For this assignment, you will try to classify some speech data. I have provided a dataset with [vowel formants](https://en.wikipedia.org/wiki/Formant) from one speaker (90% of the tokens are training, 10% are test), as well as what the vowel label of each token is.

vowelformants\_train.csv

vowelformants\_test.csv

For this programming assignment, you are to code your own estimators and not use off the shelf learners or pre-defined functions.

Part 1 (5 points): Plot the vowel formants as a 2-d scatterplot with a different color for each vowel. Comment on what you see versus the definition of [vowel formants](https://en.wikipedia.org/wiki/Formant) you read on Wikipedia.

Part 2 (20 points): Develop a Bayesian model that computes P(Vowel|Formants), calculating P(Formants|Vowel) using a 2-d diagonal covariance Gaussian, and calculating P(Vowel) from the training data. Predict the most likely vowel for each test set token, and calculate an accuracy on the test set.

Notes on calculating P(Formants|Vowel):

1) Gaussians (like any real probability distribution) are probability density functions, not probabilities, so technically one needs to integrate a region around a point to get a probability. However, since the points are fixed, the common usage is to use the value of the PDF as a scaled probability - since you'll be eventually normalizing this over all classes in the Bayesian network, it will work out fine.

2) With a 2-dimensional Gaussian, you'll need a 2-d mean and 2-d covariance matrix, which will be diagonal. HOWEVER, you can think of this as the product of 2 1-d Gaussians if it makes it easier, giving P(Formant1|Vowel) and P(Formant2|Vowel). What kind of conditional independence assumptions does this imply?

Part 3 (25 points): Replace P(Formants|Vowel) in your Bayesian network from Part 2 with a Mixture of Gaussian model, where you create one mixture model for each vowel. Train the P(Formants|Vowel) model using the EM algorithm. For testing, marginalize over the mixture components to compute P(Formants|Vowel). Report test accuracy for 2, 3, and 4 mixtures per vowel.

Notes on using MoG for P(Formants|Vowel):

The example for MoG that we had talked about estimated P(X) using a mixture class \sum\_C P(X|C)P(C). Think for a minute what the Bayesian network would look like. How many variables would there be in a MoG for P(Formants|Vowel)? What is the connectivity of the BN?

Bonus (10 points): Reimplement Part 2 (5 points) and Part 3 (5 points) using full-covariance Gaussians. How does this affect performance? Particularly compare a single full-covariance Gaussian against your results in part 3.