For this assignment, I want you to fit a psychometric function to some data using the techniques we discussed in class today and last week. You will need to implement the code to do that, like the code I demonstrated in class.

For this assignment, there are a number of MATLAB files you can use online: `homework6.m`, `hook.m`
The data is in the MATLAB file. It’s the same data I used in some of the examples shown in class.

Unlike past assignments, I have not given you much skeleton code. Feel free to borrow from other assignments and in-class demonstrations. But for this assignment, I want to see you put all the pieces together yourself.

**Part A**

For the first part of assignment, I want you to implement code to fit the psychometric function to the data. In this case, you’ll just be fitting a single psychometric function to a single set of observed data. The data are in the form of frequencies, not probabilities, and I want you to fit the data by maximizing likelihood in the manner we discussed in class.

\[
p_{\text{cor}}(x; \alpha, \beta, \gamma, \lambda) = \gamma + (1 - \lambda - \gamma) F(x; \alpha, \beta)
\]

\[
F(x; \alpha, \beta) = 1 - \exp\left(\frac{-x}{\beta}\right)^\alpha
\]

This is the Weibull version of the psychometric function.

You should start with code used for other assignments and in class. Recall that you’ll need to “pack” the parameter vector with the four parameters (alpha, beta, gamma, and delta). If using hook, you’ll need to set the min, max, and increment appropriately. If using fminsearch, you’ll need to make sure you handle the parameter values appropriately inside your model function (i.e., recall that fminsearch lets parameter values be any real number, while some of the parameters of the psychometric function need to be greater than zero and others need to be between zero and one).

In your model function, you’ll need to unpack the parameter values, calculate the predicted psychometric function given those parameters, calculate the log likelihood of the data given the parameters, and pass the fit back to hook (recall that in this case that will be the negative of the log likelihood).

**Part B**

Does the psychometric function with four parameters fit significantly worse than a saturated model? While that question is of little interest when fitting process models to data (models that attempt to explain the data) it can be useful when fitting simple functions to data (when the goal...
is to summarize the data in another form). (In statistical data analysis, the $G^2$ value comparing the saturated model to the model you are fitting is called the deviance.)

**Part C**

Now I’d like you to implement one part of the code you would need to implement parametric bootstrapping.

The first step is creating a parametric bootstrapped sample of simulated data from the model. Use the best-fitting parameters from Part A to create the bootstrap sample. This is a simple Monte Carlo simulation.

Now fit the psychometric function to the bootstrapped sample (this is mainly copying and pasting the code from Part A, but using the bootstrapped simulated data rather than the actual observed data).

To do a full bootstrapping, you would repeat this process 1000’s of times.