More on t, part 1

Chriss Jordan Oboa

4/20/2022

Packages

We load our usual packages: `tidyverse` as a general tool, and `infer` to help with random sampling.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(infer)
```

The challenge of confidence intervals for small samples

We continue study confidence intervals based on a sample mean. As we saw in an earlier activity, our usual procedure for 95% intervals (point estimate +/- 2 standard errors) does not work as promised when we use the sample standard deviation to estimate the standard error.

We start by loading the same population from our first activity. This population is close to what Gosset used in his work.

```
population <- read_csv("data/population.csv")
```

```
## Rows: 3000 Columns: 1
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (1): height
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

We again store the population mean and population standard deviation with the names `mu` and `sigma`.

```
mu <- mean(population$height)
sigma <- sd(population$height)
```

Gosset's big idea

The basis for our method to make, say, 95% confidence intervals is the following:

- Sampling distributions for many statistics (e.g. sample means and sample proportions) are approximately normal.
- Assuming that the sampling distribution is normal, for almost all (95% of) samples, the sample statistic is within 2 standard errors of the population parameter.
- Or, equivalently, for almost all (95% of) samples, the population parameter is within 2 standard errors of the sample statistic.

When we study a numerical variable and want to estimate a population mean (`mu`) using a sample mean (`x_bar`), we use the sample standard deviation (`s`) to estimate the standard error. In a previous activity, we saw that making confidence intervals for the population mean using samples this way does not seem to work work for 95% of samples!

Gosset thought about this kind of issue carefully. He had the idea of graphing another kind of sampling distribution: for each sample, he figured out how many approximate standard errors (`s / sqrt(n)`) the population mean was from the sample mean. If we were using the true standard error, we would find that 95% sample means would fall within 2 standard errors of the population mean (or that for 95% of samples, the population mean is within 2 standard errors of the sample mean). Gosset found consistent patterns for his new sampling distributions; we now call these patterns "Student's t-distributions" in his honor. ("Student" was Gosset's pseudonym. You will learn from a reading why he used a pseudonym.)

Trying out Gosset's idea

Let's try out Gosset's idea:

1. Draw a sample of size 4 from the population.

```
samp1 <- population %>%
  slice_sample(n = 4)
```

2. Look at your sample by clicking on samp1 in the Environment pane to the right.
3. Use your sample to estimate the standard error for sample means with samples of size 4. How does your estimated standard error (`s / sqrt(4)`), which depends on your individual sample, compare with the actual standard error (`sigma / sqrt(4)`), which is the same for everyone's sample?

```
samp1 %>%
  summarize(sd(height) / sqrt(4))
```

```
## # A tibble: 1 x 1
##   `sd(height)/sqrt(4)`
##                  <dbl>
## 1                 2.20
```

```
sigma / sqrt(4)
```

```
## [1] 1.262883
```

The next chunk stores the samples standard deviation and sample mean for later use. Take note of what we are calling them.

```
x_bar <- mean(samp1$height)
s <- sd(samp1$height)
```

4. Calculate two statistics from your sample. Take note of their definitions - write them down for yourself.

- `z`, the number of true standard errors above the population mean that your sample mean falls. (If your sample mean is less than `mu`, then your `z` statistic will be negative.)

```
(x_bar - mu) / (sigma / sqrt(4))
```

```
## [1] 0.5969792
```

- `t`, the number of estimated standard errors above the population mean that your sample falls. (If your sample mean is less than `mu`, then your `t` statistic will also be negative.)

```
(x_bar - mu) / (s / sqrt(4))
```

```
## [1] 0.3432378
```

5. Report your results in the form on Blackboard. Once we have enough data, we will look at the distributions. Please repeat this process with a few new samples to add to our collection of data values.