Chapter 4
Probability and hypotheses

James Myers
2022/3/1 draft

1. Introduction

So far what we've been doing is **descriptive statistics** (描述統計學): tools for summarizing data. As you may recall, this is only one of the many jobs that statistics does. To do the others, we need to move beyond mere description and start looking at **inferential statistics** (推論統計學), which allows you to do a kind of magic: draw general conclusions from a finite amount of data.

The core of this magic is **probability** (機率、概率). This is another one of those very familiar concepts that becomes more complex (and hopefully more interesting) when you look at it more closely. In this chapter I'll first remind you about your own intuitions about probability, and then, using a linguistic version of a classic puzzle, show why relying on intuition alone can be dangerous; there's no escaping the need to learn a bit about the math behind probability. I'll then show how probability can help us compare our observed sample with our hypothesized populations, and illustrate this in a sociolinguistic example. In classical statistics (as opposed to Bayesian statistics), making inferences about a population from a sample is called **hypothesis testing**, the second part of this chapter's title. As we'll see, this term has a narrower technical meaning than it sounds like it does (which is partly why Bayesian statistics takes a different approach). I'll end the chapter by introducing your very first practical statistical tests, both of which are also fundamental to most of the rest of the tests we'll learn: the **binomial test** (for binary data) and the **one-sample *t* test** (for normally distributed data).

2. Probability

What does it mean when the morning weather report says there's a 30% chance of rain in the afternoon? Well, in traditional (non-Bayesian) statistics, this kind of probability claim is based on frequency (so another name for the traditional approach is **frequentist**): weather experts counted how many days have had weather like this morning (e.g., 10,000), then counted how many of those days had rain in the afternoon (e.g., 3,000), and finally divided the latter number by the former (3000/10000 = 0.3 = 30%).

See? Probability is very intuitive. Except that human intuitions about probability relate less to objective frequencies than to subjective feelings of confidence. After all, this afternoon hasn't happened yet; we can't actually count anything about it. Instead, what seems to happen

psychologically is that people start with a subjective belief of what feels probable, and then adjust this belief as they get more evidence; we don't literally count days when making a guess about the weather. As we'll see at the end of this book, this subjective approach can be formalized in terms of Bayesian statistics (see, e.g., Oaksford & Chater, 2009).

This conflict between objective frequencies and subjective beliefs can make probability surprisingly confusing. In this section, I'll illustrate this with a classic probability puzzle, and then show how to gain control over the mess by formalizing probability in a more mathematical way. The focus will be on the frequentist approach, since this is the approach adopted by most of the rest of this book (and most statistics books as well).

2.1 The three Martian genders

One of the most notorious examples of how probability intuitions can go awry is the so-called **Monty Hall problem** (蒙提霍爾問題), named after the host of an old American guessing game TV show (see, e.g., "Marilyn and the goats" in Stewart, 1997, and many other sources all over the Internet... though weirdly, pigeons may actually be better at at this puzzle than people: see Paulos, 2011). Let me tell you a new version just for you linguists, and show how to solve it with the help of logic, Excel, and R.

You're taking a Martian-for-humans class, and your teacher asks you for the gender of the word for "banana" (sure, Mars has bananas). There are three different genders for Martian nouns, A, B, C, but you have no idea which one goes with "banana", so you make a wild guess: "Um... A?" Your teacher doesn't change expression, and merely says: "Maybe, but it's definitely not C." Here's the situation in a diagram:

A ← Your wild guess for the right answer
B
C ← What your teacher say is the wrong answer, after hearing your guess

So should you stick to your original guess of A, or should you now change your guess to B? Maybe you think it doesn't make any difference. After all, your intuitions about probability tell you that your original wild guess has a 1/3 chance of being right, and the correct gender is not going to change no matter what you guess, so you may as well keep your first one.

Before going on to the hard part of this puzzle, let me say that the first part of this intuition is totally correct! Why? Because there are three (3) genders and you picked one (1) at **random** (隨意). We'll explore the philosophical notion of randomness later, but intuitively the idea is that there was no reason for you to pick any one gender over the others. Randomness is crucial not just to the logic of statistics, but also to the logic of research more generally. For example, if you run a psycholinguistic experiment, testing one word at a time to see if nouns are

processed differently from verbs, it would be a bad idea to present the items nonrandomly, such as putting all the nouns before all the verbs, since then you couldn't tell if any processing difference is really due to nouns versus verbs, and not due to the order (maybe people are more confused about what to do at the start of the experiment, or maybe they get more tired at the end of the experiment). That's why psycholinguistic experiments always present items in random order (unless there's some good reason not to, like if you really do want to study this order effect).

You can randomize the order of test items very easily in both Excel and R. In Excel, just list your items in one column, then put **=RAND()** in another column (in every cell in this column), and then sort both columns by the **random number** (亂數) column (note that they get recalculated after sorting). Try it! In R, use the **sample()** function to select a "sample" from your sample that's exactly the same size as your sample, without replacing any of the elements (the default setting for this function, as we'll see below). For example, if **items** is a character vector of items in some order, then the vector **sample(items)** has exactly the same items, except in random order. Try it!

Now let's return to the Martian gender puzzle. While you're right to think that you have a 1/3 chance of guessing the right gender on your first try, most people also have the intuition that this is still true even after the teacher says that the gender isn't C. Saying this can't magically change the rightness or wrongness of A versus B. If at the start the right answer had a 1/3 chance of being any of the three genders, then there must still be a 1/3 chance for A being right, and a 1/3 chance for B being right too, so why change your guess?

This intuition is wrong, though. Why? Because you *have* been given additional information, information that changes the original probabilities. Specifically, what we have now are the two probabilities that the right gender is A or B *given that* (assuming that) the gender is *not* C. Such a "given that" probability is called a **conditional probability** (條件機率), and human intuitions just don't handle them well at all.

To end your suspense, the correct answer is that after the teacher declares that the gender is not C, the probabilities about the real gender are now like this: 1/3 chance for A, but 2/3 chance for B. Since it's twice as probable that the correct gender is B, that should be your next guess.

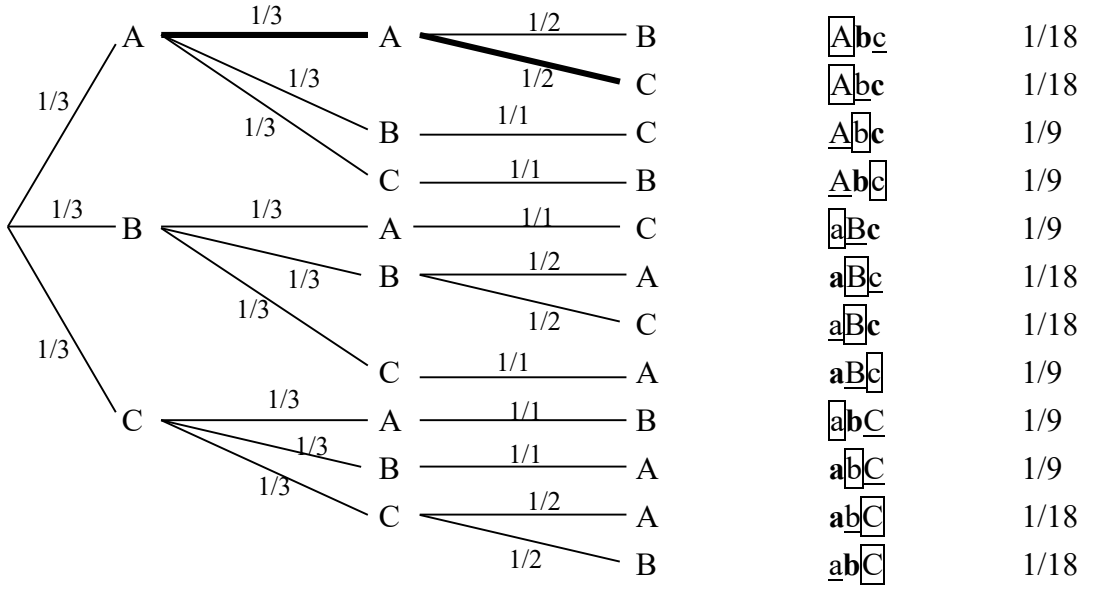
Let me explain why this is so, in three different ways (and even after all that, your intuitions may still rebel). To start with, basic logic says that the total probability that the gender is A, B, or C is 100%. Thus it's impossible for the probabilities for A and B to both be 1/3, since the probability for C is now 0, so adding up these three probabilities would only give you 1/3 + 1/3 + 0 = 2/3, not 3/3 (100%). By the same logic, when you make your first guess, because there is a 1/3 chance that the gender is A, there must be a 2/3 chance that the true gender is not A (i.e., that it is B or C). Now that you know that the gender is not C, "B and C" becomes "B and C but not C", which is just "B", but the total probability for this is still 2/3 (so that A = 1/3

plus B-or-C = 2/3 still adds up to 1). Thus, now that C has been removed, this probability of 2/3 now only applies to B.

Here's another way to put this same logical argument. Imagine another alien language where nouns have 100 genders, not just three. You guess that the gender for "banana" is gender #1, and your teacher says "Maybe, but its gender is definitely not #2 to #76 or #78 to #100!" Just like the Martian teacher, this one knows the real gender, and purposely mentions all of them except your guess of #1, plus #77. Now your intuitions are probably crystal clear: your first guess of gender #1, chosen randomly, still has a random 1/100 chance of being right, but gender #77 was chosen nonrandomly, and in fact obviously has a chance of 99/100, and so obviously you should guess that gender next!

Not obvious? All right, let's try this more systematically, examining all logical possibilities to see whether the gender is more likely to be A or B, assuming the teacher always play this mischievous trick (i.e., identifying a wrong answer after the student makes a guess). Study the network of probabilities below: three possible real genders (1/3 chance each), three possible first guesses (1/3 chance each), then one (1/1 chance) or two (1/2 chance) possible wrong answers revealed by the teacher (depending on whether your first guess was wrong or right), yielding twelve possible final situations, where the capital letter indicates the true gender, the boxed letter indicates your first choice, the bold letter indicates the teacher's revealed wrong answer, and the underline indicates the other remaining gender. The bold lines indicates the situation in our original story (first guess = A, revealed wrong answer = C). The probabilities for each of the twelve final situations are calculated by simply multiplying each of the probabilities on the path leading to it (e.g., for the first situation, 1/3 × 1/3 × 1/2 = 1/18). I hope this **multiplication rule** is intuitive (we'll formalize it later); for example, the probability of choosing the king of hearts (K♥) from a deck of 52 cards is obviously 1/52, and that's the same as multiplying the probabilities of choosing a king (1/13: A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K) and a heart (1/4: ♠, ♣, ♥, ♦) at the same time (1/13 × 1/4 = 1/52).

Now add up the probabilities for being right on your first guess (boxed capital letter: 1/18 + 1/18 + 1/18 + 1/18 + 1/18 + 1/18 = 6/18 = 1/3) and for being right if you change your choice to a different gender (underlined capital letters: 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 = 6/9 = 2/3): so your first guess gives you a 1/3 chance, while changing gives you a 2/3 chance. I hope this **addition rule** is intuitive too; for example, the probability of choosing a king (K) from a deck of 52 cards is 4/52 = 1/13, and the probability of choosing a queen (Q) is also 4/52 = 1/13, so the probability of choosing a king or a queen (which are non-overlapping cases) is 1/13 + 1/13 = 2/13 = 8/52 (and indeed, there are eight cards that are either K or Q).



We can make this slightly clearer if we use Excel to apply the multiplication and addition rules for us. Note how I calculate the total probability for each situation, and the total probabilities given the accuracy (1 vs. 0) of the first guess and changed guess. Copy/paste this table into Excel, fill in the "etc" cells by dragging down the formulas, and compare the probabilities in the two highlighted cells: 0.3333333 (1/3) vs. 0.6666667 (2/3).

	A	B	C	D	E	F	G	H	I	J
1	Real	Prob	Guess	Prob	Wrong	Prob	TotalProb	FirstAcc	Change	ChangeAcc
2	A	=1/3	A	=1/3	B	=1/2	=B2*D2*F2	=IF(A2=C2, G2,0)	C	=IF(A2=I2, G2,0)
3	A	=1/3	A	=1/3	C	=1/2	(etc)	(etc)	B	(etc)
4	A	=1/3	B	=1/3	C	=1/1	(etc)	(etc)	A	(etc)
5	A	=1/3	C	=1/3	B	=1/1	(etc)	(etc)	A	(etc)
6	B	=1/3	A	=1/3	C	=1/1	(etc)	(etc)	B	(etc)
7	B	=1/3	B	=1/3	A	=1/2	(etc)	(etc)	C	(etc)
8	B	=1/3	B	=1/3	C	=1/2	(etc)	(etc)	A	(etc)
9	B	=1/3	C	=1/3	A	=1/1	(etc)	(etc)	B	(etc)
10	C	=1/3	A	=1/3	B	=1/1	(etc)	(etc)	C	(etc)
11	C	=1/3	B	=1/3	A	=1/1	(etc)	(etc)	C	(etc)
12	C	=1/3	C	=1/3	A	=1/2	(etc)	(etc)	B	(etc)
13	C	=1/3	C	=1/3	B	=1/2	(etc)	(etc)	A	(etc)
14							Totals:	=SUM(H2: H13)		=SUM(J2: J13)

As our third and final attempt to understand what's going on, let's take a more "empirical" approach, and try to **simulate** the story using R. The following code randomly generates 10,000 scenarios, where "banana" has an equal chance of having each of the three genders, you have an equal chance of choosing any gender on your first guess, and the teacher always reveals a wrong answer (randomly chosen if your first guess is right). The only new functions are

sample(X, n), which randomly chooses **n** elements from the set (vector) **X**, and **setdiff(X, Y)**, which outputs the set (vector) of elements in **X** that are not in **Y**. Try the code and you'll see that your chances of finding the speaker are twice as good if you change your original guess.

```
genders = c("A","B","C")
first.guess.acc = numeric(10000) # To be filled with 1 = right vs. 0 = wrong
change.guess.acc = numeric(10000) # Ditto
for (i in 1:10000) {
real.gender = sample(genders,1) # Randomly choose one of the fixed set of genders
  first.guess = sample(genders,1) # Ditto
  # Teacher reveals wrong answer, randomly if necessary:
  non.gender= sample(setdiff(genders,c(real.gender,first.guess)),1)
  change.guess = setdiff(genders,c(first.guess,non.gender)) # Your only alternative now
  first.guess.acc[i] = 1*(real.gender== first.guess) # 1 if first guess is right
  change.guess.acc[i] = 1*(real.gender== change.guess) # 1 if changed guess is right
}
prob.first.guess = mean(first.guess.acc) # Mean of 0s and 1s gives the proportion...
prob.change.guess = mean(change.guess.acc) # ...because it divides the 1s by total
prob.first.guess # About 0.33 = 1/3
prob.change.guess # About 0.66 = 2/3
```

2.2 The math of probability

Let's look into probability a bit more formally now. The most basic idea is probability itself, of course. In the frequentist approach, probability P is just the proportion of times that some event E is true in a set of all possible events (technically, an **event** is a set of **outcomes**, where an outcome is the result of a single "experiment" or process). For example, if E = choosing the king of hearts (K♥) from a normal deck of cards, where there is only 1 such card and the whole deck has 52 cards, then P(E) = 1/52 = .01923077....

Since probability is a kind of proportion, it can never go below 0 (= can't happen) or above 1 (= definitely must happen). That's why it's so convenient that the area under the standard normal curve is 1, so portions of this area can represent probabilities. That's also why I have been leaving off the "0" before the "." in probabilities, since a proportion can never go above 1. In particular, APA style says that you should always write *p*-values like this: $p < .05$, not $p < 0.05$. (You have to fix this by hand; both Excel and R always put that "0." in there anyway.)

As we've already seen with the problem of the three Martian genders, probability starts getting tricky when you deal with more than one event at a time. For example, the probability that outcome A and/or outcome B occurs depends on whether the two outcomes can ever occur at the same time. If they can't, the probability that either one or the other occurs is just the sum of the two probabilities, as in the simple formula below.

Addition rule (simple): P(A or B) = P(A) + P(B) {if A and B never cooccur}

However, if A and B can cooccur, then examples of A will also contain instances of A&B, and so will B, so the simple addition rule will count these instances twice. So the more general formula looks like the second one below.

Addition rule (general): P(A or B) = P(A) + P(B) - P(A&B)

The general form of the addition rule raises the question of how to calculate P(A&B). This is only easy to calculate if A and B are **independent** (獨立事件). Informally, independence means that P(A) tells you nothing about P(B) and vice versa. If this is true, then you can use the following formula for the multiplication rule (which we also used in solving the three-genders problem):

Multiplication rule (simple): P(A&B) = P(A) × P(B) {if A and B are independent}

But what does "independent" really mean? It turns out that there's a deep philosophical problem hidden here, since independence is actually *defined* in terms of the simple multiplication rule! That is, if A and B obey that formula, then they are independent. Circularity city!

Another way to think about it is in terms of **conditional probability** (條件機率), which is what confused us so much in the three-genders problem. Conditional probability is the probability of one event assuming another event: "P(A|B)" is read as "the probability of A, given B." For example, what is the probability of choosing a heart assuming that the card is a king? It's 1/4, right? That's the same as defining the main set as "king" (B) and the event E as "heart" (A). This works because a deck of cards is designed so that all the properties are independent of each other; choosing a king gives you no information about the color or anything else.

Conditional probability: P(A|B) = P(A&B)/P(B)

Notice that by moving stuff around in an algebraic fashion, the conditional probability formula implies the following general version of the multiplication rule. Note that if A and B are independent, P(A|B) = P(A) (since the probability of A stays the same whether or not B is also involved), and the general formula turns into the simple one we saw above.

Multiplication rule (general): P(A&B) = P(A|B) × P(B)

Even though the math of probability is undeniable, linking this math to any particular real-world situation depends, of course, on the real-world situation. For example, it was crucial in the three-genders story that if your first guess was right, the teacher had to reveal one of the other wrong answers *at random*. If instead the teacher always revealed the same wrong answer (e.g., if you correctly guessed A, it would always be B that was revealed, never C), then all of the above calculations would change. Or to take a more familiar case, which ordered set of cards is less probable: J♥, Q♥, K♥, or J♥, 2♦, 10♦? The answer is: it depends! Without any prior expectations, both are equally probable, since any combination of three cards is just as probable as any other combination of three cards, namely 1/(52×51×50) = 1/132,600 (there are 52 choices for the first card, but only 51 choices for the second since you already chose one, and so on for the third). But if we can explicitly specify ahead of time the few card sets that we will count as "interesting" (maybe only 100 three-card ordered sets seem "interesting", so 132,600-100 = 132,500 are "boring"), then the first random series becomes much less probable than the latter: 100/132,600 < 132,500/132,600.

2.3 Some applications of probability

The math of probability can be useful to linguists all by itself, even if we forget about its crucial role in inferential statistics (and go beyond these silly puzzles). For example, suppose you want to know if somebody can choose the classifier for a noun more accurately than what you'd expect by chance, where you define statistical significance with $p < .05$. What's the minimum number of nouns that you should test?

If $p = .05$, that would be mean that getting the right answer purely by chance would happen with a probability of 1/20. So you should test enough nouns so that there are at least 20 possible events in the experiment (where an "event" is a series of binary "outcomes" of choosing the correct vs. wrong classifier). If you only test four nouns, that's not enough: the number of possible patterns of "correct" vs. "incorrect" would only be $2^4 = 16$ (this follows from the simple multiplication rule, assuming that each classifier choice is independent of the others). By chance, any one of these 16 possible events is expected to happen with a probability around $p = 1/16 = .0625 > .05$. So no matter what happens in your real experiment, you can't tell if the result is statistically significant. Thus you need to test at least five nouns: $1/2^5 = 1/32 = .03125$, so if all nouns are given the right classifier, the probability of this happening by chance is less than the magical .05 ("magical" being a joke here: this cut-off value is merely an arbitrary convention).

Linguistic processing and grammar may themselves be intrinsically probabilistic. Sociolinguist William Labov (e.g., Labov, 1994) has argued that this means that competence is actually probabilistic, not categorical. Thus his theory, as well as more recent versions (e.g., Anttila, 1997; Boersma & Hayes, 2001; Sorace & Keller, 2005), focus not on either-or outputs,

but on the probabilities of getting one output versus another, and how these probabilities shift depending on the inputs.

Probabilistic variation can also undermine claims about grammatical patterns. For example, Myers (2006) argues that the *chance* probability of getting the famous Southern Min tone circle is so high, just by randomly combing tones, that there's no need for phonologists to try to explain it: there is probably no real pattern there at all.

This is a general probability issue that you should keep in mind. After all, think carefully about what that magical $p < .05$ cut-off value really means: it means that even if there is *no* real pattern, running the same experiment on the pattern will result in approximately 5% of the experiments being "significant" by pure luck! And it gets worse: as has often been pointed out (e.g., Woods et al., 1986, p. 128; Ioannidis, 2005), the probability that at least one such experiment will be "significant" increases the more times the experiment is run. Thus if a totally false hypothesis is nevertheless popular for other reasons (e.g., it sounds plausible), then many people will run experiments on it, and about 5% will get "significant" results by chance alone, and then they'll be able to publish their results (while the poor researchers who run the 95% experiments that fail will just give up), inspiring further useless research on the totally false hypothesis. Thus the probability of a published result being significant can relate more to the *popularity* of a topic than to truth! (There's a link on the statistics resources page to a little animated video explaining this logic: "Why most published scientific research is probably false"). This is the first of many clues that "statistically significant" doesn't necessarily mean "significant" in the ordinary sense.

I should also give a brief note on the terms **probability** (機率) and **likelihood** (似然). In ordinary language (as in most of my discussions in this book), they mean exactly the same thing. As we'll see in later chapters, however, there is a subtle technical difference in terms of conditional probability. Probability describes outcomes given the input assumptions (e.g., the parameters of a statistical model), which is what we've been talking about so far. By contrast, likelihood, in its technical sense, describes the inverse, namely the parameters given the outcomes (i.e., how likely a statistical model is, given the observed data).

3. Sampling

Probability comes into classical (frequentist) statistics because inferential statistics tests hypotheses about an abstract (maybe infinite) **population** (總體、母體) based on an observed **sample** (樣本). The most intuitive way to do this would be to make inferences about the population *from* the sample *to* the population. Sadly, this is *not* what is done in traditional statistics; instead you make a mathematical model of a chance population that assumes that there is *no* pattern, then you imagine selecting an infinite number of samples from this no-

pattern population, creating a distribution of samples, and checking to see if your actual sample is an outlier in this distribution.

Confused? I hope so, or else I'd feel bad about being confused myself. According to that statistician Gelman (2005), this concept is probably the most confusing idea in all of statistics, so we'll need to explain it repeatedly, from many different angles, again and again through this whole book.

In this section, I'll start by looking at a common sort of linguistic sampling problem: making inferences about linguistic productivity from a finite set of corpus examples. Then we'll formalize the logic, linking it back into our favorite distribution (the normal distribution, remember?), with the help of an amazing theorem (more proof that the normal distribution deserves its huge role in statistics).

3.1 Which word form is more common?

Having mastered Martian, you fly to Mars to conduct fieldwork on the last speaker of another Martian language, Extinctish, and collect a small corpus of his/her speech. While examining your corpus, you notice that Extinctish has two different word forms for "banana", namely [banana] and [ananab]. As far as you can tell, this is a purely **sociolinguistic variable**, which, as in the famous definition of Fasold (1990, pp. 223-4), represents "alternative ways of saying the same thing". In this case, there doesn't seem to be any social effect of the variation. But you still wonder which of the two forms is preferred in this language: [banana] or [ananab].

Well, in your corpus, [banana] appears 23 times and [ananab] appears 37 times. From this you might be tempted to say that [ananab] is preferred, but let's think it through first. After all, this Martian was probably talking about bananas for many years before you ever came to Mars, and probably talks about them even when you're not around (why not? bananas are delicious), and for all you know, she/he may keep talking about them long after you fly back home (Martians live a long time). Your observations are merely a finite (and relatively small) *sample*, but what you really care about is the *population* of all references to bananas. So maybe that 23-versus-37 difference is just the result of chance, and in the full population the ratio is more like one-to-one: 50% [banana] and 50% [ananab].

The hypothesis that the population shows no pattern is called the **null hypothesis** (虛無假說). This is the opposite of the **alternative hypothesis** (對立假說), in a strictly mathematical sense: if the probability of the null hypothesis is p (a number between 0 and 1), then the probability of the alternative hypothesis is $1-p$ (i.e., these are the only two logical possibilities so the total probability adds up to 1: only one or other other must be true). Usually the alternative hypothesis is what you predict if your **research hypothesis** is true, since usually scientific claims predict that there *is* a pattern rather than that there is *no* pattern. (On rare occasions we *do* want to predict that there is no pattern, which creates a situation that

frequentist statistics can't handle, so we would have to move over to Bayesian statistics, as we'll see in the final chapter.) So in this ordinary type of statistical hypothesis testing, you are hoping that you will be able to **reject** (falsify) the null hypothesis, and that will automatically make the alternative hypothesis the "winner".

To get this process started, then, we need a mathematical model of this no-pattern null hypothesis situation, where choosing [banana] or [ananab] is just like flipping a coin. What might this model look like...? Hm, let's try to remember... remember all the way back to the previous chapter... where we learned about something called... what was it again... ah yes! The **binomial distribution** (二項分配)! This represents the probability of getting a particular number of "heads" (H: 正面) versus a particular number of "tails" (T: 反面) by chance.

Thus if you flip a coin three times, there are $2\times2\times2 = 2^3 = 8$ possible events: HHH, **HHT**, **HTH**, HTT, **THH**, THT, TTH, TTT. How many have exactly two Hs? Three, so the probability is 3/8 = .375. How many have *at least* two Hs? Four, so the probability is 4/8= .5. Similarly, there are $2^4 = 16$ ways to flip a coin four times: HHHH, HHHT, HHTH, **HHTT**, HTHH, **HTHT**, **HTTH**, HTTT, THHH, **THHT**, **THTH**, THTT, **TTHH**, TTHT, TTTH, TTTT. So there are five events with exactly two Hs, so $p = 6/16$, and 11 with at least two Hs ($p = 11/16$).

And so on, so a computer would be helpful. Fortunately, the binomial formula is built into Excel's **=BINOMDIST()** function and R's **dbinom()** function (for density = height of the binomial curve) and **pbinom()** function (for probability = proportional area under the binomial curve). Note 1: As is often the case, the "improved" version of Excel's function has a dot inside: **=BINOM.DIST()**. Note 2: By default, Excel's function acts like R's **dbinom()**; to make it act like **pbinom()**, you need to set its "cumulative" argument to TRUE. Note 3: To understand the **pbinom()** results, it helps to look at the plot, and remember that R's distribution area functions always give the area to the *left* (because they're cumulative, accumulating values from lowest to highest).

```
p0 = dbinom(x=0,size=3,prob=0.5) # Probability of exactly 0 H in 3 coin flips
p1 = dbinom(x=1,size=3,prob=0.5) # Probability of exactly 1 H in 3 coin flips
p2 = dbinom(x=2,size=3,prob=0.5) # Probability of exactly 2 Hs in 3 coin flips
p3 = dbinom(x=3,size=3,prob=0.5) # Probability of exactly 3 Hs in 3 coin flips
plot((0:3),dbinom((0:3),size=3,prob=0.5),ylim=c(0,0.5)) # The four probabilities
p0 + p1 + p2 # Probabiilty of at most 2 Hs in 3 coin flips (0, 1, or 2)
p2 + p3 # Probability of at least 2 Hs in 3 coin flips (2 or 3)
pbinom(q=2,size=3,prob=0.5) # Probability of at most 2 Hs in 3 coin flips (left side!)
pbinom(q=(3-2),size=3,prob=0.5) # Probability of at least 2 Hs in 3 coin flips
```

OK, returning to bananas, here we have a total of 23+37=60 references. What is the probability that a random selection of [banana] versus [ananab] would give a difference in their counts as large as what we actually observed? Or better, a difference *at least* that large, since of course we'd be even more impressed if the contrast had been 10 versus 50 or even 0 versus 60. In other words, how many ways can you get 0 vs. 60 or 1 vs. 59 or 2 vs. 58 or ... down to

23 vs. 37, out of all of the possible ways of randomly choosing between [banana] and [ananab] 60 times?

```
pbinom(q=23, size=60, prob=0.5) # Probability of at most 23 Hs in 60 coin flips
[1] 0.04623049
```

As you can see, the answer is .04623049 (removing that initial zero). That's less than our "magic" cut-off of .05, meaning that getting a difference at least as big as we observe, by chance, will happen less than 1 out of 20 random tries of this 60-outcome experiment. By convention, maybe we want to say, then, that our observed result is too unlikely to have happened by chance (though of course, it might still be due to chance anyway: even rare accidents can happen).

Congratulations! You have just performed your very first bit of inferential statistics. This test is called, naturally, the **binomial test**. We'll add a bit more complexity to it shortly, but as you can see, the basic logic is based on notions that we've already discussed, in particular the connection between distribution area and probability.

The binomial test is a kind of **exact test**: we can calculate the chance probabilities exactly, since we know that the binomial distribution suits our situation perfectly, and the precise probabilities depend only on our total number of observations, and the number of outcomes of one type versus the other type. R even calls it an exact test if you use the special-purpose **binom.test()** function. I'll explain some of the other terminology later, but for now just look for the *p* value, which is the same as we calculated above.

```
binom.test(23, 60, 0.5, alternative="less") # The "less" gives you the area to the left

        Exact binomial test

data:  23 and 60
number of successes = 23, number of trials = 60, p-value = 0.04623
alternative hypothesis: true probability of success is less than 0.5
95 percent confidence interval:
 0.0000000 0.4975667
sample estimates:
probability of success
             0.3833333
```

Unfortunately, for most realistic situations, exact tests aren't practical; calculating exact probabilities can get very complicated. Fortunately, we can get results that are almost as good by building on the sample/population logic. Let's see how this would work for our simple case (where it's not really necessary), so we can apply it again in more complicated cases (where it is necessary).

Remember that our null population here is some imaginary (maybe infinite) set of all references to bananas, maybe not even just by this speaker, but by any imaginable speaker of Extinctish. For computational convenience, let's say the population size *n* is some large but finite number, say 1,000,000. In this population, [ananab] is used *a* times and [banana] is used *b* times (note my clever choice of abbreviations!), where $a + b = n$. According to the null hypothesis (which we want to try to falsify), $a = b$. Our actual sample size is $n = 60$, and our observed counts are $a = 37$ vs. $b = 23$. What we want to know is: Is our sample an outlier in the set of all possible samples of 60 banana references from this null-hypothesis population?

Let's simulate this situation. The R code below takes a null hypothesis population of 1,000,000 banana references, where $a = b = 500{,}000$ (which obeys the null hypothesis), and randomly selects samples of 60 items each (with **replacement**, so we could sample multiple copies of "the same" *a* or *b*, since that's what most statistical tests assume), producing many samples (10,000 of them), then counting the number of *a* vs. *b* in each sample. Finally, it computes the proportion of results where *a* is at least 37 (equivalently, where *b* is 23 or fewer): that's the probability of getting results (**hits**) at least as impressive as we actually got, by chance alone.

```
null.population = c(rep("a",50000),rep("b",50000))
hit.count = 0 # Will count how often there are at least 37 a's

for (i in 1:10000) {
  our.sample = sample(null.population,60,replace=T) # Sampling with replacement
  a.count = sum(our.sample == "a")
  if (a.count >= 37) { # if a.count is greater or equal to 37
    hit.count = hit.count+1
  }
}
hit.count/10000 # Proportion of hits
```

What did you get? Due to the random sampling, the result will be a bit different every time, but when I ran the code, I got .0476, which is quite close the "exact" probability of .04623049. Is that amazing or what?

3.2 The math of sampling

It's important to remember that samples and populations are sets of *numbers* (**data points** or **observations**). Recognizing that statistics deal with sets of numbers is particularly important in language experiments, where the results usually have two sources of random variation: from the participants and from the test items. The traditional response to this situation is to do two separate analyses, "by participants" (older term: "by subjects") and "by items." (Later we'll learn how to analyze all of the data together, but it will take us a while to get to that point.)

For example, imagine that you run an experiment comparing reaction times (RTs) for nouns versus verbs, using a **within-participants design** (i.e., every participant gets both nouns and verbs, as opposed to a **between-participants design**, where half the participants gets only nouns and the other half gets only verbs). Let's say you test 20 nouns, 20 verbs, and 30 participants. This would give you 1,200 ([20+20]×30) RTs.

For the by-participant analysis, you'd first average the noun RTs and the verb RTs, so each participant would only have two numbers. Then the two samples would be 30 averaged noun RTs and 30 averaged verb RTs. For the by-item analysis, you'd first average all the participants' RTs for each noun, and likewise for each verb, so the two samples would be 20 averaged noun RTs and 20 averaged verb RTs. Putting this together, you get something like Table 1 below.

Note that the by-participant and by-item analyses involve different samples and populations, so it is quite common for the by-item analysis to give different results from the by-participant analysis. In fact, basic algebra implies that if there is missing data (NA = not available), even the by-participant and by-item means may be different. Thus in Table 1, the overall by-participant means for nouns and verbs are 250 ms and 300 ms, respectively, and so are the overall by-item means for nouns and verbs (try it!). By contrast, replace one RT with NA, as in Table 2, and the overall by-participant mean for verbs is now 325 ms, even though the overall by-item means for verbs is still 300 ms.

Table 1. Mean RTs in full data set

	Noun 1	Noun 2	Noun (subj means)		Verb 1	Verb 2	Verbs (subj means)	
Subject 1	100	300	200	250	200	300	250	**300**
Subject 2	200	400	300		400	300	350	
Nouns (item means)	150	350	Verb (item means)		300	300		
	250				**300**			

Table 2. Mean RTs if a data point is missing

	Noun 1	Noun 2	Noun (subj means)		Verb 1	Verb 2	Verb (subj means)	
Subject 1	100	300	200	250	200	300	250	**325**
Subject 2	200	400	300		400	**NA**	400	
Noun (item means)	150	350	Verb (item means)		300	300		
	250				**300**			

Another crucial issue to keep in mind is that if you want to generalize from your sample, it must not be **biased** (偏頗). In fact, as we have seen, the math of probability and inferential statistics assumes that the sample is selected at random, with values (or groups of values) that

are either totally independent of each other, or related in known ways so that their effects can be extracted from the statistical model.

Intuitively, "randomness" means "no pattern", so one way to define it is as the opposite of a pattern. One way to define "pattern" is a description of a set of observations that is somehow simpler or more elegant than a mere listing of the observations themselves (formalized as **Kolmogorov complexity** 柯氏複雜性). In other words, a set of numbers is fully random if the only way to describe it is to just list all the numbers. For example, the non-random sequence (1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0) could be described much more elegantly in R as **c(rep(1,8), rep(0,8))**. By contrast, the randomly generated sequence (1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0) cannot be compressed so efficiently.

I find this way of understanding randomness kind of fascinating, since it reveals that randomness is fundamentally a cognitive notion: since there's no way to be sure that some future genius will not be able to find a pattern in a set that today seems to be entirely random, there's no way to be sure when we really have a random set.

Because brains evolved to detect patterns (it's safer to hide whenever you think you see a tiger, even if there is no tiger at all), it's basically impossible to generate random numbers by hand. For example, here's my attempt to type a random series of 0s and 1s, where 0 was typed with my right hand and 1 with my left hand (as they're arranged on the keyboard):

1101010110010101100011010101001010101010001101000110101101010011000110111001101000011101010011001001101011010000

If each digit is independent of the others, the simple multiplication rule says that there should be about equal numbers of the digit pairs 00, 01, 10, 11, which here would be about 30 (=120/4) each. But look at the frequency table below showing the actual counts:[1]

Type	00	01	10	11
Count	21	39	40	19

Obviously I was strongly biased to alternate my two hands, presumably because there is a cognitive illusion that tells us, falsely, that randomness should not produce too many clumps of 0s or 1s. But randomness does not mean "evenly spread out"; it means "no pattern at all", and there's definitely a pattern here.

So don't rely on your intuitions when randomizing things. The standard solution is to use a **random-number generator**, like the **=RAND()** cell function in Excel or **runif()** and **rnorm()** and other random distribution functions in R (see also Random Number Generation [亂數產

[1] I computed this using a trick from http://stackoverflow.com/questions/35563375/count-number-of-occurrences-when-string-contains-substring. The total doesn't add up to 120 because the counts can overlap (e.g., "110" contains both "11" and "10").

生器] in Excel's Anaysis Toolpak [分析工具箱]). Since these functions generate numbers through a programmed algorithm, they do not generate truly random numbers (in R, type **?Random** to learn a bit more about how the algorithms work). For most practical jobs, however, the algorithm is designed to generate output that is so variable that it is essentially unpredictable unless you know the initial input value used by the algorithm (called the **seed**). By default, Excel's and R's random number generators use your computer's clock as the seed, making the results different every time. In R, you can also set the seed manually with **set.seed()**. So if you enter **set.seed(1)**, then **runif(1)**, you will always get the result 0.2655087. This trick is useful if you need to replicate somebody else's "random" example (as we will occasionally have to do in this book).

For practical reasons, sampling in language studies is usually done nonrandomly, but this is not considered to be a problem if there is no known bias. Experimental participants are generally selected arbitrarily within a convenient population (often, university students), though it's not obvious that this is representative of any larger population (e.g., all Chinese speakers, or even all human beings). If you feel bad about this, it helps to remember the cognitive underpinnings of the notion of randomness: if you (and your potential critics) cannot see any bias in your sample, then that may be all we can ask for.

If your study depends crucially on the make-up of your speaker group, then sampling must be done much more carefully. For example, if you want to know what the "typical Taiwanese person" thinks about some linguistic issue, you'll have to leave the university and sample people from a representative range of socioeconomic groups, making sure that the number of people you ask in each group is proportional to the size of each group, a method called **stratified sampling** (分層抽樣); a classic guidebook is Kish (1965).

In language experiments, you also have to sample across the test items. If you're testing sentences, make them as varied as possible, so your sample will be more representative of all sentences of the tested types (it wouldn't make sense to choose a "random sample" of sentences, given that there is no agreement among linguists about what the population of all sentences in a language should look like anyway). If you're testing words, however, it's basically impossible to choose an unbiased sample. For example, if you sample words randomly from a dictionary, most will be very low frequency, and if you sample words randomly from a corpus, most will be very high frequency (both due to Zipf's law). Sometimes there are so few words that fit your experimental criteria (e.g., monosyllabic animal names) that it may be possible to test the entire population. Again, just try to choose as unbiased a sample as possible, even if it's not truly random.

Typological linguists also face the challenge of sampling languages. Note that the relevant population here is not the set of all *existing* languages, but the set of all *possible* languages, since the goal of typology is to understand the human language capacity in general, not the specific history of actual languages. The problem is that the set of existing languages is highly

biased, due to borrowing and historical relationships across languages and non-linguistic accidents favoring certain languages (e.g., English or Mandarin). Strategies for overcoming such biases when selecting typological samples are discussed in Cysouw (2005).

3.3 Samples as potential outliers

As noted above, we are usually not lucky enough to be able to compute probabilities using exact tests like the binomial test. In particular, the binomial test only works for independent sets of binary outcomes, but as I emphasized in earlier chapters, many real-life statistical hypotheses involve the means of (mostly) normally distributed continuous variables, like phonetic measures or reaction times. Fortunately we can deal with this kind of data using a clever trick (unfortunately, it is the very trick that Gelman, 2005, says always confuses his statistics students).

The trick works like this. Using the logic of traditional statistics, we want to know the probability *p* of choosing a sample at least as exciting as ours from the boring null hypothesis population (so we hope this *p* is low). The key to computing this probability is to see that if the null hypothesis is true, our sample is not merely a subset of the null hypothesis population, but is itself also an *element* in a more abstract kind of set: the *set of all possible samples* (of the same size as our actual sample) from the null hypothesis population.

For example, say we're studying accuracy rates in a child acquisition study, so we mark a correct pronunciation for some word as 1 and an incorrect pronunciation as 0. Now, these measurements are on a binary scale, so the most appropriate test would be one based on the binomial distribution, like the binomial test. But darn it, it's my textbook, and I can break the rules if I want to teach you something, and what I want to teach you right now is the logic of doing the statistics of means and the normal distribution. Sets of 0s and 1s are easy to understand, and because the binomial distribution is related to the normal distribution (as we saw in chapter 2), and because proportions of 0s and 1s can be computed using means (think about it!), it's actually quite common for real researchers to analyze accuracy rates using **parametric** tests (which, as you remember from chapter 3, are tests based on means and standard deviations of mostly normally distributed data).

All right, so we run a tiny study on child pronunciation accuracy, and get the following tiny sample of accuracy scores: {0, 0, 0, 1, 1}. The accuracy rate here is the same as the sample mean *M*:

```
samp = c(0,0,0,1,1)
samp.mean= mean(samp) # Same as the accuracy rate of 2/5
```

The result is .4 (remember the rule about not writing the first 0 for proportions?). So the question is: is this lower than we would expect by chance? By chance, the accuracy rate should

be around .5. Yes, we could (in fact *should*) do a binomial test, to compute the exact *p* value, which is .5, way higher than .05:

```
pbinom(q=2, size=5, prob=0.5) # At most 2 accurate in 5 random pronunciations
[1] 0.5
```

OK, now that we know the correct answer, let's see if we can get it again using the trick for means and normal distributions that I promised to teach you.

To explain the trick, we need to be explicit about what our "no pattern" population looks like. To keep things simple, let's say the population is also tiny: {0, 0, 0, 0, 0, 1, 1, 1, 1, 1}, so the population size $N = 10$, and the population mean (μ) is **mean(c(rep(0,5),rep(1,5)))** = 0.5.

Now comes the trick. If the null hypothesis is right, our observed *sample* is actually also an *element* in the set of all possible five-element samples from this boring population. In classic mathematical fashion, this is the first step in a series of ever more abstract steps.

The set of all five-element samples from our ten-element population contains exactly 252 members. How do I know this? I computed it in Excel using the function **=COMBIN(10,5)**, and double-checked it in R using **choose(10,5)**. Exactly one of these samples will be {0, 0, 0, 0, 0}, because there's only way to sample from {0, 0, 0, 0, 0, 1, 1, 1, 1, 1} and have all zeroes, but there will be ten samples that look like {0, 0, 0, 0, 1}, because there are five ways to choose four 0s from the five 0s in the population (because the population has five different 0s that you can decide *not* to choose each time), and there are five ways to choose one 1 (because there are five 1s in the population), and these two choices are independent and non-overlapping, so by the addition rule of probability, you get 5 + 5 = 10 possible sets of {0, 0, 0, 0, 1}. And likewise for all other samples.

Let's let R compute this for us:

```
choose(10,5)
[1] 252
```

R can even show us what this set of all possible samples looks like, using the function **combn()**, which generates all of the samples that we counted with **choose()**, putting them all into a **matrix** (矩陣: rectangle of numbers) with each subset arranged in a column. Try it out:

```
pop = c(rep(0,5),rep(1,5)) # Our null population
pop # This is what it looks like
all.samps = combn(pop,5) # All 252 5-element samples from pop, each in a column
all.samps # Many samples look the same because "different" 0s and 1s were selected
```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]	[,8]	[,9]	[,10]	[,11]	[,12]	[,13]
[1,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[2,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[3,]	0	0	0	0	0	0	0	0	0	0	0	0	0
[4,]	0	0	0	0	0	0	0	0	0	0	0	1	1
[5,]	0	1	1	1	1	1	1	1	1	1	1	1	1

...

The point of reimagining of our sample as an element in a set of samples from the same boring "chance" population is that we can now compute whether our sample is an **outlier** in this boring distribution. If our sample is an outlier, then it's not boring. Specifically, we want to know if our sample mean (accuracy rate, in our case) is "sufficiently" far away from the boring population mean (chance accuracy of 50%, in our case) for us to claim that our result is "significantly" different from chance.

How can we measure how close our sample mean is to the population mean? The next insight (or the next step up on the mathematical abstraction ladder) is to realize that each of those 252 samples has its own mean too, so we can create a set of all of these means: this very important thing is called the **distribution of sample means**. Since, according to the null hypothesis, our actual sample mean is just a point in this distribution, we can figure out how unusual it is in terms of some kind of standard distance measure, similar to (in fact, almost exactly like) how we used the standard deviation to test for outliers in chapter 3.

We could compute the distribution of sample means by looping through the matrix of samples in **all.samps**, applying **mean()** to each column, but R has lots of special functions for avoiding loops, and **apply()** is one of them. This function applies any other one-vector function (like **mean()**) to a matrix either by row or by column (here, by column, using the code 2) to each of the samples. So we can see the mean 0 for the sample {0, 0, 0, 0, 0}, the mean 0.2 for the ten samples of the form {0, 0, 0, 0, 1}, and so on.

```
dist.samp.means = apply(all.samps,2,mean) # Means of columns (1=row, 2=column)
dist.samp.means # Take a look!
```

```
  [1] 0.0 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.2 0.4 0.4 0.4 0.4 0.4 0.4 0.4
 [19] 0.4 0.4 0.4 0.2 0.2 0.2 0.2 0.2 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4 0.4
...
```

Let's plot what we've done so far. The following code produces the three density plots (smoothed histograms) in Figure 1, namely for the null population (pop), for all 252 samples (overlapped), and for the distribution of their sample means.

```
par(mfrow=c(1,3)) # Put three plots in a row
plot(density(pop),main="Population")
```

```
# Plot first column (sample)
plot(density(all.samps[,1]),main="Samples", # Plot first sample (column)
  xlim=c(-2,3), ylim=c(0,1.15),) # x-axis and y-axis adjust to fit other samples
for (i in 2:252) {
  lines(density(all.samps[,i])) # Add all of the other columns (samples) to this plot
}
plot(density(dist.samp.means),main="Sample means")
```

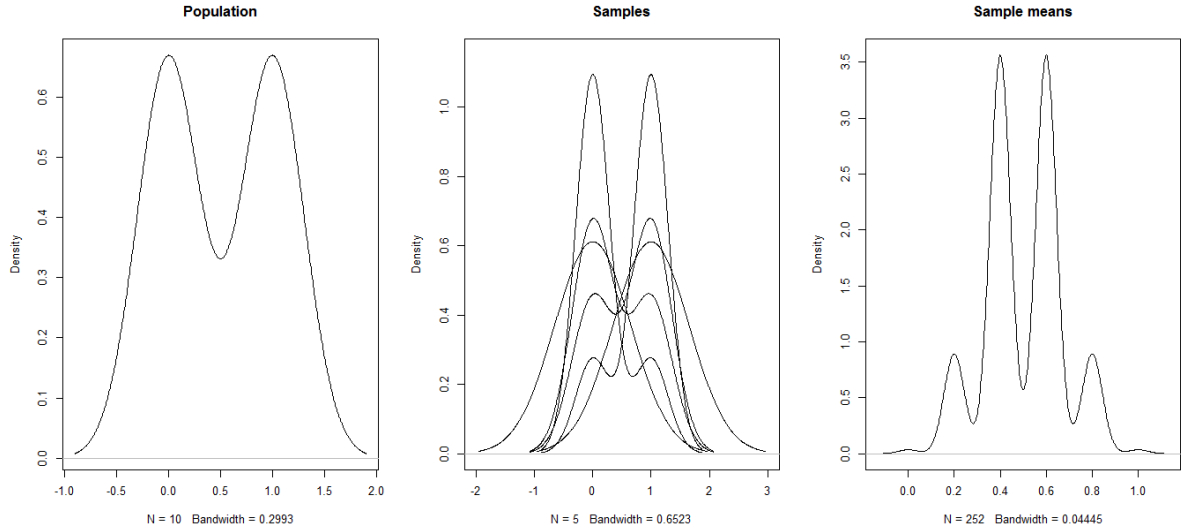


Figure 1. Population, samples, and sample means

Now that we have a distribution of sample means, we can calculate a *p* value indicating the proportion of samples taken from the "boring" population that have a mean at least as "interesting" (here, as low) as our sample's mean of .4. The result is $p = .5$, exactly the same as we computed using the binomial test.

```
sum(dist.samp.means <= samp.mean)/length(dist.samp.means)
[1] 0.5
```

Before moving on, let's try this again with a different sample, {0, 0, 0, 0, 1}, from the same population, and compute the *p* value both ways:

```
# Binomial test
pbinom(1,5,0.5)
[1] 0.1875
```

```
# Via distribution of sample means
samp2 = c(0,0,0,0,1)
samp2.mean = mean(samp2)
sum(dist.samp.means <= samp2.mean)/length(dist.samp.means)
[1] 0.1031746
```

The results are slightly different: what we're doing now is not exactly the same as the binomial test, since our population is not just finite but pretty small, and also it's based on the means rather than the raw counts. The difference in *p* values relates to different background assumptions within which we're counting our "events", kind of like the different probabilities we calculated above for selecting the jack, queen, and king of hearts in order from a deck of cards.

In this ridiculously unrealistic case, our population was so small that we could actually list all samples. We can also estimate a *p* value by taking a **simulation** or **resampling** approach, by randomly selecting samples; this method is always possible even if the population is too large to examine all of the samples. Just to illustrate this, let's simulate the distribution of sample means by choosing 1000 samples from the population (and thus repeating many, since we know there are only 252 in total). When I ran this the first time, I got $p = .508$, which is pretty close to the exact value of .5.

```
rand.samp.means = numeric(1000) # That's how many we'll create
for (i in 1:1000) {
  # Mean of 5-element sample from pop:
  rand.samp.means[i] = mean(sample(pop,5,replace=T))
}
sum(rand.samp.means <= samp.mean)/length(rand.samp.means)
```

Another way to estimate the *p* value is to assume that this distribution is basically normal. And indeed, as shown by the QQ-norm plot in Figure 2, the distribution of sample means does look sort of normal (as is also implied by the rightmost plot in Figure 1, if we imagine smoothing over all the bumps):

```
qqnorm(dist.samp.means)
qqline(dist.samp.means)
```

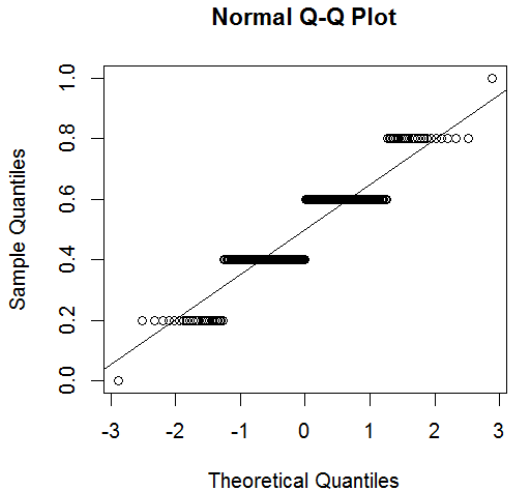


Figure 2. The sort of normal distribution of sample means

We shouldn't be too surprised that the distribution of sample means is basically normal, since the normal distribution keeps popping up all over the place in statistics. Knowing that it's basically normal then justifies using the **pnorm()** function to calculate the area to the left of the *z* score associated with our actual sample's mean (I hope you still remember how to do this from chapter 3):

```
mean.of.means = mean(dist.samp.means)
mean.of.means
[1] 0.5

sd.of.means = sd(dist.samp.means)
sd.of.means
[1] 0.1669983

samp.mean.z = (samp.mean - mean.of.means)/sd.of.means # z score of our sample mean
samp.mean.z
[1] -0.5988083
```

So it turns out that $z = -0.599$ (this time I include the 0, since *z* scores have no lowest or highest possible values). The value is negative, since our sample's mean of .4 is below the mean. Does this look like an outlier to you? Please try to remember what we discussed in chapter 3! In the standard normal distribution, the area in the tails two standard deviations away from the mean is around 5%, so you need a *z* score of around ±2 (more precisely, ±1.96) to create a two-tailed area this small. By contrast, the *z* score for the mean of our sample in the distribution of sample means is not even one standard deviation away from the mean of zero: no way is this an outlier!

To make this precise, let's use **pnorm()** to calculate the area in the tail to the left of our sample mean's *z* score:

```
pnorm(samp.mean.z)
[1] 0.2746503
```

So using this estimated method, $p = .27$. That's different from the exact $p = .5$ that we calculated using the actual proportion of sample means in the distribution of sample means at least as low as our sample mean, but that's because this estimated *p* value makes the assumption that our distribution is perfectly normal, which it isn't. In other words, an estimate only works if the assumptions used to generate it are true. Nevertheless, $p = .27 > .05$, so if our goal is just to decide if our sample mean is "significantly" different from chance, we can still conclude that it isn't.

3.4 The most important theorem in statistics

Now, we did all these calculations, both exact and estimated, based on our supernatural knowledge of the null hypothesis population. Sadly, in real life we don't really have access to the whole population; not only is it usually infinite, but it's about the null hypothesis, which is usually imaginary anyway. While a resampling approach may still be helpful (for basic introductions to resampling methods, see Good, 2005, and Vasishth & Broe, 2011), statistics has been around a lot longer than computers, of course, and even with computers, all the looping needed for resampling can make things pretty slow. This is where the **Central Limit Theorem** (中央極限定理) comes in: the most important theorem in statistics.

Here's how it works. In order to compute the tail area defined by our sample mean's z value in the null distribution of sample means, we need to know three things: this distribution's overall shape (is it really normal as we assumed above?), its mean (sometimes symbolized as $\boldsymbol{\mu_M}$), and its standard deviation (sometimes symbolized as $\boldsymbol{\sigma_M}$). The standard deviation of the distribution of sample means is more commonly called the **standard error** [of the mean] (標準誤), often symbolized as ***SE***, where "error" here means "measurement noise", not "stupid mistake". The Central Limit Theorem gives us all three of these things, justifying our use of **pnorm()** (or related functions we'll see shortly) to compute the p values.

The Central Limit Theorem states that as a sample size n gets larger and larger, the distribution of sample means becomes more and more normal, its mean gets closer and closer to the population mean, and its standard deviation (standard error) gets closer and closer to the population standard deviation, divided by the square root of our sample size:

Mean of distribution of sample means: $\mu_M \approx \mu$

Standard deviation of distribution of sample means = standard error: $\sigma_M = SE \approx \frac{\sigma}{\sqrt{n}}$

Yes, scary math again, but look more closely, and you'll see that these formulas aren't so crazy. First, even if the population distribution isn't normal at all, it makes sense that the distribution of sample means will tend to be more normal as the sample size increases, because computing means tends to move things towards the middle. We even saw this in our tiny fake example above: randomly selecting samples with $n = 5$ from the population {0, 0, 0, 0, 0, 1, 1, 1, 1, 1} almost always gave us samples containing a mixture of 0s and 1s (only the samples {0, 0, 0, 0, 0} and {1, 1, 1, 1, 1} are "pure"), so their means tended to be more in the middle, and indeed, due to the relationship between the binomial and normal distributions, the most common mean was right in the middle at .5. So the Central Limit Theorem gives us yet another reason to treat the normal distribution as "normal": it just keeps showing up in one situation after another.

Second, it seems reasonable that the mean of the distribution of sample means is the same as the mean for the whole population, especially as the sample gets larger and larger (and thus closer and closer to the whole population itself). Again, this was true for our tiny null hypothesis population and the tiny samples it generated.

Third, although it is trickier to see, it also makes a kind of sense to divide the population standard deviation by something to get the standard deviation for the distribution of sample means (i.e., the standard error), since the distribution of sample means should indeed be narrower than the population as a whole. In our fake example, the population itself had many extreme values (i.e., five 0s and five 1s), but the distribution of sample means, since it consists of means, moved values towards the center: it only contains one 0 (for the sample {0,0,0,0,0} and one 1 (for the sample {1,1,1,1,1}). Thus in our fake case, for the whole population, σ = **sd(pop)**= 0.5270463, while the distribution of sample means is narrower: σ_M = SE = **sd(dist.samp.means)** = 0.1669983. It also makes sense that the formula involves dividing by something relating to the sample size n, since as n gets bigger, our sample gets closer to the whole population, so the standard error gradually drops to 0, indicating that we get more and more sure of the population mean (less measurement error).

As for why you divide specifically by the square root of n, I can't give an informal explanation of this, but it relates to two facts: the normal distribution is ultimately defined by the variance (σ^2), not the standard deviation (σ) (as in that crazy formula I showed in section 4.2 in chapter 3), and computing the distribution of means of n-sized samples involves dividing by n. So at some point in deriving the Central Limit Theorem you divide one by the other to get σ^2/n, and the square root of that is the standard deviation of the distribution of sample means, or standard error: $\sigma/\sqrt{n}$. Satisfied?

In any case, note that this estimate sort of works for our simulated example, as shown below. The fit isn't great, since our sample is very small, and the Central Limit Theorem is a claim about the limit values as the sample size increases.

```
sd(dist.samp.means) # 0.1669983
sd(pop)/sqrt(5) # 0.2357023 (closer to 0.167 than sd(pop) = 0.527)
```

What would happen if we had a larger sample? Let's redo the simulation using randomly selected samples, starting with a much larger population so it makes sense to take a larger sample too:

```
big.pop = c(rep(0,5000),rep(1,5000)) # Same null hypothesis population (but N = 10,000)
big.samp = c(rep(0,30),rep(1,20)) # Same 3 vs. 2 proportion of 0 vs. 1, but now n = 50
# Too big to use combn() - try it (it just gives a fatal error)
big.rand.samp.means = numeric(100000) # That's how many samples we'll create
```

```
for (i in 1:100000) { # Many, many resamples to improve near-exact estimate (slow!)
  # Mean of 50-element sample
  big.rand.samp.means[i] = mean(sample(big.pop,50,replace=T))
}
sd(big.rand.samp.means) # Resampled estimate for SE: I got 0.07063253
sd(big.pop)/sqrt(50) # Central Limit Theorem estimate for SE: 0.07071421: very close!
```

Of course I won't give the proof for this theorem. Apparently it took until the 1930s before it was proved in general, a long time after tests using it were becoming popular (Salsburg, 2001, implies that this is a common theme in the history of statistics: applications come before the math!). But I hope the above discussion at least makes it seem plausible.

The Central Limit Theorem allows us to estimate the parameters of the distribution of sample means in a simple formula: we just plug the null hypothesis parameters directly into the *z* score formula to find out if our sample mean is an outlier in the distribution of sample means:

z score for the distribution of sample means: $z_M = \frac{M-\mu_M}{\sigma_M} = \frac{M-\mu_M}{SE} = \frac{M-\mu}{\sigma/\sqrt{n}}$

This *z* score tells us where our observed sample mean falls on the distribution of sample means hypothesized for the "boring" population of the null hypothesis. As we saw earlier, finding a *z* score on a standard normal distribution allows us to compute the area beyond the *z* (towards the tail). This area represents the *p* value, which indicates statistical significance: a smaller *p* means that our sample is a more extreme outlier on the random distribution of possible samples, and thus is less likely to be due to chance.

Let's try it out for our fake case (please note the careful use of parentheses, to make sure the right things are divided by the right things):

```
z.score = (mean(samp)-mean(pop)) / (sd(pop) / sqrt(length(samp)))
[1] -0.4242641

pnorm(z.score) # Area to the left of this value
[1] 0.3356866
```

Now we get $z = -0.423$ and $p = .336$, which are close to, though not exactly the same as, the $z = -0.599$ and $p = .275$ that we calculated using the "actual" 252-element distribution of sample means.

If you are paying attention, you may feel that I've tricked you, since to use the Central Limit Theorem, we still need to have supernatural knowledge about the population, namely its mean and standard deviation. How could we know all that, just by pondering our null hypothesis and looking at our sample? Well, the population we're testing is actually the boring null-hypothesis population, so we can just stipulate whatever "boring" population mean that

we want. For example, if we're doing a parametric test on accuracy rates, the most boring population mean would be .5, the accuracy rate from flipping a coin. The real problem is how we determine the population standard deviation, which is needed to compute the standard error, which estimates the width of the distribution of sample means, and hence the area of the tail(s) beyond our *z* score.

This problem confused early statisticians too, and it took them a little while to come up with a solution. I'll tell you the solution later in this chapter, but before I do so, it's more urgent to tell you a bit more about the logic of *p* values.

4. Hypothesis testing

Since (traditional) inferential statistics seems to be obsessed with *p* values, it is important to understand exactly what they mean - and what they don't mean. I'll illustrate the logic by returning to our original binomial case, involving the relative frequencies of [banana] and [ananab].

4.1 Back to bananas

Remember how we computed the *p* value for our case of 23 tokens of [banana] versus 37 tokens of [ananab], using the exact binomial test?

```
pbinom(q=23, size=60, prob=0.5) # Probability of at most 23 Hs in 60 coin flips
[1] 0.04623049
```

To get a sense of what this number is actually measuring, let's make the plot in Figure 3 (with the help of the function **segments(x0,y0,x1,y1)** to mark the area up to the twenty-third probability point):

```
plot((0:60),dbinom((0:60),size=60,prob=0.5),ylim=c(0,0.11)) # The 61 probabilities
segments((0:23),rep(0,23),(0:23),dbinom((0:23),size=60,prob=0.5)) # Up to 23rd point
```

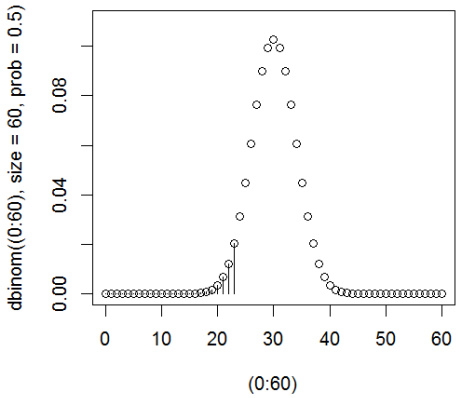


Figure 3. Area to left of the 23rd probability

Since $p = .04623049 < .05$, we might want to say that this pattern is too unlikely to be due to chance, and thus is statistically significant.

But wait a moment! Before we collected our banana data, we had no idea that one variant would be more common than the other. That's why the null hypothesis was that the frequencies for [banana] and [ananab] would be the same. But this null hypothesis could have been rejected *two* different ways, not just one: [ananab] could have been more common than [banana], as is in fact the case, but it could have instead turned out that [banana] was more common than [ananab]. If we really want to test the null hypothesis that both frequencies are equal, we should compute a *p* value that reflects both of these possibilities. That is, the *p* value should reflect the chance probability that the difference between the two frequencies is at least as high as we observed, regardless of the direction of this difference. This is a so-called **nondirectional hypothesis**, as opposed to the **directional hypothesis** reflected in our computation and figure above.

Putting this into graphical terms, what we computed was a **one-tailed test** (單尾檢定), when what we really want is a **two-tailed test** (雙尾檢定), as shown in Figure 4:

```
plot((0:60),dbinom((0:60),size=60,prob=0.5),ylim=c(0,0.11)) # The 61 probabilities
segments((0:23),rep(0,23),(0:23),dbinom((0:23),size=60,prob=0.5)) # Left tail
segments((37:60),rep(0,23),(37:60),dbinom((37:60),60,0.5)) # Right tail
```

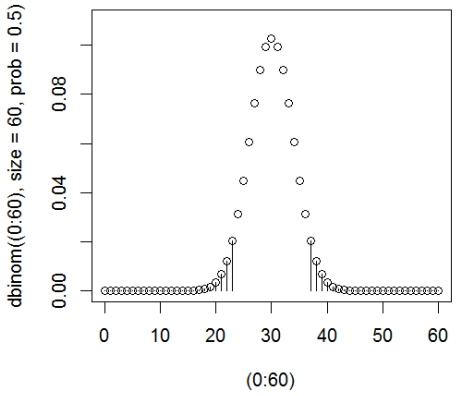


Figure 4. Areas beyond the 23rd probability in both left and right tails

When the chance probability for each outcome is 1/2 (as it is here), the binomial distribution is symmetrical, so computing a two-tailed *p* value merely involves multiplying the one-tailed *p* value by two:

```
2*pbinom(23, 60, 0.5) # Twice the probability of at most 23 Hs in 60 coin flips
[1] 0.09246098
```

Or we can make R do the thinking for us, and just use the **binom.test()** function with the default value for the **alternative** argument (**alternative="two.sided"**, i.e., two-tailed):

```
binom.test(23, 60, 0.5) # As usual, ?binom.test tells you more about defaults & options

        Exact binomial test

data:  23 and 60
number of successes = 23, number of trials = 60, p-value = 0.09246
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.2607071 0.5178850
sample estimates:
probability of success
             0.3833333
```

OH NO!!! Now *p* =.09 > .05, so it's no longer significant by the standard convention. We should've stayed with the one-tailed *p* value, so we would win! Winning is everything!!!

No, respecting the truth is everything, at least within the limits of our knowledge. Even though one-tailed *p* values are always smaller than two-tailed *p* values, making it easier to get "significant" results, they are rarely used, for some very good reasons. First, unlike two-tailed

tests, one-tailed tests have to be justified case by case, since they depend on an argument that your null hypothesis has some intrinsic direction. This may make skeptics wonder if you're trying to trick them. Second, since one-tailed tests give lower *p* values, it's human nature to feel tempted to change your hypothesis from two-tailed to one-tailed *after* running the experiment. For example, suppose you start with the null hypothesis that the mean reaction times for nouns and verbs are equal, but after the experiment you find that nouns are significantly faster, but only by a one-tailed test. Can you change your null hypothesis to the directional hypothesis that nouns are slower than verbs? No! This is like drawing the target after shooting the arrow. Finally, if you do choose a one-tailed test, you have to be ready to ignore any result that goes directly opposite your officially stated alternative hypothesis, and treat such a result as a "null result," even if it obviously isn't. For example, if your research hypothesis predicts that nouns should be faster, but instead you find that they are much, much, *much* slower, then by a one-tailed test this still counts as a "null result."

Hence this simple rule of thumb: Always use two-tailed *p* values if possible. That's why the default type of *p* value for **binom.test()** is two-tailed.

4.2 The math of hypothesis testing

Let's make this logic a bit more formal. In traditional statistics, the logical problem of **induction** (歸納推理) means that you can't prove that claims about the (infinite) chance population are *true*. You can only prove that claims about it are *false*. For example, if you want to show that nouns are easier to process than verbs, then your experiment is actually trying to *falsify* the **null hypothesis** (虛無假設) that there is *no difference* in the processing of nouns versus verbs. Thus if you find $p < .05$, this implies that your results are so unlikely to have arisen by chance that you are justified in claiming that your results are not due to chance, but if you find $p > .05$, you can conclude exactly nothing. Maybe nouns and verbs really are processed the same, or maybe they're processed differently but you failed to detect it. Another way to put this is that you can't prove a negative: you can show that the human mind *can* handle OSV word order just by finding one language with OSV word order, but to try to show that the human mind *cannot* handle OSV word order you would have to show that no *possible* human language could *ever* have OSV word order (finding no OSV word order in the set of attested languages would not be enough). In reality, OSV languages do exist, but I hope you get my general point.

The null hypothesis is formally symbolized as H_0. This is the "chance" hypothesis, the one that is modeled precisely by the math (e.g., the normal curve). Thus the *p* value represents the precise probability that H_0 is true, given your results and the assumptions you made in your analysis (e.g., that the normal curve was appropriate for your data).

Following the above logic, though, we can't prove that H_0 is true. All we can say is that the probability of H_0 is so high that we have no reason to reject it. If the probability of the null hypothesis is not low enough, we have a **null result**, which only shows that we failed to falsify the null hypothesis, *not* that the null hypothesis is true. The distinction here relates to what linguists call the "scope of negation": a null result is not(show(different)), rather than show(not(different)).

How low does the probability of H_0 have to be before we reject it? This is the **level of significance** (顯著水準) or **alpha level** (α水準), and it's set by convention, most commonly (in social sciences like linguistics) at $\alpha = .05$. Thus if we calculate p and find that it's less than α, then we are allowed to claim (by convention) that the results are statistically significant.

Why is alpha usually set to .05? This apparently goes back to a habit of the great British statistician Ronald Fisher (1890-1962), who invented so many modern statistical things (like ANOVA) that linguists might think of him as the Chomsky of statistics. However, he didn't like how the p value came to be used, where alpha represents a sharp criterion point separating "significant" from "non-significant". This approach was pushed by Polish statistician Jerzy Neyman (1894 - 1981) and British statistician Egon Sharpe Pearson (1895 - 1980), son of British statistician Karl Pearson (1857-1936); the elder Pearson invented so many fundamental statistical things (like correlations and the chi-squared test) that linguists might think of him as the Saussure of statistics (despite his German-sounding first name, he was an Englishman originally named "Carl"; he changed it to honor Karl Marx!).

The disagreement over p values, what they mean and what they don't mean, continues up to today. We'll see some reasons why shortly (and still more in the last chapter of this book), but one is illustrated by the popular (but controversial) notion of **marginal significance**. If you set your alpha level at .05 (as usual) but find $p = .06$, a strict application of the Neyman-Pearson approach means that you cannot count the result as significant. But since .06 feels so close to .05, many researchers (desperate for their research hypothesis to be right) will call any p value between .05 and .1 "marginally" significant. Fisher would approve of this approach, since for him the p value was just one piece of information among many that the researcher could use to argue for or against a research hypothesis; for him, there was truly nothing magical about the .05 threshold. Other researchers, however, starting with Neyman and Pearson themselves, would say that "marginal" significance is nonsense: you must fix your alpha level ahead of time, and stick with it throughout your analysis, no matter what you find.

One way to think about α is that it is the probability of making a certain type of error, namely rejecting the null hypothesis when it's actually true. If we set $\alpha = .05$, so that we reject the null hypothesis whenever $p < .05$, then about 5% of the time we'll be wrong just by pure bad luck: actually, the "pattern" we saw was just random noise. This kind of error is called a **false alarm**: believing in a pattern that's actually fake. A false alarm is considered so serious

that it's called a **Type I error**, and inferential statistics is designed to minimize it (we don't want to believe fake things).

However, another way we might be wrong is when the null hypothesis is *false* (so there really *is* a pattern) but by bad luck we got a null result and so aren't justifying in rejecting it. The probability of this kind of error (a **miss**), called a **Type II error**, is symbolized by β. Type II errors can also cause real-life problems (e.g., not being able to publish a paper because you missed a true pattern), but missing some truths is considered less serious than believing in lies (Type I error). In case you're confused by the terms, there's an old statistics joke that says that a Type III error is forgetting which is Type I and which is Type II....

The statistical **power** (検定力) of a test is the probability that a test will avoid a Type II error (i.e., will be able to detect a real pattern, and not confuse it with random noise). Since it's the probability of *not* getting a Type II error (β), and the total probability is always 1, power equals 1-β. Power can be increased by using a larger sample (i.e., closer to the actual population size) or by choosing a test that uses more information (e.g., a parametric test based on the raw values of a continuous variable, rather than another type of test that only looks at the relative ranking of the values). R includes some functions for estimating power and sample size, as we'll see shortly, but these aren't very practical in real life so I won't make a big deal out of them.

Crucially, the term "statistical significance" merely indicates that a pattern is unlikely to be due to chance. It may not be "significant" in real life. For example, a 1% difference in syntactic judgments may be statistically significant ($p < .05$), but if this tiny effect can only be detected when you have 100 speakers judging 500 sentences, it probably doesn't have any relevance to our understanding of grammar (Cowart, 1997, p. 123, discusses a real example of this). Moreover, even if a linguistic pattern seems "large" to us as linguists, it may still be "small" to actual language users. For example, maybe a morphological pattern has so few exceptions that it's not only statistically significant, but has been analyzed by linguists as being part of the grammar. But language learners need not use the same *p*-values that researchers use to define "significant": there may still be too many exceptions for the pattern to be learnable by real children. Yang (2016) discusses exactly this kind of situation.

The alternative to H_0 is symbolized as H_1: the alternative hypothesis. Again, in traditional statistics, you can't really prove that H_1 is *true*, but only argue that you are justified in rejecting H_0, which leaves H_1 (the logical opposite of H_0) as the only alternative. Thus if your null hypothesis is that reaction times for nouns and verbs are the same, then the alternative hypothesis would be like so, where μ_{noun} and μ_{verb} are the means for the populations of all noun and verb RTs:

H_1: $\mu_{noun} \neq \mu_{verb}$

The null hypothesis, which is what the *p* value actually represents, is the logical inverse:

H_0: $\mu_{noun} = \mu_{verb}$

If you ignore my warnings and want to test a directional null hypothesis anyway, for example that nouns are faster, the alternative and null hypotheses will be like so (note the ≥ for the null hypothesis, since > is *not* the logical inverse of <):

H_1: $\mu_{noun} < \mu_{verb}$
H_0: $\mu_{noun} \geq \mu_{verb}$

Another way to see what the *p* value means is to see what's wrong with common misconceptions about it. For example, finding that $p < .05$ doesn't guarantee that a replication of your study will also find $p < .05$. We can demonstrate this by going back to the banana example yet again, but this time using a population where the ratio of *a* to *b* is truly 37 to 23, that is, the alternative hypothesis population rather than the null hypothesis population. Then we'll take lots of 60-item samples, run a one-tailed binomial test on each one (for consistency with our original analysis), and count how often $p < .05$:

```
population = c(rep("a",37000),rep("b",23000)) # So our original sample was just right
sig.count = 0 # Will count how often the one-tailed binomial test is significant
for (i in 1:10000) {
  our.sample = sample(population,60,replace=T)
  a.count = sum(our.sample == "a")
  pval = pbinom(min(a.count, 60-a.count), 60, 0.5) # The left tail for a vs. b
  if (pval< 0.05) { # if significant
    sig.count = sig.count+1
  }
}
sig.count/10000 # Proportion of significant results
```

When I ran this the first time, I got .5515: only slightly more than half of the samples from this population show a significant effect! So even though the alternative hypothesis is true here (since we faked the data to make it true), people who want to replicate our study are quite likely to be disappointed (or happy to criticize us, if they're mean people).

The lesson here is: don't worship *p* values like some kind of magical road to capital-T Truth. In fact, the counterintuitive nature of *p* values has led to a lot of criticism (see, e.g., Nuzzo, 2014; Simmons et al., 2011; Amrhein et al., 2019), and has been yet another reason for the increasing use of Bayesian statistics, which doesn't rely on *p* values.

4.3 More on hits and false alarms

It's also worth noting that the concepts of hits and false alarms are logically important in empirical research far beyond their use in interpreting *p* values. For example, just as human

psychology is biased towards finding patterns, making it impossible for us to generate truly random numbers, it is also biased towards detecting hits rather than false alarms. Lilienfeld et al. (2010) illustrate this point with what they call the Great Fourfold Table of Life, representing all four possible ways a hypothesis could correspond to reality:

		Hypothesis	
		Pattern	No pattern
Reality	Pattern	**Hit**	**Miss**
	No pattern	**False alarm**	**Correct rejection**

Because (as we noted earlier) it's safer to hide whenever you think you see a tiger, even if there is no tiger at all, the human brain evolved to be very good at detecting hits, and terrible at detecting misses, false alarms, or even correct rejections. Particularly pernicious is **confirmation bias** (確認偏誤), which is the tendency to look only for evidence consistent with your hypothesis. To be a good scientist, then, you have to learn to go beyond your tiger-fearing evolutionary roots, and try to pay attention to all of the data, not just the bits that conform to your expectations. If you properly fill out a table like the above, you get what's called a **contingency table** (列聯表); we'll come back to this in a later chapter when we discuss chi-squared tests.

The notion of hits and false alarms also play a crucial role in a statistical method called **detection theory** (信號檢測理論), which can be useful in linguistics. For example, how can we estimate a speaker's vocabulary size (whether a child, a student, or a second-language learner)? One approach would be to give the person a lexical decision task, asking him or her to decide which items are real words and which items are fake. The overall accuracy on such a task tells us something about how good the person is at detecting real words, but there's a logical problem: it's possible to correctly identify all of the real words by responding "real" to *all* items, including the fake ones. This kind of result would be hard to interpret: the person may really know all those real words, but simply want to please you by responding "real" all the time. Detection theory provides a solution by separating the **sensitivity** of the person to the real/fake contrast from the person's overall response **bias** (here, to respond "real" no matter what). Sensitivity can be estimated by taking the hit rate (here, responding "real" to real words) and the false alarm rate (here, responding "real" to fake words), treating these rates as areas under a standard normal distribution and converting them into *z* scores (e.g., with Excel's **=NORMSINV()** or R's **qnorm()** functions), and then subtracting the latter from the former, thereby producing a normally distributed value called ***d′*** (pronounced "d prime") that can be studied using parametric statistics. I won't bother you with the details, but if you're curious, there are many places where you can learn more, including Huibregtse et al. (2002), Macmillan & Creelman (2005), and Myers et al. (2007).

4.4 Hypothesis testing summary

To summarize, here are the steps when testing a null hypothesis in traditional statistics. First, formulate the alternative hypothesis, and thus the null hypothesis, which is usually nondirectional, and which therefore uses a two-tailed test. Then choose α, which is .05 unless you have some reason to make it lower (like you want to be extra-sure that you don't commit a Type I error; .01 and .001 are other common alpha levels in social science, while in physics and engineering they prefer alphas of .000001 or lower).

Then choose the correct statistical test. For example, for a set of independent binary outcomes, choose the binomial test; for a set of normally distributed data, choose a **parametric** test (i.e., a test that builds on the normal distribution's two key parameters: the mean and the standard deviation). Then compute the **test statistic** (検定統計量), which is the number that comes out of the test. For a binomial test, the test statistic is just the numbers we plug into the **pbinom()** function, namely the number of "hits" and the total number of outcomes defining our event. In our example using the Central Limit Theorem (which I'll expand on in the next section), the test statistic was the z score for our sample mean on the distribution of sample means.

If you don't care about the precise p value, but just want to know whether it's statistically significant, you can compare the test statistic against the **critical value** (臨界性): this is the value of the test statistic that makes $p = \alpha$. For example, what is the critical value for a two-tailed test involving the standard normal distribution, where $\alpha = .05$? It's ± 1.96, because that's the z value defining the two tails with a total area of .05 (remember?). If you want the actual p value, you can compute it (using a function like **pnorm()** or some test-specific function that has a p-computing component inside it). No matter what the test is, the p value is always the area under a distribution relative to the position of the line marked by the test statistic. The p value will also depend on whether your hypothesis is directional, which determines if you use a one- or two-tailed test.

If the test statistic is beyond the critical value, or if the p value is below α, then you can call your results "significant". If not, it's a null result that doesn't really tell you very much at all.

Finally, report the results. This includes not just the p value, but also test name, test statistic, sample size information, and for some tests, information about standard error (*SE*). For example, to report our banana analysis, we might write something like this: "The numbers of [ananab] and [banana] utterances were, respectively, 37 and 23, but this trend failed to reach statistical significance by a two-tailed exact binomial test ($p = .09$)." Note that by reporting the number of each form and the name of the test, this tells the readers all they need to know to check our calculations.

5. One-sample parametric tests

The method we used in section 3.4 to compute *z* scores in the distribution of sample means (using the Central Limit Theorem to estimate its mean and standard deviation) is called the **one-sample *z* test**. It's a one-sample test since it tests a hypothesis about just one actual sample (in our case, the mean of our sample of 0s and 1s). Because the *z* test requires us to magically know the null hypothesis standard deviation (usually our null hypothesis is only about the mean), this test is not very useful in real life, but it leads directly into the very useful **one-sample *t* test**. This test is the secret power lying at the heart of many of the other tests we'll learn about in this book, from the very commonly used two-sample *t* test through ANOVA through linear regression. So let me end this chapter by explaining the short hop up in abstraction from *z* to *t*.

5.1 The one-sample *z* test (review)

Suppose you study 20 two-year-old kids acquiring Taiwanese (Southern Min) as their first language, and you discover that their mean vocabulary size is 160 words. So you can summarize your results like so. (The *z* test doesn't use the sample standard deviation; the *t* test does, though, as we'll see in 5.2).

$M_{\text{Taiwanese}} = 160$
$n = 20$

You wonder if this mean vocabulary size is significantly different from (i.e., higher or lower than) the vocabulary size for English. Now, unlike Taiwanese, English is very well studied in language acquisition research, so let's say that it's been fully established, based on a huge number of studies, that the mean vocabulary size for two-year-old kids acquiring English as a first language is 100 words, with a standard deviation of 50, forming a mostly normal distribution (of course I'm just making up these numbers for us to practice with). Since these values are so solid, you believe that it's legitimate to take them as population values:

$\mu_{\text{English}} = 100$
$\sigma_{\text{English}} = 50$

So your null hypothesis and alternative hypothesis are as follows:

H_1: $\mu_{\text{Taiwanese}} \neq \mu_{\text{English}} = 100$
H_0: $\mu_{\text{Taiwanese}} = \mu_{\text{English}} = 100$

You have one sample, you have a normally distributed population where the standard deviation is known, and your hypothesis involves your sample mean. This is just the kind of situation where you can use the one-sample *z* test.

Here's how it works. If the null hypothesis is true, your sample of Taiwanese mean comes from the distribution of sample means from the English population (the numbers, not the actual kids, of course). So what you want to know is whether your sample mean is an outlier on this distribution, in the sense that the area beyond your mean is $p < .05$. Fortunately, we don't have to generate or simulate the distribution of sample means, since the Central Limit Theorem lets us estimate the mean and standard deviation of this distribution for plugging into the *z* score formula, from which we can compute the *p* value.

As a reminder, here's the *z* score formula again, but now applied to a sample (of size *n*) with mean *M* and the null hypothesis distribution of the means of samples (with size *n*) which itself has mean μ_M and standard deviation σ_M:

z score in distribution of sample means: $z = \frac{M-\mu_M}{\sigma_M}$

And because of the Central Limit Theorem, we know that $\mu_M = \mu$ (the null hypothesis population mean) and σ_M is the standard error (*SE*), which is $\sigma_M/\sqrt{n}$:

z score using Central Limit Theorem: $z = \frac{M-\mu_M}{\sigma_M} = \frac{M-\mu_M}{SE} = \frac{M-\mu}{\sigma/\sqrt{n}}$

Now we just plug in the above values:

z test for our example: $z = \frac{M-\mu_M}{\sigma_M} = \frac{M-\mu_M}{SE} = \frac{M-\mu}{\sigma/\sqrt{n}} = \frac{160-100}{50/\sqrt{20}}$

Using Excel or R, we get *z* = **(160-100)/(50/sqrt(20))** = 5.366563 (careful with the parentheses!). Does this test significant represent a significance effect by a two-tailed test? Yes, definitely! You don't even need to do any more calculations to see that. Remember that -1.96 and 1.96 represent the critical values for the two tails of the standard normal distribution whose areas add up to .05, and 5.37 is obviously much larger than 1.96. Our test statistic is definitely an outlier in the distribution of sample means, so our results must be significant: the two-tailed *p* value must be below .05.

If you want a precise *p*-value, we can compute it in R as shown below (similarly in Excel). Notice the **-abs()** part, which uses the absolute value (絕對賦值) function to make sure the difference between *M* and μ is positive and then makes it negative, since R's function **pnorm()**

and Excel's function **=NORMSDIST(... CUMULATIVE=TRUE)** both give you the area to the left (negative half of the standard normal distribution).

```
2*pnorm(-abs(160-100)/(50/sqrt(20)))
[1] 8.025111e-08
```

Hm, 8-point-something.... Oh right: that "e-08" at the end means $1/10^8$, so this p value is actually .00000008...: a very tiny p value indeed! Not surprising, given that $z = 5.37$ is so far beyond the critical value of 1.96. To report these results, we could write them up like so: "A two-tailed one-sample z test showed that the mean Taiwanese vocabulary size is significantly different from the mean English vocabulary size: $z(20) = 5.37$, $p < .05$." Or we might write $p < .001$ or $p < .0001$ to emphasize how tiny the p value really is (but not $p = 8.025 \times 10^{-8}$, since when it's so tiny the precise value doesn't matter, and definitely not rounding it to $p = .000$, since that would be a lie). Note also that I wrote "significantly different from", not "significantly higher than", since we did a nondirectional two-tailed test; if the Taiwanese vocabulary size were much smaller than 100, that would still have been a significant difference from the null hypothesis.

5.2 The one-sample *t* test

The z test depends on knowing the standard deviation of the population (σ), which almost never happens in real life. However, we do have easy access to the standard deviation of the sample (s). Wouldn't it be nice if we could use that in our calculations instead...?

Well, due to the research of an Englishman named William Gosset (1876-1937), we can! Gosset showed, however, that if we use s as our estimate for σ, the reliability of the estimate depends even more on our sample size than is assumed by the plain old Central Limit Theorem alone: the bigger the sample, the more confident we should be that the sample s properly estimates the null hypothesis population σ.

The first step in doing this is to replace the universal z distribution with a family of ***t* distributions**, where each member of the t family depend on the sample size (the choice of the letter "t" was made, for unknown reasons, in a series of letters between Gosset and the great Fisher; Eisenhart, 1979). t distributions are shaped almost exactly like the z distribution, but they have fatter tails, to reflect their greater uncertainty about where the population mean really is. The bigger n is, the thinner the tails in the t distribution, and in fact, as $n \rightarrow \infty$, t turns into z (the normal distribution reappears yet again!).

We can get a sense of this from the plot in Figure 5, created with the help of R's t distribution density function **dt()** (cf. **dnorm()** for the normal distribution). Note the **plot()** argument **add=T**, which lets us add a new plot to an already-created plot; I'll explain the **df** argument in the **dt()** function shortly. (And **Sys.sleep()** is a way to make R pause for effect.)

```
plot(function(x) {dnorm(x)}, xlim=c(-3,3), lwd=3) # Normal distribution with thick line
for (i in c(1,5,10)) { # Ever larger samples
  plot(function(x) {dt(x, df=i)}, xlim=c(-3,3),add=T) # t distributions
  Sys.sleep(1)# Add a one-second pause so you can see each t distribution getting added
}
```

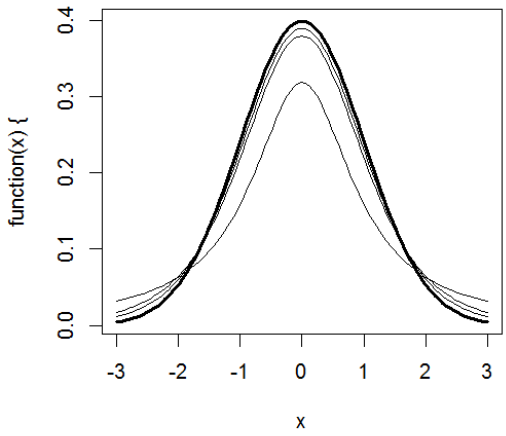


Figure 5. Some *t* distributions on top of a normal (*z*) distribution

The function giving the *t* distribution is not based on *n* directly, but on the **degrees of freedom** (自由度), abbreviated ***df*** (yet another invention of Fisher). You might remember that I mentioned *df* in chapter 3 (and apologized for the mysterious name). As I said then, roughly speaking, *df* represents the amount of unique information that you need to take into consideration (i.e., not counting other values that you can, or have to, guess, from the values you have). Generally, $df = n - m$, where *n* is the sample size and *m* is a small number that varies from test to test and is rarely obvious (Excel and R will compute it for you automatically anyway).

With this background, we can now perform a **one-sample *t* test**, where you compare an observed sample mean with the mean of the null hypothesis, while assuming that $\sigma = s$. The one-sample *t* test works exactly like the one-sample *z* test, except that now you use the fat-tailed *t* distribution, with $df = n - 1$.

(Before we go on here, here's a funny story about the *t* distribution. If you look up help in R about *t* functions, you'll see that statisticians call it "Student's *t*." Why? Because Gosset was forced to publish his research under a pseudonym, and he called himself "Student", to show that he was, like, humbly learning stuff. He had to use a pseudonym because he didn't want his bosses to know he was "wasting time" doing science, instead of focusing on his "real job". And what was his real job? Maintaining quality control in a *beer factory*! This story might also hint at the choice of the letter "t": it's the second letter of "Student", and "s" was already being used for the standard deviation.)

So as I was saying, the formula for the one-sample *t* test is exactly like the *z* test, except that you use *s* instead of *σ* when applying the Central Limit Theorem to estimate σ_M (*SE*). That is, for the *z* test the standard error (*SE*) was $\sigma/\sqrt{n}$, but for the *t* test it's $s/\sqrt{n}$:

t test statistic: $t = \frac{M-\mu}{SE} = \frac{M-\mu}{s/\sqrt{n}}$

The remaining steps then build on the same logic we've been using all along: to get the *p* value, instead of using the universal *z* distribution, you use the specific *t* distribution for your specific *df*, which for this test is *n*-1.

For example, suppose that you suspect that your sister may be a Martian in disguise. As is well known (remember, textbooks never lie), Martians pronounce /t/ with a mean voice onset time (VOT) of 20 ms, but for some stupid reason nobody knows what the standard deviation is. In the more realistic uses of the one-sample *t* test that we'll see in later chapters, the null hypothesis mean is always zero, literally "null", rather than being taken from an existing population, but I want to show that the logic works for any null hypothesis mean.

So you secretly record 16 of your sister's /t/ productions, getting *M* = 22 ms and *SD* = 3 ms, and you want to compare the mean of this sample with a population mean representing the null hypothesis (that your sister's VOT is just like a Martian's). Since VOT is normally distributed (as are most natural processes) and you don't know the population standard deviation, this is exactly the kind of situation where you can use a one-sample *t* test.

Using the above formula, you get *t* = **(22-20)/(3/4)** = 2.667. That's our test statistic for the one-sample *t* test. It's bigger than the magic number 1.96, but that magic number only applies to *z* scores, not *t* values (especially for small samples, where the *t* distribution has very thick tails), so just to be safe, let's compute the actual *p* value.

You can compute the *p* value using Excel's function **=T.DIST(*t*, *df*, TRUE)**. Notice the dot! Like **=NORMDIST()** and **=BINOMDIST()**, if **cumulative** = TRUE, then **=T.DIST()** gives the area to the *left* of the *t* value (or the density curve if **cumulative** = FALSE). This is the updated version of the old "dotless" function **=TDIST(*t*, *df*, tails)**, which works in a confusingly different way, giving the area of the *right* tail (if **tails** = 1) or both tails (if **tails** = 2) and only works at all if $t \geq 0$. So... let's not use that version.

In our dumb example, then, the two-tailed *p* value can be computed with **=T.DIST(*t*, *df*, cumulative)** by using the negative value of t, in order to get the left tail, and setting cumulative = TRUE to get the area, and then doubling it, to get the area of both tails:

=2*T.DIST(-ABS(22-20)/(3/4), (16-1), TRUE)
0.017595153

So that gives us two-tailed $p = .018 < .05$. So your sister's VOT is significantly different from the mean Martian VOT, so it seems safe to conclude that your sister is not a Martian

(whew!). You can then report this in your master's thesis (*Is My Sister a Martian?*) like so: "My sister's mean /t/ VOT was 22 ms (*SD* 3 ms), significantly different from the Martian mean VOT of 20 by a two-tailed one-sample *t* test ($t(15) = 2.67$, $p < .05$)." Notice that unlike the *z* test report we gave above, the 15 here is the degrees of freedom *df*, not the full sample size *n* (in a sense, the syntax *t*(*df*) implies that *t* is a function that takes the input *df* and gives you a value in that particular distribution).

The equivalent function in R is **pt(t, df)**, except that following the nerdy logic of R, it is consistent with R's other functions **pnorm()** and **pbinom()** in that it gives the *one-tailed* area to the *left* of the *t* value, unlike Excel's **=TTEST()** function. So the safest way to properly compute the two-tailed *p* value for any possible *t* value (whether it's negative or positive) is to use the **abs()** function to make *t* positive, then use - (minus) to make it negative, so you can be sure that you're computing the tail to the left: **pt(-abs(t),df)**. Then to get a two-tailed value (since they're better than one-tailed values), you double it: **2*pt(-abs((22-20)/(3/4)),16-1)** ≈ .018.

Unlike Excel, R also has a built-in function that can do all of the computations for the one-sample *t* test, called **t.test()** (as we'll see later, this function can also do other kinds of *t* tests). To replicate the Martian sister example with this function, we need the actual data, which don't exist. So let's generate fake data with $n = 16$, M = 22, SD = 3:

```
set.seed(19483) # To make sure you match my random sample (it was hard to find!)
sister = round(rnorm(n=16,mean=22,sd=3)) # Make fake sister data for this example
sister # These are the fake data points
[1] 20 23 29 21 22 21 18 25 25 16 21 21 25 22 21 22

mean(sister) # 22
sd(sister) # 3.03315: hopefully close enough to 3 to match our analysis above
t.test(sister,mu=20) # H0: mu = 20 (by default, t.test sets mu = 0)
```

Running the last function makes R print out a little text report for us:

```
        One Sample t-test

data:  sister
t = 2.6375, df = 15, p-value = 0.01865
alternative hypothesis: true mean is not equal to 20
95 percent confidence interval:
 20.38375 23.61625
sample estimates:
mean of x
       22
```

This text report (which is actually a character string, including the invisible line-break symbol "\n") gives us the name of the test ("One Sample t-test"), the *t* value (2.6375), *df* (15),

p (.01865), says that it's two-tailed, and gives μ ("alternative hypothesis: true mean is not equal to 20") and *M* (22), plus some other stuff we'll explain in a later chapter. The results are quite close to what we got above (not exactly the same, since my random sample has a slightly different standard deviation from the original version), and it also gives you everything you need for your report, except for *SD* (*s*), which you can compute yourself using **sd(sister)**. Compare the *p* value in the output of **t.test()** with the hand-computed value:

```
2*pt(-abs((mean(sister)-20)/(sd(sister)/sqrt(length(sister)))),length(sister)-1)
[1] 0.01865043
```

The simplest way to extract the *t* test statistic and other values from the report is simply to copy/paste from the R screen, but if you ever need to write additional code to process them, you can extract them from the object produced by the **ttest()** function. For example, to pull out the *t* test statistic (AKA the *t* value), you can refer to the property **statistic** using the $ operator (which I learned about by opening the help page with **?t.test**, then looking in the "Value" section):

```
t.test(sister,mu=20)$statistic # The t test statistic (t value)
        t
2.637522

t.val = t.test(sister,mu=20)$statistic # Put this info in a variable to use later, if you want
2*pt(-abs(t.val),df=length(sister)-1) # p value (the "t" is its old vector element name)
             t
0.01865043
```

It's good to know your sister isn't a Martian, but when would you ever need to use a one-sample *t* test in real life? Almost never by itself, but very often as part of another test: the tests for correlations, two-sample *t* tests, paired *t* tests, and regressions all involve calculations closely related to the one-sample *t* test. Moreover, the basic logic is also used in ANOVA, which is a generalization of the *t* test. In fact, the *F* value used in ANOVA (*F* for Fisher!) is related to the square of the *t* value. As noted earlier, in all such realistic uses of the one-sample *t* test, the null hypothesis mean is zero.

5.3 A note on statistical power

And that's it for all the important things to know about *t* for now. But let me tell you just one more unimportant thing, in case you see discussion of it elsewhere (e.g., in Johnson, 2008). Namely, now that we have a specific test in hand, we can return to the notion of power (i.e., 1-β, where β is the probability of a Type II error, where you miss a real pattern). Because of the interrelations of all of the values we've been playing with, R has a built-in function, called

power.t.test(), that can compute the sample size *n* needed to get a significant result, given *α* (significance level), the size of the expected effect (delta = *Δ*, for difference), in a sample with a given standard deviation (*sd*), for a particular sort of *t* test (e.g., one sample), and the type of alternative hypothesis (e.g., two-tailed). So if we know the expected effect size and variance, we can save money and time by collecting just enough data to give us the power we desire (by convention, often set at .8). More generally, we can compute any of the values (**n**, **sd**, **delta**, **power**, **sig.level**) from all of the others, by leaving it out when we run the function.

For example, to compute the power of the test of your sister, I enter everything I know except for the power information, and look at the text report produced by R (I like to put initial zeroes in my proportional values inside functions, to avoid typos while typing):

```
power.t.test(n=16, sig.level=0.05, delta=abs(mean(sister)-20), sd=sd(sister),
type="one.sample", alternative="two.sided")
```

```
     One-sample t test power calculation
              n = 16
          delta = 2
             sd = 3.03315
      sig.level = 0.05
          power = 0.6936063
    alternative = two.sided
```

This analysis shows that the power of our test was lower than we might like: only .69, instead of the conventional .8. So to be even more sure that your sister isn't a Martian, we might want to increase the sample size by collecting more data. In fact, to reach the target power of .8, we would need to collect a total of about 20 data points, as you can see by running the following function:

```
power.t.test(power=0.8, sig.level=0.05, delta= abs(mean(sister)-20), sd=sd(sister),
type="one.sample",alternative="two.sided")
```

Still, this kind of analysis isn't very useful in real life. If you haven't even run the study yet, how can you know values like the sample standard deviation or the expected difference size? One situation in which you might need to do this is if you're writing a grant proposal, and need to convince the money providers that you can run your study with as few measurements (i.e., as cheaply) as possible.

6. Summary

As usual, we covered a lot of ground in this chapter. First we reviewed what you already know about probability, with basic notions like its limits of 0 (never) to 1 (always), the addition

rule, and the multiplication rule, but also showed that when conditional probability comes into play, things can get very counterintuitive. Conditional probability is crucial, however, since it defines the notion of independence, which is assumed by the simplest statistical tests. The very simplest of these is the binomial test, for events composed of independent binary outcomes, which you can compute in Excel or R using functions for the binomial distribution. This test gives you *p* values representing the area of the tails of the distribution, with the emphasis on the plural: two-tailed tests, which test nondirectional null hypotheses, are almost always preferred over one-tailed tests. The binomial test is an exact test, but most statistical tests merely estimate *p* values by building on the logic of sampling. In traditional (non-Bayesian) statistics, we imagine that we select all possible samples (of the same size as ours) from a population representing the null hypothesis, and then compute a test statistic for our sample to see if it is an outlier in the distribution of these samples. When we're doing a parametric test, the distribution is assumed to be normal, and our focus is on means (and variances, as we'll see more when we get to ANOVA). Thus the sampling distribution of interest is the distribution of sample means, the parameters for which can be estimated from the population parameters (mean and standard deviation) using the Central Limit Theorem; the standard deviation of the distribution of sample means is called the standard error, and is computed as $\sigma/\sqrt{n}$. This logic allows us to run a one-sample *z* test. In most real-life situations, we just assume that the null population standard deviation is identical to our sample's standard deviation, forcing us to take sample size into account when computing the test statistic. Thus we shift over from the universal standard normal distribution to the sample-size-dependent family of *t* distributions, choosing the proper distribution from this family based on the degrees of freedom (*df*), here computed as one less than the sample size (*n*-1); we tried this out in the one-sample *t* test. Along the way, we saw how to estimate *p* values using simulations involving resampling, the use and misuse of *p* values, and the logic of hits and false alarms.

References

Amrhein, V., Greenland, S., & McShane, B. (2019). Retire statistical significance. Nature, 567, 305-307. https://www.nature.com/articles/d41586-019-00857-9

Anttila, A. (1997). Deriving variation from grammar. In F. Hinskens, R. Van Hout, & W. L. Wetzels (Eds.) *Variation, change and phonological theory* (pp. 35-68). John Benjamins.

Boersma, P., & Hayes, B. (2001). Empirical tests of the Gradual Learning Algorithm. *Linguistic Inquiry, 32*, 45-86.

Cowart, W. (1997). *Experimental syntax: Applying objective methods to sentence judgments*. Sage.

Cysouw, M. (2005). Quantitative methods in typology. In R. Kohler, G. Altmann, & R. G. Piotrowski (Eds.) *Quantitative Linguistik: Ein internationales Handbuch* [Quantitative linguistics: An international handbook] (pp. 554-578). Walter de Gruyter.

Eisenhart, C. (1979). On the transition from "Student's" *z* to "Student's" *t*. *The American Statistician, 33*(1), 6-12.

Fasold, R. W. (1990). *The sociolinguistics of language*. Basil Blackwell.

Gelman, A. (2005). The sampling distribution of the sample mean. http://andrewgelman.com/2005/08/12/the_sampling_di/

Good, P. I. (2005). *Introduction to statistics through resampling methods and R/S-Plus*. Wiley.

Huibregtse, I., Admiraal, W., & Meara, P. (2002). Scores on a yes-no vocabulary test: Correction for guessing and response style. *Language Testing, 19*(3), 227-245.

Ioannidis, J. P. A. (2005). Why most published research findings are false. *PLoS Medicine, 2* (8), e124.

Johnson, K. (2008). *Quantitative methods in linguistics*. Wiley.

Kish, L. (1965). *Survey sampling*. Wiley.

Labov, W. (1994). *Principles of linguistic change: internal factors*. Blackwell.

Lilienfeld, S. O., Lynn, S. J., Ruscio, J., & Beyerstein, B. L. (2010). *50 great myths of popular psychology: Shattering widespread misconceptions about human behavior*. Wiley-Blackwell.

Macmillan, N. A., & Creelman, C. D. (2005). *Detection theory: A user's guide* (second edition). Lawrence Erlbaum.

McGrayne, S. B. (2011). *The theory that would not die: How Bayes' rule cracked the Enigma code, hunted down Russian submarines, & emerged triumphant from two centuries of controversy*. Yale University Press.

Myers, J. (2006). Tone circles and chance. National Chung Cheng University ms.

Myers, J., Taft, M., & Chou, P. (2007). Character recognition without sound or meaning. *Journal of Chinese Linguistics, 35* (1), 1-57.

Nuzzo, R. (2014). Statistical errors. *Nature, 506*(7487), 150-152.

Oaksford, M., & Chater, N. (2009). Précis of *Bayesian rationality: The probabilistic approach to human reasoning*. *Behavioral and Brain Sciences, 32*, 69-120.

Paulos, J. A. (2011). Animal instincts: Are creatures better than us at computation? Scientific American, 18. https://www.scientificamerican.com/article/animal-instincts/

Salsburg, D. (2001). *The lady tasting tea: How statistics revolutionized science in the twentieth century*. New York: W. H. Freeman & Company.

Simmons, J. P., Nelson, L. D., & Simonsohn, U. (2011). False-positive psychology undisclosed flexibility in data collection and analysis allows presenting anything as significant. *Psychological Science, 22*(11) 1359-1366.

Sorace, A., & Keller, F., (2005). Gradience in linguistic data. *Lingua 115*(11), 1497-1524.

Stewart, I. (1997). *The magical maze*. Phoenix.

Vasishth, S., & Broe, M. (2011). *The foundations of statistics: A simulation-based approach*. Springer.

Woods, A., Fletcher, P., & Hughes, A. (1986). *Statistics in language studies*. Cambridge.

Yang, C. (2016). *The price of linguistic productivity: How children learn to break the rules of language*. MIT Press.