Lab 3 - Math 58B: Randomization Test with `infer`

due Tuesday Feb 10, 2026

your name here

```r
library(tidyverse)
library(infer)
library(praise)
```

Building on the work we've done this week to run a hypothesis test, we will use the **infer** package to complete an entire randomization test.

The goals for today include:

- using **infer** to complete a **randomization** hypothesis test
- creating a histogram representing a null sampling distribution
- calculating a p-value
- summarizing the hypothesis test in words of the problem

Advice for turning in the assignment

- render early and often. In fact, go ahead and render your .amd file right now. Maybe set a timer so that you render every 5 minutes. Do **not** wait until you are done with the assignment to render

- The **assignment** part of the lab is **ONLY** the last six questions at the very bottom. However, the commands in the first half of the assignment are key to doing the second half.

- Save the .qmd file somewhere you can find it. Don't keep everything in your downloads folder. Maybe make a folder called `StatsHW` or something. That folder could live on your Desktop. Or maybe in your Dropbox.

Getting started

The infer vignette is excellent: https://infer-dev.netlify.app/index.html

As we go through the lab today, focus on the names of the function to make sure that you connect the **name** of the function to the **action** of the function.

Load packages / data

In this lab we will use new syntax from the **infer** package. The syntax is meant to focus understanding on the hypothesis testing process. So for each line, pay attention to what the code is doing.

The data come from a randomized clinical trial to discern the difference between a sugar gargle versus licorice gargle after undergoing elective thoracic surgery on the amount of coughing post-surgery (reference and data here: https://www.causeweb.org/tshs/licorice-gargle/).

```
library(infer)
licorice_study <- readr::read_csv("http://pages.pomona.edu/~jsh04747/courses/math58/Licorice
  mutate(gargle = case_when(
    treat == 0 ~ "sugar",
    treat == 1 ~ "licorice")) |>
  mutate(cough = case_when(
    pod1am_cough == 0 ~ "none",
    TRUE ~ "some")) |>
  select(gargle, cough) |>
  drop_na()

licorice_study
```

```
# A tibble: 235 x 2
   gargle   cough
   <chr>    <chr>
 1 licorice none
 2 licorice none
 3 licorice none
 4 licorice none
 5 licorice none
 6 licorice none
 7 licorice none
 8 licorice none
 9 licorice none
```

```
10 licorice none
# i 225 more rows
```

Logic for Hypothesis Testing (spoiler)

1. We know that the study was an experiment, so there should be no **systematic** differences (in other variables) between the group who received the sugar versus the licorice gargle.
2. We hope to rule out random chance as the reason for the difference in proportions of the who is coughing after surgery. (We hope to reject the null hypothesis.)
3. If we can reject the null hypothesis, we conclude that the gargle type and coughing outcome are not independent. That is, the type of gargle used changes the probability that the patient will have coughing after surgery.

A randomization test

```
# to control the randomness
set.seed(47)

licorice_study |> table()
```

```
          cough
gargle     none some
  licorice   86   32
  sugar      68   49
```

Step 1. Observed Statistic

The first thing we need to do is to find the observed statistic of interest (here the difference in sample proportions). Note that R is happy to act as a calculator, but we're going to use the syntax associated with the test to get the value of interest.

Well, okay, first as a calculator, the difference in proportion who still had coughing post surgery with licorice minus surgar is 0.147617.

```
(49/117) - (32/118)
```

```
[1] 0.147617
```

Now using the infer syntax, we'll specify the variables of interest (which is response and which is explanatory?). Also, we'll need to specify what a "success" means and the order of subtraction. Note that we get the same value as we did when using R as a calculator (whew!).

```r
diff_obs <- licorice_study |>
    specify(cough ~ gargle, success = "some") |>
    calculate(stat = "diff in props", order = c("sugar", "licorice")) |>
  pull()

diff_obs
```

```
[1] 0.147617
```

Step 2. Shuffle the data under H0

The point of the hypothesis test structure is to have an understanding of what types of values might be seen just by chance (if the type of gargle really wasn't doing anything). The idea is that 81 people were going to have some coughing after surgery anyway (regardless of gargle) – that's the null hypothesis! And we expect some variability in which group those 81 people end up. How big are the differences in proportions? is 14.8% big? or is it small?

Hint: go back to this applet to see how the shuffling works!
http://www.rossmanchance.com/applets/2021/chisqshuffle/ChiSqShuffle.htm?FET=1

A note on the code… pay attention to the steps here:

- `specify()` gives the appropriate information on the variable types. Always: `responsevariable ~ explanatoryvariable`
- `hypothesize()` gives the null hypothesis of interest. Here we are performing a test of independence, but that will change over the semester as we do different types of tests.
- `generate()` swaps things around as if the null hypothesis were true. Here we are permuting the data (randomly reassigning it) as if the gargle didn't have an impact.

Take a look at the output of the first three steps, before calculating the difference in proportions. It's hard to tell just by looking at the dataframe, but really what you see is that different people have been assigned to different treatments (sugar or licorice).

```r
licorice_study |>
  specify(cough ~ gargle, success = "some") |>
  hypothesize(null = "independence") |>
  generate(reps = 4, type = "permute")   # set reps=4 just to see the process
```

```
Response: cough (factor)
Explanatory: gargle (factor)
Null Hypothesis: independence
# A tibble: 940 x 3
# Groups:   replicate [4]
   cough gargle   replicate
   <fct> <fct>        <int>
 1 none  licorice         1
 2 none  licorice         1
 3 some  licorice         1
 4 none  licorice         1
 5 none  licorice         1
 6 some  licorice         1
 7 none  licorice         1
 8 none  licorice         1
 9 none  licorice         1
10 some  licorice         1
# i 930 more rows
```

The last step ties it all together:

- `calculate()` finds the statistic (here our statistic is the difference in proportions) for each of the permuted datasets.

Keep all of those differences, and take a look at them. Is 14.8% is big? or is it small? can you tell?

```
set.seed(4774)
null_licorice <- licorice_study |>
  specify(cough ~ gargle, success = "some") |>
  hypothesize(null = "independence") |>
  generate(reps = 1000, type = "permute") |>  # need a lot of reps to truly understand
  calculate(stat = "diff in props", order = c("sugar", "licorice"))

null_licorice
```

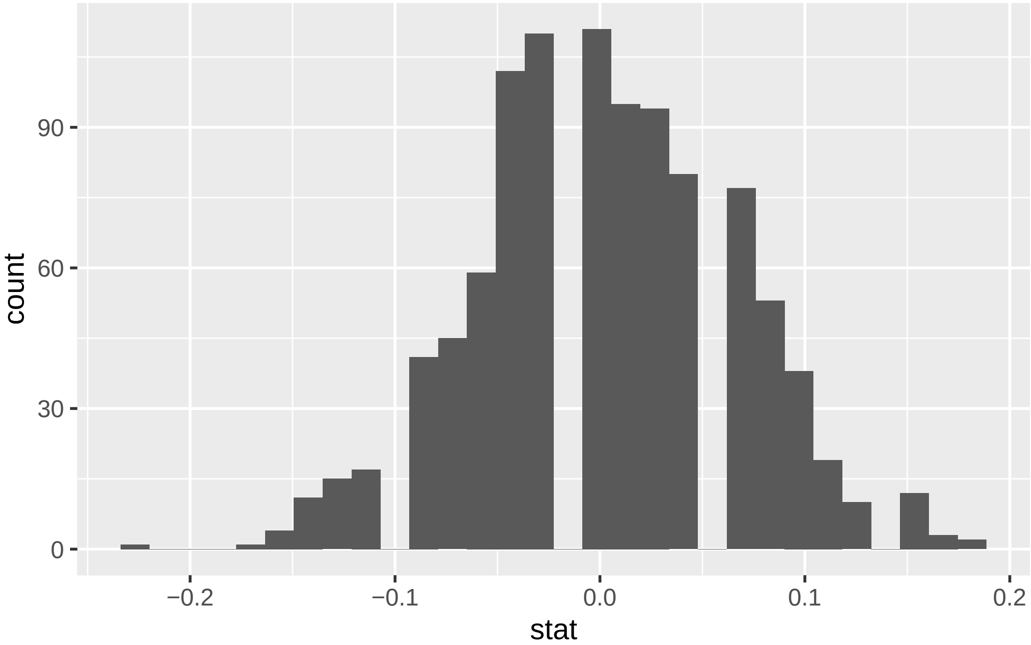
```
Response: cough (factor)
Explanatory: gargle (factor)
Null Hypothesis: independence
# A tibble: 1,000 x 2
   replicate     stat
       <int>    <dbl>
```

```
 1         1 -0.0566
 2         2 -0.125
 3         3 -0.00558
 4         4 -0.00558
 5         5  0.0285
 6         6  0.0114
 7         7  0.114
 8         8  0.0114
 9         9  0.0625
10        10  0.0114
# i 990 more rows
```

Step 3. Look at all the differences

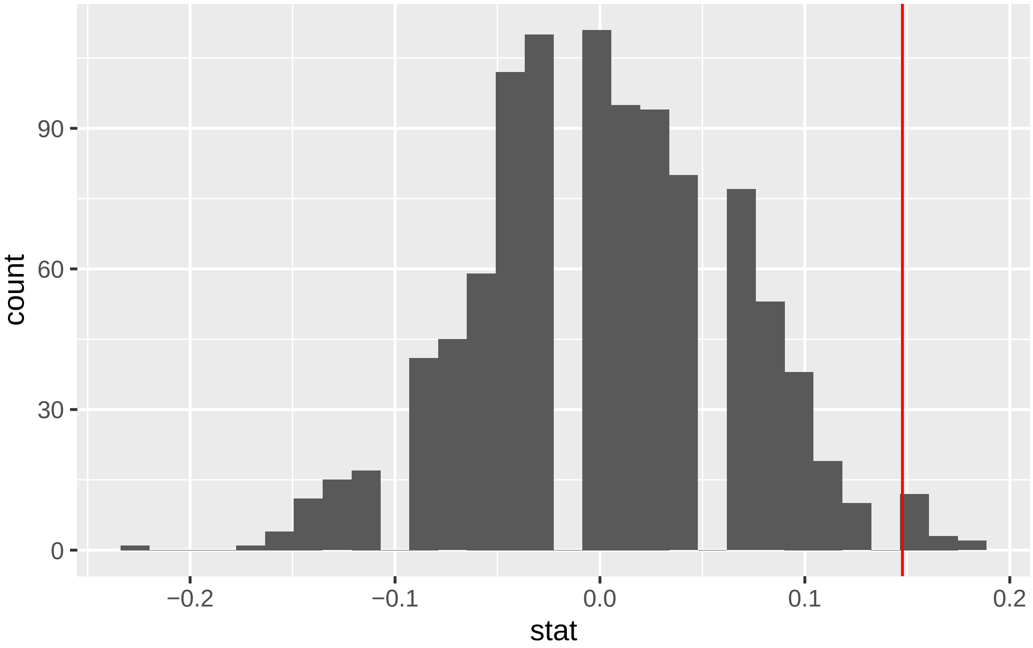
Fortunately for us, we saved all the differences in proportions into an object that was called `null_licorice`. We can use `ggplot()` to make a histogram!

```
null_licorice |>
  ggplot(aes(x = stat)) +
  geom_histogram()
```



It is still a little bit hard to tell where that 14.8% falls. It doesn't seem way outside the range, but it is in the tails. Let's plot the observed difference on the plot of null differences. Remember, when adding layers to a plot we use `+` instead of `|>`.

```
null_licorice |>
  ggplot(aes(x = stat)) +
  geom_histogram() +
  geom_vline(xintercept = diff_obs, color = "red")
```



Step 4. Calculate the p-value

Recall that the p-value is the probability of the observed difference or more extreme if the variables are independent (that is, if the null hypothesis is true and licorice doesn't impact whether or not you are more likely to cough post-surgery).

Note that `stat >= diff_obs` produces TRUE and FALSE values. If you let TRUE = 1 an FALSE = 0, then the average will be the proportion of times the null statistics are greater than the observed statistic.

Pay attention to the difference between the one-sided and two-sided p-values. How are they different?

```
null_licorice |>
  summarize(one_sided_p = mean(stat >= diff_obs),
            two_sided_p = mean((abs(stat) >= diff_obs)))
```

```
# A tibble: 1 x 2
```

```
  one_sided_p two_sided_p
        <dbl>       <dbl>
1       0.017       0.023
```

Step 5. Make a conclusion

The p-value here is 0.023. We're saying that if those 81 people were going to cough anyway, regardless of gargle, only 2.3% of the time would so many of them have landed in the sugar group just by chance. The probability of the observed difference seems pretty small. It makes us think that chance was probably not the mechanism that put so many patients into the sugar group. Instead, this process makes us believe that it was actually the licorice (in contrast to the sugar) that reduced the coughing.

Because this was a randomized trial, all other characteristics of the patients are balanced out (i.e., no confounding variables!). And we seem to have ruled out chance as the mechanism (p-value is small!).

Conclusion: the difference in probability of coughing is due to the choice of gargle. That is, licorice gargle reduces the probability of coughing post-operative for this particular elective thoracic surgery.

To Turn In

The data

The data for the write-up part of the lab is on a diabetes clinical trial[1] and available from the **openintro** package. (Note: in the original study there were three treatments, today we'll just compare metformin with a lifestyle-intervention program.)

> Three treatments were compared to test their relative efficacy (effectiveness) in treating Type 2 Diabetes in patients aged 10-17 who were being treated with metformin. The primary outcome was lack of glycemic control (or not); lacking glycemic control means the patient still needed insulin, which is not the preferred outcome for a patient.

```r
library(infer)  # for doing the randomization test
library(openintro)  # home of the dataset for the lab
data(diabetes2)

diabetes <- diabetes2 |>
  filter(treatment != "rosi") |>
  mutate(treatment = droplevels(treatment))

diabetes |> table()
```

```
           outcome
treatment   failure success
  lifestyle     109     125
  met           120     112
```

Q1. The study

Answer the following questions with respect to the study (feel free to read about the study more on your own, or you might type into the console: `?diabetes2`).

- What are the observational units?
- What are the variables? Label them as explanatory and response.
- Is it an experiment or an observational study?
- Was the treatment randomly assigned? If so, what might you conclude at the end of this lab? If not, what are you unable to conclude? Explain.

[1] Zeitler P, et al. 2012. A Clinical Trial to Maintain Glycemic Control in Youth with Type 2 Diabetes. N Engl J Med.

- Were the observational units randomly selected from the population? (Please specify the population which you think is most relevant here.). If so, what might you conclude at the end of the lab? If not what are you unable to conclude? Explain.

Q2. Hypotheses

Write out the null and alternative hypotheses. Use words like diabetes in your claims. For this lab, let's say you don't have scientific background to think that one treatment over the other might be best. Instead, the goal is to find out whether there is a difference between the two treatments.

Q3. Observed test statistic

Using the infer syntax, calculate the observed test statistic.

Q4. Null test statistics

Calculate 1000 null test statistics from 1000 different permutations of the data.

Q5. Visualize

Using the results from Q5, make a histogram to visualize the observed test statistic. Include the observed statistic, and convince yourself which area of the histogram which is "more extreme" from the observed statistic. (Remember that the hypothesis claim here is that the treatments are different.)

Q6. p-value

Calculate the p-value. Remember that the hypothesis claim here is that the treatments are different.

Q7. Conclusion

Give a conclusion about the hypothesis claims. Do you reject H0 or not? Do you have evidence for cause? To what population can you / can't you generalize the results?

```r
praise()
```

```
[1] "You are amazing!"
```