Due: Sunday, March 17, by 11:59pm (hard deadline, solutions to be posted)

1. Find the 'HouseVotes84' data set from the 'mlbench' library. It includes voting records from 1984 for the US House of Representatives on 16 bills. Note that in some cases I'm going to give you broad goals. I expect your code to work without errors or warnings, in many cases additional packages (that may not have been covered in any of your modules) might be useful. Use your web searching skills!

   (a) Cleaning: There are many NA values, replace these with the character string "NoVote". Run 'head()' on your data to show that this has been done.

   (b) Distance: Compute all pairwise distances with Gower distance for the observations using all variables except the party affiliation (which is a natural response variable).

   (c) Hierarchical: Perform hierarchical clustering on the pairwise distance matrix. Explore linkages and select which one you believe provides the best argument for groups in the data. How many are suggested? Show the dendrogram. Cut the tree at the suggested number of groups and provide a classification table vs party affiliation.

   (d) MDS: Run classical multidimensional scaling on the pairwise distance matrix for a 2D mapping. Show a scatterplot with the points coloured by party affiliation.

   (e) Kmeans: Perform kmeans clustering on the 2D mapping for k=2. Provide a classification table vs party affiliation. Show a scatterplot with the points coloured by group membership.

   (f) Mixture models: Perform MCLUST on the 2D mapping under default settings. How many groups are suggested by the BIC? Provide a classification table vs party affiliation. Show a scatterplot with the points coloured by group membership.

2. The covariance matrix "ability.cov" is available in base R. The original variables measured are a 'general' test of intelligence, a 'picture' completion test, a 'block' design test, a 'maze' test, a 'reading' comprehension test, and a 'vocabulary' test. These scores were recorded on 112 individuals, and the resulting covariance matrix estimate is the matrix we now have. Note: including number of observations is essential for the goodness of fit tests (a fact I briefly mentioned in lecture).

   (a) Run factor analysis with one factor on the covariance matrix with no rotation. Does the output suggest that this is a suitable model? Explain.

   (b) Run factor analysis with one factor on the correlation matrix (hint: cov2cor) with no rotation. Are there any differences in the model to the previous question?

   (c) Run factor analysis with two factors on the covariance matrix with no rotation. Does the output suggest that this is a suitable model? Explain. Provide an interpretation of the two factors.

   (d) Run factor analysis with two factors on the covariance matrix with 'varimax' rotation. Which elements of the output have not changed? Provide an interpretation of the two factors. Is this easier to interpret?

(e) Run factor analysis with two factors on the covariance matrix with 'promax' rotation. What element is added to the output? What assumption have we relaxed that necessitates that output? Provide an interpretation of the two factors. Is this easier to interpret?

3. The MNIST database is a famous benchmarking data set of handwritten digits. However, let's play with a similarly structured set of data on clothing items! The main source of the data can be found at `https://github.com/zalandoresearch/fashion-mnist`. Download the "t10k-images-idx3-ubyte" and "t10k-labels-idx1-ubyte" files from the data folder therein and unarchive them. While these are actually the test sets used for benchmarking, they are smaller in n than the training sets, so that is why we will play around with them instead. You will have to figure out how to get this data into readable form in R, by exploring the github page, googling, etc. Note that it is structured VERY similarly to the original MNIST data, so there are many sources available to learn from.

(a) Provide a plot of the first 25 images in the data. Ensure they are oriented properly!

(b) Run principal components (withOUT scaling) on the images (should run for a minute or two). I suggest saving this object in an Rdata file so that it doesn't run every time you knit your Rmd file (you can use eval=FALSE in your code chunk to show the TA you've used the correct commands). What is the maximum number of components are permittable?

(c) Plot the first 25 resulting eigenvectors as images. What percentage of the original variation in the pixels is explained by the first 25 PCs?

(d) Reconstruct approximations of the original observations using 25 PCs. Plot side-by-sides for the first 10 digits of the reconstructions and originals in a 5x4 matrix of images.

(e) You are welcome to run our homemade NMF function on the data — if you do, add 0.01 to every value to avoid lots of dividing by 0 — or find an NMF package to fit to it: use 25 basis vectors. I ran our function for 600 iterations and am unsure how long it took (I walked away for about 10 minutes). In case it's on the upper-end of that, I have posted the resulting object on github as "fnmfres.Rdata" in the assignments area. Provide a plot of the 25 basis vectors as images. Comment on the differences between these and the eigenvectors from part (b).

(f) Reconstruct approximations of the original observations using 25 NMF bases. Plot side-by-sides for the first 10 digits of the reconstructions and originals in a 5x4 matrix of images.

(g) Fit a classification tree with labels as the response variable and the NMF 'scores' as the predictors. Plot the tree.

(h) Use 'cv.tree' with 'prune.misclass' as the function, how many nodes are suggested to be removed? What is the cross-validated misclassification rate of the best tree?