

Dealing with Materials Data: Collection, Analysis and Interpretation
Professor M.P. Gururajan
Professor Hina A. Gokhale
Department of Metallurgical Engineering and Materials Science
Indian Institute of Technology, Bombay
Lecture No. 15
R libraries for plotting

Welcome to dealing with the materials data. In this course we are going to look at the collection, analysis and interpretation of materials data. We are in module one which is an introduction to R and we are learning how to get data into R and manipulate it specifically plotted. And in this process I want to now talk about specific plotting libraries that are available.

(Refer Slide Time: 0:45)

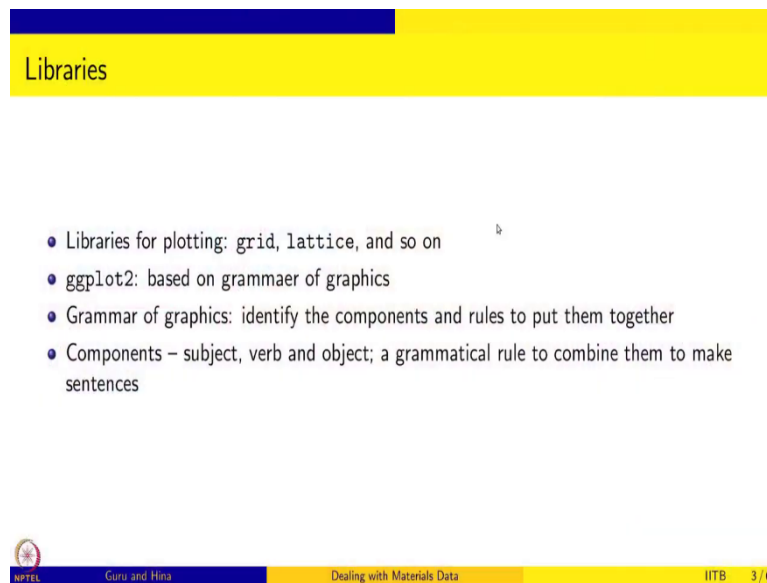
Module: Introduction to R

Plotting libraries

NPTEL Guru and Hina Dealing with Materials Data IITB 2 / 6

And so, there are R can do a plotting on its own as we have seen we can use the plot command and do plotting. But sometimes it is useful to use some of the other libraries that are available in R, they are very powerful and they also give you lots of handle on how to go about plotting. Specifically we are going to use the ggplot2 library. So, in this session we are going to learn a little bit about ggplot2.

(Refer Slide Time: 1:14)



Libraries

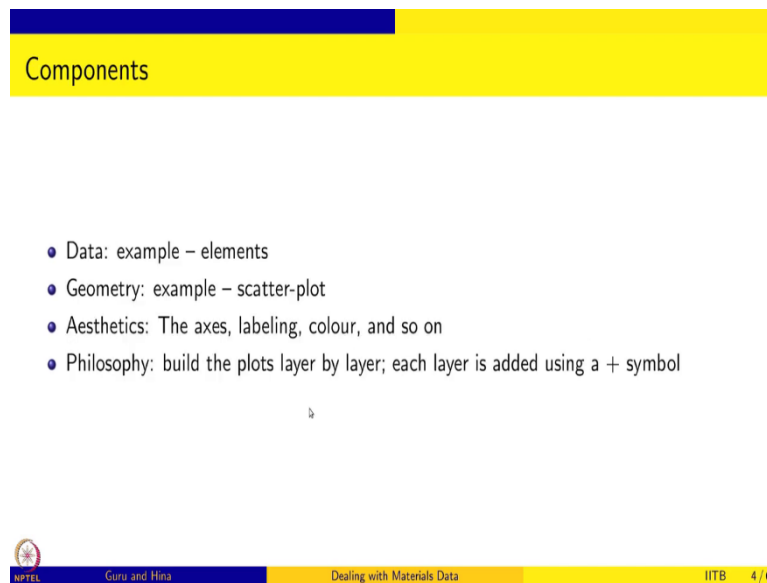
- Libraries for plotting: `grid`, `lattice`, and so on
- `ggplot2`: based on grammar of graphics
- Grammar of graphics: identify the components and rules to put them together
- Components – subject, verb and object; a grammatical rule to combine them to make sentences

NPTEL Guru and Hina Dealing with Materials Data IITB 3 / 6

So, there are many libraries for plotting, great lattice and so on and some of the R textbooks do describe these libraries, but `ggplot2`, the `gg` stands for grammar of graphics, okay. Grammar of a graphics is to identify the components and rules to put these components together. So, it is exactly like grammar for spoken language. How do we get the grammar, we say that, okay, these are the components like subject, verb, object, for example.

And how do you put them together to make a meaningful sentence, so there is a grammatical rule to combine them. And once you learn then all sentences or many many sentences can be built in this fashion. So, the grammar of graphics is to identify the different components for making plots and find the way of putting them together. So, that you know all graphs that you see or all plots that you see can be constructed using these two namely the individual components and rules for putting them together. So, that is what this `ggplot2` library is based on. It is based on the philosophy of grammar of graphics.

(Refer Slide Time: 2:36)



Components

- Data: example – elements
- Geometry: example – scatter-plot
- Aesthetics: The axes, labeling, colour, and so on
- Philosophy: build the plots layer by layer; each layer is added using a + symbol

NPTEL Guru and Hina Dealing with Materials Data IITB 4 / 6


And the components for plot for example, is a data we have been working with elements which is the data frame, so that is the data. The geometry of the plot is the scatter plot. So, we had density versus melting point and wherever you have a specific density against that melting point, we were putting a point so it is a scatter plot. So that is the geometry, it is a point plot. So, that is the geometry of the plot. And then there is aesthetics that is the axes, the labeling, the color, naming the plot and so on. So, there are many many things that can be done.

And the philosophy in ggplot2 is that we build the plots layer by layer. So, we take the data, we put the geometry, we put one component in, let us say the plot is in, then we name the axes, then we label the points and then we give colors to them and so on and so forth. And each layer that you add is added using a plus symbol. So, that is how ggplot2 works.

(Refer Slide Time: 3:38)

Help with ggplot2

- Irizarry: a nice chapter
- Cheat-sheet available online



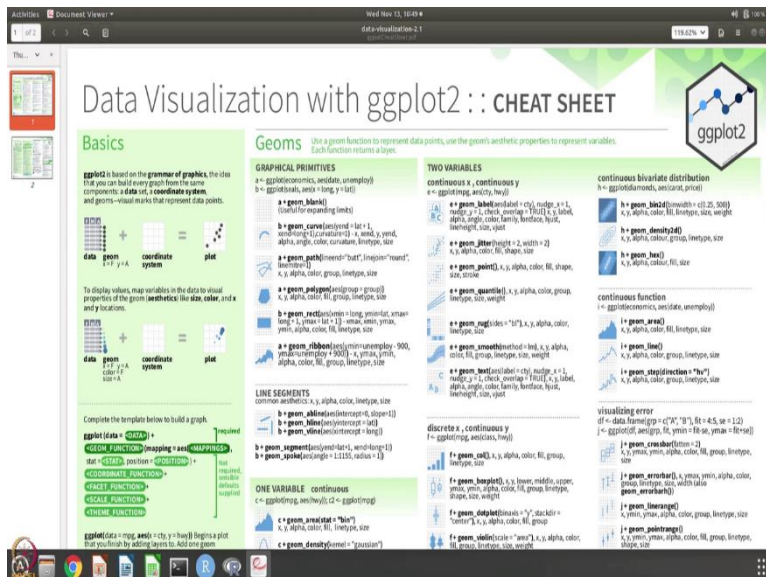
Guru and Hina

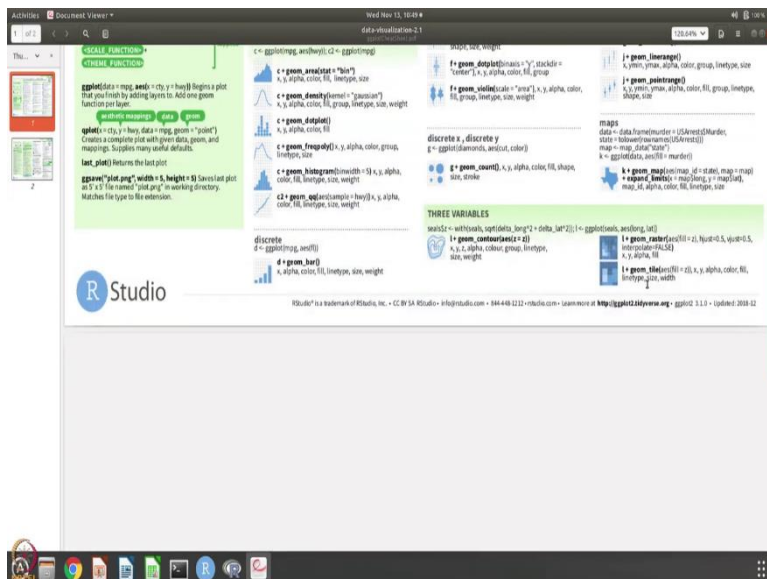
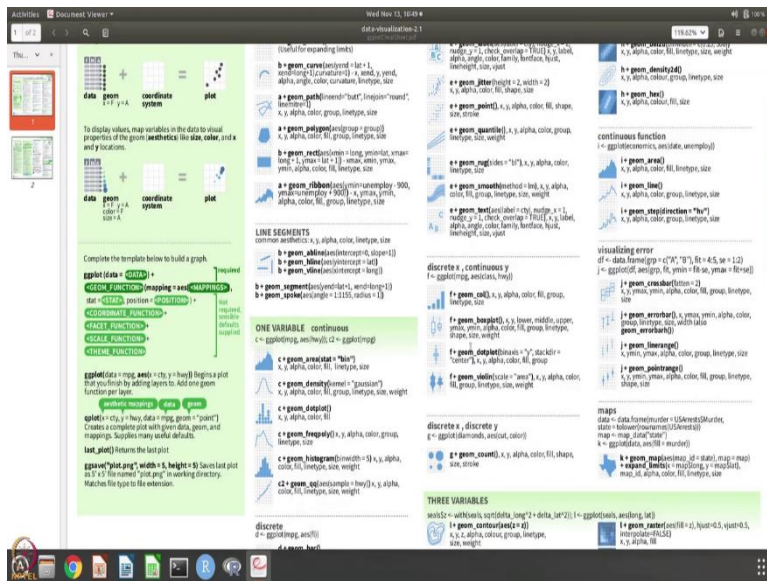
Dealing with Materials Data

IITB 5 / 6

And, of course, there is lots of help available for using ggplot2. There is also a book called grammar of graphics, which some of you might be interested. For using ggplot2 I have referred to the book by Irizarry which is freely available and it has a nice chapter on ggplot2. So I strongly recommend that you take a look at it. There is also a cheat sheet that is available online. So I want to show you that cheat sheet. And it is here.

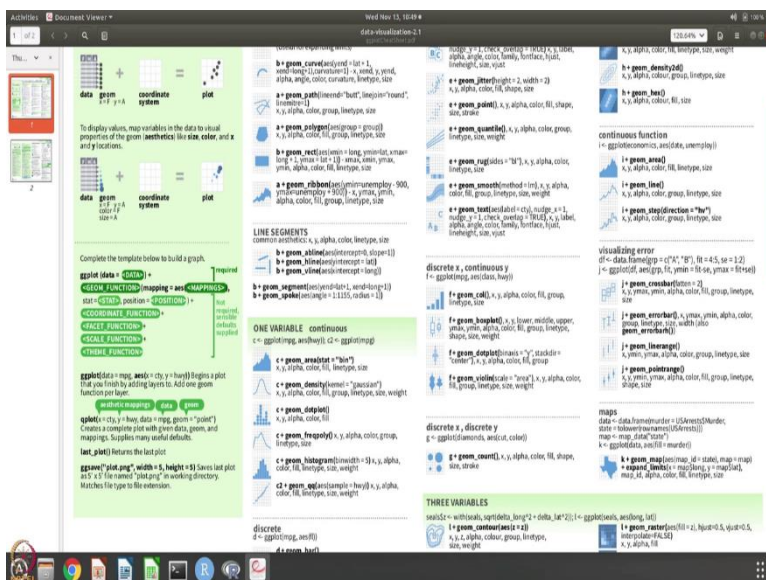
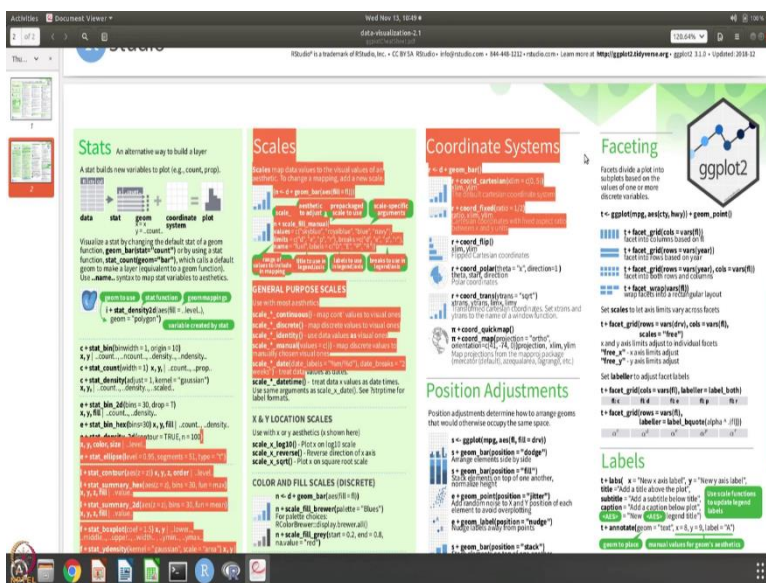
(Refer Slide Time: 4:07)





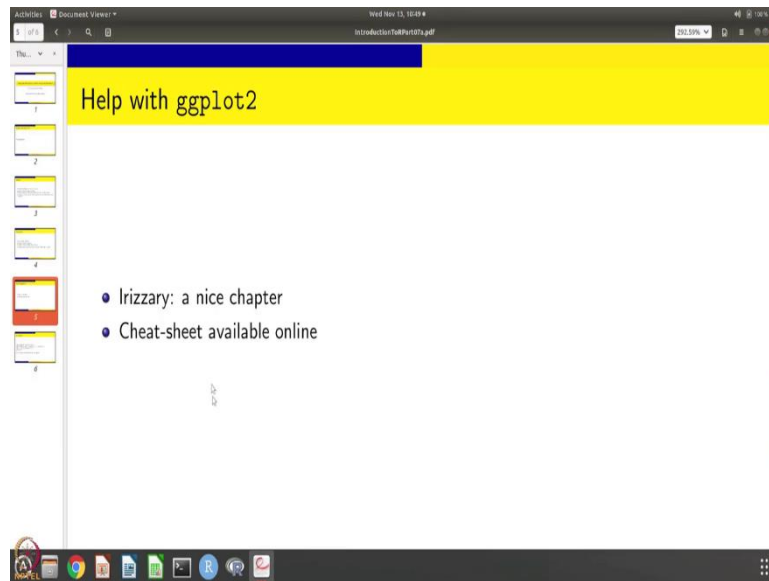
So it is called data visualization with ggplot2. So, as you can see, what is the basic? So, you have data and you have the geometry like what is x, what is y and you have a coordinate system and the plot is putting them together. And so, there are many different things that you can do and this basically, this cheat sheet gives all those commands.

(Refer Slide Time: 4:35)



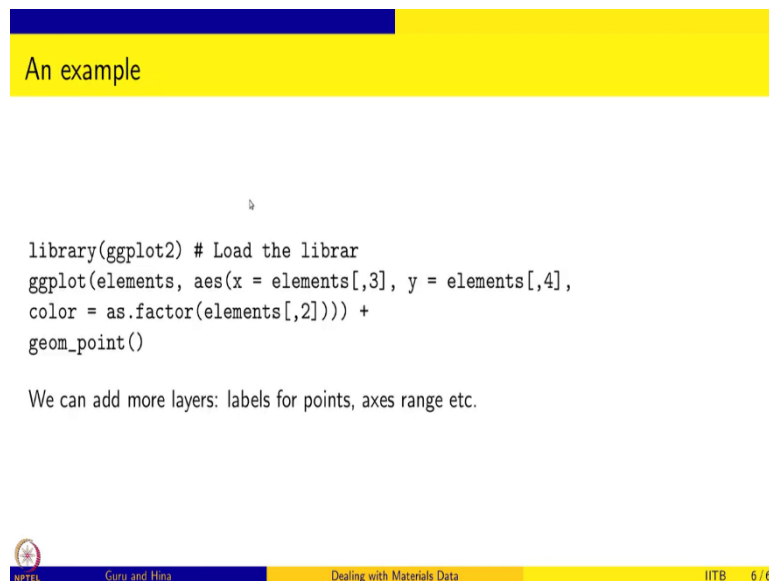
Of course, there is one more way. So, you can take the data, you can do some statistical analysis and then take the geometry coordinate system and plot. This is done for example, if you want to do a histogram plot or cumulative distribution, etc. So in which cases we have to not just take data and plot it, but we have to do some analysis and do it. So the ggplot can do that also. So it is very useful to have this cheat sheet downloaded and stored.

(Refer Slide Time: 5:11)



And this chapter by Irizarry also gives a link to this cheat sheet online. So this is what is helpful if you want to get some help with ggplot2, okay.

(Refer Slide Time: 5:22)



So, this is an example of how ggplot works, of course we have to load the library ggplot2, so we say that okay, let us do a plot. And this is the data, elements is the data. And this is the aesthetics that is which is the x axis and which is the y axis. So, elements, the third column is x axis, 4th column is the y axis, and the color should be done according to the second column. That is what we have said.

And we had added a layer, what is the layer? The layer says that at every x,y, you have to put a point, the geometry is basically a scatterplot, it is a point plots. So, so like this in you, and you can go on adding more layers for range and for labeling the points and labeling the x-y axis, and so on and so forth. So, we are going to learn about all that using our example. So, let us do that.

(Refer Slide Time: 6:19)

1 Plotting using ggplot2

In this session, we want to learn about the plotting library `ggplot2` and the philosophy behind the library. In order to use `ggplot2`, we will use the `elements` data frame. So, let us begin by inputting the data as a data frame.

```
elements <- data.frame("Element"=
c("Aluminium", "Nickel", "Gold", "Silver", "Copper",
"Iron", "Chromium", "Molybdenum", "Vanadium", "Tungsten",
"Magnesium", "Beryllium", "Zinc", "Cadmium", "Titanium"),
"Crystal Structure"=c("FCC", "FCC", "FCC", "FCC", "FCC",
"BCC", "BCC", "BCC", "BCC", "BCC",
"BCC", "BCC", "BCC", "BCC", "BCC",
"HCP", "HCP", "HCP", "HCP", "HCP"),
"Density"=c(2700, 8900, 19320, 10490, 8960, 7870, 7190, 10220,
6100, 19300, 1740, 1850, 7130, 8650, 4510),
"Melting Point"=c(660, 1453, 1063, 961, 1083, 1535, 1875,
2610, 1900, 3410, 650, 1277, 420, 321, 1668))
```

In this session, we will use `ggplot2` to plot the figures. There are several R packages for plotting, such as `grid`, and `lattice`, and `ggplot2` is one among them. `ggplot2` is very powerful because it uses the *grammar of graphics*; that is, it breaks the plots into its basic components and gives the syntax to put them together (exactly like the grammar lessons teach us to string a sentence, say, by combining a subject, verb and an object in a given manner, which then allows us to generate a large number of sentences with similar

```
+ c("Aluminium", "Nickel", "Gold", "Silver", "Copper",
+ "Iron", "Chromium", "Molybdenum", "Vanadium", "Tungsten",
+ "Magnesium", "Beryllium", "Zinc", "Cadmium", "Titanium"),
+ "Crystal Structure"=c("FCC", "FCC", "FCC", "FCC", "FCC",
+ "BCC", "BCC", "BCC", "BCC", "BCC",
+ "BCC", "BCC", "BCC", "BCC", "BCC",
+ "HCP", "HCP", "HCP", "HCP", "HCP"),
+ "Density"=c(2700, 8900, 19320, 10490, 8960, 7870, 7190, 10220,
+ 6100, 19300, 1740, 1850, 7130, 8650, 4510),
+ "Melting Point"=c(660, 1453, 1063, 961, 1083, 1535, 1875,
+ 2610, 1900, 3410, 650, 1277, 420, 321, 1668))
```

The RStudio interface also shows a 'Data' pane on the right with the following information:

Variable	Value
elements	15 obs. of 4 variables


```

+ "BCC", "BCC", "BC
+ "HCP", "HCP", "HC
+ "Density"=c(2700,8900,19320,10490,896
0,7870,7190,10220,
+ 6100,19300,1740,1850,713
0,8650,4510),
+ "Melting.Point"=c(660,1453,1063,961,1
083,1535,1875,
+ 2610,1900,3410,650,
1277,420,321,1668))
> str(elements)
'data.frame': 15 obs. of 4 variables:
 $ Element      : Factor w/ 15 levels "Aluminium","Berylli
um",...: 1 10 6 11 5 7 4 9 14 13 ...
 $ Crystal.Structure: Factor w/ 3 levels "BCC","FCC","HCP": 2
2 2 2 2 1 1 1 1 1 1 ...
 $ Density      : num 2700 8900 19320 10490 8960 ...
 $ Melting.Point : num 660 1453 1063 961 1083 ...

```

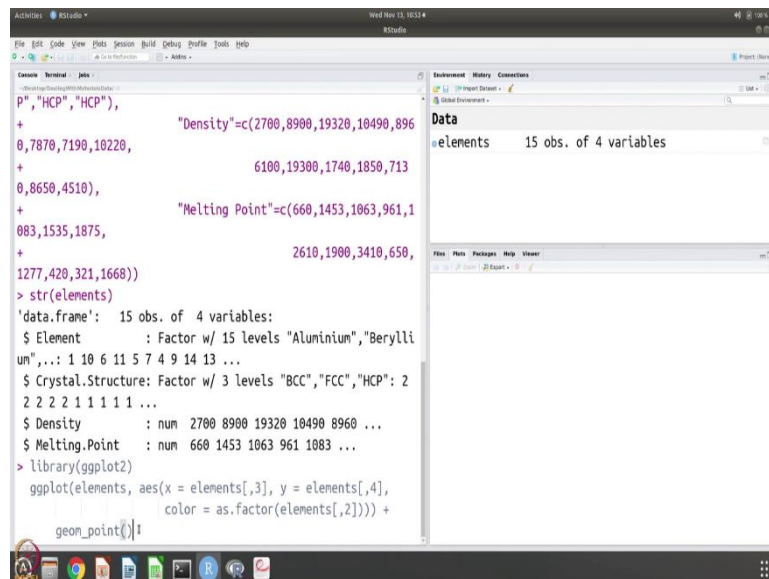
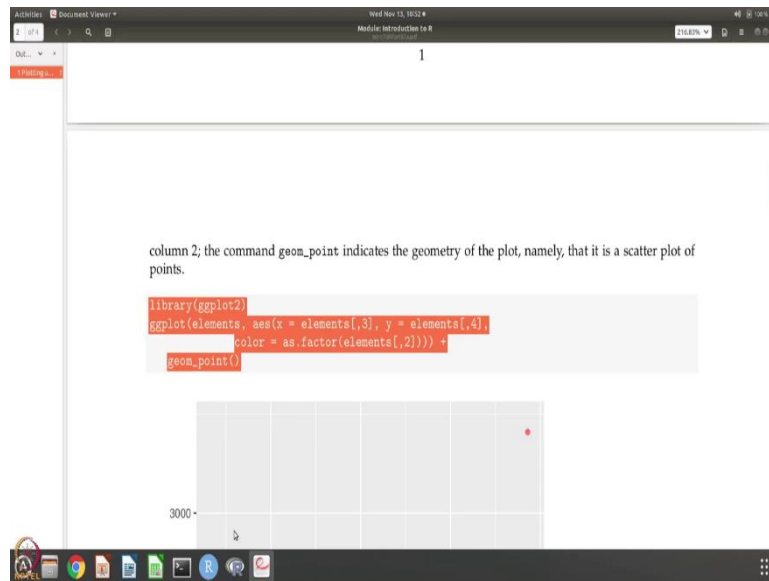
So, as we did earlier, let is open R, and let us first get the data in place. So, we copy the data. And okay, so as we did earlier, so it is a data frame. So, we just have 4 columns, and we have named the column and we have given the data for those columns. And so the data is loaded, you can see there are 15 observations and 4 variables. So always a good idea to check that everything is in place. So, it is a data frame, 15 observations, 4 variables, etcetera, okay.

(Refer Slide Time: 7:05)

```

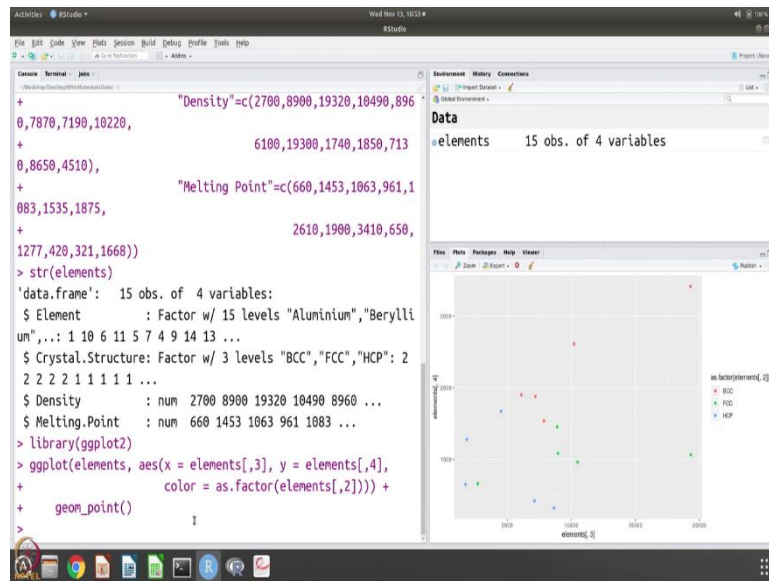
+ "HCP", "HCP"),
+ "Density"=c(2700,8900,19320,10490,896
0,7870,7190,10220,
+ 6100,19300,1740,1850,713
0,8650,4510),
+ "Melting.Point"=c(660,1453,1063,961,1
083,1535,1875,
+ 2610,1900,3410,650,
1277,420,321,1668))
> str(elements)
'data.frame': 15 obs. of 4 variables:
 $ Element      : Factor w/ 15 levels "Aluminium","Berylli
um",...: 1 10 6 11 5 7 4 9 14 13 ...
 $ Crystal.Structure: Factor w/ 3 levels "BCC","FCC","HCP": 2
2 2 2 2 1 1 1 1 1 1 ...
 $ Density      : num 2700 8900 19320 10490 8960 ...
 $ Melting.Point : num 660 1453 1063 961 1083 ...
> library(ggplot2)
ggplot(elements, aes(x = elements[,3], y = elements[,4],
color = as.factor(elements[,2]))) +
  geom_point()

```



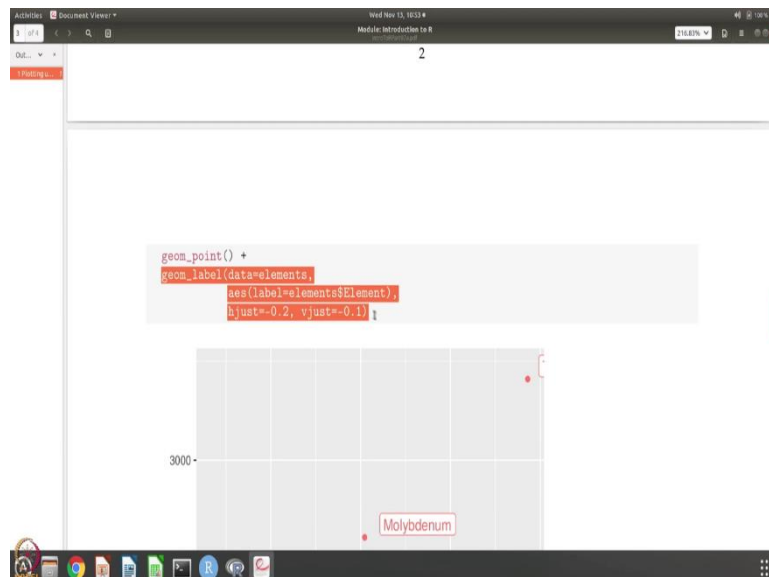
So now, let us do the plotting, to that we are going to use `ggplot` and this is the first command, this is what we saw. So, we load the library `ggplot2`, and then we say take the data `elements` that that is a data frame and the aesthetics is `x` is the third column, so, that is the density and `y` is the fourth column that is a melting point. And these data points have to be colored and the color is according to the factor. There are three levels right `bcc`, `fcc`, `hcp`, so that is what the color should be. And the geometry is that it should be a scatter plot, okay.

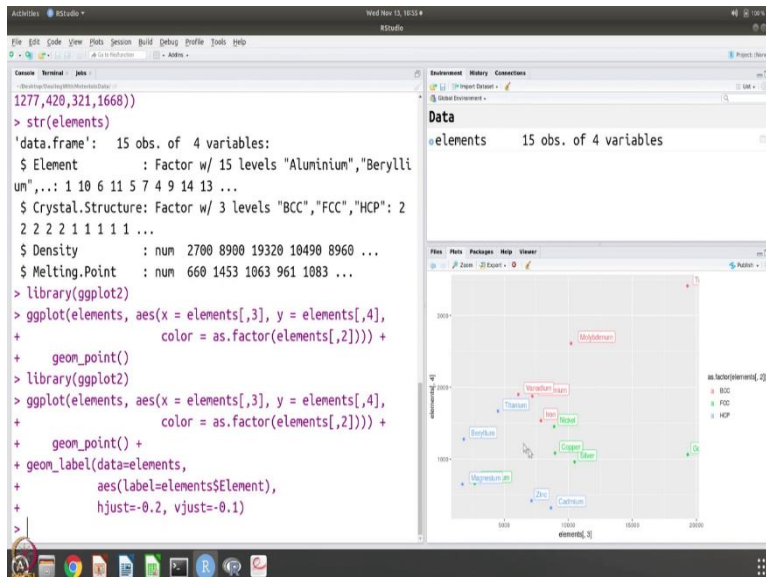
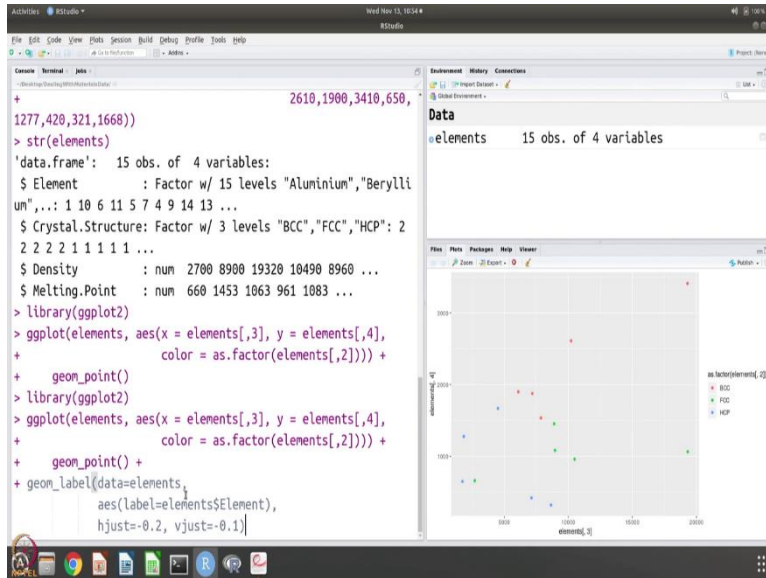
(Refer Slide Time: 7:45)



So, you can see that it is the density versus the melting point. And these are colored and you can already see that unlike the earlier case, the color scheme or the labeling is done automatically. So, it says that okay what is red is bcc and what is I think green is fcc and blue is hcp and so on. So, so it also gives you this labeling so you can easily identify what they are.

(Refer Slide Time: 8:17)



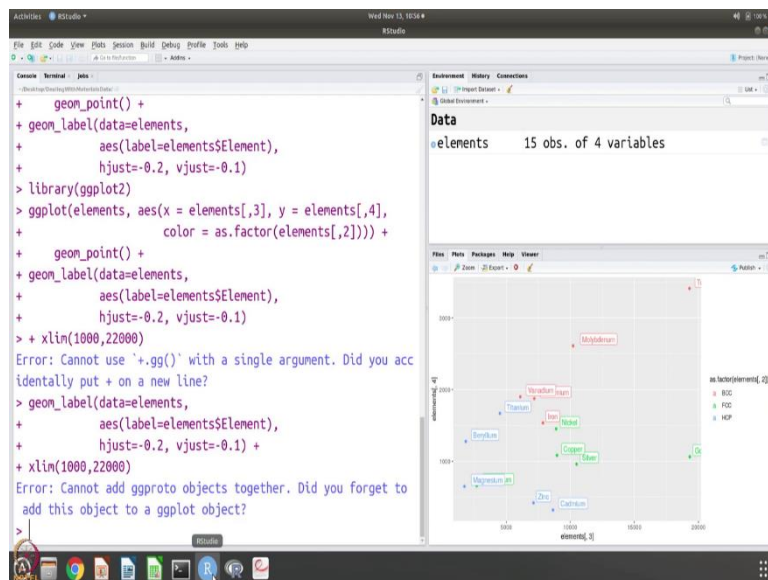


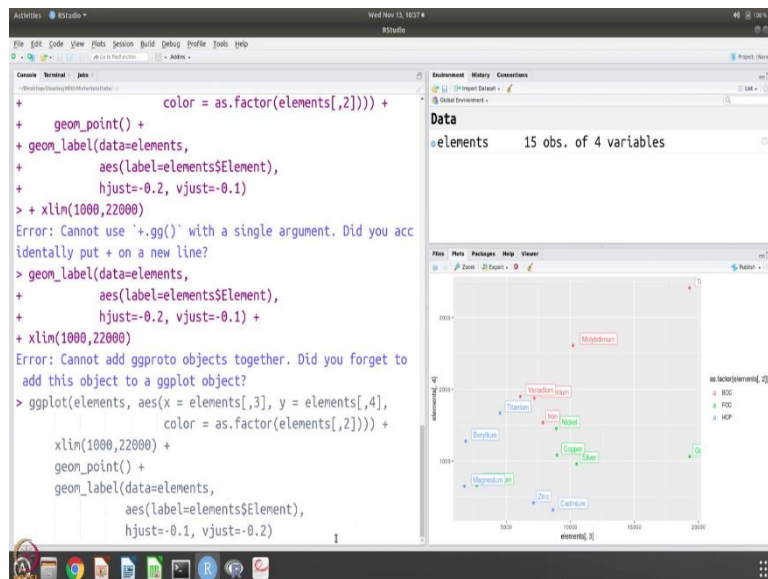
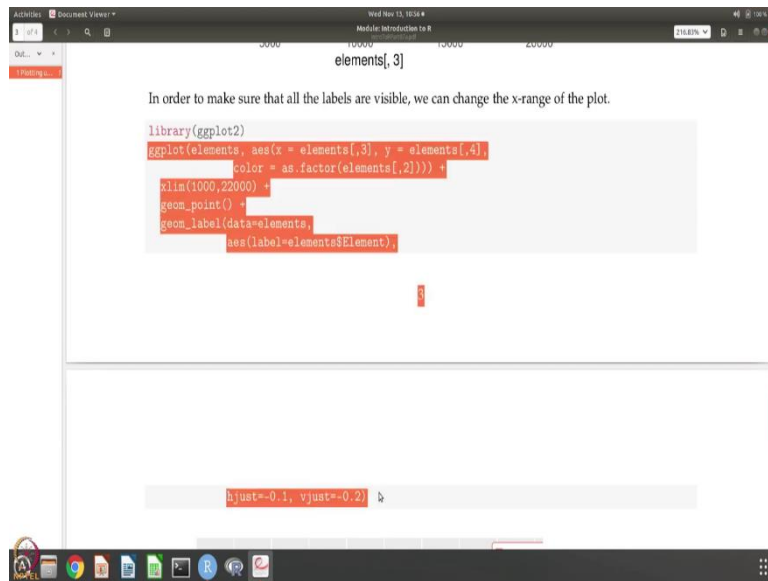
Okay, so that is what is shown here. The colors are much clearer here, this is a red, green and the blue. And so let us do the next one. And in this case, we want to add one more element, I want to name the so I am going to do this, I am going to add one more layer, and the layer is this. So, we are going to put a label and the label is from the data elements and we are going to take the element name from that data and use that as the label and this hjust and vjust are the horizontal and vertical justification that is where you should put these labels, okay.

So, you can see, so some magnesium and I did not know what was here? Maybe aluminum or something. So zinc, cadmium, beryllium, so, you can see. And unlike the earlier case, you can see that I did not explicitly I have to say that the labels are also should be color coded according to the points that is done here automatically. Check, takes the data and because that is there already, because we have already built in this layer where we said the color should be according to this.

So, it is going to use that information and do it consistently. So, it is nice that way, and very intuitive and very clear. Okay, now we can do one more thing. So you can see that these labels are cut out. So let us say that I want to change the range. So I can add one more layer, x limit is thousand to twenty two thousand.

(Refer Slide Time: 10:20)





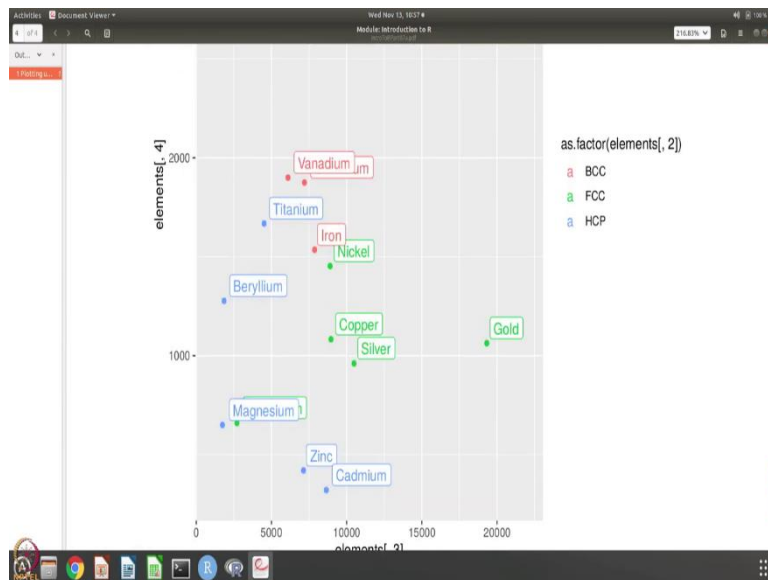
So, I do this and then I do that and then I do that okay I have to do that, so okay so problem okay, let us do this again. So, I am going to put it here. This three is not the data, okay. So, this is ggplot. So, we are going to say that this is 3 versus 4 and color according to factor and we have given an x limit. So, it will go from thousand to twenty two thousand, so, that we will see this names very clearly and then it is a point geometry and the labeling will be done.

(Refer Slide Time: 11:41)

The screenshot shows the RStudio interface. The console on the left contains the following R code and error messages:

```
+ geom_point() +  
+ geom_label(data=elements,  
+ aes(label=elements$Element),  
+ hjust=-0.2, vjust=-0.1)  
> + xlim(1000,22000)  
Error: Cannot use '+.gg()' with a single argument. Did you acc  
identally put + on a new line?  
> geom_label(data=elements,  
+ aes(label=elements$Element),  
+ hjust=-0.2, vjust=-0.1) +  
+ xlim(1000,22000)  
Error: Cannot add ggproto objects together. Did you forget to  
add this object to a ggplot object?  
> ggplot(elements, aes(x = elements[,3], y = elements[,4],  
+ color = as.factor(elements[,2]))) +  
+ xlim(1000,22000) +  
+ geom_point() +  
+ geom_label(data=elements,  
+ aes(label=elements$Element),  
+ hjust=-0.1, vjust=-0.2)  
>
```

The Environment pane on the right shows a data frame named 'elements' with 15 observations and 4 variables. The Plot pane shows a scatter plot of 'elements[, 4]' (y-axis) versus 'elements[, 3]' (x-axis). Points are colored by 'as.factor(elements[, 2])' and labeled with element names. The legend indicates three crystal structures: BCC (red), FCC (green), and HCP (blue).



The screenshot shows the RStudio interface with the console on the left and the Environment pane on the right. The console contains the following R code and error messages:

```
+ geom_label(data=elements,  
+ aes(label=elements$Element),  
+ hjust=-0.2, vjust=-0.1)  
> + xlim(1000,22000)  
Error: Cannot use '+.gg()' with a single argument. Did you acc  
identally put + on a new line?  
> geom_label(data=elements,  
+ aes(label=elements$Element),  
+ hjust=-0.2, vjust=-0.1) +  
+ xlim(1000,22000)  
Error: Cannot add ggproto objects together. Did you forget to  
add this object to a ggplot object?  
> ggplot(elements, aes(x = elements[,3], y = elements[,4],  
+ color = as.factor(elements[,2]))) +  
+ xlim(1000,22000) +  
+ geom_point() +  
+ geom_label(data=elements,  
+ aes(label=elements$Element),  
+ hjust=-0.1, vjust=-0.2)  
> help("ggplot2")  
>
```

The Environment pane on the right shows the same 'elements' data frame. The Help pane on the right displays the help page for the 'ggplot2' package, titled 'ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics'. The help page includes a description of the package and a list of authors.

So, you can see now the ranges change so, you can see the golden tungsten clearly they are readable. So, so, what we are doing is that it is the same. So, we just say what is the data and what is the aesthetics, what is the x-y and what color. Then we are adding layer by layer we are first saying okay change the x range, then we are saying okay put the data points and then we are saying okay label the data points so, you can do this layer by layer and that is the advantage of ggplot. Of course you can use labels so, we have not done x label and y label and somewhere during this one of these sessions we will do that also.

So, that is just one more layer so you add another plus and say labs I think that is for labels. x is this and y is this and title is this and so on and so forth. So, you can take a look at the gg plot help itself, okay or the cheat sheet, or the Iriarray's book, so the help itself gives you some of these links for you to learn about ggplot2. And we are going to use ggplot2 also extensively in this course. So, I recommend that you install this package and we will learn how to use it for more plots. Thank you.