

Stat 8053, Chapter 11 Cluster Analysis, rev December 1, 2011



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

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

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.

```
> library(alr3)
> 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. For a dissimilarity measure I'll use Euclidean distance, the default to the `dist` function; other choices `c("maximum", "maximum", "canberra", "binary", "minkowski")`, some of which are described in the text, and all of which are defined in the help page `?dist`. Big values mean items are dissimilar.

```
> 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:

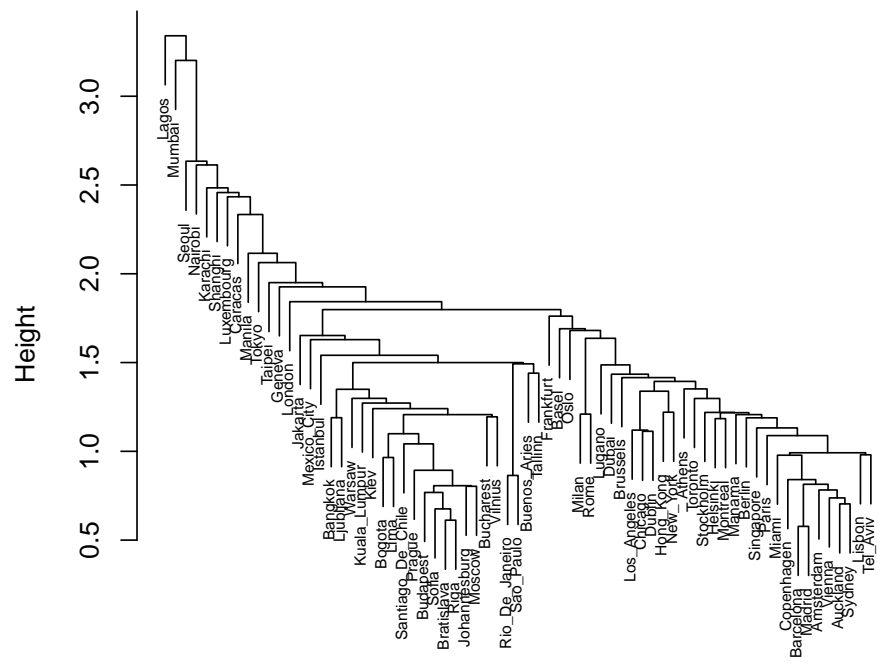
```
> as.matrix(dist(BigMac2003))[1:7,1:7]
```

	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 used¹

```
> hc <- hclust(dist(scale(BigMac2003)), "single")
> plot(hc, cex=.55, xlab="2003 Big Mac Data, single-link clustering")
```

Cluster Dendrogram



2003 Big Mac Data, single-link clustering
`hclust (*, "single")`

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

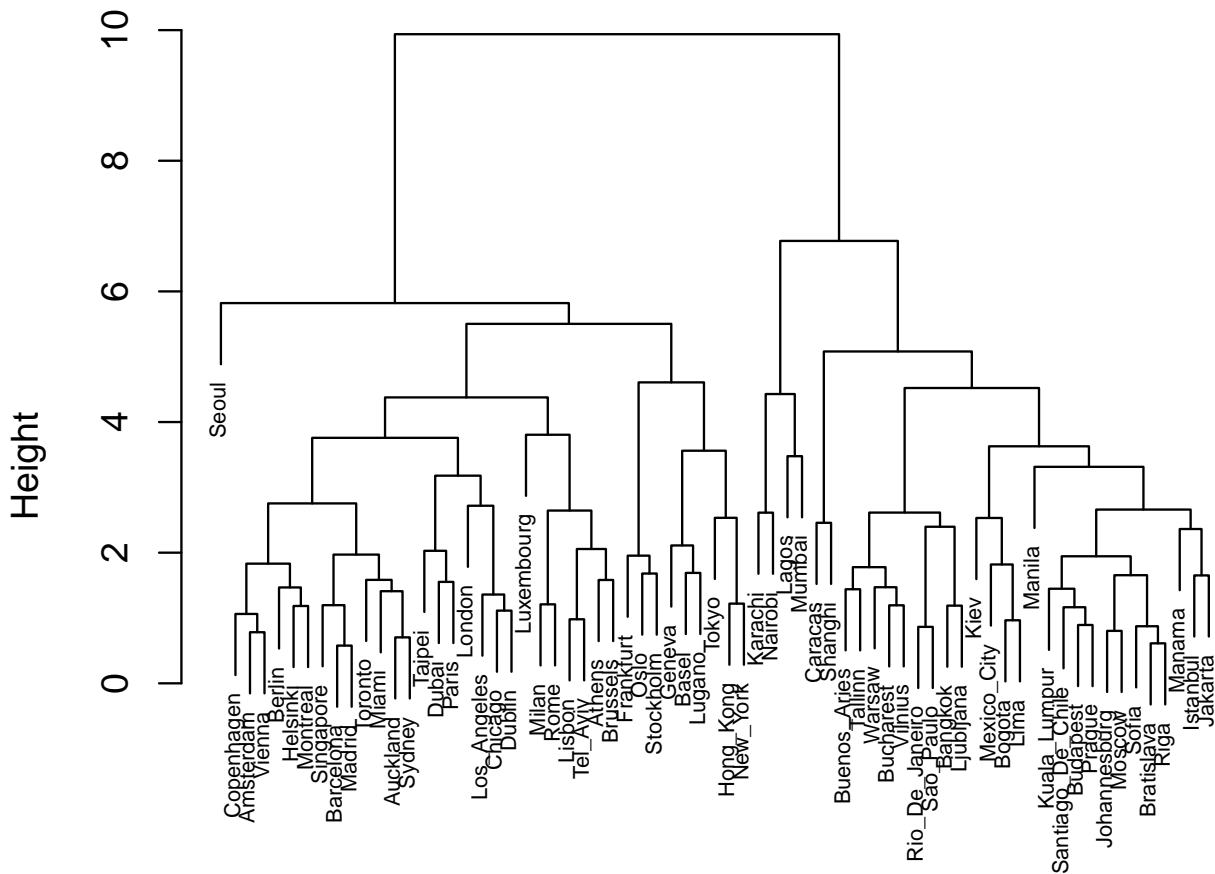
¹There is also a package called `cluster` and a taskview at <http://cran.r-project.org/web/views/Cluster.html> on clustering for more computational methods.

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=.55, xlab="2003 Big Mac Data, complete-link clustering")
```

Cluster Dendrogram



2003 Big Mac Data, complete-link clustering
hclust (*, "complete")

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

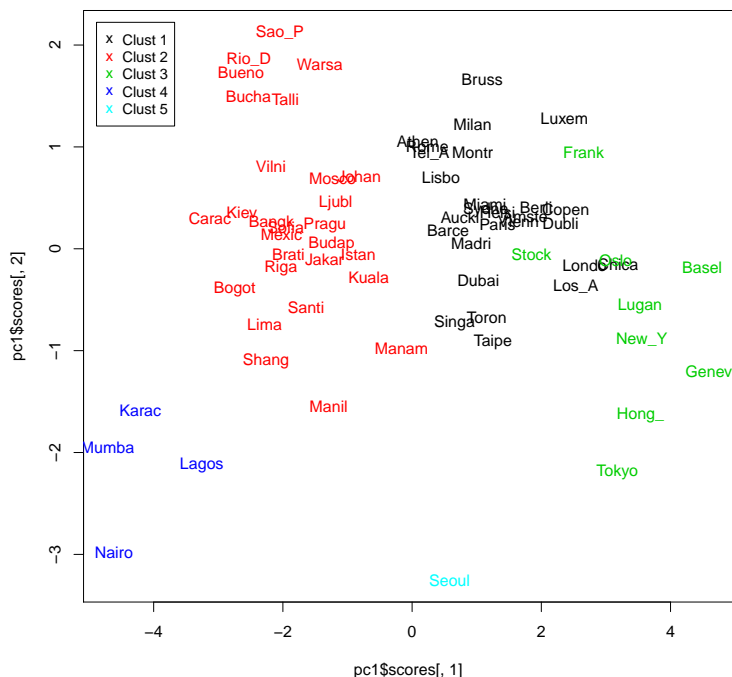
Let's see how this corresponds to Principal Component Analysis. In the plot below I've colored the data points corresponding to the complete linkage clustering with five clusters.

```
> 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 $(x_i - x_j)'(x_i - x_j)$, 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.

```

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

```

Importance of components:

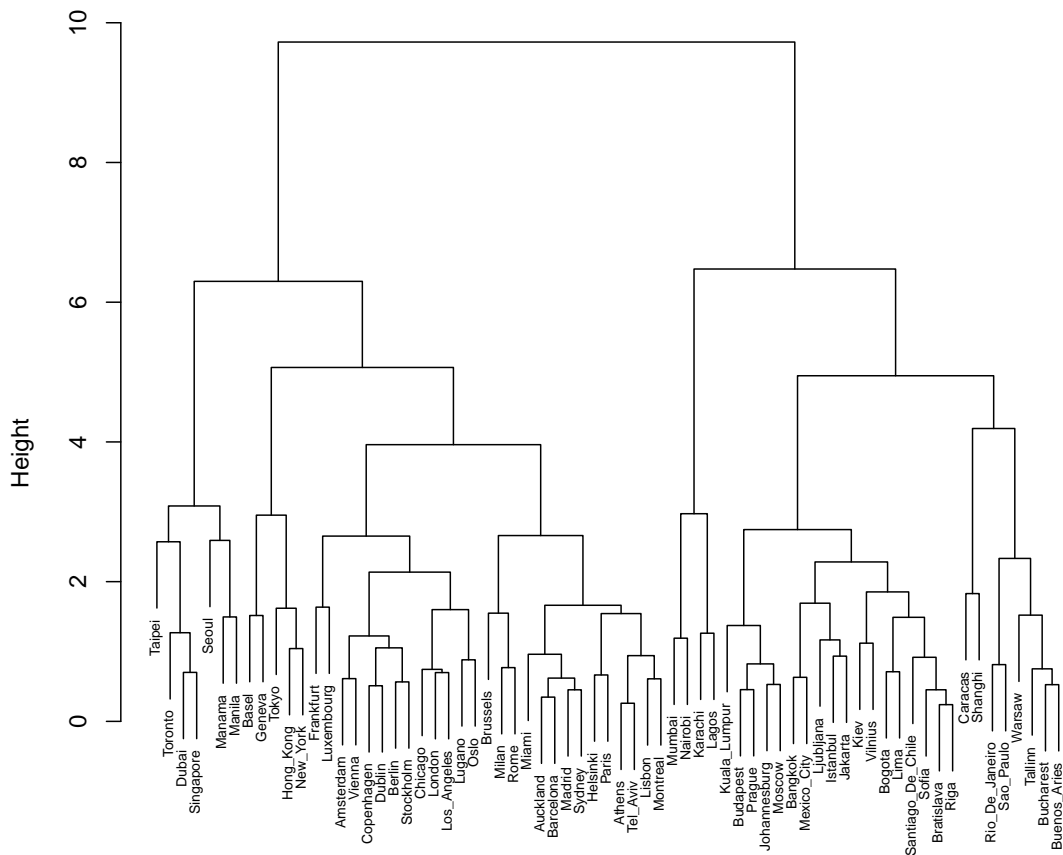
	Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8
Standard deviation	2.2540	1.0866	0.89624	0.80248	0.70239	0.59846	0.55500	0.35284
Proportion of Variance	0.5645	0.1312	0.08925	0.07155	0.05482	0.03979	0.03423	0.01383
Cumulative Proportion	0.5645	0.6957	0.78496	0.85651	0.91132	0.95112	0.98534	0.99918
	Comp.9							
Standard deviation	0.0860700							
Proportion of Variance	0.0008231							
Cumulative Proportion	1.0000000							

```

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

```

Cluster Dendrogram



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.

Let's try a final approach: transform the data towards normality first:

```
> summary(BigPow <- powerTransform(BigMac2003, family="yjPower"))
```

yjPower Transformations to Multinormality

Est.Power Std.Err. Wald Lower Bound Wald Upper Bound

BigMac	-0.3845	0.1169	-0.6136	-0.1555
Bread	-0.1469	0.1439	-0.4289	0.1351
Rice	-0.2706	0.1429	-0.5507	0.0094
FoodIndex	0.1008	0.2106	-0.3119	0.5135
Bus	-0.8378	0.2727	-1.3723	-0.3033
Apt	0.3662	0.1246	0.1221	0.6104
TeachGI	0.0733	0.0649	-0.0539	0.2005
TeachNI	0.0149	0.0704	-0.1232	0.1529
TeachHours	1.4614	0.4680	0.5442	2.3786

Likelihood ratio tests about transformation parameters

	LRT	df	pval
LR test, lambda = (0 0 0 0 0 0 0 0 0)	49.81	9	1.170e-07
LR test, lambda = (1 1 1 1 1 1 1 1 1)	500.86	9	0.000e+00
LR test, lambda = (-0.5 0 0 0 -1 0.5 0 0 1)	14.51	9	1.053e-01

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

Call:

```
princomp(x = BigTran, cor = TRUE)
```

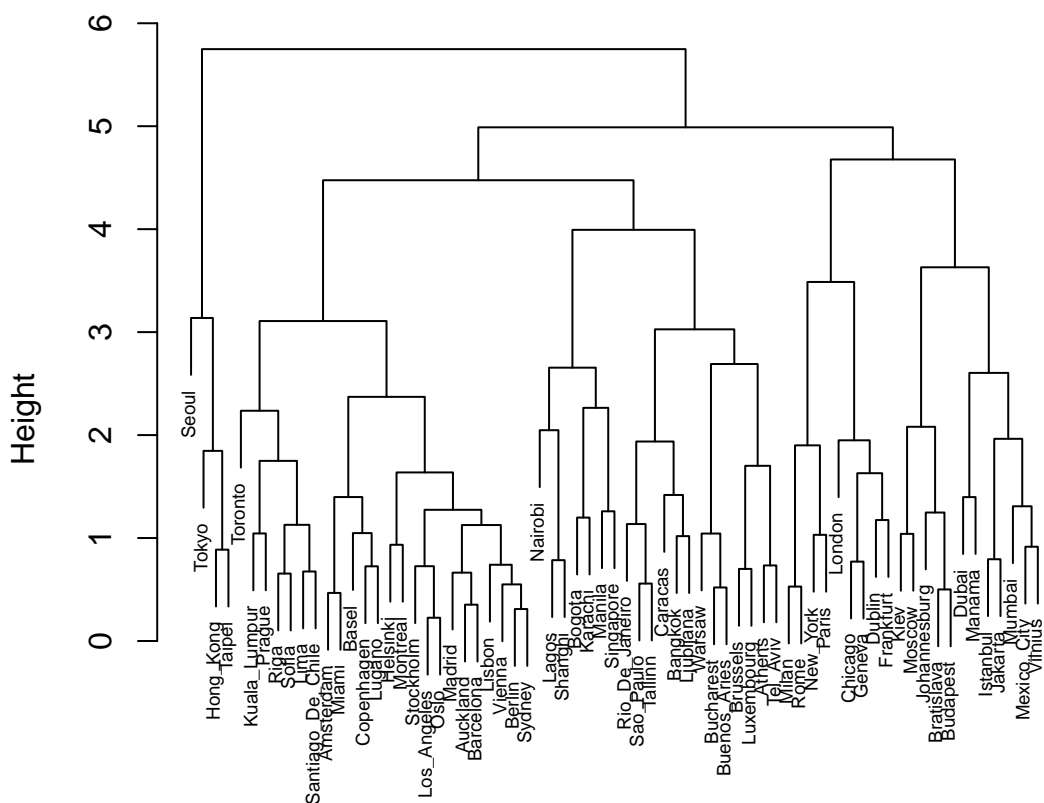
Standard deviations:

Comp.1	Comp.2	Comp.3	Comp.4	Comp.5	Comp.6	Comp.7	Comp.8	Comp.9
2.44835	1.03772	0.81551	0.63075	0.61163	0.47913	0.36598	0.35336	0.05809

9 variables and 69 observations.

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

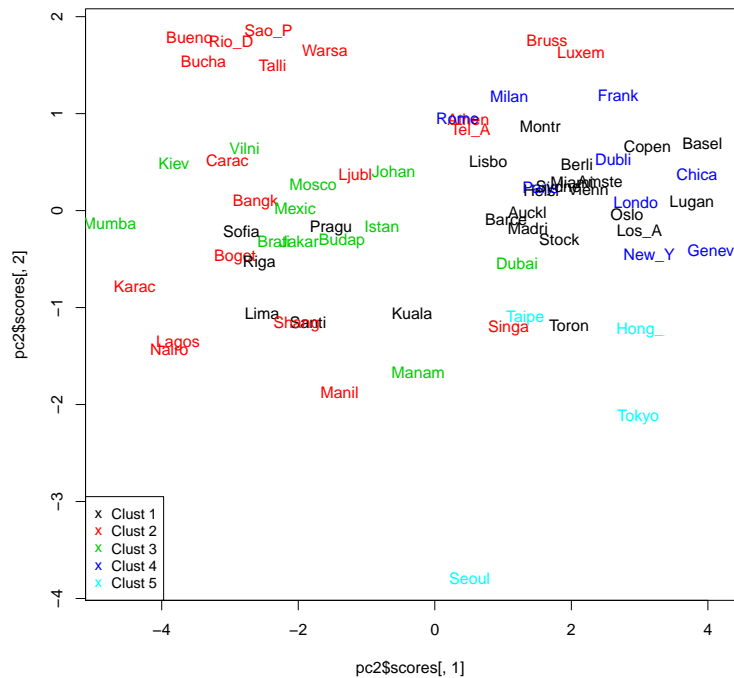
Cluster Dendrogram



Normalized Big Mac data
hclust (*, "complete")

```
> 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]
```

BigMac ^{-0.38}	Bread ^{-0.15}	Rice ^{-0.27}	FoodIndex ^{0.1}	Bus ^{-0.84}
-0.37869	-0.29818	-0.32090	0.33495	0.35053
Apt ^{0.37}	TeachGI ^{0.07}	TeachNI ^{0.01}	TeachHours ^{1.46}	
0.33669	0.39676	0.39452	0.05671	



The `cuttree` function returns a vector of cluster assignments, and the `table` function shows how the last two analyses differ.

```
> 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.

```
> 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	-1.53	0.13
length	2	82	252.60	8.23	253.00	252.32	7.41	235.0	274.0	39.0	0.25	-0.07	0.91
zyg_width	3	82	137.25	5.91	136.50	136.97	5.19	125.4	152.0	26.6	0.42	-0.05	0.65
alve_lgnth	4	82	85.02	3.04	85.00	84.97	2.30	77.0	93.1	16.1	0.20	0.32	0.34
rostrbsewid	5	82	80.23	3.87	80.00	80.18	4.23	72.2	89.1	16.9	0.13	-0.34	0.43
pal_width	6	82	30.45	2.74	30.65	30.55	3.04	24.1	35.3	11.2	-0.29	-0.67	0.30
frshldwidth	7	82	63.00	4.39	63.20	62.82	4.37	52.5	73.3	20.8	0.31	0.10	0.48
cheekhgt	8	82	38.74	2.37	39.00	38.75	2.00	34.3	44.0	9.7	-0.09	-0.37	0.26

```

jugdepth      9 82  18.97 1.57  19.05   18.97 1.41  14.7  23.6   8.9 0.02   0.52 0.17
upcarlength  10 82  25.15 1.12  25.00   25.08 1.19  23.1  28.5   5.4 0.58   0.04 0.12
X2upmolwth   11 82  13.91 0.82  14.00   13.91 0.82  12.3  16.1   3.8 0.08  -0.48 0.09

```

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

Source

```

  1 Alg   2 MN70s 3 NEMNoId   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)

```

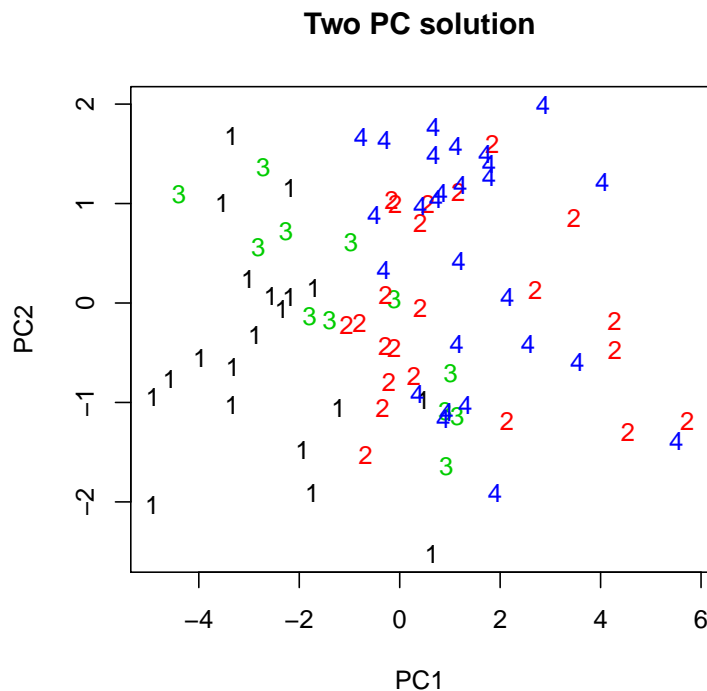
Importance of components:

	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")

```

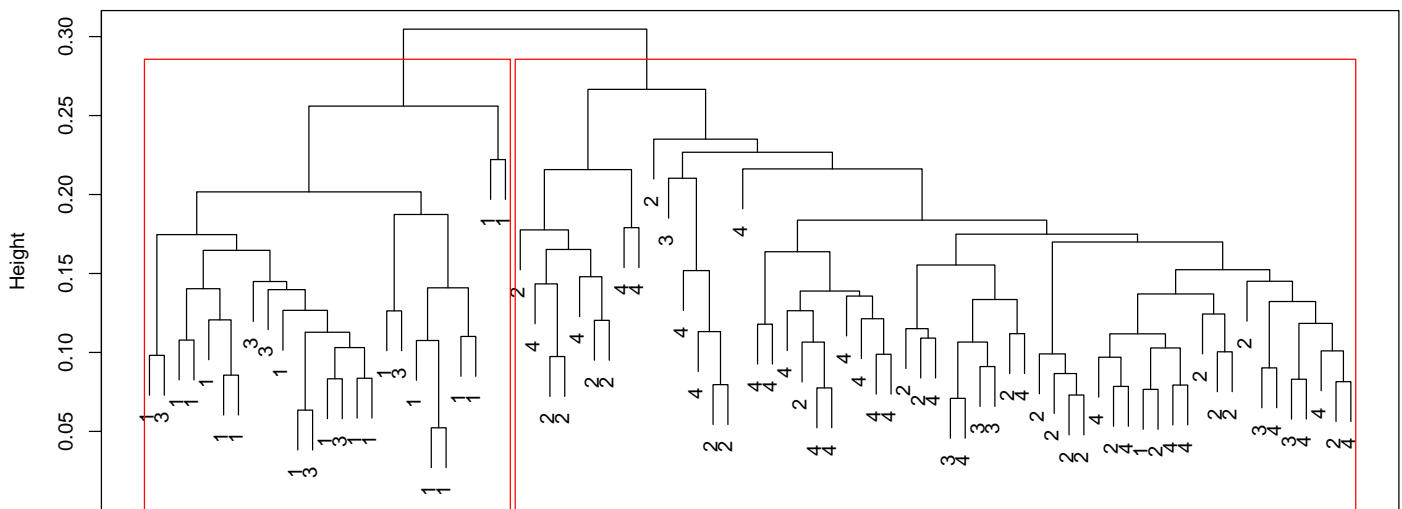


Let's see what hierarchical clustering does. The method="average" joins P and R if the average dissimilarity between points in P and R is minimized.

```
> 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 NEMNoId	4 WUS
1	19	0	6	0
2	1	23	6	27

Nowak Wolf Skulls, clustering of individuals



Average Link Clustering

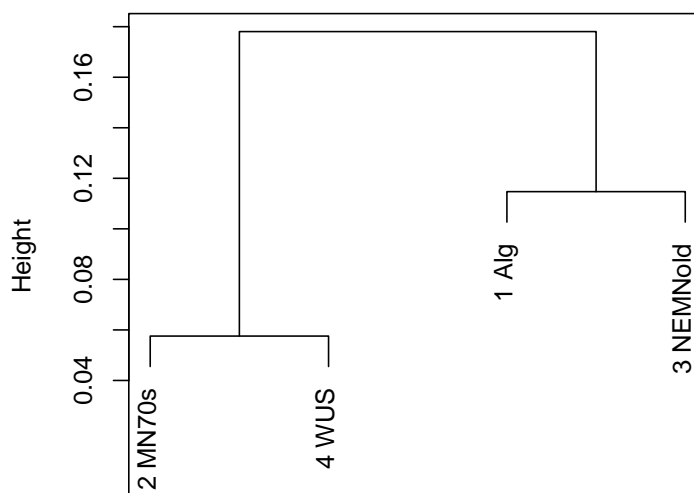
The clustering above is clustering individual skulls. Perhaps clarity could be obtained by computing means of the measurements within each of the four populations. Here is how I computed the within-population means:

```
> 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 NEMNoId	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
rostrbsewid	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
upcarlength	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="")
```



Average Link Clustering

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

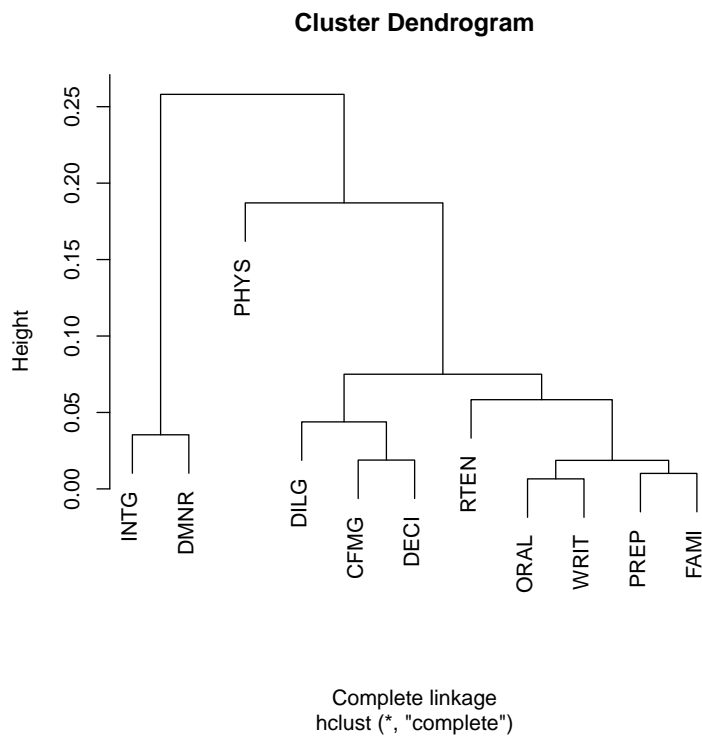
PHYS Physical ability

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="Complete linkage")
```



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 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 (π_1, \dots, π_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, \dots, 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 implicitly 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 $HXD^{-1/2}$, 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 <- pc2$scores[,1:4]
> (initial <- tapply(X, list(rep(cutree(hc4, 5), ncol(X)), col(X)), mean))
```

```
      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
```

```
> 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 16, 15, 16, 14, 8

Cluster means:

	Comp.1	Comp.2	Comp.3	Comp.4
1	1.321	0.5774	-0.3186	0.10578
2	-3.077	0.3112	-0.5086	0.31833
3	-1.836	-0.2708	0.7125	-0.24441
4	3.040	0.2348	0.3261	-0.08464
5	1.479	-1.6076	-0.4048	-0.17150

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, 8, 19, 14, 17

Cluster means:

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

Within cluster sum of squares by cluster:

[1] 32.70 12.60 31.73 29.91 33.52

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

```
> par(mfrow=c(1,2))
```

```
> plot(X[, 1], X[, 2], type="n", main="Start=complete linkage")
```

```
> text(X[, 1], X[, 2], cex=.8,
```

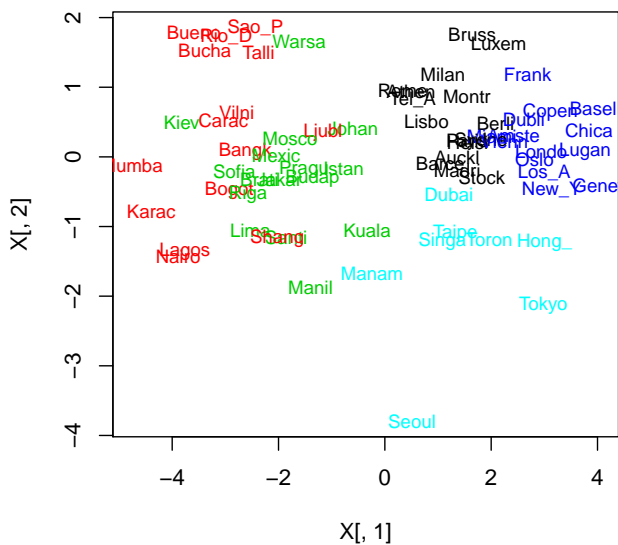
```
+ substr(rownames(BigMac2003), 1, 5), col=km$cluster)
```

```
> 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)
```

Start=complete linkage



25 random starts

