Let's begin by creating a data frame called emails using the read.csv function.

And loading up energy_bids.csv.

And as always, in the text analytics week, we're going to pass stringsAsFactors=FALSE to this function.

So we can take a look at the structure of our new data frame using the str function.

We can see that there are 855 observations.

This means we have 855 labeled emails in the data set.

And for each one we have the text of the email and whether or not it's responsive to our query about energy schedules and bids.

So let's take a look at a few example emails in the data set, starting with the first one.

So the first email can be accessed with emails$emailemails$email[1].

Almost like the first one.

So while the output you get when you type this will depend on what operating system you're running on, many of you will see what I'm displaying here.

Which is a single line of text that we need to horizontally scroll to read through.

This is a pretty tough way to read a long piece of text.

So if you have this sort of display, you can use the strwrap function and pass it the long string you want to print out, in this case emails$email.

Selecting the first one.

And now we can see that this has broken down our long string into multiple shorter lines that are much easier to read.

OK.

So let's take a look now at this email, now that it's a lot easier to read.

We can see just by parsing through the first couple of lines that this is an email that's talking about a new working

paper, "The Environmental Challenges and Opportunities in the Evolving North American Electricity Market" is the name of the paper.

And it's being released by the Commission for Environmental Cooperation, or CEC.

So while this certainly deals with electricity markets, it doesn't have to do with energy schedules or bids.

So it is not responsive to our query.

So we can take a look at the value in the responsive variable for this email using email$responsive and selecting the first one.

And we have value 0 there.

So let's take a look at the second email in our data set.

Again I'm going to use the strwrap function.

I'm going to pass it emails$emailemails$email[1].

And scrolling up the top here we can see that the original message is actually very short, it just says FYI, for your information.

And most of it is a forwarded message.

So we have all the people who originally received the message.

And then down at the very bottom is the message itself.

"Attached is my report prepared on behalf of the California State auditor." And there's an attached report, ca report new.pdf.

Now our data set contains just the text of the emails and not the text of the attachments.

But it turns out, as we might expect, that this attachment had to do with Enron's electricity bids in California.

And therefore it is responsive to our query.

And we can check this in the responsive variable.

emails$responsive[2].

And we see that that's a 1.

So now let's look at the breakdown of the number of emails that are responsive to our query using the table function.

We're going to pass it emails$responsive.

And as we can see the data set is unbalanced, with a relatively small proportion of emails responsive to the query.

And this is typical in predictive coding problems.