Dimensionality Reduction

1 Singular Value Decomposition (50 points)
First we will compute the SVD of the matrix A we have loaded
[U,S,V] = svd(A)
Then take the top k components of A for values of k = 1 through k = 10 using
Uk = U(:,1:k)
Sk = S(1:k,1:k)
Vk = V(:,1:k)
Ak = Uk*Sk*Vk'

A (30 points): Compute and report the L2 norm of the difference between A and Ak for each value of k using norm(A-Ak,2)

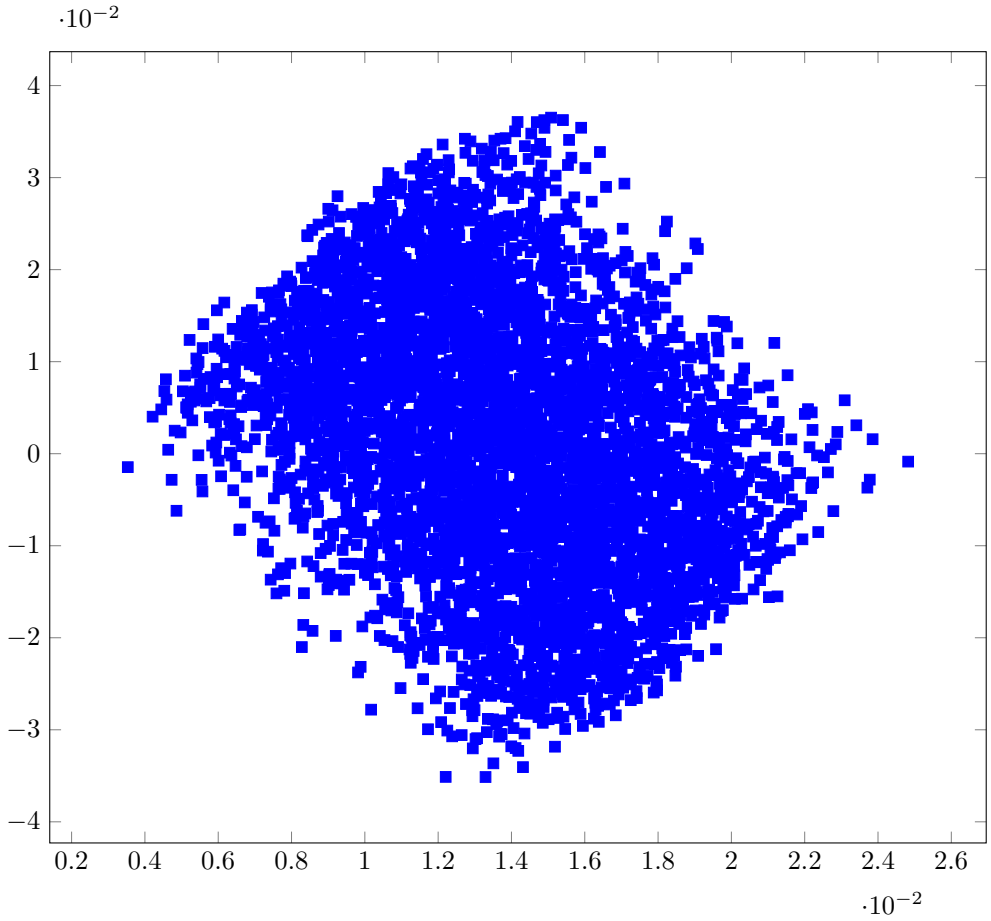
k = 1, L2 norm = 1.8626e+03
k = 2, L2 norm = 1.5257e+03
k = 3, L2 norm = 1.1719e+03
k = 4, L2 norm = 925.1212
k = 5, L2 norm = 827.8121
k = 6, L2 norm = 815.2254
k = 7, L2 norm = 639.6120
k = 8, L2 norm = 526.8578
k = 9, L2 norm = 327.0397
k =10, L2 norm = 227.2520

B (10 points): Find the smallest value k so that the L2 norm of A-Ak is less than 10% that of A; k might or might not be larger than 10.

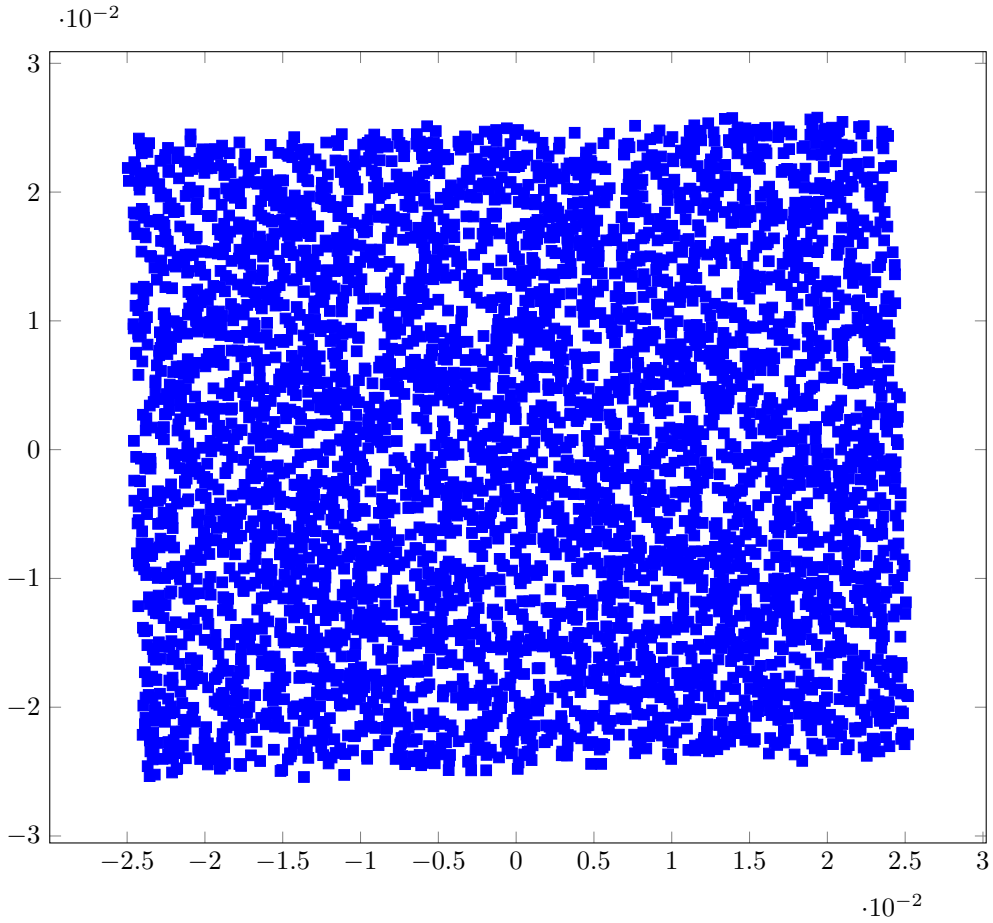
k = 7, L2 norm = 639.6120
k = 8, L2 norm = 526.8578
10% of A = 636.6475
So that means the answer is when k = 8
If you used the centered then you get k = 11, but I don't think this is what you are asking for.

C (10 points): Treat the matrix as 5000 points in 40 dimensions. Plot the points in 2 dimensions in the way that minimizes the sum of residuals squared.

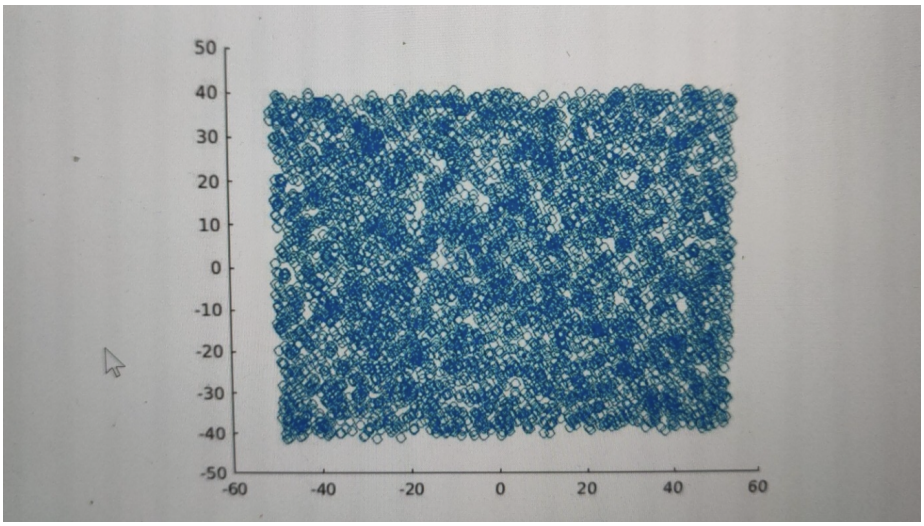
The first graphs are the [U, S, V] = svd(A), then the first two columns of U.



[U, S, V] = svd($\tilde{A}$), You need a centering matrix. You are Given A. There is a centering matrix $C_n = I_n$ - $\frac{1}{n}11^T$, where I_n is a n x n identity matrix. 11^T is is the all-ones. n x n matrix. $\tilde{A} = C_n$A. we compute [U, S, V] = svd(C_nA) = svd($\tilde{A}$). Then you take the first 2 columns of U



[U, S, V] = svd($\tilde{A}$), So let's suppose that V1 and V2 are the first two columns of V, they have length 40. And let's say we want to know the projection in 2D of the first data point A1 in $\tilde{A}$ (this is, A1 is the first row of $\tilde{A}$) using the basis V1, V2. What we do is take the dot product of A1 and V1, i.e. $< A1, V1 >$, and take the dot product of A1 and V2 i.e.$< A1, V2 >$. The 2-dimensional vector representing A1 in the new subspace will be $< A1, V1 >$ in the x-coordinate and $< A1, V2 >$ in the second coordinate. Therefore, A1, that is a 40-dimensional vector can be represented in two dimensions as the vector with coordinates ($< A1, V1 >, < A1, V2 >$).



2 Frequent Directions and Random Projections (50 points)
Use the stub file FD.m to create a function for the Frequent Directions algorithm (Algorithm 16.2.1). We will consider running this code on matrix A.

A (25 points): We can measure the error $max_{\|x\|=1}|\|Ax\|2 - \|Bx\|2|$ as norm(A'*A - B'*B, 2).

How large does l need to be for the above error to be at most $\frac{\|A\|_2^F}{10}$?

$\frac{\|A\|_2^F}{10}$ = 5.0916e+06
Error = 40532003.266658, l = 1, Nope
Error = 6499977.060382, l = 2, Nope
Error = 3746744.911215, l = 3, Yes
So at l = 3 it is finally below $\frac{\|A\|_2^F}{10}$

How does this compare to the theoretical bound (e.g. for k = 0).

$0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \frac{\|A-A_k\|_F^2}{l-k}$, k = 0
$0 \leq \|Ax\|^2 - \|Bx\|^2 \leq \frac{\|A\|_F^2}{l}$
$0 \leq 3746744.911215 \leq \frac{50915773.217866}{l}$
$0 \leq 3746744.911215 \leq \frac{50915773.217866}{3}$, l = 3
$0 \leq 3746744.911215 \leq 16971924.405955$
The result compare fairly to the theoretical bound, the statement is true

How large does l need to be for the above error to be at most $\frac{\|A-Ak\|_2^F}{10}$ (for k = 2)?
AF10 = 691449.069943
Error = 40532003.266658, l = 1, Nope

Error = 6499977.060382, l = 2, Nope
Error = 3746744.911215, l = 3, Nope
Error = 2317369.665044, l = 4, Nope
Error = 1492120.394011, l = 5, Nope
Error = 1049785.482583, l = 6, Nope
Error = 797950.797437, l = 7, Nope
Error = 517808.732668, l = 8, Yes
Error = 316702.655575, l = 9, Yes
Error = 133994.124455, l = 10, Yes
So at l = 8 it is finally below $\frac{\|A-A_k\|_2^F}{10}$

B (25 points): Create another l x d matrix B, but using random projections. You can do this by creating an l x n matrix S, and letting B = SA. Fill each entry of S by an independent normal random variable $S_{i,j} = \frac{\sqrt{n}}{\sqrt{1}}$N(0,1). Estimate how large should l be in order to achieve
$max_{\|x\|} = 1|\|Ax\|2 - \|Bx\|2| \leq \frac{\|A\|2F}{10}$ To estimate the relationship between l and the error in this randomized algorithm, you will need to run multiple trials. Be sure to describe how you used these multiple trials, and discuss how many you ran and why you thought this was enough trials to run to get a good estimate.

Out of 1000 trial I got 70 for the max and 65 for the average, I think 1000 is enough trials because it is a good amount of trials and the result does not change much no matter how many times I run it.