**Performing t-tests**

00:00 - 00:07

In the previous chapter, we calculated the z-score, which was a test statistic for a single variable.

**2. Two-sample problems**

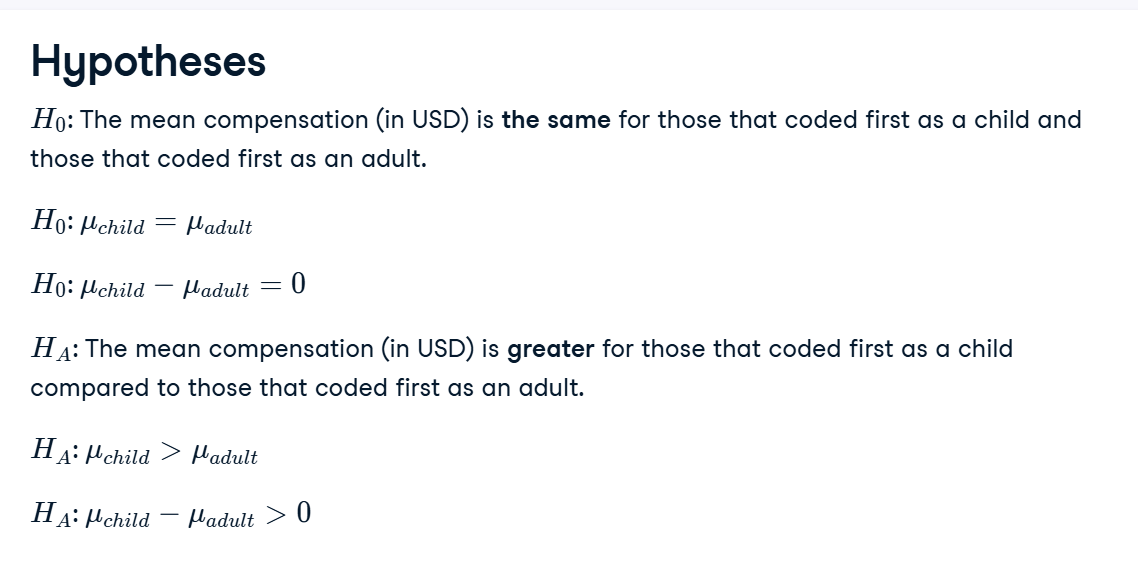
00:07 - 00:47

Here, we'll look at a related problem of comparing sample statistics across groups in a variable. In the Stack Overflow dataset, converted\_comp is a numerical variable of annual compensation. age\_first\_code\_cut is a categorical variable with two levels: child and adult, which describe when the user started programming. We can ask questions about differences in compensation across the two age groups, such as, are users who first programmed as a child better compensated than those that started as adults?

**3. Hypotheses**

00:47 - 01:28

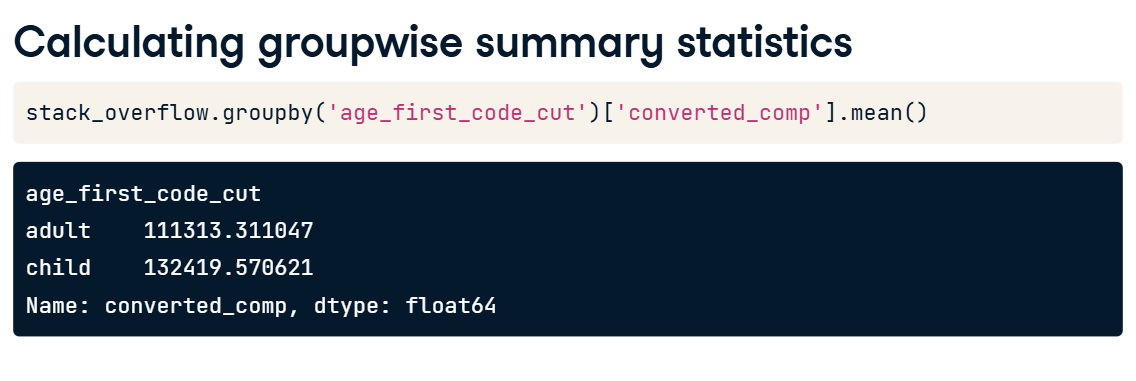
The null hypothesis is that the population mean for the two groups is the same, and the alternative hypothesis is that the population mean for users who started coding as children is greater than for users who started coding as adults. We can write these hypotheses using equations. Mu represents an unknown population mean, and we use subscripts to denote which group the population mean belongs to. An alternate way of writing the equations is to compare the differences in population means to zero. Zero here corresponds to our hypothesized value for the difference in means.



**4. Calculating groupwise summary statistics**

01:28 - 02:05

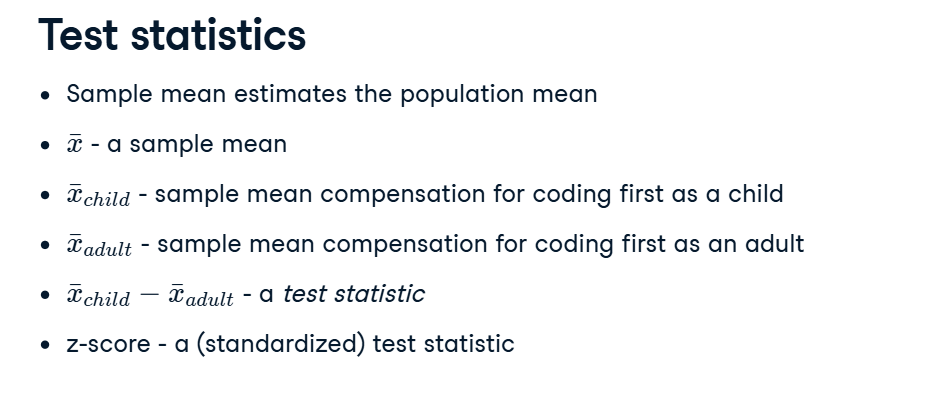
To calculate summary statistics for each group, we start with the sample, group by the categorical variable, and then compute on the numeric variable. A pandas way of doing this is shown, calculating the mean of the converted\_comp column after grouping by age\_first\_code\_cut. Here, the child programmers have a mean compensation of 132,000 dollars compared to around 111,000 for adult programmers. Is that increase statistically significant or could it be explained by sampling variability?



**5. Test statistics**

02:05 - 02:32

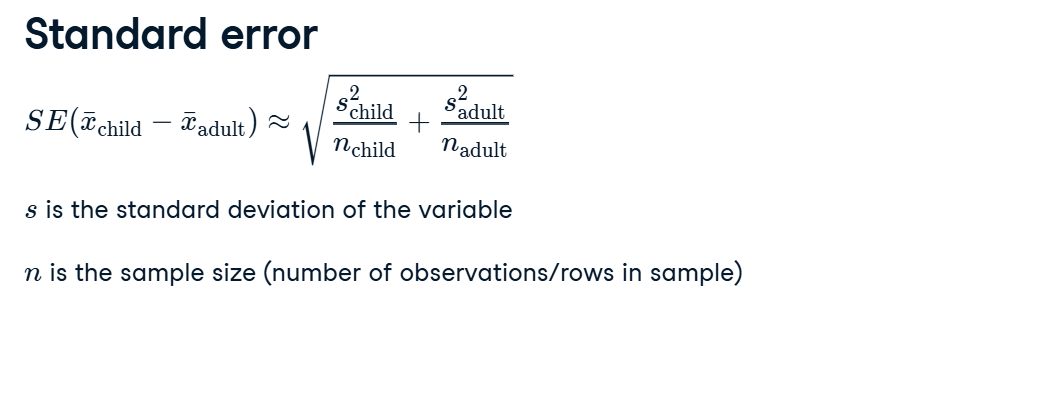
Although we don't know the population mean, we estimate it using the sample mean. x-bar is used to denote a sample mean. Then we use subscripts to denote which group a sample mean corresponds to. The difference between these two sample means is the test statistic for the hypothesis test. The z-scores we saw in Chapter 1 are a type of standardized test statistic.



**6. Standardizing the test statistic**

02:32 - 03:02

z-scores are calculated by taking the sample statistic, subtracting the mean of this statistic as the population parameter of interest, then dividing by the standard error. In the two sample case, the test statistic, denoted t, uses a similar equation. We take the difference between the sample statistics for the two groups, subtract the population difference between the two groups, then divide by the standard error.



**7. Standard error**

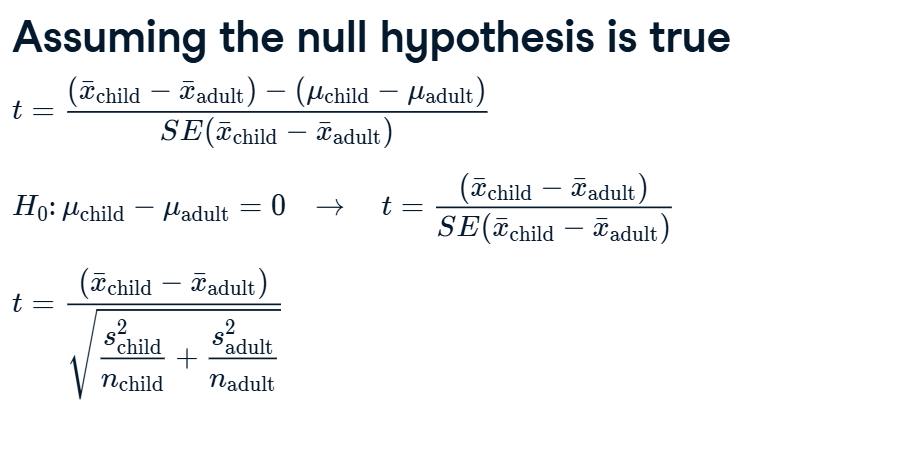
03:02 - 03:27

To calculate the standard error, needed for the denominator of the test statistic equation, bootstrapping tends to be a good option. However, there is an easier way to approximate it. We calculate the standard deviation of the numeric variable for each group in the sample, and the number of observations in each group. Then enter those values into the equation and compute the result.

**8. Assuming the null hypothesis is true**

03:27 - 03:59

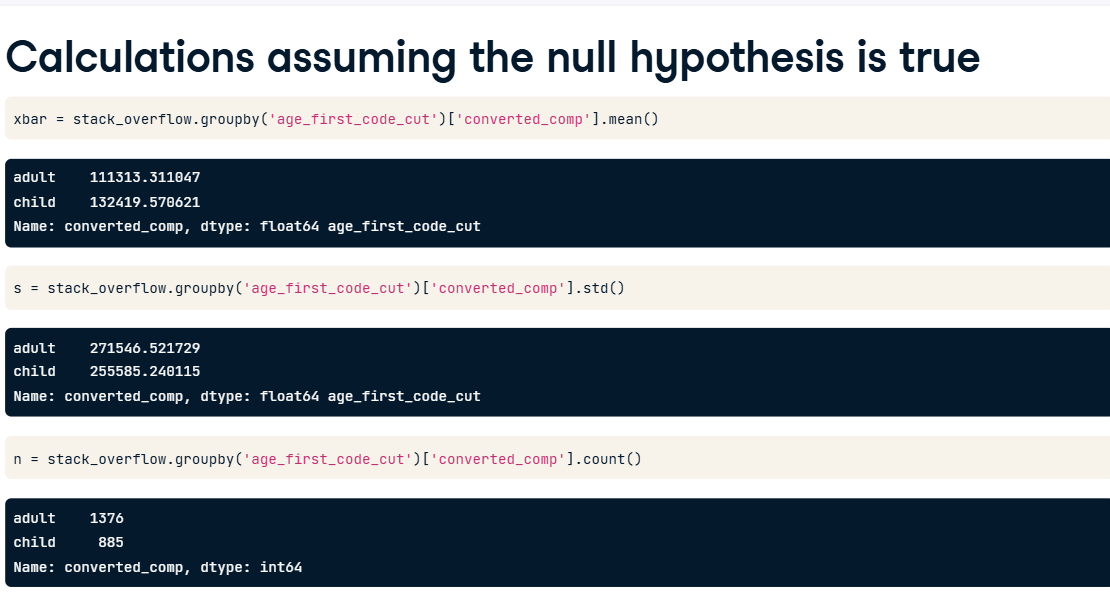
Here's the test statistic equation again. If we assume that the null hypothesis is true, there's a simplification we can make. The null hypothesis assumes that the population means are equal, and their difference is zero, so the population term in the numerator disappears. Inserting the approximation for the standard error, we now have a way of calculating the test statistic using only calculations on the sample dataset.



**9. Calculations assuming the null hypothesis is true**

03:59 - 04:15

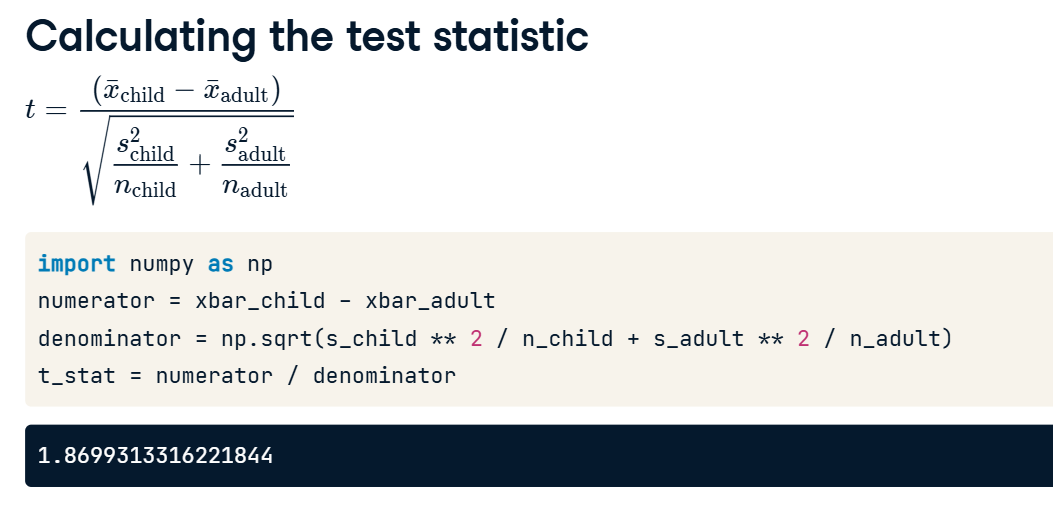
We need the mean, standard deviation, and number of observations for each group to fill in the formula for t. We again use groupby and method combinations with mean, std, and count.



**10. Calculating the test statistic**

04:15 - 04:38

Assigning the values to six different variables, the numerator is a subtraction of the sample means, and the denominator is like a weighted hypotenuse. The t-statistic is around one-point-eight-seven. Just as with z-scores, we can't draw any conclusions yet; for that, we'll need to wait for the next video.



**1. Calculating p-values from t-statistics**

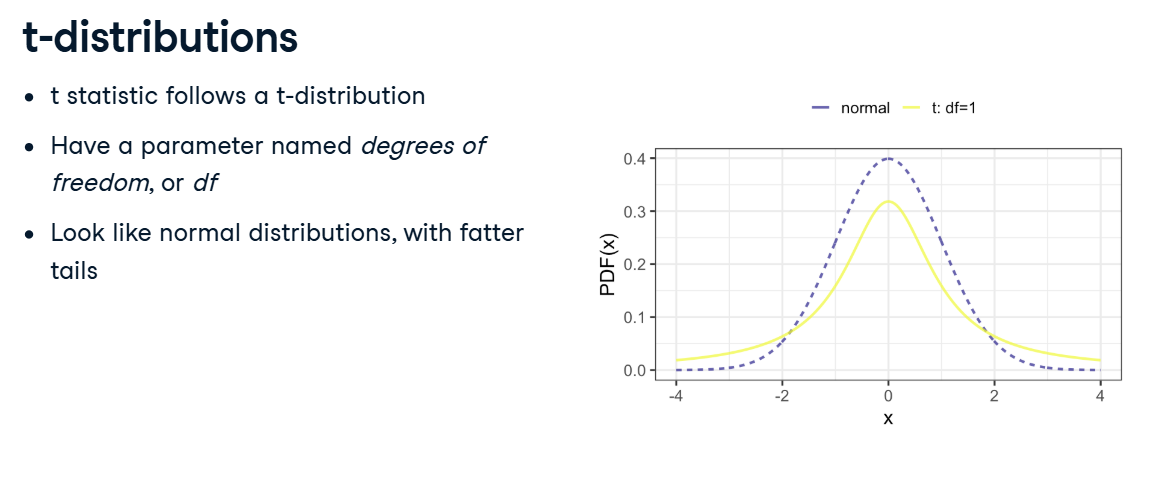
00:00 - 00:05

In the previous lesson, we calculated the test statistic t.

**2. t-distributions**

00:05 - 00:36

The test statistic, t, follows a t-distribution. t-distributions have a parameter called the degrees of freedom, or df for short. Here's a line plot of the PDF of a t-distribution with one degree of freedom in yellow, and the PDF of a normal distribution in blue dashes. Notice that the t-distribution for small degrees of freedom has fatter tails than the normal distribution, but otherwise they look similar.



**3. Degrees of freedom**

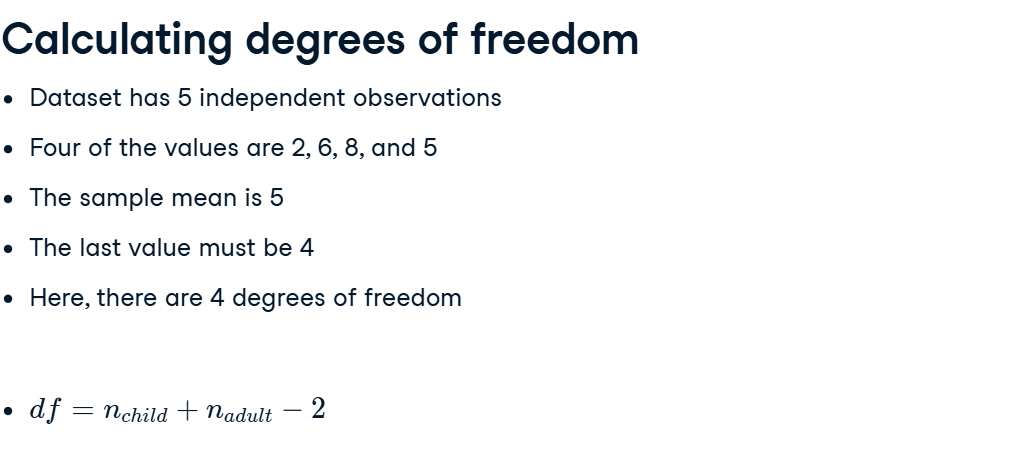
00:36 - 01:01

As we increase the degrees of freedom, the t-distribution gets closer to the normal distribution. In fact, a normal distribution is a t-distribution with infinite degrees of freedom. Degrees of freedom are defined as the maximum number of logically independent values in the data sample. That's a fairly tricky concept, so let's try an example.

**4. Calculating degrees of freedom**

01:01 - 01:45

Suppose our dataset has 5 independent observations, and that four of the values are 2, 6, 8, and 5. Suppose we also know the sample mean is 5. With this knowledge, the fifth value is no longer independent; it must be 4. Even though all five observations in the sample were independent, because we know an additional fact about the sample - that is has a mean of 5 - we only have 4 degrees of freedom. In our two sample case, there are as many degrees of freedom as observations, minus two because we know two sample statistics, the means for each group.



**5. Hypotheses**

01:45 - 01:58

Recall the hypotheses for our Stack Overflow question about compensation for the two age groups. Since this is a "greater than" alternative hypothesis, we need a right-tailed test.

**6. Significance level**

01:58 - 02:18

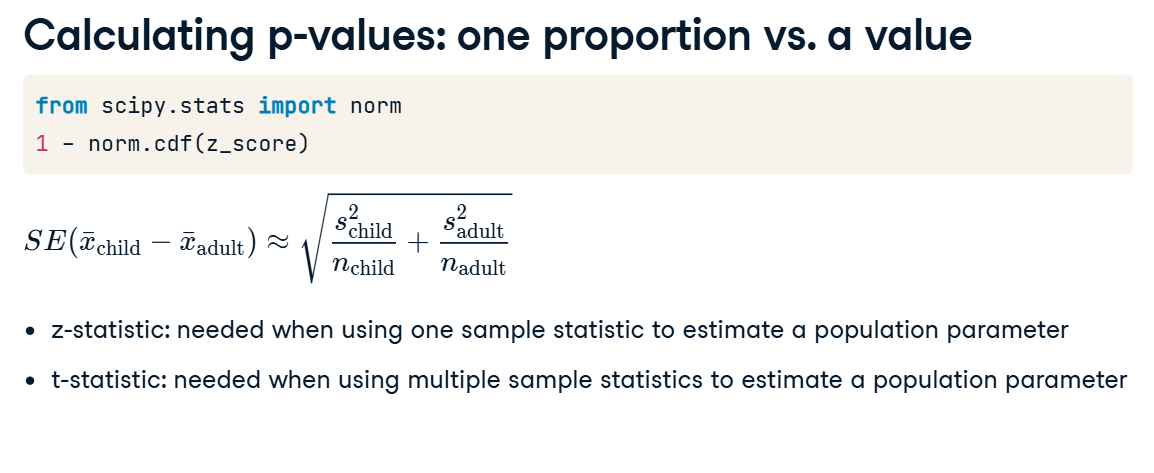
We're going to calculate a p-value in a moment, but we first need to decide on a significance level. There are several possibilities; let's use point-one. That means that we reject the null hypothesis in favor of the alternative if the p-value is less-than-or-equal-to point-one.



**7. Calculating p-values: one proportion vs. a value**

02:18 - 03:01

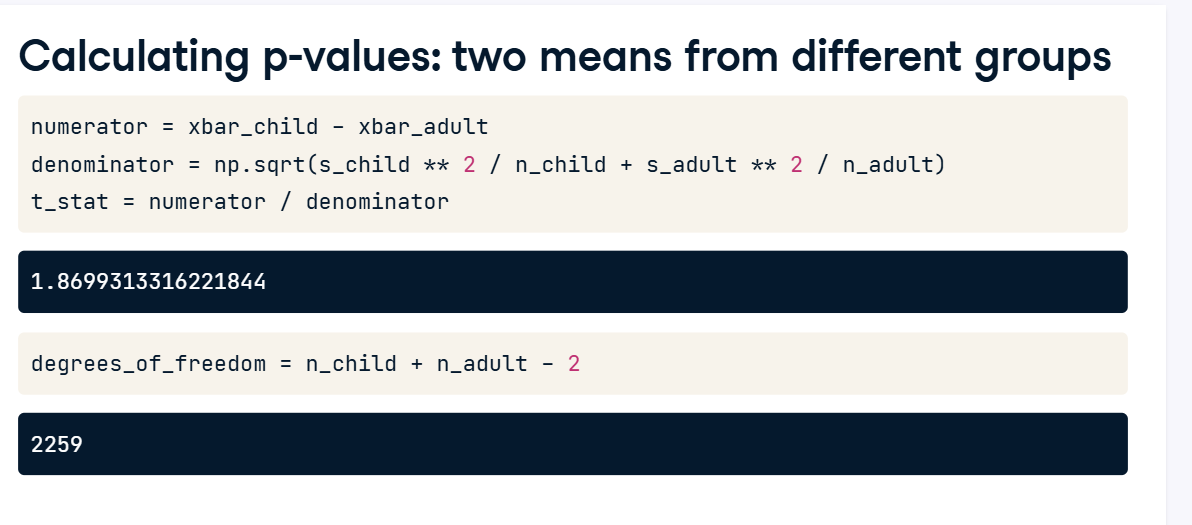
In Chapter 1, to get the p-value, we transformed the z-score with the normal CDF. Since it was a right-tailed test, we subtracted the result from one. In the previous video, we used an approximation for the test statistic standard error using sample information. Using this approximation adds more uncertainty and that's why this is a t instead of a z problem. The t distribution allows for more uncertainty when using multiple estimates in a single statistic calculation. Here, the multiple estimates correspond to the sample mean and the sample standard deviation.



**8. Calculating p-values: two means from different groups**

03:01 - 03:22

Now we are calculating means rather than proportions, the z-score is replaced with a t test statistic. This is the value calculated in the previous video. The calculation also needs the degrees of freedom, which is the total number of observations in both groups, minus two.



**9. Calculating p-values: two means from different groups**

03:22 - 03:57

To calculate the p-value, we need to transform the test statistic using the t-distribution CDF instead of the normal distribution CDF. Notice the use of t-dot-cdf instead of norm-dot-cdf, and that the df argument is set to the degrees of freedom. This p-value is less than the significance level of point-one, so we should reject the null hypothesis in favor of the alternative hypothesis that Stack Overflow data scientists who started coding as children earn more.



**Paired t-tests**

Previously, we used the t-distribution to compute a p-value from a standardized test statistic related to the difference in means across two groups.

**2. US Republican presidents dataset**

00:10 - 00:30

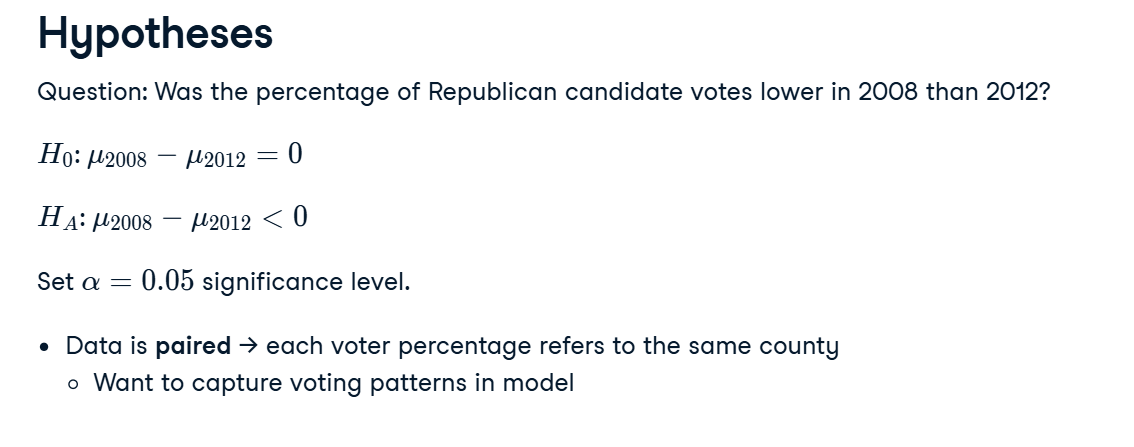
Here's a dataset of US presidential elections. Each row represents a presidential election at the county level. The variables in the dataset are the US state, the county within that state, and the percentage of votes for the Republican candidate in 2008, and in 2012.

1. 1 https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/VOQCHQ

**3. Hypotheses**

00:30 - 01:22

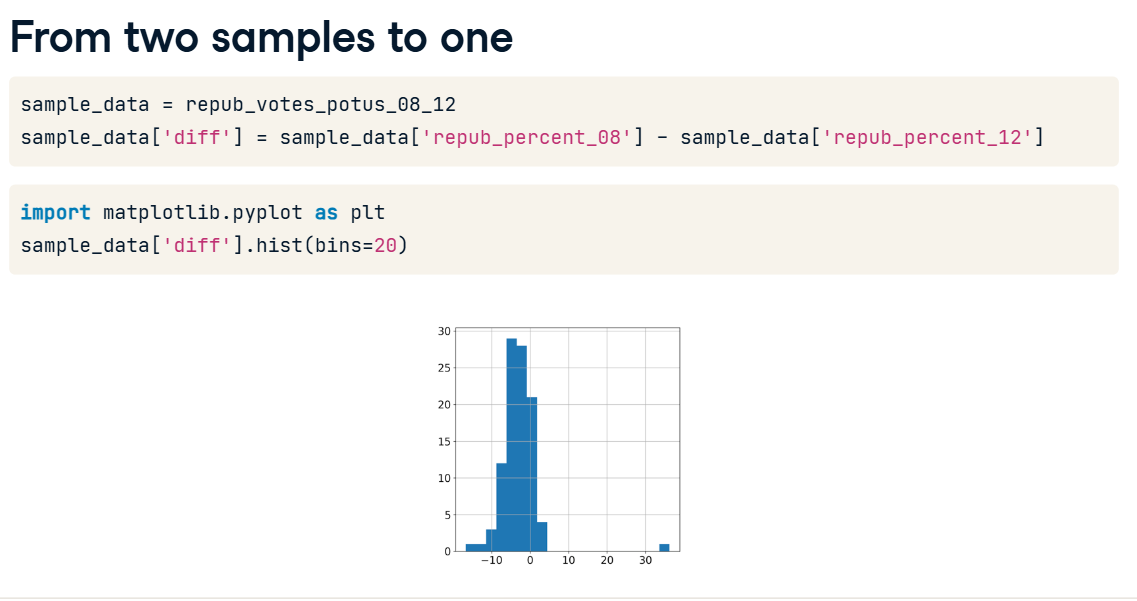
One question is whether the percentage of votes for the Republican candidate was lower in 2008 compared to 2012. To test this, we form hypotheses. As before, the null hypothesis is that our hunch is wrong, and that the population parameters are the same in each year group. The alternative hypothesis is that the parameter in 2008 was lower than in 2012. Let's set a significance level of point-zero-five. One feature of this dataset is that the 2008 votes and the 2012 votes are paired, which means they aren't independent, since they both refer to the same county. This means voting patterns may occur due to county-level demographics and local politics, and we want to capture this pairing in our model.



**4. From two samples to one**

01:22 - 01:44

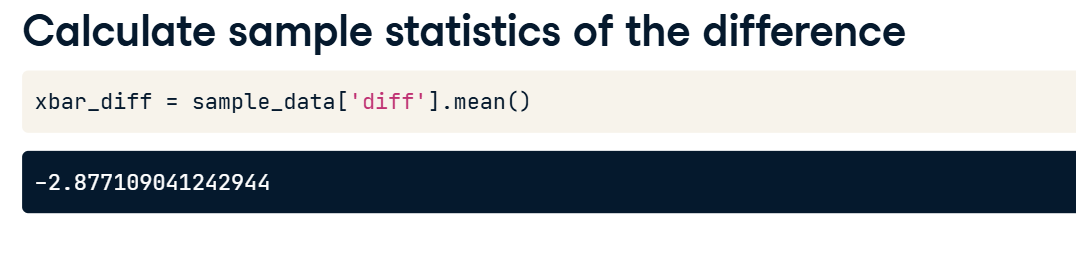
For paired analyses, rather than considering the two variables separately, we can consider a single variable of the difference. This is stored in a DataFrame called sample\_data with a column named diff. In this histogram of the difference, most values are between minus ten and ten, with at least one outlier.



**5. Calculate sample statistics of the difference**

01:44 - 01:52

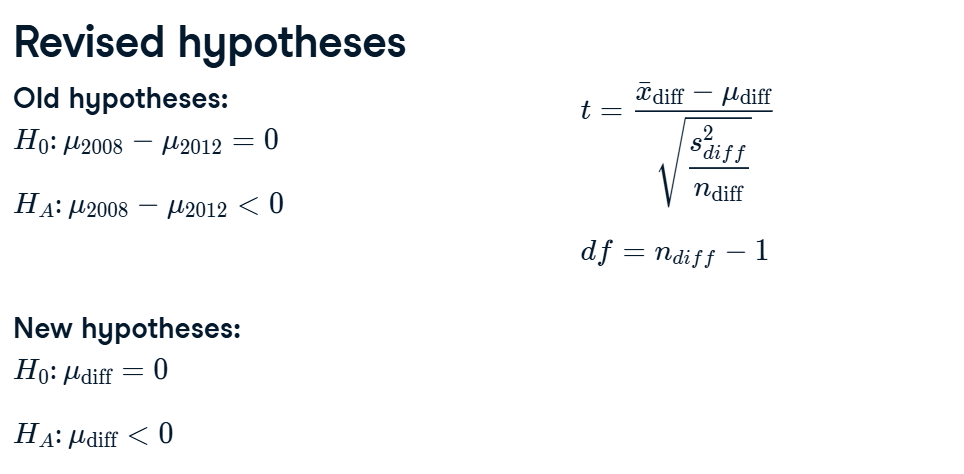
The sample mean, x-bar, is calculated from this difference. It is around minus two-point-eight-eight.



**6. Revised hypotheses**

01:52 - 02:14

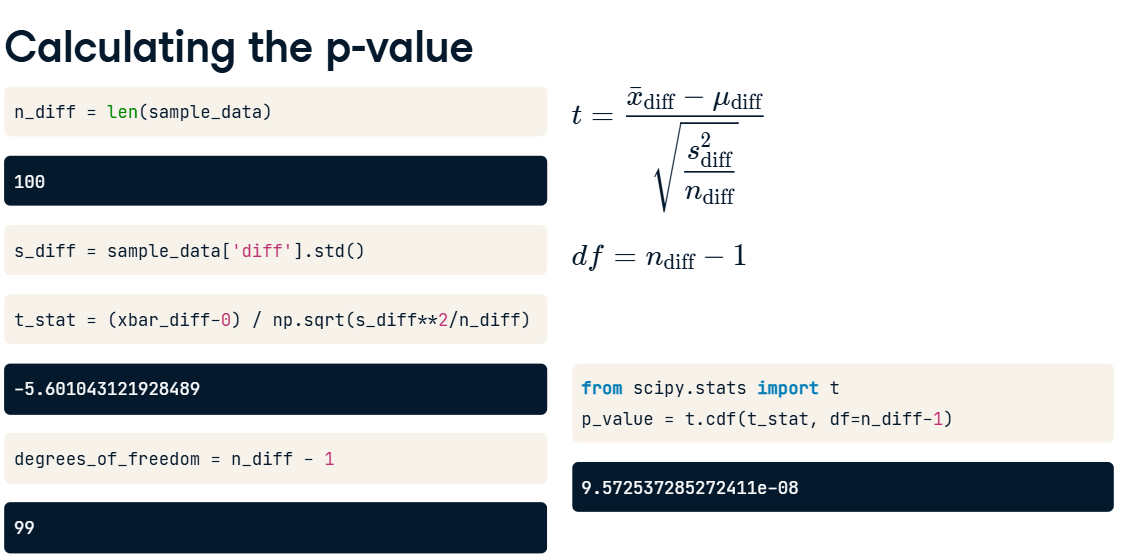
We can restate the hypotheses in terms of the single population mean, mu-diff, being equal to or less than zero. The test statistic, t, has a slightly simpler equation compared to the two sample case. We have one statistic, so the number of degrees of freedom is the number of pairs minus one.



**7. Calculating the p-value**

02:14 - 03:13

To calculate the test statistic, we need the number of rows in the dataset, one hundred, and the standard deviation of the differences. We already calculated x-bar-diff, the mean of the differences, as minus two-point-eight-eight. Assuming the null hypothesis is true means mu-diff is zero. We now have everything we need to plug into the equation to calculate t. It's minus five-point-six. The degrees of freedom are one less than n-diff at ninety nine. Finally, we transform t with the t-distribution CDF. The p-value is really small at around nine-point-six times ten to the minus eight. That means we reject the null hypothesis in favor of the alternative hypothesis that the Republican candidates got a smaller percentage of the vote in 2008 compared to 2012.

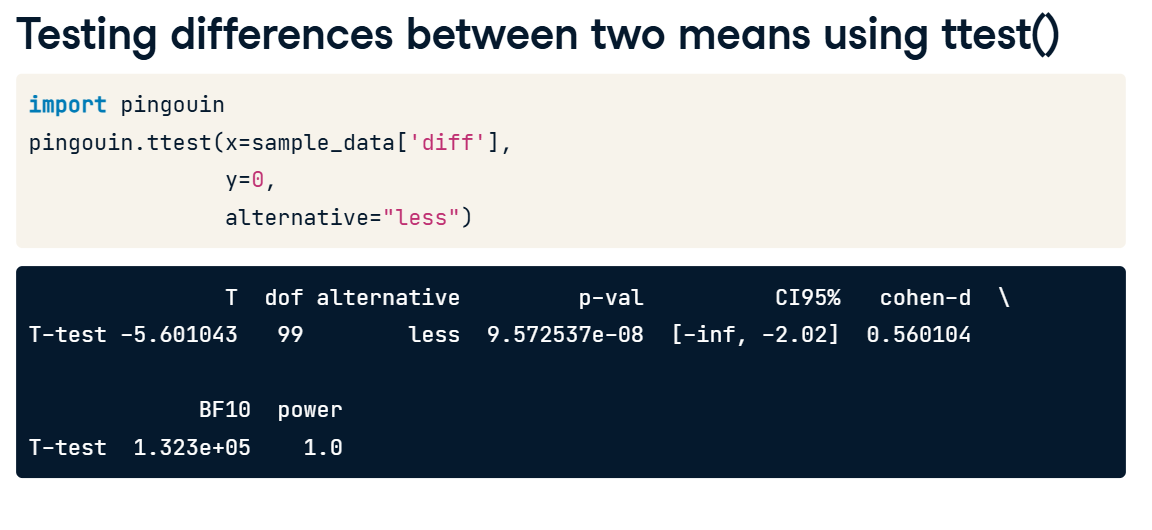


**8. Testing differences between two means using ttest()**

03:13 - 04:22

That was a lot of calculating. Fortunately, there's an easier way. The pingouin package provides a variety of different methods for hypothesis testing and returns the results as a pandas DataFrame. Its output can be a little friendlier to work with than similar methods from scipy-dot-stats. One method from pingouin is ttest and it works with array-like objects, so the first argument is the Series of differences. For a converted one sample test like this, y specifies the hypothesized difference value from the null hypothesis, which is zero. The type of alternative hypothesis can be specified as two-sided, less, or greater, corresponding to two-tailed, left-tailed, and right-tailed tests, respectively. Here's the output. We can recognize the value of the test statistic, the degrees of freedom, the alternative direction, and the p-value. The additional output refers to more advanced statistical concepts that are outside the scope of this course.

1. 1 Details on Returns from pingouin.ttest() are available in the API docs for pingouin at <https://pingouin-stats.org/generated/pingouin.ttest.html#pingouin.ttest>.



**9. ttest() with paired=True**

04:22 - 04:44

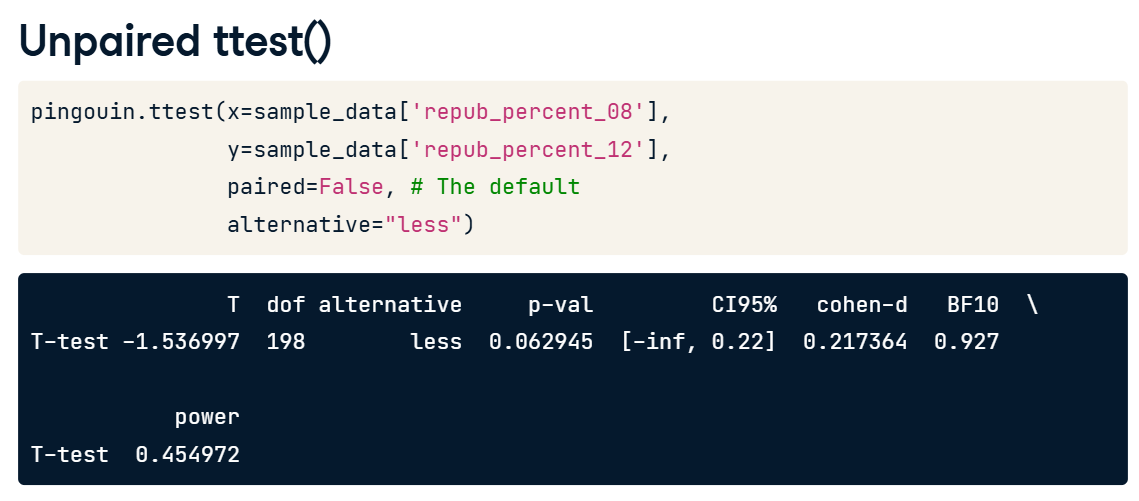
There's a variation of ttest for paired data that requires even less work. Rather than calculating the difference between the two paired variables, we can just pass them both directly to ttest as x and y, and set paired to True. Notice that the results in the first four columns are the same as before.



**10. Unpaired ttest()**

04:44 - 05:07

If we don't set paired to True and instead perform an unpaired t-test, then the numbers change. The test statistic is closer to zero, there are more degrees of freedom, and the p-value is much larger. Performing an unpaired t-test when our data is paired increases the chances of false negative errors.



**1. ANOVA tests**

00:00 - 00:08

We've seen how to compare two groups in the unpaired and paired cases. But what if there are more than two groups?

**2. Job satisfaction: 5 categories**

00:08 - 00:18

The Stack Overflow survey includes a job satisfaction variable, with five categories from "Very satisfied" down to "Very dissatisfied".

**3. Visualizing multiple distributions**

00:18 - 00:51

Suppose we want to know if mean annual compensation is different for each of the levels of job satisfaction. The first thing to do is visualize the distributions with box plots. Seaborn's boxplot method provides a nice option here with converted\_comp on the horizontal axis and job\_sat on the vertical axis using the stack\_overflow data. "Very satisfied" looks slightly higher than the others, but to see if they are significantly different, we'll need to use hypothesis tests.

**4. Analysis of variance (ANOVA)**

00:51 - 01:47

ANOVA tests determine whether there are differences between the groups. We begin by setting our significance level to point-two. This value is larger than in many situations but will help us understand the implications on comparing different numbers of groups later on. We use the pingouin anova method to compare values across multiple groups. We specify the data as stack\_overflow, the dependent variable, dv, as converted\_comp, and the column of groups to calculate between as job\_sat. The p-value is stored in the p-unc column, which is point-zero-zero-one-three, which is smaller than alpha at 20 percent. That means that at least two of the categories of job satisfaction have significant differences between their compensation levels, but this doesn't tell us which two categories they are.

**5. Pairwise tests**

01:47 - 02:07

To identify which categories are different, we compare all five job satisfaction categories, testing on each pair in turn. There are ten ways of choosing two items from a set of five, so we have ten tests to perform. Our significance level is still point-two.

**6. pairwise\_tests()**

02:07 - 02:39

To run all these hypothesis tests in one go, we can use pairwise\_tests. The first three arguments of data, dv, and between are the same as the anova method. We'll discuss p-adjust shortly. The result shows a DataFrame where A and B are the two levels being compared on each row. Next, we look at the p-unc column of p-values. Three of these are less than our significance level of point-two.

**7. As the number of groups increases...**

02:39 - 03:23

In this case we have five groups, resulting in ten pairs. As the number of groups increases, the number of pairs - and hence the number of hypothesis tests we must perform - increases quadratically. The more tests we run, the higher the chance that at least one of them will give a false positive significant result. With a significance level of point-two, if we run one test, the chance of a false positive result is point-two. With five groups and ten tests, the probability of at least one false positive is around point-seven. With twenty groups, it's almost guaranteed that we'll get at least one false positive.

**8. Bonferroni correction**

03:23 - 03:48

The solution to this is to apply an adjustment to increase the p-values, reducing the chance of getting a false positive. One common adjustment is the Bonferroni correction. Looking at the p-corr column corresponding to corrected p-values, as opposed to the p-unc column for uncorrected, only two of the pairs appear to have significant differences.

**9. More methods**

03:48 - 04:06

pingouin provides several options for adjusting the p-values with some being more conservative than others. No adjustment with none is the default, but in almost all pairwise t-testing situations choosing a correction method is more appropriate.