Problem_set_2_Kim

Daejin Kim

10/19/2019

Problem Set 2

Remember to submit a single rendered PDF or HTML file (either from .Rmd or a Jupyter Notebook) via GitHub by ***Saturday at 12 noon.***

Loading packages

```
library(tidyverse)
library(skimr)
#install.packages("dendextend")
#install.packages("ape")
library(dendextend)
library(ape)
library(tidyverse)
library(skimr)
```

Computation

You fielded a survey and collected some wildly descriptive feature vectors. Use the following vectors to address questions 1-3:

```
p{1, 2}
q{3, 4}
```

1. Calculate Manhattan, Canberra, and Euclidean distances "by hand" (i.e., create the data, program each line, and make the calculations). What are the values for each measure?

```
#p{1,2}=p{a,b}
#q{3,4}=p{c,d}
a<-1
b<-2
c<-3
d<-4
Manhattan<-abs(a-c)+abs(b-d)
Canberra<-abs(a-c)/(abs(a)+abs(c)) +abs(b-d)/(abs(b)+abs(d))
Euclidean<-sqrt((a-c)^2+(b-d)^2)
```

Manhattan distance = 4
Canberra distance = 0.8333333
Euclidean distance = 2.8284271

2. Use the `dist()` function in R to check your work. Were you right or wrong? (be honest in your reporting). If wrong, after debugging, where and why did you go wrong?

```
A=matrix(1:4, nrow=2, byrow=TRUE)
MD<-dist(A, method = "manhattan", diag = FALSE, upper = FALSE, p = 2)
CD<-dist(A, method = "canberra", diag = FALSE, upper = FALSE, p = 2)
ED<-dist(A, method = "euclidean", diag = FALSE, upper = FALSE, p = 2)

#Checking if these values are the same as the ones calculated by hand.
Manhattan==MD

## [1] TRUE

Canberra==CD

## [1] TRUE

Euclidean==ED

## [1] TRUE
```

Using the `dist()` function, I get the same answers as the ones calculated by hand.

3. What are the key differences between these measures, and why does it matter? How might you see these differences "in action" with these fictitious data?

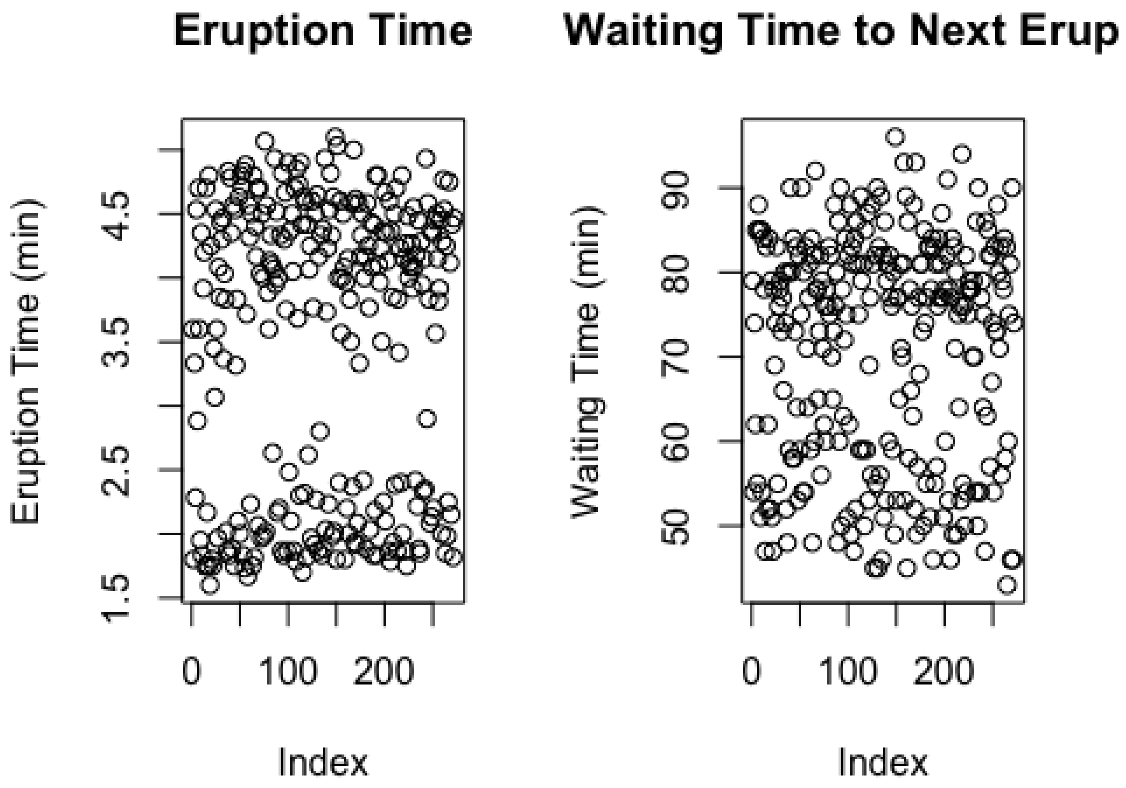
The Euclidean measure provides the closest distance without excluding any element. Therefore, using Euclidean distance may be the most intuitive. However, in a grid-like setting, the Manhattan distance may be more suitable because it has more direct information on how many blocks there are between two objects. However, the Manhattan distance is always larger than the actual distance, which will add up fast. The Canberra distance may resolve this problem by dividing Manhattan distance by the absolute value of each side.

Download the old faithful data set and use to address questions 4-6:

```
fdta<-data.frame(faithful)
```

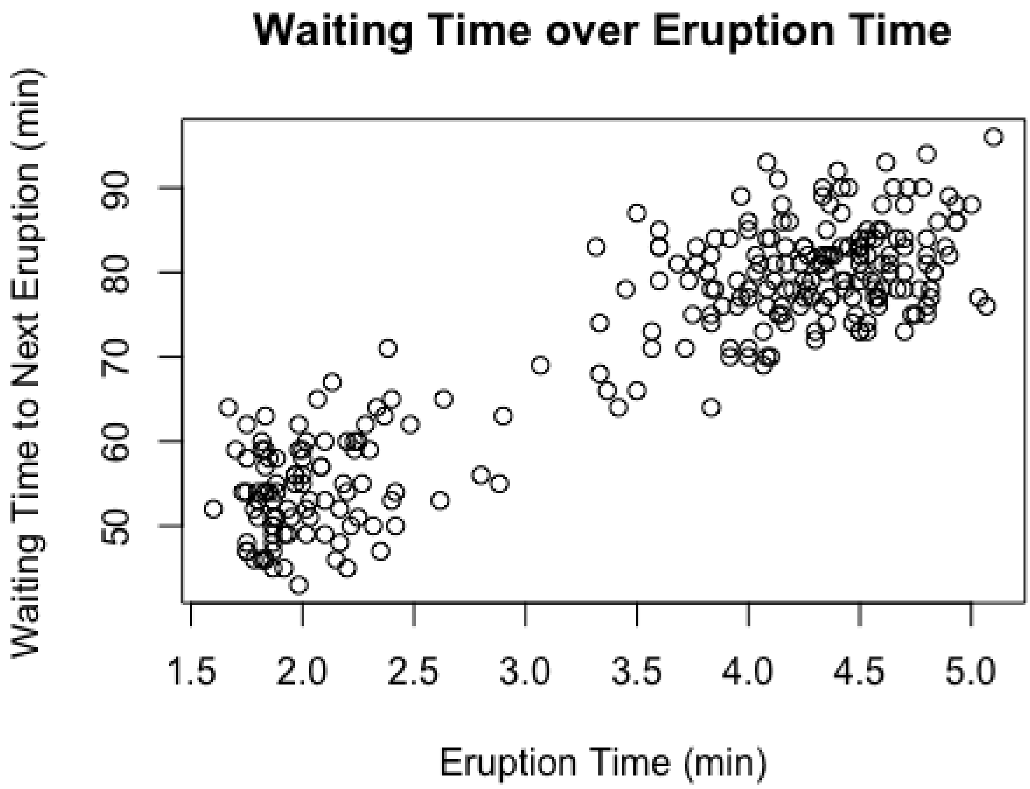
4. Use some basic EDA techniques to present and discuss the data (e.g., visualize, describe in multiple ways, etc.)

```
par(mfrow = c(1,2))
plot(fdta$eruptions, main="Eruption Time",
       ylab = "Eruption Time (min)")

plot(fdta$waiting, main="Waiting Time to Next Eruption",
       ylab = "Waiting Time (min)")
```



These graphs demonstrate that both eruption time and waiting time may have two clusters. The left graph has the most dots around 4.5 minutes and 1.8 minutes. Not as distinct as the eruption time, waiting time also shows two possible clusters around 80 minutes and 50 minutes.

```
par(mfrow=c(1,1))
plot(fdta, main="Waiting Time over Eruption Time",
     xlab = "Eruption Time (min)",
     ylab = "Waiting Time to Next Eruption (min)")
```



This graph shows the relationship between waiting time and eruption time by putting them together in the two-dimensional space. This graph shows a clear clustering of the data into two groups.

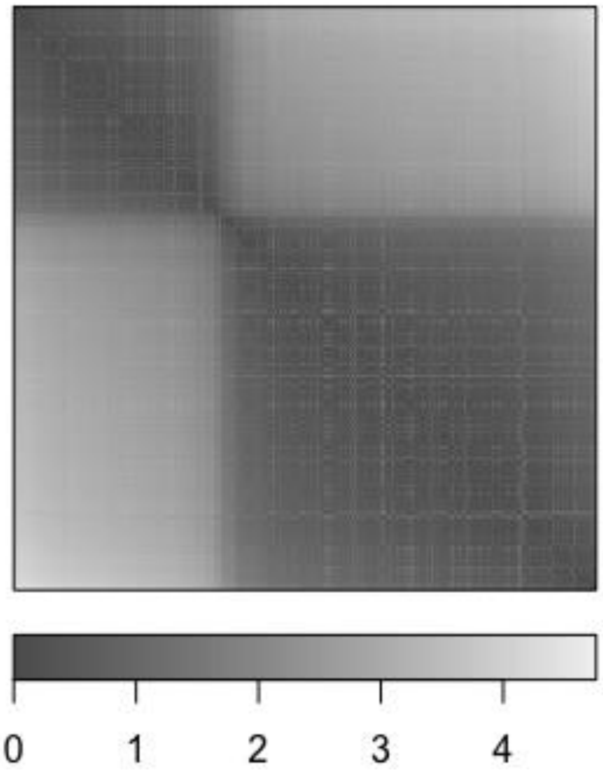
5. Calculate a dissimilarity matrix of these data.

```
#library(cluster)
#daisy(fdta, metric ="euclidean",
#      stand = TRUE, type = list())

f_sub <- fdta %>%
  scale() %>%
  dist()
```

6. Generate an ODI for the Old Faithful data. What do you see?

```
#install.packages("seriation")
library(seriation)

## Registered S3 method overwritten by 'seriation':
##   method         from 
##   reorder.hclust gclus
```

```
fdta_scaled <- scale(fdta)
fdta_dist <- dist(fdta_scaled,
                method = "euclidean")
dissplot(fdta_dist)
```



The ODI presents two dark squares aligned diagonally implying that there are two clusters. From this visualization, one can conclude that the Old Faithful data may be appropriate for clustering analysis.

Download the Iris data set we used in class (e.g., `data(iris)` in R), and use to address questions 7-10:

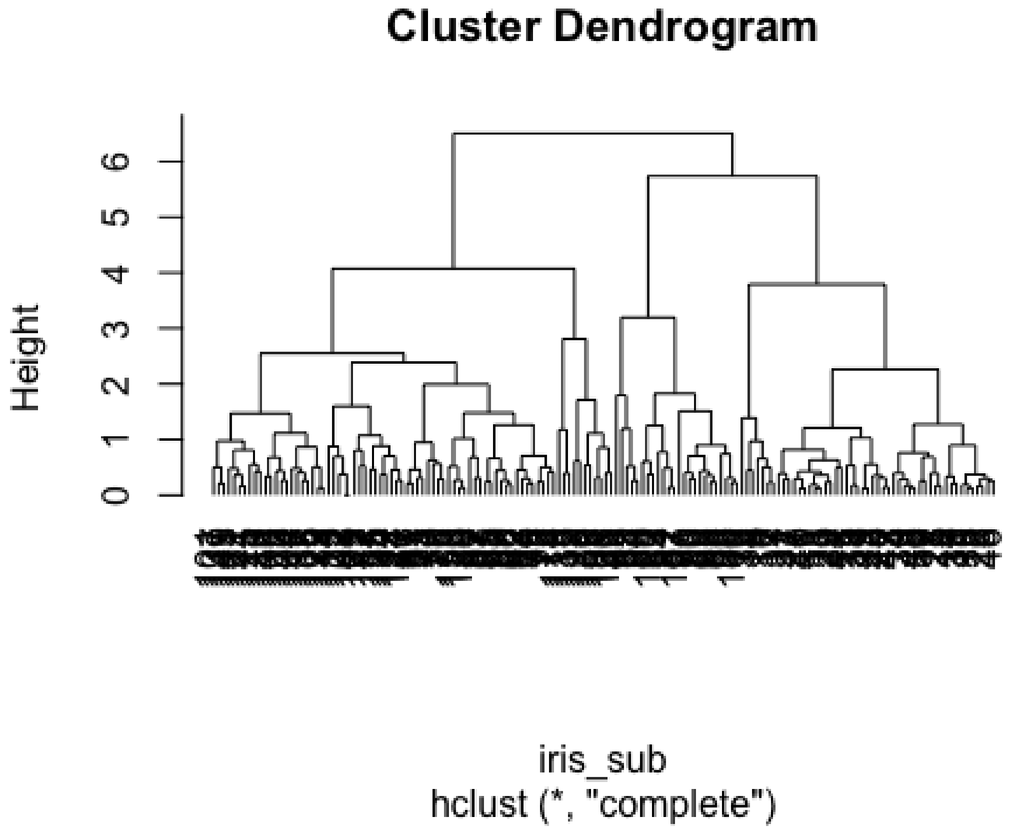
```
data(iris)
```

7. Using any munging tools you'd like (e.g., dplyr from the Tidyverse), create a subset of the data excluding the species feature, scaling the features, and calculating a dissimilarity matrix (think `%>%` for stacking functions to do this quickly, e.g.).

```
library(dplyr)
iris_sub <- iris %>%
  select(-c(Species)) %>%
  scale() %>%
  dist()
```

8. Fit an agglomerative hierarchical clustering algorithm using complete linkage on your subset data and render the dendrogram of clustering results. What do you see?

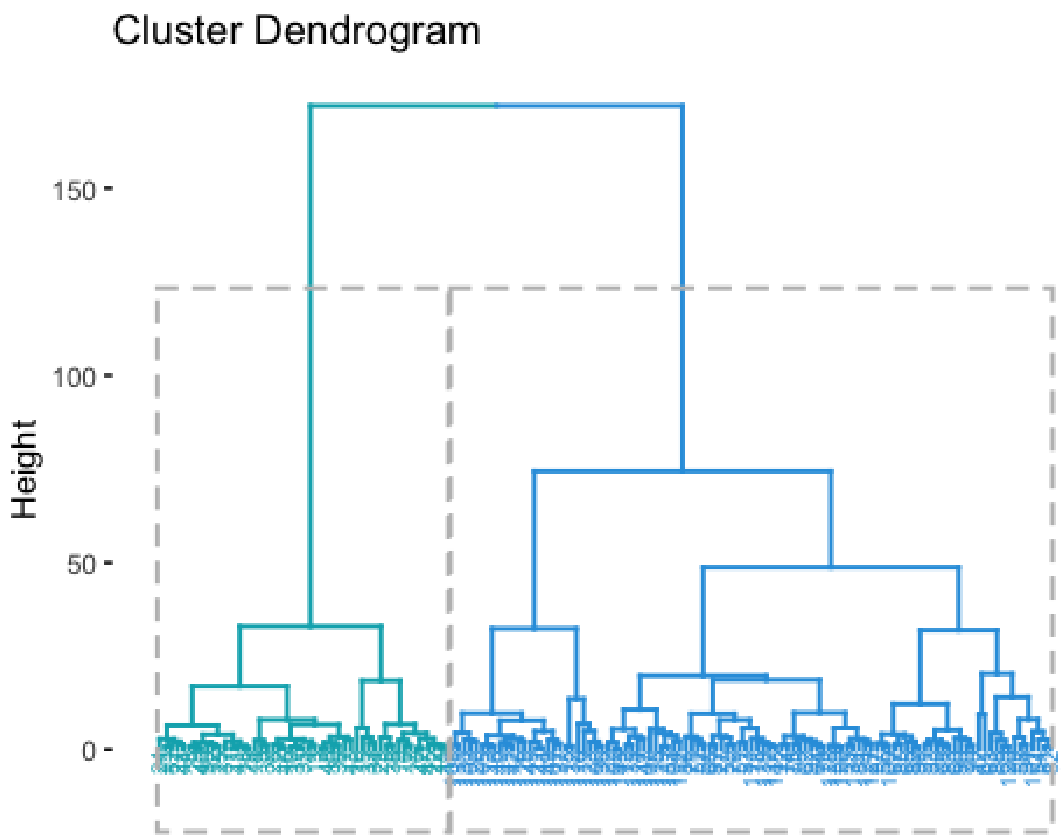
```
hc_complete <- hclust(iris_sub,
                      method = "complete"); plot(hc_complete, hang = -1)
```



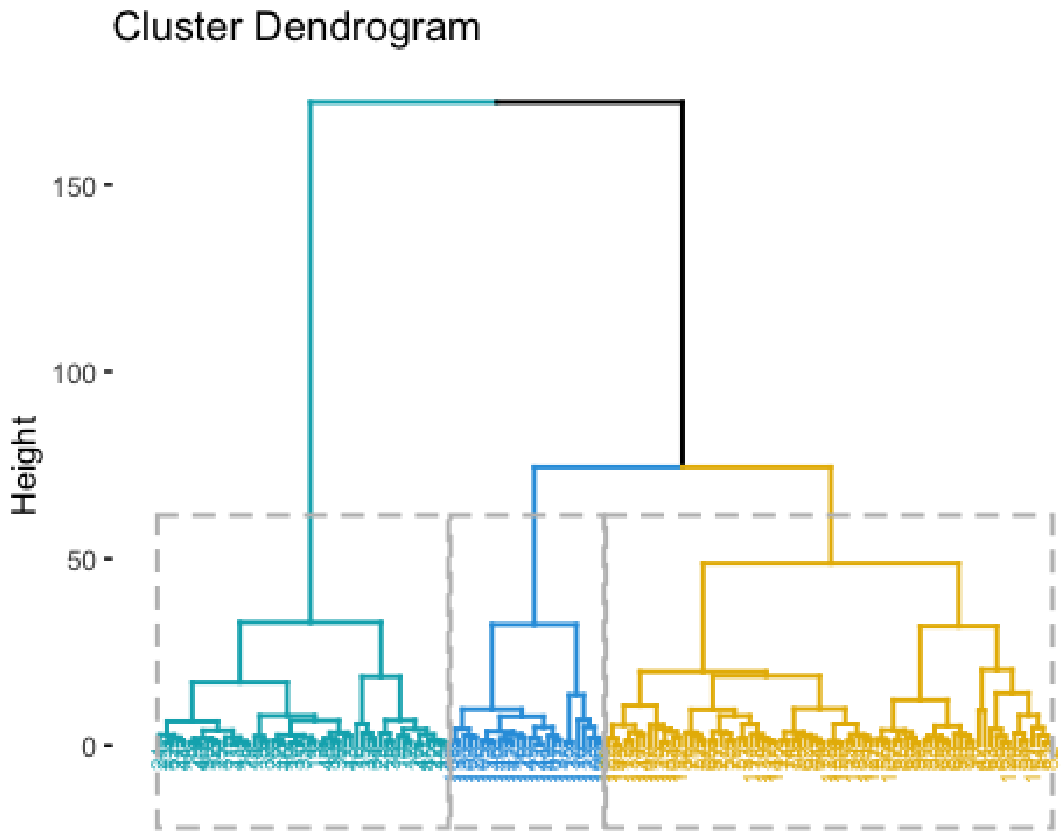
The dendrogram starts with pairs at the bottom and always increases in the number of connections as it moves upward. This characteristic results in hierarchical clustering, which means that upper clustering always includes the lower level clustering. Moreover, the dendrogram shows potential clustering. If I draw a horizontal line at the height of 4, it passes through four branches, cutting the dendrogram into four subtrees.

9. Try cutting the tree at 2 and 3 branches and show these trees side-by-side. How do they differ?

```
#install.packages("factoextra")
library("factoextra")

## Welcome! Related Books: `Practical Guide To Cluster Analysis in R` at
https://goo.gl/13EFCZ

cut_2 <- hcut(iris_sub, k = 2, stand = TRUE) #cutting iris_sub into two
clusters
fviz_dend(cut_2, rect = TRUE, cex = 0.5,     #Demonstrating a dendrogram of
cut_2
          k_colors = c("#00AFBB","#2E9FDF"))
```



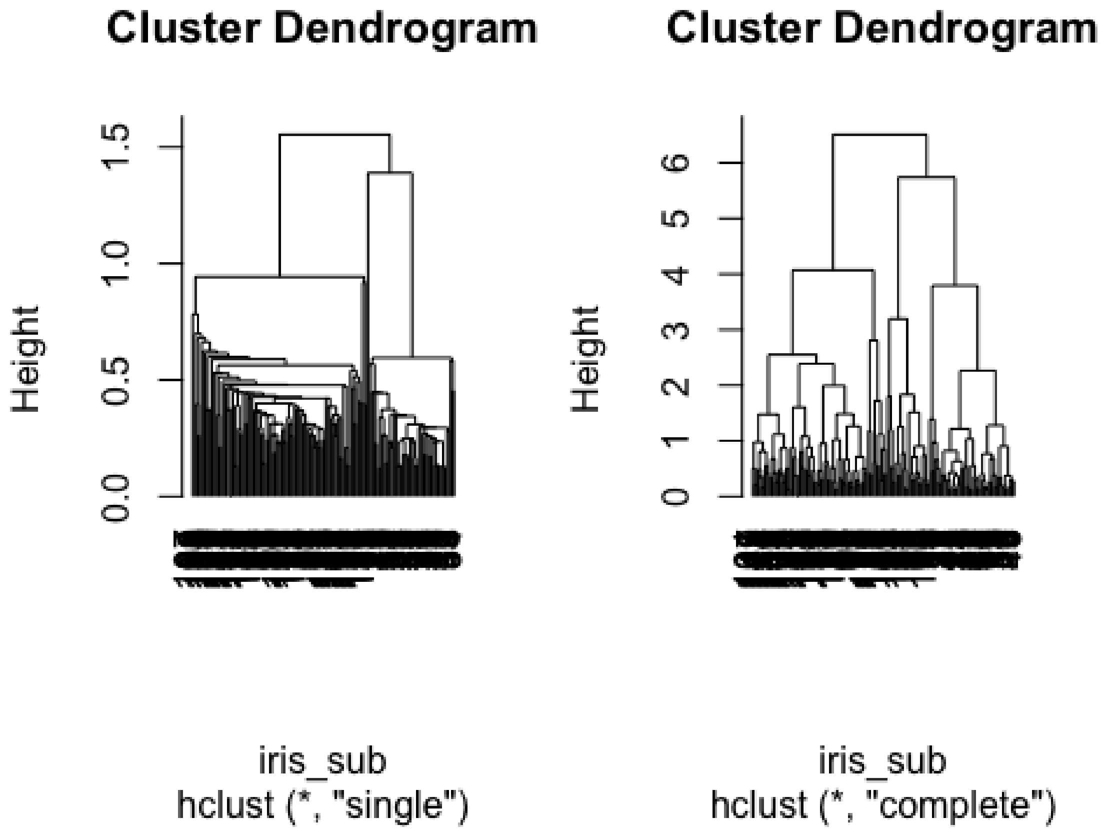
```
cut_3 <- hcut(iris_sub, k = 3, stand = TRUE)
fviz_dend(cut_3, rect = TRUE, cex = 0.5,
          k_colors = c("#00AFBB","#2E9FDF", "#E7B800"))
```



When the dendrogram is cut one more time after cut into two clusters, only the second subtree of the dendrogram is cut into another two clusters. This demonstrates the hierarchical nature of clustering that higher-level clusters include lower-level clusters.

10. Now fit the algorithm using single and complete linkage and present each dendrogram side-by-side. Discuss the differences. What effects can we see in the clustering patterns when using different linkage methods?

```r
par(mfrow = c(1,2))
hc_single <- hclust(iris_sub,
                    method = "single"); plot(hc_single, hang = -1)

hc_complete <- hclust(iris_sub,
                    method = "complete"); plot(hc_complete, hang = -1)
```



```
par(mfrow = c(1,1))
```

With the single linkage, the objects make connections to the other objects that are far away from themselves, forming elongated dendrogram. More explicitly, horizontal branches are longer when the single linkage is used than when the complete linkage is used. Consequently, it is more difficult to read the dendrogram with the single linkage than the complete one. In contrast, the dendrogram using the complete linkage is more balanced.

Critical Thinking

1. You just assessed the clusterability of some feature space, `R^n`. Address the following questions:

- How would you go about determining whether clustering made sense to consider or not?
- What are techniques you would use, and what might you be looking for from each?

Informally, I would start by plotting the data to see what they look like. Sometimes, it is clear whether they are clusterable or not simply by looking at the plot. To gain a better image of clusterability, VAT or ODI plots can be implemented. They show more distinct potential clustering by visualizing dissimilarity matrix. If the matrix demonstrates dark squares, this would indicate potential clusterability. To mathematically show clusterability, we can test Hopkins statistics. In this procedure, we assume that the data is uniformly distributed without possible clustering (null hypothesis). However, if the Hopkins statistic is larger than 0.5, we can reject the null hypothesis and suggest that the data is clusterable.

- How might these techniques work together to motivate clustering or not?

  Diagnosing clusterability with these techniques can be quite ambiguous. Therefore, one technique or approach may not be sufficient to explain clusterability. For example, simple plots are quite helpful when there is clear separation between groups. However, when the borderline is ambiguous to human eyes, VAT (ODI) can be used to reduce the ambiguity. However, VAT also relies on human eyes to recognize clusterability. In this case, we can calculate Hopkins statistics to obtain a numeric value for clusterability. On the other hand, the lack of visualization of Hopkins statistics can be complemented by plots and VAT (ODI).

- And ultimately, can/should you proceed if you find little to no support for clusterability? Why or why not?

  It might be still good to proceed to perform clustering analysis. Despite the little support for clusterability, we cannot conclude that there is no use of clustering. Unsupervised machine learning, including the clustering method, has no priors or assumptions on distributions and labeled data. It is possible that there exist clusterings but they are not detectable using these techniques.

2. Locate (and read) a paper that applies the hierarchical agglomerative clustering technique. Address the following questions:

- Describe the author(s) process.

  This paper discusses which of partitional and agglomerative algorithms works better for clustering a large array of documents. As the first step, they assigned a vector to each document after being weighted using if-idf and normalized. Next, hierarchical clustering solutions were computed by the partitional clustering algorithm as well as agglomerative algorithms. To examine the quality of clustering solutions, they compared the distribution of the nodes of the dendrograms from each algorithm.

- Do they go through similar steps as we covered this week both in setting the stage for clustering (e.g., assessing clusterability, calculating distance, etc.), as well as in fitting the algorithm? If not, what did they omit and does this omission impact their findings in your opinion?

  Setting the stage for clustering, they did not include how they assessed clusterability. However, for the purpose of the study, the impact of this assessment may be insignificant. The main concern of this study is to find a better tool to generate hierarchical clustering by comparing partitional and agglomerative algorithms. If they were using only one algorithm, it would have been essential to check for clustrability.

However, because they are measuring the difference between various algorithms, clusterability assessment seems less important.

- Describe at least one possible extension from the study that could emerge based on their findings.

  The study finds that partitional methods can better produce meaningful hierarchical clustering from documents than agglomerative algorithms. This may reflect how documents are produced. When we write an article, we first come up with a topic that is consistent throughout the entire article. Therefore, the article does not stand by each word or part of it but has the meaning as a whole. It works the same with the categorization of individual articles. We tend to consider the broader topic and categorize the documents by their topic. We seldom start by looking at the individual documents or part of an article to determine what they are about.