Learning Opportunity 2-3

2-3-1 Write a Python function that takes in a Sage matrix over the rationals and prints out the successive rank 1, rank 2, rank 3,... SVD approximations as Sage matrices over `RDF`. You may use a built-in Sage function to get the initial SVD decomposition. Do not use the PyLab code from the image compression demonstration.

Full marks for a complete Python function (not just a Sage cell with code) and a successful test run of your function on a random 4×6 matrix of rank 3. (So include this test run in what you submit.)

2-3-2 This exercise will have you determine the column space and null space of a matrix from the SVD decomposition.

1. Construct a random 6×10 matrix, A, over the integers with rank 4 and call it A. Use the `algorithm='subspaces'` keyword in `random_matrix()` so you get something manageable.
2. Construct the column space (`.column_space()`) and null space (`.right_kernel()`) for A. Recognize that these are subspaces. For each, Sage will compute a basis, which can be retrieved as a matrix of row vectors with `.basis_matrix()` applied to the subspace. Obtain the basis matrix for each vector space.
3. Now convert A to a matrix B over `RDF`. Compute the SVD of B. Examine S to see how you would determine the rank of A from just this information.
4. Take the correct columns of U to form a basis of the column space of A and the correct columns of V to form a basis of the null space of A. The `.column()` method might be helpful.
5. Use your bases to build the corresponding subspaces. For example if `null_basis` is my set of columns from V, then `(RDF^z).subspace(null_basis)` using the correct integer as `z` would accomplish this.
6. Find the basis matrix for each of your new vector spaces over `RDF` and verify that each is the same as, or very similar to, the basis matrix for the previous exact version over `QQ`.