CHAPTER 1

The Basic Idea

Suppose you want to know whether a coin is fair[1]. You toss it 17 times and it comes up heads all but 2 times. How might you determine whether it is reasonable to believe the coin is fair? (A fair coin should come up heads with probability 1/2 and tails with probability 1/2.) You could ask to compute the percentage of times that you would get this result if the fairness assumption were true. Probability theory would suggest using the binomial distribution. But, you may have forgotten the formula or the derivation. So, you might look it up or at least remember the name so you could get software to do it. The net effect is that you wouldn't understand much, unless you were up on your probability theory.

The alternative is to do an experiment 10,000 times, where the experiment consists of tossing a coin that is known to be fair 17 times and ask what percentage of times you get heads 15 times or more (see Fig. 1.1). When we ran this program, the percentage was consistently well under 5 (that is, under 5%, a result often used to denote "unlikely"), so it's unlikely the coin is in fact fair. Your hand might ache from doing this, but your PC can do this in under a second.

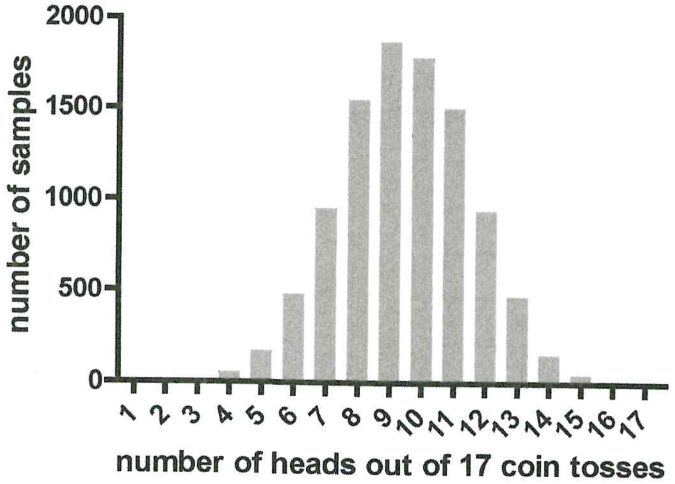

Figure 1.1: Coin toss.

[1] We take this coin example from David Howell. See http://www.uvm.edu/~dhowell/StatPages/Resampling/philosophy.html.

python™

All code was tested using Python 2.3.

download code and input files

Here is an example run of the Coinsig.py code:

```
9 out of 10,000 times we got at least 15 heads in 17 tosses.
Probability that chance alone gave us at least 15 heads in 17 tosses is 0.0009 .
```

Here is a second example.
Imagine we have given some people a placebo and others a drug. The measured improvement (the more positive the better) is:

```
Placebo: 54 51 58 44 55 52 42 47 58 46
Drug: 54 73 53 70 73 68 52 65 65
```

As you can see, the drug seems more effective on the average (the average measured improvement is nearly 63.7 (63 2/3 to be precise) for the drug and 50.7 for the placebo). But, is this difference in the average real? Formula-based statistics would use a *t*-test which entails certain assumptions about normality and variance; however, we are going to look at just the samples themselves and *shuffle* the labels.

The meaning of this can be illustrated in the following table—in which we put all the people—labeling one column 'Value' and the other 'Label' (P stands for placebo, D for drug).

Value	Label
54	P
51	P
58	P
44	P
55	P
52	P
42	P
47	P
58	P
46	P
54	D
73	D
53	D

70	D
73	D
68	D
52	D
65	D
65	D

Shuffling the labels means that we will take the P's and D's and randomly distribute them among the patients. (Technically, we do a uniform random permutation of the label column.)

This might give:

Value	Label
54	P
51	P
58	D
44	P
55	P
52	D
42	D
47	D
58	D
46	D
54	P
73	P
53	P
70	D
73	P
68	P
52	D
65	P
65	D

In Fig. 1.2, we can then look at the difference in the average P value vs. the average D value. We get an average of 59.0 for P and 54.4 for D. We repeat this shuffle-then-measure procedure 10,000 times and ask what fraction of time we get a difference between drug and placebo greater than or equal to the measured difference of 63.7 - 50.7 = 13. The answer in this case is under 0.001. That is less than 0.1%. Therefore, we conclude that the difference between the averages of the samples is real. This is what statisticians call *significant*.

Let's step back for a moment. What is the justification for shuffling the labels? The idea is simply this: if the drug had no real effect, then the placebo would often give more improvement than the drug. By shuffling the labels, we are simulating the situation in which some placebo measurements replace some drug measurements. If the observed average difference of 13 would be matched or even exceeded in many of these shufflings, then the drug might have no effect beyond the placebo. That is, the observed difference could have occurred by chance.

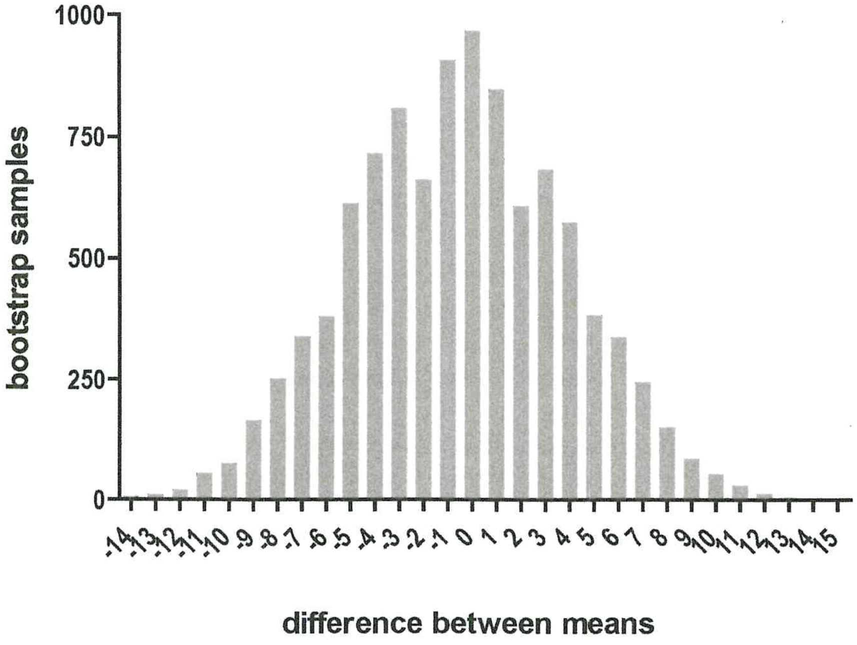

Figure 1.2: Difference between means.

To see that a similar average numerical advantage might lead to a different conclusion, consider a fictitious variant of this example. Here we take a much greater variety of placebo values: 56 348 162 420 440 250 389 476 288 456 and simply add 13 more to get the drug values: 69 361 175 433 453 263 402 489 301 469. So the difference in the averages is 13, as it was in our original example. In tabular form we get the following.

Value	Label
56	P
348	P
162	P
420	P

440	P
250	P
389	P
476	P
288	P
456	P
69	D
361	D
175	D
433	D
453	D
263	D
402	D
489	D
301	D
469	D

This time, when we perform the 10,000 shufflings, in approximately 40% of the shufflings; the difference between the D values and P values is greater than or equal to 13. So, we would conclude that the drug may have no benefit — the difference of 13 could easily have happened by chance.

All code was tested using Python 2.3.

download code and input files

Here is an example run of the Diff2MeanSig.py code, using the **first** data set from this example as input:

```
Observed difference of two means: 12.97
7 out of 10,000 experiments had a difference of two means greater than or
equal to 12.97 .
The chance of getting a difference of two means greater than or equal to 12.97 is
0.0007.
```

In both the coin and drug case so far, we've discussed **statistical significance**. Could the observed difference have happened by chance? However, this is not the same as importance, at least not always. For example, if the drug raised the effect on the average by 0.03, we might not find this important, even if the result is statistically significant. That is, the first question you should ask when someone tells you an effect is statistically significant is: “Yes, but how large is the effect?” Perhaps what is being measured here is survival. Say average survival on the placebo is 5 years, and that the drug increases survival on average by 3 days. The difference between 5 years and 5 years and 3 days may be significant, but it is not a large effect.

To get a feeling for this question of importance, we will use the notion of a *confidence interval*. Intuitively, the confidence interval of an imperfectly repeatable measurement is defined by the range of values the measurement is likely to take. In resampling statistics as in traditional statistics, this range is commonly defined as the middle 90% (or sometimes 95%) of the possible values. If you’ve been following carefully so far, you will guess that the set of possible values will be based on repeated random samples of some sort. In the drug case, we will take many samples from the patient data we have and then look at the difference between the average drug improvement and the average placebo improvement. We’ll look at the range of these differences and compute the confidence interval. This technique is called *bootstrapping*.

Here’s the method: we create new samples of the same size as the original by choosing values from the original sample “uniformly at random and with replacement.”

Let’s break down the phrase. “Uniformly at random” means each new sample element is chosen from the original sample in such a way that every original sample element has the same chance of being picked. “With replacement” means that even though an original sample element has been picked, its chance of getting picked again remains the same. Simply put, in forming a new sample (called a bootstrap sample), we choose uniformly at random on the original sample and may choose some elements twice or more and some elements no times at all.

Let’s recall our original data regarding drugs and placebos:

```
Placebo: 54 51 58 44 55 52 42 47 58 46
The average is: 50.7.

Drug: 54 73 53 70 73 68 52 65 65
The average is: 63.7.
```

NOTE:
As we will see **later**, this is in fact not enough data to justify the confidence interval procedure, but is used for easier illustration.

We subtract the placebo average from the drug average, yielding 63.7 - 50.7 = 13.

Our question now will be: “What is the 90% confidence interval of difference in the averages between the drug patients and placebo?” We answer this with experiments of the form: take a bootstrap sample of the placebo patients and compute the average; take a bootstrap sample of the drug patients and compute the average; then subtract the placebo average from the drug average. When we do this 10,000 times (the rule of thumb for bootstrapping is 1,000 times, but to increase the probability of capturing a wider range of values, we advocate increasing this to 10,000), we get many differences.

Here is a typical experiment in which a bootstrap sample of the placebo values is (note that 54 and 55 are repeated a few times, but 52 never appears):

```
55 54 51 47 55 47 54 46 54 54
The average is: 51.7.
```

Here is a bootstrap of the drug values:

```
68 70 65 70 68 68 54 52 53
The average is: 63.1.
```

We subtract the placebo average from the drug average, yielding 63.1 - 51.7 = 11.4.

When we repeated such an experiment 10,000 times and performed the subtraction each time, the lowest difference was -0.46 (the placebo is a tiny bit more effective than the drug). The highest was 23.4 (the drug is much more effective than the placebo). A more interesting range is the value 5% from the lowest and 95% from the lowest (percentile 5% and 95%). That is, arrange the differences in sorted order from smallest to largest and pick the differences that are at position 500 (500 is 5% of 10,000) and the difference at position 9,500. That is the 90% confidence interval. In our experiments, this yields a range of 7.81 to 18.11. That is, 90% of the time, drugs should yield a value that is 7.81 to 18.11 more than the placebo.

All code was tested using Python 2.3.

download code and input files

Here is an example run of the Diff2MeanConf.py code:

```
Observed difference between the means: 12.97
We have 90.0 % confidence that the true difference between the means is between:
7.81 and 18.11
```

Confidence intervals may vary vastly more for social/cultural phenomena than for physical/biological ones. Suppose we have 20 people and we compute their average income. Most have an annual income in the multi-thousand dollar range, but one person has an income of a billion dollars. The average will therefore be something like $50 million, even though most people don't make that much.

What we might be interested in is how far that average value varies if this were in fact a typical sample. Using **bootstrapping** we might have incomes (measured in thousands) as follows:

```
200 69 141 45 154 169 142 198 178 197 1000000 166 188 178 129 87 151 101 187 154
```

This gives an average of 50,142 thousands.

Now if we use the bootstrap, here is another sample:

```
151 154 166 188 154 101 1000000 129 188 142 188 129 142 188 151 87 200 178 129 166
```

This has an average of 50,146 thousands.

Another one:

```
154 87 178 151 178 87 154 169 187 129 166 154 154 166 198 154 141 188 87 69
```

This has an average of only 147 thousands (because the billionaire is missing).

Another one:

```
69 166 169 142 188 198 154 45 187 166 87 154 1000000 87 151 166 101 154 1000000
```

This has an average of 100,127 thousands because the billionaire is present twice.

The net effect is that we are going to get a wide variety of averages. In fact, when we ran bootstrap 10,000 times on our PC, we obtained a low average of 114 thousands and a high aver of 300,118 thousands. But these highs and lows are not so interesting because they vary a depending on how many times the billionaire happens to appear. A more interesting range is value 5% from the lowest and 95% from the lowest (percentile 5% and 95%). That is, arrange averages in sorted order from smallest to largest and pick the average that is at position 500 (50 5% of 10,000) and the average at position 9,500.

On our PC, the 5th percentile is 138 thousands and the 95th percentile is 200,130 thousa Because 95 - 5 = 90, this is the 90% confidence interval. Because of this vast range, we'd prob conclude that the average is not very informative.

When data has such extreme *outliers*, there are sometimes reasons for ignoring them, such faulty meter reading. If there is a good reason to ignore the billionaire in this case, then we g 90% **confidence interval** of about 132 to 173 thousands, a much more narrow range of expe values. Unfortunately, outliers are ignored incorrectly sometimes, so one must be careful. Also Nassim Taleb points out persuasively, outliers are much more common in human constructs (income or inflation) than in natural phenomena (like rainflow) 🗗. He gives a particularly conv ing example: in the early 1920s, inflation in Germany caused the exchange rate from Ger Marks to U.S. dollars to go from 3 to a dollar to 4 trillion to a dollar (that is a statistical impossi ity under the normal distribution assumption, which illustrates why blind application of the nor distribution can be dangerous).[2]

If you've been following this carefully, you might now wonder "If I have a **confidence inter** what more does **significance** bring to the party?" To answer this intuitively, consider a sin example in which you have just one element of group A having value 50 and one element of gr B having value 40. The confidence interval using replacement will say that the difference is alw 10. But intuitively this is way too little data. The significance test (in which one permutes the gr labels) will show that half the time one will get, just by random chance, a difference as big as observed one.

If all this seems easy, that's good. Several studies indicate that resampling is easier for dents to learn and they get correct answers more often than using classical methods 🗗.

[2] Why do we prefer **confidence intervals** based on the bootstrapping method to traditional confidence intervals ba on the standard deviation of the data? First, because we don t want to have to make the assumption that the und ing distribution is normal. Second, because many distributions are in fact skewed. For example, if we want to know average salary of a population and we know the salaries of a sample, we expect salaries to be positive, whereas average less the standard deviation might in fact be negative. Bootstrapping looks at the data that is present.