Now let's build the document-term matrix for our corpus.

So we'll create a variable called dtm that contains the DocumentTermMatrix(corpus).

The corpus has already had all the pre-processing run on it.

So to get the summary statistics about the document-term matrix, we'll just type in the name of our variable, dtm.

And what we can see is that even though we have only 855 emails in the corpus, we have over 22,000 terms that showed up at least once, which is clearly too many variables for the number of observations we have.

So we want to remove the terms that don't appear too often in our data set, and we'll do that using the remove sparse terms function.

And we're going to have to determine the sparsity, so we'll say that we'll remove any term that doesn't appear in at least 3% of the documents.

To do that, we'll pass 0.97 to remove sparse terms.

Now we can take a look at the summary statistics for the document-term matrix, and we can see that we've decreased the number of terms to 788, which is a much more reasonable number.

So let's build a data frame called labeledTerms out of this document-term matrix.

So to do this, we'll use as.data.fram of as.matrix applied to dtm, the document-term matrix.

So this data frame is only including right now the frequencies of the words that appeared in at least 3% of the documents, but in order to run our text analytics models, we're also going to have the outcome variable, which is whether or not each email was responsive.

So we need to add in this outcome variable.

So we'll create labeledTerms$responsive, and we'll simply copy over the responsive variable from the original emails data frame so it's equal to emails$responsive.

So finally let's take a look at our newly constructed data frame with the str function.

So as we expect, turn off a lot of variables, 789 in total.

788 of those variables are the frequencies of various words in the emails, and the last one is responsive, the outcome variable.