Cluster Analysis

The Purpose of Clustering

The purpose of cluster analysis is to classify individuals into homogeneous subgroups that have not been established in advance.

Cluster Analysis vs. Factor Analysis

Euclidean Distance

$$d_{ij} = \left[\sum_{k=1}^{q} (x_{ik} - x_{jk})^2 \right]^{\frac{1}{2}}$$

Clustering Strategies

- Hierarchical Clustering
  - Agglomerative hierarchical clustering (**hclust**)
- Non-hierarchical Clustering
  - K-Means Clustering (**kmeans**)

Clustering in R with **hclust**

- Standardize the variables first
  - z-scores
  - scale by dividing by the standard deviation (or range, mean, or maximum)
- Dendrogram
  - plot the cluster dendrogram
  - choose a number of clusters or choose a height at which to cut

Clustering in R with **kmeans**

- Standardize the variables first
  - z-scores
  - scale by dividing by the standard deviation (or range, mean, or maximum)
- Specify the number of clusters
  - plot the within-groups sum of squares against the number of clusters and look for the “knee” in the plot
- Outliers?

Using the Cluster Assignments

- Append the data set with a new variable that identifies the cluster assignment for each individual
- Examine the cluster assignments
  - how many individual are in each cluster?
  - what are the means for the other variables within each cluster?
  - do the groupings seem homogeneous in meaningful ways?
- Use cluster assignments in other analyses as needed (ANOVA & MANOVA, for example)

Criteria for Good Clustering

- It detects structures present in the data
- It enables the optimal number of clusters to be determined
- Clusters are clearly differentiable
- Clusters remain stable when there are small changes in the data
- It processes large data efficiently
- It handles both quantitative and categorical data (more advanced techniques are required with categorical variables)

Example: Olympic Figure Skating Judges

The International Olympic Committee (IOC), responding to media criticism, wants to test whether scores given by judges trained through the IOC program are “reliable”; that is, while the precise scores given by two judges may differ, good performances receive higher scores than average performances, and average performances receive higher scores than poor performances.

Consider that the IOC has asked eight trained judges to score 300 performances.

We will use this data to see if scores for the skating performances have an underlying structure that groups the performances into homogeneous clusters.

Example: Olympic Figure Skating Judges

```r
head(judges)
```

```
##   judge1 judge2 judge3 judge4 judge5 judge6 judge7 judge8
## 1    7.1    7.2    7.0    7.7    7.1    7.1    7.0    7.3
## 2    9.3    9.7    8.9    9.6    8.6    9.5    9.6    9.7
## 3    8.9    8.8    8.1    9.3    8.5    8.1    7.6    8.7
## 4    8.0    8.1    7.3    8.7    7.5    8.7    7.4    9.5
## 5    9.1    9.0    8.2    9.0    8.2    9.5    7.8    8.0
## 6    9.1    9.2    8.3    9.1    7.9    8.9    9.0    9.2
```

Standardizing the variables

```r
judges.z <- scale(judges)
round(head(judges.z),2)
```

```
##      judge1 judge2 judge3 judge4 judge5 judge6 judge7 judge8
## [1,]  -1.58  -1.96  -1.31  -1.76  -1.36  -1.74  -1.16  -1.20
## [2,]   0.93   0.93   0.94   0.91   0.81   0.67   1.46   1.19
## [3,]   0.47  -0.11  -0.01   0.48   0.67  -0.74  -0.56   0.19
## [4,]  -0.55  -0.92  -0.95  -0.36  -0.78  -0.14  -0.76   0.99
## [5,]   0.70   0.12   0.11   0.06   0.23   0.67  -0.36  -0.50
## [6,]   0.70   0.35   0.23   0.20  -0.20   0.06   0.85   0.69
```

Scaling the variables without centering

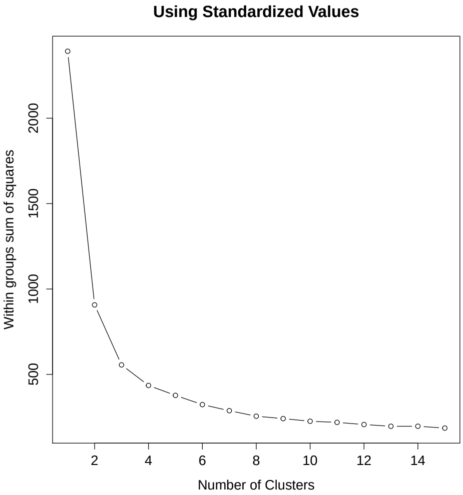
```
judges.s <- scale(judges, center = FALSE,
                 scale = apply(judges, 2, sd, na.rm = TRUE))
round(head(judges.s),2)
```

```
##      judge1 judge2 judge3 judge4 judge5 judge6 judge7 judge8
## [1,]   8.08   8.32   8.28  10.81  10.25   7.13   7.04   7.24
## [2,]  10.58  11.21  10.53  13.48  12.42   9.53   9.66   9.62
## [3,]  10.13  10.17   9.58  13.06  12.28   8.13   7.65   8.63
## [4,]   9.10   9.36   8.63  12.22  10.83   8.73   7.44   9.42
## [5,]  10.36  10.40   9.70  12.64  11.84   9.53   7.85   7.94
## [6,]  10.36  10.63   9.82  12.78  11.41   8.93   9.05   9.13
```

K-means Clustering: Selecting the Number of Groups

```
wss <- (nrow(judges.z)-1)*sum(apply(judges.z,2,var))
for (i in 2:15) wss[i] <- sum(kmeans(judges.z,
                                     centers=i)$withinss)
plot(1:15, wss, type="b", xlab="Number of Clusters",
     ylab="Within groups sum of squares",
     main="Using Standardized Values")
```

K-means Clustering: Selecting the Number of Groups



K-means Clustering

```r
kmeans(judges.z,centers=3,nstart=25)
```

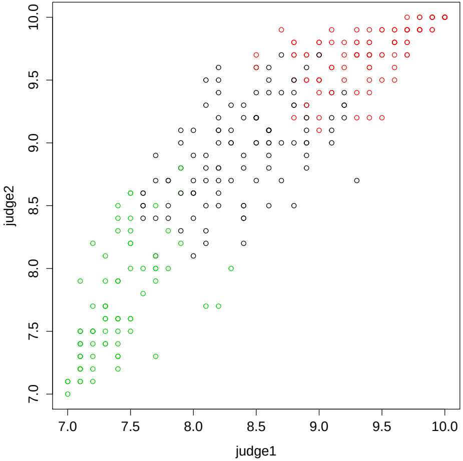
```
## K-means clustering with 3 clusters of sizes 80, 106, 114
##
## Cluster means:
##       judge1      judge2     judge3      judge4     judge5      judge6
## 1 -1.2497998 -1.36834986 -1.1028627 -1.33813779 -1.1353215 -1.39784195
## 2  1.0566879  0.96477517  1.1226637  0.96497523  1.0912848  0.96105924
## 3 -0.1054819  0.06317387 -0.2699415  0.04178639 -0.2179866  0.08732523
```

K-means Clustering: Handling the Output

```r
fit <- kmeans(judges.z,3,nstart=25)
round(aggregate(judges,by=list(fit$cluster),FUN=mean),2)
```

```
##   Group.1 judge1 judge2 judge3 judge4 judge5 judge6 judge7 judge8
## 1       1   7.39   7.71   7.17   8.00   7.25   7.44   7.10   7.64
## 2       2   9.41   9.73   9.06   9.64   8.79   9.79   9.30   9.31
## 3       3   8.39   8.95   7.88   8.99   7.89   8.92   7.83   8.36
```

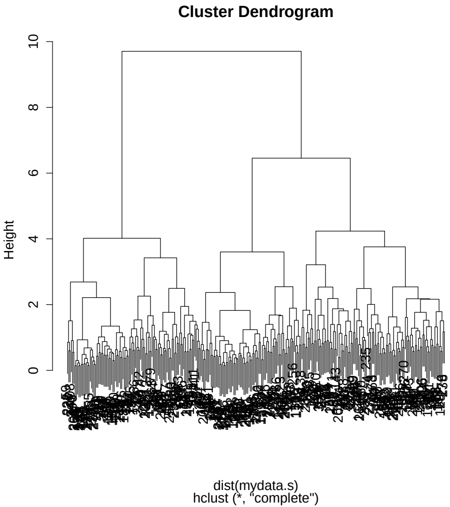
Creating plots that show clustering for two variables



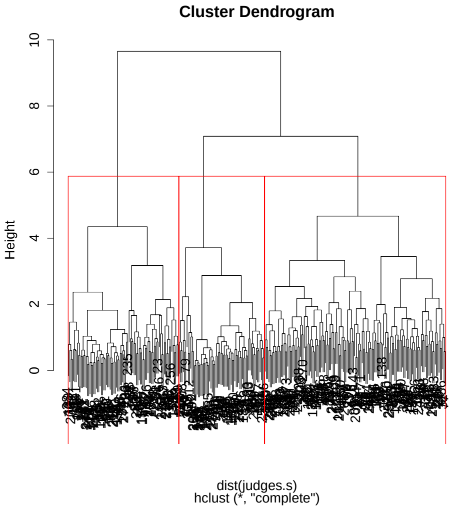
Agglomerative Hierarchical Clustering

```r
output <- hclust(dist(judges.s))
plot(output)
```

The Dendrogram



The Dendrogram



Appending the original data frame with cluster identifier, looking at means and cluster sizes

```
output<-hclust(dist(judges.s))
judges$clusternumber <- cutree(output, h=5)
round(aggregate(judges[,1:8],by=list(judges$clusternumber),FUN=mean),2)
```

```
##   Group.1 judge1 judge2 judge3 judge4 judge5 judge6 judge7 judge8
## 1       1   7.44   7.79   7.20   8.06   7.29   7.51   7.12   7.81
## 2       2   9.62   9.83   9.30   9.74   9.01   9.93   9.57   9.49
## 3       3   8.59   9.13   8.09   9.13   8.04   9.13   8.11   8.47
```

```
summary(as.factor(judges$clusternumber))
```

```
##   1   2   3
##  88  68 144
```