

Machine Learning for Soil and Crop Management
Professor Somsubhra Chakraborty
Agricultural and Food Engineering Department
Indian Institute of Technology Kharagpur
Lecture – 56
Digital Soil Mapping with Categorical Variables

Welcome friends to this last week of NPTEL online certification course of machine learning for soil and crop management. And in this week, we are talking about digital soil mapping with categorical variables. Although there will be some spillover from our last week of lectures.

So, in the last week we have discussed about the continuous modeling using continuous modeling in digital soil mapping. And how to execute those different types of continuous modelling using R. We have learned about simple linear regression, we have learned about multiple linear regression, we have learned stepwise regression, we have learned how to do the mapping with multiple linear regression and stepwise regression. Also we have learned how to do the soil mapping with decision tree also with random forest so, and also with cubist.

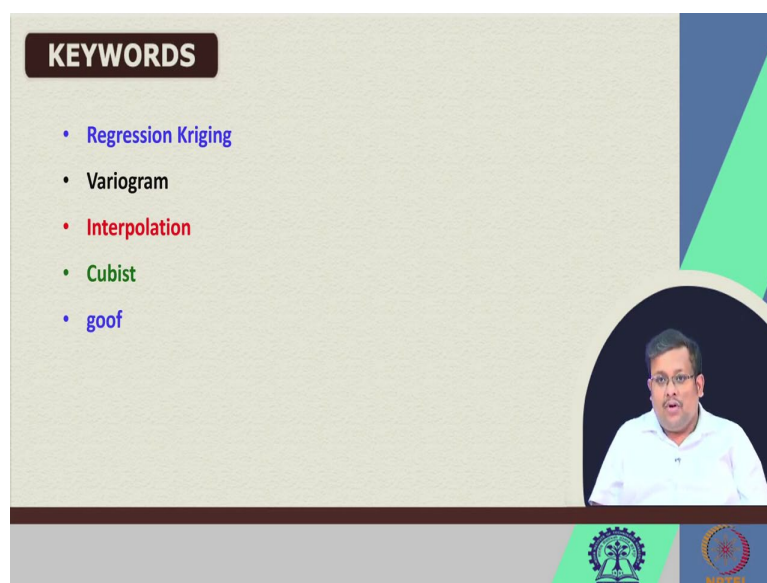
So, we have learned all these things apart from that we have also discussed one hybrid approach that is universal Kriging. Now, in the universal Kriging, remember, it is a combination of both regression and Kriging however, in universal Kriging we generally assume that the regression is always linear. And so that universal Kriging is dependent on some strict assumptions. So, in this week, we are going to focus on the general regression Kriging.

(Refer Slide Time: 2:14)



So, this is the concept which we are going to cover in this week this regression Kriging. And in the regression Kriging I am going to show you what is regression Kriging and why it is important. And secondly, how we execute the regression Kriging in using R software.

(Refer Slide Time: 2:35)



So, these are the keywords for these lectures, for this lecture, lecture number 56 is the first lecture. And these are the keywords regression Kriging, then Variogram, interpolation, cubist goof function. So, of course, these regression Kriging it is a spillover from our last week lectures it is not actually the categorical modeling, but we would like to discuss it before going to the actual categorical modeling in R or in DSN in digital soil mapping.

(Refer Slide Time: 3:10)

Regression Kriging

- The Best Linear Unbiased Predictor of spatial data
- Matheron (1969) proposed that a value of a target variable at some location can be modelled as a sum of the deterministic and stochastic components:


$$Z(\mathbf{s}) = m(\mathbf{s}) + \varepsilon'(\mathbf{s}) + \varepsilon''$$

We know that both deterministic and stochastic components of spatial variation can be modelled separately. By combining the two approaches, we obtain:

$$\hat{z}(\mathbf{s}_0) = \hat{m}(\mathbf{s}_0) + \hat{\varepsilon}(\mathbf{s}_0)$$
$$= \sum_{k=0}^p \hat{\beta}_k \cdot q_k(\mathbf{s}_0) + \sum_{i=1}^n \lambda_i \cdot e(\mathbf{s}_i)$$

where

- $\hat{m}(\mathbf{s}_0)$ = fitted deterministic part
- $\hat{\varepsilon}(\mathbf{s}_0)$ = interpolated residual
- $\hat{\beta}_k$ = estimated deterministic model coefficients
- λ_i = kriging weights determined by the spatial dependence structure of the residual and where residual at location \mathbf{s}_i



So, this is the regression Kriging approach and remember that regression Kriging is considered as the best linear unbiased predictor of spatial data and Matheron in 1969 propose that a value of a target variable at some location can be modeled as a sum of the deterministic and stochastic components we know that.

According to the universal model of variation, we know that the value of a target variable at any given location can be modeled as a sum of the deterministic as well as the stochastic component. Now, from the mapping point of view, we may not be very much interested to deal with the pure noise because this is, this can be measurement error.

So, this is the deterministic part and this is a stochastic part. So, we know that both deterministic and stochastic components of spatial variation can be modeled separately, we can model the deterministic component using regression model. However, the stochastic component can be modeled using Kriging interpolation or in a Kriging interpolation.

Now, combining these two approaches, we obtain this type of representation. So, this is the final prediction. And here you can see we have this is the deterministic part this is the stochastic part and we can model this deterministic part by using the regression equation. And here we can fit the stochastic part using the ordinary Kriging interpolation.

Where in this expression of course, please understand that this is the fitted deterministic part and this is the interpolated residuals. This $\hat{\beta}_k$ is basically estimated deterministic model coefficient is the deterministic model and these are the deterministic model coefficient and

then these λ_i is a Kriging was determined by the special dependence structure of the residuals and where e_{S_i} is the residual at location S_i .

So, from here you can understand that using this regression free hybrid approach, it is possible to capture both the deterministic component as well as the stochastic component to objectively map the spatial variability of any continuous variable. And this is the most appropriate way to predict and map the special variability of the soil properties.

Because it not only takes care about the deterministic component, but also takes care of the stochastic component and holistically it addresses the special autocorrelation or special dependence. Due to this special dependence there is a we can say that the samples which we are taking are, can show some dependence up to a certain distance after that there is we can consider after the range distance, we can consider that the samples are more or less independent.

So, if we want to make a realistic representation of the special structure, we have to take care of this dependent part as well as the independent part because any sample which you take below the range distance that will be considered as dependent with your initial samples.

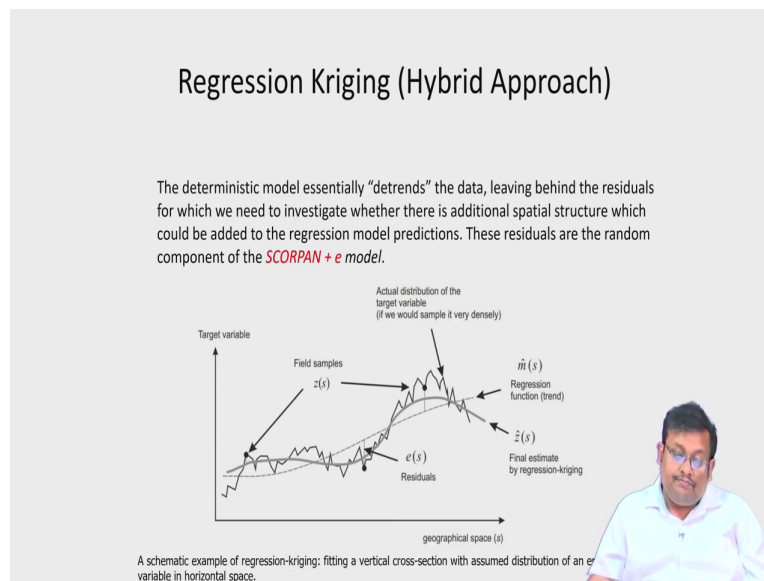
And so, this dependency will violate the assumption of linear regression, which says that the observations are independent to each other. So, here observations are not independent to each other here observations are related up to the range distance and after that of course, they are independent to each other.

So, due to the special covariation due to the spatial autocorrelation due to the spatial dependence it is very important that the spatial structure of the soil properties should not be consider, should not be captured using only the simple the models which depends on the independent target variable.

However, here we can see that the target variable is somewhat both the dependent as well as independent or observation I would say instead of observed instead of variable I would say the observations are independent and also dependent to each other depending on the spatial autocorrelation.

So, that is why we generally go for these regression Kriging and in contemporary DSM modeling, this regression Kriging has been widely popular and you will see that in most of the cases scientists are using this regression Kriging for capturing the spatial structure of the soil properties.

(Refer Slide Time: 8:35)



Now, if we see the graphical representation of this regression Kriging hybrid approach, we can see that if we just represent the target variable the value of the target variable in the y axis and the geographical space the distance in the x axis then we can see that these line is showing the field samples which shows some noise that is noise corrupted this field samples.

Now, of course, if we fit a regression function which is denoted by this dashed line the idea behind fitting this regression function is to detrend the data because this is a deterministic model and using this deterministic model we generally want to detrend the data. So, if the predictive variables appeared somewhere in this regression function and this is the original observation so, of course, this distance will be the error or the residual.

Now, if we interpolate these residuals so, this is the field samples again and this dashed line is a regression function. So, the difference linear difference between the any point to this to their corresponding point on the regression function is known as the residual. And so, if we interpolate this residual and add to this regression function, you can say that ultimately we will get this final estimate by regression Kriging.

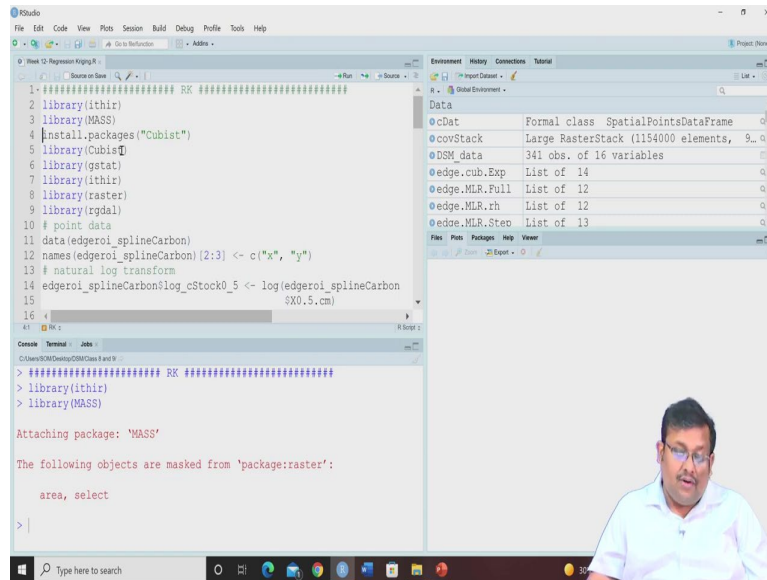
So, this final estimate by regression Kriging will try to model the variation using both regression or deterministic function as well as the stochastic function. So, this is the regression Kriging representation graphical representation. So, again remember that this deterministic model the detrends the data and then leaving behind the residuals.

So, the original actual sample distribution of the target variable can be seen more noise corrupted and so, for which we need to and these residual we need to investigate for

additional spatial structure and then we add to the deterministic component. So, these residuals are the random component in this SCORPAN plus e model.

Remember we talked about SCORPAN plus e model. So, these e or error is basically this residual. So, this is how we captured this regression Kriging in terms of these this graphic, it graphically we can capture this regression Kriging.

(Refer Slide Time: 11:45)

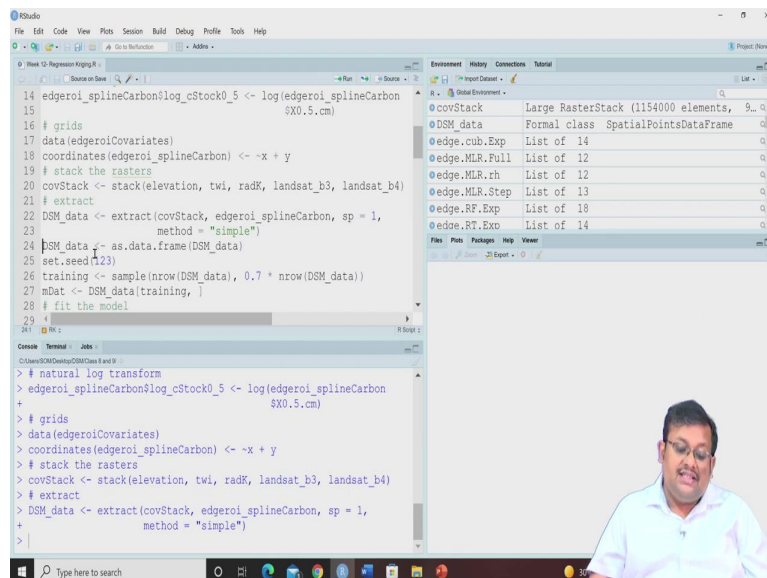


```
1 ##### RK #####
2 library(ithir)
3 library(MASS)
4 install.packages("Cubist")
5 library(Cubist)
6 library(gstat)
7 library(ithir)
8 library(raster)
9 library(rgdal)
10 # point data
11 data(edgeroi_splineCarbon)
12 names(edgeroi_splineCarbon)[2:3] <- c("x", "y")
13 # natural log transform
14 edgeroi_splineCarbon$log_cStock0_5 <- log(edgeroi_splineCarbon
15 $X0.5.cm)
```

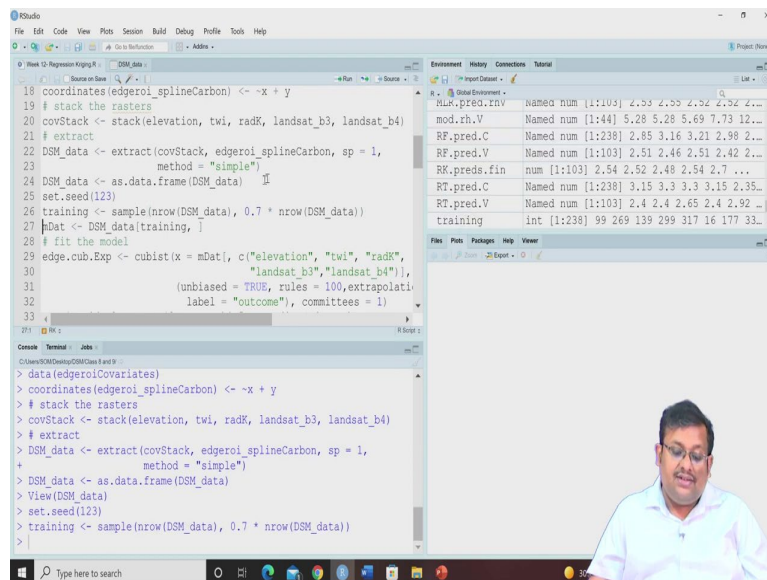
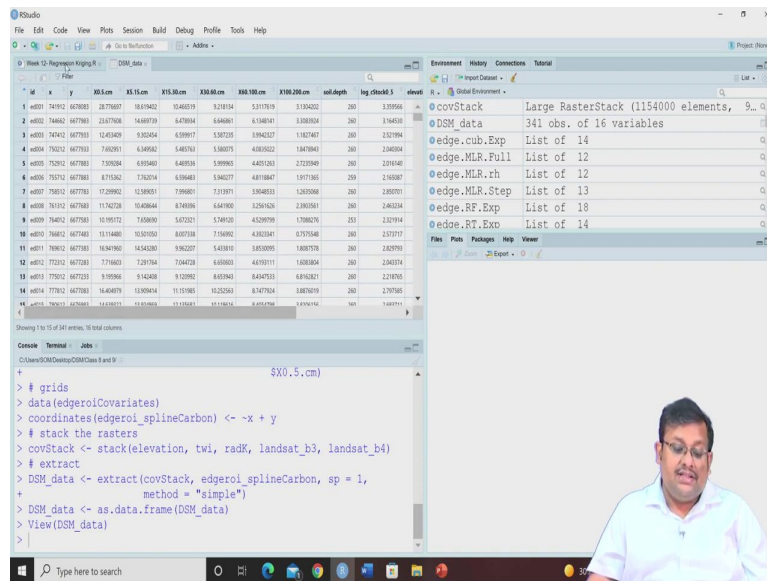
Attaching package: 'MASS'

The following objects are masked from 'package:raster':

- area, select



```
14 edgeroi_splineCarbon$log_cStock0_5 <- log(edgeroi_splineCarbon
15 $X0.5.cm)
16 # grids
17 data(edgeroiCovariates)
18 coordinates(edgeroi_splineCarbon) <- ~x + y
19 # stack the rasters
20 covStack <- stack(elevation, twi, radK, landsat_b3, landsat_b4)
21 # extract
22 DSM_data <- extract(covStack, edgeroi_splineCarbon, sp = 1,
23 method = "simple")
24 DSM_data <- as.data.frame(DSM_data)
25 set.seed(123)
26 training <- sample(nrow(DSM_data), 0.7 * nrow(DSM_data))
27 nDat <- DSM_data[training, ]
28 # fit the model
```



Now, let us go ahead and see how we can execute these regression Kriging in R. So, this is the code. So, again we are going to call these library `ithir`. And then we are going to call this library `mass` we have installed this package `cubist` we have already installed it previously, so, I am going to use it the library `cubist` and then library `gstat` because we are going to fit the Variogram, you know it.

And then let us call the library `ithir` library `raster` because we are going to deal with different raster layers and also `rgdal` a library `rgdal`. So, that we can execute different types of GIS operations. So, for this regression Kriging approach we are going to use the `edgeroi`'s `splinecarbon` data which we have used previously.

So, let me call these data `edgeroi`'s `splinecarbon` again and then let us know that these `edgeroi`'s `splinecarbon` is 341 observation with 10 variables. So, let us instruct R that the

second and third column are considered as x and y are the coordinates. Now, we are doing the natural log transformation of the 0 to 5 centimeter organic carbon data we have done it previously also.

So, we have converted this organic carbon stock of 0 to 5 centimeter by taking the natural log and then we are going to download the data of the covariates that is edgeroi covariates. So, and then we are also we want to see the coordinates, we want to also see the coordinates of the x and y.

So, we are instructing R that you should understand these x and y are the coordinates of the edgeroi's splinecarbon. Now, as usual, we are going to stack the rasters using the stack function and we are going to name it as covstack. So, here you can see we are stacking these elevation twi radk landsat b3 landsat b4 together.

So, just like previously using the stack function, then we are going to as usual we are going to extract the we are going to extract the data using the extract function. So, for that, we are going to use the extract function our data set is covstack and we are going to extract these based on the edgeroi's splinecarbon and the myth simple extraction will be there and then DSM data of course, once we do that, then we are going to save this as a simple data frame.

So, if we save this as simple data frame like previously you can see this is 341 observation with ID, x and y of course coordinates then all the organic carbon values at different depths, and then we can see soil depth and then finally, we are going to see the log converted organic carbon values of 0 to 5 centimeter followed by 5 covariates just like previous exercise. So, our data preparatory step is completed.

Now, the next step is to execute the modeling. So, we are going to set the seed at 1, 2, 3 what is the meaning of setting seed and then just like previously, we are going to fit the training model using 70 percent of the data by selecting the 70 percent of the rows by that dumbly so, this is called the training data.

(Refer Slide Time: 15:53)

RStudio interface showing R code for data extraction and model fitting. The code includes:

```
21 # extract
22 DSM_data <- extract(covStack, edgeroi_splineCarbon, sp = 1,
23                    method = "simple")
24 DSM_data <- as.data.frame(DSM_data)
25 set.seed(123)
26 training <- sample(nrow(DSM_data), 0.7 * nrow(DSM_data))
27 mDat <- DSM_data[training, ]
28 # fit the model
29 edge.cub.Exp <- cubist(x = mDat[, c("elevation", "twi", "radK",
30                                "landsat_b3", "landsat_b4")],
31                      (unbiased = TRUE, rules = 100, extrapolati
32                       label = "outcome"), committees = 1)
33 mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
34                                             newdata = mDat)
35 mean(mDat$residual)
36
```

The Environment pane shows the following objects:

- @mDat: 238 obs. of 16 variables
- @mod: 2 obs. of 9 variables
- @mod.l: List of 14
- @mod.data: 146 obs. of 2 variables
- @mod.rh: List of 14
- @model.l: 2 obs. of 9 variables
- @pred.stack: Formal class RasterStack
- @radK: Large RasterLayer (230800 elements, 1...

RStudio interface showing R code for coordinate extraction and data extraction. The code includes:

```
21 # extract
22 DSM_data <- extract(covStack, edgeroi_splineCarbon, sp = 1,
23                    method = "simple")
24 DSM_data <- as.data.frame(DSM_data)
25 set.seed(123)
26 training <- sample(nrow(DSM_data), 0.7 * nrow(DSM_data))
27 mDat <- DSM_data[training, ]
28 # fit the model
29 edge.cub.Exp <- cubist(x = mDat[, c("elevation", "twi", "radK",
30                                "landsat_b3", "landsat_b4")],
31                      (unbiased = TRUE, rules = 100, extrapolati
32                       label = "outcome"), committees = 1)
33 mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
34                                             newdata = mDat)
35 mean(mDat$residual)
36
```

The Environment pane shows the following objects:

- @mDat: 238 obs. of 16 variables
- @mod: 2 obs. of 9 variables
- @mod.l: List of 14
- @mod.data: 146 obs. of 2 variables
- @mod.rh: List of 14
- @model.l: 2 obs. of 9 variables
- @pred.stack: Formal class RasterStack
- @radK: Large RasterLayer (230800 elements, 1...

RStudio interface showing a data table and residual calculation. The data table includes columns: id, x, y, BS.0cm, BS.15cm, BS.30cm, BS.60cm, BS.100cm, BS.200cm, soil.depth, log_cStock0_5, elevat.

id	x	y	BS.0cm	BS.15cm	BS.30cm	BS.60cm	BS.100cm	BS.200cm	soil.depth	log_cStock0_5	elevat
89	40300	77312	665383	1310734	1035747	735486	4487742	3471742	2378954	260	263484
89	4055	76812	665383	2306143	1846495	6368759	3198558	1560395	1301329	460	177500
116	4748	76915	665383	2443913	1832059	6309717	3143865	1544213	6405558	100	320944
165	4487	77422	668363	2141024	1547812	2262573	3366123	2148873	1252988	110	318132
117	4862	76952	667053	2143304	1448020	8263287	3006265	1588234	2353960	260	395416
16	4074	76432	667983	1646106	1346973	6200288	10370100	8193870	5476456	260	260994
4178	74912	665483	2789425	2310889	1856346	17477394	16403975	14148927	160	332002	
163	4638	76482	664423	4177668	3124	6773285	61969757	10114205	360	1428116	
141	4818	76112	665383	1546252	1174422	785628	6247324	4491456	4188407	260	276869
152	4538	77452	668183	1475362	1248274	6532863	9304286	6959760	4318541	260	269196
167	4652	76622	666953	2443913	1832059	6309717	3143865	1544213	6405558	100	320944
4175	76912	668363	2306143	1846495	6368759	3198558	1560395	1301329	3700511	260	176768
223	4624	76422	665123	2218281	1732407	1474963	11158271	1538247	3124204	260	150606
4179	77112	663383	1812248	1533439	9491529	22968959	6396660	6421489	260	269751	
4	4074	76432	667983	1646106	1346973	6200288	10370100	8193870	5476456	260	260994

The Console shows the following code and output:

```
> edge.cub.Exp <- cubist(x = mDat[, c("elevation", "twi", "radK",
+                                     "landsat_b3", "landsat_b4")], y
+ mDat$log_cStock0_5, cubistControl
+ (unbiased = TRUE, rules = 100, extrapolation
+ = 5, sample = 0,
+ label = "outcome"), committees = 1)
> mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
+                                             newdata = mDat)
> mean(mDat$residual)
[1] 5.819761e-09
> View(mDat)
```

RStudio

```

R Console:
> edge.cub.Exp <- cubist(x = mDat[, c("elevation", "twi", "radk",
+ "landsat_b3", "landsat_b4")], y
+ mDat$log_cStock0_5, cubistControl
+ (unbiased = TRUE, rules = 100, extrapolation
+ = 5, sample = 0,
+ label = "outcome"), committees = 1)
> mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
+ newdata = mDat)
> mean(mDat$residual)
[1] 5.819761e-09
> View(mDat)

```

Environment: Global Environment, @map.MLR.rl: up, @map.MLR.rl: Formal class 'RasterLayer', @map.RF.rl: Formal class 'RasterLayer', @map.RK1: Formal class 'RasterLayer', @map.RK2: Formal class 'RasterLayer', @map.RK3: Formal class 'RasterLayer', @map.RT.rl: Formal class 'RasterLayer', @mDat: 238 obs. of 17 variables

RStudio

```

R Console:
31 (unbiased = TRUE, rules = 100, extrapolation
32 label = "outcome"), committees = 1)
33 mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
34 newdata = mDat)
35 mean(mDat$residual)
36
37 coordinates(mDat) <- ~x + y
38 crs(mDat) <- "+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84
39 +units=m +no_defs"
40
41 vgm1 <- variogram(residual ~ 1, mDat, width = 250, cressie = TRU
42 cutoff = 10000)
43 mod <- vgm(psill = var(mDat$residual), "Sph", range = 5000,
44 nugget = 0)
45 model_1 <- fit.variogram(vgm1, mod)
46
47
48 (unbiased = TRUE, rules = 100, extrapolation
49 +
50 label = "outcome"), committees = 1)
51 mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
52 newdata = mDat)
53 mean(mDat$residual)
54 [1] 5.819761e-09
55 View(mDat)
56 coordinates(mDat) <- ~x + y

```

Environment: Global Environment, @map.MLR.rl: up, @map.MLR.rl: Formal class 'RasterLayer', @map.RF.rl: Formal class 'RasterLayer', @map.RK1: Formal class 'RasterLayer', @map.RK2: Formal class 'RasterLayer', @map.RK3: Formal class 'RasterLayer', @map.RT.rl: Formal class 'RasterLayer', @mDat: Formal class 'SpatialPointsDataFrame'

RStudio

```

R Console:
34 newdata = mDat)
35 mean(mDat$residual)
36
37 coordinates(mDat) <- ~x + y
38 crs(mDat) <- "+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84
39 +units=m +no_defs"
40
41 vgm1 <- variogram(residual ~ 1, mDat, width = 250, cressie = TRU
42 cutoff = 10000)
43 mod <- vgm(psill = var(mDat$residual), "Sph", range = 5000,
44 nugget = 0)
45 model_1 <- fit.variogram(vgm1, mod)
46 model_1
47
48 # Residual kriging model
49 rkr <- predict(model_1, mDat, newdata = mDat)
50
51 (unbiased = TRUE, rules = 100, extrapolation
52 +
53 label = "outcome"), committees = 1)
54 mDat$residual <- mDat$log_cStock0_5 - predict(edge.cub.Exp,
55 newdata = mDat)
56 mean(mDat$residual)
57 [1] 5.819761e-09
58 View(mDat)
59 coordinates(mDat) <- ~x + y
60 crs(mDat) <- "+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84
61 +units=m +no_defs"

```

Environment: Global Environment, @map.MLR.rl: up, @map.MLR.rl: Formal class 'RasterLayer', @map.RF.rl: Formal class 'RasterLayer', @map.RK1: Formal class 'RasterLayer', @map.RK2: Formal class 'RasterLayer', @map.RK3: Formal class 'RasterLayer', @map.RT.rl: Formal class 'RasterLayer', @mDat: Formal class 'SpatialPointsDataFrame'

We are calling it a mdat or model data by 70 percent training samples and then we are going to fit the model deterministic model using the Cubist algorithm. Now, in the Cubist algorithm, we have already seen previous lecture that it is a data mining advanced machine learning approach, which can mind that nonlinear regression.

But also in each of these cluster or each of this turning node terminal node it can fit the ordinary least squares regression. So, it is basically a hybrid approach between the nonlinear as well as linear regression. So, we are going to use these mdat and then our, so, this cubist here x is of course model data and here our target our predictors are elevation twi radk landsat b3 landsat b4.

And of course, we are going to give the default parameters of course the rules are we are giving 100 but it depends the model will select the number of rules based on its own optimization and then the extrapolation is 5 and sample 0 and we want to grow the committee's up to 1.

So, here we let us run this and based on this cubist prediction, based on this cubic prediction, we are going to calculate the residuals how to calculate the residuals, the calculate the residuals this is the model data. Now, how to calculate the residual? Residual is calculated by subtracting the predicted values from the model from the original values.

So, here original values is mDat block converted carbon stock of 0 to 5 centimeter we know that minus if we subtract the predicted values by using this particular model which you have created in the previous step, then we will get the residuals. So, let us calculate this residual because these residuals are going to help us for Kriging interpolation.

Now, if you see the mean of these residuals, then you will see it is 5.81 10 to the power minus 09. So, these are the means of the residuals. Now, in this mDat if you see, if we click on the same data of course they will see that a 238 observations and this mDat since it is a 70 percent of the 348, 341 observation, so, it is 238 observations and you can see these are the, their values.

So, you can see along with this mdat we have already calculated there, there is an added column with the residual also. So, once we do that, we are instructing so, far it is in tabular format. So, it is in data simple data frame. Now, to execute special operation we need to convert it to the special points data frame.

So, to convert it to the special points data frame, we are going to instruct R that you consider these x and y as the coordinates. So, we are giving these coordinate function and these x and y and then you will see that the coordinate reference system let us see what is the coordinate reference system of this. Let us assume that this coordinate reference system of this mDat is UTM zone 55 South WGS 84.

So, we are assigning this coordinate reference system to the model data. So, once we have assigned the coordinate reference system of this model data, now, we are going to fit our experimental Variogram and then we are going to fit that experimental Variogram with the model.

So, here you can see we are going to first develop our experimental Variogram which is called VGM 1 using the Variogram function and we are going to do that using the residual because that this Kriging interpolation will be built based on these residuals. So, here we are going to use this residual and our data is model data and then just like previously, we are going to fix these default parameters like width 250 cressle Variogram and then cutoff value is 10,000.

So, we are going to use this. So, this is the experimental Variogram once we fit the experimental Variogram, next we are going to fit that Variogram using a spherical model earlier we use the exponential model now, let us use the spherical model. So, you can see we are going to use this spherical model and then and from that now it has been fit this Variogram has been fitted with the with the spherical model.

So, we are going to fit this. Now, let us see how this model looks like. So, you can see the nugget value is 0.017 then the partial 6 sill of the spherical model is 0.13 and the range is 489.58 meter. So, once we have developed this model once we have fitted this model using the spherical model, let us go ahead and do the regression Kriging.

So, the residual Kriging model you can see we are going to use the gstat function and then let us go ahead and interpolate this and let us see the coordinate reference system of our model data it is of course UTM zone 55 South WGS 84 we know that.

(Refer Slide Time: 22:25)

```
47 # Residual kriging model
48 gRK <- gstat(NULL, "RKresidual", residual ~ 1, mDat,
49             model = model_1)
50
51 crs(mDat)
52 # External Validation
53 # Cubist model only
54 Cubist.pred.V <- predict(edge.cub.Exp, newdata = DSM_data[-training,
55 # Cubist model with residual variogram
56 vDat <- DSM_data[-training, ]
57 coordinates(vDat) <- ~x + y
58 crs(vDat) <- "+proj=utm +zone=55 +south +ellps=WGS84 +datum=WGS84
59 +units=m +no_defs"
60 # make the residual predictions
61 RK.preds.V <- as.data.frame(krige(residual ~ 1, mDat, model = mo
62
```

Environment: Global Environment

- gRK: List of 3
- gUK: List of 3
- gXY: 201313 obs. of 2 variables
- landsat_b3: Large RasterLayer (230800 elements, 1...
- landsat_b4: Large RasterLayer (230800 elements, 1...
- looModel: List of 14
- map.cubist.rl: Formal class 'RasterLayer'
- mac.MLR: 201313 obs. of 3 variables

Console:

```
+ nugget = 0)
> model_1 <- fit.variogram(vgm1, mod)
> model_1
model      ps111  range
1  Nug 0.01322824  0.000
2  Sph 0.13637650 489.587
> gRK <- gstat(NULL, "RKresidual", residual ~ 1, mDat,
+             model = model_1)
> crs(mDat)
CRS arguments:
+proj=utm +zone=55 +south +datum=WGS84 +units=m +no_defs
>
```

```
47 # Internal Validation
48 # Cubist model only
49 l, "RKresidual", residual ~ 1, mDat,
50 l = model_1)
51
52 # Internal Validation
53 # Cubist model only
54 # Cubist model with residual variogram
55 # Internal Validation
56 # Cubist model only
57 # Cubist model with residual variogram
58 # Internal Validation
59 # Cubist model only
60 # Cubist model with residual variogram
61 # Internal Validation
62 # Cubist model only
```

Environment: Global Environment

- gRK: List of 3
- gUK: List of 3
- gXY: 201313 obs. of 2 variables
- landsat_b3: Large RasterLayer (230800 elements, 1...
- landsat_b4: Large RasterLayer (230800 elements, 1...
- looModel: List of 14
- map.cubist.rl: Formal class 'RasterLayer'
- mac.MLR: 201313 obs. of 3 variables

Console:

```
+ nugget = 0)
> model_1 <- fit.variogram(vgm1, mod)
> model_1
model      ps111  range
1  Nug 0.01322824  0.000
2  Sph 0.13637650 489.587
> gRK <- gstat(NULL, "RKresidual", residual ~ 1, mDat,
+             model = model_1)
> crs(mDat)
CRS arguments:
+proj=utm +zone=55 +south +datum=WGS84 +units=m +no_defs
>
```

Now, external validation, so, we will now will do the internal external validation. So, for we can do the external validation for so, we have built the cubist model now, we have interpolate the residuals. So, both of the operations are done. So, now we are going to do the external validation.

So, external validation for the cubist model we are going to use this validation dataset again the minus training data set just like previously. So, we have fitted that now cubist model with residual Variogram. So, cubist model with the residual Variogram. So, we are again using these minus training samples so, coordinates of the it is very important that the coordinates of the validation sample should be the same.

So, we are going to assign first that x y are the coordinates of the vDat and then we are assigning the same coordinate which we have used for the calibration data or trading data. So, you can see UTM zone 55 South WGS 84. Now, based on these validation data set we are going to predict the residual.

So, using the validation data set we are going to predict the residual based on the Kriging model. So, we are going to use this as data frame Kriging and we are going to do the residual and our new data is the validation data. So, based on the Kriging model, which we have created, we are going to now validate we are going to we are going to predict our validation dataset.

So, once we have done that, now, we have we know both the predicted values from the model using the validation data. And now we also know the predicted values of the validation data set using the Kriging interpolation. So, the final outcome will be the predicted final outcome for the validation data set will be the cubies prediction plus the Kriging interpolation prediction.

Because we know that regression Kriging is a hybrid approach which combines both the deterministic component as well as the stochastic component. So, using the validation data from the already developed cubist model we have predicted the values simultaneously we have used the same validation dataset to predict their value using the creaking interpolation.

Now, we have know there values separately now we are going to combine them together to get the final prediction. So, the final prediction you can see sum the two components together in the regression Kriging. So, cubist predicted validation and then regression Kriging prediction.

So, we are going to add this type then let us go for this goof function for doing different types of validation. So, using the only the cubist data cubist or if we go for the goodness of statistics, you can see this is the model results based on the validation data set and then if we go for validating the regression Kriging using the cubist model then you can see that we are going to based on our observed values are there.

And the predicted values is these final values which are predicted in our last line. So, we are going to run this and we are going to have this final validation result based on the regression based on the cubist also cubist model also. So, here we are going to here you can see we have

done two types of validation the first one is based on the regression based on the cubist model only and the second one is based on the regression Kriging using the cubist model.

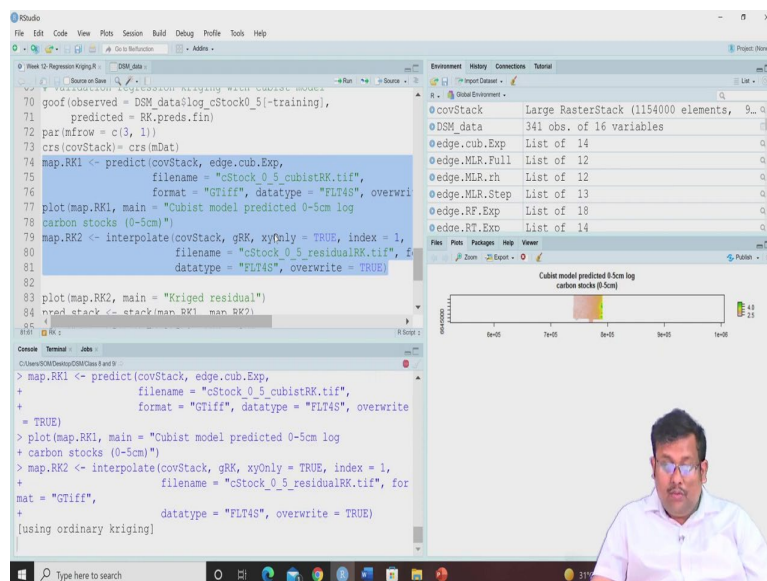
So, you try to understand the logic how we have proceeds, how we have how these different steps were implemented. So, we started with the point dataset and then we downloaded the covariate dataset after downloading the covariate dataset, we have done the stacking of the covariates.

After stacking the covariate we extracted this covariates we converted this to simple data frame after we converted this to simple data frame then we have done the cubist model and based regression fitting then we have calculated the residuals after we calculated the residuals we have made the Kriging interpolation.

So, and then we have used the validation dataset to predict from this Kriging already developed cubist regression model and simultaneously we have also predicted based on the Krig residuals and then combine them together to get the final values of the prediction. Once we get the final value of the prediction then we have done the goof function executed the goof function to see what is the validation results.

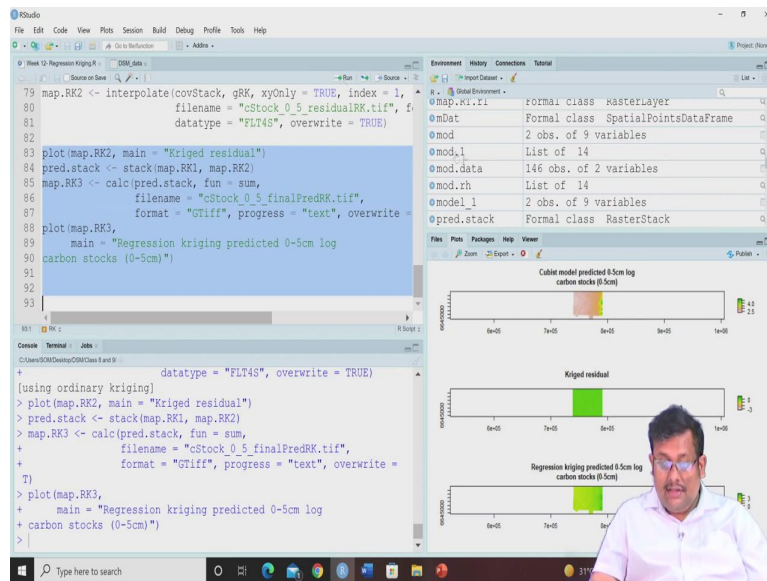
When we have done the validation in two ways one is using only the cubist model and second using the regression Kriging model using the cubist. So, all the operations of these regression Kriging are now done.

(Refer Slide Time: 28:07)



The screenshot displays the RStudio interface. The main editor window contains R code for regression kriging. The code includes a 'goof' function call, parameter setting, data crs assignment, prediction using 'predict', and interpolation using 'interpolate'. The console shows the execution of these commands. The Environment pane on the right lists objects like 'covStack', 'DSM_data', and 'edge.cub.Exp'. A plot window shows a map titled 'Cubist model predicted 0-5cm log carbon stocks (0-5cm)' with a color scale from 0 to 14. A small video inset of a man is visible in the bottom right corner.

```
70 goof(observed = DSM_data$log_cstock_0_5[-training],
71      predicted = RK.preds.fin)
72 par(mfrow = c(3, 1))
73 crs(covStack) = crs(mDat)
74 map.RK1 <- predict(covStack, edge.cub.Exp,
75                  filename = "cStock_0_5_cubistRK.tif",
76                  format = "GTiff", datatype = "FLT4S", overwrite = TRUE)
77 plot(map.RK1, main = "Cubist model predicted 0-5cm log
78      carbon stocks (0-5cm)")
79 map.RK2 <- interpolate(covStack, gRK, xyOnly = TRUE, index = 1,
80                      filename = "cStock_0_5_residualRK.tif", format = "GTiff",
81                      datatype = "FLT4S", overwrite = TRUE)
82
83 plot(map.RK2, main = "Kriged residual")
84 final_stack <- stack(map.RK1, map.RK2)
```



Now, we are going to map. So, for development of the for developing the different types of map we are going to this is very important step, this is very important that the coordinate reference system of covstack should be always should be equal to the coordinate reference system of our model data, we have to ensure that. Once we have ensured that now we are going to develop that map of we are going to develop two maps.

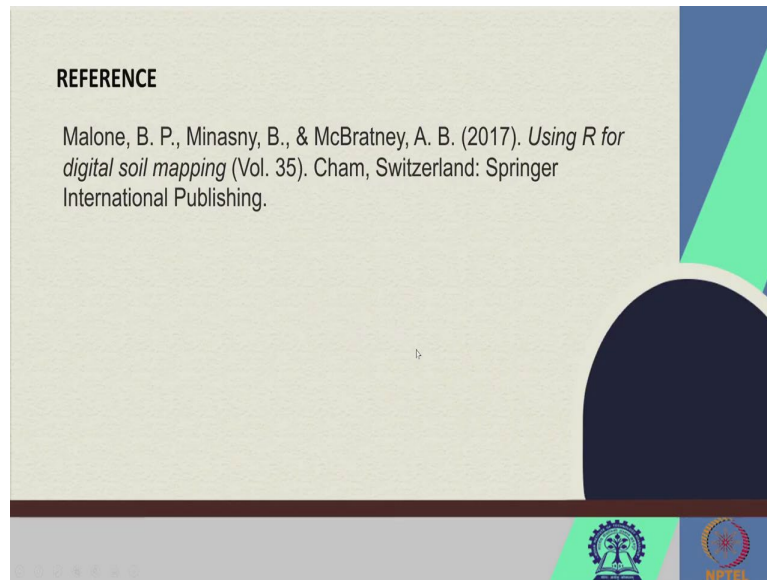
So, first one is map of the so you are going to see that we are going to get the cubist model predicted 0 to 5 centimeter lock carbon stock. So, and then we are going to get two other maps. So, here you can see in the first instance we are going to get that cubist model predicted 0 to 5 centimeter log carbon stock in the second instance we are going to get the krig residuals and in the final map it will be a combination of both these. So, it is a regression Kriging predicted 0 to 5 centimeter log carbon stock.

So, here you can see we are changing some index. So, here when we are going to the there are three maps. So, in the first map, we are going to use the cubist model predicted carbon stock in the second, we are going to interpolate via the Krig residuals. And in the third map, we are going to use the regression Kriging final by combining these two approaches by combining these two outcome. So, this is how guys you can develop this final map based on the regression Kriging.

And so, I think the flow of operation which we have seen is under is you have understood it, and please try to execute this, please try to run this thing. I will try to give you an assignment home assignment based on this, in our, I will post it in our class forum another homework assignment based on this regression Kriging. So, that you can try these codes and you can

feel much more confident about running the regression Kriging using your own data set. So, I will be posting another confidence builders by for executing this regression Kriging.

(Refer Slide Time: 31:21)



So, just like previously the reference for this lecture is this and let us wrap up this lecture here and let us meet in our next lecture to see the categorical model the mapping the categorical variables using R in digital soil mapping. So, guys thank you for listening this lecture. I hope that you have learned something new and let us meet in our next lecture to discuss it in more details. Thank you.