Lecture 8: t-tests in R

Contents

Learning Objectives

- Learn how to use R to test for a difference in population means across two groups.
- Reproduce the results that R gives you "by hand" to ensure that you understand where R's results come from.
- Get mini-quiz credit for attendance and effort/progress.

Loading the data

We will work with the `nycflights` dataset from the `openintro` package. Please start by loading the required packages and the dataset.

```
library(tidyverse)
library(openintro)
data(nycflights)
```

Even though this dataset is massive, the `openintro` documentation says that this is a random sample of flights from the population of all flights that departed from NYC in 2013. To see the documentation, you can type `?nycflights` in your console.

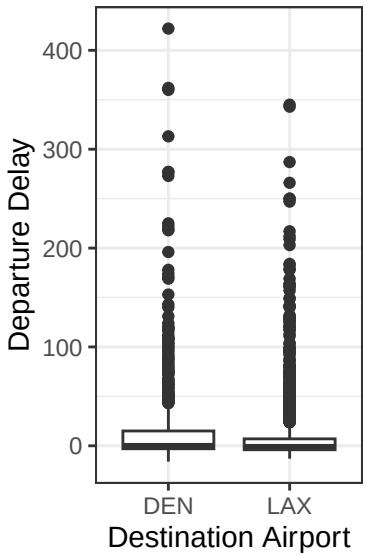
Setting up a research question

I would like to know if flights headed to Denver are more delayed, on average, than flights headed to Los Angeles.

Since my research question relates only to a certain subset of flights in my dataset, my first step is to subset my dataset.

```
nycflights_subsamp <- nycflights %>%
  filter(dest=="LAX" | dest == "DEN")
```

Next, I can either visually or numerically compare the distribution of flights headed to Los Angeles vs. Denver in my sample. I decided to do this with boxplots, but you could do this however you choose. I came up with the following plot:



Based on the plot, I can tell that there is a difference in means between flights headed to `LAX` and `DEN` in my sample. However, I cannot tell if this is driven by a true underlying population difference, or just a few unlucky outliers in my sample.

We will use a t-test to get to the bottom of whether or not there is evidence of an underlying population difference here.

A hypothesis test for a difference in means

Let μ_1 be the population average of delay time for flights headed from NYC to Denver. Let μ_2 be the population average of delay time for flights headed from NYC to Los Angeles. The null hypothesis is

$$H_0 : \mu_1 - \mu_2 = 0.$$

We want to see if our sample provides evidence that $\mu_1 - \mu_2 \neq 0$. We can actually get R to do all of this work for us. The following line of code will conduct a t-test to see if `dep_delay` differs by destination airport in our subsetted dataset.

```
t.test(dep_delay~dest, data=nycflights_subsamp)
```

```
##
##  Welch Two Sample t-test
##
## data:  dep_delay by dest
## t = 3.4621, df = 1118.2, p-value = 0.0005562
## alternative hypothesis: true difference in means between group DEN and group LAX is not equal to 0
## 95 percent confidence interval:
##   2.816468 10.184631
## sample estimates:
## mean in group DEN mean in group LAX
##         16.282609          9.782059
```

When you run this code, it prints out a lot of information for you!

At a high level, we see that the p-value is equal to 0.00055. You should recognize that this means we *reject* the null hypothesis, and conclude that there is convincing evidence of a difference in means.

But how did `R` come up with this p-value? And what is all of the other information that is printed? We will dive into this below!

Your task in groups

Your task is to reproduce all of the numbers in the `t.test()` output using the dataset and the theory that we learned in class.

1. Your sample statistic should have the form $\bar{x}_1 - \bar{x}_2$. Find the values of $\bar{x}_1$ and $\bar{x}_2$ directly from your dataset, and save then in `R` to variables called `xbar1` and `xbar2`. Do these show up in the `t.test()` output?

2. R is basing this t-test off of a sampling distribution, whose standard deviation is supposed to be

$$\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}.$$

   However, σ_1 and σ_2 are both unknown. Obtain estimates from them using your data, and save the result to variables called `s1` and `s2` in R. While you are at it, also save the appropriate values of n_1 and n_2 to `R` as `n1` and `n2`. Do any of these numbers appear in the `t.test()` output?

3. Finally, plug all of your estimates in to obtain your estimated standard deviation of your sampling distribution:

$$\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}.$$

   be sure to save this value in `R`, to a variable called `se`.

4. The `t.test()` output prints out `t=3.46`. Based on what you have calculated so far, reproduce the number 3.46. On your mini quiz, explain briefly what you did.

5. When we do a two-sample t-test for a difference in means and the two samples have different sizes and different standard deviations, a complicated formula tells us how many degrees of freedom our t-test has. In this case, I would argue that the complicated formula does not matter. Looking at the output of `t.test()` where it says `df=1118.2`, why might I argue that this does not matter?

6. Use `R` to reproduce the p-value given by the `t.test()` output. You may *either* use the `pt()` function and tell it that `df=1118.2`, or you may simply use `pnorm()`. You can even try both and compare your answers. On your mini quiz, briefly explain how you computed the p-value. A picture might help.

7. Use `R` to reproduce the confidence interval printed by '`t.test()`. Once again, you may use either a `t`-multiplier (with the appropriate degrees of freedom) or a `z`-multiplier. On your mini quiz, explain briefly how you did this, and interpret your confidence interval in your own words.