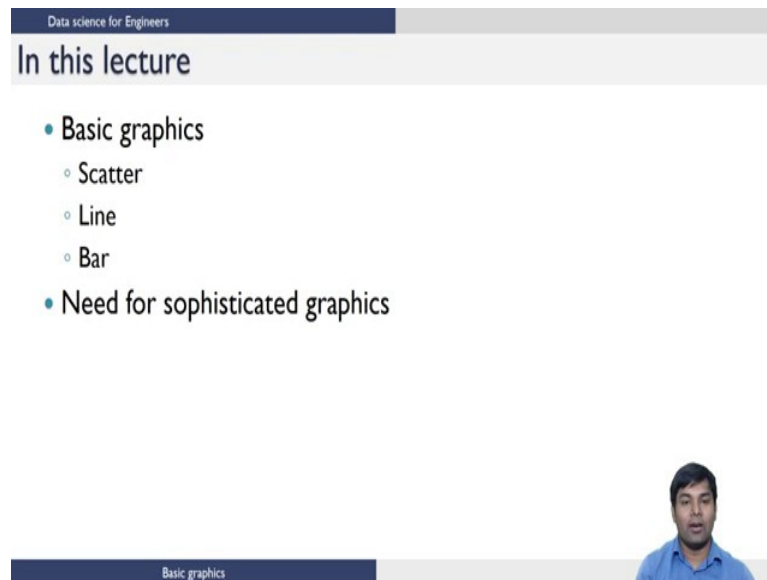**Data Science for Engineers**
Prof. Raghunathan Rengaswamy
Department of Computer Science and Engineering
Indian Institute of Technology, Madras

Lecture – 11
Data visualization in R Basic graphics

Welcome to the lecture 10 and the final lecture in the R module of the course data science for engineers. In the previous lectures, we have seen; what are the basic data types that are supported by R, how to write scripts, how to write functions and how to do control structures, how to do programming and so on.
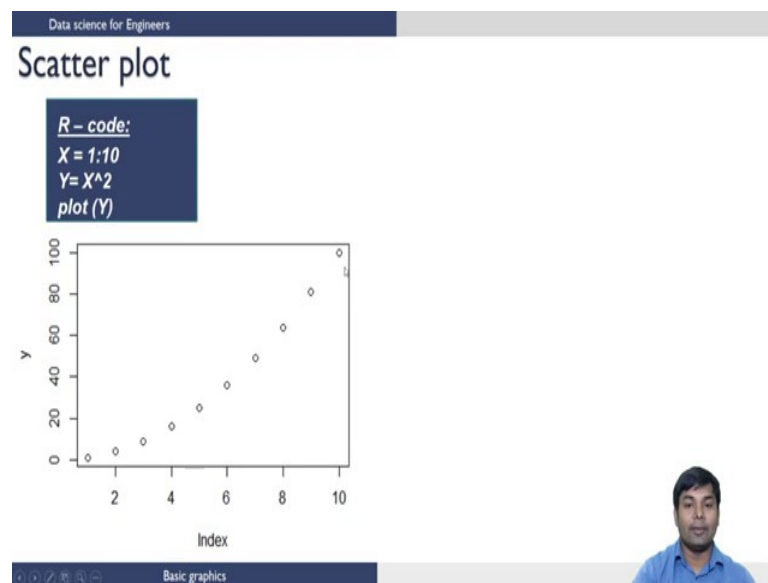
(Refer Slide Time: 00:35)

In this lecture, we are going to show you how to generate some basic graphics; such as scatter plot, line plot and bar plot using R, and we will also give a brief idea on why there is a need for more sophisticated graphics and how R does it.

(Refer Slide Time: 00:54)

First we will consider scatter plot; scatter plot is one of the most widely used plots where we have some independent variable and dependent variable, when you want to see how a dependent variable is dependent on the independent variable. We can use scatter plot generating the scatter plot in R, is quite simple. The first command here shows, it is creating a vector which is having the elements from 1 to 10, and the next command here takes this x and calculates the element wise square of the x and then assign it to value y.

When you plot y it will generate this plot here. Since we have not specified, what is x which is independent variable, the R generates its own independent variable as the index, since this vector contains 10 elements. It will create the index based on the number of elements in the vector and then the y values which are the squares of elements 1 to 10 that are 1 4 9 and so on are shown in the y axis, and $10^2 = 100$. We have the final value here on the y axis as 100.

So, let us illustrate the scatter plot using some inbuilt data set; that is available in R. So, we are talking about a data set by name empty cars. So, you can access this data set by just typing empty cars. This data set is a data frame which contains 32 observations on 11 variables. The variables are listed here such as number of cylinders, which is represented by variable c y l and m p g. What is the mileage that this cars gives; that is miles per us gallon and weight w t, which is weight of the car and so on.

Now, let us try to plot a scatter plot between weight and m p g of this data frame . To do that what we need to use, is plot command. This is your independent variable car weight, this is your dependent variable miles per gallon and this main helps in naming the title of the graph x lab to give a label for x axis, y lab is used to give a label for y axis and the this pch corresponds to different shapes for points, and this pch = 19 corresponds to the shape that is shown in this screen. You can use different pch values to obtain different shapes for the points in a scatter plot.

(Refer Slide Time: 04:02)
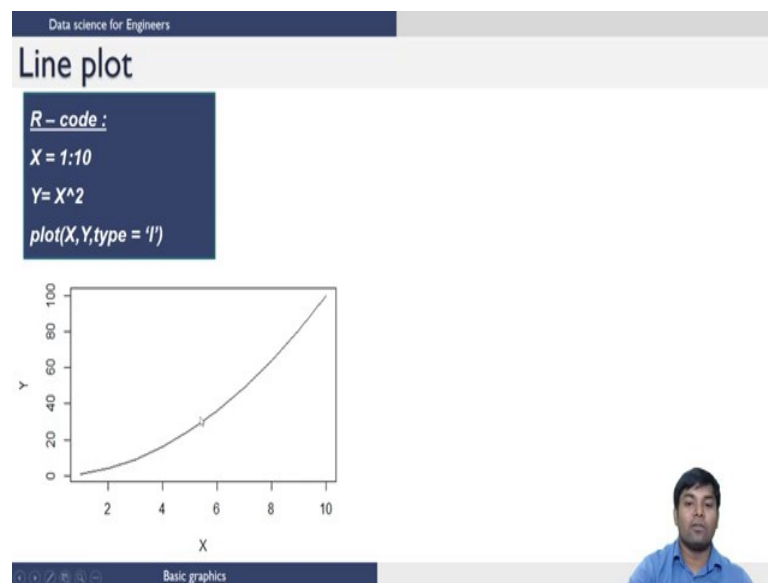
Next we move the line plot. You can take same example what we have seen earlier. If the same plot command can be used what you need to do to generate a line plot, is to specify an extra argument type which is l.

(Refer Slide Time: 04:24)

Bar plot

Syntax:
barplot(H, names.arg, xlab, ylab, main, names.arg, col)

R – code :
H <- c(7,12,28,3,41)
M <- c("Mar","Apr","May","Jun","Jul")
barplot(H,names.arg = M, xlab = "Month", ylab = "Revenue",
col = "blue", main = "Revenue chart",border = "red")

Basic graphics

So, type = l generates a line, instead of the scatter plot. Next we move on to the bar plot, the syntax to generate bar plot in R is as follows; bar plot of h. These are the heights which can be a vector or matrices to keep it simple. We will deal with only vectors and names dot argument. What this argument does is? it will print the names under the each attribute in the H x lab and y lab, and main has a same meanings as what we have seen for the scatterplot, and colour gives us an option to give colour to the bar plot. This is the R code that can be used to generate the bar plot. I want to define h heights of the barcodes as a vector, which is having the values 7, 12, 28, 3 and 41, and I want to create another vector which is of character variable, which is having the values March, April, May, June and July, and now, I am trying to create a bar plot with h as heights and name start arguments as m x lab as month, y lab as revenue and the colour of the bar notes I want, is blue and the title is revenue chart and the border is red.

(Refer Slide Time: 05:48)
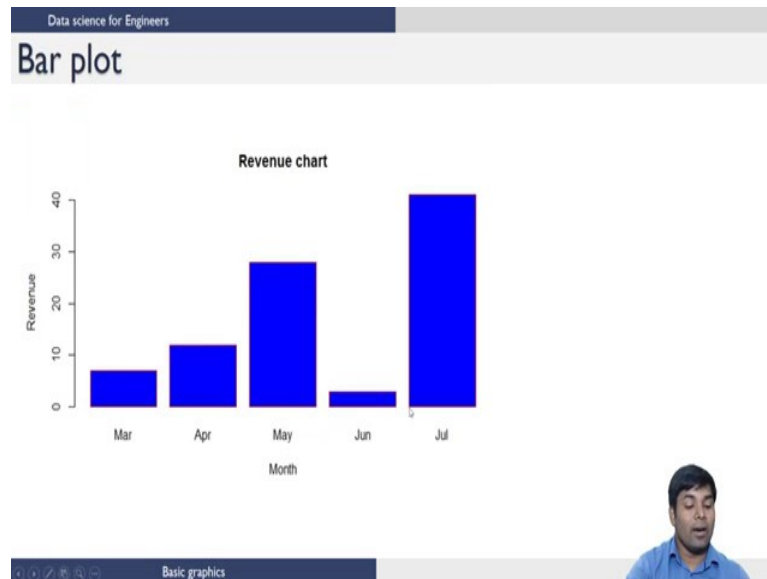
So, when you execute these commands, this is how the bar plot looks. These are the heights of the bar charts, this is A 3. And then in the names dot variable we have March, April, May, June and July, which is printed at the bottom of each height, and the x axis is month, y axis is revenue and the title is revenue chart.

(Refer Slide Time: 06:10)
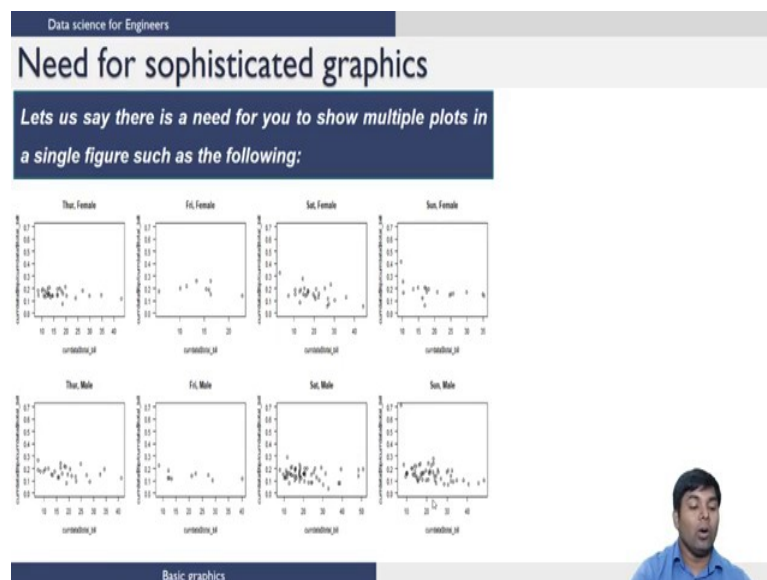
Now, let us see why there is a need for sophisticated graphics. Let us say there is a need for you to show multiple plots in a single figure as shown below. How do you do this?

(Refer Slide Time: 06:21)

What are the challenges that you face when you want to create figure that was shown in the earlier slide. So, the exact figure can be reproduced using this code which is shown here for this.

(Refer Slide Time: 06:37)

What you have to know, is you have know where to introduce for loop, which columns of data frame to be selected for plotting, and you have to also position each graph in the grid etcetera. Even though you do all of this operations, the visuals are less pleasing that is where we need more sophisticated graphics packages in R . This is where the ggplot2 comes into picture. The ggplot2 provides a very beautiful package for generating graphics in R in this course, we have not deal much with ggplot2.

(Refer Slide Time: 07:14)

In summary, we have seen how to generate scatter plots, line plots and bar plots in the R. We have also seen the challenges and disadvantages of basic graphics and the need for using the advanced packages; such as ggplot2 for generating beautiful graphics in R. With this we end the R module for this course. Wish you all the best for the next modules in this course.

Thank you.