ModernDive

Chapter 9 Hypothesis Testing

Now that we've studied confidence intervals in Chapter 8, let's study another commonly used method for statistical inference: hypothesis testing. Hypothesis tests allow us to take a sample of data from a population and infer about the plausibility of competing hypotheses. For example, in the upcoming "promotions" activity in Section 9.1, you'll study the data collected from a psychology study in the 1970s to investigate whether gender-based discrimination in promotion rates existed in the banking industry at the time of the study.

The good news is we've already covered many of the necessary concepts to understand hypothesis testing in Chapters 7 and 8. We will expand further on these ideas here and also provide a general framework for understanding hypothesis tests. By understanding this general framework, you'll be able to adapt it to many different scenarios.

The same can be said for confidence intervals. There was one general framework that applies to *all* confidence intervals and the `infer` package was designed around this framework. While the specifics may change slightly for different types of confidence intervals, the general framework stays the same.

We believe that this approach is much better for long-term learning than focusing on specific details for specific confidence intervals using theory-based approaches. As you'll now see, we prefer this general framework for hypothesis tests as well.

If you'd like more practice or you're curious to see how this framework applies to different scenarios, you can find fully-worked out examples for many common hypothesis tests and their corresponding confidence intervals in Appendix B. We recommend that you carefully review these examples as they also cover how the general frameworks apply to traditional theory-based methods like the t-test and normal-theory confidence intervals. You'll see there that these traditional methods are just approximations for the computer-based methods we've been focusing on. However, they also require conditions to be met for their results to be valid. Computer-based methods using randomization, simulation, and bootstrapping have much fewer restrictions. Furthermore, they help develop your computational thinking, which is one big reason they are emphasized throughout this book.

Needed packages

Let's load all the packages needed for this chapter (this assumes you've already installed them). Recall from our discussion in Section 4.4 that loading the `tidyverse` package by running `library(tidyverse)` loads the following commonly used data science packages all at once:

- `ggplot2` for data visualization
- `dplyr` for data wrangling
- `tidyr` for converting data to "tidy" format
- `readr` for importing spreadsheet data into R
- As well as the more advanced `purrr` , `tibble` , `stringr` , and `forcats` packages

If needed, read Section 1.3 for information on how to install and load R packages.

```r
library(tidyverse)
library(infer)
library(moderndive)
library(nycflights13)
library(ggplot2movies)
```

9.1 Promotions activity

Let's start with an activity studying the effect of gender on promotions at a bank.

9.1.1 Does gender affect promotions at a bank?

Say you are working at a bank in the 1970s and you are submitting your résumé to apply for a promotion. Will your gender affect your chances of getting promoted? To answer this question, we'll focus on data from a study published in the *Journal of Applied Psychology* in 1974. This data is also used in the *OpenIntro* series of statistics textbooks.

To begin the study, 48 bank supervisors were asked to assume the role of a hypothetical director of a bank with multiple branches. Every one of the bank supervisors was given a résumé and asked whether or not the candidate on the résumé was fit to be promoted to a new position in one of their branches.

However, each of these 48 résumés were identical in all respects except one: the name of the applicant at the top of the résumé. Of the supervisors, 24 were randomly given résumés with stereotypically "male" names, while 24 of the supervisors were randomly given résumés with stereotypically "female" names. Since only (binary) gender varied from résumé to résumé, researchers could isolate the effect of this variable in promotion rates.

While many people today (including us, the authors) disagree with such binary views of gender, it is important to remember that this study was conducted at a time where more nuanced views of gender were not as prevalent. Despite this imperfection, we decided to still use this example as we feel it presents ideas still relevant today about how we could study discrimination in the workplace.

The `moderndive` package contains the data on the 48 applicants in the `promotions` data frame. Let's explore this data by looking at six randomly selected rows:

```
promotions %>%
  sample_n(size = 6) %>%
  arrange(id)
```

```
# A tibble: 6 × 3
     id decision gender
  <int> <fct>    <fct>
1    11 promoted male
2    26 promoted female
3    28 promoted female
4    36 not      male
5    37 not      male
6    46 not      female
```

The variable `id` acts as an identification variable for all 48 rows, the `decision` variable indicates whether the applicant was selected for promotion or not, while the `gender` variable indicates the gender of the name used on the résumé. Recall that this data does not pertain to 24 actual men and 24 actual women, but rather 48 identical résumés of which 24 were assigned stereotypically "male" names and 24 were assigned stereotypically "female" names.

Let's perform an exploratory data analysis of the relationship between the two categorical variables `decision` and `gender`. Recall that we saw in Subsection 2.8.3 that one way we can visualize such a relationship is by using a stacked barplot.

```
ggplot(promotions, aes(x = gender, fill = decision)) +
  geom_bar() +
  labs(x = "Gender of name on résumé")
```

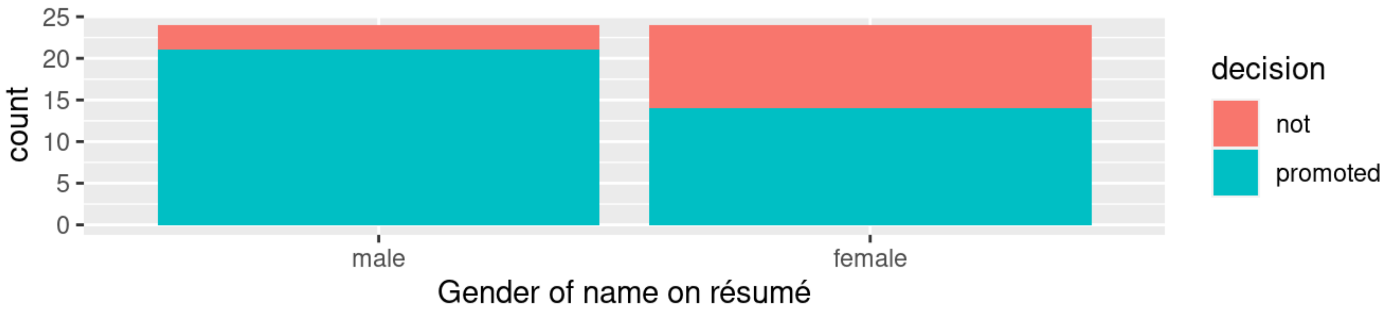


FIGURE 9.1: Barplot relating gender to promotion decision.

Observe in Figure 9.1 that it appears that résumés with female names were much less likely to be accepted for promotion. Let's quantify these promotion rates by computing the proportion of résumés accepted for promotion for each group using the `dplyr` package for data wrangling.

Note the use of the `tally()` function here which is a shortcut for `summarize(n = n())` to get counts.

```
promotions %>%
  group_by(gender, decision) %>%
  tally()
```

```
# A tibble: 4 × 3
# Groups:   gender [2]
  gender decision     n
  <fct>  <fct>    <int>
1 male   not          3
2 male   promoted    21
3 female not         10
4 female promoted    14
```

So of the 24 résumés with male names, 21 were selected for promotion, for a proportion of 21/24 = 0.875 = 87.5%. On the other hand, of the 24 résumés with female names, 14 were selected for promotion, for a proportion of 14/24 = 0.583 = 58.3%. Comparing these two rates of promotion, it appears that résumés with male names were selected for promotion at a rate 0.875 - 0.583 = 0.292 = 29.2% higher than résumés with female names. This is suggestive of an advantage for résumés with a male name on it.

The question is, however, does this provide *conclusive* evidence that there is gender discrimination in promotions at banks? Could a difference in promotion rates of 29.2% still occur by chance, even in a hypothetical world where no gender-based discrimination existed? In other words, what is the role of *sampling variation* in this hypothesized world? To answer this question, we'll again rely on a computer to run *simulations*.

9.1.2 Shuffling once

First, try to imagine a hypothetical universe where no gender discrimination in promotions existed. In such a hypothetical universe, the gender of an applicant would have no bearing on their chances of promotion. Bringing things back to our `promotions` data frame, the `gender` variable would thus be an irrelevant label. If these `gender` labels were irrelevant, then we could randomly reassign them by "shuffling" them to no consequence!

To illustrate this idea, let's narrow our focus to 6 arbitrarily chosen résumés of the 48 in Table 9.1. The `decision` column shows that 3 résumés resulted in promotion while 3 didn't. The `gender` column shows what the original gender of the résumé name was.

However, in our hypothesized universe of no gender discrimination, gender is irrelevant and thus it is of no consequence to randomly "shuffle" the values of `gender`. The `shuffled_gender` column shows one such possible random shuffling. Observe in the fourth column how the number of male and female names remains the same at 3 each, but they are now listed in a different order.

TABLE 9.1: One example of shuffling gender variable

résumé number	decision	gender	shuffled gender
1	not	male	male
2	not	female	male
3	not	female	female
4	promoted	male	female
5	promoted	male	female
6	promoted	female	male

Again, such random shuffling of the gender label only makes sense in our hypothesized universe of no gender discrimination. How could we extend this shuffling of the gender variable to all 48 résumés by hand? One way would be by using standard deck of 52 playing cards, which we display in Figure 9.2.

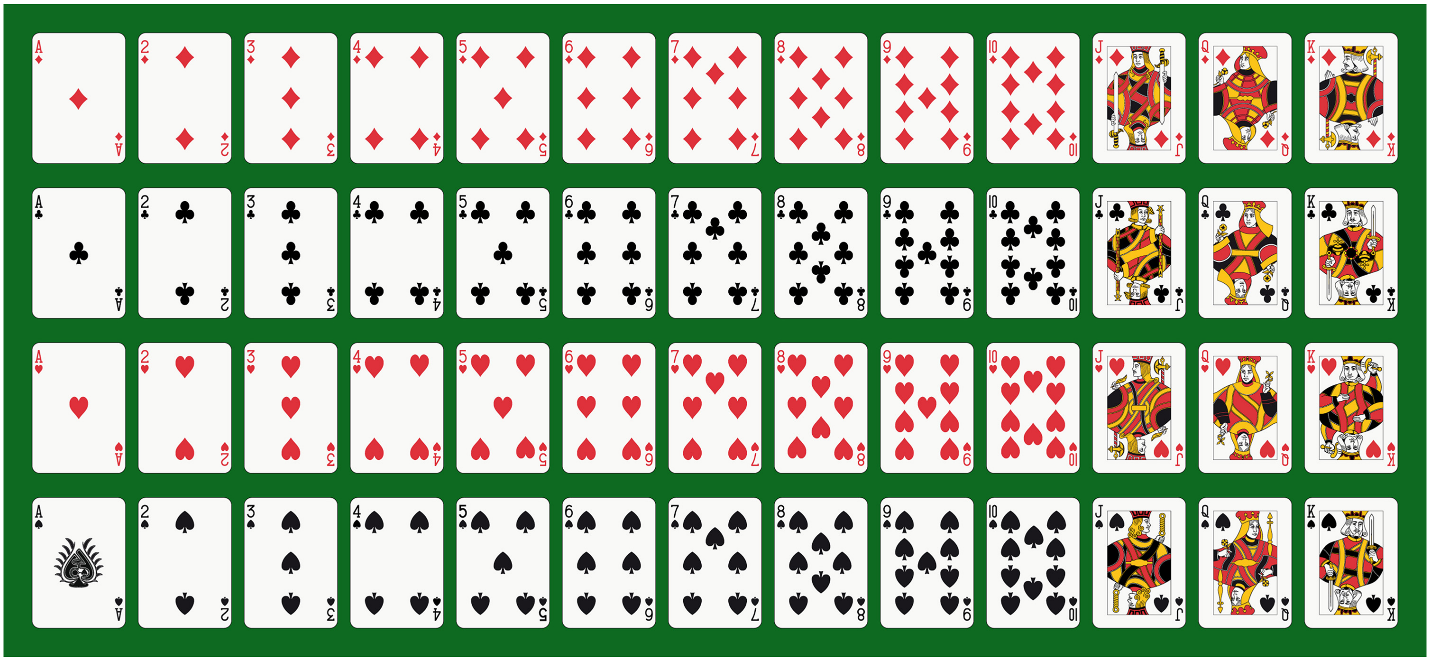


FIGURE 9.2: Standard deck of 52 playing cards.

Since half the cards are red (diamonds and hearts) and the other half are black (spades and clubs), by removing two red cards and two black cards, we would end up with 24 red cards and 24 black cards. After shuffling these 48 cards as seen in Figure 9.3, we can flip the cards over one-by-one, assigning "male" for each red card and "female" for each black card.

FIGURE 9.3: Shuffling a deck of cards.

We’ve saved one such shuffling in the `promotions_shuffled` data frame of the `moderndive` package. If you compare the original `promotions` and the shuffled `promotions_shuffled` data frames, you’ll see that while the `decision` variable is identical, the `gender` variable has changed.

Let’s repeat the same exploratory data analysis we did for the original `promotions` data on our `promotions_shuffled` data frame. Let’s create a barplot visualizing the relationship between `decision` and the new shuffled `gender` variable and compare this to the original unshuffled version in Figure 9.4.

```
ggplot(promotions_shuffled,
       aes(x = gender, fill = decision)) +
  geom_bar() +
  labs(x = "Gender of résumé name")
```

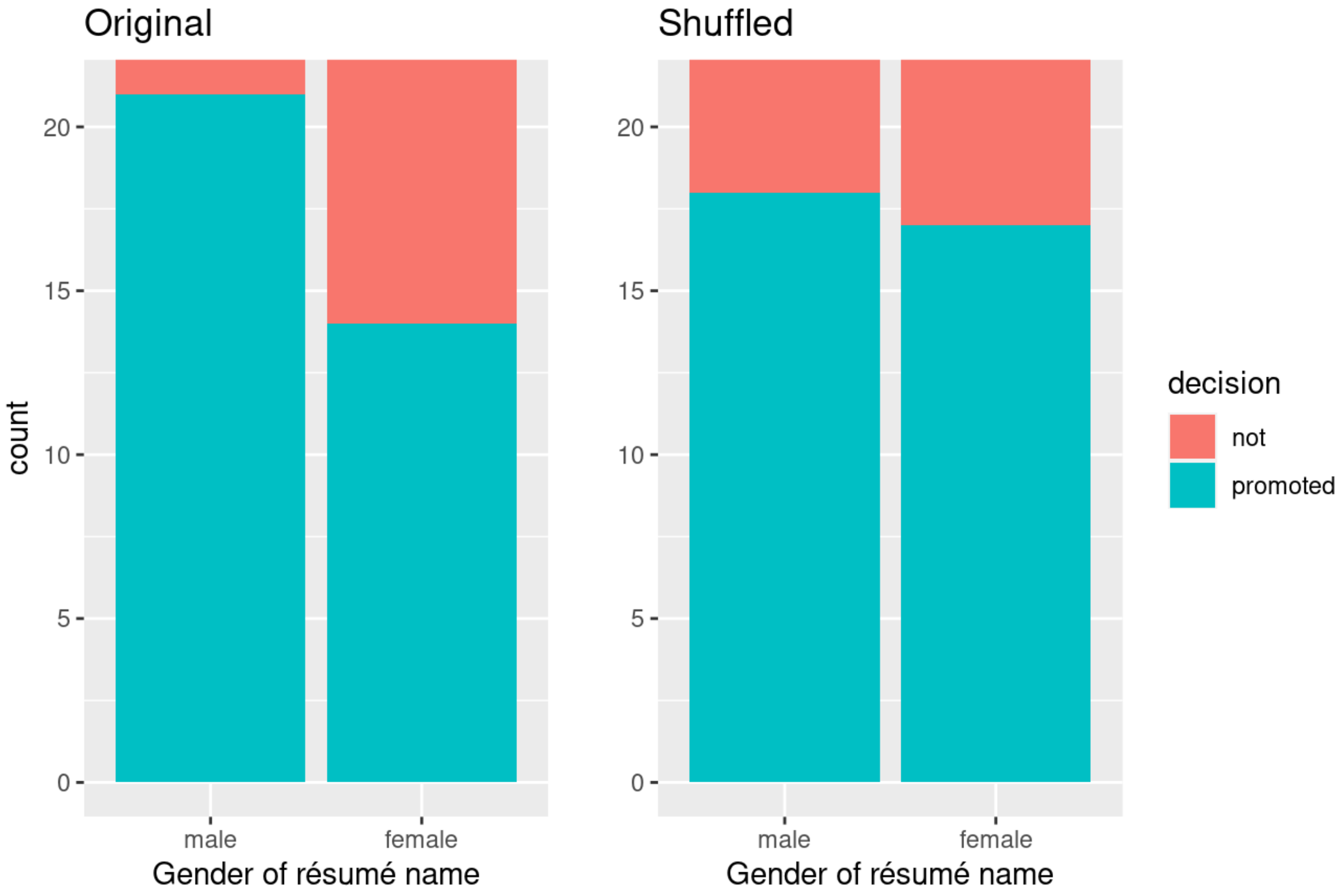


FIGURE 9.4: Barplots of relationship of promotion with gender (left) and shuffled gender (right).

It appears the difference in “male names” versus “female names” promotion rates is now different. Compared to the original data in the left barplot, the new “shuffled” data in the right barplot has promotion rates that are much more similar.

Let’s also compute the proportion of résumés accepted for promotion for each group:

```
promotions_shuffled %>%
  group_by(gender, decision) %>%
  tally() # Same as summarize(n = n())
```

```
# A tibble: 4 × 3
# Groups:   gender [2]
  gender decision     n
  <fct>  <fct>    <int>
1 male   not          6
2 male   promoted    18
3 female not          7
4 female promoted    17
```

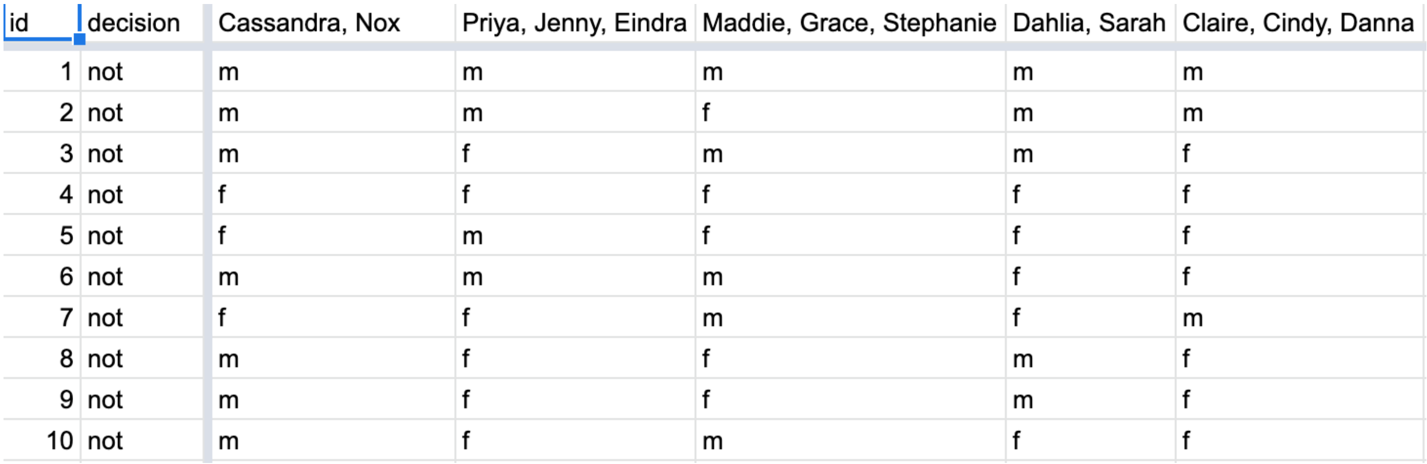
So in this hypothetical universe of no discrimination, $18/24 = 0.75 = 75\%$ of “male” résumés were selected for promotion. On the other hand, $17/24 = 0.708 = 70.8\%$ of “female” résumés were selected for promotion.

Let’s next compare these two values. It appears that résumés with stereotypically male names were selected for promotion at a rate that was $0.75 - 0.708 = 0.042 = 4.2\%$ different than résumés with stereotypically female names.

Observe how this difference in rates is not the same as the difference in rates of 0.292 = 29.2% we originally observed. This is once again due to *sampling variation*. How can we better understand the effect of this sampling variation? By repeating this shuffling several times!

9.1.3 Shuffling 16 times

We recruited 16 groups of our friends to repeat this shuffling exercise. They recorded these values in a shared spreadsheet; we display a snapshot of the first 10 rows and 5 columns in Figure 9.5.



id	decision	Cassandra, Nox	Priya, Jenny, Eindra	Maddie, Grace, Stephanie	Dahlia, Sarah	Claire, Cindy, Danna
1	not	m	m	m	m	m
2	not	m	m	f	m	m
3	not	m	f	m	m	f
4	not	f	f	f	f	f
5	not	f	m	f	f	f
6	not	m	m	m	f	f
7	not	f	f	m	f	m
8	not	m	f	f	m	f
9	not	m	f	f	m	f
10	not	m	f	m	f	f

FIGURE 9.5: Snapshot of shared spreadsheet of shuffling results (m for male, f for female).

For each of these 16 columns of *shuffles*, we computed the difference in promotion rates, and in Figure 9.6 we display their distribution in a histogram. We also mark the observed difference in promotion rate that occurred in real life of 0.292 = 29.2% with a dark line.

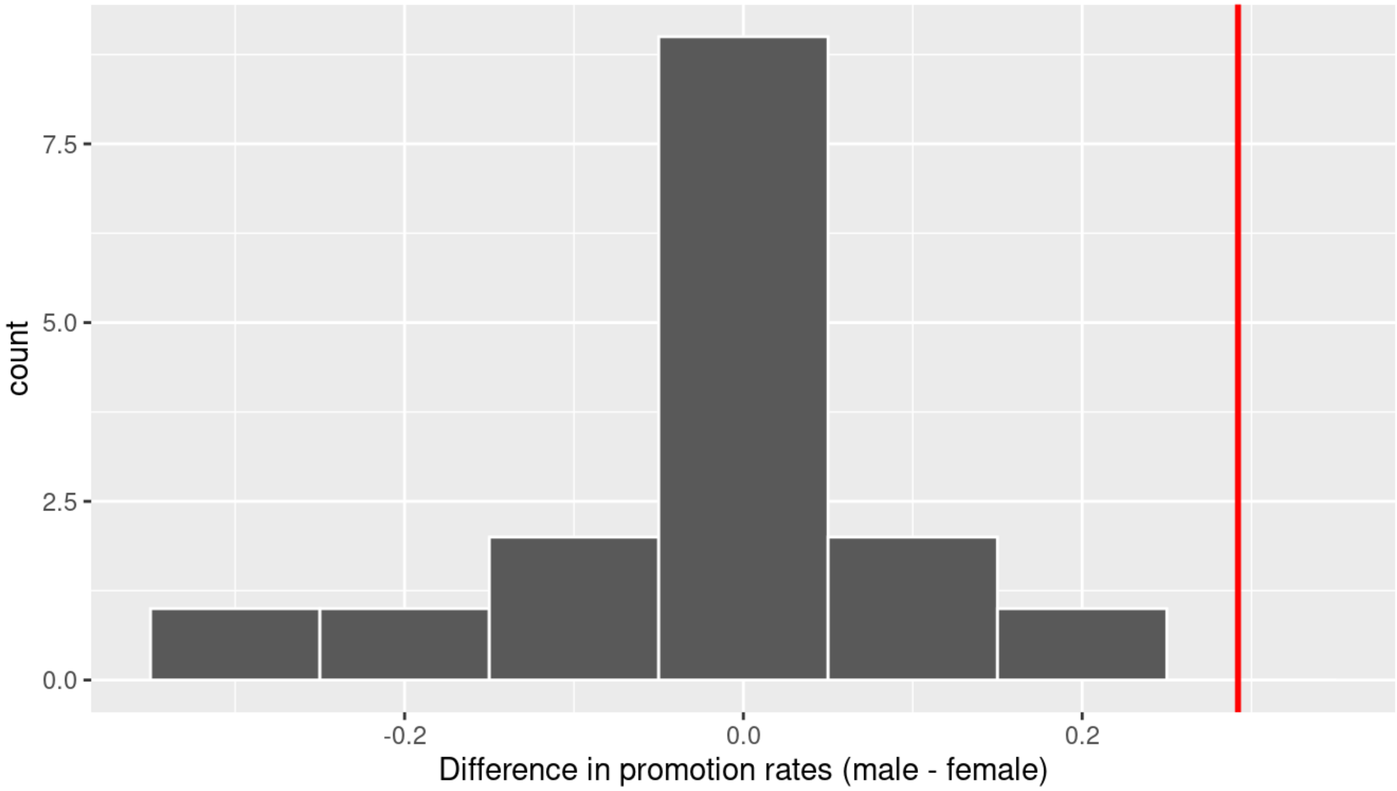


FIGURE 9.6: Distribution of shuffled differences in promotions.

Before we discuss the distribution of the histogram, we emphasize the key thing to remember: this histogram represents differences in promotion rates that one would observe in our *hypothesized universe* of no gender discrimination.

Observe first that the histogram is roughly centered at 0. Saying that the difference in promotion rates is 0 is equivalent to saying that both genders had the same promotion rate. In other words, the center of these 16 values is consistent with what we would expect in our hypothesized universe of no gender discrimination.

However, while the values are centered at 0, there is variation about 0. This is because even in a hypothesized universe of no gender discrimination, you will still likely observe small differences in promotion rates because of chance *sampling variation*. Looking at the histogram in Figure 9.6, such differences could even be as extreme as -0.292 or 0.208.

Turning our attention to what we observed in real life: the difference of 0.292 = 29.2% is marked with a vertical dark line. Ask yourself: in a hypothesized world of no gender discrimination, how likely would it be that we observe this difference? While opinions here may differ, in our opinion not often! Now ask yourself: what do these results say about our hypothesized universe of no gender discrimination?

9.1.4 What did we just do?

What we just demonstrated in this activity is the statistical procedure known as *hypothesis testing* using a *permutation test*. The term “permutation” is the mathematical term for “shuffling”: taking a series of values and reordering them randomly, as you did with the playing cards.

In fact, permutations are another form of *resampling*, like the bootstrap method you performed in Chapter 8. While the bootstrap method involves resampling *with* replacement, permutation methods involve resampling *without* replacement.

Think of our exercise involving the slips of paper representing pennies and the hat in Section 8.1: after sampling a penny, you put it back in the hat. Now think of our deck of cards. After drawing a card, you laid it out in front of you, recorded the color, and then you *did not* put it back in the deck.

In our previous example, we tested the validity of the hypothesized universe of no gender discrimination. The evidence contained in our observed sample of 48 résumés was somewhat inconsistent with our hypothesized universe. Thus, we would be inclined to *reject* this hypothesized universe and declare that the evidence suggests there is gender discrimination.

Recall our case study on whether yawning is contagious from Section 8.6. The previous example involves inference about an unknown difference of population proportions as well. This time, it will be $p_m - p_f$, where p_m is the population proportion of résumés with male names being recommended for promotion and p_f is the equivalent for résumés with female names. Recall that this is one of the scenarios for inference we've seen so far in Table 9.2.

TABLE 9.2: Scenarios of sampling for inference

Scenario	Population parameter	Notation	Point estimate	Symbol(s)
1	Population proportion	p	Sample proportion	$\hat{p}$
2	Population mean	μ	Sample mean	$\overline{x}$ or $\widehat{\mu}$
3	Difference in population proportions	$p_1 - p_2$	Difference in sample proportions	$\hat{p}_1 - \hat{p}_2$

So, based on our sample of n_m = 24 "male" applicants and n_w = 24 "female" applicants, the *point estimate* for $p_m - p_f$ is the *difference in sample proportions* $\hat{p}_m - \hat{p}_f$ = 0.875 - 0.583 = 0.292 = 29.2%. This difference in favor of "male" résumés of 0.292 is greater than 0, suggesting discrimination in favor of men.

However, the question we asked ourselves was "is this difference meaningfully greater than 0?". In other words, is that difference indicative of true discrimination, or can we just attribute it to *sampling variation*? Hypothesis testing allows us to make such distinctions.

9.2 Understanding hypothesis tests

Much like the terminology, notation, and definitions relating to sampling you saw in Section 7.3, there are a lot of terminology, notation, and definitions related to hypothesis testing as well. Learning these may seem like a very daunting task at first. However, with practice, practice, and more practice, anyone can master them.

First, a **hypothesis** is a statement about the value of an unknown population parameter. In our résumé activity, our population parameter of interest is the difference in population proportions $p_m - p_f$. Hypothesis tests can involve any of the population parameters in Table 7.5 of the five inference scenarios we'll cover in this book and also more advanced types we won't cover here.

Second, a **hypothesis test** consists of a test between two competing hypotheses: (1) a **null hypothesis** H_0 (pronounced "H-naught") versus (2) an **alternative hypothesis** H_A (also denoted H_1).

Generally the null hypothesis is a claim that there is "no effect" or "no difference of interest." In many cases, the null hypothesis represents the status quo or a situation that nothing interesting is happening. Furthermore, generally the alternative hypothesis is the claim the experimenter or researcher wants to establish or find evidence to support. It is viewed as a "challenger" hypothesis to the null hypothesis H_0. In our résumé activity, an appropriate hypothesis test would be:

$$\begin{aligned} H_0 &: \text{men and women are promoted at the same rate} \\ \text{vs } H_A &: \text{men are promoted at a higher rate than women} \end{aligned}$$

Note some of the choices we have made. First, we set the null hypothesis H_0 to be that there is no difference in promotion rate and the "challenger" alternative hypothesis H_A to be that there is a difference. While it would not be wrong in principle to reverse the two, it is a convention in statistical inference that the null hypothesis is set to reflect a "null" situation where "nothing is going on." As we discussed earlier, in this case, H_0 corresponds to there being no difference in promotion rates. Furthermore, we set H_A to be that men are promoted at a *higher* rate, a subjective choice reflecting a prior suspicion we have that this is the case. We call such alternative hypotheses *one-sided alternatives*. If someone else however does not share such suspicions and only wants to investigate that there is a difference, whether higher or lower, they would set what is known as a *two-sided alternative*.

We can re-express the formulation of our hypothesis test using the mathematical notation for our population parameter of interest, the difference in population proportions $p_m - p_f$:

$$\begin{aligned} H_0 &: p_m - p_f = 0 \\ \text{vs } H_A &: p_m - p_f > 0 \end{aligned}$$

Observe how the alternative hypothesis H_A is one-sided with $p_m - p_f > 0$. Had we opted for a two-sided alternative, we would have set $p_m - p_f \neq 0$. To keep things simple for now, we'll stick with the simpler one-sided alternative. We'll present an example of a two-sided alternative in Section 9.5.

Third, a **test statistic** is a *point estimate/sample statistic* formula used for hypothesis testing. Note that a sample statistic is merely a summary statistic based on a sample of observations. Recall we saw in Section 3.3 that a summary statistic takes in many values and returns only one. Here, the samples would be the n_m = 24 résumés with male names and the n_f = 24 résumés with female names. Hence, the point estimate of interest is the difference in sample proportions $\hat{p}_m - \hat{p}_f$.

Fourth, the **observed test statistic** is the value of the test statistic that we observed in real life. In our case, we computed this value using the data saved in the `promotions` data frame. It was the observed difference of $\hat{p}_m - \hat{p}_f = 0.875 - 0.583 = 0.292 = 29.2\%$ in favor of résumés with male names.

Fifth, the **null distribution** is the sampling distribution of the test statistic *assuming the null hypothesis H_0 is true*. Ooof! That's a long one! Let's unpack it slowly. The key to understanding the null distribution is that the null hypothesis H_0 is *assumed* to be true. We're not saying that H_0 is true at this point, we're only assuming it to be true for hypothesis testing purposes. In our case, this corresponds to our hypothesized universe of no gender discrimination in promotion rates. Assuming the null hypothesis H_0, also stated as "Under H_0," how does the test statistic vary due to sampling variation? In our case, how will the difference in sample proportions $\hat{p}_m - \hat{p}_f$ vary due to sampling under H_0? Recall from Subsection 7.3.2 that distributions displaying how point estimates vary due to sampling variation are called *sampling distributions*. The only additional thing to keep in mind about null distributions is that they are sampling distributions *assuming the null hypothesis H_0 is true*.

In our case, we previously visualized a null distribution in Figure 9.6, which we re-display in Figure 9.7 using our new notation and terminology. It is the distribution of the 16 differences in sample proportions our friends computed *assuming* a hypothetical universe of no gender discrimination. We also mark the value of the observed test statistic of 0.292 with a vertical line.

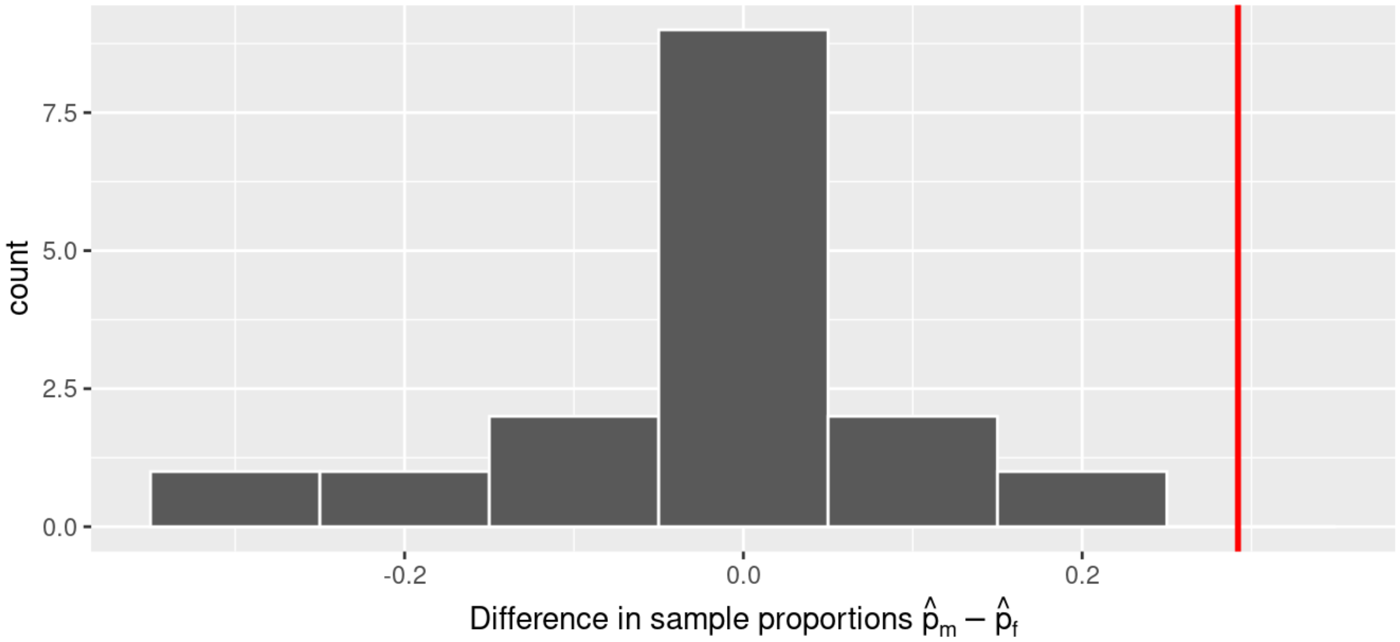


FIGURE 9.7: Null distribution and observed test statistic.

Sixth, the p**-value** is the probability of obtaining a test statistic just as extreme or more extreme than the observed test statistic *assuming the null hypothesis* H_0 *is true*. Double ooof! Let's unpack this slowly as well. You can think of the p-value as a quantification of "surprise": assuming H_0 is true, how surprised are we with what we observed? Or in our case, in our hypothesized universe of no gender discrimination, how surprised are we that we observed a difference in promotion rates of 0.292 from our collected samples assuming H_0 is true? Very surprised? Somewhat surprised?

The p-value quantifies this probability, or in the case of our 16 differences in sample proportions in Figure 9.7, what proportion had a more "extreme" result? Here, extreme is defined in terms of the alternative hypothesis H_A that "male" applicants are promoted at a higher rate than "female" applicants. In other words, how often was the discrimination in favor of men *even more* pronounced than $0.875 - 0.583 = 0.292 = 29.2\%$?

In this case, 0 times out of 16, we obtained a difference in proportion greater than or equal to the observed difference of 0.292 = 29.2%. A very rare (in fact, not occurring) outcome! Given the rarity of such a pronounced difference in promotion rates in our hypothesized universe of no gender discrimination, we're inclined to *reject* our hypothesized universe. Instead, we favor the hypothesis stating there is discrimination in favor of the "male" applicants. In other words, we reject H_0 in favor of H_A.

Seventh and lastly, in many hypothesis testing procedures, it is commonly recommended to set the **significance level** of the test beforehand. It is denoted by the Greek letter α (pronounced "alpha"). This value acts as a cutoff on the p-value, where if the p-value falls below α, we would "reject the null hypothesis H_0."

Alternatively, if the p-value does not fall below α, we would "fail to reject H_0." Note the latter statement is not quite the same as saying we "accept H_0." This distinction is rather subtle and not immediately obvious. So we'll revisit it later in Section 9.4.

While different fields tend to use different values of α, some commonly used values for α are 0.1, 0.01, and 0.05; with 0.05 being the choice people often make without putting much thought into it. We'll talk more about α significance levels in Section 9.4, but first let's fully conduct the hypothesis test corresponding to our promotions activity using the `infer` package.

9.3 Conducting hypothesis tests

In Section 8.4, we showed you how to construct confidence intervals. We first illustrated how to do this using `dplyr` data wrangling verbs and the `rep_sample_n()` function from Subsection 7.2.3 which we used as a virtual shovel. In particular, we constructed confidence intervals by resampling with replacement by setting the `replace = TRUE` argument to the `rep_sample_n()` function.

We then showed you how to perform the same task using the `infer` package workflow. While both workflows resulted in the same bootstrap distribution from which we can construct confidence intervals, the `infer` package workflow emphasizes each of the steps in the overall process in Figure 9.8. It does so using function names that are intuitively named with verbs:

1. `specify()` the variables of interest in your data frame.
2. `generate()` replicates of bootstrap resamples with replacement.
3. `calculate()` the summary statistic of interest.
4. `visualize()` the resulting bootstrap distribution and confidence interval.

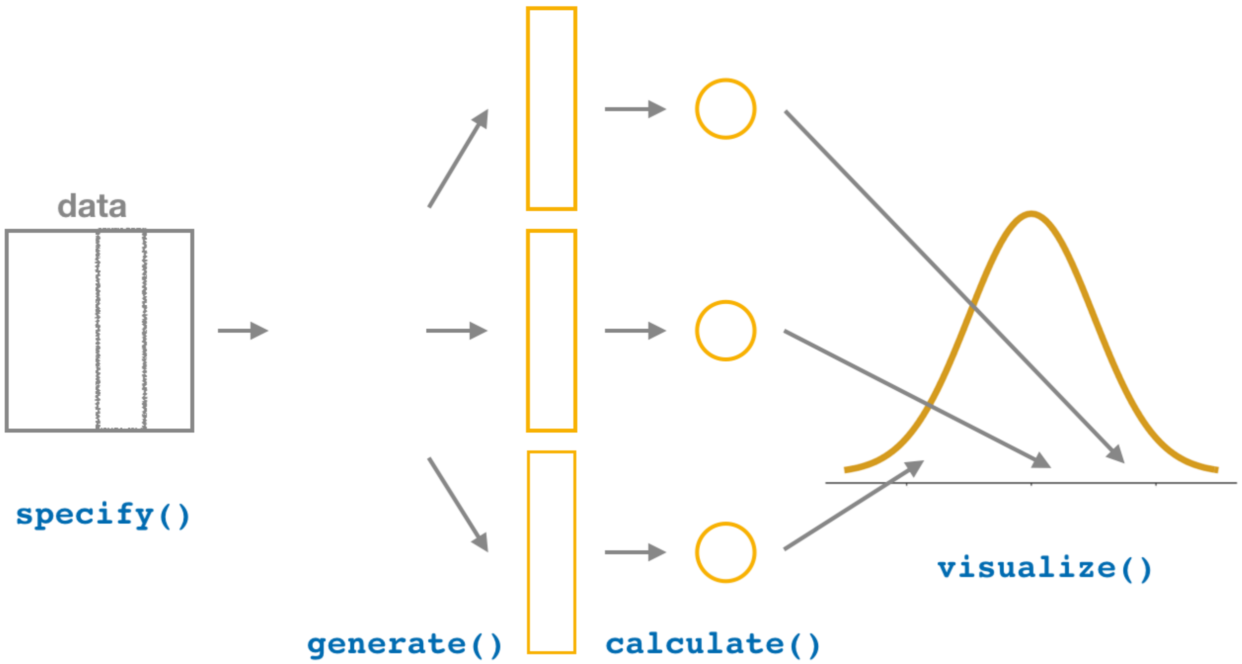


FIGURE 9.8: Confidence intervals with the infer package.

In this section, we'll now show you how to seamlessly modify the previously seen `infer` code for constructing confidence intervals to conduct hypothesis tests. You'll notice that the basic outline of the workflow is almost identical, except for an additional `hypothesize()` step between the `specify()` and `generate()` steps, as can be seen in Figure 9.9.

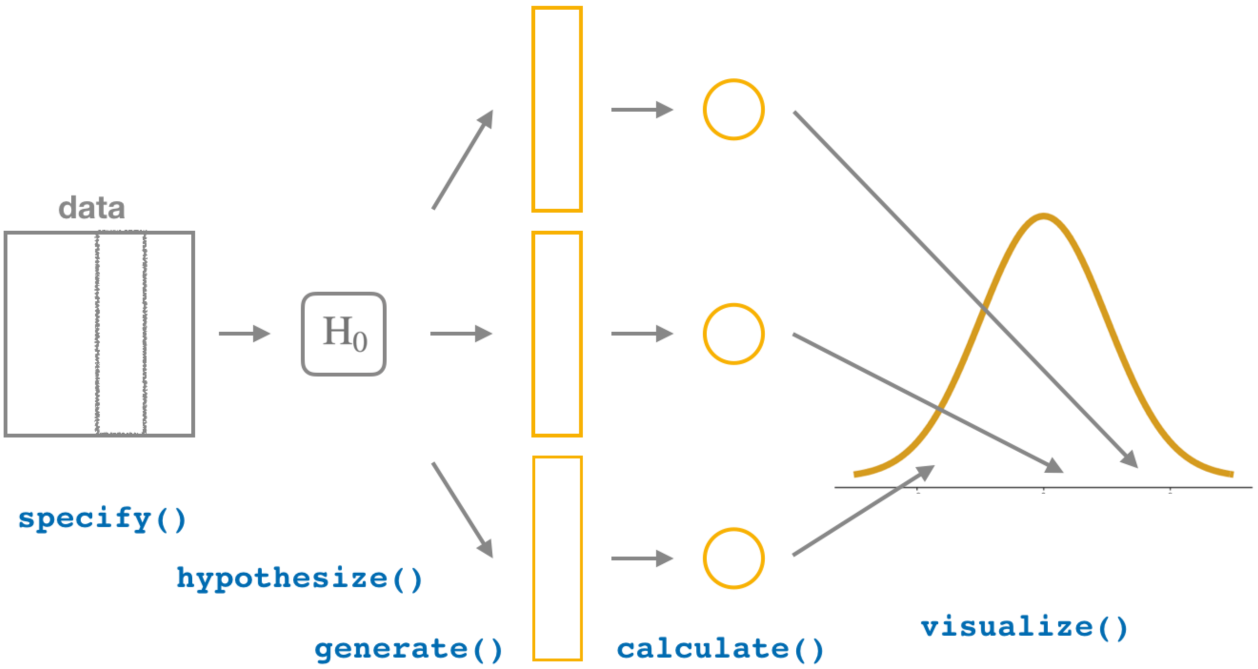


FIGURE 9.9: Hypothesis testing with the infer package.

Furthermore, we’ll use a pre-specified significance level $\alpha = 0.05$ for this hypothesis test. Let’s leave discussion on the choice of this α value until later on in Section 9.4.

9.3.1 `infer` package workflow

1. `specify` variables

Recall that we use the `specify()` verb to specify the response variable and, if needed, any explanatory variables for our study. In this case, since we are interested in any potential effects of gender on promotion decisions, we set `decision` as the response variable and `gender` as the explanatory variable. We do so using `formula = response ~ explanatory` where `response` is the name of the response variable in the data frame and `explanatory` is the name of the explanatory variable. So in our case it is `decision ~ gender` .

Furthermore, since we are interested in the proportion of résumés `"promoted"` , and not the proportion of résumés `not` promoted, we set the argument `success` to `"promoted"` .

```
promotions %>%
  specify(formula = decision ~ gender, success = "promoted")
```

```
Response: decision (factor)
Explanatory: gender (factor)
# A tibble: 48 × 2
   decision gender
   <fct>    <fct>
 1 promoted male
 2 promoted male
 3 promoted male
 4 promoted male
 5 promoted male
 6 promoted male
 7 promoted male
 8 promoted male
 9 promoted male
10 promoted male
# ℹ 38 more rows
```

Again, notice how the `promotions` data itself doesn't change, but the `Response: decision (factor)` and `Explanatory: gender (factor)` *meta-data* do. This is similar to how the `group_by()` verb from `dplyr` doesn't change the data, but only adds "grouping" meta-data, as we saw in Section 3.4.

2. `hypothesize` the null

In order to conduct hypothesis tests using the `infer` workflow, we need a new step not present for confidence intervals: `hypothesize()` . Recall from Section 9.2 that our hypothesis test was

$$\begin{aligned} H_0 &: p_m - p_f = 0 \\ \text{vs. } H_A &: p_m - p_f > 0 \end{aligned}$$

In other words, the null hypothesis H_0 corresponding to our "hypothesized universe" stated that there was no difference in gender-based discrimination rates. We set this null hypothesis H_0 in our `infer` workflow using the `null` argument of the `hypothesize()` function to either:

- `"point"` for hypotheses involving a single sample or
- `"independence"` for hypotheses involving two samples.

In our case, since we have two samples (the résumés with “male” and “female” names), we set `null = "independence"` .

```
promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  hypothesize(null = "independence")
```

```
Response: decision (factor)
Explanatory: gender (factor)
Null Hypothesis: independence
# A tibble: 48 × 2
   decision gender
   <fct>    <fct>
 1 promoted male
 2 promoted male
 3 promoted male
 4 promoted male
 5 promoted male
 6 promoted male
 7 promoted male
 8 promoted male
 9 promoted male
10 promoted male
# ℹ 38 more rows
```

Again, the data has not changed yet. This will occur at the upcoming `generate()` step; we’re merely setting meta-data for now.

Where do the terms `"point"` and `"independence"` come from? These are two technical statistical terms. The term “point” relates from the fact that for a single group of observations, you will test the value of a single point. Going back to the pennies example from Chapter 8, say we wanted to test if the mean year of all US pennies was equal to 1993 or not. We would be testing the value of a “point” μ, the mean year of *all* US pennies, as follows

$$H_0 : \mu = 1993$$
$$\text{vs } H_A : \mu \neq 1993$$

The term “independence” relates to the fact that for two groups of observations, you are testing whether or not the response variable is *independent* of the explanatory variable that assigns the groups. In our case, we are testing whether the `decision` response variable is “independent” of the explanatory variable `gender` that assigns each résumé to either of the two groups.

3. `generate` replicates

After we `hypothesize()` the null hypothesis, we `generate()` replicates of “shuffled” datasets assuming the null hypothesis is true. We do this by repeating the shuffling exercise you performed in Section 9.1 several times. Instead of merely doing it 16 times as our groups of friends did, let’s use the computer to repeat this 1000 times by setting `reps = 1000` in the `generate()` function. However, unlike for confidence intervals where we generated replicates using `type = "bootstrap"` resampling with replacement, we’ll now perform shuffles/permutations by setting `type = "permute"` . Recall that shuffles/permutations are a kind of resampling, but unlike the bootstrap method, they involve resampling *without* replacement.

```
promotions_generate <- promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute")
nrow(promotions_generate)
```

```
[1] 48000
```

Observe that the resulting data frame has 48,000 rows. This is because we performed shuffles/permutations for each of the 48 rows 1000 times and $48,000 = 1000 \cdot 48$. If you explore the `promotions_generate` data frame with `View()` , you’ll notice that the variable `replicate` indicates which resample each row belongs to. So it has the value `1` 48 times, the value `2` 48 times, all the way through to the value `1000` 48 times.

4. `calculate` summary statistics

Now that we have generated 1000 replicates of “shuffles” assuming the null hypothesis is true, let’s `calculate()` the appropriate summary statistic for each of our 1000 shuffles. From Section 9.2, point estimates related to hypothesis testing have a specific name: *test statistics*. Since the unknown population parameter of interest is the difference in population proportions $p_m - p_f$, the test statistic here is the difference in sample proportions $\hat{p}_m - \hat{p}_f$.

For each of our 1000 shuffles, we can calculate this test statistic by setting `stat = "diff in props"` . Furthermore, since we are interested in $\hat{p}_m - \hat{p}_f$ we set `order = c("male", "female")` . As we stated earlier, the order of the subtraction does not matter, so long as you stay consistent throughout your analysis and tailor your interpretations accordingly.

Let’s save the result in a data frame called `null_distribution` :

```
null_distribution <- promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("male", "female"))
null_distribution
```

```
# A tibble: 1,000 × 2
   replicate       stat
       <int>      <dbl>
 1         1 -0.0416667
 2         2 -0.125
 3         3 -0.125
 4         4 -0.0416667
 5         5 -0.0416667
 6         6 -0.125
 7         7 -0.125
 8         8 -0.125
 9         9 -0.0416667
10        10 -0.0416667
# ℹ 990 more rows
```

Observe that we have 1000 values of `stat`, each representing one instance of $\hat{p}_m - \hat{p}_f$ in a hypothesized world of no gender discrimination. Observe as well that we chose the name of this data frame carefully: `null_distribution`. Recall once again from Section 9.2 that sampling distributions when the null hypothesis H_0 is assumed to be true have a special name: the *null distribution*.

What was the *observed* difference in promotion rates? In other words, what was the *observed test statistic* $\hat{p}_m - \hat{p}_f$? Recall from Section 9.1 that we computed this observed difference by hand to be 0.875 - 0.583 = 0.292 = 29.2%. We can also compute this value using the previous `infer` code but with the `hypothesize()` and `generate()` steps removed. Let's save this in `obs_diff_prop`:

```
obs_diff_prop <- promotions %>%
  specify(decision ~ gender, success = "promoted") %>%
  calculate(stat = "diff in props", order = c("male", "female"))
obs_diff_prop
```

```
Response: decision (factor)
Explanatory: gender (factor)
# A tibble: 1 × 1
      stat
     <dbl>
1 0.291667
```

5. `visualize` the p-value

The final step is to measure how surprised we are by a promotion difference of 29.2% in a hypothesized universe of no gender discrimination. If the observed difference of 0.292 is highly unlikely, then we would be inclined to reject the validity of our hypothesized universe.

We start by visualizing the *null distribution* of our 1000 values of $\hat{p}_m - \hat{p}_f$ using `visualize()` in Figure 9.10. Recall that these are values of the difference in promotion rates assuming H_0 is true. This corresponds to being in our hypothesized universe of no gender discrimination.

```
visualize(null_distribution, bins = 10)
```

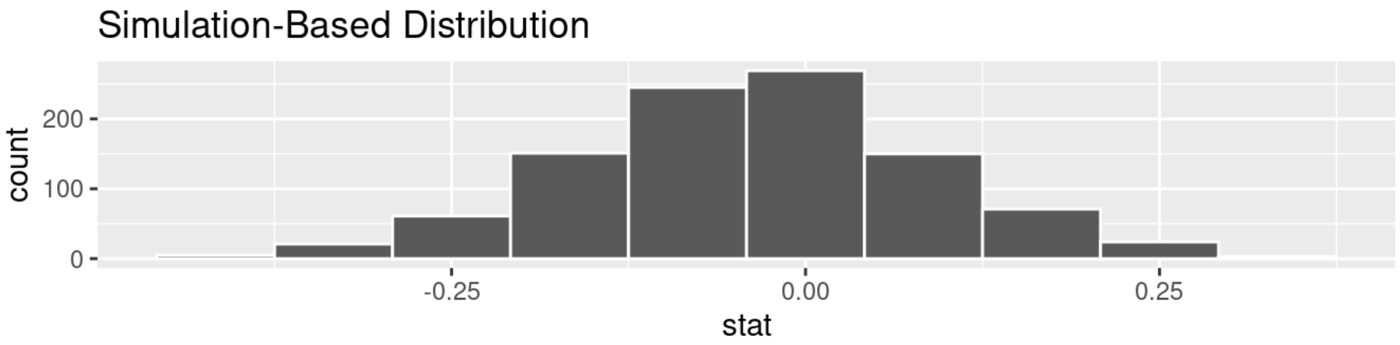


FIGURE 9.10: Null distribution.

Let's now add what happened in real life to Figure 9.10, the observed difference in promotion rates of 0.875 - 0.583 = 0.292 = 29.2%. However, instead of merely adding a vertical line using `geom_vline()` , let's use the `shade_p_value()` function with `obs_stat` set to the observed test statistic value we saved in `obs_diff_prop` .

Furthermore, we'll set the `direction = "right"` reflecting our alternative hypothesis $H_A: p_m - p_f > 0$. Recall our alternative hypothesis H_A is that $p_m - p_f > 0$, stating that there is a difference in promotion rates in favor of résumés with male names. "More extreme" here corresponds to differences that are “bigger” or “more positive” or “more to the right.” Hence we set the `direction` argument of `shade_p_value()` to be `"right"`.

On the other hand, had our alternative hypothesis H_A been the other possible one-sided alternative $p_m - p_f < 0$, suggesting discrimination in favor of résumés with female names, we would’ve set `direction = "left"`. Had our alternative hypothesis H_A been two-sided $p_m - p_f \neq 0$, suggesting discrimination in either direction, we would’ve set `direction = "both"`.

```
visualize(null_distribution, bins = 10) +
  shade_p_value(obs_stat = obs_diff_prop, direction = "right")
```

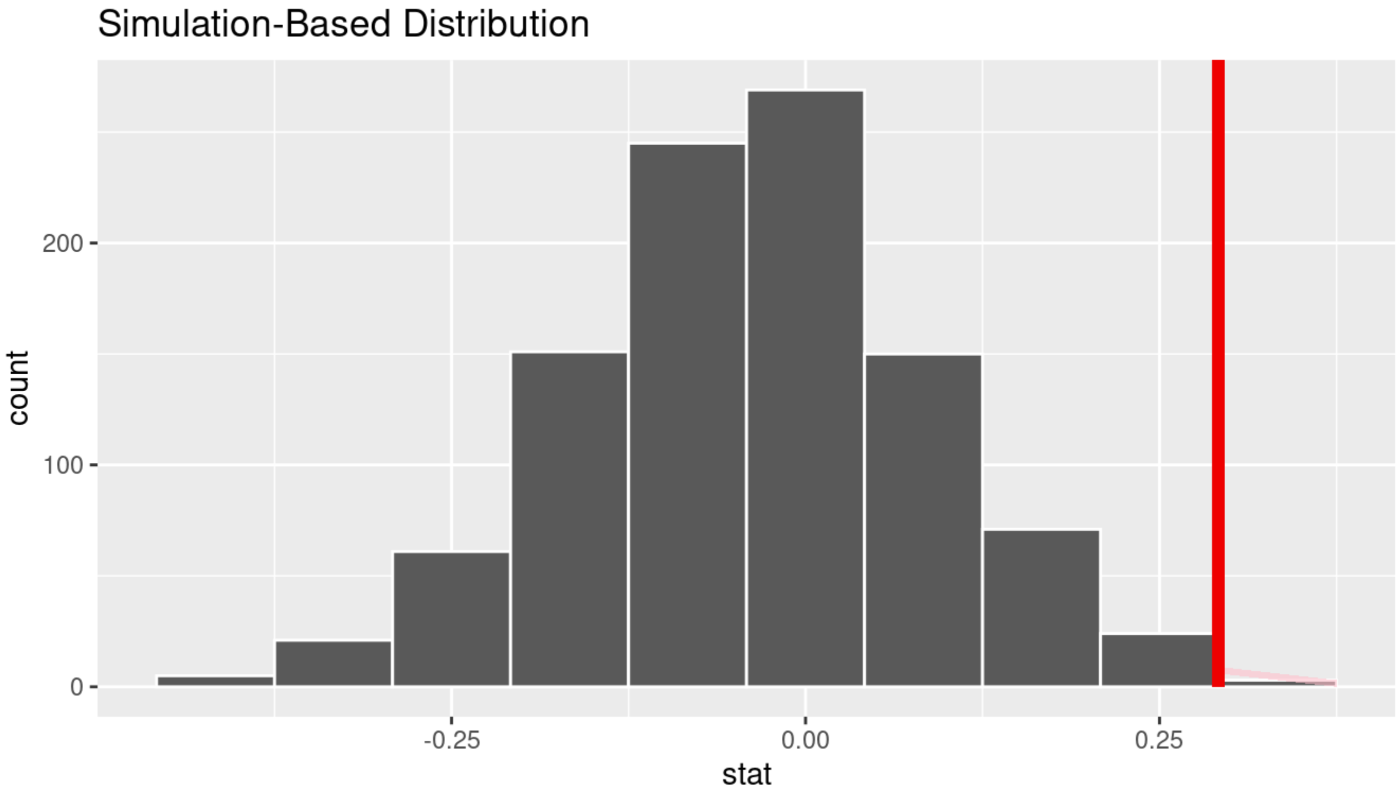


FIGURE 9.11: Shaded histogram to show p-value.

In the resulting Figure 9.11, the solid dark line marks 0.292 = 29.2%. However, what does the shaded-region correspond to? This is the p*-value*. Recall the definition of the p-value from Section 9.2:

> A p-value is the probability of obtaining a test statistic just as or more extreme than the observed test statistic *assuming the null hypothesis* H_0 *is true*.

So judging by the shaded region in Figure 9.11, it seems we would somewhat rarely observe differences in promotion rates of 0.292 = 29.2% or more in a hypothesized universe of no gender discrimination. In other words, the p-value is somewhat small. Hence, we would be inclined to reject this hypothesized universe, or using statistical language we would "reject H_0."

What fraction of the null distribution is shaded? In other words, what is the exact value of the p-value? We can compute it using the `get_p_value()` function with the same arguments as the previous `shade_p_value()` code:

```
null_distribution %>%
  get_p_value(obs_stat = obs_diff_prop, direction = "right")
```

```
# A tibble: 1 × 1
  p_value
    <dbl>
1   0.027
```

Keeping the definition of a p-value in mind, the probability of observing a difference in promotion rates as large as 0.292 = 29.2% due to sampling variation alone in the null distribution is 0.027 = 2.7%. Since this p-value is smaller than our pre-specified significance level α = 0.05, we reject the null hypothesis $H_0 : p_m - p_f = 0$. In other words, this p-value is sufficiently small to reject our hypothesized universe of no gender discrimination. We instead have enough evidence to change our mind in favor of gender discrimination being a likely culprit here. Observe that whether we reject the null hypothesis H_0 or not depends in large part on our choice of significance level α. We'll discuss this more in Subsection 9.4.3.

9.3.2 Comparison with confidence intervals

One of the great things about the `infer` package is that we can jump seamlessly between conducting hypothesis tests and constructing confidence intervals with minimal changes! Recall the code from the previous section that creates the null distribution, which in turn is needed to compute the p-value:

```
null_distribution <- promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in props", order = c("male", "female"))
```

To create the corresponding bootstrap distribution needed to construct a 95% confidence interval for $p_m - p_f$, we only need to make two changes. First, we remove the `hypothesize()` step since we are no longer assuming a null hypothesis H_0 is true. We can do this by deleting or commenting out the `hypothesize()` line of code. Second, we switch the `type` of resampling in the `generate()` step to be `"bootstrap"` instead of `"permute"` .

```
bootstrap_distribution <- promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  # Change 1 - Remove hypothesize():
  # hypothesize(null = "independence") %>%
  # Change 2 - Switch type from "permute" to "bootstrap":
  generate(reps = 1000, type = "bootstrap") %>%
  calculate(stat = "diff in props", order = c("male", "female"))
```

Using this `bootstrap_distribution` , let's first compute the percentile-based confidence intervals, as we did in Section 8.4:

```
percentile_ci <- bootstrap_distribution %>%
  get_confidence_interval(level = 0.95, type = "percentile")
percentile_ci
```

```
# A tibble: 1 × 2
   lower_ci upper_ci
      <dbl>    <dbl>
1 0.0444444 0.538542
```

Using our shorthand interpretation for 95% confidence intervals from Subsection 8.5.2, we are 95% “confident” that the true difference in population proportions $p_m - p_f$ is between (0.044, 0.539). Let’s visualize `bootstrap_distribution` and this percentile-based 95% confidence interval for $p_m - p_f$ in Figure 9.12.

```
visualize(bootstrap_distribution) +
  shade_confidence_interval(endpoints = percentile_ci)
```

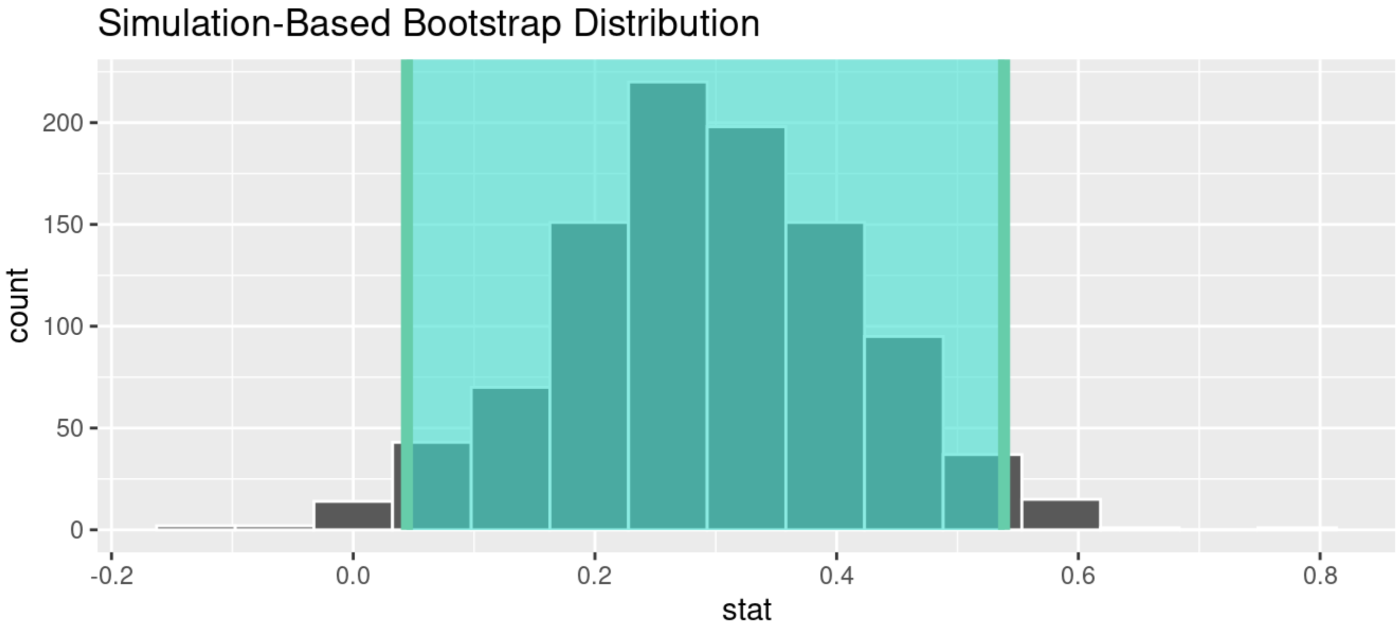


FIGURE 9.12: Percentile-based 95% confidence interval.

Notice a key value that is not included in the 95% confidence interval for $p_m - p_f$: the value 0. In other words, a difference of 0 is not included in our net, suggesting that p_m and p_f are truly different! Furthermore, observe how the entirety of the 95% confidence interval for $p_m - p_f$ lies above 0, suggesting that this difference is in favor of men.

Since the bootstrap distribution appears to be roughly normally shaped, we can also use the standard error method as we did in Section 8.4. In this case, we must specify the `point_estimate` argument as the observed difference in promotion rates 0.292 = 29.2% saved in `obs_diff_prop`. This value acts as the center of the confidence interval.

```
se_ci <- bootstrap_distribution %>%
  get_confidence_interval(level = 0.95, type = "se",
                          point_estimate = obs_diff_prop)
se_ci
```

```
# A tibble: 1 × 2
   lower_ci upper_ci
      <dbl>    <dbl>
1 0.0514129 0.531920
```

Let's visualize `bootstrap_distribution` again, but now the standard error based 95% confidence interval for $p_m - p_f$ in Figure 9.13. Again, notice how the value 0 is not included in our confidence interval, again suggesting that p_m and p_f are truly different!

```
visualize(bootstrap_distribution) +
  shade_confidence_interval(endpoints = se_ci)
```

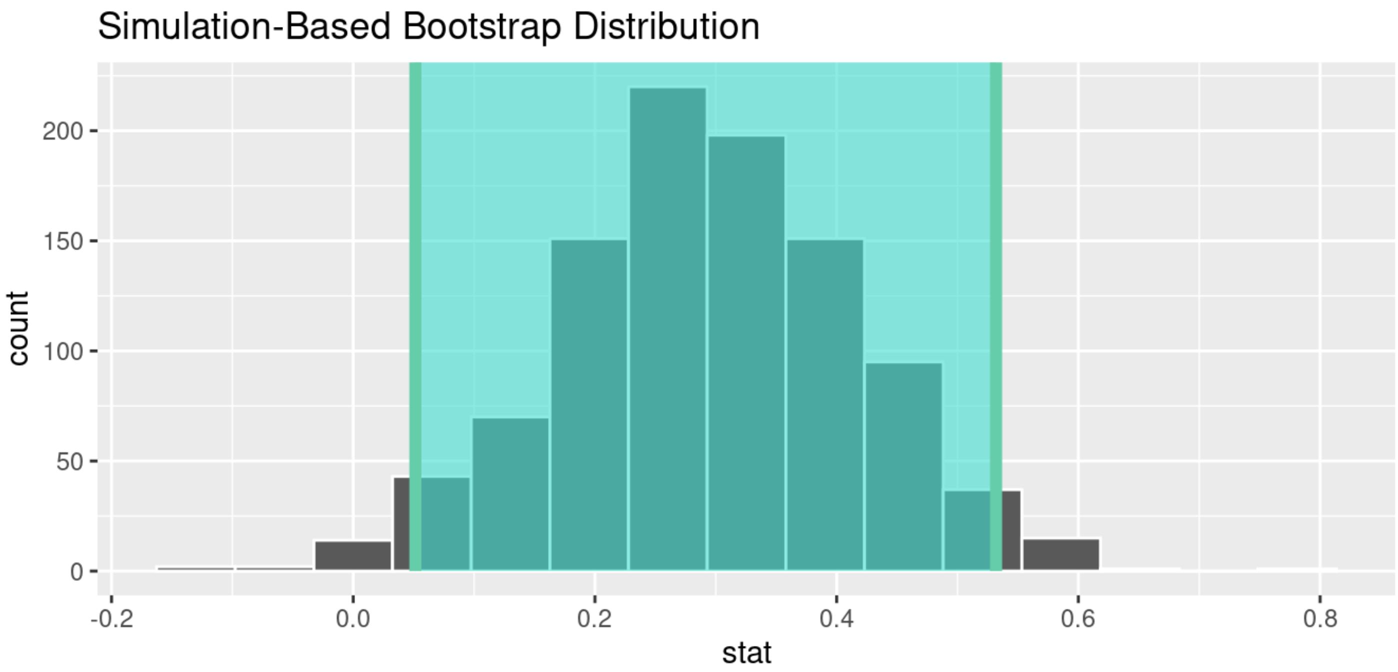


FIGURE 9.13: Standard error-based 95% confidence interval.

Learning check

(LC9.1) Why does the following code produce an error? In other words, what about the response and predictor variables make this not a possible computation with the `infer` package?

```
library(moderndive)
library(infer)
null_distribution_mean <- promotions %>%
  specify(formula = decision ~ gender, success = "promoted") %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("male", "female"))
```

(LC9.2) Why are we relatively confident that the distributions of the sample proportions will be good approximations of the population distributions of promotion proportions for the two genders?

(LC9.3) Using the definition of *p-value*, write in words what the p-value represents for the hypothesis test comparing the promotion rates for males and females.

9.3.3 "There is only one test"

Let's recap the steps necessary to conduct a hypothesis test using the terminology, notation, and definitions related to sampling you saw in Section 9.2 and the `infer` workflow from Subsection 9.3.1:

1. `specify()` the variables of interest in your data frame.
2. `hypothesize()` the null hypothesis H_0. In other words, set a "model for the universe" assuming H_0 is true.
3. `generate()` shuffles assuming H_0 is true. In other words, *simulate* data assuming H_0 is true.
4. `calculate()` the *test statistic* of interest, both for the observed data and your *simulated* data.
5. `visualize()` the resulting *null distribution* and compute the p*-value* by comparing the null distribution to the observed test statistic.

While this is a lot to digest, especially the first time you encounter hypothesis testing, the nice thing is that once you understand this general framework, then you can understand *any* hypothesis test. In a famous blog post, computer scientist Allen Downey called this the "There is only one test" framework, for which he created the flowchart displayed in Figure 9.14.

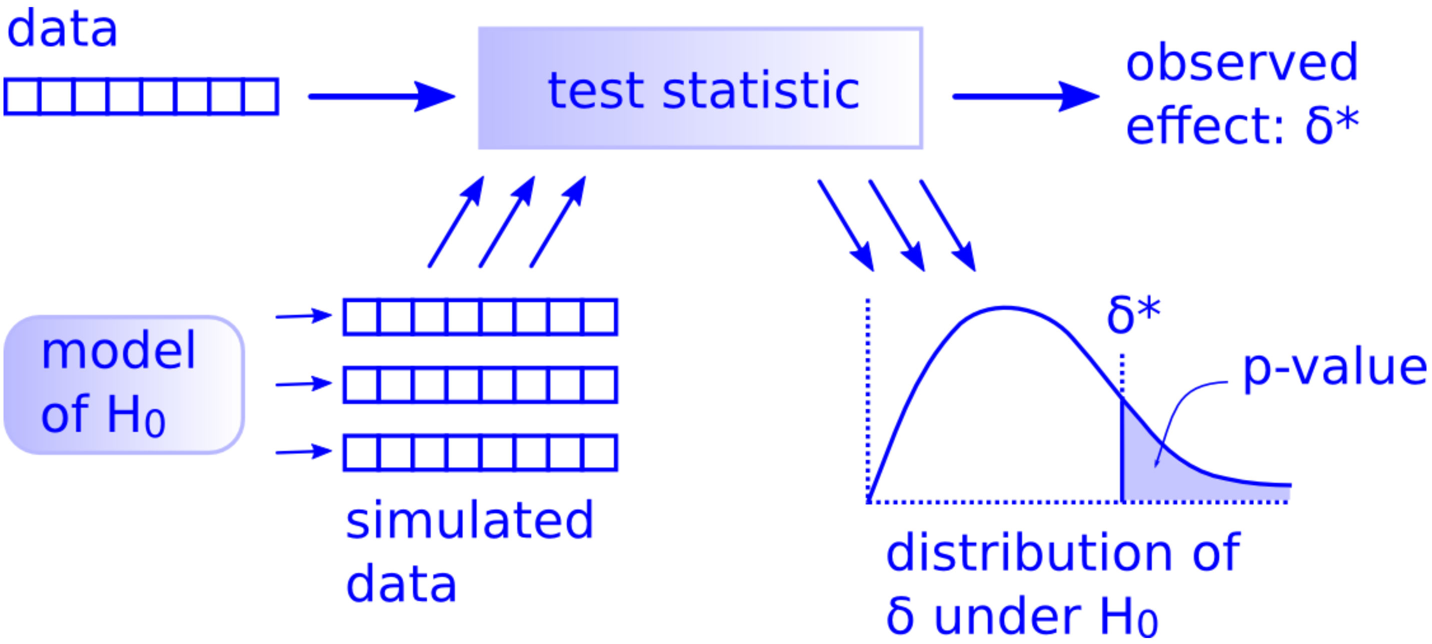


FIGURE 9.14: Allen Downey's hypothesis testing framework.

Notice its similarity with the "hypothesis testing with `infer` " diagram you saw in Figure 9.9. That's because the `infer` package was explicitly designed to match the "There is only one test" framework. So if you can understand the framework, you can easily generalize these ideas for all hypothesis testing scenarios. Whether for population proportions p, population means μ, differences in population proportions $p_1 - p_2$, differences in population means $\mu_1 - \mu_2$, and as you'll see in Chapter 10 on inference for regression, population regression slopes β_1 as well. In fact, it applies more generally even than just these examples to more complicated hypothesis tests and test statistics as well.

Learning check

(LC9.4) Describe in a paragraph how we used Allen Downey's diagram to conclude if a statistical difference existed between the promotion rate of males and females using this study.

9.4 Interpreting hypothesis tests

Interpreting the results of hypothesis tests is one of the more challenging aspects of this method for statistical inference. In this section, we'll focus on ways to help with deciphering the process and address some common misconceptions.

9.4.1 Two possible outcomes

In Section 9.2, we mentioned that given a pre-specified significance level α there are two possible outcomes of a hypothesis test:

- If the p-value is less than α, then we *reject* the null hypothesis H_0 in favor of H_A.
- If the p-value is greater than or equal to α, we *fail to reject* the null hypothesis H_0.

Unfortunately, the latter result is often misinterpreted as "accepting the null hypothesis H_0." While at first glance it may seem that the statements "failing to reject H_0" and "accepting H_0" are equivalent, there actually is a subtle difference. Saying that we "accept the null hypothesis H_0" is equivalent to stating that "we think the null hypothesis H_0 is true." However, saying that we "fail to reject the null hypothesis H_0" is saying something else: "While H_0 might still be false, we don't have enough evidence to say so." In other words, there is an absence of enough proof. However, the absence of proof is not proof of absence.

To further shed light on this distinction, let's use the United States criminal justice system as an analogy. A criminal trial in the United States is a similar situation to hypothesis tests whereby a choice between two contradictory claims must be made about a defendant who is on trial:

1. The defendant is truly either "innocent" or "guilty."
2. The defendant is presumed "innocent until proven guilty."
3. The defendant is found guilty only if there is *strong evidence* that the defendant is guilty. The phrase "beyond a reasonable doubt" is often used as a guideline for determining a cutoff for when enough evidence exists to find the defendant guilty.
4. The defendant is found to be either "not guilty" or "guilty" in the ultimate verdict.

In other words, *not guilty* verdicts are not suggesting the defendant is *innocent*, but instead that "while the defendant may still actually be guilty, there wasn't enough evidence to prove this fact." Now let's make the connection with hypothesis tests:

1. Either the null hypothesis H_0 or the alternative hypothesis H_A is true.
2. Hypothesis tests are conducted assuming the null hypothesis H_0 is true.
3. We reject the null hypothesis H_0 in favor of H_A only if the evidence found in the sample suggests that H_A is true. The significance level α is used as a guideline to set the threshold on just how strong of evidence we require.
4. We ultimately decide to either "fail to reject H_0" or "reject H_0."

So while gut instinct may suggest “failing to reject H_0” and “accepting H_0” are equivalent statements, they are not. “Accepting H_0” is equivalent to finding a defendant innocent. However, courts do not find defendants “innocent,” but rather they find them “not guilty.” Putting things differently, defense attorneys do not need to prove that their clients are innocent, rather they only need to prove that clients are not “guilty beyond a reasonable doubt”.

So going back to our résumés activity in Section 9.3, recall that our hypothesis test was $H_0 : p_m - p_f = 0$ versus $H_A : p_m - p_f > 0$ and that we used a pre-specified significance level of α = 0.05. We found a p-value of 0.027. Since the p-value was smaller than α = 0.05, we rejected H_0. In other words, we found needed levels of evidence in this particular sample to say that H_0 is false at the α = 0.05 significance level. We also state this conclusion using non-statistical language: we found enough evidence in this data to suggest that there was gender discrimination at play.

9.4.2 Types of errors

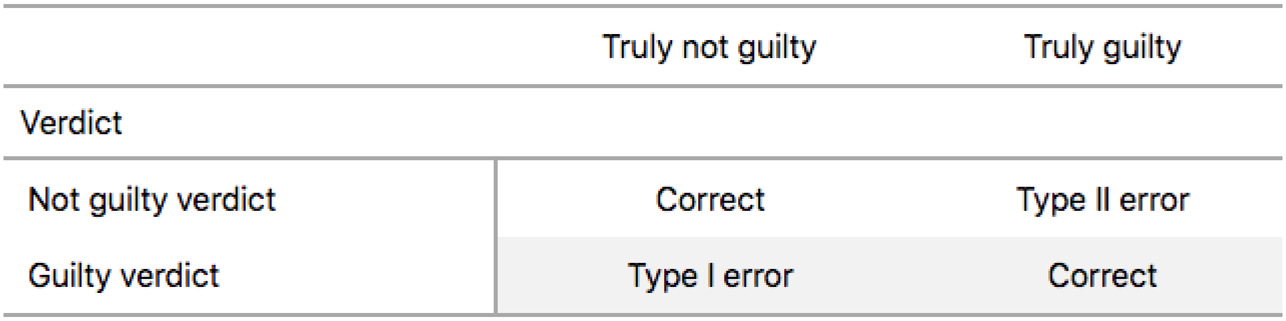
Unfortunately, there is some chance a jury or a judge can make an incorrect decision in a criminal trial by reaching the wrong verdict. For example, finding a truly innocent defendant “guilty”. Or on the other hand, finding a truly guilty defendant “not guilty.” This can often stem from the fact that prosecutors don’t have access to all the relevant evidence, but instead are limited to whatever evidence the police can find.

The same holds for hypothesis tests. We can make incorrect decisions about a population parameter because we only have a sample of data from the population and thus sampling variation can lead us to incorrect conclusions.

There are two possible erroneous conclusions in a criminal trial: either (1) a truly innocent person is found guilty or (2) a truly guilty person is found not guilty. Similarly, there are two possible errors in a hypothesis test: either (1) rejecting H_0 when in fact H_0 is true, called a **Type I error** or (2) failing to reject H_0 when in fact H_0 is false, called a **Type II error**. Another term used for “Type I error” is “false positive,” while another term for “Type II error” is “false negative.”

This risk of error is the price researchers pay for basing inference on a sample instead of performing a census on the entire population. But as we’ve seen in our numerous examples and activities so far, censuses are often very expensive and other times impossible, and thus researchers have no choice but to use a sample. Thus in any hypothesis test based on a sample, we have no choice but to tolerate some chance that a Type I error will be made and some chance that a Type II error will occur.

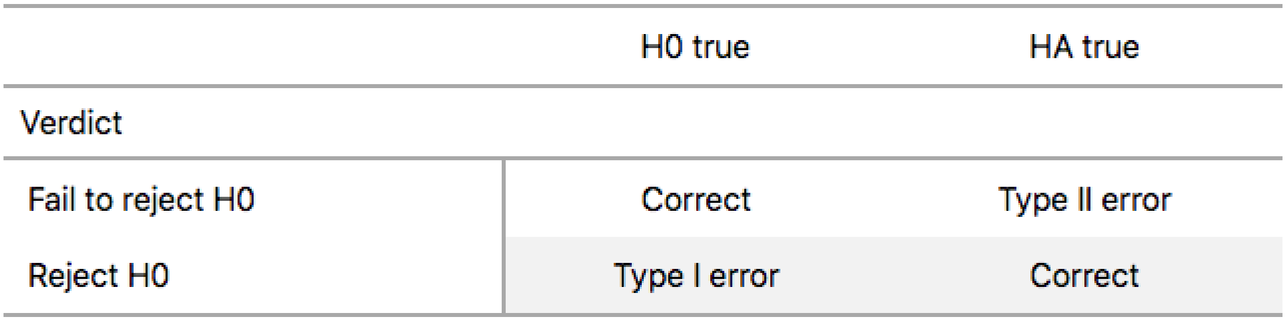
To help understand the concepts of Type I error and Type II errors, we apply these terms to our criminal justice analogy in Figure 9.15.



	Truly not guilty	Truly guilty
Verdict		
Not guilty verdict	Correct	Type II error
Guilty verdict	Type I error	Correct

FIGURE 9.15: Type I and Type II errors in criminal trials.

Thus a Type I error corresponds to incorrectly putting a truly innocent person in jail, whereas a Type II error corresponds to letting a truly guilty person go free. Let's show the corresponding table in Figure 9.16 for hypothesis tests.



	H0 true	HA true
Verdict		
Fail to reject H0	Correct	Type II error
Reject H0	Type I error	Correct

FIGURE 9.16: Type I and Type II errors in hypothesis tests.

9.4.3 How do we choose alpha?

If we are using a sample to make inferences about a population, we run the risk of making errors. For confidence intervals, a corresponding "error" would be constructing a confidence interval that does not contain the true value of the population parameter. For hypothesis tests, this would be making either a Type I or Type II error. Obviously, we want to minimize the probability of either error; we want a small probability of making an incorrect conclusion:

- The probability of a Type I Error occurring is denoted by α. The value of α is called the *significance level* of the hypothesis test, which we defined in Section 9.2.
- The probability of a Type II Error is denoted by β. The value of $1 - \beta$ is known as the *power* of the hypothesis test.

In other words, α corresponds to the probability of incorrectly rejecting H_0 when in fact H_0 is true. On the other hand, β corresponds to the probability of incorrectly failing to reject H_0 when in fact H_0 is false.

Ideally, we want $\alpha = 0$ and $\beta = 0$, meaning that the chance of making either error is 0. However, this can never be the case in any situation where we are sampling for inference. There will always be the possibility of making either error when we use sample data. Furthermore, these two error probabilities are inversely related. As the probability of a Type I error goes down, the probability of a Type II error goes up.

What is typically done in practice is to fix the probability of a Type I error by pre-specifying a significance level α and then try to minimize β. In other words, we will tolerate a certain fraction of incorrect rejections of the null hypothesis H_0, and then try to minimize the fraction of incorrect non-rejections of H_0.

So for example if we used α = 0.01, we would be using a hypothesis testing procedure that in the long run would incorrectly reject the null hypothesis H_0 one percent of the time. This is analogous to setting the confidence level of a confidence interval.

So what value should you use for α? Different fields have different conventions, but some commonly used values include 0.10, 0.05, 0.01, and 0.001. However, it is important to keep in mind that if you use a relatively small value of α, then all things being equal, p-values will have a harder time being less than α. Thus we would reject the null hypothesis less often. In other words, we would reject the null hypothesis H_0 only if we have *very strong* evidence to do so. This is known as a “conservative” test.

On the other hand, if we used a relatively large value of α, then all things being equal, p-values will have an easier time being less than α. Thus we would reject the null hypothesis more often. In other words, we would reject the null hypothesis H_0 even if we only have *mild* evidence to do so. This is known as a “liberal” test.

Learning check

(LC9.5) What is wrong about saying, “The defendant is innocent.” based on the US system of criminal trials?

(LC9.6) What is the purpose of hypothesis testing?

(LC9.7) What are some flaws with hypothesis testing? How could we alleviate them?

(LC9.8) Consider two α significance levels of 0.1 and 0.01. Of the two, which would lead to a more *liberal* hypothesis testing procedure? In other words, one that will, all things being equal, lead to more rejections of the null hypothesis H_0.

9.5 Case study: Are action or romance movies rated higher?

Let’s apply our knowledge of hypothesis testing to answer the question: “Are action or romance movies rated higher on IMDb?”. IMDb is a database on the internet providing information on movie and television show casts, plot summaries, trivia, and ratings. We’ll investigate if, on average, action or romance movies get higher ratings on IMDb.

9.5.1 IMDb ratings data

The `movies` dataset in the `ggplot2movies` package contains information on 58,788 movies that have been rated by users of IMDb.com.

```
movies
```

```
# A tibble: 58,788 × 24
   title      year length budget rating votes    r1    r2    r3    r4    r5    r6
   <chr>     <int>  <int>  <int>  <dbl> <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
 1 $          1971    121     NA    6.4   348   4.5   4.5   4.5   4.5  14.5  24.5
 2 $1000 a…   1939     71     NA    6      20   0    14.5   4.5  24.5  14.5  14.5
 3 $21 a D…   1941      7     NA    8.2     5   0     0     0     0     0    24.5
 4 $40,000    1996     70     NA    8.2     6  14.5   0     0     0     0     0  
 5 $50,000…   1975     71     NA    3.4    17  24.5   4.5   0    14.5  14.5   4.5
 6 $pent      2000     91     NA    4.3    45   4.5   4.5   4.5  14.5  14.5  14.5
 7 $windle    2002     93     NA    5.3   200   4.5   0     4.5   4.5  24.5  24.5
 8 '15'       2002     25     NA    6.7    24   4.5   4.5   4.5   4.5   4.5  14.5
 9 '38        1987     97     NA    6.6    18   4.5   4.5   4.5   0     0     0  
10 '49-'17    1917     61     NA    6      51   4.5   0     4.5   4.5   4.5  44.5
# ℹ 58,778 more rows
# ℹ 12 more variables: r7 <dbl>, r8 <dbl>, r9 <dbl>, r10 <dbl>, mpaa <chr>,
#   Action <int>, Animation <int>, Comedy <int>, Drama <int>,
#   Documentary <int>, Romance <int>, Short <int>
```

We'll focus on a random sample of 68 movies that are classified as either "action" or "romance" movies but not both. We disregard movies that are classified as both so that we can assign all 68 movies into either category. Furthermore, since the original `movies` dataset was a little messy, we provide a pre-wrangled version of our data in the `movies_sample` data frame included in the `moderndive` package. If you're curious, you can look at the necessary data wrangling code to do this on GitHub.

```
movies_sample
```

```
# A tibble: 68 × 4
   title                      year rating genre  
   <chr>                     <int>  <dbl> <chr>  
 1 Underworld                 1985    3.1 Action 
 2 Love Affair                1932    6.3 Romance
 3 Junglee                    1961    6.8 Romance
 4 Eversmile, New Jersey      1989    5   Romance
 5 Search and Destroy         1979    4   Action 
 6 Secreto de Romelia, El     1988    4.9 Romance
 7 Amants du Pont-Neuf, Les   1991    7.4 Romance
 8 Illicit Dreams             1995    3.5 Action 
 9 Kabhi Kabhie               1976    7.7 Romance
10 Electric Horseman, The     1979    5.8 Romance
# ℹ 58 more rows
```

The variables include the `title` and `year` the movie was filmed. Furthermore, we have a numerical variable `rating`, which is the IMDb rating out of 10 stars, and a binary categorical variable `genre` indicating if the movie was an `Action` or `Romance` movie. We are interested in whether `Action` or `Romance` movies got a higher `rating` on average.

Let's perform an exploratory data analysis of this data. Recall from Subsection 2.7.1 that a boxplot is a visualization we can use to show the relationship between a numerical and a categorical variable. Another option you saw in Section 2.6 would be to use a faceted histogram. However, in the interest of brevity, let's only present the boxplot in Figure 9.17.

```
ggplot(data = movies_sample, aes(x = genre, y = rating)) +
  geom_boxplot() +
  labs(y = "IMDb rating")
```

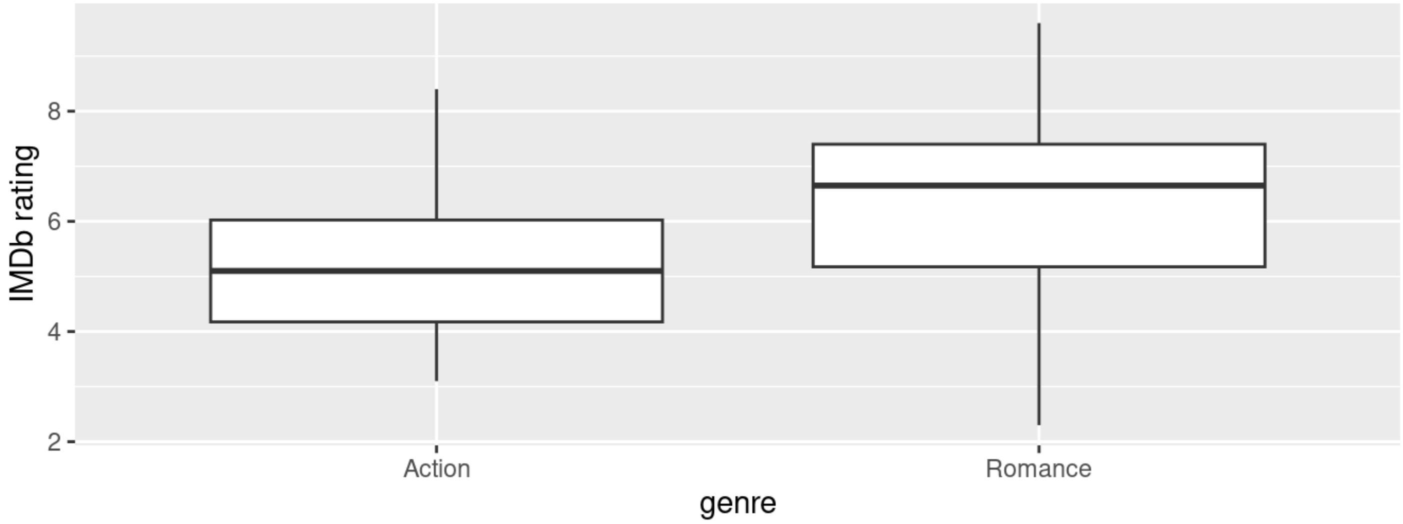


FIGURE 9.17: Boxplot of IMDb rating vs. genre.

Eyeballing Figure 9.17, romance movies have a higher median rating. Do we have reason to believe, however, that there is a *significant* difference between the mean `rating` for action movies compared to romance movies? It's hard to say just based on this plot. The boxplot does show that the median sample rating is higher for romance movies.

However, there is a large amount of overlap between the boxes. Recall that the median isn't necessarily the same as the mean either, depending on whether the distribution is skewed.

Let's calculate some summary statistics split by the binary categorical variable `genre` : the number of movies, the mean rating, and the standard deviation split by `genre` . We'll do this using `dplyr` data wrangling verbs. Notice in particular how we count the number of each type of movie using the `n()` summary function.

```
movies_sample %>%
  group_by(genre) %>%
  summarize(n = n(), mean_rating = mean(rating), std_dev = sd(rating))
```

```
# A tibble: 2 × 4
  genre       n mean_rating std_dev
  <chr>   <int>       <dbl>   <dbl>
1 Action     32     5.275   1.36121
2 Romance    36     6.32222 1.60963
```

Observe that we have 36 movies with an average rating of 6.322 stars and 32 movies with an average rating of 5.275 stars. The difference in these average ratings is thus 6.322 - 5.275 = 1.047. So there appears to be an edge of 1.047 stars in favor of romance movies. The question is, however, are these results indicative of a true difference for *all* romance and action movies? Or could we attribute this difference to chance *sampling variation*?

9.5.2 Sampling scenario

Let's now revisit this study in terms of terminology and notation related to sampling we studied in Subsection 7.3.1. The *study population* is all movies in the IMDb database that are either action or romance (but not both). The *sample* from this population is the 68 movies included in the `movies_sample` dataset.

Since this sample was randomly taken from the population `movies`, it is representative of all romance and action movies on IMDb. Thus, any analysis and results based on `movies_sample` can generalize to the entire population. What are the relevant *population parameter* and *point estimates*? We introduce the fourth sampling scenario in Table 9.3.

TABLE 9.3: Scenarios of sampling for inference

Scenario	Population parameter	Notation	Point estimate	Symbol(s)
1	Population proportion	p	Sample proportion	$\hat{p}$
2	Population mean	μ	Sample mean	$\overline{x}$ or $\widehat{\mu}$
3	Difference in population proportions	$p_1 - p_2$	Difference in sample proportions	$\hat{p}_1 - \hat{p}_2$
4	Difference in population means	$\mu_1 - \mu_2$	Difference in sample means	$\overline{x}_1 - \overline{x}_2$ or $\widehat{\mu}_1 - \widehat{\mu}_2$

So, whereas the sampling bowl exercise in Section 7.1 concerned *proportions*, the pennies exercise in Section 8.1 concerned *means*, the case study on whether yawning is contagious in Section 8.6 and the promotions activity in Section 9.1 concerned *differences in proportions*, we are now concerned with *differences in means*.

In other words, the population parameter of interest is the difference in population mean ratings $\mu_a - \mu_r$, where μ_a is the mean rating of all action movies on IMDb and similarly μ_r is the mean rating of all romance movies. Additionally the point estimate/sample statistic of interest is the difference in sample means $\overline{x}_a - \overline{x}_r$, where $\overline{x}_a$ is the mean rating of the n_a = 32 movies in our sample and $\overline{x}_r$ is the mean rating of the n_r = 36 in our sample. Based on our earlier exploratory data analysis, our estimate $\overline{x}_a - \overline{x}_r$ is $5.275 - 6.322 = -1.047$.

So there appears to be a slight difference of -1.047 in favor of romance movies. The question is, however, could this difference of -1.047 be merely due to chance and sampling variation? Or are these results indicative of a true difference in mean ratings for *all* romance and action movies on IMDb? To answer this question, we'll use hypothesis testing.

9.5.3 Conducting the hypothesis test

We'll be testing:

$$\begin{aligned} H_0 &: \mu_a - \mu_r = 0 \\ \text{vs } H_A &: \mu_a - \mu_r \neq 0 \end{aligned}$$

In other words, the null hypothesis H_0 suggests that both romance and action movies have the same mean rating. This is the "hypothesized universe" we'll *assume* is true. On the other hand, the alternative hypothesis H_A suggests that there is a difference. Unlike the one-sided alternative we used in the promotions exercise $H_A : p_m - p_f > 0$, we are now considering a two-sided alternative of $H_A : \mu_a - \mu_r \neq 0$.

Furthermore, we'll pre-specify a low significance level of α = 0.001. By setting this value low, all things being equal, there is a lower chance that the p-value will be less than α. Thus, there is a lower chance that we'll reject the null hypothesis H_0 in favor of the alternative hypothesis H_A. In other words, we'll reject the hypothesis that there is no difference in mean ratings for all action and romance movies, only if we have quite strong evidence. This is known as a "conservative" hypothesis testing procedure.

1. `specify` variables

Let’s now perform all the steps of the `infer` workflow. We first `specify()` the variables of interest in the `movies_sample` data frame using the formula `rating ~ genre` . This tells `infer` that the numerical variable `rating` is the outcome variable, while the binary variable `genre` is the explanatory variable. Note that unlike previously when we were interested in proportions, since we are now interested in the mean of a numerical variable, we do not need to set the `success` argument.

```
movies_sample %>%
  specify(formula = rating ~ genre)
```

```
Response: rating (numeric)
Explanatory: genre (factor)
# A tibble: 68 × 2
   rating genre
    <dbl> <fct>
 1    3.1 Action
 2    6.3 Romance
 3    6.8 Romance
 4    5   Romance
 5    4   Action
 6    4.9 Romance
 7    7.4 Romance
 8    3.5 Action
 9    7.7 Romance
10    5.8 Romance
# ℹ 58 more rows
```

Observe at this point that the data in `movies_sample` has not changed. The only change so far is the newly defined `Response: rating (numeric)` and `Explanatory: genre (factor)` *meta-data*.

2. `hypothesize` the null

We set the null hypothesis $H_0 : \mu_a - \mu_r = 0$ by using the `hypothesize()` function. Since we have two samples, action and romance movies, we set `null` to be `"independence"` as we described in Section 9.3.

```
movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence")
```

```
Response: rating (numeric)
Explanatory: genre (factor)
Null Hypothesis: independence
# A tibble: 68 × 2
   rating genre
    <dbl> <fct>
 1    3.1 Action
 2    6.3 Romance
 3    6.8 Romance
 4    5   Romance
 5    4   Action
 6    4.9 Romance
 7    7.4 Romance
 8    3.5 Action
 9    7.7 Romance
10    5.8 Romance
# ℹ 58 more rows
```

3. `generate` replicates

After we have set the null hypothesis, we generate “shuffled” replicates assuming the null hypothesis is true by repeating the shuffling/permutation exercise you performed in Section 9.1.

We’ll repeat this resampling without replacement of `type = "permute"` a total of `reps = 1000` times. Feel free to run the code below to check out what the `generate()` step produces.

```
movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  View()
```

4. `calculate` summary statistics

Now that we have 1000 replicated "shuffles" assuming the null hypothesis H_0 that both `Action` and `Romance` movies on average have the same ratings on IMDb, let's `calculate()` the appropriate summary statistic for these 1000 replicated shuffles. From Section 9.2, summary statistics relating to hypothesis testing have a specific name: *test statistics*. Since the unknown population parameter of interest is the difference in population means $\mu_a - \mu_r$, the test statistic of interest here is the difference in sample means $\overline{x}_a - \overline{x}_r$.

For each of our 1000 shuffles, we can calculate this test statistic by setting `stat = "diff in means"` . Furthermore, since we are interested in $\overline{x}_a - \overline{x}_r$, we set `order = c("Action", "Romance")` . Let's save the results in a data frame called `null_distribution_movies` :

```
null_distribution_movies <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("Action", "Romance"))
null_distribution_movies
```

```
# A tibble: 1,000 × 2
   replicate      stat
       <int>     <dbl>
 1         1  0.511111
 2         2  0.345833
 3         3 -0.327083
 4         4 -0.209028
 5         5 -0.433333
 6         6 -0.102778
 7         7  0.387153
 8         8  0.168750
 9         9  0.257292
10        10  0.334028
# ℹ 990 more rows
```

Observe that we have 1000 values of `stat`, each representing one instance of $\overline{x}_a - \overline{x}_r$. The 1000 values form the *null distribution*, which is the technical term for the sampling distribution of the difference in sample means $\overline{x}_a - \overline{x}_r$ assuming H_0 is true. What happened in real life? What was the observed difference in promotion rates? What was the *observed test statistic* $\overline{x}_a - \overline{x}_r$? Recall from our earlier data wrangling, this observed difference in means was $5.275 - 6.322 = -1.047$. We can also achieve this using the code that constructed the null distribution `null_distribution_movies` but with the `hypothesize()` and `generate()` steps removed. Let's save this in `obs_diff_means`:

```
obs_diff_means <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  calculate(stat = "diff in means", order = c("Action", "Romance"))
obs_diff_means
```

```
Response: rating (numeric)
Explanatory: genre (factor)
# A tibble: 1 × 1
      stat
     <dbl>
1 -1.04722
```

5. `visualize` the p-value

Lastly, in order to compute the p-value, we have to assess how "extreme" the observed difference in means of -1.047 is. We do this by comparing -1.047 to our null distribution, which was constructed in a hypothesized universe of no true difference in movie ratings. Let's visualize both the null distribution and the p-value in Figure 9.18. Unlike our example in Subsection 9.3.1 involving promotions, since we have a two-sided $H_A: \mu_a - \mu_r \neq 0$, we have to allow for both possibilities for *more extreme*, so we set `direction = "both"` .

```
visualize(null_distribution_movies, bins = 10) +
  shade_p_value(obs_stat = obs_diff_means, direction = "both")
```

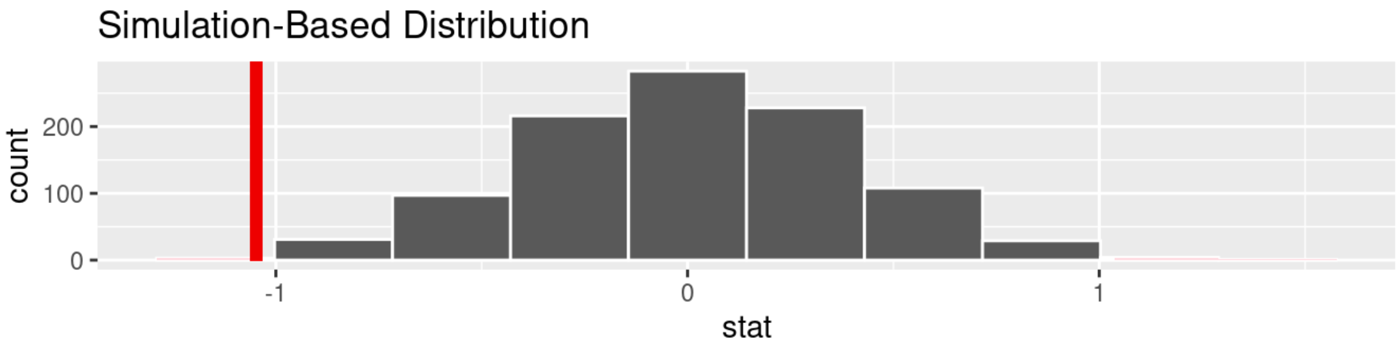


FIGURE 9.18: Null distribution, observed test statistic, and p-value.

Let's go over the elements of this plot. First, the histogram is the *null distribution*. Second, the solid line is the *observed test statistic*, or the difference in sample means we observed in real life of $5.275 - 6.322 = -1.047$. Third, the two shaded areas of the histogram form the *p-value*, or the probability of obtaining a test statistic just as or more extreme than the observed test statistic *assuming the null hypothesis H_0 is true*.

What proportion of the null distribution is shaded? In other words, what is the numerical value of the p-value? We use the `get_p_value()` function to compute this value:

```
null_distribution_movies %>%
  get_p_value(obs_stat = obs_diff_means, direction = "both")
```

```
# A tibble: 1 × 1
  p_value
    <dbl>
1   0.004
```

This p-value of 0.004 is very small. In other words, there is a very small chance that we'd observe a difference of 5.275 - 6.322 = -1.047 in a hypothesized universe where there was truly no difference in ratings.

But this p-value is larger than our (even smaller) pre-specified α significance level of 0.001. Thus, we are inclined to fail to reject the null hypothesis $H_0 : \mu_a - \mu_r = 0$. In non-statistical language, the conclusion is: we do not have the evidence needed in this sample of data to suggest that we should reject the hypothesis that there is no difference in mean IMDb ratings between romance and action movies. We, thus, cannot say that a difference exists in romance and action movie ratings, on average, for all IMDb movies.

Learning check

(LC9.9) Conduct the same analysis comparing action movies versus romantic movies using the median rating instead of the mean rating. What was different and what was the same?

(LC9.10) What conclusions can you make from viewing the faceted histogram looking at `rating` versus `genre` that you couldn't see when looking at the boxplot?

(LC9.11) Describe in a paragraph how we used Allen Downey's diagram to conclude if a statistical difference existed between mean movie ratings for action and romance movies.

(LC9.12) Why are we relatively confident that the distributions of the sample ratings will be good approximations of the population distributions of ratings for the two genres?

(LC9.13) Using the definition of p-value, write in words what the p-value represents for the hypothesis test comparing the mean rating of romance to action movies.

(LC9.14) What is the value of the p-value for the hypothesis test comparing the mean rating of romance to action movies?

(LC9.15) Test your data wrangling knowledge and EDA skills:

- Use `dplyr` and `tidyr` to create the necessary data frame focused on only action and romance movies (but not both) from the `movies` data frame in the `ggplot2movies` package.
- Make a boxplot and a faceted histogram of this population data comparing ratings of action and romance movies from IMDb.
- Discuss how these plots compare to the similar plots produced for the `movies_sample` data.

9.6 Conclusion

9.6.1 Theory-based hypothesis tests

Much as we did in Subsections 7.6.2 and 8.7.2 when we showed you theory-based methods for compututing standard errors and constructing confidence intervals that involved mathematical formulas, we now present an example of a traditional theory-based method to conduct hypothesis tests. This method relies on probability models, probability distributions, and a few assumptions to construct the null distribution. This is in contrast to the approach we've been using throughout this book where we relied on computer simulations to construct the null distribution.

These traditional theory-based methods have been used for decades mostly because researchers didn't have access to computers that could run thousands of calculations quickly and efficiently. Now that computing power is much cheaper and more accessible, simulation-based methods are much more feasible. However, researchers in many fields continue to use theory-based methods. Hence, we make it a point to include an example here.

As we'll show in this section, any theory-based method is ultimately an approximation to the simulation-based method. The theory-based method we'll focus on is known as the *two-sample t-test* for testing differences in sample means. However, the test statistic we'll use won't be the difference in sample means $\overline{x}_1 - \overline{x}_2$, but rather the related *two-sample t-statistic*. The data we'll use will once again be the `movies_sample` data of action and romance movies from Section 9.5.

Two-sample t-statistic

A common task in statistics is the process of "standardizing a variable." By standardizing different variables, we make them more comparable. For example, say you are interested in studying the distribution of temperature recordings from Portland, Oregon, USA and comparing it to that of the temperature recordings in Montreal, Quebec, Canada. Given that US temperatures are generally recorded in degrees Fahrenheit and Canadian temperatures are generally recorded in degrees Celsius, how can we make them comparable? One approach would be to convert degrees Fahrenheit into Celsius, or vice versa. Another approach would be to convert them both to a common "standardized" scale, like Kelvin units of temperature.

One common method for standardizing a variable from probability and statistics theory is to compute the z-score:

$$z = \frac{x - \mu}{\sigma}$$

where x represents one value of a variable, μ represents the mean of that variable, and σ represents the standard deviation of that variable. You first subtract the mean μ from each value of x and then divide $x - \mu$ by the standard deviation σ. These operations will have the effect of *re-centering* your variable around 0 and *re-scaling* your variable x so that they have what are known as "standard units." Thus for every value that your variable can take, it has a corresponding z-score that gives how many standard units away that value is from the mean μ. z-scores are normally distributed with mean 0 and standard deviation 1. This curve is called a "z-distribution" or "standard normal" curve and has the common, bell-shaped pattern from Figure 9.19 discussed in Appendix A.2.

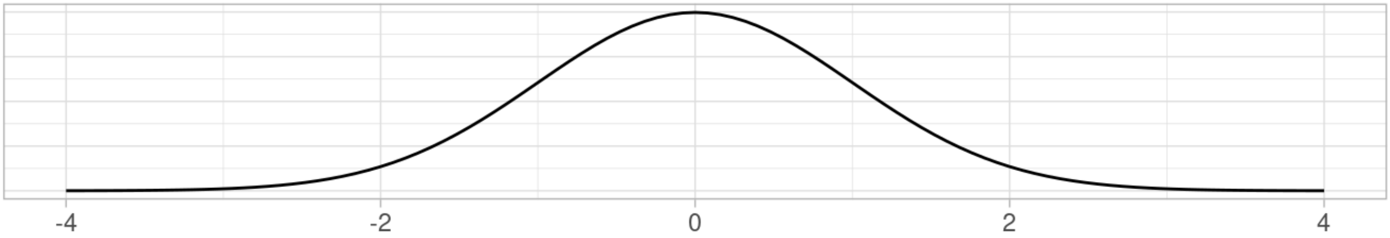


FIGURE 9.19: Standard normal z curve.

Bringing these back to the difference of sample mean ratings $\overline{x}_a - \overline{x}_r$ of action versus romance movies, how would we standardize this variable? By once again subtracting its mean and dividing by its standard deviation. Recall two facts from Subsection 7.3.3. First, if the sampling was done in a representative fashion, then the sampling distribution of $\overline{x}_a - \overline{x}_r$ will be centered at the true population parameter $\mu_a - \mu_r$. Second, the standard deviation of point estimates like $\overline{x}_a - \overline{x}_r$ has a special name: the standard error.

Applying these ideas, we present the *two-sample t-statistic*:

$$t = \frac{(\bar{x}_a - \bar{x}_r) - (\mu_a - \mu_r)}{\text{SE}_{\bar{x}_a - \bar{x}_r}} = \frac{(\bar{x}_a - \bar{x}_r) - (\mu_a - \mu_r)}{\sqrt{\dfrac{{s_a}^2}{n_a} + \dfrac{{s_r}^2}{n_r}}}$$

Oofda! There is a lot to try to unpack here! Let’s go slowly. In the numerator, $\bar{x}_a - \bar{x}_r$ is the difference in sample means, while $\mu_a - \mu_r$ is the difference in population means. In the denominator, s_a and s_r are the *sample standard deviations* of the action and romance movies in our sample `movies_sample`. Lastly, n_a and n_r are the sample sizes of the action and romance movies. Putting this together under the square root gives us the standard error $\text{SE}_{\bar{x}_a - \bar{x}_r}$.

Observe that the formula for $\text{SE}_{\bar{x}_a - \bar{x}_r}$ has the sample sizes n_a and n_r in them. So as the sample sizes increase, the standard error goes down. We’ve seen this concept numerous times now, in particular (1) in our simulations using the three virtual shovels with n = 25, 50, and 100 slots in Figure 7.15, (2) in Subsection 8.5.3 where we studied the effect of using larger sample sizes on the widths of confidence intervals, and (3) in Subsection 7.6.2 where we studied the formula-based approximation to the standard error of the sample proportion $\hat{p}$.

So how can we use the two-sample t-statistic as a test statistic in our hypothesis test? First, assuming the null hypothesis $H_0 : \mu_a - \mu_r = 0$ is true, the right-hand side of the numerator (to the right of the $-$ sign), $\mu_a - \mu_r$, becomes 0.

Second, similarly to how the Central Limit Theorem from Subsection 7.5 states that sample means follow a normal distribution, it can be mathematically proven that the two-sample t-statistic follows a *t distribution with degrees of freedom* “roughly equal” to $df = n_a + n_r - 2$. To better understand this concept of *degrees of freedom*, we next display three examples of t-distributions in Figure 9.20 along with the standard normal z curve.

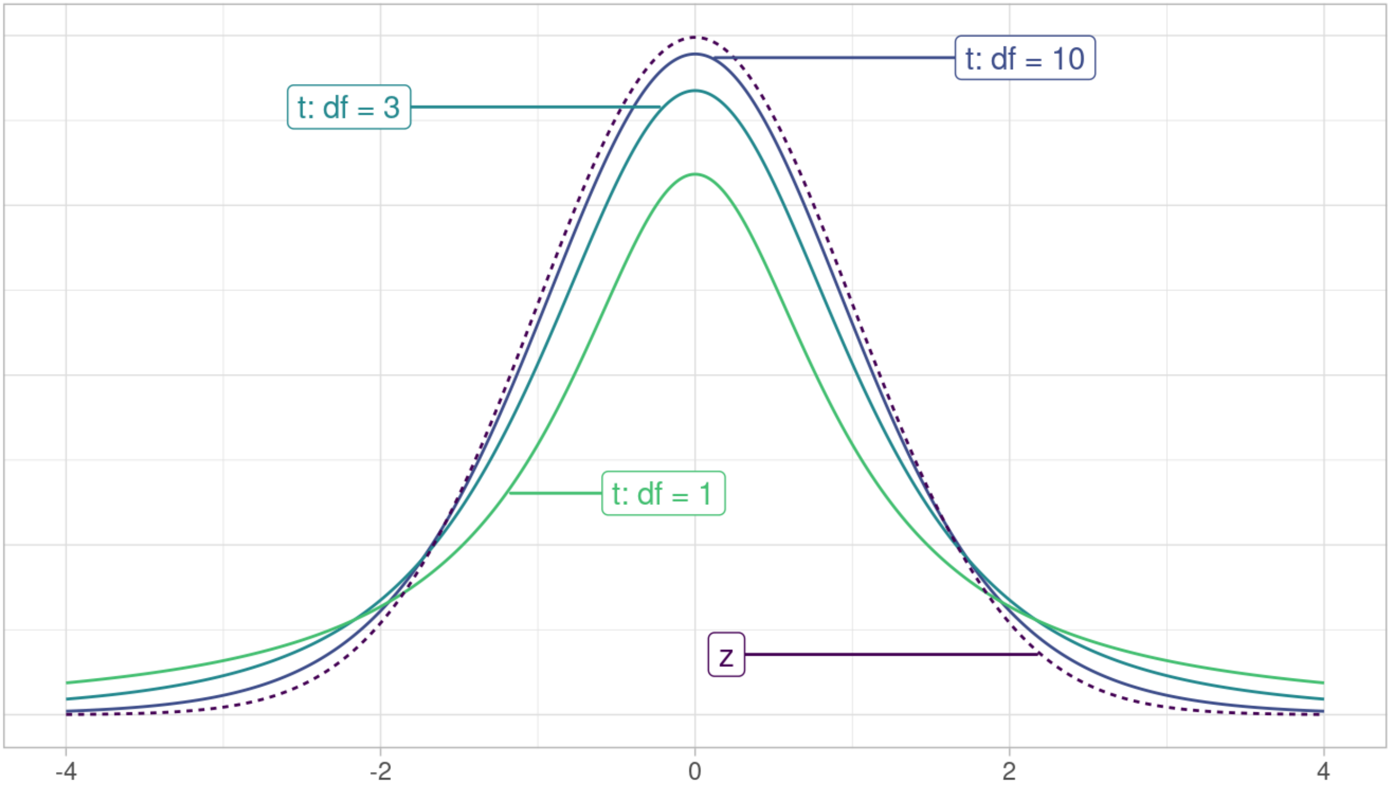


FIGURE 9.20: Examples of t-distributions and the z curve.

Begin by looking at the center of the plot at 0 on the horizontal axis. As you move up from the value of 0, follow along with the labels and note that the bottom curve corresponds to 1 degree of freedom, the curve above it is for 3 degrees of freedom, the curve above that is for 10 degrees of freedom, and lastly the dotted curve is the standard normal z curve.

Observe that all four curves have a bell shape, are centered at 0, and that as the degrees of freedom increase, the t-distribution more and more resembles the standard normal z curve. The "degrees of freedom" measures how different the t distribution will be from a normal distribution. t -distributions tend to have more values in the tails of their distributions than the standard normal z curve.

This "roughly equal" statement indicates that the equation $df = n_a + n_r - 2$ is a "good enough" approximation to the true degrees of freedom. The true formula is a bit more complicated than this simple expression, but we've found the formula to be beyond the reach of those new to statistical inference and it does little to build the intuition of the t-test.

The message to retain, however, is that small sample sizes lead to small degrees of freedom and thus small sample sizes lead to t-distributions that are different than the z curve. On the other hand, large sample sizes correspond to large degrees of freedom and thus produce t distributions that closely align with the standard normal z-curve.

So, assuming the null hypothesis H_0 is true, our formula for the test statistic simplifies a bit:

$$t = \frac{(\bar{x}_a - \bar{x}_r) - 0}{\sqrt{\dfrac{{s_a}^2}{n_a} + \dfrac{{s_r}^2}{n_r}}} = \frac{\bar{x}_a - \bar{x}_r}{\sqrt{\dfrac{{s_a}^2}{n_a} + \dfrac{{s_r}^2}{n_r}}}$$

Let's compute the values necessary for this two-sample t-statistic. Recall the summary statistics we computed during our exploratory data analysis in Section 9.5.1.

```
movies_sample %>%
  group_by(genre) %>%
  summarize(n = n(), mean_rating = mean(rating), std_dev = sd(rating))
```

```
# A tibble: 2 × 4
  genre       n mean_rating std_dev
  <chr>   <int>       <dbl>   <dbl>
1 Action     32      5.275   1.36121
2 Romance    36      6.32222 1.60963
```

Using these values, the observed two-sample t-test statistic is

$$\frac{\bar{x}_a - \bar{x}_r}{\sqrt{\dfrac{{s_a}^2}{n_a} + \dfrac{{s_r}^2}{n_r}}} = \frac{5.28 - 6.32}{\sqrt{\dfrac{1.36^2}{32} + \dfrac{1.61^2}{36}}} = -2.906$$

Great! How can we compute the p-value using this theory-based test statistic? We need to compare it to a null distribution, which we construct next.

Null distribution

Let's revisit the null distribution for the test statistic $\bar{x}_a - \bar{x}_r$ we constructed in Section 9.5. Let's visualize this in the left-hand plot of Figure 9.21.

```
# Construct null distribution of xbar_a - xbar_r:
null_distribution_movies <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("Action", "Romance"))
visualize(null_distribution_movies, bins = 10)
```

The `infer` package also includes some built-in theory-based test statistics as well. So instead of calculating the test statistic of interest as the `"diff in means"` $\bar{x}_a - \bar{x}_r$, we can calculate this defined two-sample t-statistic by setting `stat = "t"`. Let's visualize this in the right-hand plot of Figure 9.21.

```
# Construct null distribution of t:
null_distribution_movies_t <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  # Notice we switched stat from "diff in means" to "t"
  calculate(stat = "t", order = c("Action", "Romance"))
visualize(null_distribution_movies_t, bins = 10)
```

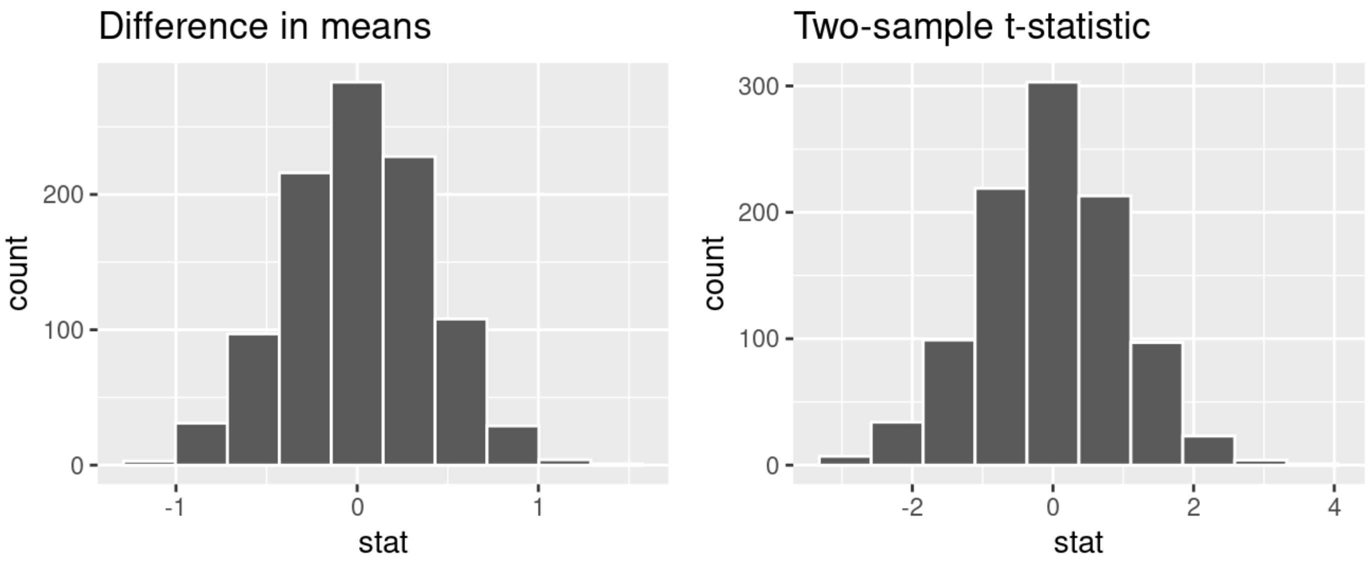


FIGURE 9.21: Comparing the null distributions of two test statistics.

Observe that while the shape of the null distributions of both the difference in means $\bar{x}_a - \bar{x}_r$ and the two-sample t-statistics are similar, the scales on the x-axis are different. The two-sample t-statistic values are spread out over a larger range.

However, a traditional theory-based t-test doesn't look at the simulated histogram in `null_distribution_movies_t`, but instead it looks at the t-distribution curve with degrees of freedom equal to roughly 65.85. This calculation is based on the complicated formula referenced previously, which we approximated with $df = n_a + n_r - 2 = 32 + 36 - 2 = 66$. Let's overlay this t-distribution curve over the top of our simulated two-sample t-statistics using the `method = "both"` argument in `visualize()`.

```
visualize(null_distribution_movies_t, bins = 10, method = "both")
```

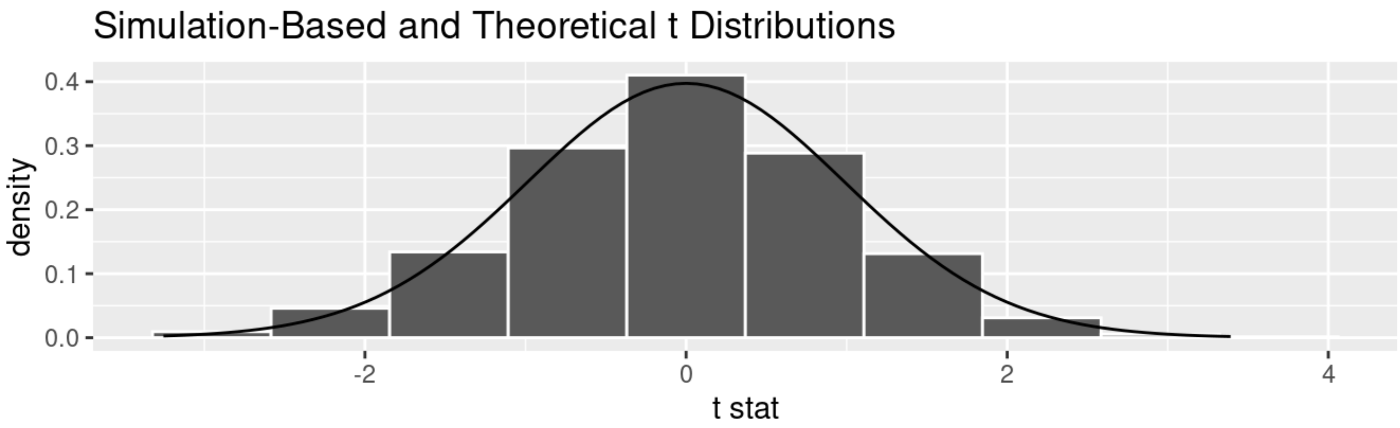


FIGURE 9.22: Null distribution using t-statistic and t-distribution.

Observe that the curve does a good job of approximating the histogram here. To calculate the p-value in this case, we need to figure out how much of the total area under the t-distribution curve is at or "more extreme" than our observed two-sample t-statistic. Since $H_A : \mu_a - \mu_r \neq 0$ is a two-sided alternative, we need to add up the areas in both tails.

We first compute the observed two-sample t-statistic using `infer` verbs. This shortcut calculation further assumes that the null hypothesis is true: that the population of action and romance movies have an equal average rating.

```
obs_two_sample_t <- movies_sample %>%
  specify(formula = rating ~ genre) %>%
  calculate(stat = "t", order = c("Action", "Romance"))
obs_two_sample_t
```

```
Response: rating (numeric)
Explanatory: genre (factor)
# A tibble: 1 × 1
       stat
      <dbl>
1 -2.90589
```

We want to find the percentage of values that are at or below `obs_two_sample_t` $= -2.906$ or at or above `-obs_two_sample_t` $= 2.906$. We use the `shade_p_value()` function with the `direction` argument set to `"both"` to do this:

```
visualize(null_distribution_movies_t, method = "both") +
  shade_p_value(obs_stat = obs_two_sample_t, direction = "both")
```

```
Warning: Check to make sure the conditions have been met for the theoretical
method. {infer} currently does not check these for you.
```

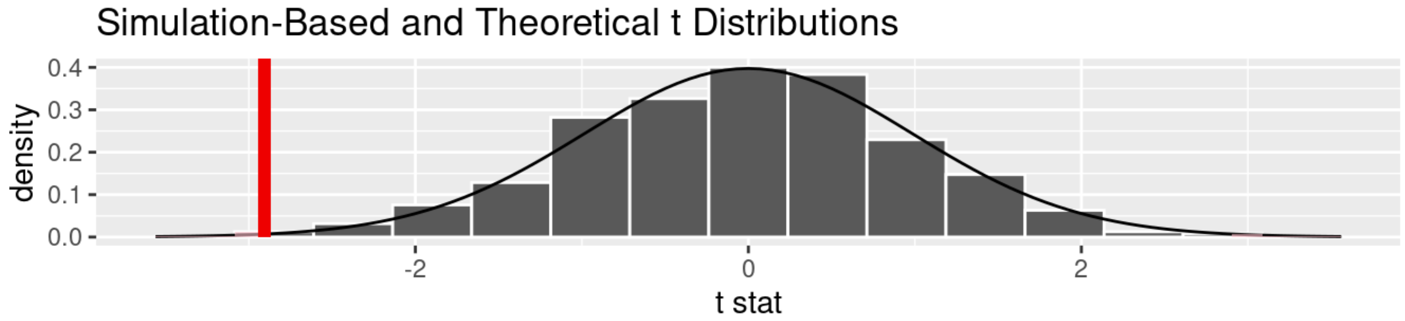


FIGURE 9.23: Null distribution using t-statistic and t-distribution with p-value shaded.

(We'll discuss this warning message shortly.) What is the p-value? We apply `get_p_value()` to our null distribution saved in `null_distribution_movies_t` :

```
null_distribution_movies_t %>%
  get_p_value(obs_stat = obs_two_sample_t, direction = "both")
```

```
# A tibble: 1 × 1
  p_value
    <dbl>
1   0.002
```

We have a very small p-value, and thus it is very unlikely that these results are due to *sampling variation*. Thus, we are inclined to reject H_0.

Let's come back to that earlier warning message: `Check to make sure the conditions have been met for the theoretical method. {infer} currently does not check these for you.` To be able to use the t-test and other such theoretical methods, there are always a few conditions to check. The `infer` package does not automatically check these conditions, hence the warning message we received. These conditions are necessary so that the underlying mathematical theory holds. In order for the results of our two-sample t-test to be valid, three conditions must be met:

1. Nearly normal populations or large sample sizes. A general rule of thumb that works in many (but not all) situations is that the sample size n should be greater than 30.
2. Both samples are selected independently of each other.
3. All observations are independent from each other.

Let's see if these conditions hold for our `movies_sample` data:

1. This is met since n_a = 32 and n_r = 36 are both larger than 30, satisfying our rule of thumb.
2. This is met since we sampled the action and romance movies at random and in an unbiased fashion from the database of all IMDb movies.
3. Unfortunately, we don't know how IMDb computes the ratings. For example, if the same person rated multiple movies, then those observations would be related and hence not independent.

Assuming all three conditions are roughly met, we can be reasonably certain that the theory-based t-test results are valid. If any of the conditions were clearly not met, we couldn't put as much trust into any conclusions reached. On the other hand, in most scenarios, the only assumption that needs to be met in the simulation-based method is that the sample is selected at random. Thus, in our experience, we prefer simulation-based methods as they have fewer assumptions, are conceptually easier to understand, and since computing power has recently become easily accessible, they can be run quickly. That being said since much of the world's research still relies on traditional theory-based methods, we also believe it is important to understand them.

You may be wondering why we chose `reps = 1000` for these simulation-based methods. We've noticed that after around 1000 replicates for the null distribution and the bootstrap distribution for most problems you can start to get a general sense for how the statistic behaves. You can change this value to something like 10,000 though for `reps` if you would like even finer detail but this will take more time to compute. Feel free to iterate on this as you like to get an even better idea about the shape of the null and bootstrap distributions as you wish.

9.6.2 When inference is not needed

We've now walked through several different examples of how to use the `infer` package to perform statistical inference: constructing confidence intervals and conducting hypothesis tests. For each of these examples, we made it a point to always perform an exploratory data analysis (EDA) first; specifically, by looking at the raw data values, by using data visualization with `ggplot2`, and by data wrangling with `dplyr` beforehand. We *highly* encourage you to always do the same. As a beginner to statistics, EDA helps you develop intuition as to what statistical methods like confidence intervals and hypothesis tests can tell us. Even as a seasoned practitioner of statistics, EDA helps guide your statistical investigations. In particular, is statistical inference even needed?

Let's consider an example. Say we're interested in the following question: Of *all* flights leaving a New York City airport, are Hawaiian Airlines flights in the air for longer than Alaska Airlines flights? Furthermore, let's assume that 2013 flights are a representative sample of all such flights. Then we can use the `flights` data frame in the `nycflights13` package we introduced in Section 1.4 to answer our question. Let's filter this data frame to only include Hawaiian and Alaska Airlines using their `carrier` codes `HA` and `AS`:

```
flights_sample <- flights %>%
  filter(carrier %in% c("HA", "AS"))
```

There are two possible statistical inference methods we could use to answer such questions. First, we could construct a 95% confidence interval for the difference in population means $\mu_{HA} - \mu_{AS}$, where μ_{HA} is the mean air time of all Hawaiian Airlines flights and μ_{AS} is the mean air time of all Alaska Airlines flights. We could then check if the entirety of the interval is greater than 0, suggesting that $\mu_{HA} - \mu_{AS} > 0$, or, in other words suggesting that $\mu_{HA} > \mu_{AS}$. Second, we could perform a hypothesis test of the null hypothesis $H_0 : \mu_{HA} - \mu_{AS} = 0$ versus the alternative hypothesis $H_A : \mu_{HA} - \mu_{AS} > 0$.

However, let's first construct an exploratory visualization as we suggested earlier. Since `air_time` is numerical and `carrier` is categorical, a boxplot can display the relationship between these two variables, which we display in Figure 9.24.

```
ggplot(data = flights_sample, mapping = aes(x = carrier, y = air_time)) +
  geom_boxplot() +
  labs(x = "Carrier", y = "Air Time")
```

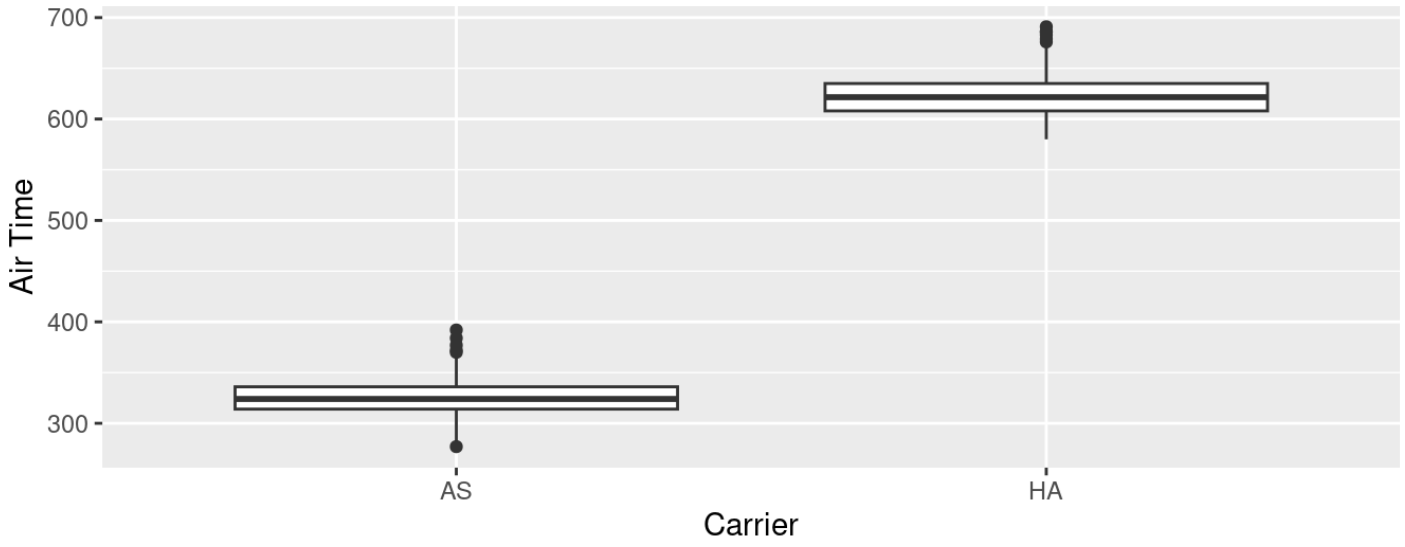


FIGURE 9.24: Air time for Hawaiian and Alaska Airlines flights departing NYC in 2013.

This is what we like to call "no PhD in Statistics needed" moments. You don't have to be an expert in statistics to know that Alaska Airlines and Hawaiian Airlines have *significantly* different air times. The two boxplots don't even overlap! Constructing a confidence interval or conducting a hypothesis test would frankly not provide much more insight than Figure 9.24.

Let's investigate why we observe such a clear cut difference between these two airlines using data wrangling. Let's first group by the rows of `flights_sample` not only by `carrier` but also by destination `dest`. Subsequently, we'll compute two summary statistics: the number of observations using `n()` and the mean airtime:

```
flights_sample %>%
  group_by(carrier, dest) %>%
  summarize(n = n(), mean_time = mean(air_time, na.rm = TRUE))
```

```
# A tibble: 2 × 4
# Groups:   carrier [2]
  carrier dest      n mean_time
  <chr>   <chr> <int>     <dbl>
1 AS      SEA     714   325.618
2 HA      HNL     342   623.088
```

It turns out that from New York City in 2013, Alaska only flew to `SEA` (Seattle) from New York City (NYC) while Hawaiian only flew to `HNL` (Honolulu) from NYC. Given the clear difference in distance from New York City to Seattle versus New York City to Honolulu, it is not surprising that we observe such different (*statistically significantly different*, in fact) air times in flights.

This is a clear example of not needing to do anything more than a simple exploratory data analysis using data visualization and descriptive statistics to get an appropriate conclusion. This is why we highly recommend you perform an EDA of any sample data before running statistical inference methods like confidence intervals and hypothesis tests.

9.6.3 Problems with p-values

On top of the many common misunderstandings about hypothesis testing and p-values we listed in Section 9.4, another unfortunate consequence of the expanded use of p-values and hypothesis testing is a phenomenon known as "p-hacking." p-hacking is the act of "cherry-picking" only results that are "statistically significant" while dismissing those that aren't, even if at the expense of the scientific ideas. There are lots of articles written recently about misunderstandings and the problems with p-values. We encourage you to check some of them out:

1. Misunderstandings of p-values
2. What a nerdy debate about p-values shows about science - and how to fix it
3. Statisticians issue warning over misuse of P values
4. You Can't Trust What You Read About Nutrition
5. A Litany of Problems with p-values

Such issues were getting so problematic that the American Statistical Association (ASA) put out a statement in 2016 titled, "The ASA Statement on Statistical Significance and P-Values," with six principles underlying the proper use and interpretation of p-values. The ASA released this guidance on p-values to improve the conduct and interpretation of quantitative science and to inform the growing emphasis on reproducibility of science research.

We as authors much prefer the use of confidence intervals for statistical inference, since in our opinion they are much less prone to large misinterpretation. However, many fields still exclusively use p-values for statistical inference and this is one reason for including them in this text. We encourage you to learn more about "p-hacking" as well and its implication for science.

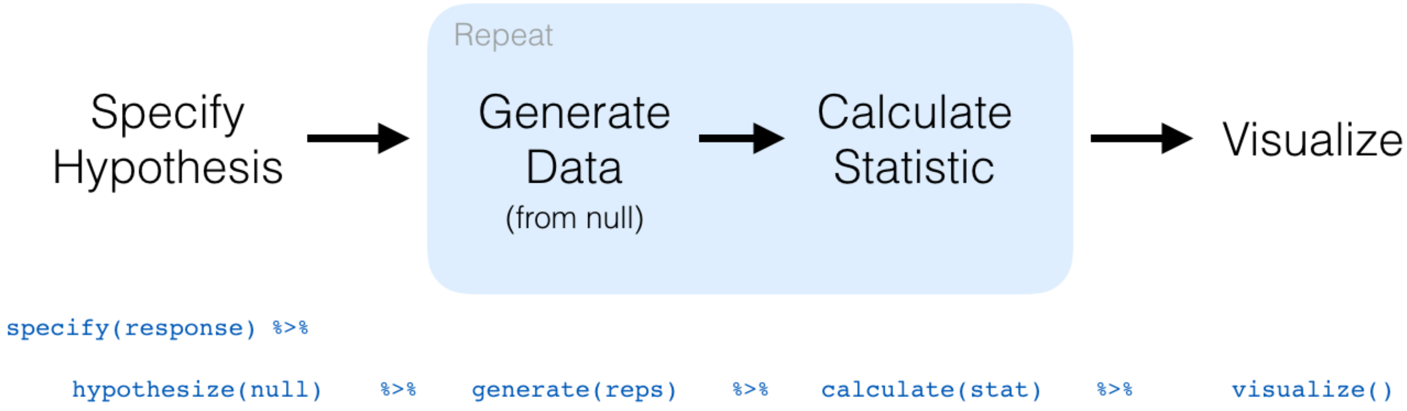
9.6.4 Additional resources

An R script file of all R code used in this chapter is available here.

If you want more examples of the `infer` workflow for conducting hypothesis tests, we suggest you check out the `infer` package homepage, in particular, a series of example analyses available at https://infer.netlify.app/articles/.

9.6.5 What's to come

We conclude with the `infer` pipeline for hypothesis testing in Figure 9.25.



```
specify(response) %>%
     hypothesize(null)    %>%    generate(reps)    %>%    calculate(stat)    %>%       visualize()
```

FIGURE 9.25: infer package workflow for hypothesis testing.

Now that we've armed ourselves with an understanding of confidence intervals from Chapter 8 and hypothesis tests from this chapter, we'll now study inference for regression in the upcoming Chapter 10.

We'll revisit the regression models we studied in Chapter 5 on basic regression and Chapter 6 on multiple regression. For example, recall Table 5.2 (shown again here in Table 9.4), corresponding to our regression model for an instructor's teaching score as a function of their "beauty" score.

```r
# Fit regression model:
score_model <- lm(score ~ bty_avg, data = evals)

# Get regression table:
get_regression_table(score_model)
```

TABLE 9.4: Linear regression table

term	estimate	std_error	statistic	p_value	lower_ci	upper_ci
intercept	3.880	0.076	50.96	0	3.731	4.030
bty_avg	0.067	0.016	4.09	0	0.035	0.099

We previously saw in Subsection 5.1.2 that the values in the `estimate` column are the fitted intercept b_0 and fitted slope for "beauty" score b_1. In Chapter 10, we'll unpack the remaining columns: `std_error` which is the standard error, `statistic` which is the observed *standardized* test statistic to compute the `p_value`, and the 95% confidence intervals as given by `lower_ci` and `upper_ci`.