

## Cluster Analysis

Today . . .

- Goals of cluster analysis
- Basic methods
- Ability to recover a known structure

Cluster analysis is an exploratory procedure that is used to identify groups of similar objects (e.g., people, stimuli, books, singers, etc.) in a large collection of objects. The identified groups have members that are similar to each other and different from the members in other groups.

The approach is similar in spirit to MDS, but produces discrete groups without any spatial representation.

The identified groups can be used in subsequent analyses.

The approach is decidedly “discovery” oriented—there is usually no prior knowledge of how or why objects might be distributed into particular groups.

There is an assumption that some natural grouping might exist and the goal of cluster analysis is to help find it.

---

---

---

---

---

---

---

---

The most common clustering methods are ***hierarchical*** and ***agglomerative***—forming clusters by joining nearby objects or clusters, beginning with as many clusters as there are objects and ending with a single cluster.

In between there MAY be a number of clusters that provides a convenient simplification of the data.

---

---

---

---

---

---

---

---

The clustering solution depends on:

- Method of joining nearby objects
- Type of distance or similarity measure used (e.g., Euclidean distance, squared Euclidean distance, Minkowski metric, correlation, binary matches)
- The information contained in the distance or similarity measure (e.g., absolute distance, profile similarity)
- Nature of the data (standardized or unstandardized, by case or by variable)

---

---

---

---

---

---

---

---

There are a sizeable number of ways to define clusters, reflecting the different desirable properties that the clusters might have (e.g., small distances within clusters, large distances between clusters). Five methods in particular are fairly common:

- Single linkage (nearest neighbor)
- Complete linkage (farthest neighbor)
- Average linkage (between-groups)
- Centroid method
- Ward's method

---

---

---

---

---

---

---

All of the clustering procedures share these steps:

1. Begin with N clusters each containing one case, numbered 1 through N.
2. Find the most similar pair of clusters  $p$  and  $q$ .
3. Reduce the number of clusters by one through merger of clusters  $p$  and  $q$ . Update the similarity matrix (by the method specified) to reflect revised similarities or dissimilarities between the new cluster and all other clusters. Delete the rows and columns of the elements joined in the new cluster.
4. Perform the previous two steps until all entities are in one cluster.

---

---

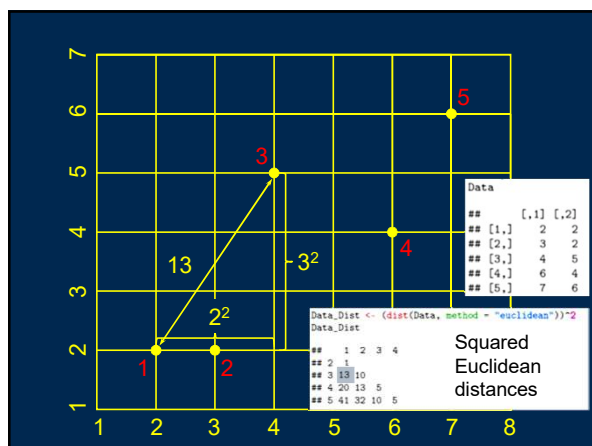
---

---

---

---

---




---

---

---

---

---

---

---

### Single Linkage

The dissimilarity between two clusters (A and B) is the minimum of all possible distances between the cases in Cluster A and the cases in Cluster B.

```
hc_1 <- hclust(Data_Dist, method = "single")
```

---

---

---

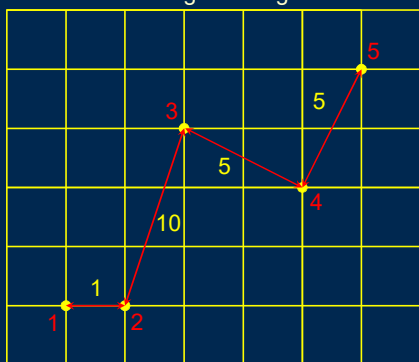
---

---

---

---

### Single Linkage




---

---

---

---

---

---

---

```
hc_1 <- hclust(Data_Dist, method = "single")
hc_1$merge
```

```
##      [,1] [,2]
## [1,]  -1  -2  Objects 1 and 2 are joined
## [2,]  -3  -4  Objects 3 and 4 are joined
## [3,]  -5   2  Object 5 and Cluster from Step 2 are joined
## [4,]   1   3  Clusters from Step 1 and 3 are joined
```

---

---

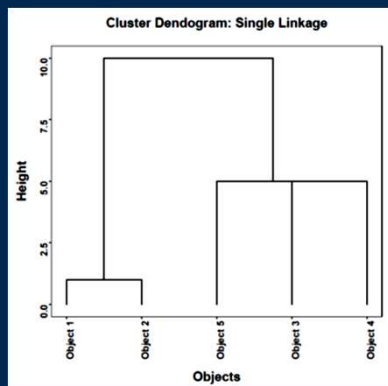
---

---

---

---

---



Height is the squared Euclidean distance using the single linkage criterion.

---

---

---

---

---

---

---

---

### Complete Linkage

The dissimilarity between two clusters (A and B) is the maximum of all possible distances between the cases in Cluster A and the cases in Cluster B.

The minimum of these maximums identifies the clusters to be joined.

```
hc_2 <- hclust(Data_Dist, method = "complete")
```

---

---

---

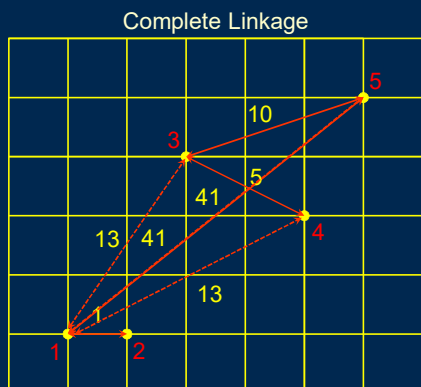
---

---

---

---

---




---

---

---

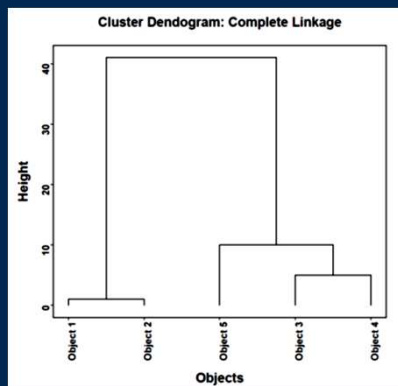
---

---

---

---

---



Height is the squared Euclidean distance using the complete linkage criterion.

---

---

---

---

---

---

---

---

### Average Linkage

The dissimilarity between two clusters (A and B) is the average of all possible distances between the cases in Cluster A and the cases in Cluster B.

The minimum of these average distances determines the clusters to be joined.

```
hc_3 <- hclust(Data_Dist, method = "average")
```

---

---

---

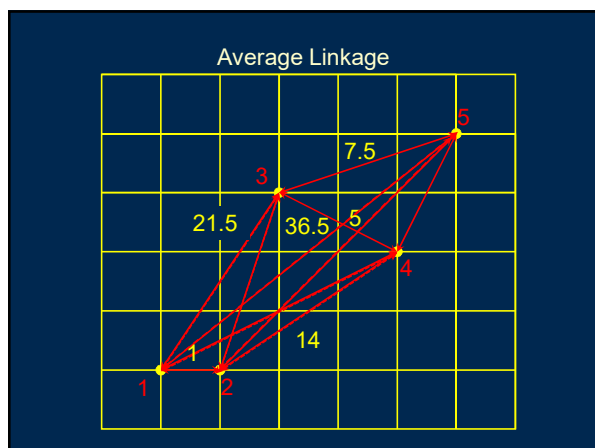
---

---

---

---

---




---

---

---

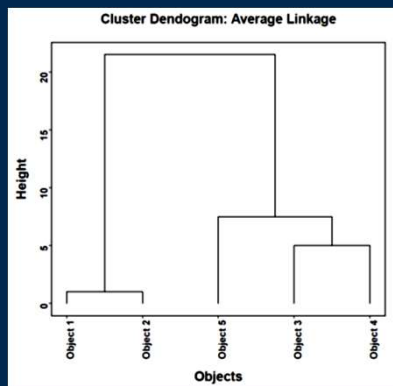
---

---

---

---

---



Height is the squared Euclidean distance using the average linkage criterion.

---

---

---

---

---

---

---

---

### Centroid Method

The dissimilarity between two clusters (A and B) is the distance between the centroid for the cases in Cluster A and the centroid for the cases in Cluster B.

The minimum of these centroid distances determines the clusters to be joined.

```
hc_4 <- hclust(Data_Dist, method = "centroid")
```

---

---

---

---

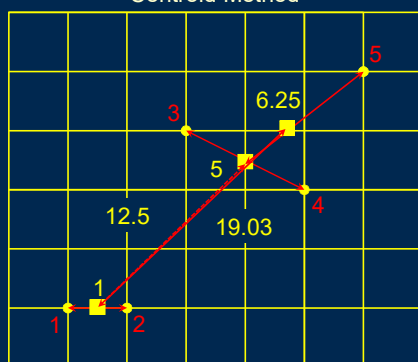
---

---

---

---

### Centroid Method




---

---

---

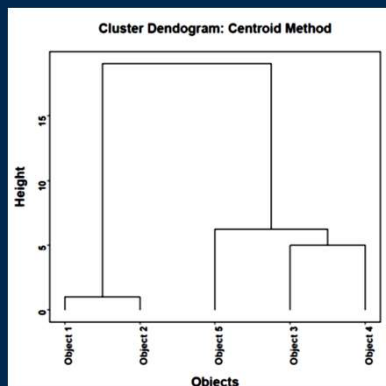
---

---

---

---

---



Height is the squared Euclidean distance using the centroid method criterion.

---

---

---

---

---

---

---

---

### Ward's Method

The dissimilarity between two clusters (A and B) is the *loss of information* from joining the clusters, measured by the increase in error sum of squares.

The sum of squares for a cluster is the sum of squared deviations of each case from the centroid for the cluster. The error sum of squares is the total of these for all clusters. The two clusters among all possible combinations that have the minimum increase in error sum of squares are joined.

```
hc_5 <- hclust(Data_Dist, method = "ward.D")
```

---

---

---

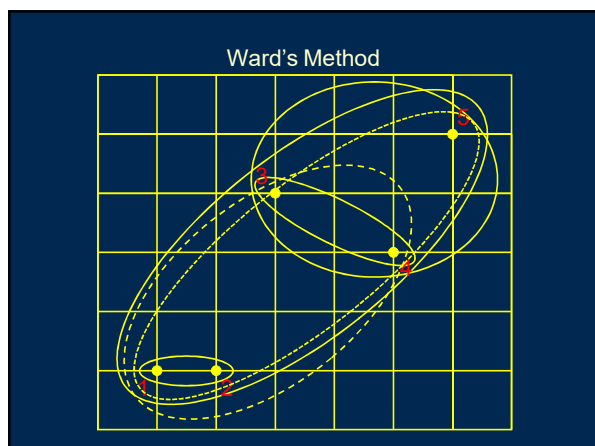
---

---

---

---

---




---

---

---

---

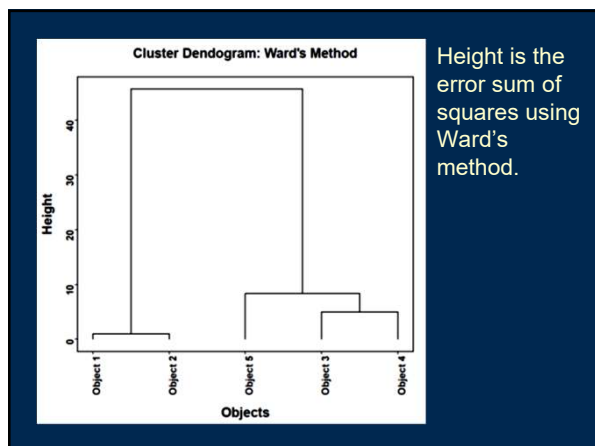
---

---

---

---






---

---

---

---

---


---

---

---

An important question is how well the different clustering methods can recover a group structure when it is known in advance. That can lend insight into the ability of the methods to identify any group structure when that structure is not known in advance.

A classic test data set was introduced by R. A. Fisher (1936): three species of iris, varying in their petal length, petal width, sepal length, and sepal width.



R. A. Fisher (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2), 173-186.

---

---

---

---

---

---

---

---




---

---

---

---

---

---

---

---

```
describeBy(Iris[, c(1:4)], group = Iris$Species, digits = 2)
```

```
## group: Setosa
##      vars  n mean  sd median trimmed mad min max
## Sepal.Length 1 50 50.06 3.52 50 50.02 2.97 43 58
## Sepal.Width 2 50 34.28 3.79 34 34.15 3.71 23 44
## Petal.Length 3 50 14.62 1.74 15 14.60 1.48 10 19
## Petal.Width 4 50 2.46 1.05 2 2.38 0.00 1 6
```



```
## group: Versicolor
##      vars  n mean  sd median trimmed mad min max
## Sepal.Length 1 50 59.36 5.16 59.0 59.38 5.19 49 70
## Sepal.Width 2 50 27.70 3.14 28.0 27.80 2.97 20 34
## Petal.Length 3 50 42.60 4.70 43.5 42.92 5.19 30 51
## Petal.Width 4 50 13.26 1.98 13.0 13.25 2.22 10 18
```



```
## group: Virginica
##      vars  n mean  sd median trimmed mad min max
## Sepal.Length 1 50 65.88 6.36 65.0 65.72 5.93 49 79
## Sepal.Width 2 50 29.74 3.22 30.0 29.62 2.97 22 38
## Petal.Length 3 50 55.52 5.52 55.5 55.10 6.67 45 69
## Petal.Width 4 50 20.26 2.75 20.0 20.32 2.97 14 25
```



```
anova(aov(Iris$Sepal.Length ~ as.factor(Species), data = Iris))
## Analysis of Variance Table
##
## Response: Iris$Sepal.Length
##      Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Species) 2 6321 3161 119 <2e-16 ***
## Residuals 147 3896 27
```

```
anova(aov(Iris$Sepal.Width ~ as.factor(Species), data = Iris))
## Analysis of Variance Table
##
## Response: Iris$Sepal.Width
##      Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Species) 2 1134 567 49.2 <2e-16 ***
## Residuals 147 1696 12
```

```
anova(aov(Iris$Petal.Length ~ as.factor(Species), data = Iris))
## Analysis of Variance Table
##
## Response: Iris$Petal.Length
##      Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Species) 2 43710 21855 1180 <2e-16 ***
## Residuals 147 2722 19
```

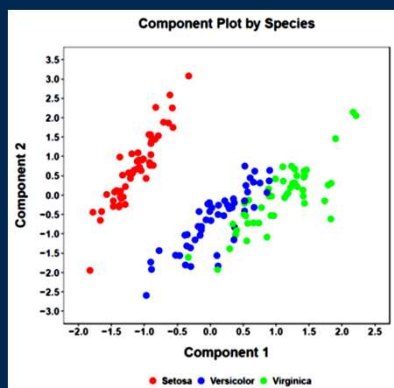
```
anova(aov(Iris$Petal.Width ~ as.factor(Species), data = Iris))
## Analysis of Variance Table
##
## Response: Iris$Petal.Width
##      Df Sum Sq Mean Sq F value Pr(>F)
## as.factor(Species) 2 8041 4021 960 <2e-16 ***
## Residuals 147 616 4
```

The three species are quite different for each feature.

```
PCA <- principal(Iris[, 1:4], nfactors = 2, rotate = "varimax", scores = TRUE)
PCA
```

```
## Principal Components Analysis
## Call: principal(r = Iris[, 1:4], nfactors = 2, rotate = "varimax",
## scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PC1 PC2 h2 u2 com
## Sepal.Length 0.96 0.05 0.92 0.0774 1.0
## Sepal.Width -0.14 0.98 0.99 0.0091 1.0
## Petal.Length 0.94 -0.30 0.98 0.0163 1.2
## Petal.Width 0.93 -0.26 0.94 0.0647 1.2
##
##      PC1 PC2
## SS loadings 2.70 1.13
## Proportion Var 0.68 0.28
## Cumulative Var 0.68 0.96
## Proportion Explained 0.71 0.29
## Cumulative Proportion 0.71 1.00
```

The features can be reduced to two dimensions that capture nearly all of the variance. We can use this reference system to get a initial glimpse of clustering.



We can expect some difficulty in separating *versicolor* from *virginica*.

---

---

---

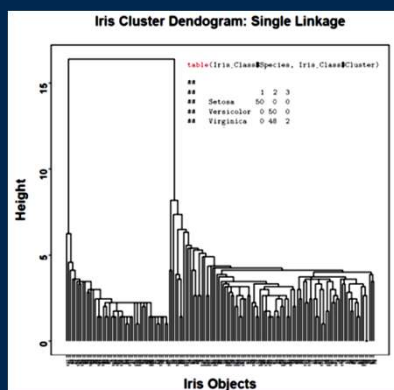
---

---

---

---

---



Single linkage is unable to separate *versicolor* and *virginica*.

---

---

---

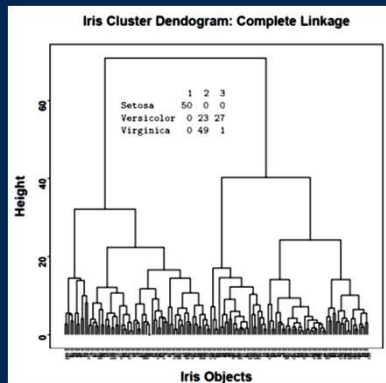
---

---

---

---

---



Complete linkage is better able to separate *versicolor* and *virginica*.

---

---

---

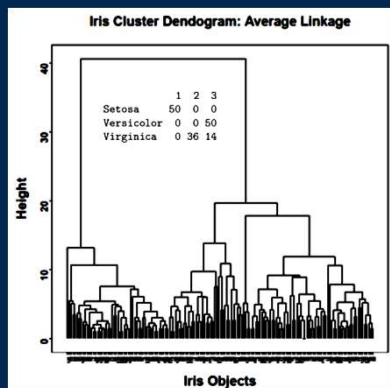
---

---

---

---

---



Average linkage is even better able to separate *versicolor* and *virginica*.

---

---

---

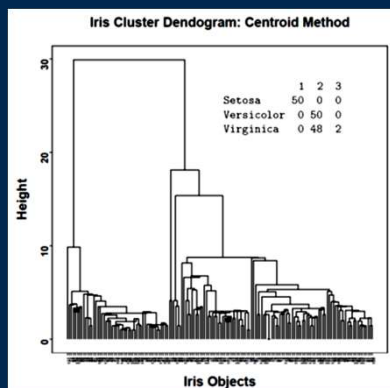
---

---

---

---

---



The centroid method is no better at separating *versicolor* and *virginica* than the single linkage method.

---

---

---

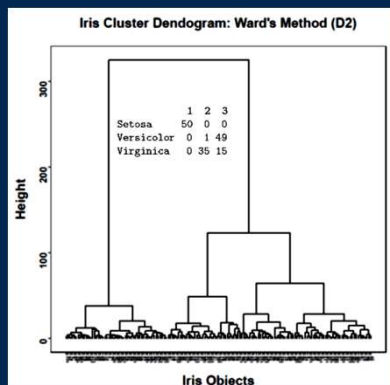
---

---

---

---

---



Ward's method separates *versicolor* and *virginica* about as well as the average linkage method.

---

---

---

---

---

---

---

---

## Additional issues . . .

- Variables can be continuous, binary, or counts. If the variables differ in their scaling, they should be standardized. The most common binary distance measure is the Jaccard distance.

Object 2

	0	1
Object 1	0	1
	d	b
	1	c
		a

$$Jaccard = 1 - \frac{a}{(a + b + c)}$$

---

---

---

---

---

---

---

---

## Additional issues . . .

- If tied distances or similarities exist in the input data or occur among updated clusters, the resulting solution may depend on the order of cases in the file. Test with different random orderings.
- The distance or similarity measures used should be appropriate for the data analyzed.

---

---

---

---

---

---

---

---

## For interval level data:

- Euclidean distance. The square root of the sum of the squared differences between values for the items. This is the default for interval data.
- Squared Euclidean distance.
- Pearson correlation. The product-moment correlation between two vectors of values.
- Cosine. The cosine of the angle between two vectors of values.

---

---

---

---

---

---

---

---

For interval level data:

- Chebychev. The maximum absolute difference between the values for the items.
- Block. The sum of the absolute differences between the values of the item. Also known as Manhattan distance.
- Minkowski. The  $p^{\text{th}}$  root of the sum of the absolute differences to the  $p^{\text{th}}$  power between the values for the items.

---

---

---

---

---

---

---

Additional issues . . .

- All relevant variables should be included in the analysis. Omission of influential variables can result in a misleading solution.
- Have I said this before? Because cluster analysis is a highly exploratory method, results should be treated as (really, really) tentative until they are confirmed with an independent sample. Unlike the example, we usually do not know the actual underlying structure to the data.

---

---

---

---

---

---

---

Next time . . .

K-Means clustering

---

---

---

---

---

---

---