Modern Statistics:

Intuition, Math, Python

Dr. Mike X Cohen

0.1 Front matter

This page contains some important details about the book that basically no one reads but somehow is always in the first page.

ISBN: ???????????????

This book was written and formatted in LaTeX by Mike X Cohen. (Mike X Cohen is the friendlier and more approachable persona of Professor Michael X Cohen; Michael deals with the legal and business aspects while Mike gets to have fun writing.)

Book edition 1.

0.2 Dedication

If you're reading this, then the book is dedicated to you. I wrote this book for *you*. Now turn the page and start learning statistics!

0.3 Forward

The past is immutable and the present is fleeting. Forward is the only direction.

Contents

Chapter 11

The t-test family

11.1 Purpose and interpretation of the t-test

The t-test is one of the most important and most commonly used inferential statistics. There are several specific implementations of the t-test that depend on the nature of the data (number of groups, sample sizes, variances, and so on), but they all share a common framework.

In this section, you will learn that fundamental framework, how to interpret the t-test, how to derive a p-value from a t-value, and the assumptions that underlie t-tests. In later sections I will introduce the specific variants of t-tests for different data situations (one-sample vs. two-sample; equal vs. unequal variance, and so on), and non-parametric alternatives to use when your data violate assumptions of the t-test.

I titled this chapter "The t-test *family*," because you will see that there are different variants of the t-test that are used in different scenarios and for different assumptions about the data, but they are all conceptually and analytically similar to each other. Understanding the foundational principles that underlie all t-tests will allow you to use and interpret the t-test more effectively. There are also members of the extended t-test family that I will present later in the book, for example the t-test for statistical significance of correlation coefficients.

11.1.1 The purpose of a t-test

The purpose of a t-test is to determine whether the mean of a sample is different from a specified H_0 value. There are three scenarios where you would use a t-test, described below and visualized in Figure 11.2. Although these may seem like *distinct* situations, the t-tests used to evaluate these situations are conceptually and mathematically similar.

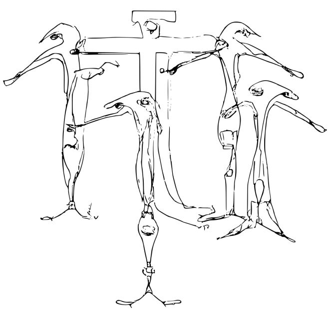

Figure 11.1: The t-test family from outer space.

One-sample t-test (Figure 11.2A)

In this scenario, you have one data sample, and the objective is to determine if the sample mean significantly deviates from a predetermined H_0 value (each circle in panel A represents an individual data point, and the horizontal dashed line represents the H_0 value). For example, perhaps the dashed line is IQ=100 and the data values are IQs of children in a particular classroom. Clearly, not *all* data samples are above the H_0 value, but it is possible that the average is significantly above that H_0 line.

Paired-samples t-test (Figure 11.2B)

In this scenario, you have one group of individuals that were measured twice. For example, imagine a study in which a research team recorded sales volume from 30 companies before ("pre") and after ("post") a corporate team-building retreat. Some companies experienced an increase in sales volume while others experienced a decrease. The question is whether sales volume increased *on average* across the sample of 30 companies.

Independent samples t-test (Figure 11.2C)

In this scenario, you have two separate groups of individuals, and want to determine whether the means of the two groups differ. Because these are different samples, the sample sizes and variances might differ, although we are only interested in testing for differences of the means. An example is a study comparing exam scores of students who attended a specific study session (group "1") to those who did not attend the session (group "2"). The goal is to determine whether attending the study session had a significant impact on the students' exam scores. The two lines in panel C depict histograms of exam scores from the two groups.

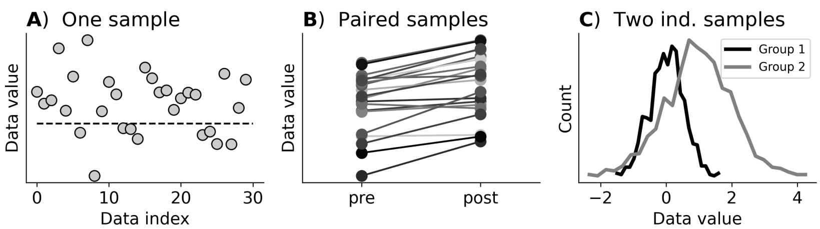

Figure 11.2: Visualizing the three t-test scenarios.

11.1.2 General t-test formula

Equation 11.1 shows a general formula for a t-test. Later in this chapter you will learn about modifications to this formula for the specific cases I highlighted above, but this formula is a "template" for all members of the t-test family, and I encourage you to commit this formula to memory.

$$t_{df} = \frac{\overline{x} - h_0}{s/\sqrt{n}} \tag{11.1}$$

where $_{df}$ is the degrees of freedom, h_0 is the null-hypothesis value, s is the sample standard deviation, and n is the sample size. The denominator is the standard error of the mean, which you learned about in section **??**.

A few remarks about the t-test:

1. A t-test against an *a priori* chosen value is a *one-sample* t-test. A typical h_0 value is zero, that is, a one-sample t-test is often used to determine whether the mean of a dataset is different from zero.

2. A t-test between two groups is called a *two-samples* t-test. Two-samples t-tests can be paired or unpaired, with equal or distinct standard deviations and sample sizes. These lead to modifications of the t-test formula, and I will discuss them later in this chapter.

3. The t-test is based on means and standard deviations. A t-test can be statistically significant even if some data points show opposite effects from the group mean. Imagine, for example, a significant t-test comparing the heights of men to women. Generally, adult men are taller than adult women, but not every man is taller than every woman.

4. The way to conceptualize the t-test equation is "normalized difference of means." In other words, the average effect scaled by the variability. This conceptualization links the t-test to the general concept of a test statistic as a signal-to-noise ratio. It is also conceptually similar to z-score normalization.

5. The t-value is influenced by the sample size. In particular, increasing the sample size will increase the t-statistic, even if the mean and standard deviation do not change. This means that the distribution of H_0 t-values depends, in part, on the sample size (indeed, imagine putting the $\sqrt{n}$ factor in the numerator: The t-value will increase with sample size even if the mean and standard deviation remain the same.). That's why we need to know the degrees of freedom to associate a t-value with a p-value.

6. The sign of the t-test is arbitrary. You can write $(\overline{x} - h_0)$ or $(h_0 - \overline{x})$. The magnitude of the t-value — and its associated p-value — will be the same. You can choose the sign to facilitate interpretation. For example, if you are testing for an increase in exam scores after reading a textbook, it makes sense to have a positive t-value. On the other hand, if you are testing for decreases in post-operative pain with a new surgical technique, then a negative t-value is more interpretable.

11.1.3 Degrees of freedom of t-tests

Remember that *df* is the maximum number of data values that can independently vary in a sample dataset. Because the t-test involves testing sample means, the *df* associated with a t-value will be the number of data values that can vary given that we know the sample means (generally, the number of descriptive statistics minus one).

- **One-sample t-test**: Because there is one sample with one mean, the *df* is $N - 1$ (where N is the sample size).
- **Paired-samples t-test**: You will learn later in this chapter that the paired samples t-test is actually the one-sample t-test in disguise (spoiler alert: subtract the two measurements to get one data value per individual), which means that the *df* is again $N - 1$.
- **Independent-samples t-test**: The *df* of a two-samples t-test is $N_1 + N_2 - 2$, where N_1 and N_2 are the sample sizes of the first and second group. Why minus 2? There are two samples and two means, and so the total *df* is actually $(N_1 - 1) + (N_2 - 1)$, which is then simplified. In some cases, the *df* calculation is more complicated to incorporate differences in sample variances. More on this point later.

In terms of typographical formatting, t-values are reported with the *df* in either subscript or parentheses. For example, a t-value of 2.56 with 13 degrees of freedom can be written as $t_{13} = 2.56$ or as $t(13) = 2.56$. Which format to use is sometimes a matter of personal preference, and sometimes dictated the style of the publisher.

11.1.4 P-values from t-values

If H_0 were true, then we expect $\overline{x} = h_0$ (alternatively: $\overline{x} - h_0 = 0$), which would make the t-value zero. Of course, with sampling variability and noise, we cannot expect t-values to equal zero exactly. And if we were to collect lots and lots of samples, we would expect the H_0 t-values to have some distribution around zero.

You've already computed and visualized a family of t-distributions in Exercise **??** (if you haven't done that exercise yet, I recommend working through it before proceeding in this chapter; or at least flip back to page **??** to look at the distributions). As a quick reminder, Figure 11.3 shows a t-value pdf for one *df* parameter. The question we ask when performing a t-test is this: What is the probability that a t-value more extreme than our empirical t-value could have been observed if the null hypothesis were true?

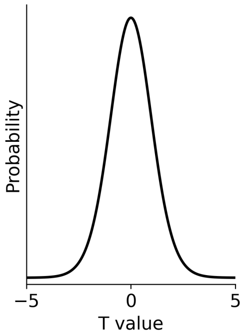

Figure 11.3: A distribution of H_0 t values with *df*=20.

In Chapter 10 and Figure 11.3, I got the p-values from a Python function. But where do they actually come from? The pdf of a t-distribution comes from the following equation (ν indicates the degrees of freedom):

$$p(t,\nu) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\sqrt{\nu\pi}\,\Gamma\left(\frac{\nu}{2}\right)}\left(1+\frac{t^2}{\nu}\right)^{-(\nu+1)/2} \tag{11.2}$$

$$\Gamma(z) = \int_0^\infty t^{z-1}e^{-t}\,dt \tag{11.3}$$

Please don't ask me to derive those equations from first principles; that's something you would learn in an advanced course on mathematical statistics. The point here is that p-values come from formulas, most of which are dense and complicated — and which provide basically zero insights or useful information about how to use or interpret p-values. Pdf's for other distributions you'll use in statistics can be found online, e.g., on wikipedia or on the `scipy.stats` library website. I won't say not to look them up, but I will warn you that staring at those equations is unlikely to make you a better statistics practitioner.

Instead, you can focus on the idea of determining statistical significance and deriving a p-value, which I introduced in Chapter 10: Compute the probability of observing a t-value as large or larger than the t-value observed in the real data.

Now let me come back to the practical interpretation of the question: How do you get p-values in Python? For this, you use functions available in the `scipy.stats` library. You need to know the t-value and the *df*, both of which you compute from your data.

The p-value comes from evaluating the cdf at your observed t-value, using the following code:

```
tval,df = 2.1,13
pval = 1-stats.t.cdf(tval,df)
```

The p-value for this t-value is 0.028 (using more professional-looking typographical formatting: $t_{13} = 2.1, p < .028$). I imagine you might have two questions about this code.

First, why do we use the cdf and not the pdf? The reason is that we're not interested in the probability of obtaining an H_0 t-value that *exactly* equals the t-value in our empirical data[1]. Instead, we're interested in the probability of obtaining a t-value that is as extreme *or more extreme* than the value we obtained in real data. For example, we don't want the probability of t=2.56 given that H_0 were true; we want the total probability of the H_0 t-value being anywhere between 2.56 and ∞ (and for the negative tail of the distribution: the total probability of the H_0 t-value being anywhere between -2.56 and $-\infty$).

Second, why use 1-cdf? Remember that the cdf is the cumulative sum of all probability values *less than* the specified value (that is, to the left in the distribution), so for a positive t-value, we want to know the probabilities *larger than* that value (to the right in the distribution). If the total area of the cdf is 1, then the area to the right of some value equals 1 minus the area to the left of that value.

In fact, this means that the line of code I wrote above is valid only for the one-tailed p-value on the right side of the distribution. A two-tailed test would require computing the areas of the probability distributions of both the left and the right sides:

```
pvalL = stats.t.cdf(-tval,df)  # area of left tail
pvalR = 1-stats.t.cdf(tval,df) # area of right tail
pval2 = pvalR+pvalL            # area of both tails
```

Because the t-distribution is symmetric, you don't actually need to compute the p-values for each tail separately. Instead, you can compute the area in one tail and double that value. I've demonstrated their equivalence below.

```
One-tailed p-value on the left:  0.027906302135628887
One-tailed p-value on the right: 0.027906302135628946

Two-tailed p-value as the sum:   0.05581260427125784
Two-tailed p-value by doubling:  0.055812604271257775
```

There seems to be some difference in the two one-tailed p-values, but that's just a minuscule rounding error.

There is a class of functions called *survival functions*, which are so named

[1] Technically speaking, the probability of getting the *exact* value is zero, but we can imagine the probability of an H_0 value close to our empirical value; that probability would be nonzero but tiny.

because they are used to determine the probability that a person or device survives past a certain date. Conceptually, the survival function is simply 1-cdf, but the details of their implementation allow for slightly more accurate probability estimates compared to the code I show above for `pvalR`. This is illustrated in the online code.

Later in this chapter, I will show you how to use Python functions to return the t- and p-values without having to use the `cdf` functions yourself. But you need to put in the effort to understand where these values come from before you can take the easy route.

11.1.5 T-values from p-values

Now you know how to get a p-value from a t-value and *df* parameter. It is conceptually easy to get a t-value from a specific p-value (and *df* parameter). Consider the cdf shown in Figure 11.4A; you've been reading this plot as going from the x-axis to the y-axis (that is, finding the y-axis value that corresponds to a specific t-value). Now read the plot the other way around: Pick a specific cdf value on the y-axis to discover the corresponding t-value on the x-axis. The plot in panel B shows the axes swapped to facilitate the comparison.

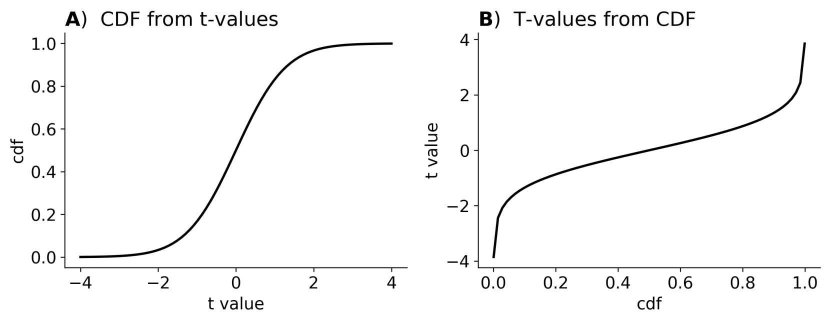

Figure 11.4: Obtaining t-values from p-values simply involves inverting the functions, which you can conceptualize as swapping the two axes.

The Python function to invert the cdf is `stats.t.isf`, which is the inverse survival function. Remember that the survival function itself is 1-cdf, so you need to flip the sign of the inverse survival function. Putting this together, you can use the following code to compute a t-value from a specific p-value.

```
pval = .05
tFromP = -stats.t.isf(pval,df)
```

But to make things more confusing, remember that ".05" here means the sum of all probabilities to the left of a particular value. Indeed, the value of `tFromP` is -1.771. So if you want to get a positive t-value, you need to enter 1-p as the first input. In other words, the following two lines of code will output the same positive t-value.

```
tFromP_R1 = -stats.t.isf(1-pval,df)
tFromP_R2 =  stats.t.isf(  pval,df)
```

I know, it's all quite confusing with the minus signs and the two tails. The good news is that you will rarely (if ever) need to compute t-values from p-values. The bad news is that the confusion of one- vs. two-tailed tests persists, as you will soon encounter...

11.1.6 Determining significance of a t-test

This subsection is a reminder of what you learned in Chapter 10: The statistical test associated with the t-value is considered statistically significant if a t-value of that magnitude, or more extreme, has less than a 5% chance of being observed if the null hypothesis were true.

Figure 11.5 illustrates the idea. Imagine we have a t-value of 1.6 with df=20. This t-value is not statistically significant, because the area of the H_0 t-pdf for t>1.6 is .0626, or 6.26%. The t-value would need to be larger than the critical value of 2.09 to be significant (because this is a two-tailed test, the t-value could also be significant if it were less than -2.09).

Here's a question for you to answer before reading the next paragraph: What is the two-tailed p-value associated with this t-value?

Based on the text and on inspection of the figure, you might have guessed that the p-value is p=.0626. **This is wrong**. It is an easy mistake to make, and reveals the trickiness of one- and two-tailed tests. In fact, the p-value of this t-value is p=.125. Consider that the area to the right of t=1.6 is 6.26% of the total t-value distribution, but in a two-tailed test, we are interested in the area that is more extreme than t=|1.6| — that is, greater than 1.6 *and* less than -1.6. Each tail contains 6.26% of the total area, so the area in both tails is 12.52%.

With that in mind, now inspect Figure 11.6, which shows both tails of the distribution. I hope now it's more clear: there is a 6.26% chance of finding a t value greater than 1.6 under H_0, and there is also a 6.26%

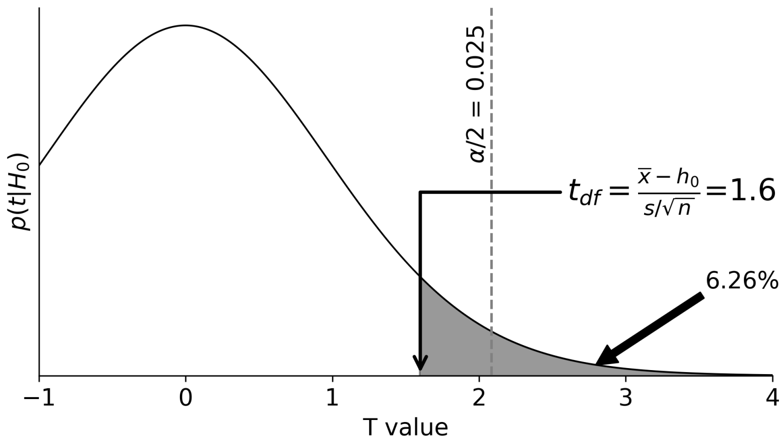

Figure 11.5: Visualizing the process of determining significance of a t-value. Imagine we have a t-value of $t_{20} = 1.6$, indicated by the downward arrow. Given our two-tailed α threshold of .05 (vertical dashed line shows the positive tail), this t-value would not be considered statistically significant, because there is a 6.26% chance of observing a t-value of 1.6 or larger if the null hypothesis were true.

chance of finding a t value less than −1.6 under H_0. Therefore, the p-value associated with the two-tailed test of t=1.6 is the sum of both areas, which is p=.1252.

(Quick reminder that having a p-value larger than .05 does not mean that we have evidence that the means are equal; instead, p>.05 indicates insufficient evidence that the means are different.)

11.1.7 Determining significance by critical t-values

The procedure described above is to (1) compute the t-statistic of the data, (2) compute the p-value associated with that t-value, and (3) label the finding as significant or not, based on the p-value. Steps 1 and 2 are done in Python.

There is another approach that bypasses step 2: compare your t-statistic to a "critical t-value." A critical t-value is the t-value corresponding to a certain *df* and α threshold. For example, the critical t-value associated with a 2-tailed p<.05 and *df*=20 is t=2.086. If your empirical t-value is larger than this, then your test is significant at p<.05. You don't need to compute the actual p-value.

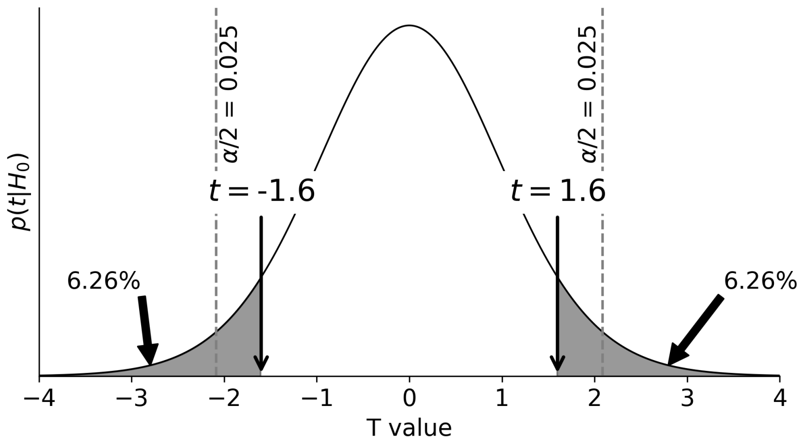

Figure 11.6: Same as Figure 11.5 but showing both tails of the distribution.

"Critical values" are old-fashioned. They are a relic of pre-computer days when p-values could not reasonably be computed by hand (*cf.* Equation 11.2!), and therefore statistics books came with really long tables that listed critical test statistic values for different degrees of freedom and α thresholds. Nowadays, we just have computers compute the p-value for the observed test statistic value. I considered printing a table of critical values here for your horror and enjoyment, but it would take up too much space, and this book is already long enough. You can search the web for "critical t-values table."

I included this section here more as an historical observation. For your general statistical knowledge, it is good to know what the term "critical value" means, and I do reference it a few times in this chapter and other chapters (for example, I will call the critical t-value τ in the discussion of sample size calculations in Chapter 17), but it's not something you'd actually use unless you're a statistician in a post-apocalyptic civilization without electricity.

11.1.8 Assumptions of the t-test

There are specific assumptions made by each t-test variant (e.g., assumptions about equal variances), which you will learn about later in the chapter. The list below describes general assumptions of all t-tests.

- **Normality**: The main assumption of a t-test is that the mean and standard deviation are valid and useful characterizations of the samples. This basically means that the data are roughly normally distributed. If the mean is not a useful characteristic of a dataset, then it doesn't make sense to perform a statistical test on that mean.

  Fortunately, many non-normal distributions can be transformed into a normal distribution, as you learned in Chapter 8. That said, t-tests are fairly robust to violations of this assumption; don't stress about getting your data distribution to be a perfect Gaussian. Alternatively, you can use non-parametric t-tests to evaluate differences in *medians* instead of means.

- **Interval or ratio data**: As you know from Chapters 2 and 4, quantities like mean and standard deviation are valid only for interval or ratio scale data. That said, discrete-numeric data and ordinal data might be OK for a t-test if there is a relatively broad range of values and if the sample size is large.

- **Independent observations**: The data samples should be independent of each other. This means that the outcome of one observation should not influence the outcome of another observation. Dependencies in the data can inflate the t-value and reduce the generalizability of the finding.

  Autocorrelations in the data are common in spatial, image, and time series data, and can be addressed with additional corrections or alternative methods.

- **Random sampling**: Related to independence, the data should be collected through random and representative sampling, i.e., each member of the population has an equal chance of being sampled.

Although the t-test is generally robust to violations of assumptions, it is a good habit to check for these assumptions in your data. Severe violations can lead to inaccurate results and misinterpretations. You'll see several examples of incorrect conclusions resulting from severe violations of assumptions in the exercises of this and later chapters.

11.1.9 Testing for normality

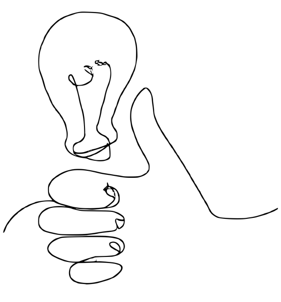

Figure 11.7: Sometimes, normal is good.

There are several ways to examine whether the distribution is normal. You've already learned about visual inspection-based methods like the histogram and the QQ plot.

Quantitative analyses for testing for a normal distribution involve evaluating the null hypothesis that the data are normally distributed. Therefore, a p-value larger than .05 would indicate that we cannot reject H_0, which means that we accept that the data are normally distributed. It's one of the few scenarios in statistics where we *want* a non-significant result. Conversely, if the p-value is less than .05, then we reject H_0 and interpret that the data are non-normally distributed.

Here I will introduce two statistical tests for normality:

Omnibus test
This test compares the skew and kurtosis of the data distribution to the values expected for a normal distribution. The test is implemented in the `scipy.stats` library with the function `stats.normaltest(X)`.

Shapiro-Wilk test
This test works by comparing the distribution of the data to values that would be expected given a normal distribution with the same mean and standard deviation as the empirical data. It is conceptually similar to a QQ plot, but the comparison is quantitative instead of qualitative. The resulting test statistic value is called W and varies between 0 and 1, with values closer to 1 indicating a closer match to a normal distribution. The Shapiro-Wilk test is called and interpreted similarly to the Omnibus test using the function `stats.shapiro(X)`.

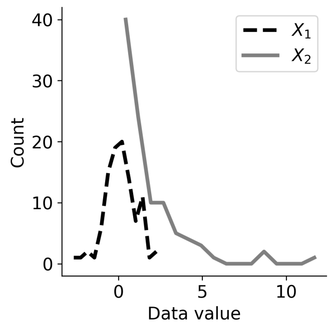

Figure 11.8: Distributions of two datasets to illustrate tests of normality. Both tests were significant for X_2 and non-significant for X_1.

As an illustration, I applied both tests to two N=100 datasets comprising random numbers drawn from a Gaussian distribution and an exponential distribution (Figure 11.8). The p-values for the Gaussian distribution were p=.25 and p=.24 for the Omnibus and Shapiro-Wilk test, respectively; and were both p<.001 for the exponential distribution.

Both of these tests are sensitive to sample size. In particular, very large samples can produce a small p-value with only minor and inconsequential deviations from a normal distribution. Therefore, these tests should be used as guides to help you make decisions alongside qualitative data inspection. Don't make important decisions about the data from these tests only from the p-value without looking at the data, especially if the sample size is large ("large" is a subjective term, but let's say >50).

11.2 How to make a t-test significant

It is very noble to say that we don't care how the result turns out; we simply collect data, run the appropriate statistical tests, and then report the results.

But the truth is that we all *want* to get significant results in our research. That's not a bad thing — wanting to get significant results can help ensure the experiments are well designed and the data are high quality.

And with this in mind, I will explain the three different ways to maximize your t-value. It's useful to think about the different dimensions by which the t-value can become large, because you may have control over some but not other of these dimensions.

For reference and elucidation, I have rewritten Equation 11.1 below:

$$t_{df} = \frac{\overline{x} - h_0}{s/\sqrt{n}} = \frac{(\overline{x} - h_0)\sqrt{n}}{s} \tag{11.4}$$

Increase average differences (Figure 11.9B)

Obviously, the larger the numerator of the fraction, the larger the t-value. This is relevant for experiment design because if you anticipate a large amount of variability and/or a small sample size, you should attempt to maximize the magnitude of the mean differences.

Reduce variability (Figure 11.9C)

The smaller the denominator, the larger the fraction. This means that if you know the effect will be modest, you should try to minimize the variability. You can do this by refining the experiment manipulations, making sure the measurement sensors are precise, selecting a sample that is more homogeneous, and cleaning the data to remove outliers.

Increase sample size

This is the reason why I rewrote Equation 11.4 with the $\sqrt{n}$ in the numerator: Increasing the sample size will increase the t-value, even if the mean difference and standard deviation remain the same. It is a nonlinear impact, so increasing the sample size by a factor of four will only double the t-value. Still, every bit helps. You can also think back to the discussion of the standard error of the mean: increasing the sample size increases our confidence in the effect. Increasing sample size is a good strategy when data are cheap but less well-controlled. This is a typical strategy, for example in medical studies that use hospital records: The variability is likely to be large and the effect size may be small (think, for example, of the impact of coffee on longevity), but researchers can acquire sample sizes in the tens of thousands.

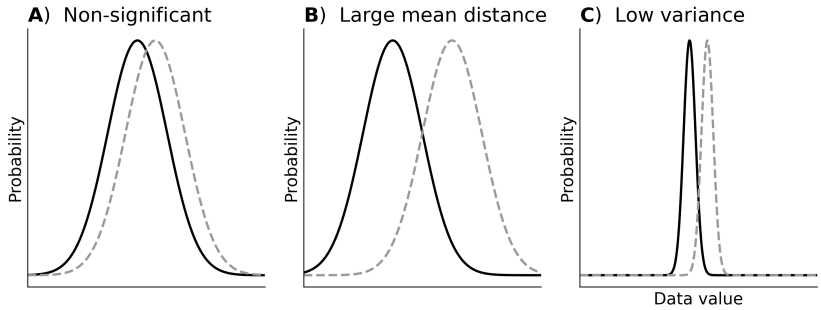

Figure 11.9: Each panel shows two lines that depict histograms of two data samples in a two-sample t-test. The distributions in Panel A are mostly overlapping, so the t-test on their mean differences will not be significant. The distributions in Panel B have the same variance as those in panel A but the means are further apart, so the t-test will be significant. The two distributions in panel C have the same means as those in panel A, but the variances are much smaller, leading to a significant t-test.

I separated these different components of the t-value because different kinds of research and different kinds of experiments allow for control over different factors. For example, if you are doing clinical research on Schizophrenia patients over 60 years old, then you will probably have small sample sizes, and you can assume that the variability will be large. So you will need to design the research to look for large effects. On the other hand, if you are performing market research on whether the color of an advertisement increases sales, you can assume that the magnitude of the effect will be small and the variability will be high, so you will need to plan on collecting large samples.

Datasets can differ from each other in characteristics other than their means. In fact, there are many descriptive characteristics that are independent of the mean, including variance and higher statistical moments. Although the mean is the most appropriate characteristic to evaluate in many cases, a non-significant t-test does not indicate that the samples do not differ; it indicates only that the means of those samples do not significantly differ.

11.3 One-sample t-test

I hope you have found the chapter thus far enlightening. My main goal was to help you grasp the fundamental principles that underpin the t-test. This is the crucial conceptual content; the remainder of this chapter delves into finer details regarding the adaptation of the t-test formula for various scenarios that I introduced at the outset of this chapter.

Let's begin with the one-sample t-test. It is the simplest form of the t-test, because it involves only one sample and therefore one standard deviation. The formula is presented in Equation 11.5:

$$t_{n-1} = \frac{\overline{x} - h_0}{s/\sqrt{n}} \tag{11.5}$$

Let's work through an example: A teacher wants to know if the average exam score of her students is significantly different from the national average of 75 points. There are 15 students, and their grades are[2] as follows.

$$X = [80, 85, 90, 70, 75, 72, 88, 77, 82, 65, 79, 81, 74, 86, 68]$$

The Shapiro test for normality had p>.05, indicating that the data meet the normality assumption of a t-test.

Before writing any code, we need to translate the hypotheses into a model — a pair of mathematical statements that define the null and alternative hypotheses. For t-tests, the mathematical versions of the hypotheses are usually simple and easy to define.

- $\mathrm{H_0}$: $\overline{X} = 75$
- $\mathrm{H_A}$: $\overline{X} \neq 75$

Notice that the test here is two-tailed.

The mean and standard deviation of this sample (rounded to the nearest tenth) are $\overline{X} = 78.1, s = 7.5$. The t-statistic is:

$$t_{14} = \frac{78.1 - 75}{7.5/\sqrt{15}} = 1.624 \tag{11.6}$$

We can compute the p-value using the cdf of the t-distribution. In this case, $t_{16} = 1.624, p < .127$. Because the p-value is larger than .05, we cannot reject the null hypothesis. We therefore conclude that the average exam score in this classroom was not significantly different from the national average.

[2] These are fake data made up for this example.

Even without conducting a formal t-test, you can see that the effect is unlikely to be significant. Consider that the difference between the classroom and national average scores is around 3, which is less than half of the standard deviation. A mean difference smaller than its standard deviation is unlikely to be significant. You can gain a lot of insight into data just by looking at the numbers.

T-test with the `scipy.stats` library The Python function for the one-sample t-test works as follows:

```
ttest = stats.ttest_1samp(X,h0)
```

In other words, you input the dataset and the H_0 value, and the function outputs a variable that here I call `ttest`. This is not a float, `numpy` array, list, or any other datatype that you've encountered so far in this book. It is a `TtestResult` object that contains the three key pieces of information we need from a t-test:

```
print( type(ttest) )
print(ttest)

>> <class 'scipy.stats._stats_py.TtestResult'>
>> TtestResult(statistic=1.62, pvalue=0.12, df=14)
```

I've truncated the output so it would fit on the page; the t- and p-values are calculated to a ridiculous precision. Importantly, you can see that those values match what I wrote above.

11.4 Two-sample t-tests

Two-sample t-tests are used to test whether the means of two samples are significantly different. The general formulation of the null and alternative hypotheses are:

- H_0: $\overline{X} = \overline{Y}$
- H_A: $\overline{X} \neq \overline{Y}$

It is sometimes useful to think of the hypotheses as equations set to zero:

- H_0: $\overline{X} - \overline{Y} = 0$
- H_A: $\overline{X} - \overline{Y} \neq 0$

Two-sample t-tests come in two flavors: paired-samples and independent-samples.

11.4.1 Paired samples t-test

This is also called a *dependent t-test*.

The paired samples t-test is used when the two samples come from the same individuals. This is common for experiments in which an intervention or manipulation is introduced, and people are measured before and after the intervention. A few examples[3]:

1. **Anti-aging supplement**: A research lab is testing whether a newly developed molecule slows biological aging. The paired-samples t-test would compare the average telomere lengths[4] before and after the treatment.
2. **Flipped classroom model**: A teacher switches from the traditional classroom model (lectures during the day and homework in the evening) to a flipped classroom model (video lectures in the evening and individual/group work during the classroom period) in order to determine whether test scores improve. The paired-samples t-test would be used to compare the average difference in test scores before and after implementing the new teaching method.
3. **Background noise on reading comprehension**: A researcher wants to investigate the effect of background noise on reading comprehension. Participants answer true/false questions based on text that they read with and without background auditory noise. The paired-samples t-test would be used to compare the average difference in reading comprehension scores between the quiet and noisy conditions.

A paired t-test is implemented by subtracting the two data values from each individual and then performing a one-sample t-test exactly as described in the previous section. This is not only a simple method, but it is also quite powerful because it reduces the variability in the data. I will illustrate this with an example.

[3] Proper experiment design for these examples should include a placebo or control condition, but let's ignore that in the interest of simplicity.

[4] Telomeres are DNA snippets that protect chromosome boundaries. They shorten with age and predict negative health outcomes, and are therefore used as a biological marker of aging.

Let's continue with the reading comprehension example. Imagine that comprehension test scores range from 0 to 100, and I'll use X_N and X_Q to indicate the comprehension scores in the noisy and the quiet conditions. Here are the data[5]:

Both samples had Shapiro p's>.05.

$$X_N = [60, 52, 90, 20, 33, 95, 18, 47, 78, 65] \tag{11.7}$$

$$X_Q = [65, 60, 84, 23, 37, 95, 17, 53, 88, 66] \tag{11.8}$$

Visual inspection of the data (Figure 11.10A) reveals a large amount of variability in the scores, which indicates that our research participants have very different baseline reading comprehension abilities.

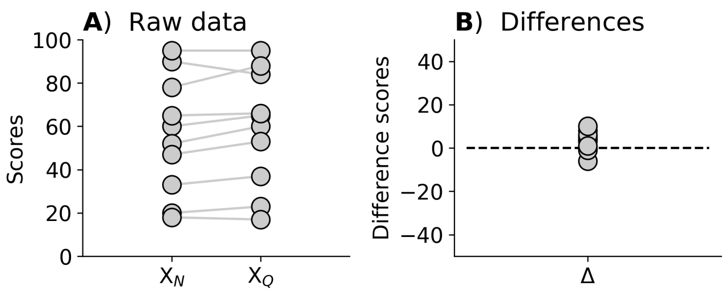

Figure 11.10: Panel A shows the raw scores for the two reading conditions. Note that each participant contributes two data points, and the linked data values are indicated with the gray lines. Panel B shows the change in the comprehension scores ($\Delta = X_Q - X_N$).

But we get a different sense of the data when we examine the difference scores (I'll call variable $\Delta = X_Q - X_N$).

In this example, a subtraction was sufficient to normalize the inter-subject differences. In other cases, it might be useful to normalize the data by computing percent change. See Exercise 11.7.

$$\Delta = [5, 8, -6, 3, 4, 0, -1, 6, 10, 1]$$

The difference scores are shown in Figure 11.10B. Notice that both y-axes have the same range (spanning 101 units), meaning that the variabilities in the two plots are directly visually comparable. In other words, although the *baseline* reading comprehension is very different across individuals, the *change* in reading comprehension due to background noise is comparable across individuals. This highlights the power of within-subjects analyses for reducing variability (a theme that will re-appear when you learn about repeated-measures ANOVAs).

[5]Fake data.

By the way, why did I compute Δ as $X_Q - X_N$ and not $X_N - X_Q$? As I mentioned earlier in the chapter, the order of the subtraction doesn't matter for the statistical significance of the two-tailed t-test (of course the sign matters for a one-tailed test). Although the sign is statistically arbitrary, the order can facilitate interpretation. In this example, I expect that comprehension will be higher in quiet compared to noisy backgrounds, so I set up the equation such that a *positive* t-value would correspond to an *increase* in performance in a quiet environment. Had I computed $\Delta = X_N - X_Q$, we'd expect a *negative* t-value, which we'd interpret as a *decrease* in performance in a noisy environment. Perhaps you prefer that interpretation.

Now that we've reduced the data from two samples into one, we proceed exactly as we did with the one-sample t-test. In this case, $\mathrm{H}_0 : \overline{\Delta} = 0$ and $\mathrm{H}_\mathrm{A} : \overline{\Delta} \neq 0$. With the numbers I made up, the result is $t(9) = 2.023, p < 0.074$. In other words, the t-test is not statistically significant.

Let's imagine that these are real data for a real PhD dissertation. What do we do with this p-value? It is not statistically significant at the accepted α level, but it is close. And when looking at the difference scores Δ and in Figure 11.10B, there are only two individuals who showed a negative change. Now, I could sit here in my high horse inside the ivory tower, give you a patronizing look, and say "it is not significant, end of story." But the truth is that this study has important implications for education and society (e.g., libraries, offices, coworking spaces) — not to mention the importance to the junior researcher who needs this study for their dissertation. It does "look like" there is a real effect in the data, and the direction of the effect is consistent with common sense. Some people will be tempted to try various "tricks" to get the p-value down, like removing a participant, trying various data normalizations until something works (see Exercise 11.7), or different ways of calculating the comprehension scores. These are all dangerously close to "p-hacking," which refers to unethical manipulative statistical practices to obtain a desired result. One ethical approach, for example, would be for the researcher to double the sample size and not perform statistics again until the final sample is collected. I will have much more to say about this topic in Chapter 18, but I wanted to introduce the issue now.

Missing data Because the goal of a paired t-test is to evaluate a *change* in a variable, missing data are extremely problematic (see also discussion about missing data in Chapter 7). There are two options for dealing with missing data in a paired-sample t-test:

Row-wise removal: Remove all data from any individual with one missing data value. This is a good option when you have a large dataset and can afford to lose data.

Imputation (replace with interpolated values): This involves "guessing" the missing data value based on the mean of other individuals or a prediction based on a regression or machine-learning analysis.

To be honest, neither of these options is very savory. My preference is row-wise removal because I am slightly uncomfortable with making inferences based on data that are modeled instead of measured. But that's my opinion and intuition, and not everyone agrees with me.

Either way, if you anticipate a large amount of missing data before collecting the data, then do whatever you can to maximize the amount of data you collect. Having a large dataset will help with both missing-data strategies.

11.4.2 Independent samples t-test

An independent two-samples t-test, also called an unpaired t-test, tests whether the means of two separate groups significantly differ. This is different from the paired-samples t-test where the same individuals are measured twice; with the independent samples t-test the two data points are not paired, because they do not come from the same individuals. Indeed, the sample sizes might differ between the two groups.

Here is an example of when an independent samples t-test would be appropriate: A start-up company that makes a card-playing app has two user-interaction designs, and wants to know which leads to higher engagement times. They randomly assign 50 people to use design "A" and 40 people to use design "B." The DV is time spent on the app.

Because the two samples come from different individuals, we need to account for differences in standard deviations and sample sizes. Equation 11.9 shows a formula that separates the sample sizes and variances for each group. This is also called Welch's test.

$$t = \frac{\overline{X} - \overline{Y}}{\sqrt{\frac{s_x^2}{n_x} + \frac{s_y^2}{n_y}}} \tag{11.9}$$

where s_x^2 is the sample variance of variable X and n_x is its sample size. In fact, this is not the *only* formula for a two-samples t-test; there are several variants of this formula that are applied depending on whether the two groups have equal sample sizes and/or variances. Please take a moment to simplify the denominator of Equation 11.9 assuming that the variances and sample sizes are equal; you will find that the t-value reduces to the same form as that of the one-sample t-test. You can also simplify the denominator assuming equal variances but different sample sizes. In practice, you determine the appropriate conditions on your own and provide the appropriate input instructions to Python functions.

Degrees of freedom If the variances are roughly equal, the *df* are $n_x + n_y - 2$. If the variances are unequal, a correction factor is applied, which leads to a more complicated *df* formula. I will show that formula in Exercise 11.9, but essentially, it involves adjusting the sample sizes according to the variances.

Testing for equal variances The statistical lingo for equal variances is "homogeneity of variances" (the opposite — unequal variances — is called "heterogeneity of variances"). The question is, How do you know whether the two groups have "equal" variances? Of course, due to sampling variability, the variances won't be *exactly* equal even if they are drawn from populations with equal variances. Therefore, the question is whether the variances are close enough to assume homogeneity.

There are three ways to determine whether the variances of the two populations (as estimated by the variances of the two groups) are equal: One is to visualize the data and make a qualitative determination. This is feasible when testing a relatively small number of datasets. A second method is to use the "doubling rubric," which means to determine whether the standard deviation from one group is less than twice the standard deviation of the other. In other words, if $s_{max} < 2s_{min}$, then you can assume homogeneity of variance. The third, and most rigorous, method is to use Levene's test[6], which tests the null hypothesis that $s_1 = s_2$. I won't present the math of Levene's test here, but it is based on the principles of a one-way ANOVA. If the p-value of Levene's test is non-significant (that is, if p>.05), then you can assume homogeneity of variances.

[6] There are other inferential statistics for testing homogeneity of variance; Levene's is a common one.

An example Let's work through an example. I will create two groups that differ in means, standard deviations, and sample sizes. And just to make things interesting, I will draw them from different distributions.

I created two datasets, one of 50 numbers drawn from an exponential distribution, and one of 42 numbers drawn from a Gumbel distribution (see online code). You can see the data values and their histograms in Figure 11.11.

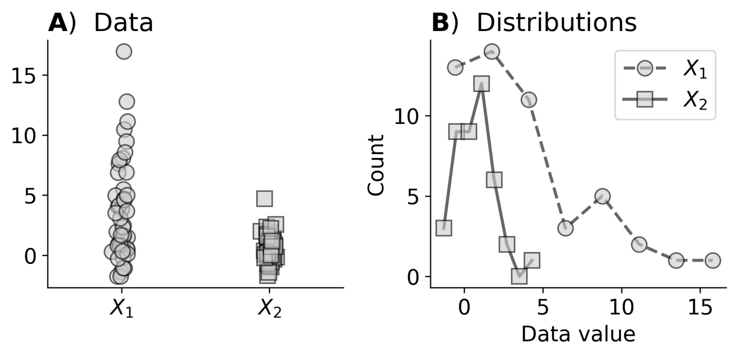

Figure 11.11: Two independent samples. Panel B shows histograms displayed as lines.

Panel A allows us to perform the visualization check for homogeneity of variance. It does seem like the variance is higher for X_1 compared to X_2. The doubling ratio is only 1.74, but the Levene's test has a p-value of .005, which is statistically significant. Remember that the Levene's test has $\mathrm{H}_0 : s_1 = s_2$, which means that we reject the null hypothesis of variance homogeneity, ergo we assume that the variances are unequal. (With sampling variability and these sample sizes, you will sometimes find the variances to conform to the homogeneity assumption.)

I also tested both samples for normality. These tests were somewhat inconsistent, in that some random datasets had evidence for normality while other random datasets had evidence against normality. Furthermore, the conclusions from the Shapiro and Omnibus tests were not always consistent with each other. In the case of the data shown in Figure 11.11, the p-values were .03 and .18 for X_1 and X_2. Nonetheless, t-tests are fairly robust to minor violations of the normality assumption.

Let us now proceed to the t-test assuming unequal variances. The Python code is:

```
tres = stats.ttest_ind(data1,data2,equal_var=False)
```

The df of this test is 90, because the two samples have 42 and 50 observations, so $n_1 + n_2 - 2 = 90$

Notice that I specified that the variances are unequal. The default value of the parameter `equal_var` is `True`. The results are $t_{90} = 5.95$ with $p < .0001$, meaning that the means of the two groups are statistically significantly different.

What does this result signify, given the data? X_1 is non-normally distributed, so it is questionable whether the mean is really a useful description. Nonetheless, visual inspection of the data clearly shows that data X_1 has larger data values than X_2. A log transform could make X_1 more normal, but it might make X_2 less normal. I believe in this case that the t-test is useful even if the data do not appear to be a "textbook" example of the ideal circumstances. Reality is rarely ideal, and applied statistics books should reflect reality.

You might be wondering whether it really matters if we assume equal or unequal variances. In this example, the t-test result was highly significant regardless of this assumption; you'll have the opportunity to explore this assumption in Exercise 11.9.

11.5 Effect size

An *effect size* is a quantitative measure of the magnitude of the observed effect. There are two measures of effect size in t-tests: Cohen's d and R^2. In this section, I will describe and define these effect size measures, and then discuss how they are related to the t-value.

Cohen's d[7] is calculated as the difference between two means divided by a standard deviation. It provides an estimate of the magnitude of the difference between the two groups under study, transformed into standard deviation units. Here are the formulas (I'm using d_1 to indicate Cohen's d for a one-sample t-test, d_p to indicate a paired-sample test, and d_2 to indicate a two-sample test):

[7] Unrelated.

$$d_1 = \frac{\overline{x} - h_0}{s} \tag{11.10}$$

$$d_p = \frac{\overline{x_a} - \overline{x_b}}{s_\delta} \tag{11.11}$$

$$d_2 = \frac{\overline{x} - \overline{y}}{\sqrt{((n_x - 1)s_x^2 + (n_y - 1)s_y^2)/(n_x + n_y - 2))}} \tag{11.12}$$

In Equation 11.11, s_δ is the standard deviation of the difference. These equations look really similar to the equation for the t-test except without the $\sqrt{N}$ factor. That leads to an important distinction between effect size and t-value, which I'll discuss more later.

Because Cohen's d has units of standard deviation, you can interpret this measure of effect size the same way that you would interpret a z-transformed variable. Although Cohen's d can be negative, it is customary to take the absolute value or arrange the numerator to get a positive result. For some reason, people find negative effect sizes uncomfortable.

R-squared (R^2 or R2) is also called *coefficient of determination*, and is a different measure of effect size. You'll see R^2 appear several times in statistics, including correlation, regression, and ANOVA. The formulas differ by application, but the interpretation is the same: It represents the proportion of the variance in the DV that can be explained by the IV.

$$R^2 = \frac{t^2}{t^2 + df} \tag{11.13}$$

R^2 is a value between 0 and 1, where 0 indicates no effect (t=0) and values closer to 1 indicate a stronger effect. For a t-test, R^2 can never truly be 1, because there will never be zero degrees of freedom. Nonetheless, as the t-value increases relative to the degrees of freedom, R^2 will approach 1.

Cohen's d is more commonly reported than R^2, in part because it was more commonly reported in the past (in statistical reporting, like in real life, traditions maintain inertia). Anyway, you'll discover in Exercise 11.11 that the two quantities are closely related.

11.5.1 Effect size vs. t-value

Effect size and t-value are different but complementary quantities.

The t-value is a measure of the departure from the null hypothesis t-value distribution, and, when transformed into a p-value, gives an indication of how unlikely the observed data would be, assuming the null hypothesis were true. On the other hand, the effect size is a measure of the magnitude of the effect, regardless of its probability relative to an H_0 distribution.

The key difference between the t-value and Cohen's d is the scaling by $\sqrt{n}$ in the t-value. The implication of this scalar is that a large t-value might result from a large sample size, even if the effect size is small (more on this in Exercise 11.11).

One way to think about the distinction is that the t-value is a measure of *statistical* significance while the effect size is a measure of *practical* significance.

11.6 Nonparametric t-test alternatives

Nonparametric t-tests involve comparing medians instead of means. You can use these tests when the data strongly violate the normality assumption (and when data transformations are not desirable), or when the data contain outliers that you do not want to remove because they are valid though non-representative.

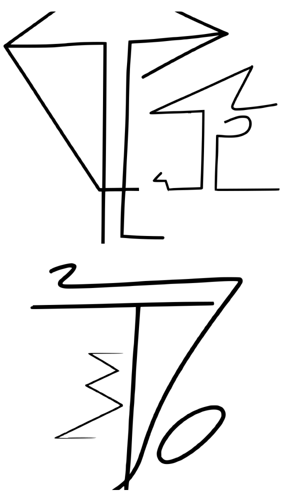

Figure 11.12: Nonparametric options for the non-normal life.

It would be nice if the formulas were as simple as replacing the mean with the median in all the equations presented in this chapter. Unfortunately, non-parametric t-tests are based on formulas that are more complicated and less intuitive; fortunately, the use and interpretations of the tests and their p-values are the same.

The methods presented in this section are not formally *t-tests*, because they do not produce a t-statistic, nor are their test statistic values evaluated against a t-distribution. But they have the same function as a t-test — evaluating whether the central tendency of the data differs from a pre-specified value — so they're considered non-parametric alternatives to the t-test.

11.6.1 Wilcoxon signed-rank

The *Wilcoxon signed-rank test*, also called the Wilcoxon test or the signed-rank test, is a medians-based replacement for the one-sample t-test or the paired t-test.

The steps below describe the algorithm for the Wilcoxon signed-rank test, but the summary version is that if the data were evenly distributed around the H_0 value, then the number of data points to the left of H_0 should equal the number of data points to the right of H_0.

1. **Step 1**: Remove data points that equal the H_0 value (for a one-sample test) or pairs that are equal (for a paired-sample test). The reasoning is that these values do not provide evidence for or against the null hypothesis.

2. **Step 2**: Compute the difference between each data point and H_0 value (for a one-sample test) or between the pairs of data points (for a paired-sample test).

   See implementation notes below for cases where $H_0 \neq 0$.

3. **Step 3:** Rank-transform the absolute values of these differences, and sort the ranks in ascending order. Multiply these ranked absolute differences by the signs of the data from Step 2. Essentially, this involves multiplying the ranks from the data below the H_0 value by -1. These are the "signed rank" values from which this analysis gets its name (also Professor Frank Wilcoxon). Let's call the result of this step variable r.

   This step also removes outliers.

4. **Step 4**: Count the number of negative r values and the number of positive r values. The smaller of these sums is called variable w.

   See implementation notes below about Step 4.

5. **Step 5**: Transform w into a z value from which a p-value can be obtained. The transformation is done through a somewhat complicated formula:

$$z = \frac{w - n(n+1)/4}{\sqrt{\frac{n(n+1)(2n+1)}{24}}} \tag{11.14}$$

   This z-score can be interpreted as a standard z-score, to which a p-value can be associated, and that p-value is the significance of the Wilcoxon signed-rank test.

I have an example for you. I created non-normally distributed data as x^2 for $x \in \mathcal{N}(0,1)$ and tested against the null hypothesis value of $H_0 = 1$ (see Figure 11.13). I then computed the Wilcoxon test using the code:

```
wtest = stats.wilcoxon(data-h0,method='approx')
```

In these data, the z-score was -.95, which has an associated p-value of .341. This is greater than the threshold of .05, so we do not reject the null hypothesis; the empirical median is not statistically significantly different from H_0=1.

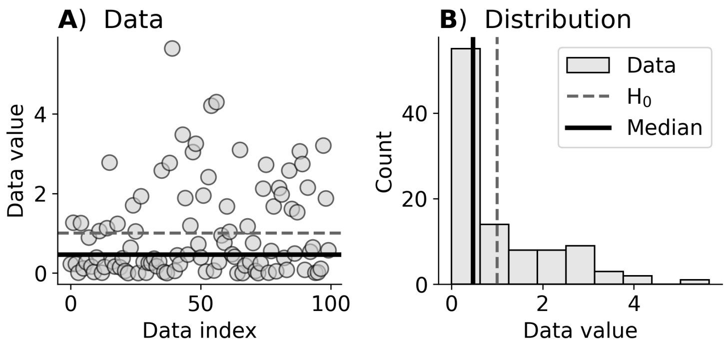

Figure 11.13: Example of Wilcoxon rank-sign test on non-normally distributed data.

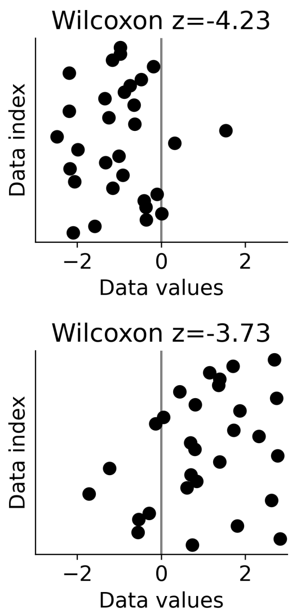

Figure 11.14: The Wilcoxon z in Python reflects the negative asymmetry around the H_0 value (vertical gray line).

A few notes about implementing the Wilcoxon test using the Python function `stats.wilcoxon()`:

- This function evaluates the null hypothesis that the median of the data is zero, i.e., that $H_0 = 0$. Therefore, you need to input a data vector that has already been shifted by the null hypothesis value. That is, set $\widetilde{X} = X - H_0$, and then proceed to work with $\widetilde{X}$ in the steps outlined earlier.
- However, for a paired-samples test, input both data vectors separately (i.e., `stats.wilcoxon(X,Y)`). You can equivalently subtract the vectors and then input one vector, i.e., input $\widetilde{X} = X - Y$.
- The function outputs the W value and its p-value. If you want a z-value, use the optional input `method='approx'` (see online code).
- There are some minor differences in implementation among different software packages (Python vs. MATLAB vs. R). Confusingly, Python takes the *smaller* of the signed-rank count, which means the z-value will be negative even if most of the data points are above H_0 (Figure 11.14). In practice, you should either avoid interpreting the sign of the z-value, or visualize the data to determine how the concentration of data points relates to the H_0 value.

11.6.2 Mann-Whitney U test

This test is variously called the Mann-Whitney U test, the Mann-Whitney-Wilcoxon U test, or the Wilcoxon rank-sum test. It is a medians-based alternative to the independent two-samples t-test, and it can be used on any numerical data type and for data with any distribution shapes and characteristics.

After some deliberations, I decided not to describe the algorithm in as much detail here as I did with the Wilcoxon test, because it is more involved than that of the signed-rank test, and I don't think it provides much insight into the nature or interpretation of the test. Briefly: the test is based on ranking the combined data from both groups. A variable U encodes whether the ranked observations in one group tend to be higher than the ranked observations in the other group. That variable U is then transformed into a z-value that is normally distributed under the null hypothesis that the medians of the two groups are equal. The corresponding p-value is used to determine the statistical significance of the Mann-Whitney U test.

Because the Mann-Whitney U test is based on medians and ranks, you don't need to worry about variance assumptions like with the independent two-samples t-test.

An example I applied the Mann-Whitney U test to the data I used to illustrate the independent t-test (Figure 11.11). We already know from the normality tests that random data from these distributions are sometimes non-normally distributed, which justifies the use of a non-parametric test. The test is easy to implement in Python:

```
mwu = stats.mannwhitneyu(data1,data2)
print(f'U={mwu.statistic:.2f}, p={mwu.pvalue:.3f}')
```

The p-value was very small ($p<.001$), indicating that the medians of the two distributions are statistically significantly different from each other.

11.6.3 Permutation testing

Another non-parametric alternative to determining the statistical significance of a t-test is to use permutation testing. Permutation testing for t-values has several advantages for data that contain outliers or are non-normally distributed, or for applying corrections for multiple comparisons when there are many tests to perform in correlated data.

Chapter 16 is dedicated to permutation testing, so I will postpone a detailed elucidation until then.

11.7 More than two samples?

The t-test variants I introduced in this chapter are for one or two samples. What do you do if your experiment design has more than two samples? Perhaps you have data from a medical experiment that compared three different medications in two different patient groups. That's six groups in total.

You might think of running a series of t-tests to compare all pairs of samples (12 two-sample t-tests in the example above). Although this is technically possible and (very) occasionally acceptable, it leads to a multiple comparisons problem, and can incorrectly specify the variance of paired samples. It also limits the ability to test for interactions between experiment factors (e.g., if the effect of medication depends on the patient group). Therefore, if you have more than two samples to compare, the appropriate analysis is an ANOVA.

11.8 Exercises

11.1. The goal of this exercise is to implement a one-sample t-test by implementing the formulas I showed in this chapter, and then compare your results against the output of `scipy`'s t-test routine. The purpose is to make sure you fully understand how to create t- and p-values.

Begin by creating a dataset of N=50 numbers randomly drawn from an asymmetric Laplace distribution with κ=2 (use the `laplace_asymmetric` module in `scipy.stats`), and test whether the mean of that dataset is significantly different from $H_0 = -\pi/2$. Before coding the statistics, visualize the data as in Figure 11.15.

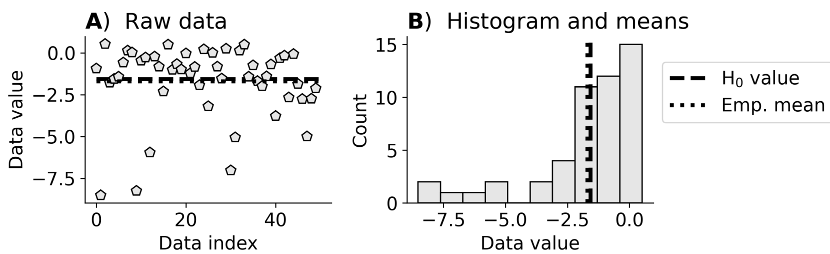

Figure 11.15: Visualization for Exercise 11.1. Panel A shows the data and panel B shows the histogram. The dashed line is the H_0 value against which to compare the empirical sample mean, which is depicted by the dotted line.

Next, compute the t-value of this test using Equation 11.5, and compute the corresponding p-value using `stats.t.cdf`. Then, obtain the t-test result using `stats.ttest_1samp`. Print both results to make sure they match. My results were:

If your results do not match those of `scipy`, check that you're using a two-tailed test!

```
Manual ttest: t(49)=-1.376, p=0.175
Scipy  ttest: t(49)=-1.376, p=0.175
```

11.2. In the previous exercise, the sample mean was not significantly different from the H_0 value. How stable is that result for this simulation? To find out, copy the code from the previous exercise into a for-loop that generates 500 datasets (each using the same distribution and sample size parameters) and counts the number of times that a $p<.05$ result were obtained. In one of my code-runs, I found that

43/500 datasets had a $p<.05$ result[8].

What is different about those subthreshold datasets? To find out, make scatter plots of the sample means and sample standard deviations for the datasets that had a corresponding p-value less than vs. greater than .05. Visualize your results as in Figure 11.16.

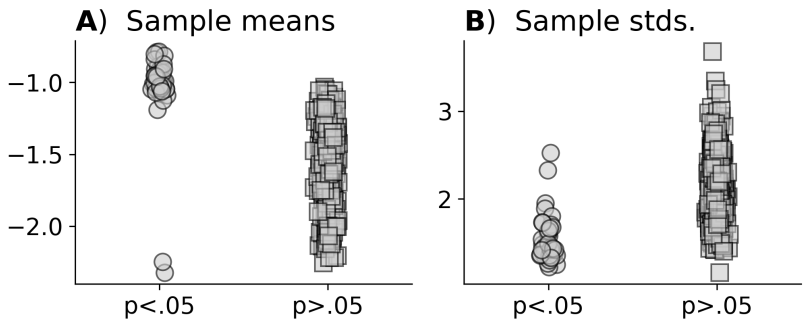

Figure 11.16: Visualization for Exercise 11.2.

It is interesting, though not surprising, to see that the "significant" samples had — purely by chance — means that were at the edges of the distribution, and also relatively small sample standard deviations. It is not surprising because the formula for the t-test selects for samples that have these characteristics.

I sometimes find these kinds of simulations troubling. There is one of two possible states of the world here: Either H_A is correct or it is incorrect. If it is correct, then we have lots of Type-II errors (incorrectly failing to reject H_0). And if H_A is incorrect, then we have quite a few false alarms. It would be nice if statistics would give us an absolute result that we could absolutely trust; but even in simulated data, all we get are probabilities that help us make decisions.

11.3. The `scipy` function `stats.ttest_1samp` accepts a matrix as input, with rows corresponding to data observations and columns corresponding to datasets. The function output will be a vector of t- and p-values, one for each dataset. Thus, if you have multiple datasets of the same sample size, you can run t-tests on all datasets at once, without a for-loop.

[8]If these were real data, t-tests on 500 independent samples would require a correction for multiple comparisons.

Create a data matrix with size 40×25, corresponding to 25 datasets each of size N=40. I generated normally distributed numbers from $\mathcal{N} \in (1,1)$ and tested against $H_0 = 0$, but the data characteristics don't matter for this exercise. The important thing is to repeat the t-tests twice: Once inputting the entire matrix into `stats.ttest_1samp`, and once using a for-loop to input each column separately. Print out the t-values to confirm that they are the same.

```
Matrix   |  Vector
--------|--------
8.3109   |  8.3109
6.6441   |  6.6441
6.2328   |  6.2328
6.1774   |  6.1774
```

I displayed only the first four tests here, but of course your result will have 25 rows.

Now that you know how to implement many t-tests without a for-loop, revisit the previous exercise to eliminate that for-loop.

11.4. In this exercise, you will empirically confirm the importance of the sample standard deviation for statistical significance, a concept I illustrated in Figure 11.9C.

Create 300 datasets of 40 numbers drawn from normal distributions that have a standard deviation (σ) ranging linearly from .1 to 3, and a theoretical population mean of $\mu = 0$. Then, force each sample mean to be $\overline{x} = .5$. Perform a one-sample t-test against the null hypothesis value of zero on each dataset, and plot the t-values and p-values as a function of the standard deviation. Visualize the results as in Figure 11.17A-B. Then plot the p-values as a function of the t-values (11.17C).

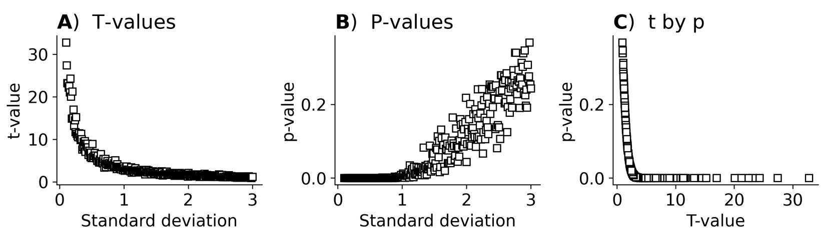

Figure 11.17: Visualization for Exercise 11.4.

It is interesting to see such a huge range of t-values even though the numerator is identical in all simulations.

11.5. Following from the previous exercise: The x-axis shows the population standard deviation (the second input into the `np.random.normal` function), but the t-test uses the empirical sample standard deviation. Would this have changed the results? Adapt the code for the previous exercise to compute the sample standard deviation, and recreate the figure. Does this affect your interpretations, or lead you to a different conclusion? (See the online code for my answer.)

11.6. One more exercise on the one-sample t-test. Here I want to impress upon you the concept that small effects sizes can become statistically significant with large sample sizes. This has considerable implications both for detecting small effects, and for the risk of false alarms in large datasets.

This experiment involves manipulating two factors: sample size and theoretical population mean. Vary the sample sizes between 10 and 810 in steps of 50, and vary the theoretical population mean between 0 and .3 in 51 linearly spaced steps. Inside a double for-loop for each combination of sample size and theoretical mean, create 250 independent datasets of normally distributed random numbers using $\sigma = 1.5$ for all simulations. Compute a t-test against $H_0 = 0$, and compute the proportion of datasets with $p<.05$. Store and visualize the results as a matrix like in Figure 11.18.

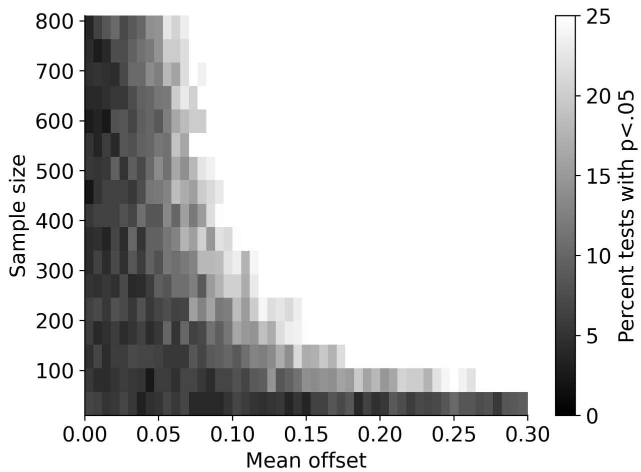

Figure 11.18: Visualization for Exercise 11.6.

The key take-home message of this exercise is that the smaller the sample size, the larger the effect size needs to be. Alternatively: If the sample is very large, even small effects can become statistically significant.

11.7. Let's combine data transformations with the paired-samples t-test. The idea will be to re-run the t-test on the "reading comprehension" data (data were presented on page 23 and are also printed in the online code) after various transformations. The first step is to apply four transformations to the data and visualize their pairwise inter-relationships.

$$
\begin{aligned}
\text{Simple subtraction:}\quad & Y_1 = X_Q - X_N \\
\text{Z-score subtraction:}\quad & Y_2 = z(X_Q) - z(X_N) \\
\text{Percent change:}\quad & Y_3 = 100(X_Q - X_N)/X_N \\
\text{Normalized ratio:}\quad & Y_4 = (X_Q - X_N)/(X_Q + X_N)
\end{aligned}
$$

$z(...)$ indicates the z-score transformation of each variable. Produce a scatter plot like Figure 11.19. Notice that all transformations are strongly correlated with each other, but they are not identical. Y_1 and Y_2, and Y_3 and Y_4, are the most closely related.

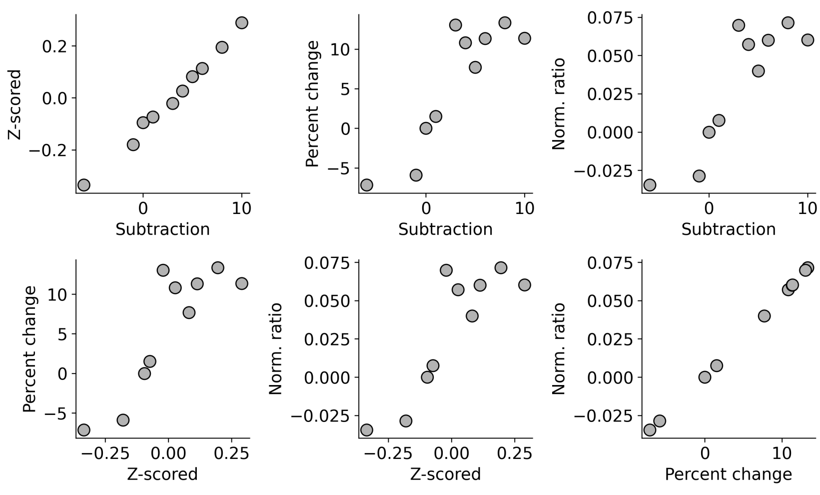

Figure 11.19: Inter-relationships for all pairs of transformations.

Now for the real question: Do these transformations have a noteworthy impact on the results of the t-test? To find out, compute and report the t-values and p-values from each transformation. My results are below (these are not random data, so you should get the same answers).

```
Subtraction (Y1): t(9)=2.023, p<0.074
Percent chg (Y2): t(9)=2.445, p<0.037
Z subtract  (Y3): t(9)=0.000, p<1.000
Norm. ratio (Y4): t(9)=2.353, p<0.043
```

Yikes! The data transformation had an *extreme* impact on the results — changing the finding from non-significant to significant. That is... thought-provoking, disturbing, and fascinating.

Before discussing the implications, let me discuss the z-score subtraction result. Are you surprised that the t-value is zero? It may seem strange at first, because the z-transformed variable is nearly perfectly correlated with the "raw" subtraction variable. But, recall that z-transformed data have a mean of zero, which means the numerator of the t-value is zero minus zero.

The fact that applying different transformations of the same data can make the finding significant or non-significant highlights the importance of understanding the underlying assumptions of the statistical test, and of the justification of the transformation. Applying transformations arbitrarily or solely to obtain a significant result can lead to false conclusions and is bad statistical practice (possibly unethical). As I discussed in Chapter 6, any transformation should be justifiable and cause minimal change to the analysis.

In a broader sense, analyses whose statistical outcomes vary considerably after minor variations in analysis parameters or design decisions tend to be less reliable. In other words, results that demonstrate robustness across a range of analytical choices inspire greater confidence.

11.8. This exercise follows from Exercise 11.6 (relationship between t-test and sample size), but for the independent-samples t-test. Create two samples of normally distributed random numbers, one with a theoretical population mean of 1 and the other with a theoretical population mean of 1.2. Use a standard deviation of 1/2 for both samples. In a for-loop, vary the sample sizes of both groups between 10 and 200 in steps of 10. For each sample size, create 100 pairs of random datasets, and perform an independent-samples t-test on each dataset. Visualize the t-values and p-values as in Figure 11.20. (Interesting to see that one significant outlier result with a *negative* t-value!)

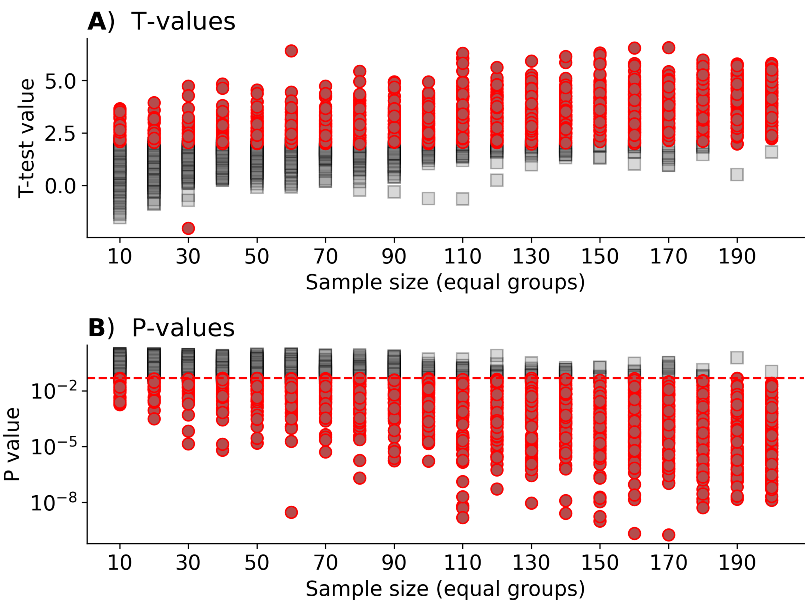

Figure 11.20: Visualization for Exercise 11.8. Each marker corresponds to the t-value and p-value from each of 100 two-samples t-tests for each sample size. The red circle markers indicate tests results with $p<.05$. The y-axis was logarithmically scaled in panel B to highlight the diversity of p-values.

It appears that the statistically significant t-tests are all above roughly the same t-value for all sample sizes. That is, there appears to be little if any impact of the sample size on the critical t-value. That may seem surprising, although if you consult Figure **??** (page **??**) you'll see that the H_0 t-distributions are quite similar across a range of *df* parameters. In fact, you can compute the critical t-values for a two-sample t-test using the same range of sample sizes you used above (10-200); you will discover that the t-value corresponding to $p<.05$ changes relatively little with sample size (figure not shown here but it's in the online code).

11.9. In this exercise, you will explore whether the homogeneity of variance assumption is crucial for evaluating the results of an independent t-test. Generate two groups of data:

$$X_1 \in \mathcal{N}(1, 1), N_1 = 50 \tag{11.15}$$

$$X_2 \in \mathcal{N}(1.1, \sigma^2), N_2 = 40 \tag{11.16}$$

After writing code to generate those two datasets (soft-coding the σ^2), write code to compute (1) the p-value from Levene's test, (2) the t-value assuming equal variance, (3) the t-value assuming *un*equal variance, and (4) the critical t-value using the adjusted *df* in Welch's method. The formula for the adjusted *df* is:

$$df = \frac{\left(s_1^2/N_1 + s_2^2/N_2\right)^2}{\frac{s_1^2}{N_1^2(N_1-1)} + \frac{s_1^2}{N_1^2(N_1-1)}} \tag{11.17}$$

Once you've coded these calculations, embed the code in a for-loop over 41 linearly spaced values of σ ranging from .01 to 15. Organize the results graphically as in Figure 11.21.

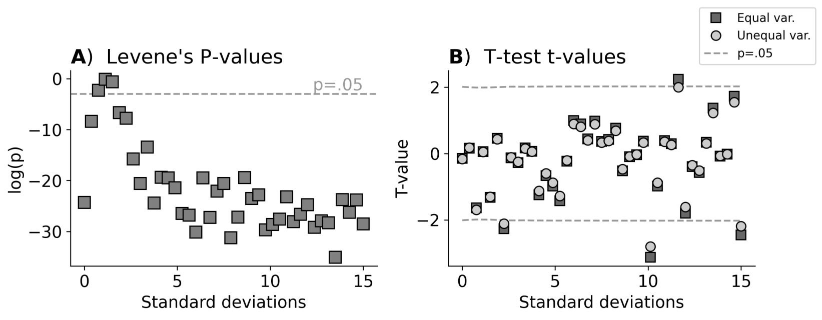

Figure 11.21: Visualization for Exercise 11.9. The horizontal dashed lines indicate the statistical significance thresholds.

The key question in this exercise is whether assuming homogeneity of variance matters. The gray squares and light-gray circles in panel B do not perfectly overlap, meaning that there is at least a numerical implication of the assumption.

The more important question is whether you would draw different conclusions about the data based on the formula adjustments (nearly all of the Levene's p-values are significant, justifying the use of unequal variance t-tests). To answer that question, look for cases where the result crosses the significance threshold without vs. with the adaptation for unequal variance. In the experiment shown above, that happened once. You can also see that the t-values are slightly inflated (that is, further from zero) when assuming equal variance. On the other hand, the statistical significance label would be the same for most of the tests done here.

The conclusion is using equal vs. unequal variance in the independent-samples t-test may not have a substantial impact on the conclusions of your research, but it's good to use the correct form to be on the safe side.

11.10. The one-sample t-test and Wilcoxon signed-rank test are not directly quantitatively comparable, because the former evaluates means and scales by standard deviations whereas the latter evaluates medians and transforms the data to rank. However, applying both tests to the same dataset does provide additional insights into evaluating the central tendency of data, as well as the sensitivity of the tests to their underlying assumptions.

Simulate 100 data points as $\exp(X\sigma)$ for $X \in \mathcal{N}(0,1)$. Mean-center the data. Perform a one-sample t-test and a Wilcoxon signed-rank test, both against the null hypothesis of .5. Repeat this procedure for 20 values of σ ranging from .1 to 1.2.

Create a set of visualizations like Figure 11.22. Panel A shows the histogram of every 3^{rd} iteration, with darker lines corresponding to smaller σ values. Panel B shows the distance to the H_0 value. The mean is always exactly .5 away from H_0 because the data are mean-centered. The median, on the other hand, drifts below the H_0 value because the data become more left-concentrated as σ increases.

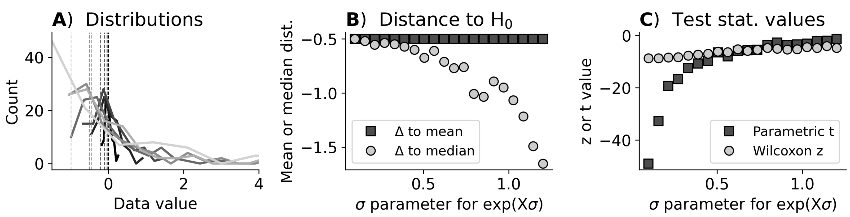

Figure 11.22: Visualization for Exercise 11.10. The x-axis in panel A is clipped to facilitate visual inspection of the lower part of the distribution; the values extend up to 20. Thin vertical lines indicate the median of each displayed distribution.

Panel C shows the statistical test results. The one-sample t-test (dark gray squares) result is striking: The numerator of the t-test is constant because the data mean and H_0 value are both constant, and yet the t-value changes dramatically as a function of the σ parameter, due to its impact on the denominator of the t-value. On the other hand, the Wilcoxon z score increases together with the decreasing distance to median. It may seem counter-intuitive that the statistical significance decreases as the median gets further away from the H_0 value; please ponder why this is the case before reading the answer below.

The relationship between the central-tendency to H_0 distance and statistical test value is better observed using a scatter plot (Figure 11.23). Panel A shows that the t-value is unrelated to the mean distance to .5, which is trivial because that's how the data were generated. On the other hand, panel B shows that the Wilcoxon z closely follows the median distance from .5. You might have expected the opposite pattern: stronger significances as the distance increases.

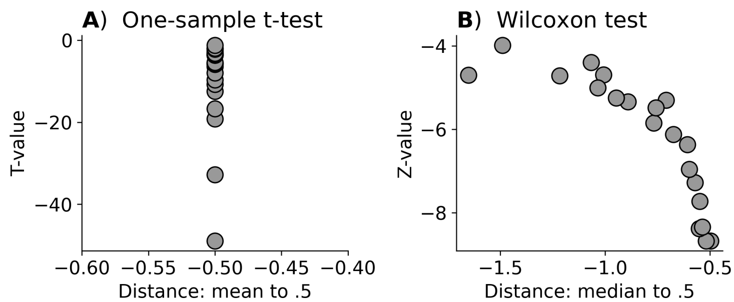

Figure 11.23: Additional visualization for Exercise 11.10, showing the relationship between central tendency distances and test statistic values.

The key insight is that the σ parameter affects not only the mean of a log-normal distribution, but also its dispersion. Indeed, for the smallest values of σ, nearly the entire distribution is left of the H_0 value (black lines in Figure 11.22A), whereas more of the distribution is to the right of the H_0 value for larger σ (lighter gray lines) even though the median itself is shifting to the left. I hope that makes sense. Statistics is not always straightforward, and working through confusing exercises like this one can help you gain a deeper understanding of how to investigate and think about data.

11.11. The purpose of this exercise is for you to explore the relationship between the p-value from the t-test and the two measures of effect size. A lot of the code for this exercise comes from the code for Exercise 11.6, so I recommend copying and pasting that code and modifying as appropriate.

Compute one-sample t-tests for a range of population means and sample sizes, but have the population means range from 0 to 2 in 71 steps, only compute one t-test per parameter pair (instead of 250 as in Exercise 11.6), and instead of storing the binary significance outcome, store the p-value, and also compute and store Cohen's d and R^2, according to Equations 11.10 and 11.13.

Plot the p-values by Cohen's d as in Figure 11.24A, and then plot Cohen's d by R^2 as in Figure 11.24b. You can see that there is a tight relationship between Cohen's d and R^2, although it is a non-linear relationship. So Cohen's d and R^2 are not identical, but they do provide very similar information.

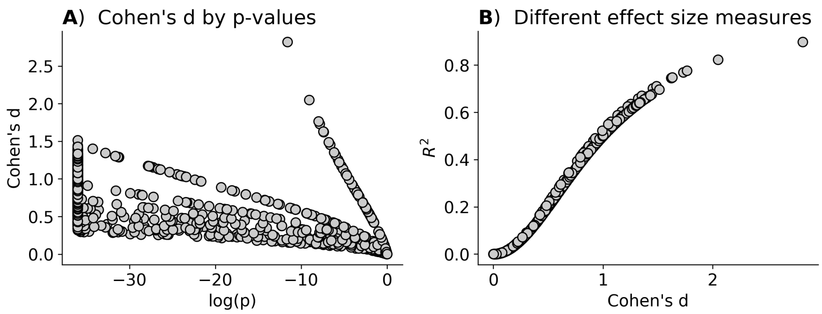

Figure 11.24: Visualization for Exercise 11.11.

On the other hand, there are relationships between the p-value and Cohen's d, but each relationship (that is, each "string" of dots) depends on sample size. The important take-home message from this exercise is that the same p-value can have very different effect sizes; and the same effect size can have very p-values. This is an illustration of how a p-value cannot be used to infer effect size. Indeed, the points in the lower-left of panel A show tiny p-values (i.e., extremely statistically significant) from very small effect sizes, which happens because of large sample sizes. This association between p-value and effect size will come up again in Chapter 15.

11.12. As you know, I like using simulated data to explore fundamental concepts in the t-test. But there's no substitute for real data, and so the goal of this and the next exercises will be to import a public dataset and apply an independent two-samples t-test.

The dataset is about predicting wine quality ratings[9]. The dataset contains 1599 observations and twelve features: eleven about the chemistry of the wine (acidity, sugar, pH, alcohol content, etc.) and one containing a subjective quality rating. The website for reading about the dataset, and downloading the data, is in this footnote[10] and linked in the online code. The goal of this exercise is to import and work with the data, and in the next exercise you will perform t-tests and corrections for multiple comparisons. The purpose is to determine which chemical properties of wine are significantly different between low- and high-quality rated wines.

I encourage you to perform these exercises using the `pandas` and `seaborn` libraries. If you understand the statistics but get stuck with the code, feel free to peek at my solutions to get the coding aspects.

Use `pandas` to import the data from the online csv file. Print out the dataframe to inspect the dataset columns and some of the rows. Then use the `describe()` method to examine some descriptive statistics of the data. Your screen should look like Figure 11.25.

	fixed acidity	citric acid	chlorides	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.270976	0.087467	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.194801	0.047065	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.000000	0.012000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.090000	0.070000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.260000	0.079000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.420000	0.090000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.000000	0.611000	1.003690	4.010000	2.000000	14.900000	8.000000

Figure 11.25: Descriptives for the dataframe used in Exercise 11.12.

Next, compute and print the number of unique data values for each column. This is important to check, because t-tests rely on means and standard deviations, which are only sensible data characteristics if there is sufficient variability. You should be able to print a report like this:

```
   fixed acidity has  96 unique values
volatile acidity has 143 unique values
     citric acid has  80 unique values
  residual sugar has  91 unique values
       chlorides has 153 unique values
```

[9]P. Cortez, A. Cerdeira, F. Almeida, T. Matos and J. Reis. *Modeling wine preferences by data mining from physicochemical properties*. In Decision Support Systems, Elsevier, 47(4):547-553, 2009.
[10]archive.ics.uci.edu/ml/datasets/Wine+Quality

```
 free sulfur dioxide has  60 unique values
total sulfur dioxide has 144 unique values
             density has 436 unique values
                  pH has  89 unique values
           sulphates has  96 unique values
             alcohol has  65 unique values
             quality has   6 unique values
```

Notice that the main IV, `quality`, has only six unique values. I'll get back to this later. Use Seaborn's `boxplot` method to visualize box plots of all columns. I don't show the Figure here, but it's in the online code.

You will see that the data have very different numerical ranges. Therefore, the next step is to z-score all columns except for the `quality` column. There is no built-in method in `pandas` to z-score, so you can either loop over the columns and transform the data using the formula for z-score, or you can use the `pandas apply` method using the `stats.zscore` function. You can create new variables or overwrite the existing variables. Confirm that the z-score transform has been successfully applied by inspecting the descriptive statistics and box plots (shown in the online code).

Are these data normally distributed? Test each column for normality. My results are shown below (I used only the Shapiro-Wilk test; you can also use the Omnibus test):

```
       fixed acidity: p<0.0000
    volatile acidity: p<0.0000
         citric acid: p<0.0000
      residual sugar: p<0.0000
           chlorides: p<0.0000
 free sulfur dioxide: p<0.0000
total sulfur dioxide: p<0.0000
             density: p<0.0000
                  pH: p<0.0000
           sulphates: p<0.0000
             alcohol: p<0.0000
```

Huh, so it appears that all variables are highly significantly non-normally distributed. Let's take a closer look. Use `seaborn`'s histogram function to visualize the distribution of all variables. I put them all into one figure, as you can see in Figure 11.26.

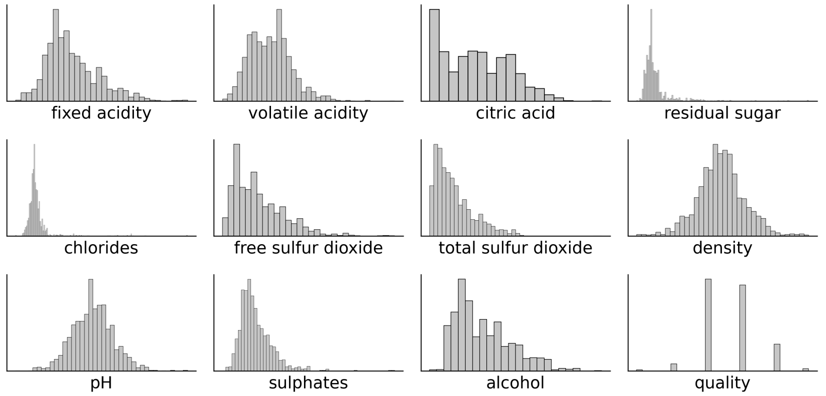

Figure 11.26: Visualization used in Exercise 11.12.

All variables look somewhat normally distributed in the sense of having one peak that tapers down on both sides, although several of the variables have some positive skew. This is a case where the highly significant Shapiro test results could be due to the large sample size (1599). In the next exercise, you will try both parametric and non-parametric tests to see whether the conclusions about the data are affected by the choice of analysis method.

Notice the distribution of the `quality` variable (lower-right in Figure 11.26). The values are nearly perfectly symmetrically distributed. You'll get a better sense of this by creating a histogram only of that variable (figure not shown here, but it's in the online code). That histogram suggests that we could binarize the quality ratings according to ratings 3-5 ("low quality") vs. 6-8 ("high quality"). Create a new column in the data for binarized quality as a boolean variable, with `False` corresponding to low-quality ratings and `True` corresponding to high-quality ratings.

Congrats on importing and inspecting the data. You are now ready for the analyses :)

11.13. Loop through all columns and compute an independent two-sample t-test to determine whether each feature is significantly different between low- and high-quality rated wines. Based on the distributions of the variables, you can assume unequal variances. Print the results as I have below. The p-values are uncorrected, the `*` indicates significance when using Bonferroni correction, and the `+` indicates significance when using FDR correction.

The `stats.ttest_ind` outputs the corrected *df*, but you can re-implement Equation 11.17 if you want additional practice at translating equations into code.

```
        fixed acidity: t(1596)=  3.86, p=0.0001, *+
     volatile acidity: t(1515)=-13.48, p=0.0000, *+
          citric acid: t(1593)=  6.48, p=0.0000, *+
       residual sugar: t(1575)= -0.09, p=0.9311,
            chlorides: t(1266)= -4.29, p=0.0000, *+
  free sulfur dioxide: t(1523)= -2.46, p=0.0141,  +
 total sulfur dioxide: t(1355)= -9.34, p=0.0000, *+
              density: t(1576)= -6.55, p=0.0000, *+
                   pH: t(1567)= -0.13, p=0.8962,
            sulphates: t(1495)=  8.85, p=0.0000, *+
              alcohol: t(1517)= 19.78, p=0.0000, *+
```

(This exercise is tricky. I have a few tips in the footnote if you need[11].)

Two final aspects to explore in this exercise: Use the Mann-Whitney U test to determine whether that impacts the significance of the results (not the p-value *per se*, but the conclusions drawn about the data); re-run the t-tests without z-transforming the data (before you implement this, think about whether you would expect the results to differ, and why).

Final comment for this exercise: recall the discussion about the dangers of discretization in Section **??** (from Chapter 3 on visualization). Here, the quality ratings are Gaussian distributed, and yet we binarized them according to the center value. This means that many data values *across bins* are closer to each other than data values *within bins*. That is not wrong per se, but does suggest that we might be ignoring meaningful nuances in the data.

[11]Some tips: Inside the for-loop over variables, extract the columns of the dataframe as separate variables. Use separate for-loops to compute the t-test vs. report the results, because FDR needs all p-values. Consider storing the results of the tests in a dictionary.

Index