Unsupervised Learning

Brady Lamson

2022-04-04

Exercise 1

```
my.data <- data.frame(X1 = c(3, 5, 4, 7),
                      X2 = c(6, 4, 9, 9),
                      X3 = c(1, 7, 2, 1))
rownames(my.data) <- c("Obs1", "Obs2", "Obs3", "Obs4")

my.data_dist <- dist(my.data, method = "euclidean")
my.data_dist
```

```
##          Obs1     Obs2     Obs3
## Obs2 6.633250
## Obs3 3.316625 7.141428
## Obs4 5.000000 8.062258 3.162278
```

a) The distance between Obs1 and Obs 2 is **6.633**.

b) Obs4 and Obs3 have a distance of **3.16** which appears to be the shortest.

c) Obs4 and Obs3 would be merged in the first step.

Exercise 2

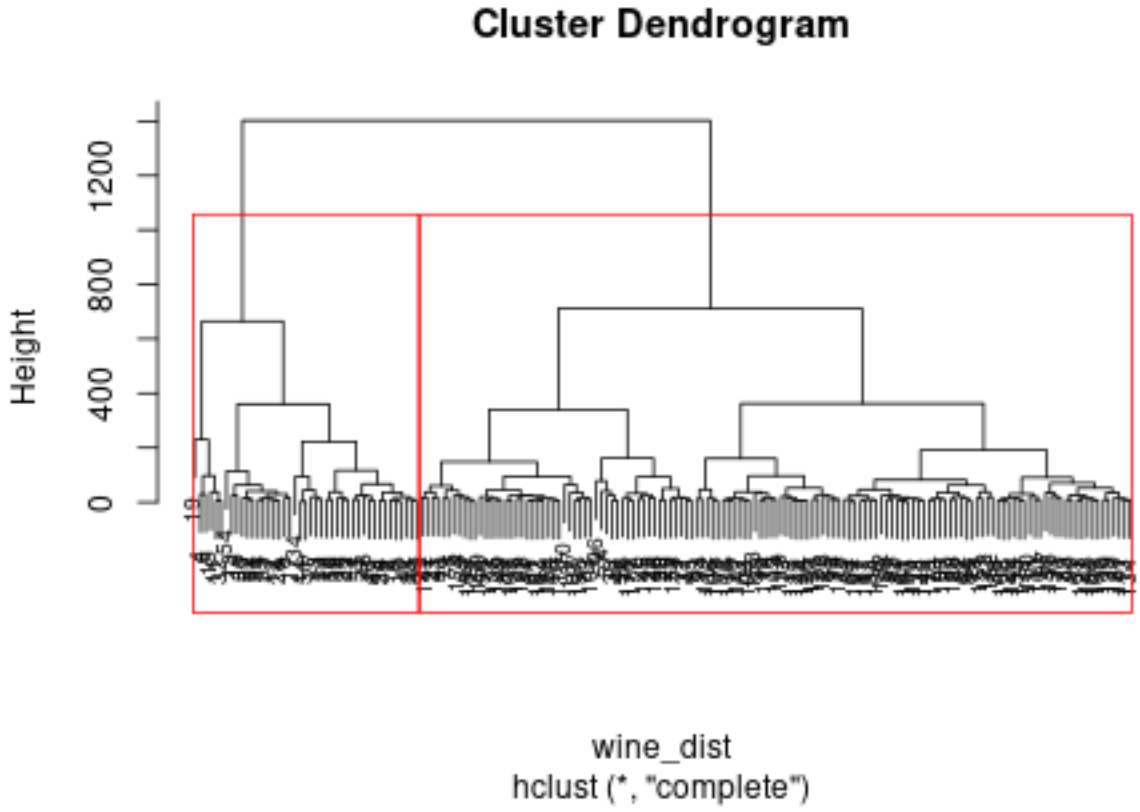
```
arr_dist <- dist(USArrests, method = "euclidean")
#arr_dist
```

The distance between **Florida** and **Alabam** is 102.001618.

Exercise 3

```
wine <- rattle::wine %>% select(-Type)
```

```
wine_dist <- dist(wine, method = "euclidean")
wine_hclust <- hclust(wine_dist)
plot(wine_hclust, cex = 0.7)
rect.hclust(wine_hclust, k = 2, border = "red")
```




```
my_clusters <- cutree(wine_hclust, k = 2)
my_clusters %>% table()
```

```
## .
##   1   2
##  43 135
```

Exercise 4

```
my.x1 <- c(5.2, 4.6, 5.9, 6.8, 10.5, 10.7, 8.6, 10.5, 14.1, 16.4, 14.3, 12.4)

my.x2 <- c(3.6, 4.7, 2.2, 4.5, 7.2, 7.3, 7.1, 9.9, 6.3, 4.2, 6.2, 3.3)

my.data <- data.frame(x1 = my.x1, x2 = my.x2)
```

```r
# So that everyone has the same randomly selected starting cluster centers:
set.seed(27)

# Carry out the k means cluster analysis with k = 3:
my_kmclust <- kmeans(my.data, centers = 3)
my_kmclust$cluster %>% table()
```

```
## .
## 1 2 3
## 4 4 4
```

We have three clusters each with **4 observations** inside of them.

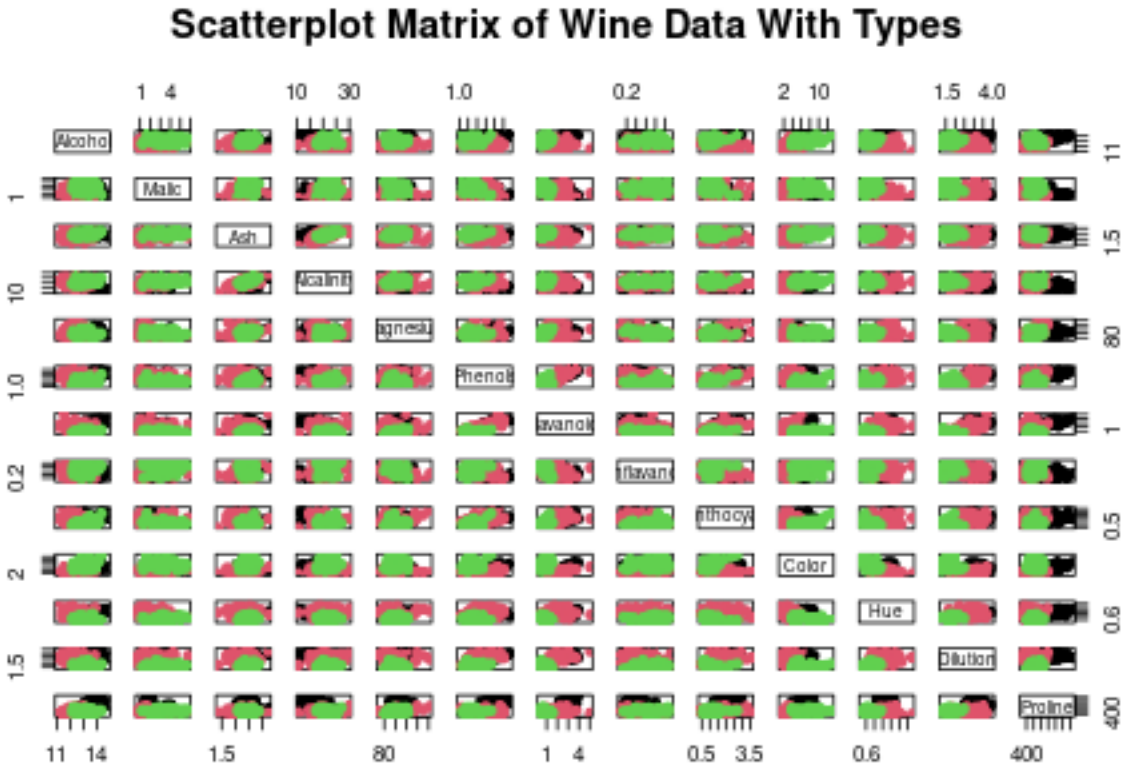
Exercise 5

```r
set.seed(20)

wine_kmclust <- kmeans(wine, centers = 3)
kmclusters <- wine_kmclust$cluster
```

```r
wine %>%
    pairs(
        col = kmclusters,
        main = "Scatterplot Matrix of Wine Data With Clusters",
        pch = 19
    )
```


```
wine %>%
    pairs(
        col = rattle::wine$Type,
        main = "Scatterplot Matrix of Wine Data With Types",
        pch = 19
    )
```



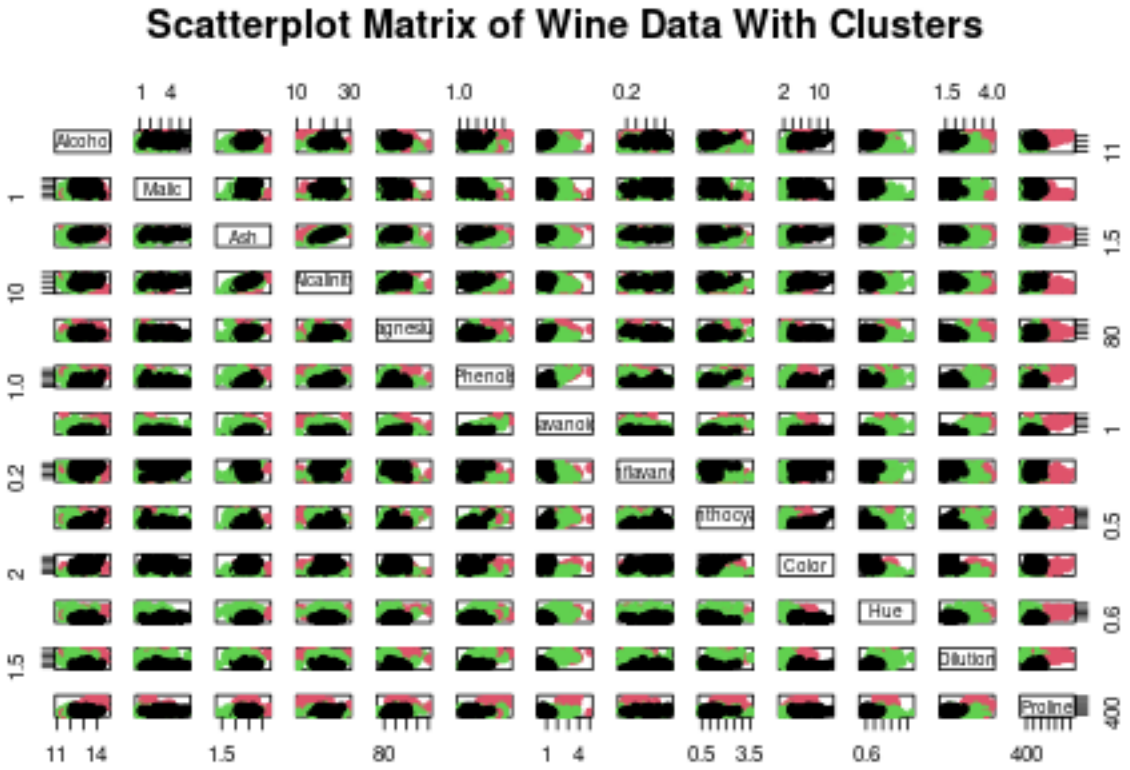
a) The clusters don't appear to correspond to type very well. With one exception, the bottom row does a fairly decent job of separating out the groups.

```
# Standardize each of the 13 variables:
wine2_std <- scale(wine, center = TRUE, scale = TRUE)

# So that everyone has the same randomly selected starting cluster centers:
set.seed(20)

# Carry out the k means cluster analysis with k = 3:
wine_kmclust_std <- kmeans(wine2_std, centers = 3)
```

```
my.clusters_std <- wine_kmclust_std$cluster
pairs(wine,
      col = my.clusters_std,
      main = "Scatterplot Matrix of Wine Data With Clusters",
      pch = 19)
```



Yes, the clusters here do seem to be corresponding to wine types.

Exercise 6

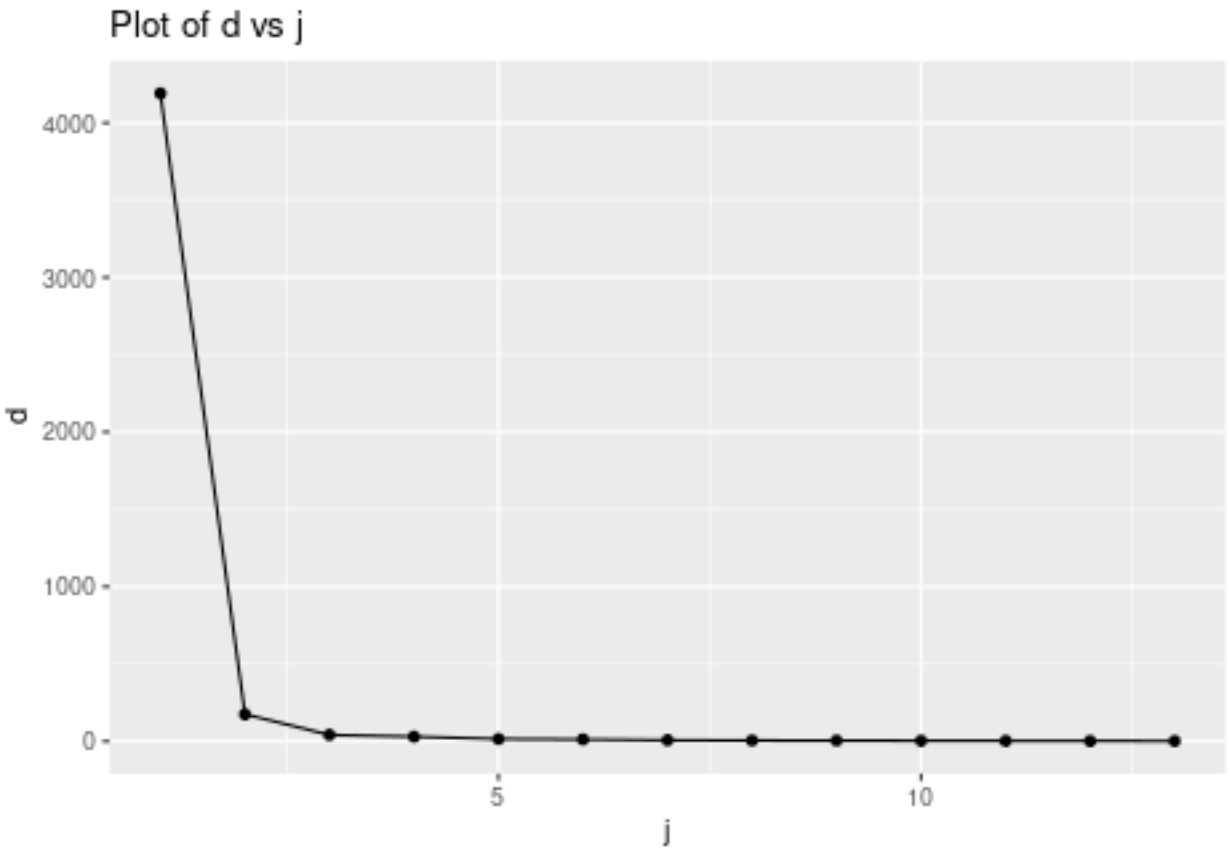
```
wine %>%
    summarise(across(
        everything(), list(mean = mean)
    ))
```

```
##   Alcohol_mean Malic_mean Ash_mean Alcalinity_mean Magnesium_mean Phenols_mean
## 1     13.00062   2.336348 2.366517        19.49494       99.74157     2.295112
##   Flavanoids_mean Nonflavanoids_mean Proanthocyanins_mean Color_mean  Hue_mean
## 1         2.02927          0.3618539             1.590899    5.05809 0.9574494
##   Dilution_mean Proline_mean
## 1      2.611685     746.8933
```

```
wine_cntr <-
    wine %>%
    scale(center = TRUE, scale = FALSE) %>%
    as.data.frame()

my_pca <-
    svd(wine_cntr)
```

```
ggplot(data = data.frame(d = my_pca$d, j = 1:13),
       mapping = aes(x = j, y = d)) +
    geom_point() +
    geom_line() +
    ggtitle("Plot of d vs j")
```




```
my_pca$d
```

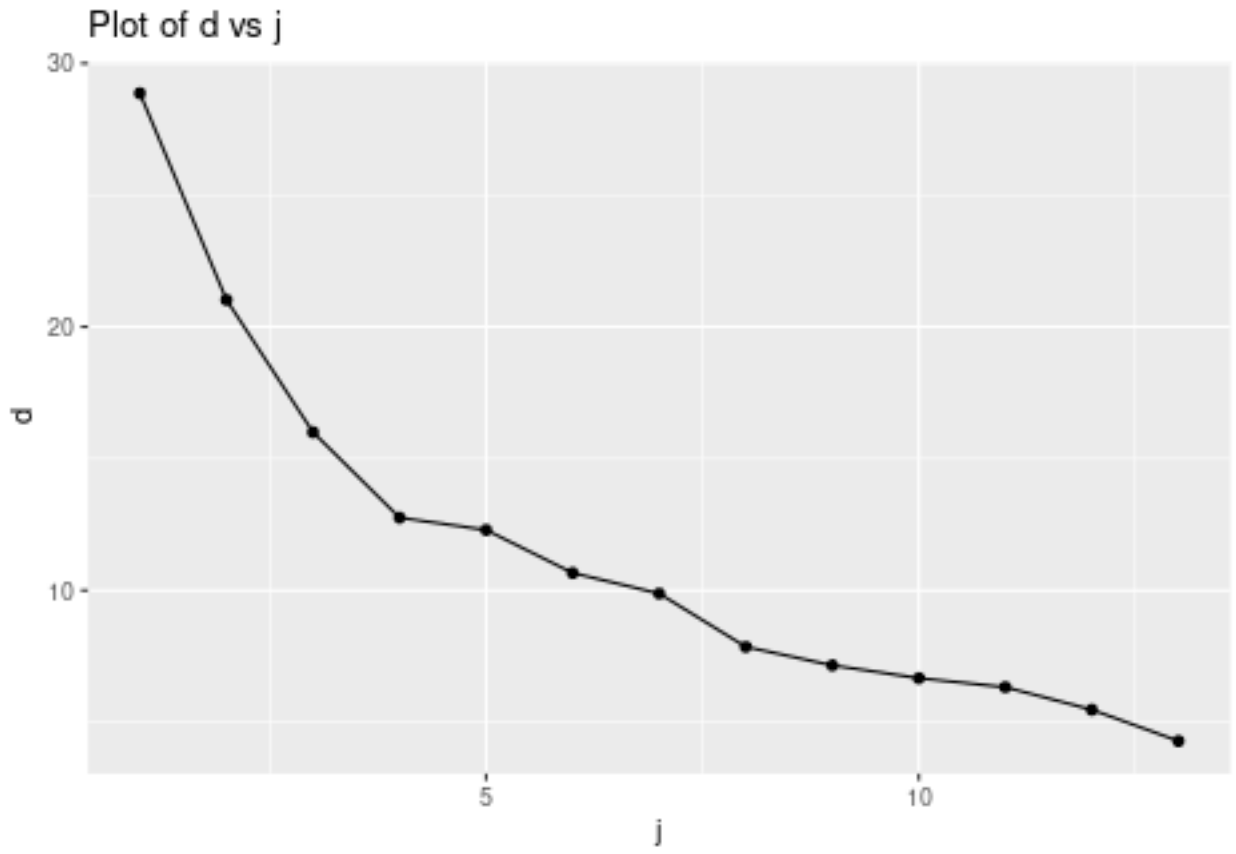
```
##  [1] 4190.312249  174.753375   40.872315   29.722695   14.748071   12.201160
##  [7]    7.026970    5.176339    4.454338    3.562494    2.578943    1.931271
## [13]    1.205013
```

What we can see from the vector is that most of the information is kept in the first 4 V_j's. What we need to decide here is what our cutoff is, what do we define as *"close to zero"*. In this case I would abandon all of the V_j's less than 5, keeping the first 8.

```
wine_cntr <-
    wine %>%
    scale(center = TRUE, scale = TRUE) %>%
    as.data.frame()

my_pca <-
    svd(wine_cntr)

ggplot(data = data.frame(d = my_pca$d, j = 1:13),
       mapping = aes(x = j, y = d)) +
    geom_point() +
    geom_line() +
    ggtitle("Plot of d vs j")
```




```
my_pca$d
```

```
##  [1] 28.860622 21.022948 15.998586 12.753760 12.289076 10.657077  9.875830
##  [8]  7.853918  7.150647  6.664063  6.321755  5.465559  4.277604
```

I'm still not 100% sure here, but I feel like a lot of the V_j's here are still quite valuable. If any, I would only abandon the bottom 2.

Exercise 7

```
virginica <-
    iris %>%
    filter(Species == "virginica") %>%
    select(-Species)

vir_cntr <-
    virginica %>%
    scale(center = TRUE, scale = FALSE) %>%
    as.data.frame()

my_pca <-
    svd(vir_cntr)

#-----[my_pca$v]------
my_pca$v
```

```
##            [,1]       [,2]       [,3]       [,4]
## [1,] 0.7410168 -0.1652590  0.5344502  0.3714117
## [2,] 0.2032877  0.7486428  0.3253749 -0.5406841
## [3,] 0.6278918 -0.1694278 -0.6515236 -0.3905934
## [4,] 0.1237745  0.6192880 -0.4289653  0.6458723
```

```
#-----[my_pca$d]------
my_pca$d
```

```
## [1] 5.836736 2.284953 1.600774 1.295773
```

V_1 would be reflecting length and V_2 width. I base this guess off of the derived variables. In V_1, the coefficients that would be tied to both length variables are high, in V_2, the higher coefficients are on what would be the width variables.