Section 1

Announcements & Recap

Announcements

Upcoming due dates:

- ECoach journal entry 6 due Monday, November 4
- Homework 5 due by 11:59pm on Friday, November 8

Section 2

Learning Objectives

R Learning Objectives

1. Revisit simulation for one proportion
2. Revisit `pnorm()` and `qnorm()`
3. Introduce `prop_test()` to understand test statistics and confidence intervals

Statistical Learning Objectives

1. Understand how area under the normal curve relates to probability
2. Understand how to move between probabilities and quantiles of the normal distribution
3. Build intuition for the relationship between simulated p-values and p-values which arise from a normal approximation.
4. Understand the standard normal distribution and its corresponding z test statistic.
5. Understand how confidence intervals can provide an estimate for the true parameter.

Functions covered in this lab

1. `plot_norm()`
2. `pnorm()`
3. `qnorm()`
4. `prop_test()`

Section 3

Lab Tutorial

Another Simulation Example

According to the American Pet Products Association, prior to the pandemic, 67% of all U.S. households had pets. Many people have speculated that the proportion of households with pets changed during the pandemic. Recently, a group of veterinarians surveyed a random sample of 480 U.S. households and found that 336 had at least one pet.

Is there evidence to support the claim that the proportion of U.S. households with pets differs from the 67% before the pandemic?

Our hypotheses to test if there is a difference are

$$H_0 : p = 0.67 \text{ and } H_A : p \neq 0.67$$

Setting Up the Simulation

What are the elements of the simulation?

Assuming the chance model. . .
One draw
Blue poker chip
Yellow poker chip
Chance of blue
One repetition

Running the Simulation

Recall that we need to set a seed before running a simulation so that we can talk about the results as a class (it will be helpful for us to have the same results). Let's arbitrarily set the seed to 734. Run this in the `setSeed` code chunk in your notes file.

Running the Simulation

Then let's run the simulation 500 times. In the `tryIt1` code chunk, you'll need to enter values for:

- `chanceSuccess` (what we assume p is when the null hypothesis is true)
- `numDraws` (this will equal the sample size)
- `numRepetitions` (the number of times you want to run the simulation so you can get a sense of the distribution of $\hat{p}$ when the null hypothesis is true)

```
sim1 <- simulate_chance_model(chanceSuccess = 0.67,
                              numDraws = 480,
                              numRepetitions = 500)
```

`sim1` should now be in your Environment pane.

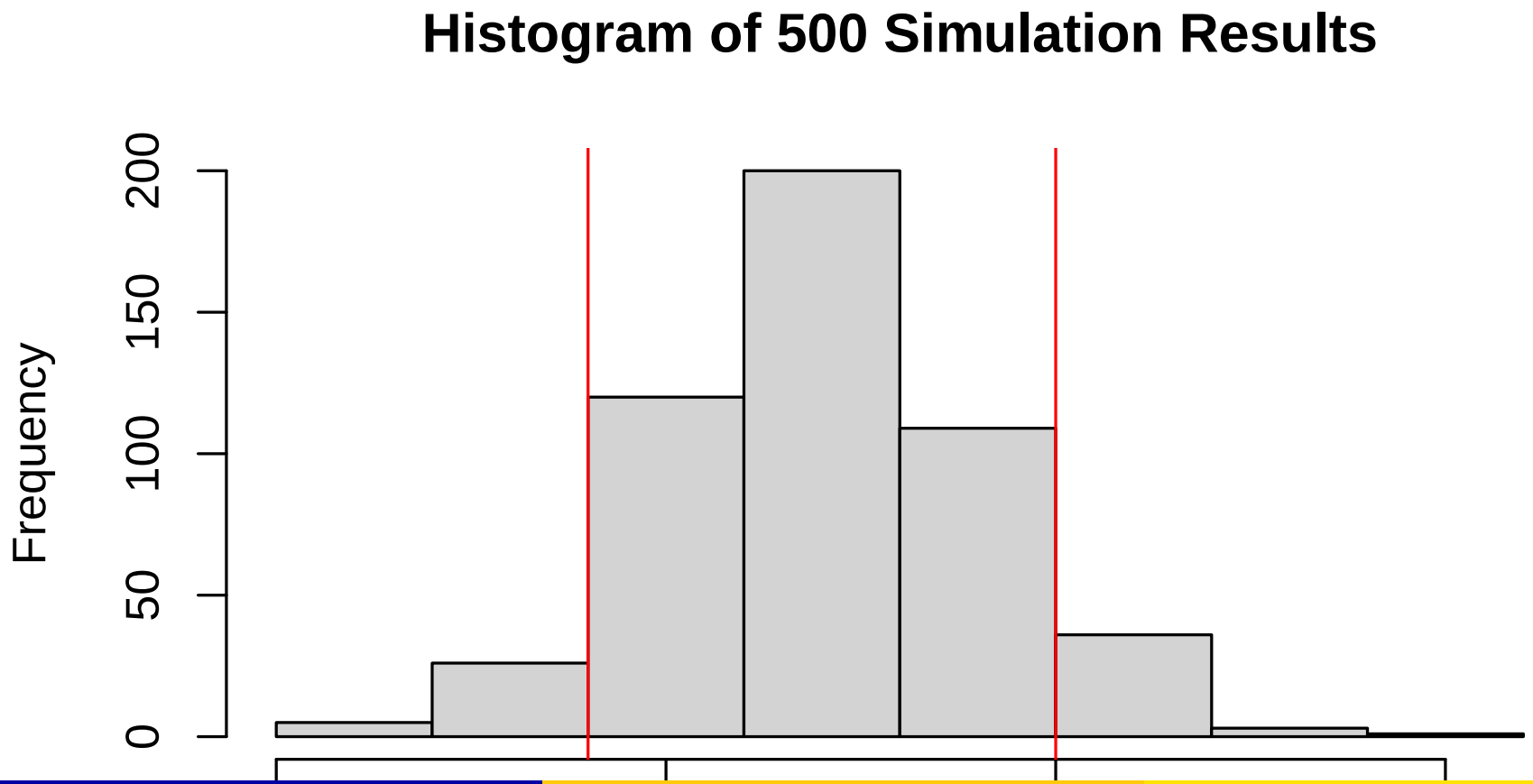
Histogram of Simulation Results

The sample statistic is $\hat{p} = \frac{336}{480} = 0.70$, so let's put a line in the histogram to indicate our observed statistic so that we can get a sense about how unusual it it to get $\hat{p} = 0.70$ when $p = 0.67$.

We have a two-sided hypothesis test, so it would have been just as unusual to see a sample proportion of 0.64, so we will put a line in the histogram to indicate this as well.

Histogram of Simulation Results

Enter the values 0.64 and 0.70 in the `tryIt2` code chunk to create a histogram of the simulated proportions from the 500 repetitions of the simulation.



Estimated p-value

The estimated p-value is the sum of the number of simulations that are as weird as or weirder than we observed (so, 0.70 or larger) *and* the sum of the simulations that are as weird or weirder on the other side (at 0.64 or less). Run this in the `tryIt3` code chunk.

```
total <- sum(sim1 >= 0.70) + sum(sim1 <= 0.64)
estimated_pvalue <- total/500
estimated_pvalue
```

```
## [1] 0.156
```

What is our estimated p-value?

Conclusion

After running the simulation 500 times, assuming that the chance of success was indeed 0.67, the probability that we would get an observed sample proportion of 336/480 or more extreme was computed to be 0.156.

The p-value of 0.156 is larger than even a significance level of 0.10. We have very little to support the claim that the proportion of U.S. households with pets differs from the 67% before the pandemic.

But We Observed a Different Result?!

You might have asked yourself, "But wait, why do we have little to no evidence to support the claim? I did in fact observe a rate different than 67% in my sample (I observed 70%, in fact). Isn't a 3% difference strong evidence?"

This is where normal theory can help us understand that the difference of 3% between the $\hat{p}$ and the assumed p was not enough evidence.

Histogram from Simulation

It turns out, the histogram that we created using simulation is in fact approximately normal.

We talked about how the distribution is centered at H_0, the chance of success. Thus, we can assume the mean is 0.67 in our example about U.S. households with pets prior to the pandemic.

To find the standard error, which is an estimate of the standard deviation for our sample statistic $\hat{p}$, we will use the following formula:

$$\sqrt{\frac{p_0(1-p_0)}{n}}$$

Let's compute the standard error, $SE(\hat{p})$.

Histogram from Simulation

The standard error for $\hat{p}$ can be found with the formula

$$\sqrt{\frac{p_0(1-p_0)}{n}}$$

Let's have R compute this for us. You will need to enter values for `p_0` and `n` in the `tryIt4` chunk in your notes and then run the chunk to calculate the SE.

```
p_0 <- 0.67
n <- 480
SE <- sqrt(p_0*(1 - p_0)/n)
SE
```

```
## [1] 0.02146218
```

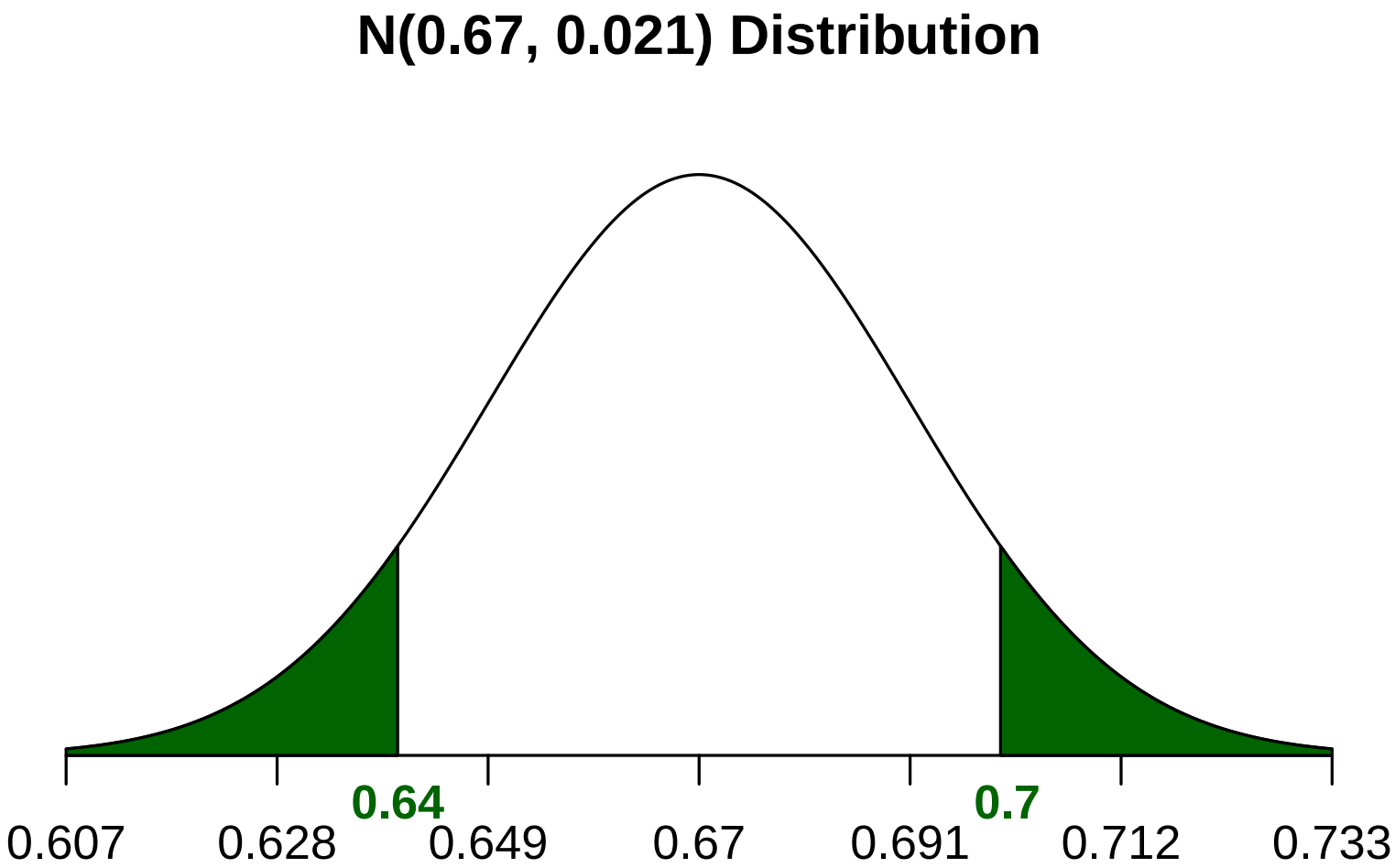
We will use the SE that R calculated to visualize the p-value for the hypothesis test.

Normal Distribution for Our Example

Let's take a look at the approximate normal distribution for our U.S. households with pets example. Recall that, when the null hypothesis is true, the mean is **0.67**, the standard error is **0.021**, and we want to shade **beyond** the values of 0.64 and 0.7. Enter the `mean` and `sd` in the `tryIt5` chunk in your notes and then run the chunk to visualize the p-value.

```
plot_norm(mean = 0.67,
          sd = 0.021,
          shadeValues = c(0.64, 0.7),
          direction = "beyond",
          col.shade = "darkgreen")
```

Normal Distribution for Example



There Is in Fact Insufficient Evidence

Now that we see that the standard deviation is 0.021, or 2.1%, we can start to understand why observing a rate that was only 3% different from the assumed mean is not all that rare.

Looking at the normal distribution plot, we can observe that 0.7 is only between 1 and 2 standard deviations above the mean. Again, not all that rare.

Another Way to Calculate the p-value

The `stats250sbi` package we have been using also contains a function called `prop_test()` that will calculate the value of the test statistic and the p-value for us. Professor Miller showed you this function briefly in lecture.

You will need to send the function the following arguments:

- `x`: the number of "successes" in the sample
- `n`: the sample size
- `p`: the hypothesized population proportion
- `alternative`: where to shade (`"two.sided"`, `"less"`, `"greater"`)
- `conf.level` (optional): used to get a confidence interval (we will use this later)

Using prop_test for Our Pet Example

We are testing $H_0 : p = 0.67$ and $H_A : p \neq 0.67$. You will need to enter values for

- `x` the number of successes (here the number of households with a pet)
- `n` the sample size
- `p` the value we assume p to be when H_0 is true
- `alternative` the direction of H_A (`"two.sided"`, `"less"`, `"greater"` – make sure the direction is in quotes)

in the `tryIt6` chunk in your notes before you run it.

```
prop_test(x = 336,
          n = 480,
          p = 0.67,
          alternative = "two.sided")
```

Using prop_test for Our Pet Example

```
##
##  1-sample proportions test without continuity correction
##
## data:  x out of n, null probability p
## Z = 1.3978, p-value = 0.1622
## alternative hypothesis: true p is not equal to 0.67
## 95 percent confidence interval:
##  0.6590044 0.7409956
## sample estimates:
##   p
## 0.7
```

The pnorm() Function

The p-value from `prop_test()` is pretty close to the p-value from our simulation.

We can also compute the p-value using the `pnorm()` function. Recall the arguments you need to send to `pnorm()`:

- `q`: the quantile (value on the axis) for the normal distribution
- `mean`: the mean of the normal distribution (μ)
- `sd`: the standard deviation of the normal distribution (σ)
- `lower.tail`: set to 'TRUE' initially, signifying that R will compute the probability **to the** LEFT of `q`; if you would like R to compute the probability *to the* right of `q`, set `lower.tail` to FALSE

Computing the p-value for the Simulation Example

Let's compute the approximate p-value using the normal distribution for our pet example. Recall that the mean is **0.67**, and the standard deviation is **0.021**.

Because the normal distribution is **symmetric** about the mean, we can find the probability of observing 0.7 or greater, then **double it**.

Or, we can find the probability of observing 0.64 or less, then **double it**.

Or, we can find the probability of observing 0.70 or greater, then the probability of observing 0.64 or less, and add the result.

Since 0.7 is the $\hat{p}$ for our sample, we will utilize the first option in the `pvalue` code chunk.

Computing the p-value for the Simulation Example

Run this code in the `tryIt7` code chunk.

```
2 * pnorm(q = 0.7,
      mean = 0.67,
      sd = 0.021,
      lower.tail = FALSE)
```

```
## [1] 0.1531275
```

Comparing the Various p-values

To recap, we have *three* ways to compute the p-value for a one proportion hypothesis test:

1. Create a vector of simulated proportions using `simulate_chance_model()`, then using the `sum()` function to count the number of observations at or beyond the sample proportion divided by the number of observations
2. Use the `prop_test()` function (which uses normal theory) by sending the number of successes observed, the sample size, the value of H_0, and the direction of the alternative hypothesis
3. Compute the μ and σ for the approximate normal distribution. Use the `pnorm()` function by sending the value of the sample proportion, μ, σ, and the direction of the probability

Each of these will produce a slightly different result. **No need to worry about how close the values should be, or which value is "best".**

Comparing the Various p-values

```
sum(sim1 <= 0.64) / 500 + sum(sim1 >= 0.7) / 500
```

```
## [1] 0.156
```

```
prop_test(x = 336, n = 480, p = 0.67)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  x out of n, null probability p
## Z = 1.3978, p-value = 0.1622
## alternative hypothesis: true p is not equal to 0.67
## 95 percent confidence interval:
##  0.6590044 0.7409956
## sample estimates:
##   p
## 0.7
```

```
pnorm(q = 0.7, mean = 0.67, sd = 0.021, lower.tail = FALSE) * 2
```

```
## [1] 0.1531275
```

Comparing the Various p-values

1. The simulation is the most accurate, because it is computing the p-value with simulated values.
2. `prop_test()` and `pnorm()` will lose some precision due to utilizing the normal approximation. This loss of precision should not affect our results.
3. `pnorm()` will lose some precision if we round the standard deviation to 3 decimal places. This loss of precision should not affect our results.

Using prop_test() to Find Confidence Intervals

The output from `prop_test()` also provides a confidence interval for the population proportion. The default confidence level is 0.95 for a 95% confidence interval. The 95% confidence interval for this example is (0.659, 0.741).

```
##
##  1-sample proportions test without continuity correction
##
## data:  x out of n, null probability p
## Z = 1.3978, p-value = 0.1622
## alternative hypothesis: true p is not equal to 0.67
## 95 percent confidence interval:
##  0.6590044 0.7409956
## sample estimates:
##   p
## 0.7
```

Using `prop_test()` to Find Confidence Intervals

We can change the confidence level to a level other than 95% by adding the argument `conf.level` to the `prop_test()` function.

Also, **if we just need to calculate a confidence interval** we don't have a hypothesized value of p and we don't have an alternative hypothesis, so we drop the `p` and `alternative` arguments from `prop_test`.

Using prop_test() to Find Confidence Intervals

In the `tryIt8` chunk, set `conf.level` to 0.98.

```
prop_test(x = 336,
          n = 480,
          conf.level = 0.98)
```

Using prop_test() to Find Confidence Intervals

```
prop_test(x = 336, n = 480, conf.level = 0.98)
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  x out of n, null probability 0.5
## Z = 8.7636, p-value < 2.2e-16
## alternative hypothesis: true p is not equal to 0.5
## 98 percent confidence interval:
##  0.6513409 0.7486591
## sample estimates:
##   p
## 0.7
```

We estimate (at the 98% confidence level) that the population proportion of homes with pets is 0.651 to 0.749.

Using `prop_test()` to Find Confidence Intervals

Caution: To get a two-sided confidence interval, the `alternative` argument *must* be set to `two.sided`. If it isn't, you will get a *confidence bound*.

Note that `two.sided` is the default for `prop_test()`, so if you just want a confidence interval you can leave the `alternative` argument off.

Confidence bounds can be useful when we have one-sided hypothesis tests, but we will leave them to your later statistics courses.

One More Example

X, the social media platform formerly known as Twitter, saw a name change and policy changes once it was under new ownership. Shortly after the name change, seventy-five percent (75%) of all X users said that they planned to keep using X in 2024. Researchers at X would like to know if this rate differs for Millennials, those born between 1981 and 1996. A random sample of 100 Millennial users of X revealed that 65% planned to keep using X in 2024.

What Would We Expect to See in the Sample?

If the rate of current Millennial users of X is the same as the rate for all X users, how many Millennial users of X from the sample of 100 would we expect to say that they planned to keep using X in 2024?

Incorrect Alternative

The researchers at X wanted to know if the rate of Millennials who planned to continue using X in 2024 differs from the rate of all users of X. The researcher in charge accidentally ran a one-sided hypothesis test and got the following output

```
prop_test(x = 65, n = 100, p = 0.75, alternative = "less")
```

```
##
##  1-sample proportions test without continuity correction
##
## data:  x out of n, null probability p
## Z = -2.3094, p-value = 0.01046
## alternative hypothesis: true p is less than 0.75
## 95 percent confidence interval:
##  0.0000000 0.7284545
## sample estimates:
##    p
## 0.65
```

Correcting the Mistake

The value of the test statistic for the incorrect alternative was $z = -2.309$. What should the test statistic be for the correct two-sided test?

- It should be half as much; that is, $z = -1.155$.
- It should be positive instead of negative; that is, $z = 2.309$.
- It should be the same; that is, $z = -2.309$.
- It should be twice as much; that is, $z = -4.618$.

Correcting the Mistake

The p-value for the incorrect alternative was 0.01046. What should the p-value be for the correct two-sided test?

- It should be half as much; that is, p-value $= 0.00523$.
- It should be negative instead of positive; that is, p-value $= -0.01046$.
- It should be the same; that is, p-value $= 0.01046$.
- It should be twice as much; that is, p-value $= 0.02092$.