Assignment #3

MSAN 593

***DUE**: Saturday, August 6, 23:45*

Be sure to upload a single `*.R` script file named `asn3_q1.R` to Canvas by the due date and time. No write-up is required. Late submissions will receive a grade of zero. Your `*.R` files will be run on local machines by graders. If your file does not run, you will automatically lose 30% of the grade. If you resubmit your corrected homework by the last class of the module, it will be graded out of the remaining 70%. Failure to resubmit will result in a grade of zero. Assume that the data file(s) being read into your `*.R` files are in the current local directory, e.g., `read.csv('myFile.csv')` will work. Do **not** hard code a specific directory structure.

Question 1

Create a function which will take four arguments:

- `nReps`: a non-negative integer value
  - `nReps` $\in \mathbb{Z}_{>0}$
- `myScatterInput`: a data frame with `n` rows and `m` columns, where all entries will be real numbers
  - `myScatterInput` $\in \mathbb{R}^m$
- `myClusterNum`: an integer value greater than or equal to 2 but less than or equal to n
  - $2 <$ `myClusterNum` $< n$: `myClusterNum` $\in \mathbb{Z}$ *`maxIter`: a non-negative integer value
  - `maxIter` $\in \mathbb{Z}_{>0}$

Your function should:

1. Randomly choose `myClusterNum` points from the `myScatterInput` data frame. These are the *cluster centers*.
2. Compute the Euclidean distance from each *cluster center* to **each data point**.
3. Assign each point to the *cluster center* where the Euclidean distance is minimized.
4. Compute the centroid of each cluster. These are your new *cluster centers*.
5. Repeat steps 2, 3 and 4 until one of two stopping conditions are met
   - subsequent cluster assignments are unchanged
   - you have repeated steps 2, 3 and 4 `maxIter` number of times
6. Once you have reached a terminating condition, compute the sum of all Euclidean distances from each point to their respective centroids.
7. Repeat steps 1-6 `nReps` number of times.
8. Identify the replication with the lowest sum of Euclidean distances from points to centroids as your best result and print the value to the console.
9. **IF** the data frame provided to you has `m=2` or `m=3`, generate a 2- or 3-dimensional graph, respectively, plotting all points and coloring each set of points based on the cluster they are in, i.e., all points associated with a certain cluster should all be the same color in the scatter plot.

n.b.

- Be sure to `set.seed()` in the following fashion: for the first run, prior to selecting the `myClusterNum` centers, `set.seed(1001)`, and for each subsequent iteration, increment the seed by one, e.g., on the third run, prior to selecting `myClusterNum` centers, the seed should be set to `set.seed(1003)`
- Your `R` script should include code which reads in a `csv` file named `hw3data.csv` **which does NOT have headers**. This is the file which will provide the data for the n by m data frame of points, i.e., `myScatterInput`.
- Your `R` script should include code which reads in a `csv` file named `hw3params.csv` **which does NOT have headers**. This file is a vector of three values which will provide values for (in order): `nReps`, `myClusterNum` and `maxIter`.
- The homework will be graded by `source()`-ing the file, which should read in the data as well as the parmeters, run as described, and if conditions are met, produce the aforementioned plot. **This means that the last line of your code should run your function with the appropriate parameters.**
- If done properly, this code should run almost instantaneously for `nReps = 100`.