Assigned: 02/04/2021
Due: Wed. 02/17/2021, 11:59pm

Instructions: This project will cover topics of data preprocessing and clustering.

Submission Requirements:

This assignment is split into two main parts.

Part A: You will be asked to walk through methods **manually**, that is not using the available functions to perform clustering analysis. You can use R, Matlab, or Python for plotting of the data when asked and for support in calculations (compute distances, averages, etc.). But, the process should be done manually with your answer presented in **typed form**.

You may then prepare your solution to Part A as a separate document using Google Docs, Word, LaTeX, with computer generated plots inserted. Note, if you prepare your solutions in Rmd, Sweave, or Python notebooks, your solution to Part A can also be included in that document using Markdown or LaTeX to enter your answers (not a separate document).

Part B: You will again have a final document submission that includes text responses to questions, tables, R/Matlab/Python code used to calculate answers, and figures.

Formatting of submissions: The following methods are acceptable ways to submit your assignment:

- If you are using MATLAB consider:
  - .m file + markup, publishing matlab code → PDF
    Incorporate your answers directly into your MATLAB code (code, comments, results), publish the code creating an PDF file.
  - .mlx files + LiveScript Editor → PDF
    Answer your questions in the Matlab LiveScript editor, embedding code and results from matlab .m file
- If you are using R consider:
  - Rmd → PDF
    Use `knitr` or `rmarkdown` to collect all text responses, figures, tables, and code in the R Markdown file and process it to produce a PDF file.
  - Snw → PDF
    Use `R Sweave` to collect all text responses, figures, tables, and code in the Snw file and process it to produce a PDF.
- If you are using Python consider:
  - Jupyter notebook (.ipynb) → PDF
    Incorporate all text responses, figures, tables, and code in the jupyter notebook and process it to produce a PDF file.
  - Colab notebook (.ipynb) → PDF
    Incorporate all text responses, figures, tables, and code in the jupyter notebook and process it to produce a PDF file.

Any other packages or tools, outside those listed in the assignments, for reproducible research should be cleared by Dr. Brown before use in your submission.

For this project you are allowed to work in **groups of up to 3**. All students must sign-up on Canvas into a group (e.g., P2-Rho, P2-Tau, etc.).

Name your main submission files as *P2_GroupName*, create a zip-file called *Project2_GroupName.zip* and submit on Canvas. For example, if I was using R, with a separte response for Part A, and member of GroupChi, I would submit either:

- *P2_PartA_GroupChi.pdf*, *P2_PartB_GroupChi.Rmd*, *P2_PartB_GroupChi.pdf*, or
- *P2_PartA_GroupChi.pdf*, *P2_PartB_GroupChi.Snw*, *P2_PartB_GroupChi.pdf*

Questions:

Part A

1. (1 point) List your project GroupName and all group members names.

2. K-means Clustering
   Perform k-means clustering **manually** with k=2 on the example data given below of $n = 8$ samples over $p = 2$ features.

Sample	X_1	X_2	Initial Groups
1	0	4	2
2	1	3	1
3	1	4	1
4	2	5	2
5	4	0	1
6	5	1	1
7	5	2	2
8	6	1	2

   (a) (2 points) Plot the sample data.

   (b) (4 points) Assign samples to be the initial groupings given in the table. Compute and report the centroid for each cluster.

   (c) (4 points) Assign each sample to the centroid to which it is closest (Euclidean distance). Report the cluster labels for each observation.

   (d) (20 points) Repeat (b) and (c) until the clusters remain stable.
   To make this process easier to grade, I have supplied a template of a table which shows how the information from Q2(b) - (d) can be presented.

   (e) (2 points) Plot the sample data colored by cluster labeling and adding centroid points.

3. Hierarchical Clustering
   Suppose you have 5 samples (1, 2, 3, 4, 5), for which the dissimilarity matrix is shown below:

$$D = \begin{bmatrix} -- & 0.3 & 0.4 & 0.7 & 0.6 \\ 0.3 & -- & 0.5 & 0.8 & 0.2 \\ 0.4 & 0.5 & -- & 0.45 & 0.4 \\ 0.7 & 0.8 & 0.45 & -- & 0.35 \\ 0.6 & 0.2 & 0.4 & 0.35 & -- \end{bmatrix}$$

   That is, the distance between sample 1 and 2 is 0.3; the distance between sample 1 and 4 is 0.7.

(a) (12 points) Trace running through hierarchical clustering **manually** with complete linkage and sketch the dendrogram. Estimate the heights in the dendrogram from the dissimilarity distances.

For each step, say what are the next two items (samples of clusters) to be combined and report the new distance matrix (have samples groups ordered alphabetically). You only need to report the lower triangular part of the distance matrix.

Step 1. Combine ______________ and ______________.
Report new distance matrix between groups, ordered numerically (e.g., if just combined 2 and 4 - then distance matrix should be ordered - 1, 24, 3, 5).

	1	24	3	5
1	0	-	-	-
24	?	0	-	-
3	?	?	0	-
5	?	?	?	0

Step 2. Combine ______________ and ______________.
. . .

(b) (12 points) Repeat (a), with single linkage clustering

(c) (6 points) Use the dendrogram from (a) and (b), cut the dendrograms to form three clusters. Which samples are in each cluster?

Part B:

4. (9 points) Report the normalized values for the following vector of data:

$$20, 30, 40, 60, 120$$

using the following methods:

- min-max normalization with *min=0* and *max=1*
- min-max normalization with *min=-1* and *max=1*
- z-score normalization

Helpful functions: R - `scale` and `preProcess, predict` from `caret` library, Matlab - `normalize`, Python - `MinMaxScaler` and `StandardScaler` or `scale` in `sklearn.preprocessing` library.

5. Consider the following data set of with 5 samples and 3 variables:

	A	B	C
x_1	1.4	1.3	2.9
x_2	1.8	1.1	3.2
x_3	1.3	1.2	2.9
x_4	0.9	3.3	3.1
x_5	1.5	2.1	3.3

You have a new data point $x = (1.25, 1.74, 3.01)$.

(a) (5 points) Calculate and present the distance between the new data point and each of the points in the data set using Manhattan distance, Euclidean distance, Minkowski distance ($\lambda = 3$), supremum distance, and cosine similarity.

(b) (5 points) Normalize the data using min-max normalization to be between 0 and 1. What is the Euclidean distance between the new data point and $x_1, \ldots, x_5$.

6. Pokemon Data
Consider methods to group Pokemons based on various statistics of skills [1].
   (a) (6 points) After loading in the data, look at the distribution of Pokemon features we will use for clustering: `HP`, `Attack`, `Defense`, `SpAtk`, `SpDef`, and `Speed`. For example, a boxplot for each feature in one figure.
   Helpful functions: R - `melt` from `reshape2` library with `ggplot2`
   (b) (4 points) The features have different ranges, therefore we should scale the data before considering the clustering analysis. Scale the data using min-max normalization with range of $[0, 1]$.
   Helpful functions: R - `preProcess, predict` from `caret` library, Matlab - `normalize`, Python - `MinMaxScaler` from `sklearn.preprocessing`.
   (c) (16 points) Run Kmeans clustering on the data of (b) with $k = [2, 3, \ldots, 8]$.
   (d) (4 points) Determine the "best" number of clusters using gap statistic.
   Helpful functions: R - `clusGap` in `cluster` library use the "globalSEmax" method with 100 bootstraps, Matlab - `evalclusters`, Python - `gapstat.py` available on Canvas[2].
   (e) (6 points) Report the mean skill values (centers) of each group, best number of groups determined in (d), as a table/data frame.
   (f) (2 points (bonus)) Report the mean skill values (center) of each group, best number of groups determined in (d), as a table/data frame using original data scaling (reverse the scaling back to the original data range).
   (g) (8 points) Create a single figure with a radar plot showing the mean skill values (center) for each cluster group, set from (d).
   Helpful functions: R - `radarchart` using `fmsb` library, Matlab - `spider_plot`[3], Python - `polar` plots in `matplotlib` or `line_polar` plots in `plotly.express`.

7. (24 points) Music Data
For this problem you will consider several properties that have been measured from music recordings.[4]

   Consider only the numeric variables from the data: `music2.csv`.

   First, standardize the variables.

   Then, perform hierarchical clustering two times, with single and complete linkage. Label the clusters by the 'Type' of music.

   Repeat the analysis as above, but label the samples by the musical 'Artist'.

   Which method seems best? Explain why.

   Helpful Functions: R - `hclust` in `cluster` library, Matlab - `linkage, cluster, dendrogram`, Python - `linkage` in `scipy.cluster.hierarchy`, `dendrogram` in `scipy.cluster.hierarchy`, and `AgglomerativeClustering` in `sklearn.cluster`.

[1] The data has been collected from several sources: pokemon.com, pokemondb, bulbapedia

[2] File from `https://github.com/jmmaloney3/gapstat`

[3] Available in Matlab File Exchange: `https://www.mathworks.com/matlabcentral/fileexchange/59561-spider_plot`

[4] The original music data: `http://www.public.iastate.edu/~dicook/stat503/music-plusnew-sub-full.csv`