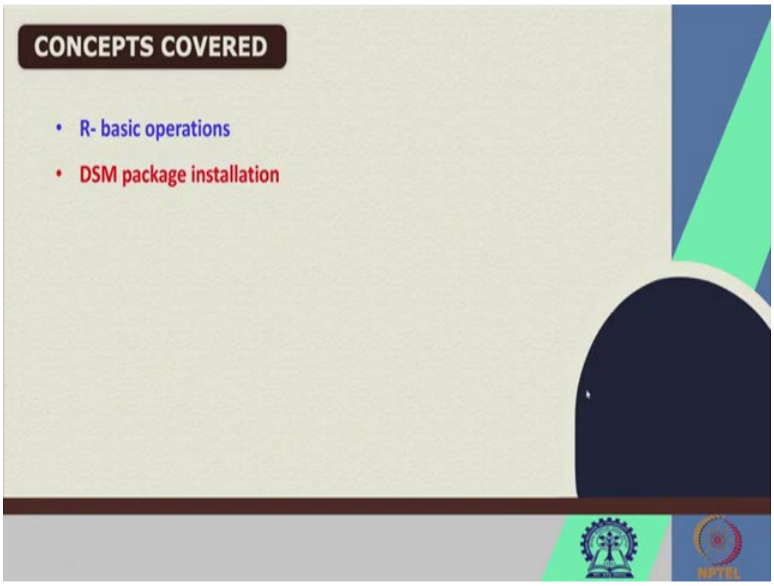
Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology, Kharagpur
Lecture 50
Digital Soil Mapping - General Overview (Contd.)

(Refer Slide Time: 0:27)

Welcome friends to this 50th lecture of this NPTEL Online Certification Course of Machine Learning for Soil and Crop Management. And in this week, we are having some basic overview of Digital Soil Mapping. In my previous lectures, I have discussed about, what is DSM, what are the different applications of DSM, why we prefer the SCORPAN plus e model for DSM we have discussed.

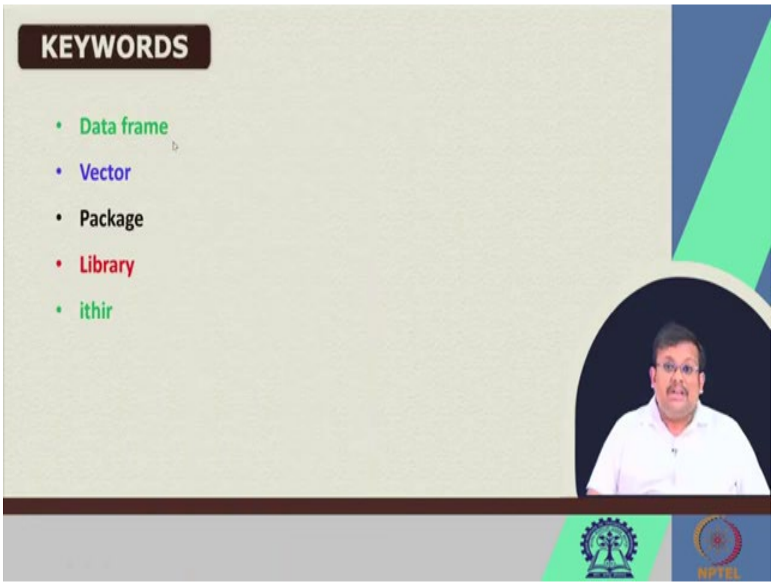
At the same time we have also discussed the GIS Geostatistics and all the important aspects, which are required for executing different types of DSM applications. Now, also we have started discussing about R and R studio in our previous lecture. I have showed you how to install R, how to install R studio, what are the basic R commands. So, we will continue from there where we left and we will try to finish this week using the basic R codes for digital soil mapping.

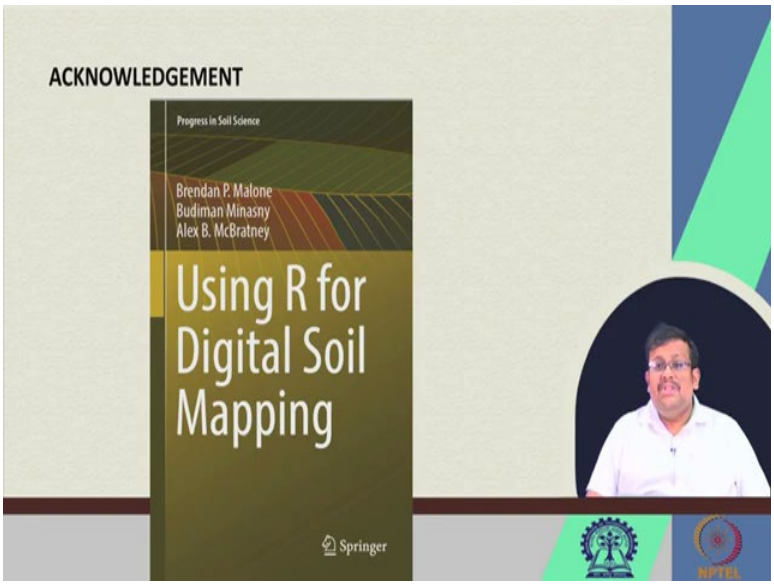
(Refer Slide Time: 1:41)



So, these are the concepts which we are going to cover today, we are going to have some basic R operations. And also we will see some DSM package installation of some DSM packages, which are required for DSM operations.

(Refer Slide Time: 1:57)



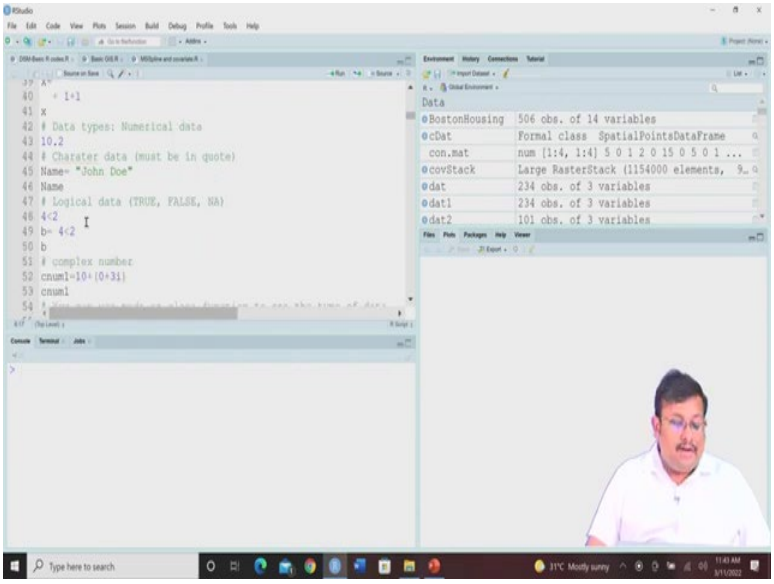


And these are some of the keywords, like data frame, vector, package, library, ithir, so these are the some of the keywords for this for this lecture. So, let us but before going to that I must acknowledge for this DSM codes acknowledge the contribution of Brendan Malone and Budiman Minasni and Alex McBratney, who wrote this book using R for digital soil mapping, I have gathered whatever knowledge is required for using R for digital soil mapping from this book.

So, I highly recommend you to get this book buy this book and learn in details about how to use R for digital soil mapping not only this book will give you the extensive over extensive discussion on digital soil mapping but also will give you the practical application which of DSM and you will know from the scratch how you can use this software for digital soil mapping, all the packages all the data sets are created and they are already uploaded in different, different host websites, I am going to show you those.

But at the same time but if you want to have more detailed knowledge about this digital soil mapping and how to use R for executing digital soil mapping you should go through this book. I personally thank Professor Budiman Minasni from the university of Sydney who were very helpful to give me some the concept of using some of the codes from this which they have mentioned in this book. So, I am really thankful to Professor Minasni.

(Refer Slide Time: 4:14)

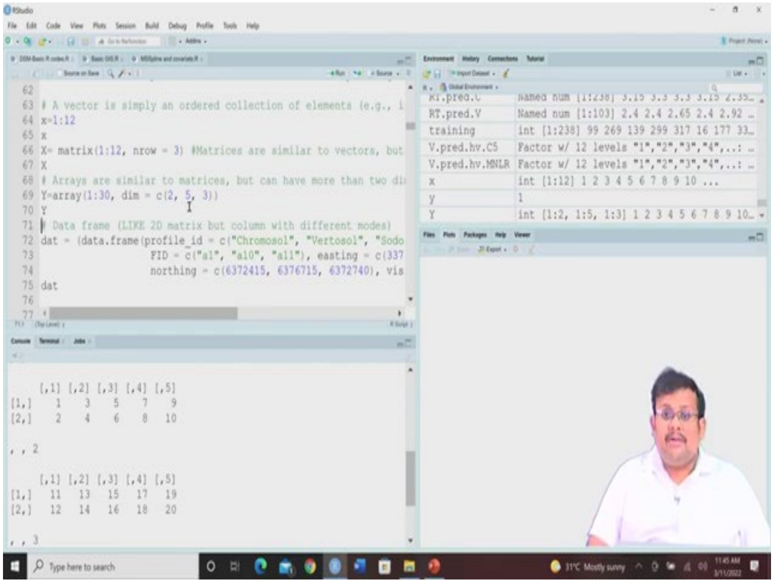


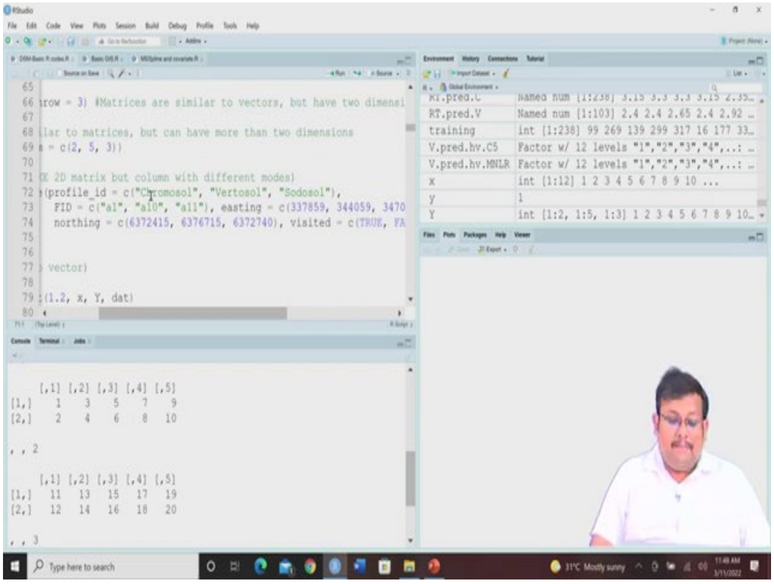
So, let me just go back and show you some of the basic R codes. So, in my previous lecture, we have discussed about the complex number and different types of from the basic calculation like simple sum and then how we can use R as a calculator and how to get different types of values like infinity values in R we have also discussed.

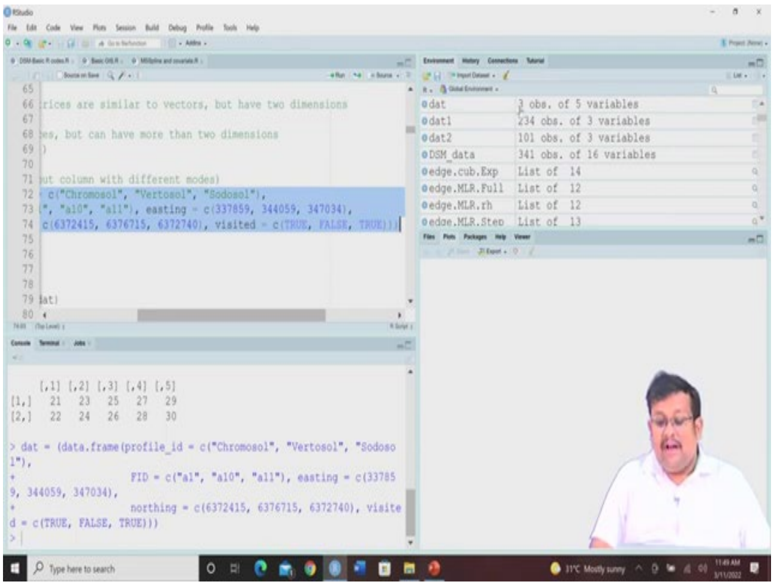
And then how to how to remove an object from my workspace, how to get the new line separator or how to get the constants and how to extract, how to know what is actually there in this constant we have already discussed.

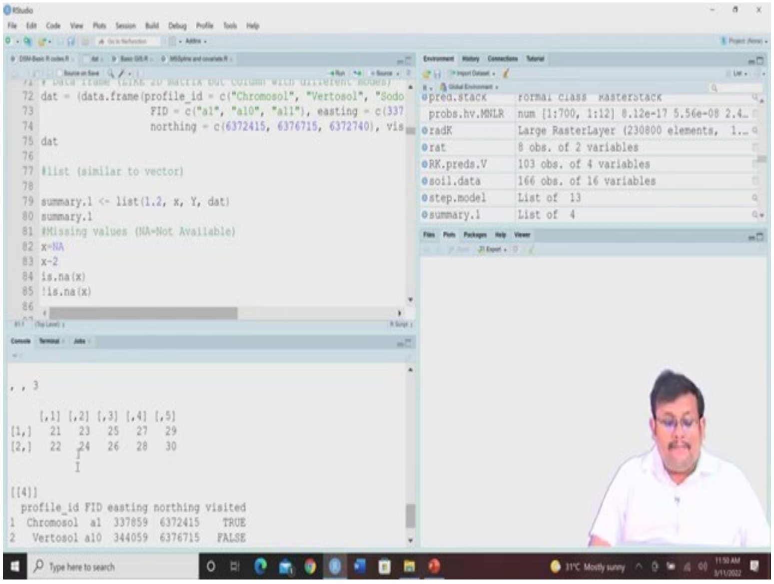
And also we have seen how to use this list command and also we have seen that how we can input different types of data in R, starting from numerical data, then character data, then logical data as well as complex number. So, we have discussed in our previous lecture.

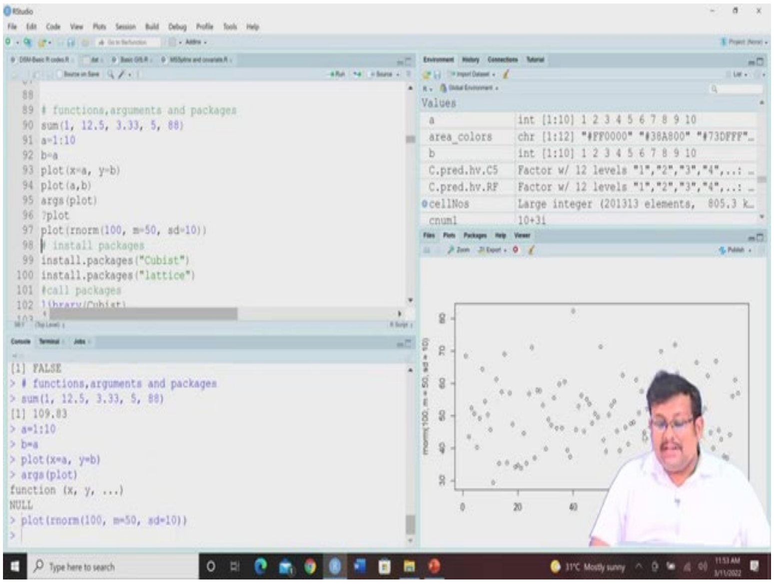
(Refer Slide Time: 5:22)











Now, let us start from here, so R data structure so will this R data structure is the platform for all the calculations and plotting, so we are going to start with the defining the vector, so vector is a simply an ordered collection of elements for example, it can be individual, so it is basically order collection of individual numbers.

So, here you can see suppose x is a vector which assumes the value of 1 to 12 and here if they are they are arranged like 1, 2, 3, 4, up to 12, then we can just separate them with this colon and express this like this way. So, if you want to see what x means, so you can see that it stands for 1 to 12.

Then the x matrix, so let us now let us do let us deal with matrix, you all know what is matrix, now suppose we want to develop a matrix from the values of 1 to 12 and the number of rows will be 3, so matrix remember these are similar to the vectors but they have some defined number of rows and columns. So, here if we want to develop a matrix called x with the values from 1 to 12 and number of row equal to 3.

So, if you want to see how it looks like, so you can see that this is the matrix and by default the values are arranged in columns from 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, so this is the matrix of this is a matrix taking the values of 1 to 12 with the number of rows three.

Now, next is arrays, arrays are similar to matrices, but they have more than two dimensions. So, here you can see we are can use this array function and using this 1 to 30 numerical values and our dimension is 2, 5, 3. So, if you see what that means, so if we just run y you will see that there are three matrix and in this three matrix individual matrices individual matrix is having 2 number of rows and 5 number of columns.

So, by default the first number is number of rows and second number is the number of columns. So, then a very important concept, that is called data frame. Now, what is data frame? So, data frame remember it is a it is a table it is a two dimensional table it is a kind of a array like structure and in which each column contains values of one variable and each row contains one set of values for from each column.

So, this is basically a two dimensional table and this data frame is a basic requirement for performing any type of statistical calculation, specifically if we want to develop models using R for DSM we need to convert the data any special data into a specific data frame. I will show you how to do that, but at this point of time I just want to show you how to create a data frame.

So, to create a data frame let us we will be using this data dot frame com function, so let us give a name to this data frame that is dat and we are using this data dot frame function and these are the arguments within this data dot frame function. So, here we have four different variables profile id, FID, sorry five different variables profile id and then FID, then easting northing and whether you have visited that profile or not.

So you can see here profile id since it is a character, so we have put those names it is a nominal variable, so you can see Chromosol, Vertosol, Sodosol these are different types of soils, so these are categories. So, since they are categorical attribute or character attribute, then that is why we have kept them within code and FID is also some identifier these are also characters.

Now, easting and northing, easting northing are basically they are some values and the term these easting and northing are geographic cartesian coordinates for a point and so easting refers to the eastward, eastward measure distance or the coordinate whereas the northing refers to the northward measured distance or the coordinates.

So, this easting and northing generally we use to indicate the geographic cartesian coordinate or the location of any given point. So, while we are using this easting and northing you please understand that these are showing the location of these three soil samples which we are going to use for creating this data frame.

And next is visited, so you can see true, false, true, so this is a logical variable, so you can see either if you have visited that place that soil so it is true, if it is not then it is false. So, this is how we create a data frame, let us first create this data frame and once you create this data frame you can see already in the environment this data frame is created. So, you can see the data frame profile id, FID, easting, northing and visited.

So, Chromosol, Vertosole, sodosole, A1, A10, A11, easting, northing and visited or non visited. So, you see the data frame has been created you can also see this data frame in this R console by just typing this dat and then running this dat, so you can see that this data frame is created also you can use this list command which is similar to vector, so list is basically showing the list of all the all the variables.

So, here suppose we are using this list 1.2 which is a numerical variable, then x comma y comma dat, so let us see how it looks like. So, if we if we see the summary you will see that it will start with the numerical variable 1.2, then x is a vector from 1 to 12, then y is of course a we have already created y is an array and the last point is our dat, which is the data frame. So, all of them will be appeared in a list when we use this list function.

Now, missing values are generally appeared in R as na that is not available. So, if you see that x is not available then if you want to have x minus 2, then you will see that it is also not available, then is dot na so it is basically making an enquiry whether x is missing or not, so if we do that you will see that if x is missing value, so we use is dot na that means it shows it is a missing value when it is gives a true output.

And when we use this exclamatory sign before this is na that is just showing the opposite that means we are asking R that whether x is a real value or not. So, when you do that then it gives the false, so that means it is not a real value it is a missing value.

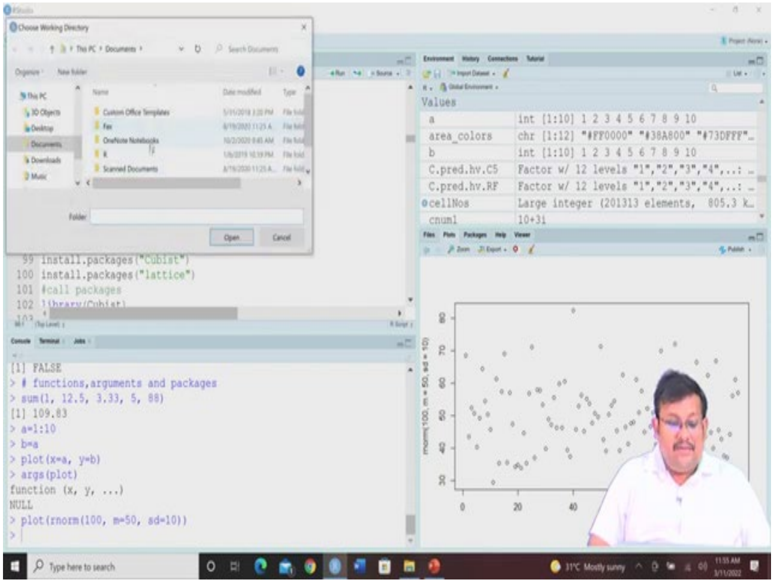
Now, function arguments and different packages we are going to use so let us use this sum function. So, in the sum function you can see we are having these numerical values, so we are getting the sum and then here 109.83 also let us create a vector of a using 1 to 10 and let us assume that b is equal to a, so if you want to plot this x equal to a and y equal to b, so we will get a plot like this.

So, our x variable is a and our y variable is b and if both since both of them are similar therefore we are getting a linear graph while plotting that. Now, once we do that you remember that when we use a function the parameters which we use inside a function those are known as arguments. So, these arguments for plot what are the arguments for the plot, so if you click if you click this arguments of the plot, so you can see that it is function then x and y and then other things.

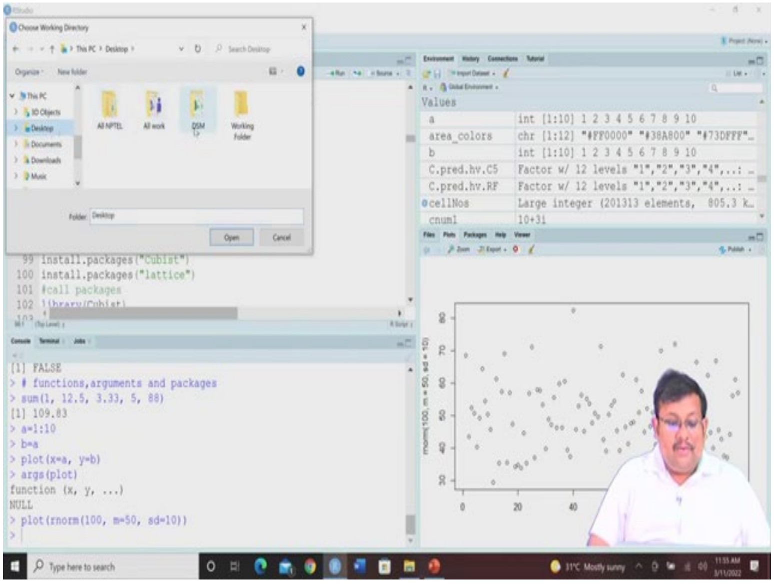
So, basically the minimum requirement for creating a plot is to put the x and y variable and then you can use all other optional variables to show the desired characters to show the desired shape to show the desired size of those markers and then colour you can give different types of colours for those markers are also.

So, let me show you one plot where we are creating 100 random normal variable with mean of 50 and standard deviation of 10. So, here we are using this plot function and then R norm followed by this 100, so 100 number of sample random number normal number we are we are sampling and then where the mean is 50 and standard deviation is 10. So, this will look like this.

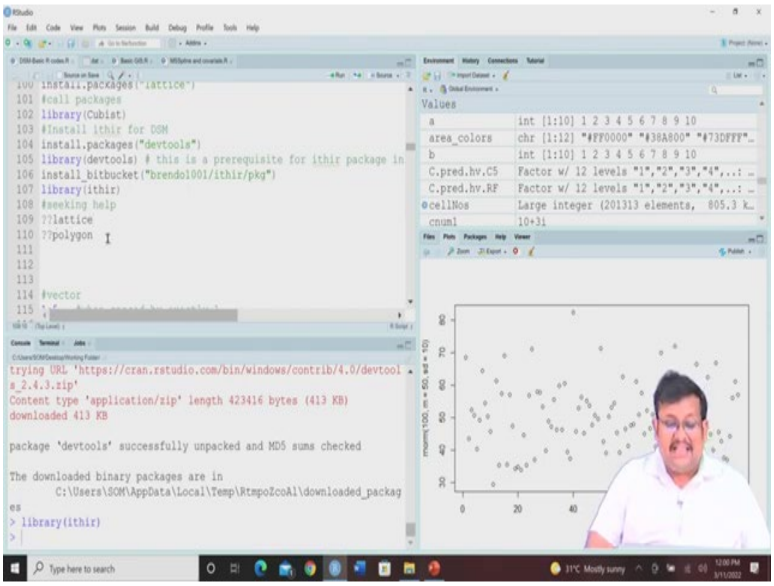
(Refer Slide Time: 16:46)



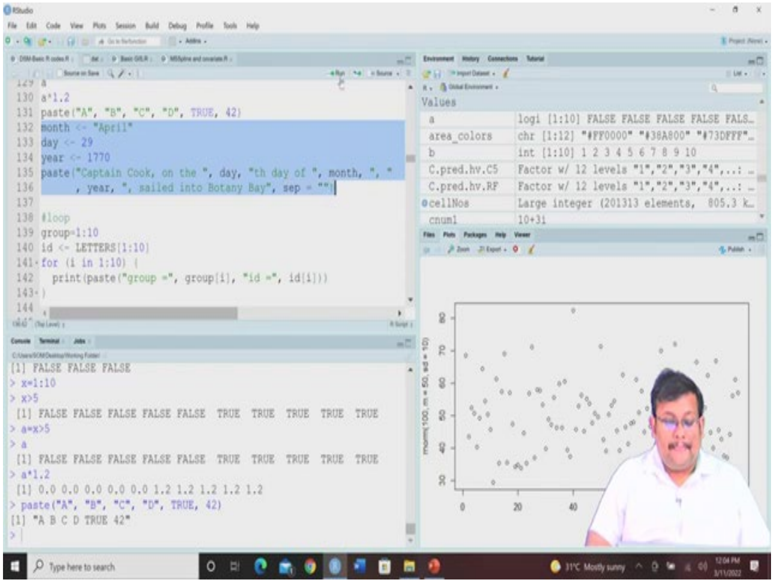
99 install.packages("Cubist")
100 install.packages("lattice")
101 #call packages
[1] FALSE
> # functions,arguments and packages
> sum(1, 12.5, 3.33, 5, 88)
[1] 109.83
> a=1:10
> b=a
> plot(x=a, y=b)
> args(plot)
function (x, y, ...)
NULL
> plot(rnorm(100, m=50, sd=10))
Values
a int [1:10] 1 2 3 4 5 6 7 8 9 10
area_colors chr [1:12] "#FF0000" "#38A800" "#73DFFF"…
b int [1:10] 1 2 3 4 5 6 7 8 9 10
C.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",..:
C.pred.hv.RF Factor w/ 12 levels "1","2","3","4",..:
cellNos Large integer (201313 elements, 805.3 k…
cnum1 10+3i

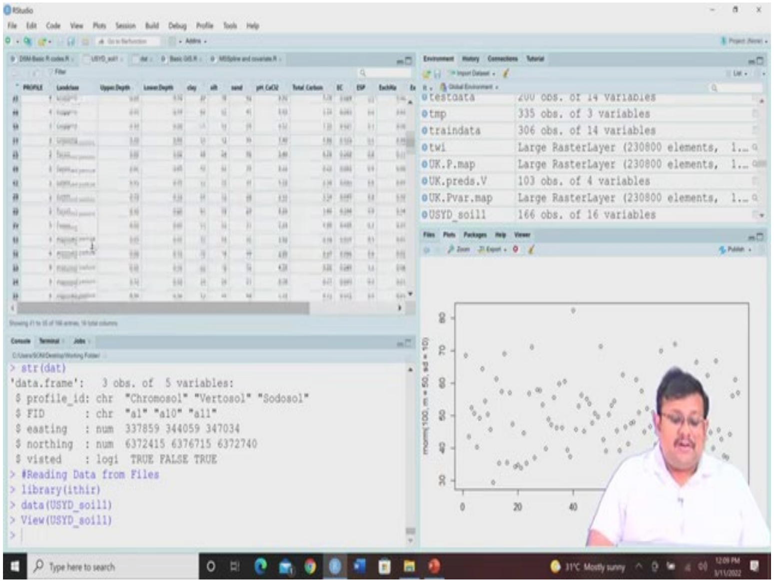


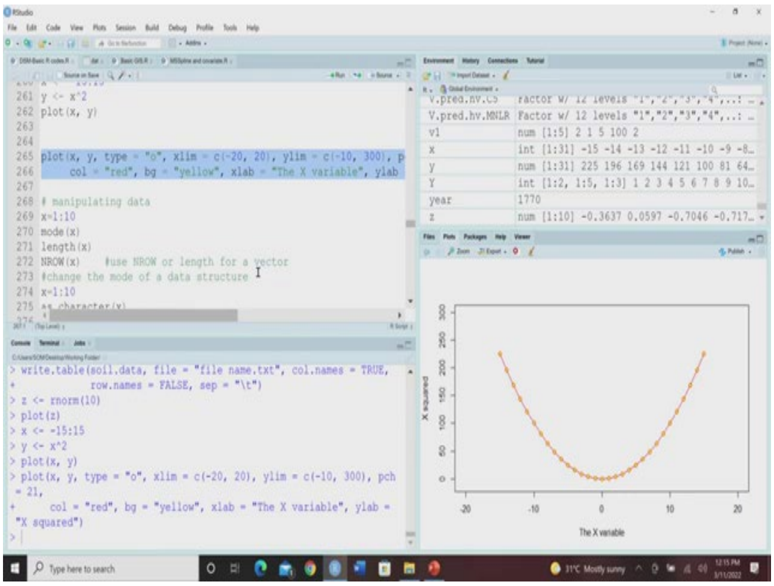
99 install.packages("Cubist")
100 install.packages("lattice")
101 #call packages
[1] FALSE
> # functions,arguments and packages
> sum(1, 12.5, 3.33, 5, 88)
[1] 109.83
> a=1:10
> b=a
> plot(x=a, y=b)
> args(plot)
function (x, y, ...)
NULL
> plot(rnorm(100, m=50, sd=10))
Values
a int [1:10] 1 2 3 4 5 6 7 8 9 10
area_colors chr [1:12] "#FF0000" "#38A800" "#73DFFF"…
b int [1:10] 1 2 3 4 5 6 7 8 9 10
C.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",..:
C.pred.hv.RF Factor w/ 12 levels "1","2","3","4",..:
cellNos Large integer (201313 elements, 805.3 k…
cnum1 10+3i



101 #call packages
102 library(Cubist)
103 #Install ithir for DSM
104 install.packages("devtools")
105 library(devtools) # this is a prerequisite for ithir package in
106 install_bitbucket("brendo1001/ithir/pkg")
107 library(ithir)
108 #seeking help
109 ??lattice
110 ??polygon
114 #vector
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.0/devtools_2.4.3.zip'
Content type 'application/zip' length 423416 bytes (413 KB)
downloaded 413 KB
package 'devtools' successfully unpacked and MD5 sums checked
The downloaded binary packages are in
C:\Users\SOM\AppData\Local\Temp\RtmpoZcoAl\downloaded_packages
> library(ithir)


a*1.2
paste("A", "B", "C", "D", TRUE, 42)
month <- "April"
day <- 29
year <- 1770
paste("Captain Cook, on the ", day, "th day of ", month, ", "
, year, ", sailed into Botany Bay", sep = "")
#loop
group=1:10
id <- LETTERS[1:10]
for (i in 1:10) {
print(paste("group =", group[i], "id =", id[i]))
}
[1] FALSE FALSE FALSE
> x=1:10
> x>5
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
> a=x>5
> a
[1] FALSE FALSE FALSE FALSE FALSE TRUE TRUE TRUE TRUE TRUE
> a*1.2
[1] 0.0 0.0 0.0 0.0 0.0 1.2 1.2 1.2 1.2 1.2
> paste("A", "B", "C", "D", TRUE, 42)
[1] "A B C D TRUE 42"
Values
a logi [1:10] FALSE FALSE FALSE FALSE FALS…
area_colors chr [1:12] "#FF0000" "#38A800" "#73DFFF"…
b int [1:10] 1 2 3 4 5 6 7 8 9 10
C.pred.hv.C5 Factor w/ 12 levels "1","2","3","4",..:
C.pred.hv.RF Factor w/ 12 levels "1","2","3","4",..:
cellNos Large integer (201313 elements, 805.3 k…
cnum1 10+31
rnorm(100, m = 50, sd = 10)


> str(dat)
'data.frame': 3 obs. of 5 variables:
$ profile_id: chr "Chromosol" "Vertosol" "Sodosol"
$ FID : chr "a1" "a10" "a11"
$ easting : num 337859 344059 347034
$ northing : num 6372415 6376715 6372740
$ visted : logi TRUE FALSE TRUE
> #Reading Data from Files
> library(ithir)
> data(USYD_soil1)
> View(USYD_soil1)
testdata 200 obs. of 14 variables
tmp 335 obs. of 3 variables
traindata 306 obs. of 14 variables
twi Large RasterLayer (230800 elements, 1…
UK.P.map Large RasterLayer (230800 elements, 1…
UK.preds.V 103 obs. of 4 variables
UK.Pvar.map Large RasterLayer (230800 elements, 1…
USYD_soil1 166 obs. of 16 variables


y <- x^2
plot(x, y)
plot(x, y, type = "o", xlim = c(-20, 20), ylim = c(-10, 300), p
col = "red", bg = "yellow", xlab = "The X variable", ylab
# manipulating data
x=1:10
mode(x)
length(x)
NROW(x) #use NROW or length for a vector
#change the mode of a data structure
x=1:10
> write.table(soil.data, file = "file name.txt", col.names = TRUE,
+ row.names = FALSE, sep = "\t")
> z <- rnorm(10)
> plot(z)
> x <- -15:15
> y <- x^2
> plot(x, y)
> plot(x, y, type = "o", xlim = c(-20, 20), ylim = c(-10, 300), pch
= 21,
+ col = "red", bg = "yellow", xlab = "The X variable", ylab =
"X squared")
V.pred.hv.MNLR Factor w/ 12 levels "1","2","3","4",..:
v1 num [1:5] 2 1 5 100 2
x int [1:31] -15 -14 -13 -12 -11 -10 -9 -8…
y num [1:31] 225 196 169 144 121 100 81 64…
Y int [1:2, 1:5, 1:3] 1 2 3 4 5 6 7 8 9 10…
year 1770
z num [1:10] -0.3637 0.0597 -0.7046 -0.717…
X squared
The X variable

Now, I told you in my previous lecture also that you need to install certain packages before you are able to execute certain operation in R. So, for that we are going to use some packages. Now, before going to those installing packages, let me show you another very important thing, for R you need to keep all the input file as well as the output file preferably in a working folder.

So you can create a working folder in your desktop and then or any other place and you can keep all the source files and also all the inputs and you can keep all the outputs also in that particular folder. So, by default it will be saved in those particular working folder. So, all the files which are required were all the data files which are required for running the R you can preferably keep that in that working folder.

So, I have already created a working folder if you see my desktop I will show you this is my working folder and this working folder I will be using in this in this in this R program. So, how to do that so you will see that in the R studio there is a session tab, so if you click on this session tab you will get a set working directory.

Now you go to this choose directory and then you just go to the desktop and then you can just select this working folder and then it will be automatically selected, although you can also use this set w d function and followed by this path, so it is another way of setting the working directory.

So, once you do that let me show you installing some of the packages, so either you can go to this package tab and download the package search the package and download the package or you can directly use this install dot packages followed by this package within code. So, we are going to use this cubist package, so you see that when you are going to install it, it will it it will it will take some time and all the required files will be downloaded in the R environment.

And similarly if you want to install this lattice package which also we are I have already installed that so I am not going to reinstall this, so you can you can install. Now, once you install these packages you have to call those packages for calling this packages the function is library, suppose we want to do some cubist operations so we have to call this library cubist each and every time we want to run the cubist.

So, apart from that we also for this for this DSM we are going to use the data sources. So, remember that the data source for all the DSM operation which I am going to show you in this course are already been there in a package called ithir package, I t h i r, so I am going to show you how to install this ithir, package. So, for installing this ithir package you need to first install this package called devtools, so you run this devtool package it will take some time to download once it is download then you can use this ithir package.

So, this ithir package is there in the bit bucket so what is a bit bucket? So, bit bucket is a basically git based source code repository and we can we can we can source code repository and hosting service, so we can we can we can keep we can host or our codes gate based goods in this in this bit bucket.

Now, the question is what is git? The git stands for git, so this gate is basically a software for tracking changes in any set of files which is usually used for coordinating work among programmers collaboratively developing source code during software development. So, it is a kind of a software for tracking changes for any set of files when different programmers collaboratively develop the codes for developing a developing a software or developing any program. So, this is called git.

So, this bit bucket basically hosts the source code this git based source code and this package ithir package this is installed, this is already hosted in this in the bit bucket website, so you can download it from this bitbucket website, I have already downloaded it, so basically you just go here and click the run and you will see it will download. So, I have already installed this so I am not going to reinstall this.

So, once you installed this ithir package the next step is to call this library ithir, so I am going to call this library ithir and so this is how you install the all the data which is required which are required for executing this DSM operation in this course will be is contained in this ithir package. So, you can you can download this ithir and you can call the ithir, remember for doing before doing any operation you have to call this library ithir, each and every time.

So, each and every session I would say so suppose you are having a session right now so you are calling this library ithir and then you can do some operations and in the next session also you have to use again call the library, you can install the package once but you have to call the library in each session.

So, please keep this in keep this in mind you if you install any package once that will be sufficient but for each session you have to up you have to call the library. So, if you if you want to take some if you want to get some more information about this lattice and polygon you can use this double question marks followed by this package and you will have more idea.

Now, let us see one vector which we can create from 1 to 5 and then we can create another vector from minus 10 to 10 with a spacing of 2. So, generally when the spacing is exactly 1 we generally use this colon but when the spacing is 2, then we use this minus 10 to 10 with the spacing of 2 and then we can use this rep package to replicate this 4, 5 times you can see here.

Then c is basically to combine all these values. So, you can see when we are doing this, so it will output like this, now again c let us consider a equal to 2, b equal to 1, c equal to 5, d equal to 100 and e equal to 2 and then you will get this type of output.

And also let us consider v 1 is another vector with the values of 2, 1, 5, 100, 2, so will combine them and let us see how this v 1 look like you will see this 2, 1, 5, 100, 2 and then x let us consider that x equal is equal to 1, 2, 3, with 1 exactly spacing, then y equal to 4 to 10 with 1 exact spacing then we combine both x and y using this c and then we are renaming this combined vector as z, let us see how this z will look like.

So, 1, 2, 3, 4, up to 10 you can see. Similarly, if we want to see that whether x is greater than 5, so if you run it you will see that it is so sorry if x if we consider that x equal to 1 to 10 and then if we want to see that whether x is greater than 5 of course for the values of 1, 2, 3, 4, 5, it will be false but for the values of 6, 7, 8, 9, 10, it will be true. So, you see that how these logical outputs is showing whether your variable is greater than 5 or not or.

And then we can create another variable that is a, which will be a which is a which is x greater than 5. So, let us see the similar output will get, so if you multiply it with any value then we will get that for false will get 0 for true value will get this particular value. So, then we can use this paste command, so in the paste command you can see the codes are basically showing the characteristics and then the logical parameter and then 42 is a numerical parameter, so you will see this will be the output of the paste command.

So, just for an example suppose there is a variable there is a variable called month, so at this time so it is in April and the day equal to 29, so it is a numerical variable and then year is 1770. So, paste if you use this captain cook on the, so this is a character then day is a numerical variable, day of the month is also numerical variable month is a character variable and then year is of course a numerical variable, sailed into the botany bay and then separator is given.

So, let us just run it and see how it looks like, so you can see when you do that then it will represent as captain cook on the 29th day of April 1770 cent into botany bay. So, this is one command, another command is the loop command. So, in the loop command you can group the variables from you can you can do the loop you can you can rename the variables one by one using the letters.

Here as an example if you run it and you will see that groups from 1 to 10, so 1 to 10 group and then you are giving the id from a to z using the letters, so it will the the program will run in loops and it will show the results.

Then vectorized arithmetic, so vectorized arithmetic you can see that let us consider a variable x from 6 to 10 and then if we do this x plus 2 then you will see that these 2 will be added to each of these integers and it will give you the results. So, then y let us consider is a vector and then z let us consider it is x and y, so if you if you see that it will be x and y will be directly combined in the form of z.

So, again you can express any equation suppose x is a vector from 1 to 10 and m is a slope 0.8 and b is a value of 2 that is a offset, so thus the straight line formula that y equal to m x plus b, so if you just do that you will see that the value of y, because x has the values from 1 to 10. So, for all these values will get these values of y by using this equation.

Then if you want to take some suppose this is 1 to 10 and you want to take the square root of x you can you can go ahead and see the values here and then you can use the vector of square root of from 1 to 10 it will get the same result and then you can just put these values by using c you will get the same result, so all these three condition you will get the same results.

Now, let us see a matrix from 1 to taking the values of 1 to 15 a number of rows 5 number of column three, so x will look like this and another matrix let us see, so here you can see that by default they are arranged in columns but if you want to arrange them in along with the rows, so you can use the byrow argument, which is true.

So, here you can see that they will be arranged in rows, so 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, and 13, 14, 15. So, arithmetic means arithmetic with matrices and arrays that have the same dimension, so you can use this matrix command, so you will see that how it matrix 1 matrix with a value of 1 there will be number of columns 5 and number of rows 5 and number of columns will be 3, so you can do all these different types of calculations. So, x plus z you can do different types of calculation.

So, when dimensions do not match you will get some error output, so you can see that when x and z dimensions are not matching you will see these non- confirmable arrays. So, for mixed vectors array arithmetic, data you will have these different types of you can play with these different types of codes, I have already given these annotations.

Now, let us move to data frame, so data frame is very important so let us create a data frame which is the similar which I have created first, so let us create this and let us see the structure so the structure of the data frame the command is very simple. So, str function so you see the data frame three observation five variables profile id character variable, FID character variable, easting numerical variable, northing numerical variable and visited is a logical variable.

And if you want to read the data from the files you call the library ithir and then you call the data USYD soil data, so this USYD underscore soil data is already there in the in the in this in this, in this ithir package.

So, it is if you click on it you will see there are 166 observation with 16 variables starting with profile land class upper depth of the soil lower depth of the soil, clay silt sand ph, measured with calcium chloride total carbon, electrical conductivity exchangeable sodium percentage, exchangeable sodium, exchangeable potassium, exchangeable calcium, exchangeable magnesium and cc.

So, this is a data set for 166 observation which we have which we have just seen and their structure is also, so if you want to see the structure just give the structure command and it will give you all the details of what types of data is there. And if you want to see if you want to use this head command it will give you the first 6 observation of all the variables.

So, and also if you want to see whether any missing value for CEC, CEC is a variable, so you specify that variable using this dollar sign and then you are using that ease dot na command again and you will see that 9th observation 10th observation 45th observation 63 observation and 115th observations are having the missing data for CEC.

If you want to create the data frame manually you create first this you use this c and combine this chromosol, vertical, organosol and anthroposol and then use create another variable carbon and you create another vector you create another vector of carbon, then you create a data frame by combining the soil type equal to soil which you have already created and soil oc is equal to carbon.

So, then you combine them together using in this data frame and you can see that when you when you and give you name that date and you will see that the final output will look like this. If you want to specify or change some names you can use this names dot names of that, so you can change this name to soil and SOC by using the c command. So, you can see that how it looks like.

So, specifying so this you can see this the column names have been changed and specifying row names in the data frame if you want to specify the row names also you can use this row names and then you can give some row names and then you can see the data how it look like. So, you can see now the row names are appearing. Working in the data frame you can use this names command you can see what type of variables are there, work with a specific column.

So, just like before I showed you if I want to use this ESP, I will specify this ESP using this, so dollar sign and you will see all the values of 166 variables are appearing and some of them are missing values. So, if you want to see just take the mean of the ESP it will take the mean of the ESP but the problem is there are some missing values, but so if you want to omit these missing values and then take them in then it will it will give you some results, so 1.99.

So, if you can you can specify by name any particular variable or you can specify the column, so suppose the tenth column, so the first is always number of rows and then the exact column so tenth column if you want to see so it is again it is the tenth column by default if you if you want to see that it will be it will be the tenth is here, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10. So, this will be easy values.

So, if you want to add a new column you can add this the soil suppose there is a new column called upper another column is lower, so you can specify these new names and you can you can you can use this command by just multiplying the existing upper depth and lower depth and now you want to see the head of the soil data, so you can see that this upper and lower these two columns have been added.

So, you if you want to omit the missing values from a vector you first select this and then remove the all the missing values and then you can use this clean data and then so all these type of things you can do and then you can write a table using the soil data that is called file and give the name file name dot txt and column names and row names are also true. So, you run this thing and you can plot the data.

Suppose you want to plot the random normal variable 10 random number variable and so this is the plot and then if you want to have a vector of minus 15 to plus 15 and then you are taking the you are creating y which is basically taking the square of 2 and then you want to plot these x square. So, plot this both x and y, so as expected you will get this type of plot.

So, a plot function also have different types of arguments for example the type of the plots you can have idea about different types of plots are there, then xlim is the limit of the x axis, ylim is the limit of the y axis, pch is different types of markers are there, different size, different types of marker.

So, this plot character the short form is pch, pch 21 will be hollow circle then the column colour you want to give the red and the background you want to give the yellow and x level will be x axis level will be the x variable and y axis level will be the x measured. So, if you do this then you will get this type of results. So, guys let us wrap up our lecture here.

So, I hope that you have some good knowledge of these data frame, vector, packages, library and ithir, and so we will in the next week will start applying these knowledges for going for different types of continuous modelling as well as in in the week 12 will be also showing the how to do the categorical modelling using R. So, I hope that this lecture was helpful, let us meet in the week 11, where we will see how to use the R for categorical for continuous soil property modelling. Thank you very much.