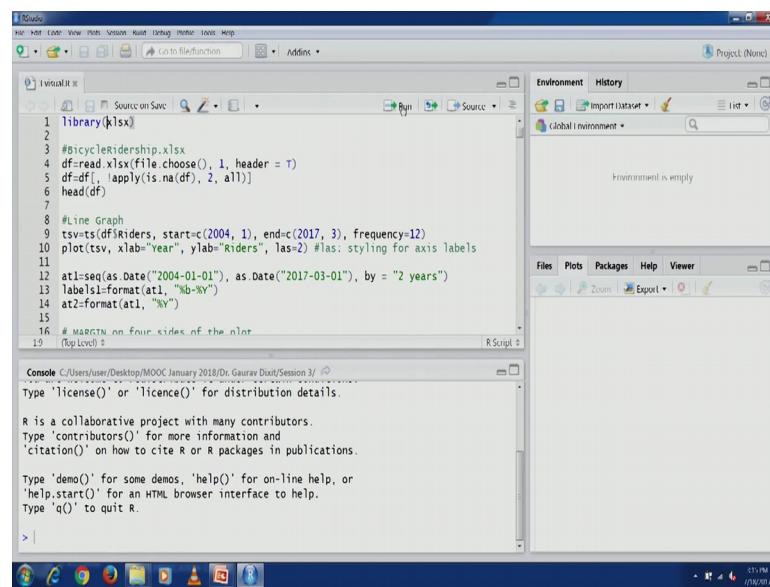


Business Analytics & Data Mining Modeling Using R
Dr. Gaurav Dixit
Department of Management Studies
Indian Institute of Technology, Roorkee

Lecture – 11
In Plot Labels

Welcome to the course Business Analytics and Data Mining Modelling Using R. In the previous lecture we did multi panel; multi panel plotting. Now in this in this particular lecture will start a with in plot labels. So, last time we also covered 10 lines while we were doing multiple panel a plotting. So, a let us start with in plot labels let us go back to r studio.

(Refer Slide Time: 00:50)



The screenshot shows the RStudio interface. The left pane displays the following R code:

```
1 library(xlsx)
2
3 #icycleridership.xlsx
4 df<-read.xlsx(file.choose(), 1, header = TRUE)
5 df<-df[, !apply(is.na(df), 2, all)]
6 head(df)
7
8 #line Graph
9 tsv<-ts(df$riders, start=c(2004, 1), end=c(2017, 3), frequency=12)
10 plot(tsv, xlab="Year", ylab="Riders", las=2) #las: styling for axis labels
11
12 at1<-seq(as.Date("2004-01-01"), as.Date("2017-03-01"), by = "2 years")
13 labels1<-format(at1, "%b-%Y")
14 at2<-format(at1, "%Y")
15
16 # MARGIN on four sides of the plot
17
18
```

The right pane shows the Environment and Global environment panes, both of which are currently empty. The bottom pane is the Console, which contains the R help text for 'R' and a prompt '>'. The taskbar at the bottom shows various application icons.

So, first let us reload the a data and the libraries let us load this library xlsx.

(Refer Slide Time: 01:17)

The screenshot shows the RStudio interface. The left pane contains an R script with the following code:

```
10 plot(tsv, xlab="", ylab="", xaxt="n", yaxt="n")
11 axis(1, at=at2, labels=labels1, las=2)
12 axis(2, las=2)
13 labels1=format(at1, "%b-%Y")
14 atz=format(at1, "%Y")
15
16 # MARGIN on four sides of the plot
17 par()$mar # number of lines
18 par(mar=c(8,4,4,2)+0.1)
19
20 plot(tsv, xlab="", ylab="", xaxt="n", yaxt="n")
21 axis(1, at=at2, labels=labels1, las=2)
22 axis(2, las=2)
23 mtext(side=1, text="Month-Year", line=5.5)
24 mtext(side=2, text="Riders", line=3.0)
25 |
```

The right pane, titled "Environment", shows the global environment is empty.

And the used cars a data set. You can see 79 observations and 11 variables.

(Refer Slide Time: 01:26)

The screenshot shows the RStudio interface. The left pane contains an R script with the following code:

```
23 mtext(side=1, text="Month-Year", line=5.5)
24 mtext(side=2, text="Riders", line=3.0)
25
26 graphics.off() # close all graphic devices
27 par()$mar
28
29 #usedCars.xlsx
30 df1<-read.xlsx(file.choose(), 1, header = T)
31 df1<-df1[, apply(is.na(df1), 2, all)]
32
33 Age=2017-df1$Mfg_year
34 df1<-cbind(df1, Age)
35 dff<-df1
36 df1<-df1[,c(1,2,3)]
37
38 head(df1)
```

The right pane, titled "Environment", shows the global environment. The "Data" section lists:

- df1: 79 obs. of 12 variables
- Values: Age num [1:79] 4 5 6 4 9 5 6 3...

Now let us recreate the age variable as we did in the previous lecture first 3 variables are not important to us. So, we are trying to get rid of those variables.

(Refer Slide Time: 01:55)

The screenshot shows the RStudio interface with the following R code in the console:

```
31 df1=df1[, apply(is.na(df1), 2, all)]
32 
33 Age=2017-df1$Mfg_year
34 df1=cbind(df1, Age)
35 dffb=df1
36 df1=df1[,-c(1,2,3)]
37 
38 head(df1)
39 str(df1)
40 df1$transmission=as.factor(df1$transmission)
41 df1$c_Price=as.factor(df1$c_Price)
42 str(df1)
43 summary(df1)
44 
45 #SCATTERPLOT
46 rrange(df1$KM)
47 |
```

The Environment pane shows two datasets: df1 (79 obs. of 9 variables) and dffb (79 obs. of 12 variables). The Values pane shows the 'Age' variable as a numeric vector from 1 to 79.

Let us also convert these 2 variables transmission and c price as factor variables

(Refer Slide Time: 02:03)

The screenshot shows the RStudio interface with the following R code in the console:

```
48 range(df1$KM)
49 range(df1$Price)
50 plot(df1$KM, df1$Price, xlim = c(18,180), ylim = c(1,75),
51 xlab="KM", ylab="Price")
52 df1=df1[-23,]
53 df1=df1[-23,]
54 
55 range(df1$KM)
56 range(df1$Price)
57 plot(df1$KM, df1$Price, xlim = c(18,180), ylim = c(1,15),
58 xlab="KM", ylab="Price")
59 
60 #BAR CHART
61 avgPrice=(mean(df1[which(df1$transmission=='0'),]$Price),
62 |
```

The Environment pane shows the same datasets as before. The Values pane shows the 'Age' variable as a numeric vector from 1 to 79.

Let us also eliminate that outlier point, like we did in the previous lectures now we are ready to go.

(Refer Slide Time: 02:10)

```

49 xlab="KM", ylab="Price")
50
51 df1[df1$Price>70,]
52 dfb=df1
53 df1=df1[-23,]
54
55 range(df1$KM)
56 range(df1$Price)
57 plot(df1$KM, df1$Price, xlim = c(18,180), ylim = c(1,15),
58 xlab="KM", ylab="Price")
59
60 #BAR CHART
61 avgPrice=c(mean(df1[which(df1$Transmission=='0'),]$Price),
62 mean(df1[which(df1$Transmission=='1'),]$Price))
63 Trans=c("0", "1")
64
531 (Top Level) §

```

Console C:/Users/user/Desktop/MOOC January 2018/Dr.Gaurav Dusit/Session 3/

```

> Age=2017-df1$Mfg_Year
> df1=cbind(df1, Age)
> dffb=df1
> df1=df1[,-c(1,2,3)]
> df1$Transmission=as.factor(df1$Transmission)
> df1$c_Price=as.factor(df1$c_Price)
> df1[df1$Price>70,]
Fuel_Type SR_Price KM Price Transmission Owners Airbag C_Price Age
23 Diesel 116 19 72 1 1 1 1 1 3
> dfb=df1
> |

```

So, will start with in plot labelling, so in plot labelling can be useful when we are dealing with when we are trying to understand a large amount of data specially in clustering. So, it is easier for us to have a look at the make a visual inspection of the data and try to understand different clusters that could be there. So, we will start with a scatter plot with label point.

(Refer Slide Time: 02:45)

```

287 title(main="Aggregation for year", adj=0, cex.main = 0.9)
288 grid()
289
290 # Scatterplot with labeled points
291 dffb=dffb[Price>70,]
292 dffb=dffb[Price<12,]
293 dffb=dffb[KM<150,]
294
295 dffb=dffb[-c(13,23,29,65,73,)]
296
297 range(dffb$KM)
298 range(dffb$Price)
299 plot(dffb$KM, dffb$Price, xlim = c(25,120), ylim = c(1,9),
300 xlab="KM", ylab="Price", panel.first = grid())
301
302
295 (Top Level) §

```

Console C:/Users/user/Desktop/MOOC January 2018/Dr.Gaurav Dusit/Session 3/

```

> df1=cbind(df1, Age)
> dffb=df1
> df1=df1[,-c(1,2,3)]
> df1$Transmission=as.factor(df1$Transmission)
> df1$c_Price=as.factor(df1$c_Price)
> df1[df1$Price>70,]
Fuel_Type SR_Price KM Price Transmission Owners Airbag C_Price Age
23 Diesel 116 19 72 1 1 1 1 1 3
> dfb=df1
> df1=df1[-23,]
> |

```

So, this particular data frame we have already created d f f b. Now there are few points if we plot the if we have if we draw this particular plot if we create this particular plot

between km and price there are few points which are far away from the major chunk of the points.

So, therefore, we are trying to get rid of those points. So, that we can go ahead with the labelling, because when we do labelling in the plot therefore, it can be slightly messy if there are too many points far away. So, there is less scope for labelling in the major chunk of the points for major chunk of the points. So, therefore, we will try to get rid of these far away points.

(Refer Slide Time: 03:32)

The screenshot shows the RStudio interface with the following details:

- Code Editor:** Displays R code for data aggregation and plotting. The code includes filtering rows where Price > 70, creating a scatterplot with labeled points, and setting axis limits for KM and Price.
- Environment View:** Shows the global environment with three datasets:
 - df1: 78 obs. of 9 variables
 - dfb: 79 obs. of 9 variables
 - dffb: 79 obs. of 12 variables
- Data View:** Shows a subset of the dffb dataset with columns Age, num, and values.
- Console:** Displays the R command history, including the execution of the R code shown in the editor.

So, this is 1 this is another one, this is the third one.

(Refer Slide Time: 03:36)

The screenshot shows the RStudio interface. The left pane displays an R script with the following code:

```

288 gria()
289 # scatterplot with labeled points
290
291 dffb[dffb$Price>70,]
292 dffb[dffb$Price<12,]
293 dffb[dffb$KM<150,]
294
295 dffb=dffb[-c(13,23,29,65,73),]
296
297 range(dffb$KM)
298 range(dffb$Price)
299 plot(dffb$KM, dffb$Price, xlim = c(25,120), ylim = c(1,9),
300      xlab="KM", ylab="Price", panel.first = grid())
301 text(dffb$KM, dffb$Price, dffb$model, adj = c(-0.4,-0.4),
302      cex = 0.5)
303
304 (Top Level) §

```

The right pane shows the "Environment" tab with the following data:

- Data:
 - dff1: 78 obs. of 9 variables
 - dffb: 79 obs. of 9 variables
 - dffb: 74 obs. of 12 variables
- Values:
 - Age: num [1:79] 4 5 6 4 9 5 6 3...

So, these points I will try to get rid of and then will look at them. So, this particular plot this is scatter plot is between a kilometre and price. So, let us look at the a new range. So, you can see that these this particular range 27 to 100 and 13.

(Refer Slide Time: 03:59)

The screenshot shows the RStudio interface. The left pane displays an R script with the following code:

```

288 gria()
289 # scatterplot with labeled points
290
291 dffb[dffb$Price>70,]
292 dffb[dffb$Price<12,]
293 dffb[dffb$KM<150,]
294
295 dffb=dffb[-c(13,23,29,65,73),]
296
297 range(dffb$KM)
298 range(dffb$Price)
299 plot(dffb$KM, dffb$Price, xlim = c(25,120), ylim = c(1,9),
300      xlab="KM", ylab="Price", panel.first = grid())
301 text(dffb$KM, dffb$Price, dffb$model, adj = c(-0.4,-0.4),
302      cex = 0.5)
303
304 (Top Level) §

```

The right pane shows the "Environment" tab with the following data:

- Data:
 - dff1: 78 obs. of 9 variables
 - dffb: 79 obs. of 9 variables
 - dffb: 74 obs. of 12 variables
- Values:
 - Age: num [1:79] 4 5 6 4 9 5 6 3...

At the bottom of the console, the output shows the removed points:

```

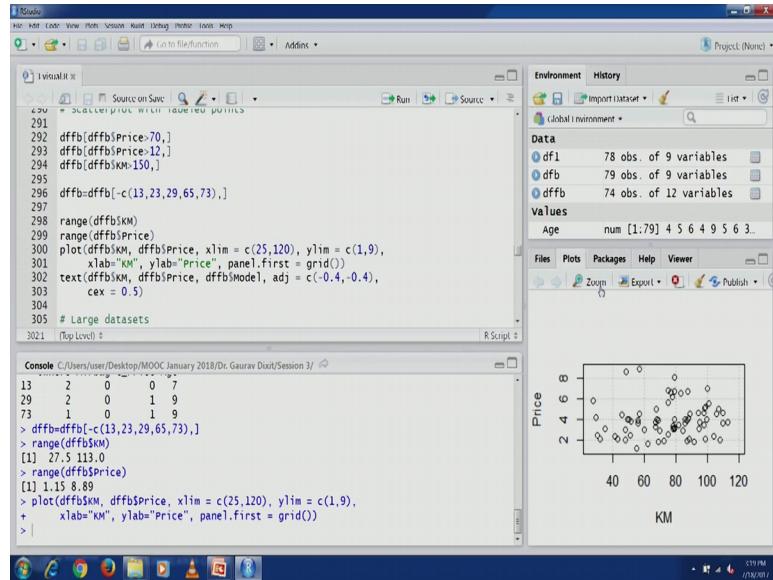
300 73 Mahindra Scorpio M-Hawk 2008 Diesel 11.30 167.000 4.1 1
Owners Airbag_C_Price Age
13 2 0 0 7
29 2 0 1 9
73 1 0 1 9
> dffb=dffb[-c(13,23,29,65,73),]
> range(dffb$KM)
[1] 27.5 113.0
> range(dffb$Price)
[1] 1.15 8.89
> |

```

And the points that we have removed they were slightly far away you can see these numbers 161, 156, 167. So, these were the far away points. So, therefore, you wanted to get rid of this point. So, that our labelling is much better. Similarly for price variable also we have been able to get rid of the other points. This 0.23 and 65 you can see that price

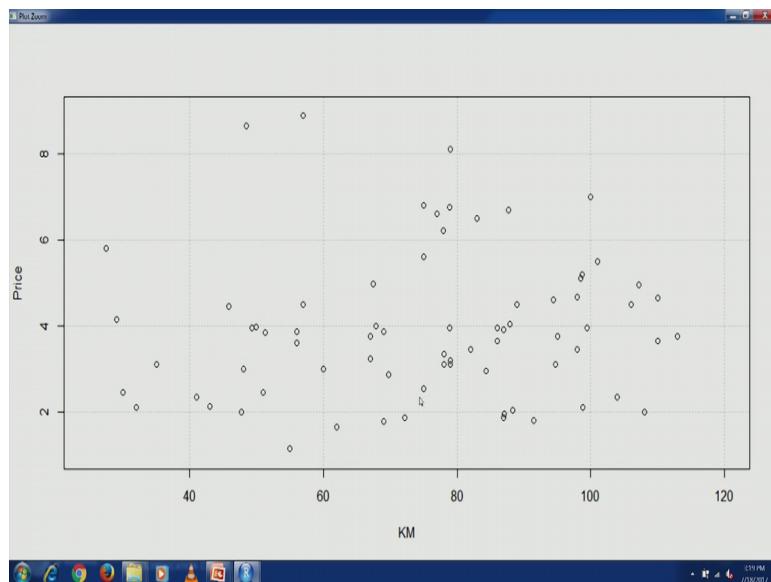
value for these 2 points 72 and 13.55 these 272 is outlier and 13.5 5 is also far away from the major chunk of values.

(Refer Slide Time: 04:42)



Let us execute the this particular code and create scatter plot. So, these are the points between price and kilometre.

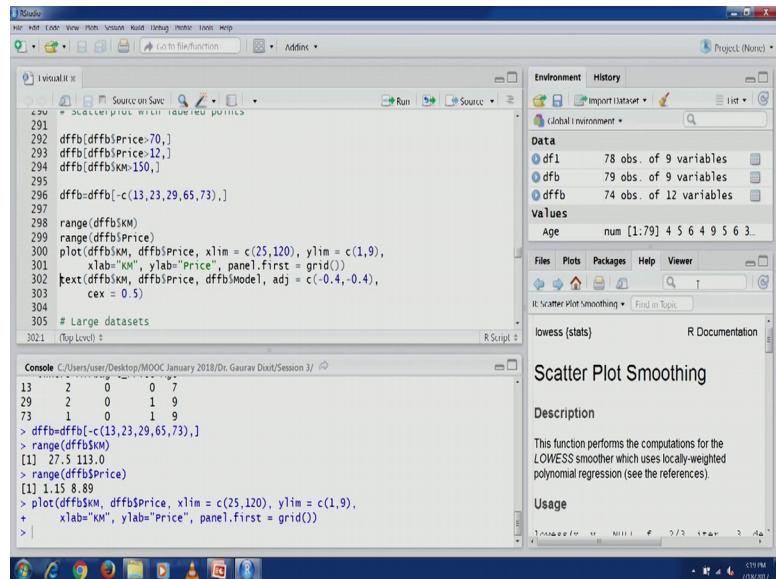
(Refer Slide Time: 04:52)



Now if you want to find out if you want to label these observations. So, this is the text is the command which can actually be used.

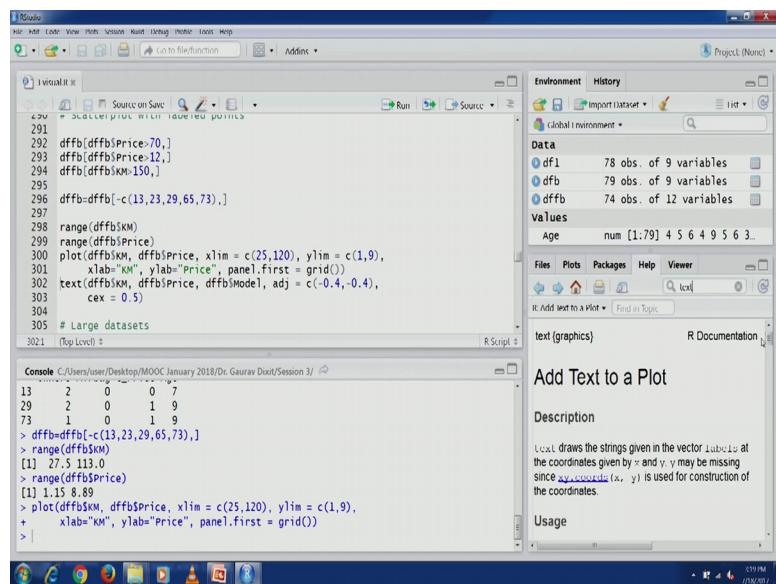
So, in the text command if you are interested in knowing more about this particular command.

(Refer Slide Time: 05:10)



And you can go into the help section.

(Refer Slide Time: 05:14)



And find out more information about this particular function.

(Refer Slide Time: 05:19)

The screenshot shows the RStudio interface with the following details:

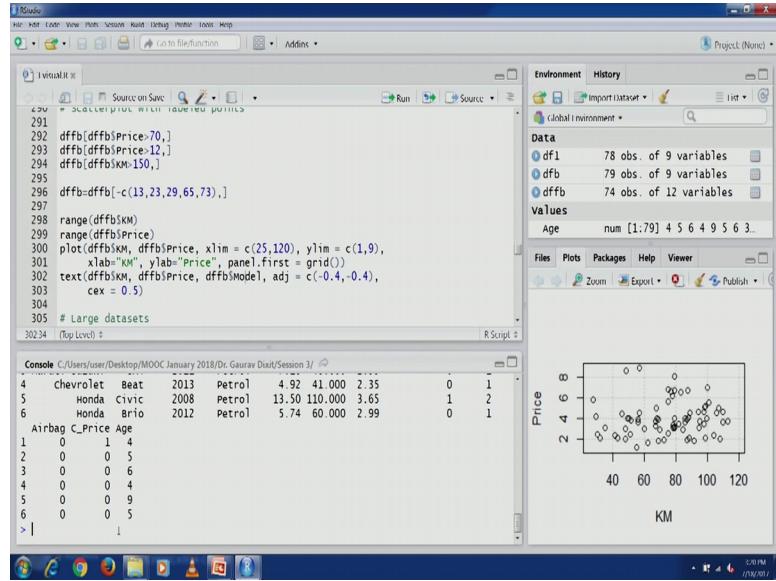
- Environment:** Shows objects dff1, dffb, and dffb.
- Data:** Shows the structure of dff1, dffb, and dffb.
- Values:** Shows the range of the Age variable.
- Console:** Displays the R script and its output. The script includes filtering data (dffb[dffb\$Price > 70,]), setting plot ranges (range(dffb\$KM), range(dffb\$Price)), creating a plot (plot(dffb\$KM, dffb\$Price, xlim = c(25, 120), ylim = c(1, 9), xlab = "KM", ylab = "Price", panel.first = grid()), text(dffb\$KM, dffb\$Price, dffb\$model, adj = c(-0.4, -0.4), cex = 0.5)), and a note about large datasets (# Large datasets).
- Help:** Shows the help documentation for the text function, which is used for labeling points in the plot.

You can see x and y the coordinates are first and second arguments. So, in this particular case KM and price KM on the x axis and price on the y axis, they are representing the x and y coordinates.

Their limits have been appropriately specified you can see a limits on x axis 25 to 120 this is again based on the range can range calculation that we just did. You can see the range this is within this particular within these particular limits. Similarly for y axis also limits are specified as 1 to 9 and you can see the range for price variable is also lying within this range.

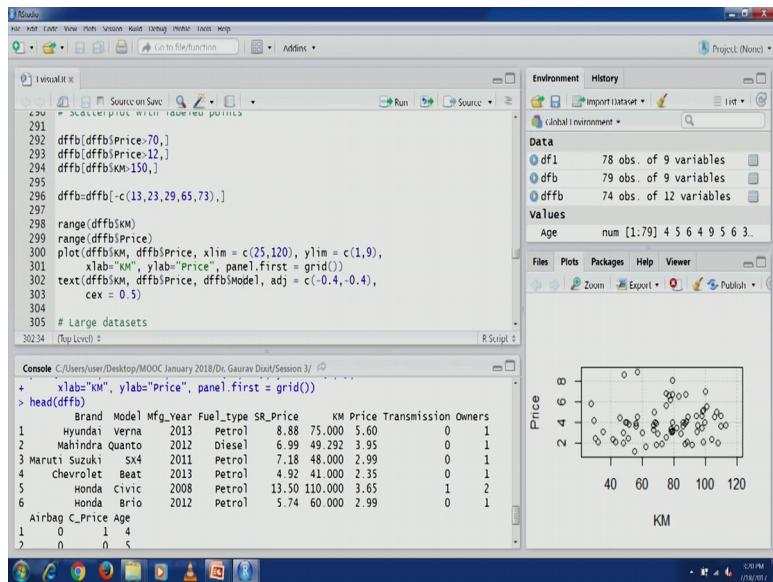
So, therefore, this plot has been generated and now let us executes the line to label this now labelling is based on the variable model. So, the model name of the car would actually be the label for the points, if you are interested in relooking at the data. So, let us look at the first 6 observations.

(Refer Slide Time: 06:21)



Let us scroll you can see the second variable model.

(Refer Slide Time: 06:27)



So, the name for each of the observation is going to be based on this variable for example, Verna Quanto SX 4 beats civic. So, the points are going to be labelled with their respective model name and then there are some adjustments. So, for every point where we want to place this particular label so that is based on this particular r argument adjustment a d j.

So, a minus .4 and minus 0.4 are the relative coordinates from the points. So, from the x and y coordinates of the actual point and relative to that coordinate this adjustment is to be done were the label is going to be plotted. And the expansion for and this expansion for this for this label point is going to be 0.5 this is half of the default size.

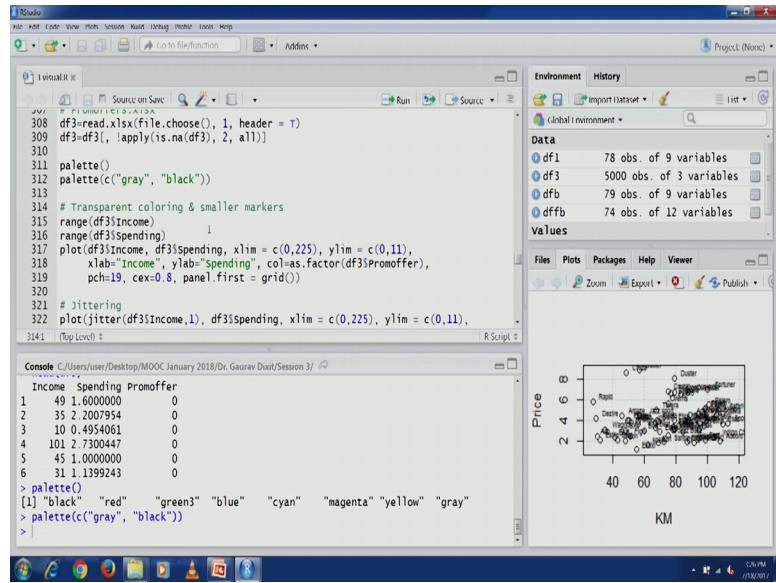
So, let us execute this line you would see the plot is slightly messy now, but every point is labelled by their model name. So, all the used cars you can see they are labelled by their label names. Now from here again you can see that on the upper side of, upper half of this particular rectangle this particular plotting region you would see slightly expensive cars are there. For example, rapid, crews Duster, so these cars Fortuner.

So, these cars are in the upper side Verna, so these car cars on the upper side of price. So, this is very understandable. Now you would see that in the lower in the in the right part right rectangle you would see the cars like which are having slightly higher mileage. So, those cars are there for example, desire Verna indigo. So, all these cars you would see on this side. So, these cars are probably being used have accumulated more kilometres. So, this kind of this kind of clustering this kind of understanding of a data can also be helpful if the points are labelled.

So, let us move to our next point now large data sets. So, if we are dealing with the really large data set what is going to happen if we start our visual analysis? The our plot would be filled with many points because they are we might be dealing with 3000, 4000, 5000 or even 10 000 points.

So, therefore, with the whole plot if we try to a you know generate a scatter plot the plot might be filled with too many points. So, how do we understand the relationships between variables or some information overlap? So, those kind of in how we can find out. So, there we have a few things that can be done while we generate our plots. So, that the we can easily see and understand the data. So, this is 1 example that 1 particular data set promotional offers that we are going to use.

(Refer Slide Time: 10:13)



So, let us import this data set, you can see this particular variable this particular data frame d f 3 with 5000 observation this. So, this particular data set is can will be in the large data set category and there are just 3 variables, but 5000 observation. Let us have a look at this particular data set first 6 observations. So, this particular data set has a 3 variables 1 is income then other one being spending and the promotional offer.

So, these 3 variables are about customers the income of a particular customer and the spending that they do. So, income and spending and then the promotional offer, whether the whether they accepted or rejected the promotional offer that was sent to them. So, these we have these 3 variables and using these 3 variables we will try to see how we can actually visualize this large data set and then try to understand. So, again here the our colour scheme is going to be slightly important in this case.

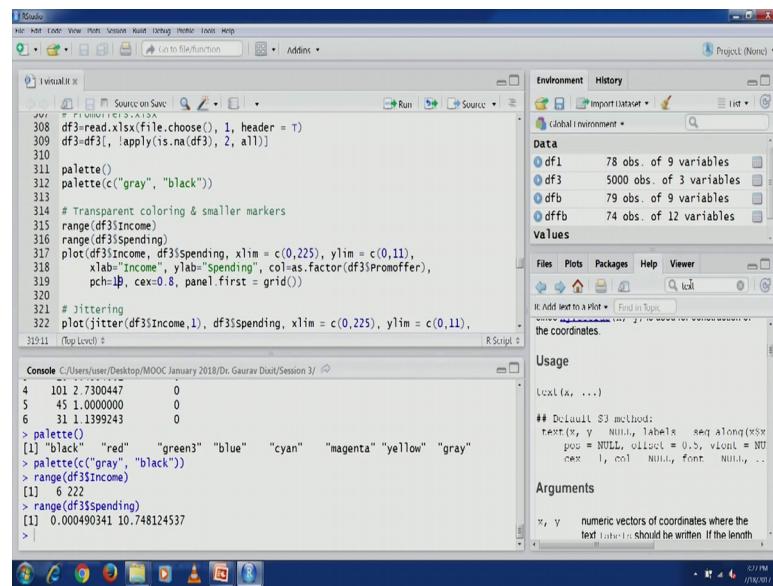
So, let us check our default colour scheme you can see the black red this is the default colour scheme in r we can change it change it to a grey and black. So, our first colour of choice is going to be grey and then black. So, let us change it now when we are dealing with a large number of data points then, we will have to in cooperate some you know transparent colouring and will also have to reduce our marker size. So, the a points that we generally see in our plots.

So, for example, you can see these circle these circles that we see in this particular scatter plot. So, the marker size is slightly on the higher side for this particular plot. So,

therefore, if we are dealing with too many points let us say for example, in this example 5 000 points therefore, will have to reduce the marker size and will also have to do a few more changes we will see.

So, let us so this particular plot scatter plot that we are going to generate this is between income and spending income on the x axis and spending on the y axis. So, let us check the range 6 to 222 and for a spending is closer to 0 and then almost 11. So, this is the range the limits have been specified appropriately and the labels and the colour, you can see the promo offer the third variable has been used as a colour and you would also see a plotting character is 19 has been specified, if you are interested in knowing more about plotting character. So plotting character is the 1 that is actually used to plot a particular graph. So, you can see circles have been used in this scatter plot.

(Refer Slide Time: 13:22)



The screenshot shows the RStudio interface. The left pane displays an R script with several lines of code. The right pane shows the environment and history panes, and the bottom pane shows the console output.

```

#> library(ggplot2)
#> df1<-read.xlsx(file.choose(), 1, header = T)
#> df3<-df1[,apply(is.na(df1), 2, all)]
#>
#> df1
#> palette()
#> palette(c("gray", "black"))
#>
#> # Transparent coloring & smaller markers
#> range(df3$income)
#> range(df3$spending)
#> plot(df3$income, df3$spending, xlim = c(0,225), ylim = c(0,11),
#>       xlab="Income", ylab="Spending", col=as.factor(df3$Promoffer),
#>       pch=19, cex=0.8, panel.first = grid())
#>
#> # jittering
#> plot(jitter(df3$income), df3$spending, xlim = c(0,225), ylim = c(0,11),
#>       xlab="Income", ylab="Spending", col=as.factor(df3$Promoffer),
#>       pch=19, cex=0.8, panel.first = grid())
#>
#> df1
#> palette()
#> palette(c("black", "red", "green3", "blue", "cyan", "magenta", "yellow", "gray"))
#> palette(c("gray", "black"))
#> range(df3$income)
#> [1] 6 222
#> range(df3$spending)
#> [1] 0.000490341 10.748124537
#>

```

If you are interested in finding out, finding about more about plotting characters you can search in the help page, help section and in the a point function, you would see there is a more description about plotting characters, you can see p c h values.

(Refer Slide Time: 13:50)

The screenshot shows the RStudio interface. The top-left pane displays R code in a script editor:

```
308 df3<-read.xlsx(file.choose(), 1, header = T)
309 df3<-df3[, !apply(is.na(df3), 2, all)]
310
311 palette()
312 palette(c("gray", "black"))
313
314 # Transparent coloring & smaller markers
315 range(df3$Income)
316 range(df3$Spending)
317 plot(df3$Income, df3$Spending, xlim = c(0,225), ylim = c(0,11),
318 xlab="Income", ylab="Spending", col=as.factor(df3$PromOffer),
319 pch=19, cex=0.8, panel.first = grid())
320
321 # jittering
322 plot(jitter(df3$Income), df3$Spending, xlim = c(0,225), ylim = c(0,11),
323 xlab="Income", ylab="Spending", col=as.factor(df3$PromOffer),
324 pch=19, cex=0.8, panel.first = grid())
325
```

The bottom-left pane is the R Console, showing the results of the executed commands:

```
4 101 2.7300447 0
5 45 1.0000000 0
6 31 1.1393243 0
> palette()
[1] "black"   "red"     "green3"  "blue"    "cyan"    "magenta" "yellow"  "gray"
> palette(c("gray", "black"))
> range(df3$Income)
[1] 6 222
> range(df3$Spending)
[1] 0.000490341 10.748124537
>
```

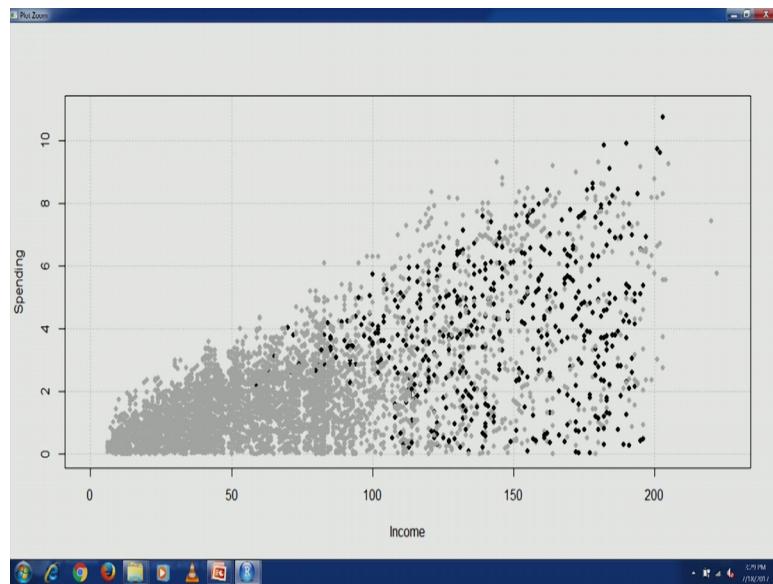
The right side of the interface includes the Environment pane showing data frames df1, df3, dfb, and dfbf, and the Plot pane displaying various plotting symbols.

So, these are first 0 to 18, 0 to 18 and even more 0 to 25 they are well defined plotting characters. For example, what we saw in our scatter plot was similar to the p c h value of 1, currently what we are going to use in this particular scatter plot is the plotting character of value 19. So, that is this this dot black dot.

So, let us use these particular plotting characters cex is again the character expansion factor pointed. So, this is going to be 80 percent of the default size. Now we in the previous lectures we talked about how we can generate grid behind a plot? Now this this another way that we are going to use to generate grid.

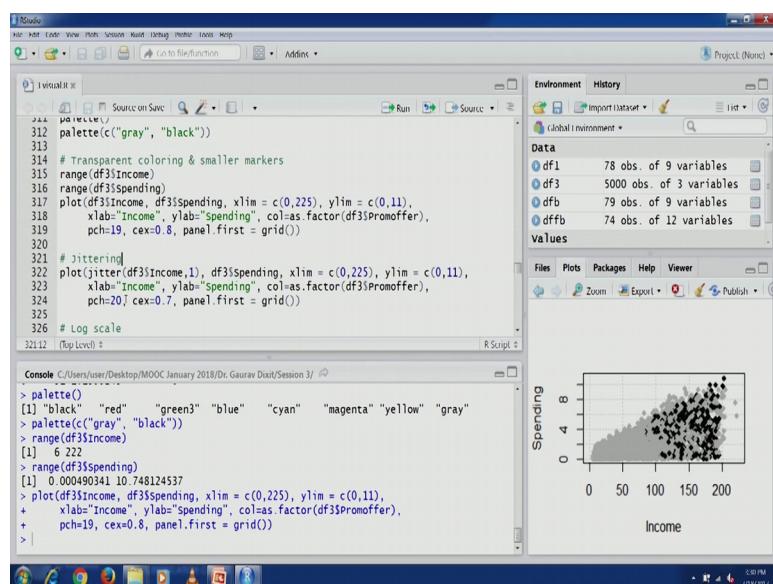
So, in the plot function it itself you can use this particular argument panel dot first and there you can pass on the grid function, which is going to generate grid for you grid behind your plots. So, let us execute this line.

(Refer Slide Time: 14:59)



So, this is our plot now, if you visualize this particular plot too many points the clarity is lost because most of the points they are in the same region. So, therefore, they are overlapping each other. So, this is what happens when we are dealing with large data set. So, it becomes slightly difficult for us to understand, what is going on here between these 2 variables. You can also see the marker size is also playing a role because of this slightly higher marker size many points are overlap, which could have been avoided with a slightly smaller marker size. So, let us do few changes. So, there is another concept which is called jittering.

(Refer Slide Time: 15:40)



Were, what happens in jittering to avoid overlap between points we add some random noise very small value, very small value compared to the actual value of that point will we generally add some random noise. So, that 2 points which are overlay overlapping they might be slightly closer to each other, but both of them could be visible instead of overlapping each other.

So, this jittering is generally added to each point. So, in this particular case you would see that it is the income variable which we have selected for jittering. So, income the so this particular x axis points. So, there has been some noise there is going to be added to all the x values for all the points. You would also see we have changed the plotting character if we go back to the help section you would see that p c h value of 20 here is actually for a smaller dot.

So, therefore, a smaller marker is going to be used and you would also see that character expansion also we have reduced it to 0.7. So, therefore, this is going to be 70 percent of the default size. So, we have reduced the character expansion we have changed the marker size and we have also done jittering which is adding the random noise, so, that the overlap could be avoided.

So, let us execute this line let us see this plot now you would see that many more points can be seen. So, there is much less overlap the marker size has reduced. So, therefore, points you know they are now of the size is very small, but much less over overlap between points. And most of the points many more points many more points in comparison to the previous graph can be seen here.

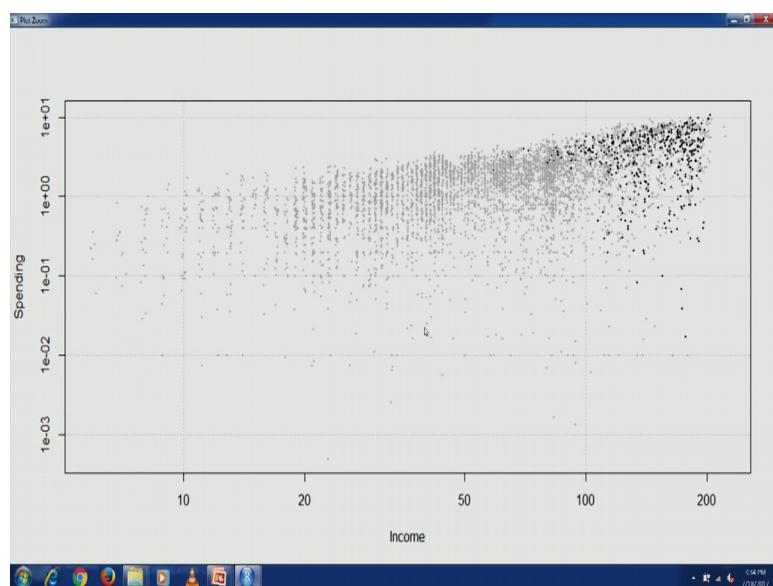
Now, you would see that most of the most of the customers who have actually accepted the promotional offer they seem to be lying on this region they seem to be lying on the right and then the upper part top right mainly. Now there is another way to further improve these particular plots. So, that is something that we have discuss in the previous lecture as well as log scale.

So, we can transform our x our both these scales x and y scales and log is logarithmic scales could be used and therefore, that is further going to improve the our visibility of the points. If you look if you again have a look at the plot you would see that this particular region between 0 to 100 this is a more messy there are more number of points

in this region, the other region is slightly you know there is some more space the points are slightly more away.

So, therefore, visibility of points is much better in this case this is bit region of 0 to hundred on x axis this is slightly more messier. So, we will try to change the scale and try to see what can be done. So, you would see that we have used the log argument in the plot function now x or y. So, scaling of both the axis is x is going to be perform and the other things remaining same colour again promo offer is same is being used plotting character is same and character transmission is also same grid is going to be there. So, let us execute this particular line and you would see a significant change in the plot this is mainly because of the change in scale.

(Refer Slide Time: 19:42)



Now, you would see that because this being log scale you would see the points this space from space between point space between points having x value are 0 to 100 it is much more and 100 to 100 less space has been the there less space is there this is, because of the usage of log scale similarly for y axis as well.

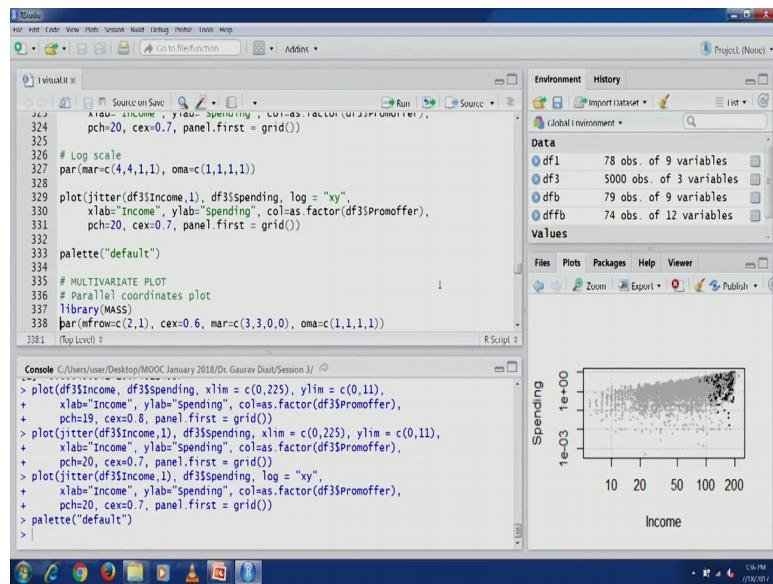
So, therefore, now most of these points they are more spaced this region of points they are more spaced and therefore, visibility of these points is these point is actually improved. So, clustering could be so this kind of a jittering and a scaling of scales axis could really be helpful, when we are trying to visualize a large amount of data.

And it can actually help us especially in unsupervised learning method task specially clustering and also sometimes for to understand the relationship between 2 variables specially when there are too many points. So, let us reset the our colour scheme for R, Now that brings us to our a next discussion points. So, we are now going our to start our discussion on multivariate plots.

So, we are going to discuss a few of multivariate plots where not just 2 variables are more than 2 variables are going to be used and because generally the kind of modelling that we do, whether it is for a statistical modelling or data mining modelling generally we are dealing generally we are in multivariate environment.

So, therefore, multivariate plots can sometimes be more useful for us to gain some more insights. So, we are going to start our discussion on multivariate plots.

(Refer Slide Time: 21:49)



So, first 1 that we are going to cover is parallel coordinates plot. So, will see later on as we go through this particular example about parallel coordinates plot is about. So, for different variables we can assume different dimensions are there and for all those dimension all those dimensions are actually given some space in our 2 D plots.

So, will see how that is done and that gives us a better picture of each observation in specially in parallel coordinates plot. And we can help understand what is happening

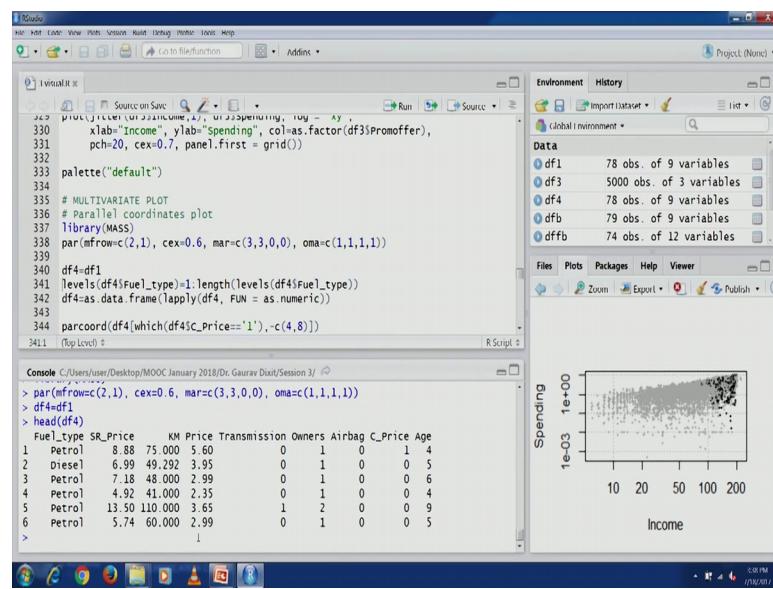
between the data in the what is happening the kind of relationship between variables and the observations as well.

So, too able to use to create parallel coordinates plot we need to reload we need to load this particular library mass. Then will also change because we would be creating a 2 back to back plots.

So, let us also change the you know parameters 2 rows, 2 rows 1 column. We also changing this cex value margins and outer margins as well, some margins are specifically for the plotting region and outer margins are these space between the plot and the remaining a area. So, let us execute this.

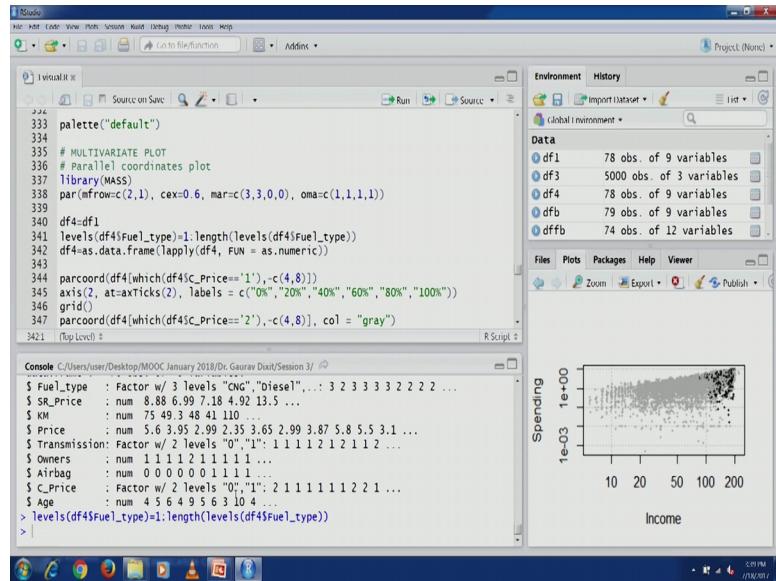
So, for this particular plot we are going to use again d f 1 let us have a look at the data frame.

(Refer Slide Time: 23:43)



So, this is main data frame that we are going to this is again used cars data set. So, what we are going to do is we need to do certain a transformation to be able to use the parallel coordinates plot. So, if we look at the data that we have right now look at this structure.

(Refer Slide Time: 24:03)



Let us look at the structure you would see that fuel type and transmission and C price all of them they are factor variable, but a parallel coordinates this function pair code that we are going to use now this actually requires us all the variables to be numeric.

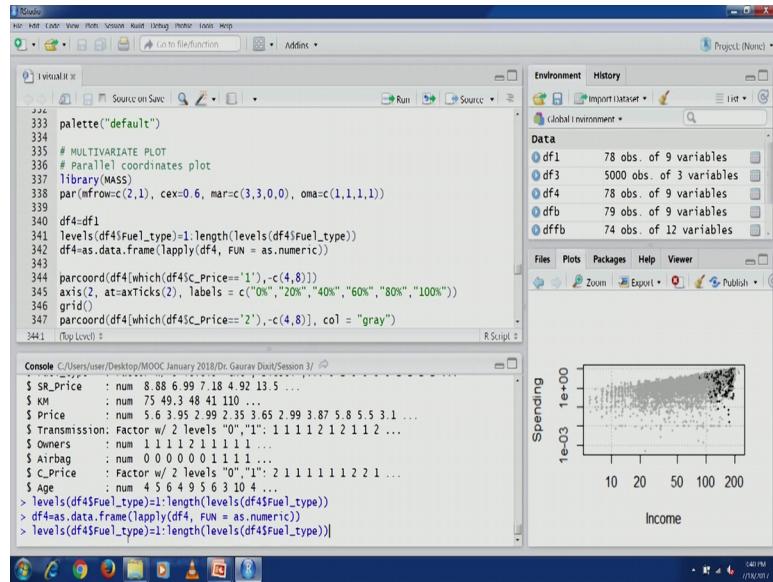
So, therefore, we need to change them. So, you would see a this is one way. So, all the labels, so fuel type labels, we are trying to make them numerical for example, right now they are labels for fuel type they are CNG diesel and petrol. So, first thing would be to change them to numbers. So, this is the this is the particular code that could be used a length because there are 3 labels.

So, therefore, we can have a 1 2 3 instead of CNG diesel and petrol. So, this is the code that can be done to change the labels name. So, once the labels name have been changed for the fuel type we are particularly ready to apply as dot numeric as dot numeric function to all the variables in this particular data frame.

Earlier even though we had 2 more factor variables transmission and C price, but their labels name were already in the numeric form 0 and 1 and 0 and 1. So, therefore, when we course when we do the convergent for convergent of these strings to a numeric, it is easier, because they are already in the numbers form you know written in the stored in the string form.

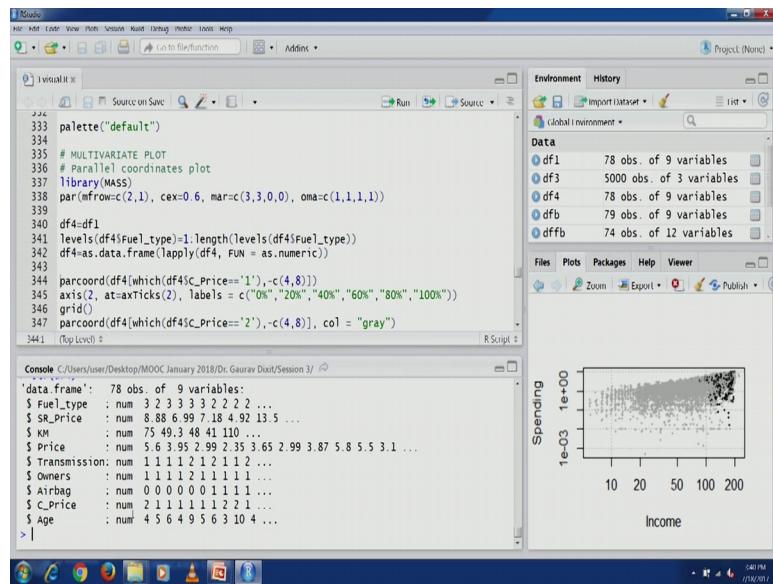
So, therefore, a, but this particular variable fuel type that was stored using CNG diesel is a texture form. So, the name of the labels they were text. So, therefore, for that we need had to change it. So, let us change this particular function, let us change this particular data frame.

(Refer Slide Time: 25:58)



Let us have a relook at the structure.

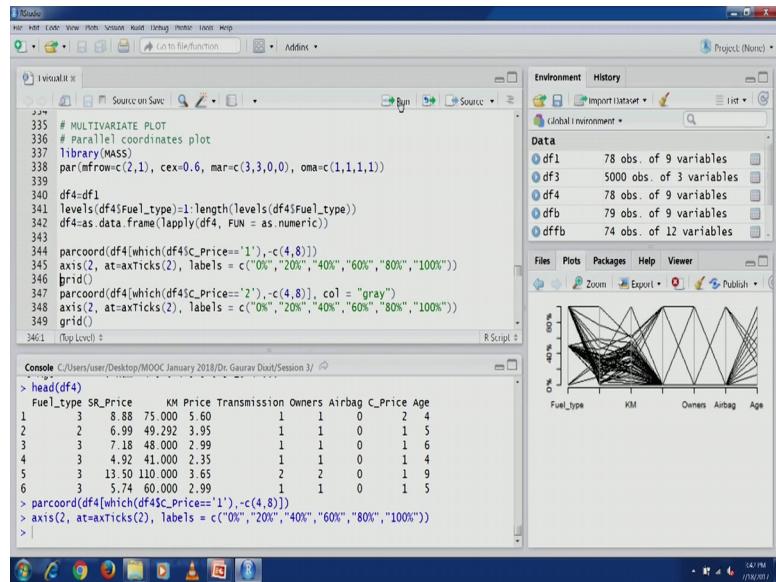
(Refer Slide Time: 26:00)



Now, you would see all the variables are now numerical and you would also see that labels have also changed, you know the values have also changed specially for fuel type

you would see same is true for this transmission and C price let us have a look at the first 6 observation.

(Refer Slide Time: 26:25)



You would see transmission earlier for 0 and 1 now you can see it is 1 and 2 this is main reason being, because we have applied numeric function. C price also earlier it was 0 and 1, but now it is 1 and 2 fuel type it was earlier you know text that was CNG petrol diesel now it is 1 2 and 3.

Now, once we have done this kind of transformation, now we are ready to use this particular function par coordinate, now here we are not going to include our outcome variable or interest. So, there are 2 variables price and C price scatter plot price is not going to be included price and C price is not going to be included in this particular plot we are interested in. So, we are going to use 2 panels and each panel for particular group of C price. So, for C price value 1, 1 panel 1 particular plot, 1 panel and the second panel for C price value of 2.

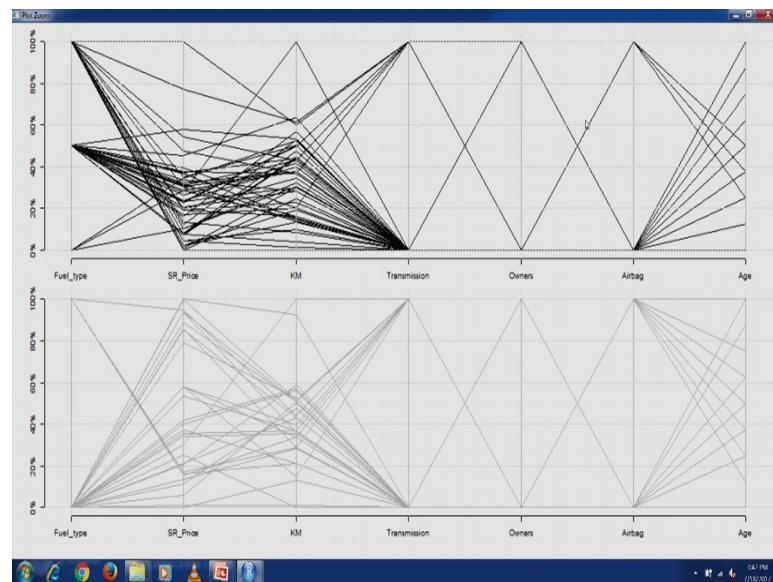
So, that is the used cars used cars with less than 4 lakh value. So, they are going to be in the panel 1 and the used cars with more than 4 lakh more than or equal to 4 lakh value they are going to be in the panel 2.

So, will try to understand the differences between these 2 groups across variables, so, this is going to be multivariate visual analysis. So, let us execute this line you can see that panel 1 has been created now let us label the axis.

Now you would see from here that par coordinate function it also scales all the variable into 0 and 100. So, that is in percentage. So, all the variables have been scaled have been brought to the same scales let us also create grid.

Now, let us plot the second panel, this is in different colour and labels and grid. Now let us zoom and find out the plot.

(Refer Slide Time: 28:23)



So, panel 1 this is for group 1, the panel 2 this is for group 2. Now from here we can actually compare these 2 panels and we can try to understand the differences between these 2 groups. So, if you can see transmission if we look at the transmission then you would see the panel 1; the panel 1 there are 2 values that are there in transmission. So, we had manual and automatic.

So, there are 2 values you would see in the panel 1 there are few values at the value 1 right. There are few values at value 1 and if you see here in the panel 2. So, the values transmissions for both these values there are they are seem to be equal number of observation.

So, equal number of equal number of lines are passing through this particular axis, for each variable Fuel-type, SR-Price, KM, Transmission, Owners for each variable we have an axis. So, that is why the name comes from this parallel coordinates. So, for each variable we have a you know coordinate system and they are they have been put in parallel. So, each observation each line is representing in observation going through this particular plot.

You would see fuel type if you look at the fuel type dimension you would see in the panel 1 petrol CNG and diesel all 3 are present, but if you see in the panel 2 only petrol and diesel are present, 1 1 is 1 is not there 1 particular category is not present. So, this kind of this kind of comparison can be done using parallel coordinates plot.

(Refer Slide Time: 30:47)

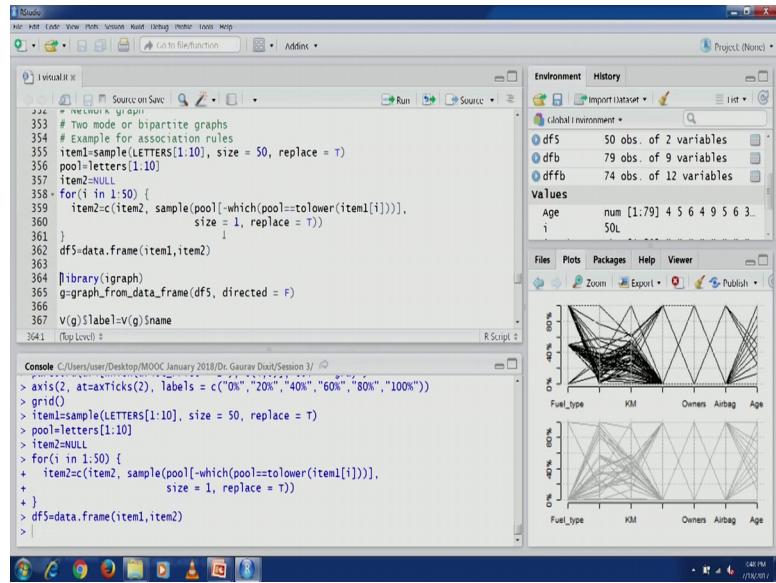
The slide has a dark blue header and footer bar. The main title 'VISUALIZATION TECHNIQUES' is centered in white font. Below it is a white content area containing a bulleted list of visualization types. At the bottom, there is a dark blue footer bar with the IIT Roorkee logo, the NPTEL Online Certification Course logo, and the number '12'.

- Multidimensional Visualization
 - Trend Lines
 - In-plot labels
 - Scaling Up
 - Multivariate plots
- Specialized Visualization
 - Network graphs
- Open RStudio

Now let us go back. So, we are going to start now our discussion on next point that is specialised visualization. Till now what we have been doing is we have been mainly dealing with the cross sectional data or the time series data. Now we are going to incorporate some other forms of data. For example, in this particular lecture we are going to cover network graphs. So, for that we require network data.

So, will go through 1 example and see how it is different from cross sectional and data cross section analysis and cross sectional data and time series data and time series analysis. So, this is this 1 hypothetical example that I have created.

(Refer Slide Time: 31:28)



This is mainly applicable in the association rules context that is going to be covered in a much later lecture. So, this is a bipartite graphs 2 more graphs. So, there are going to be 2 groups and we are going to see the interconnections between these 2 groups by plotting a network graph.

So, first what I am going to do is because this is mainly in the association rules context. So, therefore, we are essentially dealing with the transactions. So, therefore, in transaction generally we have items which are purchased together. So, we are going to create a hypothetical data set for the same.

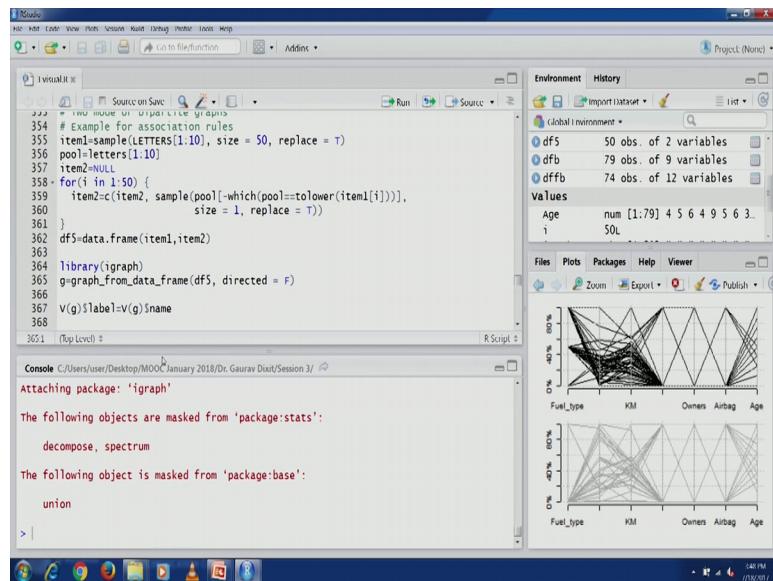
So, let us say some items 1 to 10 they are represented by letters a to the corresponding 1. So, let us create this particular you would see that item 1 has been created 50 observations and these and the labels 1 to 10. And now this is for the first transaction this is for the first item in a transaction. Second item in a transaction again the again we are going to it is going to be from the same pool of items, but it cannot be the item that has been already purchased.

So, therefore, we are going to we have written some code here to perform that. So, first let us let create these this particular pool this is 1 to 10 you can see here pools variable has been created 1 to 10 a to this particular value.

Now, what we are going to do we are going to for a particular for a particular item in item 1 the it cannot be included in the item 2. So, therefore, it is eliminated through this code you would see minus which pool to lower item i item 1 i. So, item 1 i, in the upper case we lower it down and it is it should not if it is equal, then that particular index is excluded from the pool samples from the pool from which this sampling is being done.

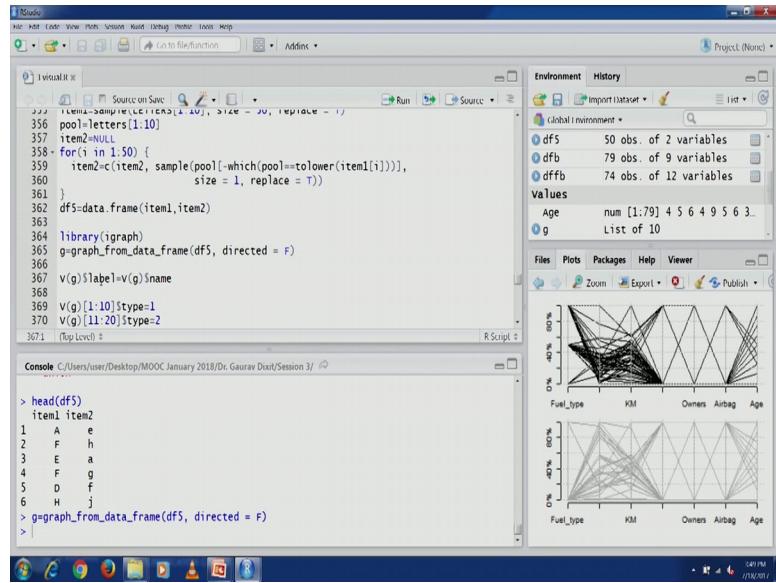
So, let us execute this code now let us create the data frame of these 2 variables, this I graph is the library that we generally required to deal with network data. So, let us load this particular library.

(Refer Slide Time: 34:03)



Now, from the data frame that we have just created let us have a look at the data frame as well.

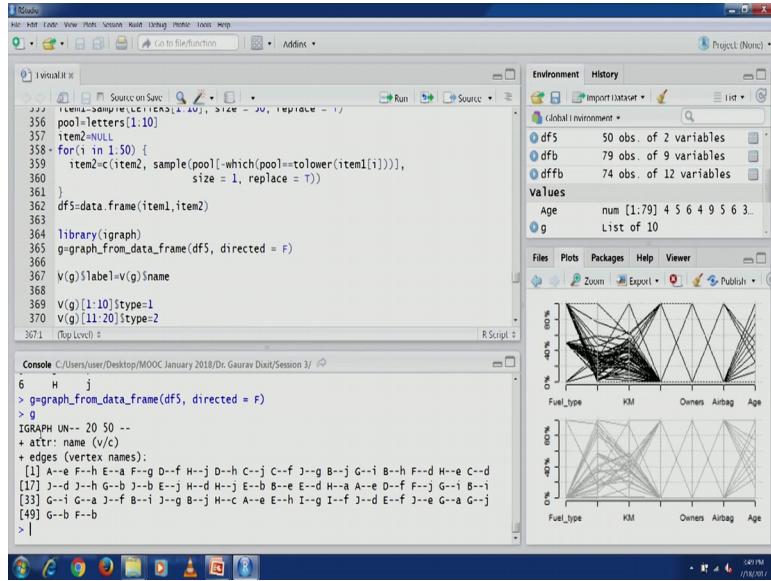
(Refer Slide Time: 34:11)



So, you can see first 6 observation items name. So, we can consider this to be 1 transaction, row number 1 is 1 transaction were items A and e are what purchase together, row number 2 is second transactions. So, these transactions base data set is mainly applicable to association and rules. So, now, let us move on.

So, from this particular data frame will try to create a network graph data. So, this is the function graph from data frame that can be used, if you need to pass on the a data frame and the this we need to set this directed argument. So, in this case we are not trying to create a directed graph. So, therefore, this is this has been set as false. So, let us execute this line. Now V is for vertices of a graph now labelling of those vertices. So, till now what we have done we have created graph. So, if you want to see what we have done.

(Refer Slide Time: 35:09)



You can see this graph has 20 vertexes and 50 edges and those ages have been displayed here right. So, now, let us try to label these vertices. So, labelling could be done using their name itself. So, let us execute this.

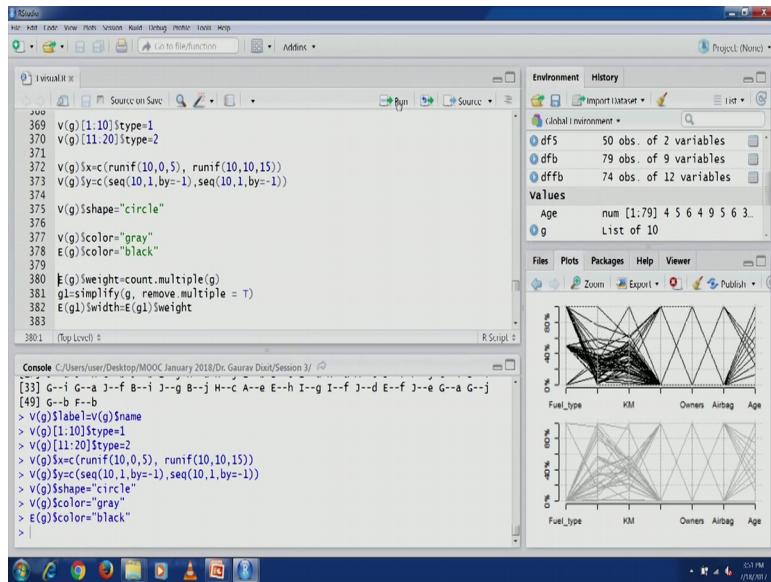
Now there are 2 groups as because 1 is what you know in a particular transaction first item and the second item. So, we are trying to put them into different groups. So, 1 to 10 is 1 is type 1 and 11 to 20 vertices as they are type 2. So, let us execute, we are trying to understand what generally happens in association rules? What course with what? If item a is purchased then, whether item b is purchased or not. So, in that kind of association we can see here through a network graph here, if in a particular group item 1 item is purchased then whether another item is purchased along with that. So, we are going to find out that through network graph.

So, we have created 2 groups then a randomly we are trying to create the coordinates were these were these vertices are going to be plotted. So, this is for the creation of layout coordinates. So, this is again randomly being done. So, x and y coordinates for all the vertices have been created. Now shape of the vertices we have selected right now circle colour grey. Now we come to our edge part. So, edge edges the colour has been selected black.

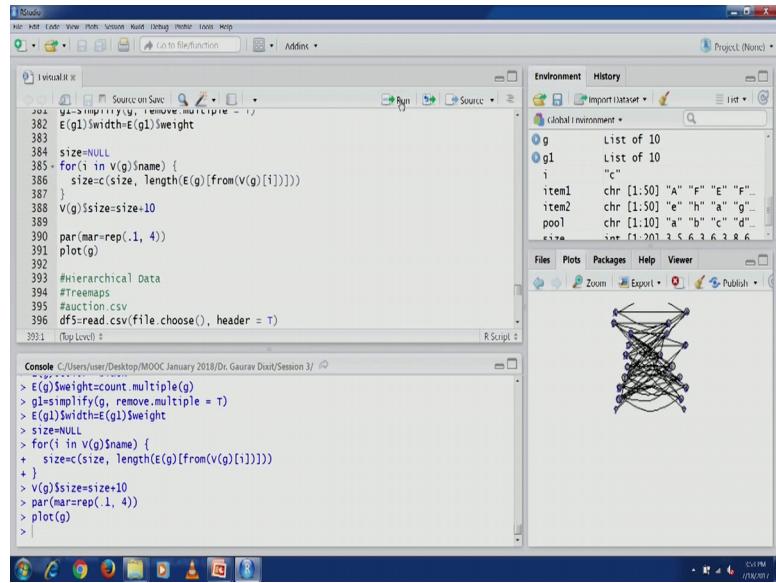
Now sometimes there might be multiple lines between 2 particular items, because 2 particular items can be bought by more than 1 customer. So, therefore, there could be

more transaction of that kind. So, we are trying to represent more number of transaction through a through the edge width.

(Refer Slide Time: 37:07)

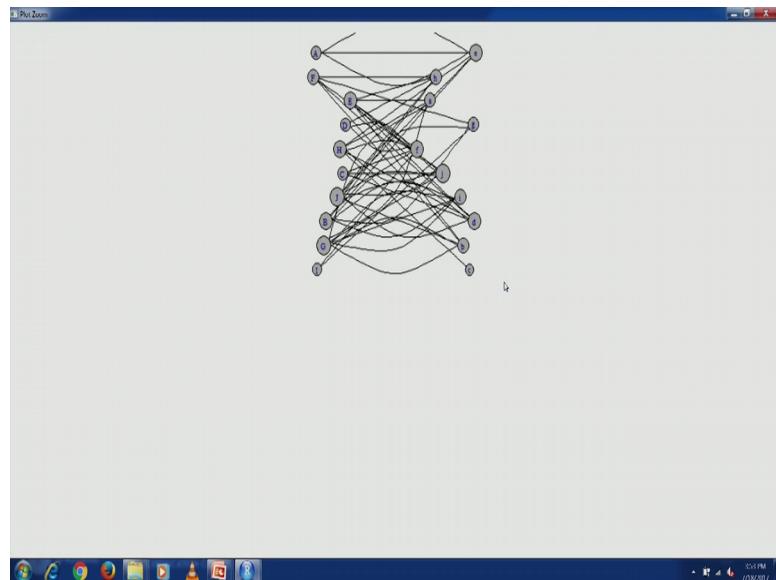


(Refer Slide Time: 38:26)



This particular network graph has been created

(Refer Slide Time: 38:48)



And you can see these are this is the 1 group 1 this is the first item that has been that was purchased by different customers, a f e and the second item that was purchased by the same customer for each transaction we can see and there is difference in the a line width that is actually signifying the a more number of transactions.

If that particular item has been purchased and the another item has been purchased in more transaction. So, that is reflect in line with the bigger size of the vertex that is reflecting the involvement of that particular item in more number of transactions.

So, therefore, network graph can really be help helpful in association rules, while we are trying to understand the relationship between different items. So, we will stop here in the next class will start with hierarchical data.

Thank you.