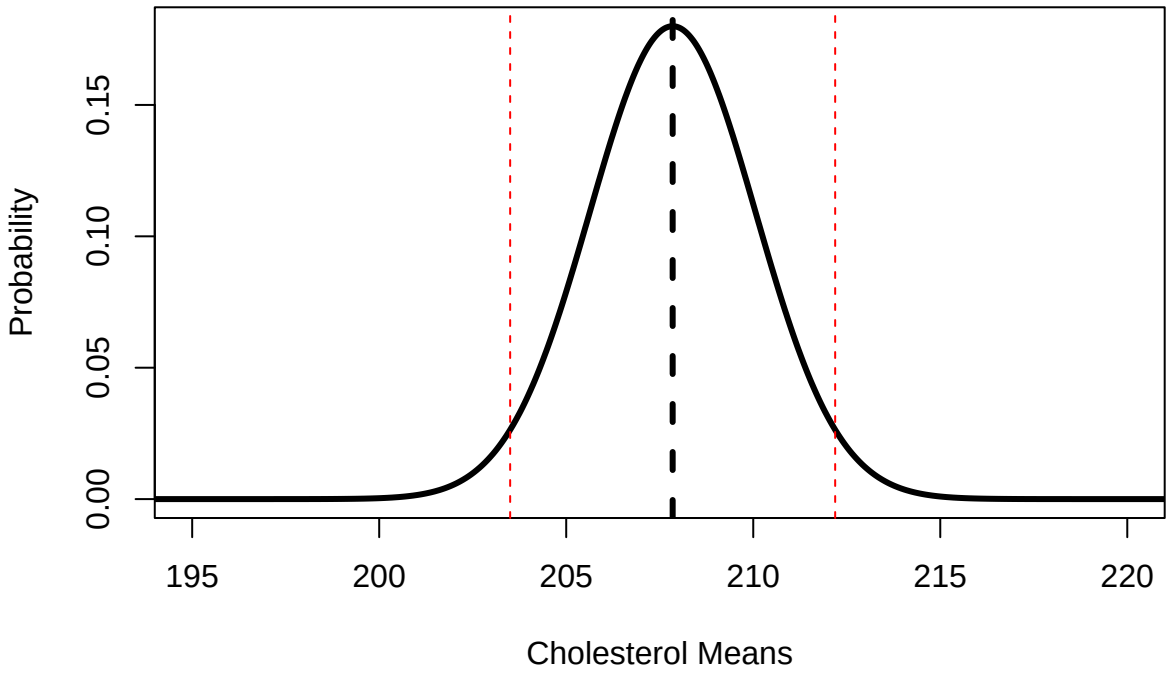
```
sampl.quantiles <- qnorm(c(0.025, 0.975), # Ask for the 2.5% and 97.5% quantile
      mean = chol.mean, # Specify the mean of the sampling distribution
      sd = chol.se) # Specify the standard deviation of the sampling dist.
                    # This is the standard error.
sampl.quantiles
```

```
## [1] 203.5010 212.1905
```

Let's put these on the plot as vertical lines on the plot and we will also add the mean we estimated.

```
plot(possible.mean.chol, poss.mean.prob, typ = 'l',
     xlab = 'Cholesterol Means', lwd = 3,
     ylab = 'Probability', xlim = c(195,220))
abline(v = sampl.quantiles, col = 'red', lty = 2)
```

```
abline(v = chol.mean, col = 'black', lty = 2, lwd = 3)
```



This here gives us our best estimate of the mean and a range of possible values where the actual mean of the population lies. This is called a **95% confidence interval** because it 95% of the samples (assuming random sampling), the actual population mean is in this range. Likewise, this means that 5% of the time, we collect a sample where the mean is NOT in the 95% confidence interval.

Checkpoint 7: Find the 90% confidence interval for the mean of cholesterol in this population.

Calculating Confidence Intervals in R

This is the way to do it by hand, but R has a way to do it much more efficiently. To do it more efficiently, we can use a structure in R called a **linear model**. We will learn more about linear models later, but they include everything from t-tests to ANOVA to linear regression. Statisticians have recognized that these are all variants of the same underlying model, which we now call a linear model. We will use this to estimate confidence intervals.

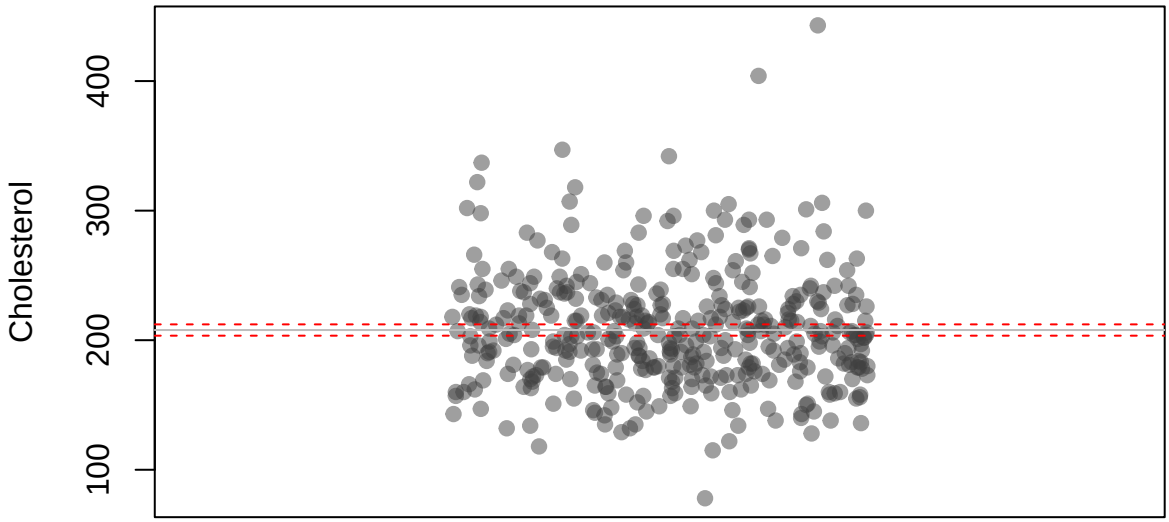
The way to run a linear model is to use the function `lm`, which stands for "linear model." We put in there formula notation on what to estimate and then we add in the data frame. The formula notation we will use is `chol ~ 1` which is the way of writing that we want to estimate the mean of the variable cholesterol. We will talk more about formula notation when we get to linear models, but the basic idea is that we want to write how the response variable depends on other factors in the data structure. Since we don't care about any dependencies of cholesterol on anything else, we just write that it is a function of a constant number, 1. The only other option we need to include is the `level`, which tells us the level of the confidence interval. If we want the 95% confidence interval, we set `level = 0.95`. If we want the 99% confidence interval, we set `level = 0.99`.

```
our.model <- lm(chol ~ 1, data = new.diabetes.df)
confint(our.model, level = 0.95)
```

```
##                 2.5 %   97.5 %
## (Intercept) 203.4879 212.2037
```

This is effectively the same number we calculated above by writing out the sampling distribution. Here is

what that looks like in visual form. The graph below has the individual data points, the mean (the gray line) and the 95% confidence interval (the red dashed lines).




```
confint(our.model, level = 0.99)
```

Checkpoint 8: Find the 99% confidence interval for cholesterol. Is it larger or smaller than the 95% confidence interval?

```
##                 0.5 %  99.5 %
## (Intercept) 202.1085 213.583
```

```
# Larger
```

Checkpoint 9: Find the 90%, the 95%, and the 99% for height in this dataset.

```
height.mdl <- lm(height~1, data = diabetes.df)
confint(height.mdl, level = 0.9)
```

```
##                   5 %     95 %
## (Intercept) 65.69627 66.34393
```

```
confint(height.mdl, level = 0.95)
```

```
##                 2.5 %   97.5 %
## (Intercept) 65.63395 66.40625
```

```
confint(height.mdl, level = 0.99)
```

```
##                 0.5 %   99.5 %
## (Intercept) 65.51172 66.52848
```