Now that we have identified a set of risk factors, let's use this data to predict the 10 year risk of CHD.

First, we'll randomly split our patients into a training set and a testing set.

Then, we'll use logistic regression to predict whether or not a patient experienced CHD within 10 years of the first examination.

Keep in mind that all of the risk factors were collected at the first examination of the patients.

After building our model, we'll evaluate the predictive power of the model on the test set.

Let's go to R and create our logistic regression model.

In our R console, we'll call our data set framingham and use the read.csv function to read in the data file "framingham.csv".

Remember to navigate to the directory containing the file "framingham.csv" before reading in the data.

Let's take a look at our data by using the str function.

We have data for 4,240 patients and 16 variables.

We have the demographic risk factors male, age, and education; the behavioral risk factors currentSmoker and cigsPerDay; the medical history risk factors BPMeds, prevalentStroke, prevalentHyp, and diabetes; and the physical exam risk factors totChol, sysBP, diaBP, BMI, heartRate, and glucose level.

The last variable is the outcome or dependent variable, whether or not the patient developed CHD in the next 10 years.

Now let's split our data into a training set and a testing set using sample.split like we did in the previous lecture.

We first need to load the library caTools.

Now, let's set our seed and create our split.

We'll start by setting our seed to 1000, and then use the sample.split function to create the split.

The first argument is the outcome variable framingham$TenYearCHD.

And the second argument is the percentage of data that we want in the training set or the SplitRatio.

Here, we'll put 65% of the data in the training set.

When you have more data like we do here, you can afford to put less data in the training set and more in the testing set.

This will increase our confidence in the ability of the model to extend to new data since we have a larger test set, and still give us enough data in the training set to create our model.

You typically want to put somewhere between 50% and 80% of the data in the training set.

Now, let's split up our data using subset.

We'll call our training set "train" and use the subset function to take a subset of framingham and take the observations for which split is equal to TRUE.

We'll call our testing set "test" and again use the subset function to take a subset of framingham and take the observations for which split equals FALSE.

Now we're ready to build our logistic regression model using the training set.

Let's call it framinghamLog, and we'll use the glm function like we did in the previous lecture to create a logistic regression model.

We'll use a nice little trick here where we predict our dependent variable using all of the other variables in the data set as independent variables.

First, type the name of the dependent variable, TenYearCHD, followed by the tilde and then a period.

This will use all of the other variables in the data set as independent variables and is used in place of listing out all of the independent variables' names separated by the plus sign.

Be careful doing this with data sets that have identifying variables like a patient ID or name since you wouldn't want to use these as independent variables.

Following the period, we need to give the argument that defines the data set to use, data = train.

And then, the final argument for a logistic regression model is family = binomial.

Let's take a look at the summary of our model.

It looks like male, age, prevalent stroke, total cholesterol, systolic blood pressure, and glucose are all significant in our model.

Cigarettes per day and prevalent hypertension are almost significant.

All of significant variables have positive coefficients, meaning that higher values in these variables contribute to a higher probability of 10-year coronary heart disease.

Now, let's use this model to make predictions on our test set.

We'll call our predictions predictTest and use the predict function, which takes as arguments the name of our model, framinghamLog, then type = "response", which gives us probabilities, and lastly newdata = test, the name of our testing set.

Now, let's use a threshold value of 0.5 to create a confusion matrix.

We'll use the table function and give as the first argument, the actual values, test$TenYearCHD, and then as the second argument our predictions, predictTest > 0.5.

With a threshold of 0.5, we predict an outcome of 1, the true column, very rarely.

This means that our model rarely predicts a 10-year CHD risk above 50%.

What is the accuracy of this model?

Well, it's the sum of the cases we get right, 1069 plus 11, divided by the total number of observations in our data set, 1069 + 6 + 187 + 11.

So the accuracy of our model is about 84.8%.

We need to compare this to the accuracy of a simple baseline method.

The more frequent outcome in this case is 0, so the baseline method would always predict 0 or no CHD.

This baseline method would get an accuracy of 1069 + 6-- this is the total number of true negative cases-- divided by the total number of observations in our data set, 1069 + 6 + 187 + 11.

So the baseline model would get an accuracy of about 84.4%.

So our model barely beats the baseline in terms of accuracy.

But do we still have a valuable model by varying the threshold?

Let's compute the out-of-sample AUC.

To do this, we first need to load the ROCR package.

And then, we'll use the prediction function of the ROCR package to make our predictions.

Let's call the output of that ROCRpred and use the prediction function, which takes as a first argument our predictions, predictTest, and then as a second argument the true outcome, test$TenYearCHD.

Then, we need to type as.numeric(performance(ROCRpred, "auc")@y.values).

This will give us the AUC value on our testing set.

So we have an AUC of about 74% on our test set, which means that the model can differentiate between low risk patients and high risk patients pretty well.

As we saw in R, we were able to build a logistic regression model with a few interesting properties.

It rarely predicted 10-year CHD risk above 50%.

So the accuracy of the model was very close to the baseline model.

However, the model could differentiate between low risk patients and high risk patients pretty well with an out-of-sample AUC of 0.74.

Additionally, some of the significant variables suggest possible interventions to prevent CHD.

We saw that more cigarettes per day, higher cholesterol, higher systolic blood pressure, and higher glucose levels all increased risk.

Later in the lecture, we'll discuss some medical interventions that are currently used to prevent CHD.