Here is a Matlab script that runs an example classifier. It is recommended that you run the script step by step (cut and paste), relating each step to the class slides. As you go along, answer the questions posed in red.

```matlab
%% Simulated data for a 1-D brain

clear all; close all; clc;

nVox=500;       % number of voxels
nPerc=0.2;      % percentage of 'active' voxels
nTrials=100;    % total number of trials (per condition)
nTraining=90;   % number of trials for training

% condition vector (class 1=0, class 2=1)
condition=zeros(2*nTrials,1);
condition(nTrials+1:end)=1;

% generate the design matrix
X(:,1)=1-condition; X(:,2)=condition;

% spatial activation pattern
spatial=rand(nVox,1);
spatial(spatial<1-nPerc)=0;

data=(spatial*condition').';
data=data+randn(size(data))*3;

figure(1);
imagesc(data); xlabel('voxels'); ylabel('trials');

%% estimate the beta vector and error variance
beta_hat=inv(X'*X)*X'*data;
Var_e=diag((data-X*beta_hat)'*(data-X*beta_hat)/(nTrials-rank(X))).';

%% hypothesis testing; compute the t statistic

%% split data (training / test)
idx0=find(condition==0);
idx1=find(condition==1);
```

Understand the way how the data is generated and also study the visualization in Figure 1. Do you (visually) recognize a difference between trials of the two conditions?

It very hard to see any difference between the first half of the rows (condition 0) and the second half (condition 1).

Is this high-dimensional data if you consider the voxels as features?

Yes, the number of features (i.e., 500 voxels) is higher than the number of trials (i.e., 100 per condition)

We first perform a GLM analysis (two-sample t-test) on the full data. Does any of the voxels survive statistical thresholding?

No (at least not in any of the runs I performed). There is no "magical" voxel that allows you decide upon the condition. In other words, univariate analysis does not reveal any difference between the conditions.

Then we split the data into training and test subsets.
We use a Naive Bayes (NB) classifier, which uses the marginal (Gaussian) distribution of the different features (voxels). What are the parameters of the NB model?

The mean and variance per feature and per condition.

Does the NB model use different type of information than the GLM above?

No. The same noise model (Gaussian, through the likelihood function) is applied with the same assumption of independence.

Study the eigenvalues of the empirical covariance matrix? What do you observe? Try to estimate its inverse! Which additional assumption is made in NB to solve this problem?

Only 90 eigenvalues are non-zeros, which corresponds to the number of training samples. The 500x500 feature covariance matrix cannot be estimated reliably and its inversion is problematic (would require regularization). As NB assumes independence, it only relies on the diagonal of the covariance matrix, which can be inverted.

Next, we evaluate the model for the test data by computing the posterior probability.

Run the code a couple of times. Is the NB classifier successful (on average) to retrieve the condition for (individual) test trials? Yes

Do you expect more advanced classifiers that incorporate correlation structure between voxels to perform better for this data? Yes, the same voxels are activated for one condition and exploiting correlation can help.