You will start by learning what “clustering” is. It is an unsupervised learning technique, where you try to find patterns based on similarities in the data. Then, you will be introduced to a case study that shows the applicability of clustering in the industry.

You will learn the two most commonly used types of clustering algorithms - **K-Means Clustering**and **Hierarchical Clustering**, as well as their application in R. Then, you will also look at what segmentation is and how it is different from clustering.

**Understanding Clustering**

In the previous modules, you saw various supervised machine learning algorithms. Supervised machine learning algorithms make use of labelled data to make predictions.

For example, an email will be classified as spam or ham, or a bank’s customer will be predicted as ‘good’ or ‘bad’. You have a target variable Y which needs to be predicted.

On the other hand, in unsupervised learning, you are not interested in prediction because you do not have a target or outcome variable. The objective is to discover interesting patterns in the data, e.g. are there any subgroups or ‘clusters’ among the bank’s customers?

In the next segment, you will be introduced to a real life application of clustering — grouping customers of an online store into different clusters and making a separate targeted marketing strategy for each group. We will be using this example throughout the module.



**A baby is given some toys to play. These toys consist of various animals, vehicles and houses, but the baby is unaware of these categories. The baby chooses different toys and starts making different groups with the toys based on what he feels are similar toys.**

**Feedback :***Since the baby is classifying objects based on its features and has no idea of any pre existing classification, it is a clustering task.*

We have learnt about three different types of machine learning techniques - regression, classification and clustering. Which of the following techniques does not require bifurcation of data points into dependent and independent variables?



**Clustering**

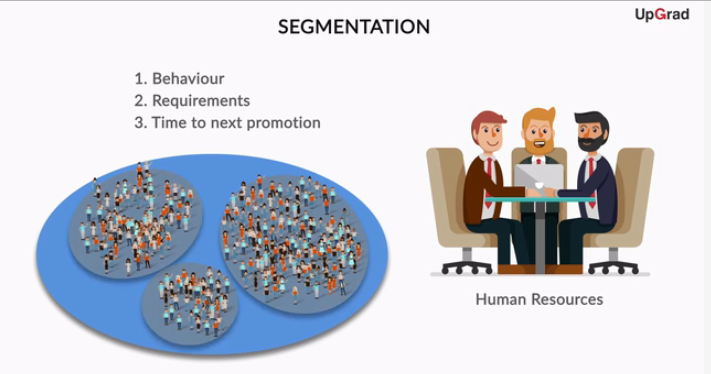
**Feedback :***In clustering, we group the data points into different categories based on the given set of attributes. There are no dependent and independent variables*

Customer segmentation for targeted marketing is one of the most vital applications of the clustering algorithm. Here, as a manager of the online store, you would want to group the customers into different clusters, so that you can make a customised marketing campaign for each of the group. You do not have any label in mind, such as good customer or bad customer. You want to just look at patterns in customer data and then try and find segments. This is where clustering techniques can help you with segmenting the customers. Clustering techniques use the raw data to form clusters based on common factors among various data points. This is exactly what will also be done in segmentation, where various people or products will be grouped together on the basis of similarities and differences between them.

As a manager, you would have to decide what the important business criteria are on which you would want to segregate the customers. So, you would need a method or an algorithm that itself decides which customers to group together based on this criteria.

Sounds interesting? Well, that is the beauty of unsupervised learning, especially clustering. But before we conclude this introductory session, it would be best to get an industry perspective on the application of clustering in the world of analytics.



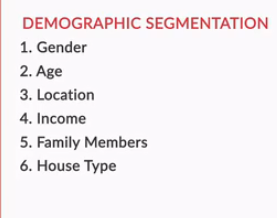


You saw that, for successful segmentation, the segments formed must be stable. This means that the same person should not fall under different segments upon segmenting the data on the same criteria. You also saw that segments should have **intra-segment homogeneity** and **inter-segment heterogeneity**. You will see in later sessions how this can be defined mathematically.

Now you will see what types of market segmentations are commonly used.







You saw that mainly 3 types of segmentation are used for customer segmentation:

* **Behavioural segmentation**: Segmentation is based on the actual patterns displayed by the consumer
* **Attitudinal segmentation**: Segmentation is based on the beliefs or the intents of people, which may not translate into similar action
* **Demographic segmentation**: Segmentation is based on the person’s profile and uses information such as age, gender, residence locality, income, etc.
* **Type of Segmentation**
* A telecom company classifies its prepaid mobile customers into three types mainly based on the number of times they recharge per month. This is a type of:
* Top of Form
* 
* **Behavioral segmentation**
* **Feedback :***Recharge is a behaviour which can be observed, opposed to attitude which resides in the mindset of the customer.*
* Bottom of Form
* An international foods and beverages company wants to look at what products it should launch in India. For that, it has first tried to segment the market. It is known that people living in the same area and having similar salaries will have similar eating habits. Then which of the following can be segmented in 1 group? (All options in 1 bracket are 1 segment)
* A. High Earning Individual from Bengaluru
* B. Low Earning Individual from Rural Uttar Pradesh
* C. Mid Earning Individual from Mumbai
* D. High Earning Individual from Hyderabad
* 
* **(A,D) - (B) - (C)**
* **Feedback :***You want to ensure that the people in One segment are very similar and the people in different segments are very dissimilar. So Option C is good segment to make*

**You learnt that clustering is commonly used for segmenting customers. Can you think of some features on which you would want to segment the customers of an online store? Go back and check the data ‘online retail’ if needed.**

*Some of the features can be how much the people spend on buying goods from the store. Another one can be the number of times they bought goods in the last one year. Finally, you can also look at the time they last bought a good from the store.*

You are an analyst at a global laptop manufacturer and are given the task of deciding whether the company should enter the Indian Market. You try to estimate the market size by first breaking the market by different types of people who use a laptop such as students, working professionals and their paying capacity to get an estimate of the total market size and the characteristics of each segment. In essence, you are doing:



**Demographic Segmentation**

**Feedback :***You are doing a demographic segmentation, since you are looking at the income and the profession of people. Notice how this is much simpler than finding data about actual laptop purchasing history of customers and then trying to estimate the market size based on that.*

* The steps in the K-Means algorithm
* How to graphically visualise the steps of K-Means algorithm
* Practical considerations while using the K-Means algorithm

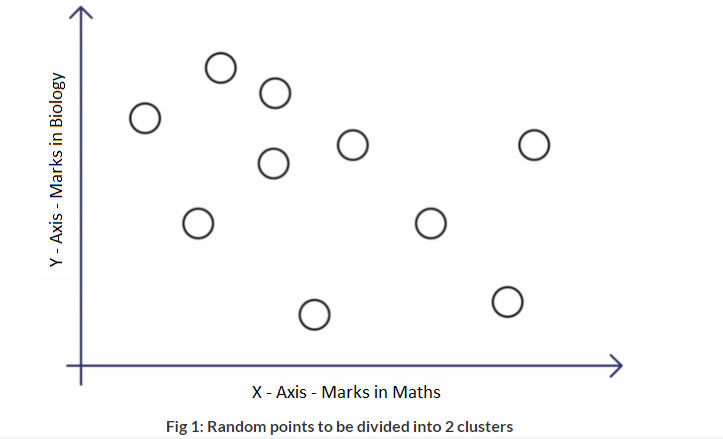
# Steps of the Algorithm

Let’s go through the K-Means algorithm using a very simple example. Let’s consider a set of 10 points on a plane and try to group these points into, say, 2 clusters. So let’s see how the K-Means algorithm achieves this goal.

Before moving ahead, think about the following problem. Let’s say you have the data of 10 students and their marks in Biology and Math (as shown in the plot below). You want to divide them into two clusters so that you can see what kind of students are there in the class.

The y-axis shows the marks in Biology, and the x-axis shows the marks in Math.

Imagine two clusters dividing this data — one red and the other yellow. How many points would each cluster have?



**Centroid**

The K-Means algorithm uses the concept of the centroid to create K clusters. Before you move ahead, it will be useful to recall the [concept of the centroid](https://en.wikipedia.org/wiki/Centroid).

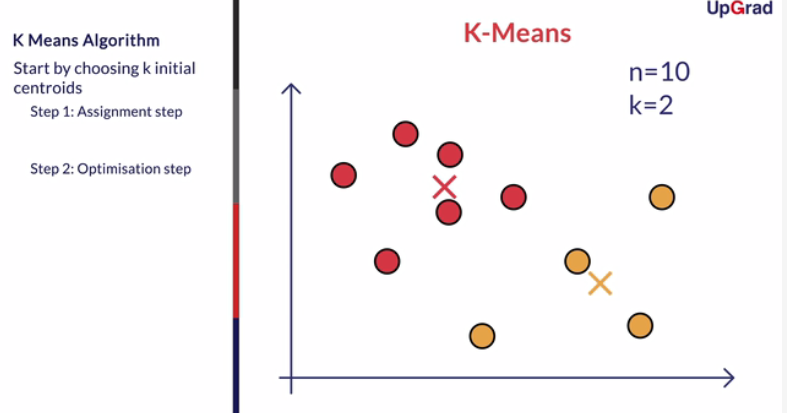
In simple terms, a centroid of n points on an x-y plane is another point having its own x and y coordinates and is often referred to as the geometric centre of the n points.

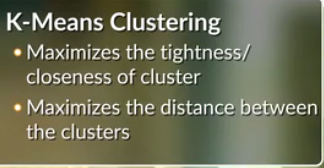
For example, consider three points having coordinates (x1, y1), (x2, y2) and (x3, y3). The centroid of these three points is the average of the x and y coordinates of the three points, i.e.

(x1 + x2 + x3 / 3, y1 + y2 + y3 / 3).

Similarly, if you have n points, the formula (coordinates) of the centroid will be:

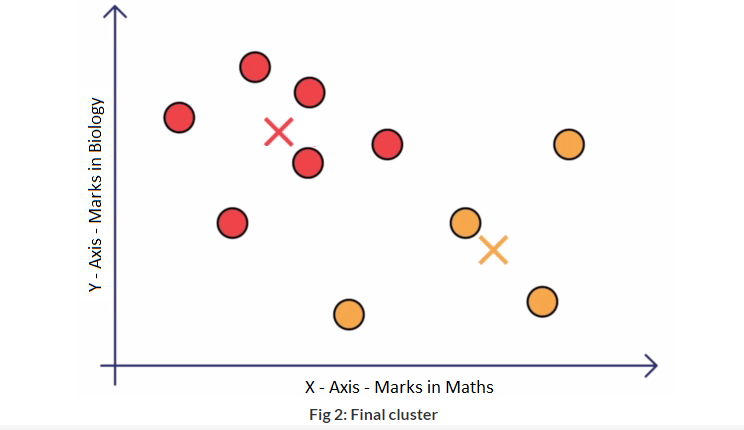
**(x1+x2…..+xn / n, y1+y2…..+yn / n).**





Each time the clusters are made, the centroid is updated. The updated centroid is the centre of all the points which fall in the cluster associated with the centroid. This process continues till the centroid no longer changes, i.e. the solution converges.

Thus, you can see that the K-means algorithm is a clustering algorithm that takes N data points and groups them into K clusters. In this example, we had N =10 points and we used the K-means algorithm to group these 10 points into K = 2 clusters.



**K-Means algorithm**

Find the distance of each of the 10 points from the two cluster centers (in column H & I) and fill the cells S6:T16. Select the distance formula to be used to find the distance of a point (xi,yi) from a centre (Xi,Yi).



**SQRT(((Xi-xi)^2) + (Yi-yi)^2))**

**Feedback :***This is the formula for the Euclidean distance. You can see that once the cells S6:T16 are filled with the required distances, the points are assigned to one of the clusters (marked in column E) based on the minimum distance.*

In the next step of k-means clustering, you need to find the new cluster centers (H22:I23). Select the correct method used to find the new cluster centers.



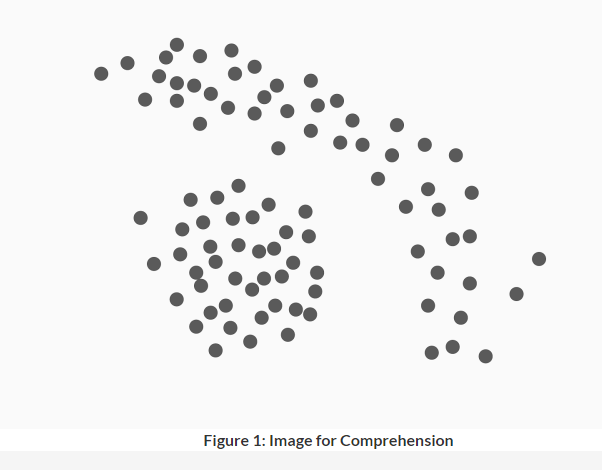
**Calculate the centroid of the points assigned to a particular cluster in the previous step**

**Feedback :***The way is new center is calculated in K-Mean clustering is the mean of all the data points belonging to that cluster.Notice that the new centers are already filled in the excel sheet.*

# Visualising the K Means Algorithm

Let’s see the K-Means algorithm in action using a visualisation tool. This tool can be found on [naftaliharris.com](http://www.naftaliharris.com/blog/visualizing-k-means-clustering/). You can go to this link after watching the video below and play around with the different options available to get an intuitive feel of the K-Means algorithm.

Upon trying the different options, you may have noticed that the final clusters that you obtain vary depending on many factors, such as choice of the initial cluster centres and the value of K, i.e. the number of clusters that you want. You will understand these factors and other practical considerations while using the K-means algorithm in more detail in the next segment.



Consider the above arrangement of points. How many clusters do you intuitively feel are present. What will happen if you use K-Means clustering here? How do you think this problem can be solved?

Suggested Answer

*Intuitively, it looks like 2 clusters are present. If we use K means, then we will get wrong clusters since the points in the outer ring like structure will not be segmented accurately. A reason for that is K-Means looks for how close the points are to a centroid and this distance or measure of closeness is the “linear distance”. One way to correct this can be to see the distance between all the points and then cluster the closest points.*

# Practical Consideration in K Means Algorithm

Let’s understand some of the factors that can impact the final clusters that you obtain from the K-means algorithm. This would also give you an idea about the issues that you must keep in mind before you start to make clusters to solve your business problem.

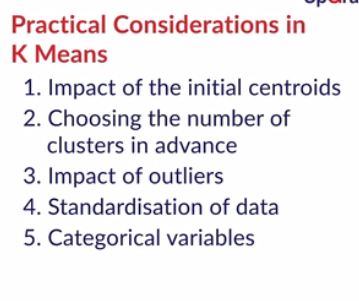
**K-Means algorithm**

Did you take a note of one of the most important underlying principles of k-means algorithm. Select the option which describes the principle correctly.



**Maximise the inter-cluster distance and minimise the intra-cluster distance**

**Feedback :***K means clustering tries to minimise the intra cluster distance and maximise the inter cluster distance.*



Thus, the major practical considerations involved in K-Means clustering are:

* The number of clusters that you want to divide your data points into, i.e. the value of K has to be pre-determined.
* The choice of the initial cluster centres can have an impact on the final cluster formation.
* The clustering process is very sensitive to the presence of outliers in the data.
* Since the distance metric used in the clustering process is the Euclidean distance, you need to bring all your attributes on the same scale. This can be achieved through standardisation.
* The K-Means algorithm does not work with categorical data.
* The process may not converge in the given number of iterations. You should always check for convergence.
* **K-Means algorithm**
* If we are worried about K-means getting stuck in bad local optima, one way to solve this problem is if we try using multiple random initializations. Is this true or false? You can read about local optimum and global optimum [here](https://en.wikipedia.org/wiki/Local_optimum)
* 
* **True**
* **Feedback :***Since each run of K-means is independent, multiple runs can find different local optima, and this can help in choosing the global optimum value.*

We covered a lot in this session. We started with understanding the K-Means intuitively by grouping the 10 random points in 2 clusters.

The algorithm begins with choosing K random cluster centres.

Then the 2 steps of **Assignment and Optimisation** continue iteratively till the clusters stop updating. This gives you the most optimal clusters — the clusters with minimum intra-cluster distance and maximum inter-cluster distance.

We covered a lot in this session. We started with understanding the K-Means intuitively by grouping the 10 random points in 2 clusters.

The algorithm begins with choosing K random cluster centres.

Then the 2 steps of **Assignment and Optimisation** continue iteratively till the clusters stop updating. This gives you the most optimal clusters — the clusters with minimum intra-cluster distance and maximum inter-cluster distance.

You also saw the different practical issues that need to be considered while employing clustering to your data set. You need to choose **how many clusters**you want to group your data points into. Secondly, the K-means algorithm is **non-deterministic**. This means that the final outcome of clustering can be different each time the algorithm is run even on the same data set. This is because, as you saw, the final cluster that you get can vary by the choice of the initial cluster centres.

You also saw that the **outliers** have an impact on the clusters and thus outlier-infested data may not give you the most optimal clusters. Similarly, since the most common measure of the distance is the Euclidean distance, you would need to bring all the attributes into the same scale using **standardisation**.

You also saw that you cannot use categorical data for the K-Means algorithm. There are other customised algorithms for such categorical data.

**K-Means algorithm**

Arrange the steps of k-means algorithm in the order in which they occur:

1. Randomly selecting the cluster centroids
2. Updating the cluster centroids iteratively
3. Assigning the cluster points to their nearest center
4. 
5. **1-3-2**
6. **Feedback :***First the cluster centers are pre-decided. Then all the points are assigned to their nearest cluster center and then the center is recalculated as the mean of all the points which fall in that cluster. Then the clustering is repeated with the new centers and the centers are updated according to the new cluster points.*
7. **K-Means algorithm**
8. Consider three cluster centres A(2,3), B(4,5) and C(6,2). A point (1,2) is to be assigned to one of these clusters. According to k-means clustering concepts and using euclidean distance as the measure of closeness, which cluster should it be assigned to?
9. Top of Form
10. 
11. **A**
12. **Feedback :***According to k-means algorithm, the point should be assigned to the centre with the minimum distance from the point. The distances for A, B, C are sqrt(2), sqrt(18) and sqrt(25)*
13. **Correct**
14. 
15. B

Which of the following options are prerequisites for k-means algorithm:

A) initial centers should be very close to each other

B) Choice of number of clusters

C) Choice of initial centroids 

**B & C**

**Feedback :***Note that the k-means algorithm does not require the initial centers to be far apart.*

Executing K-Means in R

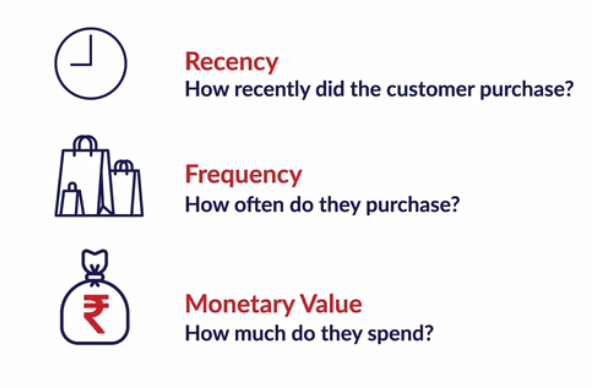
You will learn about

* Data preparation
* How to make the clusters
* Decide the optimal number of clusters
* How to interpret the results

# Data Preparation

In Marketing: Online retail shop example





You saw the steps to performing the data preparation. Now let’s do the same in R. **You can download the data set for the case study from this**[**link**](https://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx)**here. Convert the .xls file into .csv format before proceeding.**

Let’s begin the analysis by importing the data set into R.

Online.Retail <- read.csv("Online Retail.csv", stringsAsFactors=FALSE)

Let's start with some preliminary data cleaning. Now, as you can notice, the data set is at the granularity of order level. So, it doesn’t make much sense to do missing value imputation because it would be very difficult to predict the individual missing details of individual orders. Hence, you have to remove the entries with the missing values.

order\_wise <- na.omit(Online.Retail)

But, if you remember, our main objective is to cluster the customers of the online store. So, you need to transform this order-wise data into customer-wise data by finding key attributes that best characterises a customer. This is achieved through RFM analysis.

**RFM analysis**

In RFM analysis, you look at the recency, frequency and the monetary scores of all the customers for segmentation.

* **Recency:** It measures how recently you visited the store or made a purchase
* **Frequency:** It measures the frequency of the transactions the customers made
* **Monetary:** It measures how much the customer spent on purchases he/she made

So, your target is to compute the RFM numbers for each customer, which effectively means that the granularity level of your data set will change from Invoice number to the CustomerID. Thus, you will have one unique row corresponding to each customer.

Let’s start with creating customer-wise data. We begin with the computation of M of the RFM, that is the total monetary value of the purchases made by each customer.

Create a vector named Amount, which creates the total monetary value of each order, and append the column to your data set.

Amount <- order\_wise$Quantity \* order\_wise$UnitPrice

order\_wise <- cbind(order\_wise,Amount)

Now, sort the data set in order of CustomerID. Next, create a new vector — monetary — which gives the aggregated purchase amount for each customer.

order\_wise <- order\_wise[order(order\_wise$CustomerID),]

monetary <- aggregate(Amount~CustomerID, order\_wise, sum)

This vector monetary is the M of the RFM framework.

Next, let’s compute the frequency of purchase for each customer, i.e. the F of the RFM framework. For this, you will count the number of unique Invoice Numbers for each Customer ID. There are many ways to do this. One way is to convert the format of CustomerID column to “factors”. Now, tabulating the data set using the “table” function will give the count of each factor level, which is the number of times the purchase has been made by each customer. This is the “Frequency” corresponding to each customer.

frequency <- order\_wise[,c(7,1)]

temp<-table(as.factor(frequency$CustomerID))

temp<-data.frame(temp)

colnames(temp)[1]<-c("CustomerID")

Finally, merge this data frame with the “Frequency” of each customer into your earlier data set containing the “Monetary” value.

RFM <-merge(monetary,temp,by="CustomerID")

Thus, the data frame RFM contains both the monetary and the frequency attributes corresponding to each customer IDs. Now, you have to turn your attention towards the computation of the recency, i.e. for how long a customer has not visited the online store.

Begin by extracting the Customer ID and Invoice Date from the data and converting the Invoice Date to “date” format. **Please look at the format of InvoiceDate (it depends on the format you have imported Excel in) and change the code given below according to that (dd-mm-yy or mm/dd/yy etc).**

recency <- order\_wise[,c(7,5)]

recency$InvoiceDate<-as.Date(recency$InvoiceDate,"%d-%m-%Y %H:%M")

Now, find the latest “Invoice Date” which forms the reference point for the calculation of the “Recency” of each customer. For each order corresponding to each customer, you find the difference from the latest “Invoice Date” and then find the minimum “Recency” value for each customer.

maximum<-max(recency$InvoiceDate)

maximum<-maximum+1

maximum$diff <-maximum-recency$InvoiceDate

recency$diff<-maximum$diff

recency<-aggregate(recency$diff,by=list(recency$CustomerID),FUN="min")

colnames(recency)[1]<- "CustomerID"

colnames(recency)[2]<- "Recency"

Now, the data frame recency contains the recency for each customer. Let’s merge it to the RFM data set and change the format to the required form.

RFM <- merge(RFM, recency, by = ("CustomerID"))

RFM$Recency <- as.numeric(RFM$Recency)

Thus, you have obtained the RFM data corresponding to each customer. These 3 attributes will form the basis, depending on which the customers will be segregated into different clusters.

However, your data preparation is still not complete. You have already seen previously how the clustering process can be impacted due to the presence of outliers. So, let’s treat the data set for outliers. One way to do it is by eliminating all the data points which fall outside the whiskers of the box plot plotted by you.

box <- boxplot.stats(RFM$Amount)

out <- box$out

RFM1 <- RFM[ !RFM$Amount %in% out, ]

RFM <- RFM1

box <- boxplot.stats(RFM$Freq)

out <- box$out

RFM1 <- RFM[ !RFM$Freq %in% out, ]

RFM <- RFM1

box <- boxplot.stats(RFM$Recency)

out <- box$out

RFM1 <- RFM[ !RFM$Recency %in% out, ]

RFM <- RFM1

Now, you have done the basic data preparation. Let’s see if any other steps are required before you can make the clusters.

So, let’s get going with the Online Retail data set to solve the customer segmentation problem we talked about. You can download the data set for the case study from this [link](https://archive.ics.uci.edu/ml/machine-learning-databases/00352/Online%20Retail.xlsx) here. Convert the .xls file into .csv format before proceeding.

Let’s start making your very first clusters using K-Means. Prof Dinesh will guide you through the process for creating these clusters.

So, the data preparation is now complete. So, let’s reiterate the steps involved in data preparation:

* Missing value treatment
* Transforming data from Order-level to Customer-level
* Calculation of RFM values
* Outlier treatment
* Standardisation of data

In the next segment, you will begin with the actual implementation of the K-Means algorithm in R.

**Using K-Means Clustering**

Run a summary function before scaling the variables of the RFM dataset and find out the mean of various columns



**R : 105 ,F : 47 ,M : 771**

**Feedback :***Run the summary function on the data. Also remember that if you removed outliers in some other way, then your answer will differ*

What do you think will happen if you run the clustering without scaling the data? *Since the scales of the data are different, more weightage will be given to Monetary value. Data points which have very different monetary values will be classified differently, even though they might actually be very similar in Recency and Frequency*

**K-Means algorithm**

Select the option which correctly describes “betweenss” - output of str(clus) as discussed in the lecture. 

**betweenss = totss - tot.withinss**

**Feedback :***tot.withinss is the total within sum of squares of clusters. betweenss is the inter cluster sum of squares of distance, while totss is the total sum of square of distance.*

As you saw, R has a simple in-built function K - Means() to implement the K-Means algorithm. You also saw that the K - Means() function stores the output of the algorithm in a list of 9 objects.

1. **cluster:**This stores the cluster IDs for each data point
2. **centers:**This gives the location of the cluster centres
3. **totss:**This measures the total spread in the data by calculating the total sum of squares of distance
4. **withinss:**This is a measure of the within-cluster sum of squares, one component per cluster
5. **tot.withinss:**This gives the total within-cluster sum of squares, i.e. sum(withinss)
6. **betweenss:**The inter-cluster sum of squares of the distance, i.e. totss-tot.withinss
7. **size:**The number of points in each cluster
8. **iter:**The number of (outer) iterations
9. **ifault:**integer indicator of a possible algorithm problem – for experts

You can see that R stores the inter-cluster and intra-cluster variance in the form of sum-of-squares distance between the units. These values are, in fact, used to decide the optimal number of clusters that you want to divide your customers into. You will see their application in the next segment

Thus, you saw that there are different ways to decide upon the required number of clusters that you want.

1. The business understanding or the requirement can guide you, or rather constrain you, about the number of clusters that you can create.
2. You can use the elbow method to arrive at a range of the values of K for which you can get optimal clusters, i.e. clusters with the maximum inter-cluster variance and minimum intra-cluster variance. For this, you can use any of the 2 metrics:
   1. R-sq value which is the ratio of (betweenss/totss)
   2. Pseudo F statistic, which is given by (between-cluster-sum-of-squares / (c-1)) / (within-cluster-sum-of-squares / (n-c)) where c is the number of clusters and n is the total number of data points

Once you have found the optimal range of K graphically from the R-sq or Pseudo F plot, use the value of K for which the obtained clustering makes the most business sense.

**Parameters to consider**

Did you get any warning or errors while running the code? If you did, then explore the warnings further. What is the reason and how will you correct it?

Top of Form

Suggested Answer

*The warning is related to the convergence of the solution. As we saw earlier, K-Means is an iterative method and the centers keep getting updated with each iteration. However, it is possible that the solution does not converge in the default number of iterations. You can change the number of maximum iterations using iter.max option in the kmeans() command*

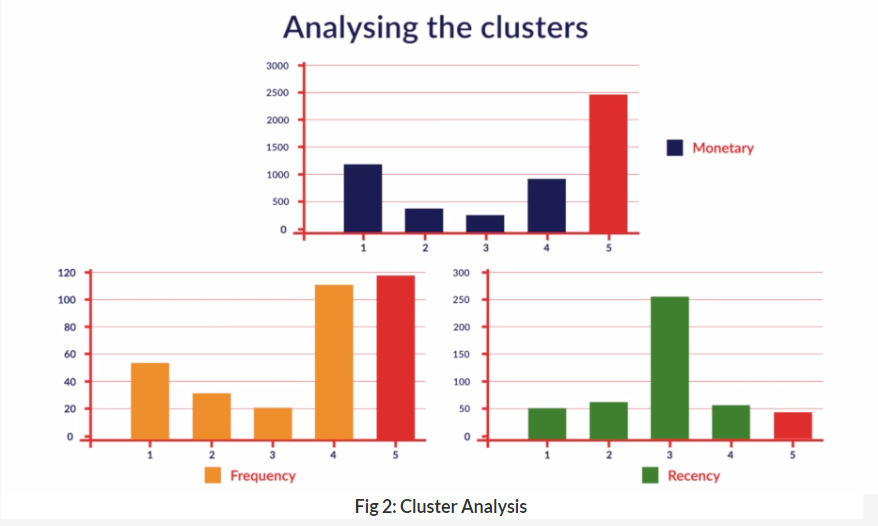
Bottom of Form

Thus, you obtained the ClusterID corresponding to each CustomerID from the kmeans() function. These ClusterIDs were appended to the original RFM data set which had the Recency, Frequency and the Monetary values corresponding to each CustomerID.

Then, using the group\_by() and the summarise() function under the “dplyr” library, you were able to find the characteristics of each cluster in terms of the mean values of RFM. This helped you profile each of the obtained 5 clusters.

Here, each graph has the cluster number on the x axis, whereas the value of Recency, Frequency and Monetary is on the y axis.

You found that cluster 5 was the best customer segment from the store’s point of view. These customers make a purchase for a higher amount, more frequently, and these customers had visited the site recently. Thus, the store may offer them a reward or loyalty points or some privileged status, to keep them attracted and coming back to the store.

.

On the other hand, cluster 3 had the worst customers from the store’s point of view. Thus, the store may decide to focus more on this group. Similarly, in cluster 1, the customers had favourable features in terms of the purchase amount and recency; however, these have low frequency. Thus, if the store can re-design its incentive strategy and entice these customers into making a purchase more frequently, they could turn profitable for the store.

# Let's Have Some Fun

You have learnt about how to make clusters using the K-Means algorithm. Let's use that knowledge to play around with clustering using K-Means.

Given below is a data set on the education status of Indian states.

The data contains state-level information on attributes such as the number of literates, illiterates, the number of literates who are graduate and above, etc.

But there’s a problem — the number of variables is quite large, and after forming the clusters, it may get difficult to describe each cluster’s characteristics.

This is not an uncommon problem. You may have noticed data sets having as many as 100-200 variables. There are techniques which are used to ‘reduce the number of variables while retaining as much information as possible’.

Two most common techniques, also called variable reduction techniques, are factor analysis and principal component analysis.

Lawmakers can cluster the states to find out which states have similar education statistics, and thus assign the budget accordingly. Clustering can also help them figure out the best policies to make for these clusters.

You can download the data set and run the K-Means algorithm on this. You can try to make the clusters on different attributes. You can choose which attributes are most important using Factor Analysis, which you will find in the [optional content](https://learn.upgrad.com/v/course/29/session/5920).

You can see the effect of various elements of the K-Means clustering on the clusters formed. You can zoom in by selecting and double-clicking For the purpose of creating the above visualisation, we have cleaned the data to include only the 2 factors under consideration. You could have chosen any other factors as well. Factors here mean the variables that you will use to build the clustering model. You can download the file below to answer the questions that follow:the map to look at the clusters.

**K - Means in R**

Which parameters do you think are the most important for segmenting the states? How did you decide this?

Suggested Answer

*The parameters used depend on the question at hand. We can choose different age groups and different categories such as graduate or above etc.*

How will you check if the segmenting is good or whether you need to use different factors for segmenting?

Suggested Answer

*One really easy way is to see if the .are logically correct. For example, we can check if states in a similar geography and economic situation are clustered together or not. You can also run hypothesis test to check whether the population of different clusters is significantly different or not, If you look at the data, you can see that some specific customers or some specific states should be grouped together.*

**K - Means in R**

How are the clusters different when we have not scaled compared to clusters formed after scaling?

Top of Form



**Literacy percentage gets higher weightage when there is no scaling**

**Feedback :***Look at the clusters formed with and without scaling. You will see that for the ones formed without scaling, the states with similar literacy rates will fall in the same cluster, even though their graduate percentage differs.*

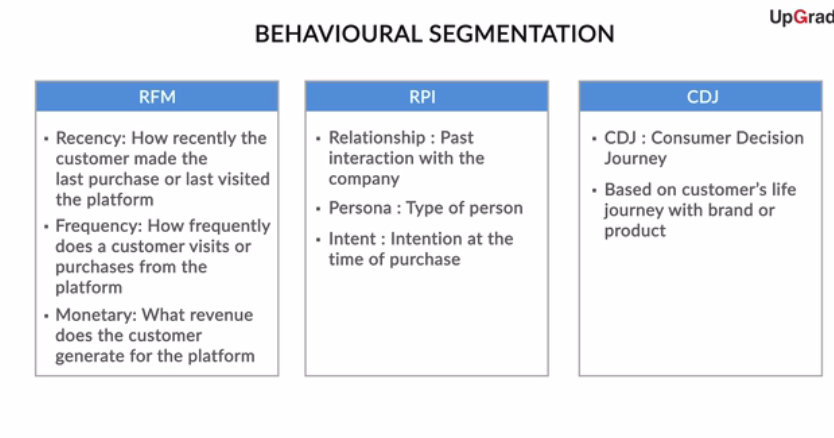
# Other Behavioural Segmentation Types

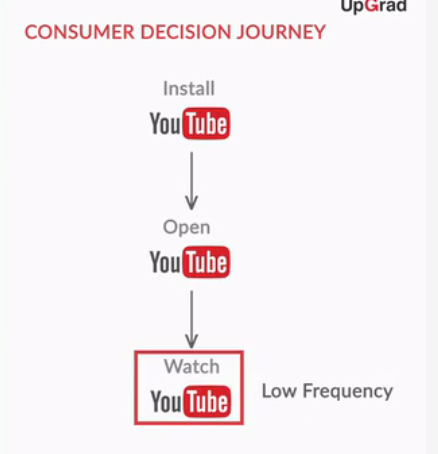
You have seen what RFM segmentation is. Now, you will look at other segmentation types commonly used in the industry.

You looked at RPI segmentation, which looks at what kind of relationship you have had with the person before, what type of person he/she is, and the intent of the person at the time of buying.

You also looked at the CDJ segmentation, which looks at the path that customers take while experiencing your product.

Now, let us turn our attention to another clustering technique — hierarchical clustering — in the next session.





You looked at RPI segmentation, which looks at what kind of relationship you have had with the person before, what type of person he/she is, and the intent of the person at the time of buying.

You also looked at the CDJ segmentation, which looks at the path that customers take while experiencing your product.

Now, let us turn our attention to another clustering technique — hierarchical clustering — in the next session.

# Summary

So what did you learn in this session?

You learnt how to create clusters using the K-means algorithm in R with the analysis of the Online Store data set. We wanted to group the customers of the store into different clusters based on their purchasing habits. The different steps involved were:

* Missing values treatment
* Data transformation
* Outlier treatment
* Data standardisation
* Finding the optimal value of K
* Implementing K Means algorithm
* Analysing the clusters of customers to obtain business insights

Depending on the initial selection of centers, the formed clusters might be different. The value of k has to be decided by the user

**Cricket**

After how many clusters does the R2R2 value reach nearly 0.7?



**4**

**Feedback :***You will have to plot the r-sq plot to see this. Some of the commands used are: cric <- Cricket[,c("Ave","SR")] cric <- as.data.frame(scale(cric, center = TRUE, scale = TRUE)) for (i in 1:10){cric.km <- kmeans(cric, center = i, iter.max = 50, nstart = 50) r\_sq[i] <- cric.km$betweenss/cric.km$totss}*



**SR Tendulkar**

**Feedback :***You can use the following commands to create clusters.and visually see them as well cric.km <- kmeans(cric, center = 4, iter.max = 50, nstart = 50) Cricket <- cbind(Cricket, cric.km$cluster) ggplot(Cricket, aes(x = SR, y = Ave, colour = as.factor(*cric.km$clustercric.km$cluster*), label = Player)) + geom\_point() + geom\_text(size = 3)*

Based on the clustering, choose the correct statement given that the clusters formed are (high SR, high Ave) - A, (low SR, low Ave) - B, (High SR, Low Ave) - C, (Low SR, High Ave) - D

Top of Form



**IVA RIchards and SR Tendulkar both belong to group A**

**Feedback :***You can even plot the graph after scaling for better separation of points and more intuitive visualisation.*

Bottom of Form

Bottom of Form

# Introduction

Welcome to the session on 'Hierarchical Clustering'. In the previous sessions, you got a basic understanding of what clustering is and how you can use the K-Means algorithm to create clusters in your data set. You also saw the execution of the K-Means algorithm in R.

## In this session

You will learn about another algorithm to achieve unsupervised clustering. This is called **Hierarchical Clustering**. Here, instead of pre-defining the number of clusters, you first have to visually describe the similarity or dissimilarity between the different data points and then decide the appropriate number of clusters on the basis of these similarities or dissimilarities.

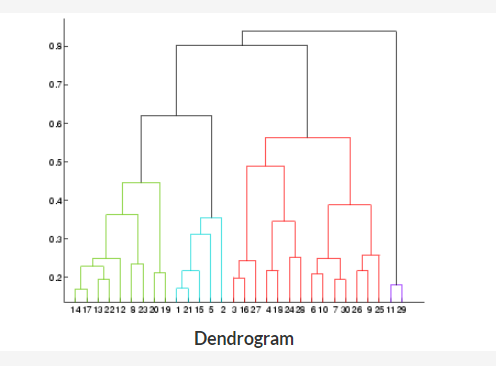
You will learn about:

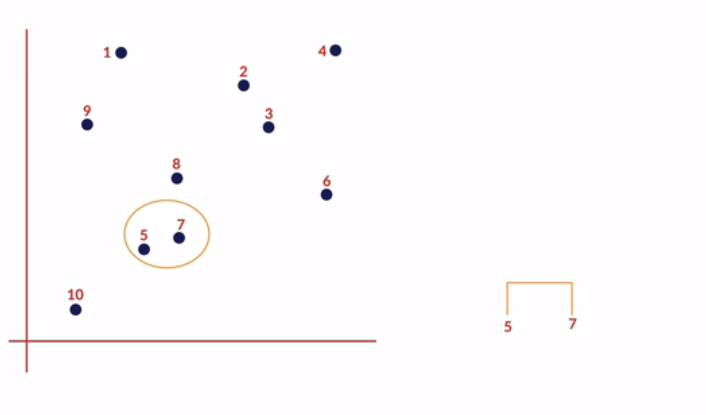
* Hierarchical clustering algorithm
* Interpreting the dendrogram
* Cutting the dendrogram
* Types of linkages

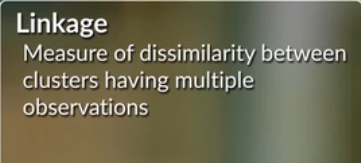
# Hierarchical Clustering Algorithm

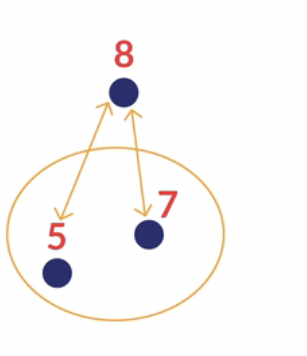
One of the major considerations in using the K-means algorithm is deciding the value of K beforehand. The hierarchical clustering algorithm does not have this restriction.

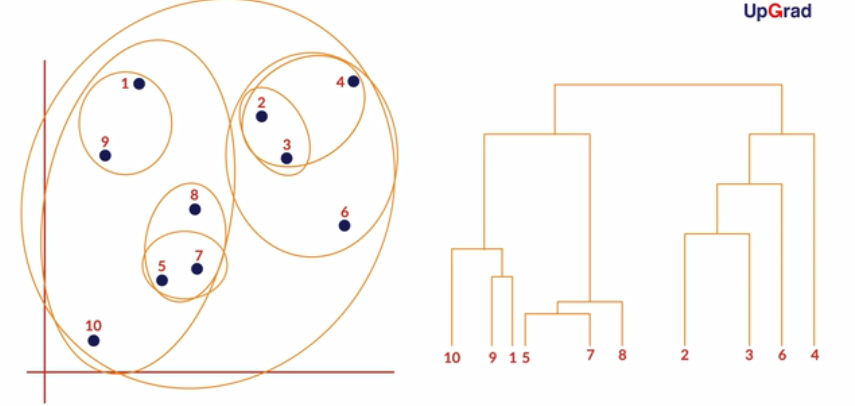
The output of the hierarchical clustering algorithm is quite different from the K-mean algorithm as well. It results in an inverted tree-shaped structure, called the dendrogram. An example of a dendrogram is shown below.











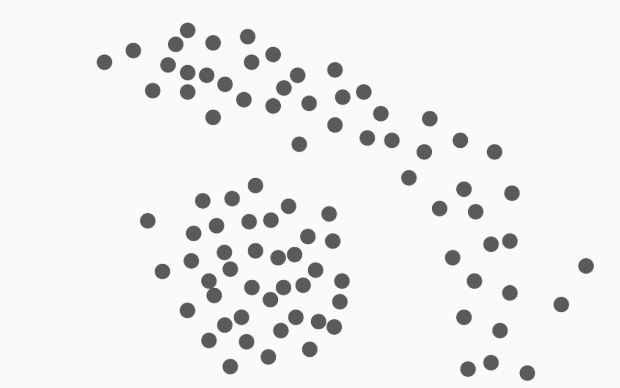
In the K-Means algorithm, you divided the data in the first step itself. In the subsequent steps, you refined our clusters to get the most optimal grouping. In hierarchical clustering, the data is not partitioned into a particular cluster in a single step. Instead, a series of partitions/merges take place, which may run from a single cluster containing all objects to n clusters that each contain a single object or vice-versa.

This is very helpful since you don’t have to specify the number of clusters beforehand.

Given a set of N items to be clustered, the steps in hierarchical clustering are:

1. Calculate the NxN distance (similarity) matrix, which calculates the distance of each data point from the other
2. Each item is first assigned to its own cluster, i.e. N clusters are formed
3. The clusters which are closest to each other are merged to form a single cluster
4. The same step of computing the distance and merging the closest clusters is repeated till all the points become part of a single cluster

Thus, what you have at the end is the dendrogram, which shows you which data points group together in which cluster at what distance. You will learn more about interpreting the dendrogram in the next segment.



**Hierarchical Clustering**

You had made clusters for it using the K-Means algorithm. How do you think clusters will be made using hierarchical algorithm on this data?

 Suggested Answer

*Since now you are looking at the closest distance between clusters, you will get two clusters - one at the center and the one which contains the points at the edges.*

**Hierarchical Clustering**

Look at the following matrix. This is the distance matrix between 4 points - A, B,C, D. Find out which 2 clusters will merge first.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | A | B | C | D |
| A |  |  |  |  |
| B | 2.24 |  |  |  |
| C | 8.06 | 10.00 |  |  |
| D | 5.83 | 8.06 | 5.00 |  |

Top of Form



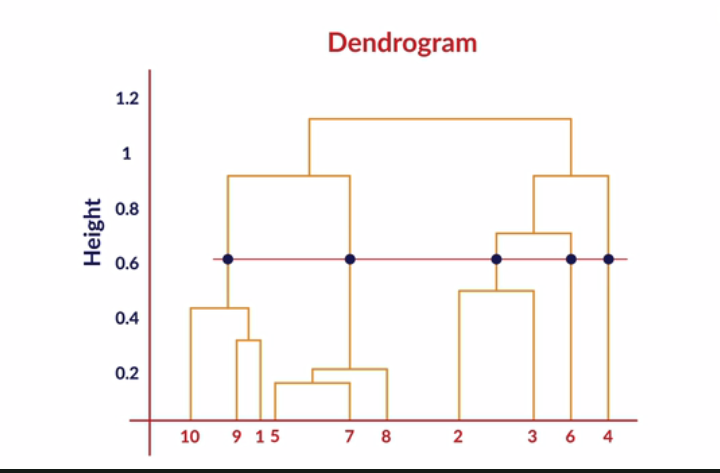
**A-B**

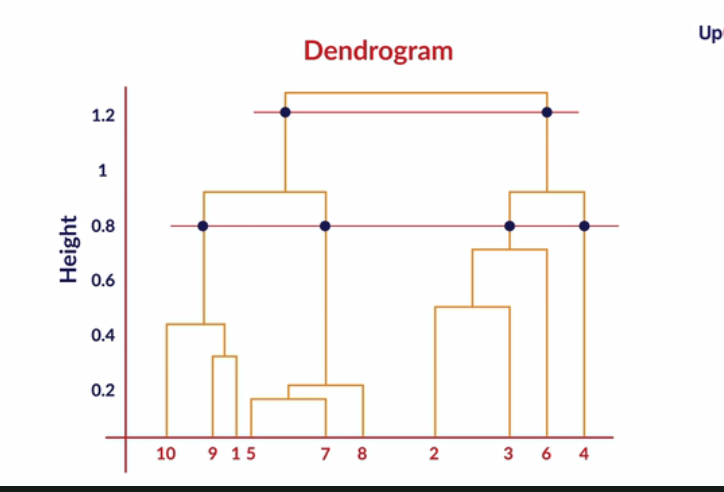
**Feedback :***Look at the data and see that the minimum distance is between A and B*

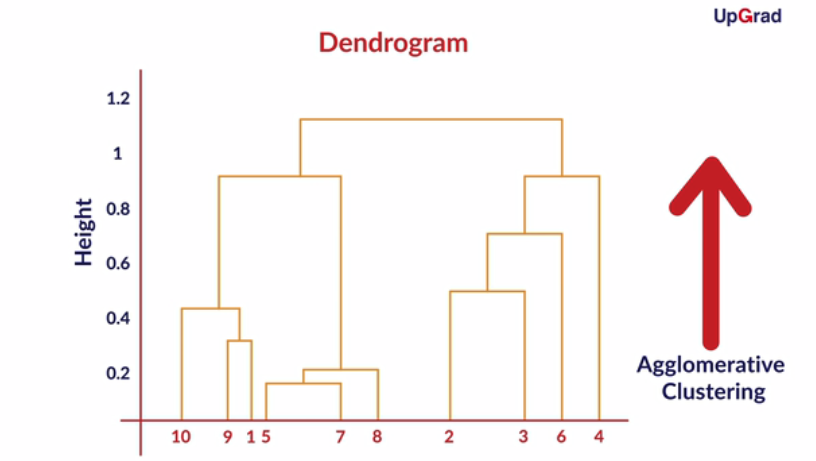
Bottom of Form

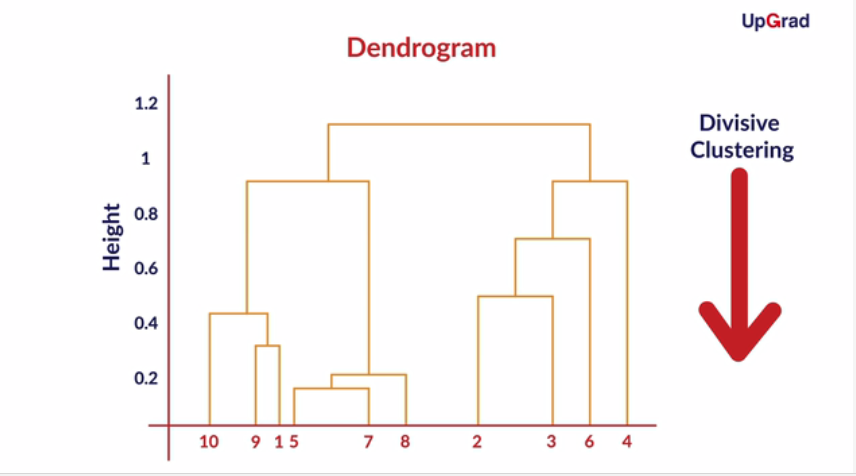
# Interpreting the Dendrogram

The result of the cluster analysis is shown by a dendrogram, which starts with all the data points as separate cluster and indicates at what level of dissimilarity any two clusters were joined.

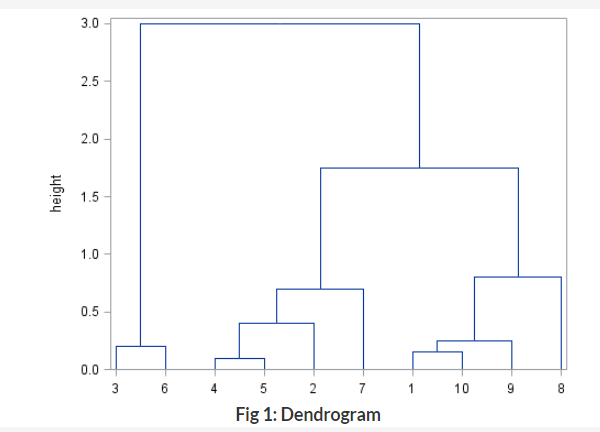








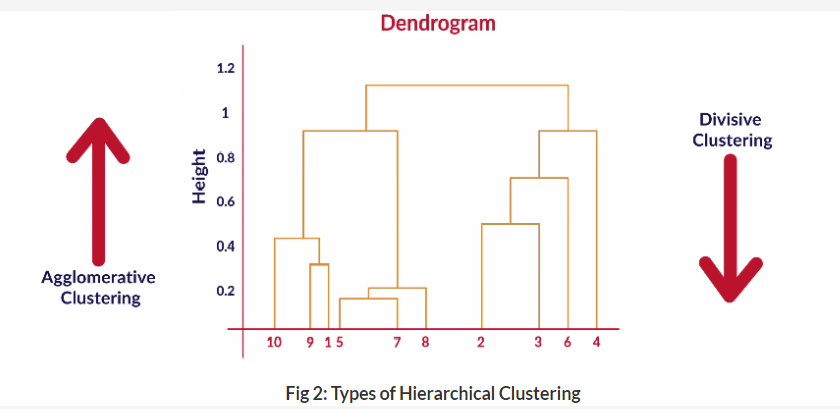
As you saw, the y-axis of the dendrogram is some measure of the dissimilarity or distance at which clusters join.



In the dendrogram shown above, samples 4 and 5 are the most similar and join to form the first cluster, followed by samples 1 and 10. The last two clusters to fuse together to form the final single cluster are 3-6 and 4-5-2-7-1-10-9-8.

Determining the number of groups in a cluster analysis is often the primary goal. Typically, one looks for natural groupings defined by long stems. Here, by observation, you can identify that there are 3 major groupings: 3-6, 4-5-2-7 and 1-10-9-8.

You also saw that hierarchical clustering can proceed in 2 ways — agglomerative and divisive. If you start with n distinct clusters and iteratively reach to a point where you have only 1 cluster in the end, it is called agglomerative clustering. On the other hand, if you start with 1 big cluster and subsequently keep on partitioning this cluster to reach n clusters, each containing 1 element, it is called divisive clustering.



**Comprehension - Hierarchical Clustering Algorithm**

Given below are five data points having two attributes x and y:

|  |  |  |
| --- | --- | --- |
| Observation | **x** | **y** |
| 1 | 3 | 2 |
| 2 | 3 | 5 |
| 3 | 5 | 3 |
| 4 | 6 | 4 |
| 5 | 6 | 7 |

The distance matrix of the points, indicating the Euclidean distance between points, is as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Label | 1 | 2 | 3 | 4 | 5 |
| 1 | 0.00 | 3.00 | 2.24 | 3.61 | 5.83 |
| 2 | 3.00 | 0.00 | 2.83 | 3.16 | 3.61 |
| 3 | 2.24 | 2.83 | 0.00 | 1.41 | 4.12 |
| 4 | 3.61 | 3.16 | 1.41 | 0.00 | 3.00 |
| 5 | 5.83 | 3.61 | 4.12 | 3.00 | 0.00 |

Take the distance between two clusters as the minimum distance between the points in the two clusters. Based on this information, answer the following questions.

**Hierarchical Clustering**

How many clusters are there initially (before any fusions have happened)?



**5**

**Feedback :***Since this is agglomerative clustering, initially, all the points are 1 cluster.*

**Hierarchical Clustering**

Which two clusters will be fused first?



**3 and 4**

**Feedback :***These two points(clusters) have the minimum distance.*

**Hierarchical Clustering**

Which clusters will be fused in step two?



**1 will be fused with the cluster(3, 4)**

**Feedback :***The distance between points 1 and 3 is 2.24 units, which is the minimum among all the new clusters. Hence they will be joined now.*

**Hierarchical Clustering**

How many total clusters are there right after point number 1 fuses with the cluster(3, 4)?

Top of Form



**3**

**Feedback :***Since three points have fused into 1 cluster, total clusters left are (1,3,4) - (2) - (5).*

Bottom of Form

**Hierarchical Clustering**

Which clusters will be fused after 1 fuses with (3, 4)?



**2 will fuse with the cluster (1, 3, 4)**

**Feedback :***The distance of point 2 from point 3 is 2.83 units, whereas the minimum distance of point 5 from the cluster (1,3,4) is 3 units*

What happens in the last step of the algorithm?

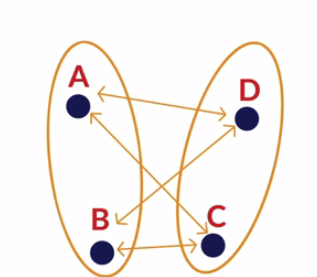


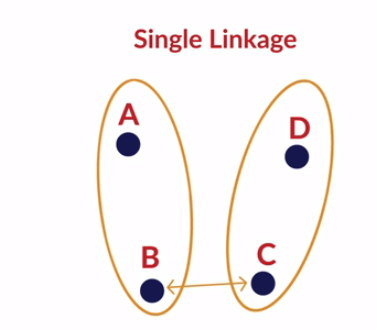
**5 fuses with ( 1, 2, 3, 4)**

**Feedback :***Since all the other points are already part of one cluster, the last point will also join that cluster at this step.*

# Types of Linkages

In our example, we took the minimum of all the pairwise distances between the data points as the representative of the distance between 2 clusters. This measure of the distance is called single linkage. Apart from using the minimum, you can use other methods to compute the distance between the clusters.





Let’s see once again the different types of linkages.

* **Single Linkage:**Here, the distance between 2 clusters is defined as the shortest distance between points in the two clusters
* **Complete Linkage:**Here, the distance between 2 clusters is defined as the maximum distance between any 2 points in the clusters
* **Average Linkage:**Here, the distance between 2 clusters is defined as the average distance between every point of one cluster to every other point of the other cluster.

You have to decide what type of linkage should be used by looking at the data. One convenient way to decide is to look at how the dendrogram looks. Usually, single linkage type will produce dendrograms which are not structured properly, whereas complete or average linkage will produce clusters which have a proper tree-like structure. You will see later what this means when you run the hierarchical clustering algorithm in R.



**In complete linkage hierarchical clustering, the inter cluster distance is defined as the longest distance between two points (one point in each cluster)**

**Feedback :***In the complete linkage, inter cluster distance is calculated as the maximum distance between 2 points (one in each cluster), However, the point is assigned to a new cluster basis it’s minimum distance from the clusters*

**Bases on clustering activity sheetHierarchical Clustering**

Select the points which get clustered in the first iteration.(First iteration is defined as the first merging of clusters - ie, from n clusters to n-1 clusters)



**C, D**

**Feedback :***Look at the distance between 2 points. You will see that the minimum distance is 2.2 units, which is between C and D*

**Hierarchical Clustering**

Select the points which get clustered in the second iteration. Use single linkage method



**A, E**

**Feedback :***You can see that the minimum distance now is between point A and E, which is 3.2 units.*

**Hierarchical Clustering**

How many iterations are required to form the final single cluster?



**5**

**Feedback :***Initially, n clusters are made and in each iteration, the number of clusters gets reduced by 1. So the number of iterations required is n-1. Here n = number of points = 6. So the correct answer is 5.*

# Summary

So what did you learn in this session?

You learnt about another clustering technique called Hierarchical clustering. You saw how it is different from K-Means clustering. One major advantage is that you do not have to pre-define the number of clusters. However, since you compute the distance of each point from every other point, it is time-consuming and needs a lot of processing power. Let's recall what you have learnt.

**Graded Questions**

Below is given a set of 6 points which have to be clustered by agglomerative clustering method. Use the single linkage method for clustering.

|  |  |  |
| --- | --- | --- |
| **Point Label** | **X** | **Y** |
| A | 6 | 0 |
| B | 1 | 2 |
| C | 2 | 7 |
| D | 4 | 6 |
| E | 5 | 3 |
| F | 11 | 1 |

Answer the following questions based on the data given above.

**Hierarchical Clustering**

What is the distance between points B and E?



**4.1**

**Feedback :***Distance is sqrt( 5-1)^2 + (3-2)^2)*

Based on the concept of agglomerative clustering, which two points will get clustered first?



**C-D**

**Feedback :***The distance between C and D is sqrt(5), which is lower than any other pair of clusters.*

**Hierarchical Clustering**

How many clusters will be formed in total after the completion of second iteration. (Each iteration is defined as the process of merging of clusters)

Top of Form

Bottom of Form



**4**

**Feedback :***After each iteration, the number of clusters decreases by 1*

# Constructing & Interpreting the Dendrogram

We will use the same online retail case study and data set that we used for the K-Means algorithm. For making the customer segments this time, we will use the hierarchical algorithm.

The hclust() function, which is an in-built function in R is used to construct the dendrogram using the distance matrix calculated earlier. You ran the algorithm using both the single and the complete linkages. As you saw using the plot() function, the shape and size of the dendrogram vary a lot with the type of linkage used in the algorithm.

**Clustering**

It is a clustering procedure where all objects start out in one giant cluster. Clusters are formed by dividing this cluster into smaller and smaller clusters.

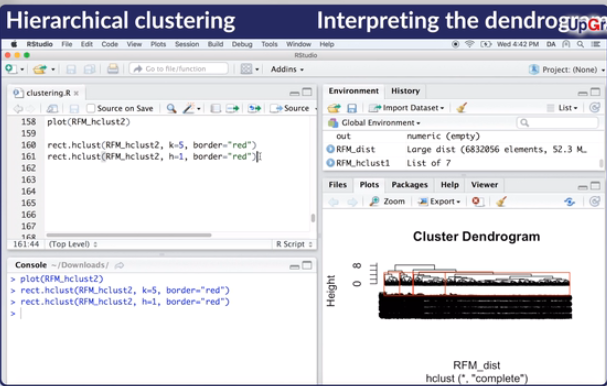


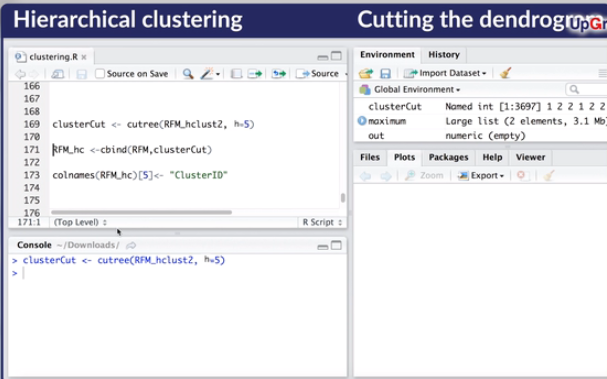
**Divisive**

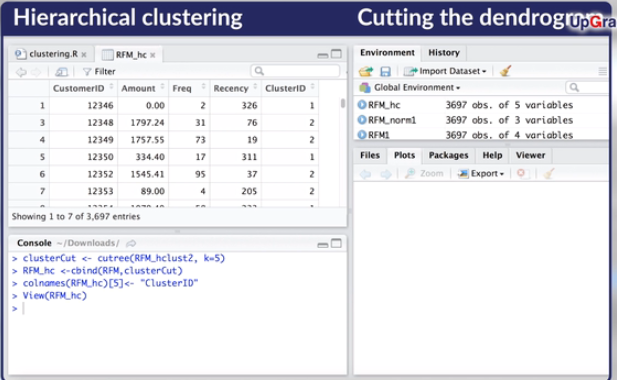
**Feedback :***This is how divisive clustering is defined, whereas agglomerative clustering starts at with n clusters and then reaches a giant cluster.*

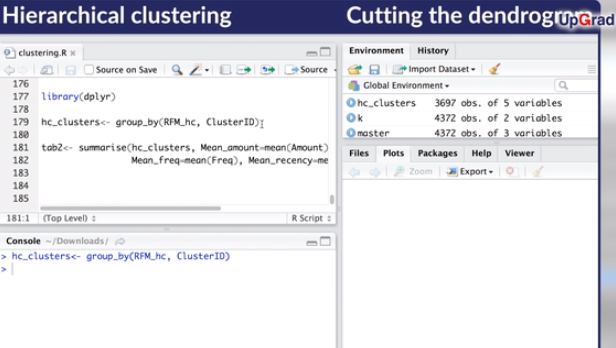
# Cutting the Dendrogram & Analyzing the Clusters

As you saw earlier, you can either cut the dendrogram depending on the number of clusters you want, or you can define the level of similarity among the clusters, i.e. the appropriate y-axis level. Both of these objectives can be achieved through the cutree() function.





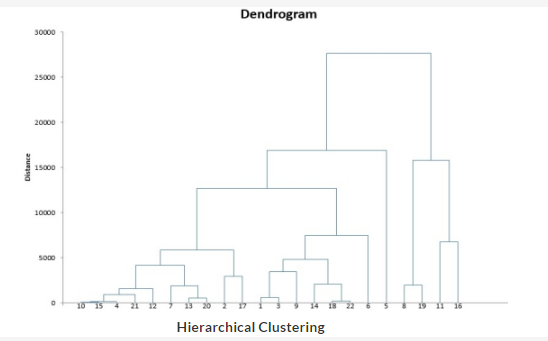




This function commonly takes 2 arguments. First is the dendrogram that has been constructed using the hclust() function. The second is the parameter which signifies where to cut the dendrogram. This argument can either be passed as "k" or "h". The parameter "k" stands for the number of clusters, whereas the parameter "h" stands for the height of the dendrogram, or the dissimilarity measure.

Thus, you observed that you can graphically visualise the cut of your dendrogram using the rect.hclust() function, but can actually cut the dendrogram to arrive at the required clusters using the cutree() function.

You then appended the obtained ClusterIDs to the RFM data set, and analysed the characteristics of each cluster to derive the business insights from the different customer segments or clusters, in the same way as you did for the K-Means algorithm.



**Hierarchical Clustering**

Consider the above dendrogram for agglomerative clustering and answer the following questions.

Find number of clusters if threshold value is 10000. (refer fig)



**5**

**Feedback :***Draw a horizontal line at that height. It cuts 5 vertical lines, all of which represent a cluster.*

**Hierarchical Clustering**

Find the threshold value if the no. of clusters to be formed is 4. (refer fig



**15000**

**Feedback :***Look at the height at which a horizontal line will cut 4 vertical lines.*

# Industry Insights

Now let's hear from our industry experts regarding the comparison between the K-Means algorithm and the Hierarchical clustering algorithm, before learning how to choose between the two based on your business problem.

So, you learnt that whether you use k-means or hierarchical clustering algorithm depends on your hardware and the data that you are dealing with.

Now, you will look at a good statistical hack to solve segmentation problems so that you get meaningful segments, where you will use both hierarchical and k-means algorithms to complement each other.

These insights were really helpful. You looked at how these clustering methods can be used to complement each other. You also looked at the differences between these methods, and the cases where you would prefer one method over the other.

You can also refer to the optional section on clustering. There, you will learn how factors are chosen to make the clusters and look at how an industrial problem was solved in the manner just described above.

*Hierarchical clustering generally produces better clusters, but is more computationallyintensive*

**Hierarchical Clustering**

Can you use the dendrogram to make meaningful clusters? (By looking at which elements leave and join at what height)

Suggested Answer

*Yes. It is a great tool. You can look at what stage an element is joining a cluster and hence see how similar or dissimilar it is to the rest of the cluster. If it joins at the higher height, it is quite different from the rest of the group. You can also see which elements are joining which cluster at what stage and can thus use business understanding to cut the dendrogram more accurately.*