PS 2

Unsupervised Machine Learning (40800)

William Parker

Contents

Computation

```
library(tidyverse)
library(dendextend)
```

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

```
p <- c(1,2)

q <-c(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?

Manhattan

$d_{manhattan}(p,q) = \sum_{i=1}^{n} |p_i - q_i|$

```
d_minkowski <- function(p, q, m){

  dist <- function(x,y) abs(x-y)^m

  (sum(mapply(dist, p, q)))^(1/m)
}

Manhattan_dist <- d_minkowski(p, q, 1)
```

the manhattan distance is 4

Canberra

$d_{manhattan}(p,q) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i| + |q_i|}$

```
d_canberra <- function(p, q){

  dist <- function(x,y) abs(x-y)/(abs(x) + abs(y))

  sum(mapply(dist, p, q))
}

Canberra_dist <- d_canberra(p,q)
```

The Canberra distance is 0.8333333

Euclidean

$d_{Euclidean} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$

```
Euclidean_dist <- d_minkowski(p, q, 2)
```

The Euclidean distance is 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?

```
dist(rbind(p,q), method = "manhattan")
```

```
##   p
## q 4
```

```
dist(rbind(p,q), method = "canberra")
```

```
##           p
## q 0.8333333
```

```
dist(rbind(p,q), method = "euclidean")
```

```
##          p
## q 2.828427
```

All my calculated distances were correct!

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 Manhattan distance is the sum of the absolute values of the differences between each vector component, here it is larger than the euclidean distance which is the square root of the sum of square of distances between components. If p and q were physical points in space this would represent the "taxicab" and "crow files" distances respectively, and explain why the manhattan distance is larger.

The Canberra is a weighted version of the manhattan distance. This would be more senstive to changes in p and q very close to the origin ((0,0) in this case), as it weighs each absolute value by the sum of the asbolute value of p and q

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

I'll start by just looking at the dataset

```
faithful <- faithful

faithful
```

```
##     eruptions waiting
## 1       3.600      79
## 2       1.800      54
## 3       3.333      74
## 4       2.283      62
## 5       4.533      85
## 6       2.883      55
## 7       4.700      88
## 8       3.600      85
## 9       1.950      51
## 10      4.350      85
## 11      1.833      54
## 12      3.917      84
```

```
## 13      4.200      78
## 14      1.750      47
## 15      4.700      83
## 16      2.167      52
## 17      1.750      62
## 18      4.800      84
## 19      1.600      52
## 20      4.250      79
## 21      1.800      51
## 22      1.750      47
## 23      3.450      78
## 24      3.067      69
## 25      4.533      74
## 26      3.600      83
## 27      1.967      55
## 28      4.083      76
## 29      3.850      78
## 30      4.433      79
## 31      4.300      73
## 32      4.467      77
## 33      3.367      66
## 34      4.033      80
## 35      3.833      74
## 36      2.017      52
## 37      1.867      48
## 38      4.833      80
## 39      1.833      59
## 40      4.783      90
## 41      4.350      80
## 42      1.883      58
## 43      4.567      84
## 44      1.750      58
## 45      4.533      73
## 46      3.317      83
## 47      3.833      64
## 48      2.100      53
## 49      4.633      82
## 50      2.000      59
## 51      4.800      75
## 52      4.716      90
## 53      1.833      54
## 54      4.833      80
## 55      1.733      54
## 56      4.883      83
## 57      3.717      71
## 58      1.667      64
## 59      4.567      77
## 60      4.317      81
## 61      2.233      59
## 62      4.500      84
## 63      1.750      48
## 64      4.800      82
## 65      1.817      60
## 66      4.400      92
```

```
## 67      4.167      78
## 68      4.700      78
## 69      2.067      65
## 70      4.700      73
## 71      4.033      82
## 72      1.967      56
## 73      4.500      79
## 74      4.000      71
## 75      1.983      62
## 76      5.067      76
## 77      2.017      60
## 78      4.567      78
## 79      3.883      76
## 80      3.600      83
## 81      4.133      75
## 82      4.333      82
## 83      4.100      70
## 84      2.633      65
## 85      4.067      73
## 86      4.933      88
## 87      3.950      76
## 88      4.517      80
## 89      2.167      48
## 90      4.000      86
## 91      2.200      60
## 92      4.333      90
## 93      1.867      50
## 94      4.817      78
## 95      1.833      63
## 96      4.300      72
## 97      4.667      84
## 98      3.750      75
## 99      1.867      51
## 100     4.900      82
## 101     2.483      62
## 102     4.367      88
## 103     2.100      49
## 104     4.500      83
## 105     4.050      81
## 106     1.867      47
## 107     4.700      84
## 108     1.783      52
## 109     4.850      86
## 110     3.683      81
## 111     4.733      75
## 112     2.300      59
## 113     4.900      89
## 114     4.417      79
## 115     1.700      59
## 116     4.633      81
## 117     2.317      50
## 118     4.600      85
## 119     1.817      59
## 120     4.417      87
```

```
## 121     2.617      53
## 122     4.067      69
## 123     4.250      77
## 124     1.967      56
## 125     4.600      88
## 126     3.767      81
## 127     1.917      45
## 128     4.500      82
## 129     2.267      55
## 130     4.650      90
## 131     1.867      45
## 132     4.167      83
## 133     2.800      56
## 134     4.333      89
## 135     1.833      46
## 136     4.383      82
## 137     1.883      51
## 138     4.933      86
## 139     2.033      53
## 140     3.733      79
## 141     4.233      81
## 142     2.233      60
## 143     4.533      82
## 144     4.817      77
## 145     4.333      76
## 146     1.983      59
## 147     4.633      80
## 148     2.017      49
## 149     5.100      96
## 150     1.800      53
## 151     5.033      77
## 152     4.000      77
## 153     2.400      65
## 154     4.600      81
## 155     3.567      71
## 156     4.000      70
## 157     4.500      81
## 158     4.083      93
## 159     1.800      53
## 160     3.967      89
## 161     2.200      45
## 162     4.150      86
## 163     2.000      58
## 164     3.833      78
## 165     3.500      66
## 166     4.583      76
## 167     2.367      63
## 168     5.000      88
## 169     1.933      52
## 170     4.617      93
## 171     1.917      49
## 172     2.083      57
## 173     4.583      77
## 174     3.333      68
```

```
## 175     4.167      81
## 176     4.333      81
## 177     4.500      73
## 178     2.417      50
## 179     4.000      85
## 180     4.167      74
## 181     1.883      55
## 182     4.583      77
## 183     4.250      83
## 184     3.767      83
## 185     2.033      51
## 186     4.433      78
## 187     4.083      84
## 188     1.833      46
## 189     4.417      83
## 190     2.183      55
## 191     4.800      81
## 192     1.833      57
## 193     4.800      76
## 194     4.100      84
## 195     3.966      77
## 196     4.233      81
## 197     3.500      87
## 198     4.366      77
## 199     2.250      51
## 200     4.667      78
## 201     2.100      60
## 202     4.350      82
## 203     4.133      91
## 204     1.867      53
## 205     4.600      78
## 206     1.783      46
## 207     4.367      77
## 208     3.850      84
## 209     1.933      49
## 210     4.500      83
## 211     2.383      71
## 212     4.700      80
## 213     1.867      49
## 214     3.833      75
## 215     3.417      64
## 216     4.233      76
## 217     2.400      53
## 218     4.800      94
## 219     2.000      55
## 220     4.150      76
## 221     1.867      50
## 222     4.267      82
## 223     1.750      54
## 224     4.483      75
## 225     4.000      78
## 226     4.117      79
## 227     4.083      78
## 228     4.267      78
```

```
## 229     3.917      70
## 230     4.550      79
## 231     4.083      70
## 232     2.417      54
## 233     4.183      86
## 234     2.217      50
## 235     4.450      90
## 236     1.883      54
## 237     1.850      54
## 238     4.283      77
## 239     3.950      79
## 240     2.333      64
## 241     4.150      75
## 242     2.350      47
## 243     4.933      86
## 244     2.900      63
## 245     4.583      85
## 246     3.833      82
## 247     2.083      57
## 248     4.367      82
## 249     2.133      67
## 250     4.350      74
## 251     2.200      54
## 252     4.450      83
## 253     3.567      73
## 254     4.500      73
## 255     4.150      88
## 256     3.817      80
## 257     3.917      71
## 258     4.450      83
## 259     2.000      56
## 260     4.283      79
## 261     4.767      78
## 262     4.533      84
## 263     1.850      58
## 264     4.250      83
## 265     1.983      43
## 266     2.250      60
## 267     4.750      75
## 268     4.117      81
## 269     2.150      46
## 270     4.417      90
## 271     1.817      46
## 272     4.467      74
```

Looks like we have 2 variables, `eruptions` and `waiting`. I'll use the `skimr` package to get some numeric summaries of each variable

```
skimr::skim(faithful)
```

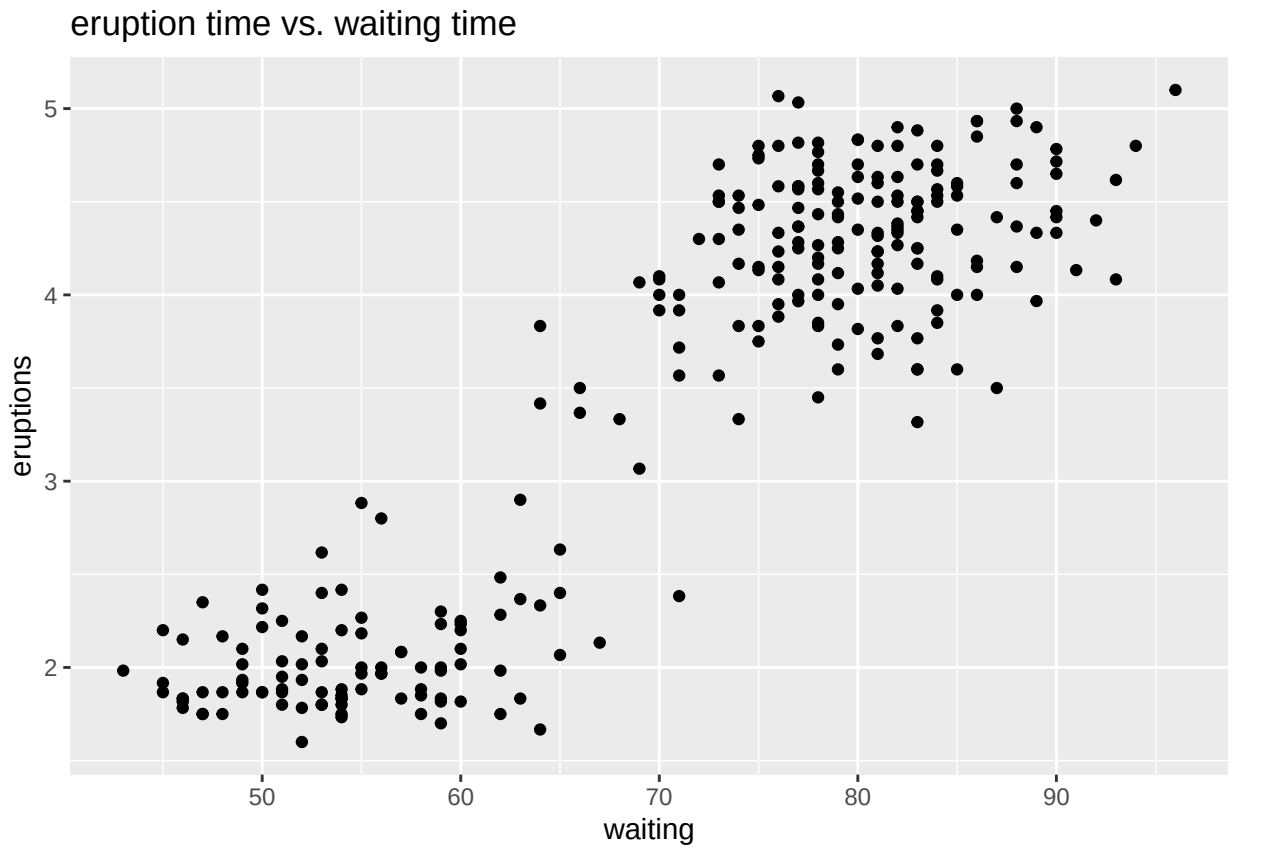
```
## Skim summary statistics
##  n obs: 272
##  n variables: 2
##
## -- Variable type:numeric ------------------------------------------------------------------------
```

```
##   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
##
##
```

We have a total of 272 observations of each variable. Reading the documentation, `waiting` is the time between each eruption event (? units) and `eruptions` is the duration of the eurption (in minutes). Regardless, both `waiting` and `eruptions` seem to have a bimodal distribution.

Next I do a simple scatter plot of the observations

```
faithful %>%
  ggplot(aes(x = waiting, y = eruptions)) +
  geom_point()+
  labs(title = "eruption time vs. waiting time")
```



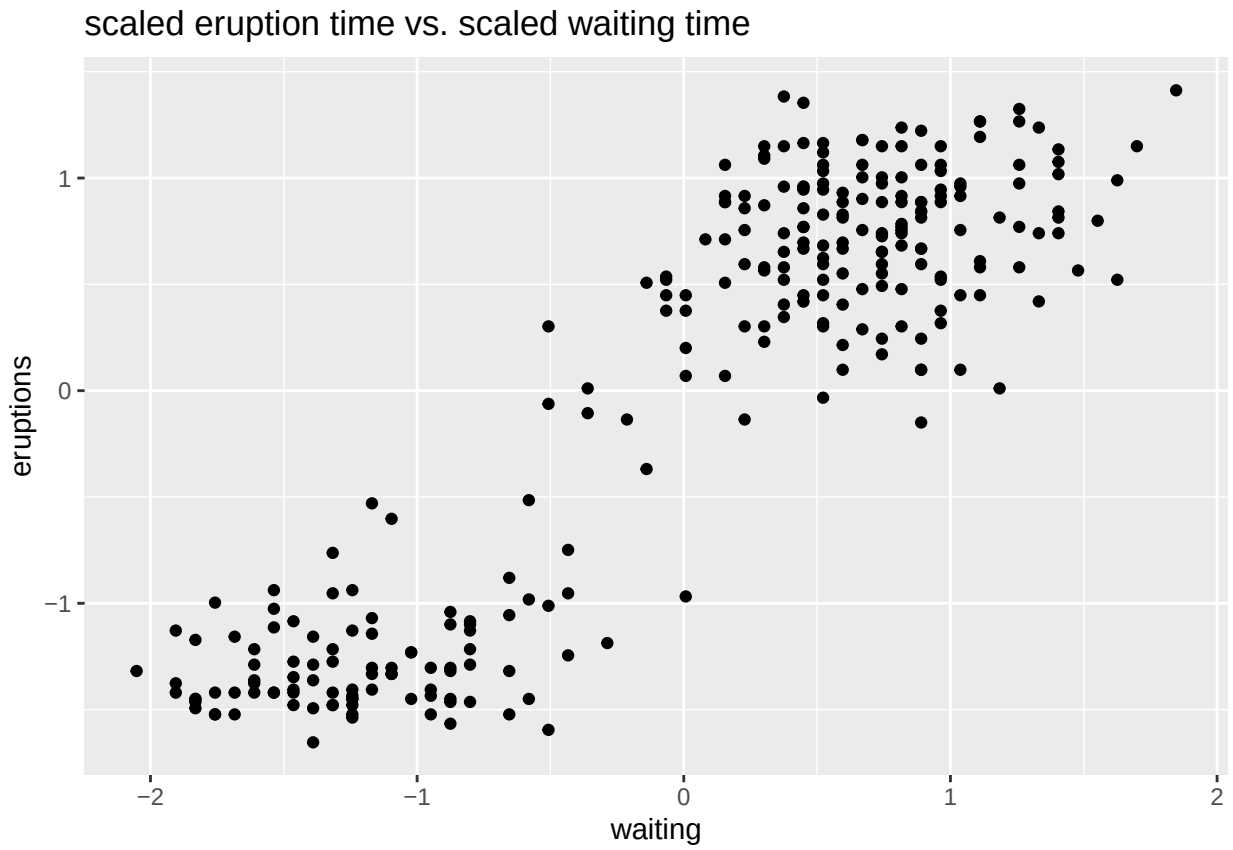
Assuming that both scales are in minutes, we seem to have two clear clusters of eurption events. If the waiting time is shorter (50-65 min) then the eruption lasts around 2 minutes. If the waiting time if longer, the eruption last longer. While there is a clear positive correlation between waiting time and eruption duration, there is a paucity of data from 65-75 minutes, suggesting something is separating these two groups clearly.

From this plot I believe we can safely conclude that a linear model of data generation like $eruptionTime = \beta_0 + \beta_1 * waiting + \epsilon$ is insufficient to fully describe what is going on.

I was curious to see how scaling (which we do in the next step) would effect the plot

```
faithful %>%
  scale() %>%
```

```
  as_tibble() %>%
  ggplot(aes(x = waiting, y = eruptions)) +
  geom_point() +
  labs(title = "scaled eruption time vs. scaled waiting time")
```



Looks exactly the same, which makes sens because `scale` just performs the same linear transformation on each axises (mean centers each column and divides each column value by it's standard deviation), which shouldn't effect the relative position of each point

5. Calculate a dissimilarity matrix of these data.

I will scale the data first with `scale`

```
faith_dist <- faithful %>%
  scale() %>%
  dist()

length(faith_dist)
```

```
## [1] 36856
```

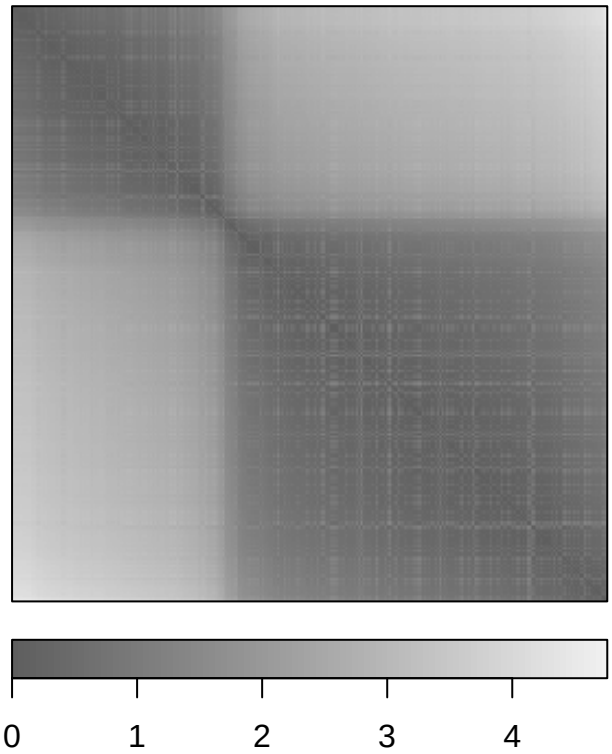
I calculate the distance matrix which is 272 x 272. This matrix should have 73984 entries, but the calculated object only has 36856 entries. The explanaiton is for this is that the matrix is symmetric and all the diagonal elements are zero, so R only returns (73984 - 272)/2 = 36856 important non-zero unique entries (i.e. $n * (n-1)/2$)

Note that I used euclidean distance which is the default distance metric in `dist()`

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

```
seriation::dissplot(faith_dist)
```

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



The Ordered Dissimilarity Image (ODI) demonstrates two clear clusters of observations.

7. Using any munging tools you'd like (e.g., dplyr from the Tidyverse), create a subset of the `iris` data excluding the species feature

```
iris_no_species <- iris %>%
  select(-Species)


iris_no_species
```

```
##     Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1            5.1         3.5          1.4         0.2
## 2            4.9         3.0          1.4         0.2
## 3            4.7         3.2          1.3         0.2
## 4            4.6         3.1          1.5         0.2
## 5            5.0         3.6          1.4         0.2
## 6            5.4         3.9          1.7         0.4
## 7            4.6         3.4          1.4         0.3
## 8            5.0         3.4          1.5         0.2
## 9            4.4         2.9          1.4         0.2
```

```
## 10          4.9         3.1         1.5         0.1
## 11          5.4         3.7         1.5         0.2
## 12          4.8         3.4         1.6         0.2
## 13          4.8         3.0         1.4         0.1
## 14          4.3         3.0         1.1         0.1
## 15          5.8         4.0         1.2         0.2
## 16          5.7         4.4         1.5         0.4
## 17          5.4         3.9         1.3         0.4
## 18          5.1         3.5         1.4         0.3
## 19          5.7         3.8         1.7         0.3
## 20          5.1         3.8         1.5         0.3
## 21          5.4         3.4         1.7         0.2
## 22          5.1         3.7         1.5         0.4
## 23          4.6         3.6         1.0         0.2
## 24          5.1         3.3         1.7         0.5
## 25          4.8         3.4         1.9         0.2
## 26          5.0         3.0         1.6         0.2
## 27          5.0         3.4         1.6         0.4
## 28          5.2         3.5         1.5         0.2
## 29          5.2         3.4         1.4         0.2
## 30          4.7         3.2         1.6         0.2
## 31          4.8         3.1         1.6         0.2
## 32          5.4         3.4         1.5         0.4
## 33          5.2         4.1         1.5         0.1
## 34          5.5         4.2         1.4         0.2
## 35          4.9         3.1         1.5         0.2
## 36          5.0         3.2         1.2         0.2
## 37          5.5         3.5         1.3         0.2
## 38          4.9         3.6         1.4         0.1
## 39          4.4         3.0         1.3         0.2
## 40          5.1         3.4         1.5         0.2
## 41          5.0         3.5         1.3         0.3
## 42          4.5         2.3         1.3         0.3
## 43          4.4         3.2         1.3         0.2
## 44          5.0         3.5         1.6         0.6
## 45          5.1         3.8         1.9         0.4
## 46          4.8         3.0         1.4         0.3
## 47          5.1         3.8         1.6         0.2
## 48          4.6         3.2         1.4         0.2
## 49          5.3         3.7         1.5         0.2
## 50          5.0         3.3         1.4         0.2
## 51          7.0         3.2         4.7         1.4
## 52          6.4         3.2         4.5         1.5
## 53          6.9         3.1         4.9         1.5
## 54          5.5         2.3         4.0         1.3
## 55          6.5         2.8         4.6         1.5
## 56          5.7         2.8         4.5         1.3
## 57          6.3         3.3         4.7         1.6
## 58          4.9         2.4         3.3         1.0
## 59          6.6         2.9         4.6         1.3
## 60          5.2         2.7         3.9         1.4
## 61          5.0         2.0         3.5         1.0
## 62          5.9         3.0         4.2         1.5
## 63          6.0         2.2         4.0         1.0
```

```
## 64          6.1         2.9         4.7         1.4
## 65          5.6         2.9         3.6         1.3
## 66          6.7         3.1         4.4         1.4
## 67          5.6         3.0         4.5         1.5
## 68          5.8         2.7         4.1         1.0
## 69          6.2         2.2         4.5         1.5
## 70          5.6         2.5         3.9         1.1
## 71          5.9         3.2         4.8         1.8
## 72          6.1         2.8         4.0         1.3
## 73          6.3         2.5         4.9         1.5
## 74          6.1         2.8         4.7         1.2
## 75          6.4         2.9         4.3         1.3
## 76          6.6         3.0         4.4         1.4
## 77          6.8         2.8         4.8         1.4
## 78          6.7         3.0         5.0         1.7
## 79          6.0         2.9         4.5         1.5
## 80          5.7         2.6         3.5         1.0
## 81          5.5         2.4         3.8         1.1
## 82          5.5         2.4         3.7         1.0
## 83          5.8         2.7         3.9         1.2
## 84          6.0         2.7         5.1         1.6
## 85          5.4         3.0         4.5         1.5
## 86          6.0         3.4         4.5         1.6
## 87          6.7         3.1         4.7         1.5
## 88          6.3         2.3         4.4         1.3
## 89          5.6         3.0         4.1         1.3
## 90          5.5         2.5         4.0         1.3
## 91          5.5         2.6         4.4         1.2
## 92          6.1         3.0         4.6         1.4
## 93          5.8         2.6         4.0         1.2
## 94          5.0         2.3         3.3         1.0
## 95          5.6         2.7         4.2         1.3
## 96          5.7         3.0         4.2         1.2
## 97          5.7         2.9         4.2         1.3
## 98          6.2         2.9         4.3         1.3
## 99          5.1         2.5         3.0         1.1
## 100         5.7         2.8         4.1         1.3
## 101         6.3         3.3         6.0         2.5
## 102         5.8         2.7         5.1         1.9
## 103         7.1         3.0         5.9         2.1
## 104         6.3         2.9         5.6         1.8
## 105         6.5         3.0         5.8         2.2
## 106         7.6         3.0         6.6         2.1
## 107         4.9         2.5         4.5         1.7
## 108         7.3         2.9         6.3         1.8
## 109         6.7         2.5         5.8         1.8
## 110         7.2         3.6         6.1         2.5
## 111         6.5         3.2         5.1         2.0
## 112         6.4         2.7         5.3         1.9
## 113         6.8         3.0         5.5         2.1
## 114         5.7         2.5         5.0         2.0
## 115         5.8         2.8         5.1         2.4
## 116         6.4         3.2         5.3         2.3
## 117         6.5         3.0         5.5         1.8
```

```
## 118          7.7         3.8          6.7         2.2
## 119          7.7         2.6          6.9         2.3
## 120          6.0         2.2          5.0         1.5
## 121          6.9         3.2          5.7         2.3
## 122          5.6         2.8          4.9         2.0
## 123          7.7         2.8          6.7         2.0
## 124          6.3         2.7          4.9         1.8
## 125          6.7         3.3          5.7         2.1
## 126          7.2         3.2          6.0         1.8
## 127          6.2         2.8          4.8         1.8
## 128          6.1         3.0          4.9         1.8
## 129          6.4         2.8          5.6         2.1
## 130          7.2         3.0          5.8         1.6
## 131          7.4         2.8          6.1         1.9
## 132          7.9         3.8          6.4         2.0
## 133          6.4         2.8          5.6         2.2
## 134          6.3         2.8          5.1         1.5
## 135          6.1         2.6          5.6         1.4
## 136          7.7         3.0          6.1         2.3
## 137          6.3         3.4          5.6         2.4
## 138          6.4         3.1          5.5         1.8
## 139          6.0         3.0          4.8         1.8
## 140          6.9         3.1          5.4         2.1
## 141          6.7         3.1          5.6         2.4
## 142          6.9         3.1          5.1         2.3
## 143          5.8         2.7          5.1         1.9
## 144          6.8         3.2          5.9         2.3
## 145          6.7         3.3          5.7         2.5
## 146          6.7         3.0          5.2         2.3
## 147          6.3         2.5          5.0         1.9
## 148          6.5         3.0          5.2         2.0
## 149          6.2         3.4          5.4         2.3
## 150          5.9         3.0          5.1         1.8
```

scale the features and calculate a dissimilarity matrix

```
iris_dist <- iris_no_species %>%
  scale() %>%
  dist()

length(iris_dist)
```

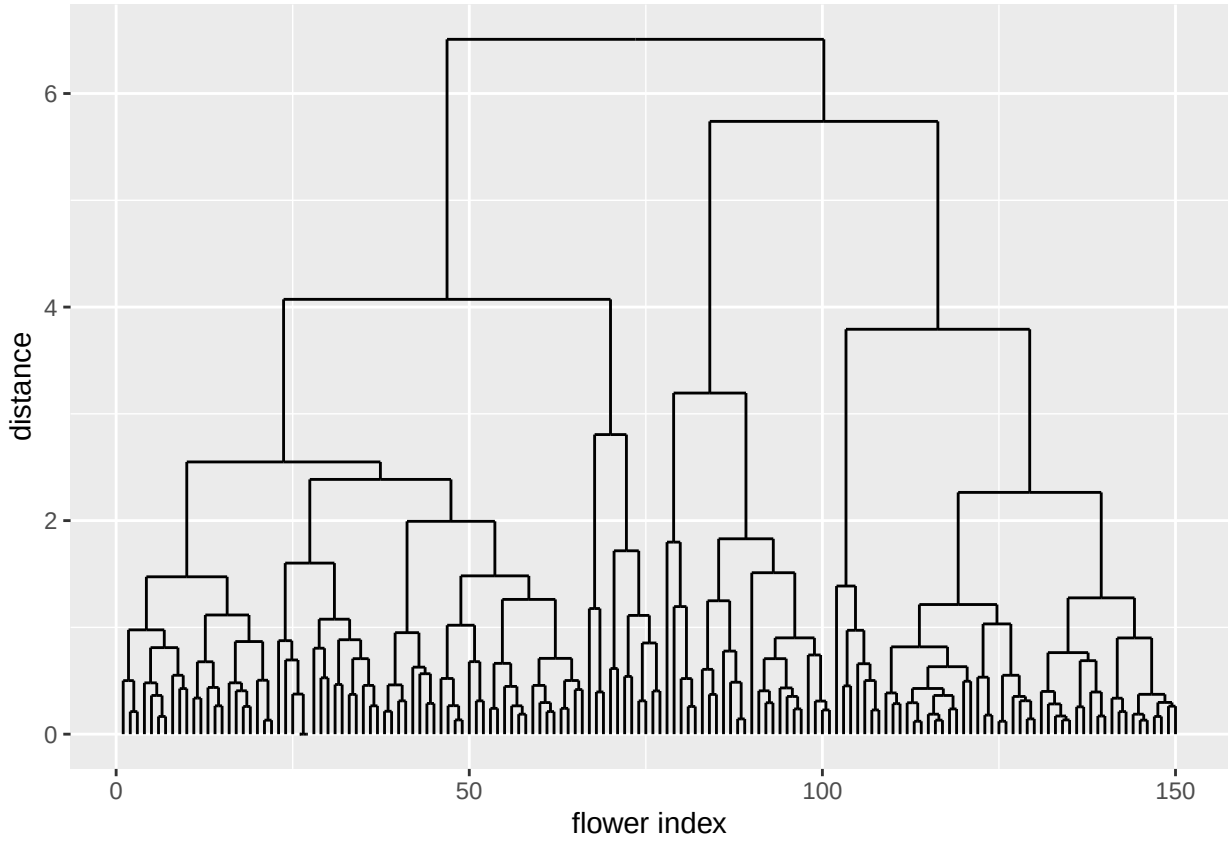
```
## [1] 11175
```

We get the expected $n * (n-1)/2$ number of unique non-zero entries

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_dist,
                      method = "complete")
```

```
ggdendro::dendro_data(hc_complete)$segments %>%
  ggplot() +
  geom_segment(aes(x = x, y = y, xend = xend, yend = yend)) +
  labs(y = "distance", x = "flower index")
```



early on the tree (distance just below 6) could be cut at $k = 3$. Another natural cut would be just below a distance of 4, yielding $k = 5$.

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

```
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 24  0
##           2  0  0 77
```

Looks like Cluster 1 in the $k = 2$ cut gets divided pretty significantly into clusters of 49 and 24 observations. Cluster 2 in the $k = 2$ cut stays intact (becoming cluster #3 in the $k = 3$ cut). this suggests that 3 cuts are probably better than 2.

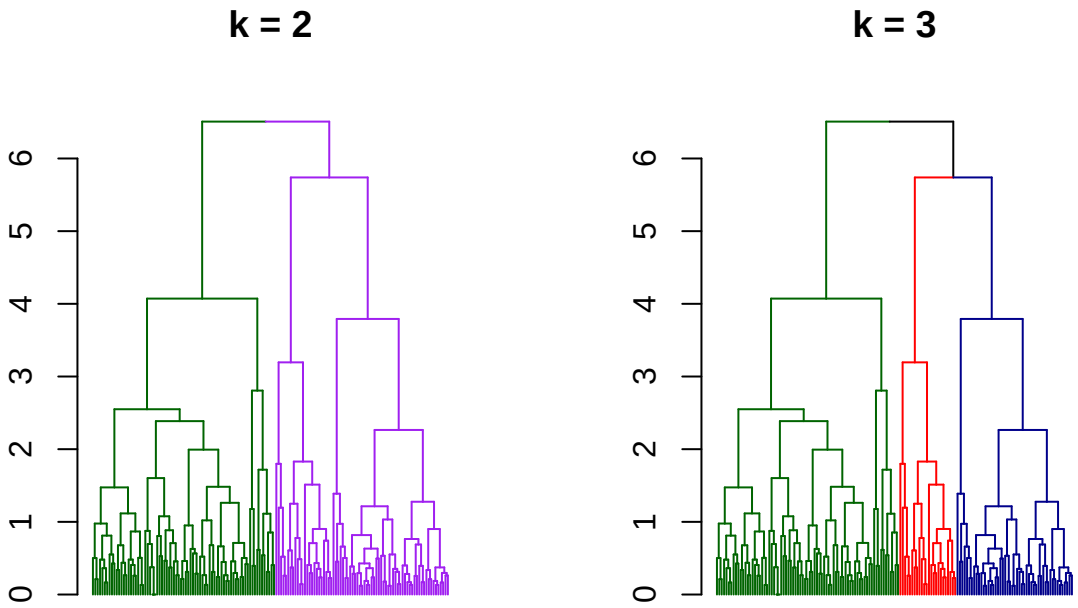
To plot the cuts with color, I use the `dendextend` package

```
dend <-  iris_no_species %>%
  scale() %>%
  dist() %>%
  hclust(method = "complete") %>%
  as.dendrogram()

par(mfrow = c(1, 2))

dend %>%
  set("branches_k_color",  value = c("darkgreen", "purple"), k = 2)  %>%
  set("labels", NA ) %>%
  plot(main = "k = 2")

dend %>%
  set("branches_k_color",  value = c("darkgreen", "red", "darkblue"), k = 3)  %>%
  set("labels", NA ) %>%
  plot(main = "k = 3")
```

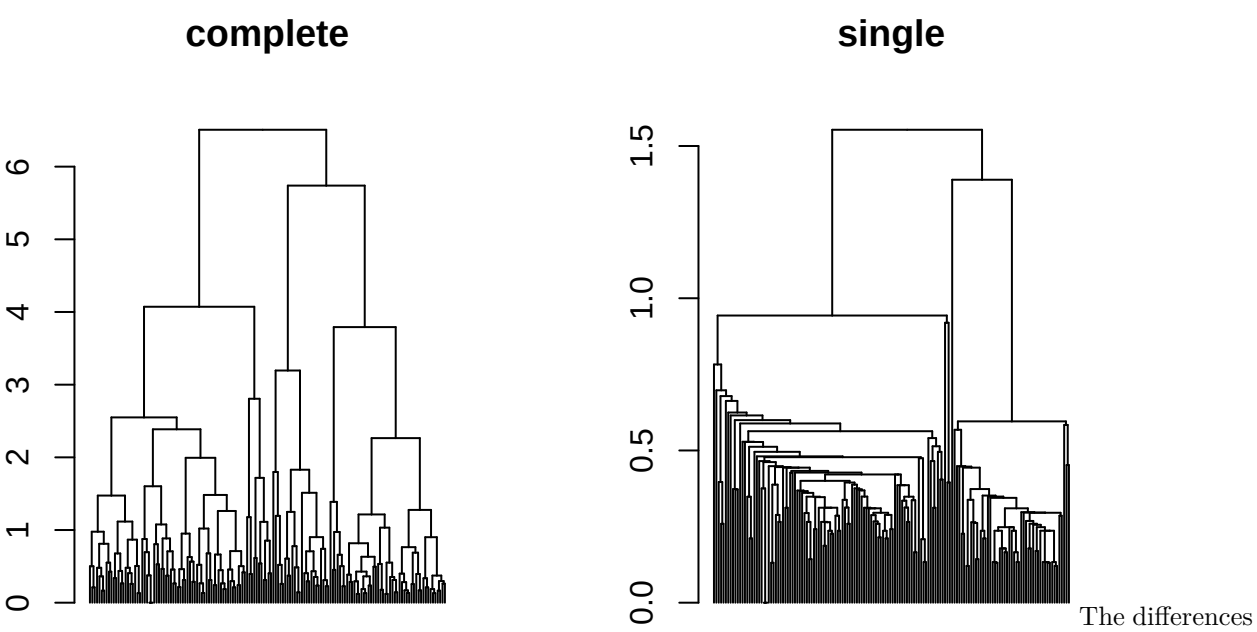


10. Now fit the algorithm using single and complete linkage and present each dendrogram side-by-side. Discuss the differences. What effects can we see in the clustering patterns when using different linkage methods?

```
par(mfrow = c(1, 2))

iris_no_species %>%
  scale() %>%
  dist() %>%
  hclust(method = "complete") %>%
  as.dendrogram() %>%
  set("labels", NA ) %>%
  plot(main = "complete")
```

```
iris_no_species %>%
  scale() %>%
  dist() %>%
  hclust(method = "single") %>%
  as.dendrogram() %>%
  set("labels", NA ) %>%
  plot(main = "single")
```



The differences between the complete and single methods are quite striking here. the complete method joins the two clusters with the smallest distance between their elements that are furthest apart, whereas the single method joins the two clusters that have the shortest distance between their closest elements.

Thus the single method is more likely to "chain" mainy roups with the same distance from single elements being close together, even though the groups may be far apart. Complete clustering finds more compact clusters of similar diameter, that is apparent in this example as well.

Interestingly despite the differing tree structure, the $k = 3$ cut groups seem to be relatively preserved, although the first group has more observations now

Critical Thinking

1. You just assessed the clusterability of some feature space $\mathbb{R}^n$. Address the following questions:

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

Before launching into techniques designed to diagnose clusterability (see my answer to b below), I would think about the nature of the data and possible models of the data generating process and ask

1. based on what we know about where the data came from, is it reasonable to suspect that the data may be clustered?
2. if the data are clustered, so what? What does the clustering add to our knowledge about the data?

To use a medical example, let's say there is reason to suspect disease X might not just be one distinct disease but instead multiple similar disease states with different causes and treatments. Proceeding to diagnosing clusterability in a high dimensional clinical dataset of patient with disease X in this context makes a lot of sense.

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

1. exploratory data analysis (EDA). Like in the old faithful example, a grid of scatter plots of different combinations of features may demonstrate a clear case for clustering. If the dimension n is very high, I would use a dimension reduction technique like principle component analysis and then perform scatter plots of the first two PCA components.
2. generate a distance matrix and then create a visual assessment of cluster tendency like the ordered dissimilarity matrix
3. Use the same distance matrix to calculate a hopkin's statistic and test the null hypothesis of spatial randomness

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

If EDA is highly suggestive of clustering, then ODI and the hopkin's statistic would just be confirmatory. However if EDA does not reveal a clear clustering pattern, then ODI and hopkin's statistic may suggest clustering where it was not readily apparent from the EDA.

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

I think proceeding would depend on your prior belief that clusters could be involved in the data generation process. If it was strong enough I would proceed regardless of the results of the diagnostic tests for clusterability, especially if the results of clustering could answer a clear research hypothesis.

On the other hand, if I was skeptical about the clusterability of the data based on what was known about the data generating process, negative diagnostics for clusterability would convince me to stop.

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

a. Describe the author(s) process.

I read Johnson, S. C. (1967). "Hierarchical clustering schemes." Psychometrika, 32(3), 241-254. Tis seems like a foundational paper in the hierarchial agglomerative clustering field. the authors set out to formally define the mechanics of agglomerative clustering and work through an example in painstanking detail to demonstrate them. They specifically demonstrate the complete (maximum) and single (minimum) linkage methods.

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 did not appear to do any clusterability diagnositics, but did follow similar steps after that point including

1. create a distance matrix using all $n * (n-1)/2$ similarity measures
2. initiate with singletons
3. form a cluster according to linkage method (either single or complete in their case)
4. record distance between clusters on linkage method scale
5. repeat steps 3-4 until only one cluster with all the observations exists

They omit the other linkage methods we learned about, as they think the single = "connectedness" and complete = "compactness" have the clearest interpreations.

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

The authors show the distance metric used in the agglomerative hierarchial cluster scheme linkage method need to statisfy not just the standard triangle inequality $d(x,z)d(x,y)+d(y,z)$ but the stronger "ultrametric inequality" that for 3 points x, y, z in clusters $x, y \in C_i$ and $y, z \in C_j$ then $d(x,z) \max[d(x,y), d(y,z)]$. They suggest an unsolved problem is finding the "closest" metric that statisfies the ultrametric inequality.