ST 453 Advanced computing for statistical reasoning
Homework problem set 3

February 3, 2026

The `R` package "jpeg" is permitted for use in this assignment.

1. In Homework 2 you were asked to propose a statistical model with population features that may be appropriate to estimate from the real data set you have chosen for your class project. Given the statistical model or models that you proposed, the next step is to determine how to fit the proposed model(s). What estimation procedures and algorithms could you use to fit your proposed model(s)? For example, common options are least squares estimates, maximum likelihood estimates, method of moments estimates. Typical algorithms for likelihood-based estimation methods are gradient decent algorithms. You do not need to fully specify an estimation procedure or algorithm, just search for resources to learn more about your proposed model(s) and list appropriate estimation procedures and/or algorithms with a brief explanation of why they are appropriate.

2. Write an `R` function that takes as input `( v, U)`, where `v` is an n-dimensional column vector and `U` is an $n \times p$ dimensional matrix. Your function should do the following:

   (a) Determine if the columns of `U` are linearly independent. If the columns of `U` are not linearly independent, then your function should determine a basis for the column space of `U`. (Hint: in the latter case, use the fact that $\mathrm{col}(U) = \mathrm{col}(UU')$ for any matrix $U \in \mathbb{R}^{n\times p}$, where $\mathrm{col}(\cdot)$ denotes the column space of a matrix argument).

   (b) Using the basis for the column space of `U` constructed in part (a), determine if `v` is in the column space. If so, determine the coefficient vector `a` such that `v` can be expressed as a linear combination of the basis vectors for the column space of `U`.

   Your function should return the basis returned in part (a) along with the coefficient vector `a` in part (b), if it exists. If `a` does not exist, then return a warning message in place of `a`. In any case, you will always be able to return the basis from part (a).

3. Obtain a .jpg file of an image. You can use a personal .jpg file, or download a .jpg file from the internet – keep it professionally appropriate.

   (a) In your homework script file, load your .jpg file, and plot the original image, to scale, in a base `R` plot.

   (b) Change the pixel colors in your image file by adding a small perturbation to each RGB value, and plot the new image, to scale, in a base `R` plot.

   (c) Plot the original image in grayscale, to scale, in a base `R` plot.

   (d) Obtain the U, D, and V matrices in the SVD of your grayscale image. Verify that the grayscale image matrix can be reconstructed from its SVD.

   (e) Plot the ordered singular values as points in a base `R` plot, and note the index k for which the ordered singular values become "close" to zero. In what sense is "close" to zero meaningfully quantified?

   (f) Reconstruct your grayscale image matrix using a *low-rank* (or *dimension reduced*) approximation starting with only the two largest singular values, and successively adding singular values one at a time to improve the approximation. At each iteration, plot the image, to scale, in a base `R` plot, and save the plot as a new page in a .pdf file. See the lecture code for an example of how to do this.

   (g) In the successive low-rank matrix approximations of your grayscale image, at what index, k, would you say adding an additional singular value to the approximation has a negligible visual effect on the image quality? Using the dimension reduced SVD with only k singular values, calculate the memory efficiency gained versus storing the full grayscale matrix (i.e., how many fewer pieces of information do you need to store in your computer memory using the low-rank approximation)?