Help Us Translate

0:04 Hi, and welcome to another Python tutorial and the statistics with Python course. Throughout week two, we learned how to do statistical inference with confidence intervals, how to interpret them, and what does 95 percent confidence truly mean. In this tutorial we'll be going over some of the examples that we discussed in lecture, and also I'll be showing you how you can streamline the calculation of confidence intervals utilizing built in Python libraries. Let's do it. Now to begin, let's go over some of the material from this week and why confidence intervals are useful tools when deriving insights from data. So, why confidence intervals? Well, confidence intervals are calculated range or boundary around a parameter or statistic that is supported mathematically within a certain level of confidence. For example, in the lecture, we estimated with 95 percent confidence that the population proportion of parents with a toddler that use a car seat for all travel with their toddler was somewhere between 82.2 percent and 87.7 percent for the population of parents with toddlers. Now, I will do want to note that this is different than having a 95 percent probability that the true population proportion is within our confidence interval. As we learned in lecture, this essentially only means that if we were to repeat this process, a bunch of times, 95 percent of our calculated confidence intervals would contain the true population proportion, which is in between our calculated range. So, how are these confidence intervals calculated? Well, our k equation for calculating confidence intervals is as follows, and it may look pretty similar to what you saw in lecture. The best estimate plus or minus your margin of error. Where the best estimate is the observed population proportion or mean, and the margin of error is the t-multiplier times your standard error, which is not included in this notebook with the t-multiplier times your standard error. The t-multiplier is calculated based off of the degrees of freedom and desired confidence level. For samples with more than 30 observations and a confidence level of 95 percent, the t-multiplier is 1.996 just give you a reference point, and you can also calculate or find the t-multiplier if you look at your degrees of freedom, and a t-table and this will give you your desire t-multiplier if you were going to replicate this process with a different sample. So, the equation to create a 95 percent confidence level can be shown with the following equation as well. So, the population proportion or mean, which is our best estimate, plus or minus the t-multiplier times standard error. One thing that we also need to cover is how to calculate the standard error. As we know from much of the standard error is calculated differently for population proportion and for population means, and the two equations are listed right here as well for you. Now, I know that seems a little bit wordy. However, what we're going to do is we're actually to input some numbers and we'll go ahead and replicate the car seat example from lecture and the example that we mentioned above when we were discussing what confidence intervals mean. So, what we're going to do we're going to go ahead and just replicate and utilize the numbers that were given in lecture and go ahead and find our confidence intervals. To start, we're to go ahead and do the confidence interval for a population proportion of parents that always travel with a car seat with their toddlers. So, to start, we're going to import our libraries. So, we'll do import as np, and then

we're going to ahead and calculate the variables that we need to calculate the confidence interval. So, we need our standard error, which we know also needs to have the number of observations and our population proportion values and a t-multiplier. For this we know that our t-multiplier, which we'll denote as t-star, is 1.96, our population proportion is 0.85, and our number of observations is 659. So, what this is saying is that we observed of the 659 parents, 85 percent of them always travel with the car seat with their toddler. Now what we're going to go ahead and do is we're actually create the standard error for this population proportion, and the equation is as follows; so it's going to be numpy.operator square root, and it's going to be our population proportion times one minus our population proportion divided by our number of observations which is n. As you can see, that's the same equation as listed above. So, I'll go ahead and run that. First, we're got to import our library, and then we can run that, and we'll go ahead and output where the standard error is. So, our standard error is 0.0139. Now, this next is to calculate our actual lower and upper bounds of our confidence interval, and we can do this as follows. So, we'll do our lower confidence bound is equal to our population proportion minus for our lower bound, and then we'll do our t-multiplier which is t-star times our standard error. Then we'll do the same for our upper bound to population proportion plus t-star times standard error, and we'll go ahead and list all those values. So, lcb, ucb. As you can see, and this will actually if you go back to lecture, this is actually the same confidence interval that we saw in lecture, where our lower bound is 82 percent and our higher bound is 87 percent. Now, this is a little bit amount of work having to calculate the standard error on your own in the lower and upper bounds. Luckily, Python has a library that will actually do these calculations for you, and it's stats models. So, I'll go ahead and import this right now, statsmodels.api and then as sm. So, we can call the functions with just sm, and the function is actually smstats.proportion_confint.

6:36 What this is looking for is it's actually looking for the count of times that you see your observed value which would be, "Yes, I always travel with a car seat when with my toddlers in the car". To get that count, because we only have the proportion and our number of parents, we can just do n times our proportion and that will just give us the number of parents that always travel with a car seat with their toddler, and then the total number of parents in our sample. If we go ahead and run that, make sure we import stats models, and then go ahead and run that. You'll see that our confidence interval is actually the same as we calculated above. Now, we're going to do one more example here and we're going to do the calculate the confidence interval of a mean of a population. To do this, we're going to actually use the cartwheel dataset that was introduced in lecture, and we're going to get the mean of the cartwheel distance of our participants. So, what we're going to do is we're going to import pandas as pd and then read in our data frame. If you're not familiar with it, before we've used it. It has the age, gender, and gender group, glasses, glasses group, height, wingspan, cartwheel distance. If they completed the cartwheel and then a complete group just another way to denote yes or no and then the score of their cartwheel that the judges gave them. What we're going to do here is we're going to calculate the mean and the standard deviation, and then also get the number of observations that we

have, and we'll just go ahead and do that now. So, we'll get mean, which is equal to cartwheel distance and then we can do.mean, and then we can do standard deviation is equal to our data frame cartwheel distance.std,

8:35 and then n is just going to be the length of our data frame and that'll give us the number of observations.

8:44 If you want to see what they are, we can print out the mean, standard deviation, then also n which is 25. One thing that I do want to mention is that when we're calculating this confidence interval, our t-multiplier is actually going to be different. It's not going to be our standard 1.96, because we don't have 30 observations, which is the assumption that we have to make. So, actually if you look at your t-table and you look for degrees of freedom 24 and for a confidence level of 95 percent, our t-multiplier is actually going to be 2.064.

9:25 The next step is also to create the standard error for our mean. To do that we know it's different than the population proportion, it's standard deviation over the square root of our number of observations, and I'll print that out so you can see, it's three. Then we'll go ahead and create our lower and upper bounds like we did above. Mean minus t-star times standard error, upper bound equals mean plus t-star times standard error, and then we'll output those, lcb, ucb. We can see that we have 95 percent confidence that our true population mean is within 76.26 and 88.69. Now that we've calculated with our standard equations and arithmetic, I'm going to go ahead and show you another way to calculate our confidence intervals using another function in stats models, and it is called smstats.Descr, select descriptive stats W, and then as inputs, we put our data frame in our column value, so we do CW Distance, MCE, and then we do D or.zconfint mean, and this is telling the Python that we want the confidence interval of our mean for our cartwheel distance column, and go ahead and run that and we get 76.57 and 88.38. I'm deducing that these kind of calculations do vary a little bit, and it's most likely just because of the assumption that we made here that and rounding error. However, you can see that these are two perfectly fine ways to calculate your confidence intervals, and utilizing the Python library stats models is a good way to streamline your calculations. All right, that's going to do it for today. In the next tutorial, we're actually going more detail about cleaning up a dataset, looking at the data, and then constructing it in a way where we can actually calculate the confidence interval for the difference of two population proportions and two population means. It's going to be a good one and looking forward to seeing you there. Bye.