Stat 8053, Cluster Analysis, rev December 2, 2013

Hierarchial clustering is done with the `hclust` method in the base `stats` package, and with more complexity in the `cluster` package. The relevant arguments to `hclust` are

```
hclust(d, method = "complete")
```

The first argument `d` is a *dissimilarity matrix*, often computed with the `dist` function. The second argument is the `method` of agglomeration. The methods we consider are `c("single", "complete", "average")`, although others are available, with `complete` the default.

The function for computing the dissimilarity measures has relevant arguments:

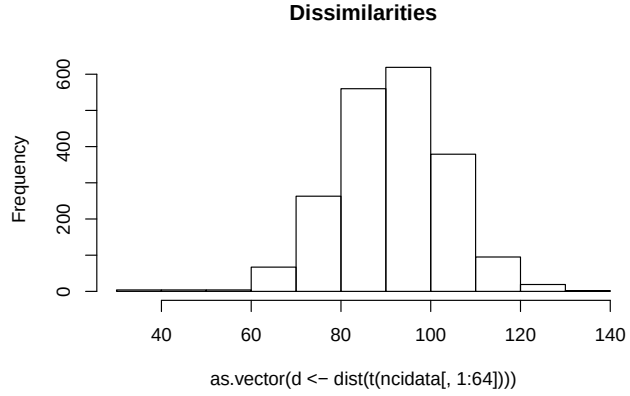
```
dist(x, method = "euclidean")
```

where `x` is a data.frame or data matrix. The returned value is a lower-triangular matrix of dissimilarities, Euclidean distances by default. There are many other options for `method`.

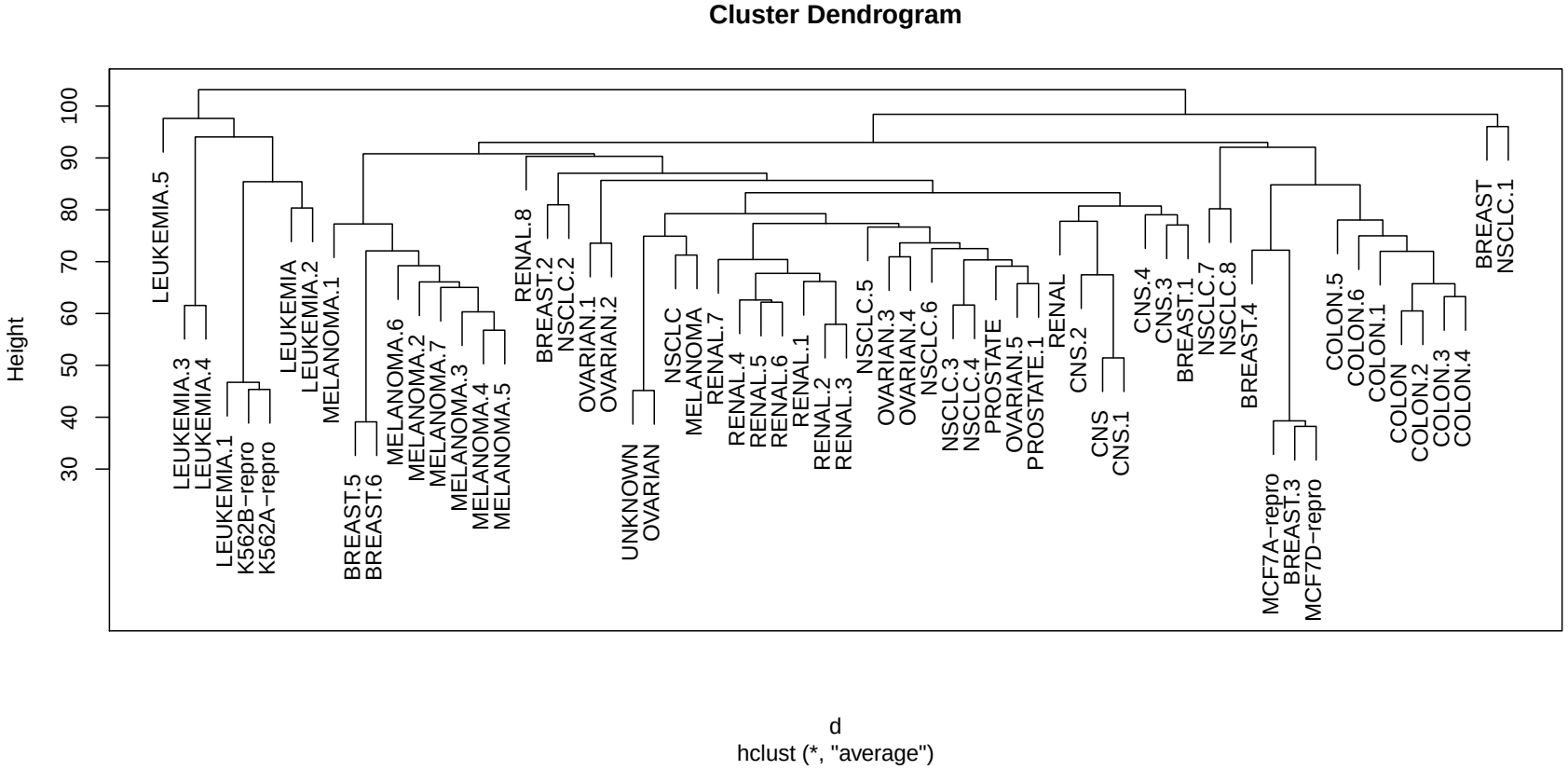
Gene Expression

This example has $p = 64$ gene profiles of $n = 6830$ genes. The column names correspond to a diagnosis for each of the 64 profiles, information that we don't use in the clustering; the rows are the log(expression) for each of the genes, so the file is 64×6830. The goal we pursue is to cluster the 64 profiles, not to cluster the genes, so we view the data as $n = 64$ vectors in $\Re^{6830}$:

```
loc <- "http://www.stat.umn.edu/~sandy/courses/8053/Data/FHT/ncidata.Rda"
load(url(loc))
hist(as.vector(d <- dist(t(ncidata[, 1:64]))), main="Dissimilarities")
```

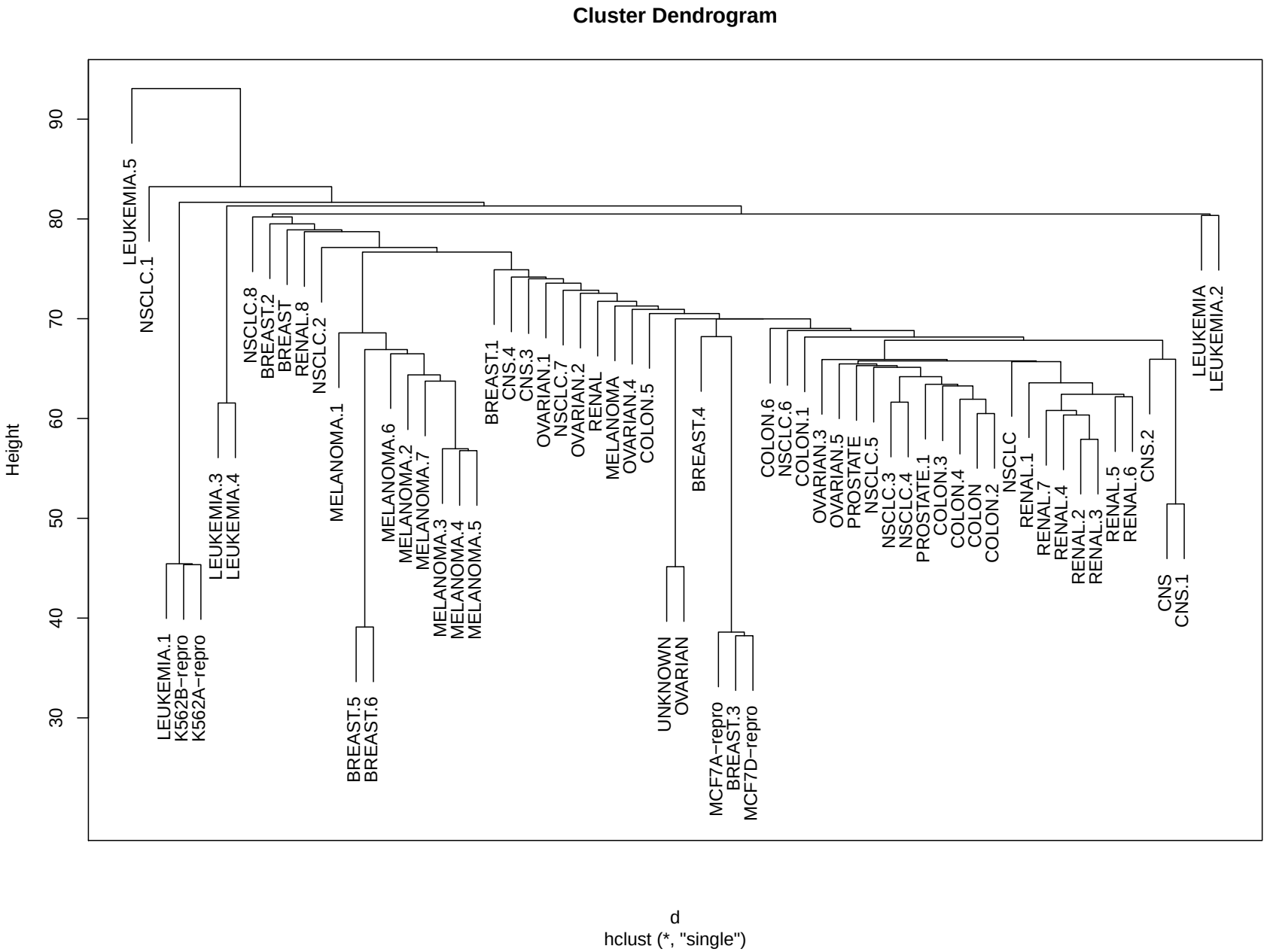


```
 (h1 <- hclust(d, method="average"))

Call:
hclust(d = d, method = "average")

Cluster method   : average
Distance         : euclidean
Number of objects: 64

 plot(h1, frame.plot=TRUE)
```



I used average clustering in `hist` and Euclidean distance in `dist`. I did not do any standardization because of the expression of each gene is on a comparable scale. The vertical axis in the dendrogram is in the units of the dissimilarity measure.

```
plot(hclust(d, method="single"), frame.plot=TRUE)
```



Big Mac, from the *Economist*, 2010 data

The first example looks at economic data from 69 world cities in 2003, provided by the Union Bank of Switzerland.

BigMac Minutes of labor to purchase a Big Mac

Bread Minutes of labor to purchase 1 kg of bread

Rice Minutes of labor to purchase 1 kg of rice

FoodIndex Food price index (Zurich=100)

Bus Cost in US dollars for a one-way 10 km ticket

Apt Normal rent (US dollars) of a 3 room apartment

TeachGI Primary teacher's gross income, 1000s of US dollars

TeachNI Primary teacher's net income, 1000s of US dollars

TaxRate Percent Tax paid by a primary teacher. This variable is defined as

$$\texttt{TaxRate} = 100 \times (\texttt{TeachGI} - \texttt{TeachNI})/\texttt{TeachGI}$$

I will not use this variable in the clustering.

TeachHours Primary teacher's hours of work per week

The variables describe aspects of the costs of food, transport, and housing. The last four variables describe earnings, based on primary school teacher's experience. The goal is to identify cities that are similar.

```
library(alr4)
head(BigMac2003)
```

```
          BigMac Bread Rice FoodIndex  Bus Apt TeachGI TeachNI TaxRate TeachHours
Amsterdam     16     9    9      65.9 2.00 890    34.3    20.5  40.233         39
Athens        21    12   19      63.5 0.61 620    19.5    15.9  18.462         29
Auckland      19    19    9      55.4 1.57 780    22.0    16.1  26.818         40
Bangkok       50    42   25      46.4 0.47 120     4.2     4.0   4.762         35
Barcelona     22    19   10      62.9 0.91 590    25.5    20.1  21.177         39
Basel         15     7    7      98.4 2.34 930    78.5    57.6  26.624         35
```

```
library(psych)
describe(BigMac2003)[ , c(2:5, 9:10)]
```

```
            n   mean     sd median     max   range
BigMac     69  37.28  31.42  25.00  185.00  175.00
Bread      69  24.58  17.81  19.00   90.00   84.00
Rice       69  19.94  15.24  16.00   96.00   91.00
FoodIndex  69  61.93  24.59  62.60  129.40  105.90
Bus        69   1.04   0.80   0.83    3.70    3.61
Apt        69 713.91 461.84 700.00 1930.00 1840.00
TeachGI    69  21.22  19.21  17.80   78.50   77.90
TeachNI    69  15.78  14.08  12.60   57.60   57.10
TaxRate    69  21.48  10.30  21.74   42.35   49.67
TeachHours 69  36.74   7.42  38.00   58.00   38.00
```

The data are clearly on different scales, and so some standardization is required or else the clustering will be dominated by the large-variance variables. Lacking any theory to suggest a scaling, I'll use correlation scale, which both centers and scales. For a dissimilarity measure I'll use Euclidean distance, the default to the `dist` function. Big values mean items are dissimliar.

```
BigMac2003 <- BigMac2003[,-9]
as.matrix(dist(scale(BigMac2003)))[1:7,1:7]
```

```
          Amsterdam Athens Auckland Bangkok Barcelona Basel Berlin
Amsterdam     0.000  2.523    1.174   4.109     1.690 3.808  1.197
Athens        2.523  0.000    2.114   2.745     1.629 5.170  2.752
Auckland      1.174  2.114    0.000   3.140     1.042 4.730  1.976
Bangkok       4.109  2.745    3.140   0.000     2.837 6.998  4.346
Barcelona     1.690  1.629    1.042   2.837     0.000 4.622  2.163
Basel         3.808  5.170    4.730   6.998     4.622 0.000  3.155
Berlin        1.197  2.752    1.976   4.346     2.163 3.155  0.000
```

The dissimilarities for only the first seven cities are displayed. For comparison, here are the same dissimilarities, unscaled:

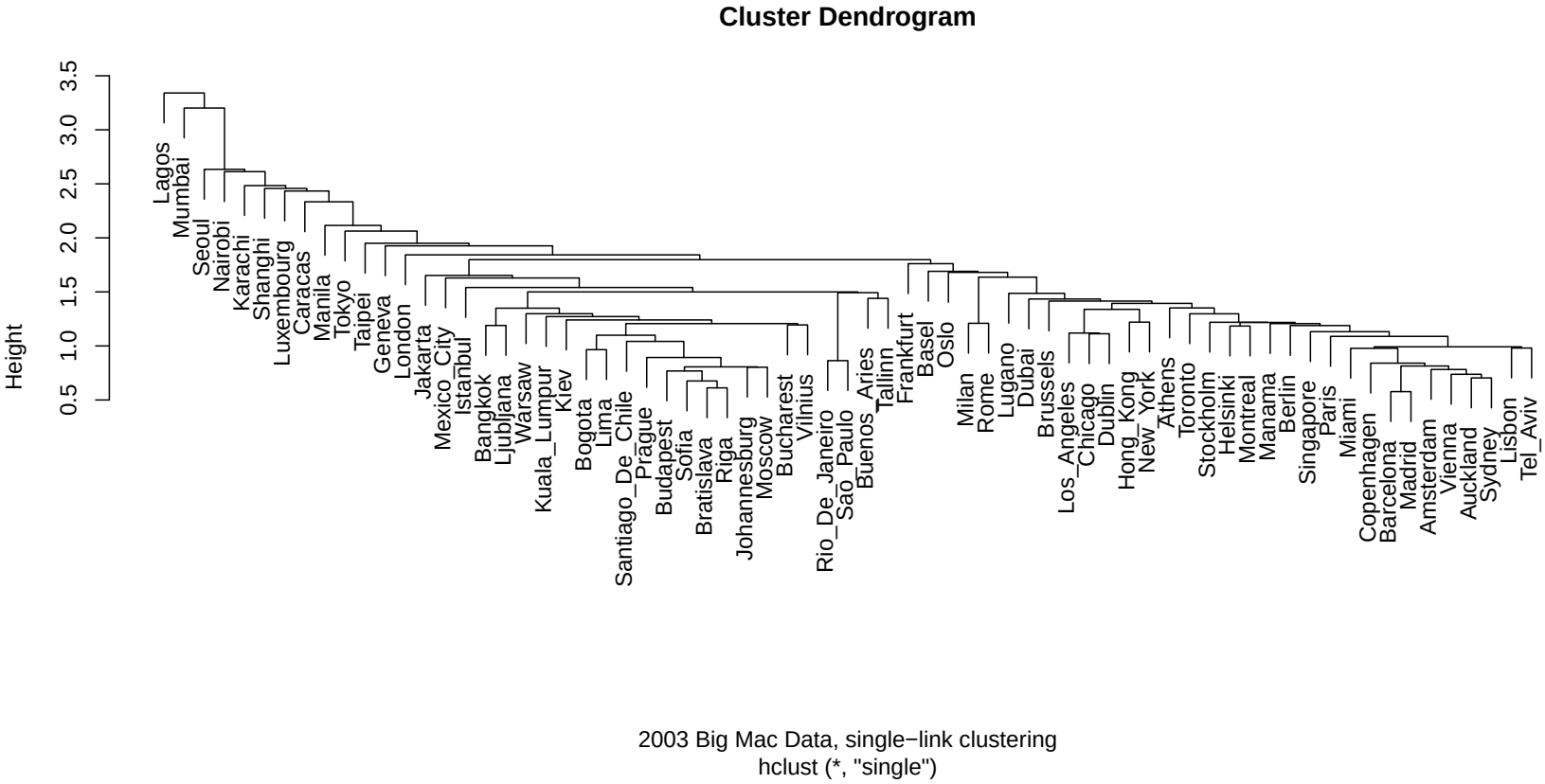
```
as.matrix(dist(BigMac2003))[1:7,1:7]
```

```
          Amsterdam Athens Auckland Bangkok Barcelona  Basel Berlin
Amsterdam      0.00 270.89    111.8   772.6    300.37  77.53 260.47
Athens       270.89   0.00    161.1   502.5     34.43 320.60  31.77
Auckland     111.76 161.08      0.0   661.8    190.25 171.62 153.07
Bangkok      772.64 502.48    661.8     0.0    472.69 818.52 514.76
```

```
Barcelona     300.37  34.43    190.3   472.7      0.00 348.28  47.16
Basel          77.53 320.60    171.6   818.5    348.28   0.00 305.28
Berlin        260.47  31.77    153.1   514.8     47.16 305.28   0.00
```

Let's try single-link clustering. Single link clustering will link current clusters P and R if the dissimilarity (Euclidean distance) between the closest element in P to the closest element in R is minimized. The function `hclust` will be used. There is also a package called `cluster` and a taskview at `http://cran.r-project.org/web/views/Cluster.html` on clustering for more computational methods.

```
hc <- hclust(dist(scale(BigMac2003)), "single")
plot(hc, cex=.75, xlab="2003 Big Mac Data, single-link clustering")
```

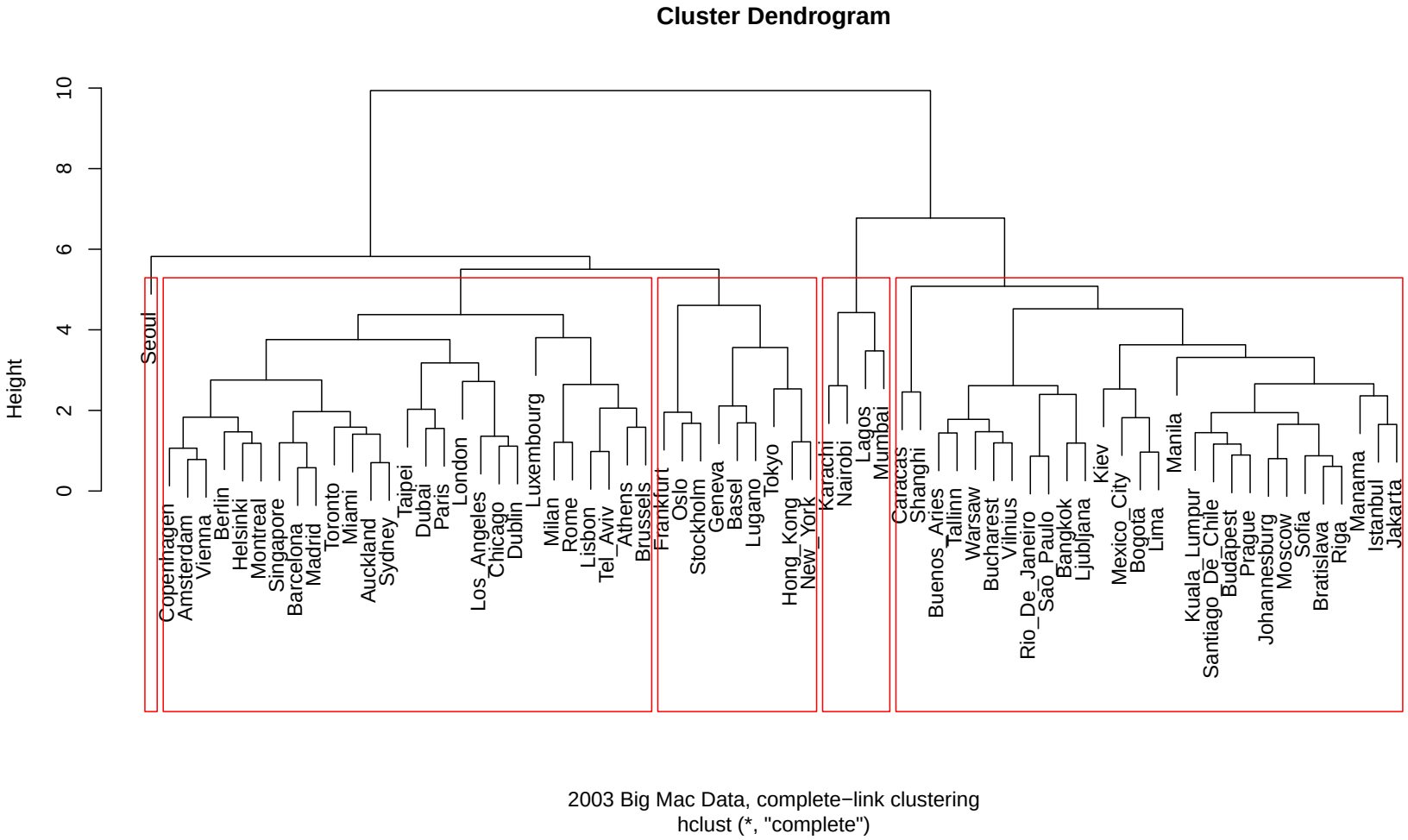


The vertical axis *Height* is the value of the criterion associated with the clustering method for the particular agglomeration. Starting at the bottom, Barcelona and Madrid appear to be the most similar, as are Auckland and Sydney, and then several

other pairs. Beyond the pairs there is no obvious clustering of the cities in this graph.

Here is the output for complete clustering. In complete clustering P and R are joined if the dissimilarity between the farthest object in P and the farthest object in R is minimized.

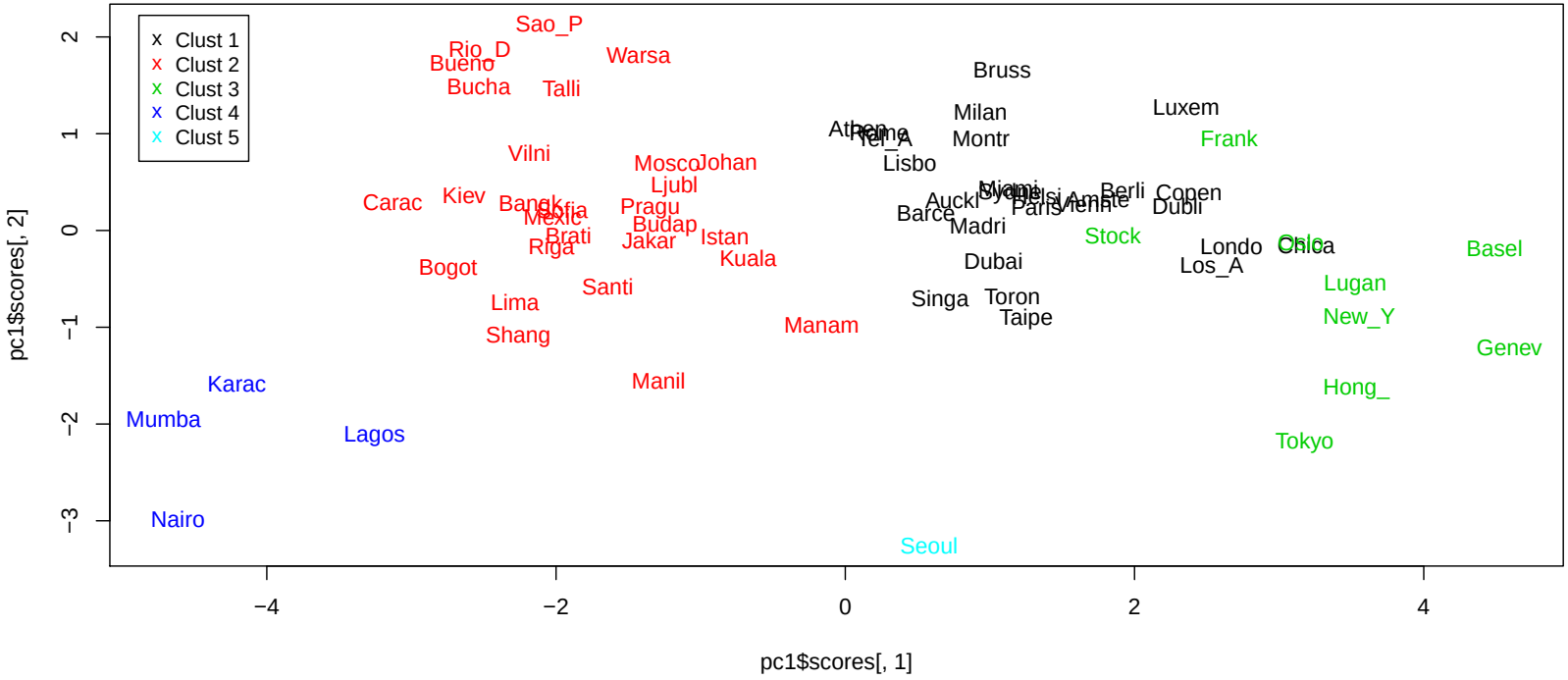
```
hc2 <- hclust(dist(scale(BigMac2003)), "complete")
plot(hc2, cex=.75, xlab="2003 Big Mac Data, complete-link clustering")
rect.hclust(hc2, k=5, border="red")
```



This clustering is more esthetic, with two or five broad clusters.

Let's see how this corresponds to Principal Component Analysis. In the plot below I've colored the data points corresponding to the complete linkage clustering with five clusters.

```
pc1 <- princomp(BigMac2003, cor=TRUE)
plot(pc1$scores[,1],pc1$scores[, 2], type="n")
text(pc1$scores[,1],pc1$scores[, 2], substr(rownames(BigMac2003), 1, 5), col=cutree(hc2, 5))
legend("topleft",paste("Clust",1:5), col=1:5, pch="x", cex=0.9,inset=0.02)
```



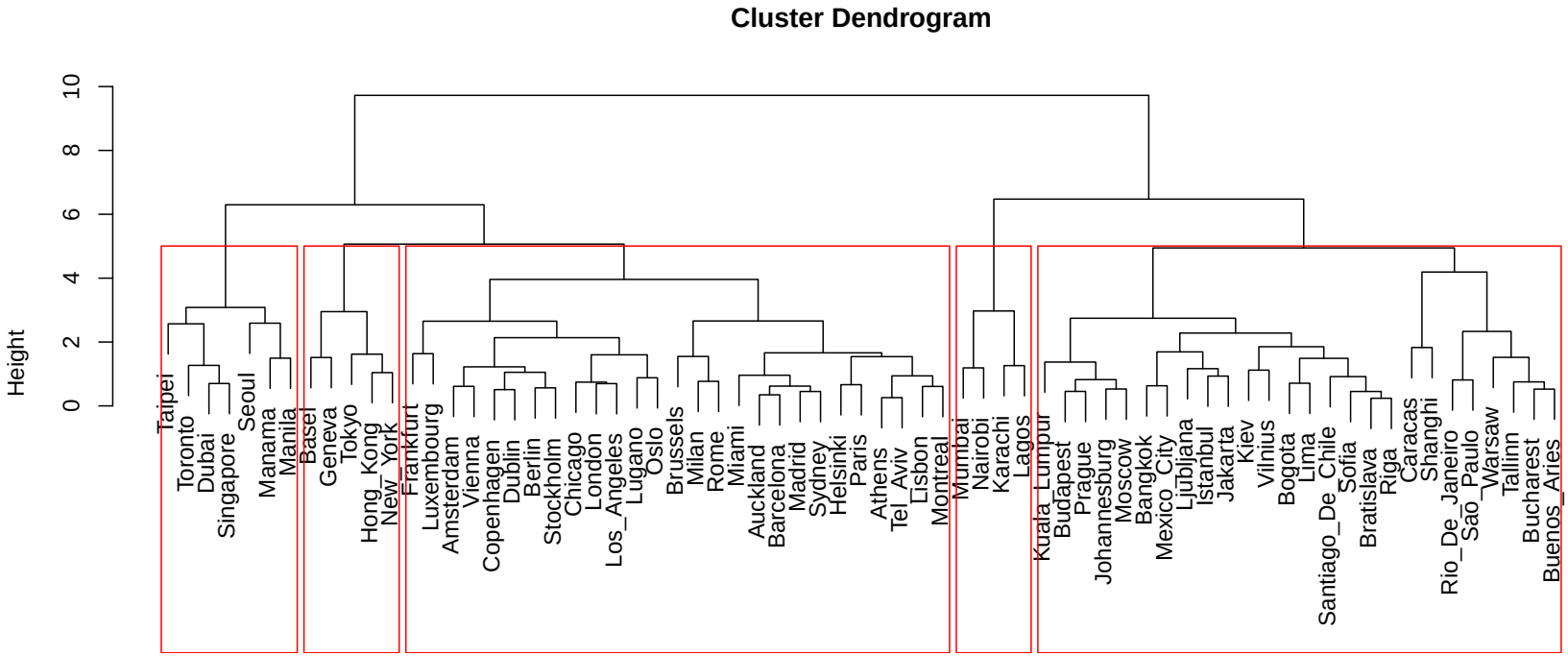
The five cluster complete-linkage solution is remarkably similar to the first principal component.

The distance measure used so far is $[(x_i - \bar{x}) - (x_j - \bar{x})]' D^{-1} [(x_i - \bar{x}) - (x_j - \bar{x})]$, where D is a diagonal matrix of column sample variances. If R is the sample correlation matrix, it seems more reasonable to me to use $[(x_i - \bar{x}) - (x_j - \bar{x})]' D^{-1/2} R^{-1} D^{-1/2} [(x_i - \bar{x}) - (x_j - \bar{x})]$ to account for covariance between the components of x. We can do that by replacing the original data X by the singular value decomposition of $(X - 1\bar{x}')D^{-1/2}$.

```
(pc1 <- princomp(BigMac2003, cor=TRUE))
```

```
Call:
princomp(x = BigMac2003, cor = TRUE)

Standard deviations:
 Comp.1  Comp.2  Comp.3  Comp.4  Comp.5  Comp.6  Comp.7  Comp.8  Comp.9
2.25402 1.08662 0.89624 0.80248 0.70239 0.59846 0.55500 0.35284 0.08607

 9  variables and  69 observations.
```

```
hc3 <- hclust(dist(pc1$scores[, 1:4]), "complete")
plot(hc3, cex=.6, xlab="2003 Big Mac Data, Mahalanobis Distance")
rect.hclust(hc3, k=5, border="red")
```



2003 Big Mac Data, Mahalanobis Distance
hclust (*, "complete")

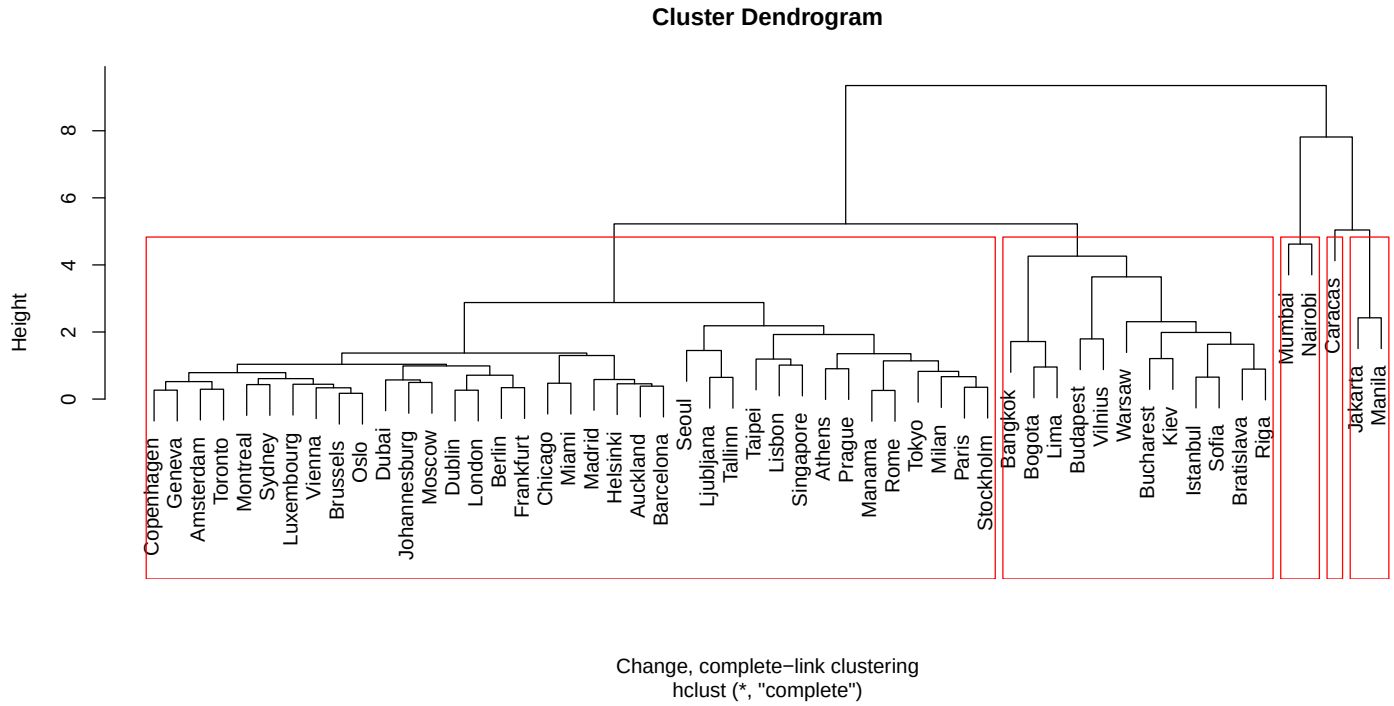
```
table(cutree(hc2, 5), cutree(hc3, 5))
```

```
    1  2  3  4  5
 1 23  0  0  4  0
 2  0 26  0  2  0
 3  4  0  5  0  0
 4  0  0  0  0  4
 5  0  0  0  1  0
```

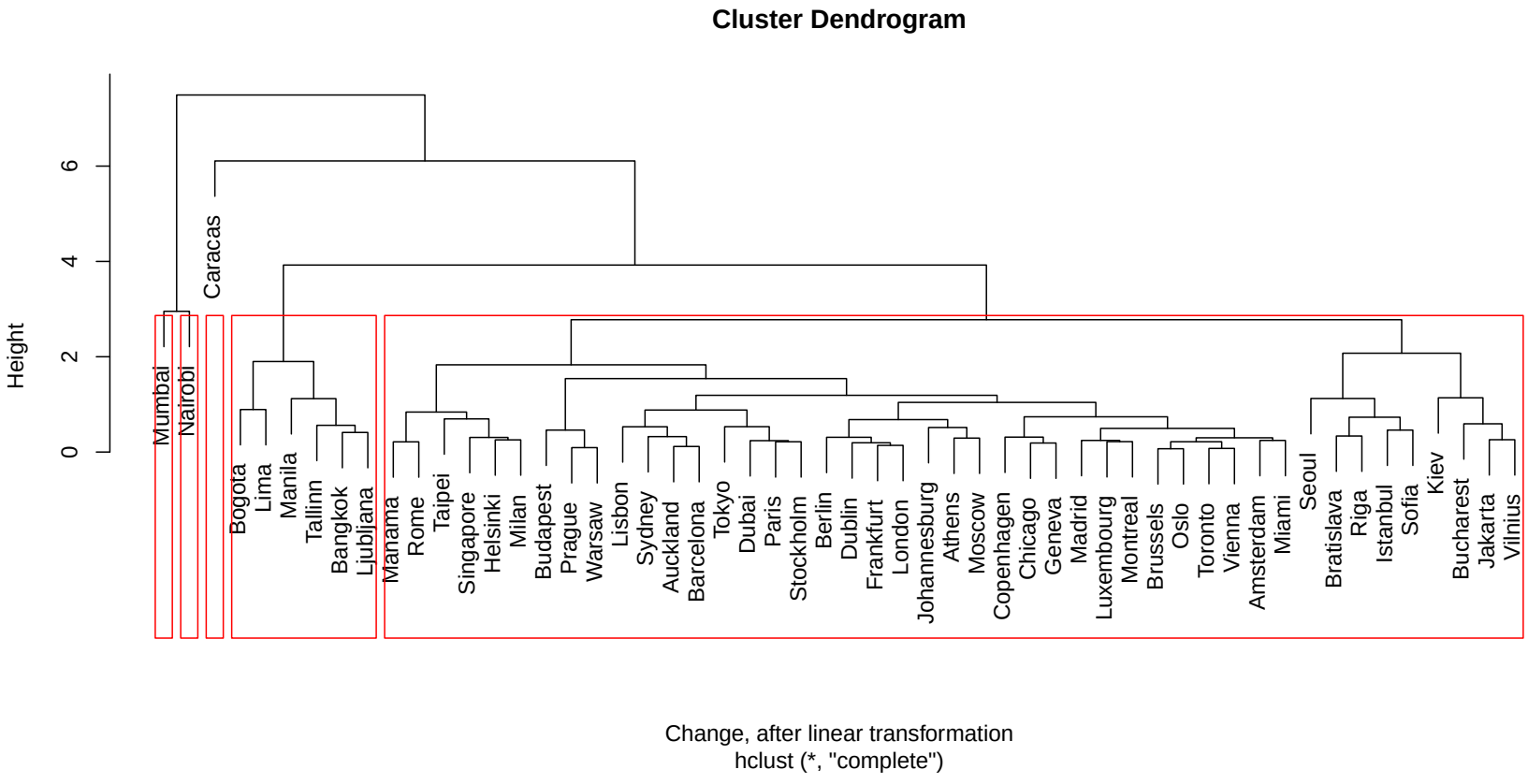
Seoul is no longer a cluster by itself.

The data in the file `UBSprices` lists the 2003 prices for a Big Mac, bread and rice along with the 2009 prices for those same commodities. Let's see how cities cluster with these variables:

```
du <- dist(scale(UBSprices))
hca <- hclust(du, "complete")
plot(hca, cex=.75, xlab="Change, complete-link clustering")
rect.hclust(hca, k=5, border="red")
```



```
UBS2 <- transform(UBSprices, macchange = bigmac2009-bigmac2003,
     breadchange=bread2009-bread2003, ricechange=rice2009-rice2003)
du2 <- dist(scale(UBSprices[, -(1:3)]))
plot(hcb <- hclust(du2, "complete"), cex=.75, xlab="Change, after linear transformation")
rect.hclust(hcb, k=5, border="red")
```



Comparison to LCA

```
library(poLCA)
data(cheating)
str(cheating)
```

```
'data.frame':        319 obs. of  5 variables:
 $ LIEEXAM : num  1 1 1 1 1 1 1 1 1 1 ...
 $ LIEPAPER: num  1 1 1 1 1 1 1 1 1 1 ...
 $ FRAUD   : num  1 1 1 1 1 1 1 1 1 1 ...
 $ COPYEXAM: num  1 1 1 1 1 1 1 1 1 1 ...
 $ GPA     : int  NA NA NA NA 1 1 1 1 1 1 ...
```

```
f <- cbind(LIEEXAM, LIEPAPER, FRAUD, COPYEXAM) ~ 1
ch2 <- poLCA(f, cheating, nclass=2, nrep=5)
```

```
Model 1: llik = -440 ... best llik = -440
Model 2: llik = -440 ... best llik = -440
Model 3: llik = -440 ... best llik = -440
Model 4: llik = -440 ... best llik = -440
Model 5: llik = -440 ... best llik = -440
Conditional item response (column) probabilities,
 by outcome variable, for each class (row)

$LIEEXAM
           Pr(1)  Pr(2)
class 1:  0.9834 0.0166
class 2:  0.4231 0.5769

$LIEPAPER
           Pr(1)  Pr(2)
class 1:  0.9708 0.0292
class 2:  0.4109 0.5891

$FRAUD
           Pr(1)  Pr(2)
class 1:  0.9629 0.0371
class 2:  0.7840 0.2160

$COPYEXAM
           Pr(1)  Pr(2)
class 1:  0.8181 0.1819
```

```
class 2:  0.6236 0.3764

Estimated class population shares
 0.8394 0.1606

Predicted class memberships (by modal posterior prob.)
 0.8307 0.1693

=========================================================
Fit for 2 latent classes:
=========================================================
number of observations: 319
number of estimated parameters: 9
residual degrees of freedom: 6
maximum log-likelihood: -440

AIC(2): 898.1
BIC(2): 931.9
G^2(2): 7.764 (Likelihood ratio/deviance statistic)
X^2(2): 8.323 (Chi-square goodness of fit)
```

```
d1 <- dist(cheating[, -5], method="manhattan")
xtabs(~ ch2$predclass + cutree(hclust(d1, method="single"), 2))
```

```
             cutree(hclust(d1, method = "single"), 2)
ch2$predclass   1   2
            1 265   0
            2  52   2
```

```
xtabs(~ ch2$predclass + cutree(hclust(d1, method="average"), 2))
```

```
             cutree(hclust(d1, method = "average"), 2)
ch2$predclass   1   2
            1 265   0
            2  45   9
```

```
xtabs(~ ch2$predclass + cutree(hclust(d1, method="complete"), 2))
```

```
             cutree(hclust(d1, method = "complete"), 2)
ch2$predclass   1   2
            1 265   0
            2  20  34
```

Wolf Skulls

The data consist of 9 physical measurements on the skulls of wolves collected from five locations in North America. I have not provided the data for you, so you can't reproduce this example.

```
describe(data)
```

```
            var  n   mean   sd median trimmed  mad   min   max range  skew kurtosis   se
Source*       1 82   2.56 1.19   2.00    2.58 1.48   1.0   4.0   3.0  0.01    -1.54 0.13
length        2 82 252.60 8.23 253.00  252.32 7.41 235.0 274.0  39.0  0.25    -0.21 0.91
zyg_width     3 82 137.25 5.91 136.50  136.97 5.19 125.4 152.0  26.6  0.42    -0.19 0.65
alve_lgnth    4 82  85.02 3.04  85.00   84.97 2.30  77.0  93.1  16.1  0.20     0.15 0.34
rostbsewid    5 82  80.23 3.87  80.00   80.18 4.23  72.2  89.1  16.9  0.13    -0.46 0.43
pal_width     6 82  30.45 2.74  30.65   30.55 3.04  24.1  35.3  11.2 -0.29    -0.76 0.30
frshldwidth   7 82  63.00 4.39  63.20   62.82 4.37  52.5  73.3  20.8  0.31    -0.05 0.48
cheekhgt      8 82  38.74 2.37  39.00   38.75 2.00  34.3  44.0   9.7 -0.09    -0.48 0.26
jugdepth      9 82  18.97 1.57  19.05   18.97 1.41  14.7  23.6   8.9  0.02     0.33 0.17
upcarlngth   10 82  25.15 1.12  25.00   25.08 1.19  23.1  28.5   5.4  0.58    -0.11 0.12
X2upmolwth   11 82  13.91 0.82  14.00   13.91 0.82  12.3  16.1   3.8  0.08    -0.58 0.09
```

```
xtabs( ~ Source, data)
```

```
Source
    1 Alg   2 MN70s 3 NEMNold     4 WUS
       20        23        12        27
```

The initial hypotheses of interest are:

1. Pre-1950 NE MN skulls tend to be similar to Algonquin Park skulls or at least intermediate between Algonquin Park skulls and post 1970 NE MN skulls.

2. Pre-1950 NE MN skulls differ from post-1970 NE MN skulls.

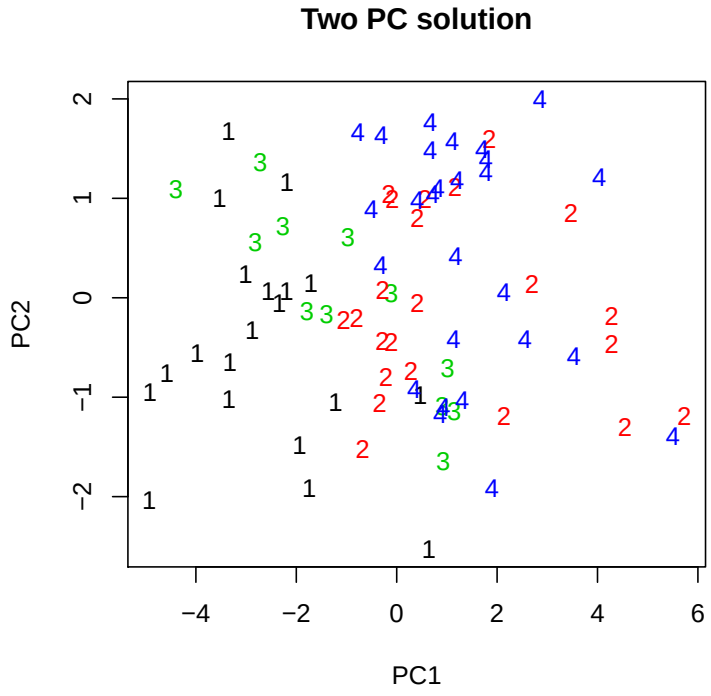
3. Post-1970 MN skulls are similar to western skulls.

For a start, let's look at a graph of the first two principal components of the data

```
p1 <- prcomp(data[, -1], scale=TRUE)
summary(p1)
```

```
Importance of components:
                          PC1   PC2    PC3    PC4   PC5    PC6    PC7    PC8    PC9   PC10
Standard deviation      2.375 1.089 0.8640 0.8242 0.768 0.6199 0.5047 0.4699 0.4379 0.3234
Proportion of Variance  0.564 0.119 0.0747 0.0679 0.059 0.0384 0.0255 0.0221 0.0192 0.0105
Cumulative Proportion   0.564 0.683 0.7575 0.8254 0.884 0.9228 0.9483 0.9704 0.9895 1.0000
```

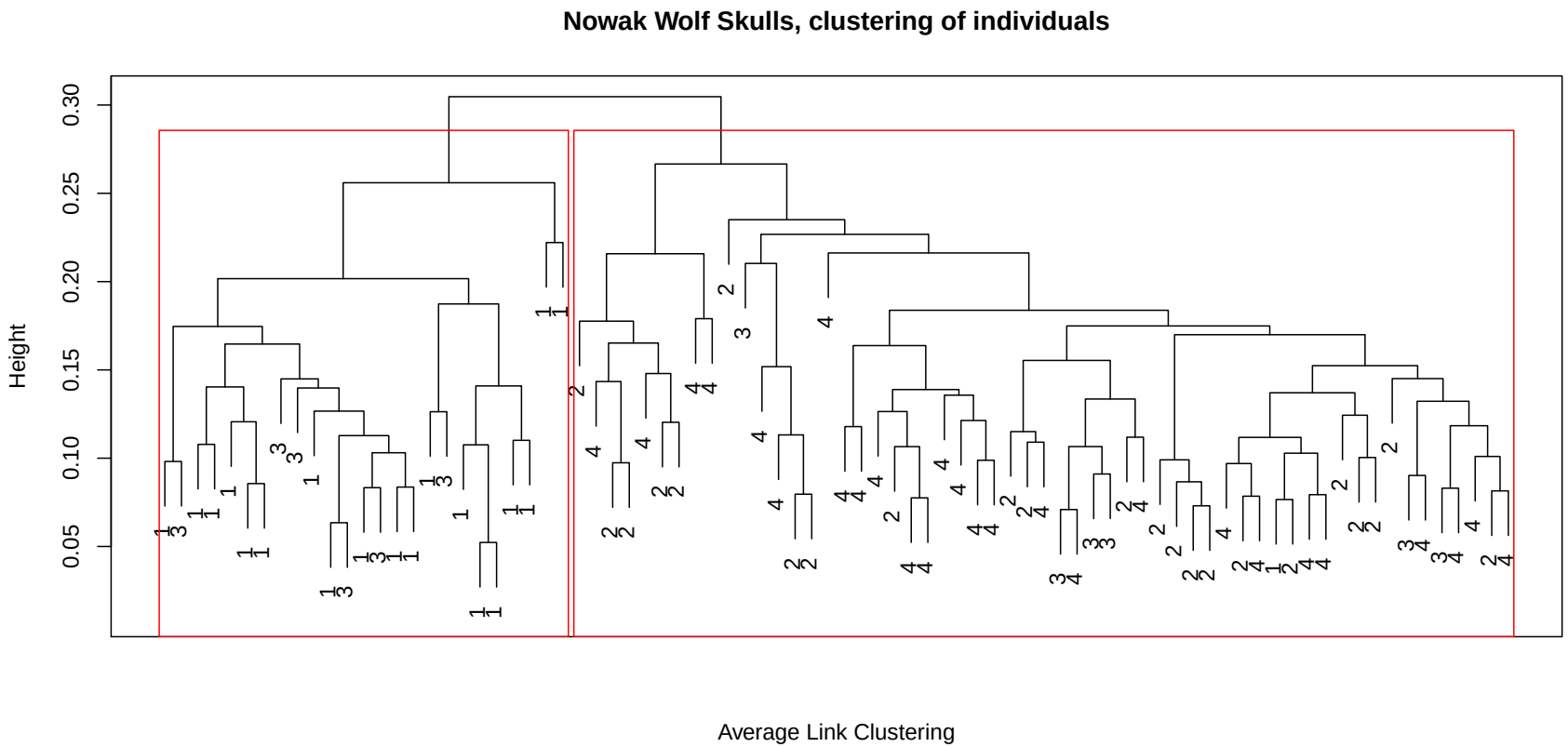
```
plot(p1$x[, 1:2], col=as.numeric(data$Source),
     pch=as.character(as.numeric(data$Source)),
     main="Two PC solution")
```



Let's see what hierarchical clustering does. The `method="average"` joins P and R if the average dissimilarity between points in P and R is minimized.

```
plot(h2<-hclust(dist(scale(data[,-1], center=FALSE)), method="average"),
    labels=substr(data$Source, 1, 1),
    frame.plot=TRUE,
    main="Nowak Wolf Skulls, clustering of individuals",
    xlab="Average Link Clustering", sub="")
rect.hclust(h2, k=2, border="red")
# confusion matrix
cluster.number <- cutree(h2, k=2)
xtabs(~cluster.number + Source, data)
```

```
               Source
cluster.number 1 Alg 2 MN70s 3 NEMNold 4 WUS
              1    19       0         6     0
              2     1      23         6    27
```



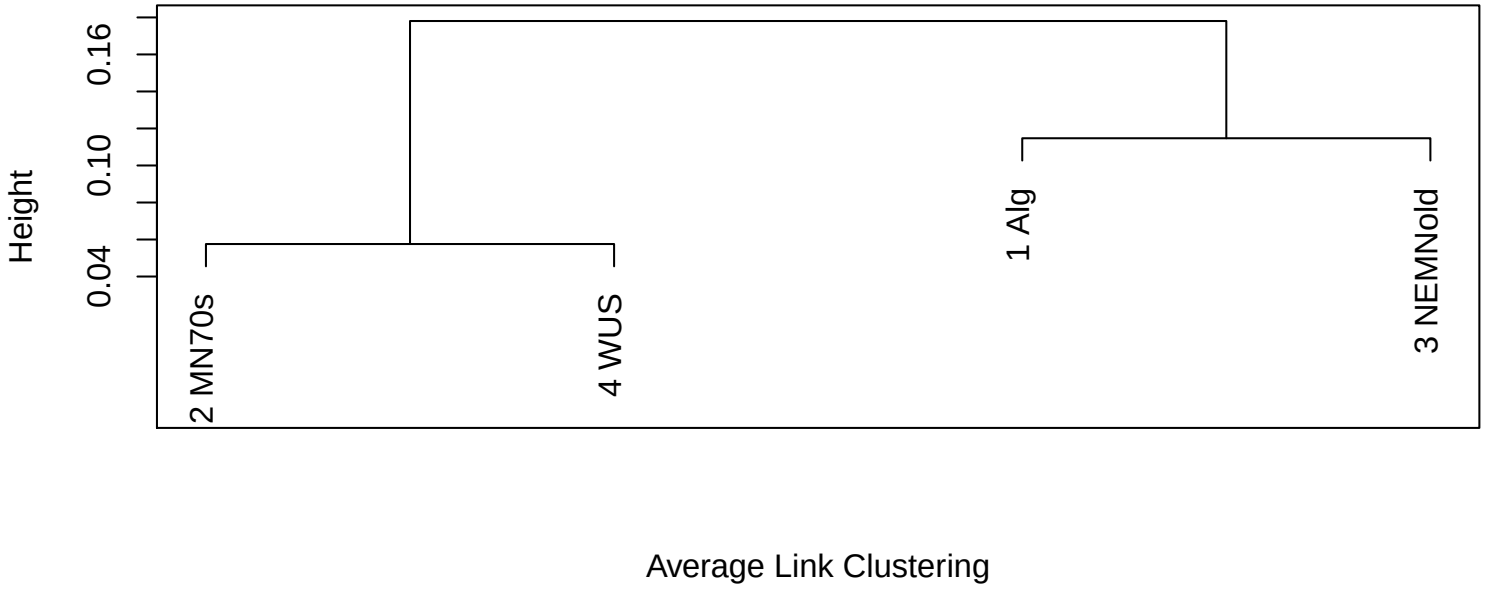
The clustering above is clustering individual skulls. Perhaps clarity could be obtained by computing means of the measurements within each of the four populations. Here is how I computed the within-population means:

```
m1 <- lm(as.matrix(data[, -1]) ~ Source -1, data)
raw.means <- coef(m1)
rownames(raw.means) <- substr(rownames(raw.means),7,20)
t(raw.means)
```

```
             1 Alg 2 MN70s 3 NEMNold  4 WUS
length      245.10  256.30    248.42 256.85
zyg_width   132.22  140.13    135.00 139.50
alve_lgnth   82.53   86.25     84.09  86.24
rostbsewid   76.20   81.82     79.12  82.34
```

```
pal_width    27.05   31.97     30.03  31.86
frshldwidth  60.69   64.30     60.86  64.57
cheekhgt     37.26   39.46     37.37  39.84
jugdepth     17.20   20.00     18.39  19.67
upcarlngth   24.51   25.09     25.07  25.70
X2upmolwth   14.31   14.20     13.85  13.39
```

```
plot(h1<-hclust(dist(scale(raw.means, center=FALSE)), method="average"),
    frame.plot=TRUE, main="",
    xlab="Average Link Clustering", sub="")
```



Lawyers' ratings of US judges

This example is included with R. A sample of 43 judges were rated on 12 characteristics. We will cluster the *variables* rather than the judges, to learn about the variables are similar over many of the judges. We can do this by using the correlations

between the characteristics as the distance. We convert from similarity to distance by subtracting the correlations from one.

The variables are

INTG Judicial integrity

DMNR Demeanor

DILG Diligence

CFMG Case flow managing

DECI Prompt decisions

PREP Preparation for trial

FAMI Familiarity with law

ORAL Sound oral rulings

WRIT Sound written rulings

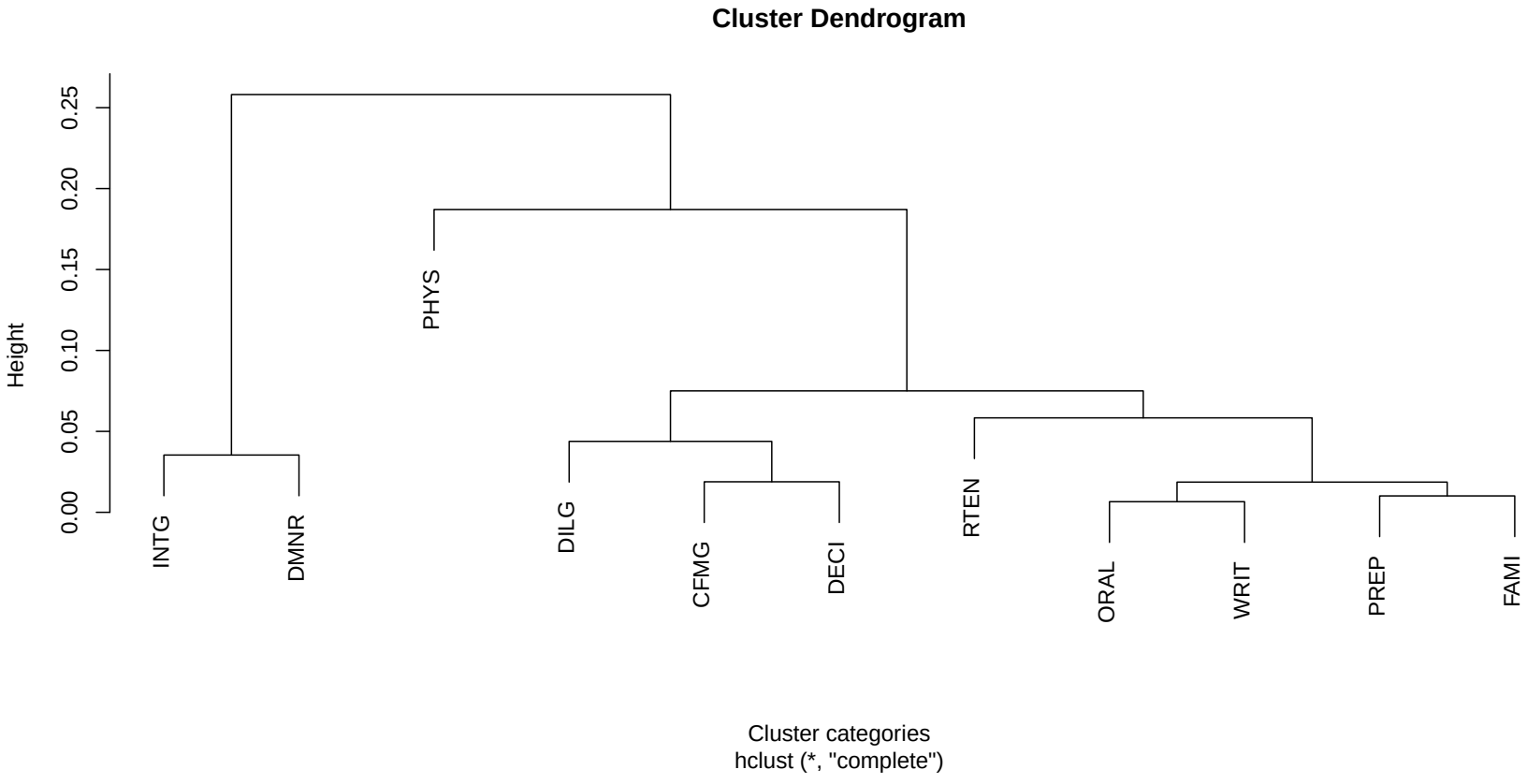
PHYS Physical ability

RTEN Worthy of retention

```
dd <- as.dist(1 - cor(USJudgeRatings[,-1]))
round(1000 * dd) # (prints more nicely)
```

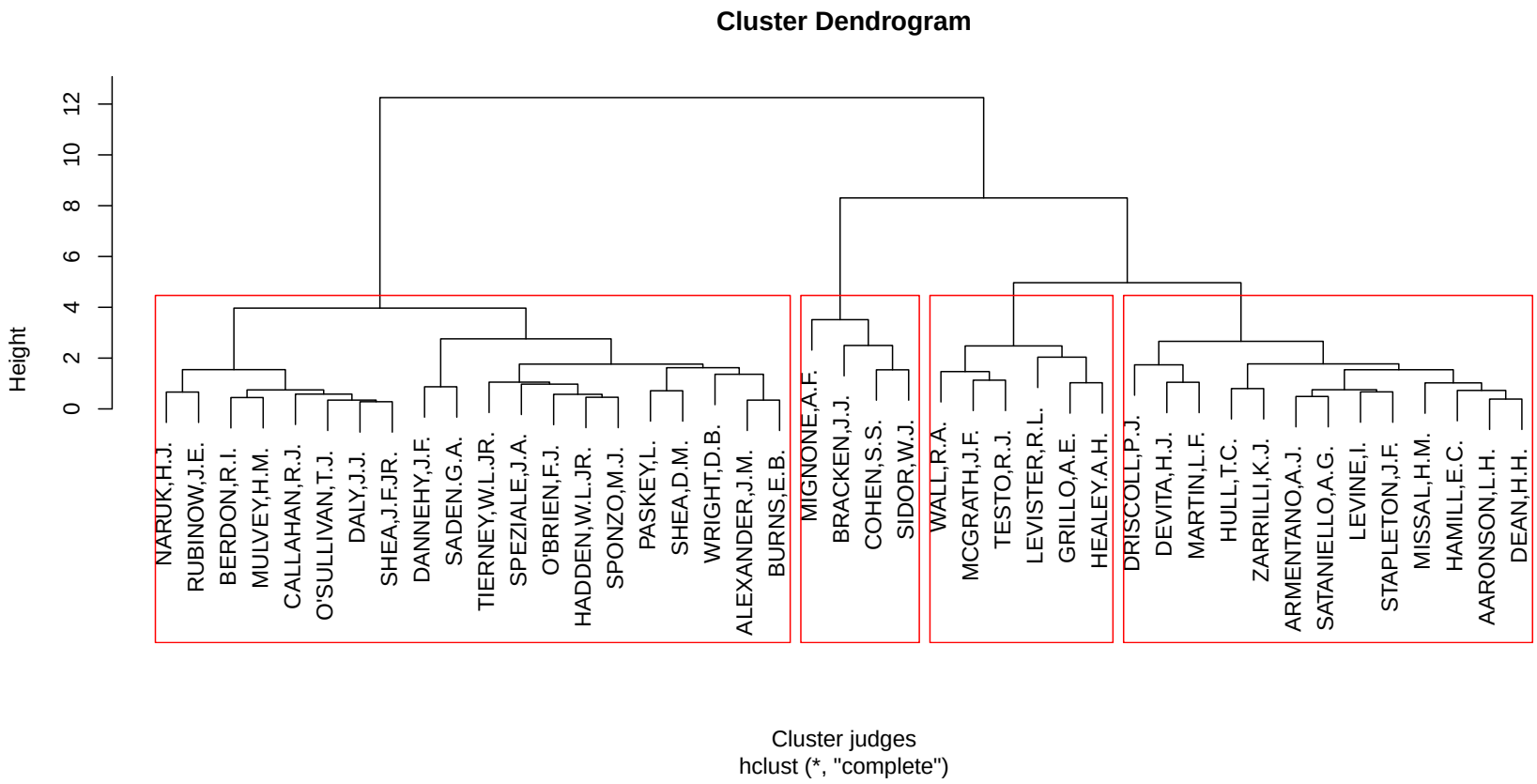
```
     INTG DMNR DILG CFMG DECI PREP FAMI ORAL WRIT PHYS
DMNR   35
DILG  128  163
CFMG  186  187   41
DECI  197  196   44   19
PREP  122  144   21   42   43
FAMI  131  159   43   65   57   10
ORAL   89   93   46   49   52   17   19
WRIT   91  107   41   58   54   13    9    7
PHYS  258  211  187  121  128  151  156  109  144
RTEN   63   56   70   73   75   50   58   18   32   93
```

```
plot(hclust(dd),xlab="Cluster categories")
```



We can also do the clustering for judges. For this direction the use of the correlation for similarity may not make any sense, so we won't use it.

```
plot(hc5<-hclust(dist(USJudgeRatings[, -1])), xlab="Cluster judges")
rect.hclust(hc5, k=4, border="red")
```



Hierarchical Divisive Clustering

Start at the top, and go to the bottom. Clearly harder.

K-means

An alternative method of clustering is called *k-means*. The k-means model is based on a normality assumption. We assume k normal populations, and

$$x_\ell | (x_\ell \in \text{cluster } j) \sim \mathrm{N}_p(\mu_j, \Sigma)$$

where the μ_j are all different. We observe only x_i, not its cluster label, so we have a missing data problem. The unconditional distribution of x_ℓ is a mixture of normal distributions with unknown mixing vector $(\pi_1, \ldots, \pi_k)$,

$$x_\ell \sim \sum \pi_j \mathrm{N}_p(\mu_j, \Sigma)$$

with $\sum \pi_j = 1$.

The k-means algorithm is very different from the clustering methods described in the book:

1. Fix k, the number of clusters. This method is not hierarchical, so a solution with k clusters is not necessarily derivable from a solution with $k-1$ or $k+1$ clusters.

2. Set the iteration counter $i = 0$, and select starting values for the k cluster centers $c_1^i, \ldots c_k^i$. For example, one could use hierarchical clustering, cut the tree to have k clusters, and compute the mean within each cluster as the c_j^i.

3. Assign observation x_ℓ to cluster j if $\ell = \arg\min_m \| x_\ell - c_m^i \|$.

4. Set $i = i + 1$ and update the cluster center c_j^i to be the average of all the observations assigned to cluster j.

5. If the cluster centers did not change at the last step, stop; else go to step 3.

There is no guarantee that this method will find the cluster centers that minimize the within-cluster sums of squares, so in some problems better answers can be obtained by repeating the algorithm with many random starts.

The use of Euclidean distance implicity assumes that the distribution of the x_ℓ in cluster j have mean μ_j and covariance matrix proportional to the identity I. One might wish to (1) scale X to have columns with the same variance and (2) replace the centered and scaled data matrix by the left singular vectors of $HXD^{-1/2}$, where H is defined as in the text to remove column means from X, although both of these concern only the *marginal distribution ignoring clusters*, rather than the *within cluster distributions*. Let's return to the Big Mac data.

```
X <- pc1$scores[,1:4]
(initial <- tapply(X, list(rep(cutree(hc3, 5), ncol(X)), col(X)), mean))
```

```
        1       2        3       4
1  1.5944  0.4419 -0.05509  0.2303
2 -1.8777  0.4315  0.25736 -0.2263
3  3.8694 -1.2220 -0.85347 -0.1351
4  0.4589 -1.2009  0.59583 -0.7492
5 -4.1973 -2.1584 -1.27685  1.3964
```

```
km <- kmeans(X, initial)
```

The magic `tapply` above computes a matrix whose columns are the means within cluster for the complete linkage clustering. I was not particularly happy with the output produced by the `summary` method for `kmeans` objects, so I wrote my own:

```
pr <- function (x, ...)
  {
      cat("K-means clustering with ", length(x$size), " clusters of sizes ",
          paste(x$size, collapse = ", "), "\n", sep = "")
      cat("\nCluster means:\n")
      print(x$centers, ...)
      cat("\nWithin cluster sum of squares by cluster:\n")
      print(x$withinss, ...)
      cat("\nAvailable components:\n")
      print(names(x))
      invisible(x)
  }
pr(km)
```

```
K-means clustering with 5 clusters of sizes 22, 25, 10, 8, 4

Cluster means:
   Comp.1  Comp.2   Comp.3   Comp.4
1  1.2719  0.6045 -0.03863  0.23364
2 -1.9260  0.4600  0.18871 -0.21132
3  3.4416 -0.7477 -0.49049  0.04018
4  0.3177 -1.0861  0.76807 -0.73054
5 -4.1973 -2.1584 -1.27685  1.39638

Within cluster sum of squares by cluster:
[1] 32.277 64.605 15.463 18.284  7.254

Available components:
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

```
table(km$cluster, cutree(hc3, 5))
```

```
    1  2  3  4  5
  1 22  0  0  0  0
  2  0 25  0  0  0
  3  5  0  5  0  0
  4  0  1  0  7  0
  5  0  0  0  0  4
```

This solution differs from the complete linkage clustering for only 6 of the cities. Let's see what happens with 25 random starts, and then draw some graphs:

```
pr(km1 <- kmeans(X, 5, nstart=25))
```

```
K-means clustering with 5 clusters of sizes 14, 12, 16, 23, 4

Cluster means:
   Comp.1  Comp.2  Comp.3   Comp.4
1  3.1502 -0.3300 -0.4916  0.25186
2 -2.1753  0.9457 -0.6634 -0.45976
3 -1.5106 -0.1659  0.9783 -0.12175
4  0.9983  0.1982  0.1869 -0.07158
5 -4.1973 -2.1584 -1.2769  1.39638

Within cluster sum of squares by cluster:
[1] 30.048 28.415 20.284 45.160  7.254

Available components:
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"
```

```
table(km1$cluster, km$cluster)
```

```
    1  2  3  4  5
  1  4  0 10  0  0
  2  0 12  0  0  0
  3  0 13  0  3  0
  4 18  0  0  5  0
  5  0  0  0  0  4
```

```
par(mfrow=c(1,2))
plot(X[, 1], X[, 2], type="n", main="Start=complete linkage")
text(X[, 1], X[, 2], cex=.8,
        substr(rownames(BigMac2003), 1, 5), col=km$cluster)
plot(X[, 1], X[, 2],type="n", main="25 random starts")
text(X[, 1], X[, 2], cex=.8,
        substr(rownames(BigMac2003), 1, 5), col=km1$cluster)
```