Clustering in R

Note: These notes are only about doing (heirarchical) clustering in R. The nuts-and-bolts will be covered in lecture, but won't be posted in on-line notes.

Clustering can be done on objects or variables. When you read a data_table into R, we assume the rows represent objects, columns variables. We'll assume here that you want to cluster objects. If not, read in the data_table and do a transpose before proceeding. (The transpose function is, simply, "t(matrix)".

Step I

If you have a dataset of variables, the first step is to create "distances" between the rows (objects). This is done with the "dist" function, which requires:

```
 dist(x, method = "euclidean", diag = FALSE, upper = FALSE)
```

Method can be

euclidean

maximum max distance between components of vectors x and y

canberra `sum(|x_i - y_i| / |x_i + y_i|)`

`manhattan absolute distance between two vectors`

binary "The vectors are regarded as binary bits, so non-zero elements are `on' and zero elements are `off'. The distance is the proportion of bits in which only one is on amongst those in which at least one is on"

"euclidian" is the default.

Example

Remember the USArrests data sets? It contained data on the number of arrests per 100,000 residents for the 50 states on three different crimes. Here are the first two rows:

```
> USArrests[c(1,2),]
        Murder Assault UrbanPop Rape
Alabama   13.2     236       58 21.2
Alaska    10.0     263       48 44.5
```

And here are their distances:

```
> dist(USArrests[c(1,2),])
[1] 37.17701
> dist(USArrests[c(1,2),],method="maximum")
[1] 27
> dist(USArrests[c(1,2),],method="canberra")
```

```
[1] 0.6410212
> dist(USArrests[c(1,2),],method="manhattan")
[1] 63.5
> dist(USArrests[c(1,2),],method="binary")
[1] 0
```

If you already have a matrix of distances, the function as.dist will convert it to the proper form.

Example:
For 10 European languages, the words for the numbers 1 through 10 were printed out. Their "similarities" were measured by counting the number of words in which the first letter were the same. For example, each language has a similarity of 10 with itself. English and Dutch have a similarity of 3, since the words for "two" ("twee"), "nine" ("negen"), and "ten" ("tien") have the same first letter, and none of the other words do.

The matrix looks like this:

E	N	DA	DU	G	FR	SP	I	P	H	FI
10	8	8	3	4	4	4	4	3	1	1
8	10	9	5	6	4	4	4	3	2	1
8	9	10	4	5	4	5	5	4	2	1
3	5	4	10	5	1	1	1	0	2	1
4	6	5	5	10	3	3	3	2	1	1
4	4	4	1	3	10	8	9	5	0	1
4	4	5	1	3	8	10	9	7	0	1
4	4	5	1	3	9	9	10	6	0	1
3	3	4	0	2	5	7	6	10	0	1
1	2	2	2	1	0	0	0	0	10	2
1	1	1	1	1	1	1	1	1	2	10

The following commands put this into a form that R likes. Assume that there is a file called "language" that contains this matrix, tab-delimited.

```
language <- read.table(“language”, header=T)
ldist <- as.dist(language)
names(ldist) <- c("E", "N", "DA", "DU","G", "FR", "SP", "I", "P",
"H", "FI")
```

There's one last step. This matrix has big numbers for objects that are the most similar, and 0's for those that are least similar. This is the reverse of a "distance" measure. To convert this to a distance measure:

```
ldist <- 10 - ldist
```

Now the most similar are 0's, and the least similar are 10's.

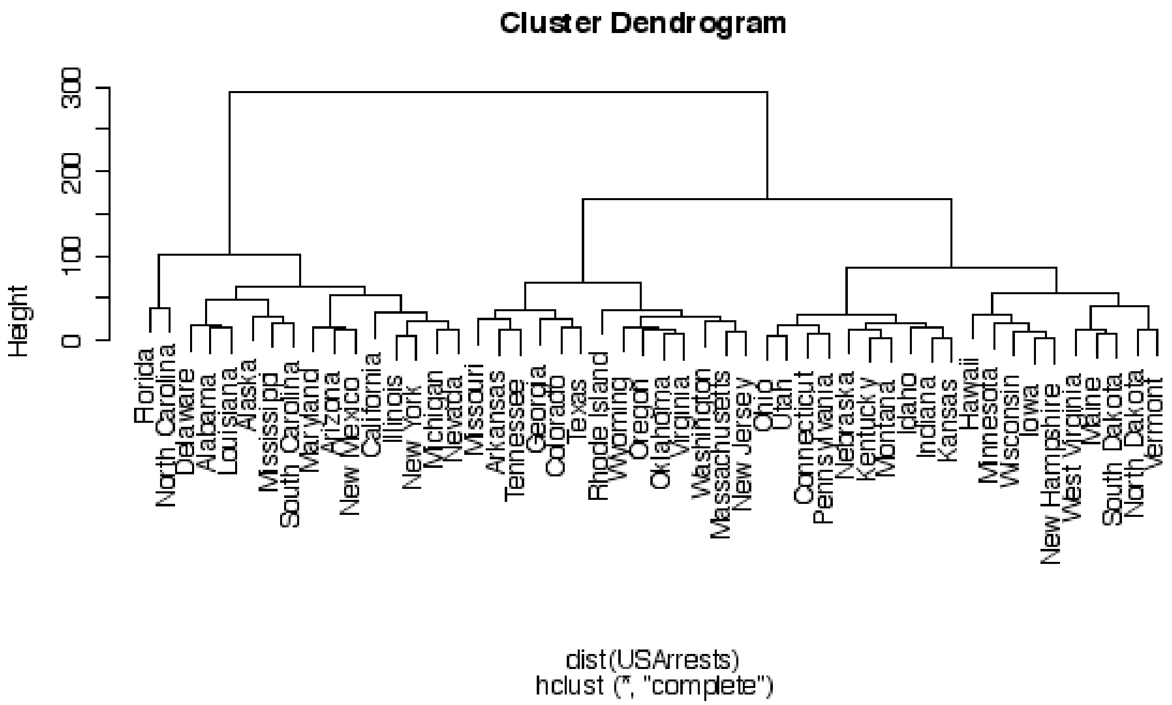
Step II

The “hclust” command allows for a variety of heirarchical clustering methods. The call looks like hclust(distance_object, method=”complete”). Here are the methods:

complete	max distance between clusters. The default
ward	error sum of squares criteron
single	minimum distrance between clusters
average	the average distance between all possible pairs
mcquitty	don’t know
median	not certain, but I think the median of all possible pairs
centroid	probably the distance between the center of each cluster

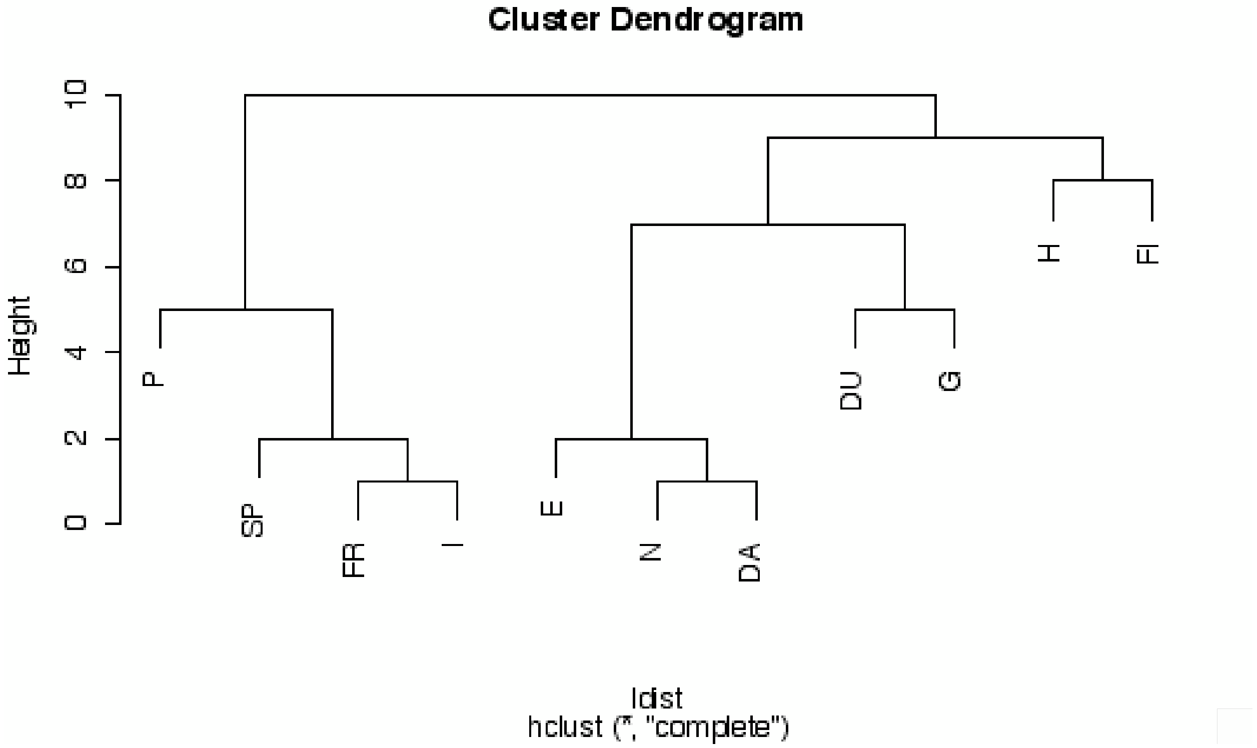
The most useful output is the “plot” function, which gives the dendogram. Here are two dendograms:

```
plot(hclust(dist(USArrests)))
```



This is hard to interpret because it’s hard to read, but it does tell us that North Dakota is more like Vermont than South Dakota.

```
plot(hclust(ldist))
```



This tells us that Finnish and Hungarian are more like each other than the others. Danish and Norweigian are alike, and English is then more like them than it is like others. French and Italian are closely related, etc.