Confidence Intervals

[Put your name here]

In this assignment we will learn about the sampling distribution model for proportions. This will allow us to define the standard error and construct confidence intervals.

Instructions

Presumably, you have already created a new project and downloaded this file into it. Please knit the document (probably to HTML as you're working) and work back and forth between this R Markdown file and the knit output to answer the following questions.

When you are finished with the assignment, knit to PDF, export the PDF file to your computer, and then submit to the corresponding assignment in Canvas. Be sure to submit before the deadline!

Sometimes you will be asked to add your own R code. That will appear in this document as a code chunk with a request for you to add your own code, like so:

```
## Add code here to [do some task]...
```

When you see that in a code chunk, you need to type some R code to complete a task.

Sometimes you will be asked to type up your thoughts. Instructions to do that will be labeled as follows. If you are currently reading this in the knit output, please look back at the R Markdown file to see the following text:

(When you knit the document, you can't see the text from the line above. That's because the crazy notation surrounding that text marks it as a "comment", and therefore it doesn't appear in the output.) In these areas of the assignment, please use contextually meaningful full sentences/paragraphs (unless otherwise indicated) and proper spelling, grammar, punctuation, etc. This is not R code (although it may include inline R code when necessary), but rather a free response section where you talk about your analysis and conclusions.

Getting started

Make sure you're in a project

If you're looking at this document, you should have already created a project and uploaded this R Markdown file to that project folder.

Save your file!

The first thing we **always** do is save our file. You'll probably want to save this under a new name. Go to the "File" menu and then "Save As". Once you've saved the file with the new name, from them on it's easier to just hit Ctrl-S (or Cmd-S on a Mac) to keep saving it periodically.

Remember that file names should not have any spaces in them. (In fact, you should avoid other kinds of special characters as well, like periods, commas, number signs, etc. Stick to letters and numerals and you should be just fine.) If you want a multiword file name, I recommend using underscores like this: `this_filename_has_spaces_in_it`.

Load Packages

We load the standard `mosaic` package. We'll also need the `openintro` package later in the assignment for the `hsb2` data set.

```
library(mosaic)
library(openintro)
```

Since we'll be working with random numbers, let also set the seed so that every time you knit this document, you'll always be looking at the same set of numbers.

```
set.seed(13579)
```

Sampling distribution models

We know that when we sample from a population, our sample is "wrong". Even when the sample is representative of the population, we don't actually expect our sample statistic to agree exactly with the population parameter of interest. Our prior simulations have demonstrated this: they are centered on the "true" value (in our case, the "true" value was the assumed null value), but there is some spread due to sampling variability.

Let's explore this idea a little further, this time considering how sample size plays a role in sampling variability.

First, let's learn how to sample a single proportion in R. (Our earlier simulations focused on the difference between two proportions, not a single proportion.)

Suppose that a certain candidate in an election actually has 64% of the support of registered voters. We conduct a poll of 10 people, gathering a representative (though not very large) sample of voters.

The command to do this in R is `rbinom`:

```
rbinom(n = 1, size = 10, prob = 0.64)
```

```
## [1] 8
```

You can think of the above calculation as taking 1 random sample of size 10 and getting a certain number of "successes" (where a "success" here is a person who votes for our candidate). In other words, in this particular sample, we surveyed 8 people who said they were voting for our candidate and 2 people who were not.

Be careful: normally we consider `n` to be the sample size, but here `n` is the number of samples we collect, whereas `size` is the sample size.

If we change the value of `n`, we can simulate many samples, all of size 10. Let's take 1000 and store them in a variable called `sims10`.

```
sims10 <- rbinom(n = 1000, size = 10, prob = 0.64)
head(sims10, n = 20)
```

```
##  [1] 5 8 4 7 7 9 3 5 8 3 5 5 6 6 5 8 6 7 8 5
```

Note that with 10 people, it is impossible to get a 64% success rate in our sample. (That would be 6.4 people!) Nevertheless, you can see that many of the samples gave us around 5–8 successes, as we'd expect if the true population rate is 64%. Also, the mean number of successes across all simulations is 6.456, which is close to 6.4.

Let's do this again, but instead of focusing on the total number of successes, let's use proportions. To get each number to be a percentage, we simply divide the whole expression by 10, the sample size.

```
sims10 <- rbinom(n = 1000, size = 10, prob = 0.64)/10
head(sims10, n = 20)
```

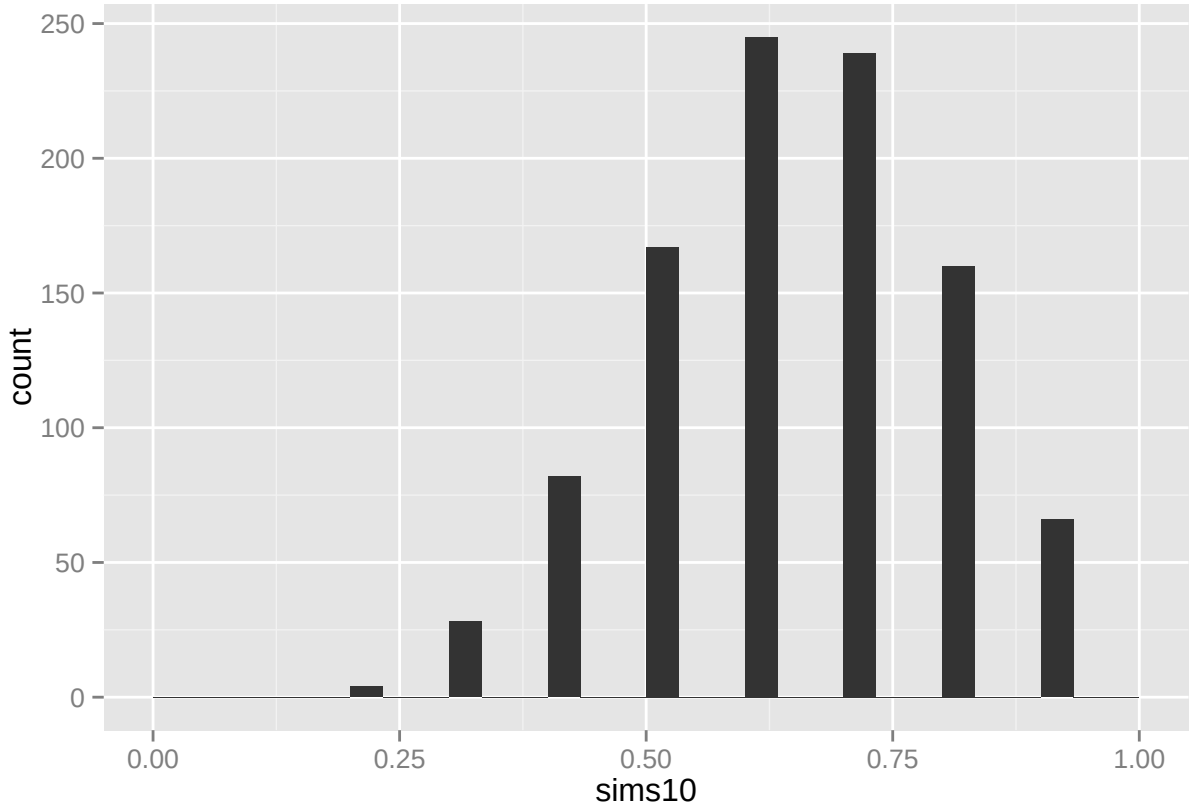
```
##  [1] 0.6 0.8 0.7 0.4 0.7 0.8 0.7 0.4 0.6 0.4 0.4 0.4 0.6 0.6 0.7 0.8 0.5
## [18] 0.8 0.7 0.7
```

(This command generated a whole new random simulation, so these proportions are not meant to correspond with the earlier simulated values.)

Let's graph our simulated values and look at the result over an x-axis that spans the whole range of possible proportions from 0 to 1.

```
qplot(sims10,  xlim = c(0, 1))
```

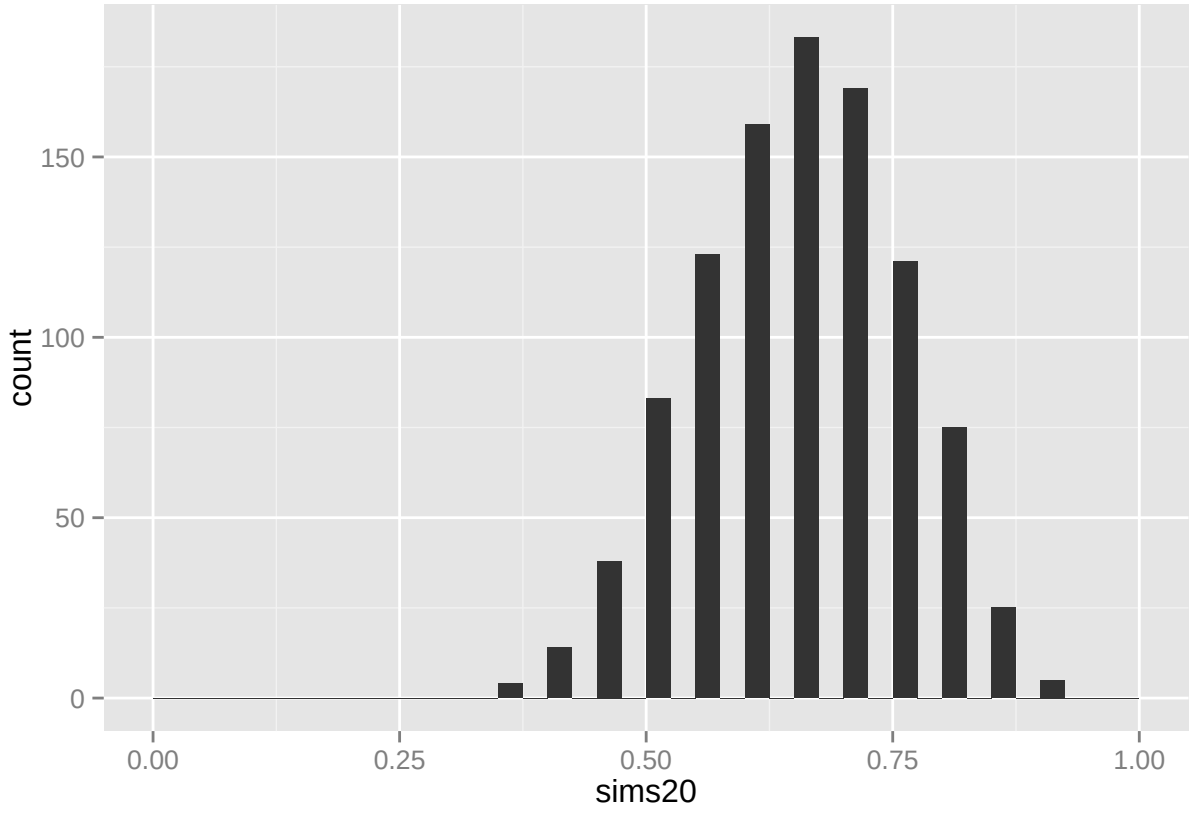
```
## stat_bin: binwidth defaulted to range/30. Use 'binwidth = x' to adjust this.
```



I won't change the bin sizes here; this picture is already an accurate depiction of the fact that the sample proportions can only be multiples of 0.1. Also note that although the distribution is more or less normally shaped, it is discrete (no values in between the bars) and there is an appreciable left skew.

What happens if I increase the sample size to 20? (I will have to change the default binwidth here to see the discrete bars.)

```
sims20 <- rbinom(n = 1000, size = 20, prob = 0.64)/20
qplot(sims20,  xlim = c(0, 1), binwidth = 0.025)
```



Question: Explain how the distribution of simulations has changed going from a sample size of 10 to a sample size of 20.

Now you try with samples of size 50. Below the plot, explain how the distribution of simulations has changed going from a sample size of 10 to 20 to 50.

```
## Add code here to simulate 1000 randon samples of size 50 and plot them.
```

The Central Limit Theorem tells us that as our sample size increases, the distribution of sample proportions looks more and more like a normal model. This model is called the *sampling distibution model* because it describes how many different samples from a population should be distributed.

The question now is, which normal model? In other words, what is the mean and standard deviation of a normal model that describes a simulation of repeated samples?

The simulations above are all centered at the same place, 0.64. This is no surprise. If the true population proportion is 0.64, then we expect most of our samples to be around 64% (even if, as above, it is actually impossible to get exactly 64% in any given sample).

But what about the standard deviation? It seems to be changing with each sample.

Question: How does the standard deviation change as the sample size increases? Intuitively, why do you think this happens?

The standard deviation of a sampling distribution is usually called the *standard error*. (The use of the word "error" in statistics does not mean that anyone made a mistake. A better word for error would be "uncertainty" or even just "variability".)

There is some complicated mathematics involved in figuring out the standard error, so I'll just tell you what it is. If p is the true population proportion, then the standard error is

$$\sqrt{\frac{p(1-p)}{n}}.$$

In other words, if the sample size is large enough, the sampling distribution model is nearly normal, and the correct normal model is

$$N\left(p, \sqrt{\frac{p(1-p)}{n}}\right).$$

Conditions

Like anything in statistics, there are conditions that have to be met before applying any technique. When we want to use a normal model, we have to make sure the sampling distribution model is truly normal (or nearly normal).

As always, we need our samples to be random and less than 10% of the population. (These are conditions that help ensure that the mathematical assumption of independence is met. Of course, in real life hardly any samples will be truly random, so being representative is the most we can hope for.)

There is now one more condition. It is called the "success/failure" condition. We need for the total number of successes in our sample to be at least 10 and, similarly, for the total number of failures in our sample to be at least 10.

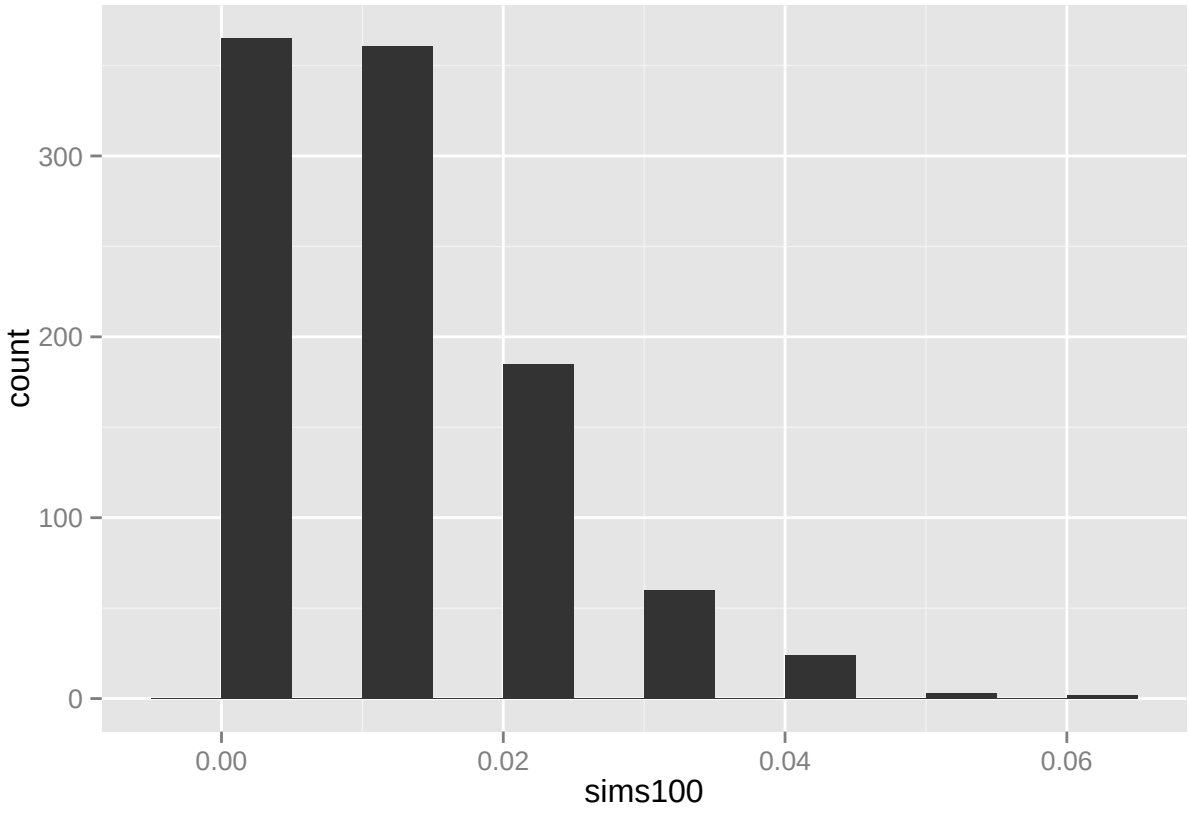
Go back and consider our first simulated sample. We obtained 8 of 10 successes. 8 successes and 2 failures are nowhere near big enough. In fact, since the sample size was 10, there was no way that any of the simulated samples could meet this condition. When we plotted the histogram of simulated proportions, we saw the problem. With such small numbers, the histogram was skewed, and not normal.

We check the success/failure condition by calculating np and $n(1-p)$. If n is the sample size, and p is the proportion of successes, then np is the total number of successes. Since $1-p$ is the proportion of failures, then $n(1-p)$ is the total number of failures. Each of these numbers needs to be bigger than 10.

Notice that when n is large, the quantities np and $n(1-p)$ will also tend to be large. This is the content of the Central Limit Theorem: when sample sizes grow, the sampling distribution model becomes more and more normal.

There is something else going on though. Suppose that $n = 100$ but $p = 0.01$. The sample seems quite large, but let's look at the sampling distribution through a simulation.

```
sims100 <- rbinom(n = 1000, size = 100, prob = 0.01)/100
qplot(sims100, binwidth = 0.005)
```



Question: What's the problem here? Despite having a fairly large sample size, why is this distribution so skewed?

In this scenario, the success/failure condition fails because

$$np = (100)(0.01) = 1 \ngeq 10.$$

In other words, in a typical sample, we expect 1 success and 99 failures.

Confidence intervals

All our simulations above are based on the assumption that we know p, the true population proportion. In reality, we do not know p. In fact, that's the whole point of statistics: we do not know the true population parameters, so we gather a sample. We will measure $\hat{p}$, our sample proportion, and we hope that $\hat{p}$ is a good estimate of p.

Of course, $\hat{p}$ will almost never be exactly the same as p, but now we have a tool for figuring out how close together they should be. The sampling distribution model tells us that when we sample a value of $\hat{p}$, it should be within a few standard errors of p.

We do need to make one adjustment to our standard error. The formula we gave above was

$$\sqrt{\frac{p(1-p)}{n}}.$$

However, in a situation where we have collected data to try to estimate p, clearly we do not know p. So we plug in the estimate from our sample instead and call this SE:

$$SE = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}.$$

Recall that about 95% of a distribution lies within two standard deviations of the mean. Therefore, it should be the case that the true proportion p should lie within two standard errors of $\hat{p}$ most of the time.

This is the idea of a confidence interval. Take $\hat{p}$ from the sample and add/subtract 2 standard errors:

$$(\hat{p} - 2SE, \hat{p} + 2SE)$$

$$= \left(\hat{p} - 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + 2\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right).$$

Question: The number 2—derived from the 68-95-99.7 rule—is, in fact, slightly incorrect. Using your knowledge of normal models and the `qdist` and `qnorm` commands, what is the number of standard errors that would enclose exactly 95% of the middle of the sampling distribution?

```
## Add code here to calculate the exact number of standard errors
## from the mean that enclose the middle 95% of a normal distribution.
```

The interpretation is that when you go collect a sample, the confidence interval you produce using your estimate $\hat{p}$ will capture the true population proportion 95% of the time.

There is no particular reason that we need to compute a 95% confidence interval, although that is the generally agreed-upon standard. We could compute a 90% confidence interval or a 99% confidence interval, or any other type of interval. (Having said that, if you choose other intervals besides these three, people might wonder if you're up to something.)

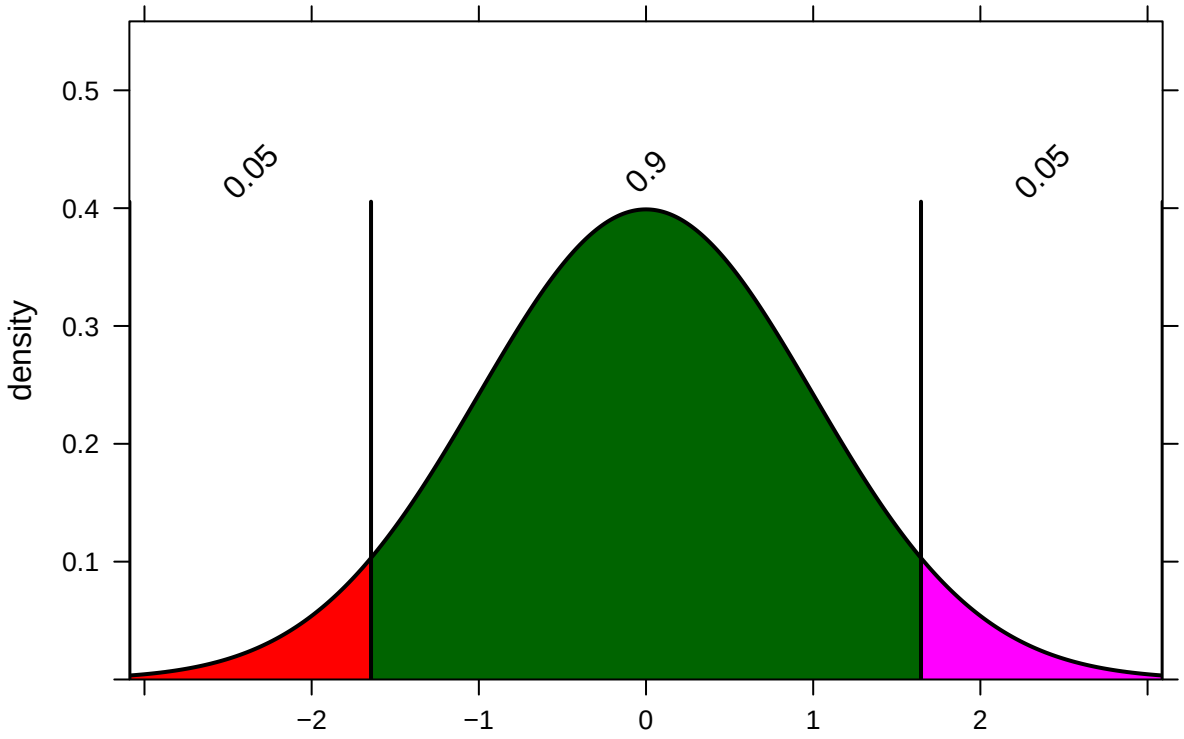
The general formula, then, is

$$\left(\hat{p} - z^*\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}, \hat{p} + z^*\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}\right).$$

The new symbol z^* is called a *critical z-score*. This is the z-score that encloses the confidence level you want.

For example, suppose we wanted a 90% confidence interval. What is the corresponding critical z-score? Well, the middle 90% of a distribution leaves 5% in each tail. Therefore, we need to use `qdist` or `qnorm` with the 5th and 95th percentiles.

```
qdist(dist = "norm", p = c(0.05, 0.95))
```




```
## [1] -1.644854  1.644854
```

The critical z-score is the positive answer, so we could use `qnorm` simply to report 1.6448536. **Be careful!** The critical z-score for a 90% confidence interval requires `p = 0.95` in the `qnorm` function. This is because the upper endpoint of a 90% interval is actually located at the 95th percentile.

(Also, if you had trouble with the question above that asked about the exact critical z-score for a 95% confidence interval, go back and try again now that you've seen another worked example.)

Now do the same thing for a 99% confidence interval. In other words, calculate the critical z-score that encloses the middle 99% of the normal distribution.

Don't forget that there are always conditions to check. Before computing a confidence interval for a proportion, you must verify them.

- Random
  - The sample must be random (or hopefully representative).
- 10%
  - The sample size must be less than 10% of the size of the population.
- Success/failure
  - $np \geq 10$ and $n(1-p) \geq 10$. (No full sentences required.) Keep in mind that if you already have the raw counts of successes and failures, you do not need to compute np and $n(1-p)$ since you already have them! Just make sure both numbers are bigger than 10.

Using R to calculate confidence intervals

There are three forms in which you might have categorical data. They all use the command `binom.test`, but in slightly different ways.

You may just have a summary of the total number of successes and failures. For example, suppose we survey 326 people and 212 of them support a new initiative. (This means that 114 do not support it.) Assuming we have a random sample that is less than 10% of the population, and seeing as we have far more than 10 successes and failures, we get the confidence interval as follows:

```
binom.test(x = 212, n = 326)
```

```
##
##
##
## data:  212 out of 326
## number of successes = 212, number of trials = 326, p-value =
## 6.237e-08
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.5958226 0.7020312
## sample estimates:
## probability of success
##              0.6503067
```

This gives us our confidence interval, but it also gives us a bunch of other stuff we don't need right now. (It's actually doing the "Mechanics" section for a full hypothesis test.)

If we just want to extract the confidence interval, the easiest thing to do is to give the results a name (in this case, `survey`) and then grab the `conf.int` piece directly.

```
survey <- binom.test(x = 212, n = 326)
survey$conf.int
```

```
## [1] 0.5958226 0.7020312
## attr(,"conf.level")
## [1] 0.95
## attr(,"method")
## [1] "Score"
```

One can even break this down further for reporting inline:

We are 95% confident that the true proportion of those who support the new initiative is captured in the interval (59.5822584%, 70.2031208%).

If you are given the percentages of successes and/or failures in your data, you'll have to convert them to whole number totals. For example, if we're told that 326 people were surveyed and 65% of them support the new initiative, then we have to do this:

```
survey2 <- binom.test(x = round(326*0.65), n = 326)
survey2$conf.int
```

```
## [1] 0.5958226 0.7020312
## attr(,"conf.level")
```

```
## [1] 0.95
## attr(,"method")
## [1] "Score"
```

We have to round the number inside this command since 326 times 0.65 is not a whole number. (It is the 65% that is, in fact, rounded, but we have no way of knowing what the exact number of successes was if all we're told is the proportion of successes.)

Finally, it is possible that we have categorical data in a data frame. For example, what percentage of U.S. high school students go to private school? We use the `schtyp` variable in the `hsb2` data set.

First, check the conditions. The sample is presumably a representative sample of high school seniors from the U.S. as the survey was conducted by the National Center of Education Statistics. The sample size is 200, which is much less than 10% of the population of all U.S. high school seniors. Finally, we look at the number of successes and failures:

```
table(hsb2$schtyp)
```

```
##
##  public private
##     168      32
```

There are more than 10 successes and more than 10 failures (where a success is defined here to mean a senior who goes to private school).

Now we are ready to compute. If we do the following, though, we get the wrong answer:

```
schooltype <- binom.test(x = hsb2$schtyp)
schooltype
```

```
##
##
##
## data:  hsb2$schtyp  [with success = public]
## number of successes = 168, number of trials = 200, p-value <
## 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.7817010 0.8879125
## sample estimates:
## probability of success
##                   0.84
```

Examine the output above and see if you can spot the problem.

Try this instead:

```
schooltype2 <- binom.test(x = hsb2$schtyp, success = "private")
schooltype2
```

```
##
##
```

```
##
## data:  hsb2$schtyp  [with success = private]
## number of successes = 32, number of trials = 200, p-value <
## 2.2e-16
## alternative hypothesis: true probability of success is not equal to 0.5
## 95 percent confidence interval:
##  0.1120875 0.2182990
## sample estimates:
## probability of success
##                   0.16
```

We are 95% confident that the true proportion of U.S. high school seniors who attend private school is captured in the interval (11.2087498%, 21.8299015%).

The default confidence level for a confidence interval is almost always 95%. It is possible, however, to use a different level.

```
schooltype3 <- binom.test(x = hsb2$schtyp, success = "private", conf.level = 0.9)
schooltype3$conf.int
```

```
## [1] 0.1188649 0.2088852
## attr(,"conf.level")
## [1] 0.9
## attr(,"method")
## [1] "Score"
```

Question: Is a 90% confidence interval wider or narrower than a 95% confidence interval? Explain why this is so. (In other words, from your understanding of how confidence intervals work, explain why it makes sense that a 90% confidence interval wider or narrower than a 95% confidence interval.)

Rubric for confidence intervals

Calculating a confidence interval is part of the inferential process. The points assigned to each step are laid out in the "Rubric for Inference" posted on Canvas.

Here is a worked example.

Some of the students in the "High School and Beyond" survey attended vocational programs. What percentage of all high school seniors attend vocational programs?

Conditions

- Random

The sample is presumably a representative sample of high school seniors from the U.S. as the survey was conducted by the National Center of Education Statistics.

- 10%

The sample size is 200, which is much less than 10% of the population of all U.S. high school seniors.

- Success/failure

```
table(hsb2$prog)
```

```
##
##    general   academic vocational
##         45        105         50
```

The number of "successes" (students in vocational programs) is 50 which is more than 10, and the number of "failures" (all other programs) is 150, also more than 10.

Calculation

```
program <- binom.test(x = hsb2$prog, success = "vocational")
program$conf.int
```

```
## [1] 0.1916072 0.3159628
## attr(,"conf.level")
## [1] 0.95
## attr(,"method")
## [1] "Score"
```

Conclusion

We are 95% confident that the true proportion of U.S. high school seniors who attend a vocational program is captured in the interval (19.160717%, 31.5962833%).

Your turn!

Use the `smoking` data set from the `openintro` package. What percentage of the population of the U.K. smokes tobacco? (The information you need is in the `smoke` variable.)

Conditions

Calculation

```
## Add code here to calculate the confidence interval.
```

Conclusion