Hierarchical 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

```r
> 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:

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

If the argument `x` is a matrix or data frame, a matrix of distances is returned. There are lots of methods available.

**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 goal we pursue is to cluster the 64 profiles, not to cluster the genes:

```r
> load(url(loc))
> hist(as.vector(d <- dist(t(ncidata))), main="Dissimilarities")
```

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

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

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

```r
> plot(h1, frame.plot=TRUE)
```
Plot is on the next page. 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.

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

![Cluster Dendrogram](image-url)

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

![Cluster Dendrogram](image-url)

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

![Cluster Dendrogram](image-url)
From the *Economist*, March 2010 data,

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

- **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

\[
\text{TaxRate} = 100 \times \frac{\text{TeachGI} - \text{TeachNI}}{\text{TeachGI}}
\]

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.

```r
> library(alr3)
> head(BigMac2003)

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

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

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

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. For a dissimilarity measure I’ll use Euclidean distance, the default to the `dist` function. Big values mean items are dissimilar.

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

<table>
<thead>
<tr>
<th></th>
<th>Amsterdam</th>
<th>Athens</th>
<th>Auckland</th>
<th>Bangkok</th>
<th>Barcelona</th>
<th>Basel</th>
<th>Berlin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amsterdam</td>
<td>0.000</td>
<td>2.523</td>
<td>1.174</td>
<td>4.109</td>
<td>1.690</td>
<td>3.808</td>
<td>1.197</td>
</tr>
<tr>
<td>Athens</td>
<td>2.523</td>
<td>0.000</td>
<td>2.114</td>
<td>2.745</td>
<td>1.629</td>
<td>5.170</td>
<td>2.752</td>
</tr>
<tr>
<td>Auckland</td>
<td>1.174</td>
<td>2.114</td>
<td>0.000</td>
<td>3.140</td>
<td>1.042</td>
<td>4.730</td>
<td>1.976</td>
</tr>
<tr>
<td>Bangkok</td>
<td>4.109</td>
<td>2.745</td>
<td>3.140</td>
<td>0.000</td>
<td>2.837</td>
<td>6.998</td>
<td>4.346</td>
</tr>
<tr>
<td>Barcelona</td>
<td>1.690</td>
<td>1.629</td>
<td>1.042</td>
<td>2.837</td>
<td>0.000</td>
<td>4.622</td>
<td>2.163</td>
</tr>
<tr>
<td>Basel</td>
<td>3.808</td>
<td>5.170</td>
<td>4.730</td>
<td>6.998</td>
<td>4.622</td>
<td>0.000</td>
<td>3.155</td>
</tr>
<tr>
<td>Berlin</td>
<td>1.197</td>
<td>2.752</td>
<td>1.976</td>
<td>4.346</td>
<td>2.163</td>
<td>3.155</td>
<td>0.000</td>
</tr>
</tbody>
</table>

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

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

```r
> 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.

```r
> hc2 <- hclust(dist(scale(BigMac2003)), "complete")
> plot(hc2, cex=.75, xlab="2003 Big Mac Data, complete-link clustering")
```

---

1There is also a package called *cluster* and a taskview at [http://cran.r-project.org/web/views/Cluster.html](http://cran.r-project.org/web/views/Cluster.html) on clustering for more computational methods.
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.

```r
> 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 \((x_i - x_j)'(x_i - x_j)\), where the components of \(x\) all have the same variance. If \(R\) is the sample correlation matrix, it seems more reasonable to me to
use \((x_i - x_j)'R^{-1}(x_i - x_j)\) to account for covariance between the components of \(x\). We can do that by replacing the original data \(X\) by \(S^{-1/2}(X - 1\bar{x})\), or effectively replace \(X\) by its singular value decomposition.

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

```
Importance of components:

          Comp.1 Comp.2 Comp.3 Comp.4 Comp.5 Comp.6 Comp.7 Comp.8 Comp.9
Standard deviation  2.2540  1.0866  0.8962  0.8025  0.7024  0.5985  0.5585  0.3528 0.0860
Proportion of Variance 0.5645  0.1312  0.0892  0.0715  0.0548  0.0398  0.0342  0.0138 0.0008
Cumulative Proportion  0.5645  0.6957  0.7849  0.8565  0.9113  0.9511  0.9853  0.9991 1.0000
```

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

```r
> 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.
Let’s try a final approach: transform the data towards normality first:
> summary(BigPow <- powerTransform(BigMac2003, family="yjPower"))

yjPower Transformations to Multinormality

<table>
<thead>
<tr>
<th></th>
<th>Est.Power</th>
<th>Std.Err.</th>
<th>Wald Lower Bound</th>
<th>Wald Upper Bound</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigMac</td>
<td>-0.3845</td>
<td>0.1169</td>
<td>-0.6136</td>
<td>-0.1555</td>
</tr>
<tr>
<td>Bread</td>
<td>-0.1469</td>
<td>0.1439</td>
<td>-0.4289</td>
<td>0.1351</td>
</tr>
<tr>
<td>Rice</td>
<td>-0.2706</td>
<td>0.1429</td>
<td>-0.5507</td>
<td>0.0094</td>
</tr>
<tr>
<td>FoodIndex</td>
<td>0.1008</td>
<td>0.2106</td>
<td>-0.3119</td>
<td>0.5135</td>
</tr>
<tr>
<td>Bus</td>
<td>-0.8378</td>
<td>0.2727</td>
<td>-1.3723</td>
<td>-0.3033</td>
</tr>
<tr>
<td>Apt</td>
<td>0.3662</td>
<td>0.1246</td>
<td>0.1221</td>
<td>0.6104</td>
</tr>
<tr>
<td>TeachGI</td>
<td>0.0733</td>
<td>0.0649</td>
<td>-0.0539</td>
<td>0.2005</td>
</tr>
<tr>
<td>TeachNI</td>
<td>0.0149</td>
<td>0.0704</td>
<td>-0.1232</td>
<td>0.1529</td>
</tr>
<tr>
<td>TeachHours</td>
<td>1.4614</td>
<td>0.4680</td>
<td>0.5442</td>
<td>2.3786</td>
</tr>
</tbody>
</table>

Likelihood ratio tests about transformation parameters

<table>
<thead>
<tr>
<th></th>
<th>LRT</th>
<th>df</th>
<th>pval</th>
</tr>
</thead>
<tbody>
<tr>
<td>LR test, lambda = (0 0 0 0 0 0 0 0 0)</td>
<td>49.81</td>
<td>9</td>
<td>1.170e-07</td>
</tr>
<tr>
<td>LR test, lambda = (1 1 1 1 1 1 1 1 1)</td>
<td>500.86</td>
<td>9</td>
<td>0.000e+00</td>
</tr>
<tr>
<td>LR test, lambda = (-0.5 0 0 0 0 -1 0.5 0 0 1)</td>
<td>14.51</td>
<td>9</td>
<td>1.053e-01</td>
</tr>
</tbody>
</table>

> BigTran <- yjPower(BigPow$y, coef(BigPow))
> (pc2 <- princomp(BigTran, cor=TRUE))

Call:
princomp(x = BigTran, cor = TRUE)

Standard deviations:

<table>
<thead>
<tr>
<th>Comp.1</th>
<th>Comp.2</th>
<th>Comp.3</th>
<th>Comp.4</th>
<th>Comp.5</th>
<th>Comp.6</th>
<th>Comp.7</th>
<th>Comp.8</th>
<th>Comp.9</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.44835</td>
<td>1.03772</td>
<td>0.81551</td>
<td>0.63075</td>
<td>0.61163</td>
<td>0.47913</td>
<td>0.36598</td>
<td>0.35336</td>
<td>0.05809</td>
</tr>
</tbody>
</table>

9 variables and 69 observations.

> hc4 <- hclust(dist(scale(pc2$scores[, 1:4])), "complete")
> plot(hc4, cex=.6, xlab="Normalized Big Mac data")

8
> plot(pc2$scores[,1], pc2$scores[,2], type="n")
> text(pc2$scores[,1], pc2$scores[,2],
+  substr(rownames(BigMac2003), 1, 5), col=cutree(hc4, 5))
> legend("bottomleft", paste("Clust", 1:5), col=1:5, pch="x", cex=0.9)
> pc2$loadings[, 1]

<table>
<thead>
<tr>
<th>BigMac</th>
<th>Bread</th>
<th>Rice</th>
<th>FoodIndex</th>
<th>Bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.38</td>
<td>-0.15</td>
<td>-0.27</td>
<td>0.1</td>
<td>-0.84</td>
</tr>
<tr>
<td>-0.37869</td>
<td>-0.29818</td>
<td>-0.32090</td>
<td>0.33495</td>
<td>0.35053</td>
</tr>
<tr>
<td>Apt</td>
<td>TeachGI</td>
<td>TeachNI</td>
<td>TeachHours</td>
<td></td>
</tr>
<tr>
<td>0.37</td>
<td>0.07</td>
<td>0.01</td>
<td>1.46</td>
<td></td>
</tr>
<tr>
<td>0.33669</td>
<td>0.39676</td>
<td>0.39452</td>
<td>0.05671</td>
<td></td>
</tr>
</tbody>
</table>

Cluster Dendrogram
Normalized Big Mac data
hclust (", "complete")

pc2$scores[, 1]
pc2$scores[, 2]

> pc2$scores[, 1]
The `cuttree` function returns a vector of cluster assignments, and the `table` function shows how the last two analyses differ.

```r
> table(cutree(hc2, 5), cutree(hc4, 5))

          1  2  3  4  5
1 14  5  1  6  1
2  6 12 10  0  0
3  4  0  0  3  2
4  0  3  1  0  0
5  0  0  0  0  1
```

### 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.

```r
> 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 0.20 -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 1.82  14.00  13.91 1.82  12.3  16.1  3.8 0.08 -0.58  0.09
```

```r
> 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.
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

```r
> p1 <- prcomp(data[, -1], scale=TRUE)
> summary(p1)
```
Importance of components:

<table>
<thead>
<tr>
<th></th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
<th>PC8</th>
<th>PC9</th>
<th>PC10</th>
</tr>
</thead>
<tbody>
<tr>
<td>SD</td>
<td>2.375</td>
<td>1.089</td>
<td>0.864</td>
<td>0.824</td>
<td>0.768</td>
<td>0.619</td>
<td>0.505</td>
<td>0.469</td>
<td>0.438</td>
<td>0.323</td>
</tr>
<tr>
<td>Var</td>
<td>0.564</td>
<td>0.119</td>
<td>0.074</td>
<td>0.068</td>
<td>0.059</td>
<td>0.038</td>
<td>0.025</td>
<td>0.022</td>
<td>0.019</td>
<td>0.011</td>
</tr>
<tr>
<td>Cumulative Var</td>
<td>0.564</td>
<td>0.683</td>
<td>0.758</td>
<td>0.825</td>
<td>0.884</td>
<td>0.923</td>
<td>0.948</td>
<td>0.970</td>
<td>0.989</td>
<td>1.000</td>
</tr>
</tbody>
</table>

> 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)

<table>
<thead>
<tr>
<th>Source</th>
<th>cluster.number</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alg</td>
<td>1</td>
<td>19</td>
<td>0</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>MN70s</td>
<td>2</td>
<td>1</td>
<td>23</td>
<td>6</td>
<td>27</td>
</tr>
</tbody>
</table>
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:

```r
> 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
```

```r
> 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
```
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(hclust(dist(USJudgeRatings[, -1])), xlab="Cluster judges")
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

\[ x_\ell | (x_\ell \in \text{cluster } j) \sim N_p(\mu_j, \Sigma) \]

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

\[ x_\ell \sim \sum \pi_j N_p(\mu_j, \Sigma) \]

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 \( x_\ell \) 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 implicitly assumes that the distribution of the \( x_\ell \) in cluster \( j \) have mean \( \mu_j \) 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 \( HX \mathbf{D}^{-1/2} \), 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.

\[ X \leftarrow \text{pc2}$scores[,1:4] \]
\[ \text{(initial} \leftarrow \text{tapply}(X, \text{list(rep(cutree(hc4, 5), ncol(X)), col(X)), mean)) \]

\[
\begin{pmatrix}
 1 & 2 & 3 & 4 \\
 1 & 1.124 & -0.07033 & 0.4799 & 0.4358 \\
 2 & -1.786 & 0.39416 & -0.5152 & 0.2917 \\
 3 & -1.776 & -0.13495 & 0.4649 & -0.7335 \\
 4 & 2.471 & 0.40870 & -0.3091 & -0.7576 \\
 5 & 1.956 & -2.06355 & -1.0025 & -0.1680
\end{pmatrix}
\]

\[ \text{km} \leftarrow \text{kmeans}(X, \text{initial}) \]

The magic \text{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 \text{summary} method for \text{kmeans} objects, so I wrote my own:

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

K-means clustering with 5 clusters of sizes 16, 15, 16, 14, 8

Cluster means:

\[
\begin{pmatrix}
\text{Comp.1} & \text{Comp.2} & \text{Comp.3} & \text{Comp.4} \\
1 & 1.321 & 0.5774 & -0.3186 & 0.10578 \\
2 & -3.077 & 0.3112 & -0.5086 & 0.31833 \\
3 & -1.836 & -0.13495 & 0.7125 & -0.24441 \\
4 & 3.040 & 0.2348 & 0.3261 & -0.08464 \\
5 & 1.479 & -1.6076 & -0.4048 & -0.17150
\end{pmatrix}
\]
Within cluster sum of squares by cluster:
[1] 22.91 42.46 35.02 18.21 23.17

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

> table(km$cluster, cutree(hc4, 5))

1 2 3 4 5
1 9 4 0 3 0
2 0 13 2 0 0
3 6 2 8 0 0
4 8 0 0 6 0
5 1 1 2 0 4

This solution differs from the complete linkage clustering for about half 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 11, 17, 19, 14, 8

Cluster means:

Comp.1 Comp.2 Comp.3 Comp.4
1 -1.030 -0.89449 0.55012 -0.30948
2 2.971 -0.07407 0.19413 -0.04819
3 1.302 0.33621 -0.33982 0.03792
4 -3.095 -0.44464 0.09976 0.02253
5 -2.573 1.36696 -0.53646 0.39843

Within cluster sum of squares by cluster:
[1] 32.70 33.52 31.73 29.91 12.60

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

> table(km1$cluster, km$cluster)

1 2 3 4 5
1 0 0 9 0 2
2 0 0 0 14 3
3 16 0 0 0 3
4 0 8 6 0 0
5 0 7 1 0 0

> 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)