K-Means

Recall Supervised Learning

- Some supervised learning methods such as: linear model, logistic model, KNN, Decision Tree, Naive Bayes, etc.
- Data provide both features and response.
- Data then 'teach' us, given a certain values for x, what is most likely corresponding outcome y.
- However, there are many situations where we do not have response in the data.

Unsupervised Learning

- Suppose we only have data on the predictors x, but not the outcome y.
- We call such data without y as 'unlabeled' data.
- *Unsupervised learning* is the task of finding hidden structure based on data without the outcome y.

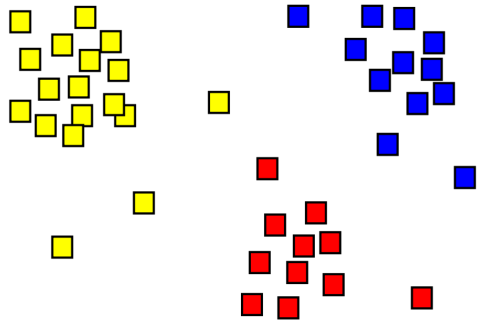
Unsupervised Learning

- A fashion company has a new dress design, and now would transform the design into products. They need to produce the design in various sizes such as S, M, L.
- Based on the weight and height of a set of 1000 people, how would they determine such sizes?
- Obviously, there is no response in the data set. Only two features, height and weight, are given.
- Simplify the question above: how should we cluster people into three clusters/groups (S, M, L)?

Some common supervised and unsupervised learning algorithms

Supervised	**Unsupervised**
Generalized Linear regression	k-means
Decision trees	Association rules
k-nearest neighbor	Hierarchical clustering
Naive Bayes	Self-organizing map
Linear discriminant analysis	Deep belief nets

- *Clustering* refers to a very broad set of techniques for finding subgroups, or clusters, in a data set.
- When we cluster the observations of a data set, we seek to partition them into distinct groups so that the observations within each group are quite similar to each other, while observations in different groups are quite different from each other.

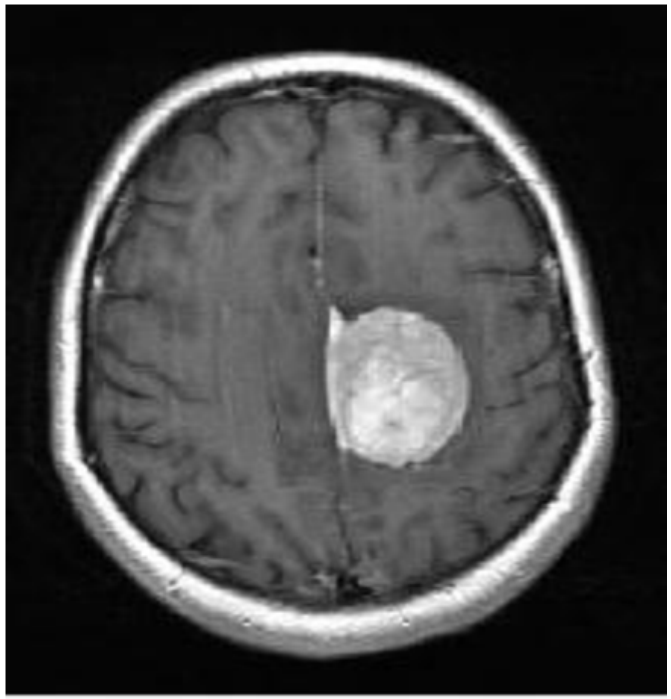


Clustering

- In marketing, clustering methods from data science can discover distinct groups of customers.
- This will allow for more targeted marketing strategies.
- There is no group label for each customer, just his or her amount spent and number of purchases as the features for clustering.
- Aim is to find certain natural groups of customers based on unlabeled data.

Clustering

- Clustering can be used in image processing to segment objects.
- The attributes of each pixel can include brightness, color, location, the x and y coordinates in the frame.
- Clustering can aid medical diagnostics, i.e. identification of clusters with darker pixels.
- Clustering, in general, is useful in biology for the classification of plants and animals as well as in the field of human genetics.



Example: HDB Flats

- We have extracted a subset of all the resale records for 3-Room flats from March 2012 to December 2014.
- For the purposes of illustration for clustering methods, it's added an artificially generated variables, `amenities`.
- Available as the data set `HDBresale_cluster.csv` on the course website.

Example: HDB Flats

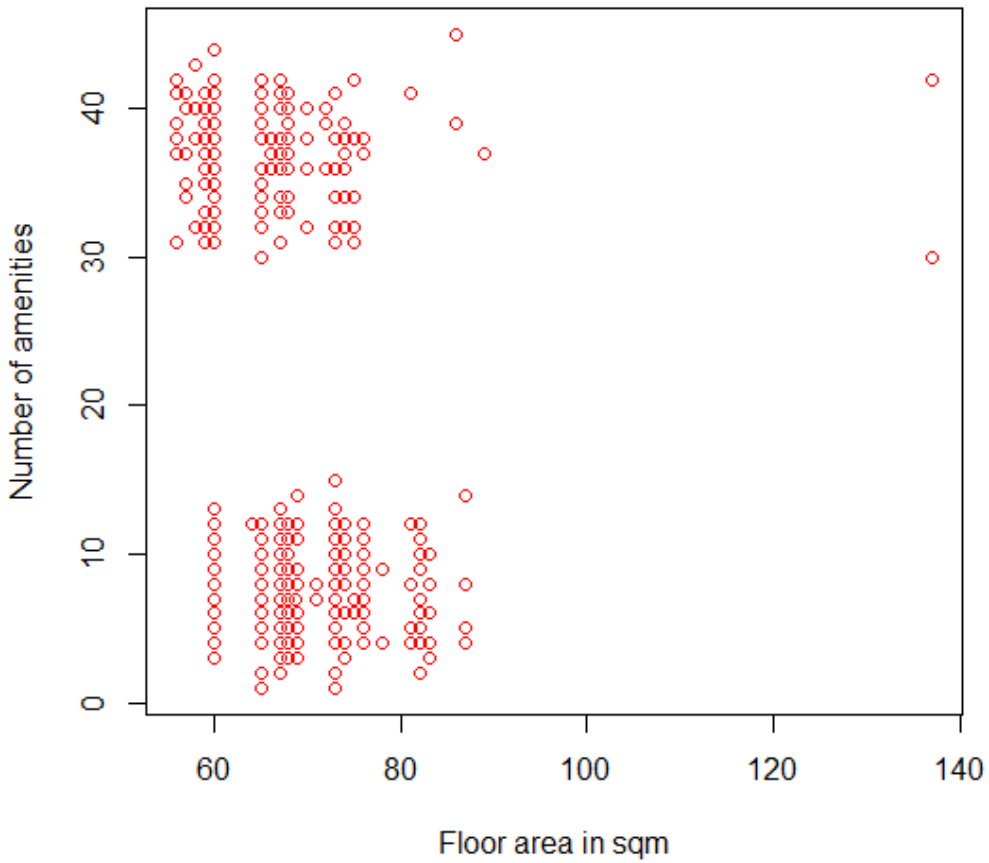
```
> hdb=read.csv("C:/Data/hdbresale_cluster.csv")
> head(hdb)
    X flat_type floor_area_sqm amenities
1 580    3 ROOM             74        32
2 581    3 ROOM             74        37
3 582    3 ROOM             73        34
4 583    3 ROOM             59        38
5 584    3 ROOM             68        39
6 586    3 ROOM             68        36
> dim(hdb)
[1] 774   4
> #flat_type: all are 3-ROOM
```

- Data has only two features:

(i) Floor area in square meters

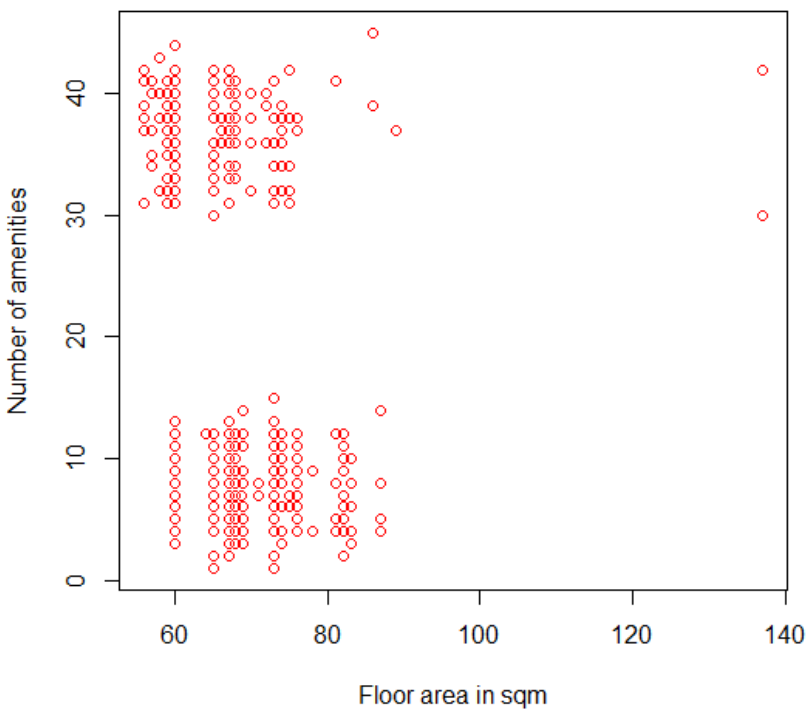
(ii) The number of amenities near the flat

Example: HDB Flats



Data points seem in two subsets.

Example: HDB Flats



- Clustering method can divide these points into subsets.
- These subsets are called clusters and are comprised of data points that are most similar to one another.
- For this data set, it appears that there are two clusters, one in the higher number of amenities group and the other in the lower number of amenities group.

Clustering

- Before running the clustering algorithm, you don't have an exact idea as to the nature of the subsets (clusters).
- However, by visual inspection, it seems that there are two distinct clusters in our HDB resale data set.
- We need an algorithm that provides us with a principled and automatic way for clustering.
- There are many clustering methods in data science that provide valuable insights from a given data.
- Examples include K-means clustering, mean-shift clustering, density-based spatial clustering of applications with noise (DBSCAN), etc.

Notations

- We have a data set of n observations with p features.
- We'll build a classifier that helps to put n observations into k clusters.
- Determining the value of k is somehow subjective. Visualizing the data will give a better option for k.

K-Mean Algorithm

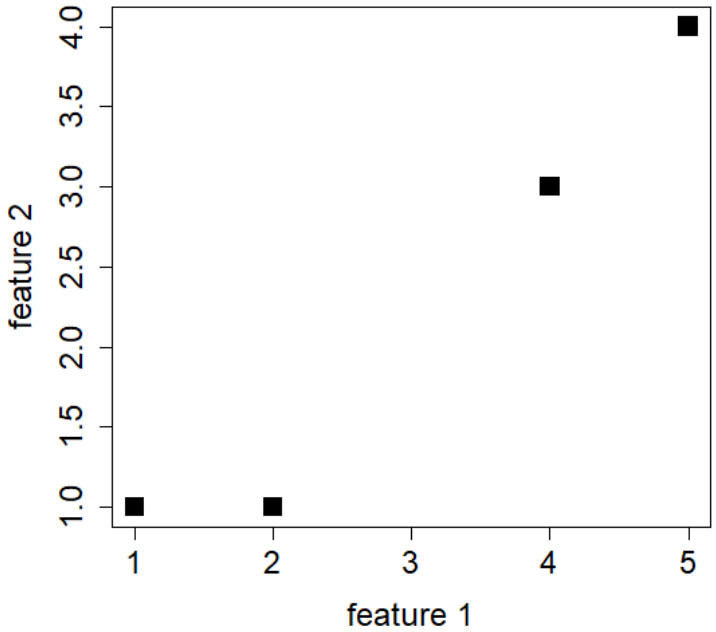
A MUST-watch video before continuing:
https://www.youtube.com/watch?v=hDmNF9JG3lo&t=6s

K-Mean Algorithm

- For illustration, we assume $k = 2$.
- We then start with two points, C_1 and C_2, that we think they could be the centroid for the two clusters.
- Then, we start to divide the data points into two groups, by:
  - (i) For each point, measure the distance from that point to C_1 and to C_2. If the point is nearer to C_1 then it's classified to first cluster; If it's nearer to C_2 then it's classified into second cluster.
  - (ii) Calculate the new centroids for the two clusters, C_1^1 and C_2^1.
- Steps (i) and (ii) above are repeated, until the two centroids are stable.
- We use Euclidean distance to measure distance between two points.

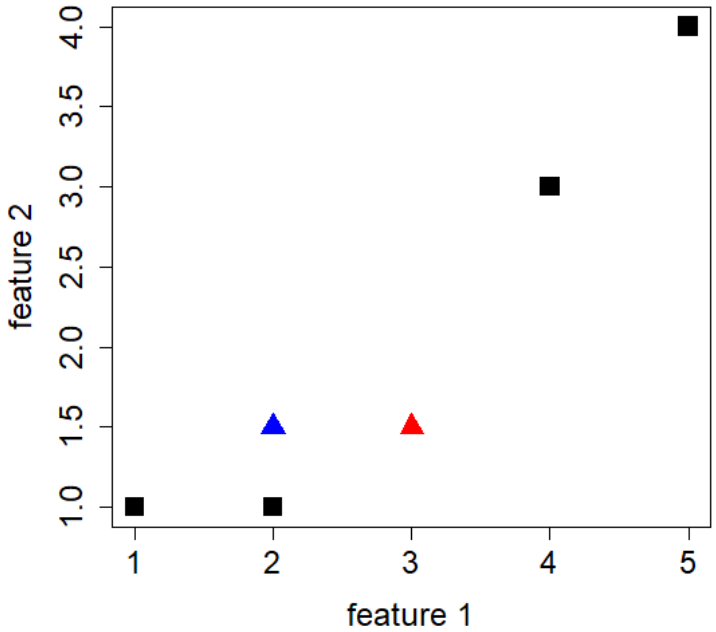
A Manual Example

- Suppose we have four data points, with two features: (1 ,1), (2, 1), (4, 3), (5, 4).
- We want to classify them into $k = 2$ clusters.



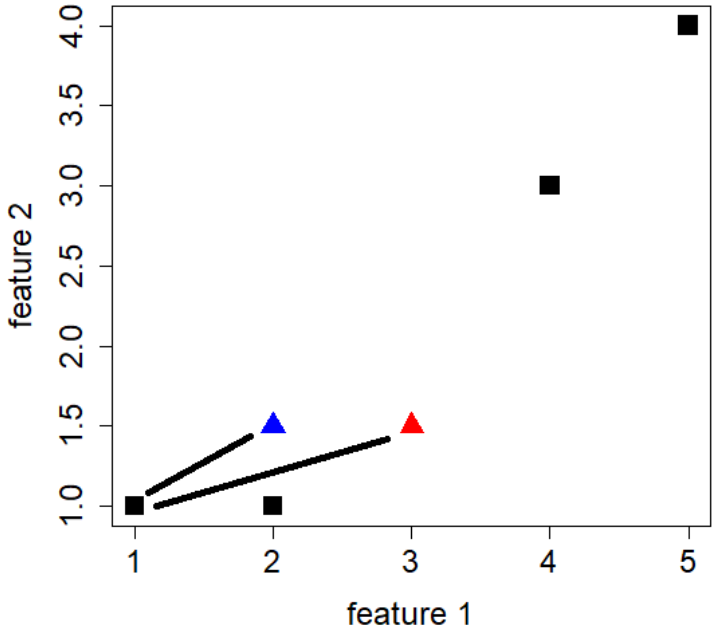
First Iteration

- We initially randomly select two starting centroids: Blue centroid: (2, 1.5) and Red centroid: (3, 1.5).
- We need to calculate the distance of each point to each centroid.

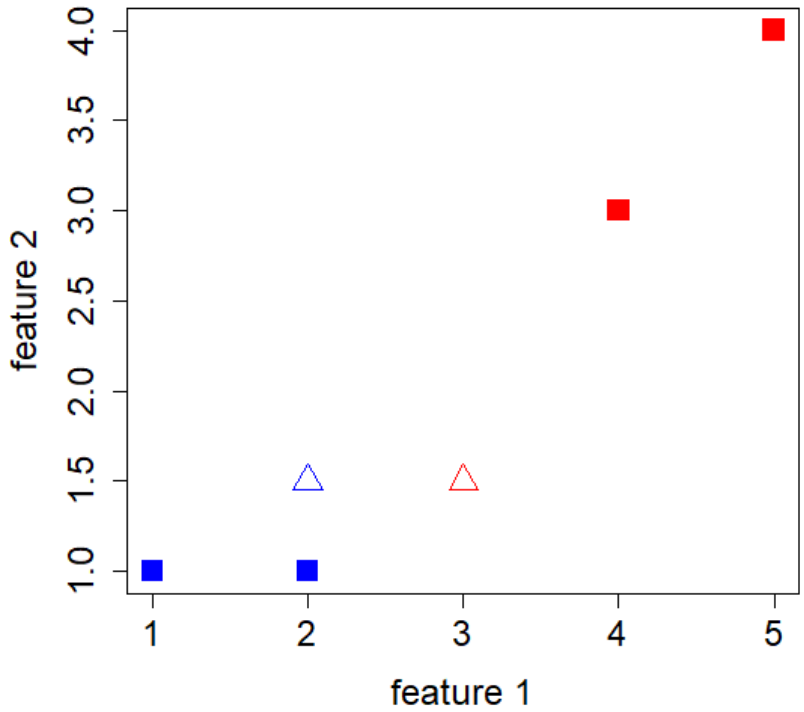


First Iteration

- Distance from first point to blue centroid: $\sqrt{(1-2)^2+(1-1.5)^2} = \sqrt{1.25}$
- Distance from first point to red centroid: $\sqrt{(1-3)^2+(1-1.5)^2} = \sqrt{4.25}$
- So the first point belongs to blue group.
- Similar steps above are applied for the other three points.

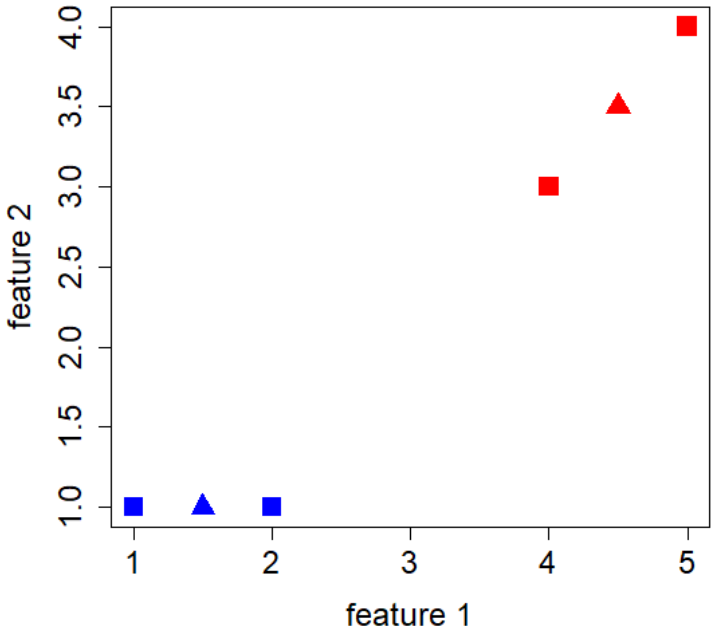


First Iteration



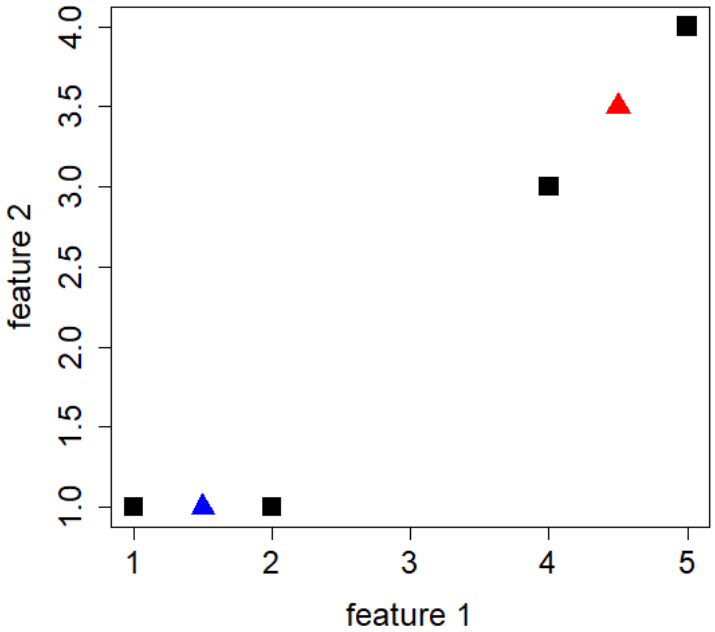
- Point 1 and Point 2 are classified as Blue while Point 3 and Point 4 are Red.
- The initial centroids now are removed. We'll calculate the new centroids, based on the points in Blue and points in Red.

Results of First Iteration



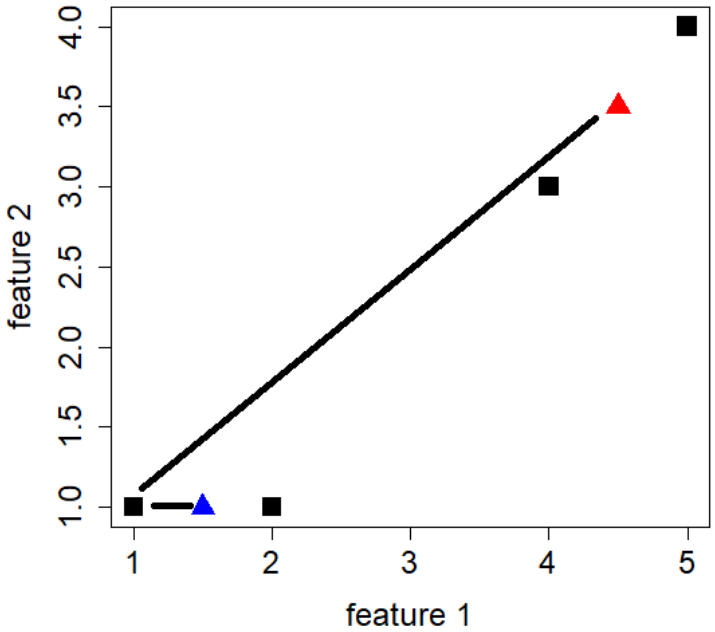
- New blue centroid (triangle):
  $(\frac{1+2}{2}, \frac{1+1}{2}) = (1.5, 1)$
- New red centroid (triangle):
  $(\frac{4+5}{2}, \frac{3+4}{2}) = (4.5, 3.5)$

Start of 2nd Iteration



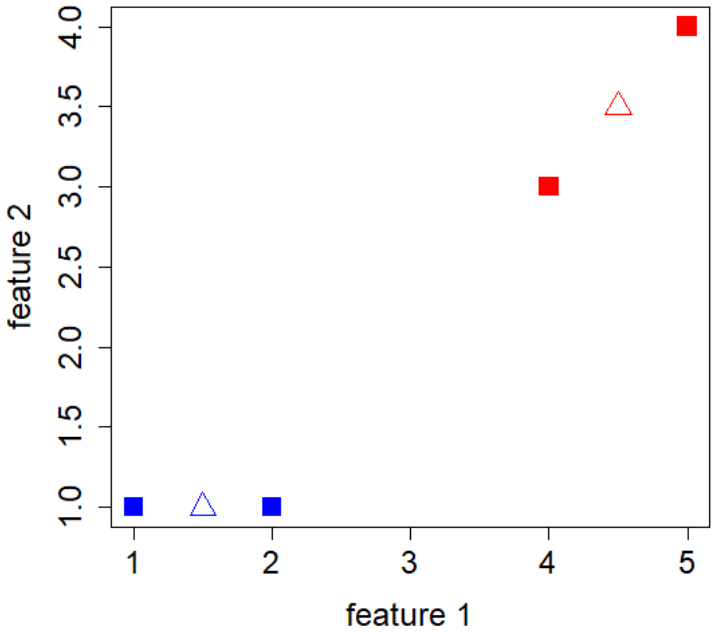
- We move on to the next iteration of the $2 - means$ algorithm with the two centroids obtained from the 1st iteration.

2nd Iteration



- First point is nearer to the Blue centroid; Hence, it's now classified to Blue.
- Similarly, we classify the rest 3 points.

2nd Iteration



- We have completed the 2^{nd} iteration of the $2 - means$ algorithm.
- Now, we calculate the new centroids in the two clusters and notice that the computed centroids do not change in their coordinates from the start of the 2^{nd} iteration.
- The algorithm has converged.

K-Means

- The k-means clustering algorithm can be generalized to cluster objects with more than two features.

- User is supposed to choose a value for k.

HDB Flats

- With the data set provided, `HDBresale_cluster.csv`, we would use K-Means method to classify the flats into clusters.

- We can use `kmeans()` function in R.

```
> set.seed(1)
> dim(hdb)
[1] 774   4
> head(hdb)
    X flat_type floor_area_sqm amenities
1 580    3 ROOM             74        32
2 581    3 ROOM             74        37
3 582    3 ROOM             73        34
4 583    3 ROOM             59        38
5 584    3 ROOM             68        39
6 586    3 ROOM             68        36
```

K-Means in R

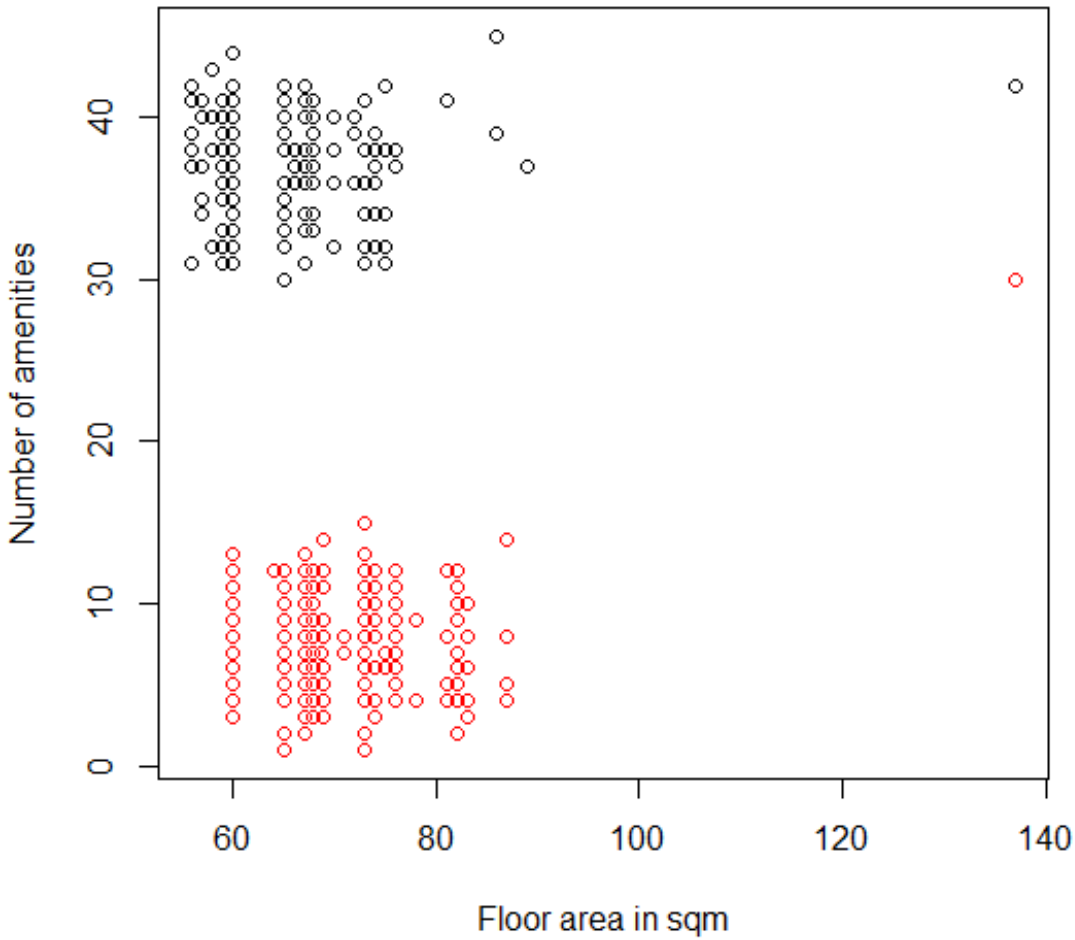
- Specify the number of clusters, and the features used for clustering:

```
> kout = kmeans(hdb[ , c("floor_area_sqm","amenities")]
+               ,centers = 2)
> # centers = number of clusters
```

- We would visualize the clusters:

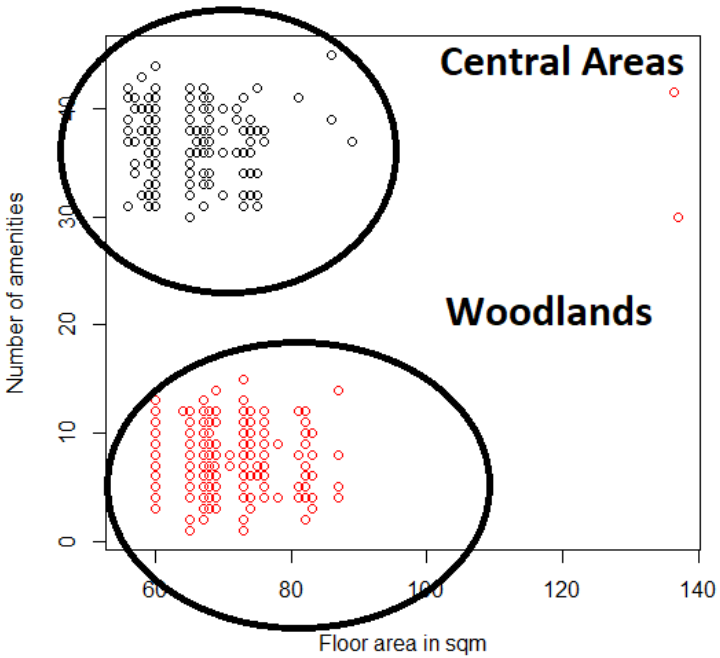
```
> plot(hdb$floor_area_sqm,
+      hdb$amenities,
+      col=kout$cluster)
> kout$centers # A matrix of cluster centers.
  floor_area_sqm amenities
1       65.40782 37.279330
2       69.46017  7.448739
> kout$size # The number of points in each cluster.
[1] 179 595
```

Prediction by K-Means



- We see that the results from $k - means$ clustering agree with our initial visual inspection.

The Fact is



- In fact, the data set includes flats from either central area or Woodlands in Singapore.
- We see that $k - means$ clustering has helped us uncover this hidden structure, based on the original unlabeled data.

K-Means with only 2 Features

- In the HDB flats example, we only have two features for clustering

  (1) Floor area in square meters

  (2) The number of amenities in the vicinity of the HDB unit

- It's relatively easy to see the data in two dimensions and determine that there are two clusters.

- Is there a way to determine the number of clusters, k, when the number of features for clustering is of higher dimensions?

Within Sum of Squares (WSS)

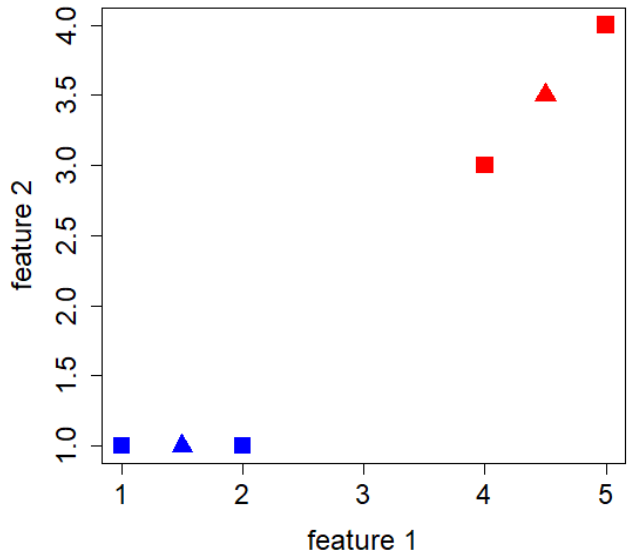
- Selection of the number of clusters k can be guided by the metric *Within Sum of Squares* (WSS).
- To ground ideas, we might group the points by k clusters with different values of k. Then, get WSS for each k.
- The value of k that gives stable WSS (WSS does not change much when k changes) could be chosen.
- With $k = 2$, WSS is then

$$WSS = SS_1 + SS_2,$$

where SS_k is the sum of squared distances from each point to its centroid.

Within Sum of Squares (WSS)

- For this simulated example, we have determined with Blue centroid at $(1.5, 1)$ and Red centroid at $(4.5, 3.5)$.
- $SS_{Blue} = \left(\sqrt{(1-1.5)^2 + (1-1)^2}\right)^2 + \left(\sqrt{(2-1.5)^2 + (1-1)^2}\right)^2 = 0.5.$
- $SS_{red} = \left(\sqrt{(4-4.5)^2 + (3-3.5)^2}\right)^2 + \left(\sqrt{(5-4.5)^2 + (4-3.5)^2}\right)^2 = 1.$
- $WSS = SS_{blue} + SS_{red} = 0.5 + 1.0 = 1.5$
- That is for $k = 2$.



WSS for HDB Flats Example

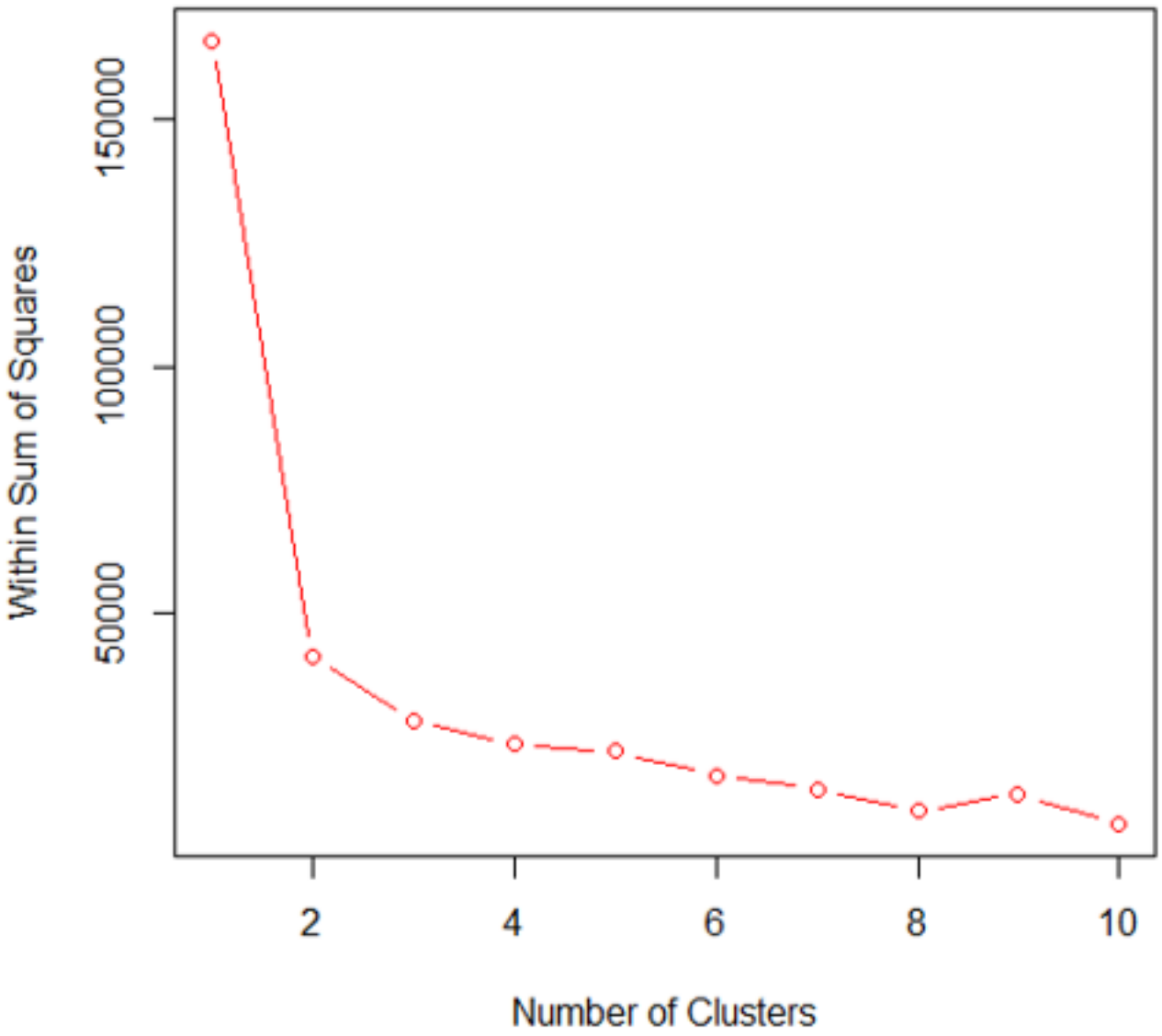
- We grouped the flats into $k = 2$ clusters. The SS_k is then

```
> kout$withinss # Vector of SS_k, one value per cluster
[1] 15347.26 26043.23
```

- The WSS is then

```
> sum(kout$withinss) # OR
[1] 41390.5
> kout$tot.withinss
[1] 41390.5
```

WSS versus K for HDB Flats



Grade Grouping

- The task is to group 620 high school seniors based on their grades in three subject areas: English, Mathematics, and Science.
- The grades are averaged over their high school career and assume values from 0 to 100.
- Data set is `grades_km_input.csv`.

```
> grade = read.csv("C:/Data/grades_km_input.csv")
> grade[1:4,]
  Student English Math Science
1       1      99   96      97
2       2      99   96      97
3       3      98   97      97
4       4      95  100      95
```

- Can you try the algorithm and determine the optimal k?