6. *Some simple visualization methods.* For this problem, we'll be using the *animals with attributes* data set. Go to

   `http://attributes.kyb.tuebingen.mpg.de`

   and, under "Downloads", choose the "base package" (the very first file in the list). Unzip it and look over the various text files.

   This is a small data set that has information about 50 animals. The animals are listed in `classes.txt`. For each animal, the information consists of values for 85 features: does the animal have a tail, is it slow, does it have tusks, etc. The details of the features are in `predicates.txt`. The full data consists of a $50 \times 85$ matrix of real values, in `predicate-matrix-continuous.txt`. Load this real-valued array.

   (a) We would like to visualize these animals in 2-d. Do this with a PCA projection from $\mathbb{R}^{85}$ to $\mathbb{R}^2$. Show the position of each animal, and label each with its name.

   Python notes: You will need to make the plot larger by prefacing your code with

   ```
   from pylab import rcParams
   rcParams['figure.figsize'] = 10, 10
   ```

   (or try a different size if this doesn't seem right).

   (b) A popular visualization method is the *t-SNE algorithm.* This method takes a numerical parameter called the *perplexity* and then obtains an embedding by solving a non-convex optimization problem.

   The t-SNE algorithm is built into `scikit-learn`. You can invoke it to get a 2-d embedding of a data set $X$ by using:

   ```
   from sklearn.manifold import TSNE
   Z = TSNE(n_components=2, perplexity=10.0).fit_transform(X)
   ```

   The *perplexity* has a significant effect on the output. Try different perplexity values (5, 10, 25, 50) and show each of these embeddings, as you did with the PCA embedding.

   Bear in mind that t-SNE is a local search algorithm with randomized initialization, and thus even for a fixed perplexity value, different calls to it can return different results.

   (c) How can we evaluate these embeddings? Some might seem more visually pleasing than others, or might seem to group animals in a way that agrees more with our own intuitions.

   Let's look at a somewhat more objective measure. Say we have a data set of $n$ points $x_1, \ldots, x_n \in \mathbb{R}^d$ and we somehow obtain a 2-d visualization $z_1, \ldots, z_n \in \mathbb{R}^2$ of them. How accurately does this 2-d embedding capture the original interpoint distances? To see this,

   - Define $D_{ij} = \|x_i - x_j\|$ and $\widehat{D}_{ij} = \|z_i - z_j\|$.
   - In general, the $D$ values might be scaled differently from the $\widehat{D}$ values; so define the scaling factor to be $c = \text{mean}(D)/\text{mean}(\widehat{D})$, where mean$(\cdot)$ denotes the average over all $n^2$ entries of the matrix.
   - The multiplicative factor by which the distance between $x_i$ and $x_j$ is distorted can be defined by
   $$\Delta_{ij} = \max\left(\frac{D_{ij}}{c \cdot \widehat{D}_{ij}}, \frac{c \cdot \widehat{D}_{ij}}{D_{ij}}\right).$$
   This ratio is always $\geq 1$.
   - The *average distortion* is then mean$(\Delta)$.

   Compute the average distortion for each of the five embeddings you found (PCA and four t-SNE embeddings). Which of them fares best under this measure?