Class_Exercises_ClassNotes_7

Section 12.1 Exercises

Exercise 1: Do the following.

Code:

```
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
```

```
##      X1 X2 X3
## Obs1  3  6  1
## Obs2  5  4  7
## Obs3  4  9  2
## Obs4  7  9  1
```

```
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) What's the distance between Obs1 and Obs2?
The distance between obs1 and obs2 is 6.633250 units.

b) Which two observations are "closest" (least dissimilar) to each other?
The two observations closest to each other are obs3 and obs4.

c) Which two observations would be merged in the first step of a hierarchical clustering procedure?
The two observations that would merge together first in a hierarchal clustering procedure would be obs3 and obs4.

Exercise 2: What's the distance between Florida and Alabama?

Code:

```
arr_dist <- dist(USArrests, method = "euclidean")
head(arr_dist, n = 64)
```

```
##  [1]  37.17701  63.00833  46.92814  55.52477  41.93256 128.20694  16.80625
##  [8] 102.00162  25.84183 191.80305 116.76198  28.45488 123.34521 180.61010
## [15] 121.51987 127.28417  15.45445 154.14529  64.99362  91.64851  28.48543
## [22] 164.65096  27.39014  59.78829 127.39262 134.43697  37.43047 179.73620
## [29]  83.24302  51.64349  33.71083 101.96102 192.41614 117.38761  85.84870
## [36]  78.38686 131.08509  70.33811  44.18292 151.08911  48.34760  41.56609
## [43] 118.50270 190.37069  80.29533  92.82047 156.79241 183.77573  75.50709
## [50]  46.59249  77.19741  45.10222  66.47594 159.40656  45.18296  79.97450
## [57]  57.03026 221.19354 146.48498  42.91165 152.80409 209.98352 151.48020
## [64] 156.61204
```

The distance between Florida and Alabama is 102.001618 units.

Exercise 3: Do the following.

Code:

```
head(wine)
```

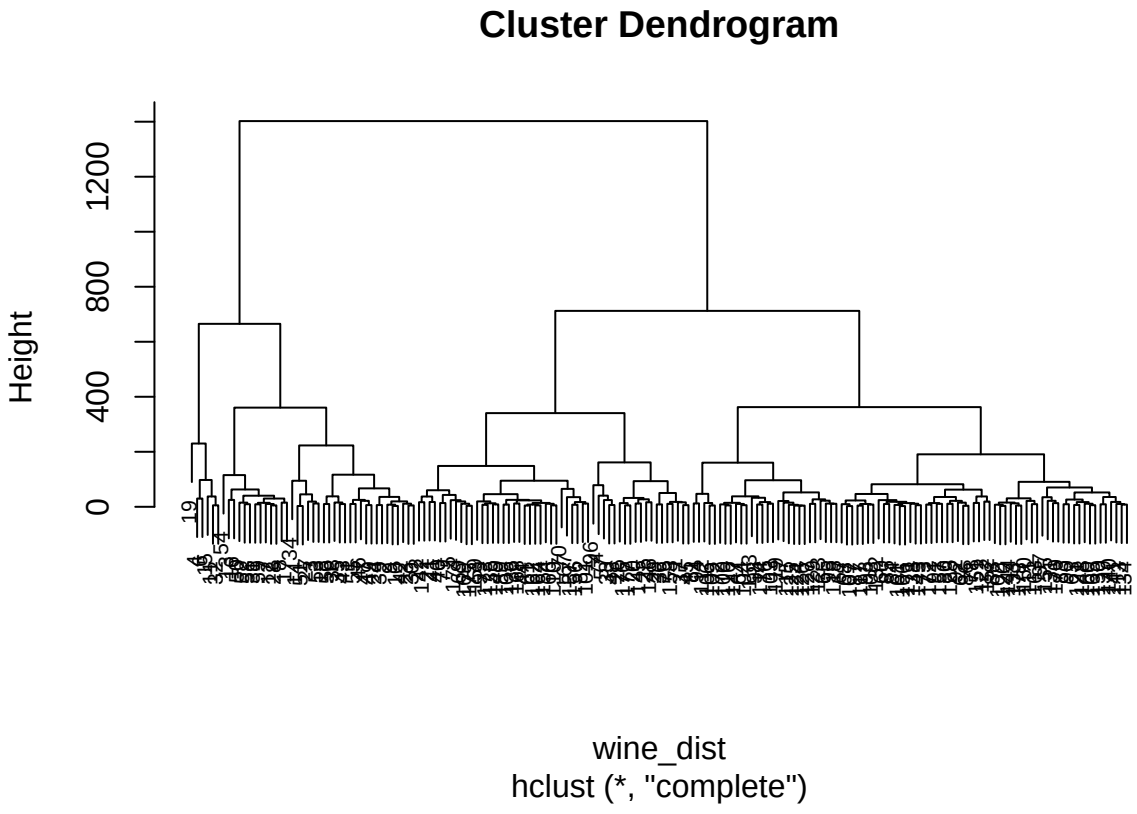
```
##   Type Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids
## 1    1   14.23  1.71 2.43       15.6       127    2.80       3.06          0.28
## 2    1   13.20  1.78 2.14       11.2       100    2.65       2.76          0.26
## 3    1   13.16  2.36 2.67       18.6       101    2.80       3.24          0.30
## 4    1   14.37  1.95 2.50       16.8       113    3.85       3.49          0.24
## 5    1   13.24  2.59 2.87       21.0       118    2.80       2.69          0.39
## 6    1   14.20  1.76 2.45       15.2       112    3.27       3.39          0.34
##   Proanthocyanins Color  Hue Dilution Proline
## 1            2.29  5.64 1.04     3.92    1065
## 2            1.28  4.38 1.05     3.40    1050
## 3            2.81  5.68 1.03     3.17    1185
## 4            2.18  7.80 0.86     3.45    1480
## 5            1.82  4.32 1.04     2.93     735
## 6            1.97  6.75 1.05     2.85    1450
```

```
wine2 <- select(wine, -Type)
wine_dist <- dist(wine2, method = "euclidean")
head(wine_dist, n = 50)
```

```
##  [1]  31.26501 122.83115 415.24540 330.17450 385.29596 227.12947 230.09160
##  [8]  36.11697  35.27498 445.55103 217.37854 257.81933  92.41867 482.66694
## [15] 245.47426 215.16396  66.27070 615.30369 220.28275 285.00238 296.08654
## [22]  39.76334  59.45623 222.22988 235.22247 134.38352 222.48121 151.38255
## [29]  43.16130 221.64285 450.50693  78.49627 170.12508  34.71830 147.57804
## [36] 185.79161  49.53119  53.60958 305.02053 270.18798  47.83494  39.71842
## [43] 385.76405 181.11954  22.34075  25.09466  84.12591  24.76023 195.96158
## [50]  92.00920
```

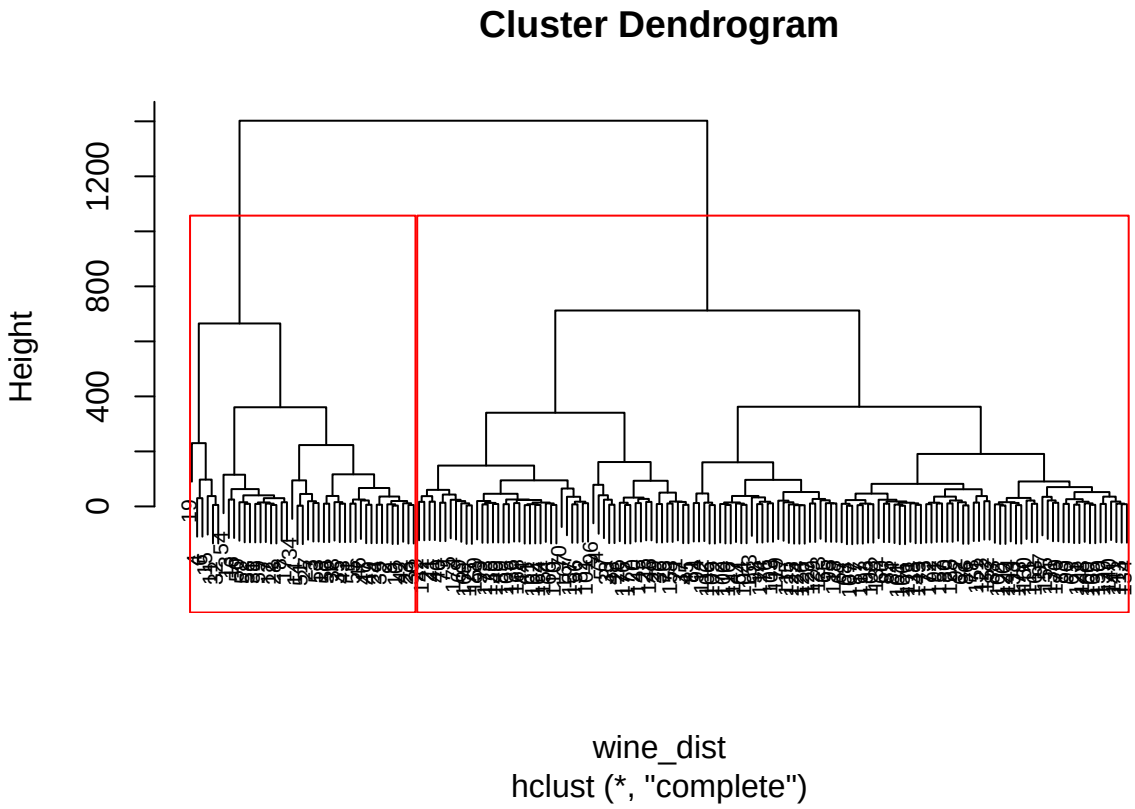
a) Now use wine_dist to carry out a hierarchical cluster analysis on the wines2 data set (which excludes Type), and produce the dendrogram. Report your R command(s).
Code:

```
wine_hclust <- hclust(wine_dist)
plot(wine_hclust, cex = 0.7)
```



b) Next, use rect.hclust() to plot red rectangles around k = 2 clusters in the dendrogram. Report your R command(s).
Code:

```
plot(wine_hclust, cex = 0.7)
rect.hclust(wine_hclust, k = 2, border = "red")
```



c) Finally, use cutree() to obtain k = 2 sets of observations (rows of the wines2 data frame) corresponding to the two clusters of part b. How many observations are in each of the two clusters?
Code:

```
wine.clusters <- cutree(wine_hclust, k = 2)
wine.clusters
```

```
##   [1] 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 1 1 2 2 1 1 2 1 1 1 2 1 1 2 2
##  [38] 1 1 2 2 1 1 2 2 1 1 2 1 1 1 1 1 1 1 1 2 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [112] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
## [149] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

```
wine.clust1 <- filter(wine2, wine.clusters == 1)
wine.clust2 <- filter(wine2, wine.clusters == 2)

head(wine.clust1, n = 8)
```

```
##   Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids
## 1   14.23  1.71 2.43       15.6       127    2.80       3.06          0.28
## 2   13.20  1.78 2.14       11.2       100    2.65       2.76          0.26
## 3   13.16  2.36 2.67       18.6       101    2.80       3.24          0.30
## 4   14.37  1.95 2.50       16.8       113    3.85       3.49          0.24
## 5   14.20  1.76 2.45       15.2       112    3.27       3.39          0.34
## 6   14.39  1.87 2.45       14.6        96    2.50       2.52          0.30
## 7   14.06  2.15 2.61       17.6       121    2.60       2.51          0.31
## 8   14.83  1.64 2.17       14.0        97    2.80       2.98          0.29
##   Proanthocyanins Color  Hue Dilution Proline
## 1            2.29  5.64 1.04     3.92    1065
## 2            1.28  4.38 1.05     3.40    1050
## 3            2.81  5.68 1.03     3.17    1185
## 4            2.18  7.80 0.86     3.45    1480
## 5            1.97  6.75 1.05     2.85    1450
## 6            1.98  5.25 1.02     3.58    1290
## 7            1.25  5.05 1.06     3.58    1295
## 8            1.98  5.20 1.08     2.85    1045
```

```
head(wine.clust2, n = 8)
```

```
##   Alcohol Malic  Ash Alcalinity Magnesium Phenols Flavanoids Nonflavanoids
## 1   13.24  2.59 2.87       21.0       118    2.80       2.69          0.39
## 2   13.64  3.10 2.56       15.2       116    2.70       3.03          0.17
## 3   14.06  1.63 2.28       16.0       126    3.00       3.17          0.24
## 4   12.93  3.80 2.65       18.6       102    2.41       2.41          0.25
## 5   13.50  1.81 2.61       20.0        96    2.53       2.61          0.28
## 6   13.05  2.05 3.22       25.0       124    2.63       2.68          0.47
## 7   13.87  1.90 2.80       19.4       107    2.95       2.97          0.37
## 8   13.68  1.83 2.36       17.2       104    2.42       2.69          0.42
##   Proanthocyanins Color  Hue Dilution Proline
## 1            1.82  4.32 1.04     2.93     735
## 2            1.66  5.10 0.96     3.36     845
## 3            2.10  5.65 1.09     3.71     780
## 4            1.98  4.50 1.03     3.52     770
## 5            1.66  3.52 1.12     3.82     845
## 6            1.92  3.58 1.13     3.20     830
## 7            1.76  4.50 1.25     3.40     915
## 8            1.97  3.84 1.23     2.87     990
```

There are 43 observations in cluster 1 and there are 135 observations in cluster 2.

Section 12.2 Exercises

Exercise 4: How many observations are in each of the three clusters (groups) identified by the k means procedure?

Code:

```
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)

# 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
```

```
## K-means clustering with 3 clusters of sizes 4, 4, 4
##
## Cluster means:
##       x1    x2
## 1 10.075 7.875
## 2  5.625 3.750
## 3 14.300 5.000
##
## Clustering vector:
##  [1] 2 2 2 2 1 1 1 1 3 3 3 3
##
## Within cluster sum of squares by cluster:
## [1]  8.4150  6.5775 14.7200
##  (between_SS / total_SS =  86.2 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

There are four observations in each cluster.

Exercise 5: Do the following.

a) Carry out a k means cluster analysis on the wine2 data set, with k = 3. Compare the scatterplot matrix showing the identified clusters with one showing the wine types. Do the clusters correspond to wine types?
Code:

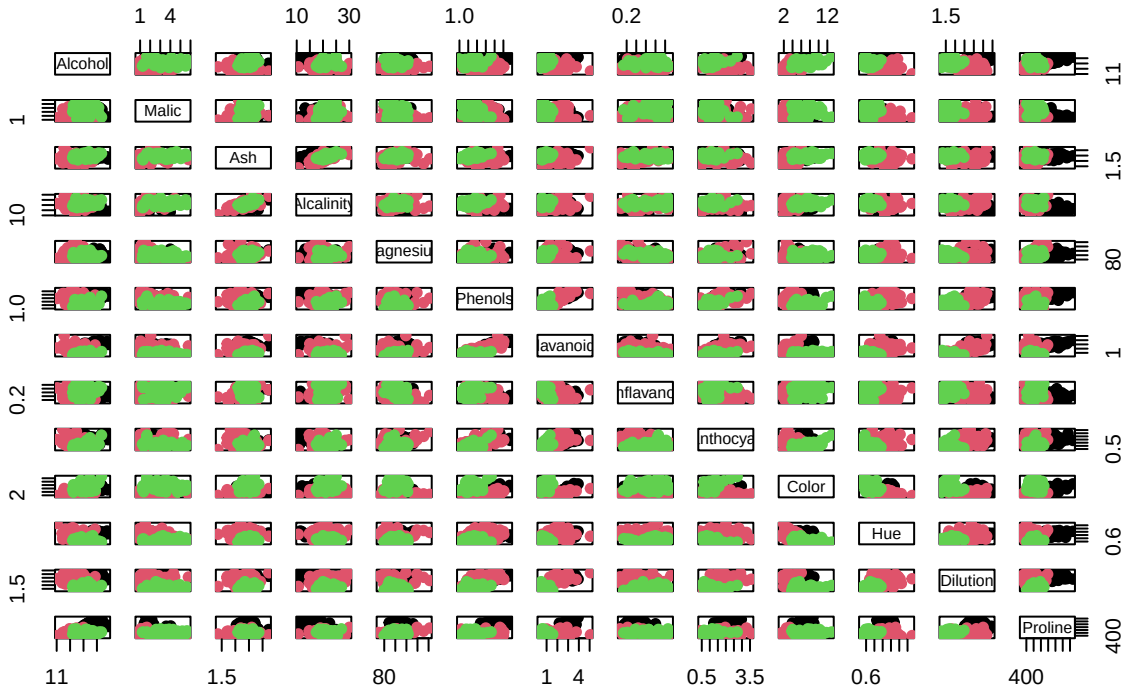
```
# So that everyone has the same randomly selected starting cluster centers:
set.seed(20)

# Carry out the k means cluster analysis with k = 3:
wine_kmclust <- kmeans(wine2, centers = 3)
wine_kmclust
```

```
## K-means clustering with 3 clusters of sizes 62, 69, 47
##
## Cluster means:
##    Alcohol    Malic      Ash Alcalinity Magnesium  Phenols Flavanoids
## 1 12.92984 2.504032 2.408065   19.89032 103.59677 2.111129   1.584032
## 2 12.51667 2.494203 2.288551   20.82319  92.34783 2.070725   1.758406
## 3 13.80447 1.883404 2.426170   17.02340 105.51064 2.867234   3.014255
##   Nonflavanoids Proanthocyanins    Color       Hue Dilution   Proline
## 1     0.3883871        1.503387 5.650323 0.8839677 2.365484  728.3387
## 2     0.3901449        1.451884 4.086957 0.9411594 2.490725  458.2319
## 3     0.2853191        1.910426 5.702553 1.0782979 3.114043 1195.1489
##
## Clustering vector:
##   [1] 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 1 1 1 3 3 1 1 3 3 1 3 3 3 3 3 3 1 1
##  [38] 3 3 1 1 3 3 1 1 3 3 3 3 3 3 3 3 3 3 3 3 3 3 2 1 2 1 2 2 1 2 2 1 1 1 2 2 3
##  [75] 1 2 2 2 1 2 2 1 1 2 2 2 2 2 1 1 2 2 2 2 2 1 1 2 1 2 1 2 2 2 1 2 2 2 2 1 2
## [112] 2 1 2 2 2 2 2 2 2 1 2 2 2 2 2 2 2 2 2 1 2 2 1 1 1 1 2 2 2 1 1 2 2 1 1 2 1
## [149] 1 2 2 2 2 1 1 1 2 1 1 1 2 1 2 1 1 2 1 1 1 1 2 2 1 1 1 1 1 2
##
## Within cluster sum of squares by cluster:
## [1]  566572.5  443166.7 1360950.5
##  (between_SS / total_SS =  86.5 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
my.clusters <- wine_kmclust$cluster
pairs(wine2,
      col = my.clusters,
      main = "Scatterplot Matrix of Wine Data With Clusters",
      pch = 19)
```

Scatterplot Matrix of Wine Data With Clusters

```
pairs(wine2,
      col = wine$Type,
      main = "Scatterplot Matrix of Wine Data With Types",
      pch = 19)
```



The clusters do not correspond to the types of wine.

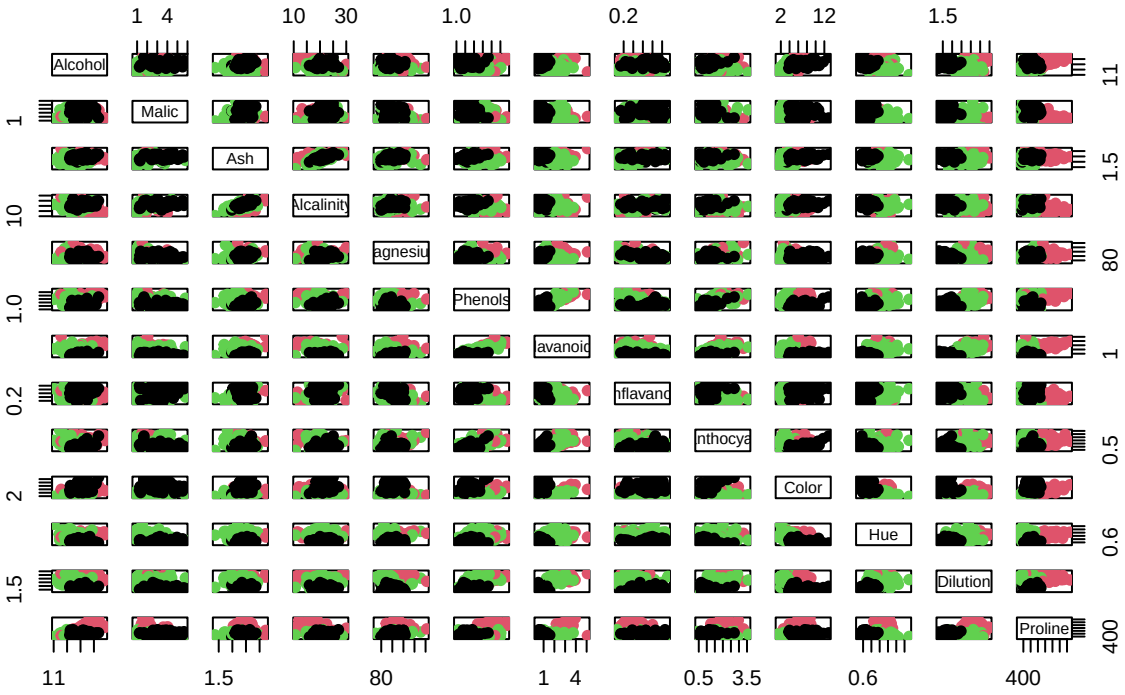
b) For cluster analysis, if the variables are measured on very different scales, it's best to standardize each one so that distances along each coordinate axis in p-dimensional space are comparable. Re-run the cluster analysis after standardizing each of the 13 variables in wine2. Now compare the scatterplot matrix showing the identified clusters with the one showing the wine Types.Now do the clusters correspond (at least approximately) to wine Types?
Code:

```
# Standardize each of the 13 variables:
wine2_std <- scale(wine2, 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)
wine_kmclust_std
```

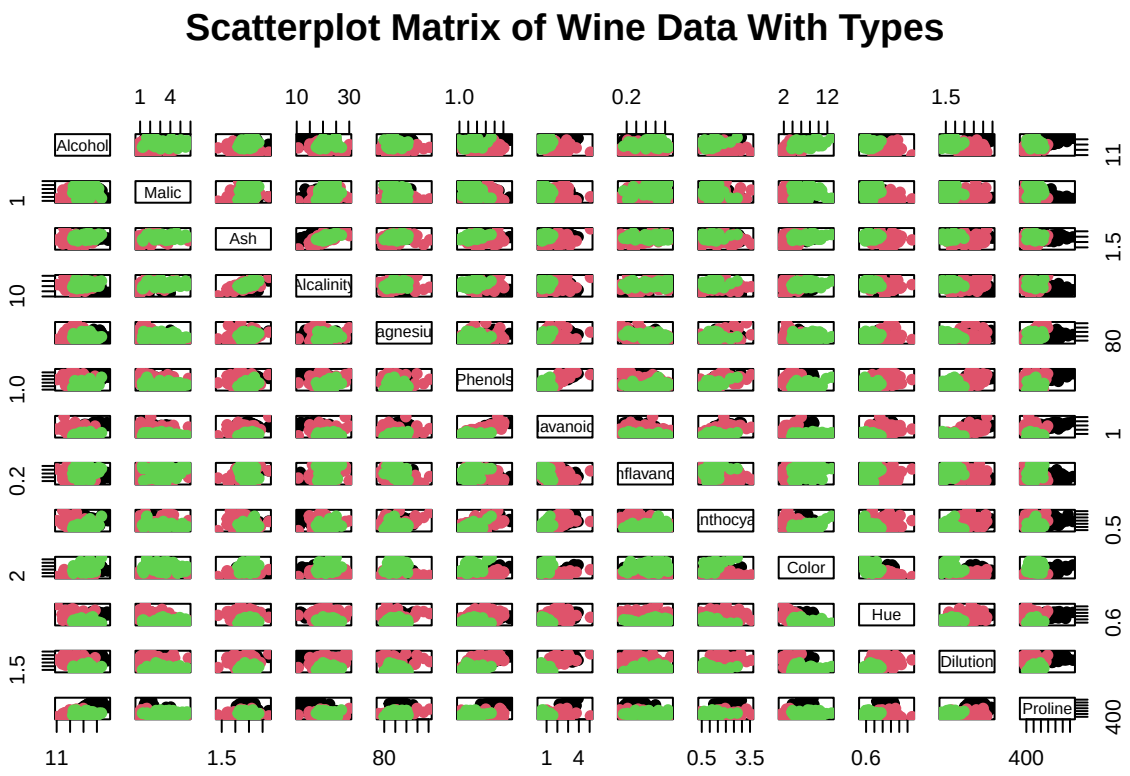
```
## K-means clustering with 3 clusters of sizes 51, 62, 65
```

```
##
## Cluster means:
##      Alcohol      Malic        Ash Alcalinity   Magnesium     Phenols
## 1  0.1644436  0.8690954  0.1863726  0.5228924 -0.07526047 -0.97657548
## 2  0.8328826 -0.3029551  0.3636801 -0.6084749  0.57596208  0.88274724
## 3 -0.9234669 -0.3929331 -0.4931257  0.1701220 -0.49032869 -0.07576891
##    Flavanoids Nonflavanoids Proanthocyanins      Color        Hue   Dilution
## 1 -1.21182921    0.72402116     -0.77751312  0.9388902 -1.1615122 -1.2887761
## 2  0.97506900   -0.56050853      0.57865427  0.1705823  0.4726504  0.7770551
## 3  0.02075402   -0.03343924      0.05810161 -0.8993770  0.4605046  0.2700025
##      Proline
## 1 -0.4059428
## 2  1.1220202
## 3 -0.7517257
##
## Clustering vector:
##   [1] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
##  [38] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 1 3 3 3 3 3 3 3 3 3 3 2
##  [75] 3 3 3 3 3 3 3 3 3 1 3 3 3 3 3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
## [112] 3 3 3 3 3 3 3 1 3 3 2 3 3 3 3 3 3 3 3 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [149] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## Within cluster sum of squares by cluster:
## [1] 326.3537 385.6983 558.6971
##  (between_SS / total_SS =  44.8 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
my.clusters_std <- wine_kmclust_std$cluster

pairs(wine2,
      col = my.clusters_std,
      main = "Scatterplot Matrix of Wine Data With Clusters",
      pch = 19)
```



```
pairs(wine2,
      col = wine$Type,
      main = "Scatterplot Matrix of Wine Data With Types",
      pch = 19)
```



The clusters correspond to the wine types.

Section 12.3 Exercises

Exercise 6: Do the following.

Setup:

```
wine2 <- select(wine,-Type)

summarise(wine2, 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
```

```r
wine2_cntr <-
  scale(wine2, center = TRUE, scale = FALSE) %>% as.data.frame()

head(wine2_cntr, n = 3)
```

```
##    Alcohol       Malic         Ash Alcalinity Magnesium   Phenols Flavanoids
## 1 1.229382 -0.62634831  0.06348315 -3.8949438 27.258427 0.5048876  1.0307303
## 2 0.199382 -0.55634831 -0.22651685 -8.2949438  0.258427 0.3548876  0.7307303
## 3 0.159382  0.02365169  0.30348315 -0.8949438  1.258427 0.5048876  1.2107303
##   Nonflavanoids Proanthocyanins      Color        Hue  Dilution  Proline
## 1   -0.08185393       0.6991011  0.5819101 0.08255056 1.3083146 318.1067
## 2   -0.10185393      -0.3108989 -0.6780899 0.09255056 0.7883146 303.1067
## 3   -0.06185393       1.2191011  0.6219101 0.07255056 0.5583146 438.1067
```

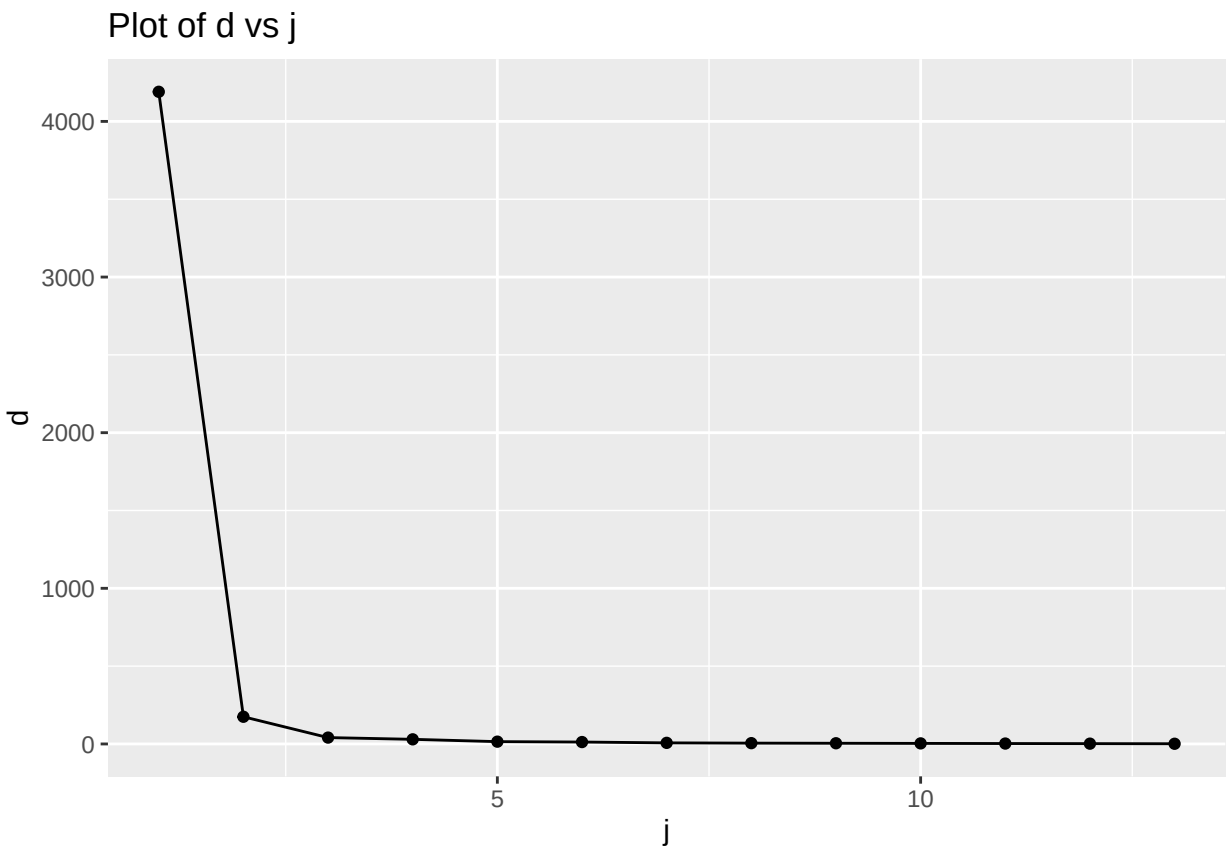
```r
my.pca <- svd(wine2_cntr)

head(my.pca$v, n = 5)
```

```
##               [,1]         [,2]        [,3]         [,4]         [,5]
## [1,] -0.0016592647 -0.001203406 -0.01687381  0.141446778  0.020336977
## [2,]  0.0006810156 -0.002154982 -0.12200337  0.160389543 -0.612883454
## [3,] -0.0001949057 -0.004593693 -0.05198743 -0.009772810  0.020175575
## [4,]  0.0046713006 -0.026450393 -0.93859300 -0.330965260  0.064352340
## [5,] -0.0178680075 -0.999344186  0.02978025 -0.005393756 -0.006149345
##              [,6]         [,7]         [,8]         [,9]         [,10]
## [1,] -0.194120104  0.923280337 -0.284820658 -0.086600612  2.245000e-03
## [2,] -0.742472963 -0.150109941  0.064674468 -0.015662138  1.850935e-02
## [3,] -0.041752912  0.045009549  0.149339532 -0.073649852  8.679965e-02
## [4,]  0.024065303  0.031526583 -0.015153912 -0.002044578 -3.554028e-03
## [5,]  0.001923782  0.001797363  0.003552212  0.001963668  4.051542e-05
##             [,11]         [,12]        [,13]
## [1,] -0.014971508 -0.0156514071  0.008029245
## [2,] -0.023187651  0.0672955455 -0.011090392
## [3,]  0.954010643 -0.1320630337 -0.173685673
## [4,] -0.052821695  0.0053938058  0.001939563
## [5,] -0.003024888  0.0006208885  0.002284536
```

```r
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
```

```r
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")
```



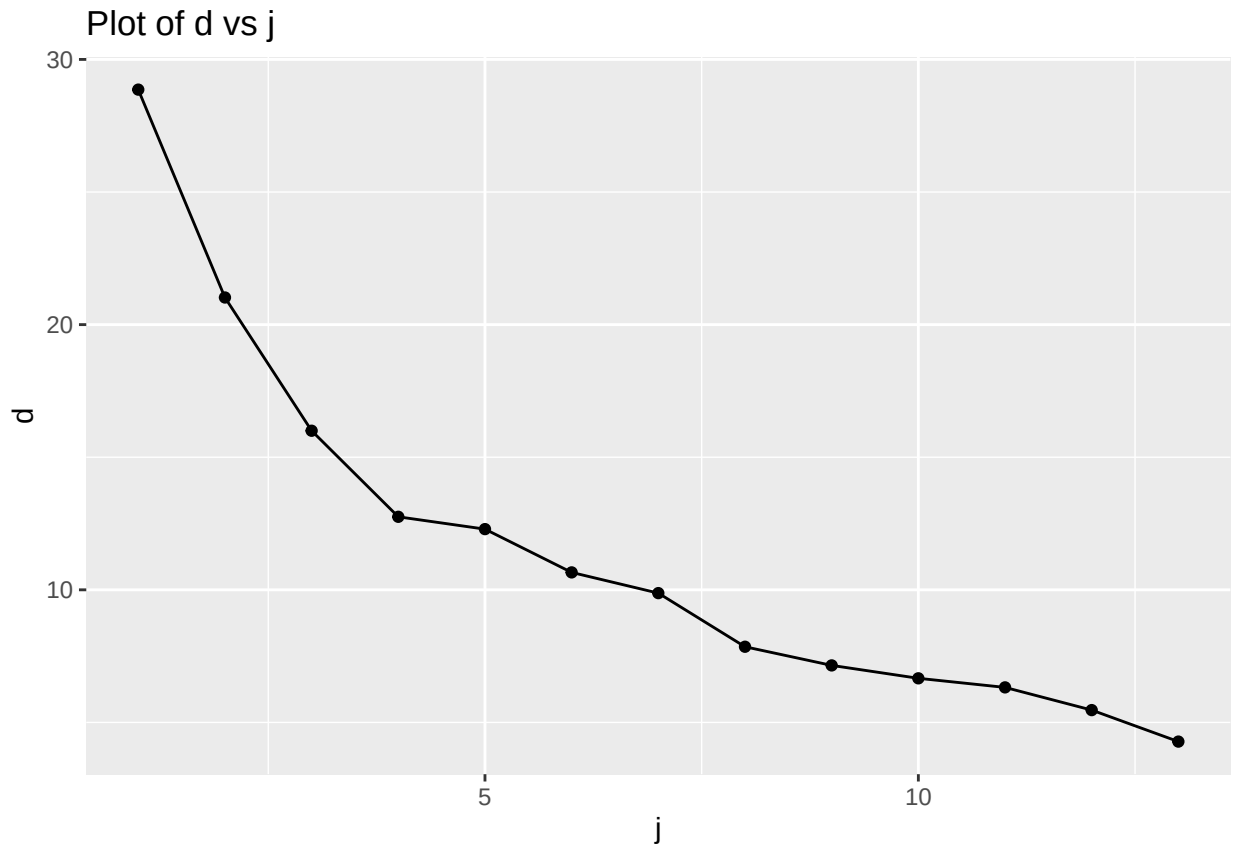
a) The Vj's whose d values are close to zero carry very little information and can be discarded. How many of the more informative Vj's would you suggest keeping?
I would consider keeping the two of the three more informative variables. However, the most significant variable would be enough for a useful prediction.

b) The Vj's whose d values are close to zero carry very little information and can be discarded. Now how many of the more informative Vj's would you suggest keeping?
Code:

```
# Standardize each of the 13 variables:
wine2_std <- scale(wine2, center = TRUE, scale = TRUE)
my.pca_std <- svd(wine2_std)
my.pca_std$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
```

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



I would suggest keeping the four to eight most informative variables as a balance of keeping complexity to a minimum while obtaining a more accurate model.

Exercise 7: Which variable, V1 or V2, is reflecting length and which is reflecting width?

Code:

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

colMeans(virginica)
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##        6.588        2.974        5.552        2.026
```

```
virginica_cntr <- scale(virginica, center = TRUE, scale = FALSE) %>%
  as.data.frame()
head(virginica_cntr, n = 3)
```

```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width
## 1       -0.288       0.326        0.448       0.474
## 2       -0.788      -0.274       -0.452      -0.126
## 3        0.512       0.026        0.348       0.074
```

```
my.pca <- svd(virginica_cntr)
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
```

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

The variable V1 is depicting length and V2 is conveying the width.