Assigned: 03/24/2022
Due: Sun. 04/03/2022, 11:59pm

Instructions: This project will cover topics of dimensionality reduction and text mining.

Name your main submission files as *A4_Group_<GroupName>.extension*. You will be submitting your "code" file and a PDF of the code + text + results of your code running.

The following are acceptable formats:

Language	"Code" File Format	PDF Generation
R	R Markdown, .Rmd	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.
	R Sweave, .Snw	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.
Python	Jupyter notebook, .ipynb Colab notebook, .ipynb	.ipynb → PDF: Incorporate all text responses, figures, tables, and code in the notebook and process it to produce a PDF file.

Submission Requirements: Your answers must be computer generated (including text and diagrams). Your final document submission should include text responses to questions and description of your efforts, tables, R/Python code used to calculate answers, and figures.

Create a zip-file called A4_Group_<*GroupName*>.zip and submit on Canvas. Your group name, <*GroupName*> will be given on Canvas.
For example, if I was using R, I would submit either:

- A4_Group_<*GroupName*>.Rmd, A4_Group_<*GroupName*>.pdf, or
- A4_Group_<*GroupName*>.Snw, A4_Group_<*GroupName*>.pdf

For Python, I would submit:

- A4_Group_<*GroupName*>.ipynb, A4_Group_<*GroupName*>.pdf

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

Questions:

1. NBA Data

   Consider methods to cluster NBA players based on their statistics.

   (a) (8 points) Load the data of NBA players in the 2018-2019 season. First, filter the players to only consider those who have played in more than 20 games. The analysis will ignore the first 7 columns as well as ignore the columns of statistics of percentages (FG%, 3P%, 2P%, eFG%, FT%).

   (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 or `scale` function, Python - `MinMaxScaler` from `sklearn.preprocessing`.

(c) (24 points) Run Kmeans clustering on the data with k=2, ..., 10. For each value of k, keep track of the within-cluster variation. This quantity is referred to as different terms such as "inertia" and total "within-cluster sum-of-squares".

Helpful functions: R - `kmeans` from the base `stats` library,
Python - `KMeans` from `sklearn.cluster`.

Plot the within-cluster variation vs. the values of k.

As discussed this value is always decreasing, therefore it is difficult to use for selecting the *best* value of k.

(d) (12 points) Determine the "best" number of clusters using gap statistic.

Helpful functions: R - `clusGap` in `cluster` library use the "globalSEmax" method with 100 bootstraps; Python - `gapstat.py` available on Canvas[1] with B1=50.

(e) (8 points) Create a data frame with the mean skill values (centers) of each group, using the best number of groups determined in (d), as a table or data frame. You may want to run Kmeans again with the best value of k.

Print out the statistics for each group (rows) and the columns of 'MP', 'FG', '3P', and 'FT'.

(f) (4 points (bonus)) Report the same statistics as in (e), but using the original data scaling (reverse the scaling back to the original data range).

(g) (10 points) Apply PCA to the filtered nba data (make sure to apply the necessary scaling for running PCA). Plot the data in the first two principal components, colored by the best group labels found in (e).

2. (54 points) Music Data

For this problem you will consider several properties that have been measured from music recordings.[2]

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

First, standardize the variables.

Then, perform hierarchical clustering with single, complete, and average linkage.

Report out the results in a dendrogram. Label the clusters by the 'Type' of music.

Report out the results in a dendrogram, but label the samples by the musical 'Artist'.

Which method seems best? Explain why.

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

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

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