Hypothesis Testing in 4 Steps

Stat 145

Testing Hypotheses in 4 steps

Hypothesis testing follows a 4-step outline.

1. State the null and alternative **hypotheses**.

   We have been practicing this step already. Remember that the null hypothesis will be a specific statement about the parameter or parameters involved.

2. Compute a **test statistic**.

   The evidence from the data must be summarized as a single number, computed from the data. We call this number the test statistic.

3. Compute a **p-value** by comparing the observed test statistic to the null distribution.

   The p-value answers the question: How often would we get a test statistic at least an unusual as the one we observed, *if the null hypothesis were true*?

   One way to estimate the p-value is to simulate random data sets that

   - match the way the original data were collected, and
   - force the null hypothesis to be true.

   We can then compute the test statistic for each of these **randomization samples**. The distribution of these statistics is called a **randomization distribution** or a **null distribution**.[1]

4. **Interpret** the results.

   The smaller the p-value, the stronger the evidence *against* the null hypothesis.

   - If the p-value is small enough, we *reject the null hypothesis.*
   - If the p-value is not small enough, we *fail to reject the null hypothesis.* (In this case our data are "compatible with" the null hypothesis, but the data may be compatible with other hypotheses as well.)

Examples

Gender discrimination in promotion decisions?

This example is discussed in Section 5.1 of IMS

Hypotheses

- null hypothesis
  - H_0: $p_M - p_F = 0$ (There is no difference in the proportion of males who are promoted and the proportion of females who are promoted.)

[1] Technically, the randomization distribution is just an approximation to the null distribution, and each time you rerun your randomization, you will get slightly different results. But we will use the terms interchangeably. Later we will learn other ways to determine or approximate the null distribution that don't involve randomization.

- alternative hypothesis – two choices
  - H_a: $p_M - p_F \neq 0$ (There is a difference in the proportion of males who are promoted and the proportion of females who are promoted.)
  - H_a: $p_M - p_F > 0$ (Males are more likely to be promoted than females.)

Our choice for alternative hypothesis will be determined by how we would deal with the case when women are promoted more often than men in our data. Will we take that as evidence agains the null hypothesis or not. The first type of alternative is called **two-sided** or **two-tailed** and the second is called **one-sided** or **one-tailed**.

The most important thing to know about choosing one-sided or two-sided alternatives: **Your choice may not depend on the data.** If you wouldn't know which side to use without looking at the data, then you should be doing a two-sided test.

Data The data are presented in a table in the book, but the table seems to be incorrect compared to the rest of the section. The table should be this:

	not promoted	promoted
female	10	14
male	3	21

We can create a data set with a row for each case like this:

```
Promotion <-
  bind_rows(
    do(21) * tibble(gender = "male", decision = "promoted"),
    do( 3) * tibble(gender = "male", decision = "not promoted"),
    do(14) * tibble(gender = "female", decision = "promoted"),
    do(10) * tibble(gender = "female", decision = "not promoted")
  )
```

Here is our table, but oriented the other way around so we can have the response on the left and explanatory variable on the right to match how we do this in regression. Also, it gives us the column totals that we need to compute proportions.

```
tally(decision ~ gender, data = Promotion, margins = TRUE) %>% pander()
```

	female	male
not promoted	10	3
promoted	14	21
Total	24	24

```
tally(decision ~ gender, data = Promotion, margins = TRUE, format = "prop") %>% pander()
```

Computing the test statistic

	female	male
not promoted	0.4167	0.125
promoted	0.5833	0.875
Total	1	1

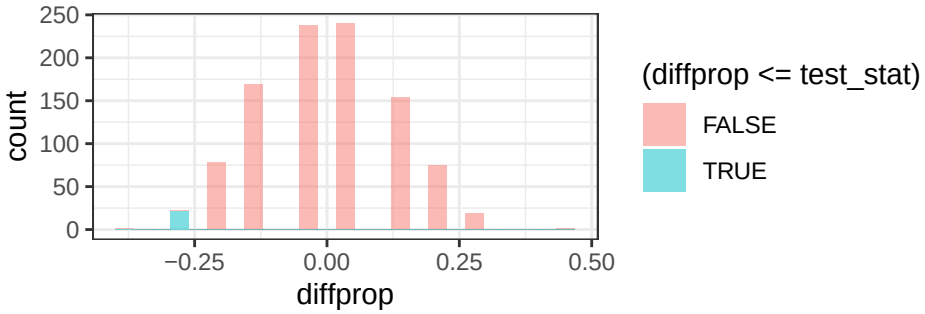
```
test_stat <- diffprop(decision ~ gender, data = Promotion)
test_stat
```

```
##   diffprop
## -0.2916667
```

```
set.seed(13579)
Promotion_null <- do(1000) * diffprop(decision ~ shuffle(gender), data = Promotion)
```

```
gf_histogram( ~ diffprop, data = Promotion_null, fill = ~(diffprop <= test_stat))
```

Computing the p-value




```
prop( ~ (diffprop <= test_stat), data = Promotion_null)
```

```
## prop_TRUE
##     0.023
```

Does caffeine make people tap fingers faster?

Hypotheses

- H_0: $\mu_N - \mu_C = 0$
- H_a: $\mu_N - \mu_C \neq 0$

```
head(CaffeineTaps)
```

Taking a look at our data:

```
##   Taps    Group Caffeine
## 1  246 Caffeine      Yes
## 2  248 Caffeine      Yes
## 3  250 Caffeine      Yes
## 4  252 Caffeine      Yes
## 5  248 Caffeine      Yes
## 6  250 Caffeine      Yes
```

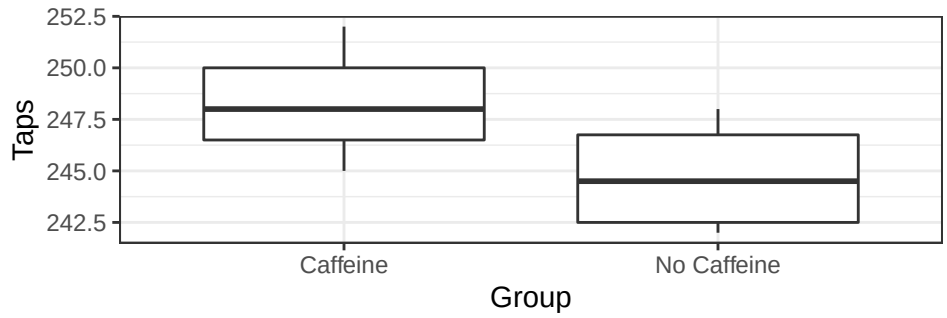
```
df_stats(Taps ~ Group, data = CaffeineTaps, mean)
```

```
##   response       Group  mean
## 1     Taps    Caffeine 248.3
## 2     Taps No Caffeine 244.8
```

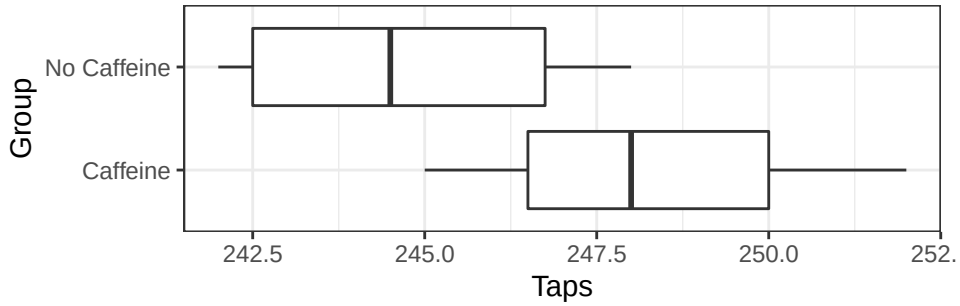
```
diffmean(Taps ~ Group, data = CaffeineTaps)
```

```
## diffmean
##     -3.5
```

```
gf_boxplot(Taps ~ Group, data = CaffeineTaps)
```



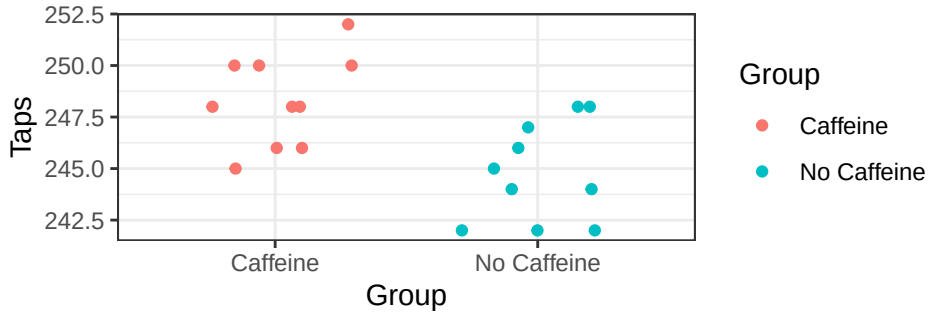

```
# horizontal to fit on the page better
gf_boxplot(Group ~ Taps, data = CaffeineTaps)
```



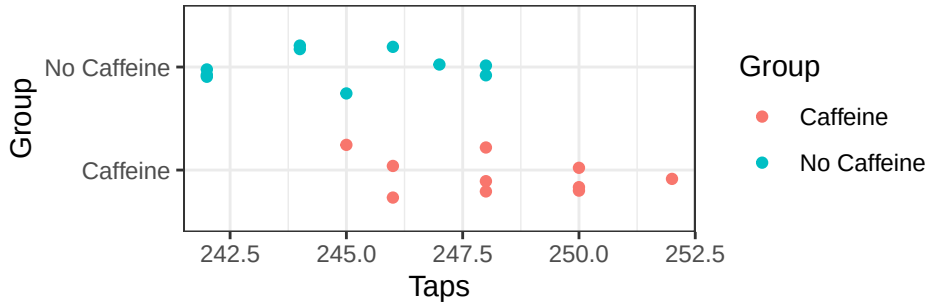
Other plots we could have used:

- jittered scatter plot

```
gf_jitter(Taps ~ Group, data = CaffeineTaps, height = 0, width = 0.3, color = ~ Group)
```




```
gf_jitter(Group ~ Taps, data = CaffeineTaps, width = 0, height = 0.3, color = ~ Group)
```



If we had more data, we might consider using * violin plots * overlaid density plots * faceted histograms, ...

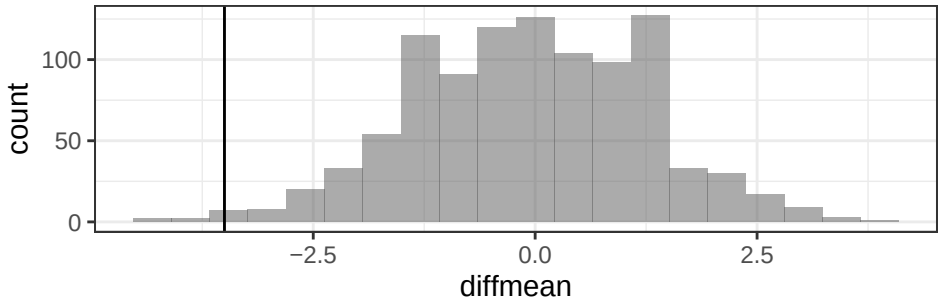
Generating the randomization distribution. Key idea: if the null hypothesis is true, then whether a person had caffeine doesn't matter. So we can shuffle those labels to get our randomization distribution.

```
set.seed(1234)
Taps.null <-
  do(1000) * diffmean(Taps ~ shuffle(Group), data = CaffeineTaps)
```

```
gf_histogram(~ diffmean, data = Taps.null, bins = 20) %>%
  gf_vline( xintercept = -3.5)
```

Determining the p-value

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```




```
prop( ~(diffmean <= -3.5), data = Taps.null)
```

```
## prop_TRUE
##     0.007
```

```
df_stats( ~(diffmean <= -3.5), data = Taps.null, props)
```

```
##                response prop_FALSE prop_TRUE
## 1 I(diffmean <= -3.5)       0.993      0.007
```

Again we double this to get our p-value because we are doing a 2-tailed test.

```
# p-value
2 * prop( ~(diffmean <= -3.5), data = Taps.null)
```

```
## prop_TRUE
##     0.014
```

That's a pretty small p-value, so we have evidence against the null hypothesis: It appears the mean number of taps differs between our two treatment groups. But it doesn't differ by very much. (We could create a

confidence interval for that difference, if we wanted.) A small p-value doesn't necessarily mean that the difference is important or meaningful – just that it is bigger than we would expect by chance alone.

Fair Coin?

Data: 57 heads in 100 flips.

1. $H_0 : p = 0.5$; $H_a : p \neq 0.5$
2. Test statistic: $\hat{p} = 0.57$.

0.57 is larger than 0.50, but we need to know whether it might have happened just by chance even though $p = 0.5$.

3. To get our p-value, we simulate world where H_0 is true using `rflip()`.

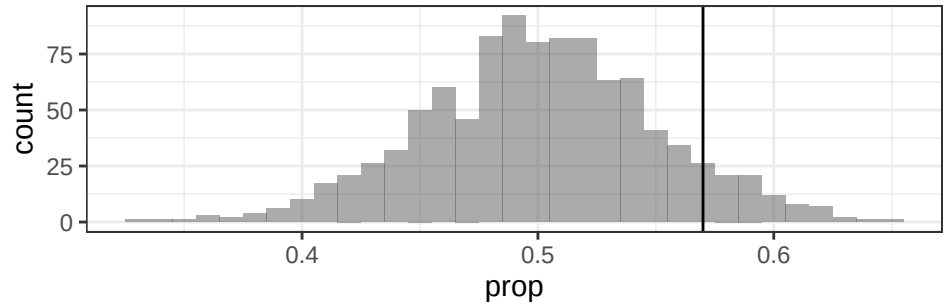
```
set.seed(1234)
Coin.rand <- do(1000) * rflip(100)
head(Coin.rand, 3)   # first 3 rows
```

```
##     n heads tails prop
## 1 100    52    48 0.52
## 2 100    52    48 0.52
## 3 100    48    52 0.48
```

Now we check how unusual our $\hat{p}$ is by comparing it to our randomization distribution.

```
gf_histogram( ~ prop, data = Coin.rand, binwidth = 0.01) %>%
  gf_vline(xintercept = 0.57)
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```




```
tally( ~(prop >= 0.57), data = Coin.rand)   # counts
```

```
## (prop >= 0.57)
##  TRUE FALSE
##    99   901
```

```
prop( ~(prop >= 0.57), data = Coin.rand)    # proportion
```

```
## prop_TRUE
##     0.099
```

Since we are doing a 2-tailed test, we will double the proportion we just computed to get our p-value.

```
2 * prop( ~(prop >= 0.57), data = Coin.rand)    # p-value
```

```
## prop_TRUE
##     0.198
```

Do dogs resemble owners?

To test this, researchers had 25 subjects attempt to match dogs with owners based on photos (one person, two dogs). 16 of 25 people correctly matched the owner to the dog.

The dogs and owners situation is nearly the same as the coin tossing example above. Again our null hypothesis is

- H_0: $p = 0.5$.

But this time it makes more sense to do a one-tailed test, since we are only looking for evidence that people do better than guessing. So

- H_a: $p > 0.5$.

As before we use `do()` and `rflip()` to generate our randomization distribution. Notice that we flip only 25 coins per randomization distribution now, since that's the number of subjects in the original sample.

```
set.seed(13)
16/25   # this is p-hat
```

```
## [1] 0.64
```

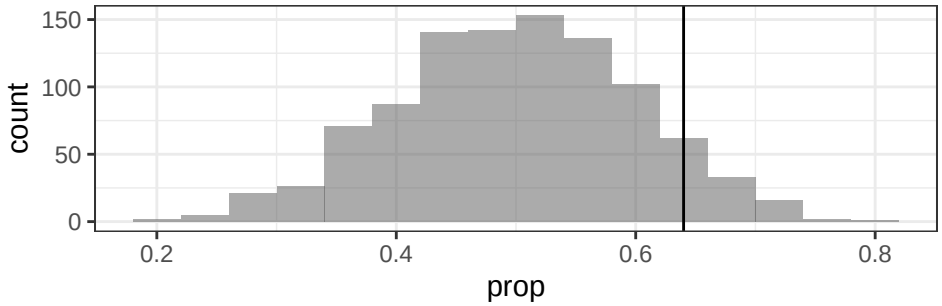
```
Dogs.null <- do(1000) * rflip(25)
head(Dogs.null, 3)      # first three rows
```

```
##    n heads tails prop
## 1 25    13    12 0.52
## 2 25     5    20 0.20
## 3 25    14    11 0.56
```

Now we compare $\hat{p} = 0.64$ to the randomization distribution.

```
gf_histogram(~prop, data = Dogs.null, binwidth = 0.04) %>%
  gf_vline(xintercept = 0.64)
```

```
## Warning: geom_vline(): Ignoring `mapping` because `xintercept` was provided.
```




```
prop( ~(prop >= 0.64), data = Dogs.null)  # p-value
```

```
## prop_TRUE
##     0.114
```

No need to double this time since we are doing a 1-tailed test.