The Mechanics of Hypothesis Testing… Utilizing Infer!

Robin H[1,]

[1] Department of Sociology & Statistics, Reed College

Step 1: Define Your Hypothesis Test

The first step of any good hypothesis test is—you guessed it!—deciding what hypothesis to test. Here are the main questions I like to ask when first approaching how to write out my null/alternative hypotheses:

1. What **test statistic** is of interest to us?
2. Determine the **range of values** the null hypothesis should cover. Is the “boring” outcome when the difference of means/proportions is zero? When the difference is zero or less than zero? If the proportion is 0.5?
3. What about the alternative hypothesis? (**Hint:** this is usually the opposite of the null values you determined, or whatever values are left over!)

Step 2: Using `specify`

In `specify`, we let R know the variables from our data frame we’ll use in our hypothesis test. This always includes a response (dependent) variable, and potentially a predictor (independent) variable.

- **Only a response variable:** `specify(response = your_var_here)`
- **Response and predictor:** `specify(response_var_name ~ predictor_var_name)`

Additionally, when your test statistic involves proportions, you must tell R in `specify` **what categorical response level should be used to calculate the proportion.** For example, consider a sample that has a variable called fruit, with 60% of responses indicating “apple” and 40% indicating “not an apple”.

To calculate **the proportion of apples in the sample**, you’d add the following code: `specify(response = fruit, success = "apple")`. Conversely, to find **the proportion of non-apples in the sample**, you’d write: `specify(response = fruit, success = "not an apple")`.

Step 3: Using `hypothesize`

It is crucial to let R know what type of test we’ll be doing based on the types of variables we’re using! With this information, R will be able to create the correct **null distribution** for our data (AKA, what the data would look like if no significant relationship existed).

- When the test only has one variable, use one of the following.*
  - **Categorical response:** `hypothesize(null = "point", p = _)`. Here, set `p = _` to the proportion indicated in your null/alternative hypotheses. This is most commonly 0.5, but could be other values between 0 and 1!
  - **Numerical response:** `hypothesize(null = "point", mu = _)`. Same procedure here—fill in the blank with the number you’re interested in testing to see if there is a significant relationship!
- When the test has **both a response and predictor variable** write: `hypothesize(null = "independence")`
- The `null = "point"` refers to the **singular point/value** the null distribution will center itself around, capturing random variance between various generated samples.

Step 4: Using `generate`

Now, to get our null distribution, we must **simulate multiple new samples** that seek to model what samples would look like if our null hypothesis was true.

See below for the three types of arguments you may feed into your `generate` command.

1. **One categorical variable:** `generate(reps = 1000, type = "draw")`
2. **One numerical variable:** `generate(reps = 1000)`
3. **Any combination of two variables:** `generate(reps = 1000, type = "permute")`

Pro-tip: You’ll use the argument `type = "bootstrap"` once we get to confidence intervals in `infer`!

Step 5: Using `calculate`

Now, to finish, we need to grab the test statistic for all of the different null samples we created! To do this, we’ll make use of the `calculate` command! Fill in the code like so, to tell R what test statistic you’ll use:

`calculate(stat = "___")`, where __ could be: “prop”, “mean”, “diff in props”, “diff in means”, etc. (View all options by running `?calculate` in your console).

Pro-tip: For any "diff in __" calculations, it’s smart to note the order you’d like R to subtract the proportions/means by. To go back to our earlier apple example, you would write `order = c("apple", "not apple")` to indicate a subtraction of apple - not apple. This is imperative to improve interpretability and consistency with your written hypotheses!

Step 6: Put It All Together!

For the last two steps, I’ll use data on admissions by gender to an unspecified department at UC Berkeley. See documentation for this dataset here, and download it yourself from the `datasets` package!

Grabbing the sample statistic

Save a new dataframe that takes your data and runs **only the specify and calculate lines**. This will calculate the test statistic of interest, but for just your original sample!

```
sample_statistic <- UCBAdmissions_dept_A %>%
  specify(Admit ~ Gender, success = "Admitted") %>%
   # Because our response variable is categorical, we'll use diff in
        props
  calculate(stat = "diff in props", order = c("Female", "Male"))
```

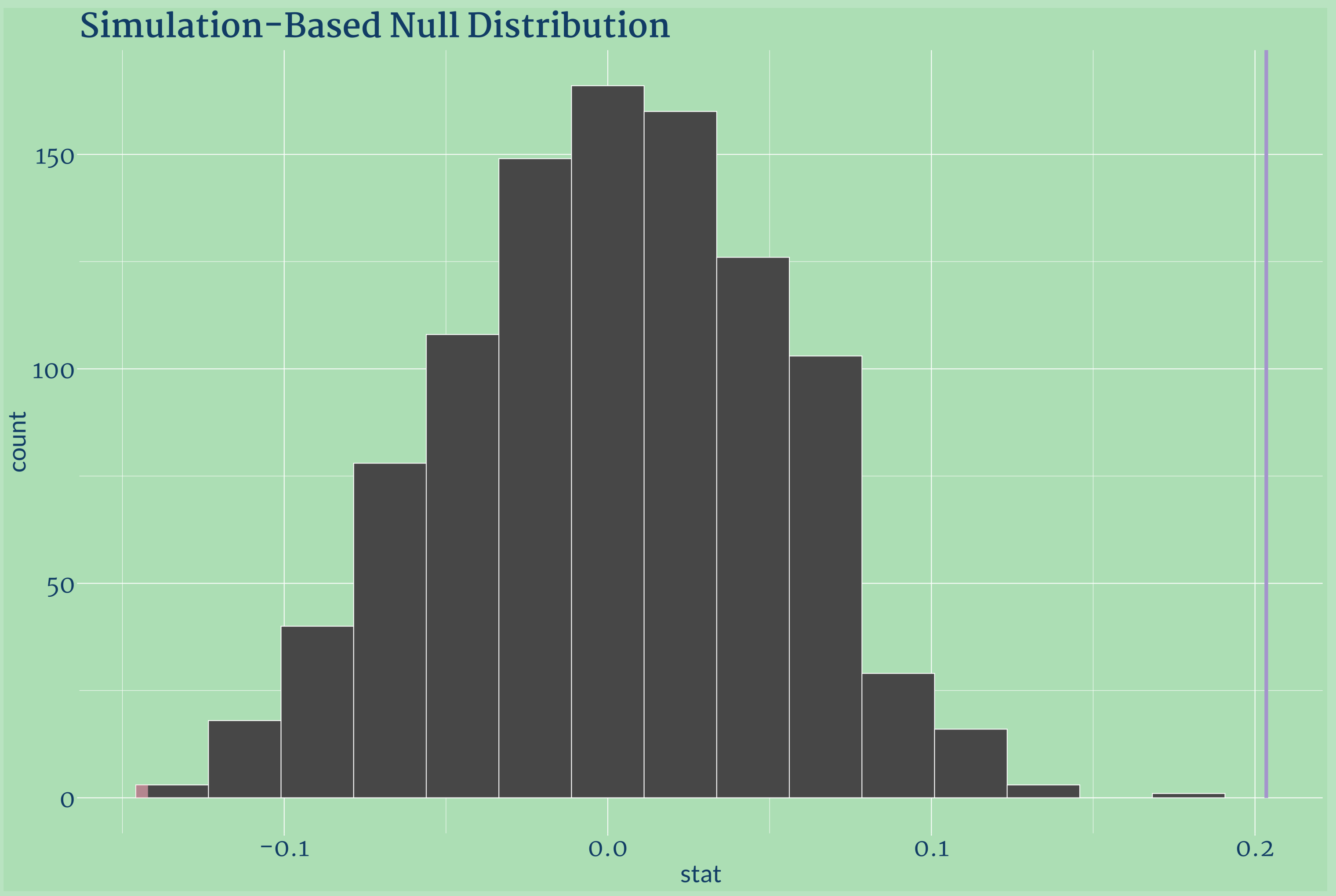
Creating a null distribution

```
null_dist <- UCBAdmissions_dept_A %>%
  specify(Admit ~ Gender, success = "Admitted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("Female", "Male"))
```

Pro-tip: Be sure to use the same original dataframe for both calculations! Using the `sample_statistic` dataframe to calculate the null distribution will not work out well :(

Step 7: Visualization and p-value Calculation

```
visualize(null_dist) +
  shade_p_value(obs_stat = sample_statistic,
                direction = "two-sided", color = "#b4a7d6") +
  college_center_theme
```



Choose the direction of your test based on how you wrote out your null/alternative hypotheses! As a rule of thumb, any hypothesis that includes a > or < sign will be one-sided (left or right depending on the direction of the sign), while tests that consider the alternative as ≠ will be two-sided.

Calculate a p-value

```
null_dist %>%
  get_p_value(obs_stat = sample_statistic, direction = "two-sided")
```

```
## # A tibble: 1 × 1
##   p_value
##     <dbl>
## 1       0
```

Further Readings

A helpful vignette from the creators of `infer`.

Your textbook (Modern Dive)’s discussion of hypothesis testing.

A brief `infer` tutorial by a seasoned professor of data science.

Another short tutorial from a data science professor, chock full of many helpful visualizations!

The github created by the writers of `infer`. The most helpful information is in the README, which you can view about halfway down the page. (Fun fact: Many Reed students and professors were involved in its creation!)