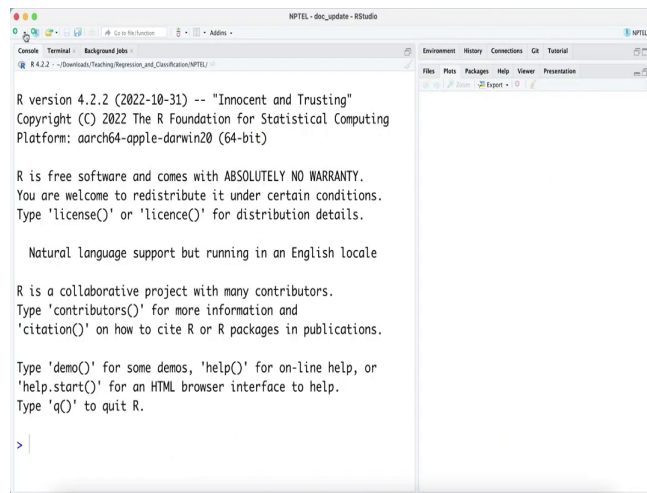


**Predictive Analytics - Regression and Classification**  
**Prof. Sourish Das**  
**Department of Mathematics**  
**Chennai Mathematical Institute**

**Lecture - 46**  
**Hands on with R: Feature Engineer in Logistic Regression**

Hello all, welcome to the Part C of lecture 13; in this video, I am going to talk about how to model non monotonic relationship between the x and y using R.

(Refer Slide Time: 00:36)



```
NPTEL - doc_update - RStudio
R 4.2.2 -- ~/Downloads/Teaching/Regression_and_Classification/NPTEL/
R version 4.2.2 (2022-10-31) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: aarch64-apple-darwin20 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

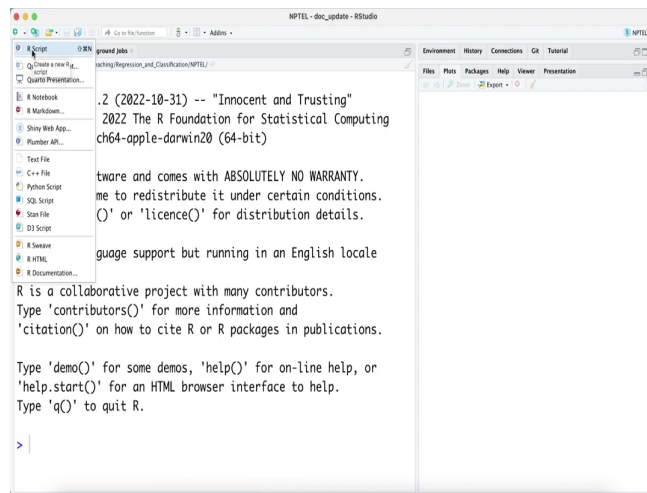
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```



(Refer Slide Time: 00:39)



```

> ?contributors
contributors()
R (2022-10-31) -- "Innocent and Trusting"
2022 The R Foundation for Statistical Computing
ch64-apple-darwin20 (64-bit)

software and comes with ABSOLUTELY NO WARRANTY.
me to redistribute it under certain conditions.
()' or 'licence()' for distribution details.

language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

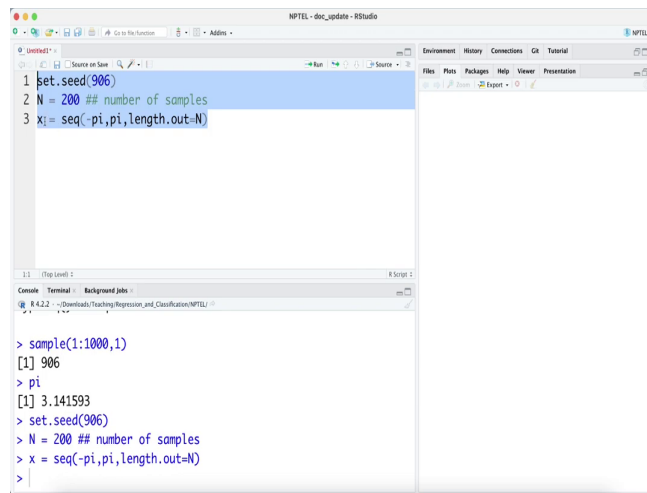
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>

```



(Refer Slide Time: 00:41)



```
1 set.seed(906)
2 N = 200 ## number of samples
3 xj = seq(-pi,pi,length.out=N)
```

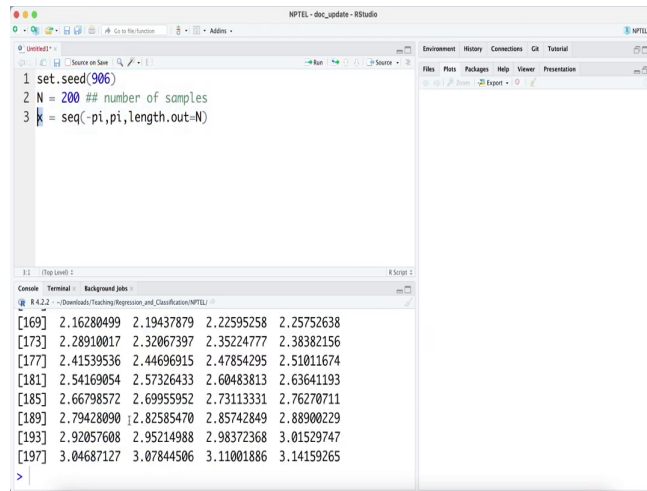
```
> sample(1:1000,1)
[1] 906
> pi
[1] 3.141593
> set.seed(906)
> N = 200 ## number of samples
> x = seq(-pi,pi,length.out=N)
>
```



So, first I am going to create a script alright; so, first what I will do, I will set its going to be a simulation study. So, I am going to put some number, maybe I will just draw a sample random sample between 1 is to 1000 and 1 ok, 906, I will put 906 and then maybe I will just simulate 200 sample ok.

So, number of samples that I will simulate, then I am going to simulate x or I will just create a sequence of number between minus pi and pi. So, pi is already there value of pi if you just play and length gth dot out equal to capital N.

(Refer Slide Time: 02:05)



The image shows a screenshot of the RStudio interface. The source editor contains the following R code:

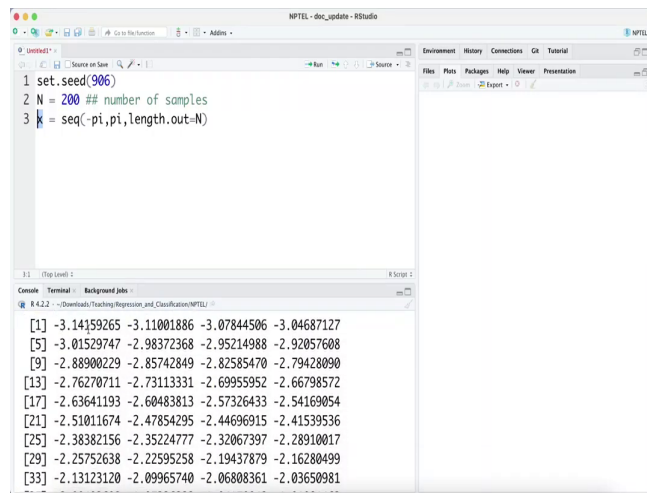
```
1 set.seed(906)
2 N = 200 # number of samples
3 k = seq(-pi,pi,length.out=N)
```

The console window shows the output of the code, displaying a sequence of 200 values for  $k$ :

```
[169] 2.16280499 2.19437879 2.22595258 2.25752638
[173] 2.28910017 2.32067397 2.35224777 2.38382156
[177] 2.41539536 2.44696915 2.47854295 2.51011674
[181] 2.54169054 2.57326433 2.60483813 2.63641193
[185] 2.66798572 2.69955952 2.73113331 2.76270711
[189] 2.79428090 2.82585470 2.85742849 2.88900229
[193] 2.92057608 2.95214988 2.98372368 3.01529747
[197] 3.04687127 3.07844506 3.11001886 3.14159265
>
```



(Refer Slide Time: 02:12)



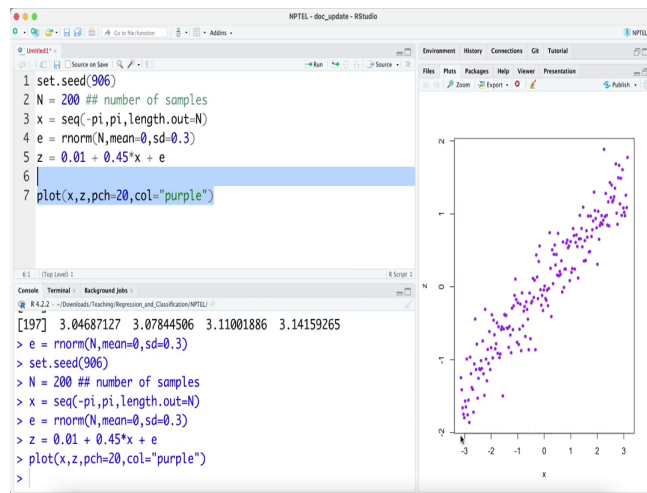
```
1 set.seed(906)
2 N = 200 ## number of samples
3 k = seq(-pi,pi,length.out=N)
```

```
[1] -3.14159265 -3.11001886 -3.07844506 -3.04687127
[5] -3.01529747 -2.98372368 -2.95214988 -2.92057608
[9] -2.88900229 -2.85742849 -2.82585470 -2.79428090
[13] -2.76270711 -2.73113331 -2.69955952 -2.66798572
[17] -2.63641193 -2.60483813 -2.57326433 -2.54169054
[21] -2.51011674 -2.47854295 -2.44696915 -2.41539536
[25] -2.38382156 -2.35224777 -2.32067397 -2.28910017
[29] -2.25752638 -2.22595258 -2.19437879 -2.16280499
[33] -2.13123120 -2.09965740 -2.06808361 -2.03650981
```



So, if I just run this, then  $x$  will be like 200 values between minus 3.141 and 3.141; so, between minus  $\pi$  and  $\pi$  there are 200 grids I have been created.

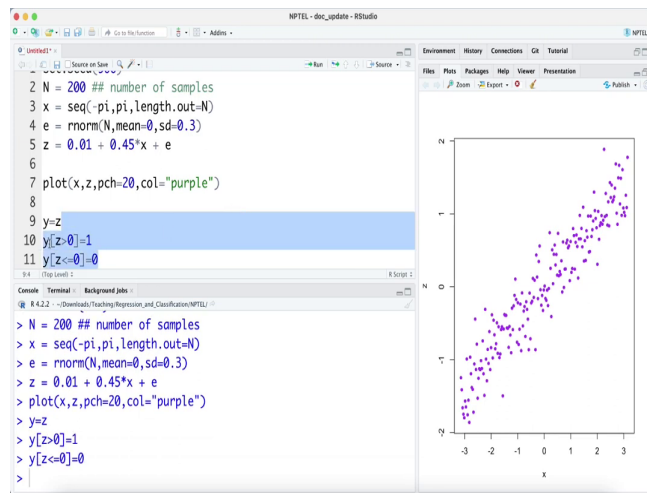
(Refer Slide Time: 02:27)



Now, what I am going to do is I am going to simulate some random numbers, I will call it e or I will say, basically these are going to be my residuals rnorm how many samples, n samples with mean equal to 0 and I am going to say sd equal to 0.3 ok. Now, z this is going to be my latent variable, but for now z is going to be like 0.01 plus 0.45 times x plus e ok.

Now, if I plot x comma z, say pch equal to 20 and color equal to say purple, if I do that. So, this is the simulated values of x and z and all x values are somewhere between negative minus pi to pi z values and they have a straight line relationship as we have done here; so, but z is unobserved, z is latent variable.

(Refer Slide Time: 04:07)

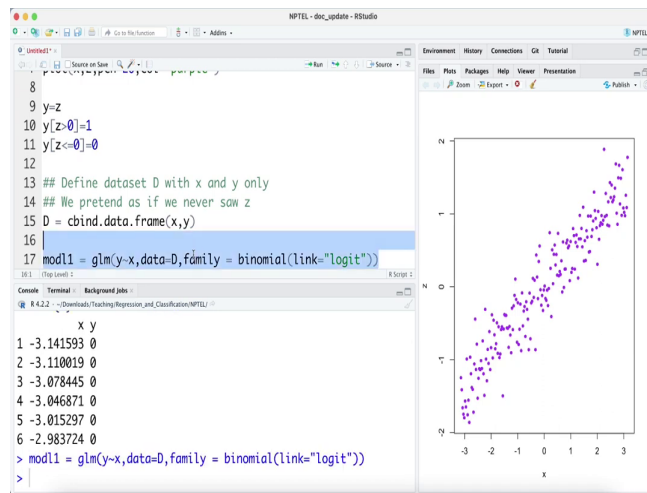


So, what I am going to do? I am going to simulate the y's ok. So, y equal to z ok, first I am going to define y equal to z and then I am going to say that first if z is greater than 0, you say 1, if z is less than equal to 0, then y is 0.





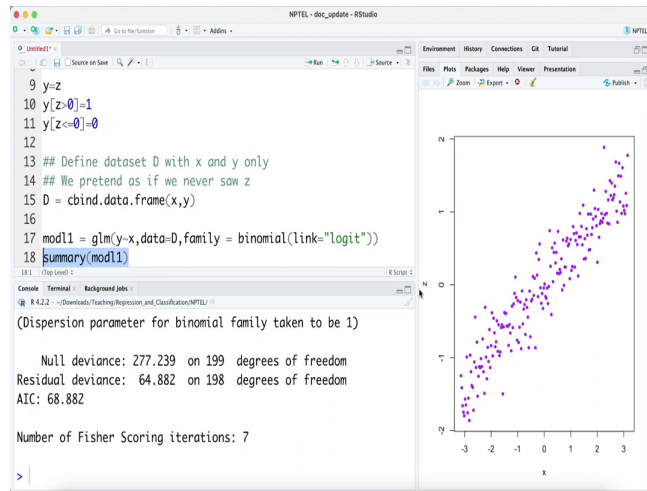
(Refer Slide Time: 04:54)



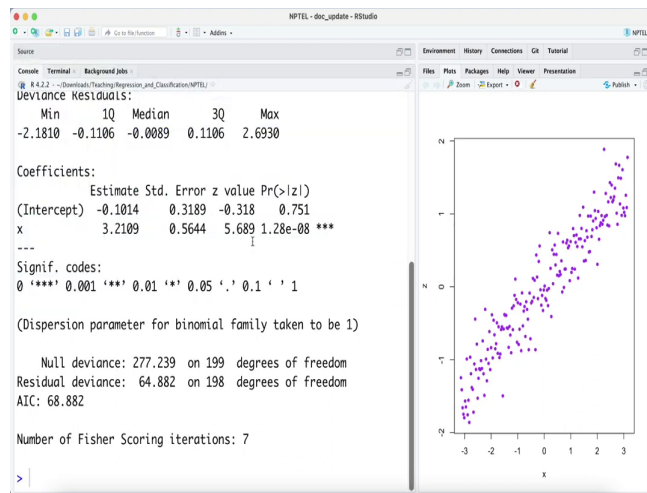
Now, y are all either 0's or 1 ok; now, so, y is the observed response. So, now, I am going to define the dataset D with x and y only, define data set D with x and y only, we will pretend, we pretend as if we never saw z ok. Now, what I am going to do is D equal to cbind dot data dot frame x comma y ok.

Now, if we just say head of t. So, these are my x values and these are my y values; so, y values either 0 or 1 alright. Now, I am going to fit a glm, say sigma, I will just say model 1 is glm, y tilde x comma data equal to D and I have to say family equal to binomial link equals to logit ok.

(Refer Slide Time: 06:41)

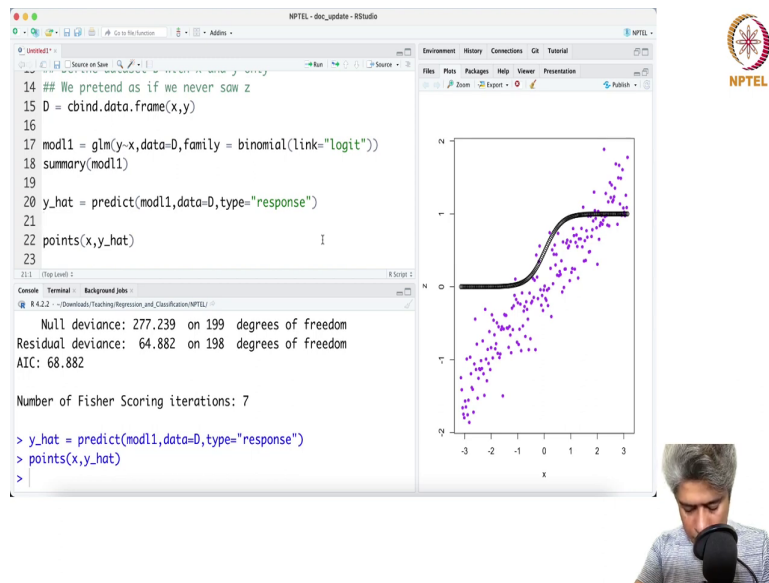


(Refer Slide Time: 06:51)



Now, if you do `summary(mod1)`; so, what is happening is, you see the coefficient of  $x$  is statistically significant, it is very small  $p$  value,  $z$  value is very high. So, we can say that as  $x$  is positive; so, probability of  $y$  equal to 1 keep increasing as  $x$  value will keep increasing ok fine.

(Refer Slide Time: 07:30)



The image shows a screenshot of the RStudio interface. The main editor window contains the following R code:

```
14 # We pretend as if we never saw z
15 D = cbind.data.frame(x,y)
16
17 mod1 = glm(y~x,data=D,family = binomial(link="logit"))
18 summary(mod1)
19
20 y_hat = predict(mod1,data=D,type="response")
21
22 points(x,y_hat)
23
```

The console window shows the output of the model fit:

```
R 4.2.2 > Downloads/Teaching/Regression_and_Classification/NPTEL/
Null deviance: 277.239 on 199 degrees of freedom
Residual deviance: 64.882 on 198 degrees of freedom
AIC: 68.882

Number of Fisher Scoring iterations: 7

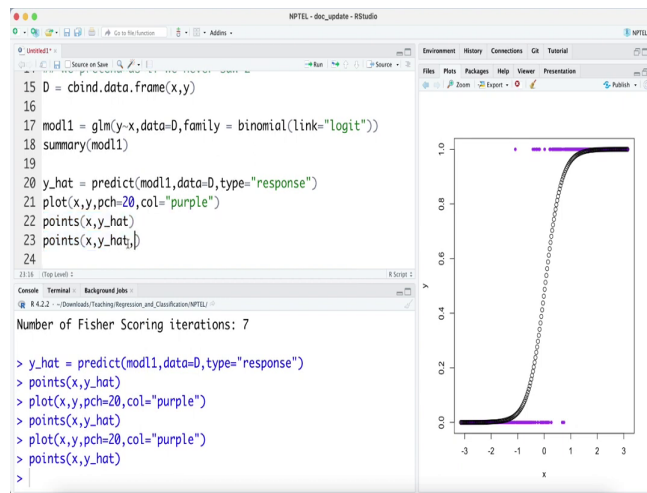
> y_hat = predict(mod1,data=D,type="response")
> points(x,y_hat)
>
```

The plot window displays a scatter plot of the data points (purple dots) and a fitted logistic curve (black line). The x-axis ranges from -3 to 3, and the y-axis ranges from 0 to 1. The fitted curve is an S-shaped curve that passes through the data points.

The NPTEL logo is visible in the top right corner of the slide.

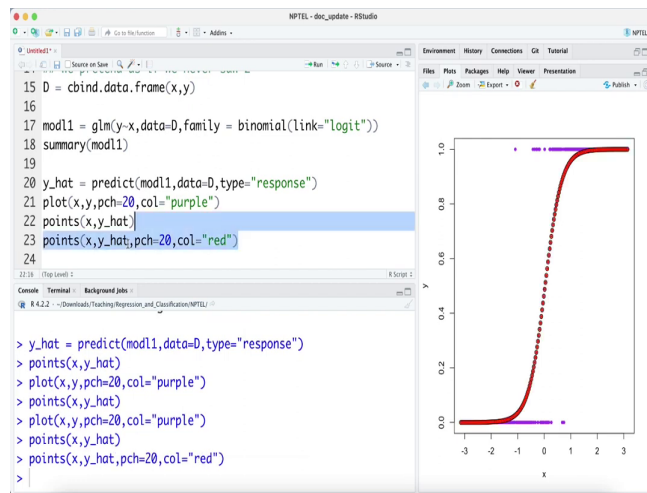
Now, what I am going to do, I am going to make some y hats, set predict, the model data equal to D and type equal to response. So, first, I am going to make some points, x comma y hat ok, x comma y hat, oopsie, sorry about that. So, what I have to do is, what I will do, I will just copy this and put it here and x and y and then points were like this, you can see these are the true actual x and y's.

(Refer Slide Time: 08:40)



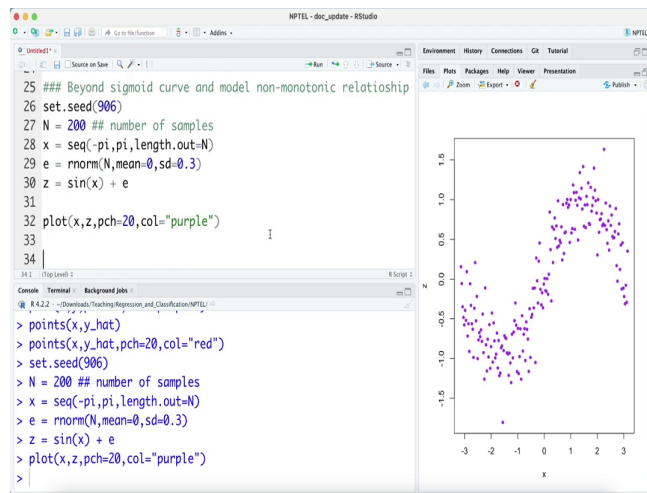
If I can I am doing it, its actual x values and the y values, y is all I taking at the 0 and 1.

(Refer Slide Time: 09:08)



And if I put the points  $x$  and  $y$  hat, and then I would be doing like  $pch$  equal to 20 with color equal to red wonderful; so, now its nicely sigmoid curve, a nice sigmoid curve. And if the, remember that if the underlying relationship is straight line, then this nice sigmoid curve will work out very nicely ok. But what happens if the underlying relationship not necessarily has to be you know straight line, that time we will, we may have a bit of a different situation.

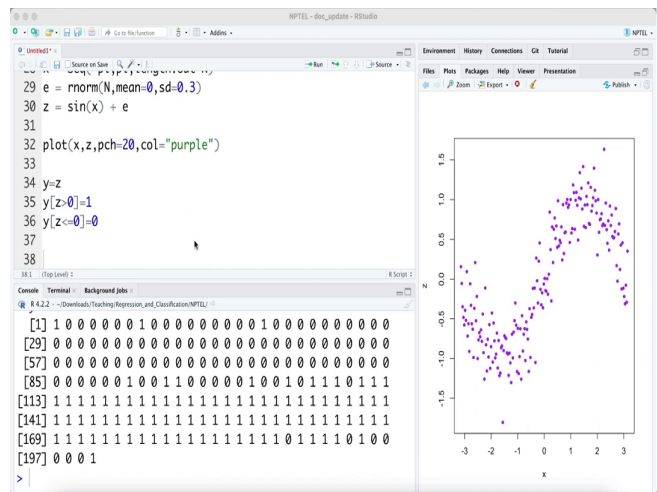
(Refer Slide Time: 09:57)



So, beyond non monotonic's and sigmoid means it is basically the relation through relationship will be always non monotonically increasing function. So, beyond our sigmoid curve; so, we want to go beyond sigmoid and sigmoid curve and model non monotonic relationship, tonic relationship ok. So, how we do that? Set dot; so, what we will do, we will just copy this part; just I will bring it here.

And this is my old model and I am going to change this model to completely different model sin x; let us see what happens ok, can it model. So, this is my y n x z plot x comma z, if I plot x comma z, pch equal to 20, color equal to purple. Ok. So, this is the relationship between x and z, but we are not seeing the z, this is the true relationship completely non monotonic behaviour, but we do not see the thing ok.

(Refer Slide Time: 12:07)



The image shows the RStudio interface. The script editor contains the following R code:

```
29 e = rnorm(N,mean=0,sd=0.3)
30 z = sin(x) + e
31
32 plot(x,z,pch=20,col="purple")
33
34 y=z
35 y[z>0]=1
36 y[z<=0]=0
37
38
```

The console shows the output of the code:

```
[1] 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
[29] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[57] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[85] 0 0 0 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1 1 1 1 0 1 1 1
[113] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[141] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[169] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 0 0
[197] 0 0 0 1
```

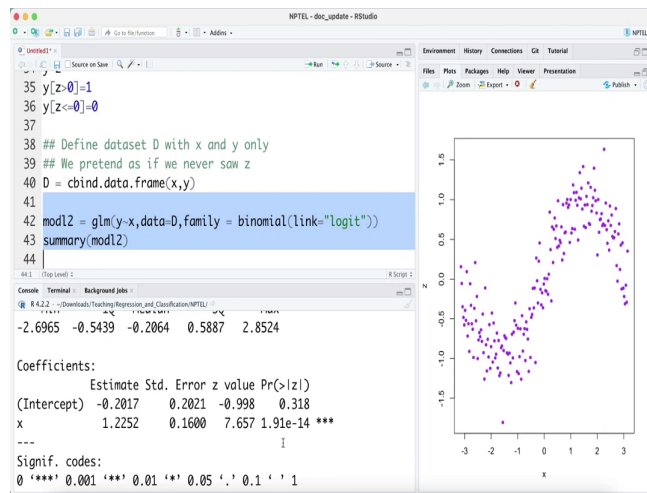
The plot shows a scatter plot of purple points forming a sine wave. The x-axis ranges from -3 to 3, and the y-axis ranges from -1.5 to 1.5. The points are colored purple and have a size of 20. The plot is titled 'plot(x,z,pch=20,col="purple")'.



So, if this is the situation and then what we can do? We can do the create the y's exactly the way we want, exactly the way we want. So, the y's are now all initially few 1s, 0s and the 1s, some 0s 1s; so, yeah; so, it is not that straightforward ok. Now, we are again, we are going to define that x, y and we will going to pretend as if we have not seen the z, the underlying, all we have the data x and y right.

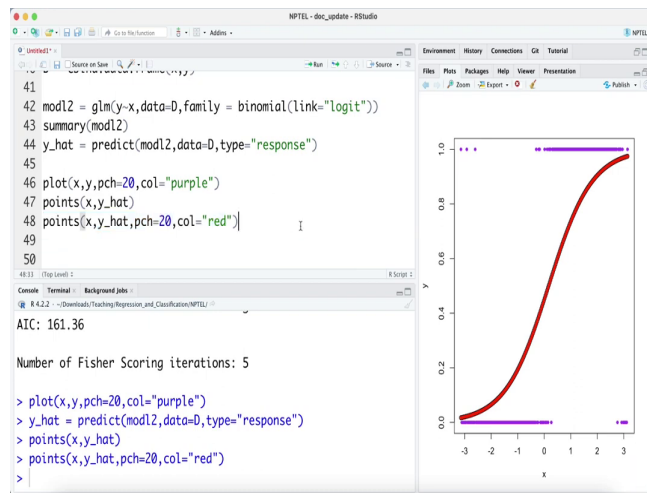


(Refer Slide Time: 12:50)



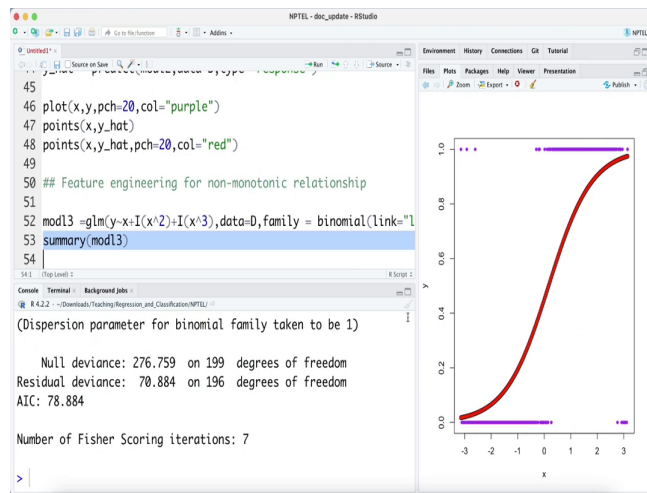
So, now we have that; let us try to model the, make the second model; let us make the second model ok. If I make the second model, now you see second model also the x is sort of statistically significant; so, that means, we are in a good shape. And if we want to have the plot this, we want to plot this.

(Refer Slide Time: 13:31)



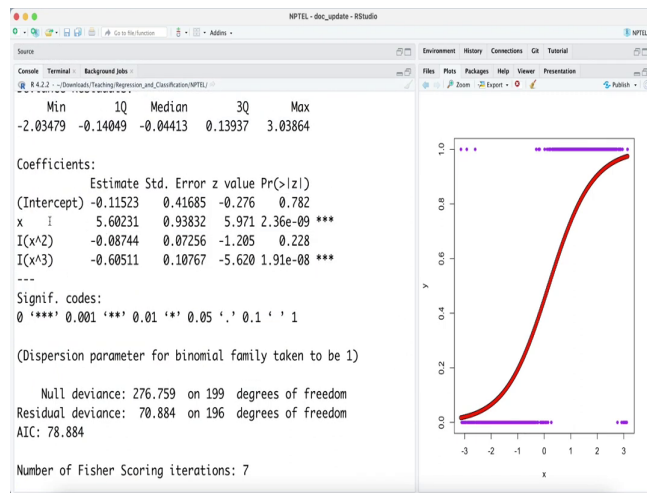
Now, clearly, we have some data points here, some points here, some points here and some points here. Whereas our points, now we have to have a y predict, y hat predict; so, let me just take this y hat credit and instead of model 1, I need model 2. Let me just run the y hat predicts through and it is a constantly non increasing. These points looks like does not have that match of Fa; so, it is going to be a straightforward model.

(Refer Slide Time: 14:30)



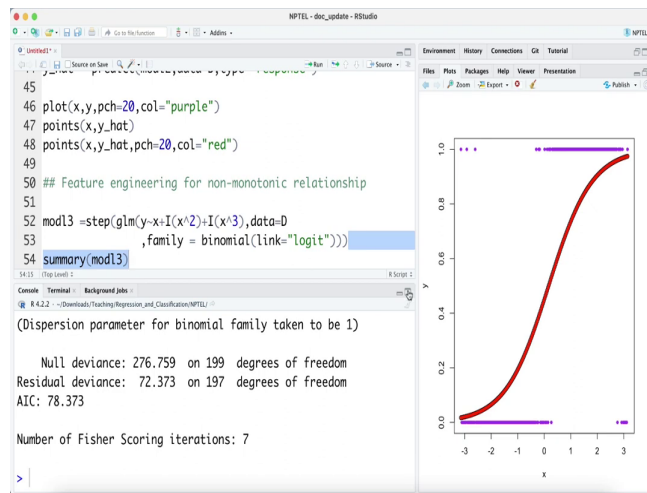
However, now what we are going to, we are thinking is, can we do some non-linear, non monotonic feature engineering with for non monotonic relationship, for non monotonic relationship ok. So, model 3 and then I am going to take, let us take this model directly and then plus  $I x$  plus  $I x$  square. Actually, this should be square and this should be cube ok.

(Refer Slide Time: 15:35)

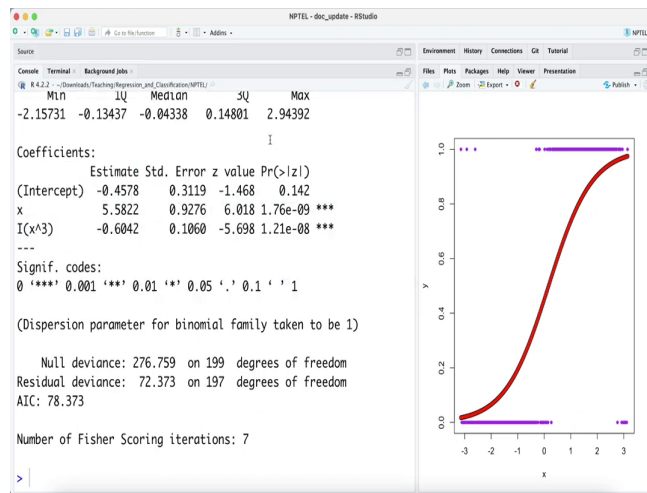


And if you see summary of model 3, oops we yeah summary of model three what we are seeing that coefficient for x and coefficient for x cube are statistically significant.

(Refer Slide Time: 15:49)

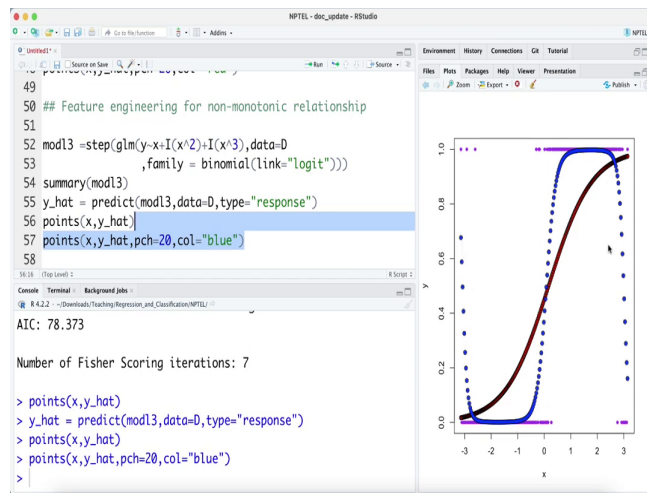


(Refer Slide Time: 16:04)



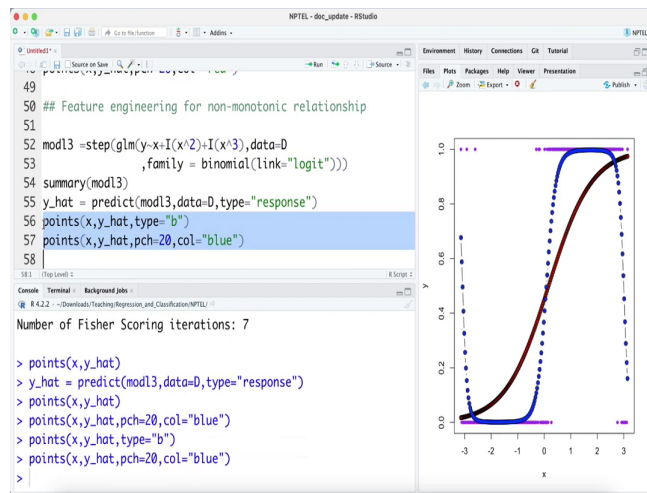
So, what I can do, we can just even actually do a step function that will do a stepwise protection stepwise selection, it will run. And if I do a stepwise selection, it will make the model with only x and x cube that is for best model.

(Refer Slide Time: 16:14)



And then what we can do, we can do a prediction based on model 3 and then now we can plot actually, yeah now, if I do this. So, this is the predicted values and now if I just do the blue, it is going to. So, the; so, this model; so, this prediction, this capturing the cubic relationship between x and y, this model is capturing the cubic relationship between x and y.

(Refer Slide Time: 17:04)



Whereas, so, non relay non monotonic, when because if you look into the real data, there is a sort of a, you know this is sin x that model does. Now, it is when I am giving higher order polynomial, it able to capture that, because there are few points here and there and that kind of gives you the best kind of fit. So, this is really the power of you know feature engineering that we see; so, I am going to share this code in the NPTEL portal.

So, please have a look, try yourself, try different feature engineering, see how it helps you ok; you can try to detect sin x, see if you use sin x, how good it would be, why not. So, you know good luck with feature engineering on this, but that is how we do in classification, binary classification; so, enjoy, see you in the next lecture, next video.