Abhishek_Pandit_hw2

Abhishek Pandit

19 October 2019

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 Manahattan, 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? 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? 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?

```
p<-c(1,2)
q<-c(3,4)
dist_data<-data.frame(p,q)
```

Now we try to calculate the 3 distance metrics.

```
manhattan <-function(p,q){
  distance<-abs(p-q)
  total_distance<-sum(distance)
  return(total_distance)
  }

euclid <-function(p,q){
  distance<-(p-q)^2
  total_distance<-sum(distance)
  final<-sqrt(total_distance)
  return(final)
}


canberra <-function(p,q){
  total_dist <-0
  for (i in length(p)){
    distance<-abs(p[i]-q[i])/(abs(p[i])+abs(q[i]))
    total_dist <-total_dist + distance}
  return(total_dist)}
```

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?

Now we apply them to our fictitious data

```
manhattan(p,q)
```

```
## [1] 4
```

```
euclid(p,q)
```

```
## [1] 2.828427
```

```
canberra(p,q)
```

```
## [1] 0.3333333
```

Now we check against the pre-existing function

```
euc = dist(dist_data, method="euclidean")
manh = dist(dist_data, method="manhattan")
canb = dist(dist_data, method="canberra")
all_dist<-c(euc, manh, canb)
all_dist
```

```
## [1] 1.4142136 2.0000000 0.4761905
```

The expected values as per the dist function were 1.4142136 2.0000000 0.4761905 respectively. I initially got all three wrong- with 4, 2.82, 0.33. On further inspecton, my mistakes were to do with how I created the matrix from the two vectors. I forgot to use cbind.

```
dist_data = data.frame(cbind(p,q))
euc = dist(dist_data, method="euclidean")
manh = dist(dist_data, method="manhattan")
canb = dist(dist_data, method="canberra")
all_dist<-c(euc, manh, canb)
all_dist
```

```
## [1] 1.4142136 2.0000000 0.4761905
```

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?

Euclidean distance considers the geometric distance as the shortest line between two data points, while Manhattan distance finds the shortest distance specifically along the feature axes. Canberra distance considers the absolute value of the distances in each dimensions (feature) after normalizing for the sum of the absolute value of each feature individually. This ensures that the distance for each feature always lies between 0 and 1 (though the sum can be greater than 1).

The choice of these measures becomes crucial in deciding the 'nearness' of data points, and thus in their allocation to clusters. The results of clustering process could vary greatly simply by changing the metric.

For these specific data, we could see these measures in action by plotting the individual vectors (since we are still operating in 2 dimensions). The Euclidean distance would be the straight line connecting them, while the Manhattan would be equivalent to dropping a vertical and horizontal perpendicular from the first and second points, noting their intersection and then adding the 2 resulting lines. The Euclidean measure would thus be akin to the hypotenuse of a right triangle, of which the other two sides can be summed to derive the Manhattan distance.

The Canberra distance is itself less amenable to visualization.

```
faith<-faithful
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 3.5.3
```

```
## -- Attaching packages ---------------------------------------------------------------------------------- t
```

```
## v ggplot2 3.1.0     v purrr   0.2.5
## v tibble  2.0.1     v dplyr   0.7.8
## v tidyr   0.8.2     v stringr 1.3.1
## v readr   1.3.1     v forcats 0.3.0
```

```
## Warning: package 'ggplot2' was built under R version 3.5.2
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
## Warning: package 'readr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
## Warning: package 'dplyr' was built under R version 3.5.2
## Warning: package 'stringr' was built under R version 3.5.2
## Warning: package 'forcats' was built under R version 3.5.2
## -- Conflicts ---------------------------------------------------------------------------- tidyve
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(skimr)
```

```
## Warning: package 'skimr' was built under R version 3.5.3
##
## Attaching package: 'skimr'
## The following object is masked from 'package:stats':
##
##     filter
```

```
library(seriation)
```

```
## Warning: package 'seriation' was built under R version 3.5.3
```

```
summary(faith)
```

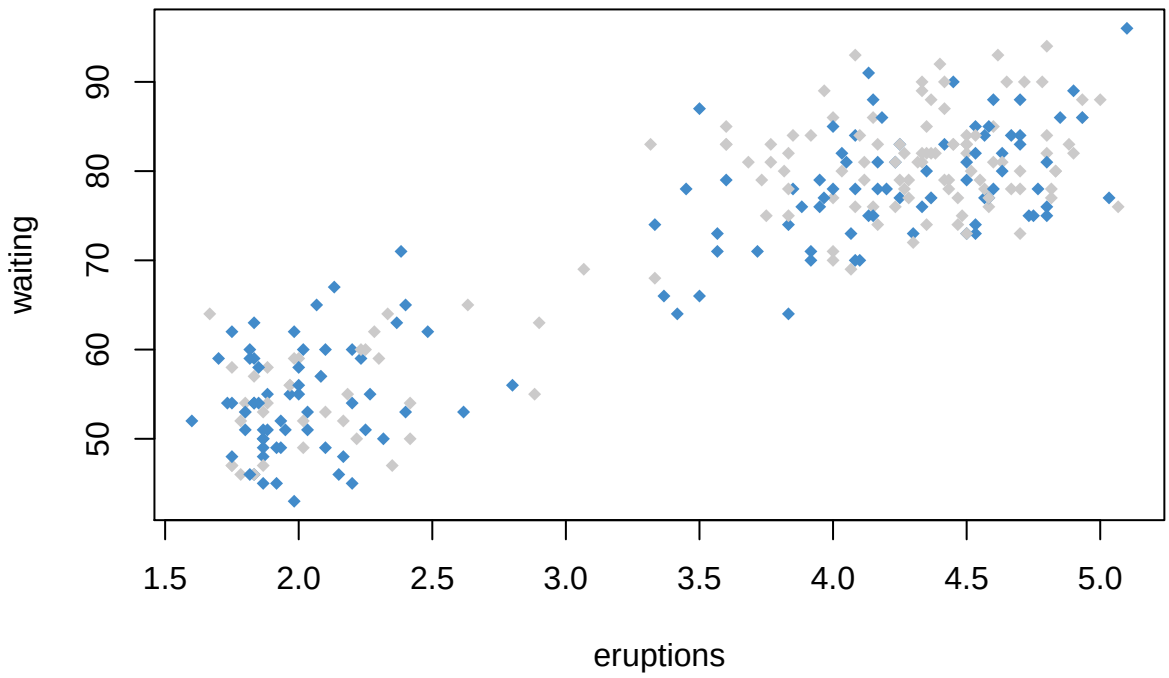
```
##    eruptions        waiting
##  Min.   :1.600   Min.   :43.0
##  1st Qu.:2.163   1st Qu.:58.0
##  Median :4.000   Median :76.0
##  Mean   :3.488   Mean   :70.9
##  3rd Qu.:4.454   3rd Qu.:82.0
##  Max.   :5.100   Max.   :96.0
```

```
skim(faith)
```

```
## Skim summary statistics
##  n obs: 272
##  n variables: 2
## Warning: package 'bindrcpp' was built under R version 3.5.2
##
## -- Variable type:numeric ---------------------------------------------------------------------
##   variable missing complete   n  mean    sd   p0   p25 p50   p75 p100
##  eruptions       0      272 272  3.49  1.14  1.6  2.16   4  4.45  5.1
##    waiting       0      272 272 70.9  13.59 43   58     76 82    96
##      hist
##  <U+2587><U+2583><U+2581><U+2581><U+2582><U+2585><U+2587><U+2583>
##  <U+2582><U+2585><U+2583><U+2582><U+2585><U+2587><U+2586><U+2582>
```

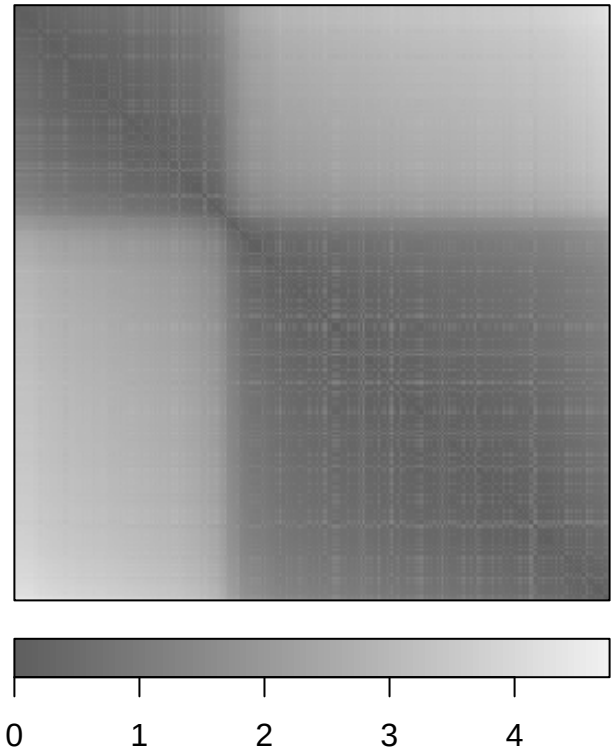
We thus see that the data consists of 2 variables. Both variables appear to be bimodal

```
plot(faith,
     col = c("#428bca", "#cbcaca"),
     pch = 18,
     cex = 0.9)
```



We can discern two clusters in this data, as were suggested by the bimodal summary statistics We now visualize the ODI.

```
faith_scaled <- scale(faith)
faith_dist <- dist(faith_scaled,
                   method = "euclidean")
dissplot(faith_dist)
```



The dark shading in two distinct squae sections of the plot indicates that we most likely have two clusters in the data.

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 "pipe" for stacking functions to do this quickly, e.g.)

```
skim(iris)
```

```
## Skim summary statistics
##  n obs: 150
##  n variables: 5
##
## -- Variable type:factor ---------------------------------------------------------------------------
##  variable missing complete   n n_unique                       top_counts
##   Species       0      150 150        3 set: 50, ver: 50, vir: 50, NA: 0
##  ordered
##    FALSE
##
## -- Variable type:numeric --------------------------------------------------------------------------
##      variable missing complete   n mean   sd  p0 p25  p50 p75 p100
##  Petal.Length       0      150 150 3.76 1.77 1   1.6 4.35 5.1  6.9
##   Petal.Width       0      150 150 1.2  0.76 0.1 0.3 1.3  1.8  2.5
##  Sepal.Length       0      150 150 5.84 0.83 4.3 5.1 5.8  6.4  7.9
##   Sepal.Width       0      150 150 3.06 0.44 2   2.8 3    3.3  4.4
##      hist
##  <U+2587><U+2581><U+2581><U+2582><U+2585><U+2585><U+2583><U+2581>
```

```
##  <U+2587><U+2581><U+2581><U+2585><U+2583><U+2583><U+2582><U+2582>
##  <U+2582><U+2587><U+2585><U+2587><U+2586><U+2585><U+2582><U+2582>
##  <U+2581><U+2582><U+2585><U+2587><U+2583><U+2582><U+2581><U+2581>
```

```
iris_sub<-iris %>%
  dplyr::select(Sepal.Length, Sepal.Width) %>%
  scale() %>%
  dist()
summary(iris_sub)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.037   1.716   1.774   2.394   5.571
```

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?

```
library(dendextend)
```

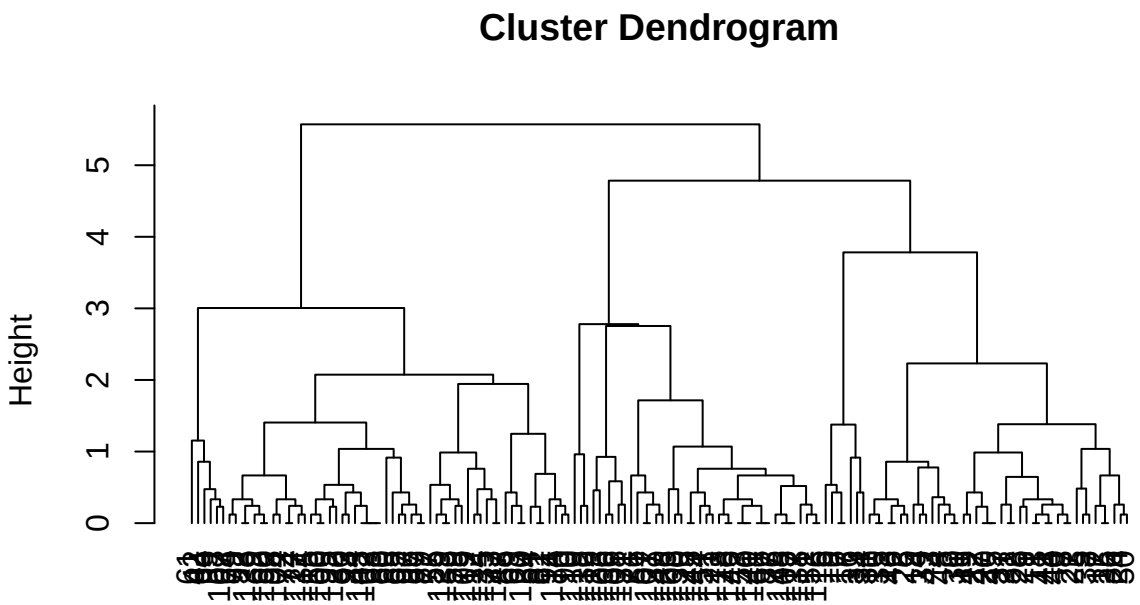
```
## Warning: package 'dendextend' was built under R version 3.5.3
```

```
##
## ---------------------
## Welcome to dendextend version 1.12.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## Or contact: <tal.galili@gmail.com>
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ---------------------
```

```
##
## Attaching package: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

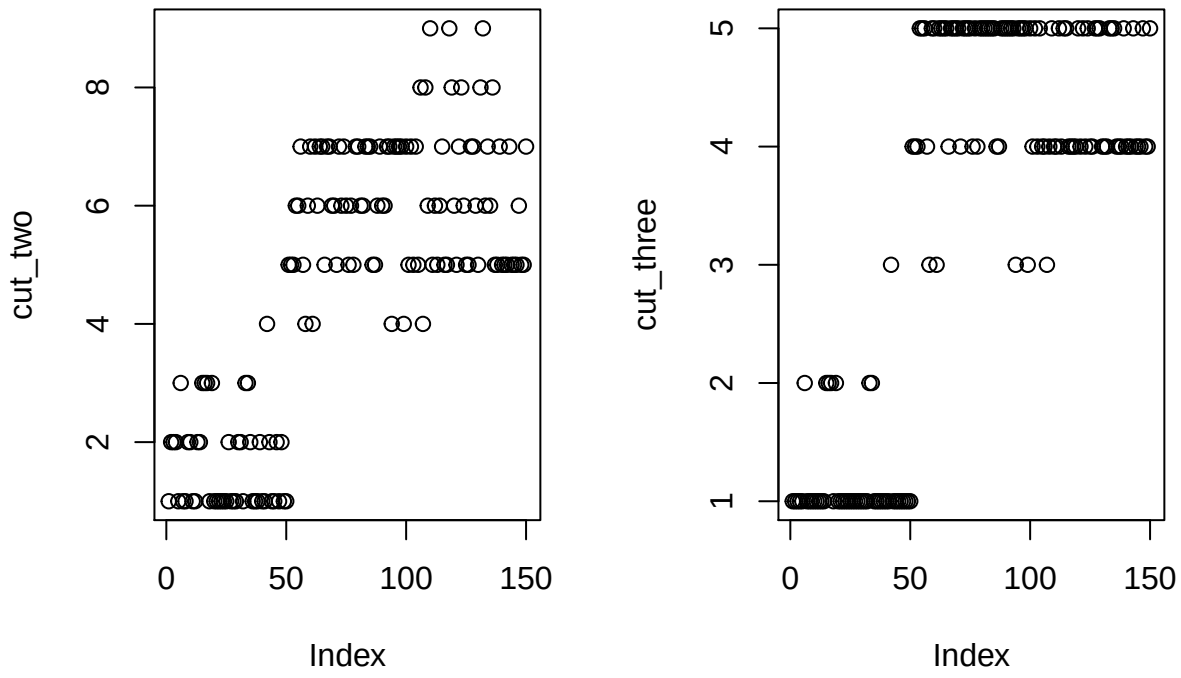
```
hc_complete <- hclust(iris_sub,
                      method = "complete"); plot(hc_complete, hang = -1)
```



One can see that three clusters emerge, before being grouped into 2. This may explain why with some of the features of the iris data set, we may find 2 species to show marked similarity.

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

```
par(mfrow = c(1,2))
cut_two <- cutree(hc_complete, h=2);plot(cut_two)
cut_three <- cutree(hc_complete, h=3) ; plot(cut_three)
```



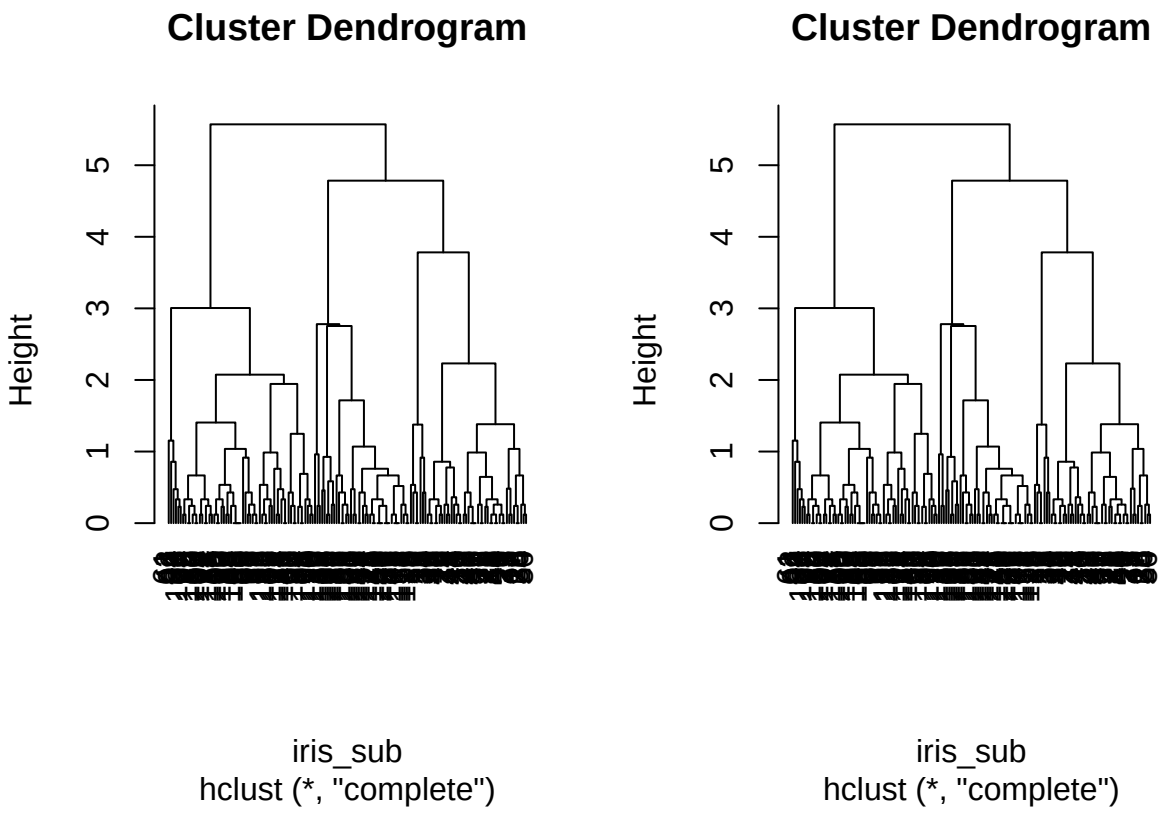
Now in tabular form

```
cuts <- cutree(hc_complete,
               k = c(2,3))
table(`2 Clusters` = cuts[,1],
      `3 Clusters` = cuts[,2])
```

```
##            3 Clusters
## 2 Clusters  1  2  3
##          1 49  0 40
##          2  0 61  0
```

From this preliminary inspection,we see that the 2 classifications roughly agree on which flowers belong in clusters 1 and 2. The disagreement lies around the 3rd cluster.

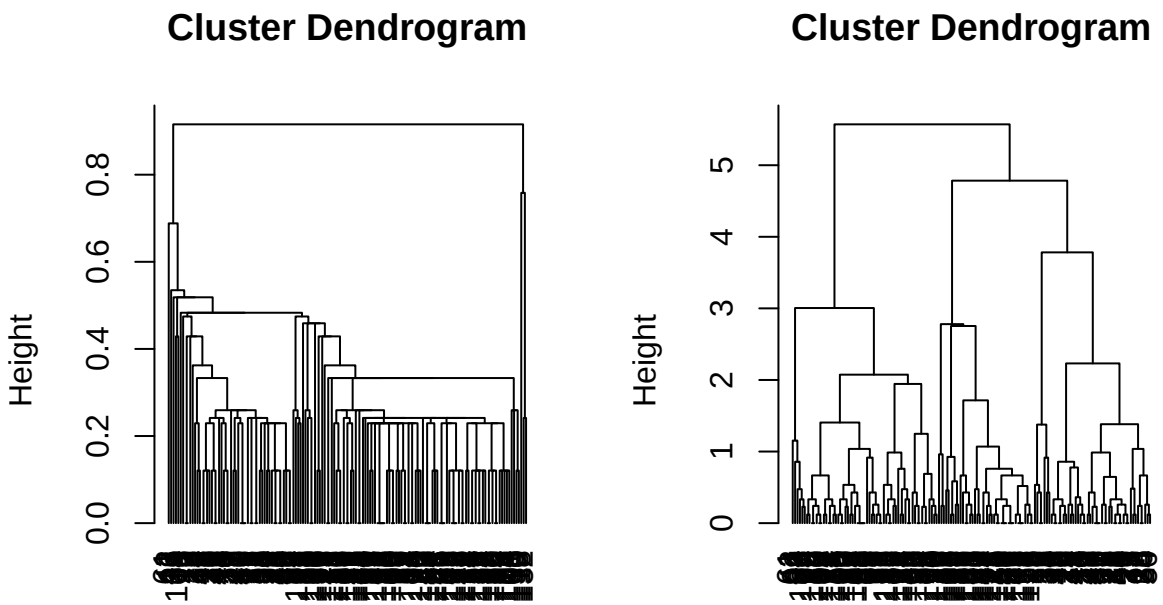
```
par(mfrow = c(1,2))
hc_complete_two <- hclust(iris_sub,
                    method = "complete"); plot(hc_complete_two, hang = -1)
hc_complete_three <- hclust(iris_sub,
                    method = "complete"); plot(hc_complete_three, hang = -1)
```



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?

```
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)
```



The core difference lies in how these methods are defined. In the single method, the distance between clusters is defined between the closest pairs, while the complete method calculates this distance between the points furthest apart.

In the clustering by the single method (on the left), we see a much higher degree of branching (considerably more sub-clusters). Interestingly the lengths of the y axis along appears consistent at each level of clustering process, leading to a more geometric and seemingly orderly pattern to the clustering. Conversely, in the complete method, we see fewer sub-clusters, with the agglomeration occurring at a variety of heights in the bottom-up process.

CRITICAL THINKING

1. You just assessed the clusterability of some feature space, ???". Address the following questions:

a. How would you go about determining whether clustering made sense to consider or not?

One method, like with the iris data, would be to plot known labels (in this case, species) and see if the clustering matches with the known labels.

If such labels do not exist, we will need subject domain expertise to be able to assess whether these clusters actually make sense. For example, in US political campaigns, we may find voters clustering into Democrats and Republicans even along non-partisan variables.

b. What are techniques you would use, and what might you be looking for from each?

First, we would need to select the appropriate distance measurement metric. These could be based on an understanding of research questions and the nature of variables at okay.

Next, techiques to use would involve informal scatterplots, calculation of dissimilarity matrices and the associated ODI plots.

Thereafter, we would begin a more formal exploration, through measures such as the Hopkin Statistic. If its value exceeds 0.5, we may need to consider whether we are barking up the wrong tree.

c. How might these techniques work together to motivate clustering or not? Initial examinations through visualizations- such as scatterplots and ODIs could help indicate the potential of cluster existence. We can narrow down the set of features that are of most interest.

Formal methods such as the H Statistics then help test for these hypotheses, based on whichever seems most promising.

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

It is well possible that we have chosen features for clustering along which there are no discernible clusters, although the data points do in fact cluster. Thus, one preliminary step may be to consider any important features that have been excluded.

However, if multiple techniques reveal the same trends, it is probably wiser to proceed with a different hypothesis or apply a statistical method uniformly to all data points rather than expecting differences in each.

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

a. Describe the author(s) process.

I will be quoting from a research paper on Genetics: Odong, T.L., van Heerwaarden, J., Jansen, J. et al. Theor Appl Genet (2011) 123: 195. https://doi.org/10.1007/s00122-011-1576-x

The authors seek to address gene diversity in cultivated crops, and harness the example of coconuts with 30 SSR markers (Simple Sequence Repeat), owing largely due to the higher number of accessions of each of the varied origins across geographies. Both real and simulated data are used, with the latter drawing on finite island and a stepping stone migration models.The ideal number of clusters was calculated using two methods- Point-Biserial Correlation and Average Silhouette Coefficient. The authors then undertake Hierarchical Agglomerative clustering. They use 2 clustering methods- Ward, STRUCTURE and UPGMA (which seems specific to this domain) and find similar results. Ultimately, coconuts from the Pacific region showed distinct genetic markers than those in the Indian or Atlantic Ocean.

b. 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?

They do not visualize the space of the 30 markers- understandably due to the dimension of the data. Nonetheless, other methods such as ODI plots may have proven more heloful in determining clusterability.

They also leverage mutliple methods to validate their methods at different heights of the dendogram tress, which leads to more rigorous analyis.

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

They could consider clustering with subsets of the 30 markers, and find those with the maximum predictive power for the clusters that were found in this method- in a sense, it is dimensionality rediction