Practice problems for CA with partial answers

1. With the goblet data set, complete the following:
	1. Apply other agglomerative methods to determine appropriate clusters for the standardized and non-standardized data. Do the results agree with those from the nearest neighbor method examined in class?

For the standardized data only, here are few of my thoughts about the analyses:

* + 1. Based on the hierarchical tree diagrams, I cut the trees at locations with the following number of clusters
			1. Furthest neighbor: 2, 3, 5, 6
			2. Centroid: 2, 3, 5 (remember to square the values in the distance matrix)
			3. Average: 2, 3, 4, 5
			4. Ward: 2, 4, 5
		2. With respect to the PC score 2D plots, I think that the lower left hand corner should have all of the observations combined into one cluster rather than separated. For example, here is the PC score bubble plot with respect to the average method and 4 clusters:



Perhaps it is a little questionable whether or not 10 and 22 are included in the black cluster. With 5 clusters, the average method would separate these observations out. Note that the other methods do not necessarily agree on which observations would be in 4 or 5 clusters.

* + 1. With respect to the parallel coordinate plots, we are looking for distinguishing qualities among the clusters.

Below is the plot for average method with 4 clusters. The green and blue clusters stand out (e.g., green has the largest w4 and w5 values). The red does not stand out too much from the black, except that the corresponding observations have larger w2 and w4 values.



Below is the plot for Ward’s method with 4 clusters (look at the PC score plots too). I do not like to see the lack of distinguishing qualities between the red and black groups. Note that you sometimes need to be careful though when examining plots like this because your eyes may focus on only one variable at a time (which may not always provide a complete picture).



Alternatively, one may say with the above plot that the red and black only overlap by one observation for the w6 value. Also, variables like w4 and w5 show tendencies toward separation between observations in the two clusters.

* + 1. Overall, I like the 4 clusters obtained through the average method. Other options may be justifiable too.
	1. Apply k-means clustering to the non-standardized data. Do the results agree with those from using the standardized data in class?

Here are few of my thoughts about the analyses:

* + 1. The plot of W vs. K stops to level off perhaps at 4 clusters.
		2. Of course, I need to use cor.use = FALSE with PCA.CA.plot2() due to using non-standardized data.
		3. Below is my bubble plot with 4 clusters. There appears to be two main clusters based on whether or not PC #1 is positive or negative. The other two clusters are much farther away.



* + 1. With respect to 4 clusters, the parallel coordinate plot below shows differences between the clusters. In particular, notice that w1 separates the red and blue completely. Also, w2, w4, and w5 show some separation between the red and blue clusters too.



* + 1. Overall, I am satisfied with 4 clusters.

1. The purpose of this problem is to examine how clustering methods work when we know the correct number of clusters and which observations should be in a particular cluster. We are able to do this through using simulated data.
	1. Simulate 100 observations each from two populations with the following distributions:

 and 

Set a seed number of 1181 before using the first distribution and a seed number of 1218 before using the second distribution. Plot the simulated data in a scatter plot where the plotting symbols correspond to the population. Based on this plot, how well do you think a cluster analysis method will separate out the observations into the correct populations?

> library(mvtnorm)

> set.seed(1181)

> pop1 <- rmvnorm(n = 100, mean = c(0,0), sigma = matrix(data = c(1, 0, 0, 1), nrow = 2, ncol = 2))

> head(pop1)

 [,1] [,2]

[1,] -0.2962019 -1.02229182

[2,] -0.2277220 2.09802300

[3,] 1.8393558 0.29361539

[4,] -0.7641729 -1.16671752

[5,] -0.2279988 0.05907115

[6,] -0.5433259 0.61151723

> set.seed(1218)

> pop2 <- rmvnorm(n = 100, mean = c(6,0), sigma = matrix(data = c(1, 0, 0, 1), nrow = 2, ncol = 2))

> head(pop2)

 [,1] [,2]

[1,] 6.005221 -0.264053889

[2,] 6.013727 0.594869118

[3,] 7.351746 0.003988692

[4,] 8.637141 0.778251779

[5,] 6.447020 0.211541760

[6,] 6.298675 -0.635608118

> dev.new()

> par(pty = "s")

> plot(x = pop1[,1], y = pop1[,2], xlab = "Variable #1", ylab = "Variable #2", pch

 = 1, col = "blue", xlim = c(-5, 10), ylim = c(-5, 10), panel.first = grid())

> points(x = pop2[,1], y = pop2[,2], xlab = "Variable #1", ylab = "Variable #2",

 pch = 2, col = "red")

> legend(x = 0, y = 8, legend = c("Pop. #1", "Pop. #2"), col = c("blue", "red"),

 pch = c(1, 2), bty = "n")



The two groups of points are fairly well separated so I would expect cluster analysis to find the correct clusters.

* 1. Combine the two sets of data into one data frame and perform nearest and farthest neighbor methods upon the non-standardized data. Examine how well the cluster analysis methods group the observations into clusters corresponding to the two populations. Because the data is simulated from normal distributions with the same variance, do not standardize the data (of course, you would not normally know this in practice so you would then need to standardize the data!).

> #Variable names become X1 and X2 by default

> set1 <- rbind(data.frame(pop1, pop = 1), data.frame(pop2, pop = 2))

> head(set1)

 X1 X2 pop

1 -0.2962019 -1.02229182 1

2 -0.2277220 2.09802300 1

3 1.8393558 0.29361539 1

4 -0.7641729 -1.16671752 1

5 -0.2279988 0.05907115 1

6 -0.5433259 0.61151723 1

> tail(set1)

 X1 X2 pop

195 5.405351 -0.5735176 2

196 7.619697 0.2910158 2

197 7.159090 1.0431127 2

198 6.625485 0.9365802 2

199 5.571953 -0.6963330 2

200 7.224140 -0.5669459 2

> dist.mat <- dist(x = set1, method = "euclidean")

> clust.nn <- hclust(d = dist.mat, method = "single")

> clust.fn <- hclust(d = dist.mat, method = "complete")

> dev.new(width = 12)

> plot(clust.nn)



> dev.new(width = 12)

> plot(clust.fn)



> clusters.nn <- cutree(tree = clust.nn, k = 2)

> clusters.fn <- cutree(tree = clust.fn, k = 2)

> compare.nn <- data.frame(set1$pop, clusters.nn)

> compare.fn <- data.frame(set1$pop, clusters.fn)

> #Forms a contingency table summarizing the results.

> # The rows denote the original population and the columns correspond to the

 clusters formed. One needs to be careful here because cluster #1 may actually

 correspond to population #2 (or vice versa) due to how hclust() and cutree()

 number their clusters.

> table(compare.nn)

 clusters.nn

set1.pop 1 2

 1 100 0

 2 0 100

> table(compare.fn)

 clusters.fn

set1.pop 1 2

 1 100 0

 2 0 100

Both cluster analysis methods put the observations into the correct clusters! Notice where observation #70 appears in the nearest neighbor tree diagram. In the scatter plot of the data, #70 has the smallest variable #2 value.

* 1. Repeat a) and b) with the variable #1 mean for population #2 being decreased from 6 to 5, 4, 3, and 2. Describe trends you see as the mean decreases. Does nearest or farthest neighbor perform better?

When the mean is 5, this is when we start to see the nearest neighbor method have difficulties. The two clusters involve one with just #70 and one with the remaining 199 observations! Overall, it is perhaps not as bad as it seems. If there were four clusters, I obtain

> clusters.nn <- cutree(tree = clust.nn, k = 4)

> compare.nn <- data.frame(set1$pop, clusters.nn)

> table(compare.nn)

 clusters.nn

set1.pop 1 2 3 4

 1 98 1 1 0

 2 0 0 0 100

which indicates that the method separates out all but two observations correctly.

* 1. Perform a similar investigation as in a), b), and c) using other agglomerative methods and k-means clustering.

I have not done this part, but it looks like fun! It would be even more fun to try different original cluster shapes!