Cluster analysis pt.1

k-means

What is clustering?

- Is a form of unsupervised learning used to group similar observations together such in a way that the observations in a cluster, are more similar to one another than they are to observations in another cluster.

- A form of Exploratory Data Analysis (EDA) where observations are divided into meaningful groups that share common characteristics (features).

- We aim to minimize the intracluster variance (within cluster variance) = WSS = «within sums of squares»

Clustering methods

01	Partition techinique

Find centers of clusters and each point is assigned to the cluster that has the closest center.

k-means

02	Hierarchical techniques

Connect the observations based on their similarity to form clusters.

Hierarchical clustering

03	Model-base methods

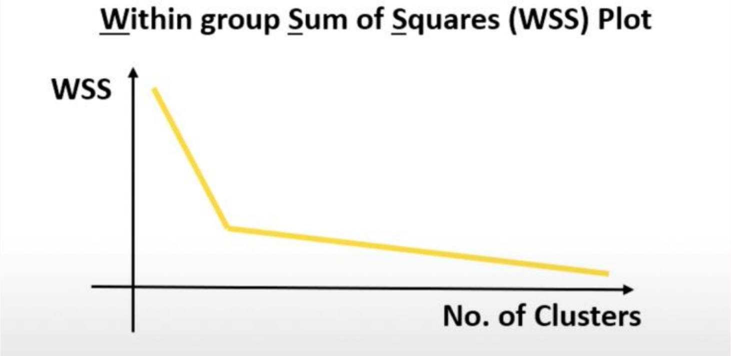
Use probabilistic distribution to create clusters.

Mixture models

The flow of cluster analysis

01	Pre-process data
02	Select similarity measure
03	Cluster
04	Analyze

How to choose optimum number of clusters?

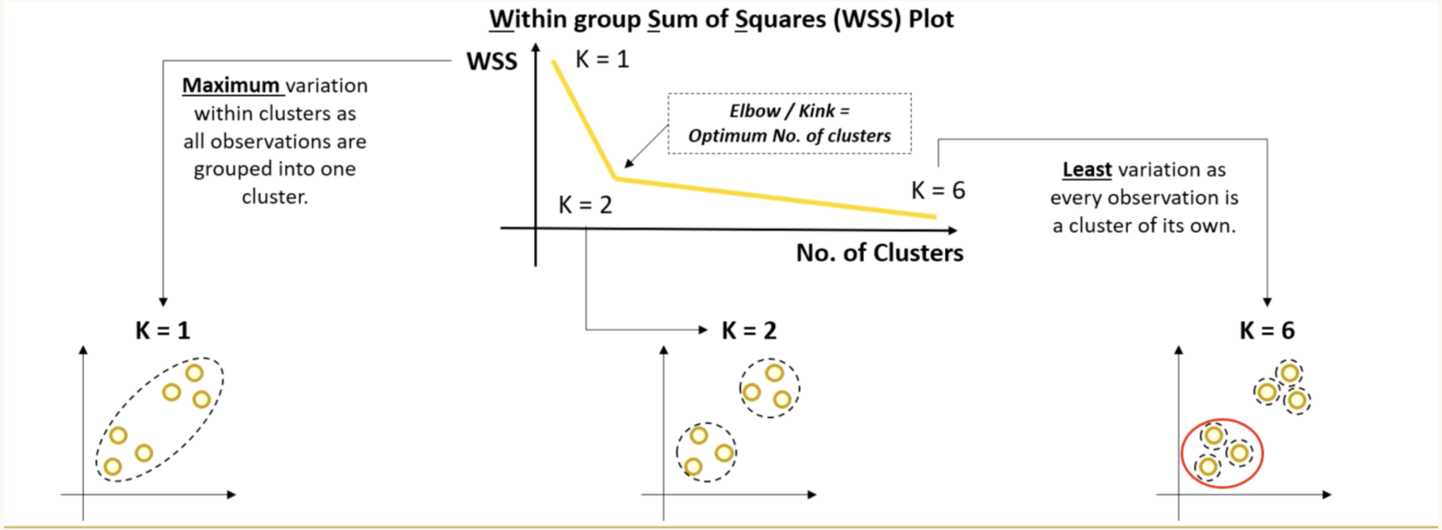


High WSS Value	Low WSS Value
Variation within the clusters is **high**	Variation within the clusters is **low**

How do we pick the value of k?

- Find the k-value for which there is no longer a meaningful decrease in the total WSS.
- If we pick the value of k that results the lowest total within SS, we might have far too many clusters. To prevent this we may use the Davies-Bouldin index, which penalizes overfitting and lower values are preferred.

How to choose optimum number of clusters?



The lowest number

The optimal number of clusters

The biggest number

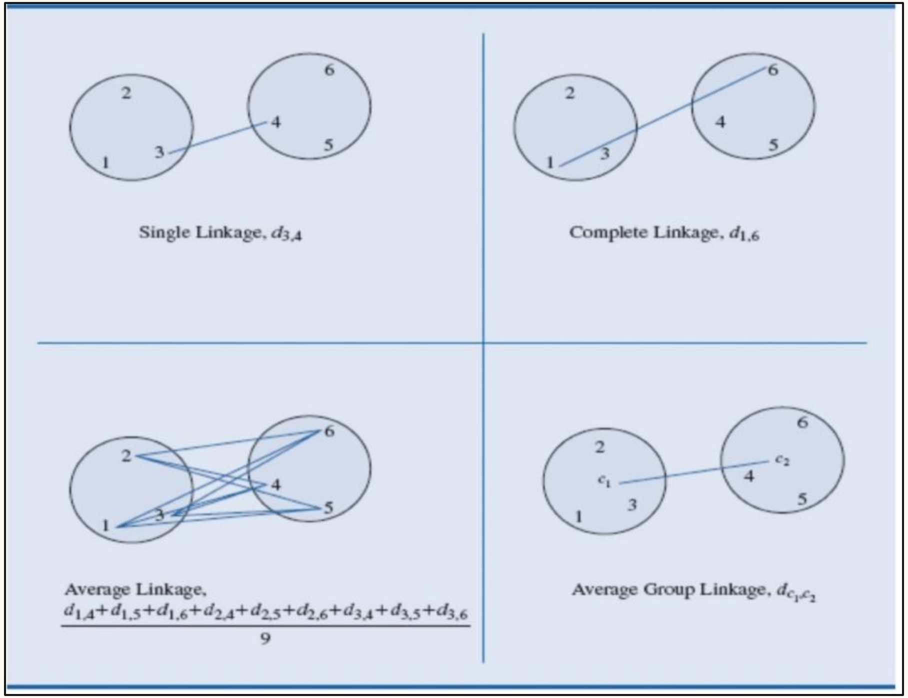
01 k-means

The k-means algorithm is an algorithm to cluster n objects, based on attributes into k partitions, where k<n.

Measuring similarity between clusters

Measuring similarity between clusters:

1. Single linkage
2. Complete linkage
3. Average linkage
4. Average group linkage



Measuring similarity between observations

- Euclidean distance is the most common method to measure distance between observations, when observations include continuous variables.
- Let observations u = (u1, u2, u3..., uq) and v = (v1, v2, v3..., vq) each comprise measurements of q variables.
- The Euclidean distance between observations u and v is:

$$d_{u,v} = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \cdots + \left(u_q - v_q\right)^2}$$

k-means algorithm

Step 1

Randomly select k observations, called «cluster centroids».

Step 2

Compute the Euclidean distance from every observation to each of the k cluster centroid.

Step 3

For each of the n observations, assign the observation to its closest cluster.

Step 4

Update the cluster centroid for each of the k clusters.

Step 5

Cluster centroids no longer «move» or maximum of iteration reached

k-means clustering example

- This example helps understand better the mechanisms of clustering with k-means methods.
- Exercise: Cluster the following 8 points into 3 clusters:
  A1 (2,19)
  A2 (2,5)
  A3 (8,4)
  A4 (5,8)
  A5 (7,5)
  A6 (6,4)
  A7 (1,2)
  A8 (4,8)

k-means clustering example

- Since k number is 3, let's decide 3 points randomly from the set of the point we have.
- Suppose that we have initial cluster centres as:
  A1(2,10)   A4(5,8)   A7(1,2)
- The distance function b $\rho(a, b) = |x2 - x1| + |y2 - y1|$ *(x2,y2)* is defined as
- Let's use k-mean algorithm to find 3 cluster centres after the second iteration:

k-means clustering example

			2 10	5 8	1 2	
	Points		Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
	x	y				
A1	2	10	0	5	9	1
A2	2	5	5	6	4	3
A3	8	4	12	7	9	2
A4	5	8	5	0	10	2
A5	7	5	10	5	9	2
A6	6	4	10	5	7	2
A7	1	2	9	10	0	3
A8	4	9	3	2	10	2

Cluster 1	Cluster 2	Cluster 3
(2, 10)	(8, 4)	(2, 5)
	(5, 8)	(1, 2)
	(7, 5)	
	(6, 4)	
	(4, 9)	

k-means clustering example

- Repeating the same thing but this time updating the center of the clusters.

- Center of each cluster is updated by calculating the centroids of each of these clusters, that is the average of all x and y.

- Next we need to recompute the new clusters (mean) by taking the mean of all points in each cluster.

- For Cluster 1, we only have A1(2,10) so remains the same.
- For Cluster 2, the new center results (6,6)
- For Cluster 3, the new center results (1,5; 3,5)

k-means clustering example

			2 10	6 6	1,5 3,5	
	Points		**Dist Mean 1**	**Dist Mean 2**	**Dist Mean 3**	**Cluster**
	x	**y**				
A1	2	10	0	8	7	1
A2	2	5	5	5	2	3
A3	8	4	12	4	7	2
A4	5	8	5	3	8	2
A5	7	5	10	2	7	2
A6	6	4	10	2	5	2
A7	1	2	9	9	2	3
A8	4	9	3	5	8	1

Cluster 1	Cluster 2	Cluster 3
(2, 10)	(8, 4)	(2, 5)
(4, 9)	(5, 8)	(1, 2)
	(7, 5)	
	(6, 4)	

k-means clustering example

	Points		Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
			3 9,5	6,5 5,25	1,5 3,5	
	x	y				
A1	2	10	1,5	9,25	7	1
A2	2	5	5,5	4,75	2	3
A3	8	4	10,5	2,75	7	2
A4	5	8	3,5	4,25	8	1
A5	7	5	8,5	0,75	7	2
A6	6	4	8,5	1,75	5	2
A7	1	2	9,5	8,75	2	3
A8	4	9	1,5	6,25	8	1

Cluster 1	Cluster 2	Cluster 3
(2, 10)	(8, 4)	(2, 5)
(5, 8)	(7, 5)	(1, 2)
(4, 9)	(6, 4)	

Recompute the cluster means again in the same way:

k-means clustering example

	Points		Dist Mean 1	Dist Mean 2	Dist Mean 3	Cluster
			3,6 9	7 4,3	1,5 3,5	
	x	y				
A1	2	10	2,6	10,7	7	1
A2	2	5	5,6	5,7	2	3
A3	8	4	9,4	1,3	7	2
A4	5	8	2,4	5,7	8	1
A5	7	5	7,4	0,7	7	2
A6	6	4	7,4	1,3	5	2
A7	1	2	9,6	8,3	2	3
A8	4	9	0,4	7,7	8	1

Cluster 1	Cluster 2	Cluster 3
(2, 10)	(8, 4)	(2, 5)
(5, 8)	(7, 5)	(1, 2)
(4, 9)	(6, 4)	

Recompute again and this time we do not see any changes, FINAL ALLOCATION.

02 k-means in R

How to perform k-means analysis in R

Arguments of the kmeans() function in R

Arguments

`x`	numeric matrix of data, or an object that can be coerced to such a matrix (such as a numeric vector or a data frame with all numeric columns).
`centers`	either the number of clusters, say k, or a set of initial (distinct) cluster centres. If a number, a random set of (distinct) rows in `x` is chosen as the initial centres.
`iter.max`	the maximum number of iterations allowed.
`nstart`	if `centers` is a number, how many random sets should be chosen?
`algorithm`	character: may be abbreviated. Note that `"Lloyd"` and `"Forgy"` are alternative names for one algorithm.
`object`	an R object of class `"kmeans"`, typically the result `ob` of `ob <- kmeans(..)`.
`method`	character: may be abbreviated. `"centers"` causes `fitted` to return cluster centers (one for each input point) and `"classes"` causes `fitted` to return a vector of class assignments.
`trace`	logical or integer number, currently only used in the default method (`"Hartigan-Wong"`): if positive (or true), tracing information on the progress of the algorithm is produced. Higher values may produce more tracing information.
`...`	not used.

Syntax in R for k-means

Example using the dataset below:

Gender	Salary	Age	Place	Weight	Company	Academic degree
Female	1,5	33	Chicago	80	BMW	Bachelor
Female	1,2	33	Chicago	82,5	Ford	No
Male	2,2	34	New York	100,8	BMW	Bachelor
Male	2,1	42	New York	90	BMW	Master
Female	1,5	29	Chicago	67	Ford	Master
Female	1,7	19	Washington	60	Ford	Master
Male	3	50	Washington	77	Ford	No
Male	3	55	Washington	77	Ford	Bachelor
Female	2,8	31	New York	87	Ford	Bachelor
Male	2,9	46	New York	70	GM	Master
Female	2,78	36	Washington	57	BMW	No
Male	2,55	48	New York	64	GM	Master

Syntax in R for k-means

- 1st, 4th, 6th and 7th columns are non-numeric so must be removed to perform clustering.

```
#Syntax in R for k-means

head(Database_for_cluster_analysis_datatab)
datatab_kmeans <- kmeans(x = Database_for_cluster_analysis_datatab[, c(2,3)],
                         centers = 3,
                         iter.max = 20)
```

- Centers represent value of k (clusters)
- iter.max represent stopping criteria

Reproducibility of k-means

- You must set a seed before running k(means) if you want it to be reproducibile.

```
set.seed(34)

head(Database_for_cluster_analysis_datatab)
datatab_kmeans <- kmeans(x = Database_for_cluster_analysis_datatab[, c(2,3)],
                         centers = 3,
                         iter.max = 20)
```

Output of kmeans() function in R

- The «cluster» element gives the cluster labels for each of the n observation.

```
datatab_kmeans$cluster
```

- The «withinss» gives the WSS for each of the k clusters.

```
datatab_kmeans$withinss
```

- The «tot.withinss» gives the total WSS

```
datatab_kmeans$tot.withinss
```

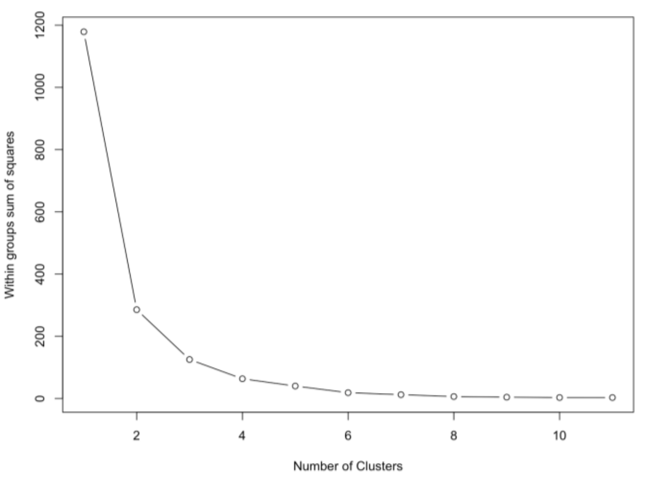
WSS function

- We can not find WSS function by default in R, so we plot the function with this code:

```r
#WSS plot function
  wssplot <- function(mydata, nc=11, seed=34){
    wss <- (nrow(mydata)-1)*sum(apply(mydata,2,var))
    for (i in 2:nc){
      set.seed(seed)
      wss[i] <- sum(kmeans(mydata, centers=i)$withinss)}
    plot(1:nc, wss, type="b", xlab="Number of Clusters",
         ylab="Within groups sum of squares")
    wss
  }
wssplot(mydata)
```

WSS function

- Then we have to spot the «kink» on the curve. There IS the right number of cluster.

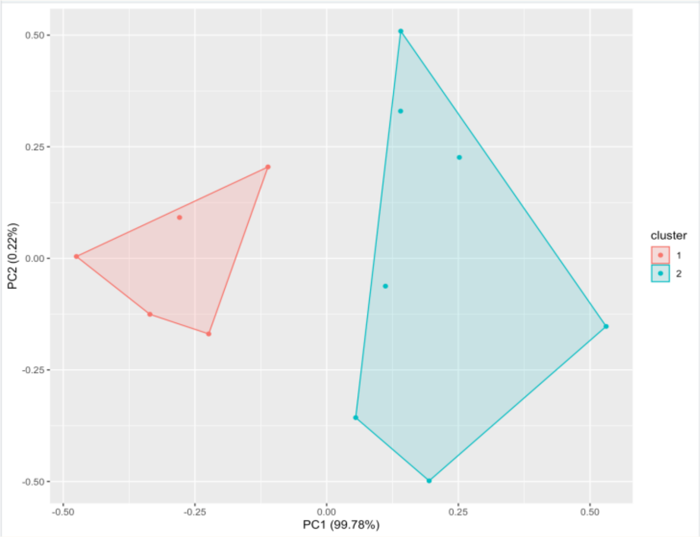



```
#Now we have to spot the kink on the curve
#K-means cluster
KM=kmeans(mydata, 2)
```

Evaluating cluster analysis

```r
#Cluster plot
autoplot(KM, mydata, frame=TRUE)

#Cluster centers
KM$centers
```



continues...