Ma221 Homework #7, Fall 2024, Due Wednesday, Oct 23, by 11:59pm

In this assignment we will empirically explore part of the design space of randomized low rank factorizations, and compare how well they can approximate the SVD of a matrix.

Using your favorite programming language (Matlab, Python with numpy, etc) you should
1) Implement a routine to produce a k x m sketching matrix F, with k < m, where F is a random orthogonal matrix**, multiplied by sqrt(m/k).**

2) Implement a routine to generate a random m x n matrix A = U*Sigma*V^T, with m >= n, and singular values depending on parameters t and r:
   U is a random m x n orthogonal matrix
   Sigma is an n x n diagonal matrix with specified singular values on the diagonal:
      There are t singular values equal to r^j, for j = 0, 1, ..., ceiling(n/t)-1,
      for t=10 and r = (1e-10)^(1/(n/10 - 1)). This means the largest singular value is 1
      and the smallest is 1e-10. These singular values are called true_sigma_i below.
   V is a random n x n orthogonal matrix

3) Perform the following tests:
   For n = [50, 500]
      For m = [2, 10]*n
         Generate an m x n test matrix A as described above, with t=10 and
         r = (1e-10)^(1/(n/10 - 1)). This means the largest singular value is 1
         and the smallest is 1e-10
            For k = [.1, .5]*n
               Repeat 20 times:
                  Generate a random k x m F as described above
                  Compute and save the singular values of F*A (called
                   sketched_sigma_i below)
               Make a whisker plot (boxplot in Matlab, or Python matplotlib) of
               the values of sketched_sigma_i / true_sigma_i versus true_sigma_i
               (with a horizontal log axis). Since each true_sigma_i takes the same
               value **up to t=10 times (one group may be smaller)**, there should be
               one whisker per distinct value of true_sigma_i

4) Make observations about the data you see, eg how accurately the sketched_sigma_i approximate the true_sigma_i, and how the quality of the approximation depends on n, m and k.

5) Extra credit: Try other singular value distributions, values of m, n and k, and sketching matrices.