HW #2
DISC 101
K-means

A full write-up of the nutrition information found at

Use a subset of the data for the calculations:
https://www.dropbox.com/s/Sihee5dd3ockr9ys/nutrition.csv

1. By hand, cluster the sandwiches in nutrition.csv using k = 2. You must use both calories and fat when computing the distance, which means you must use the Euclidean distance (http://en.wikipedia.org/wiki/Euclidean_distance). Use the Angus Deluxe and the Hamburger as your initial cluster means. Pick 6 of the sandwiches to cluster.

2. Create your first program using MATLAB to cluster the above data. Here are the steps to create your first MATLAB program:
   a. Change the current folder in MATLAB to the directory containing nutrition.csv.
   b. Now we need to enter a command to read the data from nutrition.csv:

   ```matlab
   C = textscan(fopen('nutrition.csv','r'),
   '%s%f%f%f%f%f%f%f%f%f%f%f','delimiter','\t','headerlines',1);
   ```

   c. Now we need to select just the features we want to cluster:

   ```matlab
   data = cell2mat(C(3:end));
   ```

   d. Now we need to cluster them using k-means:

   ```matlab
   [cidx,ctrs] = kmeans(data,2);
   ```

   e. Again, this has too many features to visualize, so we'll need to run principal component analysis:

   ```matlab
   [pc, score, latent, tsquare] = princomp(data);
   disp(cumsum(latent)./sum(latent));
   m = mean(data);
   plot(score(cidx==1,1),score(cidx==1,2),'r.',score(cidx==2,1)
   ,score(cidx==2,2),'b.',(ctrs(1,:) - m)*pc(:,1),(ctrs(1,:) -
   m)*pc(:,2),'kx', (ctrs(2,:) - m)*pc(:,1), (ctrs(2,:) -
   m)*pc(:,2),'kx','markersize',15);
   xlabel('Principal Comp 1');
   ylabel('Principal Comp 2');
   ```
f. Now it would be great to add some labels, so here is the code to add labels:

```matlab
for i = 1:size(data,1) % Loop through each sample
    text(score(i,1),score(i,2),C{1}(i));
end
```