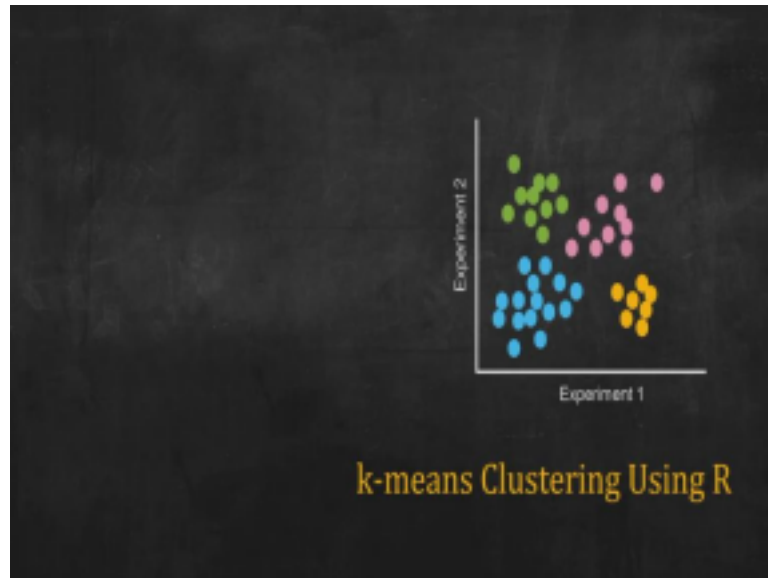


**Data Analysis for Biologists**  
**Professor Biplab Bose**  
**Department of Biosciences and Bioengineering**  
**Indian Institute of Technology Guwahati**  
**Lecture 39**  
**K-Means Clustering Using R**

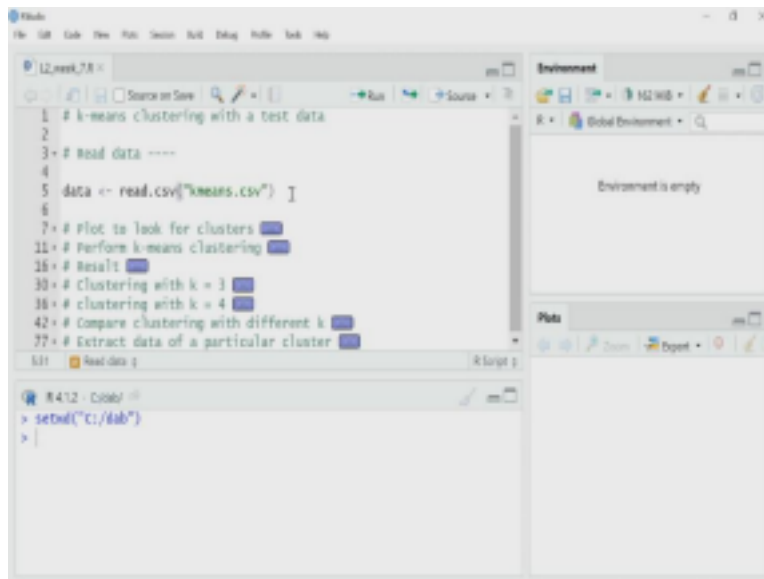
(Refer Slide Time: 00:29)



Hello everyone, welcome back. In the last lecture, we learned about k-means clustering. In k-means clustering, we are performing a flat clustering algorithm, where we divide a heterogeneous data set into relatively homogeneous clusters and the number of clusters is  $k$ . So, the number of clusters is predefined.

In this lecture, I will perform k-means clustering using R. I have a synthetic data set. I have created that data set in such a way so that we can easily understand how k-means algorithm work. So, let us jump into R studio and perform k-means clustering.

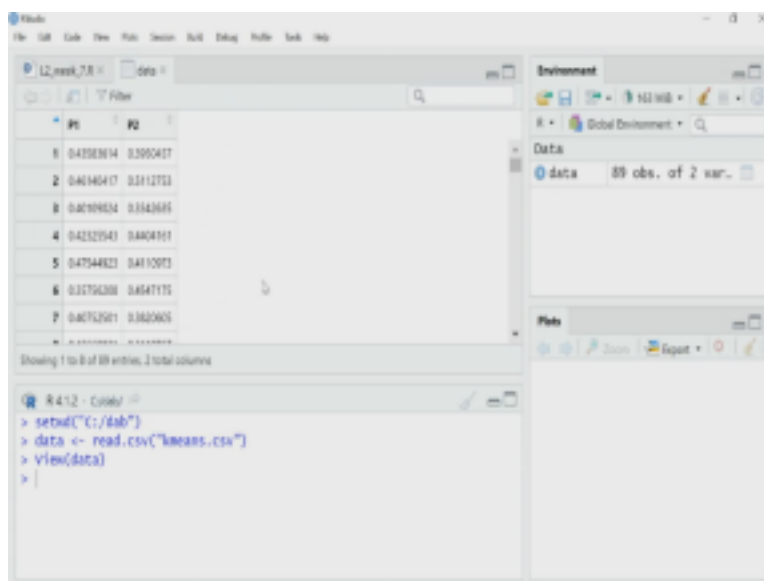
(Refer Slide Time: 01:17)



`data ← read.csv("kmeans.csv")`

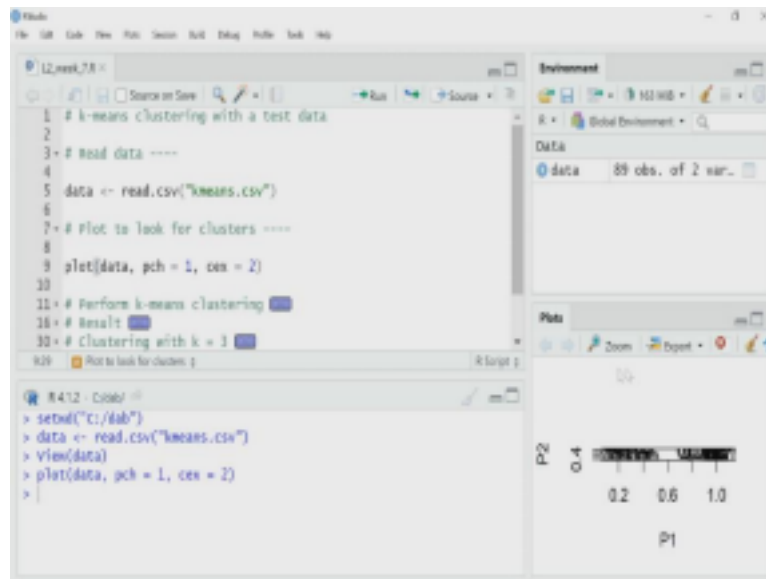
So, the name of the data file is k-mean dot csv. So, I will read a, use read dot csv function to read it and store that data as a data variable. Now, to understand what type of data it is let us open that data to see what we have here.

(Refer Slide Time: 01:36)



So, you can see it is a two column data. The first column is for P1 and the second column is for P2. You can imagine P1, P2 maybe just two parameter or variable that you may have measured in an experiment and there are 89 samples here. So, we have 89 rows.

(Refer Slide Time: 01:54)

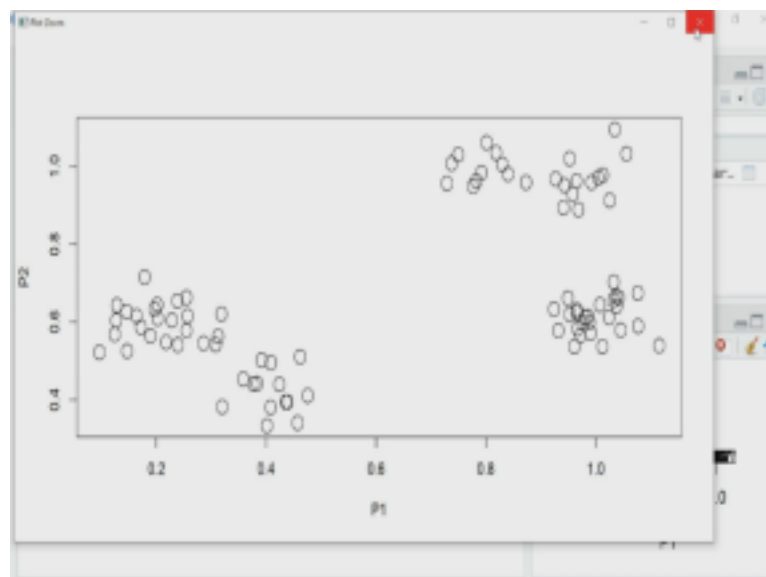


```
data <- read.csv("kmeans.csv")
```

```
plot(data, pch = 1, cex = 2)
```

Now, let me plot this data as a scatter plot to see that how that data is distributed. Does it has some clusters already or not, that will be clear if I make a scatter plot. So, I am using the plot function here and the arguments are, data is obviously the input argument and I want to use a symbol 1 so that I will get you know open circle as symbol and we want to say using size 2. So, I have made the plot here, let me zoom it out.

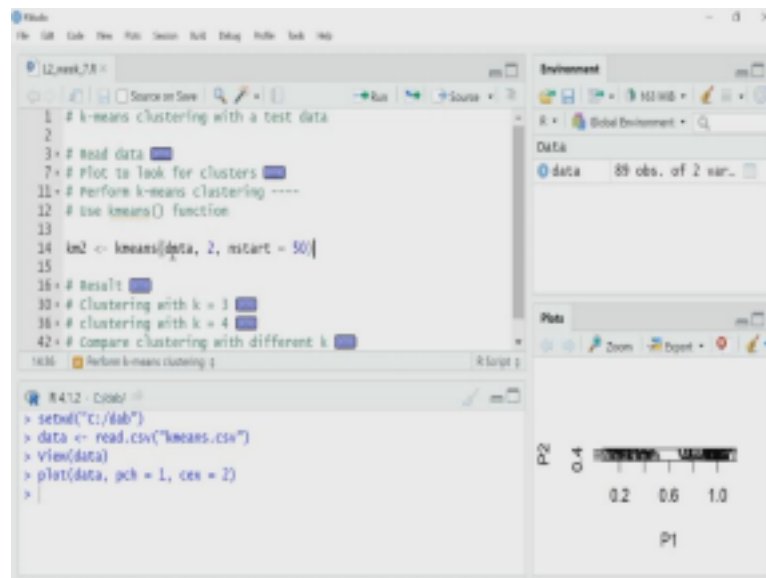
(Refer Slide Time: 02:21)



So, you can see as I said it is a synthetic data, I have intentionally created this data so that we can understand the k-means clustering easily. So, P1 and P2 are the horizontal and vertical axes respectively. And we have this 81 data point you can easily see on the left hand side of

the diagram, we have almost like two clusters I can see whereas here I may have for two or three or more clusters on the right hand side and these two are quite separated and our k-means algorithm should be able to detect it.

(Refer Slide Time: 03:00)



```
1 # k-means clustering with a test data
2
3 # Read data
4
5 # Plot to look for clusters
6
7 # Perform k-means clustering ----
8 # Use kmeans() function
9
10
11 km2 <- kmeans(data, 2, nstart = 50)
12
13
14 # Result
15
16 # Clustering with k = 3
17
18 # Clustering with k = 4
19
20 # Compare clustering with different k
21
22 Perform k-means clustering
```

```
R 4.1.2 - Console
> setwd("C:/dab")
> data <- read.csv("kmeans.csv")
> View(data)
> plot(data, pch = 1, cex = 2)
> |
```

`km2 ← kmeans(data, 2, nstart = 50)`

So, let us move and move for performing k-means. So, to perform k-means what I will do. I will use an inbuilt function called k-means it is inbuilt in R. So, what will be the arguments? Okay the first argument will be the data. The second argument here given as two is the number of cluster I want. So, I am starting the clustering with two clusters. I do not know what is the optimum number of clusters in this data.

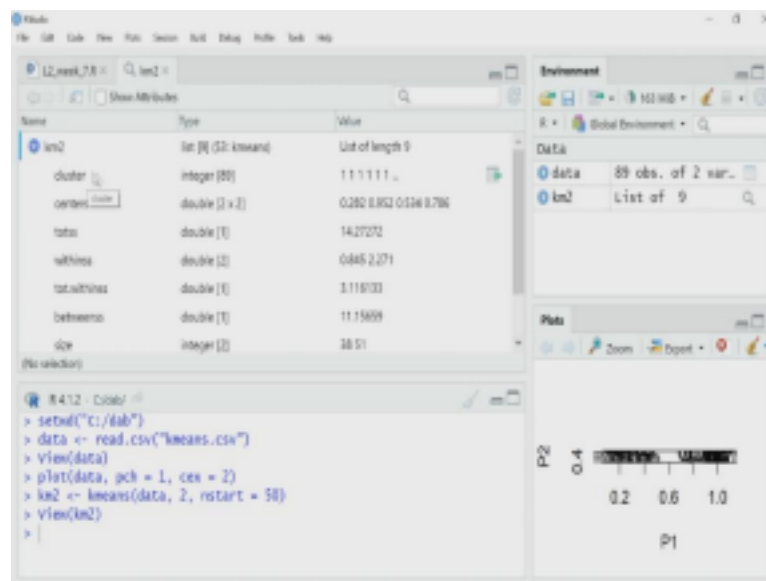
So, blindly I am starting with two because that is the minimum number of cluster you want and if you remember the last lecture, see the initial seeding of the centroid or the centre of each clusters are random. And where you have seeded that actually can change the result of k-means algorithm. So, what we usually do, we repeat the same process multiple time.

So, I will start the k-means algorithm to find two clusters once and it will reach the result and perform complete the work and then again, I will start on the same data with two clusters and I will again try to perform the clustering. So, in this way, we perform multiple clustering and then we decide, the algorithm decides by default which one is giving the best result and that best result is given as output to you. So, by n start, I am saying that the number of start, number of random start that I have to do in this algorithm is 50. Usually, it should be 20 or

more than 20 I have kept it 50 to be safe.

So, I will execute this k-means function and I will store all the output of that function into a variable called km2 k mean 2 because I have k equal to 2. So, I perform it, let us check the result of it. So, how should I check the result?

(Refer Slide Time: 04:59)



```
R 4.1.2 - Console  
> setwd("c:/dab")  
> data <- read.csv("kmeans.csv")  
> view(data)  
> plot(data, pch = 1, cex = 2)  
> km2 <- kmeans(data, 2, restart = 50)  
> view(km2)  
> |
```

Before I go into details of that let me click on this km2 variable and see what we have, we have lots of information in km2, it has this cluster variable which I will come and discuss it is actually a list. It is the vector which stores the number, the label of the cluster to which a particular observation belongs. So, suppose I have 89 data points, some of them has k equal to 2, k means clustering, so, I have two clusters in my data.

So, now, some of the data some of the observation will be in cluster 1, some of them will be in cluster 2. So, this cluster list or vector store that information, the centre which we will call again, the centre's variable stores the coordinate of the centroid of each of these clusters. So, I have 2 cluster. So, that means 2 centroid that is why a 2 by 2 because it is a two dimensional data. So, I have a two by two centres information.

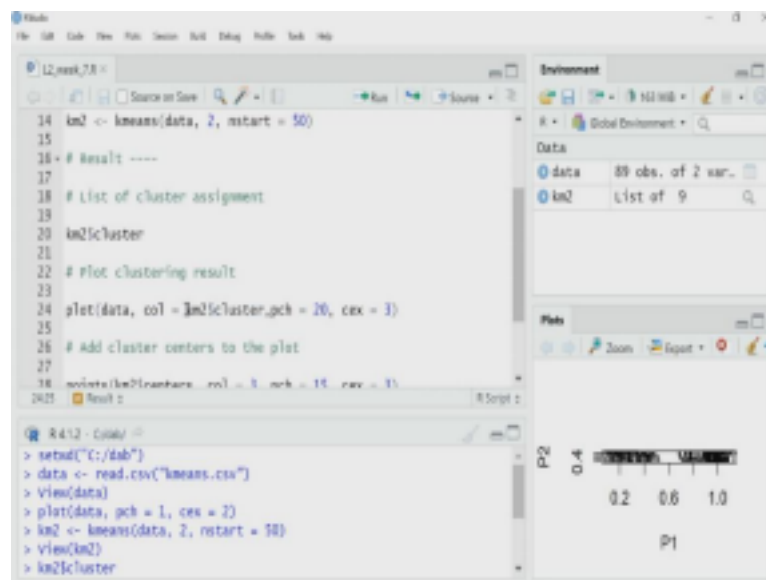
So, we will collect and use this information to understand how the clustering is working. And also, if you remember to evaluate the quality of clustering, we use the dispersion of the data within each cluster that is by within clusters sum of square and we compare the dispersion of the clusters among themselves by between clusters sum of square. So, I will go and discuss



So, my first observation is this one for which P1 is 0.435 and P2 is 0.39, my k-means algorithm for k equal to 2 is telling that okay these data belongs to, this observation belongs to cluster 1 whereas the last observation the 89th observation right 89th observation my algorithm is saying that, that belongs to cluster 2, that is why it has stored the label 2 in this cluster variable. So, this is how it has stored.

So, each of these observation is now labelled right either it is in cluster 1 or in cluster 2, this information will be very useful because when you want to plot the k-means clustering result or I want to extract suppose I want all the data or all the observation for cluster 2 in one as a one set. So, in that case, I have to use these labels.

(Refer Slide Time: 08:01)



```
14 km2 <- kmeans(data, 2, restart = 50)
15
16 # Result ----
17
18 # List of cluster assignment
19
20 km2$cluster
21
22 # Plot clustering result
23
24 plot(data, col = km2$cluster, pch = 20, cex = 3)
25
26 # Add cluster centers to the plot
27
28 mtext(km2$centers, col = 1, xnk = 15, yaxp = 1)
29
```

Environment

Object	Class
data	89 obs. of 2 var.
km2	List of 9

```
R 4.1.2 - Console
> setwd("C:/dab")
> data <- read.csv("kmeans.csv")
> View(data)
> plot(data, pch = 1, cex = 2)
> km2 <- kmeans(data, 2, restart = 10)
> View(km2)
> km2$cluster
```

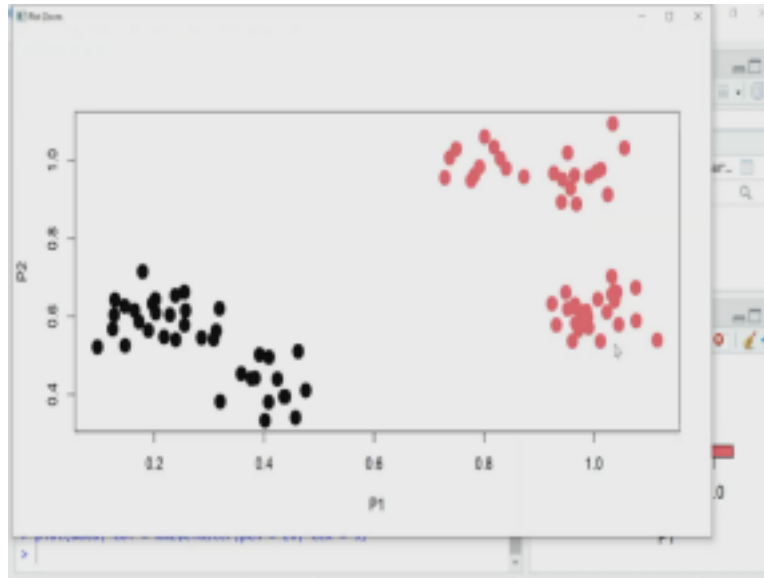
`plot(data, col = km2$cluster, pch = 20, cex = 3)`

Let me go back to my script and see what we will do now. So, now, I have got the idea that which observation is in cluster 1 which observation is cluster 2 something like that. Now, I want to visualize it. So, I will make a scatter plot where each of these observations those dots will be coloured based upon in which cluster it is. So, for example, I see okay all observations which are in cluster one will be coloured by black whereas, all observation is cluster 2 will be coloured by red something like that.

So, what I am doing, I am using the plot function which can plot a scatter plot and I am using the data as my input argument, the same data I want to plot it as a scatter plot but what I am giving extra to this this time is that I am telling this plot function that there is an another



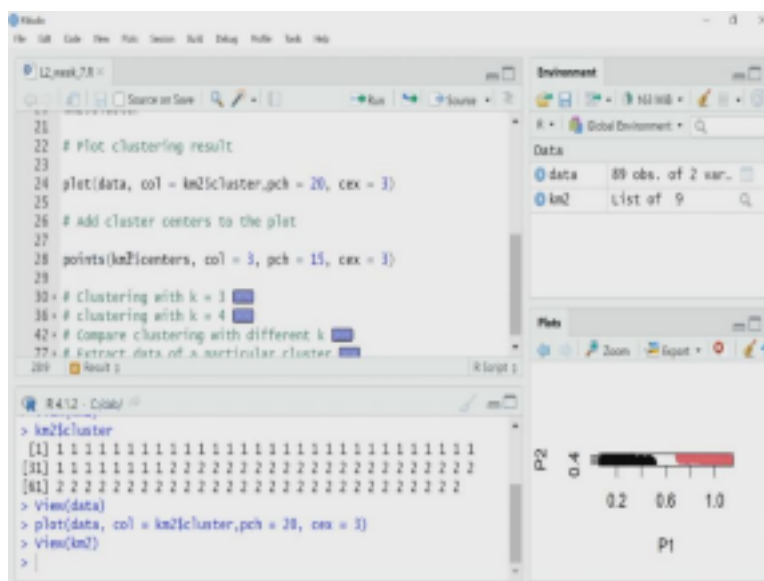




So, let me plot that, I have plotted zoom it, we have done k-means algorithm for k equal to 2 that means 2 cluster. So, this is my first cluster this all black dots are first cluster and all these red dots are the second cluster, it is obvious the way I have created the data, we can easily see that all these black dots are quite away quite far from the red dots. So obviously, that is how k-means algorithm has broken down these data into two clusters.

Now, I have plotted the scatter plot where each of these clusters are now colour coded. Now you may be interested to see where are the centroid, what is the central position for each of these cluster. So, for that on the same cluster plot, what I can do, I can overlay some extra data. So, what I will overlay?

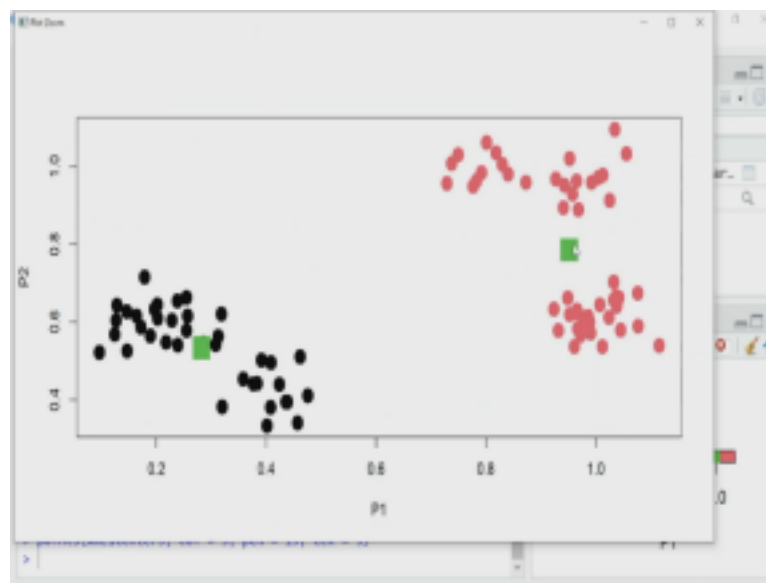
(Refer Slide Time: 11:14)



`points(km2$centers, col = 3, pch = 15, cex = 3)`

I will overlay that center data, if you remember, that km has the center data, where this information, the coordinates of each of the centroid of these two clusters are stored. So, I will take that data. So, what I am doing, I am using points as a function, points function and I am giving km2 dollar center. So, I am fetching the center's variable information and using that as an argument to this point function, so it will create points at those coordinate and I am using colour equal to 3. So, this will be green colour and I am using a symbol square field square. So, pch is equal to 15 and the size is 3.

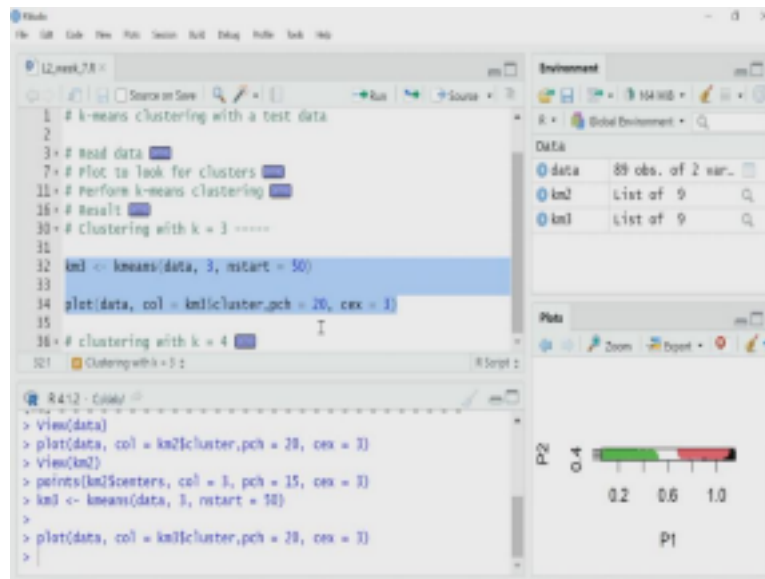
(Refer Slide Time: 11:56)



So, if I execute this, let me zoom on the same plot what I have now, on this two green squares are there. They are the centroid for the corresponding cluster. So, this one on the right-hand side near P1 equal to 1, this green box with green squares shows the centroid of my cluster 2, whereas the other green here on the left hand side is the centroid for cluster 1.

Now, when you are performing k-mean algorithm, you do not really know how many clusters should be there. So, let us in this case, also, I do not know whether these two clusters are good enough or I can actually break down this data in further in more clusters. So, what I will try now, I will use the same algorithm on the same data, but I will set k equal to 3 first, and then I will move into k equal to 4.

(Refer Slide Time: 12:56)



```
km3 <- kmeans(data, 3, nstart = 50)
```

```
plot(data, col = km3$cluster, pch = 20, cex = 1)
```

So, let us do the k equal to 3. I will use the same k-means function, everything remains same, I will plot the same scatterplot with the same way of colour coding. So, in this case, I will have a three colour, 1 for the first cluster 1, 2 for the cluster 2 that means the cluster 1 will be black cluster 2 will red and the third cluster the colour will be 3 that means it will be green. I execute the script for that, the plot is ready let us zoom and see.

(Refer Slide Time: 13:16)

```
km3 <- kmeans(data, 3, nstart = 50)
```

```
plot(data, col = km3$cluster, pch = 20, cex = 1)
```

Earlier we have two clusters but now we have three because  $k$  equal to 3. So, what the algorithm has done, it has broken down this right-hand side cluster in the previous result into two part and you can easily use the logic. It is obvious that you can easily actually see that this one the red one the second cluster is quite far from the black one. So, it has broken it into two parts. Now, if I say okay now perform for  $k$  equal to 4 possibly what can happen either maybe these two can be broken or maybe this green thing this green cluster may be broken into two, I do not know let us see which one it will do.

(Refer Slide Time: 14:03)

```
km4 ← kmeans(data, 4, nstart = 50)
```

```
plot(data, col = km4$cluster, pch = 20, cex = 3)
```

So, I will now perform for  $k$  equal to 4. Same  $k$ -means function and the same way I will plot the scatterplot with colour code.

(Refer Slide Time: 14:12)

```
km4 ← kmeans(data, 4, nstart = 50)
```

```
plot(data, col = km4$cluster, pch = 20, cex = 3)
```

So here is the diagram. So that is what it has done. So, now what is it has done it has broken down this earlier, this left-hand side we had only one cluster so it has now broken it into two parts, one is green one is black. So, I have now four clusters in this data. Now, while we are discussing the k-means algorithms, the theory of that in the earlier class we have discussed that there is no way that I can know that what will be the optimum value of k before I perform k-means clustering.

And there is a method to systematically decide that what should be the optimum value of k. So, while I am executing this, you must be wondering that how far I will go. Should I keep on doing k equal to 5 ,6, k equal to 10 something like that, obviously not. So, I have to decide, I will stop somewhere. So, how to do that? In case of k-means algorithm we can use two information to decide that what should be the optimum number of k, optimum value for k that is the optimum number of clusters in my data that is decided based upon the within cluster sum of square and between cluster sum of square for each of the k.

So, what I have done till now I have performed k-means algorithm for k equal to 2, k equal to 3, k equal to 4, for all these I have stored the data in km2, km3 and km4. So, now I can fetch from these km2, km3, km4 the within cluster sum of square and between clusters sum of square data. But before I fetch that and visualize it to decide what should be the optimum value of k what I will do, I will perform the k-means clustering for a few more values of k,

like 5, 6 and 7, that is will be similar to what we have done, we do not need much explanation.

(Refer Slide Time: 16:08)

```
km5 ← kmeans(data, 5, nstart = 50)
```

```
km6 ← kmeans(data, 6, nstart = 50)
```

```
km7 ← kmeans(data, 7, nstart = 50)
```

So, I have written down them here, I am changing k from 5, 6 to 7. So, at a go, I will execute all those. So, I have now completed all the clustering from k equal to 2 to k equal to 7.

(Refer Slide Time: 16:27)

```
w2 ← km2$tot.withinss
```

```
w3 ← km3$tot.withinss
```

```
w4 ← km4$tot.withinss
```

```
w5 ← km5$tot.withinss
```

```
w6 ← km6$tot.withinss
```

```
w7 ← km7$tot.withinss
```

```
barplot( c(w2, w3, w4, w5, w6, w7), names = seq(2,7), xlab = "k",  
        ylab = " Within cluster SS", ylim = c(0, 4))
```

Now, I will fetch the within cluster sum of squared data for each of these clustering output. Now, what is within cluster sum of data? Within cluster sum of square is the measure of dispersion of data within a cluster. So, if I have three cluster, for each cluster, I can calculate the sum of squared deviation of each of the data in that cluster from its centroid. So, if this dispersion is small, that means the cluster is very cohesive or very homogeneous that means my clustering is good.

So, what we will do, we will take the total within cluster sum of square. So, if I have three cluster for each you have within cluster sum of square and for all these three, you sum together. So, you get total within cluster sum of square. So, for each of these clustering, I start with for example km2 stores that data for k equal to 2, km3 so for k equal to 3 and so on up to km equal to 7, it will be 7.

So, from each of these output I fetch the total within cluster sum of squared data. So, what is how it is written? Km2 dollar total tot dot within SS. So, that will fetch the total within cluster sum of square for that particular k-mean clustering result. In this way, I will go up to the 7, k equal to 7 one. So, I execute them at a go. So, I have already fetched them and stored them as w2, w3 up to w7. So, now I will plot these data as a bar plot. And that is what I am doing in the next part of the script, I am calling the bar plot function.

What should the input? I am creating a vector for each of these values that I collected, so I have collected w2 to w7 so I am making a vector using these w2, w3, w4 up to w7 and I am using the C function to create the vector. And then I have to put the label below each of the bar right and so we have to put the names for each of these bars.

So, the names variable is another argument for bar plot that is equal to 3, I am creating a sequence of number starting from 2 to 7 because k equal to 2 up to k equal to 7. So, I am using the sequence functions seq. And I am labelling the x axis horizontal axis equal as k and y axis the vertical axis I am labelling as within cluster sum of square and I am putting the limit from 0 to 4 because you can see that w2 two has the highest one which is 3, so, I want to scale it from the vertical axis from 0 to 4.

(Refer Slide Time: 19:23)

```
barplot( c(w2, w3, w4, w5, w6, w7), names = seq(2,7), xlab = "k",  
        ylab = " Within cluster SS", ylim = c(0, 4))
```

Let me plot it. It will be clear. So, let me zoom here. So, I have zoomed here. So, the horizontal axis is k. I have performed 7 k-means clustering with values of k starting from 2 to 7. For each of them, we have extracted the within cluster sum of squares, this is the total within cluster sum of squares. So, you can see when I have two clusters, I have a very high value.

That means I have quite high heterogeneity within the cluster. But as I increase the number of clusters, I changed k from 2 towards 7, you can easily see that total within cluster sum of squares is dropping very fast and it is becoming very shallow asymptotically stabilized, it has become like asymptote. So it is becoming a fixed value after around 6 or 5, something like that.



So, maybe looking at this data, you can easily see that maybe if I stop at 5 k equal to 5, or k equal to 6, it will be optimal, I do not need to go beyond 6 actually. So, maybe I should choose k equal to 5 or k equal to 6 for my final clustering result. So, before I move into this, this is that within cluster sum of squares which gives you the dispersion of the data within the cluster. I now will calculate also the between clusters sum of squares. So, this is the dispersion between the clusters. And this dispersion should be high, my clusters should be separated from each other.

(Refer Slide Time: 20:49)

```
b2 ← km2$betweenss
```

```
b3 ← km3$ betweenss
```

```
b4 ← km4$ betweenss
```

```
b5 ← km5$ betweenss
```

```
b6 ← km6$ betweenss
```

```
b7 ← km7$ betweenss
```

```
barplot( c(b2, b3, b4, b5, b6, b7), names = seq(2,7), xlab = "k",
```

```
  ylab = " Between cluster SS", ylim = c(0, 15))
```

So, what I will do, just like within cluster sum of square, I will fetch the between clusters sum of square from each of these k-means algorithm output using this between SS variable. So, that is what I am doing here I am saying km2, km2 is the first clustering that I have done with

k equal to 2 dollar sign and between SS, so it will fetch that between SS data for km2 and I will assign that to b2, I am doing that up to b7. I execute all those together. So, I have already calculated got the b2 up to b7. Now I will create a bar plot just like I did for the within cluster. Let me check the bar plot.

(Refer Slide Time: 21:33)

```
barplot( c(b2, b3, b4, b5, b6, b7), names = seq(2,7), xlab = "k",  
        ylab = " Between cluster SS", ylim = c(0, 15))
```

See, we can see from 2 to 3 there is slight increase in the between clusters sum of square that is good, we want that the clusters should be separated. And then as I move forward, after 5 actually there is not much change, they are almost same. So, I have seen that in within cluster sum of square from 5 to 6, it has become almost same, it has become very low. So, 5 or 6 I could have picked k equal to 5 or 6 I could check pick for my final clustering.

And whereas the between cluster SS is showing me that 5 maybe the optimum one. So, using this within cluster sum of square data and between cluster sum of square data, I finalize, I decide that I will use k equal to 5 for k-mean clustering for this data set. So let me plot that.

(Refer slide Time: 22:21)

```
plot(data, col = km5$cluster, pch = 20, cex = 3)
```

So, I actually selected k equal to 5 so I have selected k equal to 5 because that is given optimum. So, I have already performed the k-mean clustering for k equal to 5 and that is stored in km5. So, I will just plot that as a scatterplot. Here it is.

(Refer Slide Time: 22:42)

```
plot(data, col = km5$cluster, pch = 20, cex = 3)
```

So, it has what it has done, it has broken down these, the right-hand side clusters into three part where the left hand side cluster is divided in two parts. So, this is I believe, is optimum clustering for my data set.

(Refer Slide Time: 22:58)

```
c1 ← which(km5$cluster == 1)
cluster.1 ← data[c1, ]
```

Now the last thing that we learn in this particular lecture is that okay, I have done optimum clustering, with k equal to 5. Now I want to fetch the observation for a particular cluster that may be useful for you. So, for example, suppose I want to extract the data of cluster 1. So, how I will do? Remember this km5 dollar cluster that is the cluster information in km5 where k was equal to 5 stores the label for cluster for each of these observation.

Now, what I will do, I will use the which function to extract the index of those observation, index of those observation, which has, which belongs to cluster 1. So, what I am doing? I want to find the index of those observation which belongs to cluster 1. So, how I am doing it? I am using which function and I am saying km5 cluster is equal to 1. If I want to fetch the data for cluster 5, I will say it will be equal to 5.

So, that indexes, those indexes are now stored at C1, cluster 1 and now I extract the data from that C1 data from my original data variable. So, how do I do it? I say data and then I index the row and column. The row number is this C1 and the column values because I want both the columns right P1 and P2. So, I keep that empty. So, this will fetch me all the data in observation, in my data variable where those observations belongs to cluster 1. And I assign that to cluster dot 1. Got it? Let us check the cluster 1 in the data frame I can check it.

(Refer Slide time: 24:54)

```
c1 ← which(km5$cluster == 1)
cluster.1 ← data[c1, ]
```

So, you can see Cluster 1 has 24 observation, it is written here 24 observation, 2 variable, so, it has maintained the original indexes. So, 15, 16, 17, 18 these all are in part of the cluster 1. These are the original observations index and they all belong to cluster 1. So, what we have learned in this lecture, we have seen how to perform k-means clustering using R and we have seen how to plot it in a scatterplot with the colour code for each of these cluster.

Then I have shown you how to decide the optimum number of clusters that is value of k based upon the within clusters sum of squared data and between clusters sum of squared data and at the end, we have seen how can I fetch the data for, observations for a particular cluster. You can do it for other clusters also. That is all for this video. Thank you for learning with me today.