Chapter 6

1. Commands in R to know.

   prcomp, drawst, dist, plot, text, as.matrix, hclust, cutree, cmdscale, isoMDS, round, scale, cov, rownames, colnames, Stir2, pairsbg, mahalanobis, kmeans, scatterplot3d, cclust, clusters, barplot, clusterSim, dist2, parameters, head, info, shadow, shadowStars, stepFlexclust, polygon, chull, points, stripes.

2. Be able to implement by hand for very small data sets (and with R for larger ones):

   a. perform agglomerative hierarchical clustering (using single, complete and average linkage); plot cluster dendograms, obtain clusters.

   b. plot clusters on principal components, or other lower dimensional representations.

   c. obtain approximate K-means clusters.

   d. display cluster solutions graphically (using flexclust ideas such as shadows, barplots, shadow stars, stripes).

   e. determine reasonable numbers of clusters.

   f. determine the sum of within cluster distances.

Problems

Show your work. Check your answers with R. Feel free to liberally round solutions to, say, three decimal places for display purposes.

1. (Use R only for Part (a)) Consider the Euclidean distance matrix.

	1	2	3	4	5	6
1	0	8	9	13	18	16
2	8	0	1	5	10	8
3	9	1	0	4	9	7
4	13	5	4	0	5	3
5	18	10	9	5	0	2
6	16	8	7	3	2	0

   (a) Find a data set with (integer coordinates and) the above Euclidean distances (you can use R for this part).

   (b) Obtain cluster dendograms using the single and complete linkage methods.

   (c) State any differences in the dendograms found in (b).

   (d) Obtain the three-cluster solution for the complete method in (b). State the corresponding sum of within cluster distances.

(e) Find the centroids for the three clusters obtained in (d), and the matrix of shadow values between centroids, using the formula

$$S_{i,j} = \frac{1}{n_i} \sum_{x \in A_{i,j}} \frac{2d(x, c_i)}{d(x, c_i) + d(x, c_j)}, \qquad 1 \leq i \neq j, \leq 2, \tag{1}$$

where $A_{i,j}$ is the set of points whose closest centroid is c_i and second closest centroid is c_j, and n_i is the number of points closest to the centroid c_i.

(f) Draw a rough shadow stars plot for the clusters in (b).

2. (Use R, throughout) Consider the data matrix given in the file that was E-mailed to you. The matrix begins

```
> head(Q2)
           V1         V2          V3
1 -0.09580978 0.06216742 -0.07684248
2  0.79640420 0.08455216  1.43648000
3  1.16606100 0.84623770 -0.77053320
4 -0.68268140 0.49050940 -0.12621180
5  0.96462750 0.94635740  1.13541900
6  0.08437982 0.05538653  1.66976800

> dim(Q2)
[1] 600   3
```

You can read the data into the matrix Q2, (check the "Change dir" tab under "File", first) by typing

```
> Q2<-read.table(file="Q2data.txt")
```

Note that matrix was written out with the command

```
> write(t(Q2), file="Q2data.txt",ncol=3)
```

(a) Plot the points in three dimensional space.

(b) Use `cclust`/flexclust a few times to obtain some insight on appropriate number of clusters.

(c) Obtain a `cclust` run that gives a sum of within cluster distances that is less than 312.03. Plot the `cclust` run (using `prcomp`). You may need to run a few iterations of `cclust` to obtain a low enough sum of within cluster distances. If you don't get a fairly nice looking (perhaps even "cool") plot, you may have an error in your coding. :)

(d) Draw a new three dimensional scatterplot, this time with colored cluster numbers at each of the points (with colors distinguishing clusters).

(e) The data arose by adding some noise to some nice points in three dimensional space. Come up with a theory on what those points were. Use any information from the package `flexclust` that may be helpful.

(f) Obtain a barplot for the `cclust` result in (c).

(g) For the `cclust` result from (c), use `clusterSim` to obtain the matrix of shadow values. Is the matrix symmetric?

(h) For the cclust result from (c), determine the centroids, c_a, closest to the point $(0, 0, 0)$ and the centroid, c_b, closest to the point $(1, 1, 0)$. Find the set of points $A_{a,b}$ (as in Equation (1), above), whose closest centroid is c_a and second closest centroid is c_b. Hint: Use the function `getshads`, below.

(i) (Show some manipulations.) Verify the value, $S_{a,b}$ in the matrix of shadow values corresponding to the centroids c_a and c_b (as found in Part (h)). Hint: Use the interim values outputted by the function `getshads`, below, as a check.

(j) Locate the lines for the corresponding values, $2d(x, c_a)/(d(x, c_a) + d(x, c_b))$ (as in (1)), for $x \in A_{a,b}$, in a shadow stars plot. Hint: The corresponding values comprise the [[5]] component in the list of values outputted by `getshads`.

```
> getshads<-
function (M,clM,i,j)
{
L<-1:dim(M)[1]
DM<-dist2(M,parameters(clM))
DMo<-t(apply(DM,1,order))
DMoi<-DMo[clusters(clM)==i,]
ni<-dim(DMoi)[1]
v<-L[(DMo[,1]==i)&(DMo[,2]==j)]
dii<-DM[(DMo[,1]==i)&(DMo[,2]==j),i]
dij<-DM[(DMo[,1]==i)&(DMo[,2]==j),j]
d<-2*dii/(dii+dij)
list(v,ni,dii,dij,2*dii/(dii+dij),sum(d)/ni)
}
```

3. (Use R, throughout) Consider the crime data for states. You may need to run

```
> demo("Ch-CA")
```

to load the data into R.

(a) Restrict to the data on the variables rape, burglary and theft. Scale the variables accordingly.

(b) Find the mahalanobis distances for the states in the new data set.

(c) Use `pairsbg` to obtain bagplots for the data.

(d) Remove the two states with the highest mahalanobis distances, and rescale the data. Note that the following will scale the data and remove the attributes ("scaled:scale" and "scaled:scale").

```
> MQ4s<-matrix(scale(MQ4,scale=TRUE,center=TRUE),ncol=3)
```

The data should now begin as follows.

```
> head(MQ4s)
          Rape    Burglary      Theft
ME -1.3711582 -0.95946793 -0.7384367
NH -0.8828718 -1.07153371 -0.9243492
VT -0.8610082 -0.61860119 -0.2703116
MA -0.2852675 -0.33376734 -0.9497617
RI -0.8901596  0.18687159 -0.4428491
CT -0.7152511 -0.03725997 -0.1887240
```

(e) Perform a principal components analysis on the new data set in (d). Interpret the coefficients.

(f) Plot the states, using the first two principal components.

(g) Cluster the states using `hclust` (and complete linkage). Plot the cluster dendogram and use the result to form four clusters. State the states in each cluster. Give the sum of the within cluster distances; you can use the following program.

```
> sumcldist<-
function (M,cl)
{
mv<-NULL
Dt<-0
t<-unique(cl)
for (i in t)
{
U<-cbind(M[cl==i,])
m<-colMeans(U)
mv<-rbind(mv,m)
Dm<-dist2(U,m)
print(Dm)
Dt<-Dt+sum(Dm)
```

```
}
rownames(mv)<-t
list(mv[order(t),],Dt)
}
```

(h) Obtain a four-clustering of the data using `cclust`. Ensure that your clustering has a sum of within cluster distances that is less than 34.30 and that exactly six $S_{i,j}$ (produced by `clusterSim`) are zero ($1 \le i \le j \le 4$). List the states in each cluster.

(i) Plot the result of `cclust`, from Part (h) (without projecting onto the principal components). Label the states using `text` (and a reasonable value of `cex`). Plot the result of `cclust` (projecting onto the principal components). Label the states using `text`.

(j) Obtain a second four-clustering of the data using `cclust`, this time ensuring that your clustering has a sum of within cluster distances that is greater than 39.50. Plot the results, as in Part (i) (with and without projecting onto the principal components).

(k) For the result of `cclust` in Part (h) (with sum of within cluster distances less than 34.30), draw a barplot, and give the cluster centroids.

(l) Give the distances of the data point corresponding to NC, from all four centroids (for the clustering in Part (h)). Locate (approximately) the stripes corresponding to these distances on a `stripes` (type="all") plot.