RuiHe_PS2

Rui He

10/15/2019

##Computation

```
p <- c(1,2)
q <- c(3,4)

manhattan_distance <- abs(p[1]-q[1])+abs(p[2]-q[2])
euclidean_distance <- sqrt((p[1]-q[1])^2+(p[2]-q[2])^2)
canberra_distance <- abs(p[1]-q[1])/(p[1]+q[1])+abs(p[2]-q[2])/(p[2]+q[2])

x_scaled <- scale(cbind(p,q))
e_dist <- dist(x_scaled, method = "euclidean")
m_dist <- dist(x_scaled, method = "manhattan")
c_dist <- dist(x_scaled, method = "canberra")
```

1. manhattan distance is 4, canberra distance is 0.833 and euclidean distance is 2.83.
2. When I first try using 'dist' function, my answer was half smaller compared to the calculation due to lack of scaling. After scale the variables correctly, I got the same results.
3. The key difference of the three measurements of distance lies in the way how each differences between two vectors are weighted. The different weights may generate different distance, which describe the similarity/dissimilarity between two vectors. For the ficitious data, we can see that these three methods give different distance results.

EDA - old faithful

4. The table summarize the means and medians of the two variables in the data: eruption duration and waiting time between two eruptions. The scatterplot and the fitting line provide a general visualization of the data, showing that waiting time generally positively correlates to the eruption duration, and it seems that there are two clusters in the data.

5 & 6. See figure above. There are two clusters appeared on ODI.

IRIS Data

```
library(dendextend) # for "cutree" function
```

```
##
## --------------------
## 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
```

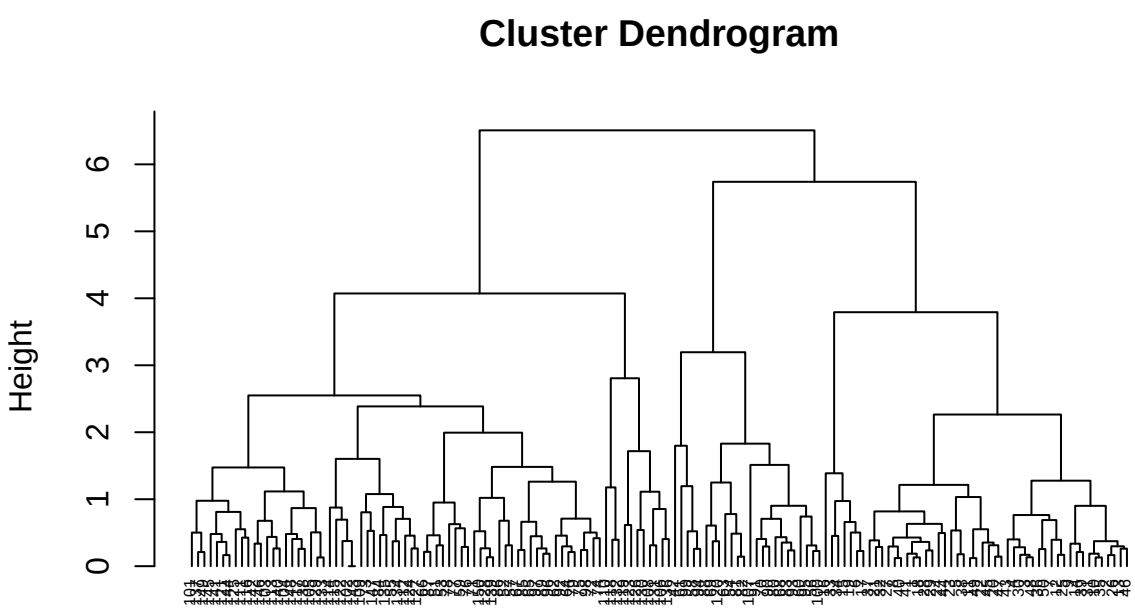
```
library(ape)
```

```
##
## Attaching package: 'ape'
```

```
## The following objects are masked from 'package:dendextend':
##
##     ladderize, rotate
```

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


hc_complete <- hclust(iris_sub,
                      method = "complete"); plot(hc_complete, hang = -1,cex=0.5)
```




```
cuts <- cutree(hc_complete,
               k = c(2,3))

cuts
```

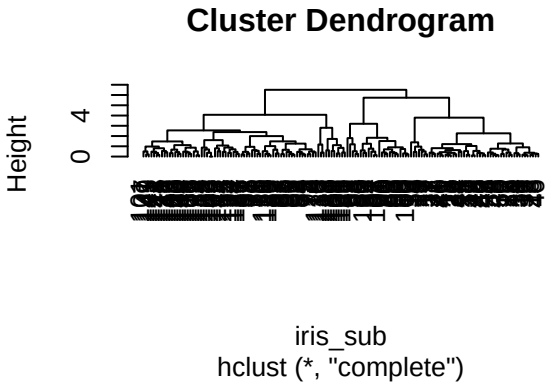
```
##        2 3
##   [1,] 1 1
##   [2,] 1 1
##   [3,] 1 1
##   [4,] 1 1
##   [5,] 1 1
##   [6,] 1 1
##   [7,] 1 1
##   [8,] 1 1
##   [9,] 1 1
##  [10,] 1 1
##  [11,] 1 1
##  [12,] 1 1
##  [13,] 1 1
##  [14,] 1 1
##  [15,] 1 1
##  [16,] 1 1
##  [17,] 1 1
##  [18,] 1 1
##  [19,] 1 1
##  [20,] 1 1
```

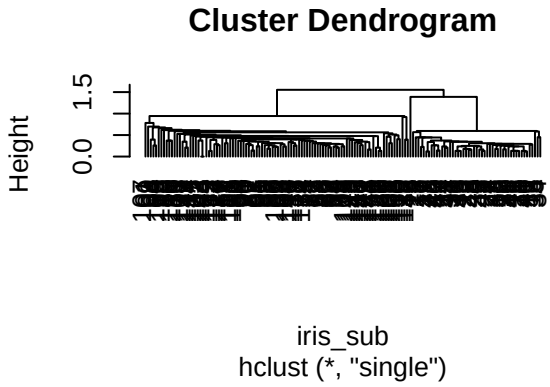
```
##  [21,] 1 1
##  [22,] 1 1
##  [23,] 1 1
##  [24,] 1 1
##  [25,] 1 1
##  [26,] 1 1
##  [27,] 1 1
##  [28,] 1 1
##  [29,] 1 1
##  [30,] 1 1
##  [31,] 1 1
##  [32,] 1 1
##  [33,] 1 1
##  [34,] 1 1
##  [35,] 1 1
##  [36,] 1 1
##  [37,] 1 1
##  [38,] 1 1
##  [39,] 1 1
##  [40,] 1 1
##  [41,] 1 1
##  [42,] 1 2
##  [43,] 1 1
##  [44,] 1 1
##  [45,] 1 1
##  [46,] 1 1
##  [47,] 1 1
##  [48,] 1 1
##  [49,] 1 1
##  [50,] 1 1
##  [51,] 2 3
##  [52,] 2 3
##  [53,] 2 3
##  [54,] 1 2
##  [55,] 2 3
##  [56,] 1 2
##  [57,] 2 3
##  [58,] 1 2
##  [59,] 2 3
##  [60,] 1 2
##  [61,] 1 2
##  [62,] 2 3
##  [63,] 1 2
##  [64,] 2 3
##  [65,] 2 3
##  [66,] 2 3
##  [67,] 2 3
##  [68,] 1 2
##  [69,] 1 2
##  [70,] 1 2
##  [71,] 2 3
##  [72,] 2 3
##  [73,] 2 3
##  [74,] 2 3
```

```
##  [75,] 2 3
##  [76,] 2 3
##  [77,] 2 3
##  [78,] 2 3
##  [79,] 2 3
##  [80,] 1 2
##  [81,] 1 2
##  [82,] 1 2
##  [83,] 1 2
##  [84,] 2 3
##  [85,] 2 3
##  [86,] 2 3
##  [87,] 2 3
##  [88,] 1 2
##  [89,] 2 3
##  [90,] 1 2
##  [91,] 1 2
##  [92,] 2 3
##  [93,] 1 2
##  [94,] 1 2
##  [95,] 1 2
##  [96,] 2 3
##  [97,] 2 3
##  [98,] 2 3
##  [99,] 1 2
## [100,] 1 2
## [101,] 2 3
## [102,] 2 3
## [103,] 2 3
## [104,] 2 3
## [105,] 2 3
## [106,] 2 3
## [107,] 1 2
## [108,] 2 3
## [109,] 2 3
## [110,] 2 3
## [111,] 2 3
## [112,] 2 3
## [113,] 2 3
## [114,] 2 3
## [115,] 2 3
## [116,] 2 3
## [117,] 2 3
## [118,] 2 3
## [119,] 2 3
## [120,] 1 2
## [121,] 2 3
## [122,] 2 3
## [123,] 2 3
## [124,] 2 3
## [125,] 2 3
## [126,] 2 3
## [127,] 2 3
## [128,] 2 3
```

```
## [129,] 2 3
## [130,] 2 3
## [131,] 2 3
## [132,] 2 3
## [133,] 2 3
## [134,] 2 3
## [135,] 2 3
## [136,] 2 3
## [137,] 2 3
## [138,] 2 3
## [139,] 2 3
## [140,] 2 3
## [141,] 2 3
## [142,] 2 3
## [143,] 2 3
## [144,] 2 3
## [145,] 2 3
## [146,] 2 3
## [147,] 2 3
## [148,] 2 3
## [149,] 2 3
## [150,] 2 3
```

```r
par(mfrow=c(2,2))
plot(hc_complete,hang = -1)
hc_single <- hclust(iris_sub,
                    method = "single"); plot(hc_single, hang = -1) #everything on a line
```





7. see code
8. The dendrogram with the complete linkage shows that the data can be separate into 2-3 main categories with furthes distance. Numbers within a similar range seem to be clustered together as well. For example, the branches on the right and in the middle are the ones with number id above 100.
9. When cutting the tree at three branches, the ones previously clustered into the second category became clustered into the third category whereas the ones that previously clustered into the first cluster now splited into two separate clusters. It suggest that while one cluster is more dissimilar from the rest of the data, the other cluster can be roughly divided into two branches.
10. These two methods both generate clusters where higher numbers (>100) and lower numbers tend to aggregate into two different clusters. The algorithm using single linkage show a more distinctive three-branches The algorithm using complete linkage have 3~4 main branches.

Critical thinking

1.

- There are three major ways to determine clusterability. First, we can do it informally by plotting the data on a scatterplot or a histogram, to have a general idea of the distribution of data and postulate if the plots show some trends of clusterable groups.

Secondly, we can rely on visualization including Visual Assessment of Tendency (VAT) or Ordered Dissimilarity Images (ODI). ODI is calculated with a dissimilarity matrix and a darker block along the diagonal may suggest that there exists a group that is spatially close to each other.

Thirdly, we can use mathematical method to calculate the Hopkin Statistic, which assume that the data is spatially random and test this null hypothesis with a sparse sampling test. Basically, this methods compare the dissimilarity across all observations in the actual data with that of a set of simulated data drawn from a random uniform distribution with the same st.dev as the actual data. If H stats > 0.5, the null hypothesis is rejected, and this suggest that the actual data is not randomly distributed (i.e. clusterable).

- While the informal way to assess clusterability may be faster and easier to do, it can be hard to detect clusterability when there are too many variables in the data and a large number of plots. ODI provides a more concrete way of data visualization, and H stats allows us to process data in a way that may yield statistically sginificant results. These three methods can cross-validate the clusterability of teh data.
- Personally, I will proceed if I find litter support for clusterability. Clusterability may not be salient in the visualization, and a statistical test can have false negative. The clustering result may yield something surprising. However, it is important to keep in mind that the clusters may not be as distinctive as we would want due to the lack of support for clusterability.

2. Paper: Suomi, A., Dowling, N. A., & Jackson, A. C. (2014). Problem gambling subtypes based on psychological distress, alcohol abuse and impulsivity. Addictive Behaviors, 39(12), 1741-1745.

https://reader.elsevier.com/reader/sd/pii/S0306460314002548?token=13896F50DA33365B0CBA0EBD270888E99DCED65D

- The authors perform HCA to explore whether there exists subtypes of gamblers on three comorbidities including psychological distress, alcohol abuse and impulsivity. They found that there are four different subgroups: gamblers with comorbid psychological problem, gamblers without other comorbidities, gamblers with comorbid alchol abuse and multimorbid gamblers. They also found that the four groups differed on the demographic factors and general well-being factors. For example, gamblers with psychological distress tend to be old age female and use gambling as a way to cope with negative emotion. Whereas gamblers with multiple comorbidity were more often charaterised by being male, hostile and aggresive. The most common form of gambling was horse/dog racing and less likely to report abstinenece.
- In the paper, they did not report checking for clusterability. In the discussion, they mentioned other research that found correlations between gambiling and multiple psychological disorders, impulsivity, and substance abuse. So although they did not perform statistical test or visualization to support their rationale for clusterability, the introduction sort of set up a stage for possible clusters in the dataset. Their results seem pretty robust and was not significantly affected by the lack of assessment.
- There is possible clinical implication for the results. For example, the group of gamblers who used gambling to cope with negative emotions may respond to treatment that provides alternative distress coping mechnism. The group with multiple comorbidity may present a more complex situation that requires an integrative treatments combing impulse control strategies, psycho-pharmarcotherapy and etc.