Data Science and Statistics in Research: unlocking the power of your data

Session 3.4:

Clustering

Outline

Overview

Clustering

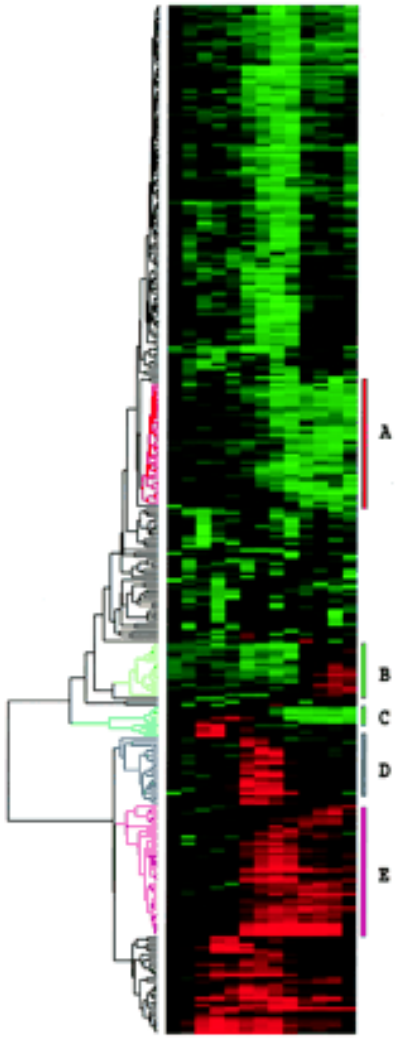
- Clustering is a statistical technique which creates groupings within data.
- Objects within the same cluster are more similar to each other than they are to objects in a different cluster.

Example: Customer Preferences

- A retailer wishes to provide each user with a unique set of recommendations.
- If we can identify similarities between customers based on shopping history, we could group customers into K groups.
- Within each group customers have similar purchasing patterns but differences in the purchases could form the basis of a recommendation system.
- Items could also be grouped based on the customers they were bought by. If a group have bought the same set of items, these items could be considered similar - thus it may be sensible to recommend similar items.

Example: Gene Function Prediction

- Much research in molecular biology is focussed on categorising what function a particular gene serves.
- A useful source of information is from microarray data which yields numerical values of how active a particular gene is under given circumstances. For a set of genes this can be measured over time.
- Genes can be clustered based on this data so that the genes in each cluster show similar behaviour over time.



CLUSTERING IN R

- One type of clustering method is known as **hierarchical clustering**. This available through the `hclust` function in `R`.
- Another method of cluster analysis is known as **k-means cluster analysis**, and is available in `R` through the `kmeans` function.

Hierarchical Clustering

Hierarchical Clustering

- Hierarchical clustering starts out by putting each observation into its own separate cluster.
- It examines all the distances between all the observations and pairs together the two closest ones to form a new cluster.
- This process repeats until there is one single cluster.

A Simple Example

{{1},{2},{3},{4},{5},{6},{7},{8}}

{{1,2},{3},{4},{5},{6},{7},{8}}

{{1,2},{3},{4},{5,6},{7},{8}}

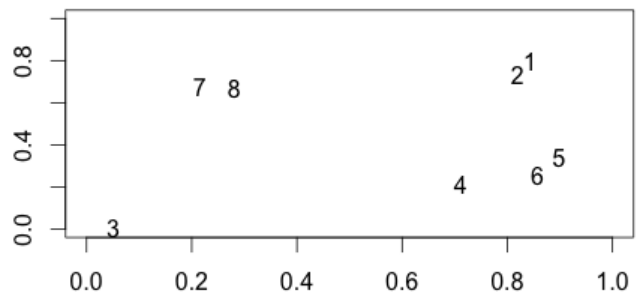
{{1,2},{3},{4},{5,6},{7,8}}

{{1,2},{3},{4,5,6},{7,8}}

{{1,2},{4,5,6},{3},{7,8}}

{{1,2},{4,5,6},{3},{7,8}}

{{1,2,4,5,6,3,7,8}}



A Simple Example

{{1},{2},{3},{4},{5},{6},{7},{8}}

{{1,2},{3},{4},{5},{6},{7},{8}}

{{1,2},{3},{4},{5,6},{7},{8}}

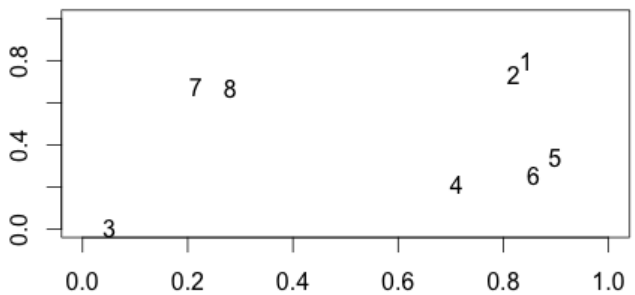
{{1,2},{3},{4},{5,6},{7,8}}

{{1,2},{3},{4,5,6},{7,8}}

{{1,2},{4,5,6},{3},{7,8}}

{{1,2},{4,5,6},{3},{7,8}}

{{1,2,4,5,6,3,7,8}}



A Simple Example

{{1},{2},{3},{4},{5},{6},{7},{8}}

{{1,2},{3},{4},{5},{6},{7},{8}}

{{1,2},{3},{4},{5,6},{7},{8}}

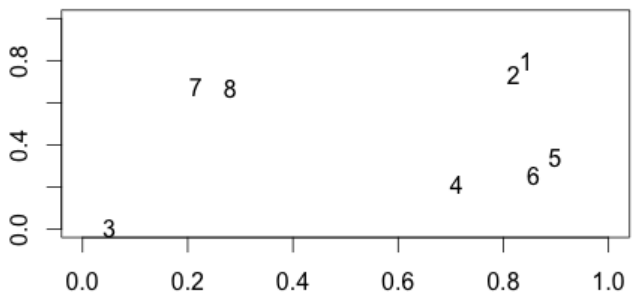
{{1,2},{3},{4},{5,6},{7,8}}

{{1,2},{3},{4,5,6},{7,8}}

{{1,2},{4,5,6},{3},{7,8}}

{{1,2},{4,5,6},{3},{7,8}}

{{1,2,4,5,6,3,7,8}}



A Simple Example

$\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7\},\{8\}\}$

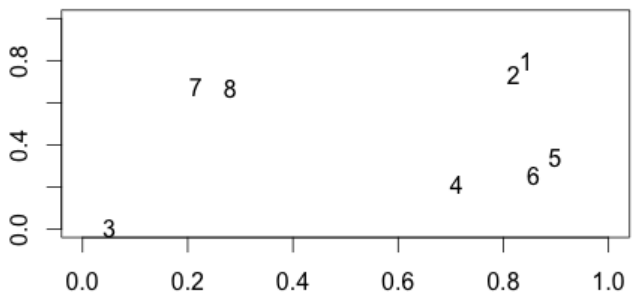
$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7,8\}\}$

$\{\{1,2\},\{3\},\{4,5,6\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2,4,5,6,3,7,8\}\}$



A Simple Example

$\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7\},\{8\}\}$

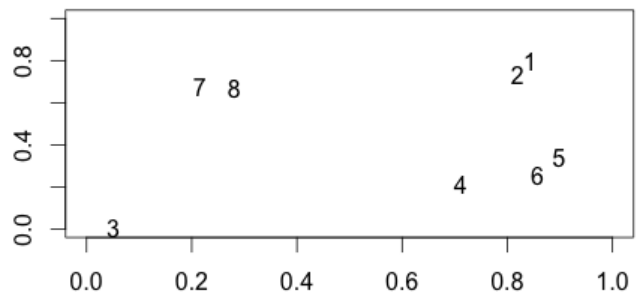
$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7,8\}\}$

$\{\{1,2\},\{3\},\{4,5,6\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2,4,5,6,3,7,8\}\}$



A Simple Example

$\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7\},\{8\}\}$

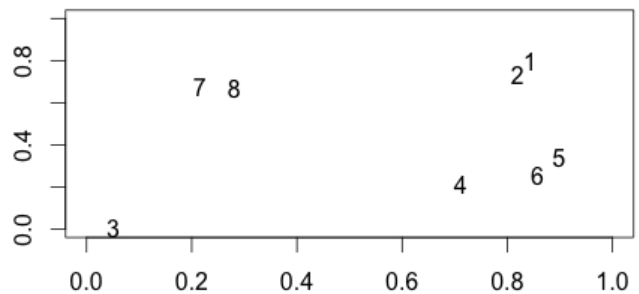
$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7,8\}\}$

$\{\{1,2\},\{3\},\{4,5,6\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2,4,5,6,3,7,8\}\}$



A Simple Example

$\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7\},\{8\}\}$

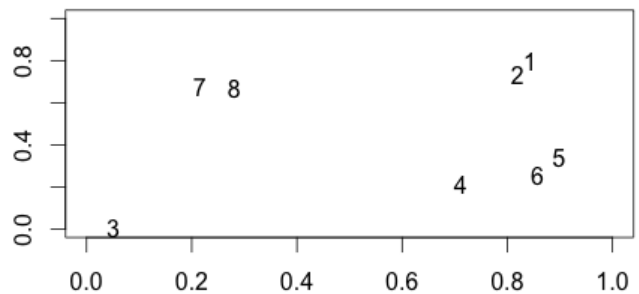
$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7,8\}\}$

$\{\{1,2\},\{3\},\{4,5,6\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2,4,5,6,3,7,8\}\}$



A Simple Example

$\{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5\},\{6\},\{7\},\{8\}\}$

$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7\},\{8\}\}$

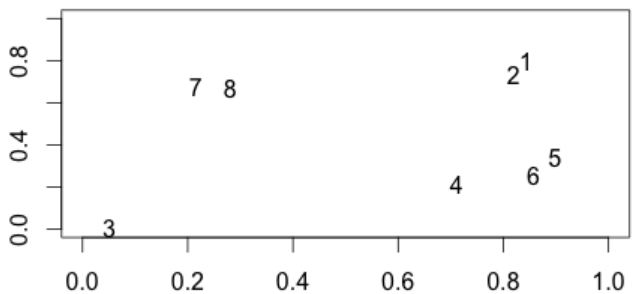
$\{\{1,2\},\{3\},\{4\},\{5,6\},\{7,8\}\}$

$\{\{1,2\},\{3\},\{4,5,6\},\{7,8\}\}$

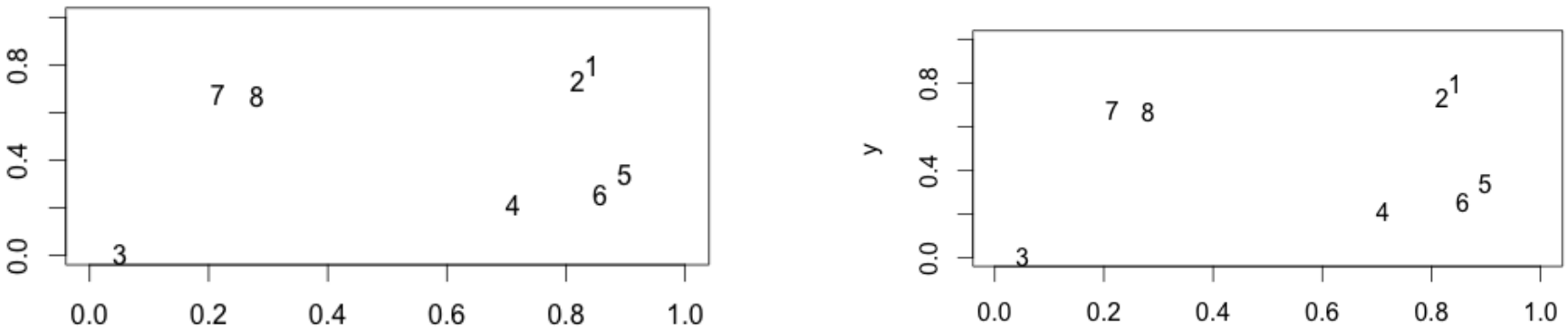
$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2\},\{4,5,6\},\{3\},\{7,8\}\}$

$\{\{1,2,4,5,6,3,7,8\}\}$



Visualising a hierarchical clustering: The Dendrogram

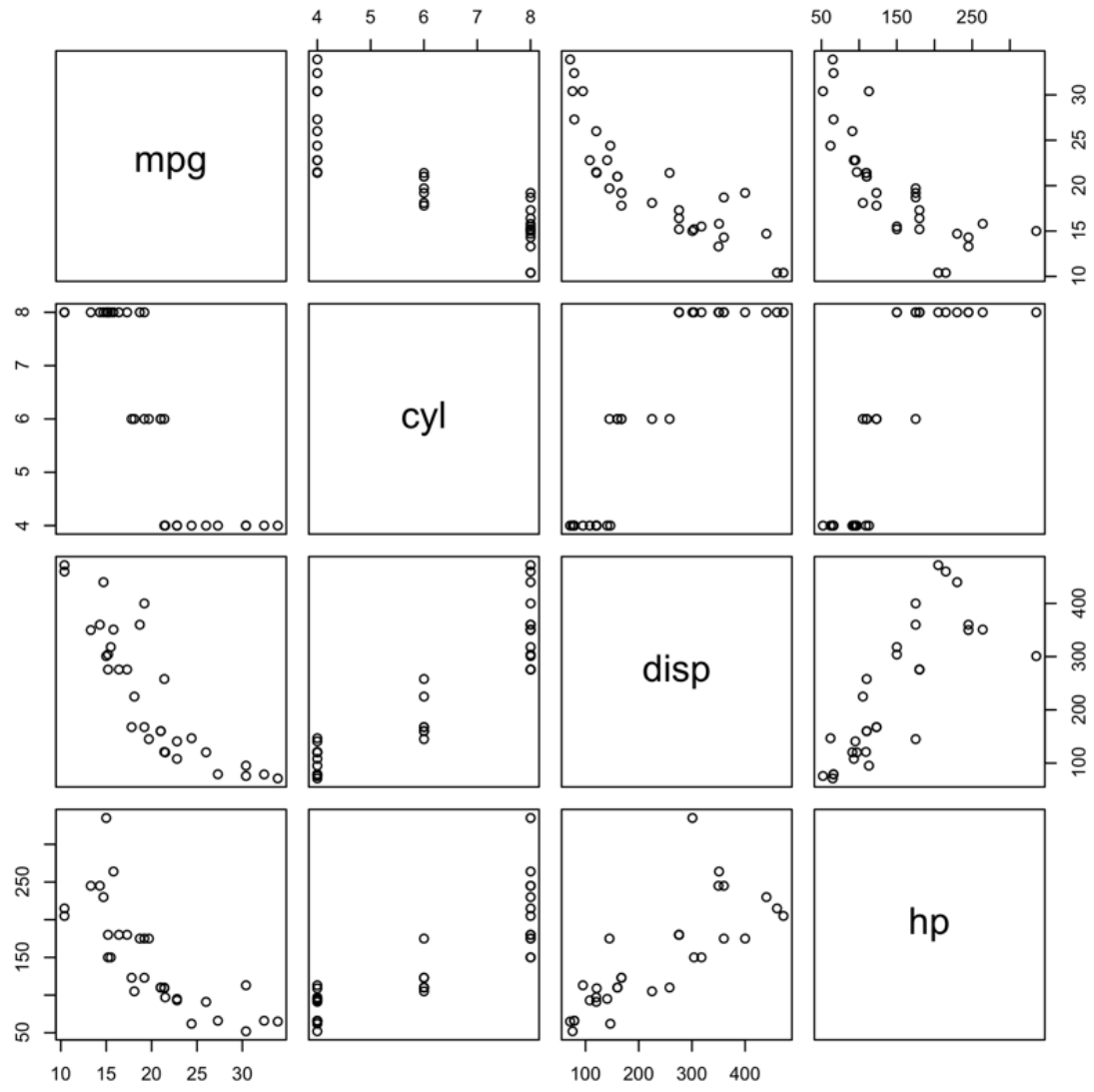


- A specific clustering can be constructed by **cutting** the dendrogram.
- When you use `hclust` to perform a cluster analysis, you can see the dendrogram by passing the result of the clustering to the `plot` function.

EXAMPLE: `mtcars` DATASET IN R

- The `mtcars` dataset comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973–74 models).
- For these methods, the dendrogram is the main graphical tool for gaining an insight into a cluster solution.
- When you use `hclust` to perform a cluster analysis, you can see the dendrogram by passing the result of the clustering to the `plot` function.

EXAMPLE: `mtcars` DATASET IN R



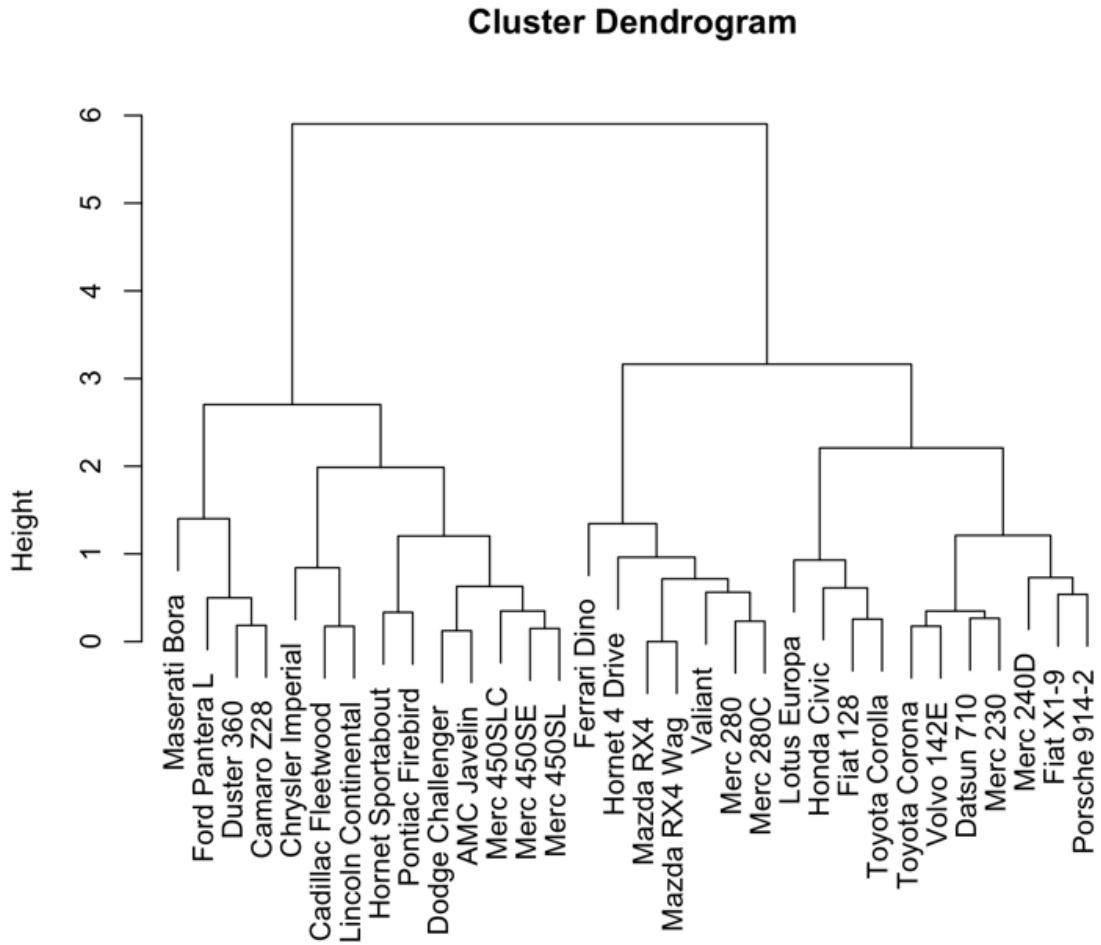
Distance Matrix

- The first requirement for hierarchical clustering is calculating a matrix of dissimilarity or distance.
- We make use of the `dist` function in `R` to calculate these distance matrices.
- For example, in the `mtcars` dataset, we calculate the distance using `car.dist=dist(mtcars[,1:4])`.

EXAMPLE: `mtcars` DATASET IN R

Using the syntax below we obtain a dendrogram.

```
cars.hclust = hclust(cars.dist) plot(cars.hclust)
```



Interpreting a cluster analysis

- One of the first things we can look at is how many cars are in each of the groups.
- We can create a vector showing the cluster membership of each observation by using the `cutree` function.
- Since the object returned by a hierarchical cluster analysis contains information about solutions with different numbers of clusters, we pass the cutree function the cluster object and the number of clusters we're interested in.
- To get cluster memberships for the three cluster solution we use `groups.3 = cutree(cars.hclust,3)`.

Interpreting a cluster analysis

- We can then summarise the data in each of the clusters.
- A good first step is to use the table function to see how many observations are in each cluster, with the code `table(groups.3)`.

```
> table(groups.3)
groups.3
 1  2  3
 8 20 10
```

- We'd like a solution where there aren't too many clusters with just a few observations, because it may make it difficult to interpret our results.

Advantages / Disadvantages

- The **advantages** of hierarchical clustering are
  - It provides consist results given a specified dissimilarity between two different groups of points
  - We can compute multiple, nested clusterings.
- One **disadvantage** of hierarchical clustering is that the best clusters may not be nested!
  - Imagine a situation where the data contains Gender (M/F) and Nationality (UK, US, Australia)
  - The best 2-clustering could split the data by Gender
  - The best 3 clustering could split the data by Nationality
  - These clusterings are not nested, and therefore, hierarchical clustering will perform poorly.
- Another **disadvantage** is that it is slow and computationally expensive for big datasets.

K-means Clustering

K-means Clustering

- In this approach, observations are divided into k groups and reshuffled to form the most cohesive clusters possible according to a given criterion. We use the K-means approach.
- Unlike the hierarchical clustering, K-means require that we specify the number of clusters that will be formed in advance.
- We implement this method in `R` using the `kmeans` function.

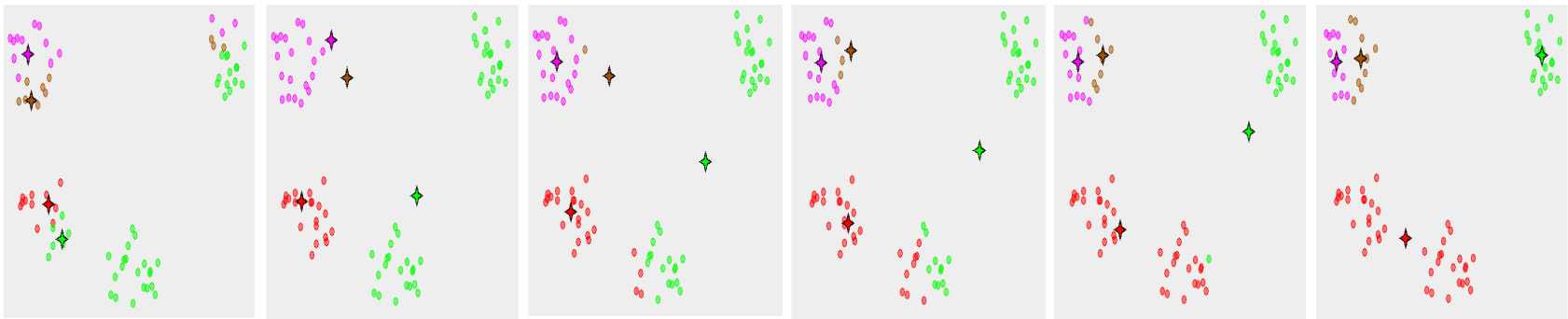
K-means Algorithm

1. Selects K cluster centres.
2. Assigns each data point to its closest cluster centre.
3. Recalculates the locations of the cluster centres as the average of all data points.
4. Assigns data points to their nearest cluster centres.
5. Repeat steps 3 and 4 until observations are not reassigned or the maximum number of iterations is reached. Note: `R` uses 10 as default.

Feature of K-Means Clustering

- K-means clustering can handle larger datasets than hierarchical cluster approaches.
- Additionally, observations are not permanently committed to a cluster.
- They are moved if doing so improves the overall solution.
- However, the use of means implies that all variables must be continuous and the approach can be severely affected by outliers.

K-means Clustering in R



EXAMPLE: mtcars DATASET IN R

- Let's look at the sort of output that the kmeans function provides for the mtcars dataset. We'll specify the number of clusters as 5.
- We can view the clusters that result from this.

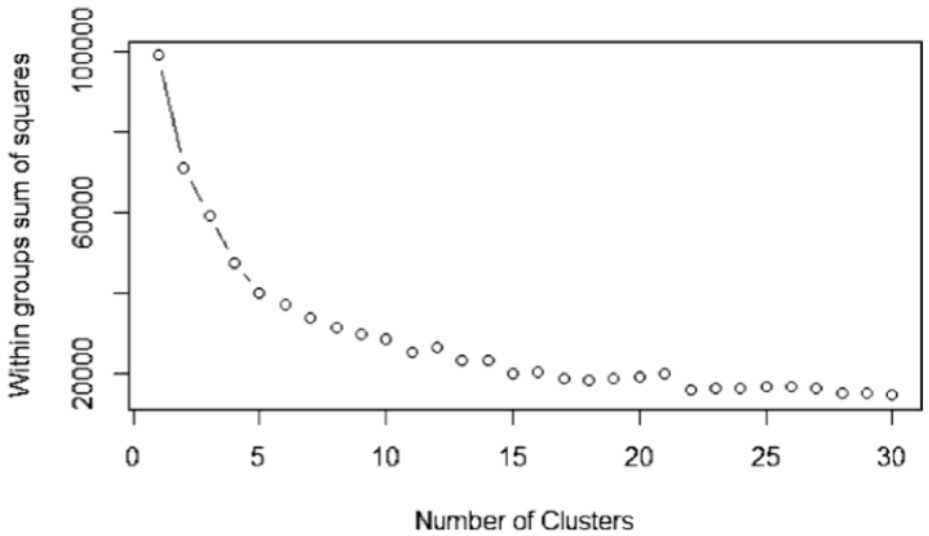
```
> kmeans(data.matrix(mtcars[,1:4]), 5)$cluster
          Mazda RX4       Mazda RX4 Wag          Datsun 710      Hornet 4 Drive
                  1                   1                   4                   3
  Hornet Sportabout             Valiant          Duster 360           Merc 240D
                  5                   1                   5                   1
           Merc 230            Merc 280           Merc 280C          Merc 450SE
                  1                   1                   1                   3
          Merc 450SL         Merc 450SLC  Cadillac Fleetwood Lincoln Continental
                  3                   3                   5                   5
 Chrysler Imperial            Fiat 128          Honda Civic       Toyota Corolla
                  5                   4                   4                   4
      Toyota Corona     Dodge Challenger           AMC Javelin          Camaro Z28
                  4                   3                   3                   5
   Pontiac Firebird            Fiat X1-9       Porsche 914-2         Lotus Europa
                  5                   4                   4                   4
     Ford Pantera L         Ferrari Dino        Maserati Bora          Volvo 142E
                  5                   2                   5                   4
```

- We can also view the cluster centre points.

```
> kmeans(data.matrix(mtcars[,1:4]), 5)$centers
       mpg      cyl     disp        hp
1 31.00000 4.000000  76.1250  62.25000
2 14.64444 8.000000 388.2222 232.11111
3 24.18571 4.000000 121.7143  94.28571
4 16.83333 7.666667 284.5667 158.33333
5 19.46667 6.000000 170.8667 124.33333
```

Interpreting K-Means Output

- Unlike hierarchical clustering, K-means clustering requires that the number of clusters to extract be specified in advance.
- A plot of the total within-groups sums of squares against the number of clusters in a K-means solution can be helpful.
- Inspection of the graph can help to suggest the appropriate number of clusters.



Final Thoughts

- Clustering is an important part of exploratory data analysis for large data with multiple variables.
- There are no “hard and fast" rules on how to do it (especially on how to choose k)
- But that doesn’t mean that you can’t glean insight.
- In practice, you should cluster subsets of the data, and see if the clusters remain the same.

Any Questions?