Multidimensional Scaling (MDS): A tool for visualizing psychological spaces

In this week's lectures, we've looked at how similarities can be represented in a psychological "space". In fact, you can visualize these "spaces" very easily by performing **multidimensional scaling (MDS)** using MATLAB, provided that you've got some "distance" representations.

In similarity studies, subjects are often asked to give ratings on how similar two items are to each other. We can then transform these similarity ratings to "distances in psychological space" based on Shepard's "universal" law. With these "psychological distances between items", we can make use of some nice built-in functions in the MATLAB Statistics Toolbox, which do MDS for you in just a blink.

You may find the example of transforming distances between US cities to a 2D map very cool.
In fact, the MATLAB guys do, too. So, they've used that as an example in the Help for the MDS function.

If you go to Help, and search for "multidimensional scaling", you'll get this (scroll down a little):

Example: Multidimensional Scaling

Given only the distances between 10 US cities, `cmdscale` can construct a map of those ci
`cmdscale`. In this example, `D` is a full distance matrix: it is square and symmetric, has pos

```
cities =
{'Atl','Chi','Den','Hou','LA','Mia','NYC','SF','Sea','WDC'};
D = [    0  587 1212  701 1936  604  748 2139 2182   543;
       587    0  920  940 1745 1188  713 1858 1737   597;
      1212  920    0  879  831 1726 1631  949 1021  1494;
       701  940  879    0 1374  968 1420 1645 1891  1220;
      1936 1745  831 1374    0 2339 2451  347  959  2300;
       604 1188 1726  968 2339    0 1092 2594 2734   923;
       748  713 1631 1420 2451 1092    0 2571 2408   205;
      2139 1858  949 1645  347 2594 2571    0  678  2442;
      2182 1737 1021 1891  959 2734 2408  678    0  2329;
       543  597 1494 1220 2300  923  205 2442 2329     0];
[Y,eigvals] = cmdscale(D);
```

In the last line, **`cmdscale()`** is the built-in function for the **classical (or metric) MDS**. All you need is to plug in a distance matrix, as show in this city example. The really cool thing is that, for our purpose, this function works just as fine if you plug in a similarity matrix, provided that you accept their way of transforming similarities to distances (read more about this on the Help).

In order to see the plot of MDS result, you'll also need the following bit of code (just scroll down a little more in MATLAB Help):

```
plot(Y(:,1),Y(:,2),'.')
text(Y(:,1)+25,Y(:,2),cities)
xlabel('Miles')
ylabel('Miles')
```

You should get exactly the same plot as shown in Help, and it's the same as the one we saw in lecture.

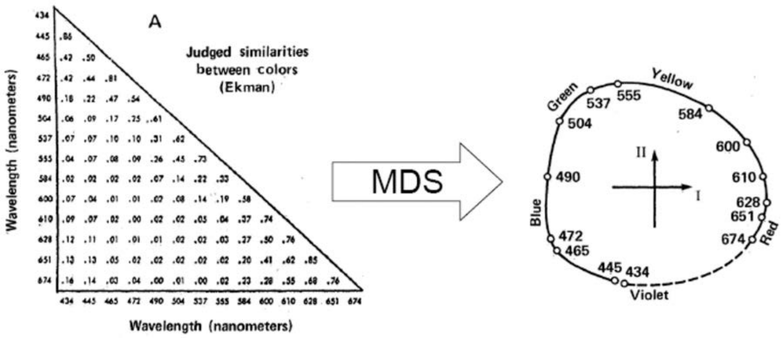
Another built-in function is **`mdscale()`** , which performs **non-metric MDS** (you can also use it for metric MDS if you plug in correct parameters; see Help for details).

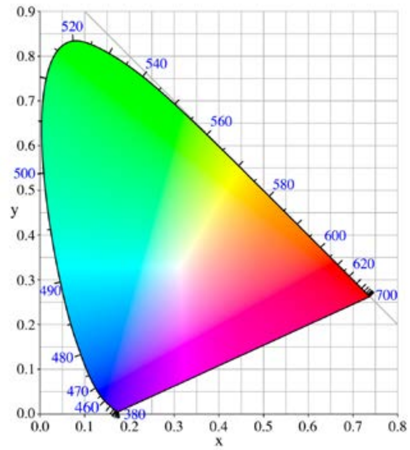
In-class Exercise (submit mdscode.m at CCLE)

Prof. Michael Lee at UC Irvine provided a dataset that contain similarity/distance measurements from many psychological experiments. You'll find references to all these studies, and details about all the data sets: http://faculty.sites.uci.edu/mdlee/similarity-data/

As detailed in the website, each data file contains 4 variables:

1. `n` : the number of items
2. `d` : a normalized distance matrix (i.e., maximum distance = 1, minimum = 0)
3. `s` : a normalized similarity matrix (i.e., maximum similarity = 1, minimum = 0)
4. `labs` : an ordered list of strings, labeling each corresponding item in the matrices





I included a few data sets downloaded from Prof. Michael Lee's website. First, let's take a look at color similarity:

1. Load the **colour.mat** data file. Simply type load('colour.mat')
2. For this color data set, we will use the similarity matrix `s` as the input.
3. Use the MATLAB built-in function `mdscale()` with the input argument of the similarity matrix. The second argument refers to the number of "dimensions" you'd like your visualization to be. Let's consider a 2D psychological space for now.

   ```
   [Y,eigvals] = mdscale(s,2);
   ```

4. Copy and paste the plotting bit from the US-cities example. However, you'll have to:
   a. change the variable "`cities`" to "`labs`" (for labeling the plot)
   b. change the number 25 to a much smaller number, e.g., 0.01. This number controls how far away you'd like each text label to be away from each dot
   c. Remove the two lines for labeling the x- and y- axes
5. You should get something very similar to the figure above, which is what we saw in lecture.

6. Try running the *classical MDS* on the color data **colour.mat** using the `cmdscale()` function. I've attached an image (the right color image above, from Wikipedia) of a computationally-defined color space, proposed by the CIE (an international authority on light and color) as a standard for color printing. Compare the image with your plot. Do you see any similarities? Write down a couple of sentences to summarize your comparisons as comments in your Matlab code.

7. Please try the other dataset **sport.mat using non-metric MDS** to recover the sport space. After getting the MSD plot, please interpret what the two axes might mean. Write down your interpretation as comments at the end of your matlab code.

8. Please try the dataset **fruits.mat using non-metric MDS** to recover the fruit space. Write down your interpretation as comments at the end of your matlab code. Can you locate an item called "fruit"? Is it located at the center of the space or at a corner? Why would it locate in the position within the psychological space?