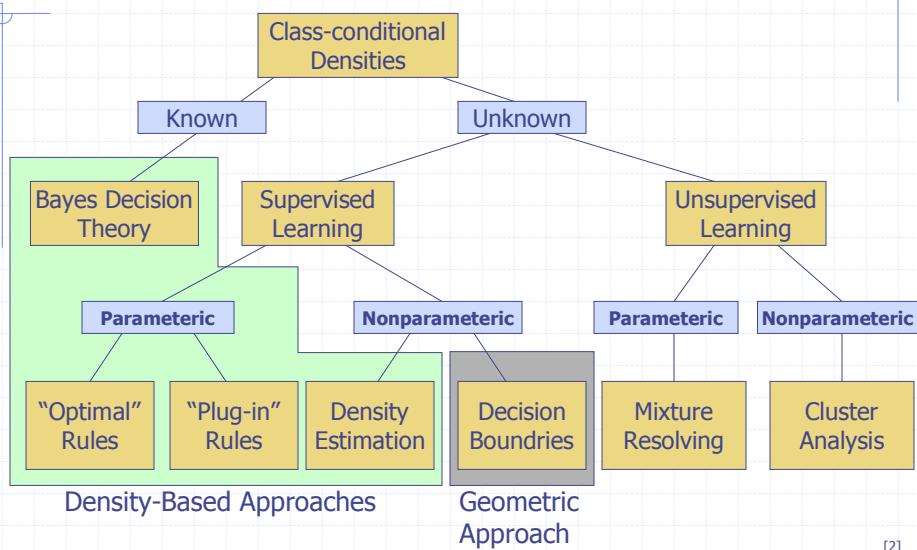


Unsupervised Learning: Clustering

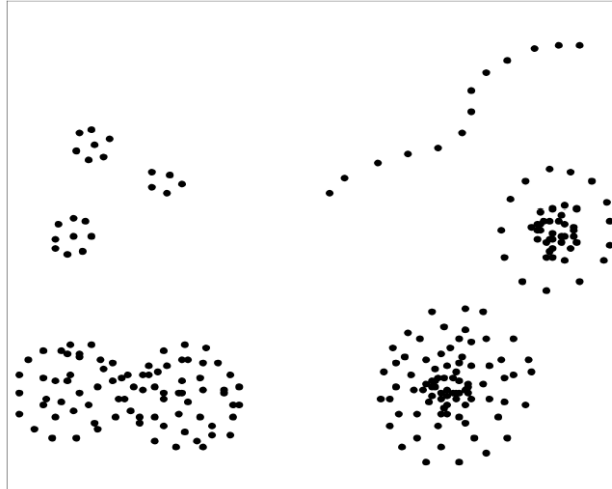
Randy Julian

Lilly Research Laboratories

Various approaches



Clusters - All shapes and sizes...

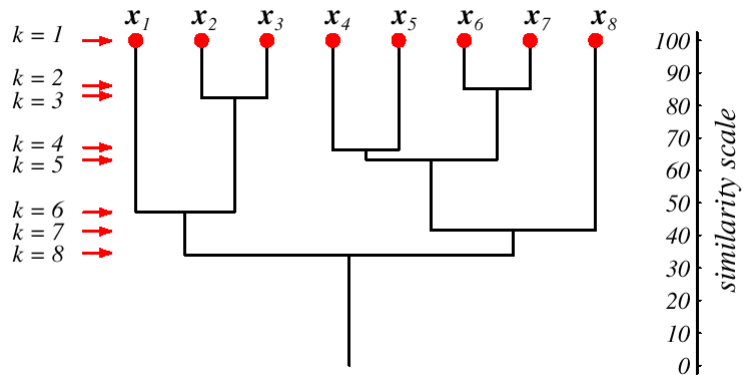


Cluster Analysis Methods

◆ Hierarchical

- Agglomerative (bottom-up)
 - ◆ Start at bottom and merge a selected pair of clusters into a single cluster
 - ◆ Pair to merge is selected as having smallest intergroup dissimilarity
- Divisive (top-down)
 - ◆ Start at the top and split one cluster into two new clusters
 - ◆ Split is chosen to produce two new groups with largest between-group dissimilarity

HCA - Hierarchical Clustering



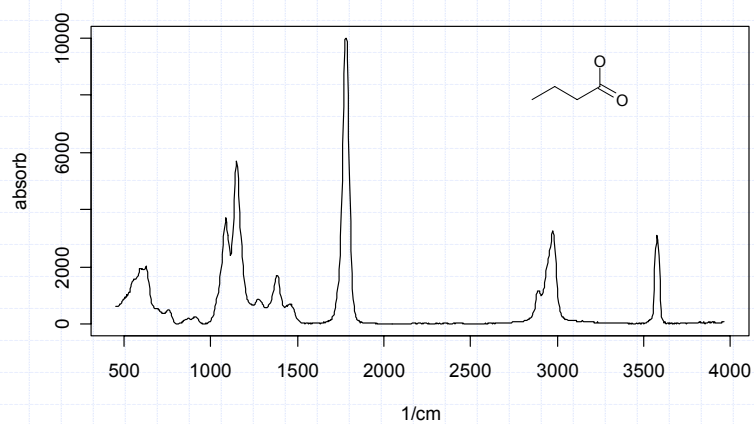
Dissimilarity Metrics

- ◆ Construct pairwise dissimilarities between observations

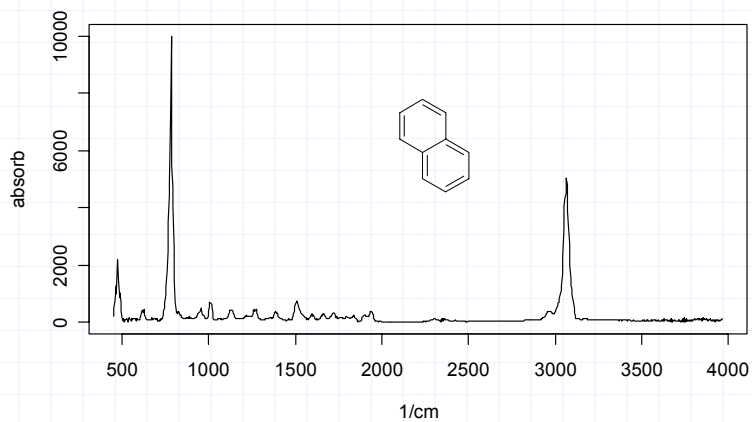
$$D(x_i, x_j) = \sum_{j=1}^p w_j \cdot d(x_{ij}, x_{ij}); \quad \sum_{j=1}^p w_j = 1$$

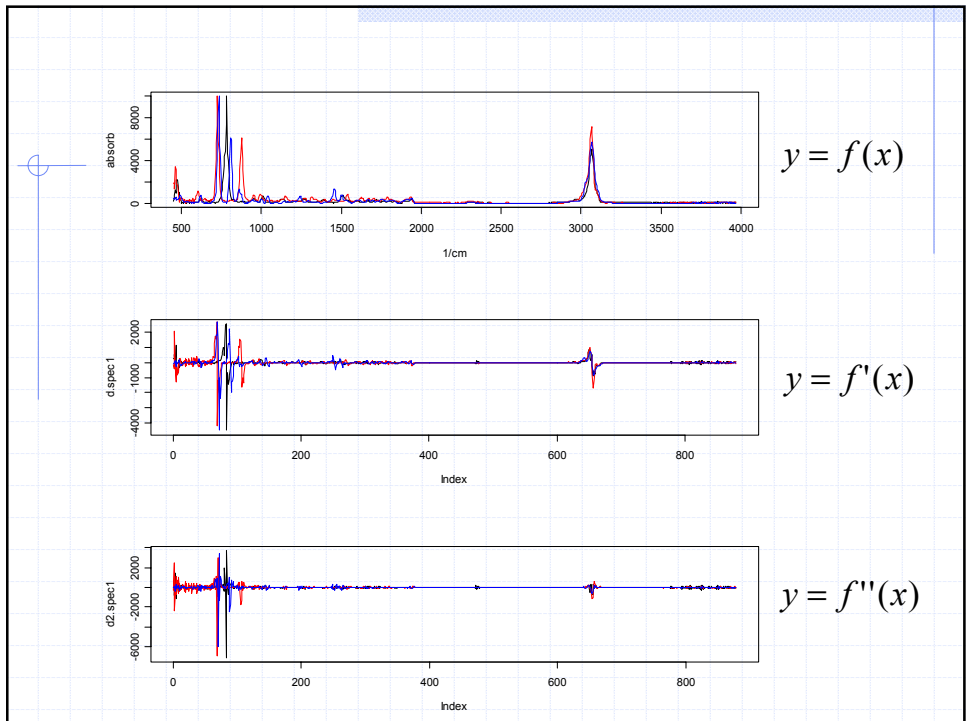
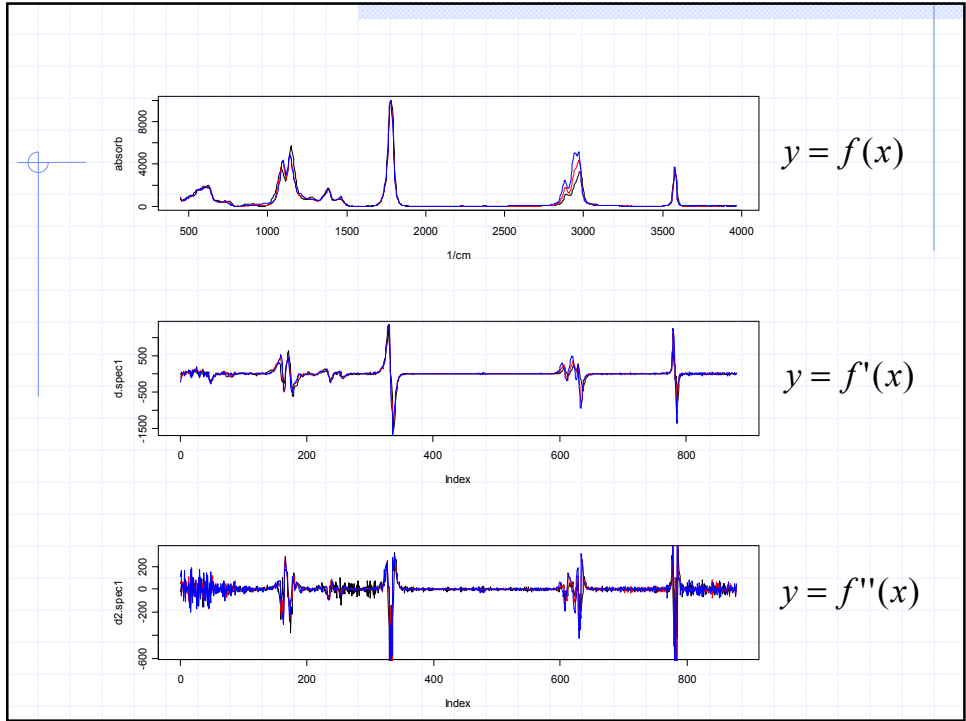
$$d(x_{ij}, x_{i'j}) = (x_{ij} - x_{i'j})^2$$

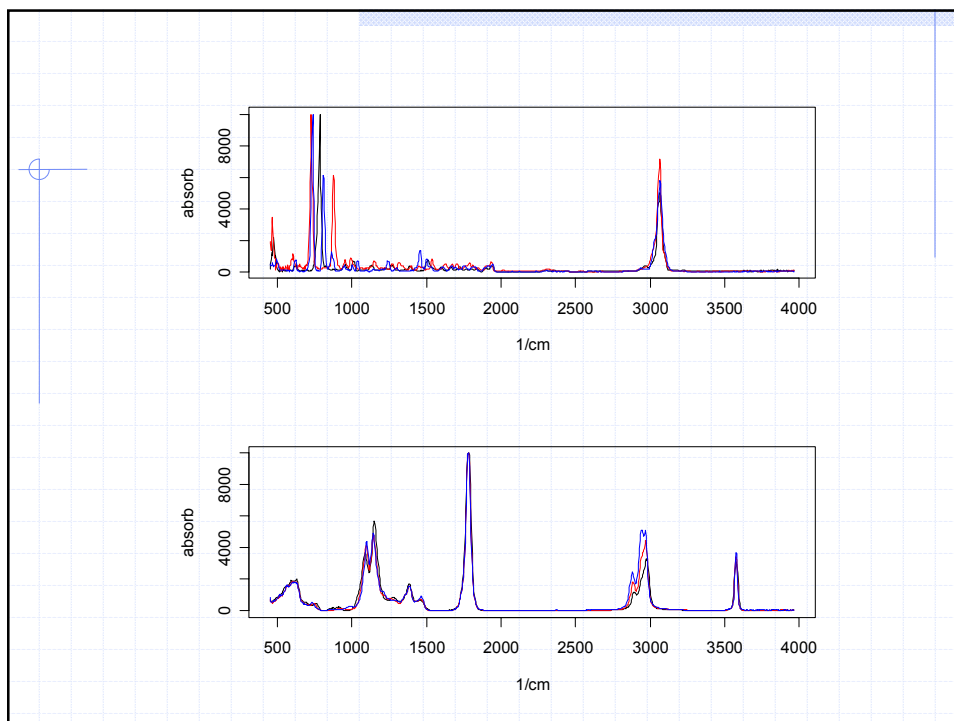
880 Dimensional Data: IR of FA's



Compared to IR from PAH's







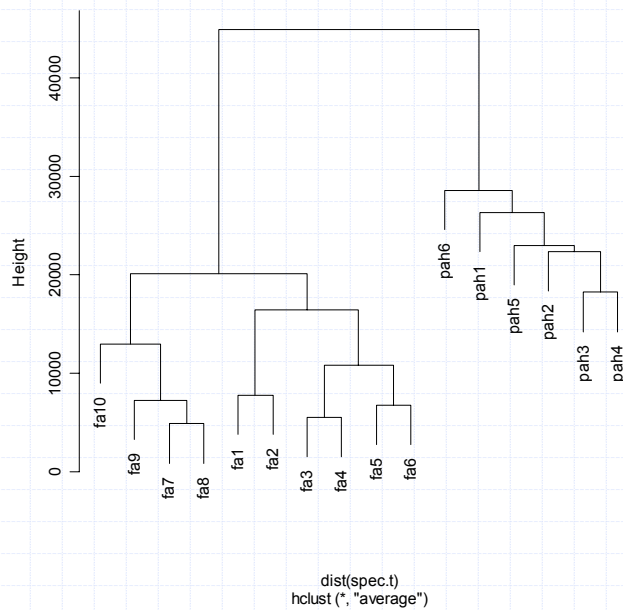
Distance Matrix: dist()

| | pah1 | pah2 | pah3 | pah4 | pah5 | pah6 |
|------|----------|----------|----------|----------|----------|----------|
| pah2 | 27023.15 | | | | | |
| pah3 | 25685.99 | 19710.36 | | | | |
| pah4 | 24860.33 | 24943.17 | 18183.68 | | | |
| pah5 | 27605.84 | 23478.77 | 21199.62 | 24132.58 | | |
| pah6 | 28452.82 | 27547.71 | 27976.13 | 28236.49 | 30536.37 | |
| fa1 | 44264.44 | 45414.67 | 46204.04 | 43519.26 | 48934.11 | 46952.86 |
| fa2 | 43133.35 | 44504.98 | 45023.27 | 42250.91 | 48064.70 | 45941.68 |
| fa3 | 44717.94 | 45981.07 | 46526.37 | 43987.55 | 49542.08 | 47444.59 |
| fa4 | 45310.17 | 46664.01 | 47071.96 | 44683.85 | 50155.72 | 48055.39 |
| fa5 | 45374.92 | 46839.65 | 47174.47 | 44921.76 | 50062.74 | 48185.37 |
| fa6 | 48232.45 | 49787.43 | 49981.15 | 47921.04 | 52808.91 | 50906.00 |
| fa7 | 44544.73 | 46283.15 | 46390.72 | 43860.58 | 49732.28 | 47449.04 |
| fa8 | 41504.64 | 43402.10 | 43402.54 | 40809.35 | 46902.19 | 44555.97 |
| fa9 | 38870.52 | 41309.74 | 40956.81 | 37788.30 | 45150.98 | 42265.29 |
| fa10 | 34546.72 | 37121.11 | 36713.06 | 33609.70 | 40832.36 | 38204.74 |

Distances between FA's

| | fa1 | fa2 | fa3 | fa4 | fa5 | fa6 |
|------|----------|-----------|-----------|-----------|-----------|----------|
| fa2 | 7718.36 | | | | | |
| fa3 | 12711.51 | 6030.209 | | | | |
| fa4 | 17392.06 | 10959.837 | 5506.861 | | | |
| fa5 | 21669.88 | 15835.046 | 10797.357 | 6588.852 | | |
| fa6 | 26259.21 | 20632.722 | 15285.311 | 10337.937 | 6678.778 | |
| fa7 | 26877.12 | 21420.239 | 16708.467 | 12226.365 | 9002.815 | 6965.94 |
| fa8 | 28217.63 | 22991.564 | 18919.696 | 14971.922 | 11467.954 | 10472.53 |
| fa9 | 29410.73 | 24437.826 | 21179.732 | 17623.957 | 14914.461 | 14345.51 |
| fa10 | 33607.62 | 29550.562 | 27283.120 | 24656.561 | 21620.286 | 21855.91 |

Cluster Dendrogram



R code for hclust()

```
library(mva)

p1<-read.table("pah1-ir.txt.out")
...
f1<-read.table("fa1-ir.txt.out")
...
names(p1)<-c("wn", "a")
...
names(f1)<-c("wn", "a")
...
spec<-data.frame(p1$a, p2$a, p3$a, p4$a, p5$a, p6$a,
                 f1$a, f2$a, f3$a, f4$a, f5$a, f6$a, f7$a, f8$a, f9$a, f10$a)

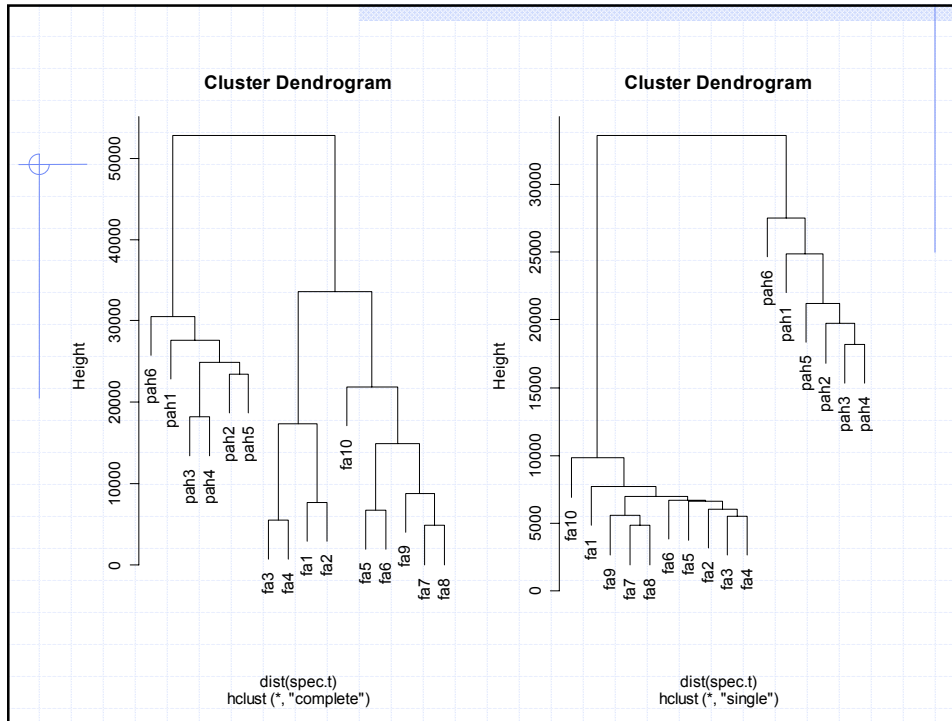
names(spec)<-c("pah1", "pah2", "pah3", "pah4", "pah5", "pah6",
              "fa1", "fa2", "fa3", "fa4", "fa5", "fa6", "fa7", "fa8", "fa9", "fa10")

spec.t <- t(spec)
hc<-hclust(dist(spec.t), "ave")

plot(hc)
```

Agglomerative Methods

- ◆ Single Linkage
 - Takes the intergroup dissimilarity as the closest (least dissimilar) pair
- ◆ Complete Linkage
 - Takes the intergroup dissimilarity as the furthest (most dissimilar) pair
- ◆ Average Linkage
 - Average dissimilarity between groups
 - Compromise between 'Single' and 'Complete'



K-means clustering

- ◆ Iterative descent cluster method
- ◆ Assumes squared-Euclidian distance is used as the dissimilarity measure
- ◆ N observations are assigned to the K clusters in such a way that within each cluster the average dissimilarity of observations from the cluster mean, is minimized.

Cluster assignment: C^*

$$C^* = \min_C \sum_{k=1}^K \sum_{C(i)=k} \|x_i - \bar{x}_k\|^2$$

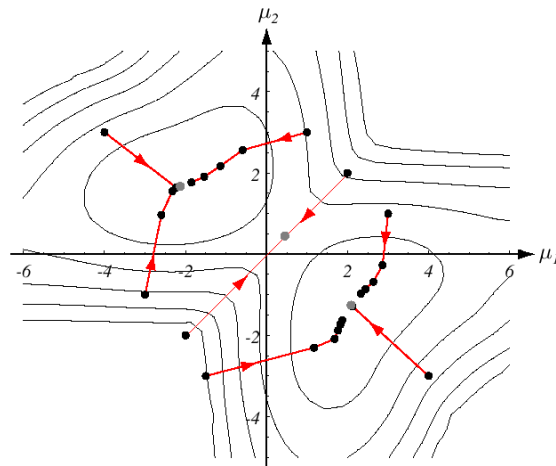
$$\bar{x}_S = \arg \min_m \sum_{i \in S} \|x_i - m\|^2 \quad m = \text{means of current clusters}$$

$$\min_{C, \{m_k\}_1^K} \sum_{k=1}^K \sum_{C(i)=k} \|x_i - m_k\|^2 \quad \text{total cluster variance}$$

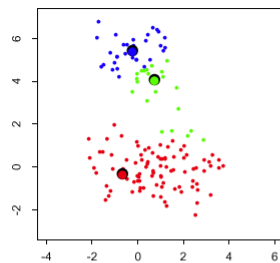
Method

1. For a given cluster assignment C , the total cluster variance is minimized with respect to the means of the currently assigned clusters
2. Given a current set of means, total cluster variance is minimized by assigning each observation to the closest (current) cluster mean.
3. Repeat 1,2 until assignments in 2 don't change.

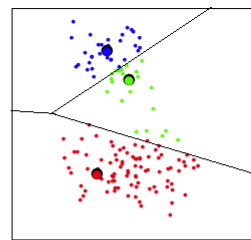
Stochastic Hill-Climbing



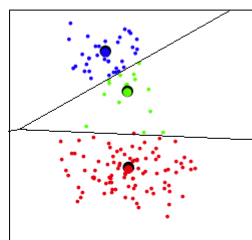
Initial Centroids



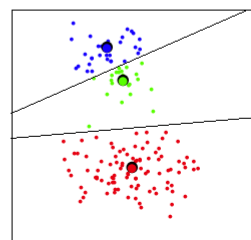
Initial Partition



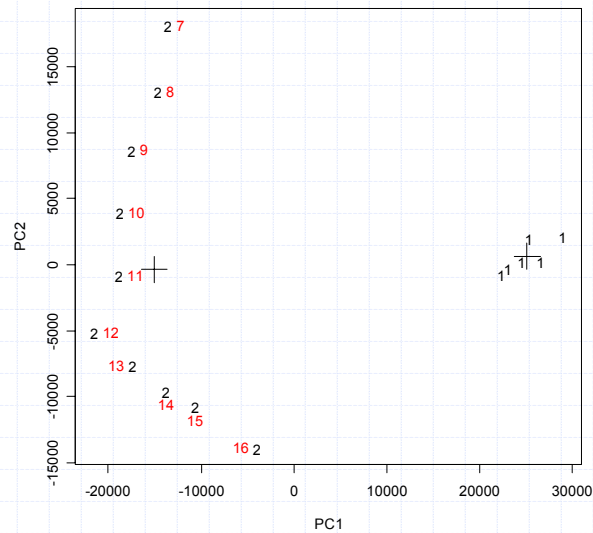
Iteration Number 2



Iteration Number 20



Guess k=2 from dendrogram



R code for PCA/K-means example

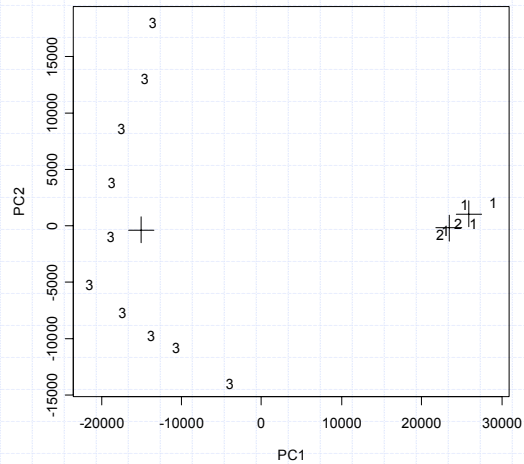
```
library(mva)
library(cluster)
...
spec.t <- t(spec)
hc<-hclust(dist(spec.t),"ave")

initial <- tapply( spec.t,
  list(rep(cutree(hc,2),ncol(spec.t)),col(spec.t)), mean)

dimnames(initial)<-list(NULL,dimnames(spec.t)[[2]])
km<- kmeans(spec.t, initial)

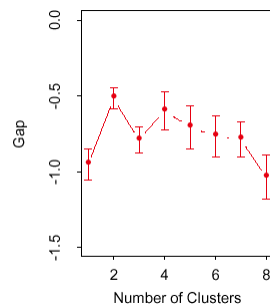
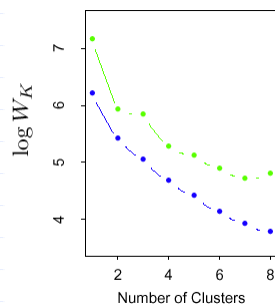
spec.pca<-princomp(spec.t)
spec.px<-predict(spec.pca)
dimnames(km$centers)[[2]]<-dimnames(spec.t)[[2]]
spec.centers<-predict(spec.pca,km$centers)
plot(spec.px[,1:2],type="n", xlab="PC1", ylab="PC2")
text(spec.px[,1:2], cex=1, labels=km$cluster)
points(spec.centers[,1:2], pch=3, cex=3)
```

Wrong Guess at K? cutree(hc,3)



Practical Issues for k-means

- ◆ Must select a number for K
- ◆ Must select an initialization
- ◆ Can use within-cluster dissimilarity or Gap:



Multidimensional Scaling

- ◆ Attempt to find a lower dimensional approximation of the data so as to preserve the pairwise distances as well as possible
- ◆ Classical: Minimize "stress function"

$x_1, x_2, x_3, \dots, x_N \in \mathfrak{R}^p$ Observations in p-dimensions
 $z_1, z_2, z_3, \dots, z_N \in \mathfrak{R}^k$ Transform to k-dimensions

$$S_D(z_1, z_2, z_3, \dots, z_N) = \left[\sum_{i \neq i'} (d_{ii'} - \|z_i - z_{i'}\|)^2 \right]^{1/2}$$

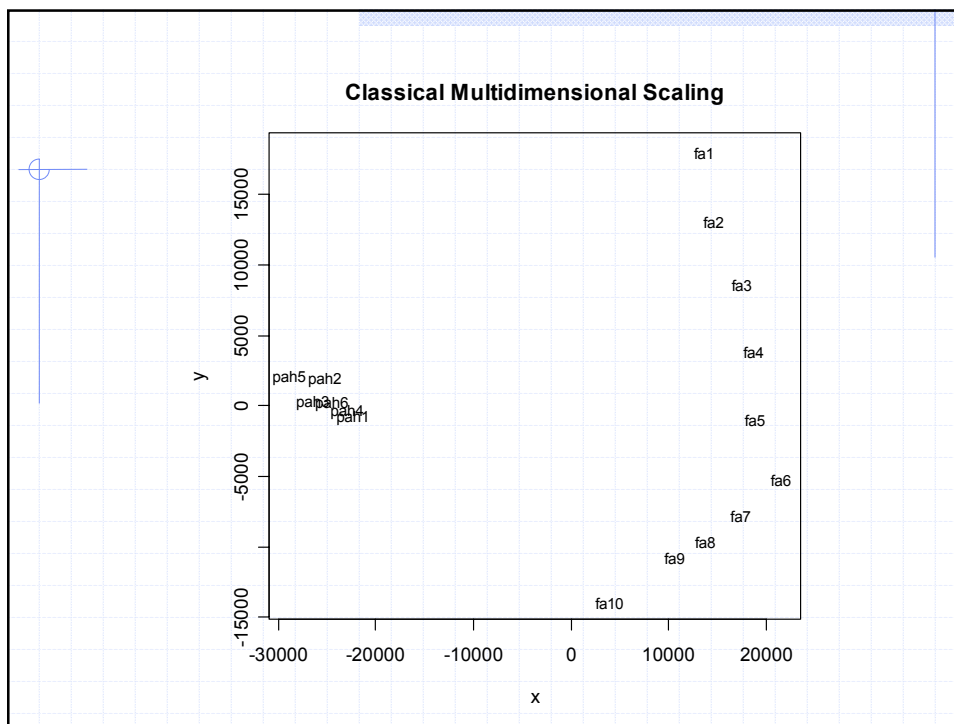
by minimizing S_D

MDS in R

```
library(mva)

...

spec.t <- t(spec)
loc <- cmdscale(dist(spec.t))
x <- loc[,1]
y <- -loc[,2]
plot(x, y, type="n", main="Classical
Multidimensional Scaling")
text(x, y, names(spec), cex=0.8)
```



A look ahead...

- ◆ IR data from NIST is in JCAMP-DX format:

```

##TITLE=Butanoic acid
##JCAMP-DX=4.24
##DATA TYPE=INFRARED SPECTRUM
##ORIGIN=Sadtler Research Labs Under US-EPA Contract
##OWNER=NIST Standard Reference Data Program
##CAS REGISTRY NO=107-92-6
##MOLFORM=C4H8O2
##$NIST SOURCE=MSDC-IR
##STATE=gas
##XUNITS=1/CM
##YUNITS=ABSORBANCE
##XFACTOR=1.0
##YFACTOR=0.000052179
##DELTAX=4.0
##FIRSTX=450.0
##LASTX=3966.0
##FIRSTY=0.033238
##MAXX=3966
##MINX=450
##MAXY=0.52179
##MINY=0
##NPOINTS=880
##XYDATA=(X++(Y..Y))
450.0 637 638 621 624 638 665 702 723 741 798

```

JCAMP Data section

```
##XYDATA=(X++(Y..Y))
450.0 637 638 621 624 638 665 702 723 741 798
490.0 839 862 888 935 928 967 1021 1000 996 1098
530.0 1133 1222 1328 1382 1488 1531 1571 1558 1559
1570
...
##END=
```

Need a x,y pair for use in R...

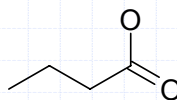
Perl to the rescue...

```
# just a fragment of the code...

$line = <SRCFILE>;
while ($line !~ /##END/) {
    @tokens = split(/ /,$line);
    for( $index = 0; $index <= $#tokens; $index++ ) {
        if( $index == 0 ) {
            $x = $tokens[$index];
        }
        else {
            printf OUTFILE ("%4.1f\t%5.1f\n",
                ($x+((($index-1)*$deltaX)), $tokens[$index]);
        }
    }
    $line = <SRCFILE>;
}
```

Structures: MOL FILES, SMILES

```
-ISIS- 03230321232D  
  
6 5 0 0 0 0 0 0 0 0 0999 V2000  
-3.0000 -1.2167 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
-2.2875 -0.8042 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
-1.5750 -1.2167 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
-0.8625 -0.8042 0.0000 C 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
-0.1475 -1.2158 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
-0.8667 0.0208 0.0000 O 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0  
  
1 2 1 0 0 0 0  
3 4 1 0 0 0 0  
4 5 2 0 0 0 0  
2 3 1 0 0 0 0  
4 6 1 0 0 0 0  
  
M END
```



SMILES: CCCC(=O)O