

Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology, Kharagpur
Lecture 53
Digital Soil Mapping with Continuous Variables (Contd.)

(Refer Slide Time: 0:29)



Welcome friends to this third lecture of week 11 of this NPTEL online certification course on machine learning for soil and crop management. And in this week our topic is digital soil mapping with continuous variables.

So, we are trying to see how we can model the continuous variable starting from the simple linear regression to multiple linear regression and advanced machine learning models using specific software, that is R, we have seen the basic R commands already, we have seen the some exploratory analysis with R and also we have seen the basics GIS operations in our of course these are very basics, we can do a lot of things with R.

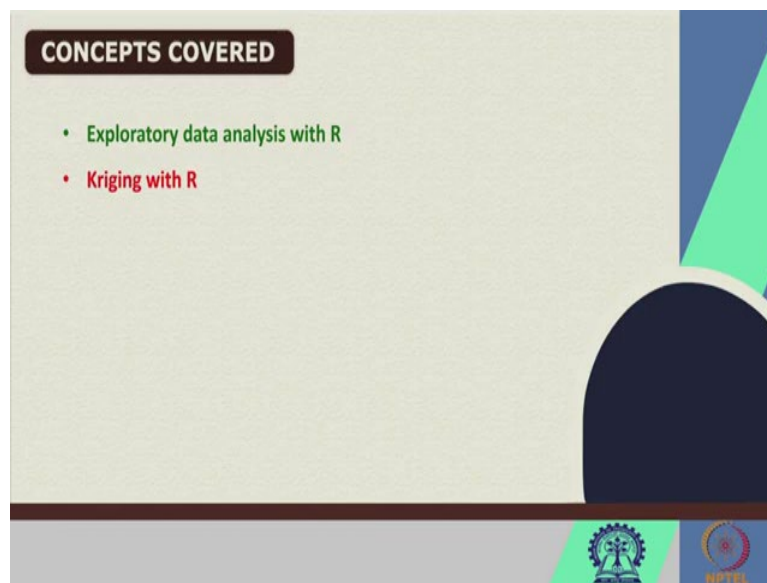
And also we have seen the mass preserving spline and we have also seen the how to extract how to stack different covariates for digital soil mapping and also how to extract the covariates value for particular point observation, how to intercept those values we have seen in our previous lectures.

Now, in this lecture, we are going to explore some more exploratory data analysis and will see their interpretation and also will see how to do trigging interpolation using R, because whenever our major objective for digital soil mapping is producing maps, using different

types of special inference models, but if you do not know how to execute geostatistics the there will not be any, there will not be any fruitful result from the machine learning models.

So, whatever we do, whatever prediction we make using machine learning models, we have to map that and for mapping we need to understand some geostatistical operations, of course these will be again I am telling you guys, there are these are very basics and you need to understand if you want to be a master on this geostatistical or digital soil mapping applications you have to go through several other literature to enrich yourself, but I am just trying to expose you to this new domain using this software.

(Refer Slide Time: 3:23)



So, these are the topics, the concepts which are going to cover using R, that is exploratory data analysis using R and also we are going to see the Kriging using R.

(Refer Slide Time: 3:35)

KEYWORDS

- Kriging
- Normality test
- Skewness
- Kurtosis
- Variogram

The slide features a speaker's video feed in the bottom right corner and logos for IIT Bombay and NPTEL at the bottom.

Now, these are the some of the keywords, which we are going to see in this lecture. Kriging, then normality test, then Skewness, Kurtosis and Variogram. So, we have already seen Variogram before, but we will see how to execute the Variogram in this lecture.

(Refer Slide Time: 3:59)

SKEWNESS

Distortion or asymmetry that deviates from the symmetrical bell curve, or normal distribution, in a set of data.

Skewness value: >1 or <-1 indicates a highly skewed distribution.
Value between 0.5 to 1 or -0.5 to -1 is moderately skewed.
Value between -0.5 and 0.5 indicates that the distribution is fairly symmetrical.

The slide includes a graph with three bell curves: a blue curve for 'positive skewness' (right-tailed), a red curve for 'skewness = zero' (symmetrical), and a green curve for 'negative skewness' (left-tailed). A speaker's video feed is in the bottom right, and IIT Bombay and NPTEL logos are at the bottom.

So, before going to discuss before going to execute the Skewness and kurtosis of any data set, let us see what is what are these statistics. So, Skewness basically shows us the distortion or asymmetry that deviates from the symmetrical bell curve or normal distribution in a set of data, suppose you have a set of data and you want to see whether they match with a normal distribution or they are deviating from a normal distribution. So, in that case we are going to see the Skewness of the data set.

So, here you can see that this is the Skewness 0 that means it follows a proper normal distribution and here you see that it is a positive Skewness distribution and it is the negative Skewness distribution. Now, in both this condition, these are deviating from the original normal distribution.

So, what are the value of Skewness? So, if the Skewness value is greater than 1 or less than 1, then it indicates a highly skewed distribution, so that deviates significantly from a normal distribution and value between 0.5 to 1 and minus 0.5 to minus 1 is moderately skewed and value between minus 0.5 to 0.5 indicates that the distribution is fairly symmetrical. So, this is about the Skewness.

(Refer Slide Time: 5:39)

KURTOSIS

The sharpness of the peak of a frequency-distribution curve.

Leptokurtic
Mesokurtic
Platykurtic

kurtosis identifies whether the tails of a given distribution contain extreme value.

If the value is greater than +1, the distribution is too peaked. Likewise, a kurtosis of less than -1 indicates a distribution that is too flat.

And Kurtosis is basically shows the sharpness of the peak of a frequency distribution curve. So, there are three different condition one is called platykurtic and then another is mesokurtic and then another is leptokurtic. So, mesokurtic generally follows the normal distribution, whereas platykurtic is generally blunt and leptokurtic has higher peak than the normal distribution.

So, what is the implication of Kurtosis? Because, Kurtosis identifies whether the tails of a given distribution contain the extreme values or not, so this is how this Kurtosis is important. If the value is greater than plus 1, then the distribution is too peaked likewise if the Kurtosis is less than minus 1, it indicates the distribution is too flat. So, you do not need these two extremes, so you already know so you know the ideal distribution is the this mesokurtic distribution ok guys.

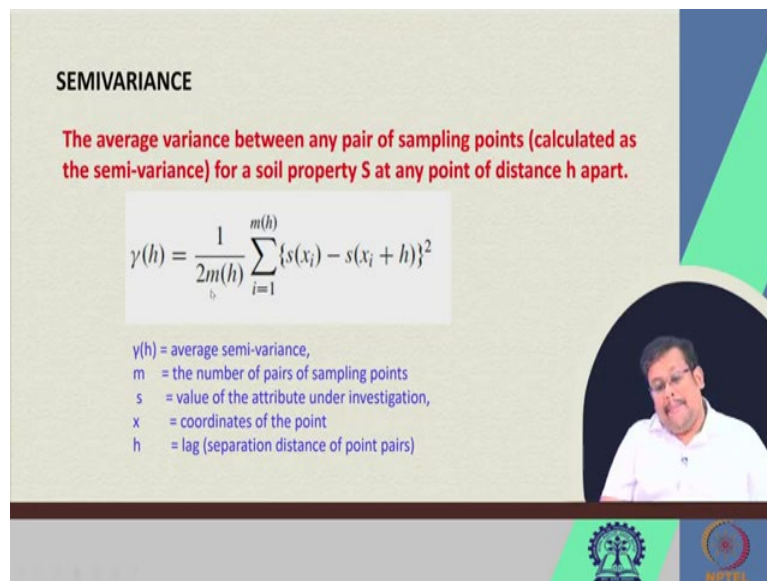
(Refer Slide Time: 6:50)

SEMIVARIANCE

The average variance between any pair of sampling points (calculated as the semi-variance) for a soil property S at any point of distance h apart.

$$\gamma(h) = \frac{1}{2m(h)} \sum_{i=1}^{m(h)} \{s(x_i) - s(x_i + h)\}^2$$

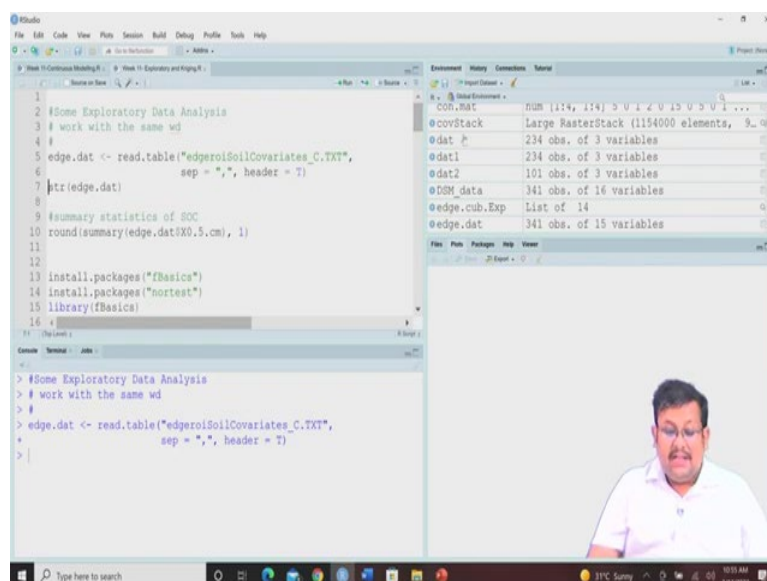
$\gamma(h)$ = average semi-variance,
m = the number of pairs of sampling points
s = value of the attribute under investigation,
x = coordinates of the point
h = lag (separation distance of point pairs)



So, let us go ahead and see another the semi variance you already know that semi variance is the average variance between any pair of the sampling point, it calculates at semi variance. So, this is the average semi variance and here this is the formula of calculating the semi variance $\frac{1}{2m}$, then summation then $s(x_i) - s(x_i + h)$ squares, here this s is a value of attribute under investigation, x is the coordinates of the points.

And then h is the lag distance or distance between the point pairs, m is the number of pairs of sampling points. So, this is how you calculate the semi variance and then you plot the semi variogram or variogram.

(Refer Slide Time: 7:45)



```
1  
2 #Some Exploratory Data Analysis  
3 # work with the same wd  
4 #  
5 edge.dat <- read.table("edgeroiSoilCovariates_C.TXT",  
6 sep = ",", header = T)  
7 #tr(edge.dat)  
8  
9 #summary statistics of SOC  
10 round(summary(edge.dat[X0.5,]), 1)  
11  
12  
13 install.packages("fBasics")  
14 install.packages("nortest")  
15 library(fBasics)  
16 #
```

```
> #Some Exploratory Data Analysis  
> # work with the same wd  
> #  
> edge.dat <- read.table("edgeroiSoilCovariates_C.TXT",  
+ sep = ",", header = T)  
> |
```

con.mat	num [1:4, 1:4] 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1 0 0 1
covStack	Large RasterStack (1154000 elements, 9_000)
@dat	234 obs. of 3 variables
@dat1	234 obs. of 3 variables
@dat2	101 obs. of 3 variables
@DSM_data	341 obs. of 16 variables
@edge.cub.Exp	List of 14
@edge.dat	341 obs. of 15 variables

RStudio interface showing a data table with columns: X100.200.cm, X20.5.cm, X50.5.cm, X100.5.cm, soilDepth, elevation, radK, landsat_b3, landsat_b4. The console shows the following R code:

```

> #Some Exploratory Data Analysis
> # work with the same wd
> #
> edge.dat <- read.table("edgeroiSoilCovariates_C.TXT",
+                       sep = ",", header = T)
> View(edge.dat)

```

The Environment pane on the right lists objects: con.mat (NUM), covStack (Large RasterStack), edat (234 obs. of 3 variables), edat1 (234 obs. of 3 variables), edat2 (101 obs. of 3 variables), edSM_data (341 obs. of 16 variables), edge.cub.Exp (List of 14), and edge.dat (341 obs. of 15 variables).



RStudio interface showing the R code from the previous step followed by summary statistics for the soil depth variable:

```

1
2 #Some Exploratory Data Analysis
3 # work with the same wd
4 #
5 edge.dat <- read.table("edgeroiSoilCovariates_C.TXT",
6                       sep = ",", header = T)
7 str(edge.dat)
8
9 #summary statistics of SOC
10 round(summary(edge.dat[X0.5.cm]), 1)
11
12
13 install.packages("fBasics")
14 install.packages("nortest")
15 library(fBasics)
16
17
18 # X100.200.cm: num 3.13 3.31 1.18 1.85 2.72 ...
19 # soil.depth : int 260 260 260 260 259 260 253 260 ...
20 # elevation : num 186 187 192 193 197 ...
21 # twi : num 22.9 23.5 23.1 22.8 22.2 ...
22 # radK : num 1.122 0.983 0.918 0.954 0.784 ...
23 # landsat_b3 : num 62.3 59.6 67.3 57.9 49 ...
24 # landsat_b4 : num 54.9 51.7 56.1 46.6 39.2 ...
25
26 > #summary statistics of SOC
27 round(summary(edge.dat[X0.5.cm]), 1)
28
29 Min. 1st Qu. Median Mean 3rd Qu. Max.
30 0.3 11.9 16.4 18.9 23.1 93.1
31

```

The Environment pane on the right remains the same as in the previous screenshot.



RStudio interface showing the R code for installing and using the fBasics and nortest packages to calculate skewness and kurtosis:

```

13 install.packages("fBasics")
14 install.packages("nortest")
15 library(fBasics)
16 library(nortest)
17
18 # skewness
19 sampleSKEN(edge.dat[X0.5.cm])
20
21 # kurtosis
22 sampleKURT(edge.dat[X0.5.cm])
23
24 #Normality test
25 ad.test(edge.dat[X0.5.cm])
26
27 #plot untransformed data
28 #

```

The console output shows the results of the skewness and kurtosis tests:

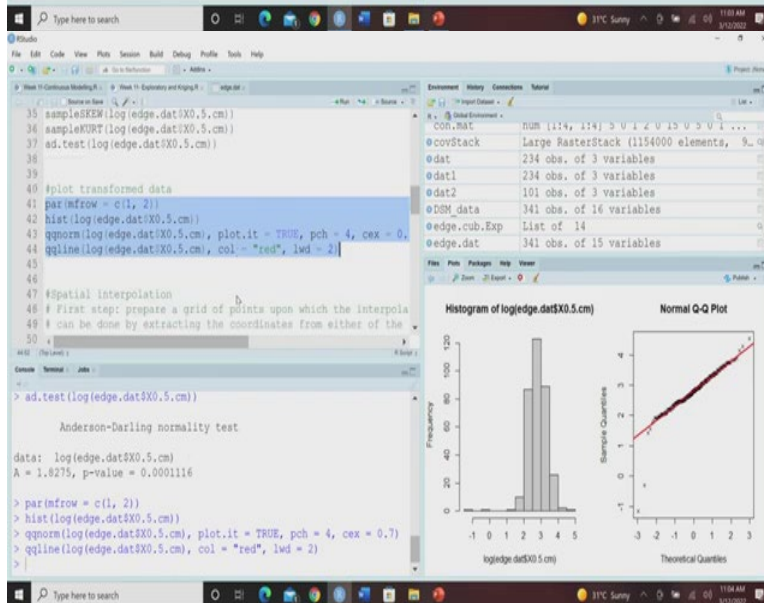
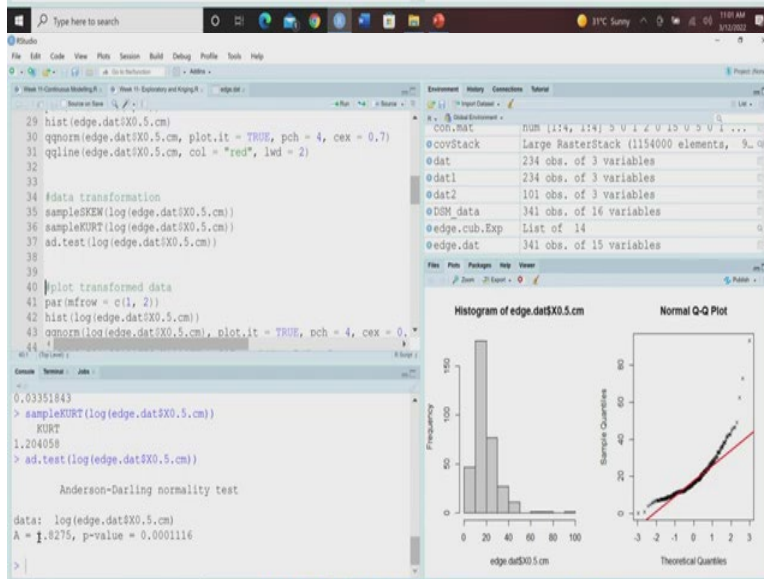
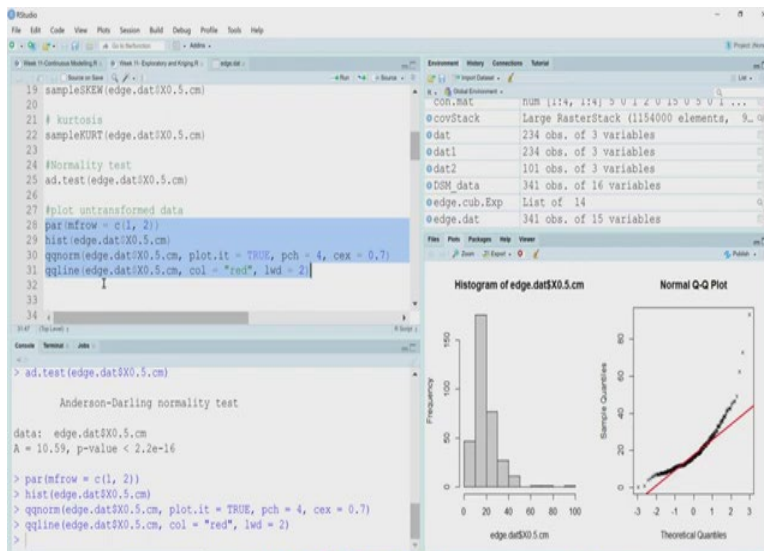
```

Loading required package: timeDate
Loading required package: timeSeries
> library(nortest)
> # skewness
> sampleSKEN(edge.dat[X0.5.cm])
SKEN
0.1964831
> # kurtosis
> sampleKURT(edge.dat[X0.5.cm])
KURT
1.307351

```

The Environment pane on the right remains the same as in the previous screenshots.





Now, let us go and ahead and see how we can execute this thing in R, so we have already covered this, the basics of GIS and some basic data exploratory data analysis in our previous

week. So, we will do, will see more exploratory data analysis in this course. So, will be dealing with this edge roi covariate underscore c dot txt, which we have already created in our last lecture and it is already saved in our working folder. So, we will start from there.

So, first we will read the table that is and we will and this table is edge roi soil covariate dot underscore c dot txt. So, let us run this, so this edge roi, so let us see how it looks like, so this is h dot that there are 341 observation of 15 variables, so we know that this i d easting northing, then the organic carbon values for all the depths, then soil depth, then elevation t w i r a d k lens and v 3 lenses before. So, it is basically the intersection of the point observation as well as the stacked covariate we already know that.

Now, in the next step, so we know this is the structure, but if we want to see the structure and their data type we can see here that id is a character variable, then easting, northing is integer are integers and then all other are numerical variable except the soil depth which is another integer. So, suppose now we are interested on interested to deal with only the top 0 to 5 centimetre depth.

So, the summary statistics of this top 0 to 5 centimetre depth, if you want to see and at the same time if we round off the value to the to 1 decimal point, then we will use this round function followed by this summary function and up to which digit you want to round up these values you can give here. So, here we are going to round of these values up to 1 decimal, so we are giving 1 here, you can give it 2, 2, decimals also.

So, here you can see that this gives us the summary statistics of our data minimum is 0.3 and first quartile 11.9, medium value is 16.4, mean value is 18.9, third quartile is 23.1 and maximum value is 93.1.

Now, for the next set of analysis, for example identifying the Skewness and Kurtosis, we are going to install these packages, we are going to install these packages like f basics and nor test. So, f basics are the, f basics contains some, a f basics contains some normal statistical analysis, which and nor test is basically helps in normality test of the data. So, I have already installed this, so I am not going to further install, I request you to install these two packages. And then let us call this library F basics and library nor test.

So, these two required packages have been uploaded and then let us see the Skewness of this 0 to 5 centimetre depth. So, if you see you use, we use the sample skew function, so we can see this is 0.19, so it is moderately positively Skewness, it is showing the moderately positive

Skewness, it is not perfect, it is not very highly deviation is there, but you can see some moderate deviation positive deviation. If you want to see the Kurtosis, again the function is sample cut and you can see here we are using this 0 to 5 centimetre depth.

Now, so you can see here the Kurtosis is also pretty high, it is more than 1.30. So, if we want to test the normality, now normality can be test by different test, Shapiro wilk test is there, Anderson darling test is there. So, there are different types of test for identifying the normality, whether the data set is normal or not.

So, here we are going to use the Anderson darling test, so the function we are going to use is called ad dot test and then you give the particular data set and specify the variable. So, here it is 0 to 5 centimetre, this is the depth variable, so here we are going to see the ad test value.

So, remember here, if the p value is less than 0.05 at 95 percent confidence interval, then Anderson darling test shows non normal data, if it is more than 0.05, then it shows the normal data. In other words, if the state statistics is less than it will show normality, but if it is high then it will show deviation from a normal distribution. So, here we are getting it is pretty low then 0.05, so we can assume that it is not following a normal distribution.

Now, let us plot this untransformed data or original data and let us see how they looks like. So, we are going to use this hist function for producing the histogram and then we are going to produce this q q normality plot and then we are going to add this q q line also. So, let us do that and let us see how this appears. So, if we do this, then you will see that histogram of this data set is appearing and also here you can see the normal q q plot and the, from the normal q q plot and the histogram you can clearly see that they are skewed, they are not following the normal distribution and so for that we need to do some kind of transformation.

So, of course you can see here the data is not arranged in the 45 degree lining the q q plot, so that shows the deviation from the normal distribution. So, that gives us the indication that we should go ahead with data transformation. So, what type of transformation, there are different types of transformation we have already covered, we can go with the natural log transformation, we can go with the power transformation, we can take square root, so different types of transformations are there.

Now, let us do the logarithmic transformation, so let us see now the Skewness of the data when we transform it using the natural logarithm and you can see here, now the Skewness

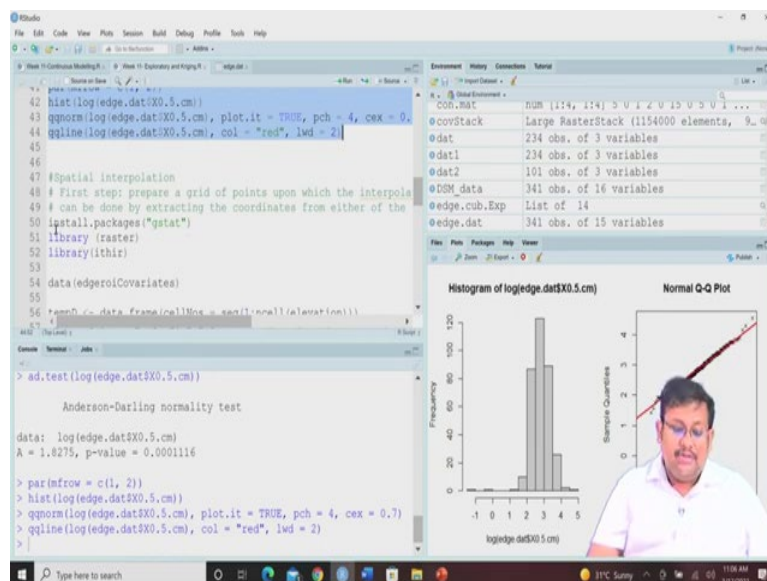
has reduce. So, earlier the Skewness was 0.19 and right now it is 0.03, so we are able to reduce the Skewness by taking the natural log of the data set.

So, let us see the Kurtosis also, so if we use this same function sample Kurt and we use this logarithmic transformed value of this 0 to 5 centimetre, let us see it is 1.20, so earlier it was 1.30, now it is 1.20, so that shows there is a quite improvement, so we are able to make it to reduce this value of Kurtosis and in other words we are try, we were able to reduce the peak of the distribution, so hopefully by running this ad test now it will be possible to see our data is following a normal like distribution.

So, let us run this thing again and you can clearly see, so earlier the value was 10, now it is 1.82, so you can see that yes it is although it is not perfect because we are not getting the now p value greater than 0.05, but it is still very, high than the earlier one, where we are getting 2.2 10 to the power minus 16, but here you can see the p value has considerably increase at the same time the test statistics has considerably decreased from 10 to 1.82, so that shows that this transformation helped us for making the data more normal like.

Now, let us see the plot using this transform data, so we are going to rerun the same script but this time using the logarithmic converted data and let us see how it looks like, so you can see clearly. Now, the histogram is showing more normal like distribution and here you can see the normal q q plot, which is having the q q line is showing the that most of the observations are lying along this q q line. So, that means if we are taking the data transformation helps in the conversion of the non-normal data to normal data.

(Refer Slide Time: 17:41)



RStudio interface showing R code and environment. The code in the console includes:

```

48 # First step: prepare a grid of points upon which the interpolation
49 # can be done by extracting the coordinates from either the
50 install.packages("gstat")
51 library(gstat)
52 library(igraph)
53
54 data(edgeroiCovariates)
55
56 tempD <- data.frame(cellNos = seq(1:nrow(elevation)))
57 tempD$vals <- getValues(elevation) # get the pixels with value
58 tempD <- tempD[complete.cases(tempD), ]
59 cellNos <- c(tempD$cellNos)
60 gXY <- data.frame(xyFromCell(elevation, cellNos, spatial = FALSE))
61
62 # IDW / IDW interpolation
63
64
65 > qqnorm(log(edge.dat$X0.5.cm), plot.it = TRUE, pch = 4, cex = 0.7)
66 > qqline(log(edge.dat$X0.5.cm), col = "red", lwd = 2)
67
68 > library(raster)
69 > library(igraph)
70 > data(edgeroiCovariates)
71 > tempD <- data.frame(cellNos = seq(1:nrow(elevation)))
72 > tempD$vals <- getValues(elevation) # get the pixels with value associated
73 > tempD <- tempD[complete.cases(tempD), ]
74 > cellNos <- c(tempD$cellNos)
75 > gXY <- data.frame(xyFromCell(elevation, cellNos, spatial = FALSE))
76

```

The Environment pane shows the following objects:

- @summary.l: List of 4
- @swiss: 47 obs. of 6 variables
- @tempD: 20133 obs. of 2 variables
- @resdata: 200 obs. of 14 variables
- @tmp: 335 obs. of 3 variables
- @raindata: 306 obs. of 14 variables
- @twi: Large RasterLayer (230800 elements, 1...)

Two diagnostic plots are displayed: a Histogram of log(edge.dat\$X0.5.cm) and a Normal Q-Q Plot. The histogram shows a distribution of values between -1 and 5, with a peak around 2.5. The Q-Q plot shows the sample quantiles following a normal distribution line.

RStudio interface showing R code and environment. The code in the console includes:

```

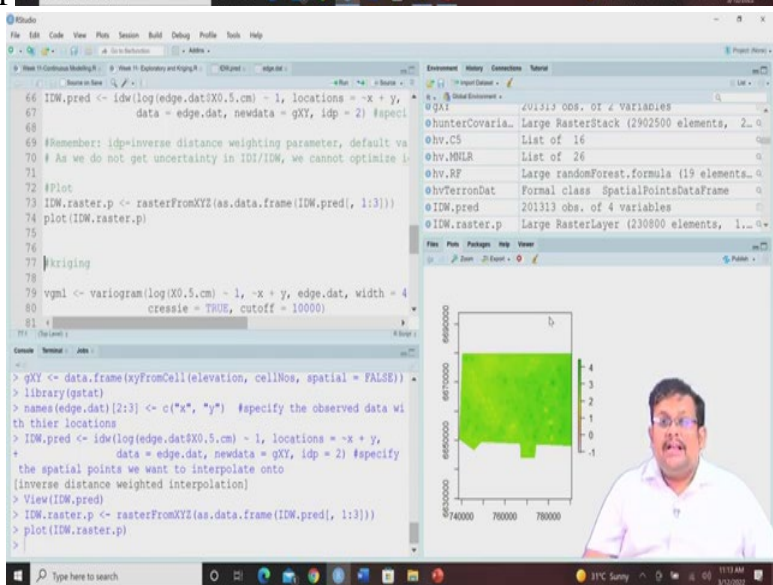
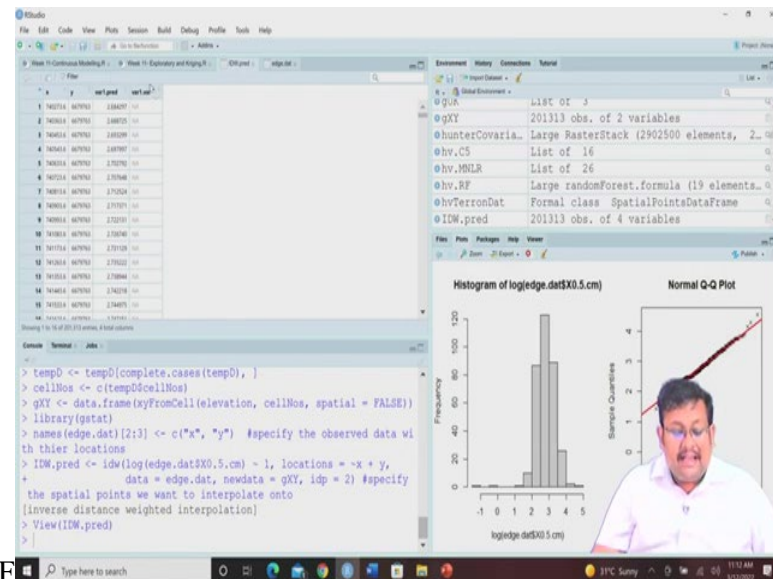
58 tempD[complete.cases(tempD), ]
59 <- c(tempD$cellNos)
60 data.frame(xyFromCell(elevation, cellNos, spatial = FALSE))
61
62 # interpolation
63
64 gstat)
65 pe.dat[2:3] <- c("x", "y") #specify the observed data with their
66 <- idw(log(edge.dat$X0.5.cm) - 1, locations = ~x + y,
67 data = edge.dat, newdata = gXY, idp = 2) #specify the spatial
68
69 # idp=inverse distance weighting parameter, default value is 2,
70 so not get uncertainty in IDW/IDW, we cannot optimize idp
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

The Environment pane shows the following objects:

- @edge.dat: 341 obs. of 15 variables
- @edge.MLR.Full: List of 12
- @edge.MLR.rh: List of 12
- @edge.MLR.Step: List of 13
- @edge.RF.Exp: Large randomForest.formula (18 elements...)
- @edge.RT.Exp: List of 14
- @edgeroi_spln_: Formal class SpatialPointsDataFrame
- @elevation: Large RasterLayer (230800 elements, 1...)

Two diagnostic plots are displayed: a Histogram of log(edge.dat\$X0.5.cm) and a Normal Q-Q Plot. The histogram shows a distribution of values between -1 and 5, with a peak around 2.5. The Q-Q plot shows the sample quantiles following a normal distribution line.



Now, the next step is so we have we are done with the exploratory data analysis, next step is to do some special interpolation. Now, thus for special interpolation it is very important that you should create a grid of points upon which you are going to make the interpolation and these grid of points, for example for doing the special interpolation you require it is a study area or boundary of the study area.

So, how to get that boundary, how to get, so for that you need to produce a grid of points upon which the interpolation will be made and it can be done by extracting the coordinates from either of the covariates, 90 meter resolution raster, which have further to edge roi, because in the edge roi data we have seen edge roi covariate data we have seen there are 4, there are there are 5 covariates, elevation t w i r a d k, then landsat b 3 landsat b 4.

So, among these five all these are having same resolution, so we can take any of them for example and we can extract their grid, because whatever prediction model will be built, they will be built based on those covariate data, because these covariate values will be the predictor for predicting a particular soil property. So, we can check any one of them.

So, let us consider this elevation and then from this elevation we are going to extract those cells which for which we have the values, excluding the cells for which we have missing values. So, after extracting the cells we are going to extract their coordinates and store as a we know will store their coordinates and then we are going to interpolate using Kriging or other methods for that graded region. So, you will see that for that we need to do some kind of operations I am going to show you.

Now, for these geostatistical applications operations you need to install this package called gstat and then once you install this package, then you can call this library raster and then library iqr again. So, we are going to deal with one of the raster file. So, let us first call these data edge roi covariates and then we are going to extract their the cell values of those of the elevation raster file.

So, we are going to select first the elevation raster file, then we are going to see for which cells it has the values and then we are going to extract those cell values and after that we are going to extract their coordinates and save it as a g x y file and then we are going to use this g x y file, which contains the coordinates for producing the interpolation of the organic carbon values for 0 to 5 centimetre using the normal Kriging interpolation or any other Kriging interpolation.

So, here you can see the script which we are going to use for doing this, so you can see first we are going to create a data frame and we are going to name it a temporary data team D. So, data frame by extracting the cell numbers from the elevation file and then we are going to get the values get the pixel for which we have the values, because there will be some pixel with no values. So, we are going to get those values for which pixels for which we have the elevation data and then from there we are going to keep only the complete cases we are going to remove all those missing cases.

And then we are going to extract the cell numbers and then we are going to extract the coordinates from those pixels. So, here for that you can see the for the function is x, y from cell, so we first select the cell for which we have data by excluding the cell for which we do not have the data and after that we take we extract the cell numbers and their spatial

coordinates by and then create a data frame and give it a name called g x y and this g x y will be used for and this x y g these coordinates will be used for further interpolation using inverse distance interpolation or Kriging interpolation.

So, let us start with inverse distance interpolation, sometime we call it inverse distance waiting. So, IDI and IDW are used synonymously, so for this IDI operation we are going to first call this library gstat and then we are going to instruct R that you should understand that these x and y's the second and third column are the x, y coordinates. So, we are specifying the observed data with their location and then we are going to use this IDW for interpolation.

So, the function is simple IDW, we are going to use this log of edge roi data 0 to 5 centimetre and then our locations are of course x and y, we have already instructed R and our data is edge dot dat and then our new data for which we are going to interpolate is g x y remember this g x y is the coordinate. So, for this region of interest we are going to interpolate and then idp, idp stands for inverse distance waiting parameter.

So, here we are going to use a default value of 2, but you can use, you can trial an error with different other values also. So, basically we are this is the function and arguments for this IDW and let us see how it looks like. So, we are going to run that, now remember this idp that is the for which we have taken the value of 2, this inverse distance wattage parameter the default value is 2, but you can play with it and as we do not have, we do not get any uncertainty estimate from IDI and IDW, we cannot optimize this parameter. So, basically it is a trial and error.

Now, that is why IDI is not any more utilized in research and applications, we are we more we are nowadays more focus towards using the Kriging and the advanced variants of Kriging. So, this is how you do the IDI prediction, now you have predicted this but the next step is to produce the plot. So, for plot remember you have to create a raster, because only unless you create a raster file how you are going to plot, so right now it is in a data frame, so this data frame, now this data frame we have to convert this data frame into raster file. So, the function we are going to use is raster from xyz.

So, here our data frame is IDW dot p r e d, so if you click on it this in the environment this has been already created. So, you see x and y and then variable 1 predicted and their variance. Now, since IDW cannot have any variance, so we are getting only the predicted values. So, now we are going to use this first to third column of course fourth column has no data values

or is missing, so we are going to use all the third, first, second, third columns showing the x and y and predicted IDI prediction and then for all the observation.

So, we are going to create the raster file and once we create the raster file, let us plot this raster by using the simple plot function. So, you can see this is the IDI predicted interpolated map of organic carbon for 0 to 5 centimetre. So, this is how you do the IDI or idp interpolation.

(Refer Slide Time: 27:01)

The image displays two screenshots of the RStudio environment, illustrating the workflow for creating a kriging model and generating a raster map.

Top Screenshot: Shows the R console with the following code and output:

```

79 vgm1 <- variogram(log(X0.5.cm) ~ 1, ~x + y, edge.dat, width = 4
80   cressie = TRUE, cutoff = 10000)
81 mod <- vgm(psill = var(log(edge.dat$X0.5.cm)),
82   "Exp", range = 5000, nugget = 0)
83 model_1 <- fit.variogram(vgm1, mod)
84 model_1
85
86 #plot variogram
87 plot(vgm1, model = model_1)
88
89 krig.pred <- krige(log(edge.dat$X0.5.cm) ~ 1, locations = ~x +
90   data = edge.dat, newdata = gXY, model = mode
91
92 par(mfrow = c(2, 1))
93
94
95
96
97
98
99
100
101
102
103

```

The Environment pane on the right shows the following objects:

- @mod: 2 obs. of 9 variables
- @mod.1: List of 14
- @mod.data: 146 obs. of 2 variables
- @mod.rh: List of 14
- @model_1: 2 obs. of 9 variables
- @models: List of 28
- @pred.stack: Formal class RasterStack
- @robs.lv.MLR: num [1:700, 1:12] 8.12e-17 5.56e-08 2.4...

The plot window shows a scatter plot of log(X0.5.cm) versus distance, with a fitted exponential variogram curve.

Bottom Screenshot: Shows the R console with the following code and output:

```

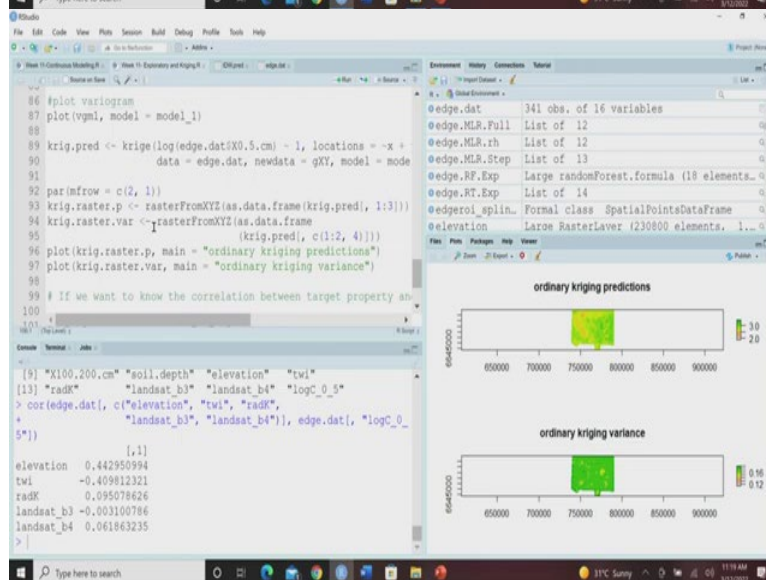
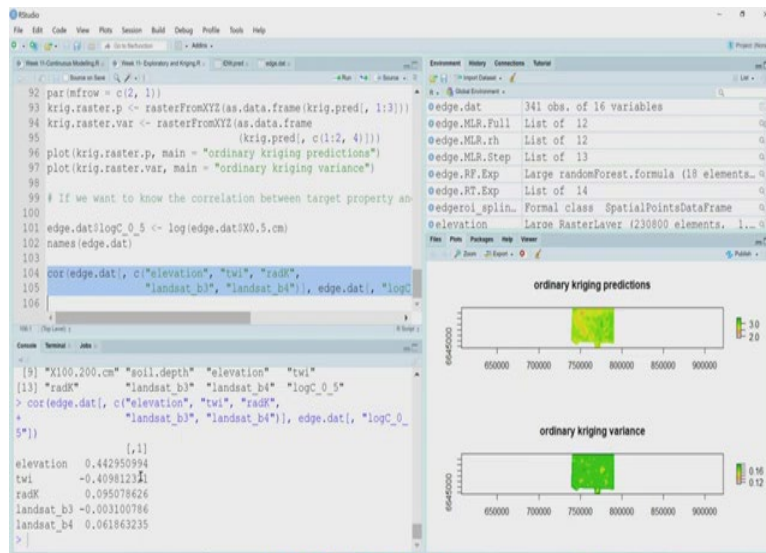
88
89 krig.pred <- krige(log(edge.dat$X0.5.cm) ~ 1, locations = ~x +
90   data = edge.dat, newdata = gXY, model = mode
91
92 par(mfrow = c(2, 1))
93 krig.raster.p <- rasterFromXYZ(as.data.frame(krig.pred[, 1:3]))
94 krig.raster.var <- rasterFromXYZ(as.data.frame
95   (krig.pred[, c(1:2, 4)])
96
97 plot(krig.raster.p, main = "ordinary kriging predictions")
98
99 # If we want to know the correlation between target property an
100
101 edge.dat$logC_0_5 <- log(edge.dat$X0.5.cm)
102 names(edge.dat)
103

```

The Environment pane on the right shows the following objects:

- @krig.raster.v...: Large RasterLayer (230800 elements, 1...)
- @landsat_b3: Large RasterLayer (230800 elements, 1...)
- @landsat_b4: Large RasterLayer (230800 elements, 1...)
- @locModel: List of 14
- @map.C5.c: Formal class RasterLayer
- @map.cubist.rl: Formal class RasterLayer
- @map.MLR: 20133 obs. of 3 variables
- @map.MLR.r: Large RasterLayer (230800 elements, 1...)

The plot window shows two raster maps: "ordinary kriging predictions" and "ordinary kriging variance".



SEMIVARIANCE

The average variance between any pair of sampling points (calculated as the semi-variance) for a soil property S at any point of distance h apart.

$$\gamma(h) = \frac{1}{2m(h)} \sum_{i=1}^{m(h)} \{s(x_i) - s(x_i + h)\}^2$$

$\gamma(h)$ = average semi-variance,
 m = the number of pairs of sampling points
 s = value of the attribute under investigation,
 x = coordinates of the point
 h = lag (separation distance of point pairs)

The next let me just remove it, so that we can get fresh plots. So, now the next is Kriging, so for Kriging first we are going to develop an experimental variogram, because the Kriging

based on variogram model, first we have to develop this variogram model and based on that the Kriging will interpolate the values. So, first you can see we are going to use an experimental variogram, we are going to use some experimental variogram by giving some default parameters.

So, here you can see this variogram is going to develop using this log converted 0 to 5 centimetre and then locations are x and y data set is edge dot dat, width is 400, so this width parameter is giving you the idea at which distance the interval you want your point pairs to you want to average the point pairs to calculate the same variance. So, this is the width parameter, this will be a crassy variogram and the cut off value we are giving the default value of 10 thousand.

So, this is an experimental variogram once we have this experimental variogram, then we are going to fit this experimental variogram using a model. So, here we are going to use this exponential model as you can see here E x p and so once we fit this, this experimental model with exponential model.

So, now you can see we are going to fit our variogram using this experimental model and our selected model, you can of course you can try this with different other models also, like spherical model, bezel model, there are different types of circular model and then you can try you can compare the results also, but here I am showing you the example using the exponential model.

So, you can run it and you can see the model results, of course the model results you can see this is the nugget value 0.10 and then exponential model the p sill stands for partial sill. So, of course this is partial sill is 0.07, the total sill will be nugget plus partial sill, so the total sill will be 0.17 and the range is 1010 meter.

So, we can see here, so if we want to see the variogram how do we want to plot the variogram, simple we are going to use this plot function with v g m 1 and our model is model 1, which is an exponential model. So, you can see this is a variogram, of course from the model we can see it is the nugget value is 0.10, which is quite high and then the partial sill will be 0.7, so the total sill will be 10 plus 0.7 it is 0.07, so it will be 0.17.

So, of course you can see somewhere here we are getting the sill and the range parameter as you can see here will be somewhere here, so that shows that this is how you can create the

variogram by plotting the semi variance along with respect to the distance between the data pairs.

Now, once we have created this variogram, next is producing the prediction values using the Kriging interpolation. So, this Kriging interpolation new data again $g \times y$ for which we are going to predict model in our case this model 1 exponential model. So, let us run it and once we have it once we run this Kriging prediction, then the next step will be let me just remove this, we will do this later, but here you can see after the Kriging prediction we are going to create the raster again from $x \ y \ z$ and also simultaneously we are going to produce the raster of the Kriging variance also.

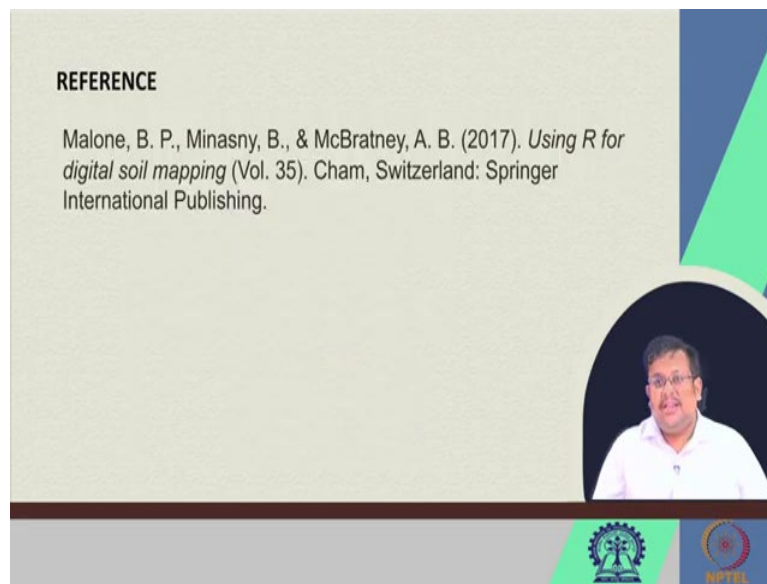
Now, once we have created you can see there are two rasters we have created one Krig dot raster dot p which is a raster from $x \ y \ z$ and this is for the predicted values and this is from the variance. So, you can see not only we can show the prediction but also we can show the variance also. So, let us plot both of them and let us see how they appear. So, you can clearly see that ordinary Kriging predictions will be there at the same time ordinary Kriging variants you can also see simultaneously. So, this is how you can do the Kriging prediction and ordinary and simultaneously you can have the ordinary Kriging variance also.

Now, if you want to know the correlation between the target property and the covariates you can go with this core function. So, let me show you suppose this is an logarithmic of a 0×0 , 0 to 5 centimetre data, so we have already and then let us call let us see their names, so these are the variables which were there and then we want to see the correlation between the logarithmic converted organic carbon for all the observation with the elevation $t \ w \ i \ r \ d \ k$ landsat b 3 landsat b 4. So, for with the all the variables if you want to see then we can clearly see the elevation $t \ w \ i \ r \ d \ k$ landsat b 3, landsat b4.

And so from this we can see the correlation of elevation is highest with the organic carbon followed by the $t \ w \ i$ and in case of $r \ a \ d \ k$ landsat b 3 landsat b 4 we are not getting very high correlation. So, this is how you can do the Kriging interpolation, you can do IDI interpolation and then you can you can do exploratory data analysis.

So, guys I think that you have learnt something new in this lecture and now you will be able to produce your extract the grid points and then do explorative data analysis and you can use IDI or Kriging to interpolate the values. And then you can produce the map and in case of Kriging you can produce a map of prediction at the same time you can produce the map of the variance.

(Refer Slide Time: 33:53)



REFERENCE

Malone, B. P., Minasny, B., & McBratney, A. B. (2017). *Using R for digital soil mapping* (Vol. 35). Cham, Switzerland: Springer International Publishing.

The slide features a light beige background with a dark blue and green geometric design on the right side. A circular video feed in the bottom right corner shows a man with glasses and a white shirt. At the bottom of the slide, there are two logos: the Indian Institute of Technology (IIT) logo on the left and the NIFT logo on the right.

So, this is the reference for this lecture again and thank you for joining, let us meet in our next lecture, where we will start actual modelling with the simple linear regression, multiple linear regression and so on so forth. Thank you guys let us meet in our next lecture.