1. Fit the polynomial $y = \theta_0 + \theta_1 x + \theta_2 x^2$ to the data points $(x, y) = (1, 4), (2, 6), (3, 15), (4, 19)$ by setting up a system of equations $A\vec{\theta} = \vec{b}$ where A is 4×3 and $\vec{b}$ is 4×1 and $\vec{\theta} = (\theta_0, \theta_1, \theta_2)^T$. Find the total least squares solution using the singular value decomposition (SVD). Make a plot in MATLAB/Python showing the four data points and the linear least squares polynomial and the total least squares polynomial. Explain the difference between the linear least squares and total least squares fit in your plot.

2. Notice that if $[A\vec{b}] = USV^T$ then we can define a matrix $C = [A\vec{b}]^T[A\vec{b}] = VSU^TUSV^T = VS^2V^T$. Recall that total least squares only requires the smallest singular vector of$[A\vec{b}]$,so find the smallest eigenvalue and the corresponding eigenvector of C using inverse iteration.

3. One method of compressing an image is to apply the SVD directly to the image matrix. Use the any sampling method to convert any of the images in the zip file to a 100×100 matrix to say that A, the matrix A is a 100×100 matrix which represents the pixels in the image. Use the SVD to find the best rank-5 approximation to the matrix A and view this matrix as an image. Repeat for rank-10 and rank-20. Plot the singular values and comment on the relationship between the singular values and the quality of the approximate image.

4. define $function \quad [X, Z] = PCA(Y, k)$ and complete it by implementing the following algorithm:
   Input: An $m \times N$ data matrix Y with columns y_i and the desired number k of principal coordinates.
   Output:A $k \times N$ matrix X with columns x_i which are the projection of the data points y_i onto the first k principal components. An $m \times N$ matrix Z which is the reconstruction of Y using the first k principal components.

   (a) Compute the mean of the data set $\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i \in \mathbb{R}^n$
   (b) Center the data set by subtracting$\bar{y}$ from each data point, let $\bar{Y}$ have columns $\bar{y_i} = y_i\bar{y}$.
   (c) Compute the singular value decomposition$(SVD)\bar{Y} = USV^T$
   (d) Define $U_k = UI_{n\times k}$ to be the first k columns of U.
   (e) Define the $k \times N$ principal coordinate matrix $X = U_k^T Y$ where x_i is the i-th column of X
   (f) Define the mN matrix $\bar{Z} = U_k U_k^T Y$with columns $\bar{z_i}$ (reconstruction of $\bar{Y}$).
   (g) Define the$m \times N$ reconstructed data matrix Z

5. (Additional Score) **Goal:**In this part , you will run PCA on a real data set, and interpret the output.

   Description: Download the genomdata.txt file from the zip file . The data represented there is from the 1000 genomes project. Each of the 995 lines in the file represents an individual. The first three columns represent respectively the individual's unique identifier, his/her sex (1=male, 2=female) and the population he or she belongs to1 . The subsequent 10101 columns of each line are a subsample of nucleobases from the individual's genome.

   We will be looking at the output of PCA on this dataset. PCA can refer to a number of related things, so to be explicit, in this section when we say "PCA" we mean

- The data should be centered (i.e., the sample mean subtracted out) but not normalized.
- The output should be the normalized principal components (i.e., unit-length eigenvectors).

Feel free to use a library implementation of PCA for the following questions. For python users, we recommend scikit learn's implementation. Matlab's built-in pca function can also be used. Note that with both python scikit and Matlab, you can specify how many principal components you want (this can save on computation time).

Exercises: First convert the data from the text file of nucleobases to a real-valued matrix (PCA needs a real-valued matrix). Specifically, convert the genetic data into a binary matrix X such that $X_{ij} = 0$ if the i^{th} individual has column js mode nucleobase2 for his or her j th nucleobase, and $X_{ij} = 1$ otherwise. Note that all mutations appear as a 1, even if they are different mutations, so if the mode for column j is "G", then if individual i has an "A","T", or "C", then X_{ij} would be 1.

The first 3 columns of the data file provide meta-data, and should be ignored when creating the binary matrix X. We will examine genotypes to extract phenotype information.

(a) Say we ran PCA on the binary matrix X above. What would be the dimension of the returned vectors?

(b) We will examine the first 2 principal components of X. These components contain lots of information about our data set. Create a scatter plot with each of the 995 rows of X projected onto the first two principal components. In other words, the horizontal axis should be v_1, the vertical axis v_2, and each individual should be projected onto the subspace spanned by v_1 and v_2. Your plot must use a different color for each population and include a legend.

(c) In two sentences, list 1 or 2 basic facts about the plot created in part (b). Can you interpret the first two principal components? What aspects of the data do the first two principal components capture? Hint: think about history and geography.

(d) We will now examine the third principal component of X. Create another scatter plot with each individual projected onto the subspace spanned by the first and third principal components. After plotting, play with different labeling schemes (with labels derived from the meta-data) to explain the clusters that you see. Your plot must include a legend.

(e) Something should have popped out at you in the plot above. In one sentence, what information does the third principal component capture?

(f) In this part, you will inspect the third principal component. Plot the nucleobase index vs the absolute value of the third principal component. What do you notice? What's a possible explanation? Hint: think about chromosomes.

Deliverables: Scatter plot for part (b). Short discussion for part (c). Scatter plots for parts (d) and (f). One sentence answers for (e) and (f). Code for the whole section in the Appendix (which doesn't have to be separated into parts).

Handling Instructions:

- dowmload dataset and description of total least squares from zip file.

- All your project should be implemented by MATLAB or Python.
- You are not allowed use any package for implementation of each part.
- you can use any package For implementation Additional Score part.
- About 20% of your score would be devoted to the report you deliver within your projects code. In this report you would explain the whole job and anything special you have done.
- All questions would be answered by the course's email: ala.spring2019@gmail.com.
- Place all your modules and report into a .zip named as "StudentID_FirstName_LastName.zip" before upload.
- Deadline of the project is at **23:55 pm 98/4/10** .
- In case of delivery, your code will be downloaded by the responsible TA from Moodle, so the only way to convey your code is Moodle and in if you need to reform your code please upload it when possible to be used in the due date.

Cheating Alert

1. This project should be done by individual.
2. Any similarity of more than 30% of the two projects causes the score of both sides to be zero.

Good Luck