Lab 3 - Math 58 / 58b: CI for a single proportion

your name here

due Feb 11, 2020

Lab Goals

Computing a confidence interval for a single proportion

- wait, what *is* a confidence interval?
- using `xqnorm` (find the number on the x-axis!) [Note: feel free to use `plot=FALSE` if you understand what is going on and don't want the visual representation.]
- practice with `ggplot`

Some thoughts

- Length of assignment turned in: the goal isn't to be as short as possible. The goal is to provide a document (e.g., to a client) that communicates as well as possible. If you leave in all the superfluous stuff, the client won't know what to read. Also goes for messages / long pages of data that the client doesn't think is necessary.

- As you go through the code, ask yourself (or a friend or me!) what each line means. You should understand every single line of code, even if you couldn't reproduce it . . . yet.

Getting started

Load packages

In this lab we will continue to use `infer` and the `xpnorm` function which is in the `mosaic` package.

Let's load the packages.

```r
library(tidyverse)  # ggplot lives in the tidyverse
library(mosaic)  # where xqnorm lives
library(infer)
```

The data

Consider the research study done in 2017 describing the support of marijuana legalization in Washington State.[1]

From the abstract of the paper:

> Data come from geographically representative general population samples of adult (aged 18 and over) Washington residents collected over five timepoints (every six months) between January 2014 and April 2016 (N=4101). Random Digit Dial was used for recruitment. Statistical analyses involved bivariate comparisons of proportions across timepoints and subgroups (defined by age, gender, and marijuana user status), and multivariable logistic regression controlling for timepoint (time) to formally test for trend while controlling for demographic and substance use covariates. All analyses adjusted for probability of selection.

The results are given as:

[1] MS Subbaraman and WC Kerr, "Support for marijuana legalization in the US state of Washington has continued to increase through 2016", *Drug and Alcohol Dependence*, Vol 175: 205-209, 2017. https://www.ncbi.nlm.nih.gov/pubmed/28448904

> Support for legalization in Washington has significantly increased: support was 64.0% (95% CI: 61.2%-67.8%) at timepoint 1 and 77.9% (95% CI: 73.2%-81.9%) at timepoint 5. With each six months' passing, support increased 19% on average. We found no statistically significant change in support for home-growing.

For this lab, we're going to **pretend** to have a population. In this case, it is the population of Washington State in 2016 (about 7 million people), 75% of whom think that marijuana should be legalized. (I made up those numbers, but they seem reasonable!)

```
legal <- data.frame(support = c(rep("yes", 5250000), rep("no", 1750000)),
                    stringsAsFactors=FALSE)
```

1. Let's start with a random sample of 50 people. How many of them support legalization? Is it the same proportion as your neighbor?

```
samp_n <- 50
legal_samp <- sample_n(legal, samp_n)
table(legal_samp)
```

```
## legal_samp
##  no yes
##  14  36
```

2. Using the following formula & your sample of 50 people, find a CI for the true proportion of people who support legalizing marijuana use. Feel free to use R as a calculator.

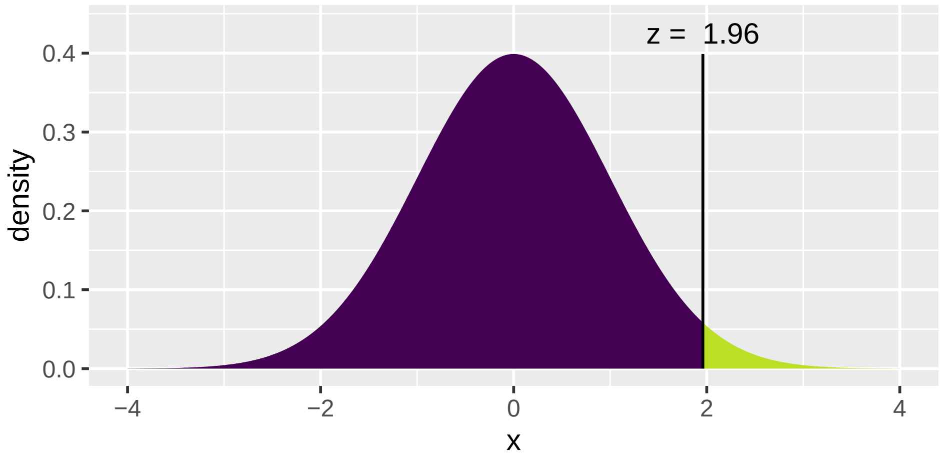
$$\hat{p} \pm 1.96 * \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$$

Note a few things… 1.96 is used instead of 2 for Z*. And $\hat{p}$ was used instead of p in the SE.

Does your CI capture the true value of p (which we know / we set to be 0.75)? Does your neighbor's CI capture the true value of p?

3. Now use the following "tidy" formula to calculate the CI one more time. Did you get the same interval as in #2?

```
(z_star_95 <- xqnorm(0.975, 0, 1))
```



```
## [1] 1.959964
```

```
legal_samp %>%
  summarize(p_hat = sum(support == "yes") / samp_n,
            se = sqrt(p_hat*(1-p_hat)/samp_n),
            me = z_star_95 * se,
            lower = p_hat - me,
            upper = p_hat + me)
```

```
##   p_hat         se        me     lower     upper
## 1  0.72 0.06349803 0.1244539 0.5955461 0.8444539
```

4. If you have run your code a few different times, you may have gotten **different** samples of size 50. If you want to get the sample sample (e.g., so that you can write about your results), set your randomness by using `set.seed()` before you sample, where the argument to the function is your favorite integer. Try it and see if you can get the same repeated results.

5. Because each student took a different random sample, we'd expect 95% of the classroom intervals to capture the true parameter value.

Using R, we're going to collect many samples to learn more about how sample means and confidence intervals vary from one sample to another.

Here is the rough outline:

- Obtain a random sample.
- Calculate the sample proportion and use it to calculate and store the lower and upper bounds of the confidence intervals.
- Repeat 100 times.

We can get many CIs using the `rep_sample_n` function. The following lines of code takes 100 random samples of size `samp_n` from the population (and remember `samp_n` = 50 as defined earlier) and computes the upper and lower bounds of the confidence intervals separately for each of the 100 samples.

```
set.seed(47)
ci <- legal %>%
  infer::rep_sample_n(size = samp_n, reps = 100, replace = FALSE) %>%
  summarize(p_hat = sum(support == "yes") / samp_n,
            se = sqrt(p_hat*(1-p_hat)/samp_n),
            me = z_star_95 * se,
            lower = p_hat - me,
            upper = p_hat + me)
```

Let's view the first five intervals:

```
ci %>%
  head(5)
```

```
## # A tibble: 5 x 6
##   replicate p_hat     se    me lower upper
##   <chr>     <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1 1          0.8  0.0566 0.111 0.689 0.911
## 2 10         0.68 0.0660 0.129 0.551 0.809
## 3 100        0.66 0.0670 0.131 0.529 0.791
## 4 11         0.68 0.0660 0.129 0.551 0.809
## 5 12         0.72 0.0635 0.124 0.596 0.844
```

Next we'll create a plot similar to the Confidence Interval applet and also to Figure 2.37 on page 104 of @isrs. The first step will be to create a new variable in the `ci` data frame that indicates whether the interval does or does not capture the true population mean. Note that capturing this value would mean the lower bound of the confidence interval is below the value and upper bound of the confidence interval is above the value. Remember that we create new variables using the `mutate` function.

```
ci <- ci %>%
  mutate(capture_p = ifelse(lower < ___ & upper > ___, "yes", "no"))

ci %>% select(p_hat, lower, upper, capture_p) %>% head(5)
```

The `ifelse` function takes three arguments: first is a logical statement, second is the value we want if the logical statement yields a true result, and the third is the value we want if the logical statement yields a false result.

We now have all the information we need to create the plot.

Note that the `geom_errorbar()` function only understands `y` values, and thus we have used the `coord_flip()` function to flip the coordinates of the entire plot back to the more familiar vertical orientation.

```
ggplot(ci, aes(x = replicate, y = p_hat, color = ___)) +
  geom_errorbar(aes(ymin = lower, ymax = upper)) +
  geom_hline(yintercept = ___, color = "darkgray") + # draw vertical line
  coord_flip()
```

You can count how many confidence intervals capture the true proportion by using the `summarize` function. [Aside: what is the difference between `=` and `==` in the code below?]

```
ci %>%
  summarize(capturecount = count(capture_p == "___"))
```

To Turn In

1. Consider the original study. Provide one sentence (not a number) describing each of the following:

- observational unit
- variable
- statistic
- parameter

2. Create a bar plot of the first sample you took (above). Use `geom_bar`.

3. Pick a confidence level of your choosing *other than* 95%. What is the number you will use for multiplying the SE? (What is Z*?)

4. Calculate 100 confidence intervals at the confidence level you chose in the previous question, plot all intervals on one plot, and calculate the proportion of intervals that include the true population proportion. How does this percentage compare to the confidence level selected (in the previous question) for the intervals? Make sure to include your plot in your answer.

5. To get a better sense of the actual coverage rate, calculate 1000 intervals (don't try to plot 1000 intervals on one graphic). How many of your intervals captured the true population value? Was it close to the confidence level you chose?

6. Repeat the previous two questions using $n = 610$ (as in the study). Answer the following: – How is the plot different across $n = 50$ and $n = 610$? – Does the coverage rate (how often a sample will capture the true value) depends on n? Explain.