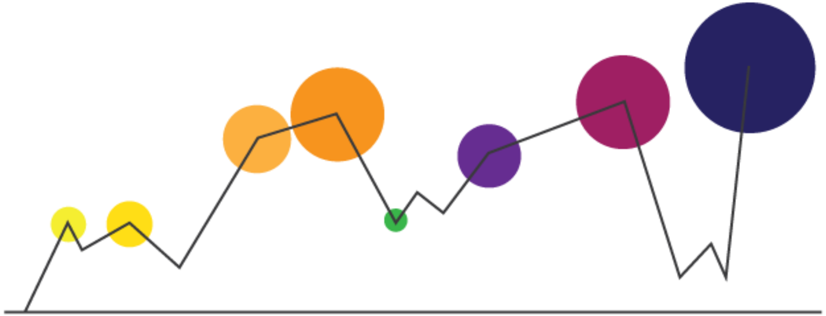
edureka!



MODULE – 7
DATA MINING – CLUSTERING TECHNIQUES

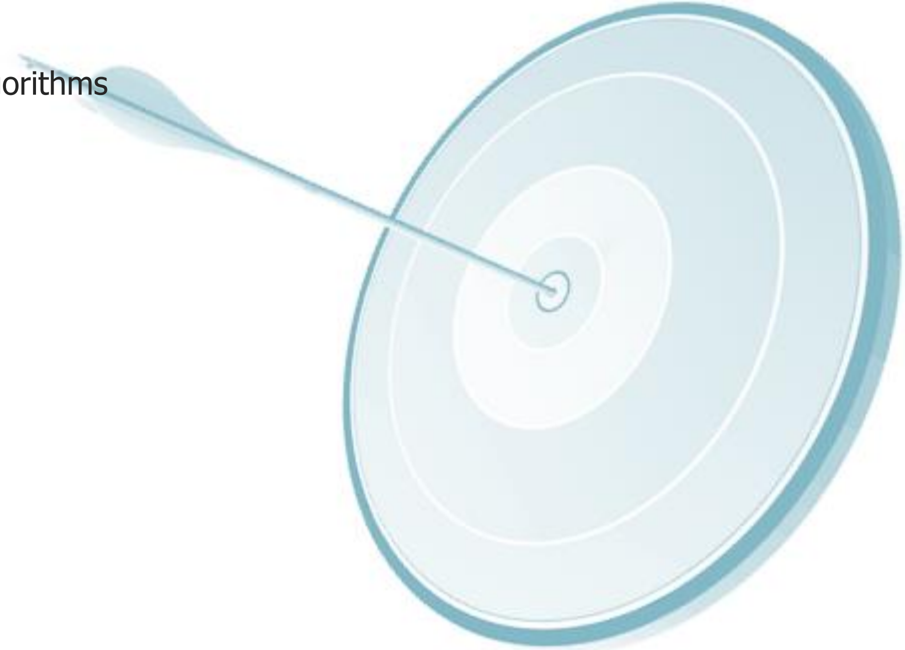
Course Topics

- →Module 1
  - » Introduction to Data Analytics
- →Module 2
  - »Introduction to R Programming
- →Module 3
  - » Data Manipulation in R
- →Module 4
  - » Data Import Techniques in R
- →Module 5
  - » Exploratory Data Analysis
- →Module 6
  - » Data Visualization in R
- →Module 7
  - » **Data Mining: Clustering Techniques**
- →Module 8
  - » Data Mining: Association Rule Mining &Sentiment Analysis
- →Module 9
  - » Linear and Logistic Regression
- →Module 10
  - » Anova and Predictive Analysis
- →Module 11
  - » Data Mining: Decision Trees and Random Forest
- →Module 12
  - »Project

Objectives

At the end of this module, you will be able to:

- → Introduction to Data Mining
- → Understand Machine Learning
- → Supervised and Unsupervised Machine Learning Algorithms
- → K-means Clustering



Introduction to Data Mining

Data today can range in size up to terabytes. Within these masses of data lies hidden information of strategic importance

Data mining is a process that uses a variety of data analysis tools to discover patterns and relationships in data that may be used to make valid predictions

The Data Mining Process include collecting, exploring and selecting the right data

The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.

In this module you will study K-means Clustering and implement the same on ‘Insurance Data’

What is Machine Learning?

Machine learning, a branch of artificial intelligence, concerns the construction and study of systems that can learn from data.

The emphasis of machine learning is on automatic methods.

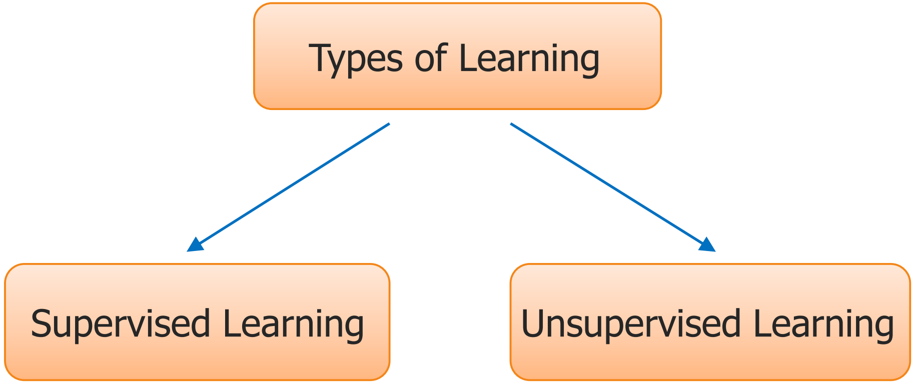
In other words, the goal is to devise learning algorithms that do the learning automatically without human intervention or assistance

Examples:

→ Medical Diagnosis: diagnose a patient as a sufferer or non-sufferer of some disease
→ Customer Segmentation: predict, for instance, which customers will respond to a particular promotion
→ Fraud Detection: identify credit card transactions (for instance) which may be fraudulent in nature
→ Weather Prediction: predict, for instance, whether or not it will rain tomorrow

Learning Techniques

Attain knowledge by study, experience, or by being taught.



Types of Learning

Supervised Learning

Unsupervised Learning

What is Supervised Learning?

Supervised learning:

Training data includes both the input and the desired results.

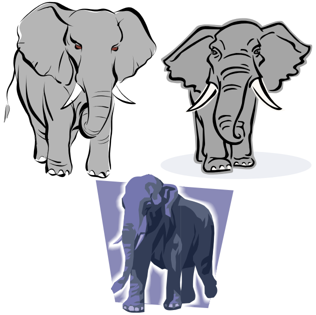
For some examples, the correct results (targets) are known and are given in input to the model during the learning process.

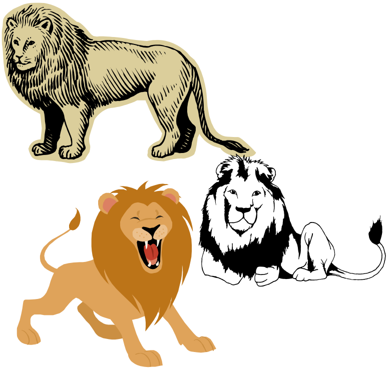
The construction of a proper training, validation and test set (Bok) is crucial.

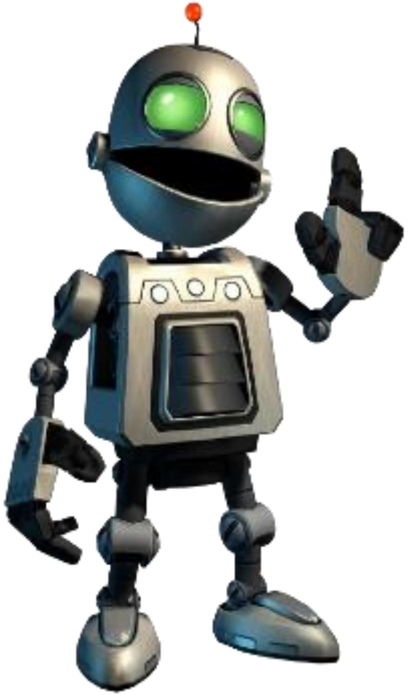
These methods are usually fast and accurate.

Have to be able to generalize : give the correct results when new data are given in input without knowing a priori the target.

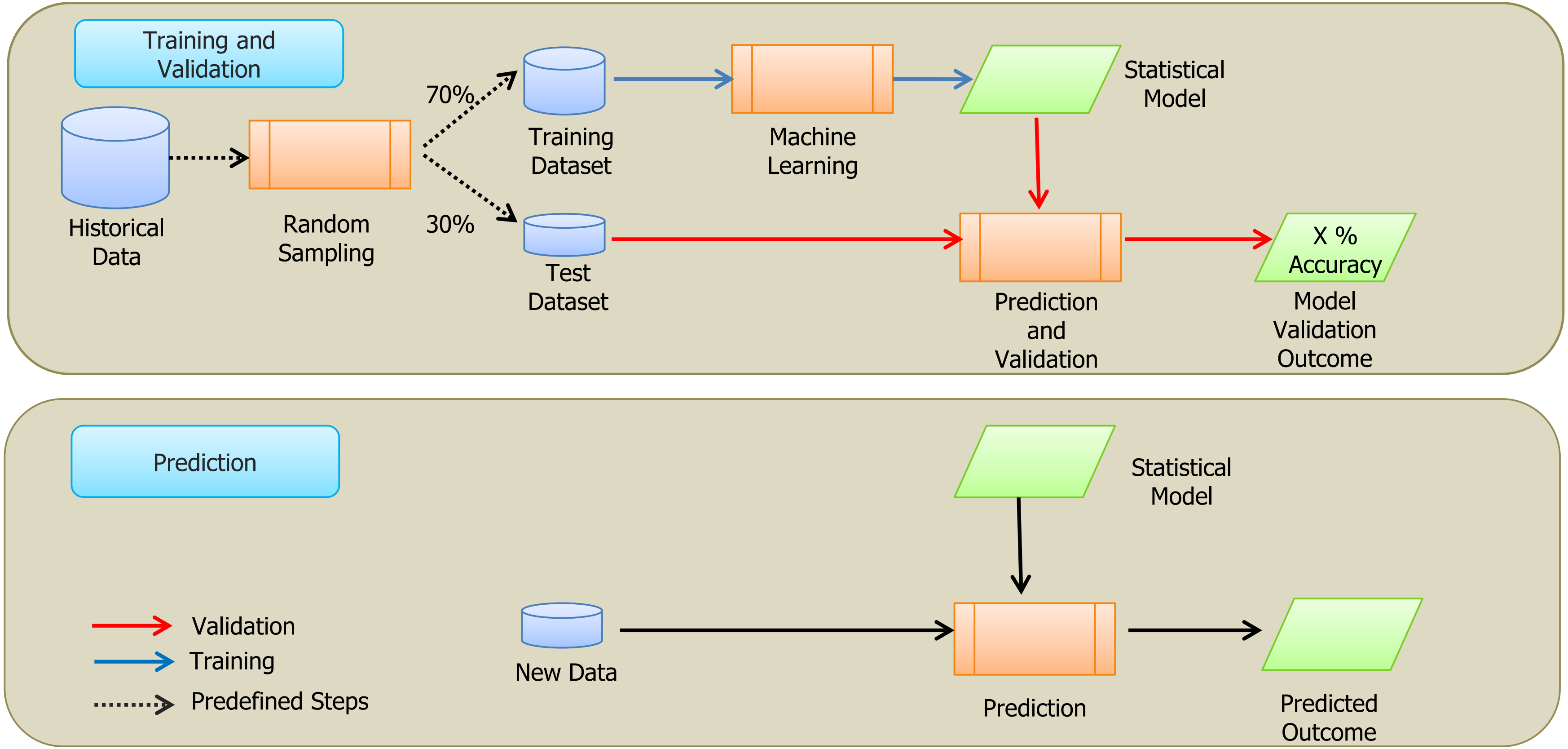
Supervised Learning?







What is Supervised Learning?



What is Unsupervised Learning?

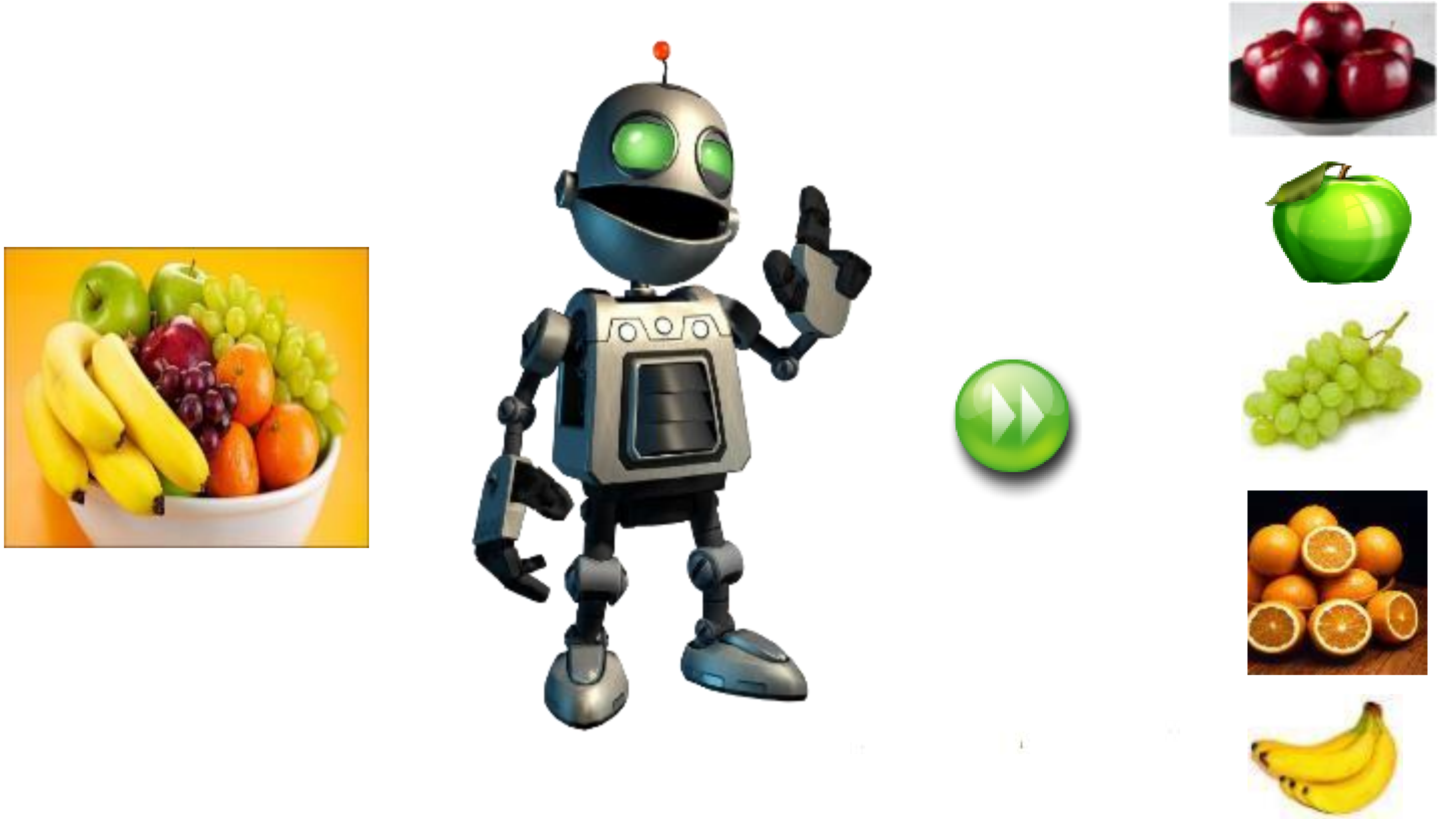
Unsupervised learning:

The model is not provided with the correct results during the training.

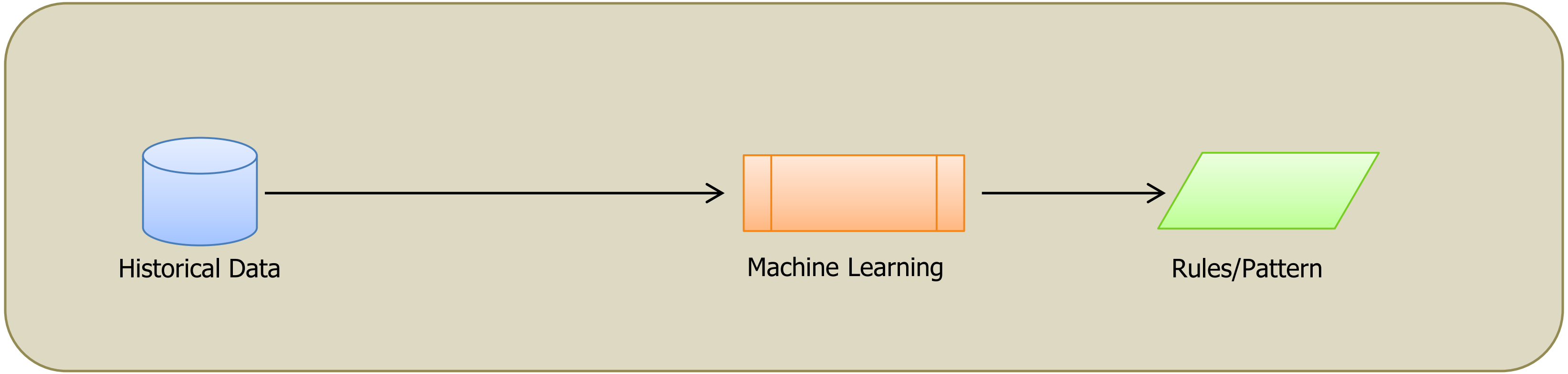
Can be used to cluster the input data in classes on the basis of their statistical properties only Cluster significance and labeling.

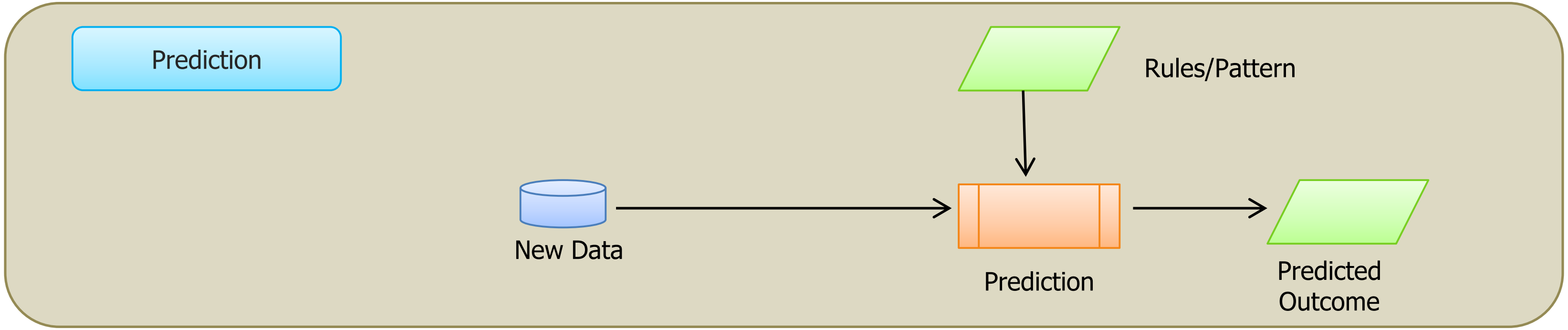
The labeling can be carried out even if the labels are only available for a small number of objects representative of the desired classes.

Unsupervised Learning?



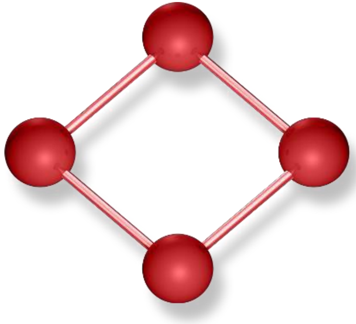
What is Unsupervised Learning?





K-means Clustering

What is Clustering?



Clustering

Organizing data into clusters such that there is:

- → High intra-cluster similarity
- → Low inter-cluster similarity
- → Informally, finding natural groupings among objects.

Why do we want to do it??

Why Clustering?

→ Organizing data into clusters shows internal structure of the data.
Ex. Clusty and clustering genes

→ Sometimes the partitioning is the goal.
Ex. Market segmentation

→ Prepare for other AI techniques.
Ex. Summarize news (cluster and then find centroid)

→ Techniques for clustering is useful in knowledge.

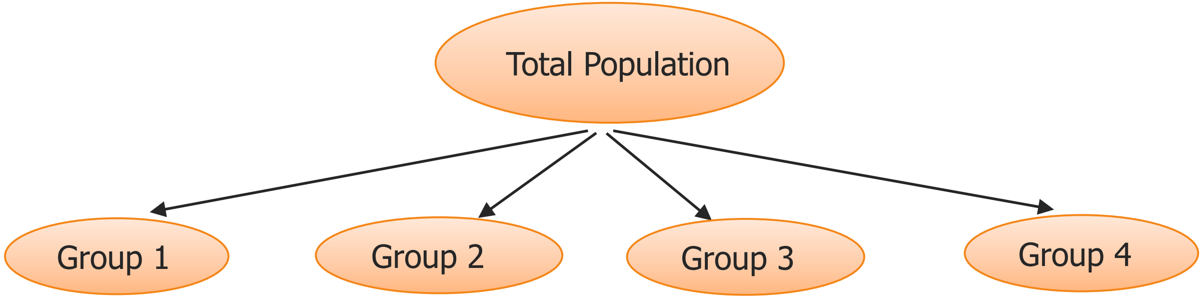
→ Discovery in data.
Ex. Underlying rules, reoccurring patterns, topics, etc.

K-Means Clustering

What is Cluster Analysis ?

The process by which objects are classified into a number of groups so that they are as much dissimilar as possible from one group to another group, but as much similar as possible within each group.

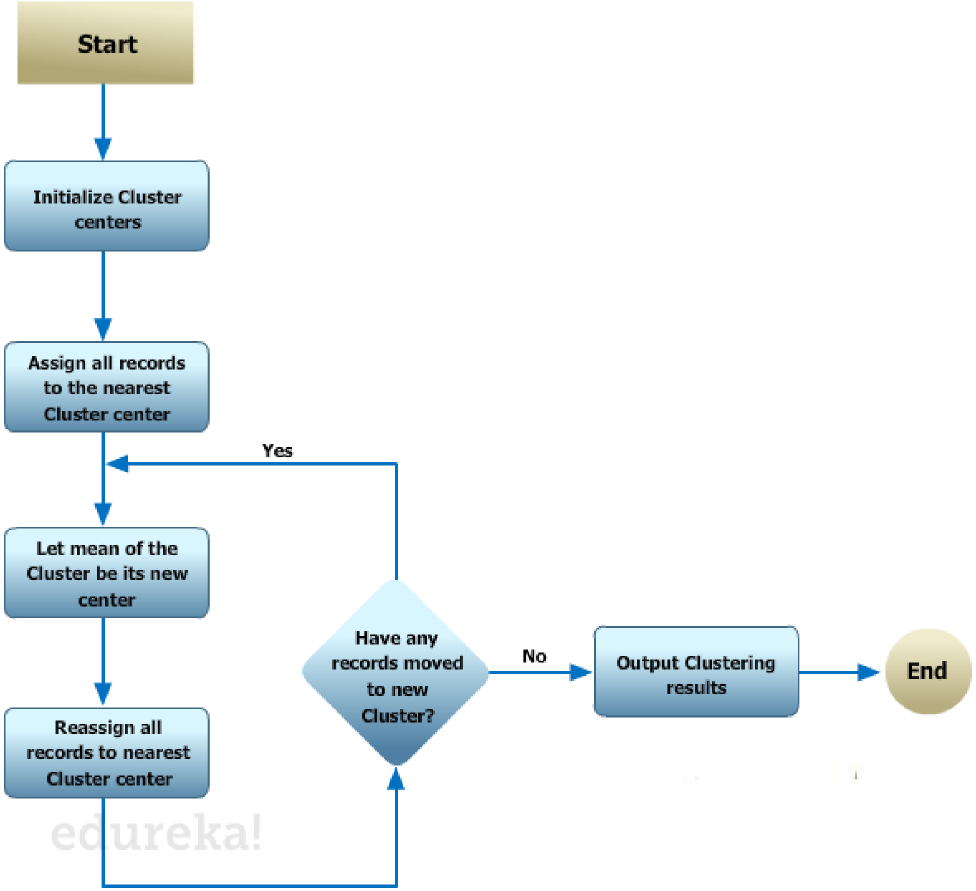
In other words Cluster analysis means dividing the whole population into groups which are distinct between themselves but internally similar.



→ The objects in group 1 should be as similar as possible. But there should be much difference between an object in group 1 and group 2.

→ The attributes of the objects are allowed to determine which objects should be grouped together.

Process Flow of K-means


Start

Initialize Cluster centers

Assign all records to the nearest Cluster center

Let mean of the Cluster be its new center

Reassign all records to nearest Cluster center

Have any records moved to new Cluster?

Yes

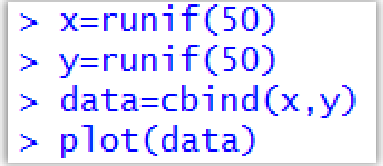
No

Output Clustering results

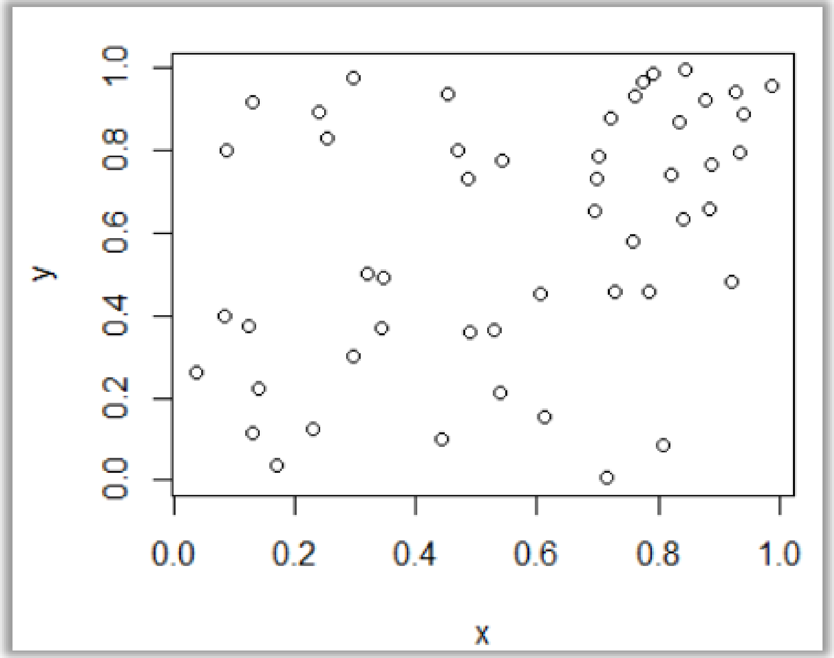
End

K-Means Clustering

Let us take a set of random data points as :



```
> x=runif(50)
> y=runif(50)
> data=cbind(x,y)
> plot(data)
```



Now let us find what are the clusters in the data.

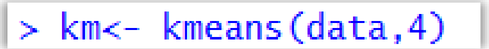
Syntax: Kmeans (x, centers)

x - Data
centers - Number of clusters

The centers can be estimated if good knowledge about the data is present with us.

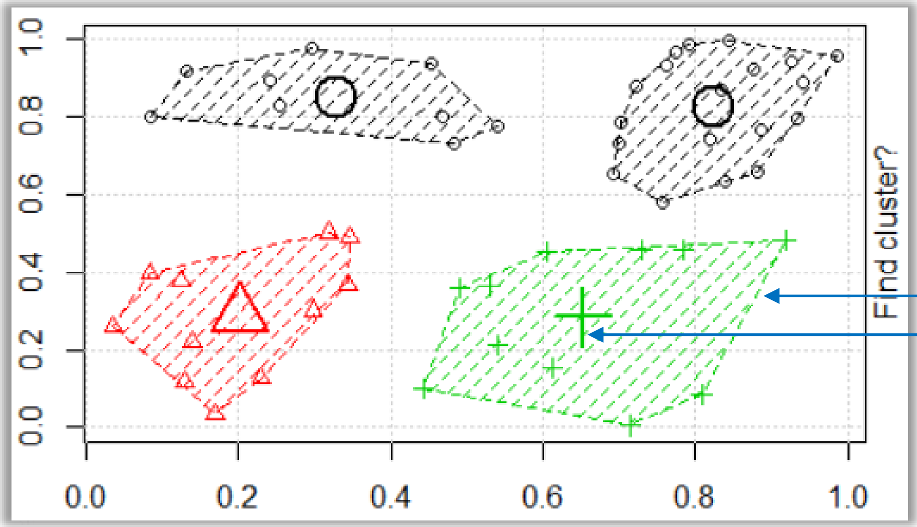
K-Means Clustering

For the data given, lets simply take centers=4.



```
> km<- kmeans(data,4)
```

The clusters thus created can be seen in the plot. The four cluster centres can be observed as well.



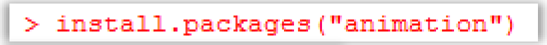
cluster

cluster center

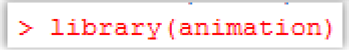
Let us now understand how these clusters were created.

K-Means Clustering

In order to plot the 'kmeans' graph you have to install 'animation' package in R, which lets you create a dynamic 'k-means clustered' plot of points from the data given.

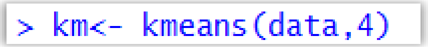


```
> install.packages("animation")
```

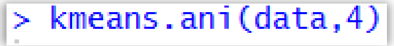


```
> library(animation)
```

'animation': A gallery of animations in statistics and utilities to create animations in R.



Now, instead of giving `> km<- kmeans(data,4)` command, alter the k-means functions as:



```
> kmeans.ani(data,4)
```

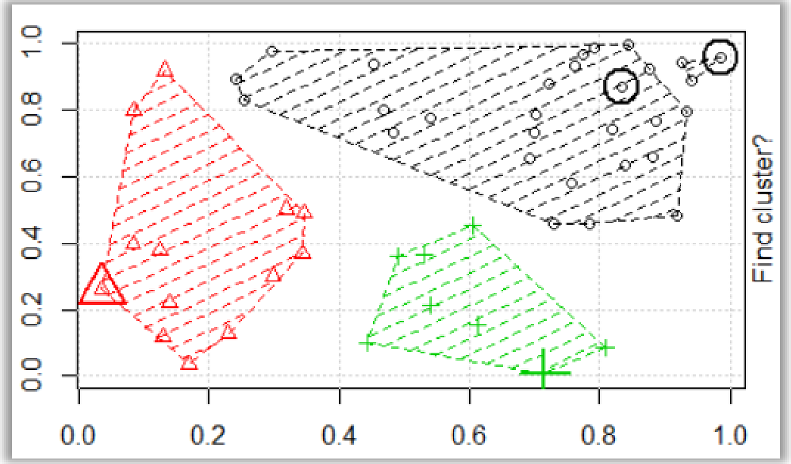
Let us see in the next few slides how the dynamic plot looks like.

K-Means Clustering

Step #1 :
Any random 4 points are assumed to be cluster-centers.

Step #2 :
Distance of the individual data points are calculated from all the 4 cluster-centers and the one which is closest, the data point is assigned to that cluster.

The data point is thus colored in the color of the cluster-center to which it is closely located. This process is followed for all the data points, until all the data points are assigned to a particular cluster.



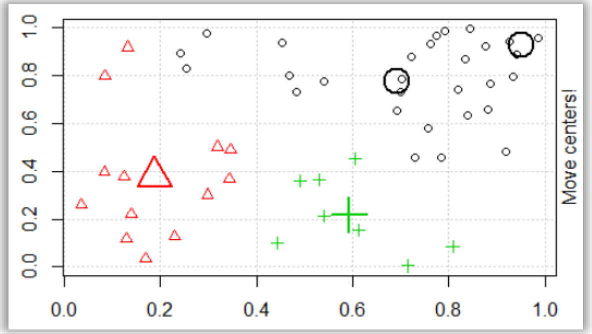
K-Means Clustering

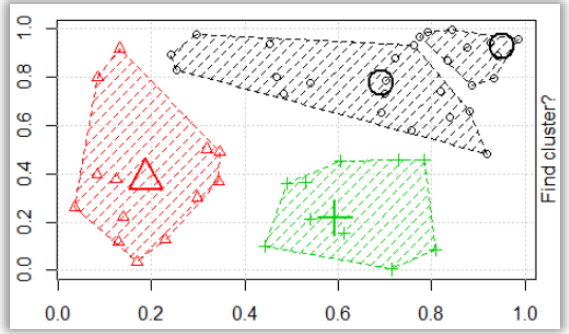
Step #3 :

Now for each cluster, the centroid is calculated and it takes the position of the 'cluster center'.

Step #4 :

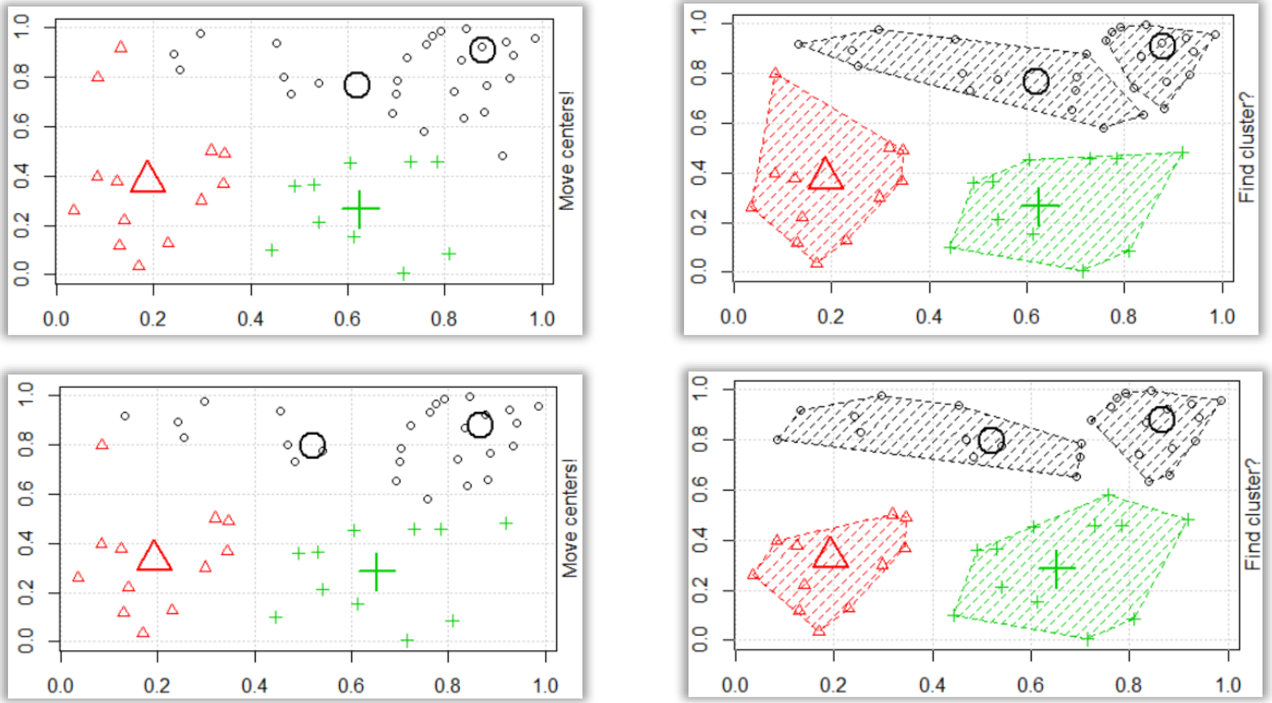
Step #2 and Step #3 are executed repeatedly until the cluster centers cease to move.





1st Iteration

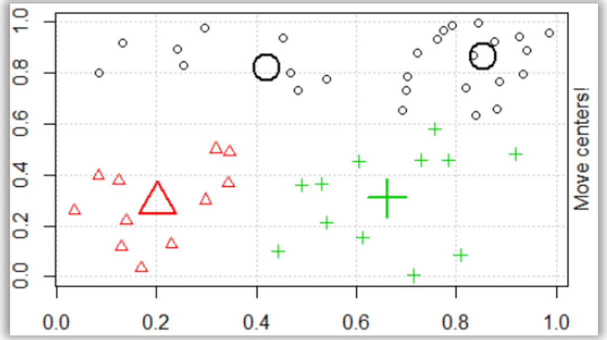
K-Means Clustering

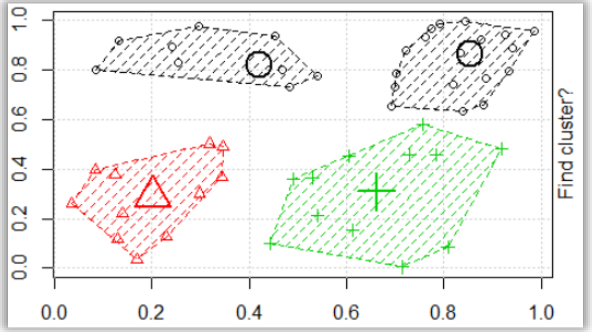


2nd Iteration

3rd Iteration

K-Means Clustering





4th Iteration

Thus after 4 iterations, we receive the final clusters created.

K-Means Clustering

Problem:

How can the cluster quality be measured?

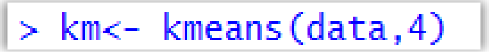
Solution:

You have to observe the following cluster matrix to decide whether the cluster created is good or bad.

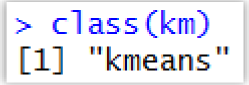
→ Cluster
→ Centers
→ Totss
→ Withinss
→ Tot.withinss
→ Betweenss

K-Means Clustering

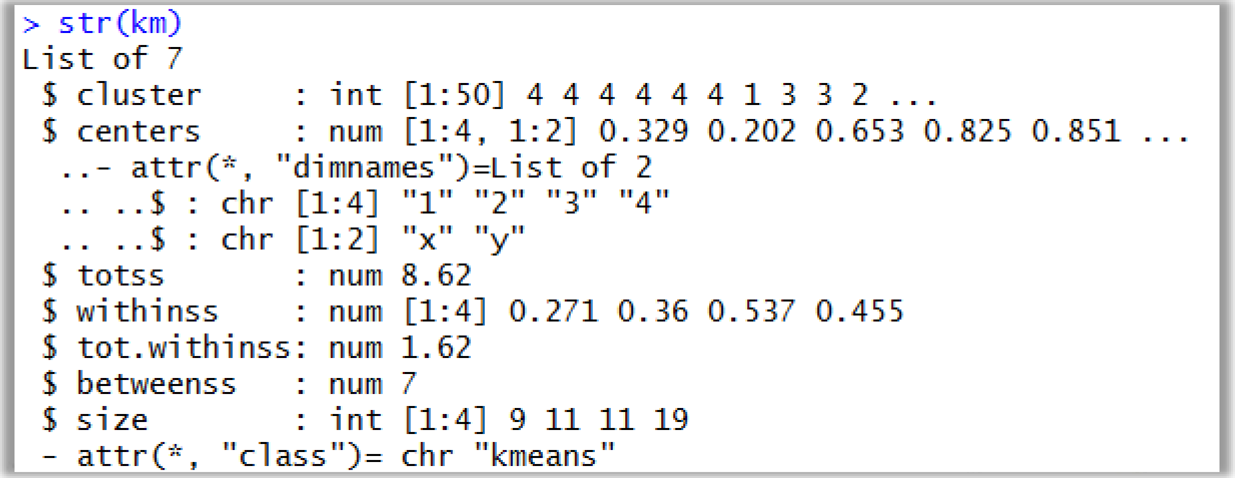
Let us see what is the class and structure of ‘km’:



```
> km<- kmeans(data,4)
```



```
> class(km)
[1] "kmeans"
```



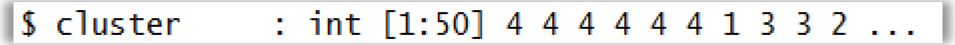
```
> str(km)
List of 7
 $ cluster     : int [1:50] 4 4 4 4 4 4 1 3 3 2 ...
 $ centers     : num [1:4, 1:2] 0.329 0.202 0.653 0.825 0.851 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:4] "1" "2" "3" "4"
  .. ..$ : chr [1:2] "x" "y"
 $ totss       : num 8.62
 $ withinss    : num [1:4] 0.271 0.36 0.537 0.455
 $ tot.withinss: num 1.62
 $ betweenss   : num 7
 $ size        : int [1:4] 9 11 11 19
 - attr(*, "class")= chr "kmeans"
```

Structure of the k-means object can tell us how the k-means has performed.

K-Means Clustering

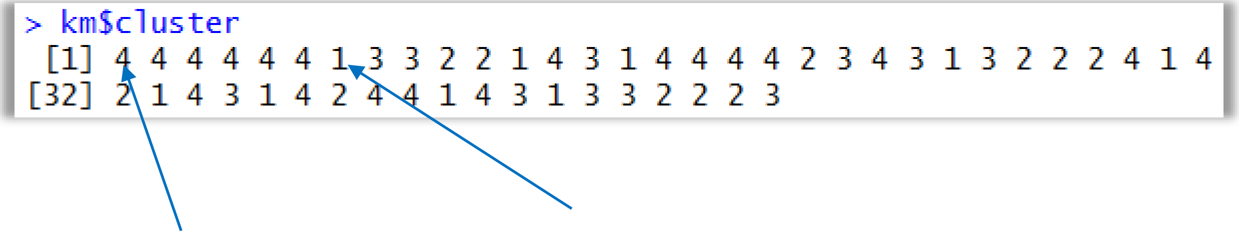
Here, structure comprises of 7 characteristics :

Cluster : It tells us to which cluster, the individual data points belong to.



```
$ cluster      : int [1:50] 4 4 4 4 4 4 1 3 3 2 ...
```

i.e. 1st datapoint : belongs to cluster number 4
      7th datapoint : belongs to cluster number 1, etc.



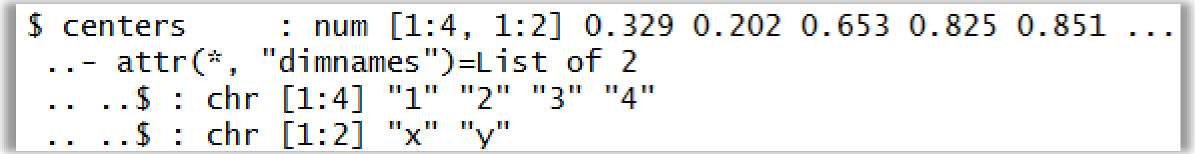
```
> km$cluster
 [1] 4 4 4 4 4 4 1 3 3 2 2 1 4 3 1 4 4 4 4 2 3 4 3 1 3 2 2 2 4 1 4
[32] 2 1 4 3 1 4 2 4 4 1 4 3 1 3 3 2 2 2 3
```

1st Datapoint

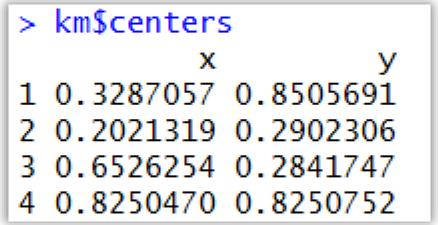
7th Datapoint

K-Means Clustering

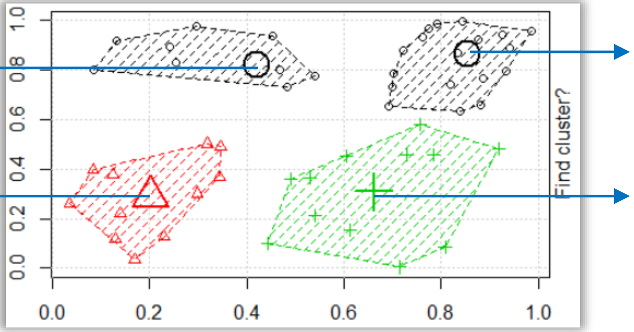
Centers : Contains the values of the four centers computed.



```
$ centers     : num [1:4, 1:2] 0.329 0.202 0.653 0.825 0.851 ...
  ..- attr(*, "dimnames")=List of 2
  .. ..$ : chr [1:4] "1" "2" "3" "4"
  .. ..$ : chr [1:2] "x" "y"
```



```
> km$centers
          x         y
1 0.3287057 0.8505691
2 0.2021319 0.2902306
3 0.6526254 0.2841747
4 0.8250470 0.8250752
```



1st cluster center

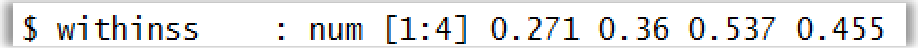
2nd cluster center

4th cluster center

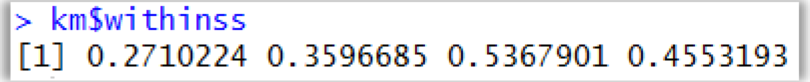
3rd cluster center

K-Means Clustering

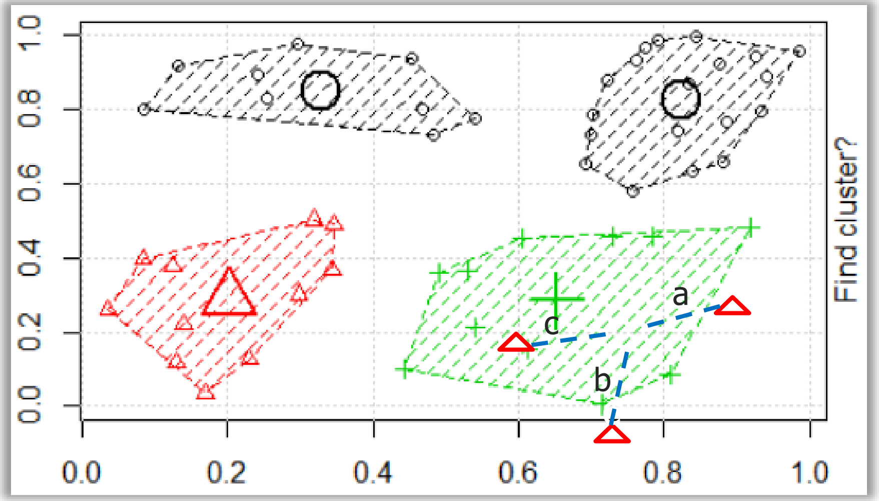
withinss : It stands for 'Within Sum of Squares' and is the total of the squared distance between a point and its cluster's centre, across all points in the cluster. It captures the intra-cluster variability.



```
$ withinss     : num [1:4] 0.271 0.36 0.537 0.455
```



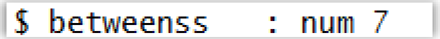
```
> km$withinss
[1] 0.2710224 0.3596685 0.5367901 0.4553193
```



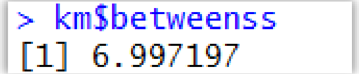
withinss = a+b+c+d

K-Means Clustering

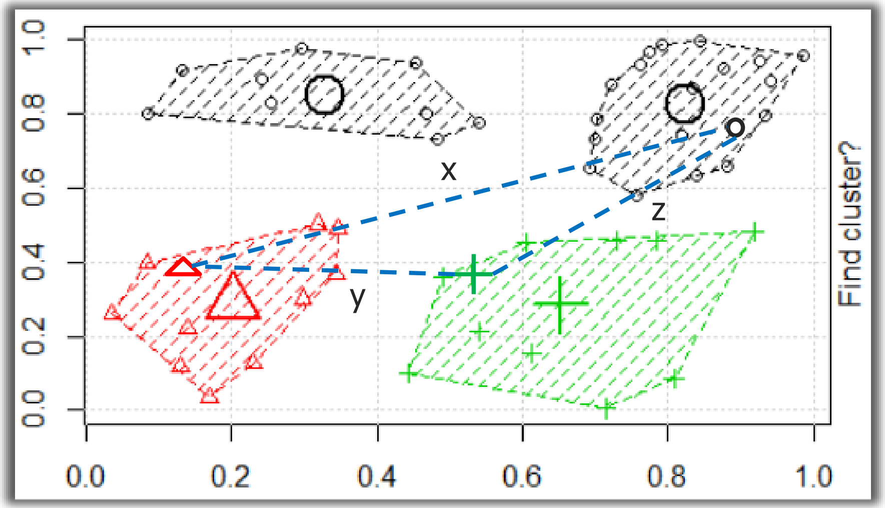
betweenss : It stands for 'Between Sum of Squares' (i.e. totss *less* tot.withinss) and it gives a sense of Inter-cluster variability



```
$ betweenss     : num 7
```



```
> km$betweenss
[1] 6.997197
```



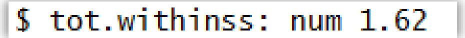
betweenss = x+y+z

K-Means Clustering

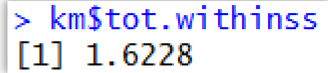
Other structure attributes are:

tot.withinss : It stands for 'Total of the Within Sum of Squares'. If the within-sum-of-squares across all clusters were to be added, we would get tot.withinss

tot.withinss = sum(withinss)



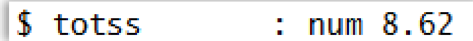
```
$ tot.withinss: num 1.62
```



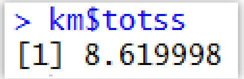
```
> km$tot.withinss
[1] 1.6228
```

totss : It stands for 'Total Sum of Squares'. It is the total of the squared distance between a point and the centre for the entire data. In other words, think of the entire data to be a single cluster and compute its withinss. Therefore,

totss = tot.withinss + betweenss



```
$ totss       : num 8.62
```



```
> km$totss
[1] 8.619998
```

Class activity

Create a cluster with k=1 and observe the withinss and betweenss values.

The value should be :

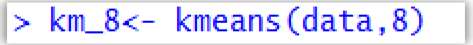
withinss = 0
betweenss = 0 (close to 0)

K-Means Clustering

Ideally you want a clustering that has the properties of internal cohesion and external separation, i.e. the BSS/TSS ratio should approach 1.

When k=4, BSS:TSS = 0.811

Let us increase the value of K to k=8.
When k=8, BSS:TSS = 0.89



```
> km_8<- kmeans(data,8)
```

As number of points are the same and cluster number have increased, it can be seen here that betweenss is large and withinss is small.

If we keep on increasing the k value, the BSS/TSS ratio approaches 1, but this is not the ideal way to cluster.
Let us see how to 'determine the optimal number of clusters'.

How to find optimal solution?

Idea 1: Careful about where we start

→ Choose first center at random
→ Choose second center that is far away from the first center
→ Choose jth center as far away as possible from the closest of centers 1 through (j-1)

Idea 2: Do many runs of K-means, each with different random starting point

Determine Optimal Number of Clusters

→ Determining the number of clusters in a data set i.e., k in the k-means algorithm

→ The correct choice of k is often ambiguous, which depends on the shape and scale of the distribution of points in a data set.

→ In addition, increasing k without penalty will always reduce the amount of error in the resultant clustering

→ The optimal choice of k will strike a balance between maximum compression of the data, and maximum accuracy by assigning each data point to its own cluster.

Determine Optimal Number of Clusters

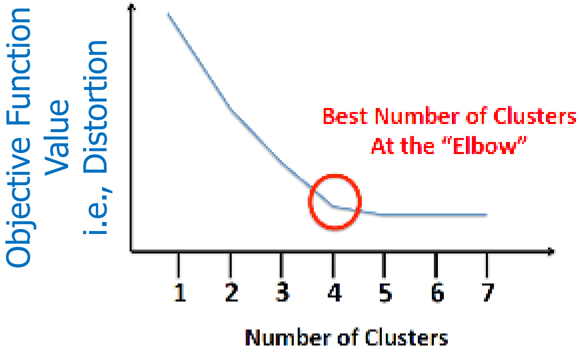
Rule of thumb :

One simple rule of thumb sets the number to with n as the number of objects (data points).

$$k \approx \sqrt{n/2}$$

The Elbow Method :

Another method looks at the percentage of variance explained as a function of the number of clusters:
You should choose a number of clusters so that adding another cluster doesn't give much better modeling of the data. More precisely, if you graph the percentage of variance explained by the clusters against the number of clusters, the first clusters will add much information (explain a lot of variance), but at some point the marginal gain will drop, giving an angle in the graph. The number of clusters are chosen at this point, hence the "elbow criterion". This "elbow" cannot always be unambiguously identified.



QUESTIONS

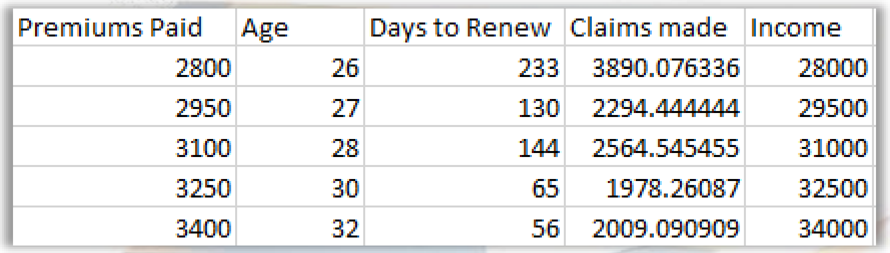
Assignment

Analyze the information given in the following ‘Insurance Policy dataset’ to create clusters of persons falling in the same type

The description of the attributes in the dataset are as follows:

→ Premium Paid – Amount paid by the person
→ Age – Indicates the age of the person
→ Days to renew – Days remaining to renew the policy
→ Claims made – Indicates the claims already made by the person
→ Income – Net income of the person

The dataset for the ‘Insurance Policy’ looks like this :



Premiums Paid	Age	Days to Renew	Claims made	Income
2800	26	233	3890.076336	28000
2950	27	130	2294.444444	29500
3100	28	144	2564.545455	31000
3250	30	65	1978.26087	32500
3400	32	56	2009.090909	34000

Agenda for Next Class

In the next class, you will:

→ Implement data mining on Twitter data

→ Understand Association Rule Mining

→ Understand Collaborative Filtering

Survey

Your feedback is important to us, be it a compliment, a suggestion or a complaint. It helps us to make the course better!

Please spare few minutes to take the survey after the webinar.

Thank you!