Sharpen your pencil

Answer the following questions from the city council about your work with LitterGitters.

Why can't you measure tonnage directly?

.....

.....

.....

Can you prove that the campaign had an effect?

.....

.....

.....

Can you guarantee that your tactics will continue to work?

.....

.....

.....

Why not spend money on cleanup rather than education?

.....

.....

.....

You guys are just as incompetent as the others.

.....

.....

.....

Sharpen your pencil Solution

How did you respond to the city council?

Why can't you measure tonnage directly?

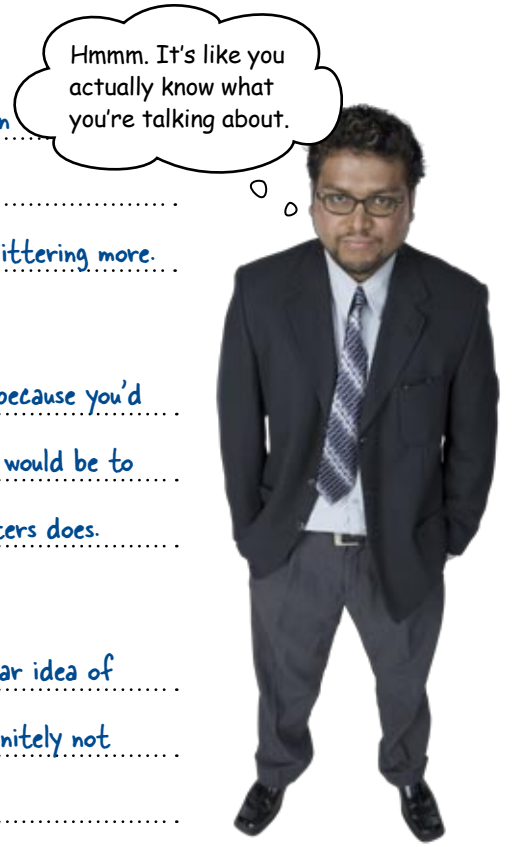
We can measure tonnage directly. The problem with doing it, though, is that it'd be too expensive. It'd cost twice the amount of money you actually pay LitterGitters to do their work. So the best course of action is to use this heuristic to assess performance. It's simple but in our belief accurate.

Can you prove that the campaign had an effect?

All the data is observational, so we can't prove that the increase in awareness of the general public about litter and the reduction that sanitation workers believe has taken place is the result of LitterGitters. But we have good reasons to believe that our program was the cause of these results.

Can you guarantee that your tactics will continue to work?

There are never guarantees in life, but as long as we can sustain the improved public awareness that came out of our outreach program, it's hard to imagine that people will suddenly resume littering more.



Why not spend money on cleanup rather than education?

But in that case, your objective wouldn't be to reduce litter, because you'd be doing nothing to get people to stop littering. The objective would be to clean it up as fast as you can, and that's not what LitterGitters does.

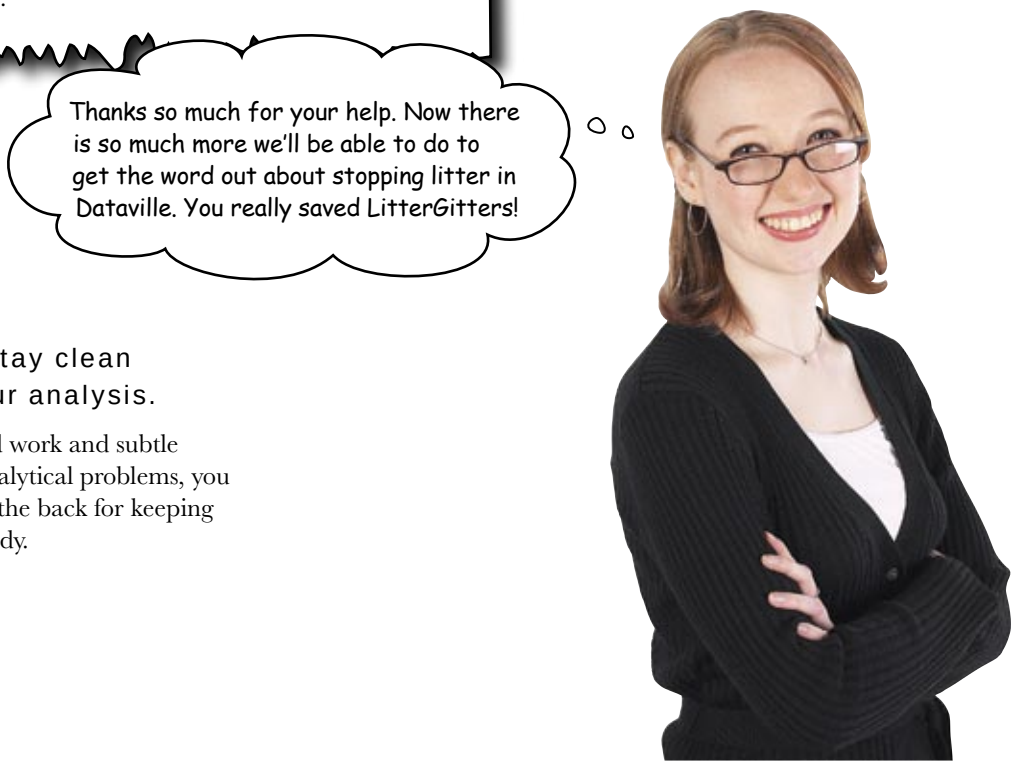
You guys are just as incompetent as the others.

We can't speak for other nonprofits, but we have a crystal clear idea of what we're doing and how to measure the results, so we're definitely not incompetent. When did you say you were up for reelection?

Looks like your analysis impressed the city council members

Memorandum
Re: LitterGitters and litter in Dataville

The city council is pleased to renew the contract of LitterGitters, thanks to the excellent work from the Head First data analyst. We recognize that our previous assessment of the work of LitterGitters did not adequately treat the whole issue of litter in Dataville, and we discounted the importance of public opinion and behavior. The new decision procedure you provided is excellently designed, and we hope the LitterGitters continue to live up to the high bar they have set for themselves. LitterGitters will receive increased funding from the Dataville City Council this year, which we expect will help…

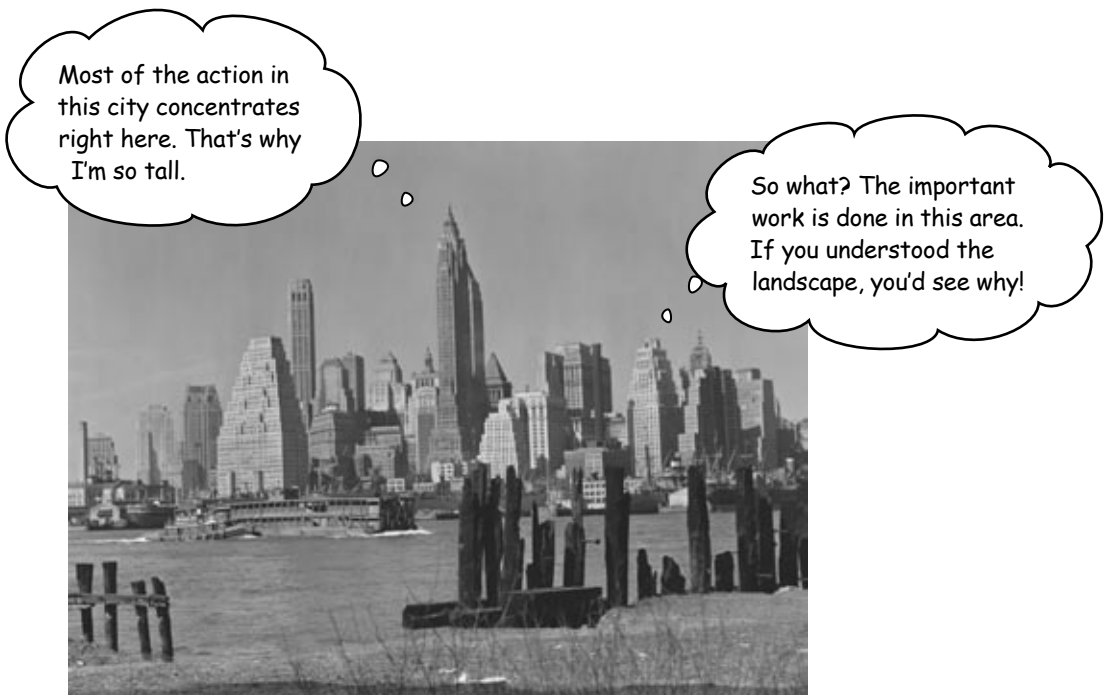


Dataville will stay clean because of your analysis.

Thanks to your hard work and subtle insight into these analytical problems, you can pat yourself on the back for keeping Dataville neat and tidy.

9 histograms

The shape of numbers



How much can a bar graph tell you?

There are about a zillion ways of **showing data with pictures**, but one of them is special. **Histograms**, which are kind of similar to bar graphs, are a super-fast and easy way to summarize data. You're about to use these powerful little charts to measure your data's **spread, variability, central tendency**, and more. No matter how large your data set is, if you draw a histogram with it, you'll be able to "see" what's happening inside of it. And you're about to do it with a new, free, crazy-powerful **software tool**.

Your annual review is coming up

You've been doing some really excellent analytical work lately, and it's high time you got what's coming to you.

The powers that be want to know what you think about your own performance.

Oh boy, a self evaluation.

Starbuzz Analyst Self-review

Thank you for filling out our self-review! This document is important for our files and will help determine your future at Starbuzz.

Date _____

Analyst Name _____

Circle the number that represents how well-developed you consider your abilities to be. A low score means you think you need some help, and a high score means you think your work is excellent.

The overall quality of your analytical work.

1 2 3 4 5

Your ability to interpret the meaning and importance of past events.

1 2 3 4 5

Your ability to make level-headed judgements about the future.

1 2 3 4 5

Quality of written and oral communication.

1 2 3 4 5

Your ability to keep your client well-informed and making good choices.

1 2 3 4 5

Bet you'd score higher now than you would have in chapter 1!

Your work is totally solid.

You deserve a pat on the back.

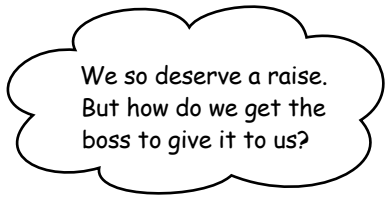
Not a literal pat on the back, though… something more. Some sort of *real* recognition. But what kind of recognition? And how do you go about actually getting it?

You'd better brainstorm about strategies to get recognized. Write down how you'd respond to each of these questions.

Should you just say thanks to your boss and hope for the best? If your boss really believes you've been valuable, he'll reward you, right?

Should you give yourself super-positive feedback, and maybe even exaggerate your talents a little? Then demand a **big raise?**

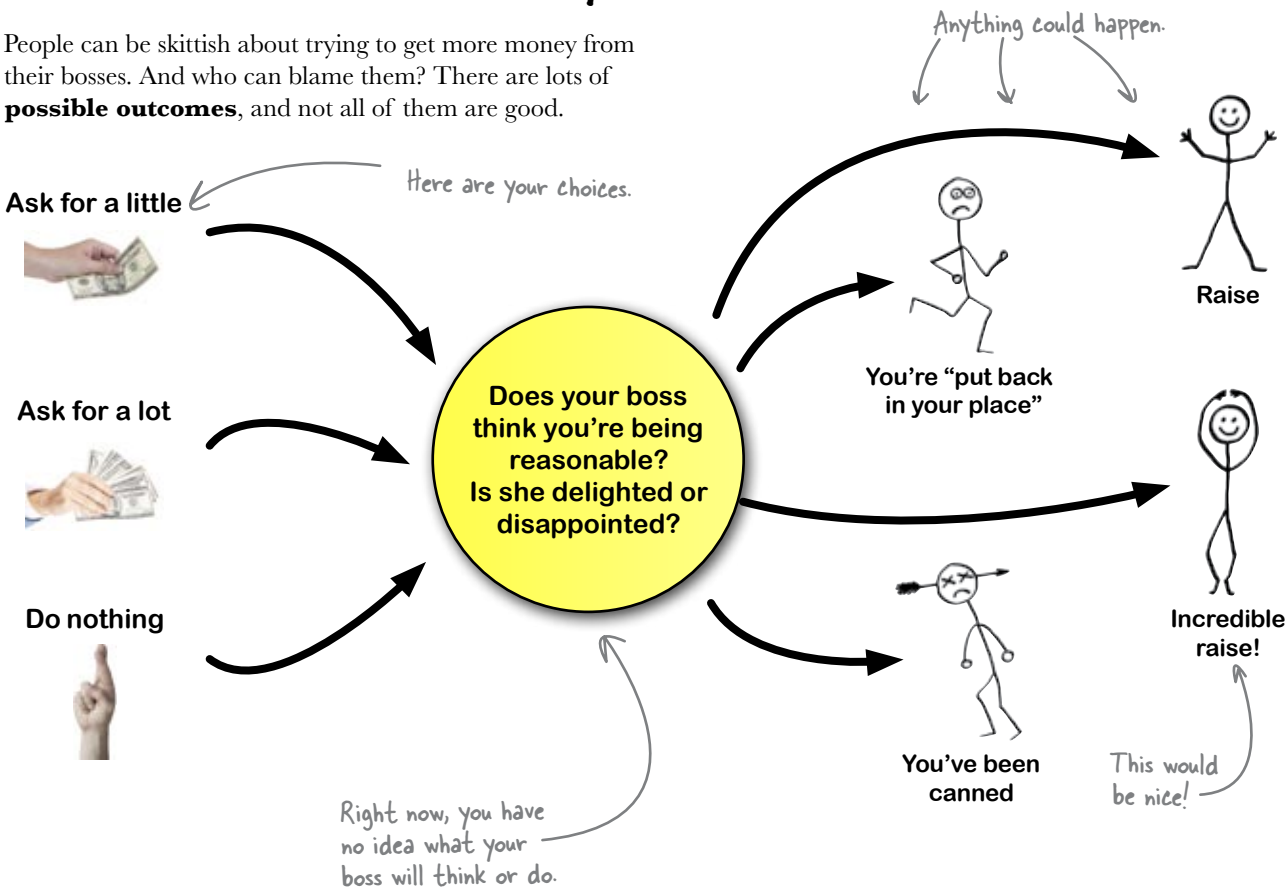
Can you envision a data-based way of deciding on how to deal with this situation?



However you answered the questions on the last page, we think you should go for more money. You're not doing this hard work for your health, after all.

Going for more cash could play out in a bunch of different ways

People can be skittish about trying to get more money from their bosses. And who can blame them? There are lots of **possible outcomes**, and not all of them are good.



Could research help you predict the outcomes?

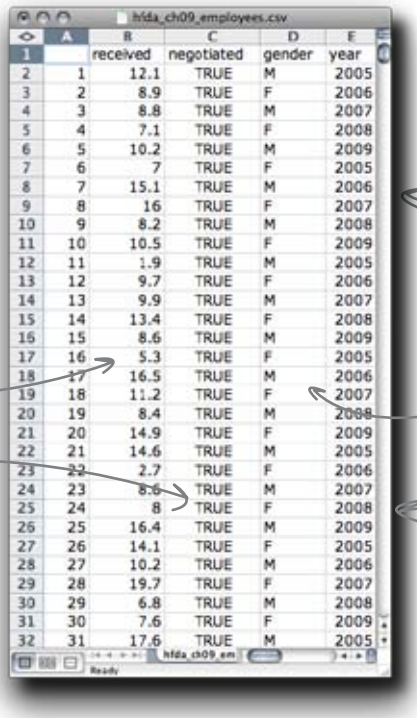
Even though your case is unique to you, it still might make sense to get an idea of your boss's **baseline expectations**.

Here's some data on raises

Because you're so plugged in to Starbuzz's data, you have access to some sweet numbers: Human Resource's records about raises for the past three years.

Load this!

www.headfirstlabs.com/books/hfda/
hfda_ch09_employees.csv



	A	B	C	D	E
1		received	negotiated	gender	year
2	1	12.1	TRUE	M	2005
3	2	8.9	TRUE	F	2006
4	3	8.8	TRUE	M	2007
5	4	7.1	TRUE	F	2008
6	5	10.2	TRUE	M	2009
7	6	7	TRUE	F	2005
8	7	15.1	TRUE	M	2006
9	8	16	TRUE	F	2007
10	9	8.2	TRUE	M	2008
11	10	10.5	TRUE	F	2009
12	11	1.9	TRUE	M	2005
13	12	9.7	TRUE	F	2006
14	13	9.9	TRUE	M	2007
15	14	13.4	TRUE	F	2008
16	15	8.6	TRUE	M	2009
17	16	5.3	TRUE	F	2005
18	17	16.5	TRUE	M	2006
19	18	11.2	TRUE	F	2007
20	19	8.4	TRUE	M	2008
21	20	14.9	TRUE	F	2009
22	21	14.6	TRUE	M	2005
23	22	2.7	TRUE	F	2006
24	23	8.6	TRUE	M	2007
25	24	8	TRUE	F	2008
26	25	16.4	TRUE	M	2009
27	26	14.1	TRUE	F	2005
28	27	10.2	TRUE	M	2006
29	28	19.7	TRUE	F	2007
30	29	6.8	TRUE	M	2008
31	30	7.6	TRUE	F	2009
32	31	17.6	TRUE	M	2005

Each line of the database represents someone's raise for the specified year.

Your company's raises

Here's the person's raise amount, measured as a percent increase.

This column says whether the person is male or female... you know, there might be a correlation between gender and raise amount.

This column tells you whether the person asked for more money or not. TRUE means they asked for more cash, FALSE means they didn't.

This data could be of use to you as you figure out what types of raises are reasonable to expect.

You might be able to wring some powerful insights out of this data. If you assume that your boss will act in a similar way to how previous bosses acted, this data could tell you what to expect.

Problem is, with approximately 3,000 employees, the data set is pretty **big**.

You're going to need to do something to make the data useful.

How would you deal with this data? Could you manage it to make it more useful?

Jim: We should forget about the data and just go for as much as we can get. Nothing in there will tell us what they think we're worth. There's a range of numbers in the boss's head, and we need to figure out how to get the upper end of that range.

Joe: I agree that most of the data is useless to tell us what they think *we* are worth, and I don't see how we find out. The data will tell us the average raise, and we can't go wrong shooting for the average.

Jim: The **average**? You've got to be joking. Why go for the middle? Aim higher!

Frank: I think a more subtle analysis is in order. There's some rich information here, and who knows what it'll tell us?

Joe: We need to stay risk-averse and follow the herd. The middle is where we find safety. Just average the `Raise` column and ask for that amount.

Jim: That's a complete cop-out!

Frank: Look, the data shows whether people negotiated, the year of the raise, and people's genders. All this information can be useful to us if we just massage it into the right format.

Jim: OK, smarty pants. *Show me.*

Frank: Not a problem. First we have to figure out how to collapse all these numbers into figures that make more sense…

Better summarize the data. There's just too much of it to read and understand all at once, and until you've summarized the data you don't really know what's in it.

Start by breaking the data down into its basic constituent pieces. Once you have those pieces, then you can look at averages or whatever other summary statistic you consider useful.

Where will you begin your summary of this data?

Sharpen your pencil

As you know, much of analysis consists of taking information and breaking it down into smaller, more manageable pieces.

Draw a picture to describe how you would break these data fields down into smaller elements.

Draw pictures here to represent how you'd split the data into smaller pieces.

Here are some examples.

Raises of 6–8%

Women

hfda_ch09_employees.csv

	A	B	C	D	E
1		received	negotiated	gender	year
2	1	12.1	TRUE	M	2005
3	2	8.9	TRUE	F	2006
4	3	8.8	TRUE	M	2007
5	4	7.1	TRUE	F	2008
6	5	10.2	TRUE	M	2009
7	6	7	TRUE	F	2005
8	7	15.1	TRUE	M	2006
9	8	16	TRUE	F	2007
10	9	8.2	TRUE	M	2008
11	10	10.5	TRUE	F	2009
12	11	1.9	TRUE	M	2005
13	12	9.7	TRUE	F	2006
14	13	9.9	TRUE	M	2007
15	14	13.4	TRUE	F	2008
16	15	8.6	TRUE	M	2009
17	16	5.3	TRUE	F	2005
18	17	16.5	TRUE	M	2006
19	18	11.2	TRUE	F	2007
20	19	8.4	TRUE	M	2008
21	20	14.9	TRUE	F	2009
22	21	14.6	TRUE	M	2005
23	22	2.7	TRUE	F	2006
24	23	8.6	TRUE	M	2007
25	24	8	TRUE	F	2008
26	25	16.4	TRUE	M	2009
27	26	14.1	TRUE	F	2005
28	27	10.2	TRUE	M	2006
29	28	19.7	TRUE	F	2007
30	29	6.8	TRUE	M	2008
31	30	7.6	TRUE	F	2009
32	31	17.6	TRUE	M	2005

What statistics could you use to summarize these elements? Sketch some tables that incorporate your data fields with summary statistics.

Sharpen your pencil Solution

What sort of pieces would you break your data into?

Here are some examples... your answers might be a little different.

You can break the data in your columns into pieces...

- Women
- Raises of 4–5%
- Men

...and you can combine those pieces of data with pieces from other columns.

- Female negotiators
- Men in 2008
- Raises of >10%
- Raises of 1% in '07

hrda_ch09_employees.csv

	A	B	C	D	E
1		received	negotiated	gender	year
2	1	12.1	TRUE	M	2005
3	2	8.9	TRUE	F	2006
4	3	8.8	TRUE	M	2007
5	4	7.1	TRUE	F	2008
6	5	10.2	TRUE	M	2009
7	6	7	TRUE	F	2005
8	7	15.1	TRUE	M	2006
9	8	16	TRUE	F	2007
10	9	8.2	TRUE	M	2008
11	10	10.5	TRUE	F	2009
12	11	1.9	TRUE	M	2005
13	12	9.7	TRUE	F	2006
14	13	9.9	TRUE	M	2007
15	14	13.4	TRUE	F	2008
16	15	8.6	TRUE	M	2009
17	16	5.3	TRUE	F	2005
18	17	16.5	TRUE	M	2006
19	18	11.2	TRUE	F	2007
20	19	8.4	TRUE	M	2008
21	20	14.9	TRUE	F	2009
22	21	14.6	TRUE	M	2005
23	22	2.7	TRUE	F	2006
24	23	8.6	TRUE	M	2007
25	24	8	TRUE	F	2008
26	25	16.4	TRUE	M	2009
27	26	14.1	TRUE	F	2005
28	27	10.2	TRUE	M	2006
29	28	19.7	TRUE	F	2007
30	29	6.8	TRUE	M	2008
31	30	7.6	TRUE	F	2009
32	31	17.6	TRUE	M	2005

This chart counts the number of raises in each interval of raise possibilities.

Here are a few ways you might integrate your data chunks with summary statistics.

You've got loads of options here.

This table shows the average raise for male and female negotiators.

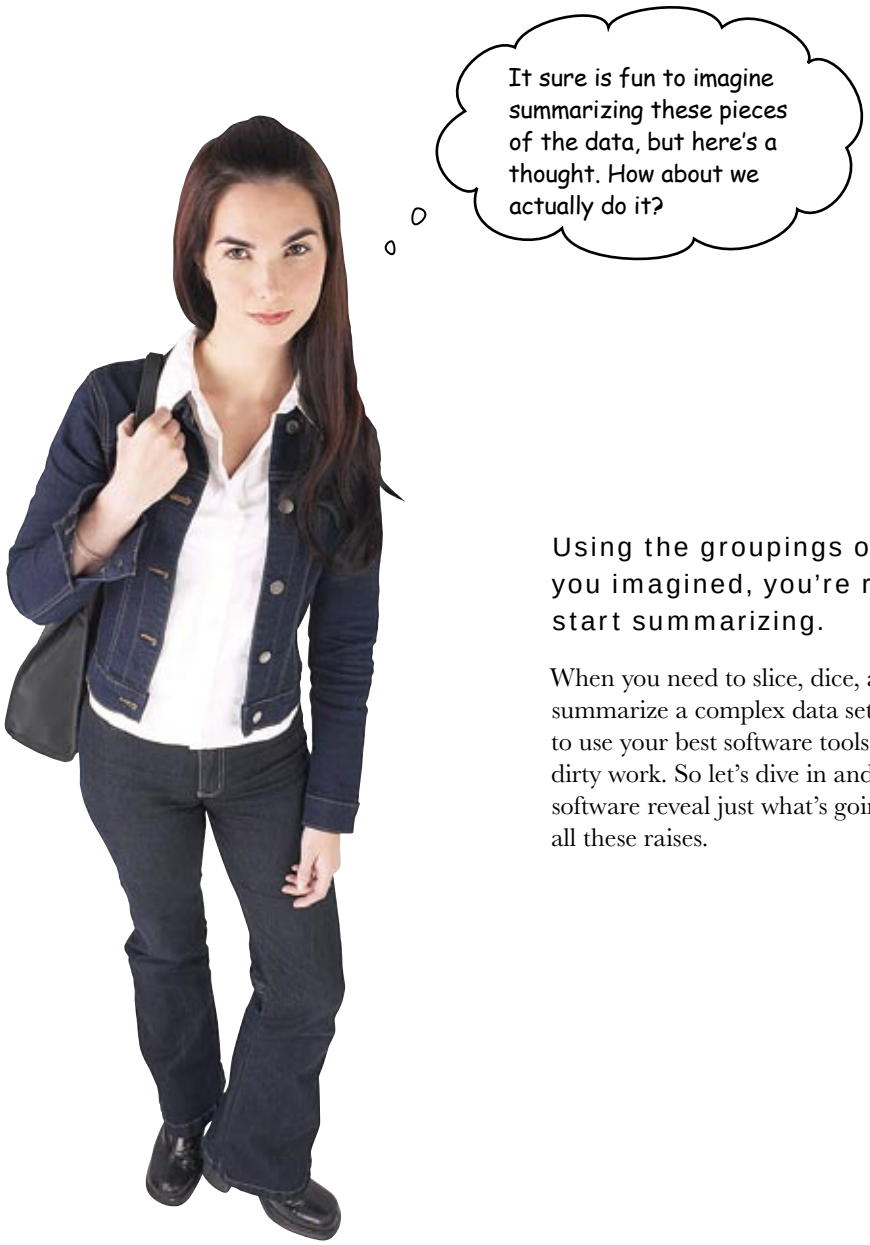
	Men	Women
Negotiators	Average raise.	Average raise.

- 0–1%
- 1–2%
- 3–4%
- 4–5%
- 5–6%
- 6–7+%

Frequencies (or counts) of raises

How many people had a raise between 6 and 7 percent?

Let's actually create this last visualization...



Using the groupings of data you imagined, you're ready to start summarizing.

When you need to slice, dice, and summarize a complex data set, you want to use your best software tools to do the dirty work. So let's dive in and make your software reveal just what's going on with all these raises.

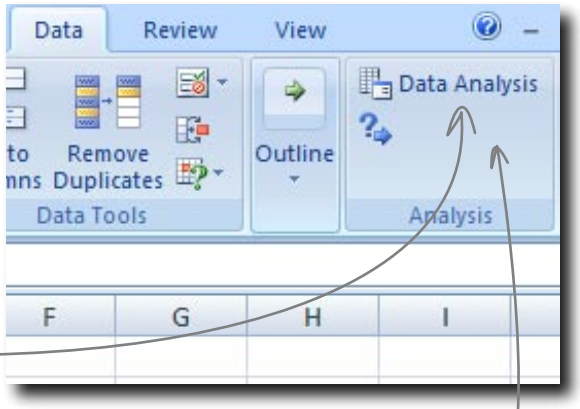
Test Drive

A visualization of the number of people who fall in each category of raises will enable you to *see* the whole data set at once.

So let's create that summary… or even better, let's do it **graphically**.

1. Open the Data Analysis dialogue box.

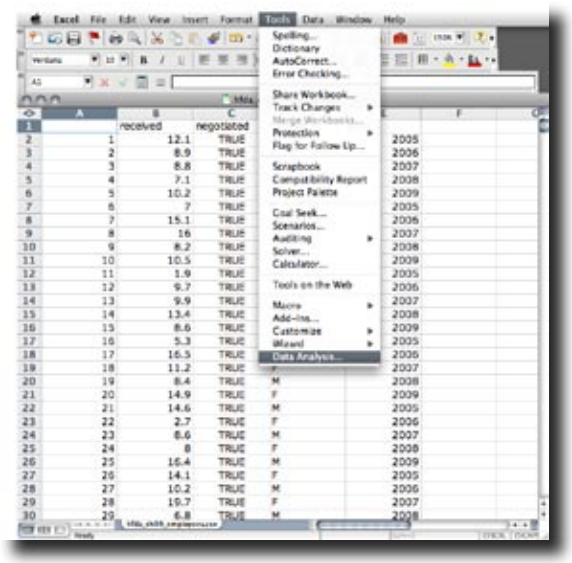
With your data open in Excel, click the Data Analysis button under the Data tab.



If you don't see the Data Analysis button, see Appendix iii for how to install it.

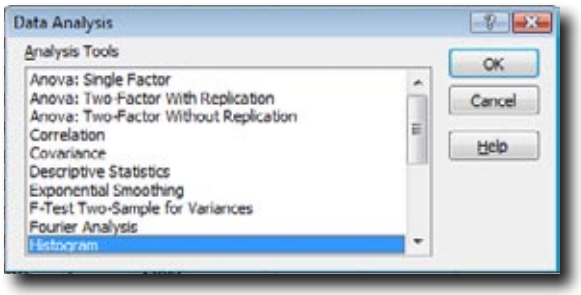
In OpenOffice and older versions of Excel, you can find Data Analysis under the Tools menu.

Here it is!



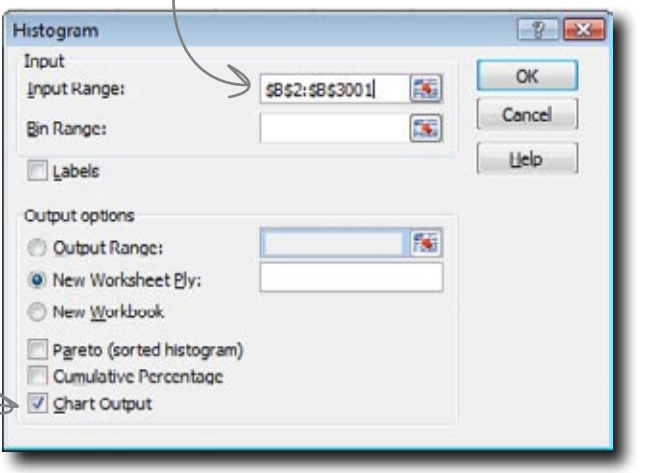
2. Select Histogram.

In the pop-up window, tell Excel that you want to create a histogram.

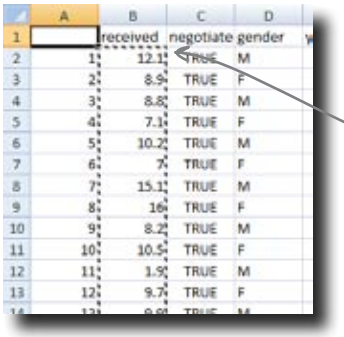


3 Select your data.

Select all your raise data under the `received` column.

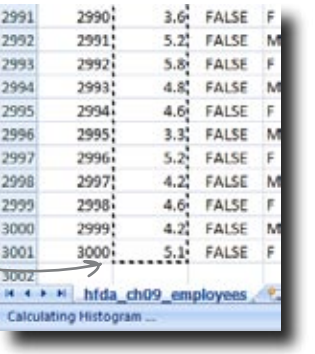


Be sure to check this box so that Excel makes a chart.



When your raise data is selected, there should be a big "marching ants" selection box all the way from the top...

...to the bottom.



4 **Run it.**

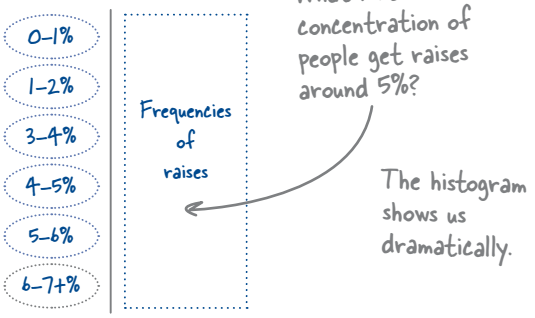
Press OK and let 'er rip!

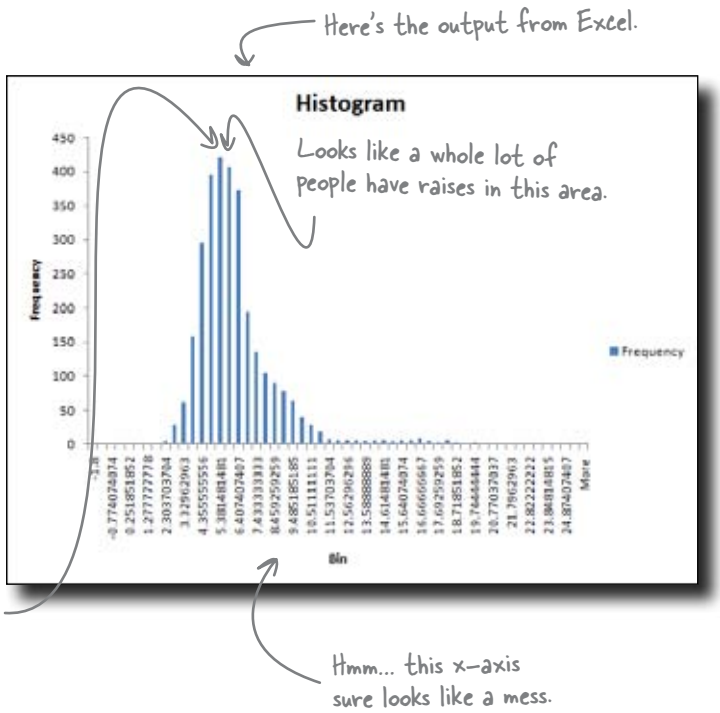
What happens when you create the chart?

Histograms show frequencies of groups of numbers

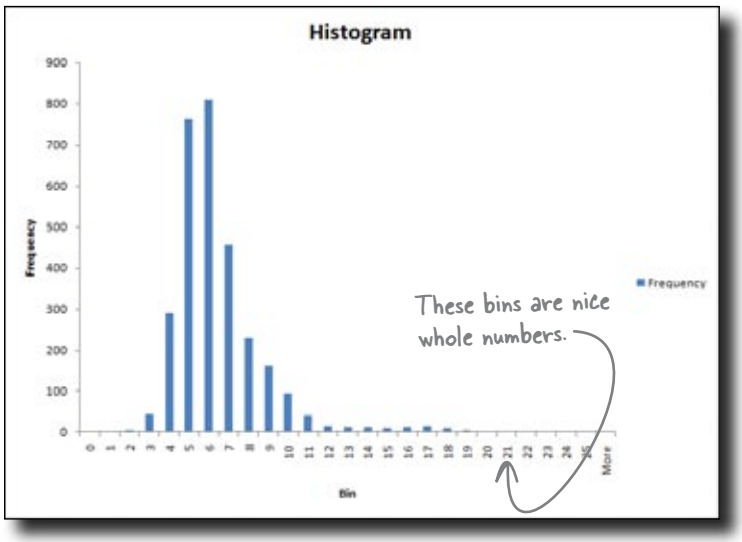
Histograms are a powerful visualization because, no matter how large your data set is, they show you the **distribution** of data points across their range of values.

For example, the table you envisioned in the last exercise would have told you how many people received raises at about 5 percent.





This histogram shows graphically how many people fall into each raise category, and it concisely shows you what people are getting across the spectrum of raises.



On the other hand, there are some problems with what Excel did for you. The default settings for the **bins** (or "class intervals") end up producing messy, noisy x-axis values. The graph would be easier to read with plain integers (rather than long decimals) on the x-axis to represent the bins.

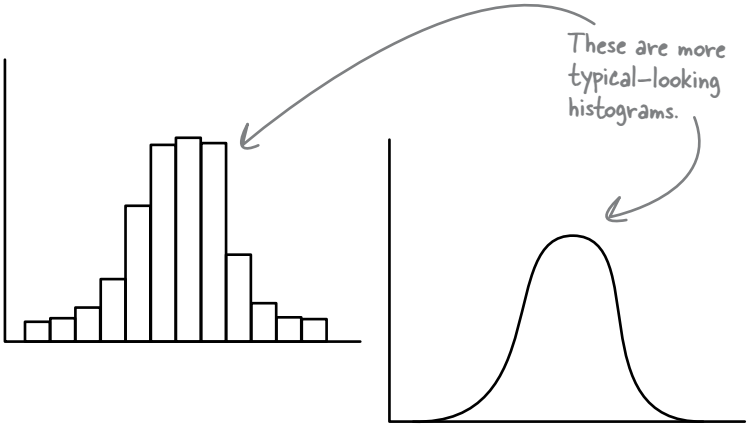
Sure, you *can* tweak the settings to get those bins looking more like the data table you initially envisioned.

But even this histogram has a serious problem. Can you spot it?

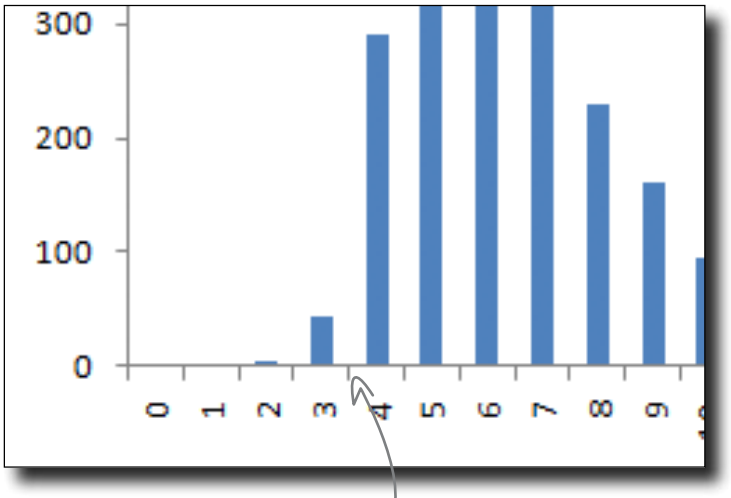
Gaps between bars in a histogram mean gaps among the data points

In histograms, gaps mean that there is data missing between certain ranges. If, say, no one got a raise between 5.75 percent and 6.25 percent, there might be a gap. If the histogram showed that, it might really be worth investigating.

In fact, there will always be gaps if there are more bins than data points (unless your data set is the same number repeated over and over).



Histograms Up Close



Does this gap mean that there are no people who got raises between 3.3% and 3.8%?

That's exactly what the gap *should* mean, at least if the histogram is written correctly. If you assumed this histogram was correct, and that there were gaps between these values, you'd get the totally wrong idea. You need a software tool to create a better histogram.

The problem with Excel's function is that it creates these messy, artificial breaks that are really deceptive.

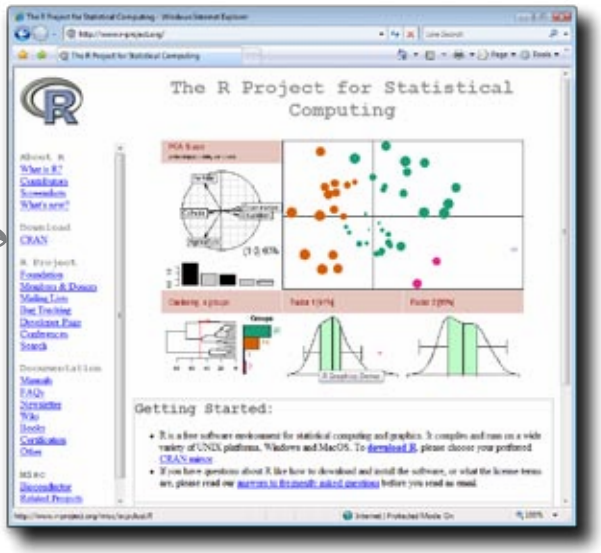
And there's a technical workaround for the issues (with Excel, there's almost always a workaround if you have the time to write code using Microsoft's proprietary language).

But it's chapter 9, and you've been kicking serious butt. You're ready for a **software tool** with more power than Excel to manage and manipulate statistics.

The software you need is called **R**. It's a free, open source program that might be the future of statistical computing, and you're about to dive into it!

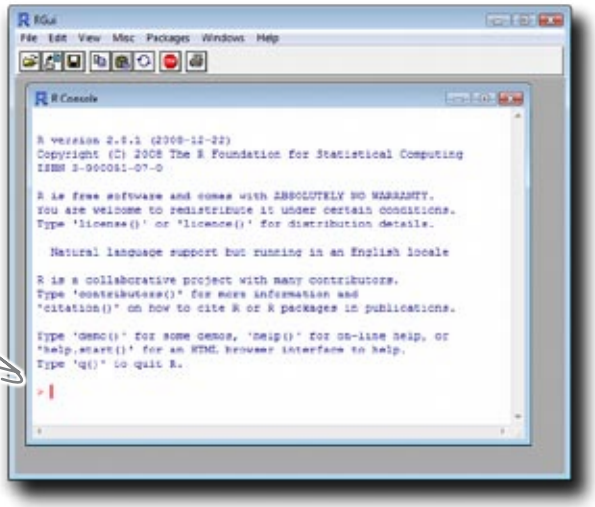
Install and run R

Head on over to ***www.r-project.org*** to download R. You should have no problem finding a mirror near you that serves R for Windows, Mac, and Linux.



Click this download link.

Once you've fired up the program, you'll see a window that looks like this.



This little cursor here represents the command prompt and is where you'll be entering your commands into R.

The command prompt is your friend.

Working from the command prompt is something you get the hang of quickly, even though it requires you to think a little harder at first. And you can always pull up a spreadsheet-style visualization of your data by typing `edit(yourdata)`.

Load data into R

For your first R command, try loading the *Head First Data Analysis* script using the `source` command:

Load this!

```
source("http://www.headfirstlabs.com/books/hfda/hfda.R")
```

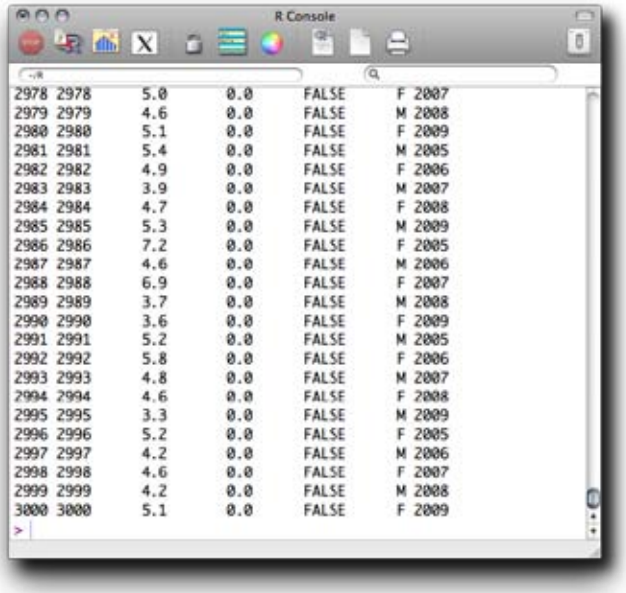
That command will load the raise data you need for R. You'll need to be connected to the Internet for it to work. If you want to save your R session so that you can come back to the *Head First* data when you're not connected to the Internet you can type `save.image()`.

So what did you download? First, take a look at the **data frame** from your download called "Employees." Just type this command and press Enter:

```
employees
```

Type the name of the data frame to get R to display it.

The output you see on the right is what R gives you in response.



The command returns a listing of all the rows in the data frame.

Generate a histogram in R by typing this command:

```
hist(employees$received, breaks=50)
```

What does this mean?

What do you think the various elements of the command mean? Annotate your response.

What do you think this histogram command means?

`hist` tells R to run the histogram function.

The second argument tells R how to construct the groupings.

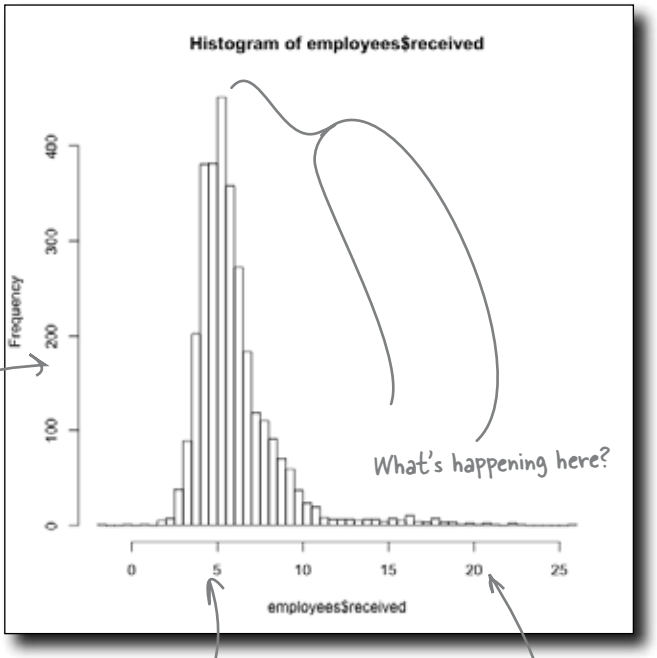
```
hist(employees$received, breaks=50)
```

The first argument specifies what data to use.

R creates beautiful histograms

With histograms, the areas under the bars don't just measure the count (or **frequency**) of the thing being measured; they also show the percentage of the entire data set being represented by individual segments.

When you run the command, a window pops up showing this.



The frequency here is the count of people.

What's happening here?

A lot of people got a raise around 5%.

Here are the highest percentage raises.

Look carefully at the contour of the curve. A few things are obvious. Not a lot of people got raises below 0 percent, and not a lot of people got raises above 22 percent.

But what's happening in the *middle* of the distribution?

What do you make of this histogram?

These commands will tell you a little more about your data set and **what people's raises look like**. What happens when you run the commands?

```
sd(employees$received)

summary(employees$received)
```

Why do you think R responds to each of these the way it does?

Type `help(sd)` and `help(summary)` to find out what the commands do.

What do the two commands do?

.....

.....

.....

Look closely at the histogram. How does what you see on the histogram compare with what R tells you from these two commands?

.....

.....

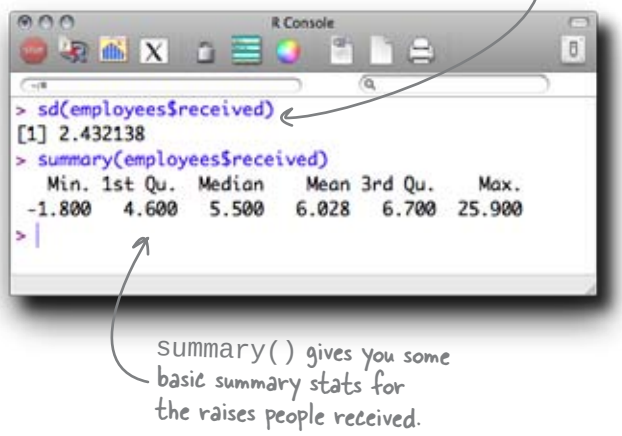
.....

Exercise Solution

You just ran some commands to illustrate the summary statistics for your data set about raises. What do you think these commands did?

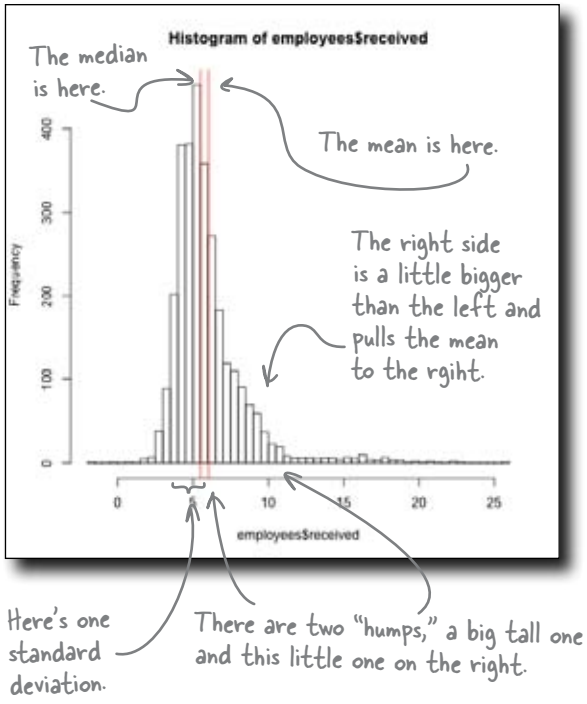
What do the two commands do?

The `sd` command returns the standard deviation of the data range you specify, and the `summary()` command shows you summary statistics about the `received` column.



Look closely at the histogram. How does what you see on the histogram compare with what R tells you from these two commands?

The histogram does a good job of visually showing mean, median, and standard deviation. Looking at it, you can't see the exact figures, but you can get a sense of those numbers by looking at the shape of the curve.



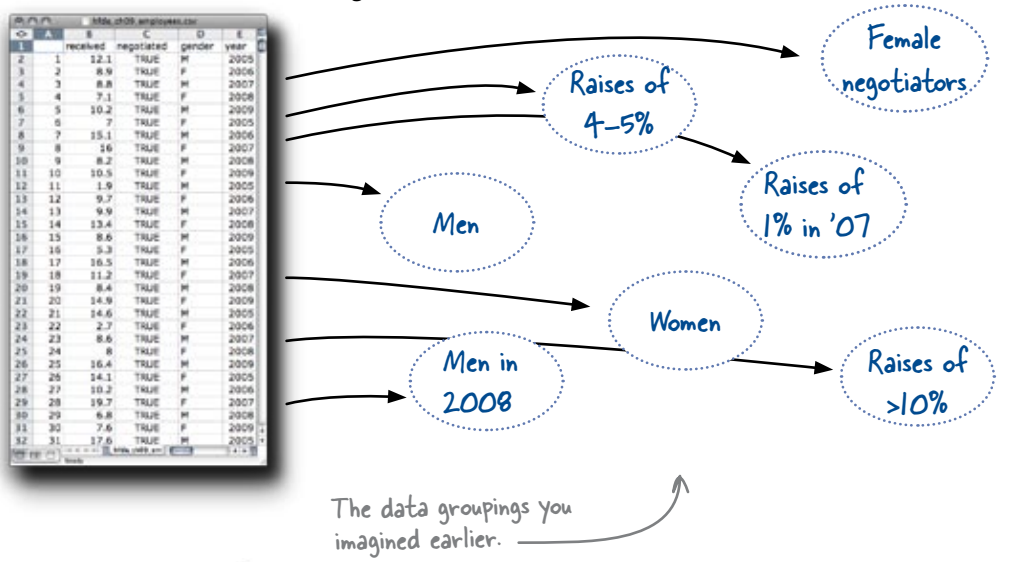
Joe: If the histogram were symmetrical, the mean and median would be in the same place—in the dead center.

Frank: Right. But in this case, the small hump on the right side is pulling the mean away from the center of the larger hump, where most of the observations are.

Joe: I'm struggling with those two humps. What do they ***mean***?

Frank: Maybe we should take another look at those pieces of data we identified earlier and see if they have any relevance to the histogram.

Joe: Good idea.



Sharpen your pencil

Can you think of any ways that the groups you identified earlier might explain the two humps on the histogram?

.....

.....

.....

How might the groupings of data you identified earlier account for the two humps on your histogram?

There could be variation among years: for example, raises in 2007 could be on average much higher than raises from 2006. And there could be gender variation, too: men could, on average, get higher raises than women, or vice versa. Of course, all the data is observational, so any relationships you discover won't necessarily be as strong as what experimental data would show.

there are no Dumb Questions

Q: So it seems like we have a lot of flexibility when it comes to how the histograms look.

A: It's true. You should think of the very act of creating a histogram as an interpretation, not something you do *before* interpretation.

Q: Are the defaults that R uses for creating a histogram generally good?

A: Generally, yes. R tries to figure out the number of breaks and the scale that will best represent the data, but R doesn't *understand* the meaning of the data it's plotting. Just as with the summary functions, there's nothing wrong with running a quick and dirty histogram to see what's there, but before you draw any big conclusions about what you see, you need to use the histogram (and redraw the histogram) in a way that remains mindful of what you're looking at and what you hope to gain from your analysis.

Q: Are either of those humps the "bell curve?"

A: That's a great question. Usually, when we think of bell curves, we're talking about the normal or Gaussian distribution. But there are other types of bell-shaped distributions, and a lot of other types of distributions that aren't shaped like a bell.

Q: Then what's the big deal about the normal distribution?

A: A lot of powerful and simple statistics can come into play if your data is normally distributed, and a lot of natural and business data follows a natural distribution (or can be "transformed" in a way that makes it naturally distributed).

Q: So is our data normally distributed?

A: The histogram you've been evaluating is definitely not normally distributed. As long as there's more than one hump, there's no way you can call the distribution bell-shaped.

Q: But there are definitely two humps in the data that look like bells!

A: And that shape must have some sort of meaning. The question is, why is the distribution shaped that way? How will you find out?

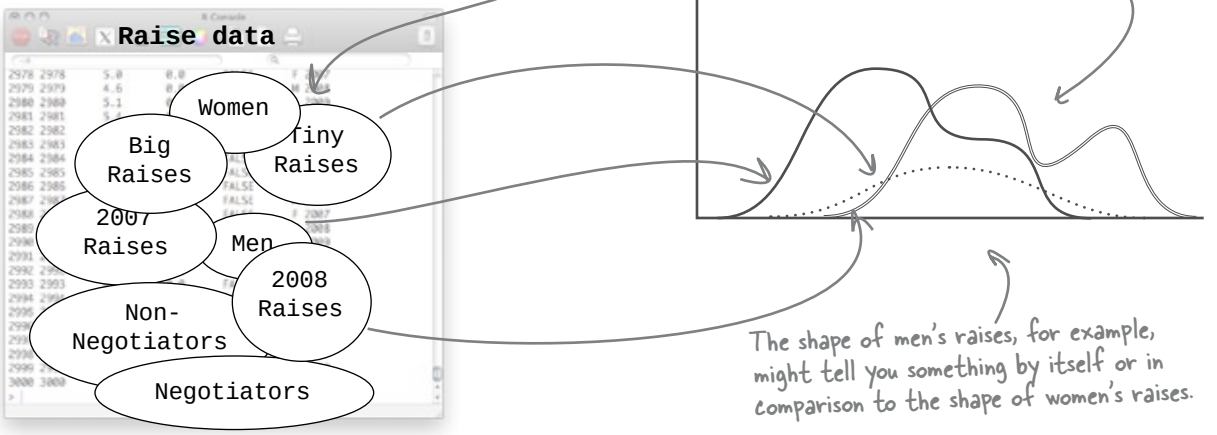
Q: Can you draw histograms to represent small portions of the data to evaluate individually? If we do that, we might be able to figure out why there are two humps.

A: That's the right intuition. Give it a shot!

Can you break the raise data down in a way that isolates the two humps and explains why they exist?

Make histograms from subsets of your data

You can make a histogram out of your entire data set, but you can also split up the data into subsets to make other histograms.



Let's make a bunch of histograms that describe subsets of the raise data. Maybe looking at these other histograms will help you figure out what the two humps on the raise histogram mean. Is there a group of people who are earning more in raises than the rest?

1) To start, look at this histogram command and annotate its syntax. What do you think its components mean?

```
hist(employees$received[employees$year == 2007], breaks = 50)
```

Write down here what you think each piece means.

2) Run the above command each of these commands. What do you see? The results are on the next page, where you'll write down your interpretations.

```
hist(employees$received[employees$year == 2008], breaks = 50)
hist(employees$received[employees$gender == "F"], breaks = 50)
hist(employees$received[employees$gender == "M"], breaks = 50)
hist(employees$received[employees$negotiated == FALSE], breaks = 50)
hist(employees$received[employees$negotiated == TRUE], breaks = 50)
```

Long Exercise

These histograms represent the raises for different subgroups of your employee population. What do they tell you?

The `hist()` command makes a histogram.

`received` is the set of values you want plotted in the histogram.

Breaks are the number of bars in the histogram.

```
hist(employees$received[employees$year == 2007], breaks = 50)
```

These brackets are the subset operator, which extracts a subset of your data.

In this case, you're extracting records where the year is 2007.

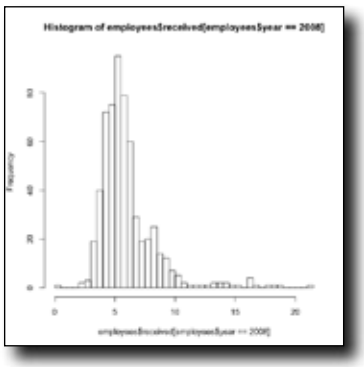
.....

.....

.....

.....

```
hist(employees$received[employees$year == 2008],
 breaks = 50)
```



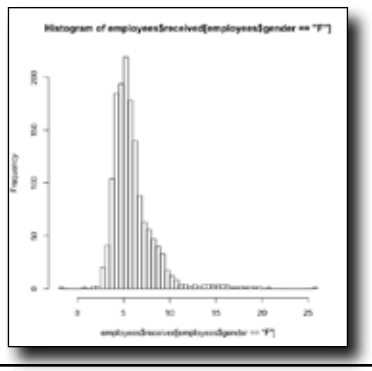
.....

.....

.....

.....

```
hist(employees$received[employees$gender == "F"],
 breaks = 50)
```



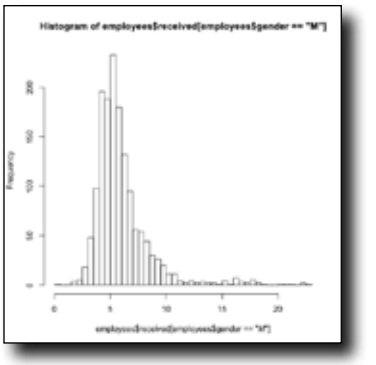
.....

.....

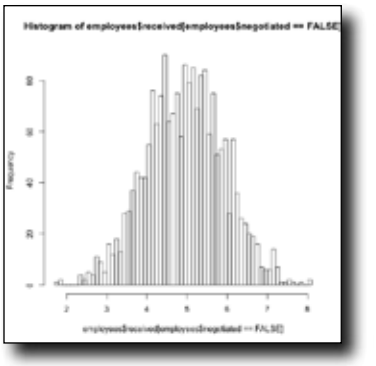
.....

.....

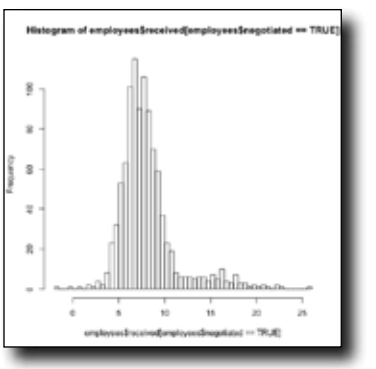
```
hist(employees$received[employees$gender == "M"],
 breaks = 50)
```




```
hist(employees$received[employees$negotiated == FALSE],
 breaks = 50)
```




```
hist(employees$received[employees$negotiated == TRUE],
 breaks = 50)
```



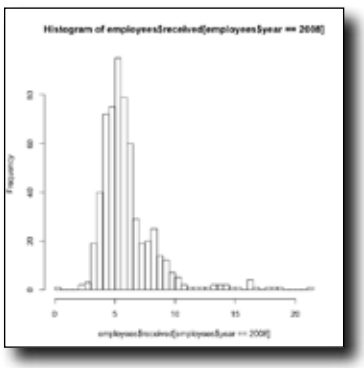
Long Exercise Solution

You looked at the different histograms in search of answers to help you understand who is getting what raises. What did you see?

```
hist(employees$received[employees$year == 2007],
 breaks = 50)
```

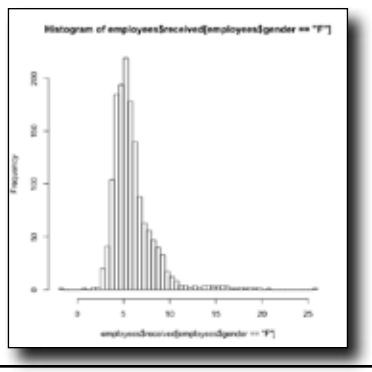
This histogram selects only raises for 2007 and has the same basic shape as the original histogram. The scale is different—e.g., only 8 people are in the largest break here. But the shape is the same, and the 2007 group might have the same characteristics as the overall group.

```
hist(employees$received[employees$year == 2008],
 breaks = 50)
```



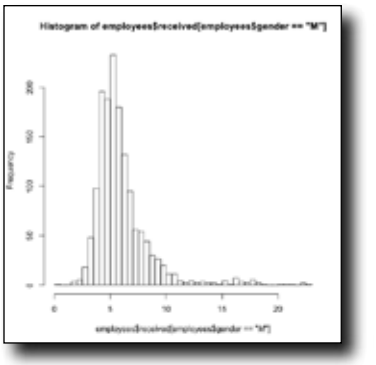
There's the exact same thing going on here as we see with the 2007 data. R even chose to plot the data using the exact same scale. At least as far as this data is concerned, 2007 and 2008 are pretty similar.

```
hist(employees$received[employees$gender == "F"],
 breaks = 50)
```



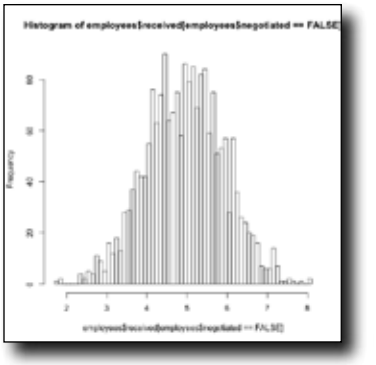
Once again, we see the big hump and the little hump attached on the right, although the scale is different on this histogram. This graph shows raises earned by women by all the years represented in the data, so there's a lot of them.

```
hist(employees$received[employees$gender == "M"],
 breaks = 50)
```



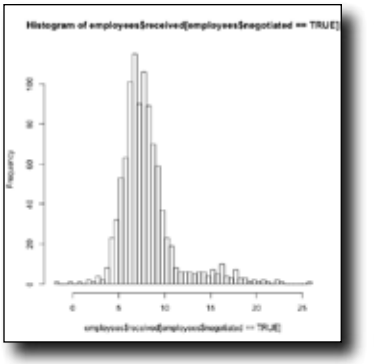
This looks a lot like the histogram for females. The scale is different, but when you count the bars, it looks like there are roughly the same number of men as women in the different categories. As usual, there are two humps.

```
hist(employees$received[employees$negotiated == FALSE],
 breaks = 50)
```



Now here's something interesting: just one hump. And the horizontal scale shows that these people—the ones who did not negotiate their raises—are on the low end of the raise range. And there are a lot of them, as you can see from the vertical scale.

```
hist(employees$received[employees$negotiated == TRUE],
 breaks = 50)
```

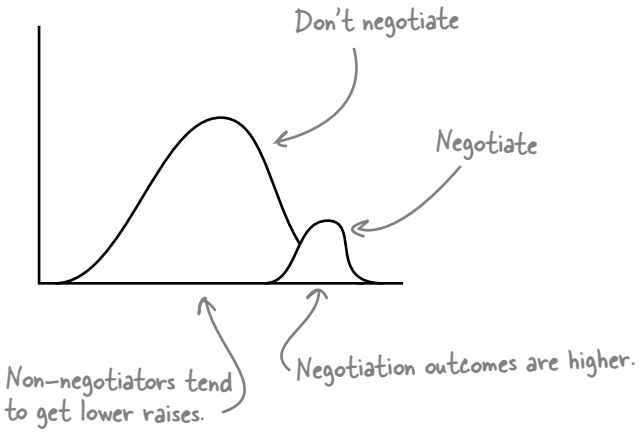


It looks like splitting those who did and did not negotiate neatly separates the two humps. Here we see people earning a lot more in raises, and there are far fewer people. It looks like negotiating for a raise gives people a completely different outcome distribution.

Negotiation pays

Your analysis of histograms of different subsets of the raise data shows that getting a larger raise is all about *negotiation*.

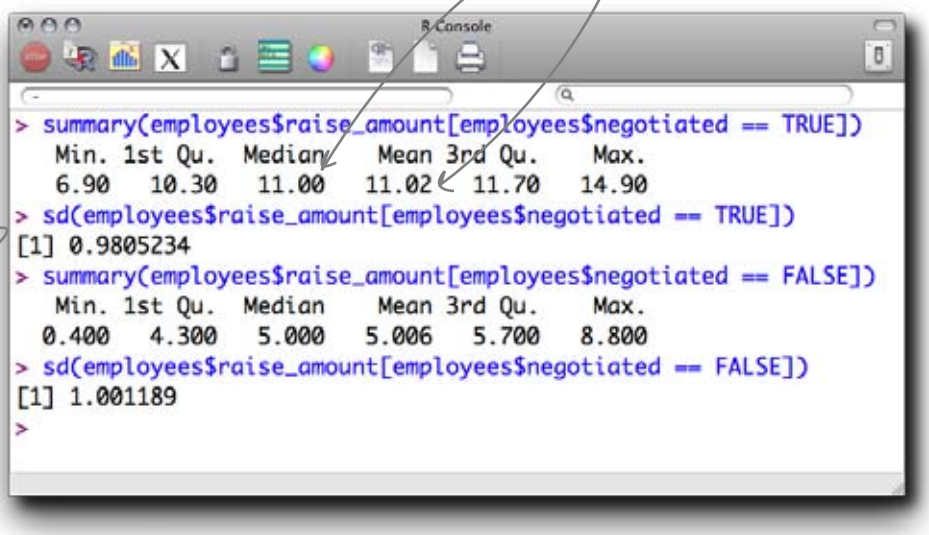
People have a different **spread of outcomes** depending on their choice of whether to negotiate. If they do, their whole histogram shifts to the right.



If you run the summary statistics on your negotiation subsets, the results are just as dramatic as what you see with the two curves.

This is the function that calculates the standard deviation.

The mean and median are about the same within each distribution.

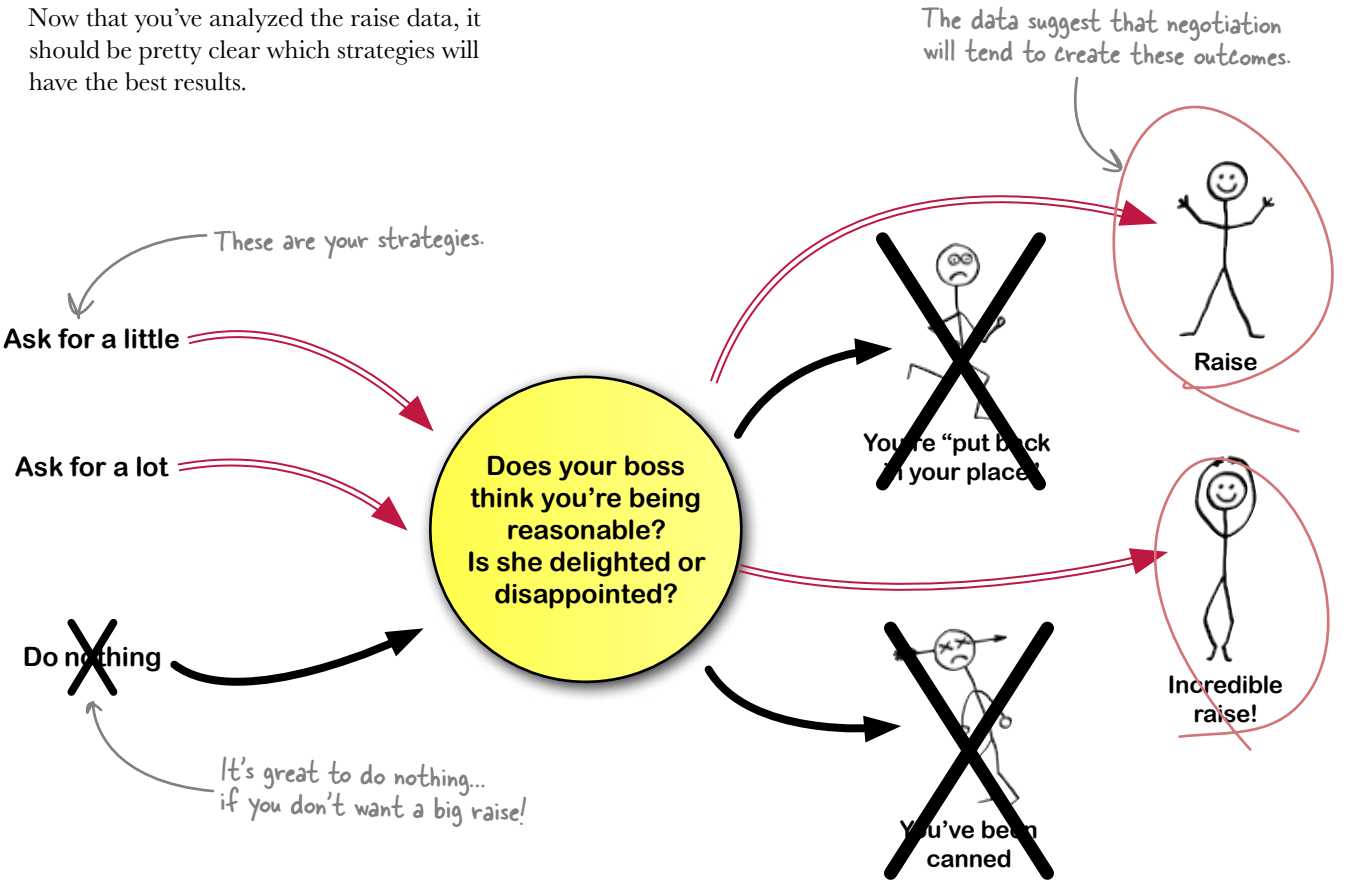


On average, for both distributions, data points are within a single percentage point of the mean.

You should definitely negotiate your salary.

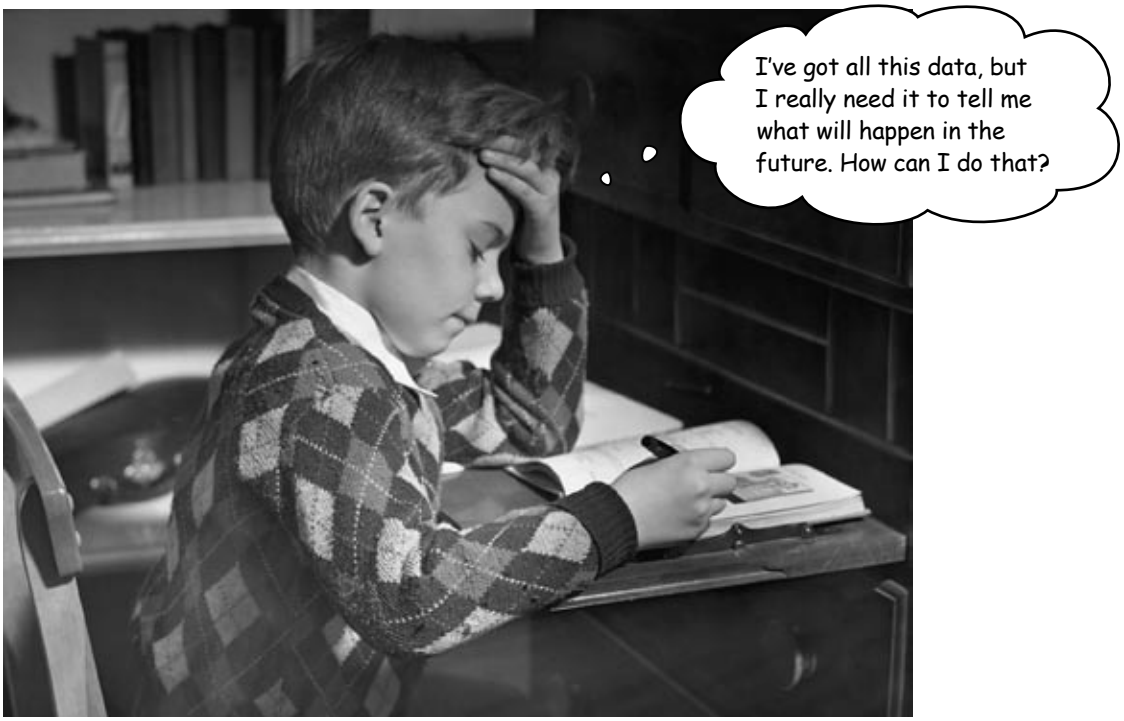
What will negotiation mean for you?

Now that you've analyzed the raise data, it should be pretty clear which strategies will have the best results.



10 regression

Prediction



Predict it.

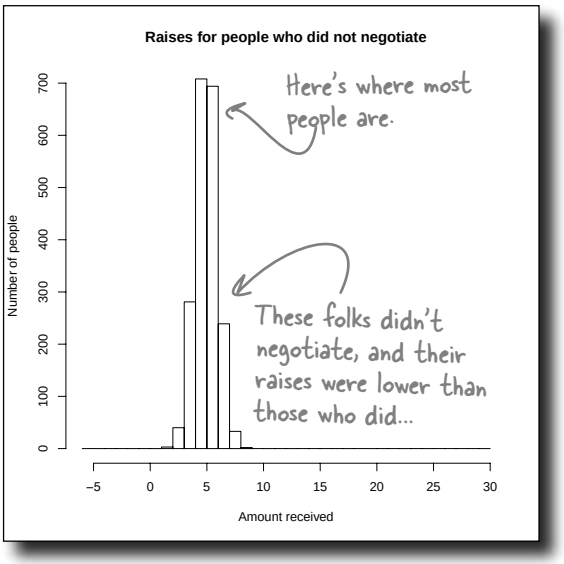
Regression is an incredibly powerful statistical tool that, when used correctly, has the ability to help you predict certain values. When used with a controlled experiment, regression can actually help you predict the future. Businesses use it like crazy to help them build models to explain customer behavior. You're about to see that the judicious use of regression can be very profitable indeed.

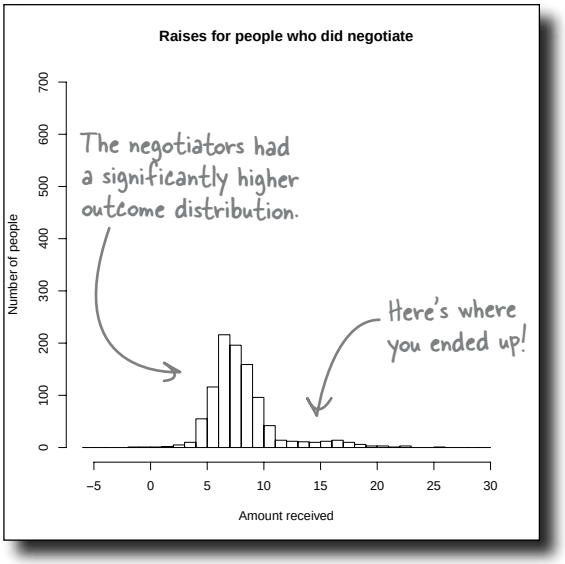
What are you going to do with all this money?

Your quest for a raise paid off. With your histograms, you figured out that people who chose to negotiate their salaries got consistently higher outcomes. So when you went into your boss's office, you had the confidence that you were pursuing a strategy that tended to pay off, and it did!

These are the histograms you looked at in the final exercises of the previous chapter, except they've been redrawn to show the same scale and bin size.

Nice work!





No point in stopping now.

Lots of people could benefit from your insight about how to get better raises. Few of your colleagues took the savvy approach your did, and you have a lot to offer those who didn't.

You should set up a business that specializes in getting people raises!

Here are a few questions to get you thinking about data-based ways of creating a business around your insights in salary negotiations.

What do you think your clients would want from a business that helps them understand how to negotiate raises?

If you ran such a business, what would be a fair way to compensate you for your knowledge?

Sharpen your pencil Solution

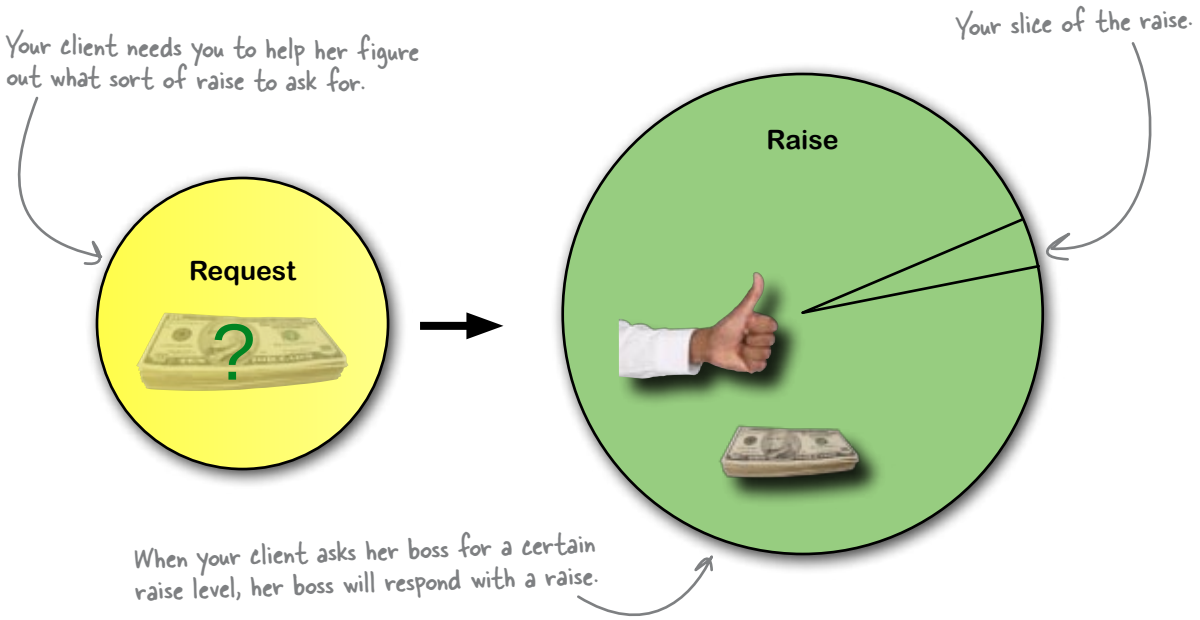
What sort of data-based compensation consulting business do you envision?

What do you think your clients would want from a business that helps them understand how to negotiate raises?

There are all sorts of ways that people negotiating for a raise could be helped: they might want to know how to dress, how to think about the issue from the perspective of their boss, what words will soften people up, and so forth. But one question is fundamental: how much do I ask for?

If you ran such a business, what would be a fair way to compensate you for your knowledge?

Clients will want you to have an incentive to make sure that their experience works out well. So why not charge them a percentage of what they actually get when they use your advice? That way, your incentive is to get them the biggest raise you can get them, not to waste their time.



An analysis that tells people what to ask for could be huge

What amount of money is reasonable to ask for? How will a request for a raise translate into an actual raise? Most people just don't know.

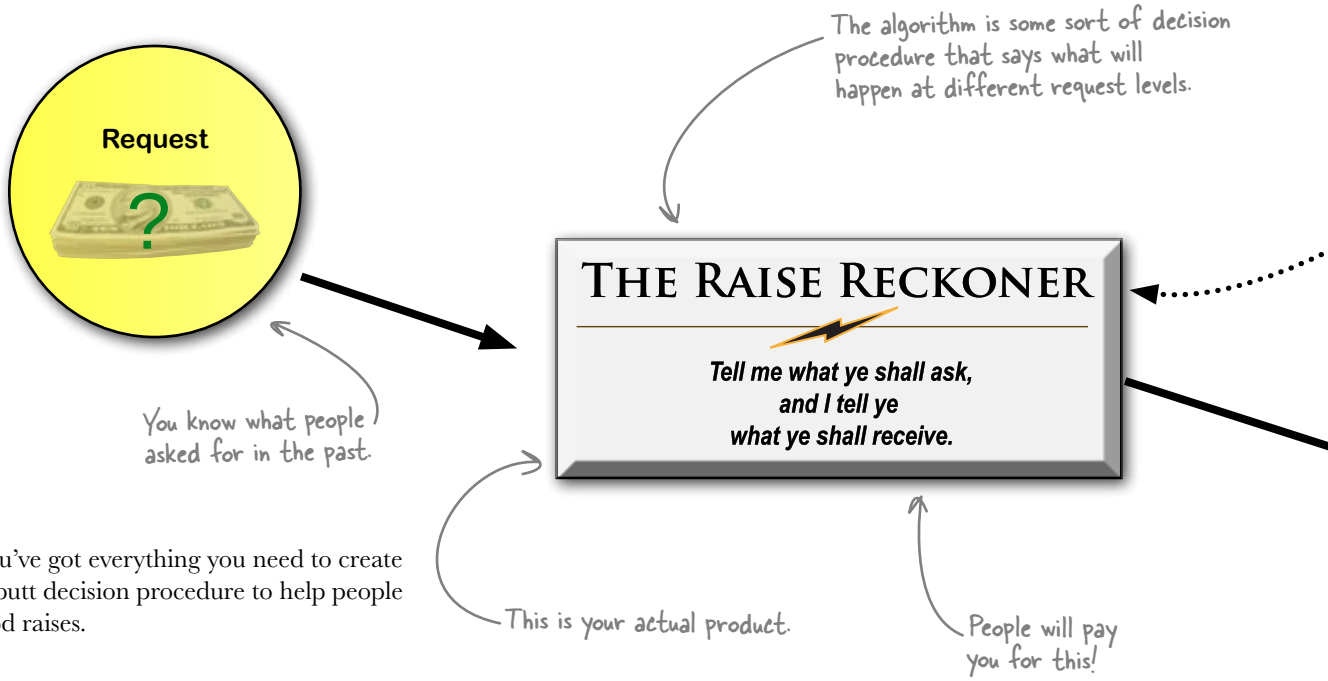
BRAIN POWER

You need a basic outline of your service so you know what you're shooting for. What will your product look like?

Behold... the Raise Reckoner!

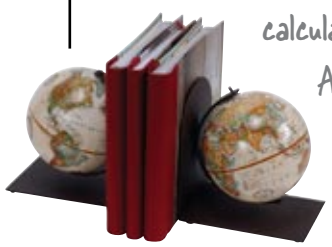
People want to know what to ask for. And they want to know what they'll get, given what they've asked for.

You need an **algorithm**.

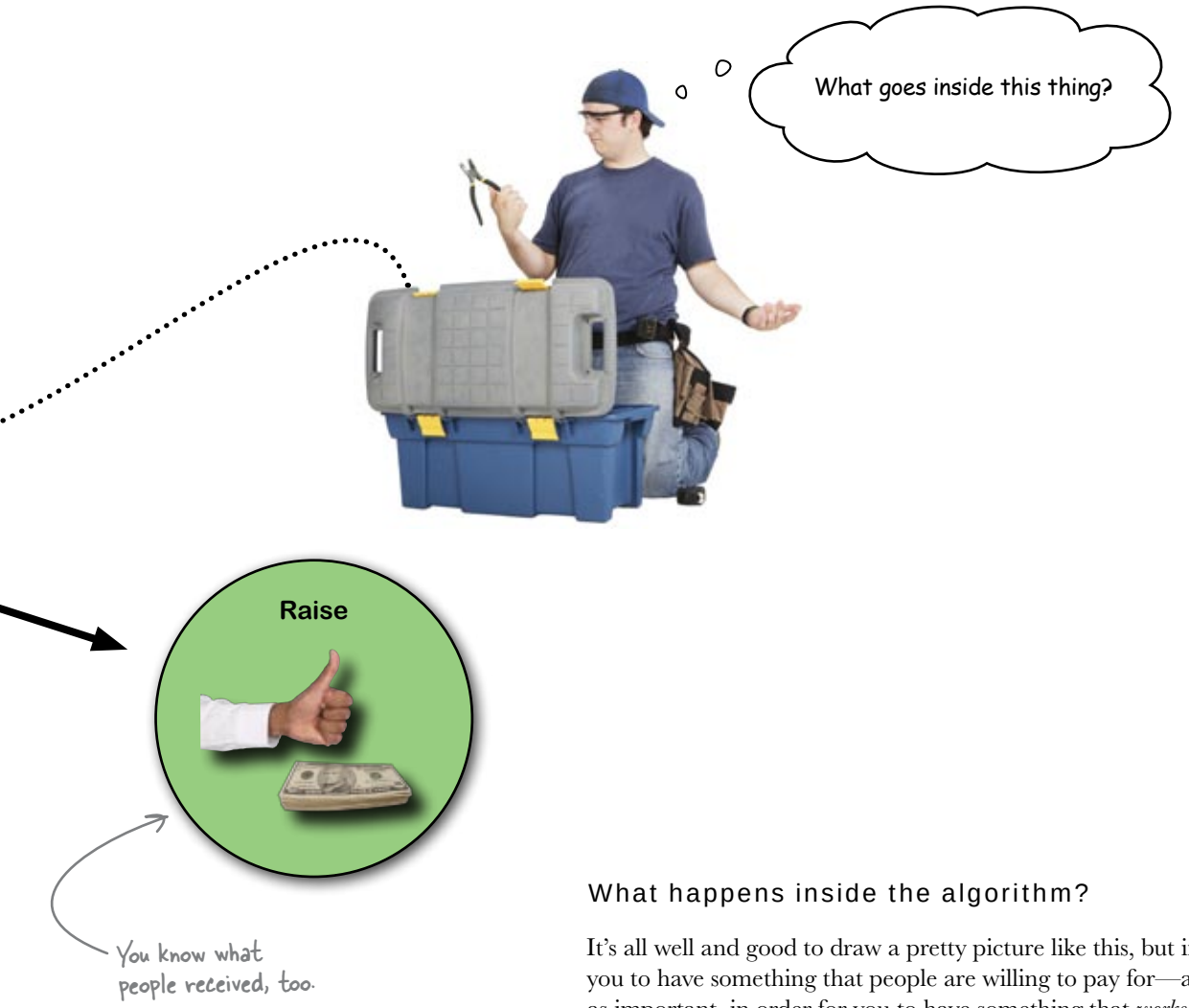


And you've got everything you need to create a kick-butt decision procedure to help people get good raises.

the Scholar's Corner



Algorithm Any procedure you follow to complete a calculation. Here, you'll take the input to the algorithm, the Amount Requested, and perform some steps in order to predict the Amount rewarded. But what steps?



What happens inside the algorithm?

It's all well and good to draw a pretty picture like this, but in order for you to have something that people are willing to pay for—and, just as important, in order for you to have something that *works*—you're going to need to do a serious analysis.

So what do you think goes inside?

Inside the algorithm will be a method to predict raises

Prediction is a big deal for data analysis. Some would argue that, speaking generally, **hypothesis testing** and **prediction** together are the *definition* of data analysis.

BULLET POINTS

Things you might need to predict:

- People's actions
- Market movements
- Important events
- Experimental results
- Stuff that's not in your data

Questions you should always ask:

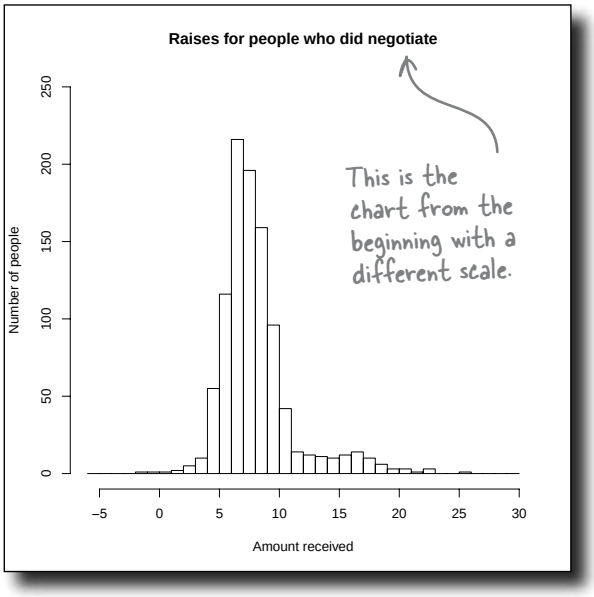
- Do I have enough data to predict?
- How good is my prediction?
- Is it qualitative or quantitative?
- Is my client using the prediction well?
- What are the limits of my prediction?

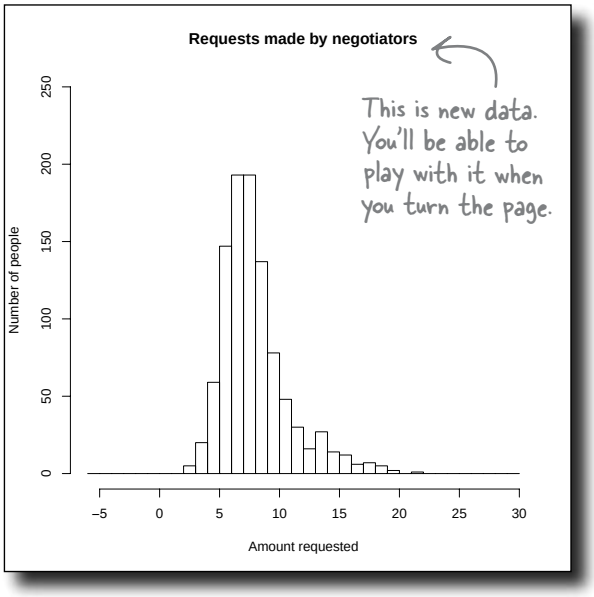
Let's take a look at some data about what negotiators asked for. Can you ***predict*** what sort of raise you'll get at various levels of requests?

Sharpen your pencil

The histograms below describe the amount of money the negotiators received and the amount of money they ***requested***.

Do the histograms tell you what people *should* request in order to get a big raise? Explain how comparing the two histograms might illuminate the relationship between these two variables, so that you might be able to predict how much you would receive for any given request.

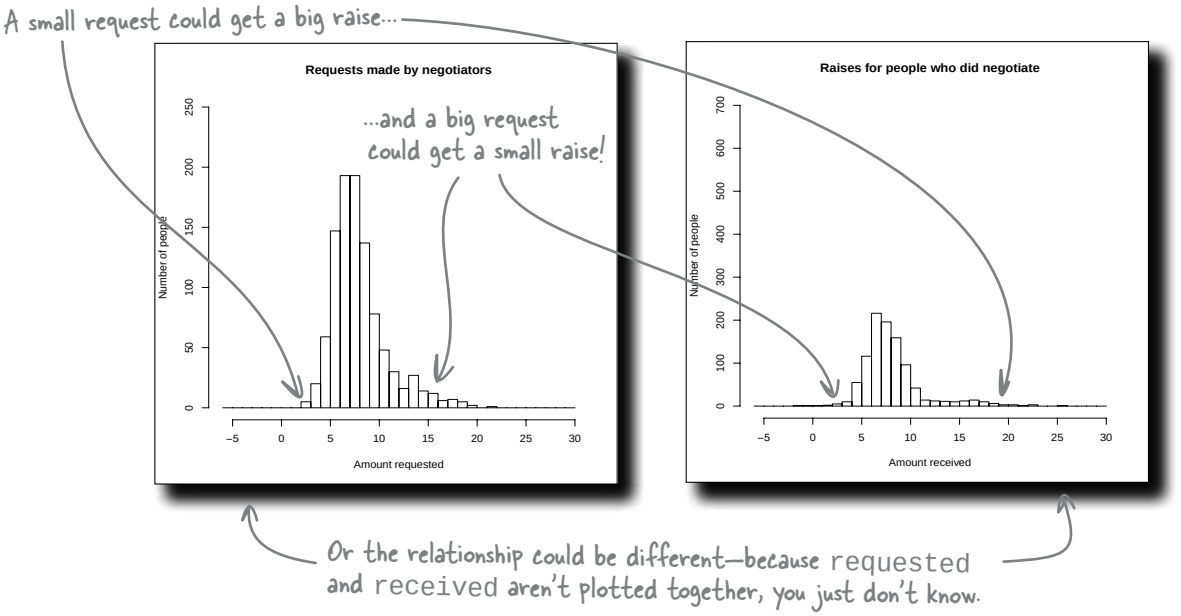




Sharpen your pencil Solution

Can you tell from looking at these two histograms how much someone should request in order to get the biggest raise?

No. The histograms show spreads of single variables, but they don't actually compare them. In order to know how these two variables relate to each other, we'd have to see where single individuals fall on both the `requested` and `received` distributions.



there are no Dumb Questions

Q: Can't I just overlay two histograms onto the same grid?

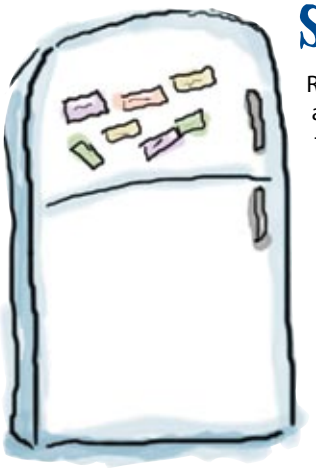
A: You totally can. But in order to make a good comparison, the two histograms need to describe *the same thing*. You made a bunch of histograms in the previous chapter using subsets of the same data, for example, and comparing those subsets to each other made sense.

Q: But Amount Received and Amount Requested are really similar, aren't they?

A: Sure, they're similar in the sense that they are measured using the same metric: percentage points of one's salary. But what you want to know is not so much the distribution of either variable but how, for a single person, one variable relates to the other.

Q: I get it. So once we have that information, how will we make use of it?

A: Good question. You should stay focused on the end result of your analysis, which is some sort of intellectual "product" that you can sell to your customers. What do you need? What will the product look like? But first, you need a visualization that compares these two variables.



Scatterplot Magnets

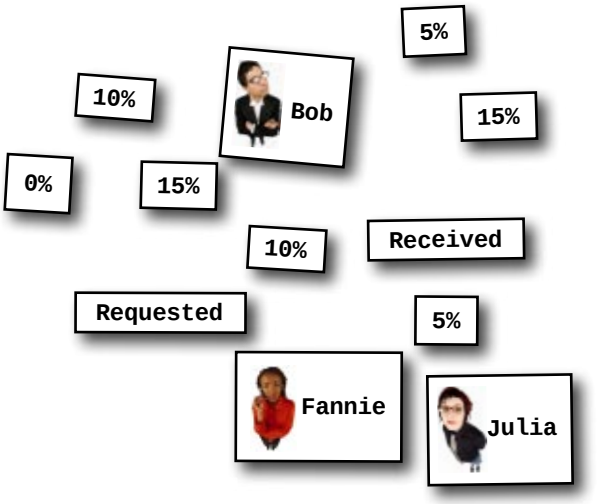
Remember scatterplots from chapter 4? They're a great visualization for looking at two variables together. In this exercise, take the data from these three people and use it to place them on the graph.

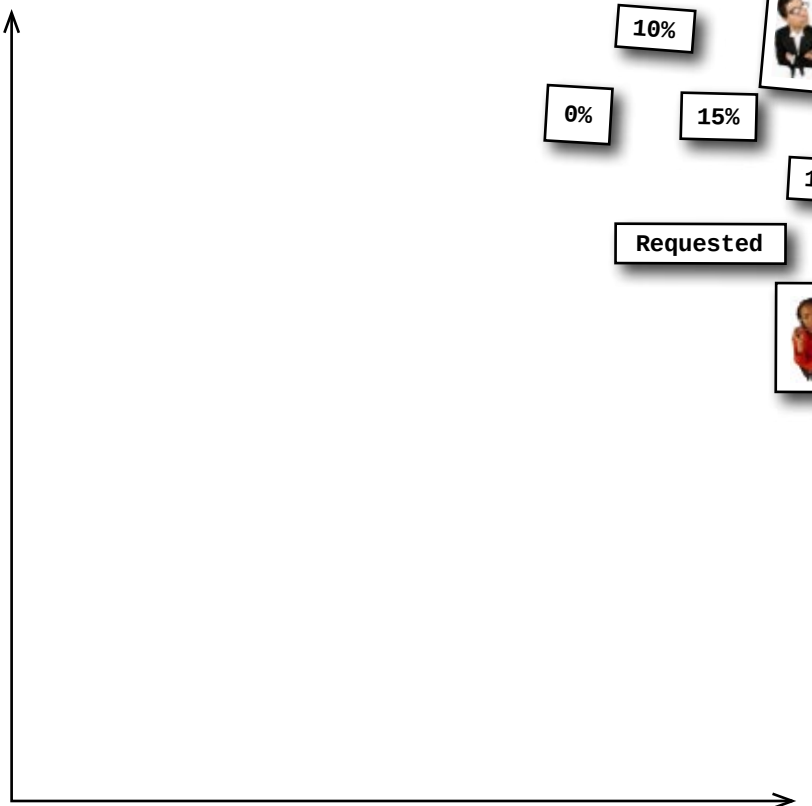
You'll need to use other magnets to draw your scale and your axis labels.

Bob requested 5% and received 5%.

Fannie requested 10% and received 8%.

Julia requested 2% and received 10%.





Use this x–y axis to plot Bob, Fannie, and Julia.

Scatterplot Magnets

You just plotted Bob, Fannie, and Julia on the axis to create a scatterplot. What did you find?

Bob requested 5% and received 5%.

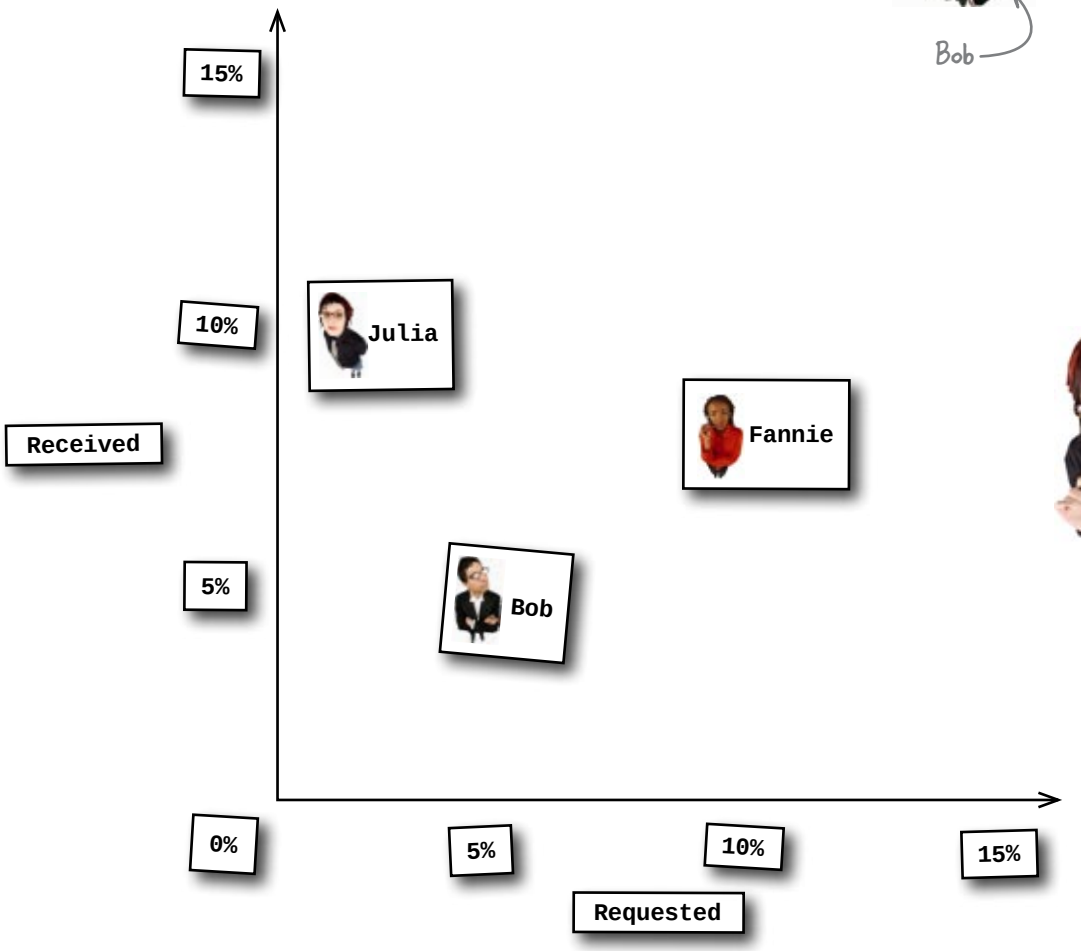
Fannie requested 10% and received 8%.

Julia requested 2% and received 10%.

Bob

Fannie

Julia



there are no Dumb Questions

Q: When can I use scatterplots?

A: Try to use them as frequently as you can. They're a quick way to show rich patterns in your data. Any time you have data with observations of two variables, you should think about using a scatterplot.

Q: So any two variables can be put together in a scatterplot?

A: As long as the two variables are in pairs that describe the same underlying thing or person. In this case, each line of our database represents an instance of an employee asking for a raise, and for each employee, we have a received and a requested value.

Q: What should I look for when I see them?

A: For an analyst, scatterplots are ultimately all about looking for causal relationships between variables. If high requests cause low raises, for example, you'll see an association between the two variables on the scatterplot. The scatterplot by itself only shows *association*, and to demonstrate causation you'll need more (for starters, you'd need an explanation of *why* one variable might follow from the other).

Q: What if I want to compare *three* pieces of data?

A: You can totally create visualizations in R that make a comparison among more than two variables. For this chapter, we're going to stick with two, but you can plot three variables using 3D scatterplots and multi-panel lattice visualizations. If you'd like a taste of multidimensional scatterplots, copy and run some of the examples of the `cloud` function that can be found in the help file at `help(cloud)`.

Q: So when do we get to look at the 2D scatterplot for the raise data?

A: Right now. Here's some ready bake code that will grab some new, more detailed data for you and give you a handy scatterplot. Go for it!

Ready Bake Code

Run these commands inside of R to generate a **scatterplot** that shows **what people requested** and **what they received**.

Make sure you're connected to the Internet when you run this command, because it pulls data off the Web.

```
employees <- read.csv("http://www.headfirstlabs.com/books/hfda/
       hfda_ch10_employees.csv", header=TRUE)

head(employees,n=30)

plot(employees$requested[employees$negotiated==TRUE],
       employees$received[employees$negotiated==TRUE])
```

This command loads some new data and doesn't display any results.

This command displays the scatterplot.

This command will show you what's in the data... always a good idea to take a look.

What happens when you run these commands?

Scatterplots compare two variables

Each one of the points on this **scatterplot** represents a single observation: a single person.

Like histograms, scatterplots are another quick and elegant way to show data, and they show the spread of data. But unlike histograms, scatterplots show ***two*** variables. Scatterplots show *how* the observations are paired to each other, and a good scatterplot can be part of how you demonstrate **causes**.

This dude asked for 7% but got 20%. He must be important.

This gentleman asked for 8% and received 8%.

This guy asked for 12% but had a 3% pay cut!

The plot command produced the scatterplot on the right.

Ready Bake Code

```
head(employees,n=30)

plot(employees$requested[employees$negotiated==TRUE],
  employees$received[employees$negotiated==TRUE])
```

The head command shows you the data below.

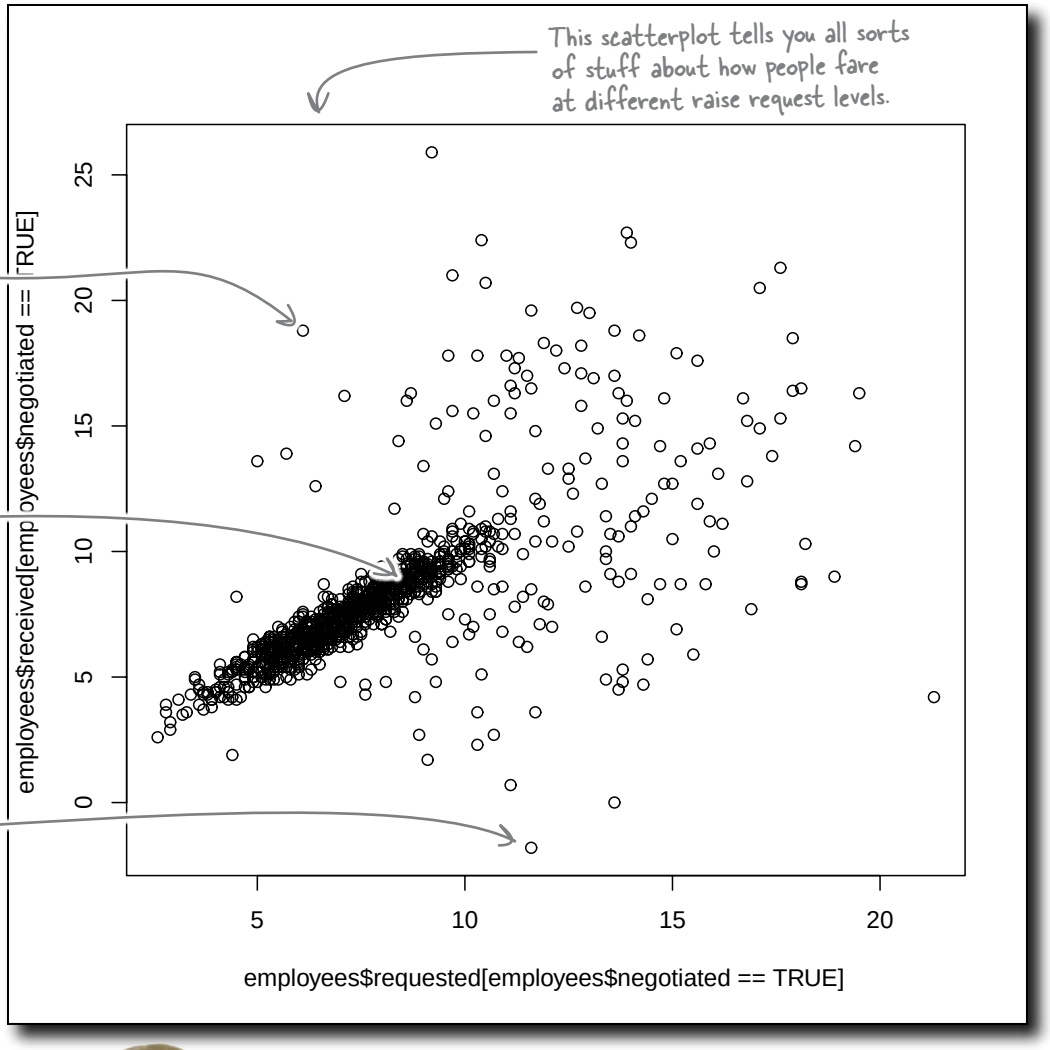
Here's the output of the head command.

R Console

```
> head(employees,n=30)
   received requested negotiated gender year
1       4.1      13.4       TRUE      M 2005
2       3.4      10.7       TRUE      F 2006
3       9.1       5.0       TRUE      M 2007
4      15.4      11.7       TRUE      F 2008
5      14.1      13.8       TRUE      M 2009
6      12.6      11.9       TRUE      F 2005
7       4.6       6.7       TRUE      M 2006
8      14.9      11.6       TRUE      F 2007
9      12.0      17.2       TRUE      M 2008
10      5.8      10.9       TRUE      F 2009
```

The head command is a quick way to peek inside any new data you load.

These three people and more are all inside this data set.



Of course you *can*, but why would you? Remember, you're trying to build an algorithm here.

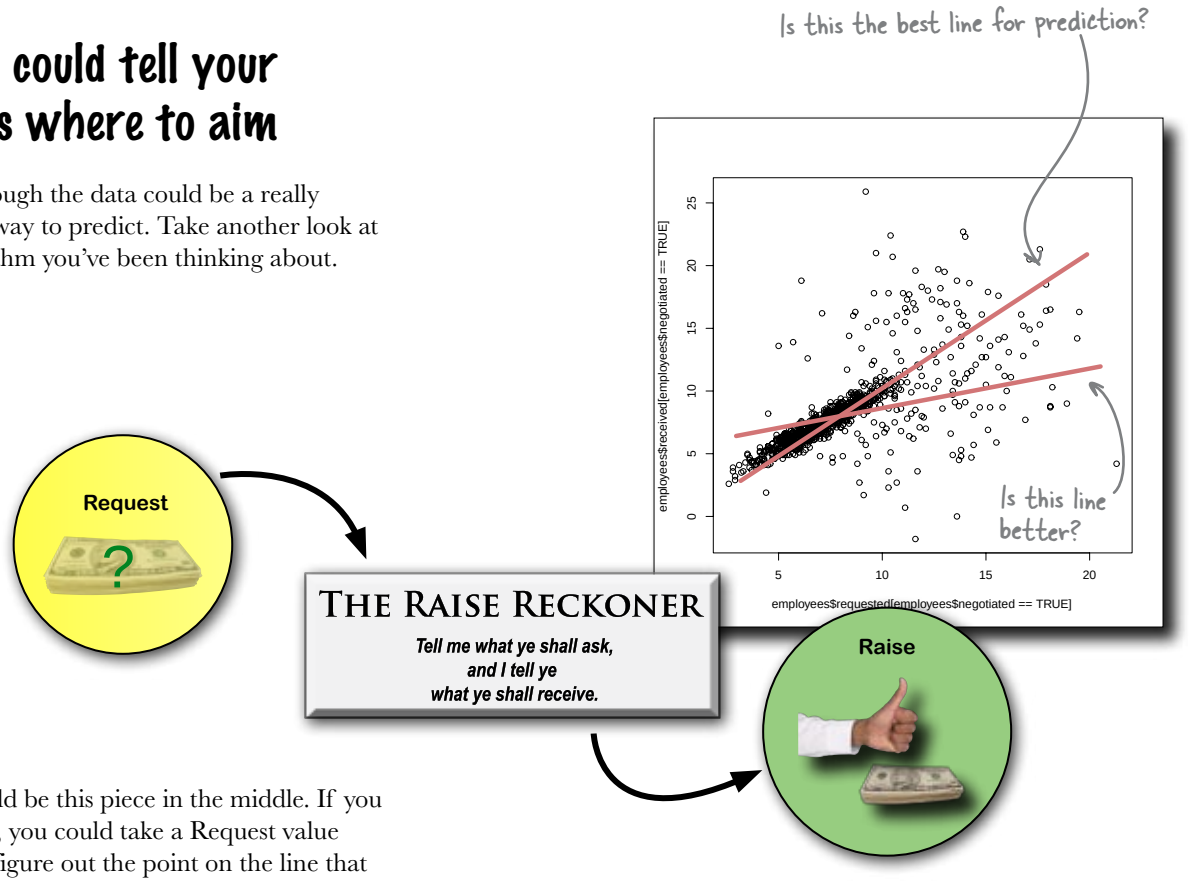
What would a line through the data do for you?

.....

.....

A line could tell your clients where to aim

A line through the data could be a really powerful way to predict. Take another look at the algorithm you've been thinking about.



A line could be this piece in the middle. If you had a line, you could take a Request value and then figure out the point on the line that matches a Received value.

If it was the ***right*** line, you might have your missing piece of the algorithm.

Sharpen your pencil

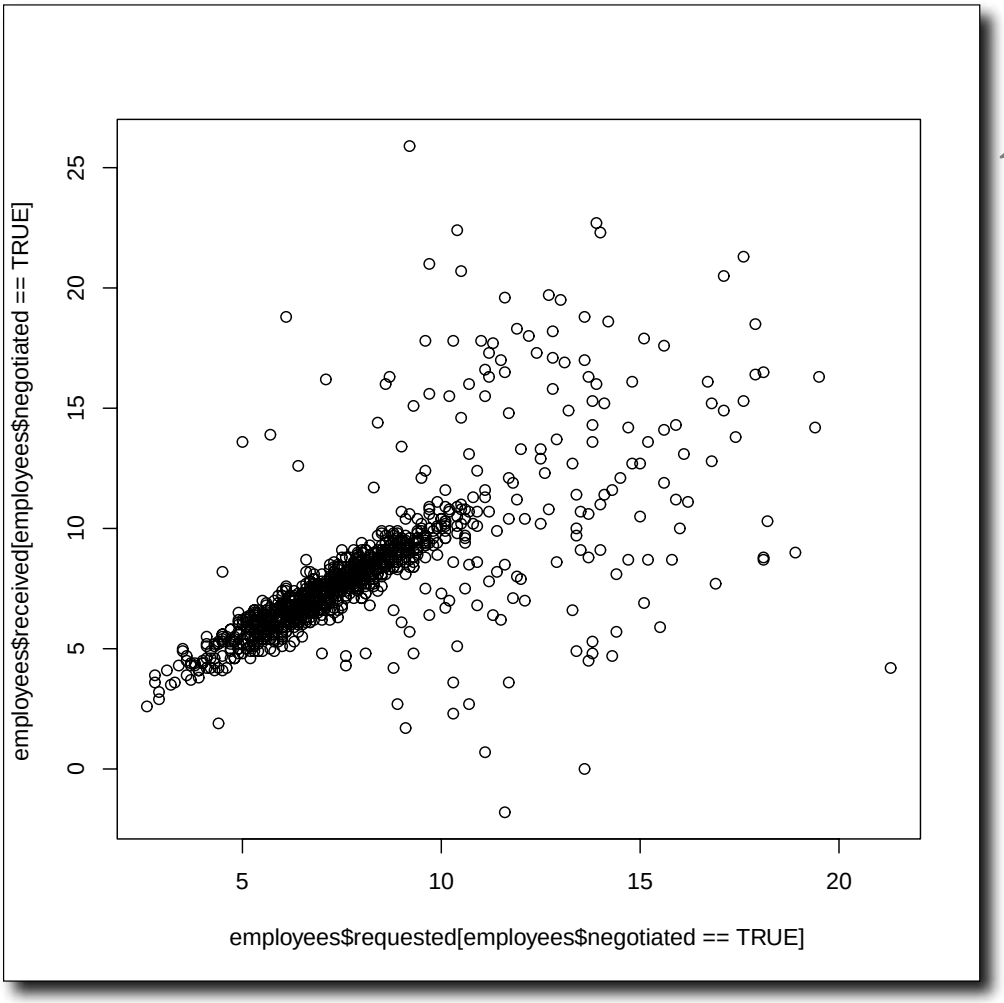
In order to figure out how to get the right line, why not just try answering a specific question about a single raise with your scatterplot? Here's an example:

If someone asks for a 8 percent raise, what is he likely to receive in return? See if looking at this scatterplot can tell you what sort of results people got from asking for 8 percent.

Take a good look at the scatterplot to answer this question.

.....

.....

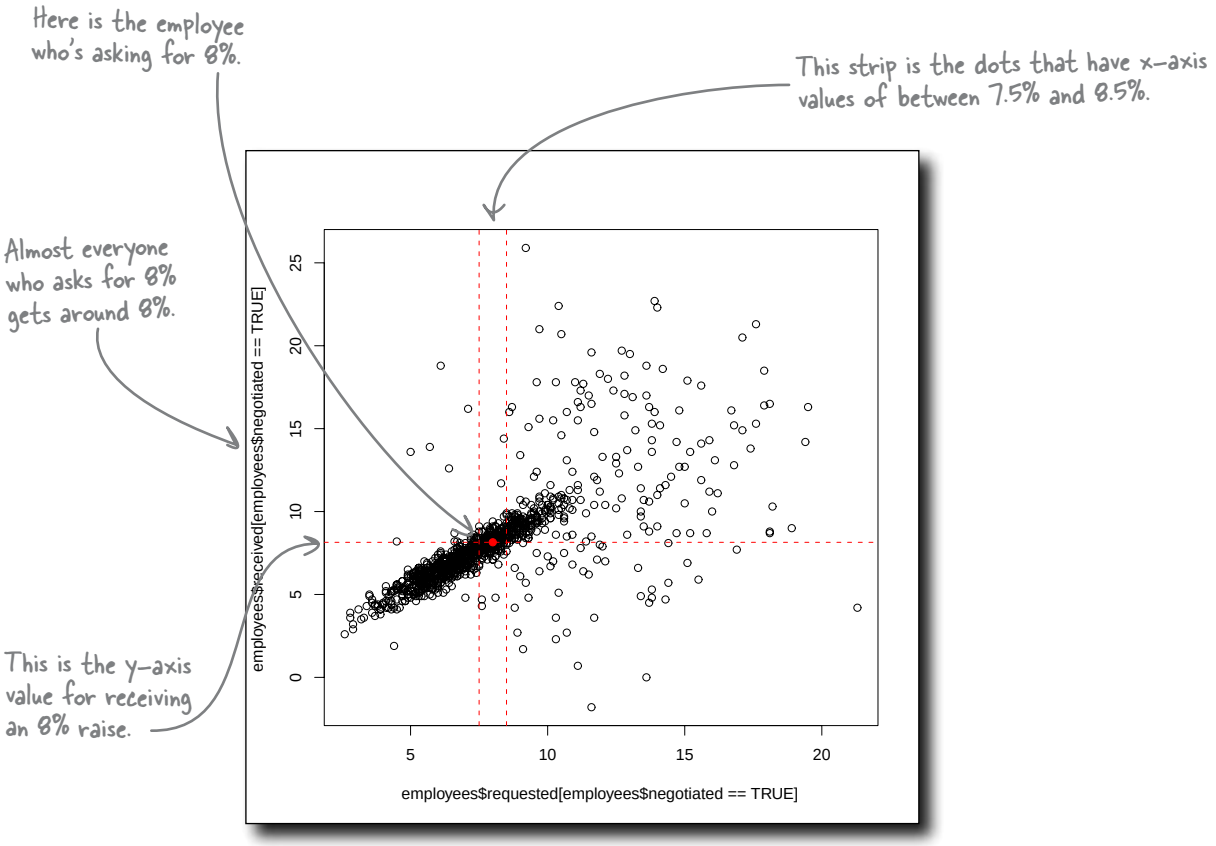


Hint: look at the dots in the 8% requested range!

Sharpen your pencil Solution

Using the scatterplot, how do you determine what an 8% raise request is likely to get you?

Just take the average amount received for dots around the range of amount requested you're looking at. If you look around 8% on the x-axis (the amount requested), it looks like the corresponding dots on the y-axis are about 8%, too. Take a look at the graph below.



If you take the **mean** of the Amount Received scores for dots in the 8 percent range (or **strip**), you get around 8 percent. On average, if you ask for 8 percent, you get 8 percent.

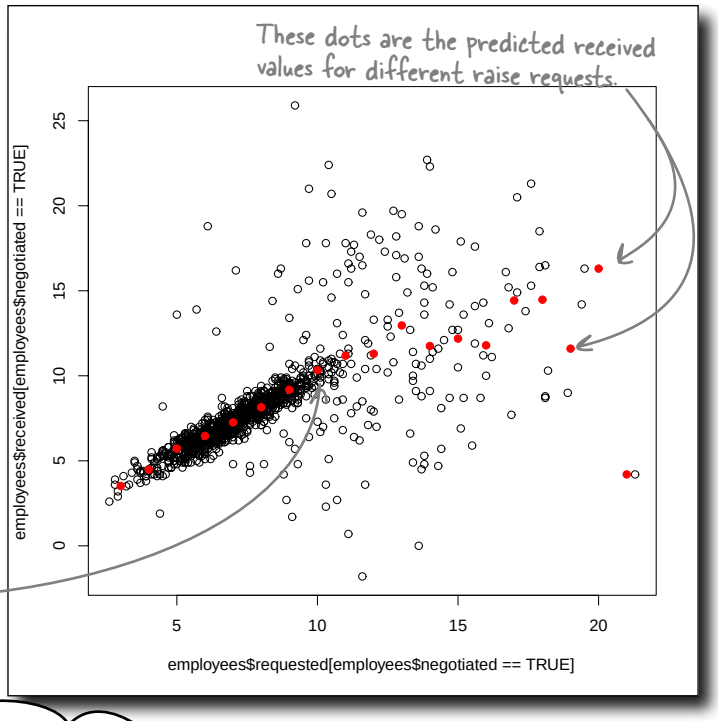
So you've solved the raise question for one group of people: those who ask for 8 percent. But other people will ask for different amounts.

What happens if you look at the average amount received for all the x-axis strips?

Predict values in each strip with the graph of averages

The **graph of averages** is a scatterplot that shows the predicted y-axis value for ***each strip on the x-axis***. This graph of averages shows us what people get, on average, when they request each different level of raise.

The graph of averages is a lot more powerful than just taking the overall average. The overall average raise amount, as you know, is 4 percent. But this graph shows you a much more subtle representation of how it all shakes out.



Man, I wanted to draw a line through the first scatterplot. I'm **dying** to draw a line through the graph of averages!

You've hit on the right line.

Seriously. Draw a line through the points on the graph of averages.

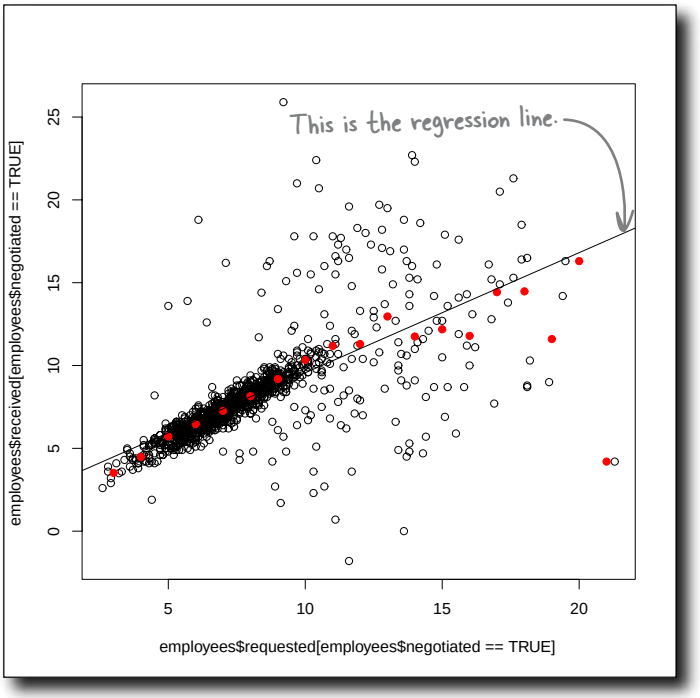
Because that line is the one you're looking for, the line that you can use to **predict raises for everybody**.

The regression line predicts what raises people will receive

Here you have it, the fascinating regression line.

The regression line is just **the line that best fits the points on the graph of averages**. As you're about to see, you don't just have to draw them on your graphs.

You can represent them with a simple equation that will allow you to predict the y variable for any x variable in your range.



there are no Dumb Questions

Q: Why is it called a regression?

A: The guy who discovered the method, Sir Francis Galton (1822-1911), was studying how the height of fathers predicted the height of their sons. His data showed that, on average, short fathers had taller sons, and tall fathers had shorter sons. He called this phenomenon "regression to mediocrity."

Q: Sounds kind of snooty and elitist. It seems that the word "regression" has more to do with how Galton felt about numbers on boys and their dads than anything statistical.

A: That's right. The word "regression" is more a historical artifact than something analytically illuminating.

Q: We've been predicting raise amount from raise request. Can I predict raise request from raise amount? Can I predict the x-axis from the y-axis?

A: Sure, but in that case, you'd be predicting the value of a past event. If someone came to you with a raise she received, you'd predict the raise she had requested. What's important is that you always do a reality check and make sure you keep track of the *meaning* of whatever it is that you're studying. Does the prediction make sense?

Q: Would I use the same line to predict the x-axis from the y-axis?

A: Nope. There are two regression lines, one for x given y and one for y given x. Think about it. There are two different graphs of averages: one for each of the two variables.

Q: Does the line have to be straight?

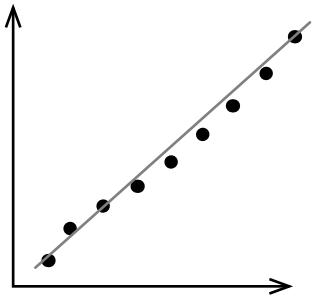
A: It doesn't have to be straight, as long as the regression makes sense. Nonlinear regression is a cool field that's a lot more complicated and is beyond the scope of this book.

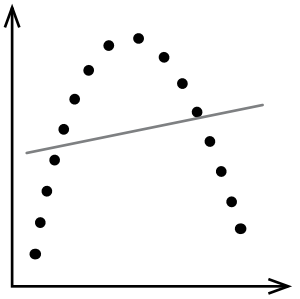
You're forgetting something. Are you sure the line is actually useful? I mean, what's it doing for ya?

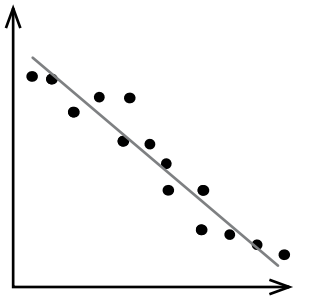
Make sure your line is actually useful.

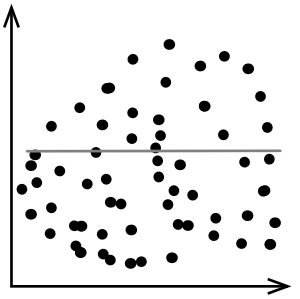
There are **a lot of different ways the scatterplot could look**, and a lot of different regression lines.

The question is how useful is the line in *your* scatterplot. Here are a few different scatterplots. Are the lines for each one going to be about as useful as the lines for any other? Or do some regression lines seem more powerful?



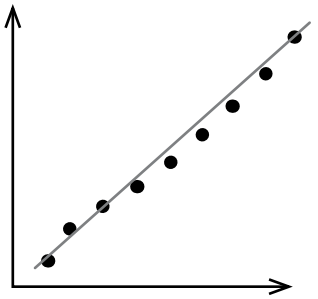




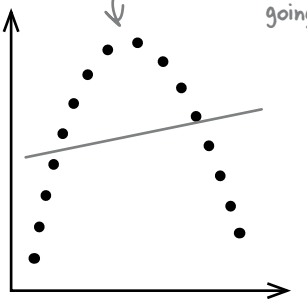


The line is useful if your data shows a linear correlation

A **correlation** is a linear association between two variables, and for an association to be linear, the scatterplot points need to roughly follow a line.



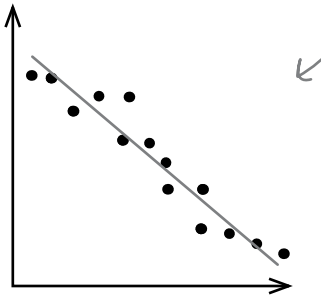
r = 0.9



Non-linear

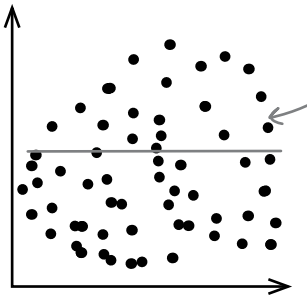
The dots on this scatterplot don't follow a straight line at all, so the regression line isn't going to predict accurately.

These two scatterplots show tight, strong correlations, and their regression lines will give you good predictions.



r = -0.7

These dots are all over the place, so the regression line might not be of much use here either.



r = 0

You can have strong or weak correlations, and they're measured by a **correlation coefficient**, which is also known as `r` (not to be confused with [big] R, the software program). In order for your regression line to be useful, data must show a strong linear correlation.

`r` ranges from -1 to 1, where 0 means *no association* and 1 or -1 means a *perfect* association between the two variables.

Does your raise data show a linear correlation?

Ready Bake R Code

Try using R (the program) to calculate **r** (the correlation coefficient) on your data raise. Type and execute this function:

```
cor(employees$requested[employees$negotiated==TRUE],
      employees$received[employees$negotiated==TRUE])
```

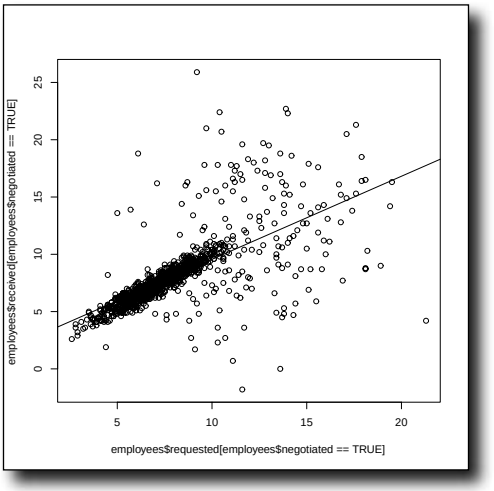
Annotate the elements of the function. What do you think they mean?

How does the output of the correlation function square with your scatterplot? Does the value match what you believe the association between these two variables to be?

.....

.....

.....



Ready Bake R Code

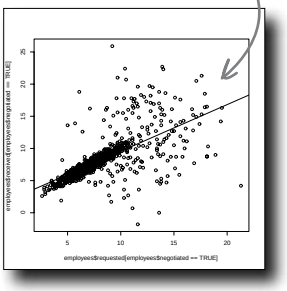
You just told R to give you the correlation coefficient of your two variables. What did you learn?

The cor function tells R to return the correlation of the two variables.

These are the two variables you want to test for correlation.

```
> cor(employees$requested[employees$negotiated==TRUE],employees
$received[employees$negotiated==TRUE])
[1] 0.6656481
>
```

You can see a linear association by looking at the chart.



How does the output of the correlation function square with your scatterplot?

Both the r-value and the scatterplot show a moderate correlation. It's not perfect, where all the points line up, but there's definitely a linear association.

Correlation Up Close

How do you got the correlation coefficient? The actual calculation to get the correlation coefficient is simple but tedious.

Here's one of the algorithms that can be used to calculate the correlation coefficient:

1 Convert each value to standard units.

Received		Requested	
9.6%	0.5 σ's	9.4%	0.5 σ's
8.6%	0.2 σ's	7.8%	-0.1 σ's
11.6%	1.1 σ's	10.1%	0.8 σ's
8.0%	0.0 σ's	7.9%	0.0 σ's
6.0%	-0.7 σ's	5.0%	-1.1 σ's
etc…	etc…	etc…	etc…

Standard units show how many standard deviations each value is from the mean.

2 Multiply each pair.

0.5 σ's	0.5 σ's	0.25
0.2 σ's	-0.1 σ's	0.02
1.1 σ's	0.8 σ's	0.88
0.0 σ's	0.0 σ's	0.00
-0.7 σ's	-1.1 σ's	0.77
etc…	etc…	etc…

3 Average the products.

0.25, 0.02, 0.88, 0.00, 0.77, etc… → 0.67

The correlation coefficient

there are no Dumb Questions

Q: I can see that a correlation of 1 or -1 is strong enough to enable me to use a regression line. But how low of a correlation is too low?

A: You just need to use your best judgment on the context. When you use the regression line, your judgments should always be qualified by the correlation coefficient.

Q: But how will I know how low of a correlation coefficient is too low?

A: As in all questions in statistics and data analysis, think about whether the regression **makes sense**. No statistical tool will get you the precisely correct answer all the time, but if you use those tools well, you will know how close they will get you on average. Use your best judgment to ask, "Is this correlation coefficient large enough to justify decisions I make from the regression line?"

Q: How can I tell for sure whether my distribution is linear?

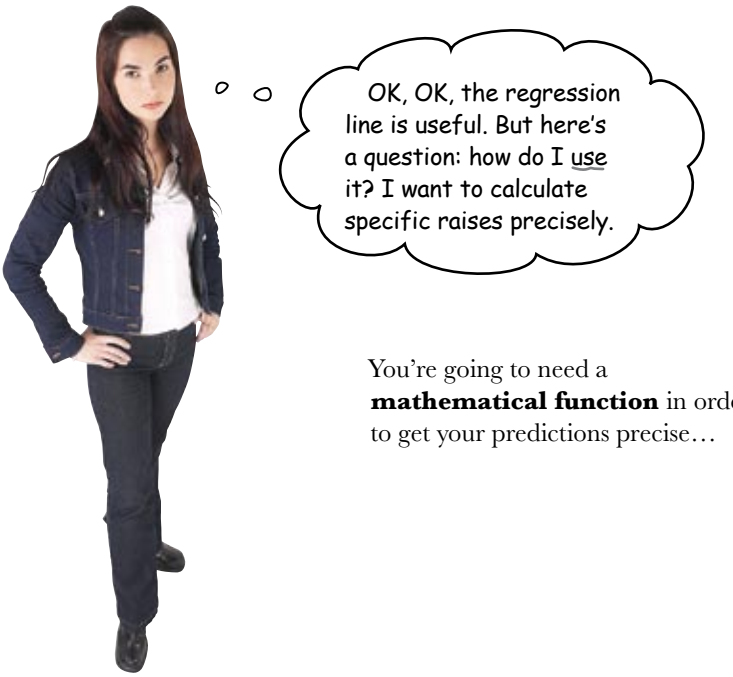
A: You should know that there are fancy statistical tools you can use to quantify the linearity of your scatterplot. But usually you're safe eyeballing it.

Q: If I show a linear relationship between two things, am I proving scientifically that relationship?

A: Probably not. You're specifying a relationship in a really useful mathematical sense, but whether that relationship *couldn't be otherwise* is a different matter. Is your data quality really high? Have other people replicated your results over and over again? Do you have a strong qualitative theory to explain what you're seeing? If these elements are all in place, you can say you've demonstrated something in a rigorous analytic way, but "proof" might be too strong a word.

Q: How many records will fit onto a scatterplot?

A: Like the histogram, a scatterplot is a really high-resolution display. With the right formatting, you can fit thousands and thousands of dots on it. The high-res nature of the scatterplot is one of its virtues.



You're going to need a **mathematical function** in order to get your predictions precise…

You need an equation to make your predictions precise

Straight lines can be described algebraically using the **linear equation**.

y is the y-axis value, which in this case in the thing we're trying to predict: raise received.

$$y = a + bx$$

x is the x-axis value, which in this case in the thing we know: the raise amount requested.

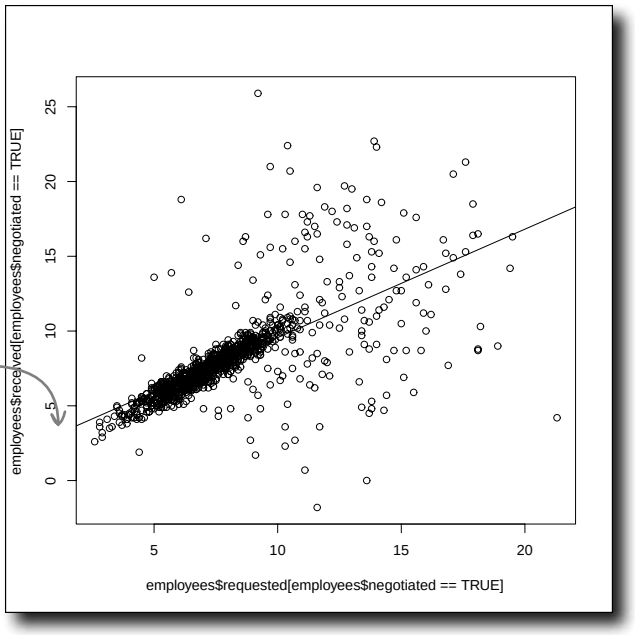
Your regression line can be represented by this linear equation. If you knew what yours was, you'd be able to plug any raise request you like into the `x` variable and get a prediction of what raise that request would elicit.

You just need to find the numerical values for `a` and `b`, which are values called the **coefficients**.

a represents the y-axis intercept

The first variable of the right side of the linear equation represents the y-axis **intercept**, where your line passes the y-axis.

Here's is the y-axis intercept.



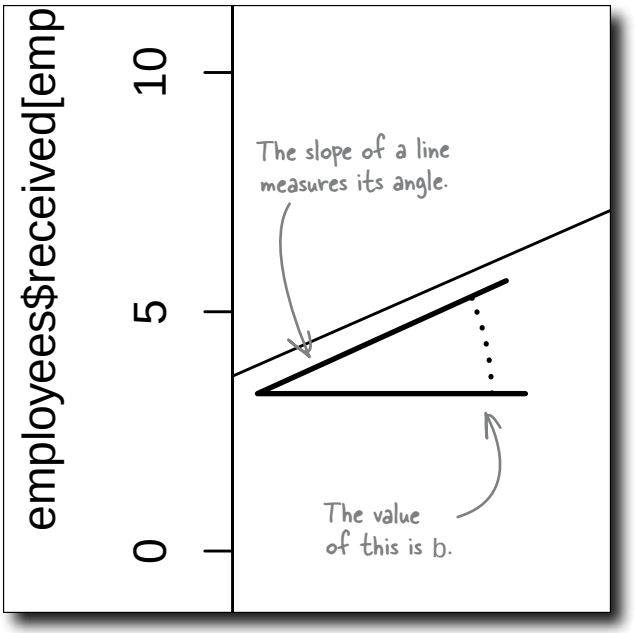
If you happen to have dots on your scatterplot that are around `x=0`, you can just find the point of averages for that strip. We're not so lucky, so finding the intercept might be a little trickier.

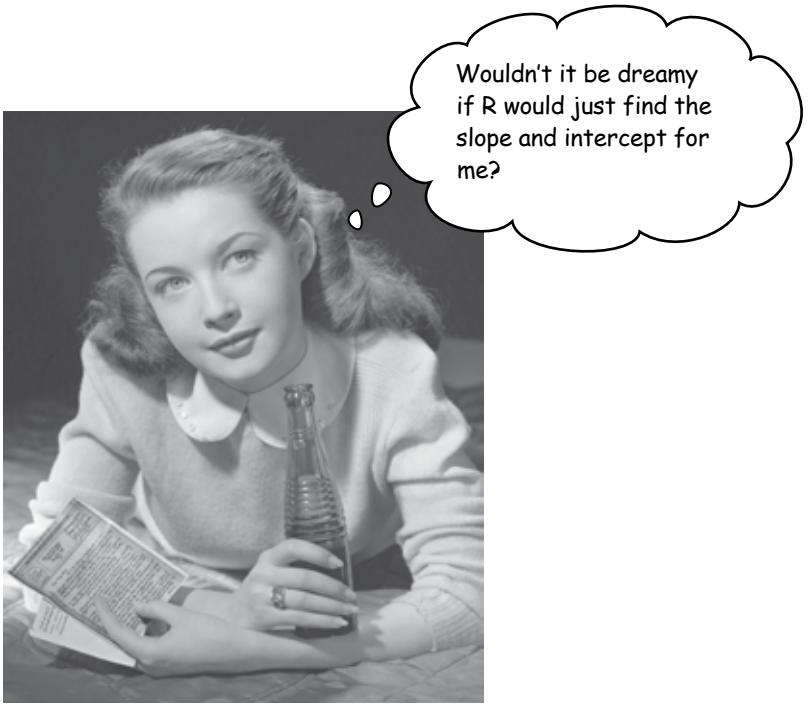
b represents the slope

The **slope** of a line is a measure of its angle. A line with a steep slope will have a large b value, and one with a relatively flat slope will have a b value close to zero. To calculate slope, measure how quickly a line rises (its "rise," or change in y-value) for every unit on the x-axis (its run).

$$\text{slope} = \frac{\text{rise}}{\text{run}} = b$$

Once you know the slope and y-axis intercept of a line, you can easily fill those values into your linear equation to get your line.



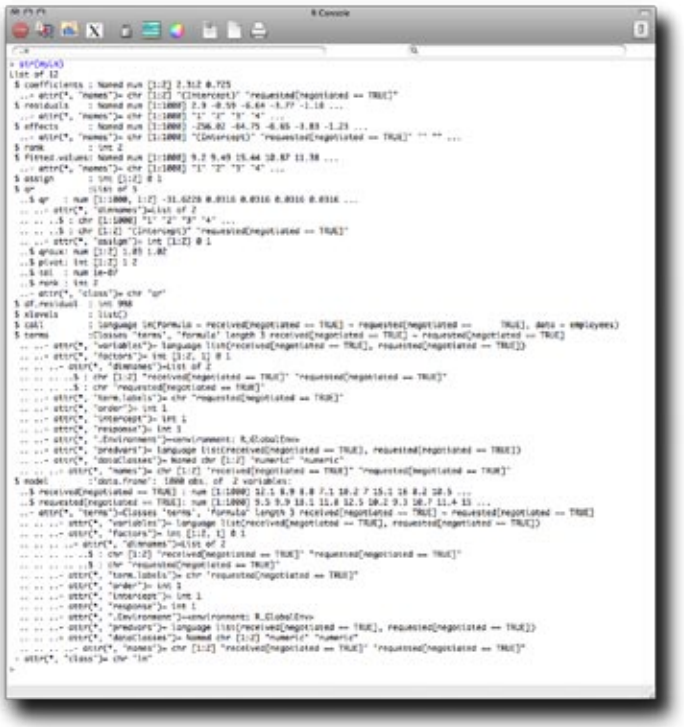


Tell R to create a regression object

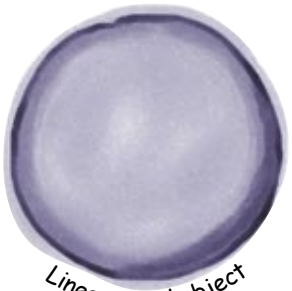
If you give R the variable you want to predict on the basis of another variable, R will generate a regression for you in a snap.

The basic function to use for this is `lm`, which stands for **linear model**. When you create a linear model, R creates an **object** in memory that has a long list of properties, and among those properties are your coefficients for the regression equation.

Behind the Scenes



Here's a list of all the properties R creates inside your linear model.



Linear model object

Watch it!

No software can tell you whether your regression makes sense.

R and your spreadsheet program can generate regressions like nobody's business, but it's up to you to make sure that it makes sense *to try to predict one variable from another. It's easy to create useless, meaningless regressions.*

Try creating your linear regression inside of R.

1. Run the formulas that create a linear model to describe your data and display the coefficients of the regression line.

```
myLm <- lm(received[negotiated==TRUE]~requested[negotiated==TRUE],
      data=employees)
myLm$coefficients
```

2. Using the numerical coefficients that R finds for you, write the regression equation for your data.

.....

$$y = a + bx$$

Here's the intercept.

Here's the slope.

Exercise Solution

What formula did you create using the coefficients that R calculated?

1. Run the formulas that create a linear model to describe your data and display the coefficients of the regression line.

```
R Console
> myLm <- lm(received[negotiated==TRUE]~requested[negotiated==TRUE],data=employees)
> myLm$coefficients
                  (Intercept) requested[negotiated == TRUE]
                    2.3121277                     0.7250664
>
```

2. Using the coefficients that R found for you, you can write your regression equation like this.

Raise received → $y = 2.3 + 0.7x$ ← This is your regression formula!

Raise requested

Here's the intercept.

Here's the slope.

Geek Bits

How did R calculate the slope? It turns out that the slope of the regression line is equal to the correlation coefficient multiplied by the standard deviation of y divided by the standard deviation of x.

$$b = r * \sigma_y / \sigma_x$$

This equation calculates the slope of the regression line.

$$b = .67 * 3.1 / 2.8 = 0.7$$

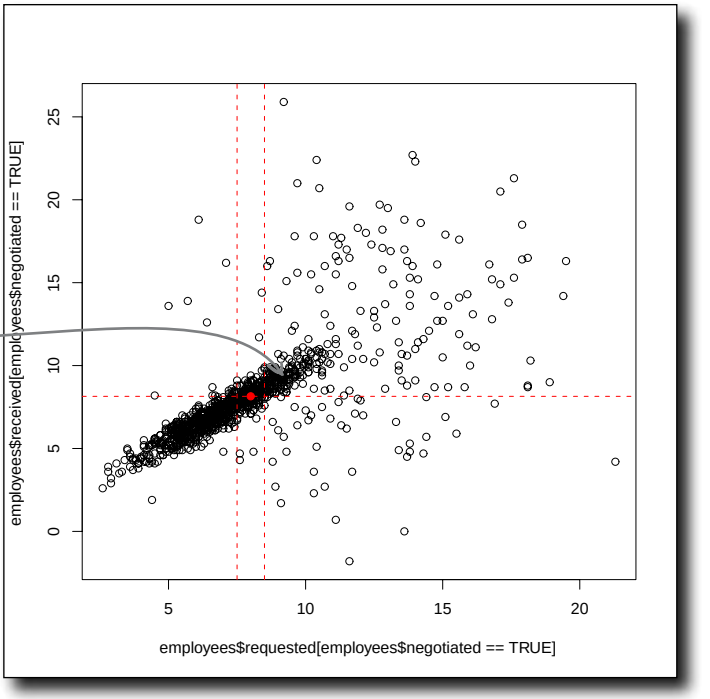
Here's your slope!

Ugh. Let's just say that calculating the slope of a regression line is one of those tasks that should make us all happy we have computers to do our dirty work. These are pretty elaborate calculations. But what's important to remember is this:

As long as you can see a solid association between your two variables, and as long as your regression *makes sense*, you can trust your software to deal with the coefficients.

The regression equation goes hand in hand with your scatterplot

Take the example of the person who wanted to know what sort of raise he'd receive if he asked for 8 percent. A few pages back, you made a prediction just by looking at the scatterplot and the vertical strip around 8 percent on the x-axis.



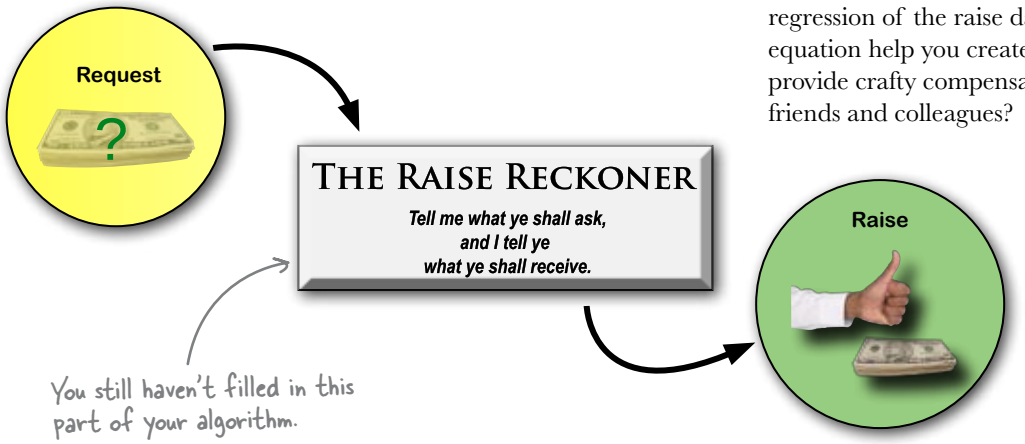
The regression equation your found with the help of the `lm` function gives you the same result.

```
y = 2.3 + 0.7x
  = 2.3 + 0.7 * 8
  = 7.9
```

Here's what the regression equation predicts he'll receive.

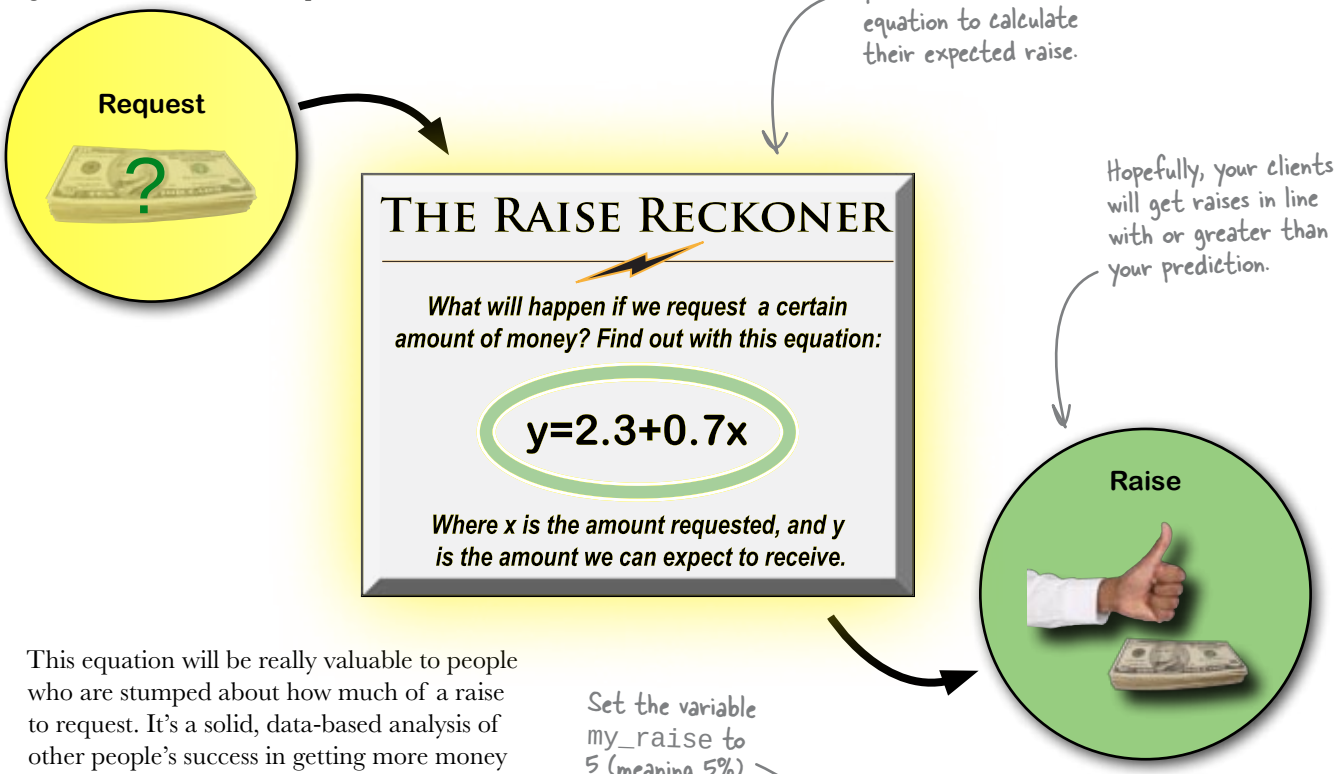
So what *is* the Raise Reckoner?

You've done a lot of neat work crafting a regression of the raise data. Does your regression equation help you create a product that will provide crafty compensating consulting for your friends and colleagues?



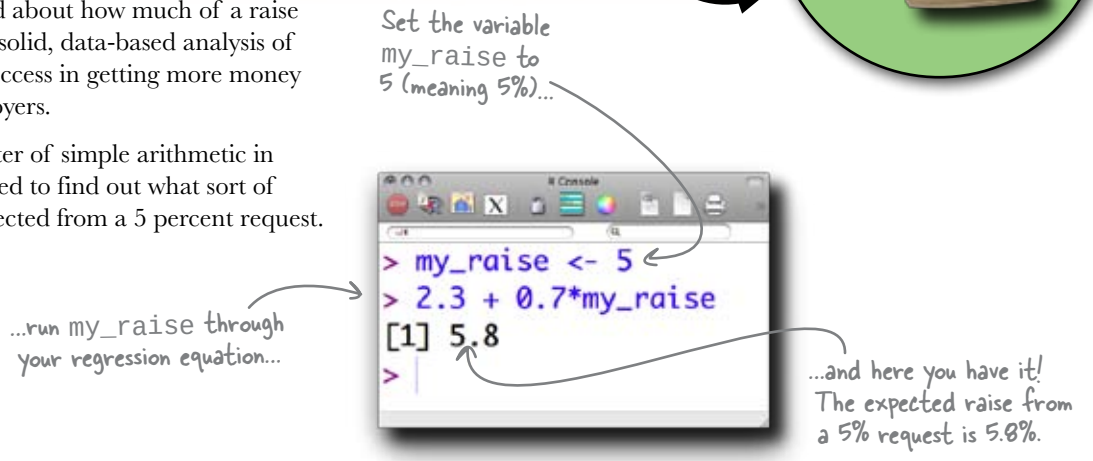
The regression equation is the Raise Reckoner algorithm

By taking a hard look at how people in the past have fared at different negotiation levels for their salaries, you identified a **regression equation** that can be used to predict raises given a certain level of request.



This equation will be really valuable to people who are stumped about how much of a raise to request. It's a solid, data-based analysis of other people's success in getting more money from their employers.

Using it is a matter of simple arithmetic in R. Say you wanted to find out what sort of raise can be expected from a 5 percent request. Here's the code.



there are no Dumb Questions

Q: How do I know that what people ask for tomorrow will be like they received today?

A: That's one of the big questions in regression analysis. Not only "Will tomorrow be like today?" but "What happens to my business if tomorrow is different?" The answer is that you don't know whether tomorrow will be like today. It **always** might be different, and sometimes completely different. The likelihood of change and its implications depend on your problem domain.

Q: How so?

A: Well, compare medical data versus consumer preferences. How likely is it that the human body, tomorrow, will suddenly change the way it works? It's possible, especially if the environment changes in a big way, but unlikely. How likely is it that consumer preferences will change tomorrow? You can bet that consumer preferences will change, in a big way.

Q: So why bother even trying to predict behavior?

A: In the online world, for example, a good regression analysis can be very profitable for a period of time, even it stops producing good predictions tomorrow. Think about your own behaviors. To an online bookseller, you're just a set of data points.

Q: That's kind of depressing.

A: Not really—it means the bookseller knows how to get you what you want. You're a set of data points that the bookseller runs a regression on to predict which books you'll want to buy. And that prediction will work until your tastes change. When they do, and you start buying different books, the bookseller will run the regression again to accommodate the new information.

Q: So when the world changes and the regression doesn't work any more, I should update the it?

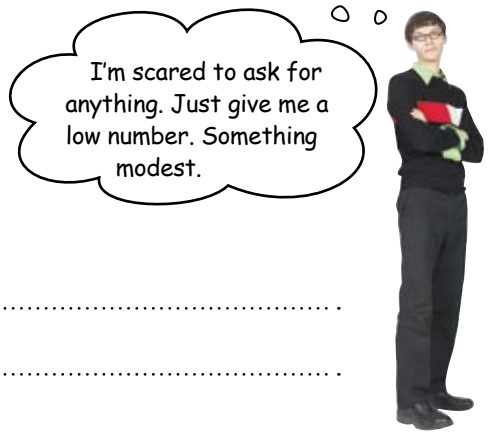
A: Again, it depends on your problem domain. If you have good qualitative reasons to believe that your regression is accurate, you might never have to change it. But if your data is constantly changing, you should be running regressions constantly and using them in a way that enables you to benefit if the regressions are correct but that doesn't destroy your business if reality changes and the regressions fail.

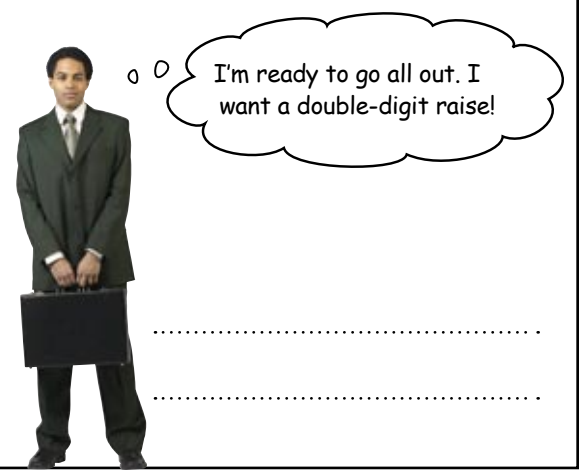
Q: Shouldn't people ask for the raise they think they *deserve* rather than the raise they see other people getting?

A: That's an excellent question. The question is really part of your mental model, and statistics won't tell you whether what you're doing is the right or fair approach. That's a qualitative question that you, the analyst, need to use your best judgment in evaluating. (But the short answer is: you deserve a *big raise*!)

Exercise

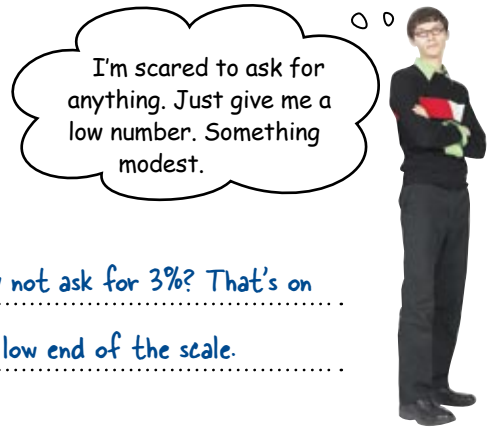
Meet your first clients! Write down what sort of raise you think is appropriate for them to request, given their feelings about asking, and use R to calculate what they can expect.



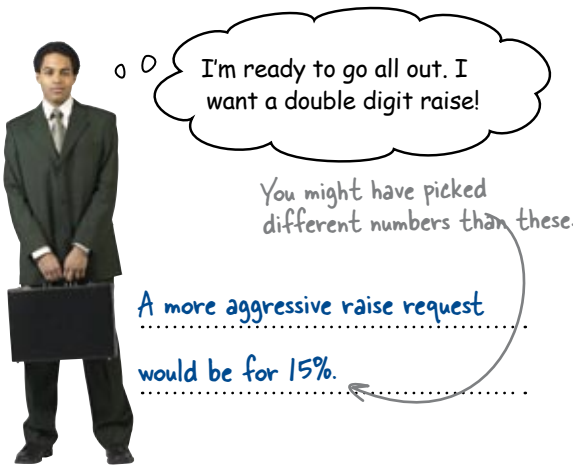


Exercise Solution

What did you recommend to your first two clients, and what did R calculate their expected raises to be?



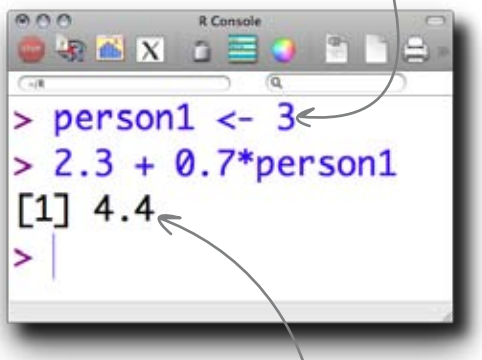
Why not ask for 3%? That's on the low end of the scale.



You might have picked different numbers than these.

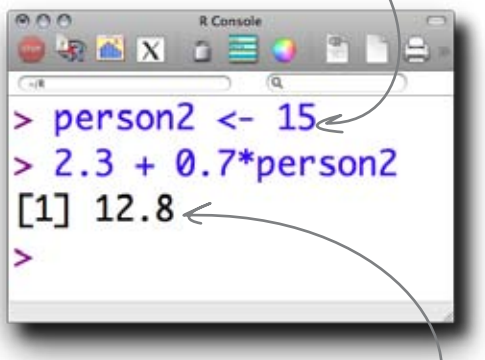
A more aggressive raise request would be for 15%.

A low raise request might be 3%.



Someone who asked for 3% would expect to get around 4.4% in return.

A higher raise request might be 15%.



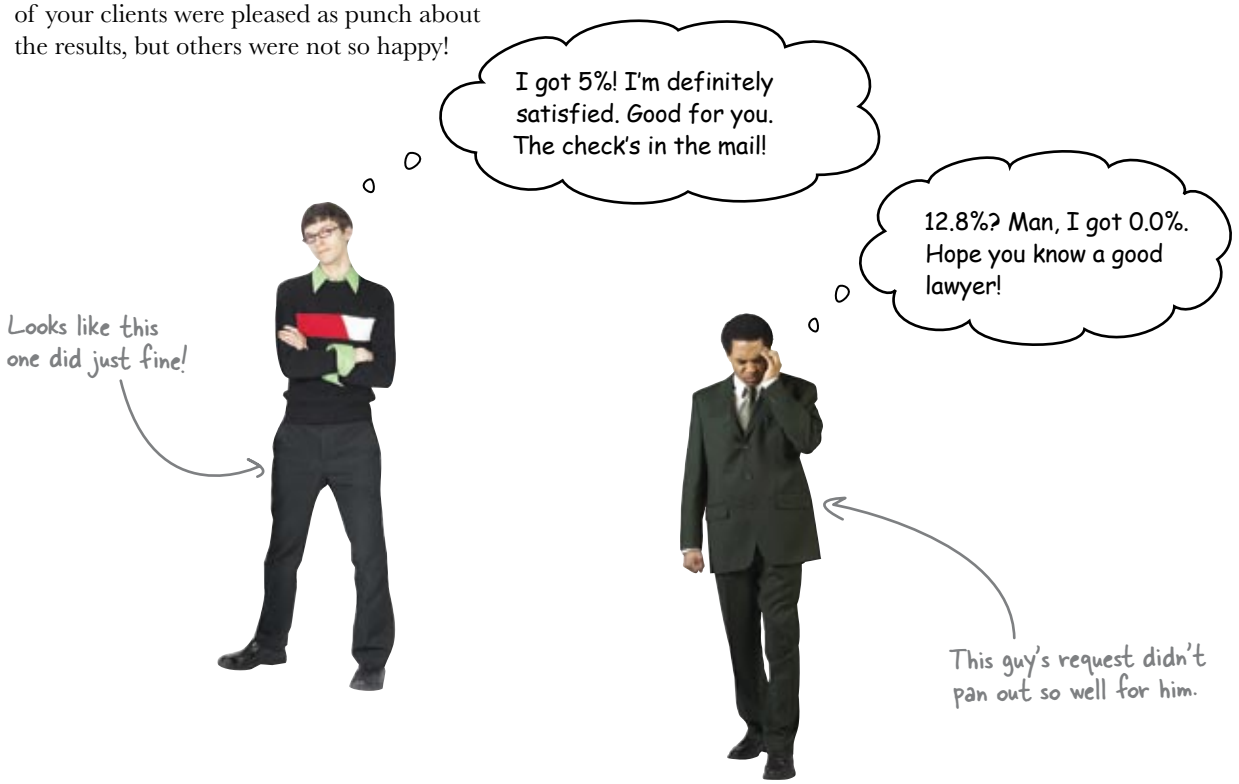
Someone who asked for 15% would expect to get around 12.8% in return.

Let's see what happened...

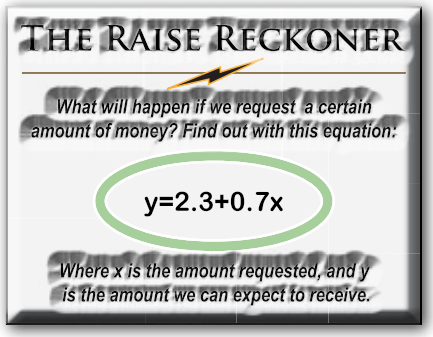
Your raise predictor didn't work out as planned...

People were falling all over themselves to get your advice, and you got off your first round of recommendations smoothly.

But then the **phone started ringing**. Some of your clients were pleased as punch about the results, but others were not so happy!



Did this thing misfire?



What did your clients **do** with your advice? What went wrong for those who came back unhappy?

You'll have to get to the bottom of this situation in the **next chapter**…

11 error

Err Well

The world is messy.

So it should be no surprise that your predictions rarely hit the target squarely. But if you offer a prediction with an **error range**, you and your clients will know not only the average predicted value, but also how far you expect typical deviations from that error to be. Every time you express error, you offer a much richer perspective on your predictions and beliefs. And with the tools in this chapter, you'll also learn about how to get error under control, getting it as low as possible to increase confidence.

Your clients are pretty ticked off

In the previous chapter, you created a linear regression to predict what sort of raises people could expect depending on what they requested.

Lots of customers are using the raise algorithm. Let's see what they have to say.

I got a 4.5% raise. It was a good raise. I think that's the sort of raise I deserved. I was so nervous in the meeting that I can't even remember what I asked for.

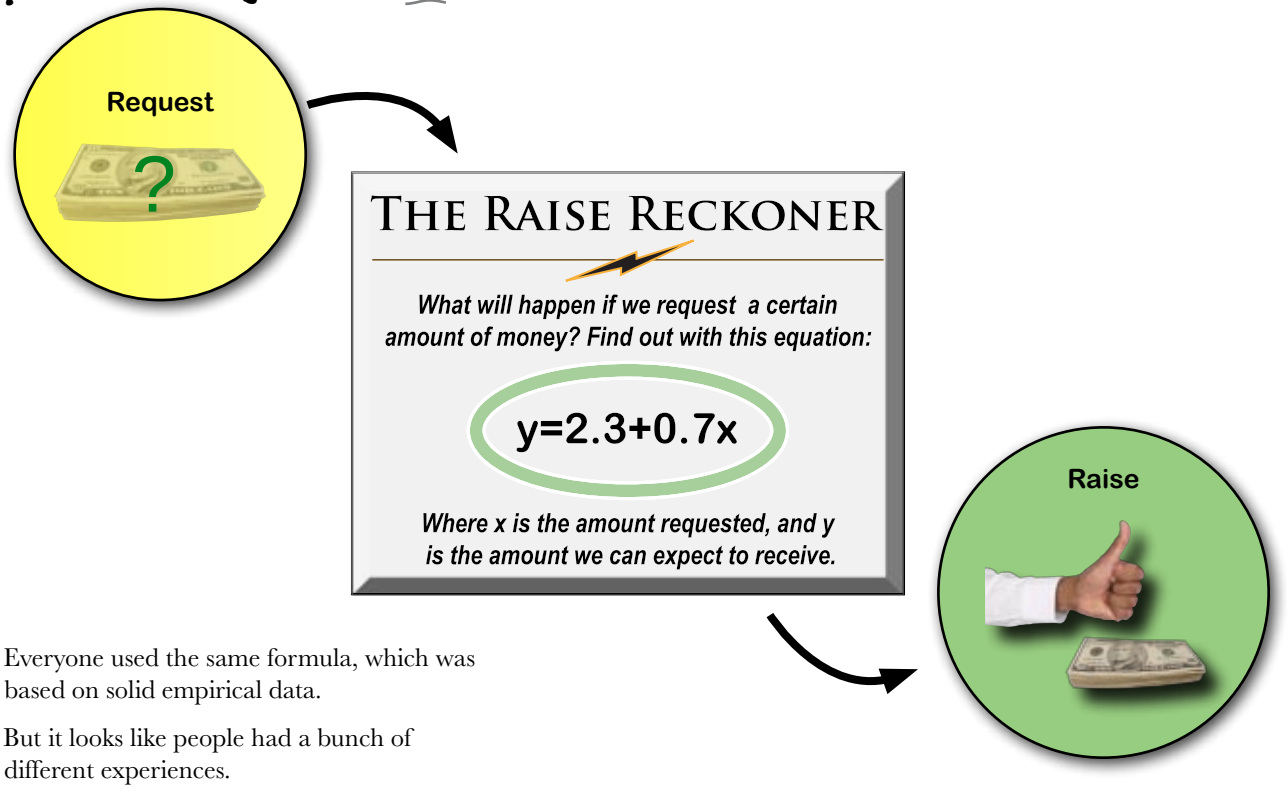
I can't believe it! I got a 5.0% bigger raise than the algorithm predicted! My negotiation must have scared my boss, and he just started throwing money at me!

Yeah, I got **no raise**. Did you hear that? **0.0%** I have some ideas for you about what you can do with your algorithm.

I'm pretty pleased. My raise was 0.5% lower than expected, but it's still a solid raise. I'm pretty sure I wouldn't have gotten it if I hadn't negotiated.

Bull's-eye! I got the exact raise the algorithm predicted. I'm telling you, it's incredible. You must be some sort of genius. You rock my world.

What did your raise prediction algorithm do?



Everyone used the same formula, which was based on solid empirical data.

But it looks like people had a bunch of different experiences.

What happened?

Sharpen your pencil

The statements on the facing page are qualitative data about the effectiveness of your regression.

How would you **categorize** the statements?

.....

.....

.....

.....

Sharpen your pencil Solution

You looked closely at your customers' qualitative responses to your raise prediction algorithm. What did you find?

The statements.

Bull's-eye! I got the exact raise the algorithm predicted. I'm telling you, it's incredible. You must be some sort of genius. You rock my world.

This one's spot on!

I'm pretty pleased. My raise was 0.3% lower than expected, but it's still a solid raise. I'm pretty sure I wouldn't have gotten it if I hadn't negotiated.

This one got a raise that was close but not exactly what you predicted.

Yeah, I got no raise. Did you hear that? **0.0%** I have some ideas for you about what you can do with your algorithm.

I can't believe it! I got a 5.0% bigger raise than the algorithm predicted! My negotiation must have scared my boss, and he just started throwing money at me!

These two appear to be way off.

It looks like there are basically three types of response, qualitatively speaking. One of them got exactly what the algorithm predicted. Another received a raise that was a little off, but still close to the prediction. Two of them got raises that were way off. And the last one, well, unless there's a trend of people who can't remember what they requested there's probably not much you can make of it.

This one's just weird. It's kind of hard to draw any conclusion off a statement like this.

I got a 4.5% raise. It was a good raise. I think that's the sort of raise I deserved. I was so nervous in the meeting that I can't even remembered what I asked for.

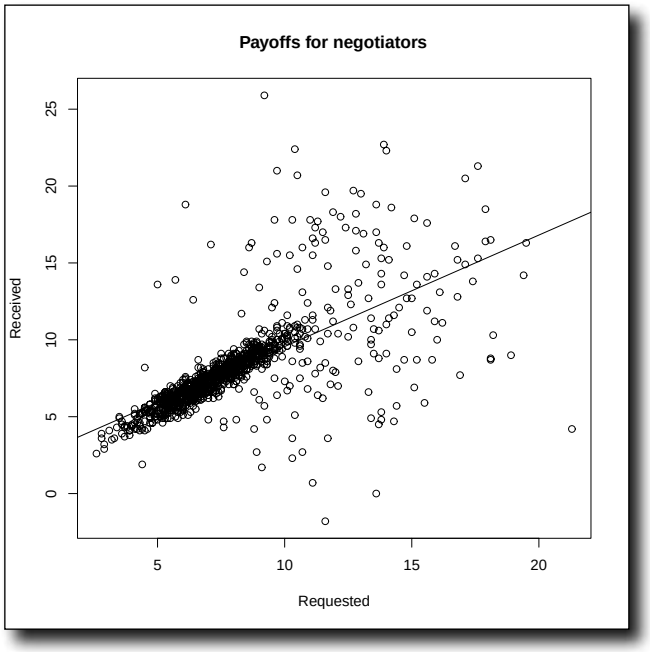
The segments of customers

Remember, the regression equation predicts what people will hit **on average**. Obviously, not everyone is going to be exactly at the average.

Let's take a few more responses from your clients. The ones below are a little more specific than the previous ones.

Draw arrows to point where each of these people would end up if you plotted their request/raise experiences on your scatterplot.

Draw arrows to show where each of these people would show up on the scatterplot.



Do you notice anything *weird*?

.....

.....

.....

Exercise Solution

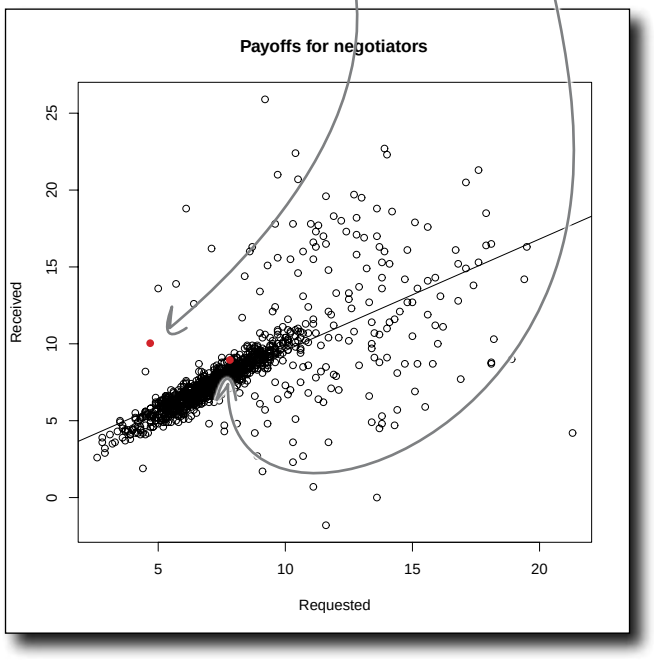
You just added new dots to your scatterplot to describe where three of your customers would be shown. What did you learn?

I demanded 5% and got 10%.

This person shows up above the regression line on the far left of the chart.

I requested 8%, and I got 7%.

This person would show up right in the middle of the biggest clump of observations.



This person doesn't show up on the scatterplot at all.

I asked for 25% and got 0.0%... because I was fired!

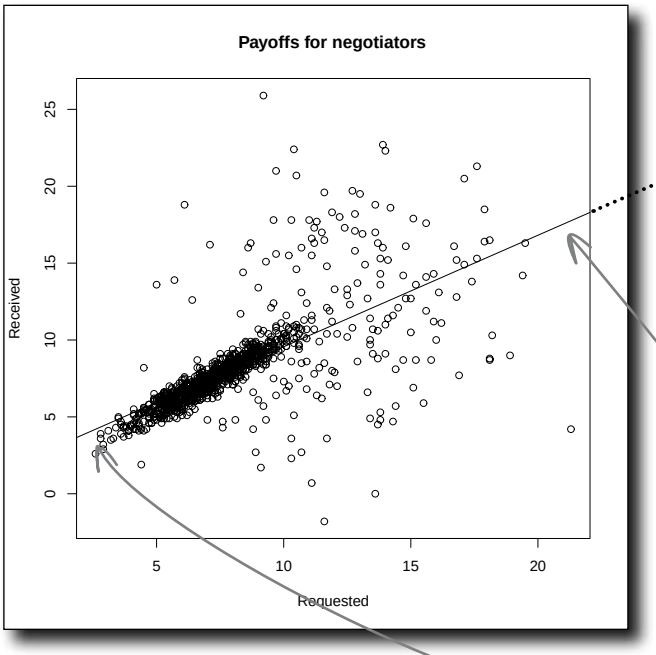
?

He's off the chart!

The guy who asked for 25% went outside the model

Using a regression equation to predict a value outside your range of data is called **extrapolation**. Beware extrapolation!

The regression line points to oblivion.



You don't know what's going on out here. Maybe if you had more data, you could use your equation to predict what a bigger request would get.

But you'd definitely have to run your regression again on the new data to make sure you're using the right line.

Extrapolation is different from **interpolation**, where you are predicting points within your range of data, which is what regression is designed to do. Interpolation is fine, but you should be leery of extrapolation.

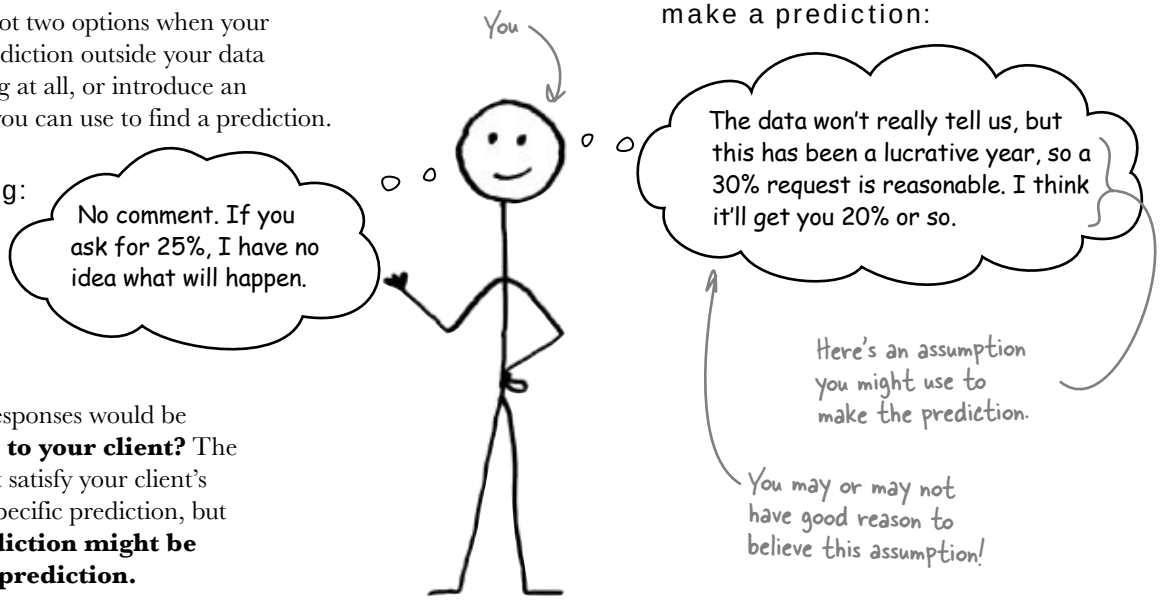
People extrapolate all the time. But if you're going to do it, you need to **specify additional assumptions** that make explicit your ignorance about what happens outside the data range.

What would you say to a client who is wondering what he should expect if he requested a 30% raise?

How to handle the client who wants a prediction outside the data range

You've basically got two options when your clients want a prediction outside your data range: say nothing at all, or introduce an assumption that you can use to find a prediction.

Say nothing:



Which of these responses would be more **beneficial to your client?** The second one might satisfy your client's desire to have a specific prediction, but a **crummy prediction might be worse than no prediction.**

there are no Dumb Questions

Q: So what exactly might happen outside the data range that's such a problem?

A: There might not even be data outside the range you're using. And if there is, the data could look totally different. It might even be nonlinear.

Q: I won't necessarily have all the points *within* my data range, though.

A: You're right, and that's a data quality and sampling issue. If you don't have all the data points—if you're using a sample—you want to make sure that the sample is representative of the overall data set and is therefore something you can build a model around.

Q: Isn't there something to be said for thinking about what would happen under different hypothetical, purely speculative situations?

A: Yes, and you should definitely do it. But it takes discipline to make sure your ideas about hypothetical worlds don't spill over into your ideas (and actions) regarding the real world. People abuse extrapolation.

Q: Isn't any sort of prediction about the *future* a type of extrapolation?

A: Yes, but whether that's a problem depends on what you're studying. Is what you're looking at the sort of thing that could totally change its behavior in the future, or is it something that is pretty stable? The physical laws of the universe probably aren't going to change much next week, but the associations that apparently explain the stock market might. These considerations should help you know how to use your model.

Watch it!

Always keep an eye on your model assumptions.

And when you're looking at anyone else's models, always think about how reasonable their assumptions are and whether they might have forgotten to mention any. Bad assumptions can make your model completely useless at best and dangerously deceptive at worst.

BE the model

Look at this list of possible assumptions for the Raise Reckoner. How might each of these change your model, if it were true?

Economic performance has been about the same for all years in the data range, but this year we made a lot less money.

One boss administered all the raises in the company for the data we have, but he's left the company and been replaced by another boss.

How you ask makes a big difference in what kind of raise you get.

The spread of dots in the 20-50 percent range looks just like the spread of dots in the 10-20 percent range.

Only tall people ask for raises.

BE the model

Look at this list of possible assumptions for the Raise Reckoner. How might each of these change your model, if it were true?

Economic performance has been about the same for all years in the data range, but this year we made a lot less money.

This year's raises could be down, on average. The model might not work.

One boss administered all the raises in the company for the data we have, but he's left the company and been replaced by another boss.

Yikes! That'd be the end of your business, at least until you have data on the new guy!

The new guy might think differently and break the model.

How you ask makes a big difference in what kind of raise you get.

This is surely true, and the data reflects the variation, so the model's OK.

You don't have data on how to ask for money... the model just says what you'll get on average at different requests.

The spread of dots in the 20-50 percent range looks just like the spread of dots in the 10-20 percent range.

If this were true, you'd be able to extrapolate the regression equation.

Only tall people have asked for raises in the past.

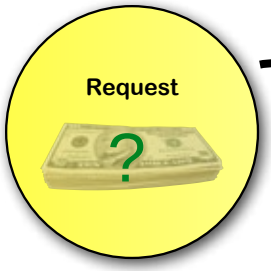
If this were true, the model wouldn't apply to shorter people.

Shorter people might do better or worse than taller people.

Now that you've thought through how your assumptions affect your model, you need to change your algorithm **so that people know how to deal with extrapolation**.

Sharpen your pencil

You need to tweak your algorithm to instruct your clients to avoid the trap of extrapolation. What would you add?



THE RAISE RECKONER

What will happen if we request a certain amount of money? Find out with this equation:

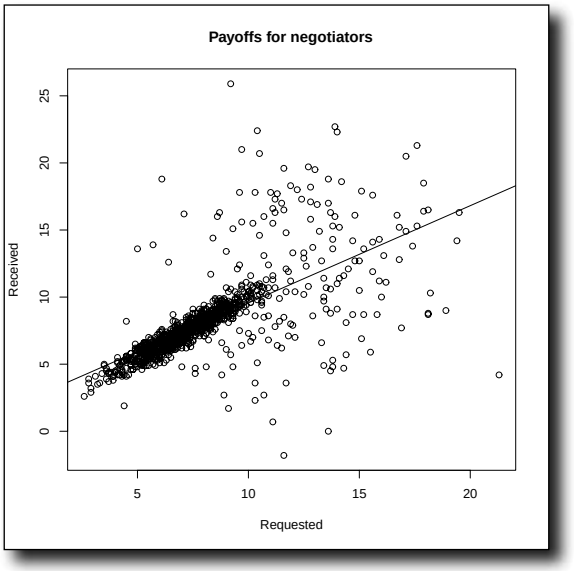
$$y=2.3+0.7x$$

Where x is the amount requested, and y is the amount we can expect to receive.

.....

.....

Write your caveat about extrapolation here.



How would you **explain to your clients** that they need to avoid extrapolation?

.....

.....

.....

.....

Sharpen your pencil Solution

How did you modify your compensation algorithm to ensure that your clients don't extrapolate beyond the data range?

THE RAISE RECKONER

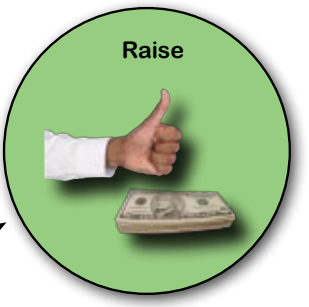
What will happen if we request a certain amount of money? Find out with this equation:

$$y=2.3+0.7x$$

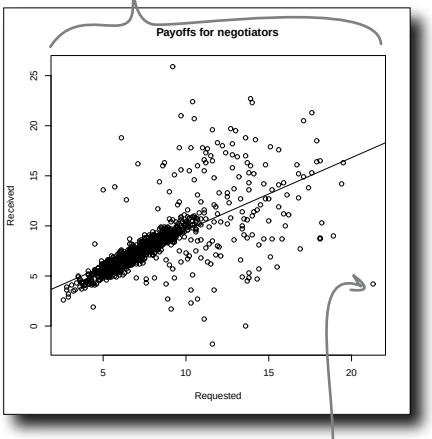
Where x is the amount requested, and y is the amount we can expect to receive.

But the formula only works if your requested amount (x) is between 0% and 22%.

Here's the language you need to add.



Your regression equation will work within this range.



Your data range for the amount requested only extends to here.

?

Beyond a 22% request, you can't say what will happen.

How would you change the algorithm to tell your clients to avoid extrapolation?

Because you only have data for people who request 22% or less in raises, your regression only applies to requests between 0% and 22%. Your clients can ask for more—and maybe they make a lot of money if they do—but as far as you're concerned, they'd be on their own.

The guy who got fired because of extrapolation has cooled off

Well, at least you're fixing your analysis as you go along. That's integrity. I'll still hit you up for advice next time I'm up for a raise.

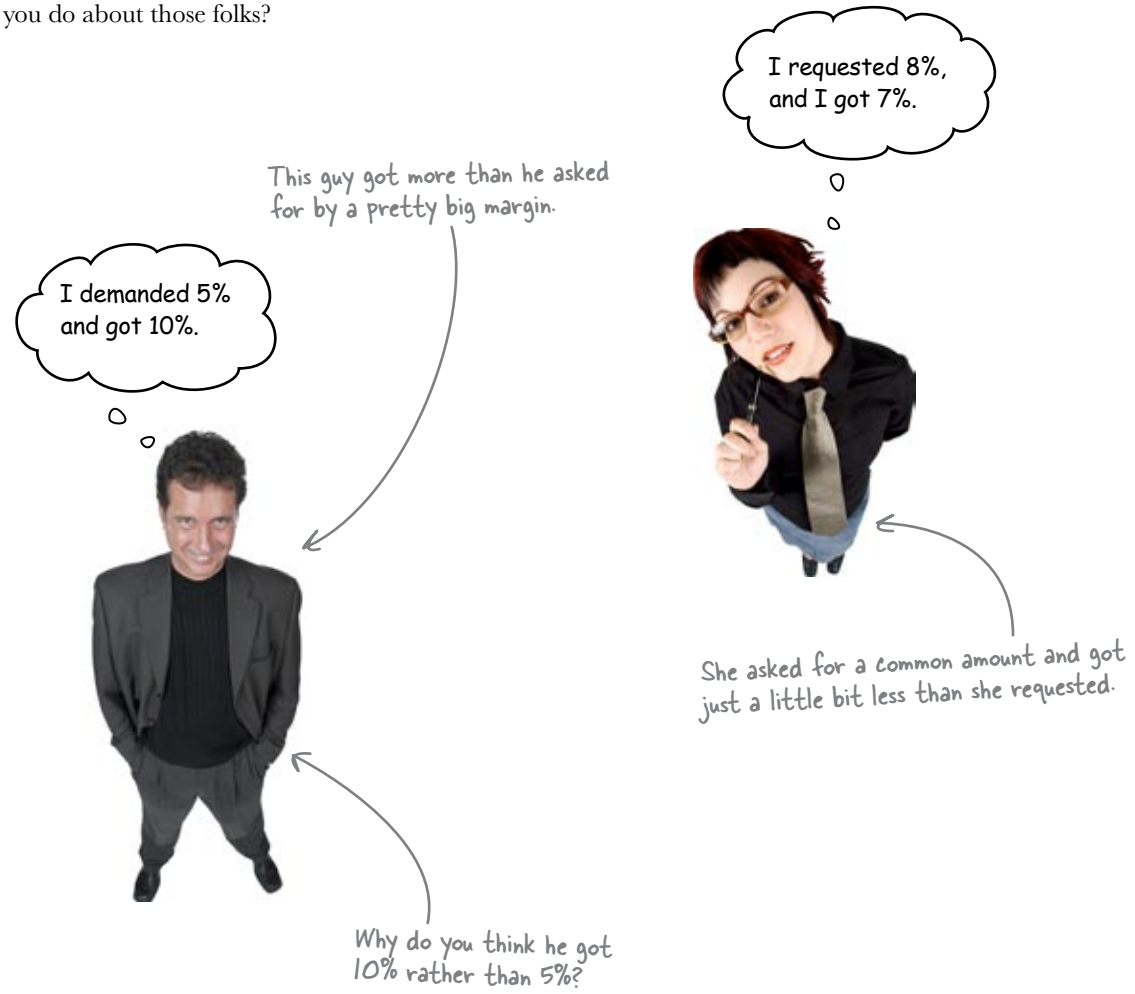
With your new-and-improved regression formula, fewer clients will run with it into the **land of statistical unknowns**.

So does that mean you're finished?

You've only solved part of the problem

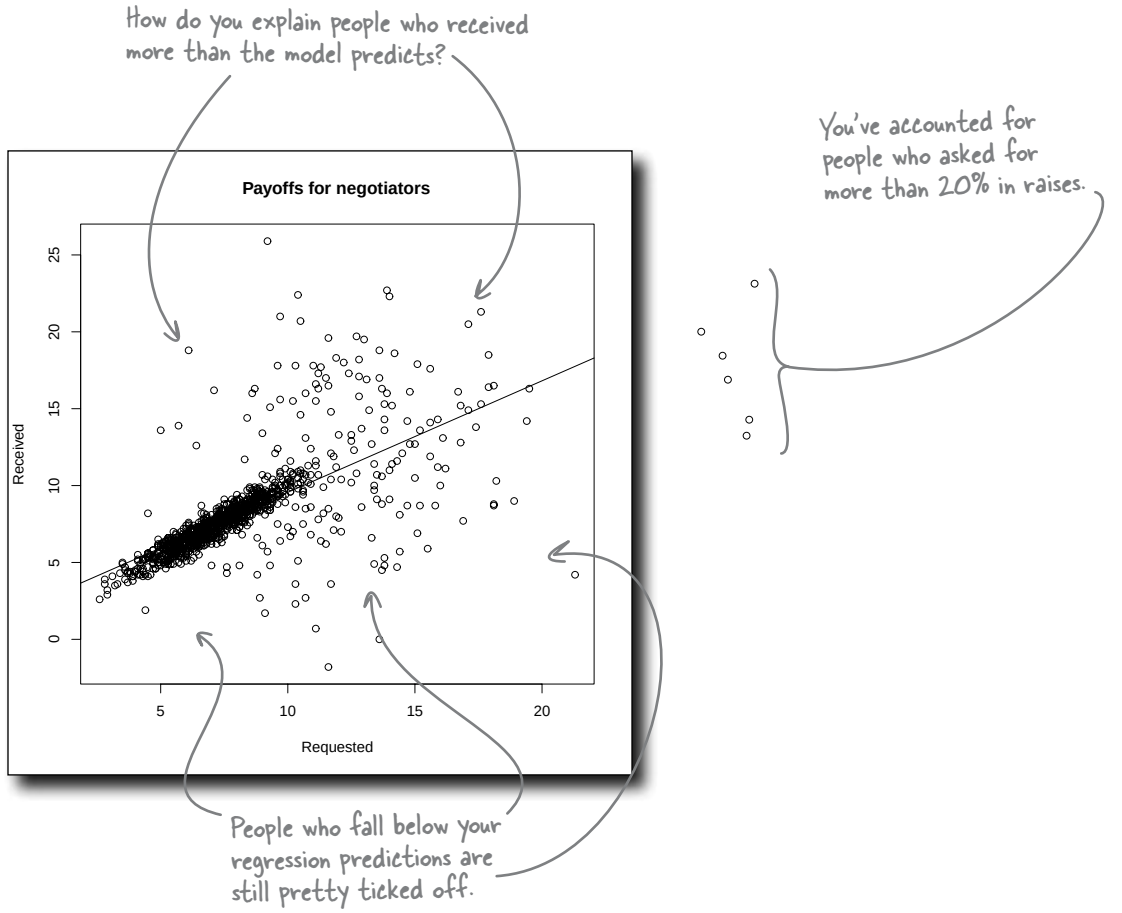
There are still lots of people who got screwy outcomes, even though they requested raise amounts that were inside your data range.

What will you do about those folks?



What does the data for the screwy outcomes look like?

Take another look at your visualization and regression line. Why don't people just get what they ask for?

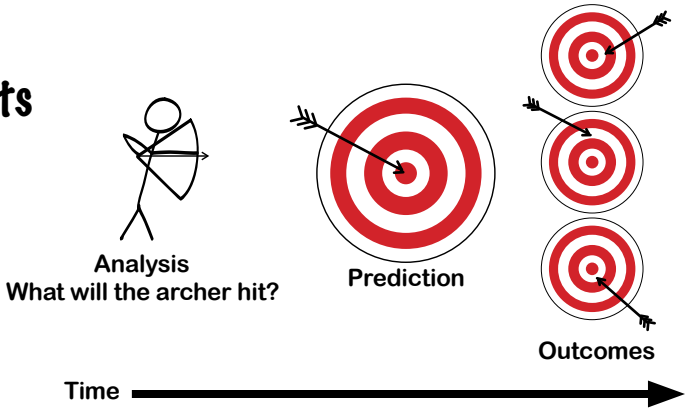


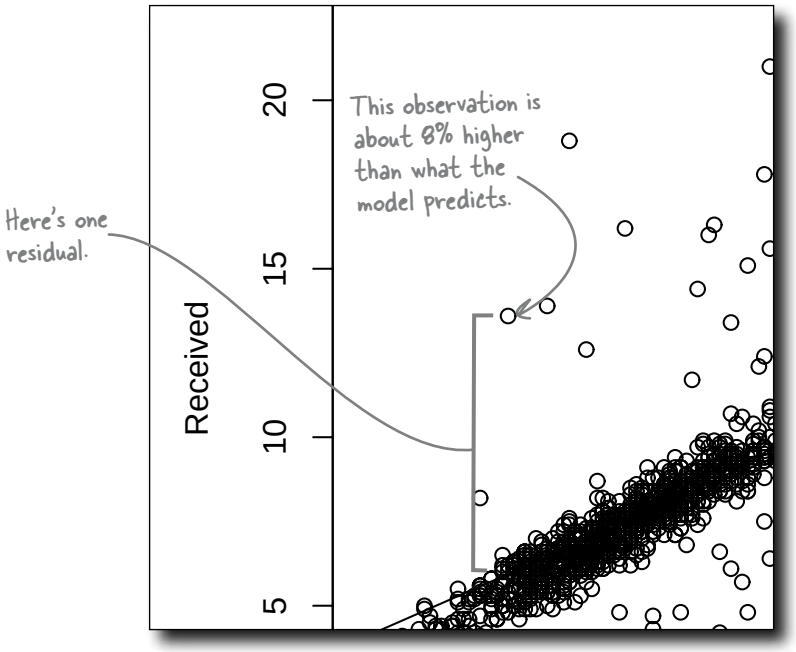
What could be causing these deviations from your prediction?

Chance errors are deviations from what your model predicts

You're always going to be making predictions of one sort or another, whether you do a full-blown regression or not. Those predictions are rarely going to be *exactly* correct, and the amount by which the outcomes deviate from your prediction is called **chance error**.

In statistics, chance errors are also called **residuals**, and the analysis of residuals is at the heart of good statistical modeling.





While you might never have a good explanation for why individuals residuals deviate from the model, you should always look carefully at the residuals on scatterplots.

If you interpret residuals correctly, you'll better understand your data and the use of your model.

You'll always have chance errors in your predictions, and you might never learn why they're in your data.

Sharpen your pencil

Better refine your algorithm some more: this time, you should probably say something about error.

Here are some possible provisions to your algorithm about chance error. Which one would you add to the algorithm?

"You probably won't get what the model predicts because of chance error."

.....

.....

.....

.....

"Your results may vary by a margin of 20 percent more or less than your predicted outcome."

.....

.....

.....

.....

"Only actual results that fit the model results are guaranteed."

.....

.....

.....

.....

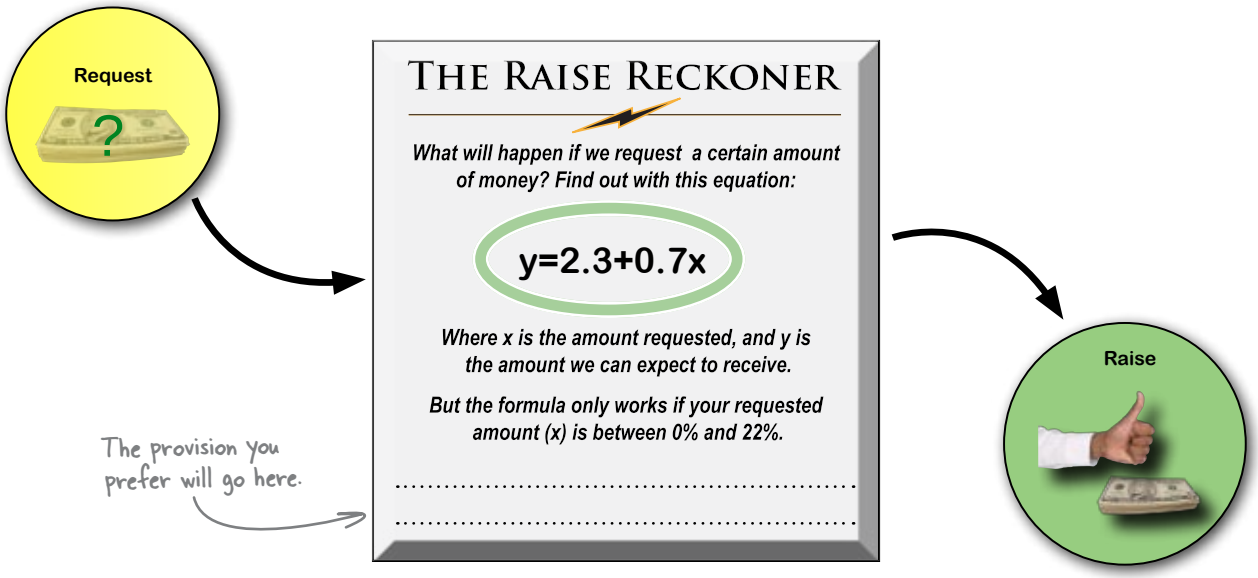
"Please note that your own results may vary from the prediction because of chance error."

.....

.....

.....

.....



Sharpen your pencil Solution

You refined the algorithm to incorporate chance errors. What does it say now?

"You probably won't get what the model predicts because of chance error."

This is true. Probably only a few people will get exactly what the equation returns. But it won't be a very satisfying explanation for the client.

"Your results may vary by a margin of 20 percent more or less than your predicted outcome."

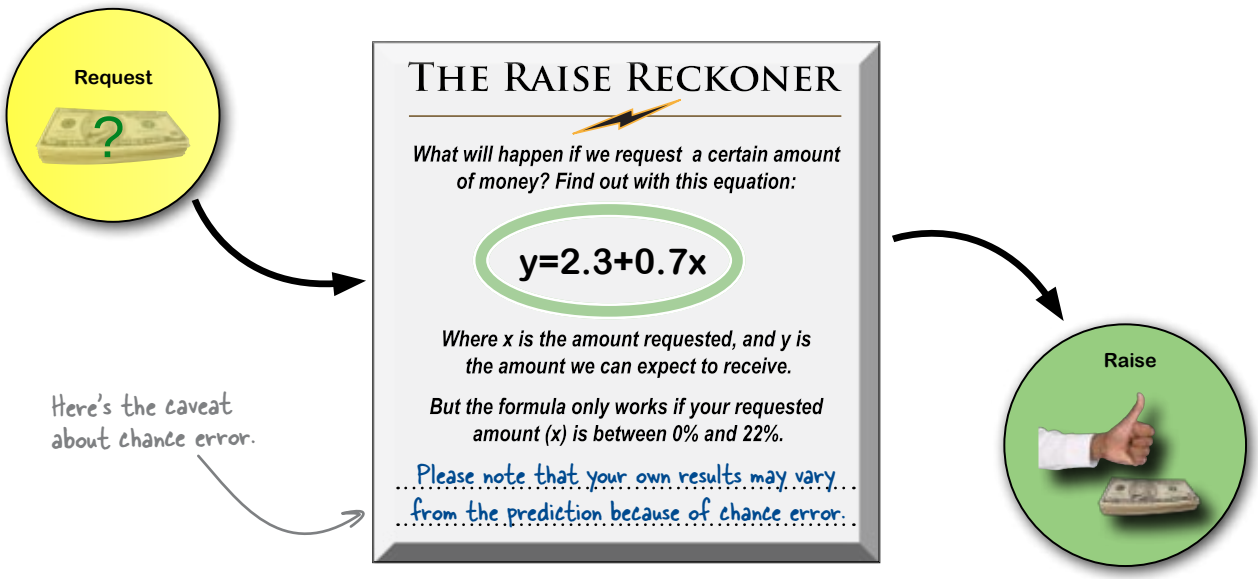
It's good to specify error quantitatively. But what reason do you have to believe the 20% figure? And if it's true, wouldn't you want less error than that?

"Only actual results that fit the model results are guaranteed."

This is just important-sounding nonsense. Your results are only guaranteed if they fit the model prediction? Well what if they don't? That's just silly.

"Please note that your own results may vary from the prediction because of chance error."

True, not terribly satisfying. Until we have some more powerful tools, this statement will have to do.



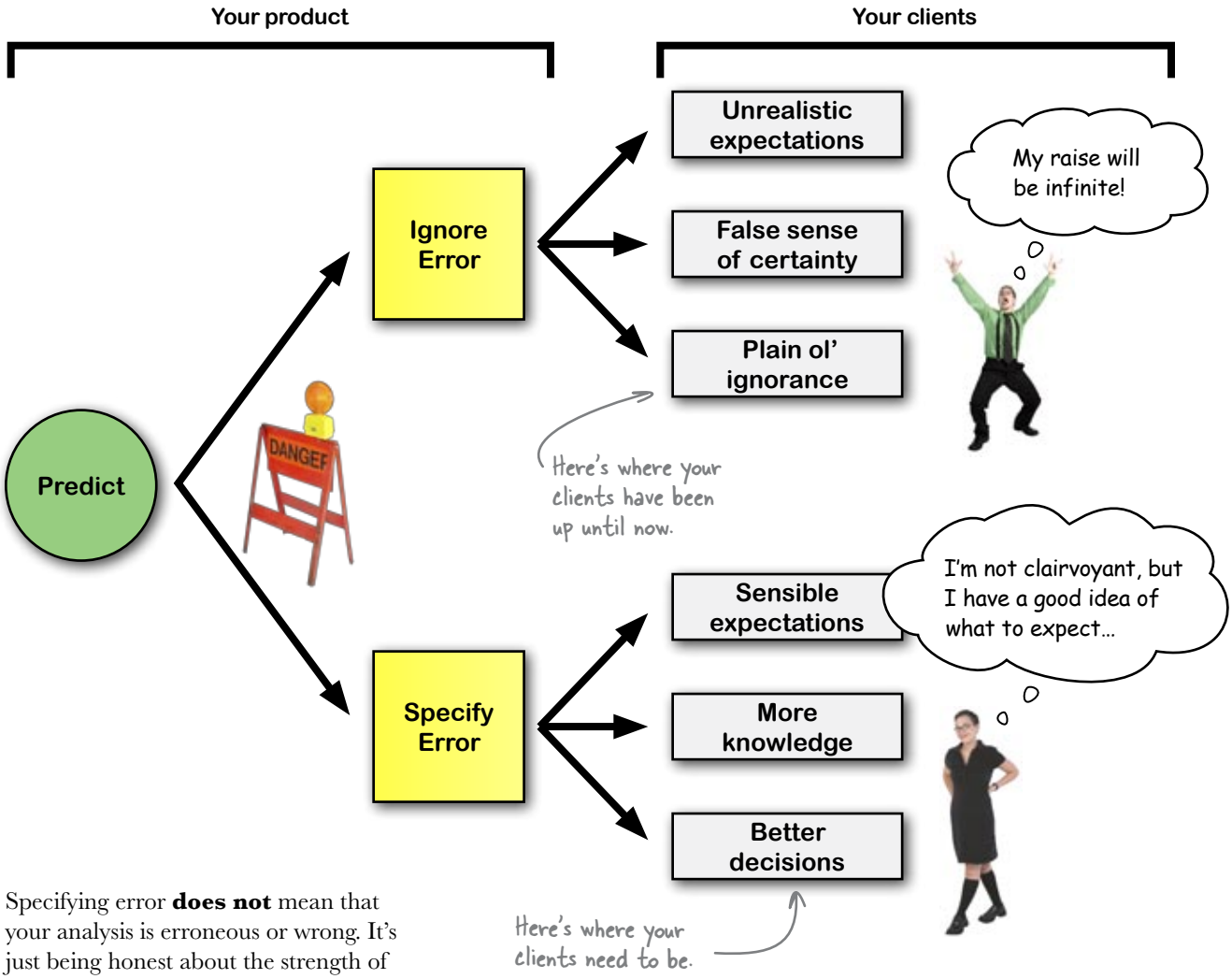
You just lost all your clients.

Hate to break it to ya, but your whole business has just fallen apart. That last line on your compensation algorithm was the difference between people feeling like you were helping them and people feeling like your product was worthless.

How are you going to fix your product?

Error is good for you and your client

The more forthcoming you are about the chance error that your clients should expect in your predictions, the better off both of you will be.



Specifying error **does not** mean that your analysis is erroneous or wrong. It's just being honest about the strength of your predictions. And the more your clients *understand* your predictions, the more likely they'll be to make good decisions using them.

Let's specify error quantitatively...

Chance Error Exposed

This week's interview:
What are the chances?

Head First: Man, you're a pain in the butt.

Chance Error: Excuse me?

Head First: It's just that, because of you, regression will never really be able to make good predictions.

Chance Error: ***What?*** I'm an indispensable part of regression in particular and any sort of measurement generally.

Head First: Well, how can anyone trust a regression prediction as long as you're a possibility? If our clients want to know how much money they'll get when they request a raise, they don't want to hear from us that it's always possible (or even likely!) that what they get will be different from what the model predicts.

Chance Error: You've got me all wrong. Think of me as someone who's always there but who isn't so scary if you just know how to talk about me.

Head First: So "error" isn't necessarily a bad word.

Chance Error: Not at all! There are so many contexts where error specification is useful. In fact, the world would be a better place if people did a better job expressing error often.

Head First: OK, so here's what I'm saying to clients right now. Say someone wants to know what they'll get if they ask for 7 percent in a raise. I say, "The model predicts 7 percent, but chance error means that you probably will get something different from it."

Chance Error: How about you say it like this. If you ask for 7 percent, you'll *probably* get between 6 percent and 8 percent. Doesn't that sound better?

Head First: That doesn't sound so scary at all! Is it really that simple?

Chance Error: Yes! Well, sort of. In fact, getting control of error is a really big deal, and there's a huge range of statistical tools you can use to analyze and describe error. But the most important thing for you to know is that specifying a **range** for your prediction is a heck of a lot more useful (and *truthful*) than just specifying a single number.

Head First: Can I use error ranges to describe subjective probabilities?

Chance Error: You can, and you really, really should. To take another example, which of these guys is the more thoughtful analyst: one who says he believes a stock price will go up 10 percent next year, or one who says he thinks it'll go up between 0–20 percent next year?

Head First: That's a no-brainer. The first guy can't seriously mean he thinks a stock will go up *exactly* 10 percent. The other guy is more reasonable.

Chance Error: You got it.

Head First: Say, where did you say you came from?

Chance Error: OK, the news might not be so good here. A lot of times you'll have no idea where chance error comes from, especially for a single observation.

Head First: Seriously, you mean it's impossible to explain why observations deviate from model predictions?

Chance Error: Sometimes you can explain some of the deviation. For example, you might be able to group some data points together and reduce the chance error. But I'll always be there on some level.

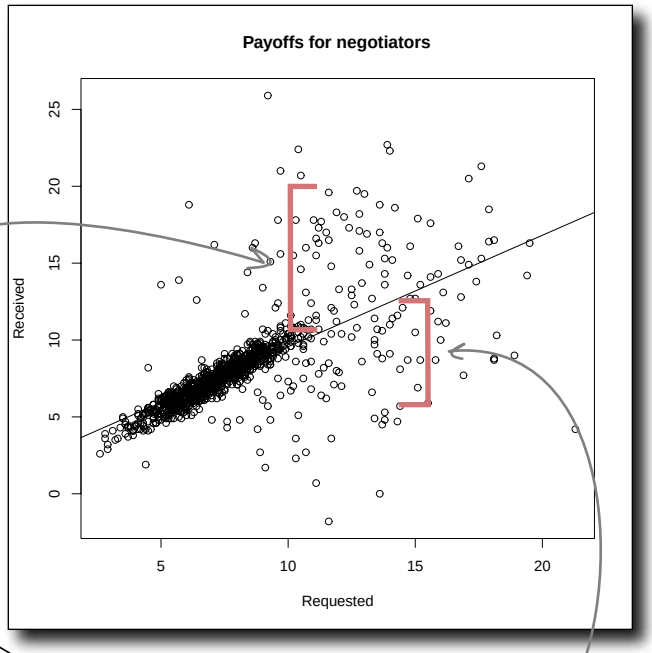
Head First: So should it be my job to reduce you as much as possible?

Chance Error: It should be your job to make your models and analyses have as much explanatory and predictive power as you can get. And that means accounting for me intelligently, not getting rid of me.

Specify error quantitatively

It's a happy coincidence if your observed outcome is exactly what your predicted outcome is, but the real question is what is the spread of the chance error (the **residual distribution**).

What you need is a statistic that shows how far typical points or observations are, *on average*, from your regression line.



The spread or distribution of residuals around the regression line says a lot about your model.

The tighter your observations are around your regression line, the more powerful your line will be.

Hey, that sounds like the standard deviation. The standard deviation describes how far typical points are from the mean observation.

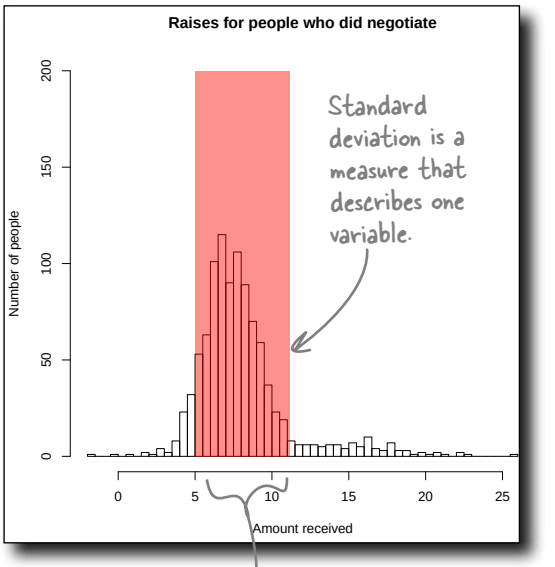
Definitely. The distribution of chance error, or R.M.S. error, around a regression line is a metric you can use just like the standard deviation around a mean.

If you have the value of the R.M.S. error for your regression line, you'll be able to use it to explain to your clients **how far away from the prediction typical outcomes will be**.

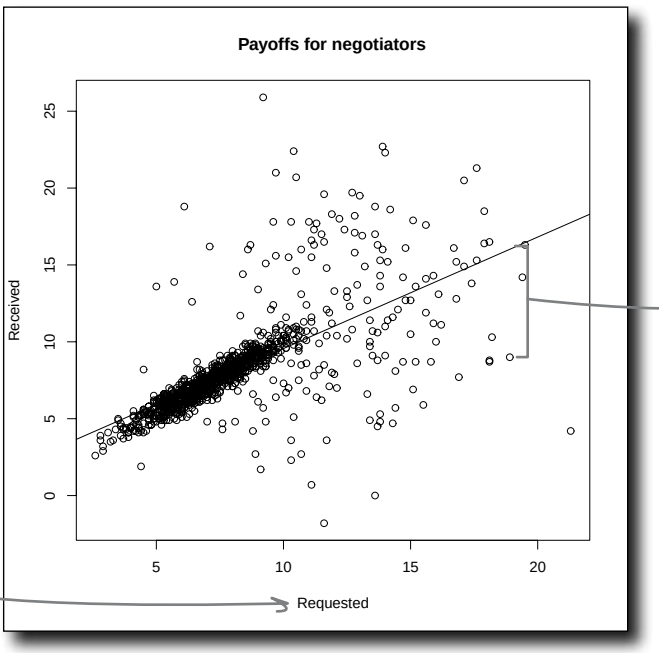
Quantify your residual distribution with Root Mean Squared error

Remember the units that you use for standard deviation? They're the same as whatever's being measured: if your standard deviation of raises received is 5 percent, then typical observations will be 5 percent away from the mean.

It's the same deal with R.M.S. error. If, say, your R.M.S. error for predicting Received from Requested is 5 percent, then the typical observation will be 5 percent away from whatever value the regression equation predicts.



The standard deviation describes the spread around the mean.



The R.M.S. error describes the spread from the regression line.

R.M.S. error refers to the relation between two variables.

So how do you calculate the R.M.S. error?

Your model in R already knows the R.M.S. error

The linear model object your created inside of R in the last chapter doesn't just know the y-axis intercept and slope of your regression line.

It has a handle on all sorts of statistics pertaining to your model, including the R.M.S. error. If you don't still have the myLm object you created in R, type in this function before the next exercise:

Make sure you have the most current data loaded.

```
employees <- read.csv("http://www.headfirstlabs.com/books/hfda/
      hfda_ch10_employees.csv", header=TRUE)
myLm <- lm(received[negotiated==TRUE]~
      requested[negotiated==TRUE], data=employees)
```

Behind the Scenes

Under the hood, R is using this formula to calculate the R.M.S. error:

$$\sigma_y * \sqrt{1-r^2}$$

The standard deviation of y.

The correlation coefficient.

there are no Dumb Questions

Q: Do I need to memorize that formula?

A: As you'll see in just a second, it's pretty easy to calculate the R.M.S. error inside of R or any other statistical software package. What's most important for you to know is that error can be described and used quantitatively, and that you should always be able to describe the error of your predictions.

Q: Do all types of regression use this same formula to describe error?

A: If you get into nonlinear or multiple regression, you'll use different formulas to specify error. In fact, even within linear regression there are more ways of describing variation than R.M.S. error. There are all sorts of statistical tools available to measure error, depending on what you need to know specifically.

Test Drive

Instead of filling in the algebraic equation to get the R.M.S. error, let's have R do it for us.

Take a look at R's summary of your model by entering this command:

```
summary(myLm)
```

Your R.M.S. error will be in the output, but you can also type this to see the error:

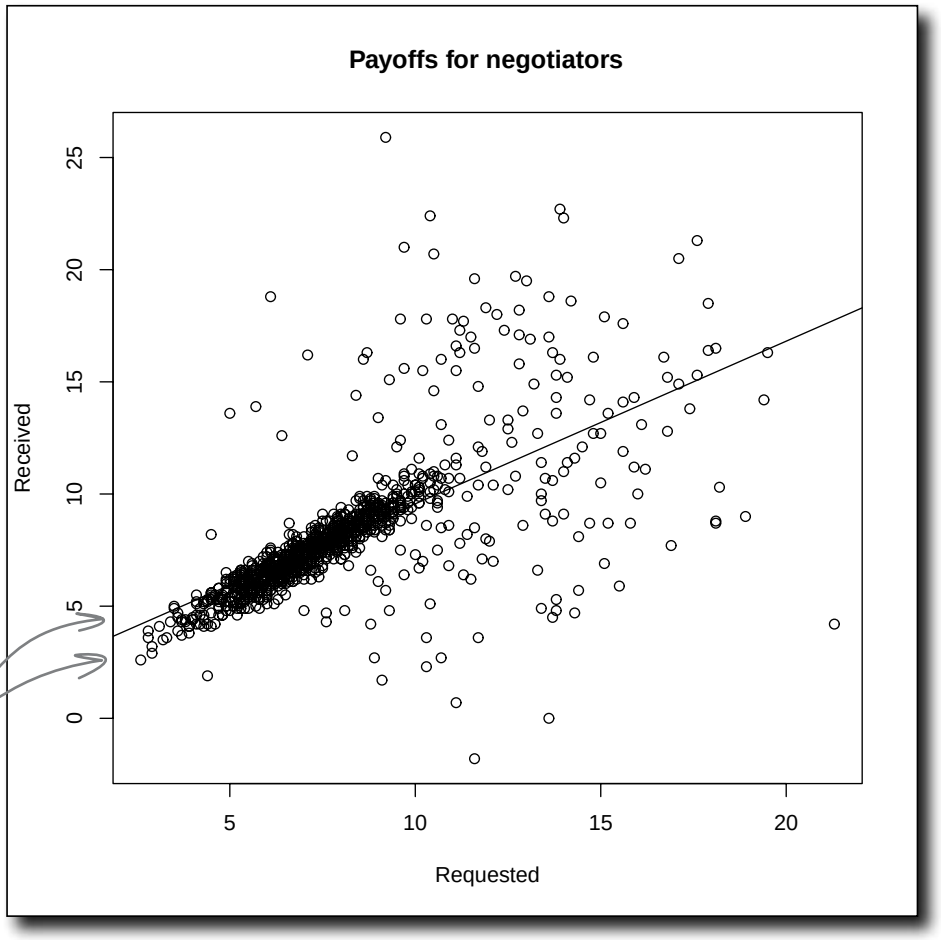
```
summary(myLm)$sigma
```

The R.M.S. error is also called "sigma" or "residual standard error."

Next, color in an error band across your entire regression line to represent your R.M.S. error.

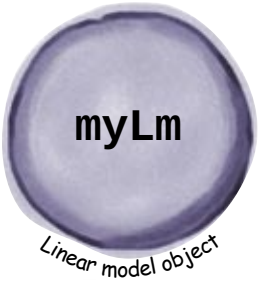
The error band should follow along the regression line and the thickness above and below the line should be equal to one R.M.S. error.

Start your error band here.



R's summary of your linear model shows your R.M.S. error

When you ask R to summarize your linear model object, it gives you a bunch of information about what's inside the object.



Here's a summary of your model.

```
> summary(myLm)

Call:
lm(formula = received[negotiated == TRUE] ~ requested[negotiated ==
    TRUE], data = employees)

Residuals:
     Min       1Q   Median       3Q      Max
-13.5560  -0.5914  -0.0601   0.3879  16.9173

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      2.31213    0.21775   10.62   <2e-16 ***
requested[negotiated == TRUE]    0.72507    0.02573   28.18   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.298 on 998 degrees of freedom
Multiple R-squared: 0.4431,  Adjusted R-squared: 0.4425
F-statistic:   794 on 1 and 998 DF,  p-value: < 2.2e-16

>
```

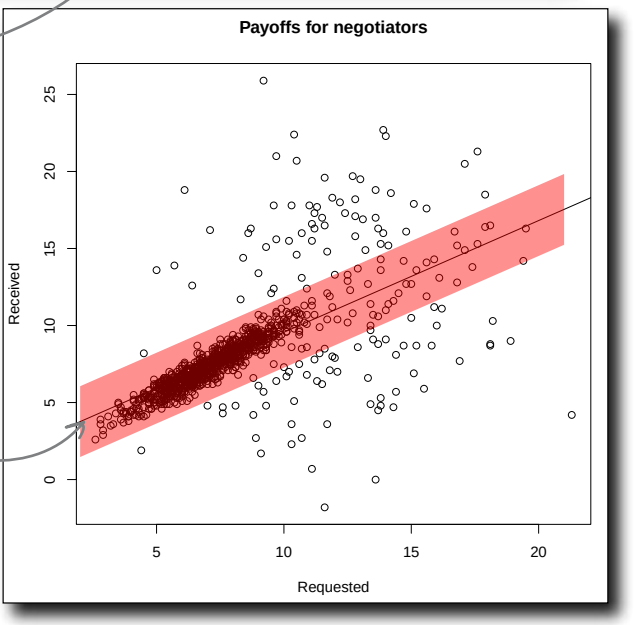
R has all sorts of things to tell you about your linear model.

These are the slope and intercept of your regression line.

Not only do you see your regression coefficients, like you saw in the previous chapter, but you also see the R.M.S. error and a bunch of other statistics to describe the model.

And here's your R.M.S. error!

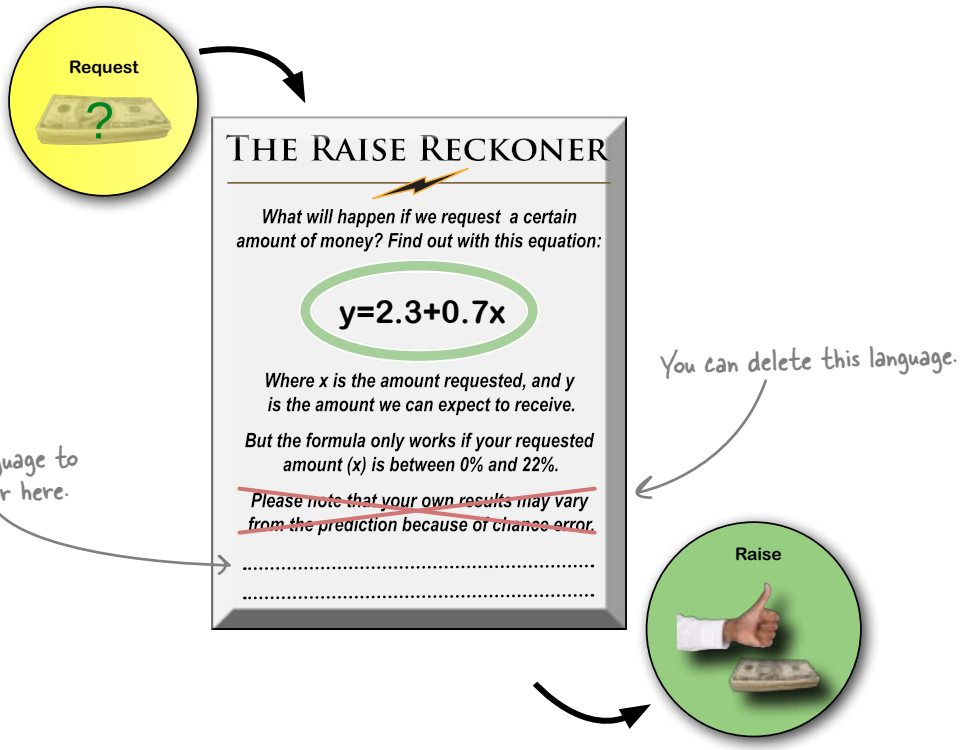
If you draw a band that's about 2.3 percentage points above and below your regression line, you get a spread that looks like this.



Sharpen your pencil

You're ready to have another go at your compensation algorithm. Can you incorporate a more nuanced conception of chance error?

How would you change this algorithm to incorporate your R.M.S. error? Write your answer inside the Raise Reckoner.



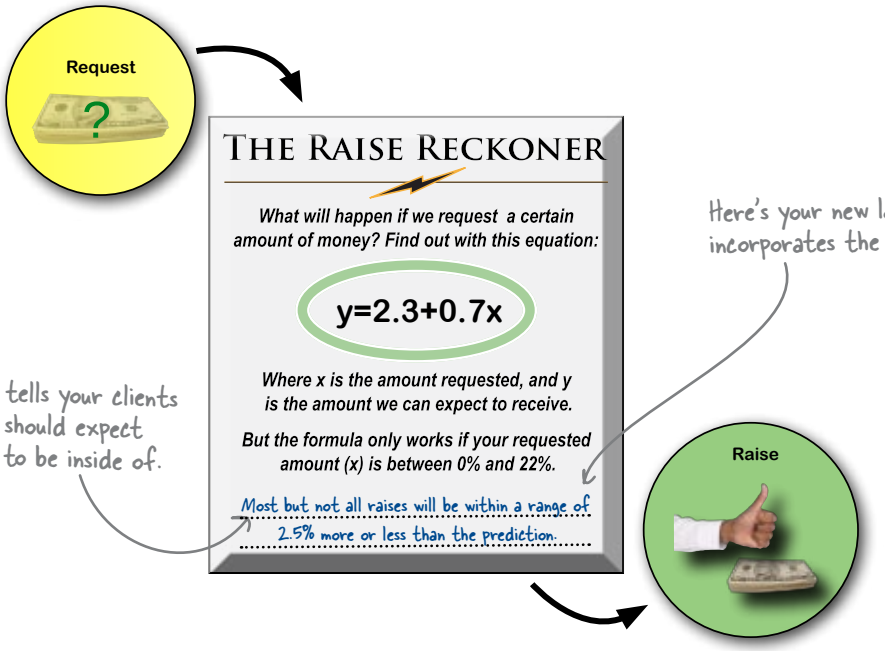

```
Signif. codes:  0 '***' 0.001 '

Residual standard error: 2.298
Multiple R-squared: 0.4431,  Ad
```

Use the R.M.S. error to improve your algorithm.

Sharpen your pencil Solution

Let's take a look at your new algorithm, complete with R.M.S. error for your regression.



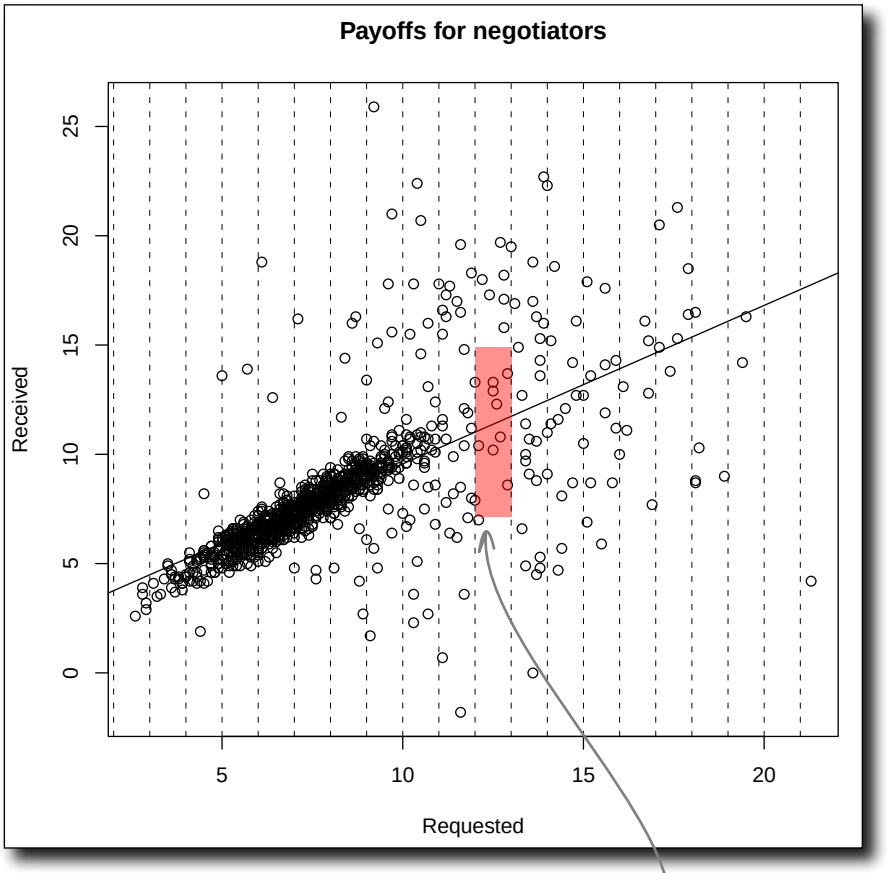
So if I ask for 7%, I'll get 4.5—9.5% back? I just need more than that if you want me to take you seriously. Can you give me a prediction with a lower amount of error, please?

She has a point.

Is there anything you can do to make this regression more useful? Can you look at your data in a way that reduces the error?

Look at different strips on your scatterplot. Is the R.M.S. error different at the various strips along the regression line?

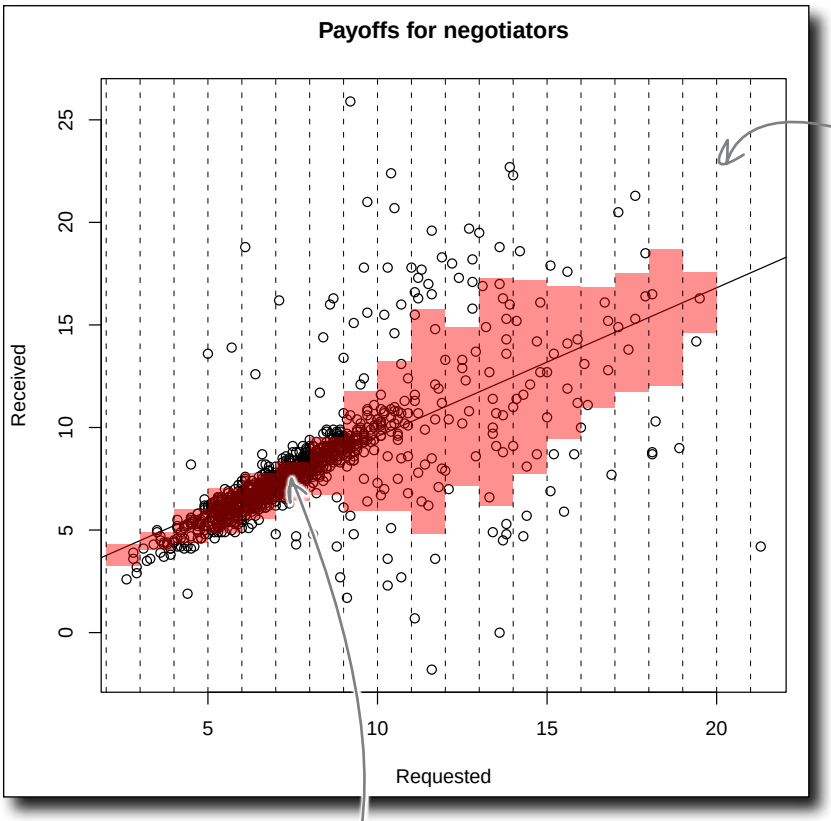
For each strip on the scatterplot, color in what you think the error is within that strip.



Do you see **segments** where the residuals are fundamentally different?

Exercise Solution

You've looked at the R.M.S. error for each strip. What did you find?



The error is a lot higher over here.

The error is lowest around here.

Why is the error higher on the right side?

Look at the data and think about what it must mean.

Jim: Oh man, that's nuts! It looks like there's a different spread of predictions for every strip along the scatterplot!

Joe: Yeah, that's crazy. Seriously. How in the world do we explain that to our customers?

Jim: They'll never buy it. If we say to them, your error is looking relatively low at 7–8 percent, but at 10–11 percent the error is through the roof, they just won't get it.

Frank: Hey, relax you guys. Maybe we should ask *why* the error bands look the way they do. It might help us understand what's happening with all these raises.

Jim: [*Scoff*] There you go being all circumspect again.

Frank: Well, we're *analysts*, right?

Joe: Fine. Let's look at what people are asking for. At the start of the scale, there's kind of a big spread that narrows as soon as we hit 5 percent or so.

Jim: Yeah, and there are only 3 people who asked for less than 5 percent, so maybe we shouldn't put too much stock in that error from 4–5 percent.

Frank: Excellent! So now we're looking at the range from 5 percent all the way up to about 10 percent. The error is lowest there.

Joe: Well, people are being conservative about what they're asking for. And their bosses are reacting, well, conservatively.

Frank: But then you get over 10 percent…

Jim: And who knows what'll happen to you. Think about it. 15 percent is a big raise. I wouldn't have the guts to request that. Who knows what my boss would do?

Frank: Interesting hypothesis. Your boss might reward you for being so bold, or she might kick your butt for being so audacious.

Jim: Once you start asking for a *lot* of money, anything can happen.

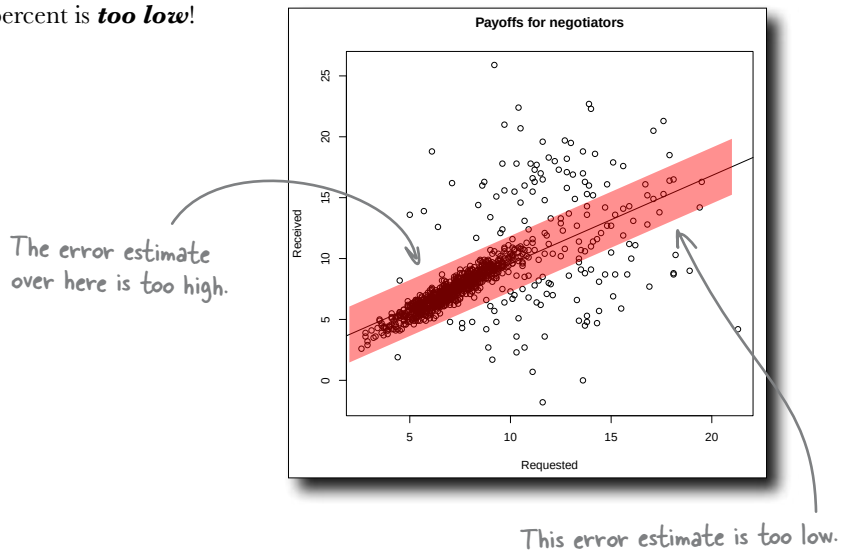
Joe: You know, guys, I think we've got two different groups of people in this data. In fact, I think we may even have two different **models**.

What would your analysis look like if you split your data?

Segmentation is all about managing error

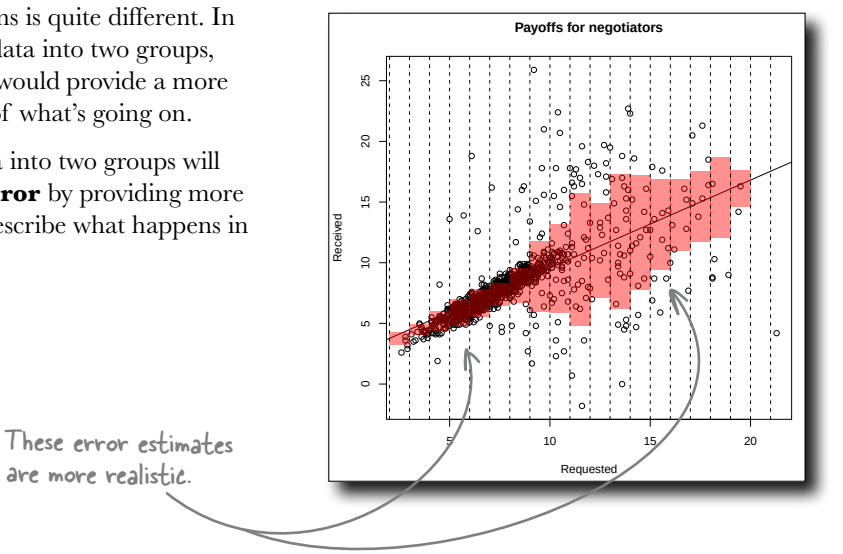
Splitting data into groups is called **segmentation**, and you do it when having multiple predictive models for subgroups will result in less error over all than one model.

On a single model, the error estimate for people who ask for 10 percent or less is ***too high***, and the error estimate for people who ask for more than 10 percent is ***too low***!



When we looked at the strips, we saw that the error in the two regions is quite different. In fact, segmenting the data into two groups, giving each a model, would provide a more realistic explanation of what's going on.

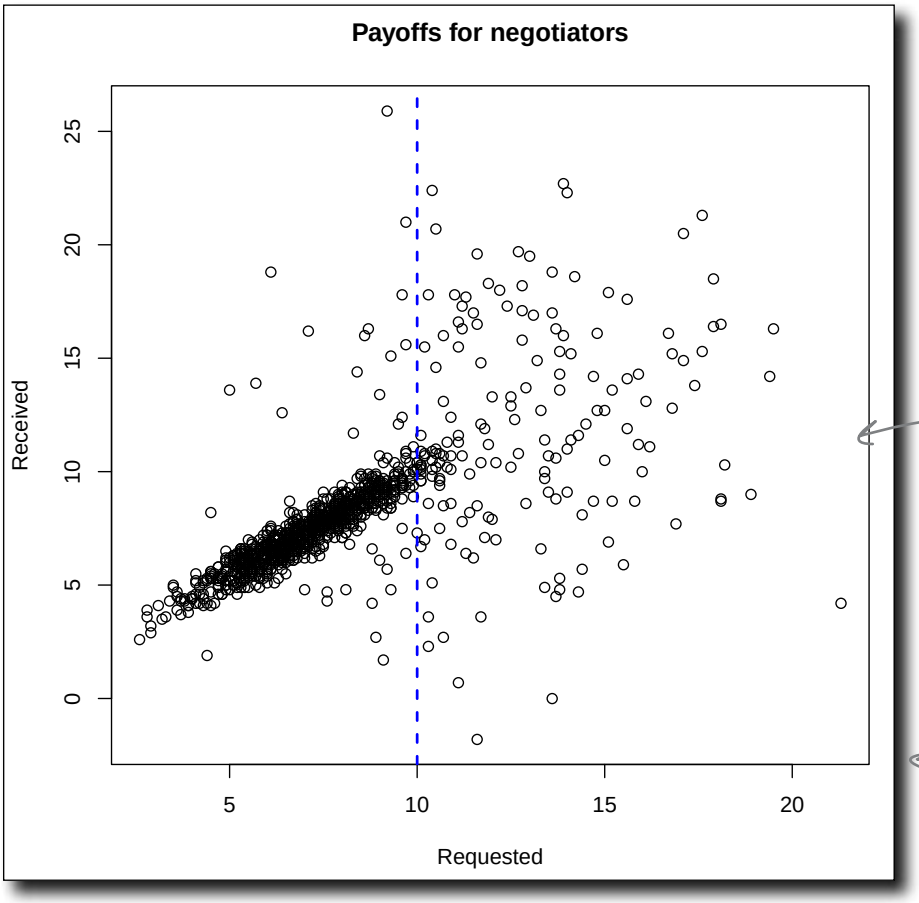
Segmenting your data into two groups will help you **manage error** by providing more sensible statistics to describe what happens in each region.



Sharpen your pencil

If you segment your data between people who requested less than 10 percent and people who requested more than 10 percent, chances are, your regression lines will look different.

Here's the split data. Draw what you think the regression lines are for these two sets of data.



Hint: the dots spread out on the right side. That's OK—just do your best to estimate where the line goes.

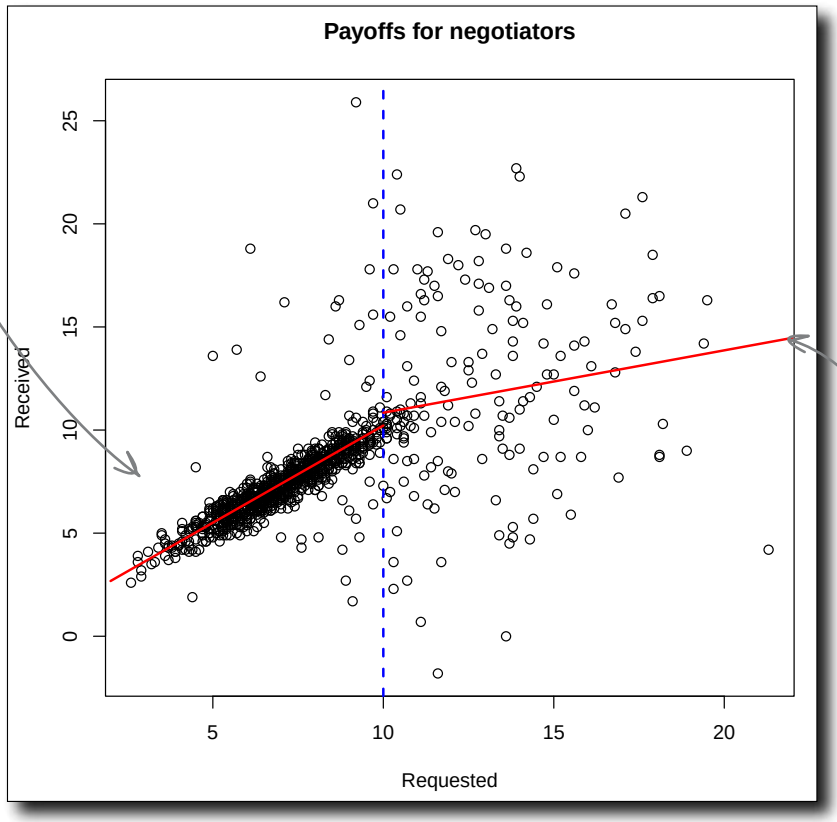
Remember: the regression line is the line that best fits the graph of averages.

Sharpen your pencil Solution

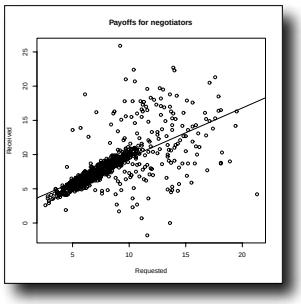
You've created two regression lines—two separate models!

What do they look like?

This line through the people who make low requests should fit the data much better than the original model.



The regression line through the more aggressive negotiators should have a different slope from the other line.



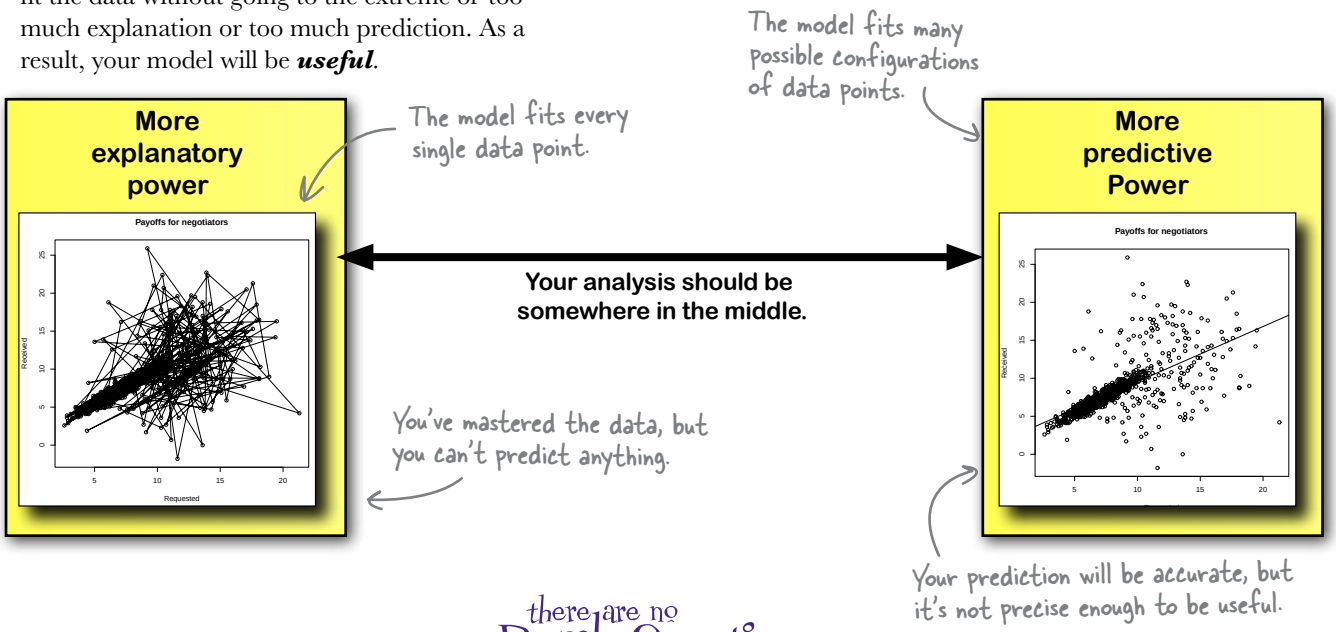
Here's your original model.

BRAIN BARBELL

This is a good one. Why stop at two regression lines? Would having more lines—a *lot* more, say—make your model more useful?

Good regressions balance explanation and prediction

Two segments in your raise regression will let you fit the data without going to the extreme of too much explanation or too much prediction. As a result, your model will be ***useful***.



there are no Dumb Questions

Q: Why would I stop at splitting the data into 2 groups? Why not split them into 5 groups?

A: If you've got a good *reason* to do it, then go right ahead.

Q: I could go nuts and split the data into 3,000 groups. That's as many "segments" as there are data points.

A: You certainly could. And if you did, how powerful do you think your 3,000 regressions would be at predicting people's raises?

Q: Ummm...

A: If you did that, you'd be able to explain everything. All your data points would be accounted for, and the R.M.S. error of your regression equations would all be zero. But your models would have lost all ability to predict anything.

Q: So what would an analysis look like that had a whole lot of predictive power but not a lot of explanatory power?

A: It'd look something like your first model. Say your model was this: "No matter what you ask for, you'll receive somewhere between -1,000 percent and 1,000 percent in raises."

Q: That just sounds dumb.

A: Sure, but it's a model that has *incredible* predictive power. The chances are that no one you ever meet will be outside that range. But the model doesn't *explain* anything. With a model like that, you sacrifice explanatory power to get predictive power.

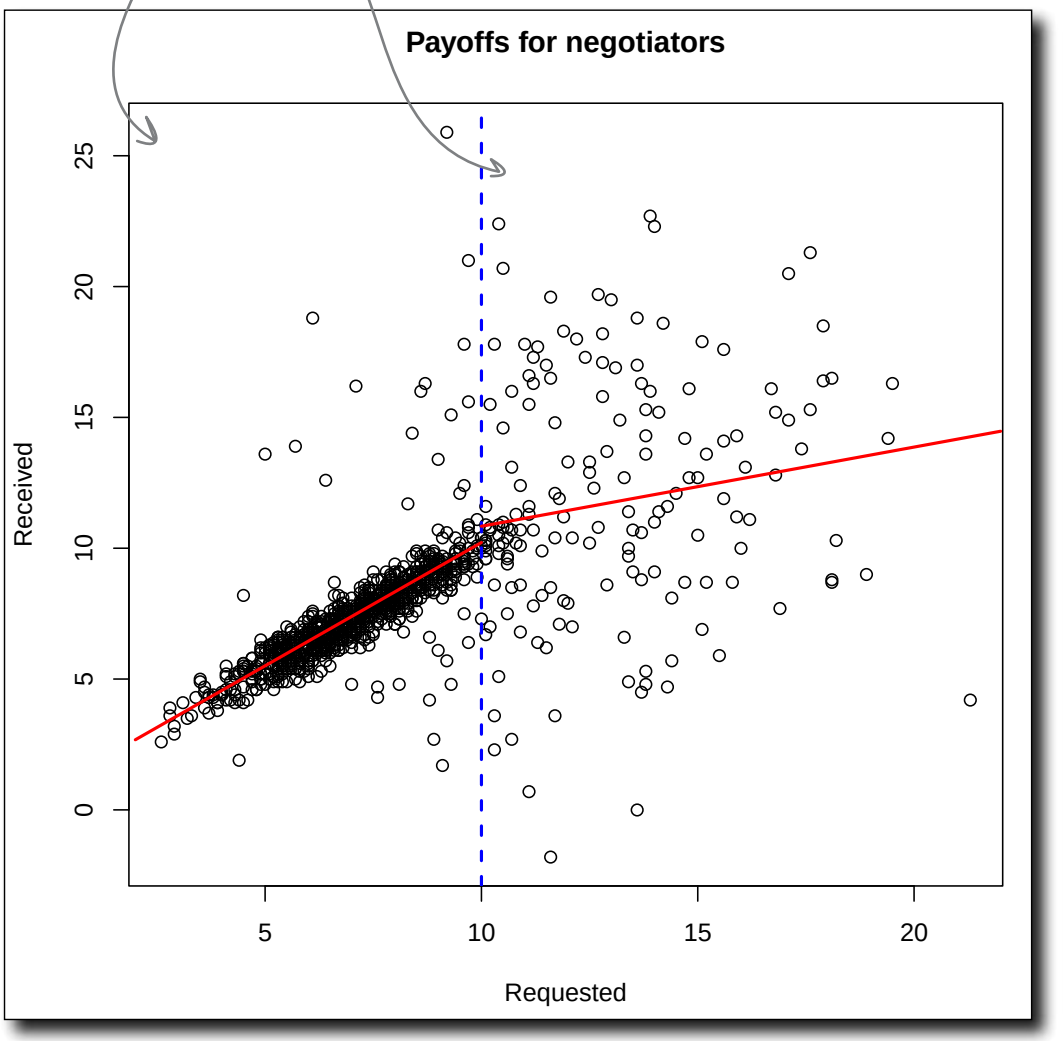
Q: So that's what zero error looks like: no ability to predict anything.

A: That's it! Your analysis should be somewhere between having complete explanatory power and complete predictive power. And where you fall between those two extremes has to do with your best judgemnt as an analyst. What sort of model does your client need?

Sharpen your pencil

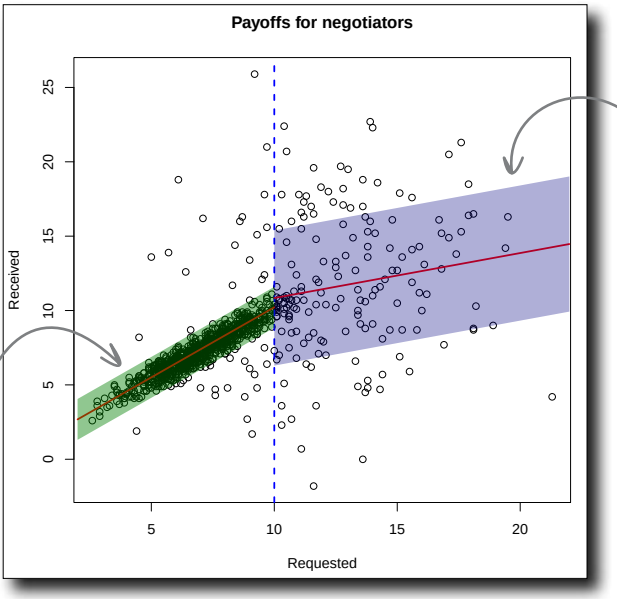
For each of these two models, color in bands that represent R.M.S. error.

Draw bands to describe the distribution of residuals for each model.



Your segmented models manage error better than the original model

They're more powerful because they do a better job of describing what's actually happening when people ask for raises.

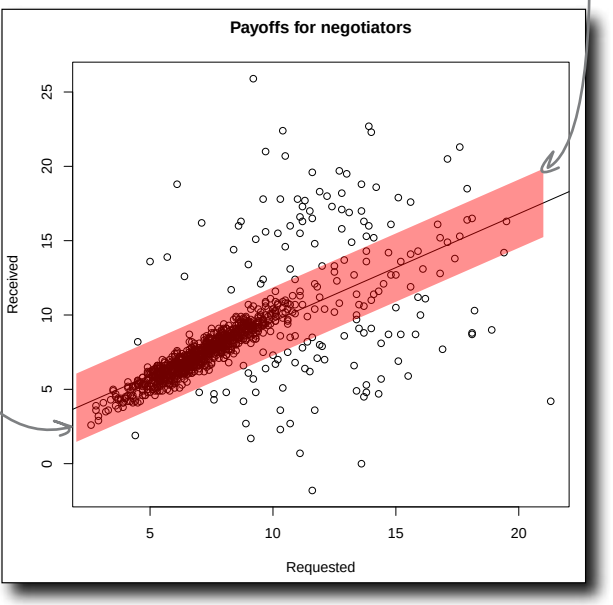


Aggressive negotiators

Your new model for **timid negotiators** does a better job fitting the data.

The slope of the regression line is more on target, and the R.M.S. error is lower.

Timid negotiators



Your new model for **aggressive negotiators** is a better fit, too.

The slope is more on target, and the R.M.S. error is higher, which more accurately represents what people experience when they ask for more than 10 percent.

Let's implement these models in R...

It's time to implement those new models and segments in R. Once you have the models created, you'll be able to use the coefficients to refine your raise prediction algorithm.

Create new linear model objects that correspond to your two segments by typing the following at the command line:

This code tells R to look only at the data in your database for negotiators...

```
myLmBig <- lm(received[negotiated==TRUE & requested > 10]~
       requested[negotiated==TRUE & requested > 10],
       data=employees)
myLmSmall <- lm(received[negotiated==TRUE & requested <= 10]~
       requested[negotiated==TRUE & requested <= 10],
       data=employees)
```

...and to split the segments at the 10% raise request range.

Look at the summaries of both linear model objects using these versions of the `summary()` function. Annotate these commands to show what each one does:

```
summary(myLmSmall)$coefficients
summary(myLmSmall)$sigma
summary(myLmBig)$coefficients
summary(myLmBig)$sigma
```

These results will make your algorithm much more powerful.

Exercise Solution

You just ran two new regressions on segmented data. What did you find?

When you tell R to create the new models, R doesn't display anything in the console.

But quite a lot happens behind the scenes!

R Console

```
> myLmBig <- lm(received[negotiated==TRUE & requested > 10]~ requested[negotiated==TRUE &
requested > 10], data=employees)
> myLmSmall <- lm(received[negotiated==TRUE & requested <= 10]~requested[negotiated==TRUE &
requested <= 10],data=employees)
> summary(myLmSmall)$coefficients
                                              Estimate Std. Error   t value      Pr(>|t|)
(Intercept)                                  0.7933468 0.22472009  3.530378  4.378156e-04
requested[negotiated == TRUE & requested <= 10] 0.9424946 0.03151835 29.903041 6.588020e-134
> summary(myLmSmall)$sigma
[1] 1.374526
> summary(myLmBig)$coefficients
                                             Estimate Std. Error  t value     Pr(>|t|)
(Intercept)                                 7.8134033  1.8760371 4.164845 4.997597e-05
requested[negotiated == TRUE & requested > 10] 0.3026090  0.1420151 2.130824 3.457618e-02
> summary(myLmBig)$sigma
[1] 4.544424
>
```

Here are the R.M.S. errors for your new models.

Here are the slope and intercepts for your new regression lines.

Sharpen your pencil

You now have everything you need to create a much more powerful algorithm that will help your customers understand what to expect no matter what level of raise they request. Time to toss out the old algorithm and incorporate everything you've learned into the new one.

Using the slopes and intercepts of your new models, write the equations to describe both of them.

.....

.....

For what levels of raises does each model apply?

Don't forget about avoiding extrapolation!

.....

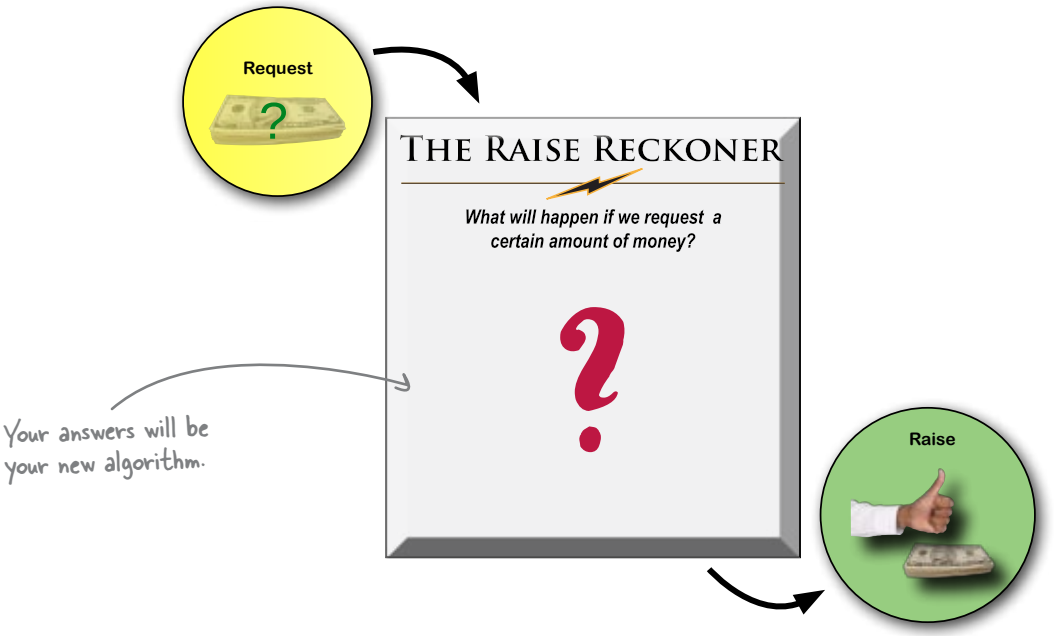
.....

How close to the prediction should your client expect her own raise to be, depending on which model she uses?

Think about the R.M.S. error.

.....

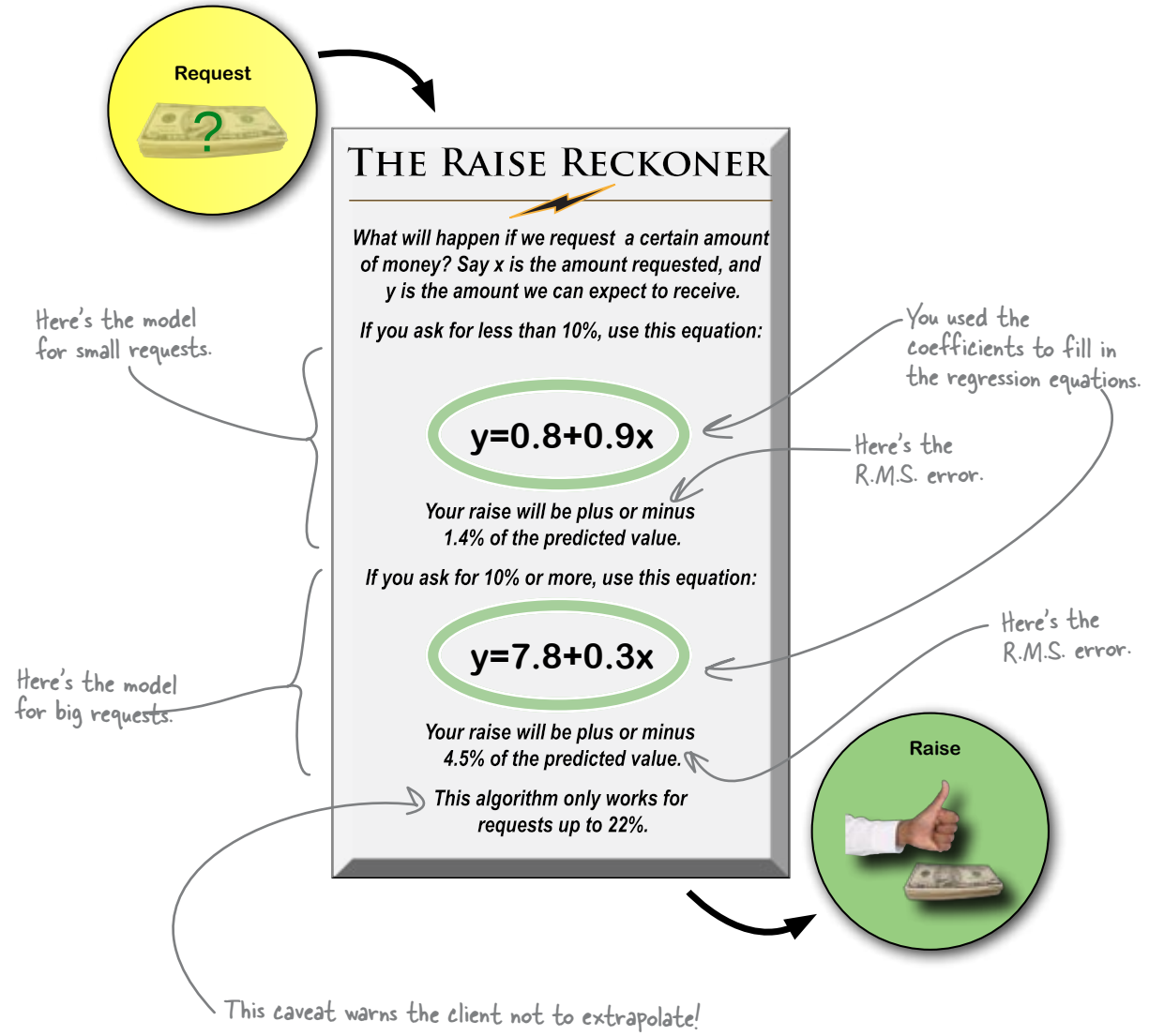
.....



Your answers will be your new algorithm.

Sharpen your pencil Solution

What is your final compensation algorithm?



Your clients are returning in droves

Your new algorithm is really starting to pay off, and everyone's excited about it.

Now people can decide whether they want to take the riskier strategy of asking for a lot of money or just would rather play it safe and ask for less.

The people who want to play it safe are getting what they want, and the risk-takers understand what they're getting into when they ask for a lot.

12 relational databases

Can you relate?

How do you structure really, really multivariate data?

A spreadsheet has only *two dimensions*: rows and columns. And if you have a bunch of dimensions of data, the **tabular format** gets old really quickly. In this chapter, you're about to see firsthand where spreadsheets make it really hard to manage multivariate data and learn **how relational database management systems** make it easy to store and retrieve countless permutations of multivariate data.

The Dataville Dispatch wants to analyze sales

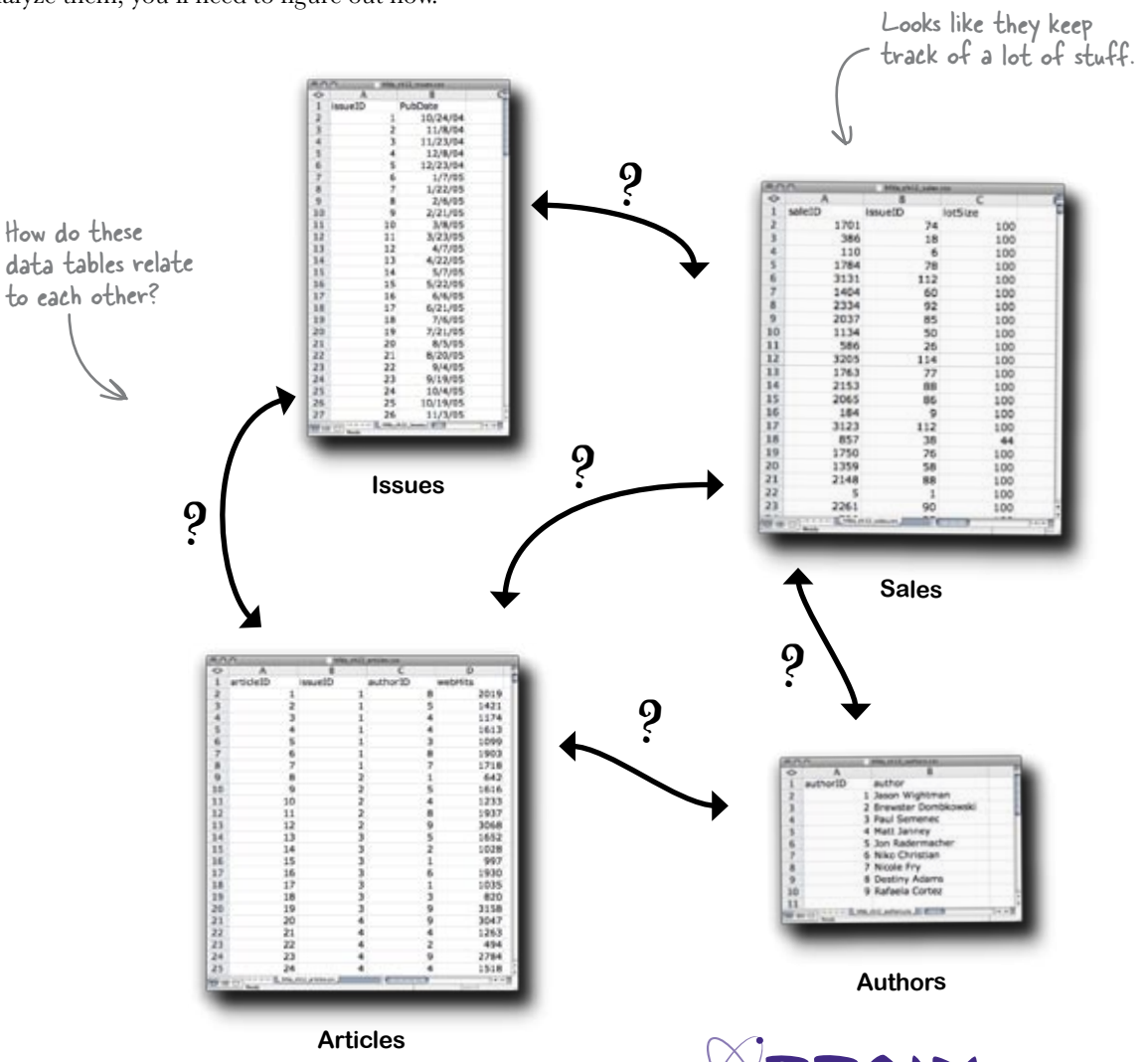
The *Dataville Dispatch* is a popular news magazine, read by most of Dataville's residents. And the *Dispatch* has a very specific question for you: they want to tie the number of articles per issue to sales of their magazine and find an optimum number of articles to write.

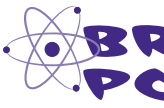
They want each issue be as cost effective as possible. If putting a hundred articles in each issue doesn't get them any more sales than putting fifty articles in each issue, they don't want to do it. On the other hand, if fifty article issues correlate to *more* sales than ten article issues, they'll want to go with the fifty articles.

They'll give you **free advertising** for your analytics business for a year if you can give them a thorough analysis of these variables.

Here's the data they keep to track their operations

The *Dispatch* has sent you the data they use to manage their operations as four separate spreadsheet files. The files all relate to each other in **some way**, and in order to analyze them, you'll need to figure out how.





What do you need to know in order to compare articles to sales?

You need to know how the data tables relate to each other

The table or tables you create to get the answers that the *Dispatch* wants will tie **article count** to **sales**.

So you need to know *how* all these tables relate to each other. What specific data fields tie them together? And beyond that, what is the **meaning** of the relationships?

Here is what the Dispatch has to say about how they maintain their data.

From: Dataville Dispatch

To: Head First

Subject: About our data

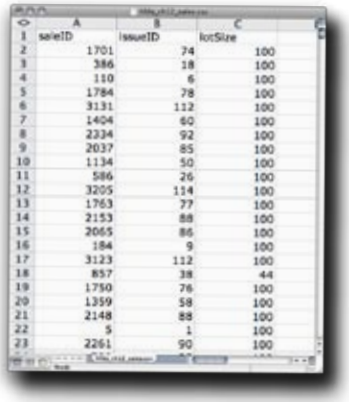
Well, each issue of the magazine has a bunch of articles, and each article has an author, so in our data we tie the authors to the articles. When we have an issue ready, we call our list of wholesalers. They place orders for each issue, which we record in our sales table. The "lot size" in the table you're looking at counts the number of copies of that issue that we sell—usually in denominations of 100, but sometimes we sell less. Does that help?

— DD

They have a lot of stuff to record, which is why they need all these spreadsheets.

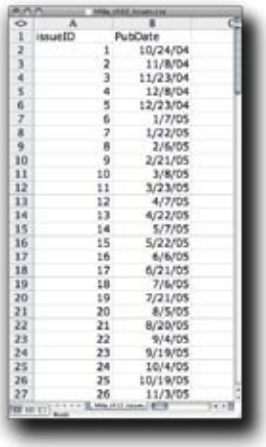
Sharpen your pencil

Draw arrows and use words to describe the relationship between the things being recorded in each spreadsheet.



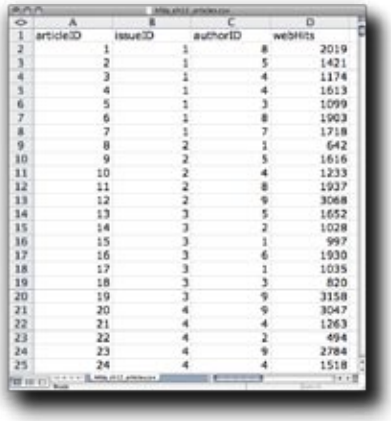
	A	B	C
1	saleID	issueID	lotSize
2	1701	74	100
3	386	18	100
4	110	6	100
5	1784	78	100
6	3131	112	100
7	1404	60	100
8	2334	92	100
9	2037	85	100
10	1134	50	100
11	586	26	100
12	3205	114	100
13	1763	77	100
14	2153	88	100
15	2065	86	100
16	184	9	100
17	3123	112	100
18	857	38	44
19	1750	76	100
20	1359	58	100
21	2148	88	100
22	5	1	100
23	2261	90	100

Sales



	A	B
1	issueID	PubDate
2	1	10/24/04
3	2	11/8/04
4	3	11/23/04
5	4	12/8/04
6	5	12/23/04
7	6	1/7/05
8	7	1/22/05
9	8	2/6/05
10	9	2/21/05
11	10	3/8/05
12	11	3/23/05
13	12	4/7/05
14	13	4/22/05
15	14	5/7/05
16	15	5/22/05
17	16	6/6/05
18	17	6/21/05
19	18	7/6/05
20	19	7/21/05
21	20	8/5/05
22	21	8/20/05
23	22	9/4/05
24	23	9/19/05
25	24	10/4/05
26	25	10/19/05
27	26	11/3/05

Issues



	A	B	C	D
1	articleID	issueID	authorID	webHits
2	1	1	8	2019
3	2	1	5	1421
4	3	1	4	1174
5	4	1	4	1613
6	5	1	3	1099
7	6	1	8	1903
8	7	1	7	1718
9	8	2	1	642
10	9	2	5	1616
11	10	2	4	1233
12	11	2	8	1937
13	12	2	9	3068
14	13	3	5	1652
15	14	3	2	1028
16	15	3	1	997
17	16	3	6	1930
18	17	3	1	1035
19	18	3	3	820
20	19	3	9	3158
21	20	4	9	3047
22	21	4	4	1263
23	22	4	2	494
24	23	4	9	2784
25	24	4	4	1518

Articles

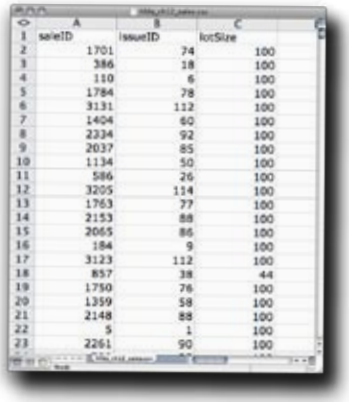
	A	B
1	authorID	author
2	1	Jason Wightman
3	2	Brewster Dombkowski
4	3	Paul Semenec
5	4	Matt Janney
6	5	Jon Radermacher
7	6	Niko Christian
8	7	Nicole Fry
9	8	Destiny Adams
10	9	Rafaela Cortez
11		

Authors

Draw arrows between the tables and describe how each relates to the other.

Sharpen your pencil Solution

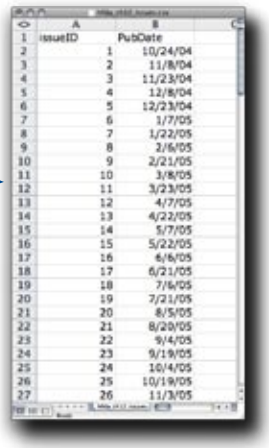
What relationships did you discover among the spreadsheets that the *Dataville Dispatch* keeps?



	A	B	C
1	saleID	issueID	lotSize
2	1701	74	100
3	386	18	100
4	110	6	100
5	1784	78	100
6	3131	112	100
7	1404	60	100
8	2334	92	100
9	2037	85	100
10	1134	50	100
11	586	26	100
12	3205	114	100
13	1763	77	100
14	2153	88	100
15	2065	86	100
16	184	9	100
17	3123	112	100
18	857	38	44
19	1750	76	100
20	1359	58	100
21	2148	88	100
22	5	1	100
23	2261	90	100

Sales

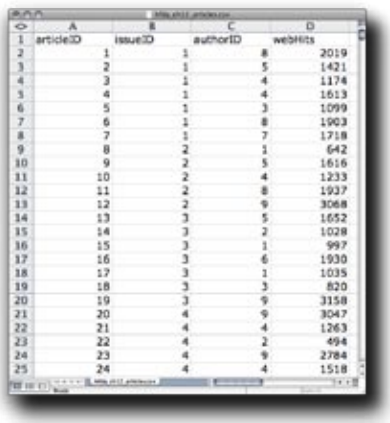
Each sale refers to a bundle of copies (usually around 100) of one issue.



	A	B
1	issueID	PubDate
2	1	10/24/04
3	2	11/8/04
4	3	11/23/04
5	4	12/8/04
6	5	12/23/04
7	6	1/7/05
8	7	1/22/05
9	8	2/6/05
10	9	2/21/05
11	10	3/8/05
12	11	3/23/05
13	12	4/7/05
14	13	4/22/05
15	14	5/7/05
16	15	5/22/05
17	16	6/6/05
18	17	6/21/05
19	18	7/6/05
20	19	7/21/05
21	20	8/5/05
22	21	8/20/05
23	22	9/4/05
24	23	9/19/05
25	24	10/4/05
26	25	10/19/05
27	26	11/3/05

Issues

Each issue consists of a bunch of articles.



	A	B	C	D
1	articleID	issueID	authorID	webHits
2	1	1	8	2019
3	2	1	5	1421
4	3	1	4	1174
5	4	1	4	1613
6	5	1	3	1099
7	6	1	8	1903
8	7	1	7	1718
9	8	2	1	642
10	9	2	5	1616
11	10	2	4	1233
12	11	2	8	1937
13	12	2	9	3068
14	13	3	5	1652
15	14	3	2	1028
16	15	3	1	997
17	16	3	6	1930
18	17	3	1	1035
19	18	3	3	820
20	19	3	9	3158
21	20	4	9	3047
22	21	4	4	1263
23	22	4	2	494
24	23	4	9	2784
25	24	4	4	1518

Articles

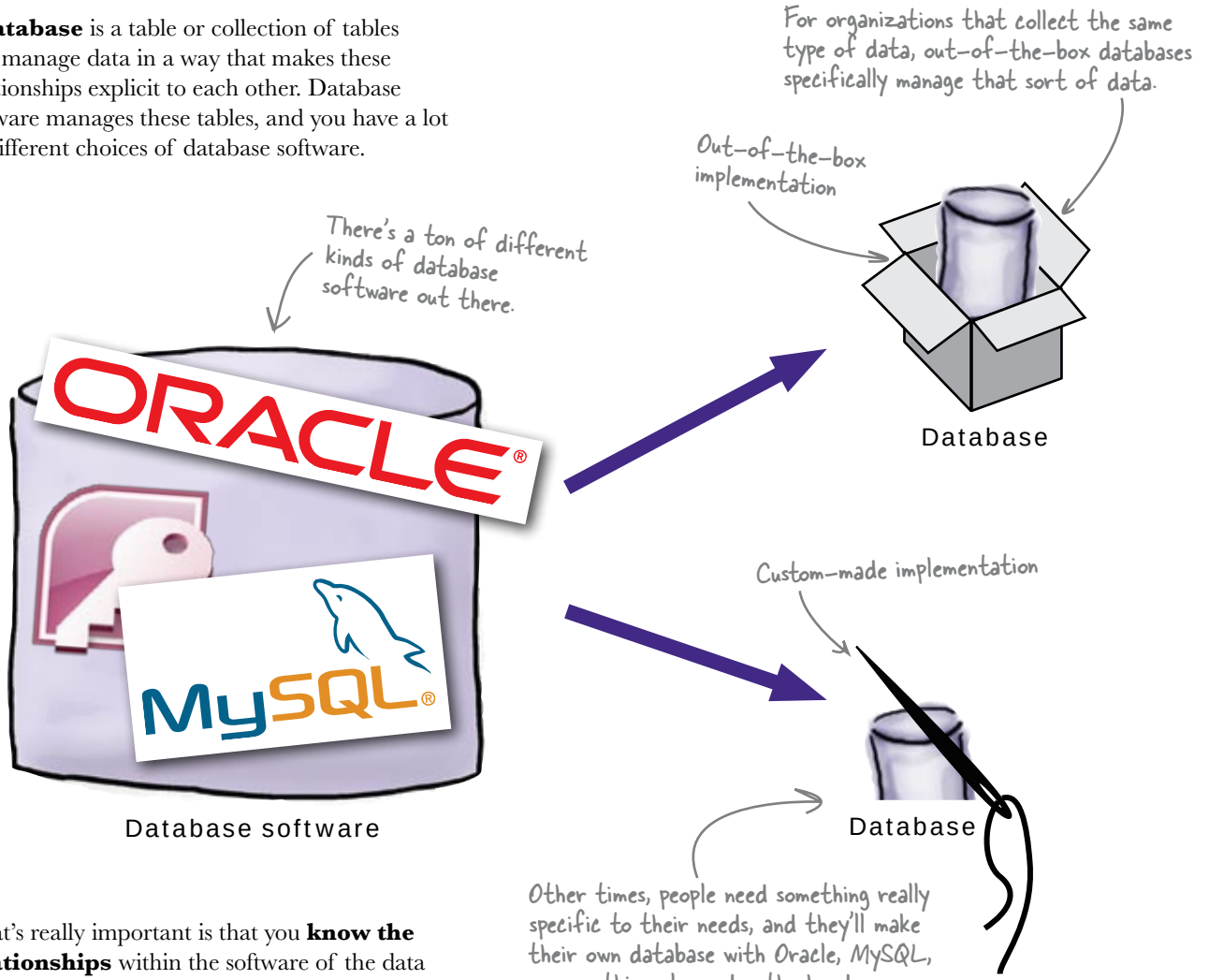
Each author writes a bunch of articles.

	A	B
1	authorID	author
2	1	Jason Wightman
3	2	Brewster Dombkowski
4	3	Paul Semenec
5	4	Matt Janney
6	5	Jon Radermacher
7	6	Niko Christian
8	7	Nicole Fry
9	8	Destiny Adams
10	9	Rafaela Cortez
11		

Authors

A database is a collection of data with well-specified relations to each other

A **database** is a table or collection of tables that manage data in a way that makes these relationships explicit to each other. Database software manages these tables, and you have a lot of different choices of database software.



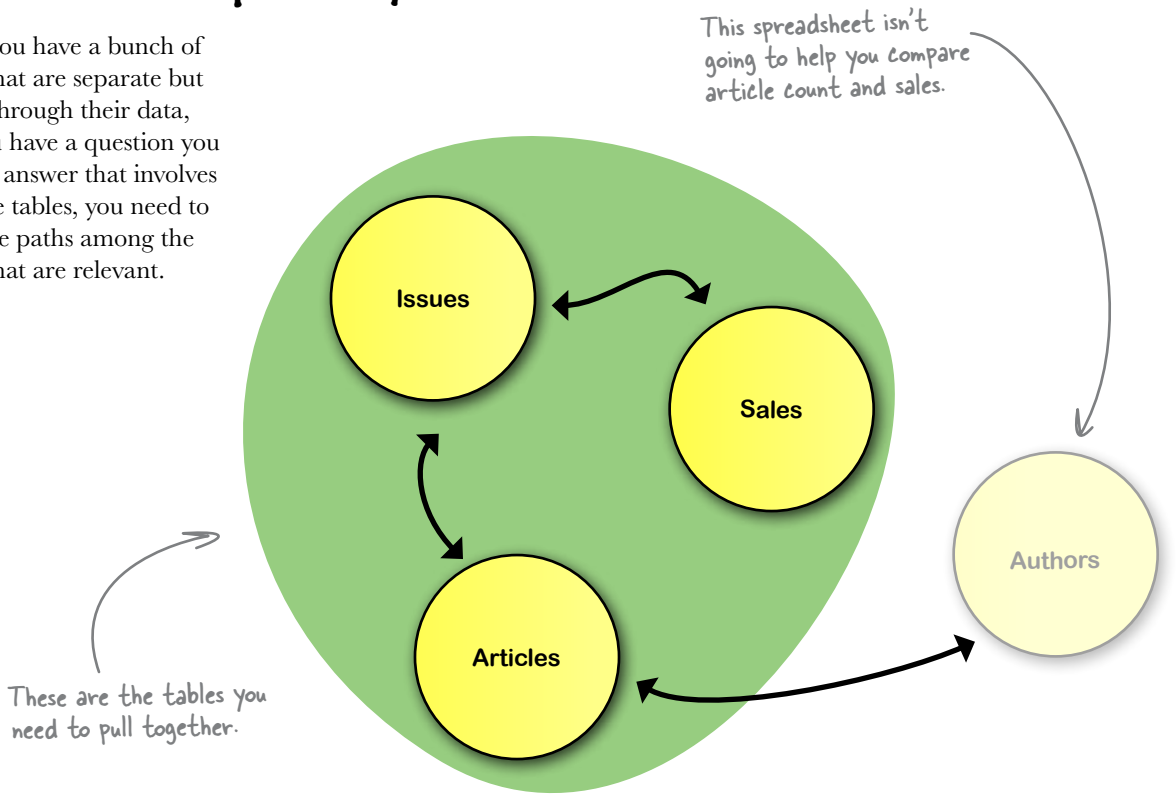
What's really important is that you **know the relationships** within the software of the data you want to record.

Here's the big question.

So how do you use this knowledge to calculate article count and sales total for each issue?

Trace a path through the relations to make the comparison you need

When you have a bunch of tables that are separate but linked through their data, and you have a question you want to answer that involves multiple tables, you need to trace the paths among the tables that are relevant.



Create a spreadsheet that goes across that path

Once you know which tables you need, then you can come up with a plan to tie the data together with formulas.

Here, you need a table that compares article count and sales for each issue. You'll need to write formulas to calculate those values.

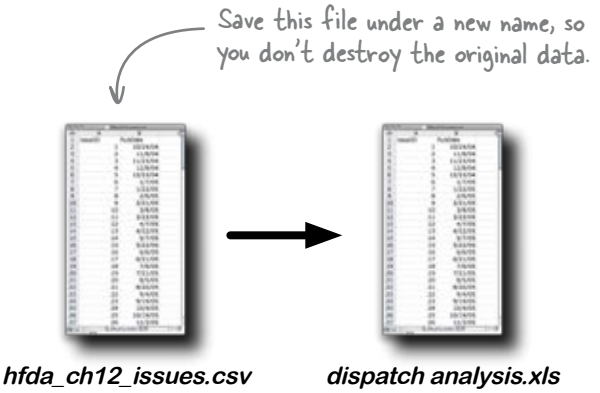
In the next exercise, you'll calculate these values.

Issue	Article count	Sales Total
1	5	1250
2	7	1800
3	8	1500
4	6	1000

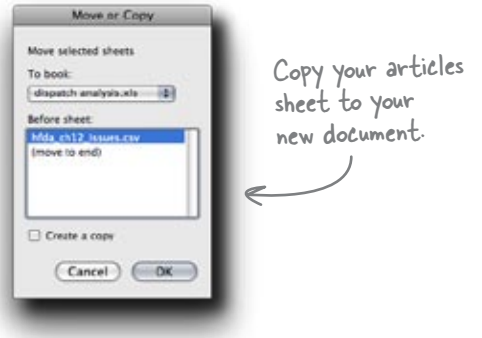
You'll need formulas for these.

Let's create a spreadsheet like the one in the facing page and start by calculating the "Article count" for each issue of the *Dispatch*.

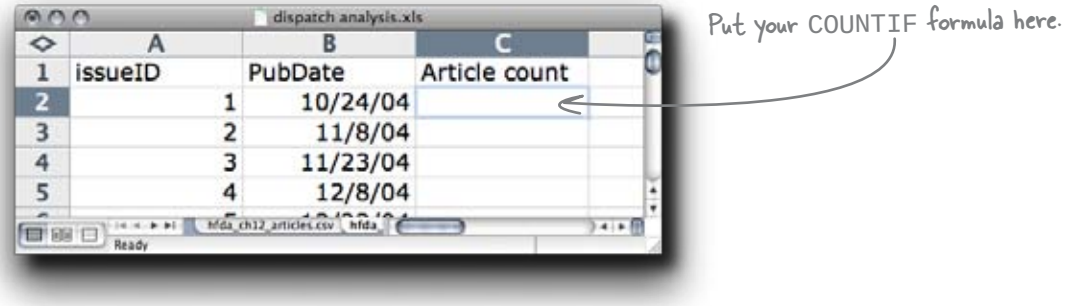
1. Open the *hfda_ch12_issues.csv* file and save a copy for your work. Remember, you don't want to mess up an original file! Call your new file "dispatch analysis.xls".



2. Open *hfda_ch12_articles.csv* and right-click an the tab that list the file name at the bottom of the sheet. Tell your spreadsheet to move the file to your *dispatch analysis.xls* document.



3. Create a column for Article count on your issue sheet. Write a `COUNTIF` formula to count the number of articles for that issue, and copy and paste that formula for each issue.



What sort of article count did you find each issue to have?

1. Open the *hfda_ch12_issues.csv* file and save a copy for your work. Remember, you don't want to mess up an original file! Call your new file "dispatch analysis.xls".

2. Open *hfda_ch12_articles.csv* and right-click an the tab that list the file name at the bottom of the sheet. Tell your spreadsheet to move the file to your *dispatch analysis.xls* document.

3. Create a column for Article count on your issue sheet. Write a `COUNTIF` formula to count the number of articles for that issue, and copy and paste that formula for each issue.

The formula looks at the "articles" tab in your spreadsheet.

```
=COUNTIF(hfda_ch12_articles.csv!B:B,hfda_ch12_issues.csv!A2)
```

It counts the number of times each issue shows up in the list of articles.

dispatch analysis.xls

	A	B	C	D
1	issueID	PubDate	Article count	
2	1	10/24/04	7	
3	2	11/8/04	5	
4	3	11/23/04	7	
5	4	12/8/04	7	
6	5	12/23/04	8	
7	6	1/7/05	7	
8	7	1/22/05	7	
9	8	2/6/05	7	
10	9	2/21/05	6	
11	10	3/8/05	5	
12	11	3/23/05	9	
13	12	4/7/05	7	
14	13	4/22/05	6	
15	14	5/7/05	6	
16	15	5/22/05	6	
17	16	6/6/05	7	
18	17	6/21/05	10	
19	18	7/6/05	7	
20	19	7/21/05	6	

	A	B	
	articleID	issueID	author
	1	1	
	2	1	
	3	1	
	4	1	
	5	1	
	6	1	
	7	1	
	8	2	
	9	2	
	10	2	
	11	2	
	12	2	
	13	3	
	14	3	
	15	3	
	16	3	
	17	3	
	18	3	
	19	3	

This is the "articles" tab in your dispatch analysis spreadsheet.

Cool! When you add the sales figures to your spreadsheet, keep in mind that the numbers just refer to units of the magazine, not dollars. I really just need you to measure sales in terms of the number of magazines sold, not in dollar terms.

Here's the Dispatch's managing editor.

Sounds good... let's add sales to this list!

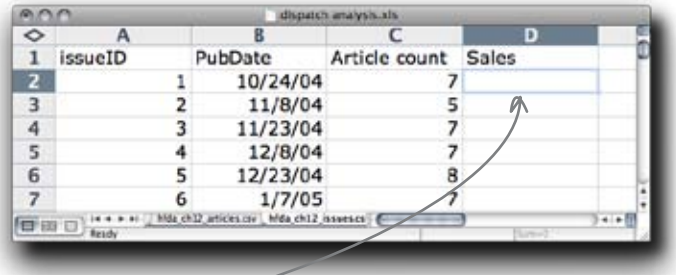
Exercise

Add a field for sales totals to the spreadsheet you are creating.

Load this!

www.headfirstlabs.com/books/hfda/hfda_ch12_sales.csv

1. Copy the *hfda_ch12_sales.csv* file as a new tab in your *dispatch analysis.xls*. Create a new column for Sales on the same sheet you used to count the articles.



Add this column and put your new formulas here.

2. Use the `SUMIF` formula to tally the sales figures for issueID #1, putting the formula in cell `C2`. Copy that formula and then paste it for each of the other issues.

Exercise Solution

What formula did you use to add sales to your spreadsheet?

This formula shows that issue #1 sold 2,227 units.

The first argument of the SUMIF formula looks at the issues.

```
=SUMIF(hfda_ch12_sales.csv!B:B, hfda_ch12_issues.csv!A2, hfda_ch12_sales.csv!C:C)
```

dispatch analysis.xls

	A	B	C	D	E
1	issueID	PubDate	Article count	Sales	
2	1	10/24/04	7	2227	
3	2	11/8/04	5	703	
4	3	11/23/04	7	2252	
5	4	12/8/04	7	2180	
6	5	12/23/04	8	2894	
7	6	1/7/05	7	2006	
8	7	1/22/05	7	2140	
9	8	2/6/05	7	2308	
10	9	2/21/05	6	1711	
11	10	3/8/05	5	1227	
12	11	3/23/05	9	3642	
13	12	4/7/05	7	2153	
14	13	4/22/05	6	1826	
15	14	5/7/05	6	1531	
16	15	5/22/05	6	1406	
17	16	6/6/05	7	2219	
18	17	6/21/05	10	4035	

The second argument looks at the specific issue whose sales you want to count.

	A	B	C	D	E
1	saleID	issueID	lotSize		
2	1701	74	100		
3	386	18	100		
4	110	6	100		
5	1784	78	100		
6	3131	112	100		
7	1404	60	100		
8	2334	92	100		
9	2037	85	100		
10	1134	50	100		
11	586	26	100		
12	3205	114	100		
13	1763	77	100		
14	2153	88	100		
15	2065	86	100		
16	184	9	100		
17	3123	112	100		
18	857	38			

The third argument points to the actual sales figures you want to sum.

Your summary ties article count and sales together

This is exactly the spreadsheet you need to tell you whether there is a relationship between the number of articles that the *Dataville Dispatch* publishes every issue and their sales.

Definitely! Let's let him have it...

Sharpen your pencil

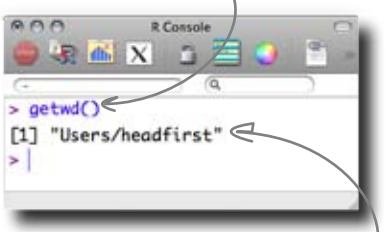
1. Open R and type the `getwd()` command to figure out where R keeps its data files. Then, in your spreadsheet, go to File > Save As... and save your data as a CSV into that directory.

   Execute this command to load your data into R:

```
dispatch <- read.csv("dispatch analysis.csv",
       header=TRUE)
```

   Name your file dispatch analysis.csv.

   This function tells you R's working directory, where it looks for files.



   Save your spreadsheet data as a CSV in R's working directory.

2. Once you have your data loaded, execute this function. Do you see an optimal value?

```
plot(Sales~jitter(Article.count),data=dispatch)
```

   You'll see how `jitter` works in a second...

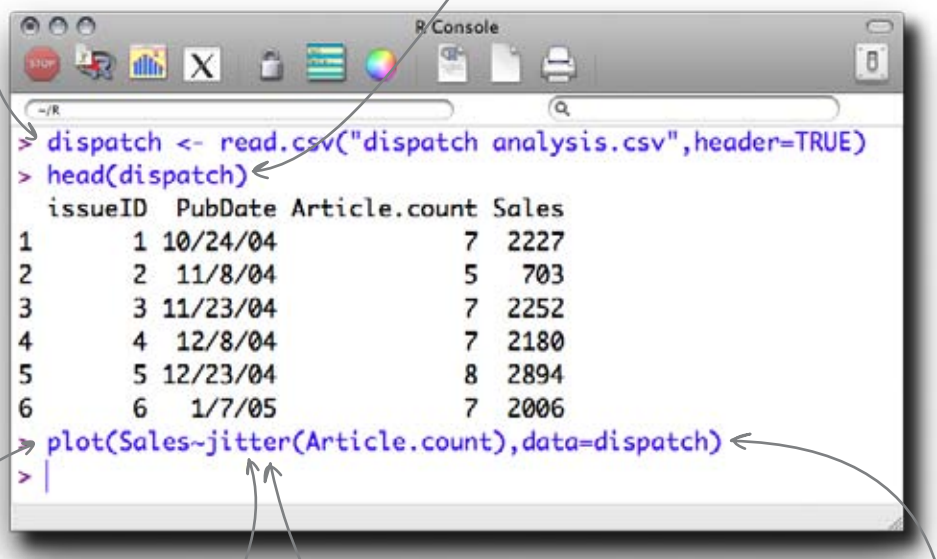
Sharpen your pencil Solution

Did you find an optimal value in the data you loaded?

The optimum appears to be around 10 articles.

Use this command to load your CSV into R.

The head command shows you what you have just loaded... it's always good to check.

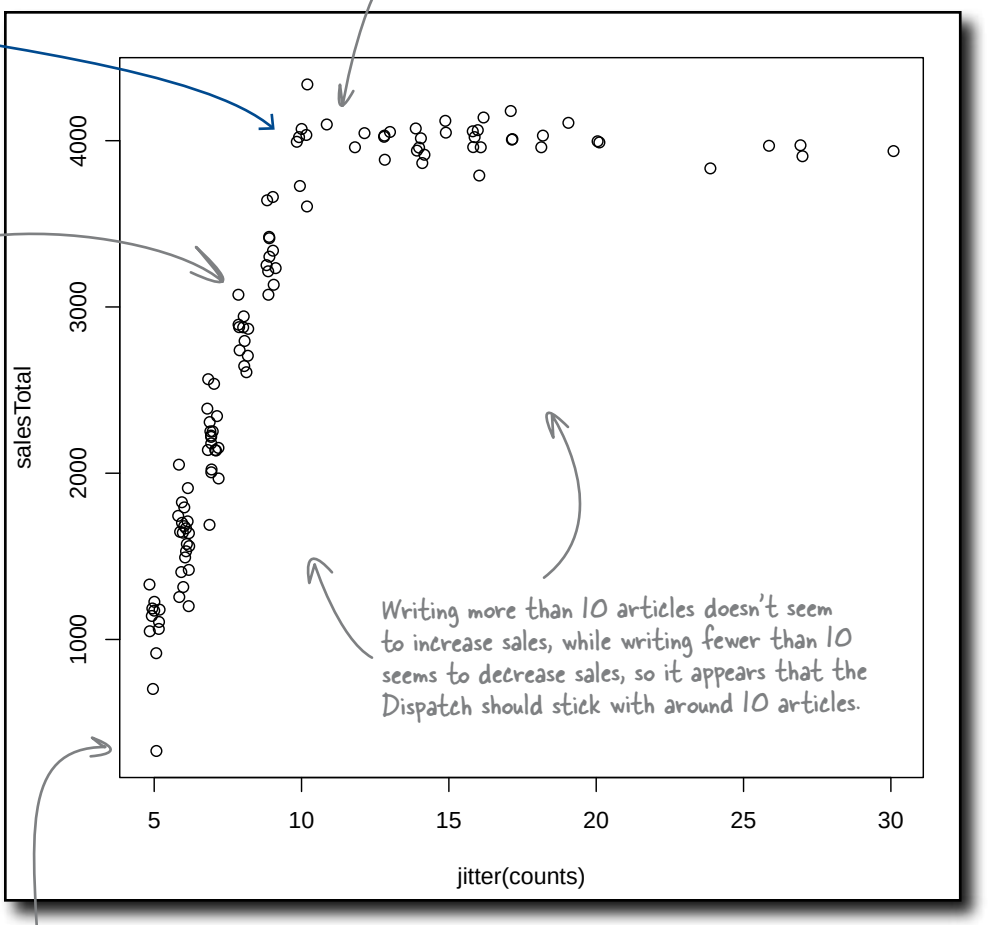


Here's the command that creates your scatterplot.

The jitter command adds a little bit of noise to your numbers, which separates them a little and makes them easier to see on the scatterplot.

Try running the same command without adding jitter; isn't the result hard to read?

Make sure that the field names in your `plot` formula match the field names that `head` shows you are contained in the data frame.



Looks like your scatterplot is going over really well

there are no Dumb Questions

Q: Do people actually store data in linked spreadsheets like that?

A: Definitely. Sometimes you'll receive extracts from larger databases, and sometimes you'll get data that people have manually kept linked together like that.

Q: Basically, as long as there are those codes that the formulas can read, linking everything with spreadsheets is tedious but not impossible.

A: Well, you're not always so lucky to recieve data from multiple tables that have neat little codes linking them together. Often, the data comes to you in a messy state, and in order to make the spreadsheets work together with formulas, you need to do some clean-up work on the data. You'll learn more about how to do that in the next chapter.

Q: Is there some better software mechanism for tying data from different tables together?

A: You'd think so, right?

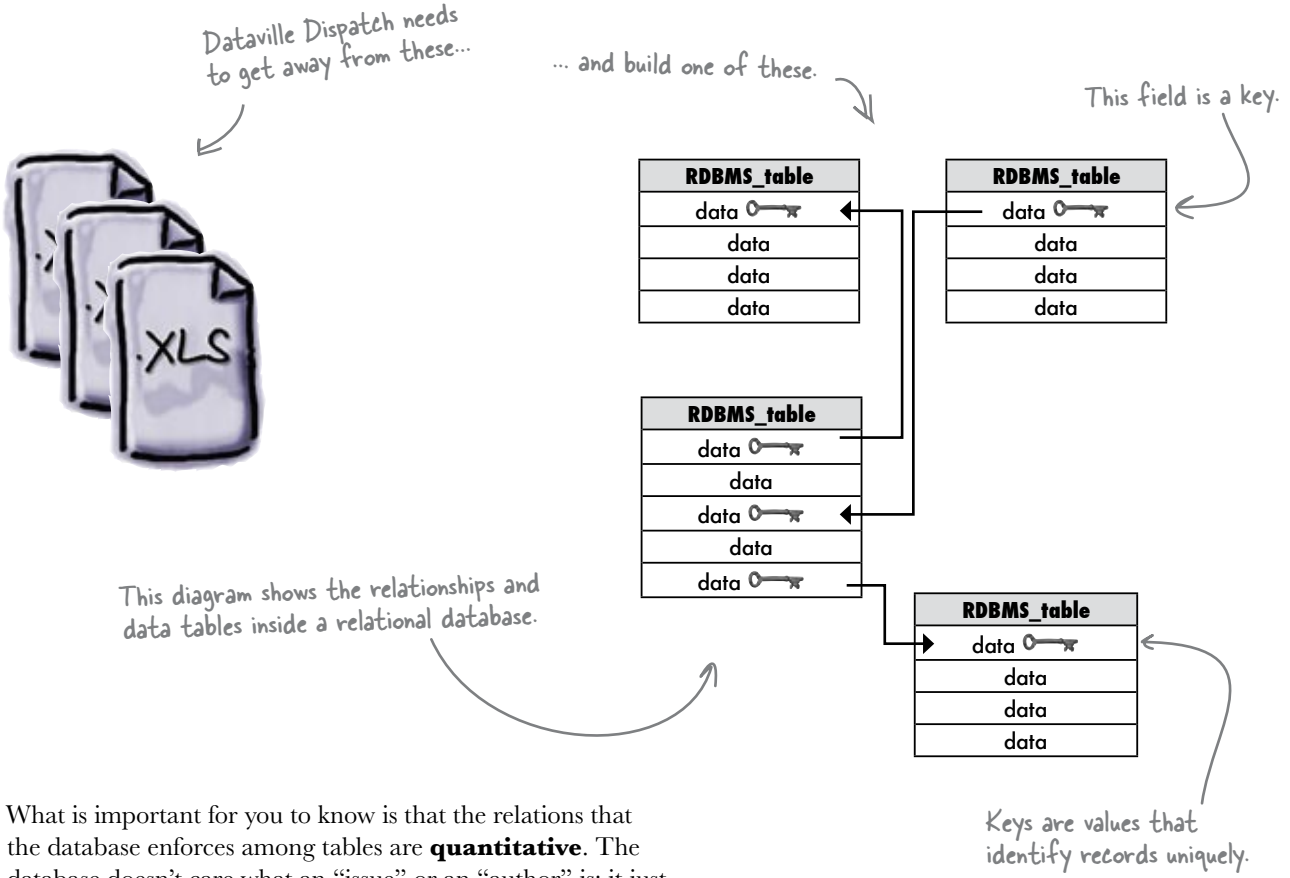
Copying and pasting all that data was a pain

It would suck to go through that process every time someone wanted to **query** (that is, to ask a question of) their data.

Besides, aren't computers supposed to be able to do all that grunt work for you?

Relational databases manage relations for you

One of the most important and powerful ways of managing data is the RDBMS or **relational database management system**. Relational databases are a huge topic, and the more you understand them, the more use you'll be able to squeeze out of any data you have stored in them.



What is important for you to know is that the relations that the database enforces among tables are **quantitative**. The database doesn't care what an "issue" or an "author" is; it just knows that one issue has multiple authors.

Each row of the RDBMS has a unique key, which you'll often see called IDs, and it it uses the keys to make sure that these quantitative relationships are never violated. Once you have a RDBMS, watch out: well-formed relational data is a treasure trove for data analysts.

If the Dataville Dispatch had a RDBMS, it would be a lot easier to come up with analyses like the one you just did.

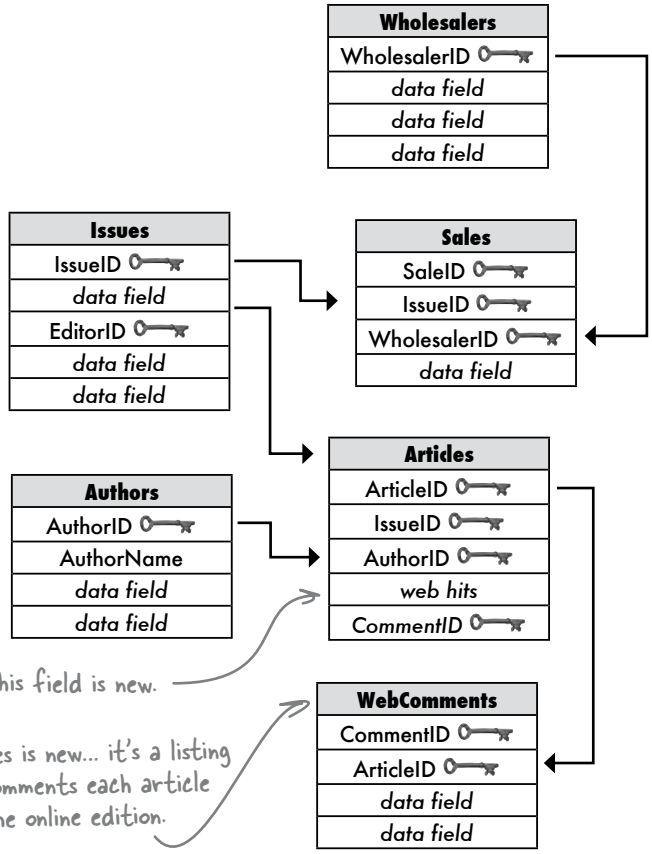
Dataville Dispatch built an RDBMS with your relationship diagram

It was about time that the *Dispatch* loaded all those spreadsheets into a real RDBMS. With the diagram you brainstormed, along with the managing editor's explanation of their data, a database architect pulled together this relational database.

Sharpen your pencil

Here is the schema for the *Dataville Dispatch*'s database. Circle the tables that you'd need to pull together into a single table in order to show which author has the articles with the most web hits and web comments.

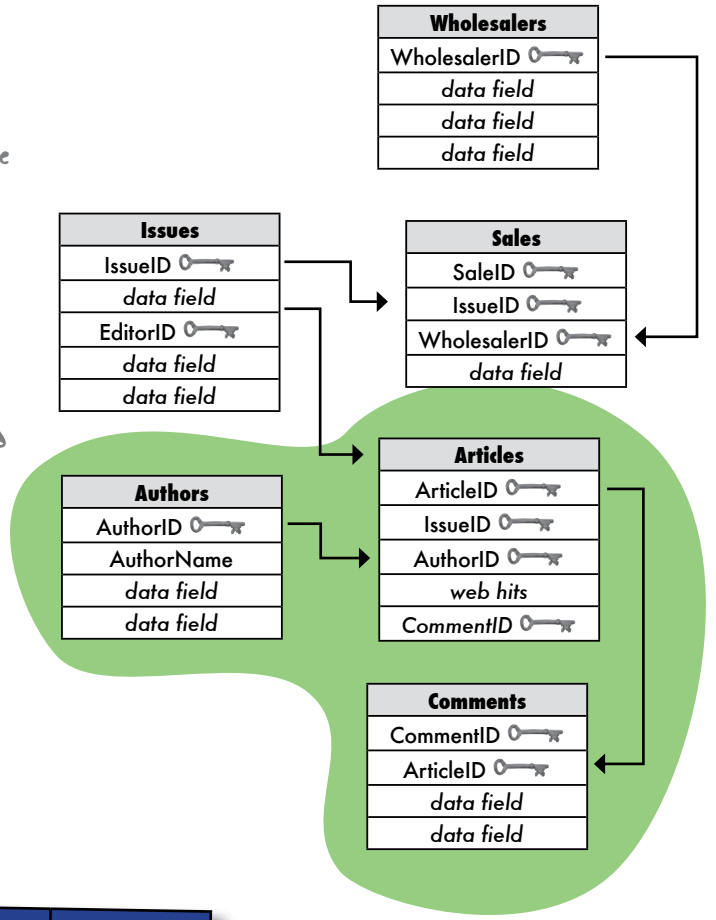
Then draw the table below that would show the fields you'd need in order create those scatterplots.



Sharpen your pencil Solution

What tables do you need to join together so that you can evaluate each author's popularity, by counting the web hits and comments that author receives?

You need a table that draws these three tables from the database together.



In the last table you used, each row represented an issue, but now each row represents an article.

Article	Author	Web Hits	Comment Count
1	Ann	2016	20
2	Ann	2016	15
3	Cathy	2164	40
4	Jeff	749	5
5	George	749	14

Ann is the author of both articles 1 and 2 for this hypothetical table.

Dataville Dispatch extracted your data using the SQL language

SQL, or *Structured Query Language*, is how data is extracted from relational databases. You can get your database to respond to your SQL questions either by tying the code directly or using a graphical interface that will create the SQL code for you.

Here's the output from the query that gets you the table you want.

Load this!

www.headfirstlabs.com/books/hfda/hfda_ch12_articleHitsComments.csv

Here's a simple SQL query.

```
SELECT AuthorName
       FROM Author WHERE
       AuthorID=1;
```

Example SQL Query

The query that created this data is much more complex than the example on the left.

This query returns the name of the author listed in the `Author` table with the `AuthorID` field equal to `1`.

You don't have to learn SQL, but it's a good idea. What's crucial is that you understand **how to ask the right questions** of the database by **understanding the tables** inside the database and the relations among them.

Exercise

1 Use the command below to load the *hfda_ch12_articleHitsComments.csv* spreadsheet into R, and then take a look at the data with the `head` command:

Make sure you're connected to the Internet for this command.

```
articleHitsComments <- read.csv(
       "http://www.headfirstlabs.com/books/hfda/
       hfda_ch12_articleHitsComments.csv",header=TRUE)
```

2 We're going to use a more powerful function to create scatterplots this time. Using these commands, load the `lattice` package and then run the `xyplot` formula to draw a "lattice" of scatterplots:

```
library(lattice)
xyplot(webHits~commentCount|authorName,data=articleHitsComments)
```

This is a new symbol!

This is the data frame that you loaded.

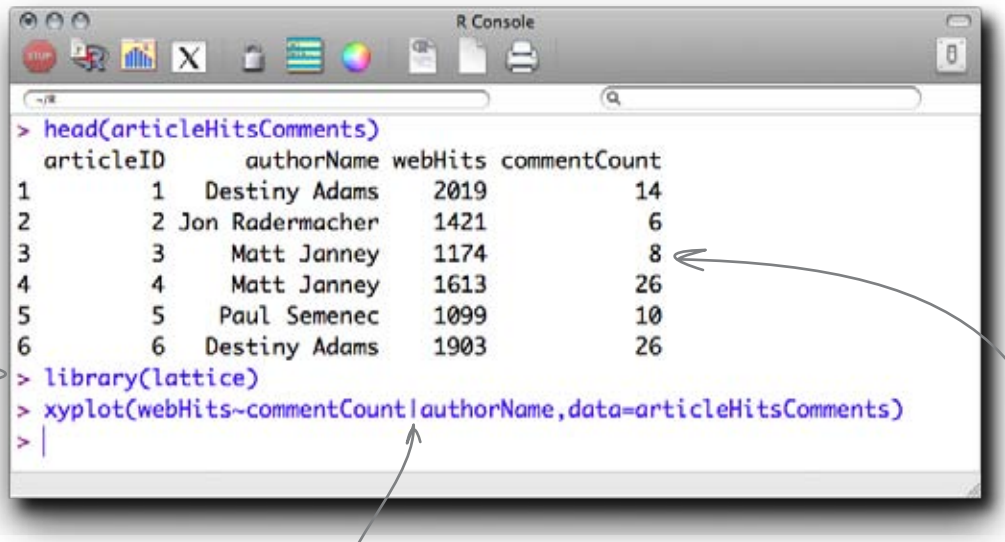
3 What author or authors perform the best, based on these metrics?

What do your scatterplots show? Do certain authors get greater sales?

1. Load the *hfda_ch12_articleHitsComments.csv* spreadsheet into R.

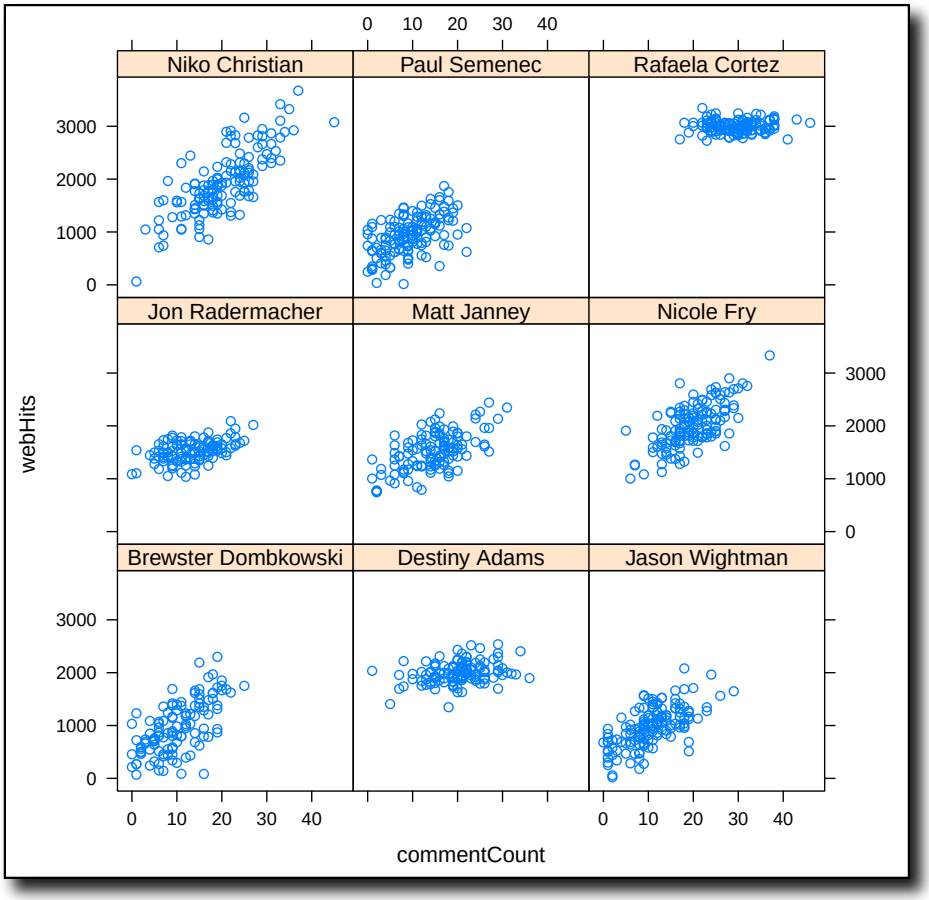
2. We're going to use a more powerful function to create scatterplots this time. Using these commands, load the `lattice` package and then run the `xyplot` formula to draw a "lattice" of scatterplots:

```
library(lattice)
xyplot(webHits~commentCount|authorName,data=articleHitsComments)
```



Article	Author	Web Hits	Comment Count
1	Ann	2016	20
2	Ann	2016	15
3	Cathy	2164	40
4	Jeff	749	5
5	George	749	14

This array of scatterplots shows web hits and comments for each article, grouped by author.



The web stats are all over the map, with authors performing very differently from each other.

3 What author or authors perform the best on these metrics?

It's pretty clear that Rafaela Cortez performs the best. All her articles have 3,000 or more web hits, and host of them show more than 20 comments. People seem really to like her. As for the rest of the authors, some (like Destiny and Nicole) tend to do better than the rest. Nike has a pretty big spread in his performance, while Brewster and Jason tend not to be too popular.

Here's what the managing editor has to say about your most recent analysis.

From:Dataville Dispatch

Subject: About our data

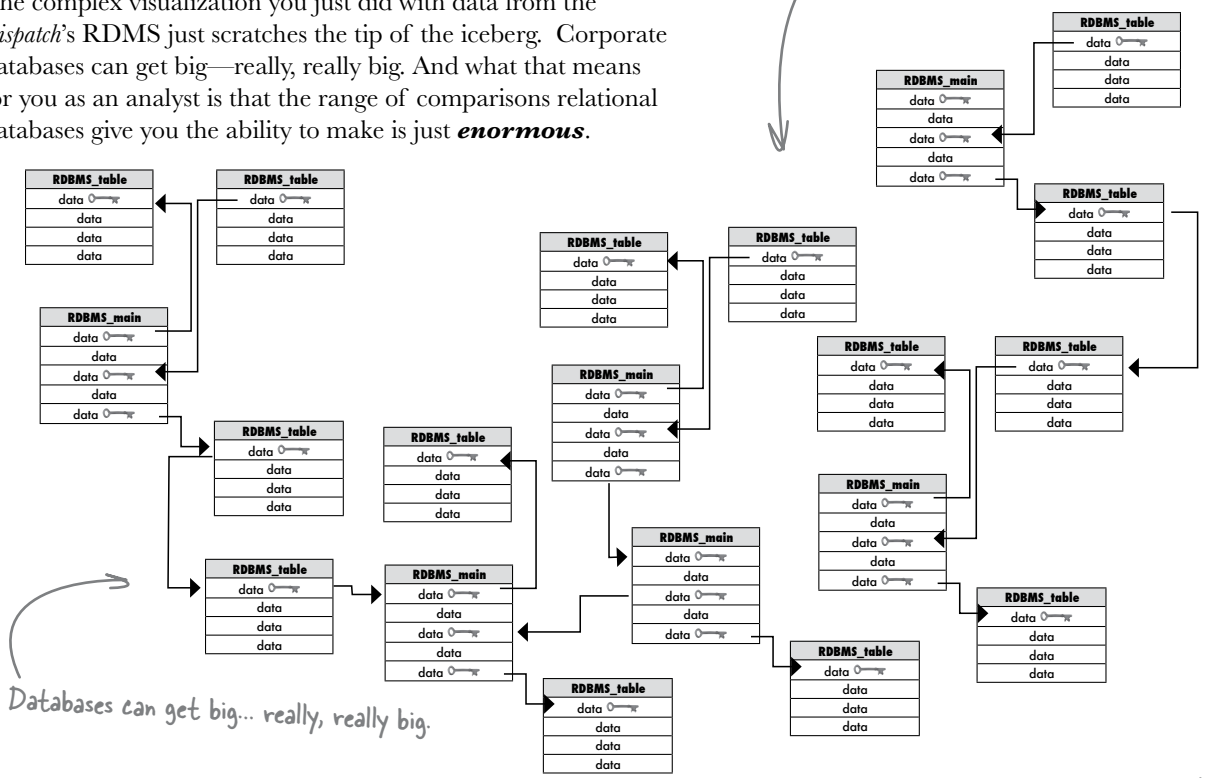
Wow, that really surprised me. I'd always suspected that Rafaela and Destiny were our star writers, but this shows that they're way ahead of everyone. Big promotion for them! All this information will make us a much leaner publication while enabling us to better reward our authors' performance. Thank you.

— DD

Comparison possibilities are endless if your data is in a RDBMS

The complex visualization you just did with data from the *Dispatch*'s RDMS just scratches the tip of the iceberg. Corporate databases can get big—really, really big. And what that means for you as an analyst is that the range of comparisons relational databases give you the ability to make is just ***enormous***.

Think about how far you can reach across this sea of tables to make a brilliant comparison!



Databases can get big... really, really big.

If you can envision it, a RDBMS can tie data together for powerful comparisons. Relational databases are a dream come true for analysts

The Dataville Dispatch's database structure isn't anywhere near this complex, but databases easily get this large.

You're on the cover

The authors and editors of the *Dataville Dispatch* was so impressed by your work that they decided to feature you in their big data issue! Nice work job. Guess who wrote the big story?

13 cleaning data

Impose order

Your data is useless…

…if it has messy structure. And a lot of people who ***collect*** data do a crummy job of maintaining a neat structure. If your data's not neat, you can't slice it or dice it, run formulas on it, or even really *see* it. You might as well just ignore it completely, right? Actually, you can do better. With a **clear vision** of how you need it to look and a few **text manipulation tools**, you can take the funkiest, craziest mess of data and **whip** it into something useful.

Just got a client list from a defunct competitor

Your newest client, Head First Head Hunters, just received a **list of job seekers** from a defunct competitor. They had to spend big bucks to get it, but it's hugely valuable. The people on this list are the best of the best, the most employable people around.

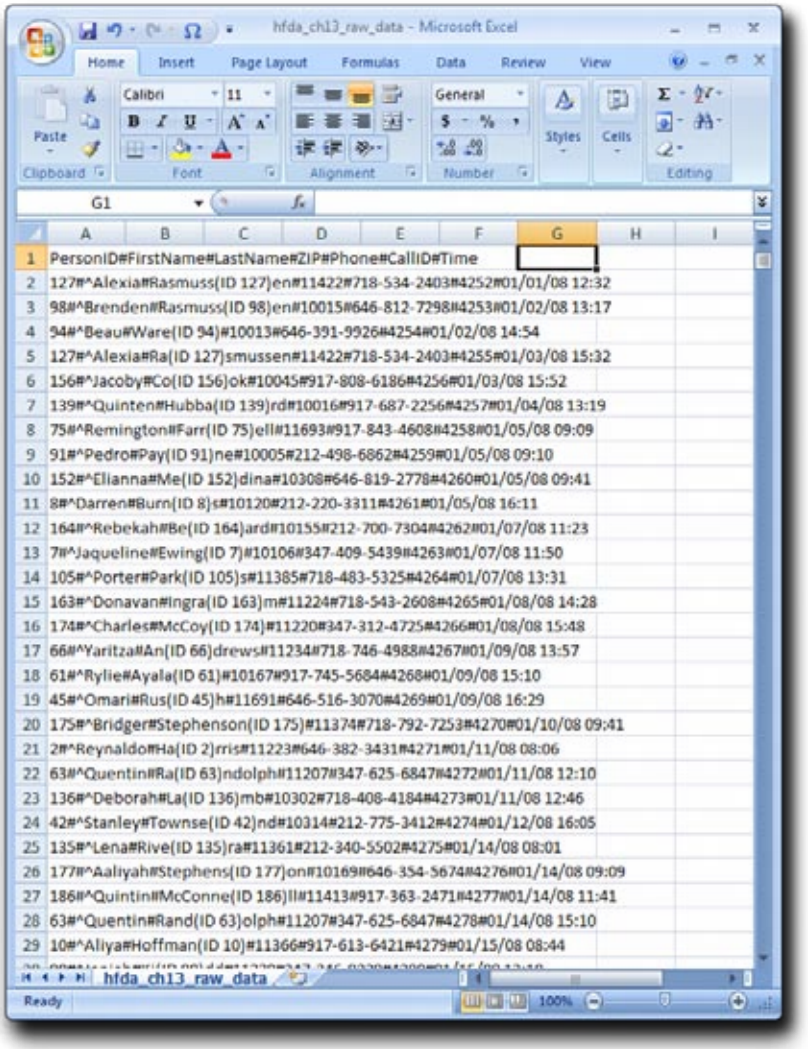
Load this!

www.headfirstlabs.com/books/hfda/hfda_ch13_raw_data.csv

This list could be a gold mine…

Look at all this stuff!

What are you going to do with this data?



…too bad the data is a **mess**! In its current form, there's not much they can do with this data. That's why they called you. Can you help?

The dirty secret of data analysis

The dirty secret of data analysis is that as analyst you might spend more time *cleaning* data than *analyzing* it. Data often doesn't arrive perfectly organized, so you'll have to do some heavy text manipulation to get it into a useful format for analysis.

Sharpen your pencil

What will be your ***first step*** for dealing with this messy data? Take a look at each of these possibilities and write the pros and cons of each.

1. Start retyping it.

   .....

   .....

2. Ask the client what she wants to *do* with the data once it's cleaned.

   .....

   .....

3. Write a formula to whip the data into shape.

   .....

   .....

Sharpen your pencil Solution

Which of these options did you choose as your first step?

1. Start retyping it.

   This sucks. It'll take forever, and there's a good chance I'll transcribe it incorrectly, messing up the data. If this is the only way to fix the data, we'd better be sure before going this route.

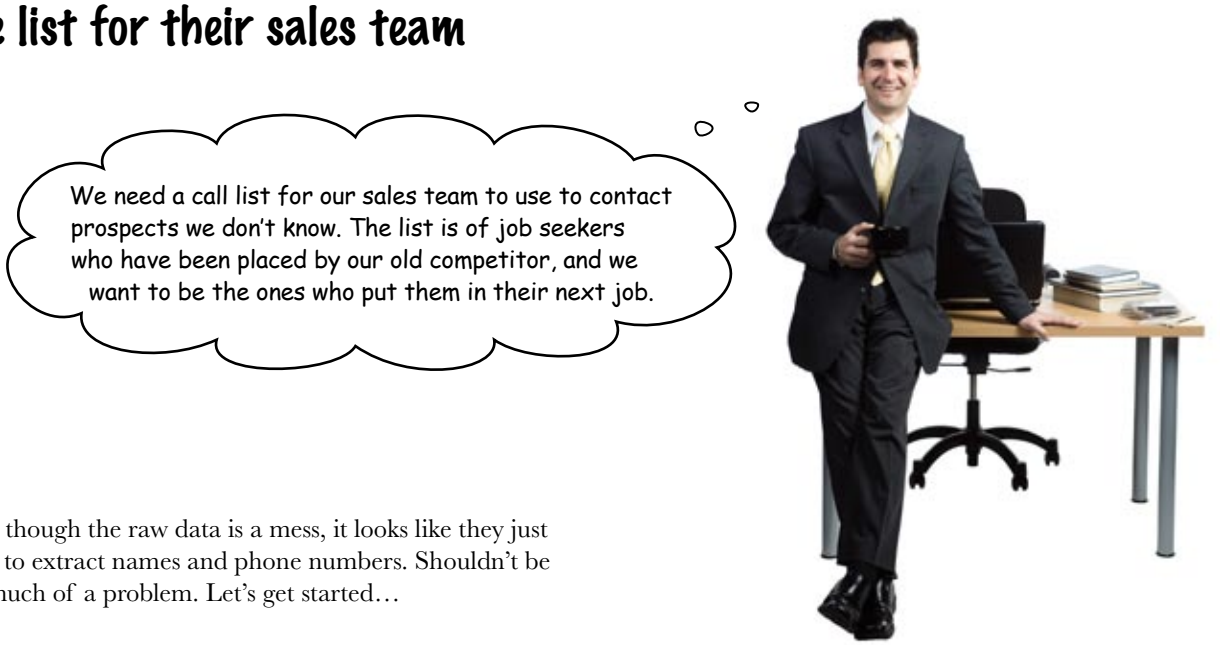
2. Ask the client what she wants to *do* with the data once it's cleaned.

   This is the way to go. With an idea of what the client wants to do with the data, I can make sure that whatever I do puts the data in exactly the form that they need.

3. Write a formula to whip the data into shape.

   A powerful formula or two would definitely help out, once we have an idea of what the data needs to look like from the client. But let's talk to the client first.

Head First Head Hunters wants the list for their sales team



Even though the raw data is a mess, it looks like they just want to extract names and phone numbers. Shouldn't be too much of a problem. Let's get started…

Sharpen your pencil

The data looks like a list of names, which is what we'd expect from the client's description of it. What you need is a clean layout of those names.

Draw a picture that shows some columns and sample data for what you *want* the messy data to look like.

Looks like these are field headings up top.

	A	B	C	D	E	F	G	H	I
1	PersonID#FirstName#LastName#ZIP#Phone#CallID#Time								
2	127#^Alexia#Rasmuss(ID 127)en#11422#718-534-2403#4252#01/01/08 12:32								
3	98#^Brenden#Rasmuss(ID 98)en#10015#646-812-7298#4253#01/02/08 13:17								
4	94#^Beau#Ware(ID 94)#10013#646-391-9926#4254#01/02/08 14:54								
5	127#^Alexia#Ra(ID 127)smussen#11422#718-534-2403#4255#01/03/08 15:32								
6	156#^Jacoby#Co(ID 156)ok#10045#917-808-6186#4256#01/03/08 15:52								
7	139#^Quinten#Hubba(ID 139)rd#10016#917-687-2256#4257#01/04/08 13:19								
8	75#^Remington#Farr(ID 75)ell#11693#917-843-4608#4258#01/05/08 09:09								
9	91#^Pedro#Pay(ID 91)ne#10005#212-498-6862#4259#01/05/08 09:10								
10	152#^Elianna#Me(ID 152)dina#10308#646-819-2778#4260#01/05/08 09:41								
11	8#^Darren#Burn(ID 8)s#10120#212-220-3311#4261#01/05/08 16:11								
12	164#^Rebekah#Be(ID 164)ard#10155#212-700-7304#4262#01/07/08 11:23								
13	7#^Jaqueline#Ewing(ID 7)#10106#347-409-5439#4263#01/07/08 11:50								
14	105#^Porter#Park(ID 105)s#11385#718-483-5325#4264#01/07/08 13:31								
15	163#^Donavan#Ingra(ID 163)m#11224#718-543-2608#4265#01/08/08 14:28								
16	174#^Charles#McCoy(ID 174)#11220#347-312-4725#4266#01/08/08 15:48								
17	66#^Yaritza#An(ID 66)drews#11234#718-746-4988#4267#01/09/08 13:57								
18	61#^Rylie#Ayala(ID 61)#10167#917-745-5684#4268#01/09/08 15:10								
19	45#^Omari#Rus(ID 45)h#11691#646-516-3070#4269#01/09/08 16:29								

Hmmm... the `Time` and `CallID` fields don't really seem relevant..

Put the column headings up here.

Draw your ideal data layout here.

Add a few lines of sample data to show what you'd like the records to look like.

Sharpen your pencil Solution

How would you like your data to look once you've cleaned it up?

You can see the information you want that's been all mashed together in Column A...

...what you need is for that information to be split into columns.

	A	B	C	D	E	F	G	H	I
1	PersonID#FirstName#LastName#ZIP#Phone#CallID#Time								
2	127#^Alexia#Rasmuss(ID 127)en#11422#718-534-2403#4252#01/01/08 12:32								
3	98#^Brenden#Rasmuss(ID 98)en#10015#646-812-7298#4253#01/02/08 13:17								
4	94#^Beau#Ware(ID 94)#10013#646-391-9926#4254#01/02/08 14:54								
5	127#^Alexia#Ra(ID 127)smussen#11422#718-534-2403#4255#01/03/08 15:32								
6	156#^Jacoby#Co(ID 156)ok#10045#917-808-6186#4256#01/03/08 15:52								
7	139#^Quinten#Hubba(ID 139)rd#10016#917-687-2256#4257#01/04/08 13:19								
8	75#^Remington#Farr(ID 75)ell#11693#917-843-4608#4258#01/05/08 09:09								
9	91#^Pedro#Pay(ID 91)ne#10005#212-498-6862#4259#01/05/08 09:10								
10	152#^Elianna#Me(ID 152)dina#10308#646-819-2778#4260#01/05/08 09:41								
11	8#^Darren#Burn(ID 8)s#10120#212-220-3311#4261#01/05/08 16:11								
12	164#^Rebekah#Be(ID 164)ard#10155#212-700-7304#4262#01/07/08 11:23								
13	7#^Jaqueline#Ewing(ID 7)#10106#347-409-5439#4263#01/07/08 11:50								
14	105#^Porter#Park(ID 105)s#11385#718-483-5325#4264#01/07/08 13:31								
15	163#^Donavan#Ingra(ID 163)m#11224#718-543-2608#4265#01/08/08 14:28								
16	174#^Charles#McCoy(ID 174)#11220#347-312-4725#4266#01/08/08 15:48								
17	66#^Yaritza#An(ID 66)drews#11234#718-746-4988#4267#01/09/08 13:57								
18	61#^Rylie#Ayala(ID 61)#10167#917-745-5684#4268#01/09/08 15:10								
19	45#^Omari#Rus(ID 45)h#11691#646-516-3070#4269#01/09/08 16:29								

When everything's separate, you can sort the data by field, filter it, or pipe it to a mail merge or web page or whatever else.

Gotta have the phone numbers... that's the most important thing for the sales team!

PersonID	FirstName	LastName	Phone
127	Alexia	Rasmussen	718-534-2403
98	Brenden	Rasmussen	646-812-7298
[Etc...]	[Etc...]	[Etc...]	[Etc...]

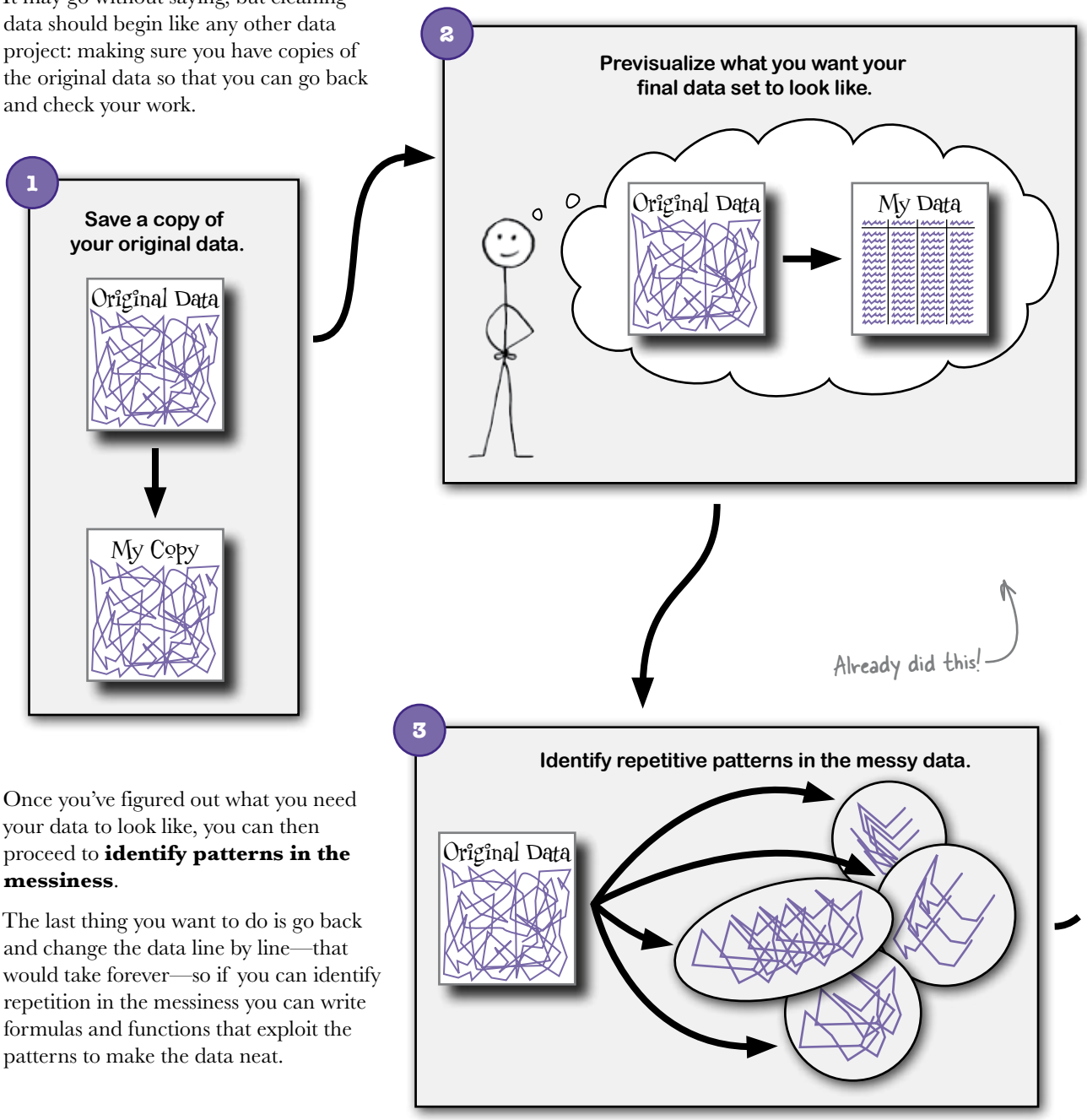
This ID field is useful, since it will let you make sure that the records are unique.

You need the name and phone fields separated from each other.

It's true: thinking about what neat data looks like won't actually make it neat. But we needed to previsualize a solution before getting down into the messy data. Let's take a look at our **general strategy** for fixing messy data and then **start coding it**…

Cleaning messy data is all about preparation

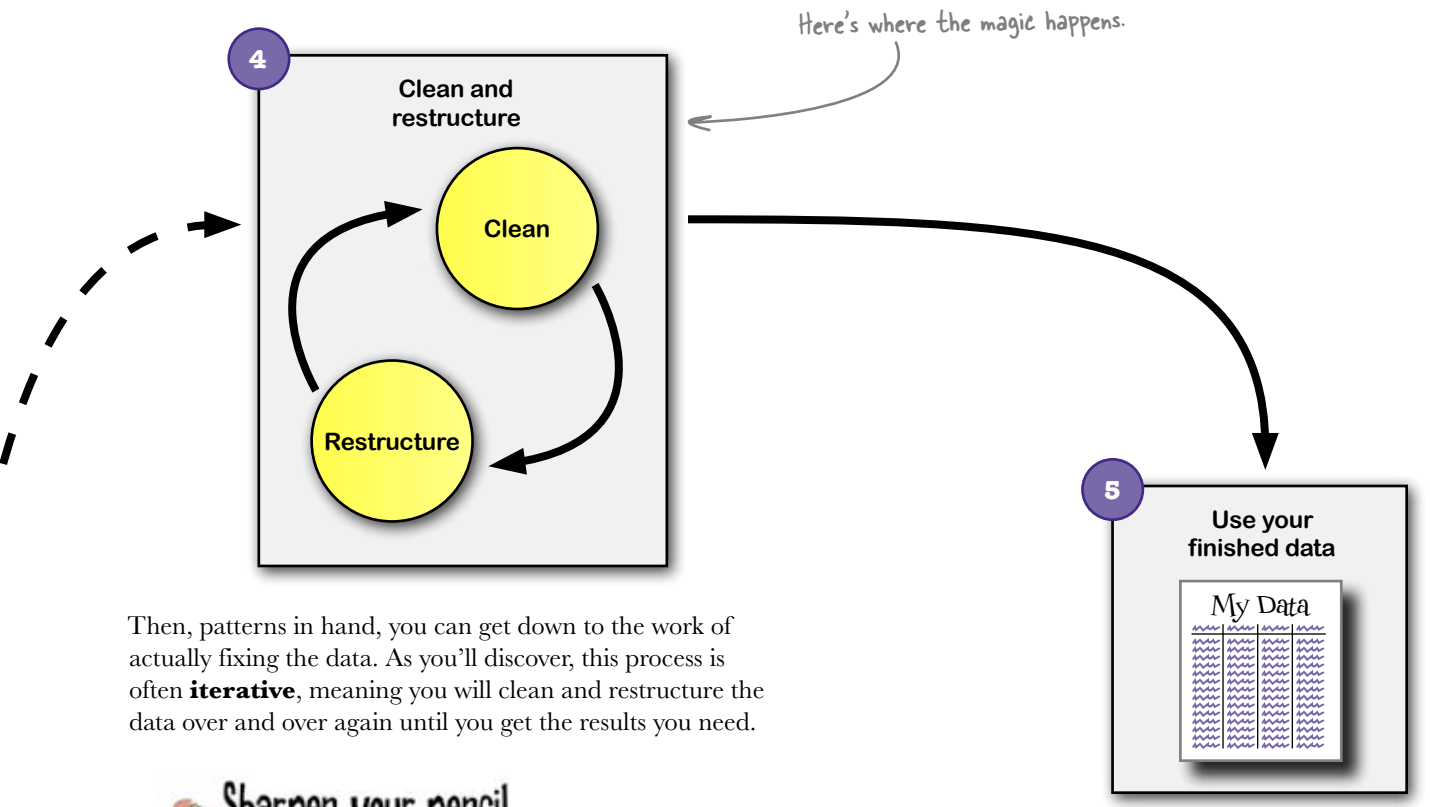
It may go without saying, but cleaning data should begin like any other data project: making sure you have copies of the original data so that you can go back and check your work.



Once you've figured out what you need your data to look like, you can then proceed to **identify patterns in the messiness**.

The last thing you want to do is go back and change the data line by line—that would take forever—so if you can identify repetition in the messiness you can write formulas and functions that exploit the patterns to make the data neat.

Once you're organized, you can fix the data itself



Then, patterns in hand, you can get down to the work of actually fixing the data. As you'll discover, this process is often **iterative**, meaning you will clean and restructure the data over and over again until you get the results you need.

Sharpen your pencil

First, let's split up the fields. Is there a pattern to how the fields are separated from each other?

.....

.....

	A	B	C	D	E	F	G	H	I
1	PersonID#FirstName#LastName#ZIP#Phone#CallID#Time								
2	127#^Alexia#Rasmuss(ID 127)en#11422#718-534-2403#4252#01/01/08 12:32								
3	98#^Brenden#Rasmuss(ID 98)en#10015#646-812-7298#4253#01/02/08 13:17								
4	94#^Beau#Ware(ID 94)#10013#646-391-9926#4254#01/02/08 14:54								
5	127#^Alexia#Ra(ID 127)smussen#11422#718-534-2403#4255#01/03/08 15:32								
6	156#^Jacoby#Co(ID 156)ok#10045#917-808-6186#4256#01/03/08 15:52								
7	139#^Quinten#Hubba(ID 139)rd#10016#917-687-2256#4257#01/04/08 13:19								
8	75#^Remington#Farr(ID 75)ell#11693#917-843-4608#4258#01/05/08 09:09								
9	91#^Pedro#Pay(ID 91)ne#10005#212-498-6862#4259#01/05/08 09:10								

Sharpen your pencil Solution

What patterns did you find in the data?

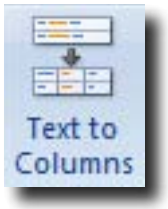
Definitely! All the data is in Column A with the fields mashed together. Between each field there is a single character: the pound (#).

	A	B	C	D	E	F	G	H	I
1	PersonID#FirstName#LastName#ZIP#Phone#CallID#Time								
2	127#^Alexia#Rasmuss(ID 127)en#11422#718-534-2403#4252#01/01/08 12:32								
3	98#^Brenden#Rasmuss(ID 98)en#10015#646-812-7298#4253#01/02/08 13:17								
4	94#^Beau#Ware(ID 94)#10013#646-391-9926#4254#01/02/08 14:54								
5	127#^Alexia#Ra(ID 127)smussen#11422#718-534-2403#4255#01/03/08 15:32								
6	156#^Jacoby#Co(ID 156)ok#10045#917-808-6186#4256#01/03/08 15:52								
7	139#^Quinten#Hubba(ID 139)rd#10016#917-687-2256#4257#01/04/08 13:19								
8	75#^Remington#Farr(ID 75)ell#11693#917-843-4608#4258#01/05/08 09:09								
9	91#^Pedro#Pay(ID 91)ne#10005#212-498-6862#4259#01/05/08 09:10								

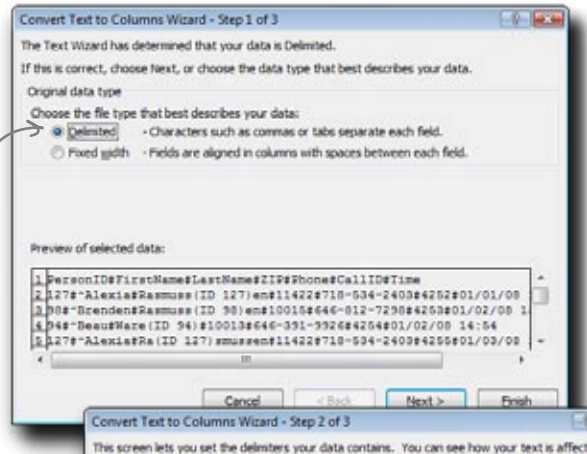
Use the # sign as a delimiter

Excel has a handy tool for splitting data into columns when the fields are separated by a **delimiter** (the technical term for a character that makes the space between fields). Select Column A in your data and press the Text to Columns button under the Data tab…

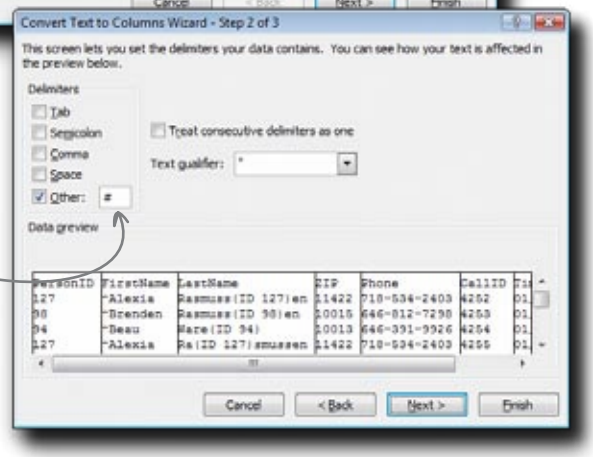
Select Column A and click this button.



Tell Excel you have a delimiter.



Specify the delimiter.



… and now you've started the Wizard. In the first step, tell Excel that your data is split up by a delimiter. In the second step, tell Excel that your delimiter is the # character. What happens when you click **Finish**?

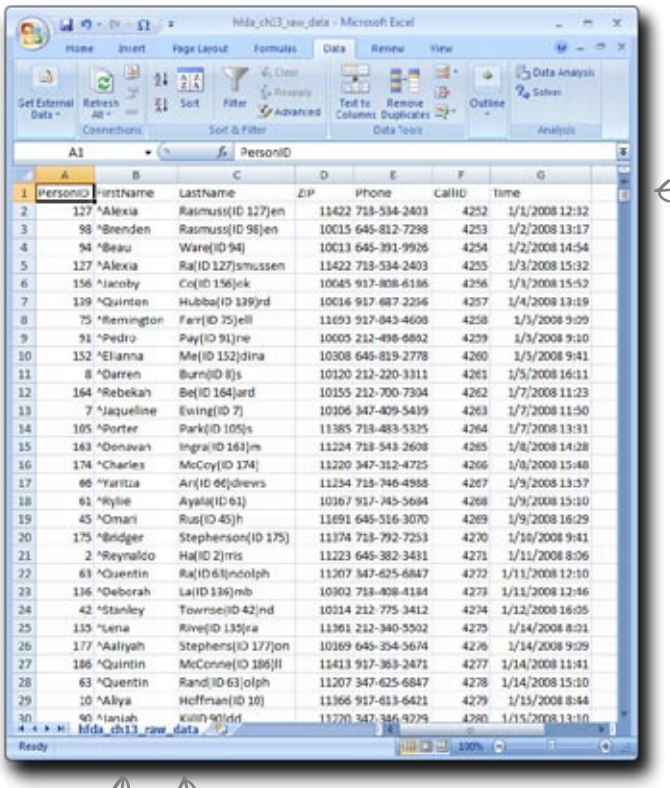
Excel split your data into columns using the delimiter

And it was no big deal. Using Excel's Convert Text to Column Wizard is a great thing to do if you have simple delimiters separating your fields.

But the data still has a few problems. The first and last names, for example, both appear to have junk characters inside the fields. You'll have to come up with a way to get rid of them!

The data is now neatly separated into columns.

Now that the pieces of data are separated, you can manipulate them individually if you want to.



PersonID	FirstName	LastName	ZIP	Phone	CallID	time
127	^Alexia	Rasmussi(ID 127)en	11422	718-534-2403	4252	1/1/2008 12:32
98	^Brenden	Rasmussi(ID 98)en	10015	646-812-7298	4253	1/2/2008 13:17
94	^Beau	Ware(ID 94)	10013	646-391-9926	4254	1/2/2008 14:54
127	^Alexia	Ra(ID 127)smussen	11422	718-534-2403	4255	1/3/2008 15:32
156	^Jacoby	Co(ID 156)ek	10045	917-808-6186	4256	1/3/2008 15:52
139	^Quinton	Hubba(ID 139)rd	10016	917-687-2256	4257	1/4/2008 13:19
75	^Remington	Farr(ID 75)ell	11693	917-843-4608	4258	1/5/2008 9:09
91	^Pedro	Pay(ID 91)ne	10003	212-498-6862	4259	1/5/2008 9:10
152	^Elianna	Me(ID 152)dina	10308	646-819-2778	4260	1/5/2008 9:41
8	^Darren	Burn(ID 8)s	10120	212-220-3311	4261	1/5/2008 16:11
164	^Rebekah	Be(ID 164)ard	10155	212-700-7304	4262	1/7/2008 11:23
7	^Jaqueline	Ewing(ID 7)	10106	347-409-5419	4263	1/7/2008 11:50
105	^Porter	Park(ID 105)s	11385	718-483-5325	4264	1/7/2008 13:31
163	^Donavan	Ingra(ID 163)m	11224	718-543-2608	4265	1/8/2008 14:28
174	^Charles	McCoy(ID 174)	11220	347-312-4725	4266	1/8/2008 15:48
66	^Yuritza	Ar(ID 66)drews	11234	718-746-4588	4267	1/9/2008 13:57
61	^Rylie	Ayala(ID 61)	10167	917-745-5684	4268	1/9/2008 15:10
45	^Omari	Rus(ID 45)h	11691	646-516-3070	4269	1/9/2008 16:29
175	^Bridger	Stepherson(ID 175)	11374	718-792-7253	4270	1/10/2008 9:41
2	^Reynaldo	Ha(ID 2)rris	11223	646-382-3431	4271	1/11/2008 8:06
63	^Quentin	Ra(ID 63)ndolph	11207	347-625-6847	4272	1/11/2008 12:10
136	^Deborah	La(ID 136)mb	10302	718-408-4184	4273	1/11/2008 12:46
42	^Stanley	Townse(ID 42)nd	10314	212-775-3412	4274	1/12/2008 16:05
135	^Lena	Rive(ID 135)ra	11361	212-340-5502	4275	1/14/2008 8:01
177	^Aaliyah	Stephens(ID 177)on	10169	646-354-5674	4276	1/14/2008 9:09
186	^Quintin	McConne(ID 186)ll	11413	917-363-2471	4277	1/14/2008 11:41
63	^Quentin	Rand(ID 63)olph	11207	347-625-6847	4278	1/14/2008 15:10
10	^Aliya	Hoffman(ID 10)	11366	917-613-6421	4279	1/15/2008 8:44

What are you going to do to fix the `FirstName` field?

What about the `LastName` field?

Sharpen your pencil

What's the pattern you'd use to fix the `FirstName` column?

.....

.....

Sharpen your pencil Solution

Is there a pattern to the messiness in the `FirstName` field?

At the beginning of every name there is a ^ character. We need to get rid of all of them in order to have neat first names.

```
^FirstName
```

This character is just in our way.

Here's the useful stuff.

B
FirstName
^Alexia
^Brenden
^Beau
^Alexia
^Jacoby
^Quinten
^Remington
^Pedro
^Elianna
^Darren
^Rebekah
^Jaqueline
^Porter
^Donavan

B
FirstName
Alexia
Brenden
Beau
Alexia
Jacoby
Quinten
Remington
Pedro
Elianna
Darren
Rebekah
Jaqueline
Porter
Donavan

This character is everywhere!

Let's see what Excel has for us…

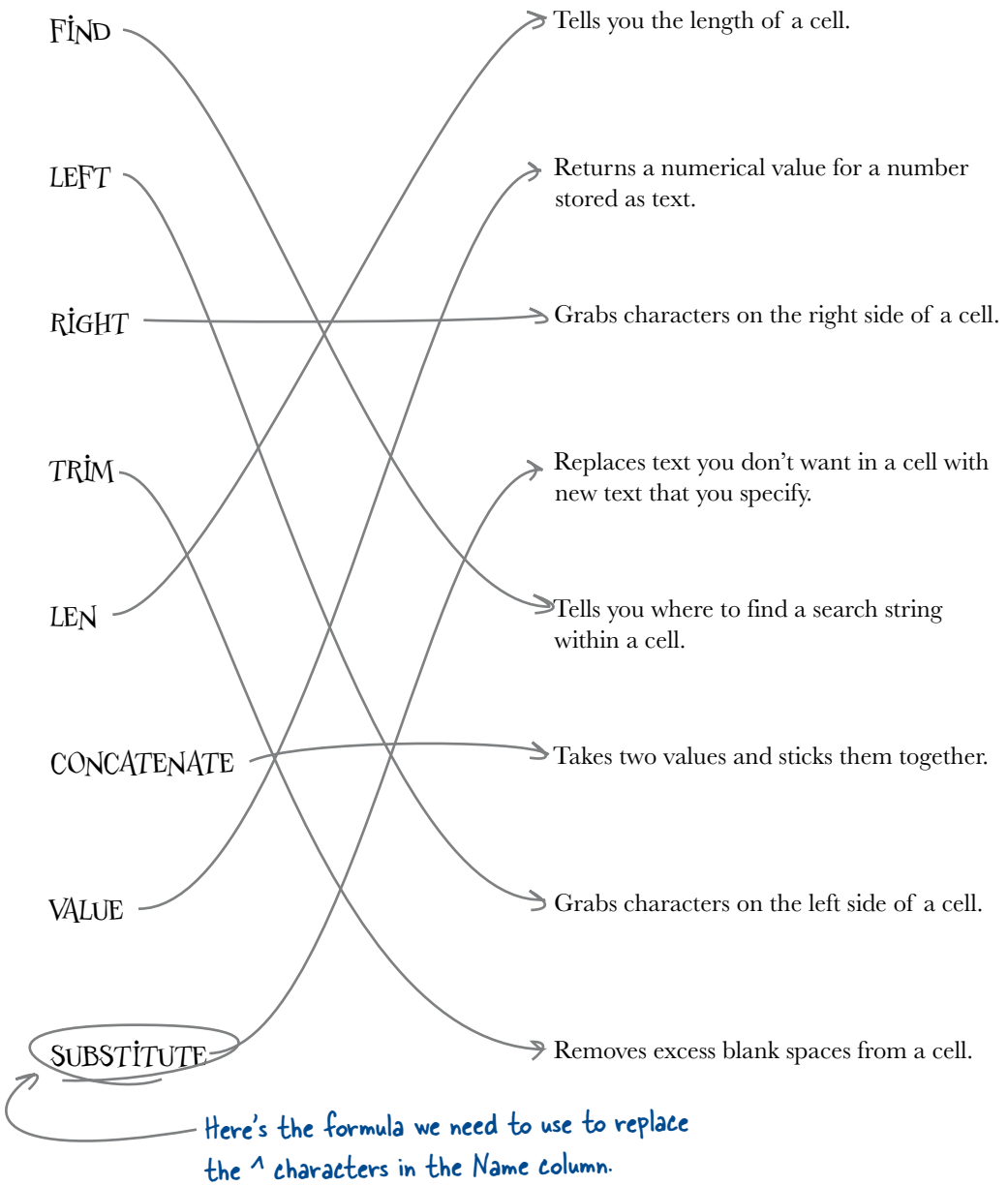
WHO DOES WHAT?

Match each Excel text formula with its function. Which function do you think you'll need to use the clean up the Name column?

FIND	Tells you the length of a cell.
LEFT	Returns a numerical value for a number stored as text.
RIGHT	Grabs characters on the right side of a cell.
TRIM	Replaces text you don't want in a cell with new text that you specify.
LEN	Tells you where to find a search string within a cell.
CONCATENATE	Takes two values and sticks them together.
VALUE	Grabs characters on the left side of a cell.
SUBSTITUTE	Removes excess blank spaces from a cell.

WHO DOES WHAT? SOLUTION

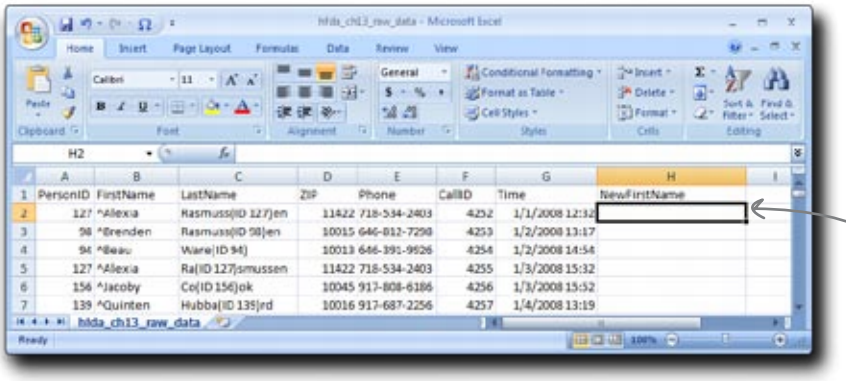
Match each Excel text formula with its function. Which function do you think you'll need to use the clean up the Name column?



Use SUBSTITUTE to replace the carat character

1. To fix the `FirstName` field, type this formula into cell H2:

```
=SUBSTITUTE(B2,"^","")
```



2. Copy this formula and paste it all the way down to the end of the data in Column H. What happens?

there are no Dumb Questions

Q: Am I limited to just these formulas? What if I want to take the characters on the left and right of a cell and stick them together? It doesn't look there's a formula that does just that.

A: There isn't, but if you nest the text functions inside of each other you can achieve much more complicated text manipulations. For example, if you wanted to take the first and last characters inside of cell `A1` and stick them together, you'd use this formula:

```
CONCATENATE(LEFT(A1,1),
    RIGHT(A1,1))
```

Q: So I can nest a whole bunch of text formulas inside of each other?

A: You can, and it's a powerful way to manipulate text. There's a problem, though: if your data is really messy and you have to nest a whole bunch of formulas inside of each other, your entire formula can be almost impossible to read.

Q: Who cares? As long as it works, I'm not going to be reading it anyway.

A: Well, the more complex your formula, the more likely you'll need to do subtle tweaking of it. And the less readable your formula is, the harder that tweaking will be.

Q: Then how do I get around the problem of formulas that are big and unreadable?

A: Instead of packing all your smaller formulas into one big formula, you can break apart the small formulas into different cells and have a "final" formula that puts them all together. That way, if something is a little off, it'll be easier to find the formula that needs to be tweaked.

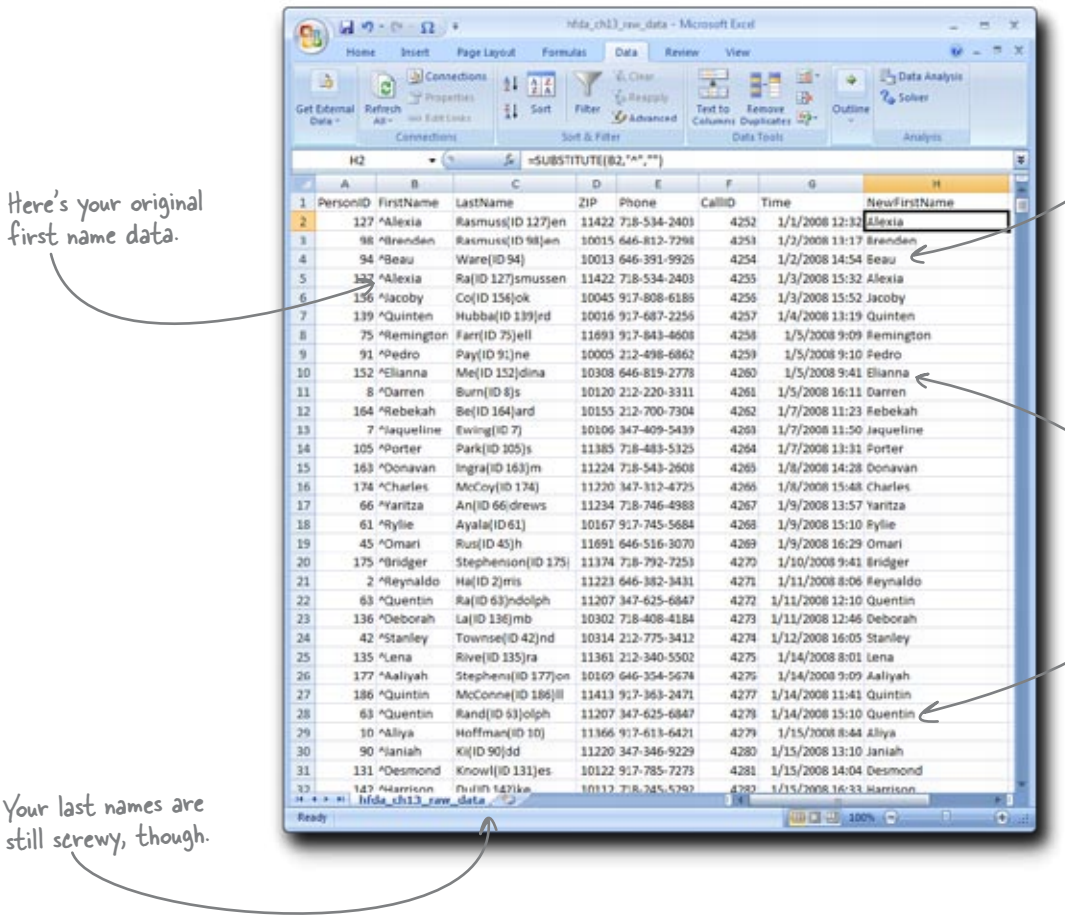
Q: You know, I bet R has much more powerful ways of doing text manipulation.

A: It does, but why bother learning them? If Excel's `SUBSTITUTE` formula handles your issue, you can save your self some time by skipping R.

You cleaned up all the first names

Using the `SUBSTITUTE` formula, you had Excel grab the ^ symbol from each first name and replace it with nothing, which you specified by two quotation marks (`""`).

Lots of different software lets you get rid of crummy characters by replacing those characters with nothing.



PersonID	FirstName	LastName	ZIP	Phone	CallID	Time	NewFirstName
127	^Alexia	Rasmuss(ID 127)en	11422	718-534-2403	4252	1/1/2008 12:32	Alexia
98	^Brenden	Rasmuss(ID 98)en	10015	646-812-7298	4253	1/2/2008 13:17	Brenden
94	^Beau	Ware(ID 94)	10013	646-391-9926	4254	1/2/2008 14:54	Beau
127	^Alexia	Ra(ID 127)smussen	11422	718-534-2403	4255	1/3/2008 15:32	Alexia
156	^Jacoby	Co(ID 156)ok	10045	917-808-6186	4256	1/3/2008 15:52	Jacoby
139	^Quinten	Hubba(ID 139)rd	10016	917-687-2256	4257	1/4/2008 13:19	Quinten
75	^Remington	Farr(ID 75)ell	11693	917-843-4608	4258	1/5/2008 9:09	Remington
91	^Pedro	Pay(ID 91)ne	10005	212-498-6862	4259	1/5/2008 9:10	Pedro
152	^Elianna	Me(ID 152)dina	10308	646-819-2778	4260	1/5/2008 9:41	Elianna
8	^Darren	Burn(ID 8)s	10120	212-220-3311	4261	1/5/2008 16:11	Darren
164	^Rebekah	Be(ID 164)ard	10155	212-700-7304	4262	1/7/2008 11:23	Rebekah
7	^Jaqueline	Ewing(ID 7)	10106	347-409-5439	4263	1/7/2008 11:50	Jaqueline
105	^Porter	Park(ID 105)s	11385	718-483-5325	4264	1/7/2008 13:31	Porter
163	^Donavan	Ingra(ID 163)m	11224	718-543-2608	4265	1/8/2008 14:28	Donavan
174	^Charles	McCoy(ID 174)	11220	347-312-4725	4266	1/8/2008 15:48	Charles
66	^Yaritza	An(ID 66)drews	11234	718-746-4988	4267	1/9/2008 13:57	Yaritza
61	^Rylie	Ayala(ID 61)	10167	917-745-5684	4268	1/9/2008 15:10	Rylie
45	^Omari	Rus(ID 45)h	11691	646-516-3070	4269	1/9/2008 16:29	Omari
175	^Bridger	Stephenson(ID 175)	11374	718-792-7253	4270	1/10/2008 9:41	Bridger
2	^Reynaldo	Ha(ID 2)rris	11223	646-382-3431	4271	1/11/2008 8:06	Reynaldo
63	^Quentin	Ra(ID 63)ndolph	11207	347-625-6847	4272	1/11/2008 12:10	Quentin
136	^Deborah	La(ID 136)mb	10302	718-408-4184	4273	1/11/2008 12:46	Deborah
42	^Stanley	Townse(ID 42)nd	10314	212-775-3412	4274	1/12/2008 16:05	Stanley
135	^Lena	Rive(ID 135)ra	11361	212-340-5502	4275	1/14/2008 8:01	Lena
177	^Aaliyah	Stephens(ID 177)on	10169	646-354-5674	4276	1/14/2008 9:09	Aaliyah
186	^Quintin	McConne(ID 186)ll	11413	917-363-2471	4277	1/14/2008 11:41	Quintin
63	^Quentin	Rand(ID 63)olph	11207	347-625-6847	4278	1/14/2008 15:10	Quentin
10	^Aliya	Hoffman(ID 10)	11366	917-613-6421	4279	1/15/2008 8:44	Aliya
90	^Janiah	Ki(ID 90)dd	11220	347-346-9229	4280	1/15/2008 13:10	Janiah
131	^Desmond	Knowl(ID 131)es	10122	917-785-7273	4281	1/15/2008 14:04	Desmond

To make the original first name data go away forever copy the H column and then Paste Special > Values to turn these values into actual text rather than formula outputs. After that you can **delete** the `FirstName` column so that you never have to see those pesky ^ symbols again.

You can delete away... as long as you saved a copy of the original file so you can refer back to it if you made a mistake.

Hmpf. That first name pattern was easy because it was just a single character at the beginning that had to be removed. But the last name is going to be harder, because it's a tougher pattern.

Exercise

Let's try using SUBSTITUTE again, this time to fix the last names.

C
LastName
Rasmuss(ID 127)en
Rasmuss(ID 98)en
Ware(ID 94)
Ra(ID 127)smussen
Co(ID 156)ok
Hubba(ID 139)rd
Farr(ID 75)ell
Pay(ID 91)ne
Me(ID 152)dina
Burn(ID 8)s
Be(ID 164)ard
Ewing(ID 7)
Park(ID 105)s
Ingra(ID 163)m

First, look for the pattern in this messiness. What would you tell SUBSTITUTE to replace? Here's the syntax again:

```
=SUBSTITUTE(your reference cell,
  the text you want to replace,
  what you want to replace it with)
```

Can you write a formula that works?

Sharpen your pencil Solution

Could you fix the `LastName` field using SUBSTITUTE?

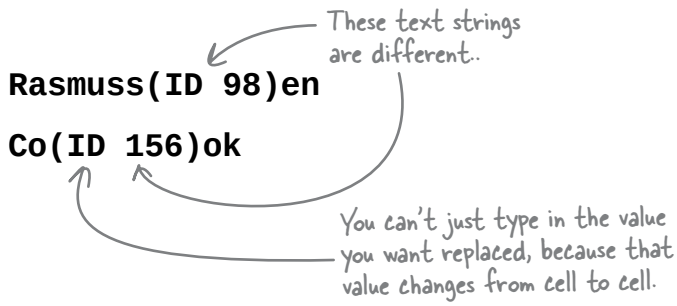
SUBSTITUTE won't work here! Every cell has different messy text. In order to make SUBSTITUTE work, you'd have to write a separate formula for each last name.

```
=SUBSTITUTE(C2, "(ID 127)", "")
=SUBSTITUTE(C3, "(ID 98)", "")
=SUBSTITUTE(C4, "(ID 94)", "")
```

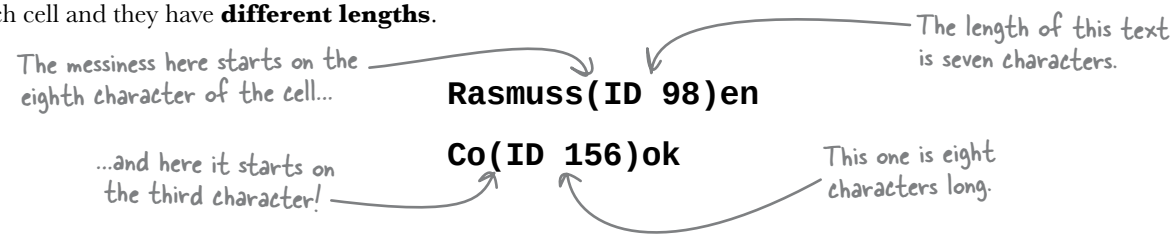
And typing a bajillion formulas like this defeats the purpose of using formulas to begin with. Formulas are supposed to *save* you the trouble of typing and retyping!

The last name pattern is too complex for SUBSTITUTE

The SUBSTITUTE function looks for a pattern in the form of a single text string to replace. The problem with the last names are that **each has a different text string** to replace.

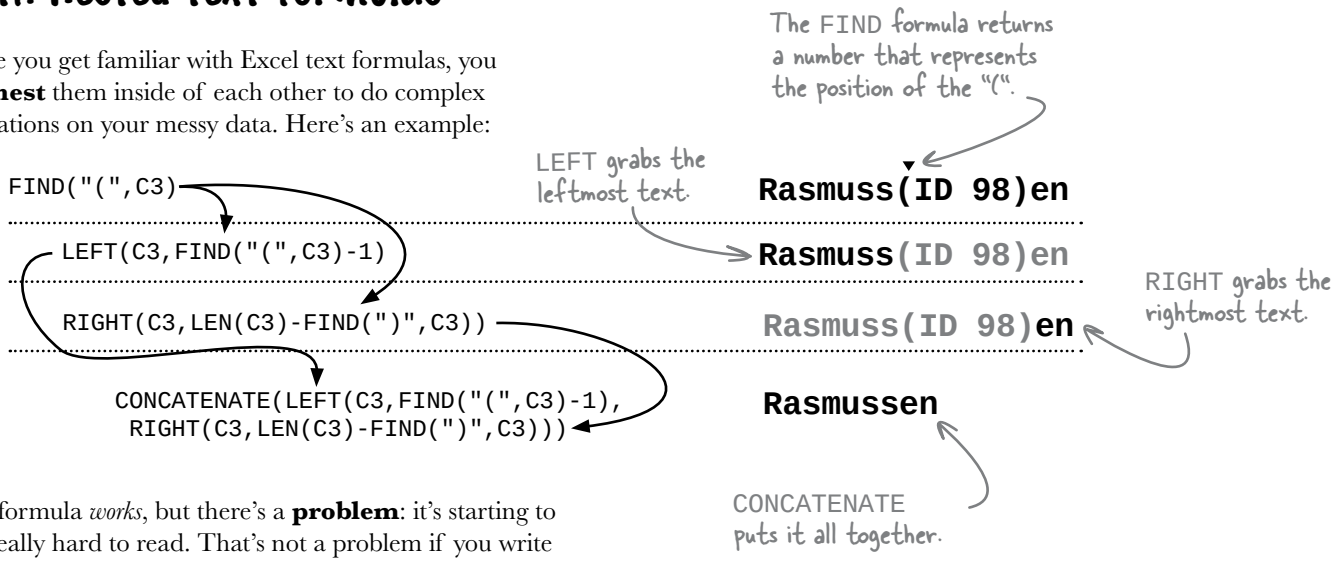


And that's not all: the pattern of messiness in the `LastName` field is more complex in that the messy strings show up in **different positions** within each cell and they have **different lengths**.



Handle complex patterns with nested text formulas

Once you get familiar with Excel text formulas, you can **nest** them inside of each other to do complex operations on your messy data. Here's an example:



The formula *works*, but there's a **problem**: it's starting to get really hard to read. That's not a problem if you write formulas perfectly the first time, but you'd be better off with a tool that has power *and* simplicity, unlike this nested CONCATENATE formula.

Wouldn't it be dreamy if there were an easier way to fix complex messes than with long, unreadable formulas like that one. But I know it's just a fantasy...

R can use regular expressions to crunch complex data patterns

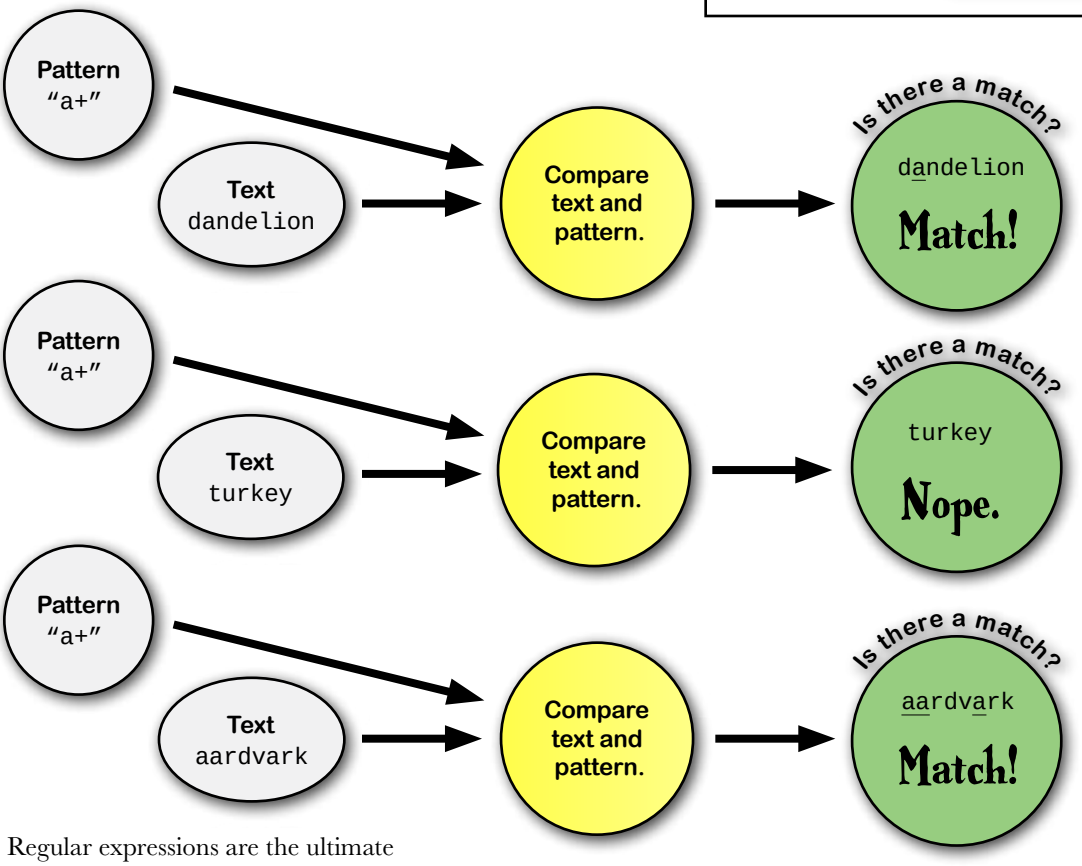
Regular expressions are a programming tool that allows you to specify complex patterns to match and replace strings of text, and R has a powerful facility for using them.

Here's a simple regular expression **pattern** that looks for the letter "a". When you give this pattern to R, it'll say whether there's a match.

Geek Bits

To learn more about the full regex specification and syntax, type `?regex` in R.

Here's R's regular expression reference in the help files.



Regular expressions are the ultimate tool for cleaning up messy data. Lots of platforms and languages implement regular expressions, even though Excel doesn't.

From: Head First Head Hunters
To: Analyst
Subject: Need those names NOW

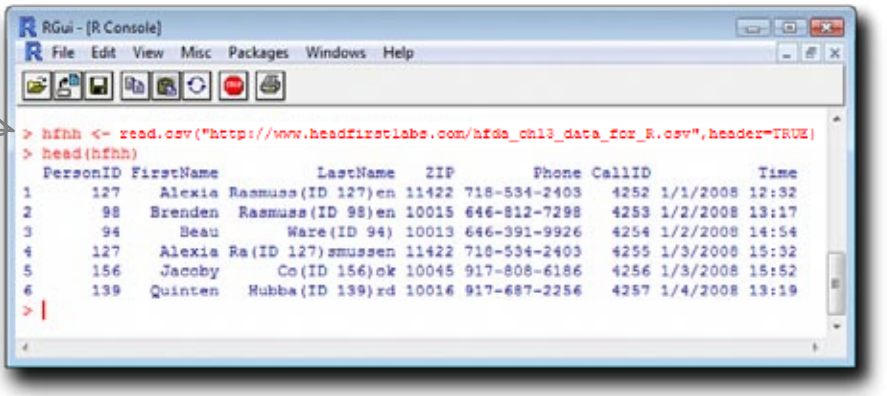
Well get on with it! Those prospects are hot and are only getting colder. I want our sales force to start calling people like yesterday!

Do this!

Better get moving! Here goes:

1. Load your data into R and take a look at what you've got with the `head` command. You can either save your Excel file as a CSV and load the CSV file into R, or you can use the web link below to get the most recent data.

This command reads the CSV into a table called `hfhh`.



2. Run this regular expressions command

```
NewLastName <- sub("\\(.*\\)","",hfhh$LastName)
```

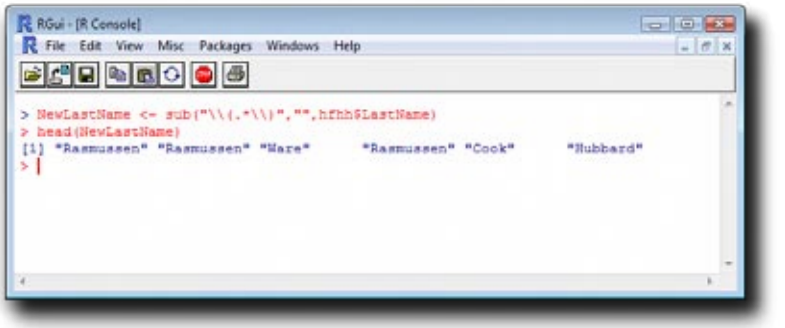
3. Then take a look at your work by running the `head` command to see the first few rows of your table.

```
head(NewLastName)
```

What happens?

The sub command fixed your last names

The `sub` command used the **pattern** you specified and replaced all instances of it with blank text, effectively deleting every parenthetical text string in the `LastName` column.



Let's take a closer look at that syntax:

Here's a new vector for your cleaned last names.

Here's your regular expression pattern.

This is blank text, which replaces text that matches the pattern with nothing.

```
NewLastName <- sub("\\(.*\\)","",hfhh$LastName)
```

If you can find a pattern in the messiness of your data, you can write a regular expression that will neatly exploit it to get you the structure you want.

No need to write some insanely long spreadsheet formula!

Your Regular Expression Up Close

Your regular expression has three parts: the left parenthesis, the right parenthesis, and everything in between.

The left parenthesis (the backslashes are "escape" characters that tell R that the parenthesis is not itself an R expression).

Everything in between.

The right parenthesis.

```
\\(.*\\)
```

The dot means "any character."

The asterisk means "any number of the preceding character."

there are no Dumb Questions

Q: Some of those regular expression commands look really hard to read. How hard is it to master regular expressions?

A: They can be hard to read because they're really concise. That economy of syntax can be a real benefit when you have crazy-complex patterns to decode. Regular expressions are easy to get the hang of but (like anything complex) hard to master. Just take your time when you read them, and you'll get the hang of them.

Q: What if data doesn't even come in a spreadsheet? I might have to extract data from a PDF, a web page, or even XML.

A: Those are the sorts of situations where regular expressions really shine. As long as you can get your information into some sort of text file, you can parse it with regular expressions. Web pages in particular are a really common source of information for data analysis, and it's a snap to program HTML tag patterns into your regex statements.

Q: What other specific platforms use regular expressions?

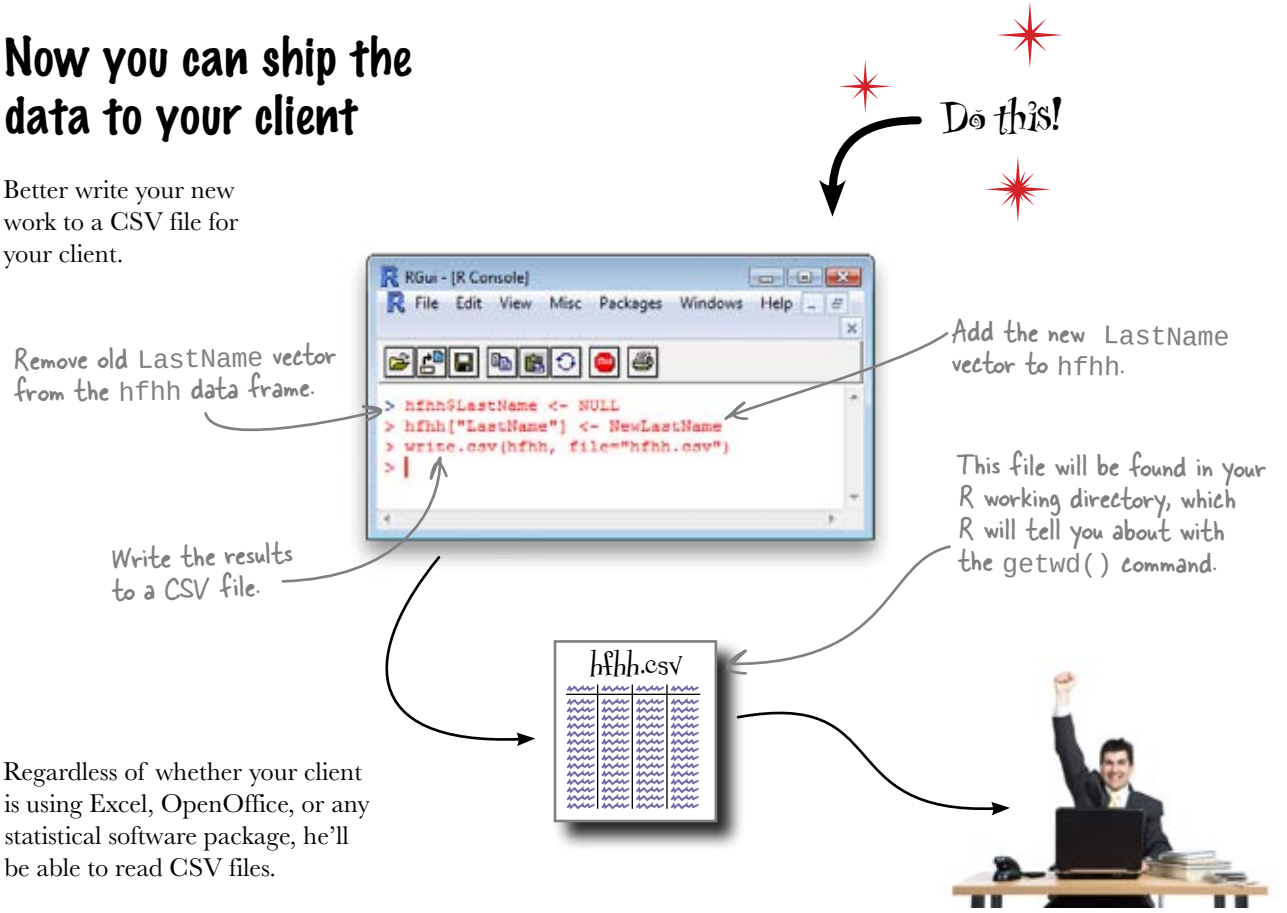
A: Java, Perl, Python, JavaScript... all sorts of different programming languages use them.

Q: If regular expressions are so common in programming languages, why can't Excel do regular expressions?

A: On the Windows platform, you can use Microsoft's Visual Basic for Applications (VBA) programming language inside of Excel to run regular expressions. But most people would sooner just use a more powerful program like R than take the trouble to learn to program Excel. Oh, and since VBA was dropped from the recent release of Excel for Mac, you can't use regex in Excel for Mac, regardless of how badly you might want to.

Now you can ship the data to your client

Better write your new work to a CSV file for your client.



Regardless of whether your client is using Excel, OpenOffice, or any statistical software package, he'll be able to read CSV files.

Maybe you're not quite done yet...

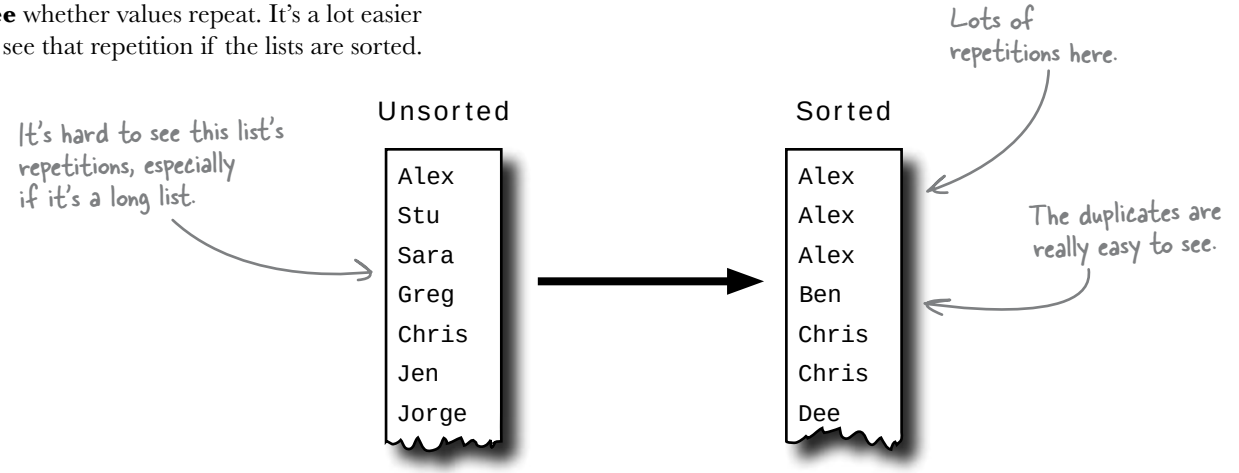
The client has a bit of a problem with your work.

He's got a point. Take "Alexia Rasmussen," for example. Alexia definitely shows up more than once. It could be that there are two separate people named Alexia Rasmussen, of course. But then again, both records here have `PersonID` equal to 127, which would suggest that they are the same person.

Maybe Alexia is **the only duplicate name** and the client is just reacting to that one mistake. To find out, you'll need to figure out how you can *see* duplicates more easily than just looking at this huge list as it is.

Sort your data to show duplicate values together

If you have a lot of data, it can be hard to **see** whether values repeat. It's a lot easier to see that repetition if the lists are sorted.



Exercise

Let's get a better look at the duplicates in your list by sorting the data.

In R, you sort a data frame by using the `order` function inside of the subset brackets. Run this command:

A new, sorted copy of your list.

```
hfhhSorted <- hfhh[order(hfhh$PersonID), ]
```

Because the `PersonID` field probably represents a unique number for each person, that makes it a good field to use it to sort. After all, it's possible that there's more than one person in the data named "John Smith."

Next, run the `head` command to see what you've created:

```
head(hfhhSorted, n=50)
```

What does R do?

Did sorting your data frame in R by `PersonID` reveal any duplicates?

```
> hfhh$PersonID <- hfhh[order(hfhh$PersonID), ]
> head(hfhh$PersonID, n=50)
    PersonID FirstName     ZIP         Phone CallID             Time    LastName
63         1    Jaylen 10006 646-376-4466   4314  1/29/2008 11:19 Christensen
935        1    Jaylen 10006 646-376-4466   4686   7/30/2008 10:27 Christensen
940        1    Jaylen 10006 646-376-4466   4791  10/8/2008 16:08 Christensen
647        1    Jaylen 10006 646-376-4466   4898  12/6/2008 15:50 Christensen
712        1    Jaylen 10006 646-376-4466   4963  1/11/2009 15:20 Christensen
831        1    Jaylen 10006 646-376-4466   5082  3/17/2009 11:13 Christensen
974        1    Jaylen 10006 646-376-4466   5225  5/22/2009 14:58 Christensen
20         2  Reynaldo 11223 646-382-3431   4271   1/11/2008 8:06      Harris
347        2  Reynaldo 11223 646-382-3431   4598   6/17/2008 8:01      Harris
421        2  Reynaldo 11223 646-382-3431   4672   7/23/2008 14:31     Harris
575        2  Reynaldo 11223 646-382-3431   4826 10/28/2008 13:58     Harris
627        2  Reynaldo 11223 646-382-3431   4878 11/26/2008 14:44     Harris
737        2  Reynaldo 11223 646-382-3431   4988   1/28/2009 8:39      Harris
892        2  Reynaldo 11223 646-382-3431   5143  4/16/2009 15:58      Harris
934        2  Reynaldo 11223 646-382-3431   5185    5/6/2009 8:36      Harris
971        2  Reynaldo 11223 646-382-3431   5222  5/19/2009 15:14     Harris
311        3    Yandel 10099 718-510-4575   4562  5/27/2008 15:13    Franklin
438        3    Yandel 10099 718-510-4575   4689   7/31/2008 9:10     Franklin
666        3    Yandel 10099 718-510-4575   4917 12/18/2008 15:36     Franklin
1078       3    Yandel 10099 718-510-4575   5325  7/14/2009 14:51     Franklin
183        4       Lia 11230 718-664-9574   4434   3/25/2008 9:13      Short
305        4       Lia 11230 718-664-9574   4556  5/24/2008 14:54      Short
528        4       Lia 11230 718-664-9574   4779   9/29/2008 9:06      Short
658        4       Lia 11230 718-664-9574   4909 12/15/2008 16:33     Short
841        4       Lia 11230 718-664-9574   5092   3/23/2009 9:54      Short
1025       4       Lia 11230 718-664-9574   5276  6/15/2009 10:54      Short
52         5      Cael 10024 646-423-7673   4303  1/23/2008 11:03     Glover
197        5      Cael 10024 646-423-7673   4448   5/29/2008 8:00      Glover
336        5      Cael 10024 646-423-7673   4587   6/12/2008 8:03      Glover
384        5      Cael 10024 646-423-7673   4645   7/5/2008 10:15      Glover
419        5      Cael 10024 646-423-7673   4670  7/30/2008 11:20      Glover
668        5      Cael 10024 646-423-7673   4819  10/25/2008 9:37     Glover
796        5      Cael 10024 646-423-7673   5047  2/25/2009 10:20     Glover
867        5      Cael 10024 646-423-7673   5108  3/28/2009 11:40     Glover
911        5      Cael 10024 646-423-7673   5162  4/25/2009 10:42     Glover
867        6   Patrick 11222 917-284-5780   5116   4/6/2009 12:51      Hodge
12         7  Jaqueline 10106 347-409-5439   4263   1/7/2008 11:50      Ewing
318        7  Jaqueline 10106 347-409-5439   4569    6/1/2008 9:04      Ewing
417        7  Jaqueline 10106 347-409-5439   4668  7/15/2008 15:03      Ewing
785        7  Jaqueline 10106 347-409-5439   5036  2/20/2009 10:49     Ewing
10         8     Darren 10120 212-220-3311   4261   1/5/2008 16:11      Burns
73         8     Darren 10120 212-220-3311   4324   2/1/2008 12:59      Burns
300        8     Darren 10120 212-220-3311   4551  5/22/2008 11:28      Burns
375        8     Darren 10120 212-220-3311   4626  6/28/2008 16:35      Burns
504        8     Darren 10120 212-220-3311   4755  9/12/2008 13:58      Burns
635        8     Darren 10120 212-220-3311   4886 11/27/2008 12:29     Burns
1080       8     Darren 10120 212-220-3311   5331   7/19/2009 11:58     Burns
139        9  Rigoberto 11358 212-537-4720   4390    3/4/2008 8:18      Mayo
560        9  Rigoberto 11358 212-537-4720   4811  10/20/2008 9:06      Mayo
882        9  Rigoberto 11358 212-537-4720   5133   4/9/2009 12:19      Mayo
>
```

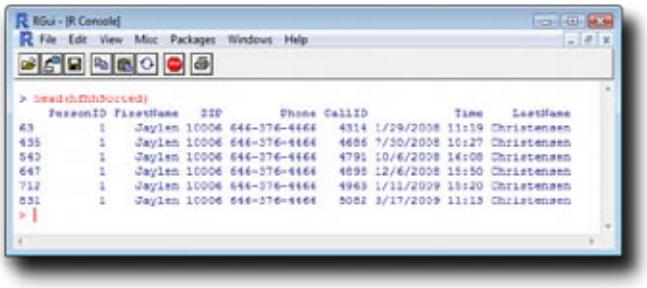
Yep, there are lots and lots of duplicate names. Lots. What a mess!

When you get messy data, you should **sort liberally**. Especially if you have a lot of records. That's because it's often hard to see all the data at once, and sorting the data by different fields lets you visualize groupings in a way that will help you find duplicates or other weirdness.

Sharpen your pencil

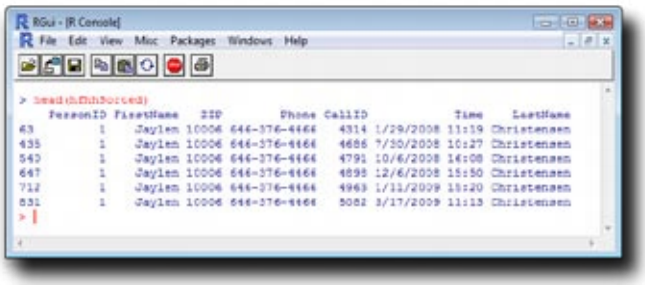
Take a close look at the data. Can you say *why* the names might be duplicated?

Write your answer here.



Sharpen your pencil Solution

Why do you think the same names show up repeatedly?

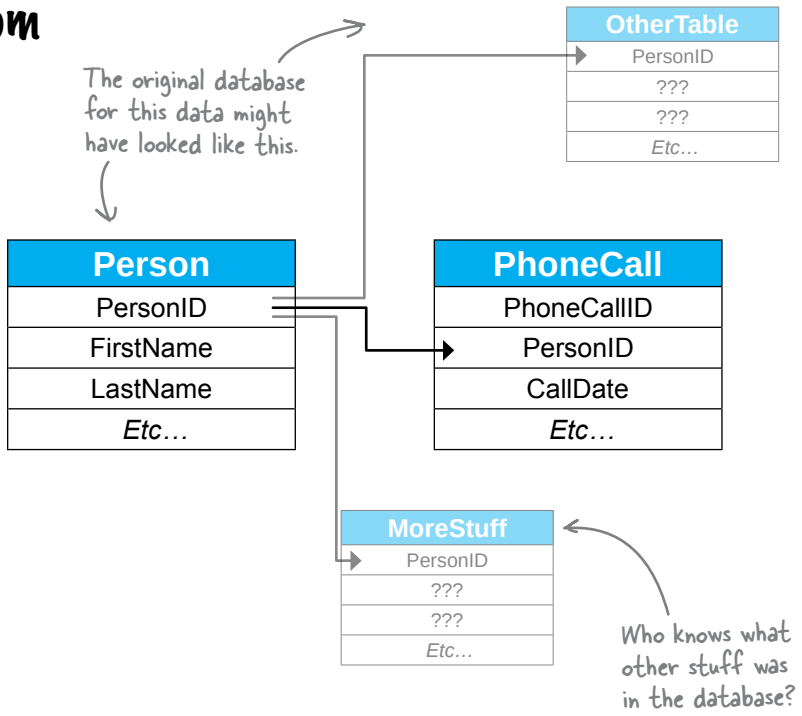


If you look at the far right column, you can see that there is a data point unique to each record: a time stamp of a phone call. That probably means that each of the lines in this database represents a phone call, so the names repeat because there are multiple calls to the same people.

The data is probably from a relational database

If elements of your messy list repeat, then the data probably come from a relational database. In this case, your data is the output of a query that consolidated two tables.

Because you understand RDBMS architecture, you know that repetition like what we see here stems from **how queries return data** rather than from **poor data quality**. So you can now remove duplicate names without worrying that something's fundamentally wrong with your data.



Remove duplicate names

Now that you know *why* there are duplicate names, you can start **removing** them. Both R and Excel have quick and straightforward functions for removing duplicates.

Removing duplicates in R is simple:

The `unique` function returns a vector or data frame like the one you specify, except that the duplicates are removed.

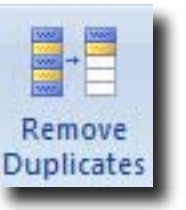
```
unique(mydata)
```

That's it! Be sure you assign the resulting value to a new name so that you can use the data `unique` returns.

In R, the unique function is what you need.

Removing duplicates in Excel is a snap:

Make sure your cursor is placed in your data and click this button:



To remove duplicates in Excel, use this button.

Excel will ask you to specify which columns contain the duplicate values, and data from other columns that isn't duplicated will be deleted.

So now that you have the tool you need to get rid of those pesky duplicate names, let's clean up your list and give it back to the client.

Fix your data once and for all in R...

1. Create a new data frame to represent your unique records:

```
hfhhNamesOnly <- hfhhSorted
```

2. Remove the `CallID` and `Time` fields, which the client doesn't need and which are the cause of your duplicate names:

```
hfhhNamesOnly$CallID <- NULL
hfhhNamesOnly$Time <- NULL
```

Here's unique in action!

3. Use the `unique` function to remove duplicate names:

```
hfhhNamesOnly <- unique(hfhhNamesOnly)
```

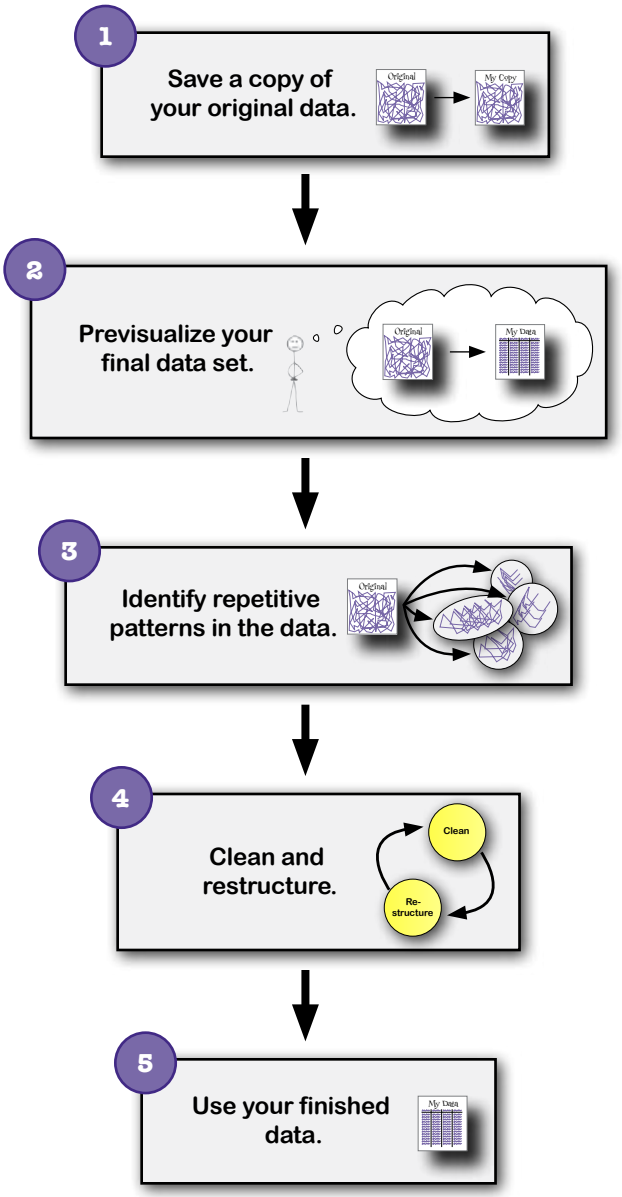
4. Take a look at your results and write them to a new CSV:

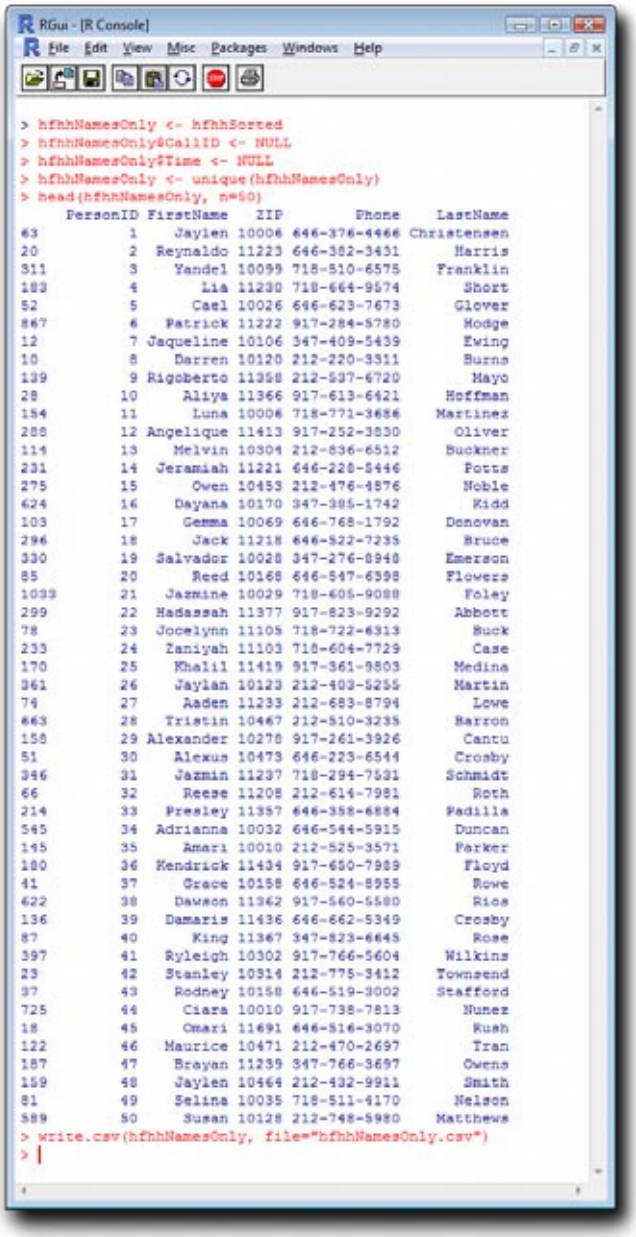
```
head(hfhhNamesOnly, n=50)
write.csv(hfhhNamesOnly, file="hfhhNamesOnly.csv")
```

You created nice, clean, unique records

This data looks totally solid.

No columns mashed together, no funny characters, no duplicates. All from following the basic steps of cleaning a messy data set:





RGui - [R Console]

File Edit View Misc Packages Windows Help

```
> hfhhNamesOnly <- hfhhSorted
> hfhhNamesOnly$CallID <- NULL
> hfhhNamesOnly$Time <- NULL
> hfhhNamesOnly <- unique(hfhhNamesOnly)
> head(hfhhNamesOnly, n=50)
     PersonID FirstName   ZIP        Phone    LastName
63          1    Jaylen 10006 646-376-4466 Christensen
20          2  Reynaldo 11223 646-382-3451      Harris
311         3    Yandel 10099 718-510-6575    Franklin
103         4       Lia 11230 718-664-9574       Short
52          5      Cael 10026 646-623-7673      Glover
867         6   Patrick 11222 917-284-5780       Hodge
12          7 Jaqueline 10106 347-409-5439       Ewing
10          8    Darren 10120 212-220-3311       Burns
139         9 Rigoberto 11358 212-597-6720        Mayo
28         10     Aliya 11366 917-613-6421     Hoffman
154        11      Luna 10006 718-771-3686    Martinez
288        12 Angelique 11413 917-252-3830      Oliver
114        13    Melvin 10304 212-836-6512     Buckner
231        14  Jeramiah 11221 646-228-5446       Potts
275        15      Owen 10453 212-476-4876       Noble
624        16    Dayana 10170 347-385-1742        Kidd
103        17     Gemma 10069 646-768-1792     Donovan
296        18      Jack 11218 646-522-7235       Bruce
330        19  Salvador 10028 347-276-8948    Emerson
85         20      Reed 10168 646-547-6398     Flowers
1033       21   Jazmine 10029 718-605-9088       Foley
299        22  Hadassah 11377 917-823-9292      Abbott
78         23  Jocelynn 11105 718-722-6313        Buck
233        24   Zaniyah 11103 718-604-7729        Case
170        25    Khalil 11419 917-361-9803      Medina
361        26    Jaylan 10123 212-403-5255      Martin
74         27     Aaden 11233 212-683-8794        Lowe
663        28   Tristin 10467 212-510-3235      Barron
158        29 Alexander 10278 917-261-3926       Cantu
51         30     Alexus 10473 646-223-6544      Crosby
346        31     Jazmin 11237 718-294-7531     Schmidt
66         32      Reese 11208 212-614-7981        Roth
214        33   Presley 11357 646-358-6884     Padilla
545        34  Adrianna 10032 646-544-5915      Duncan
145        35     Amari 10010 212-525-3571      Parker
180        36  Kendrick 11434 917-650-7989       Floyd
41         37     Grace 10158 646-524-8955         Rowe
622        38    Dawson 11362 917-560-5580         Rios
136        39   Damaris 11436 646-662-5349      Crosby
87         40      King 11367 347-823-6645        Rose
397        41   Ryleigh 10302 917-766-5604      Wilkins
23         42   Stanley 10314 212-775-3412     Townsend
37         43    Rodney 10158 646-519-3002     Stafford
725        44     Ciara 10010 917-738-7813        Nunez
18         45     Omari 11691 646-516-3070         Rush
122        46   Maurice 10471 212-470-2697         Tran
187        47    Brayan 11239 347-766-3697        Owens
159        48    Jaylen 10464 212-432-9911        Smith
81         49    Selina 10035 718-511-4170       Nelson
589        50     Susan 10128 212-748-5980     Matthews
> write.csv(hfhhNamesOnly, file="hfhhNamesOnly.csv")
>
```

Head First Head Hunters is recruiting like gangbusters!

Your list has proven to be incredibly powerful. With a clean data set of live prospects, HFHH is picking up more clients than ever, and they'd never have been able to do it without your data cleaning skills. Nice work!

Leaving town...

It's been great having you here in Dataville!

We're sad to see you leave, but there's nothing like taking what you've learned and putting it to use. You're just beginning your data analysis journey, and we've put you in the driver's seat. We're dying to hear how things go, so ***drop us a line*** at the Head First Labs website, ***www.headfirstlabs.com***, and let us know how data analysis is paying off for **YOU**!

The Top Ten Things (we didn't cover)

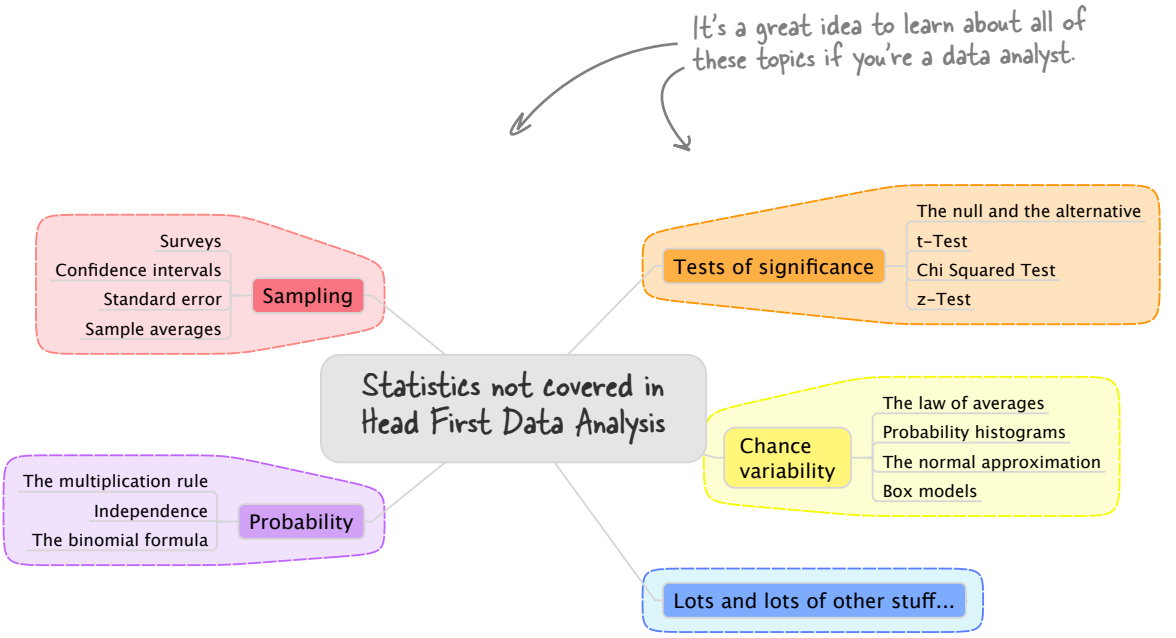
You've come a long way.

But data analysis is a vast and constantly evolving field, and there's so much left the learn. In this appendix, we'll go over ten items that there wasn't enough room to cover in this book but should be high on your list of topics to learn about next.

#1: Everything else in statistics

Statistics is a field that has a **huge array of tools and technologies** for data analysis. It's so important for data analysis, in fact, that many books about "data analysis" are really statistics books.

Here is an incomplete list of the tools of statistics not covered in *Head First Data Analysis*.



Much of what you *have* learned in this book, however, has raised your awareness of deep issues involving assumptions and model-building, preparing you not only to use the tools of statistics but also to understand their **limitations**.

The better you know statistics, the more likely you are to do great analytical work.

#2: Excel skills

This book has assumed that you have basic spreadsheet skills, but skilled data analysts tend to be spreadsheet ***ninjas***.

Compared to programs like R and subjects like regression, it's not terribly hard to master Excel. And you should!

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Analyst	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
2	Statement1	87%	88%	89%	91%	91%	92%	87%	92%	88%	92%	88%	89%	92%	88%	89%	90%
3	Statement2	68%	40%	47%	88%	37%	60%	47%	46%	59%	23%	34%	78%	70%	80%	54%	67%
4	Statement3	37%	11%	67%	7%	8%	30%	66%	41%	83%	9%	0%	46%	45%	35%	15%	63%
5	Statement4	39%	56%	33%	38%	19%	19%	27%	33%	14%	30%	58%	28%	33%	35%	16%	19%
6	Statement5	5%	28%	0%	24%	0%	18%	5%	3%	12%	9%	2%	5%	1%	13%	5%	3%
7	Statement6	77%	81%	85%	78%	72%	84%	88%	69%	74%	91%	92%	70%	3%	81%	87%	70%

#3: Edward Tufte and his principles of visualization

Good data analysts spend a lot of time reading and rereading the work of data great analysts, and Edward Tufte is unique not only in the quality of his own work but in the quality of the work of other analysts that he collects and displays in his books. **Here are his fundamental principles of analytical design**:

> "Show comparisons, contrasts, differences."
>
> "Show causality, mechanism, explanations, systematic structure."
>
> "Show multivariate data; that is, show more than 1 or 2 variables."
>
> "Completely integrate words, numbers, images, diagrams."
>
> "Thoroughly describe the evidence."
>
> "Analytical presentations ultimately stand or fall depending on the quality, relevance, and integrity of their content."
>
> —Edward Tufte

These words of wisdom, along with much else, are from pages 127, 128, 130, 131, 133, and 136 of his book *Beautiful Evidence*. His books are a gallery of the very best in the visualization of data.

What's more, his book *Data Analysis for Public Policy* is about as good a book on regression as you'll ever find, and you can download it for free at this website: *http://www.edwardtufte.com/tufte/dapp/*.

#4: PivotTables

Pivot tables are one of the more powerful data analysis tools built into spreadsheets and statistical software. They're fantastic for **exploratory data analysis** and for summarizing data extracted from **relational databases**.

From this raw data, you can create a bunch of different pivot table summaries.

hfda_ch12_articles.csv

	A	B	C	D
1	articleID	issueID	authorID	webHits
2	1	1	8	2019
3	2	1	5	1421
4	3	1	4	1174
5	4	1	4	1613
6	5	1	3	1099
7	6	1	8	1903
8	7	1	7	1718
9	8	2	1	642
10	9	2	5	1616
11	10	2	4	1233
12	11	2	8	1937
13	12	2	9	3068
14	13	3	5	1652
15	14	3	2	1028
16	15	3	1	997
17	16	3	6	1930
18	17	3	1	1035
19	18	3	3	820

hfda_ch12_articles.csv

	A	B
1	Drop Page Fields Here	
2		
3	Sum of webHits	
4	authorID	Total
5	1	138571
6	2	113546
7	3	135275
8	4	187011
9	5	173676
10	6	268413
11	7	275912
12	8	242031
13	9	383824
14	Grand Total	1918259

Here are two really simple pivot tables.

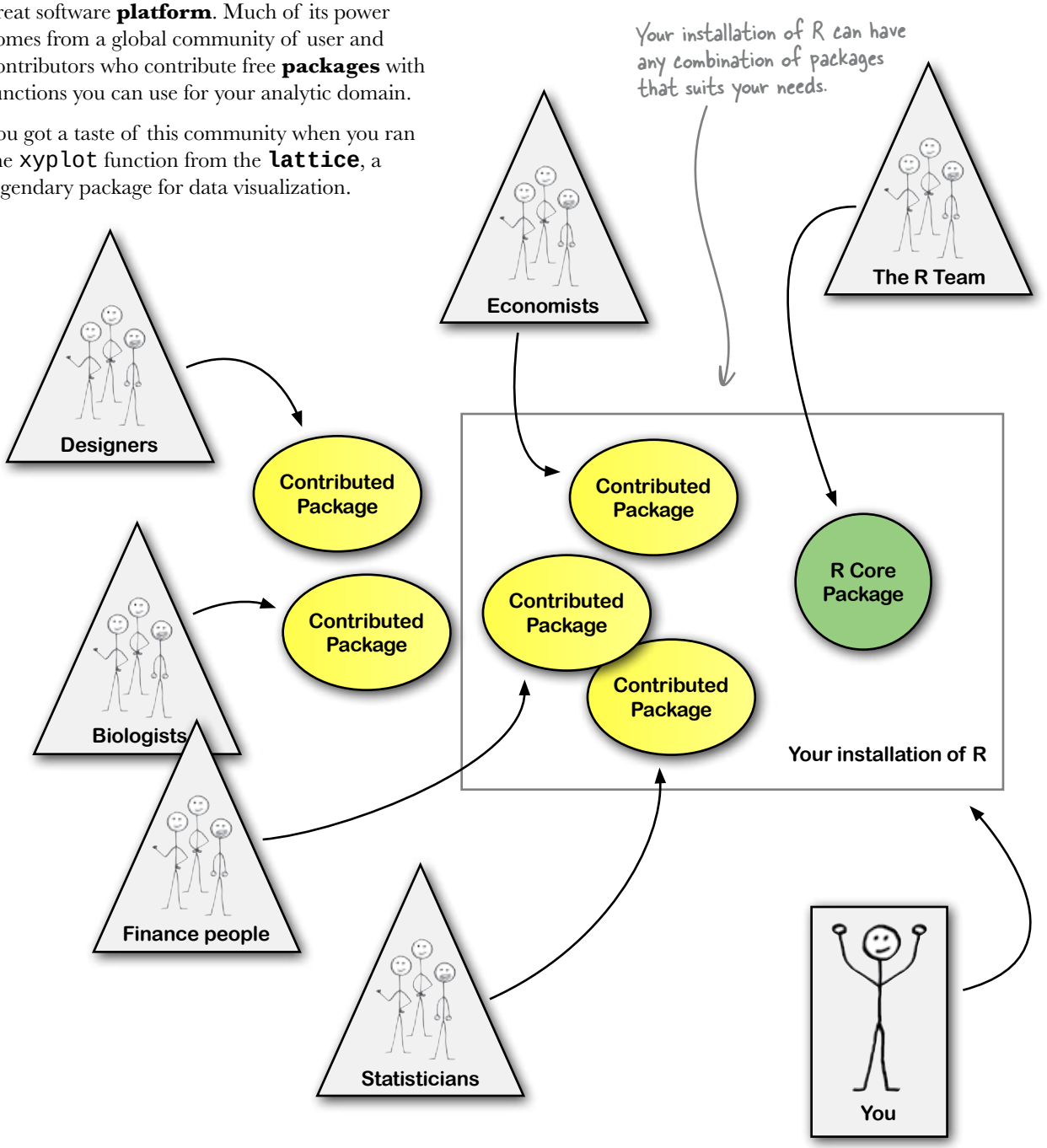
hfda_ch12_articles.csv

	A	B
1	Drop Page Fields Here	
2		
3	Sum of webHits	
4	issueID	Total
5	1	10947
6	2	8496
7	3	10620
8	4	13237
9	5	9269
10	6	10067
11	7	6953
12	8	13774
13	9	11292
14	10	9927
15	11	12719
16	12	10510
17	13	10748
18	14	9142
19	15	10728
20	16	14699
21	17	17348
22	18	10159

#5: The R community

R isn't just a great software program, it's great software **platform**. Much of its power comes from a global community of user and contributors who contribute free **packages** with functions you can use for your analytic domain.

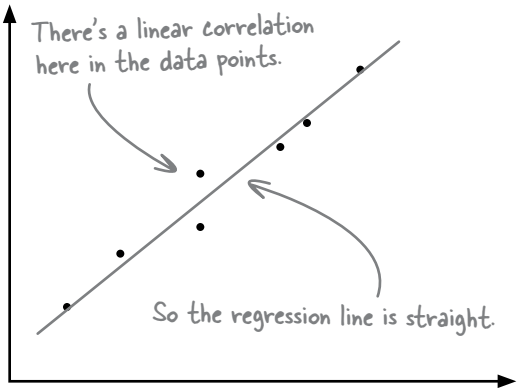
You got a taste of this community when you ran the `xyplot` function from the **`lattice`**, a legendary package for data visualization.

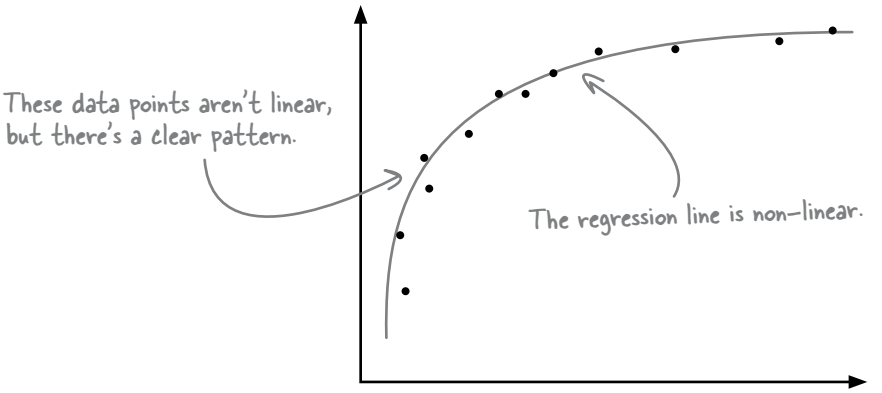


#6: Nonlinear and multiple regression

Even if your data do not exhibit a linear pattern, under some circumstances, you can make predictions using regression. One approach would be to apply a numerical **transformation** on the data that effectively makes it linear, and another way would be to draw a **polynomial rather** than linear regression line through the dots.

Also, you don't have to limit yourself to predicting a dependent variable from a single independent variable. Sometimes there are **multiple** factors that affect the variable, so in order to make a good prediction, you can use the technique of **multiple regression**.





$$y = a + bx$$

You use data this equation to predict a dependent variable from a single independent variable.

But you can also write an equation that predicts a dependent verbal from multiple independent variables.

$$y = a + bx_1 + cx_2 + dx_3 + \ldots$$

This equation is for multiple regression.

#7: Null-alternative hypothesis testing

While the hypothesis testing technique you learned in Chapter 5 is very general and can accommodate a variety of analytical problems, **null-alternative testing** is the statistical technique many (especially in academia and science) have in mind when they hear the expression "hypothesis testing."

This tool is used more often than it's understood, and *Head First Statistics* is a great place to start if you'd like to learn it.

#8: Randomness

Randomness is a big issue for data analysis.

That's because **randomness is hard to see**. When people are trying to explain events, they do a great job at fitting models to evidence. But they do a terrible job at deciding against using explanatory models at all.

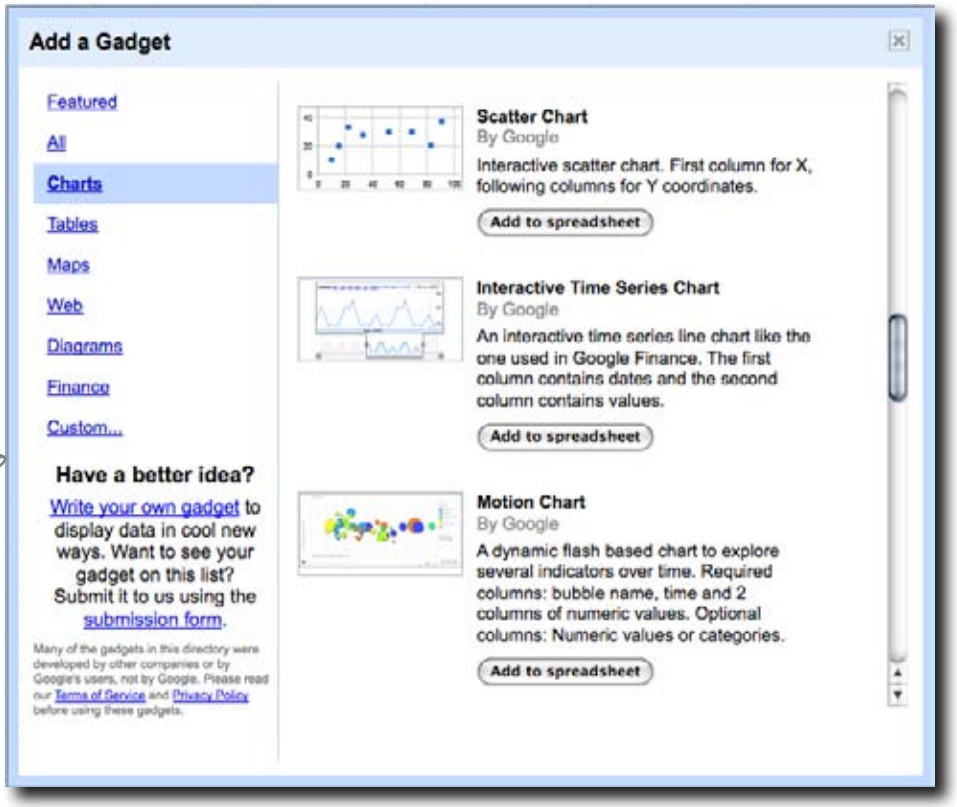
If your client asks you why a specific event happened, the honest answer based on the best analysis will often be, "the event can be explained by random variations in outcomes."

#9: Google Docs

We've talked about Excel, OpenOffice, and R, but Google Docs definitely deserves an honorable mention. Not only does **Google Docs** offer a fully functioning online spreadsheet, it has a **Gadget** feature that offers a large array of visualizations.

You can make a lot of different visualizations using the Gadget feature in Google Docs.

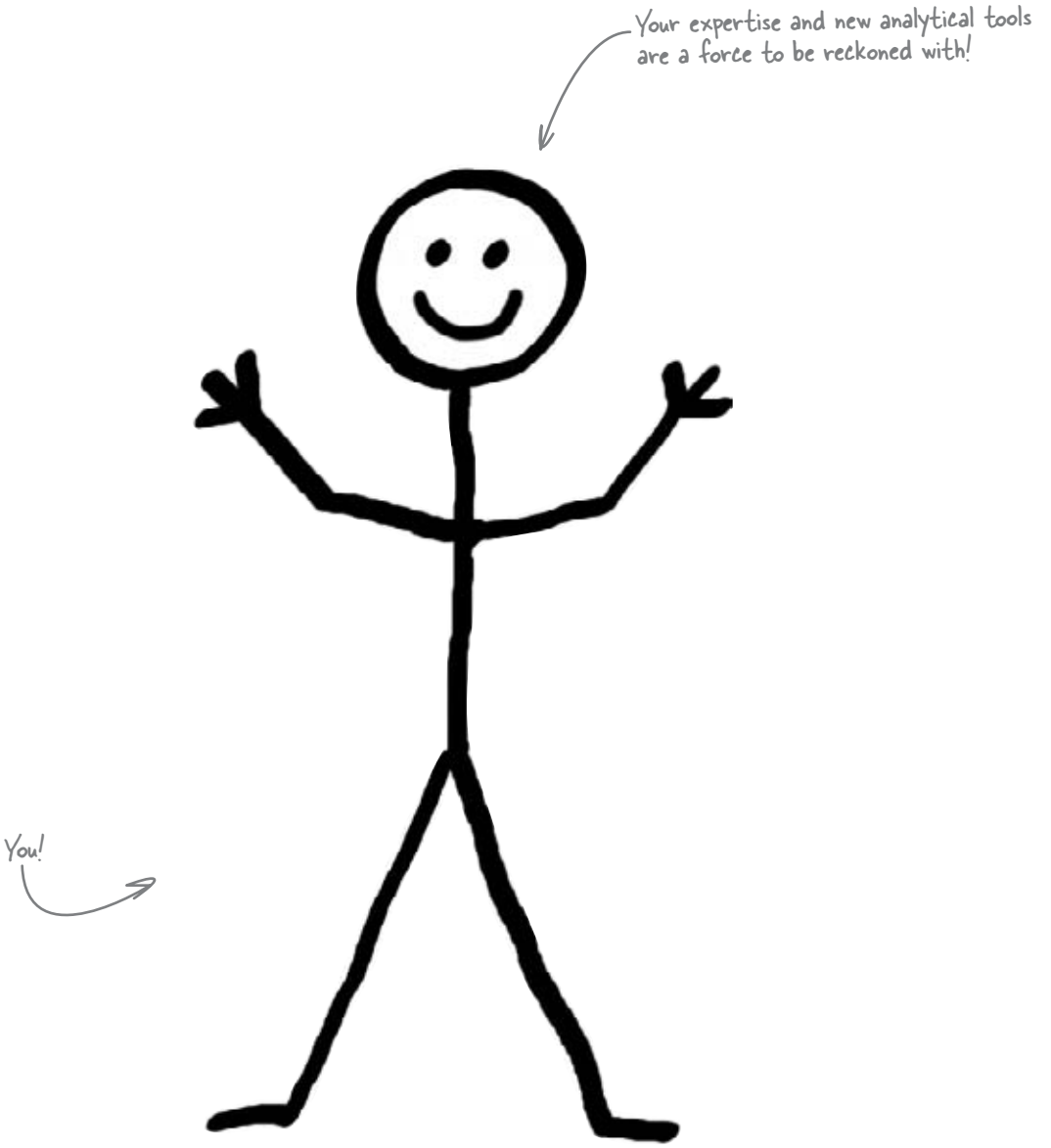
It's fun to explore the different charts that you can do with Google Docs.



What's more, Goolge Docs has a variety of functions that offer access to **real-time online data sources**. It is free software that's definitely worth checking out.

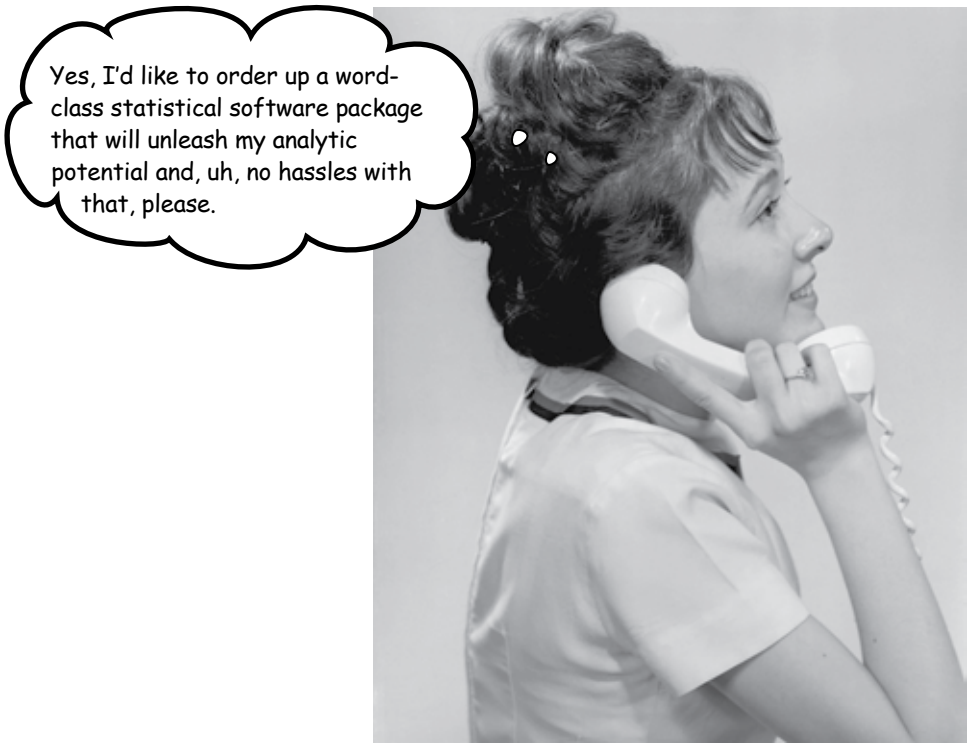
#10: Your expertise

You've learned many tools in this book, but what's more exciting than any of them is that you will combine your expertise in **your domain of knowledge** with those tools to understand and improve the world. Good luck.



appendix ii: install r

Start R up!



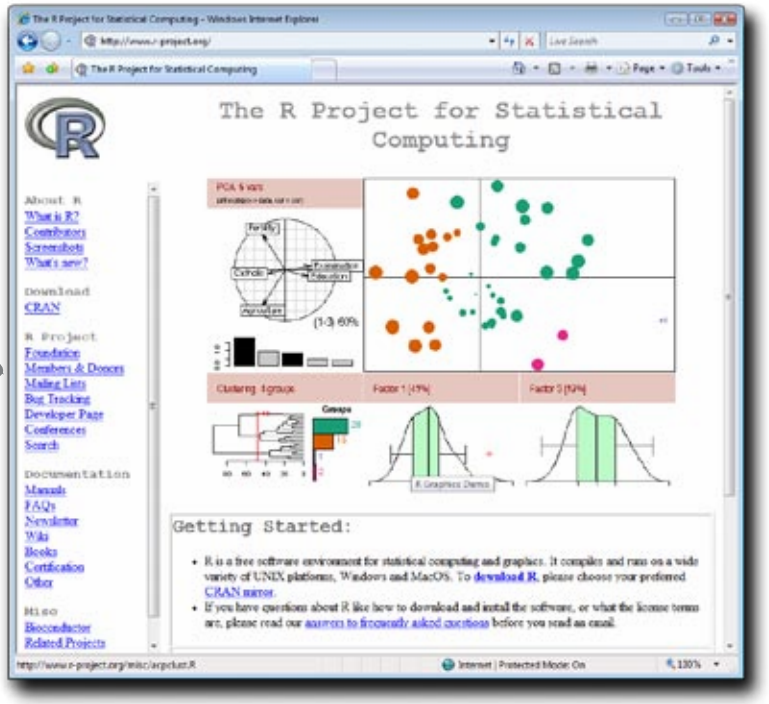
Behind all that data-crunching power is enormous complexity.

But fortunately, getting R installed and ***started*** is something you can accomplish in just a few minutes, and this appendix is about to show you how to pull off your R install without a hitch.

Get started with R

Installing the powerful, free, open source statistical software R can be done in these four quick and easy steps.

1 **Head on over** to *www.r-project.org* to download R. You should have no problem finding a mirror near you that serves R for Windows, Mac, and Linux.



Click this download link.

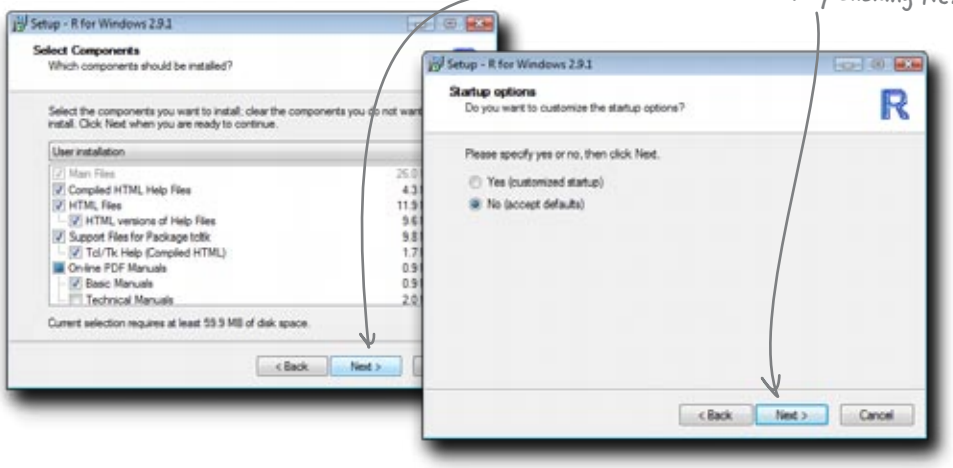
2 Once you've downloaded the program file for R, **double-click** on it to start the R installer.

This is the R program file.

Here's the R installer window.

Click here.

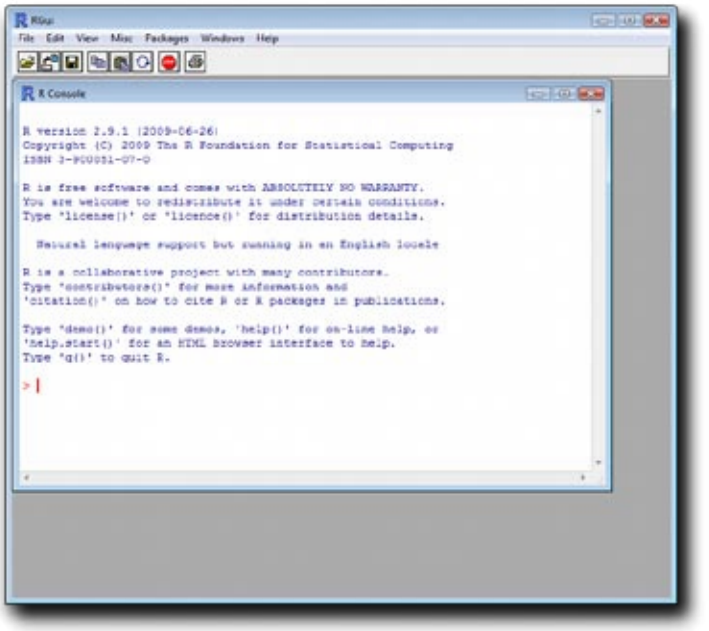
3 Accept all the default options for loading R by clicking **Next** through these windows, and let the installer do its work..



Waiting is the hardest part.

4 Click the R icon on the desktop or Start Menu, and you're ready to start using R.

Here's what R window looks like when you start it for the first time.



appendix iii: install excel analysis tools

The ToolPak

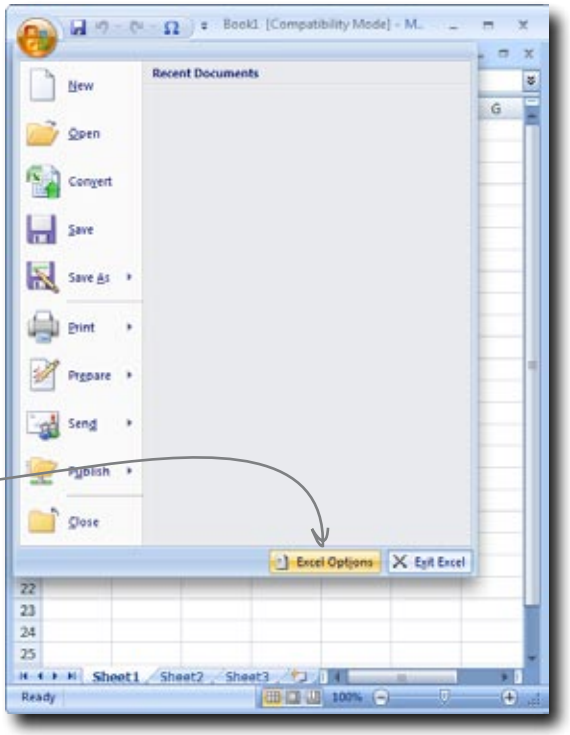
Some of the best features of Excel aren't installed by default.

That's right, in order to run the optimization from Chapter 3 and the histograms from Chapter 9, you need to activate the **Solver** and the **Analysis ToolPak**, two extensions that are included in Excel by default but not activated without your initiative.

Install the data analysis tools in Excel

Installing the Analysis ToolPak and Solver in Excel is no problem if you follow these simple steps.

This is the Microsoft Office Button.



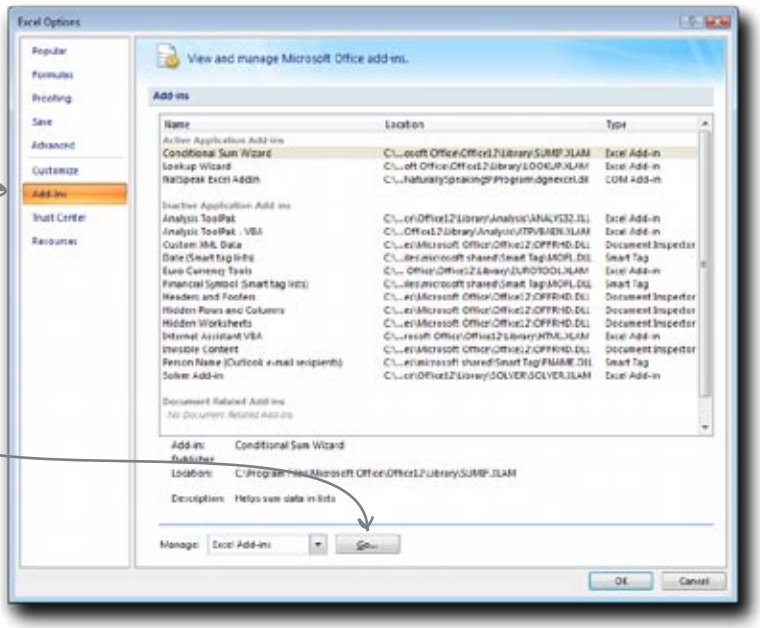
1. Click the Microsoft Office Button and select **Excel Options**.

Here's Excel Options.

2. Select the Add-Ins tab and click **Go...** next to "Manage Excel Add-Ins."

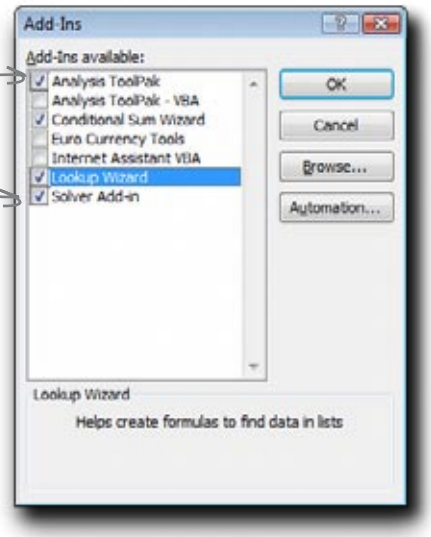
The Add-Ins tab.

Click this button.



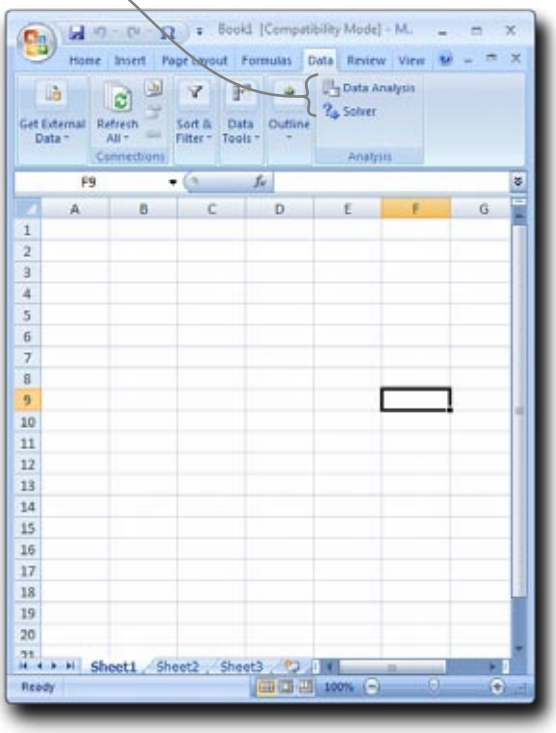
3 Make sure that the Analysis ToolPak and the Solver Add-in boxes are checked, and then press **OK**.

Make sure that these two boxes are checked.



4 Take a look at the Data tab to make sure that the Data Analysis and Solver buttons are there for you to use.

Make sure these buttons can be seen under the Data tab.



That's it!

Now you're ready to start running optimizations, histograms, and much more.

Index

Numbers

Symbols

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Y