B565: Homework 2

1. This problem treats the onset time data from the Rachmaninov Piano Concerto #2, discussed in class, available on Canvas as rachmaninov_pc2_onset.csv as well as the corresponding audio data. These data give the onset times for the notes of the first several measures in the piano's entrance. In this simple case the piano plays 12 evenly spaced notes for each measure, so there are 12 observations for each row. The observations are the onset times of the notes in seconds. These data can be read into R with

```
> read.csv("rachmaninov\_pc2\_onset.csv")
```

   (a) Plot the data on a single plot showing the onset times for each measure and making it easy to distingish the times of one measure from another.

   (b) We demonstrated in an R example that the sample covariance matrix, S, can be computed from a data matrix, X, with one observation in each row, by first "centering" X (subtracting the sample mean vector from each row) and letting $S = \frac{1}{n} X^t X$. Show that this is correct.

   (c) $u_1, \ldots, u_p$ is an *orthonormal basis* if

$$(u_i, u_j) = \begin{cases} 1 & i = j \\ 0 & \text{otherwise} \end{cases}$$

   where $(x, y) = \sum_{i=1}^{p} x_i y_i$ is the inner product of x and y. This means the vectors $u_1, \ldots, u_p$ all have length 1 and are at right angles to one another. A basic fact from linear algebra states that any vector, x, in p-dimensional space, can be expanded as a linear combination of the $\{u_i\}$ by $x = \sum_{i=1}^{p} (x, u_i) u_i$, or, equivalently,

$$x = \mu + \sum_{i=1}^{p} (x - \mu, u_i) u_i$$

   This has an interesting interpretation in terms of principal component analysis where we represent $\Sigma = UDU^t$. In the usual statistical formulation of PCA the data are assumed to have a multivariate normal distribution, which we can interpret to mean having an "ellipsoidal point cloud" distribution. In this multivariate normal case we learned that the inner product $(x - \mu, u_i)$ has a $N(0, d^i)$ distribution where $d_i = D_{ii}$ and that these p inner products are independent. This gives us a way of *generating* a representative x. Using the above equation we can synthesize representative samples by

$$x = \mu + \sum_{i=1}^{p} z_i u_i$$

   where the $\{z_i\}$ are independent and $z_i \sim N(0, d_i)$. Write R code to synthesize data 1000 data points from the Rachmaninov distribution using these ideas.

   (d) The most important part of this problem is *interpreting* the $\{u_i\}$. Since the u_1, u_2 are the directions with the greatest variance we could approximate our generation mechanism by $x \approx \mu + z_1 u_1 + z_2 u_2$.

      i. Interpret, in descriptive language, how the measures vary from one another when approximated in terms of the direction of greatest variance: $x \approx \mu + z_1 u_1$.

      ii. Same question for $x \approx \mu + z_1 u_1 + z_2 u_2$.

2. Suppose that x is a p-dimensional random vector with mean μ and covariance $\Sigma = UDU^t$ where

$$U = \begin{pmatrix} \vdots & \vdots & \vdots & \vdots \\ u_1 & u_2 & \ldots & u_p \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix}$$

$$D = \begin{pmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_p \end{pmatrix}$$

with $u_1, \ldots, u_p$ orthonormal. Show that

$$\mathrm{Cov}(\mathrm{u_i^t(x-\mu)}, \mathrm{u_j^t(x-\mu)}) = \begin{cases} d_i & i = j \\ 0 & \text{otherwise} \end{cases}$$

3. Consider the "mystery.csv" data set available from Canvas. You can read this data set in with the read.csv R function.

   ```
   > read.csv("mystery.csv")
   ```

   Using principal components, find the effective dimension of the data — this is, dimension of the hyperplane that contains the data points. Using this result perform dimensionality reduction on the data matrix to give a smaller number of columns without any loss of information.

4. The "mlbench" R package contains a number of datasets used for Machine Learning benchmarking. Install this package with

   ```
   > install.packages("mlbench")
   ```

   When you want to access one of these datasets in your R source file, you need the command

   ```
   > library("mlbench")
   ```

   If you wanted to load, for instance, the "Ionosphere" datset, you would also need the command

   ```
   > data("Ionosphere")
   ```

   as we have done before. You can see that the 35th attribute of this dataset gives a classification of "good" or "bad," thus dividing the dataset into two classes. For each of these classes compute the empirical covariance matrix, and use these to compute the Mahalanobis distance from each point to the two class means. A simple-minded classifier might associate each point with the class having the smaller Mahalanobis distance. Classify each point according to this rule, and tabulate the "confusion matrix." That is, create a 2x2 matrix where the i, j entry is the number of examples from class i (good or bad), which were classified as type j (good or bad).

5. Consider the space of binary p-tuples: $\{0,1\}^p = \underbrace{\{0,1\} \times \{0,1\} \times \ldots \{0,1\}}_{p \text{ times}}$. Show that for binary p-tuples, x, y, $d(x,y) = |\{i \in 1, \ldots, p : x_i \neq y_i\}|$ is a distance. In other words, $d(x,y)$ is the number of coordinates on which x and y disagree.