Lecture 31: Permutation Tests

Instructor: Tomer Altman

November 12, 2025

Permutation tests

The methods we've used so far for hypothesis testing (z-tests, t-tests, and chi-square tests) have depended on having large enough sample sizes for the inference to be valid. They have also required that the sample was an SRS from some larger population.

Today we will talk about another method for conducting hypothesis tests that do not require either assumption.

It might remind you of our bootstrapping lecture, but remember, bootstrapping was for confidence intervals, whereas permutation tests are for hypothesis testing.

Example: Drinking beer and mosquito bites

Background: Malaria and alcohol consumption both represent major public health problems. Alcohol consumption is rising in developing countries and, as efforts to manage malaria are expanded, understanding the links between malaria and alcohol consumption becomes crucial. Our aim was to ascertain the effect of beer consumption on human attractiveness to malaria mosquitoes in semi field conditions in Burkina Faso. - Lefevre et al, 2010, in PLOS One

Example: Drinking beer and mosquito bites

- Volunteers were randomly assigned to drink either beer or water
- Batches of mosquitoes were inside a device and could choose to fly towards the volunteer or towards the open air
- The number of mosquitoes flying towards each volunteer was counted

Example: Drinking beer and mosquito bites

The data:

```
beer <- c(27, 19, 20, 20, 23, 17, 21, 24, 31, 26, 28, 20,
          27, 19, 25, 31, 24, 28, 24, 29, 21, 21, 18, 27,
          20)

water <- c(21, 19, 13, 22, 15, 22, 15, 22, 20, 12, 24, 24,
           21, 19, 18, 16, 23, 20)

# (Students, don't need to know how to write
# the following lines of code)
mosq_data <- data.frame(num_mosquitos = c(beer, water),
                        treatment = c(rep("beer", 25),
                                      rep("water", 18)))
```

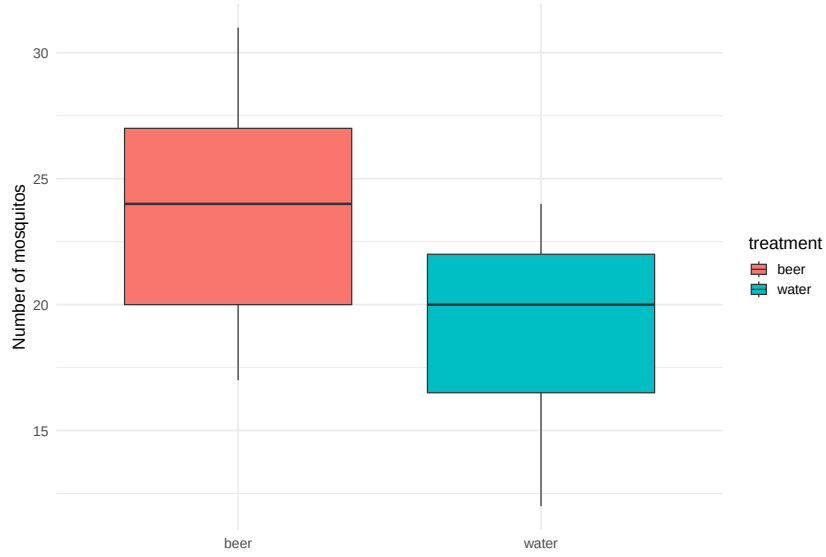
```
head(mosq_data)
```

```
##   num_mosquitos treatment
## 1            27      beer
## 2            19      beer
## 3            20      beer
## 4            20      beer
## 5            23      beer
## 6            17      beer
```

- num_mosquitos is the count of mosquitoes the flew towards the participant
- treatment is whether the person was randomized to water or beer

Example: Drinking beer and mosquito bites

Does there look to be a difference between the groups?



Example: Drinking beer and mosquito bites

Which test that we already know could we use to test whether there is a difference between the number of mosquitoes attracted to beer and water drinkers?

Example: Drinking beer and mosquito bites

Which test that we already know could we use to test whether there is a difference between the number of mosquitoes attracted to beer and water drinkers?

The average number of mosquitoes attracted to beer drinkers was 23.6 vs. 19.22 attracted to water drinkers. The p-value was 0.07% which is very small. There is evidence in favor of the alternative that there is a difference in the average number of mosquitoes attracted to beer drinkers and water drinkers.

```
t.test(beer, water, alternative = "two.sided")
```

```
##
##  Welch Two Sample t-test
##
## data:  beer and water
## t = 3.6582, df = 39.113, p-value = 0.0007474
## alternative hypothesis: true difference in means is not e
## 95 percent confidence interval:
##  1.957472 6.798084
## sample estimates:
## mean of x mean of y
##  23.60000  19.22222
```

Check your understanding!

Example: Drinking beer and mosquito bites

There is another way to perform this test. Consider the null hypothesis:

$$H_0 : \mu_1 = \mu_2$$

If the two means are the same, then we would expect no difference between the number of mosquitoes attracted to beer drinkers vs. water drinkers.

Assuming the null is true: **We could mix up the labels of who drank beer and water and re-compute the difference between beer drinkers and water drinkers in the number of mosquitoes.**

We could do this many times. For each shuffling of the labels, we could re-compute the difference and mark it on a histogram.

Example: Drinking beer and mosquito bites

Watch this clip from 8:13-9:52:
https://youtu.be/5Dnw46eC-0o?t=492.

- It shows the sampling distribution being built for this example under the null hypothesis of no difference
- It shows how the labels can be shuffled at random, and after each re-shuffling, the mean difference is computed and plotted on an evolving histogram
- Then a vertical line is added at the **observed** value of the difference (based on the data from the sample)
- An observed value in the tails of the distribution implies that it is unlikely to occur under the null hypothesis of no difference between the groups

The `infer` package

The `infer` package is relatively new to the tidyverse (which includes `ggplot2`, `readr`, `dplyr`, among others).

It is **awesome** because it interjects the steps of hypothesis testing directly into the code. It also keeps things "tidy" meaning that the output is often returned in a nice little data frame.

We will use `infer` to conduct permutation tests, but if you're interested you could also learn more here about doing all your statistical hypothesis testing using this package.

Let's have a look!

The infer package for permutation tests

First use the infer functions specify(), hypothesize(), generate(), and calculate to create the histogram of the sampling distribution for the mean difference:

```
library(infer)

null_distn <- mosq_data %>%
  specify(response = num_mosquitos, explanatory = treatment) %>%
  hypothesize(null = "independence") %>%
  generate(reps = 1000, type = "permute") %>%
  calculate(stat = "diff in means", order = c("beer", "water"))
```

```
head(null_distn)
```

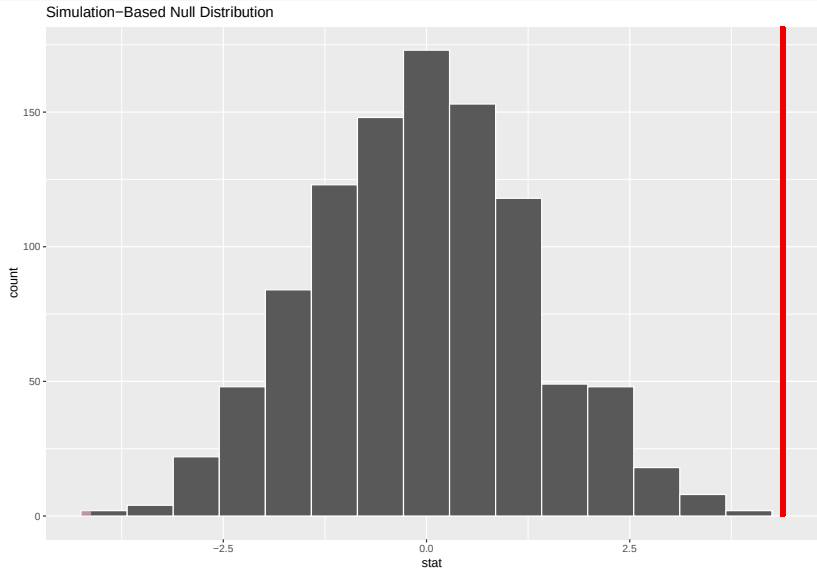
```
## # A tibble: 6 x 2
##   replicate   stat
##       <int>  <dbl>
## 1         1  0.173
## 2         2 -1.74
## 3         3 -0.304
## 4         4  1.51
## 5         5 -1.64
## 6         6  0.364
```

You won't be tested on the code for the infer package on the final exam, though you might need to write it on your next assignment. For the final, just understand the essence between how a permutation test works and the steps to conduct a permutation

The `infer` package for permutation tests

- Use the `infer` function `visualize` to plot the sampling distribution, add a line at the observed mean difference, and shade the region corresponding to the *p*-value
- Note, one of the permutations is the original observed data
- If you do enough permutations, will have at least one of the permutation-based statistics be $\geq$ the test statistic of the observed data

```
visualize (null_distn, method
  shade_p_value(23.6-19.22, di
```



The infer package for permutation tests

Finally, calculate the *p*-value by using the get_pvalue() function:

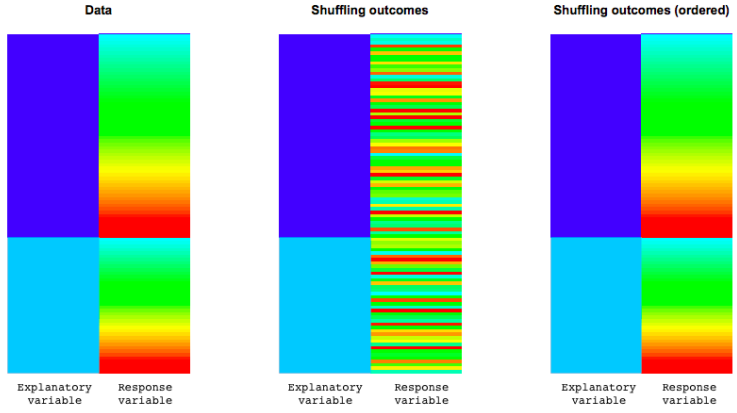
```
null_distn %>% get_pvalue(obs_stat = 23.6-19.22, direction
```

```
## Warning: Please be cautious in reporting a p-value of 0.
## approximation based on the number of `reps` chosen in th
## See `?get_p_value()` for more information.
```

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

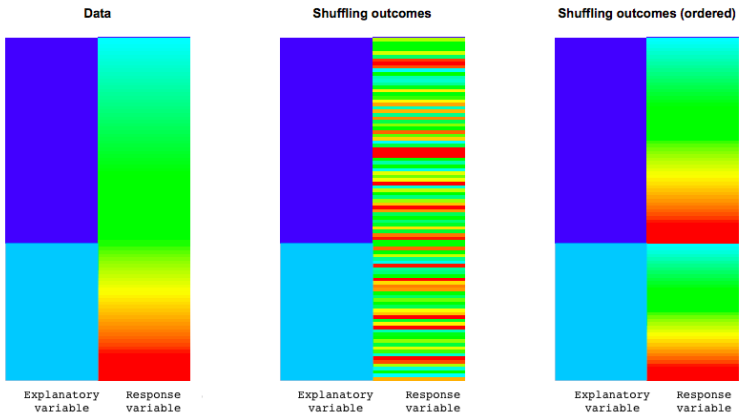
Permutation test, shown visually

If the null is *true* then the distribution of the response variable is the same for each level of the explanatory variable. This is shown by the entire spectrum of colors for both levels of the explanatory variable in this plot.
After reshuffling, the distribution comes out the same. This illustrates that if the null is true, your observed statistic will look like a random reshuffle.
Reference:
http://faculty.washington.edu/kenrice/sisg/SISG-08-06.pdf



Permutation test, shown visually

If the null is *false* the distribution of the response variable varies for each level of the explanatory variable. This is shown by the one level corresponding to the "blue-green" part of the response variable and the other level corresponding to "red-yellow". After reshuffling, the observed data looks very different from the random reshuffle.
reference:
http://faculty.washington.edu/kenrice/sisg/SISG-08-06.pdf



Another example

- So far, we've used the permutation approach to examine whether the observed difference indicated a true difference between the means of two continuous variables
- We can use permutation tests to look at all kinds of data, including categorical data

Back to the smoking example from last class

```
library(tibble)
two_way <- tribble(~ smoking, ~ non_smoking,
                   12,        238,
                   7,         743)

#We can put the data from the 2X2 table into a data frame
smoke_data <- data.frame(id = 1:1000,
                         smoking = c(rep("yes", 19),
                                     rep("no", 238+743)),
                         lung_cancer = c(rep("yes", 12),
                                         rep("no", 7),
                                         rep("yes", 238),
                                         rep("no", 743)))

# Take a look at it in the Viewer. You'll see there are 12
# people who smoke with lung cancer and so on, as specified
# by the 2X2 table.
```

```
head(smoke_data)
##   id smoking lung_cancer
## 1  1     yes         yes
## 2  2     yes         yes
## 3  3     yes         yes
## 4  4     yes         yes
## 5  5     yes         yes
## 6  6     yes         yes
```

Permutation test on the smoking data

Can we do a permutation test using these data?

Permutation test on the smoking data

Can we do a permutation test using these data?

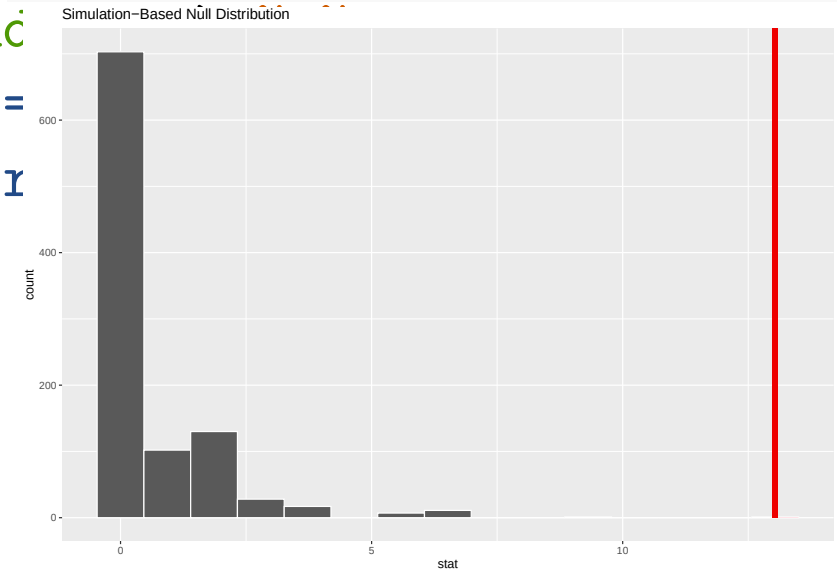
Yes! The method is strikingly similar, even though we have categorical data rather than continuous data. We just need to shuffle/permute the labels to break the association between smoking and lung cancer.

What statistic will we calculate? We can still calculate the chi-square statistic for each of the permutations and make a histogram of those values to get our p-value.

Permutation test on the smoking data

```
null_distn <- smoke_data %>%
  specify(lung_cancer ~ smoki
  hypothesize(null = "independ
  generate(reps = 1000, type =
  calculate(stat = "Chisq", or
```

```
null_distn %>% visualize() +
  shade_p_value(obs_stat = 13.
```



Permutation test on the smoking data

```
# the obs_stat is the observe
# from last class, you can also get it using this code:
smoke_data %>%
  specify(lung_cancer~smoking, success = "yes") %>%
  calculate(stat = "Chisq", order = c("yes", "no"))
```

```
## # A tibble: 1 x 1
##    stat
##   <dbl>
## 1  13.0
```

```
null_distn %>% get_pvalue(obs_
```

```
## Warning: Please be cautious
## approximation based on the
## See `?get_p_value()` for mo
## # A tibble: 1 x 1
##   p_value
##     <dbl>
## 1       0
```

The probability is 0 based on the permuted dataset because there are no values in the permutation that were larger than 13.04.

Relationship of Fisher's Exact Test to Permutation

- In the case of testing the null hypothesis of two categorical variables (as we discussed in Lecture 30), there is another test that is equivalent to the permutation test, only a bit better
- In this case, the number of different permutations can be counted easily because their distribution is known (its the hypergeometic distribution)
- So, you can think of the Fisher's exact test in this context as just the same as the permutation test, but it does *all* possible permutations (as opposed to selecting a large number, say $1,000$).

Fisher's Exact Test, continued

- Like the `chisq.test` function, the fisher's exact test function in R (`fisher.test`) takes as input a contingency table
- It selects one of the cells as the test statistic and since it is based on fixing the marginal row and column proportions, if one knows one cell, one knows them all (see last lecture)
- Thus, it can do a test of $H_0 : X$ independent of Y by looking at the observed cell to the permutation distribution to derive a p-value
- It can report the results in various ways, but often will convert the cell value to an odds ratio, or $\frac{\frac{P(L|Smoke)}{(1-P(LC|Smoke))}}{\frac{P(L|NoSmoke)}{(1-P(LC|NoSmoke))}}$
- H_0 : Smoking is independent of LC is the same as $H_0 : OR = 1$
- Don't worry, we'll just concentrate on the p-value

Fisher's Exact Test in R

```r
## Get contingency table
tble <- table(smoke_data$smoking,smoke_data$lung_cancer)
tble
```

```
##
##         no yes
##   no   743 238
##   yes    7  12
```

```r
## Put into the fisher.test function
fisher.test(tble)
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  tble
## p-value = 0.0004243
## alternative hypothesis: true odds ratio is not equal to
## 95 percent confidence interval:
##   1.912951 16.201979
```

Difference from the permutation test done above

- Note that the p-value is not 0
- That is because there is at least one test statistic in the null permutation distribution that is greater than or equal to the observed test statistic
- The permutation test above only did $1,000$ permutations and missed these
- One could change the number of permutations to $10,000$ and try again
- Not terribly important difference since the inferences are the same (reject the null even at very conservative α-level)

Check your understanding!

Assumptions

- The only assumption required for permutation tests is exchangeability
- For randomized or experimental designs this assumption is met by definition
- For observational studies, it is a bit trickier, but essentially if there is a confounder that is unmeasured or not adjusted for, then exchangeability is not met
- For example, suppose that those who drank water also applied DEET spray (which repels mosquitoes) and those who drank beer did not. Then, even if we "break" the link between treatment and the outcome by shuffling the outcomes there is still a link between DEET spray use and the outcome that will confound the association between treatment status and number of mosquitoes

In summary

- Permutation tests are another way to get p-values for hypothesis tests
- There is a permutation test equivalent for all the two sample tests that we've covered. They each rely on reshuffling (or permuting) the data to break any relationship between the two variables
- The `infer` package is a good way to conduct and visualize permutation tests in R
- Very useful when the assumptions of the related standard tests are not met (e.g., sample size too small so that CLT can not be invoked)
- Fisher's Exact Test uses exhaustive permutations; can work with small datasets that won't meet the chi-square test assumptions