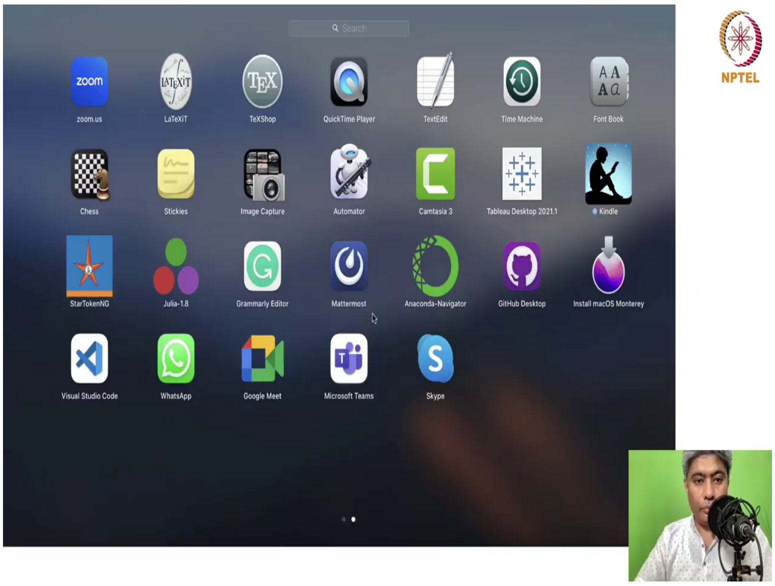
Predictive Analytics - Regression and Classification
Prof. Sourish Das
Department of Mathematics
Indian Institute of Technology, Madras

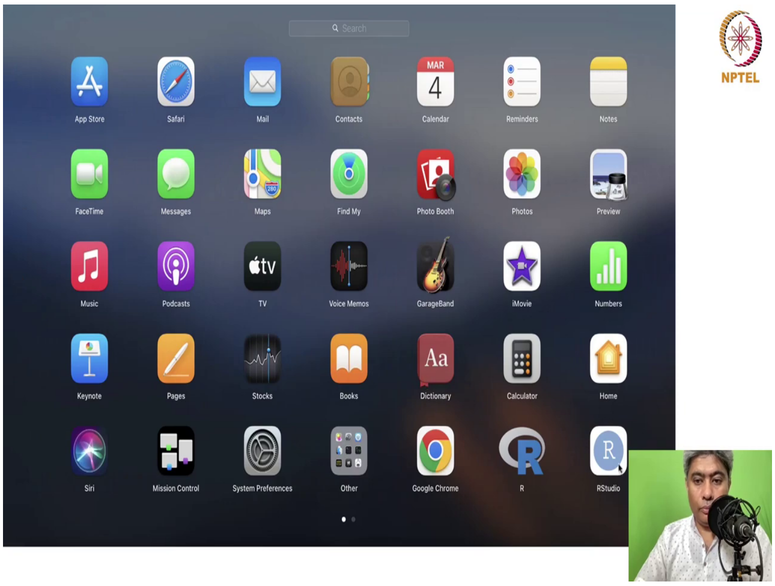
Lecture - 50
Hands on with R: Implement GP Regression from scratch

Hello all welcome back to part B of lecture 15. In this discussion, we are going to see a demo of how to Implement Gaussian Process Regression from scratch using R.

(Refer Slide Time: 00:36)

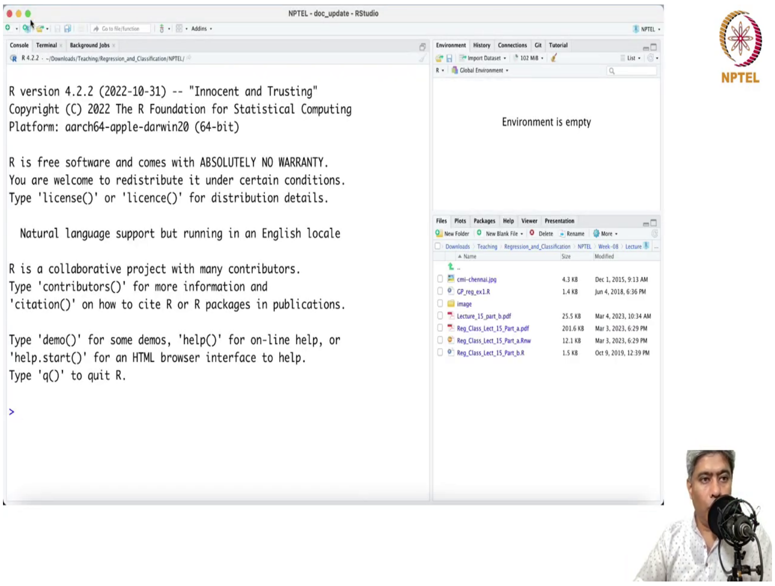


(Refer Slide Time: 00:37)



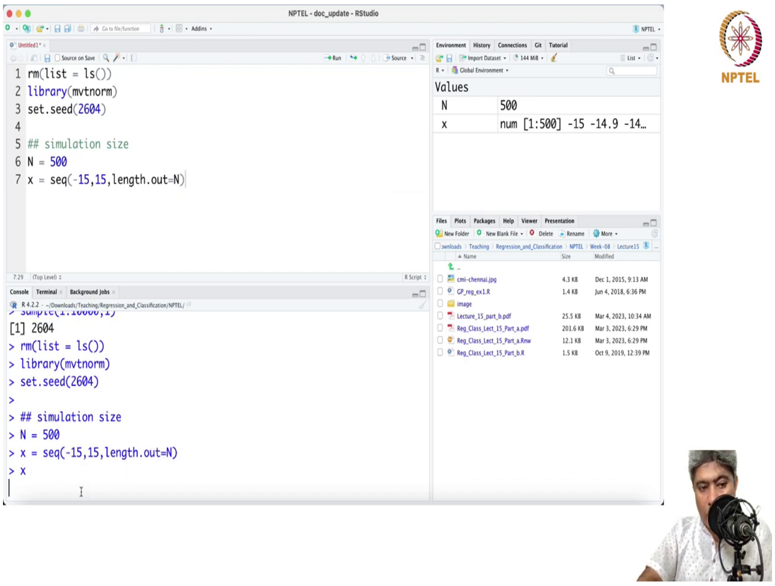
So, I am going to start my R.

(Refer Slide Time: 00:41)



I am going to open R and R script.

(Refer Slide Time: 00:49)

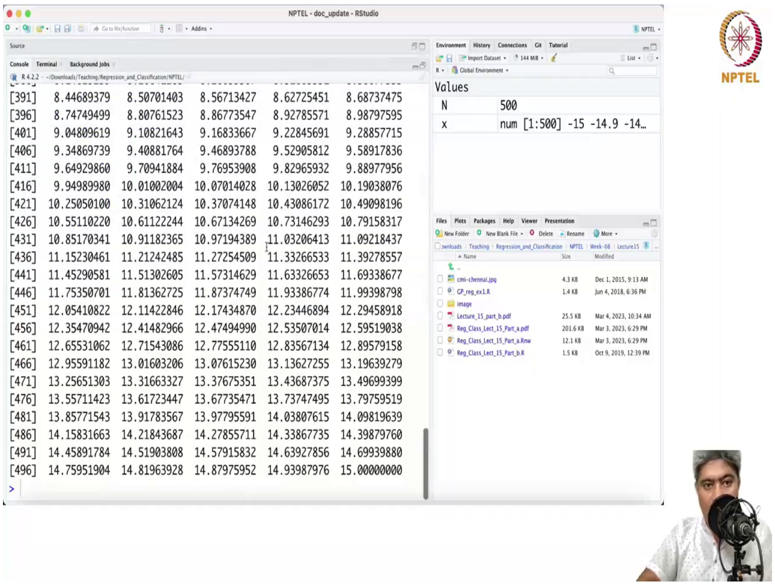


So, first thing I am going to say that ok, remove just this list anything if it is there ls. So, anything if there is any later if I you know have any just clean the environment essentially. I am going to say install a load mvtnorm package. Let me see, I hope it is there. Yeah, it is there. Then I am going to set up a seed.

So, let me just draw a random sample sample sorry, a sample between a number between 1 and say 10000 one sample is good enough 2604. So, I will just pick this random number and set dot seed as this. I am setting this number so that when you will use this number, you will get the exact same answer.

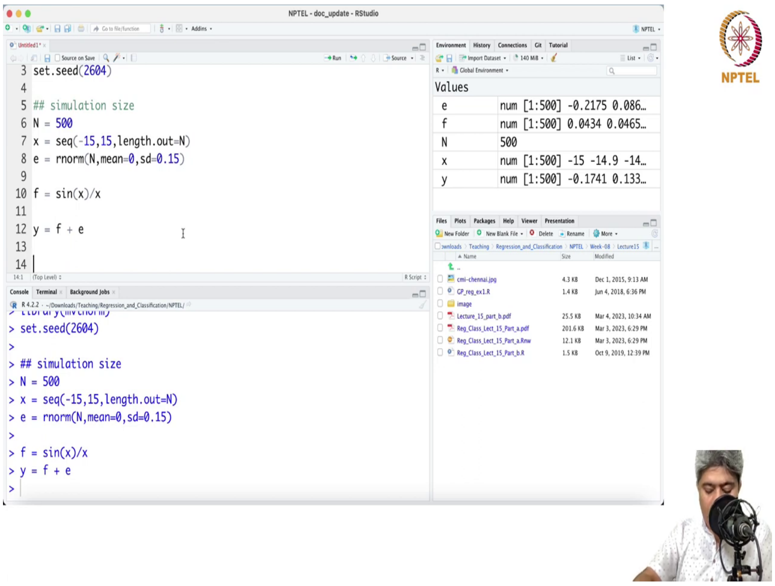
Then I will decide a simulation size simulation size of say around 500. I think ok, that many data points we will simulate, ok. So, first I am going to define a x variables sequence of number between say minus 15 to 15 and length, length dot out equal to capital N. So, if you just run this. So, now, you can see if you just run x.

(Refer Slide Time: 02:54)



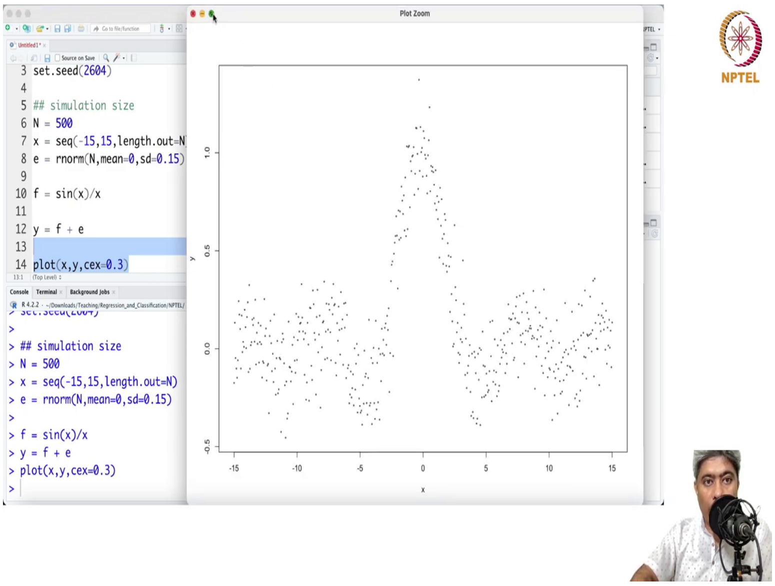
If you can see there are 500 samples 500 values are being created between minus 15 and 15 each are of equals with equal you know interval, ok. Now, I am going to draw a random samples bunch of random samples say e equals to rnorm.

(Refer Slide Time: 03:32)



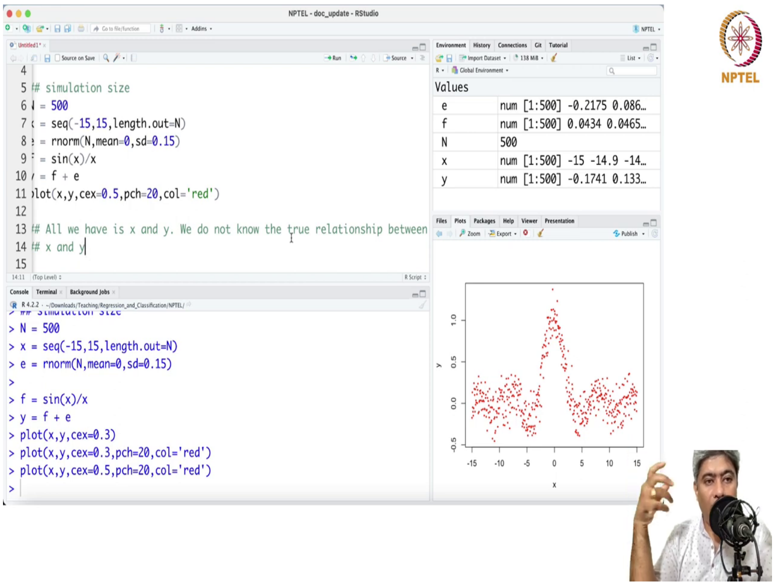
How many N many with mean equal to 0 and sd equal to 0.15, ok. And then I am going to run right, f of x. If is sin of x by x and then I just run this and then now y is equal to y values are my whatever f plus e, f of x plus e, correct.

(Refer Slide Time: 04:23)



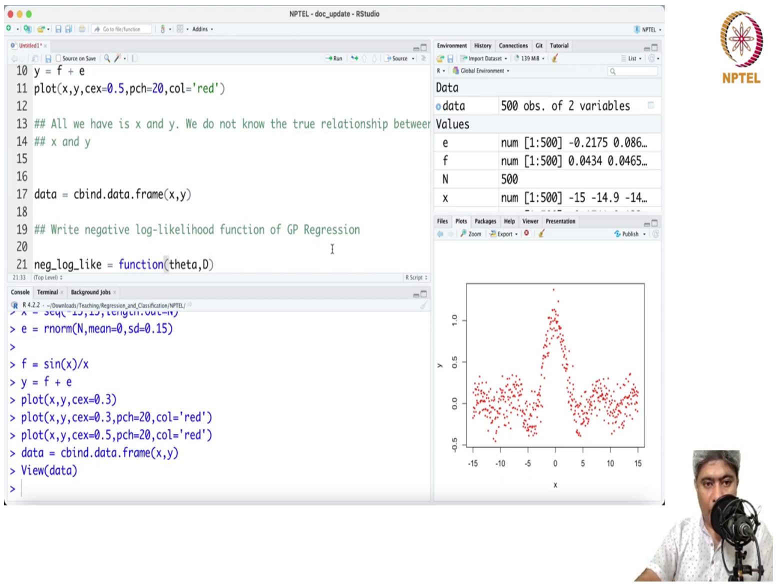
Now, if I plot x, y cex equal to 0.3. So, I can let me zoom this, ok. Let me just zoom it. So, you can see all the points are around this, ok. And in fact, you can make it pch equal to 20 maybe color equal to red.

(Refer Slide Time: 05:01)



So, this is this will be little brighter or we can just increase the value little bit cex. Yeah, it is slightly better, ok; looks much better, ok.

(Refer Slide Time: 07:05)



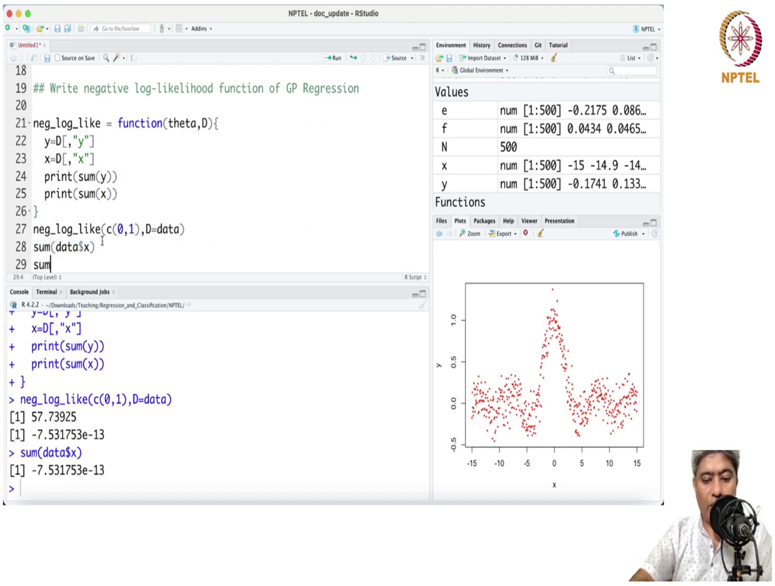
So, now this is my plot. This is my data. Now, we are going to pretend that all we have is x and y. We are going to pretend all we have is x and y, ok. All we have is x and y and we do not know the true relationship between x and y. So, given the data, can we pick up the this completely. There are some there is no trend actually there is bunch of seasonality and most of the values are hovering around 0.

And then there is at 0, there is a signal boom. There is a burst of a signal and then it came down. And then again, it is sort of a hovering around 0. So, can we capture this complete non-linear behavior using GP regression? Ok. So, this is our idea. So, now what I am going to do, we are going to create a data set. First thing what we will do, we will just say cbind data dot frame x comma y. So, here is my data x and y's and these are the. So, now we have a data which has one x and one y and we are trying to fit a model out of it.

So, the first thing is we will fit a Gaussian process regression and we will model it from scratch. There are some built-in packages are there, but I want to understand if you use built-in package, we will not really understand how GP regression really works, ok. So, what I am going to do? I am going to teach you how the GP regression works.

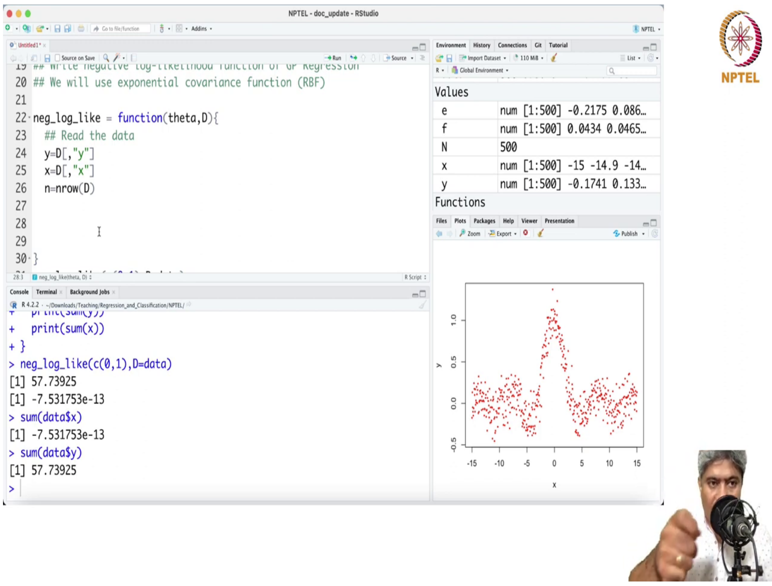
So, best way to learn is do programming from scratch, alright. So, first thing I will do, write GP regression negative log likelihood of GP regression, Write negative log-likelihood function of GP Regression, ok. So, neg log like equal to function y comma x, ok; and then no, other way we will do is theta and D. So, D for data, theta is all the parameters.

(Refer Slide Time: 08:57)



So, what we will do is y equal to D comma y and x equal to D comma x, ok. And let me just write; let me see if it is actually working, ok. So, the best way of doing it is just write sum y print, ok print sum of y and print sum of x, ok. So, let me just run only this part and negative log likelihood and say 0 comma 1 that will be for theta and D equal to data. Yeah, 57.73 and some very close to 0. So, if I take data, just check data dollar x. If I just take sum of that. And they are exactly matching and then if I just take sum of data dollar y 57.73925.

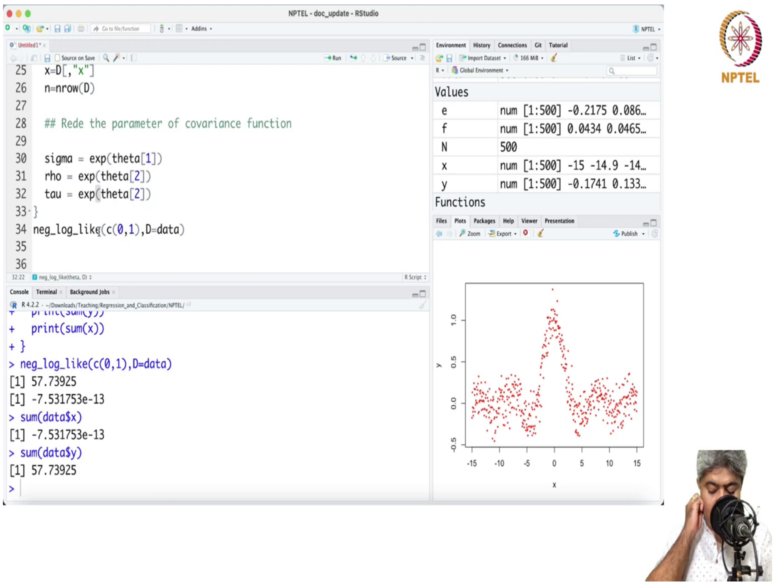
(Refer Slide Time: 10:50)



So, they are exactly matching. So, that means, this part is happening absolutely correctly. So, I can delete this part and I can delete the print also, ok. So, now we have (Refer Time: 11:09) that means, this function reading the data correctly, ok. So, next thing we will do n equal to nrow D, ok. And I think that will be not a problem.

And so read the data. This is first. These are the three things. I think first is Read the data. And in this GP regression, we are going to use exponential covariance function. So, let me write it down explicitly We will use exponential covariance function also sometimes known as RBF Radial Basis Function, ok. So, now and that will have three parameters in that covariance function.

(Refer Slide Time: 12:37)

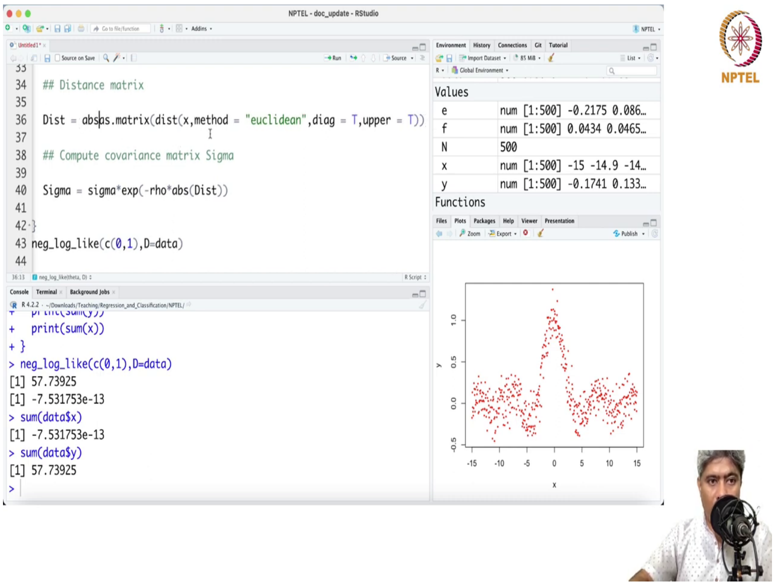


So, the read the parameters of the covariance functions. Read the parameters of covariance function, ok. And what are those? First is sigma, ok. Sigma equal to e to the power theta 1. And why I am taking e to the power theta 1? What happens eventually I want to put it through a optimization subroutine.

I am going to call the optim function in this case. And in the optim function, the theta will be expected to be varying from minus infinity to infinity. But I want my sigma obviously, on the 0 to infinity range. So, as soon as it the I get the theta, I will transform it and then I will use it.

So, we will get we will effectively optimize a transform variable or transform parameter. Then next is rho. Rho is e to the power theta 2 and tau. Tau is equal to also exponent theta 2, ok. Now, what I am going to do is I am going to calculate the distance matrix.

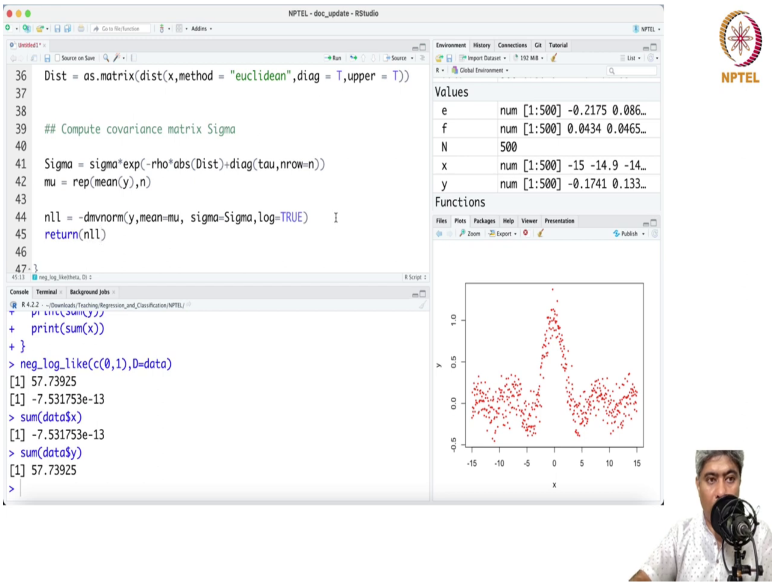
(Refer Slide Time: 14:38)



Let us first calculate the distance matrix. Distance matrix so, Dist where effectively what I am going to do call the dist function available in the R. So, Dist x method equals to euclidean diagonal equal to diagonal equals to true and also I want the upper part of the distance matrix as well. And I want to store it as a matrix so that I can do operation matrix operation on it later when we will do the optimization, ok.

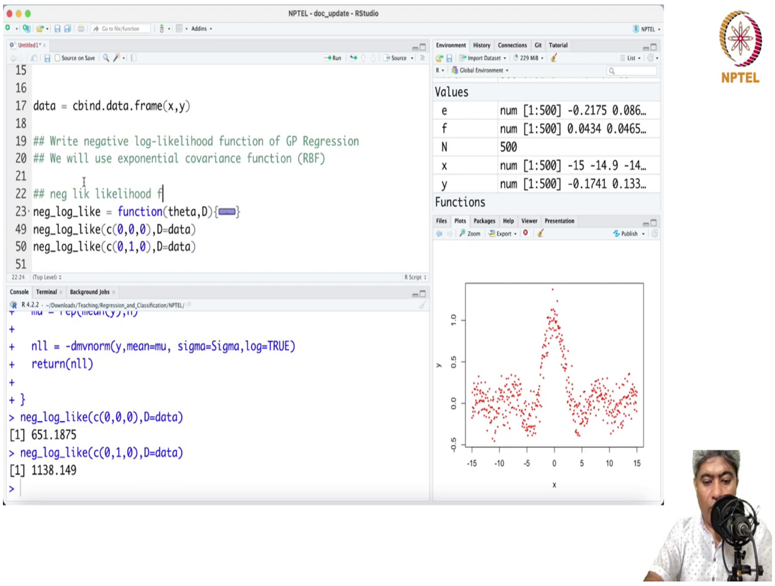
So, this is the distance matrix and then I am going to compute the covariance matrix sigma. Let us compute. Compute covariance matrix sigma, ok. What is covariance matrix sigma? So, sigma is essentially the sigma parameter we have just right here sigma parameter times e to the power e to the power minus rho times absolute value of the distance absolute value of the distance. So, in fact, what I can do? I can take the absolute value here. So, that you will or I can just write it. I do not want to make it too big. Distance I can equal to abs distance.

(Refer Slide Time: 17:01)



Or it does not matter actually we can just have it here. I think this is fine and diagonal of tau diagonal of tau where nrow equal to n, ok. Now, mu I am going to say that replicate mean of y as a function of n, ok. And then I am going to write negative log likelihood, evaluate negative log likelihood as function of negative value of first dmvnorm y as a mean equal to mu sigma equal to this capital sigma and log equal to true. And I have to have a negative sign in the beginning and return, return negative log likelihood, ok.

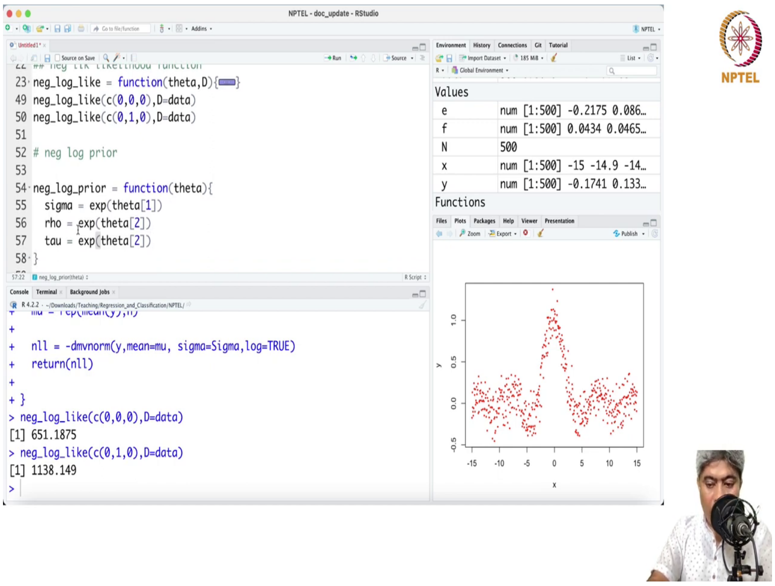
(Refer Slide Time: 18:33)



So, this is the function. And let me just call this function, ok. And try to evaluate this function at some initial value. So, this gives me one value, this could be another value, ok 1138. So, at some initial values it runs and it gives you some good reason things, ok. Now, next what I am going to do. I am going to write optimize say.

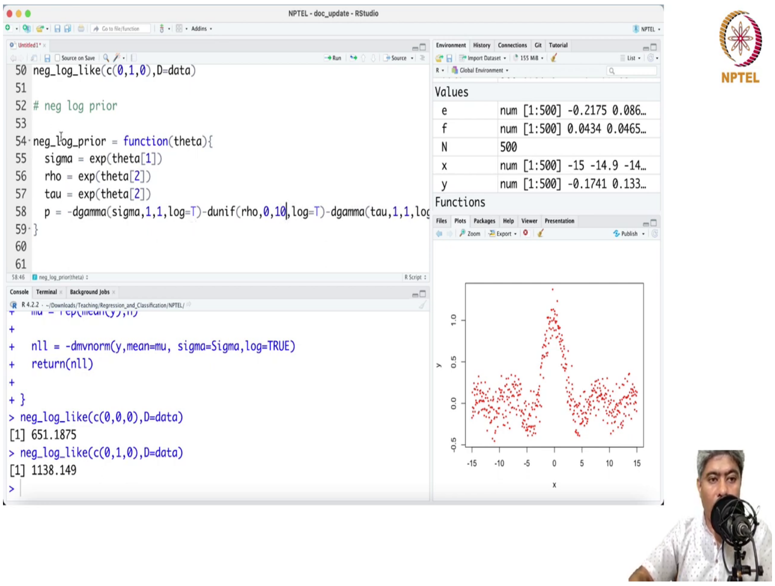
Now, I am going to write some optimization technique optimization. Actually, we have to put some prior on the top of that otherwise, optimization might throw some error. So, let us define. So, this is my negative log likelihood. So, write. Let us write some this was negative log likelihood function.

(Refer Slide Time: 20:01)



And then negative log prior. We have to write. So, we will do negative log prior function we will define as function of theta only, ok. It will not be function of data. It will be only function of theta. And what I am going to do is just copy these few lines at the beginning from the log likelihood. And paste it here. And then, p will be say dgamma.

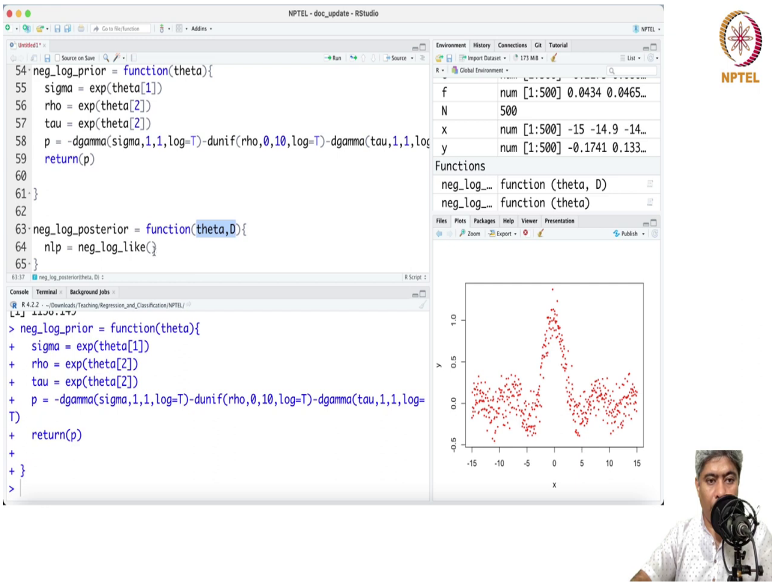
(Refer Slide Time: 20:53)



Let us suppose this is these are all dgamma sigma equal to comma 1,1, log equal to true. Then minus all I have to just do rho and then so, actually what I can do instead of saying that. Yeah, I can just take since I am giving the same prior. Yeah, let me just write it down in any way because later if you want to change anything, you can change by yourself.

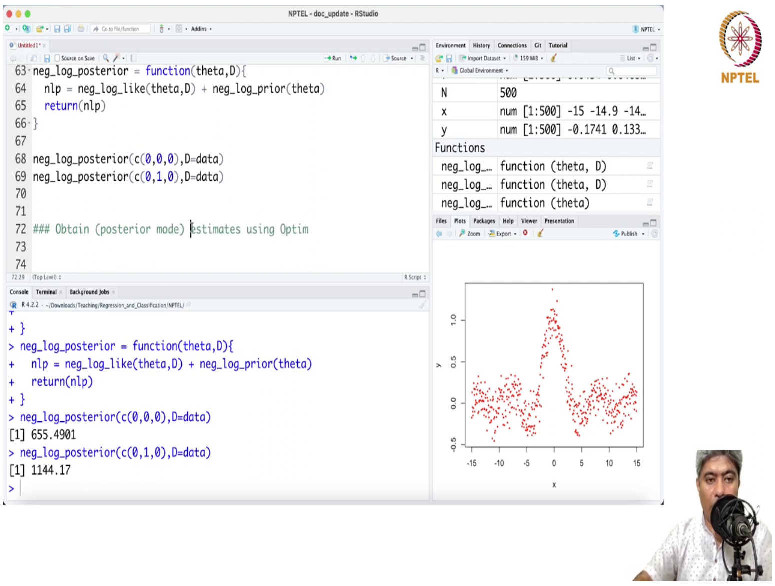
Because suppose you decide that ok, on rho, I do not want to put a gamma, I want to put say uniform between 0 and 5 or something like that, that you can do. So, if I want you can give a say uniform between say 0 and 15, ok. So, you can do that or 0 and 10 maybe. Yeah.

(Refer Slide Time: 22:32)



So, you can do this kind of thing. And then eventually return. So, you can assign a choice prior of your choice. So, return p. So, this is my negative log prior, ok. And then finally, negative log posterior which will be simply negative log posterior function theta comma D. And nlp equal to first, what I have to do is basically negative log likelihood comma theta comma D plus negative log prior comma theta.

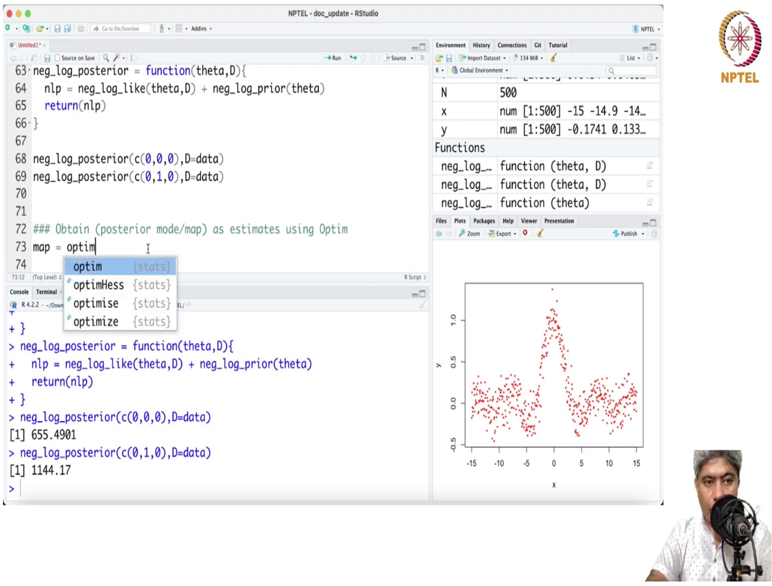
(Refer Slide Time: 23:45)



And return nlp not natural language processing it is negative log posterior, ok. And if we run this negative log posterior. Let us check if it is working perfectly fine for at least these values. The initial values, ok. Yeah, it is working reasonably, ok. We can try one. And I think this is working perfectly fine, alright. Next, what I am thinking is that essentially what we need to do? We need to find.

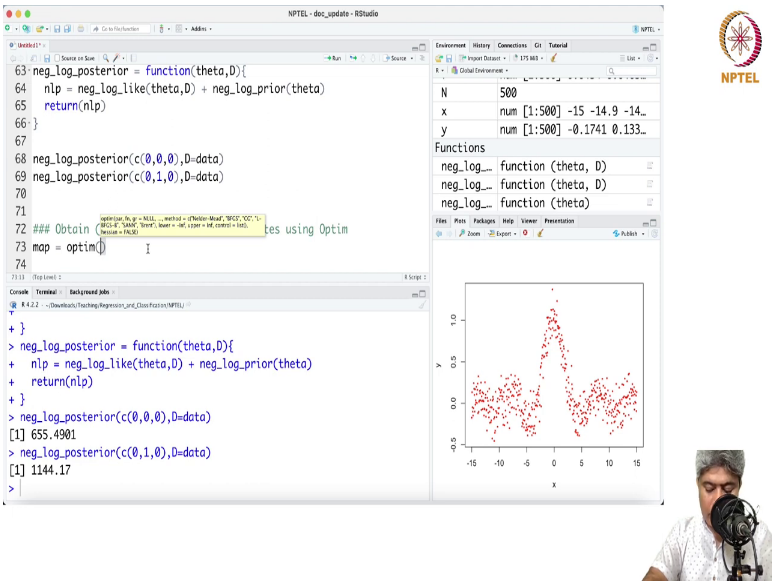
So, what we will get that posterior mode Obtain the more posterior mode or Obtain estimates parameter estimates using Optim. These are typically posterior mode Optim Estimate using Optim or posterior mode. These are all posterior mode as estimate, ok; posterior mode as estimate.

(Refer Slide Time: 25:24)



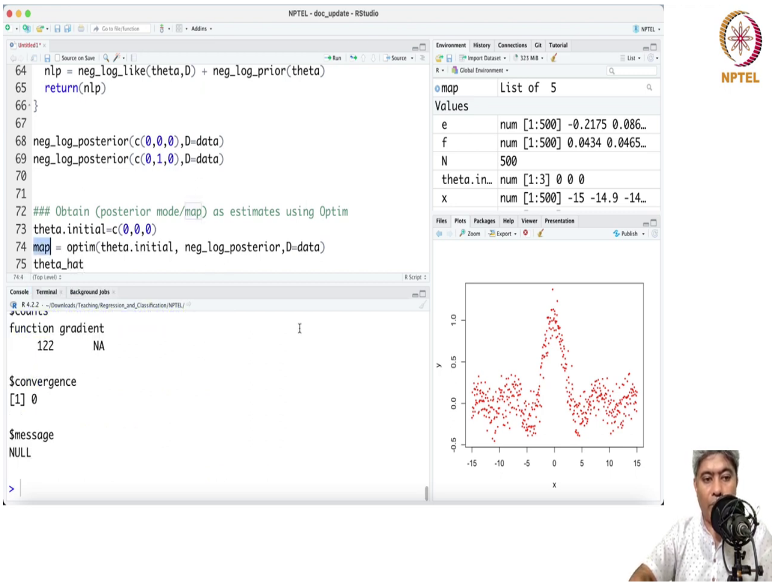
It is not mle. If you hand to mle, you have to optimize negative log likelihood function. But we are going to optimize the negative log posterior function. So, we will get posterior mode. We will obtain try to obtain both. And we will try to see what are the differences. So, first sometimes posterior mode also called map estimate. Maximum (Refer Time: 25:48) posterior estimate. So, map equal to optim, ok.

(Refer Slide Time: 25:58)



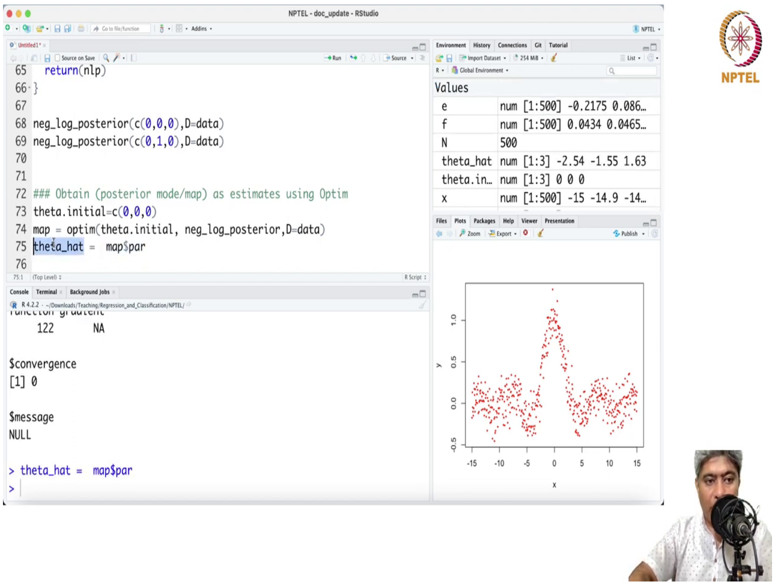
And so, we have to have some initial value of theta.

(Refer Slide Time: 26:00)



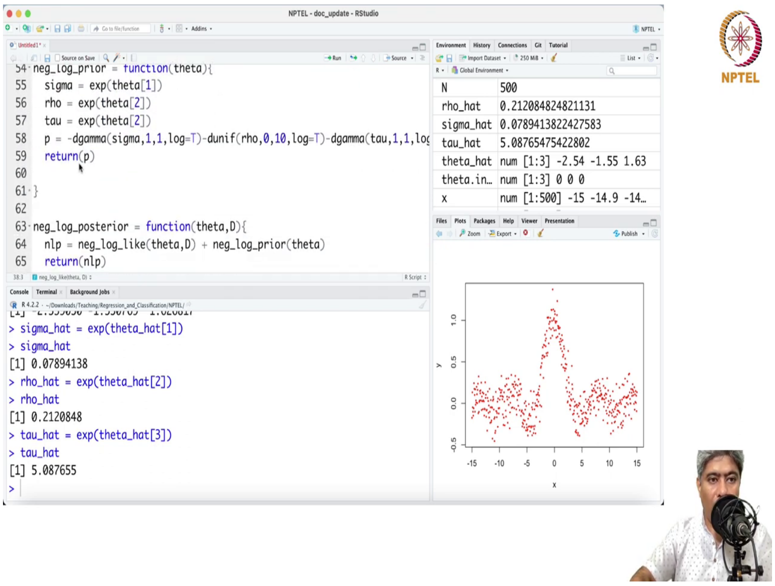
Let us take this initial value theta dot initial, ok. Initial value of theta. So, let me provide first the initial value of theta and then negative log posterior and then what we have to give D equal to data, ok. So, it takes few minutes maybe a few seconds, alright. And then from there theta hat we have to. So, if I just run the map.

(Refer Slide Time: 26:51)



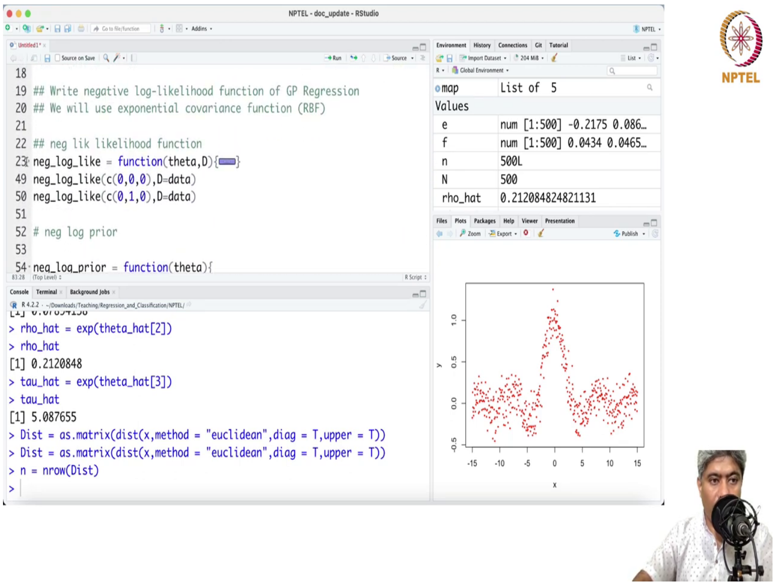
So, you can see. That it gives me three estimates. This is the value the at optimal value the negative log posterior in these estimates. And convergence equal to 0 means there is the convergence did happen and it has not find any problem with the convergence. The optimization did not had any problem. So, theta hat equal to from the map. Now, what I will do? I will just extract the parameter optimized parameters, ok. And then my theta hat these are the values and then what I am going to do sigma hat equal to e to the power theta hat 1.

(Refer Slide Time: 27:54)



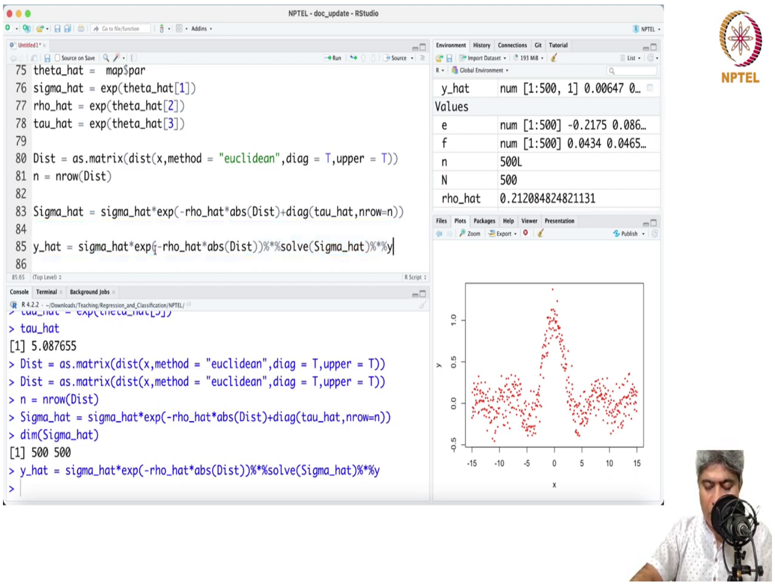
That is my sigma hat and then rho hat rho hat equal to e to the power theta hat 2. What is rho hat? Let me see so, 0.21 and then tau hat equal to e to the power theta hat 3. So, tau hat, ok. So, we got the estimates. Now, we have to have the distance. Remember that we calculated this distance matrix out of the box, ok. Out in the function we have not done it in the out in the environment. So, we have to just calculate that. So, now we have the distance.

(Refer Slide Time: 29:10)



n equal to nrow of the Distance and now first we have to calculate sigma, ok. Capital S Sigma hat equal to first sigma hat times exp. So, remember that we want this function to be here.

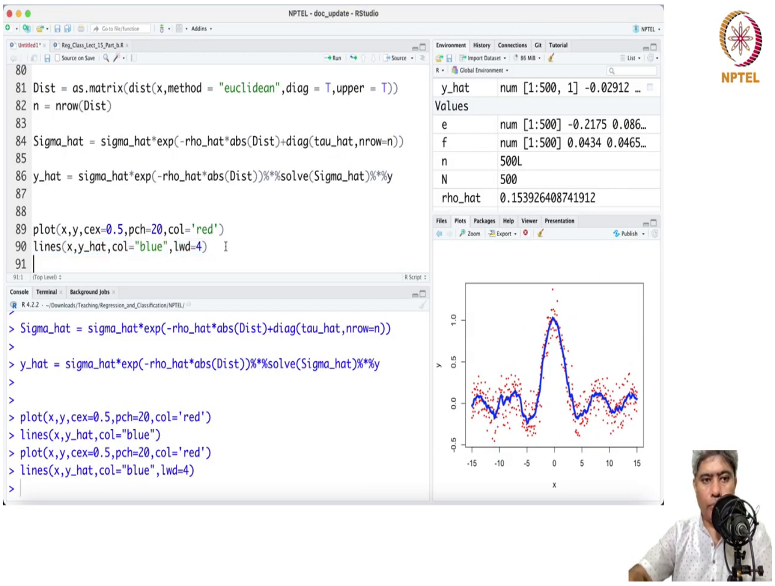
(Refer Slide Time: 30:00)



So, Sigma will be replaced by the estimated Sigma hat rho will be rho hat and tau will be by tau hat. Now that gave us Sigma hat, ok. So, remember Sigma is a very large matrix going to be. This is a very large matrix of if you just say dimension of Sigma of Sigma hat is 500 by 500. Remember that.

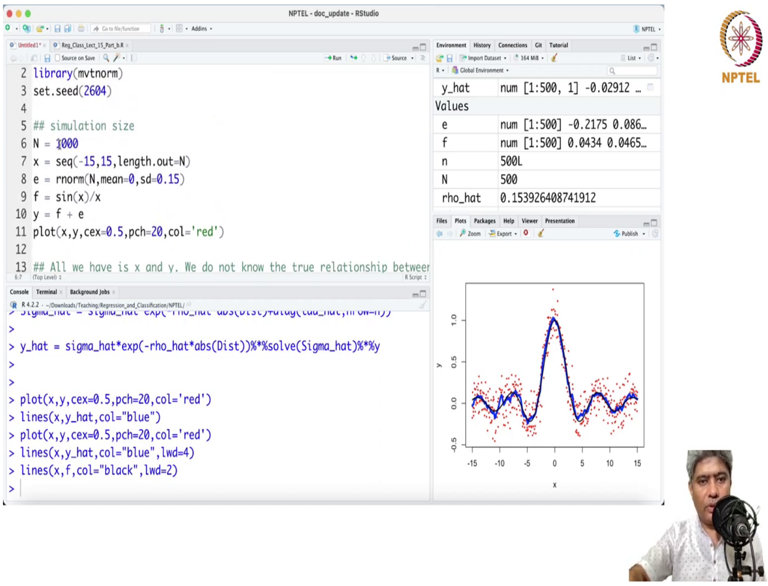
And then y hat then will be y hat will be sigma hat. Times exp minus rho hat times distribution, ok. Percentage star percentage solve Sigma hat percentage star percentage y. So, if I just run this, this will be my y hat. And now what I am going to do on the plot. Let me just bring that plot that we have done here. Let me just bring that plot here.

(Refer Slide Time: 31:45)



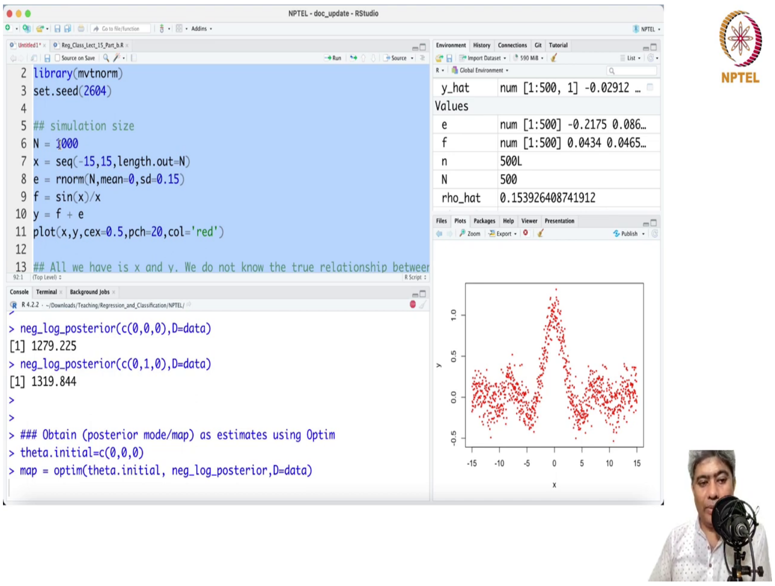
Now, I am going to write lines x comma y, x comma y hat. Now, I am going to my x color y hat color equal to blue maybe and l width equal to be 4, ok. So, if you just run it, yeah. So, you can see, let me just. So, you can see that it is bit jittery, but it is actually picking up to an extent. Let me plot the original curve here. Let me plot the original curve here, ok. Say lines equals to x comma y f; f is the original curve, I think.

(Refer Slide Time: 32:54)



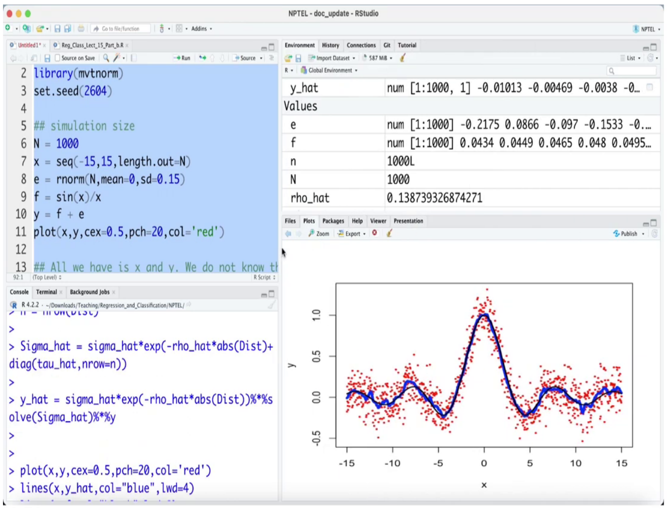
And we can take some other color. Color equal to maybe some other color may be black, ok. And line width equal to 2. So, if you now, let me just you can see this black color is the true curve. And this, this blue color is the estimated f hat and that f hat is essentially very close. Now, what I am going to do, we will see that as the sample simulation size increases. So, instead of 5000, if I just increase the simulation size to 1000, that will make the curve much more. We will see how the curve will behave.

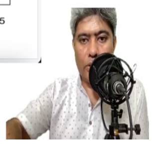
(Refer Slide Time: 34:06)



If we have 5000 samples so it will take a little bit time because we have more samples now. So, it might take about a minute so, if you.

(Refer Slide Time: 34:48)





Let me just increase it. So, this is with the 1000 samples we can see. And if this is with the 10000 samples, 500 samples, this is with the 500 samples. And this curve is with the 1000 samples.

(Refer Slide Time: 35:14)

So, almost the 1000 sample, probably a little bit closer. So, the GP regression has no clue what is the true original relationship between x and y. And yet the GP regression actually picking up the actual relationship between x and y. So, this is a very strong and very useful methodology if you want to do non-linear regression.

Now, you can ask me. If you have only one x and then we know how to do that, can I do it for more than one x, if I have, you know, multiple x. The answer is yes, you can do exactly the same way. Only thing is you have to calculate the distance, but we know how to calculate the distance.

And here we have used Euclidean distance because it was a simple toy problem. Sometimes you may have to be bit careful about what kind of distance function you want to compute. For example, if you are using a spatial data, you may have to use earth distance. But if you are using say a categorical data, then you may have to use appropriate distance like Gower distance, which do calculate the distance between the categorical variables.

So, you have to be bit careful about what kind of distance functions that you want to use. But if you are sure this is the correct distance function that you want to use. Then perhaps Gaussian process method or Gaussian process regression is one of the best methodology that you can think of. So, with that, I will stop here in this video lecture. Let us see you in the next video.