M5-07: t-Tests in Python

Part of the "Polling, Confidence Intervals, and the Normal Distribution" Learning Badge

Video Walkthrough: https://discovery.cs.illinois.edu/m5-07/

Running t-Tests in Python

Similar to running a Z test, Python provides existing libraries to help us out! Searching for a t-test using the search query pattern we used before:

Search Query:	`python 1 sample t test`

I find a function from the scipy library:

scipy.stats.ttest_1samp(a, popmean, axis=0, nan_policy='propagate', alternative='two-sided')

Calculate the T-test for the mean of ONE group of scores.

Source: https://www.statsmodels.org/stable/generated/statsmodels.stats.weightstats.ztest.html

Puzzle #1: A car company claims that their *Super Spiffy Sedan* averages 31 mpg. You randomly select 8 Super Spiffies from local car dealerships and test their gas mileage under similar conditions. Assume the gas mileage follows the normal curve. The following MPGs are provided from the eight tests:

31.7	29.8	30.3	32.4
31.2	28.1	29.6	30.0

Use the ttest_1samp function to run a ttest on the dataset:

Python:	
Result:	

Analysis: Does the data you collect suggest there may be evidence to suggest that the true MPG may not be 31?