**Assumptions in hypothesis testing**

Each hypothesis test we've seen so far makes assumptions about the data. It's only when these assumptions are met that it is appropriate to use that hypothesis test.

**Randomness**

Whether it uses one or multiple samples, every hypothesis test assumes that each sample is randomly sourced from its population. If we don't have a random sample, then it won't be representative of the population. To check this assumption, we need to know where our data came from. There are no statistical or coding tests we can perform to check this. If in doubt, ask the people involved in data collection, or a domain expert that understands the population being sampled.

1. 1 Sampling techniques are discussed in "Sampling in Python".

**Independence of observations**

Tests also assume that each observation is independent. There are some special cases like paired t-tests where dependencies between two samples are allowed, but these change the calculations, so we need to understand where such dependencies occur. As we saw with the paired t-test, not accounting for dependencies results in an increased chance of false negative and false positive errors. Not accounting for dependencies is a difficult problem to diagnose during analysis. Ideally, it needs to be discussed before data collection.

**Large sample size**

Hypothesis tests also assume that our sample is large enough that the Central Limit Theorem applies, and the sample distribution can be assumed to be normally distributed. Smaller samples incur greater uncertainty, which may mean that the Central Limit Theorem does not apply and the sampling distribution might not be normally distributed. The increased uncertainty of a small sample means we get wider confidence intervals on the parameter we are trying to estimate. If the Central Limit Theorem does not apply, the calculations on the sample, and any conclusions drawn from them, could be nonsense, which increases the chance of false negative and false positive errors. How big our sample needs to be to be "big enough" depends on the test.

**Large sample size: t-test**

For one sample t-tests, a popular heuristic is that we need at least thirty observations in our sample. For the two sample case or ANOVA, we need thirty observations from each group. That means we can't compensate for one minority group sample by making the majority group bigger. In the paired case, we need thirty pairs of observations. Sometimes we can get away with less than 30 in each of these tests; the important thing is that the null distribution appears normal. This is often the case at around 30 and that's the reason for this somewhat arbitrary threshold.

**Large sample size: proportion tests**

For one sample proportion tests, the sample is considered big enough if it contains at least ten successes and ten failures. Notice that if the probability of success is close to zero or close to one, then we need a bigger sample. In the two sample case, we require ten successes and ten failures from each sample.

**Large sample size: chi-square tests**

The chi-square test is slightly more forgiving and only requires five successes and five failures in each group, rather than ten.

**Sanity check**

One more check we can perform is to calculate a bootstrap distribution and visualize it with a histogram. If we don't see a bell-shaped normal curve, then one of the assumptions hasn't been met. In that case, we should revisit the data collection process, and see if any of the three assumptions of randomness, independence, and sample size do not hold.

**Non-parametric tests**

So what do we do if the assumptions for the hypothesis tests we've seen so far aren't met?

**Parametric tests**

The tests that we've seen so far are known as parametric tests. Tests like the z-test, t-test, and ANOVA are all based on the assumption that the population is normally distributed. Parametric tests also require sample sizes that are "big enough" that the Central Limit Theorem applies.

**Smaller Republican votes data**

Let's study a case where the sample size requirement isn't met with a subset of the US Presidential voting results for Republican candidates that we examined in a previous chapter. Here, repub\_votes\_small contains only five counties randomly sampled from the larger dataset of 2008 and 2012 county-level returns.

**Results with pingouin.ttest()**

Let's try performing a paired t-test on this small sample. Recall that we require 30 pairs to feel confident in using a t-test, and this sample only contains five. We set a significance level of one percent and use the ttest method from pingouin to perform the left-tailed paired t-test. The small p-value indicates we should reject the null hypothesis, leading us to suspect that the 2008 election had a smaller percentage of Republican votes than the 2012 election.

**Non-parametric tests**

In situations where we aren't sure about these assumptions, or we are certain that the assumptions aren't met, we can use non-parametric tests. They do not make the normal distribution assumptions or the sample size conditions that we saw in the previous video. There are many different ways to perform tests without these parametric assumptions. In this chapter, we'll focus on those relating to ranks. Consider the list, x. The first value of x, one, is the smallest value and the second value, fifteen, is the fifth smallest. These orderings from smallest to largest are known as the ranks of the elements of x. We can access them with the rankdata method from scipy-dot-stats.

Let's now use a non-parametric test to see what kind of results it gives. Remember that non-parametric tests work better than the parametric alternative in situations where the sample size is small or the data cannot be assumed to be normally distributed.

We will use the Wilcoxon-signed rank test, which was developed by Frank Wilcoxon in 1945 and was one of the first non-parametric procedures developed. We'll go over the inner workings of the test before implementing it using another pingouin method.

**Wilcoxon-signed rank test (Step 1)**

The Wilcoxon-signed rank test requires us to calculate the absolute differences in the pairs of data and then rank them. First, we take the differences in the paired values.

**Wilcoxon-signed rank test (Step 2)**

Next, we take the absolute value of the differences, using the dot-abs method, and place them in the abs\_diff column.

**Wilcoxon-signed rank test (Step 3)**

Then, we rank these absolute differences using the rankdata method from scipy-dot-stats.

**Wilcoxon-signed rank test (Step 4)**

The last part of our calculation involves calculating a test statistic called W. W uses the signs of the diff column to split the ranks into two groups: one for rows with negative differences and one for positive differences. T-minus is defined as the sum of the ranks with negative differences, and T-plus is the sum of the ranks with positive differences. For this example, all the differences are negative, so the T-minus value is the sum of the five ranks, and T-plus is zero. The test statistic W is the smaller of T-minus and T-plus, which in this case, is zero. We can calculate W, and its corresponding p-value, using a pingouin method instead of manual calculation.

**Implementation with pingouin.wilcoxon()**

The dot-wilcoxon method from pingouin takes very similar arguments to the dot-ttest method, except it doesn't have a paired argument. The function returns a W value of zero - the same as our manual calculation! This corresponds to a p-value of around three percent, which is over ten times larger than the p-value from the t-test, so we should feel more confident with this result given the small sample size. The Wilcoxon test indicates that we do not have evidence that the 2008 Republican percentages are smaller than the 2012 percentages using this small sample of five rows.

**Non-parametric ANOVA and unpaired t-tests**

In the previous video, we explored some non-parametric techniques and how they compare to their parametric counterparts. We'll continue on that theme here focusing on non-parametric alternatives to tests of independent numeric samples.

**Wilcoxon-Mann-Whitney test**

We can avoid assumptions about normally distributed data by performing hypothesis tests on the ranks of a numeric input. The Wilcoxon-Mann-Whitney test is, very roughly speaking, a t-test on ranked data. This test is similar to the Wilcoxon test we saw in the last video, but works on unpaired data instead.

**Wilcoxon-Mann-Whitney test setup**

Let's return to the StackOverflow survey and the relationship between converted compensation and the age respondents began coding. We start by focusing on just those two columns in a new DataFrame called age\_vs\_comp. To conduct a Wilcoxon-Mann-Whitney test with pingouin, we first need to convert our data from long to wide format. This is accomplished with the pivot method from pandas, which unlike pivot\_table, does not aggregate; instead, it returns the raw values for each group across the rows. We now have our data in two columns named adult and child with the values corresponding to the converted\_comp entries for each row. An adult value of NaN corresponds to a child entry and a child value of NaN corresponds to an adult entry.

**Wilcoxon-Mann-Whitney test**

Let's set a significance level of one percent. We can run a Wilcoxon-Mann-Whitney test using mwu from pingouin. It accepts x and y arguments corresponding to the two columns of numbers we want to compare, in this case, child and adult. alternative sets the type of alternative hypothesis, in this case, that those who code first as children have a higher income than those who code first as adults, which is a right-tailed test. Here, the p-value is shown as around ten to the negative nineteenth power, which is significantly smaller than the significance level.

**Kruskal-Wallis test**

In the same way that ANOVA extends t-tests to more than two groups, the Kruskal-Wallis test extends the Wilcoxon-Mann-Whitney test to more than two groups. That is, the Kruskal-Wallis test is a non-parametric version of ANOVA. We use the kruskal method from pingouin to perform a Kruskal-Wallis test to investigate if there is a difference in converted\_comp between job satisfaction groups. Unlike the Wilcoxon-Mann-Whitney test, we don't need to pivot our data here since the kruskal method works on long data. We pass in stack\_overflow as data, the dependent variable, dv, as converted\_comp, and we are comparing between the groups of job\_sat. Again, the p-value here is very small and smaller than our significance level. This provides evidence that at least one of the mean compensation totals is different than the others across these five job satisfaction groups.