**One-sample proportion tests**

The hypothesis tests in Chapter 1 measured whether or not an unknown population proportion was equal to some value. We used bootstrapping on the sample to estimate the standard error of the sample statistic. The standard error was then used to calculate a standardized test statistic, the z-score, which was used to get a p-value, so we could decide whether or not to reject the null hypothesis. A bootstrap distribution can be computationally intensive to calculate, so this time we'll instead calculate the test statistic without it.

**Standardized test statistic for proportions**

An unknown population parameter that is a proportion, or population proportion for short, is denoted p. The sample proportion is denoted p-hat, and the hypothesized value for the population proportion is denoted p-zero. As in Chapter 1, the standardized test statistic is a z-score. We calculate it by starting with the sample statistic, subtracting its mean, then dividing by its standard error. p-hat minus the mean of p-hat, divided by the standard error of p-hat. Recall from Sampling in Python that the mean of a sampling distribution of sample means, denoted by p-hat, is p, the population proportion. Under the null hypothesis, the unknown proportion p is assumed to be the hypothesized population proportion p-zero. The z-score is now p-hat minus p-zero, divided by the standard error of p-hat.

**Simplifying the standard error calculations**

For proportions, under H-naught, the standard error of p-hat equation can be simplified to p-zero times one minus p-zero, divided by the number of observations, then square-rooted. We can substitute this into our equation for the z-score. This is easier to calculate because it only uses p-hat and n, which we get from the sample, and p-zero, which we chose.

**Why z instead of t?**

We might wonder why we used a z-distribution here, but a t-distribution in Chapter 2. This is the test statistic equation for the two sample mean case. The standard deviation of the sample, s, is calculated from the sample mean, x-bar. That means that x-bar is used in the numerator to estimate the population mean, and in the denominator to estimate the population standard deviation. This dual usage increases the uncertainty in our estimate of the population parameter. Since t-distributions are effectively a normal distribution with fatter tails, we can use them to account for this extra uncertainty. In effect, the t-distribution provides extra caution against mistakenly rejecting the null hypothesis. For proportions, we only use p-hat in the numerator, thus avoiding the problem with uncertainty, and a z-distribution is fine.

**Stack Overflow age categories**

Returning to the Stack Overflow survey, let's hypothesize that half of the users in the population are under thirty and check for a difference. Let's set a significance level of point-zero-one. In the sample, just over half the users are under thirty.

**Variables for z**

Let's get the numbers needed for the z-score. p-hat is the proportion of sample rows where age\_cat equals under thirty. p-zero is point-five according to the null hypothesis. n is the number of rows in the dataset.

**Calculating the z-score**

Inserting the values we calculated into the z-score equation yields a z-score of around three-point-four.

**Calculating the p-value**

For left-tailed alternative hypotheses, we transform the z-score into a p-value using norm-dot-cdf. For right-tailed alternative hypotheses, we subtract the norm-dot-cdf result from one. For two-tailed alternative hypotheses, we check whether the test statistic lies in either tail, so the p-value is the sum of these two values: one corresponding to the z-score and the other to its negative on the other side of the distribution. Since the normal distribution PDF is symmetric, this simplifies to twice the right-tailed p-value since the z-score is positive. Here, the p-value is less than the significance level of point-zero-one, so we reject the null hypothesis, concluding that the proportion of users under thirty is not equal to point-five.

**Two-sample proportion tests**

Great work so far! In the previous lesson, we tested a single proportion against a specific value. As with means, we can also test for differences between proportions in two populations.

**2. Comparing two proportions**

The Stack Overflow survey contains a hobbyist variable. The value "Yes" means the user described themselves as a hobbyist and "No" means they described themselves as a professional. We can hypothesize that the proportion of hobbyist users is the same for the under thirty age category as the thirty or over category, which is a two-tailed test. More formally, the null hypothesis is that the difference between the population parameters for each group is zero. Let's set a significance level of point-zero-five.

**3. Calculating the z-score**

Here is the z-score equation for a proportion test. Let's break it down. The sample statistic is the difference in the proportions for each category. That's the two p-hat values in the numerator. We subtract the hypothesized value of the population parameter, and assuming the null hypothesis is true, it's zero. The denominator is the standard error of the sample statistic. We can again avoid having to generate a bootstrap distribution to calculate the standard error by using a standard error equation, which is a slightly more complicated version of the one sample case. Note that p-hat is a weighted mean of the sample proportions for each category, also is known as a pooled estimate of the population proportion. p-hat can be calculated using the following equation. This looks horrendous, but Python is great at handling arithmetic. We now only need four numbers from the sample dataset to perform these calculations and calculate the z-score: the proportion of hobbyists in each age group, and the number of observations in each age group.

**4. Getting the numbers for the z-score**

To calculate these four numbers, we group by the age category, and calculate the sample proportions using dot-value\_counts, and the row counts using dot-count. As we're looking at the proportion of hobbyists, we'll only be focusing on rows where hobbyist is Yes.

**5. Getting the numbers for the z-score**

To isolate the hobbyist proportions from p\_hats, we can use pandas' multiIndex subsetting, passing a tuple of the outer column and inner column values. This returns a sample proportion of point-77 for the at least thirty group, and point-84 for the under thirty's.

The number of observations in each age category can be extracted with simpler pandas subsetting. There are 1050 rows in the at least thirty group and 1211 for the under 30 group.

After that, we can do the arithmetic using our equations for p\_hat, the standard error, and the z-score to get the test statistic. This returns a z-score of minus four-point-two-two. Luckily, we can avoid much of this arithmetic.

**Proportion tests using proportions\_ztest()**

The proportions\_ztest function from statsmodels can calculate the z-score more directly. This function requires two objects as NumPy arrays: the number of hobbyists in each age group, and the total number of rows in each age group. We can get these numbers by grouping by age\_cat, and calling dot-value\_counts on the hobbyist column, as shown above. The numbers can then either be read-off or subsetted to create the arrays. Next, we import proportions\_ztest from statsmodels-dot-stats-dot-proportions, and pass the arrays to the count and nobs arguments. Because we're testing for a difference, we specify that this is a two-sided test using the alternative argument. proportions\_ztest returns a z-score and a p-value. The p-value is smaller than the five percent significance level we specified, so we can conclude that there is a difference in the proportion of hobbyists between the two age groups.

**Chi-square test of independence**

Just as ANOVA extends t-tests to more than two groups, chi-square tests of independence extend proportion tests to more than two groups.

**2. Revisiting the proportion test**

Here's the proportions test from the last video. The test statistic is the z-score of minus four-point-two-two.

**3. Independence of variables**

That proportion test had a positive result. The small p-value suggested that there was evidence that the hobbyist and age category variables had an association. If the proportion of hobbyists was the same for each age category, the variables would be considered statistically independent. More formally, two categorical variables are consider statistically independent when the proportion of successes in the response variable is the same across all categories of the explanatory variable.

**4. Test for independence of variables**

The pingouin package has an indirect way of testing the difference in the proportions from the previous video. To the chi2\_independence method, we pass stack\_overflow as data, hobbyist as x, and age\_cat as y. The correction argument specifies whether or not to apply Yates' continuity correction, which is a fudge factor for when the sample size is very small and the degrees of freedom is one. Since each group has over one hundred observations, we don't need it here. The method returns three different pandas DataFrames: the expected counts, the observed counts, and statistics related to the test. Let's look at stats and focus on the pearson test row and the chi2 and pval columns. The p-value is the same as we had with the z-test of around two in one hundred thousand. The chi2 value is the squared result of our z-score seen in the previous video.

**5. Job satisfaction and age category**

Let's try another example. Recall that the Stack Overflow sample has an age category variable with two categories and a job satisfaction variable with five categories.

**6. Declaring the hypotheses**

We can declare hypotheses to test for independence of these variables. Here, age category is the response variable, and job satisfaction is the explanatory variable. The null hypothesis is that independence occurs. Let's use a significance level of point-one. The test statistic is denoted chi-square. It quantifies how far away the observed results are from the expected values if independence was true.

**7. Exploratory visualization: proportional stacked bar plot**

Let's explore the data using a proportional stacked bar plot. We begin by calculating the proportions in each age group. Next, we use the unstack method to convert this table into wide format. Using the plot method and setting kind to bar and stacked to True produces a proportional stacked bar plot.

**8. Exploratory visualization: proportional stacked bar plot**

If the age category was independent of job satisfaction, the split between the age categories would be at the same height in each of the five bars. There's some variation here, but we'll need a chi-square independence test to determine whether it's a significant difference.

**9. Chi-square independence test**

Let's again use the chi-square independence test from pingouin. We have stack\_overflow as the data and job\_sat and age\_cat as x and y. We leave out a correction here since our degrees of freedom is four, calculated by subtracting one from each of the variable categories and multiplying. The p-value is point-two-three, which is above the significance level we set, so we conclude that age categories are independent of job satisfaction.

**10. Swapping the variables?**

Swapping the variables, so age category is the response and job satisfaction is the explanatory variable,

**11. Swapping the variables?**

we see that the splits for each bar are in similar places.

**12. chi-square both ways**

If we run the chi-square test with the variables swapped, then the results are identical. Because of this, we phrase our questions as "are variables X and Y independent?", rather than "is variable X independent from variable Y?", since the order doesn't matter.

**13. What about direction and tails?**

We didn't worry about tails in this test, and in fact, the chi2\_independence method doesn't have an alternative argument. This is because the chi-square test statistic is based on the square of observed and expected counts, and square numbers are non-negative. That means that chi-square tests tend to be right-tailed tests.

1. 1 Left-tailed chi-square tests are used in statistical forensics to detect if a fit is suspiciously good because the data was fabricated. Chi-square tests of variance can be two-tailed. These are niche uses, though.

**Chi-square goodness of fit tests**

Last time, we used a chi-square test to compare proportions in two categorical variables. This time, we'll use another variant of the chi-square test to compare a single categorical variable to a hypothesized distribution.

**2. Purple links**

The Stack Overflow survey contains a fun question about how users feel when they discover that they already visited the top resource, also called a purple link, when trying to solve a coding problem. We can use the dot-value-counts method to get the counts of each group in the purple\_link column. We also do a little bit of manipulation here to get a nicely structured DataFrame that we can work with later. First, we rename the leftmost column to be purple\_link, assign the counts to n, and finally sort by purple\_link, so the responses are in alphabetical order. There are four possible answers stored in the purple\_link column.

**3. Declaring the hypotheses**

Let's hypothesize that half of the users in the population would respond "Hello, old friend", and the other three responses would get one sixth each. We can create a DataFrame for these hypothesized results from a dictionary of key-value pairs for each response. We specify the hypotheses as whether or not the sample matches this hypothesized distribution. The test statistic, chi-squared, measures how far the observed sample distribution of proportions is from the hypothesized distribution. Let's set the significance level of point-zero-one.

**4. Hypothesized counts by category**

To visualize the purple\_link distribution, it will help to have the hypothesized counts for each answer, which are calculated by multiplying the hypothesized proportions by the total number of observations in the sample.

**5. Visualizing counts**

Let's create a visualization to see how well the hypothesized counts appear to model the observed counts. The natural way to visualize the counts of a categorical variable is with a bar plot. First, we use plt-dot-bar to plot the observed purple\_link counts, setting the horizontal axis to purple\_link and the vertical axis to n. We set the color of the bars and add a label for a legend. We do the same again for the hypothesized counts, but also add transparency with the alpha argument.

**6. Visualizing counts**

We can see that two of the responses are reasonably well-modeled by the hypothesized distribution and another two appear quite different, but we'll need to run a hypothesis test to see if the difference is statistically significant.

**7. chi-square goodness of fit test**

The one-sample chi-square test is called a goodness of fit test, as we're testing how well our hypothesized data fits the observed data. To run the test, we use the chisquare method from scipy-dot-stats. There are two required arguments to chisquare: an array-like object for the observed counts, f\_obs, and one for the expected counts, f\_exp. The p-value returned by the function is very small, much lower than the significance level of point-zero-one, so we conclude that the sample distribution of proportions is different from the hypothesized distribution.