Mechanical Characterization of Bituminous Materials

Prof. Dr. M.R Nivitha
Department of Civil Engineering
PSG College of Technology-Coimbatore

Lecture-53
Introduction To Curve Fitting Using Matlab

Part 01

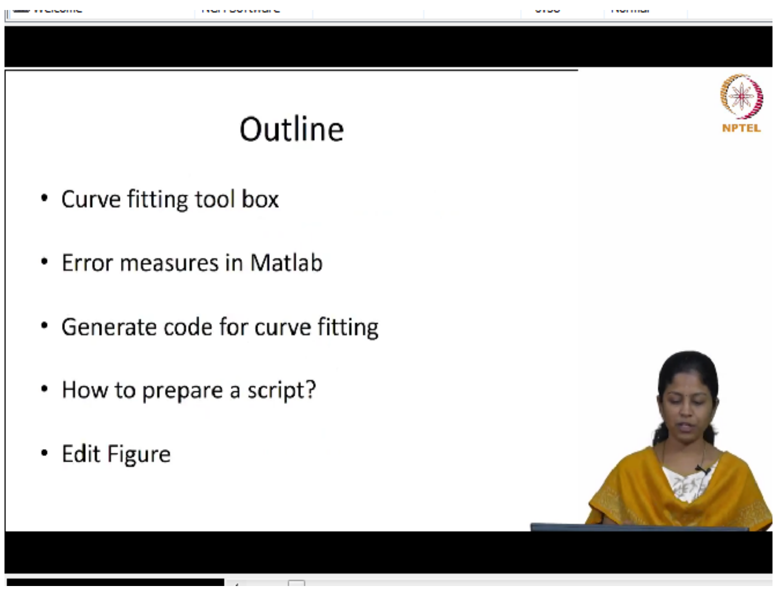
Hello all, welcome back. This session what we are going to discuss today is a very generic section which talks about curve fitting using Matlab. So, this curve fitting is something which is not specific to pavement or bituminous materials alone. So, this is very common and we use it for a lot of applications. So, this lecture here will be focused on giving some basic tips regarding the curve fitting using Matlab, right.

So, this lecture is for people who do not have any idea about what Matlab is or they are not familiar with the Matlab environment. It is not for those who already have some basics in Matlab. I am not going to tell you how to generate big codes or you know, do the curve fitting by writing your own code from the scratch. So what we are going to do here is use the curve fitting tool or app which is available in Matlab and then do some basic curve fitting.

And I will also tell you how Matlab generates a code for whatever you do. And we will be using that code to fit some data. So you are not going to write any code or nor I am going to teach you how to write a code. And this will only be an introduction session. So in the later sessions, Dr. Padmarekha will tell you how to use this curve fitting tool to fit some type of models to the creep and recovery data, which you get for your bituminous materials.

So it is going to be only an introduction part nothing related to bitumen at all right, so let us start looking at the outline of this presentation.

(Refer Slide Time: 02:04)

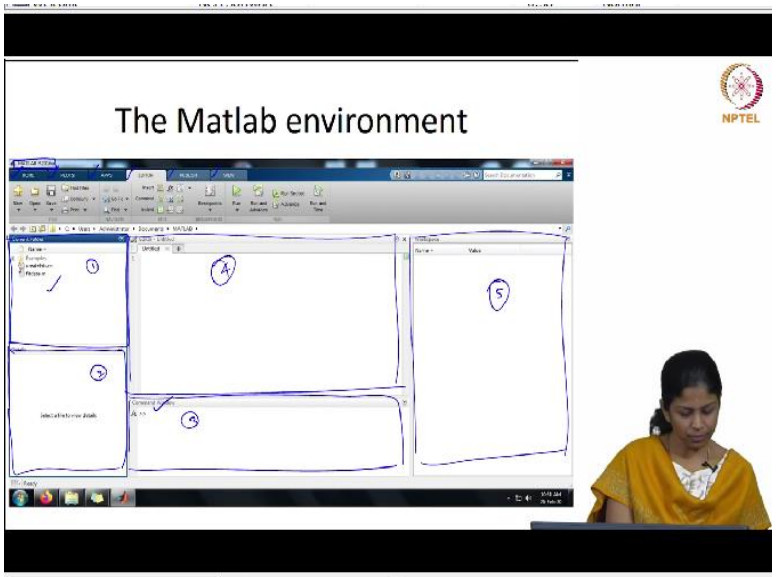


So there are 5 sections in this presentation. The first session is the introduction to the curve fitting toolbox. So, I will tell you how the Matlab environment looks like, what do you mean by an editor, what do you mean by a command window, what is a workspace. So, I will give you all the details related to that. The next one is how to calculate error measures in Matlab, because you assess the goodness of a fit using these error measures.

So, we will now look at what are these error measures and how are they calculated in Matlab. Third one is how to generate code for curve fitting. So, we will use the curve fitting toolbox, make some basic fits, and then we will ask Matlab to generate the code and we will use that, see how to use that code for a subsequent analysis. So that will be the third topic and the fourth topic is how to prepare a script.

So, every time you have to type in the command in your command window, rather what we would prefer is to save it as a file and then you just run that file, it will do all the commands whatever you have saved there. So, how to prepare a script. So, that is the fourth objective. And the last one is some basic tips how to you know, customize your figures, change your label axis, change some colors, fonts are some very basic tips. So this is the outline of this presentation. So the first one what we are going to look in here is your curve fitting toolbox.

(Refer Slide Time: 03:35)



So this is the Matlab environment. Again, I told you, it is not for those who have some basics related to Matlab; it is totally for users who just have heard the name Matlab, but have not seen it at all right. So, I will explain you the Matlab environment. So once you open the Matlab software, right, this is the window you will get, you have to note, there are different versions of Matlab, I am using Matlab 2016 a, you can see here, right.

So, this is the version I am using, and this environment looks similar for this particular version. And you will have some subtle changes for higher or lower versions, but the terminologies and the functions of each of this, you know aspects which are given in Matlab will remain the same, right. So, here you have a lot of options. You have a home button here, and you have options wherein you can edit your plots.

You have lot of apps, these apps were also previously called as toolboxes. And we have an editor where we will write the script and we can publish it, view it so a lot of options right. So, now, this box here shows the current folder, the current folder where we are in and what are the files available in our current folder. It will not only show you the Matlab files, but all the files what are available in that folder, it will list in this environment.

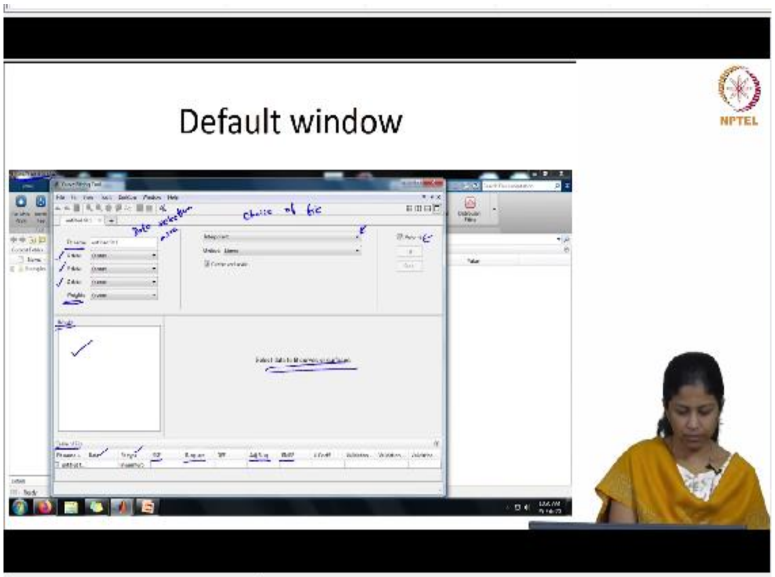
So, this is the first one. The second one is details, once you select a particular file. So, once I click some particular file, it will give you the details about that file in this particular area. So, this portion is the details related to each and every file. The third one is an editor, an editor is optional. You can give your commands in your command window itself and Matlab will execute it, only if you want to save a script and then you have to run it many number of times you can use this editor.

But the command window whatever you give here, it is continuous in time. So, as you keep you know typing some commands, it will after some time your old commands will be there but you have to retype it every time. Whereas in your editor, you can save it and then you can run it. So, that is your editor, we will come to editor later when we see where we are creating a new script.

So, for now, we will use your command window. So, editor is for writing some specific set of codes, saving it and then running it time and again; command window is to execute anything in your Matlab environment right. So, we will use command window, this is the third aspect here, right. So, an editor, we will see later, this is going to be our fourth aspect and the last one is the workspace right.

So, any variable we input here, so, all those variables will be listed in your workspace. So, if you input say x and y, 2 sets of data, your x and y will be displayed in your workspace, all your variables what you are using in your current file will be saved in the workspace. So, these are some of the basic aspects, we will be referring to them time and again. You will become more familiar with that when we look at some specific examples.

(Refer Slide Time: 07:11)



So, now let us just use the command window, we will come to the editor part later, how to prepare a script using an editor. For now we are just using the command window, right. So this is where you get 2 arrows like this. So you can start typing whatever command you want here. So, what we are going to do here is we will now open a curve fitting app straight away, get into the curve fitting app and then I will tell you how to input data when we come to selection of data in your curve fitting app.

So we have a tab here which is called as apps. In some earlier versions, they are also called as tool boxes right. So, you will have a curve fitting toolbox or an app here. So you just select this curve fitting toolboxes, you just click it. So once you click it, this is the environment that we will get, the default window which is generated.

Again this view is for your 2016 version as I mentioned earlier. So, this is the window that we will get, the default window, you can see there are a lot of options here. Automatically we have an untitled fit 1, the name is generated by default, and we have options to select our x data, y data, z data and we can also give weights to each of these variables. So, this is your data selection area right. The next one is choice of fit.

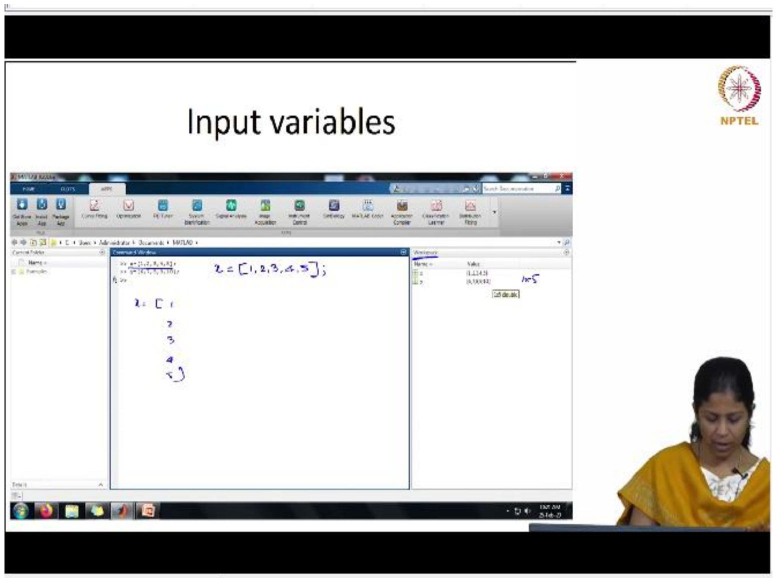
So, here we have an option, right. So it is by default it is mentioned as interpolant. So we can choose what type of fit we want for that specific data. So, that is given in this place, and how the fit should estimate the best parameters. So, we have again different methods for this and here we will be using linear least squares estimate. So, that is given as linear and you can center and scale your data.

And this option here is once you enter the data, do you want Matlab to automatically fit like select the appropriate type of fit for the data and fit automatically or you want to do it manually by yourself. So, if you want auto fit, you can check this or if you uncheck this, you have both of them enabled. So, you can fit or stop whenever you want. So, this is regarding the auto fit and the results you will see in this area.

So, here you will see what type of fit, what are the constants in the model, what are the coefficients for your variables, what is the goodness of fit. So, all those details you will see in this results area and this area here, will show you the fit, the figure, the fit for your given data. And here you can see, this is table of fits. So you have a number of parameters, your data, what type of fit it is, and these are some of your error measures, SSE, R square, adjusted R square, RMSE. So, all these things we will see in detail subsequently.

So, this gives you all these error measures and lots of other variables also. Let us not worry about them for now right. So, this is your default window, what you get when you open your curve fitting toolbox.

(Refer Slide Time: 10:39)

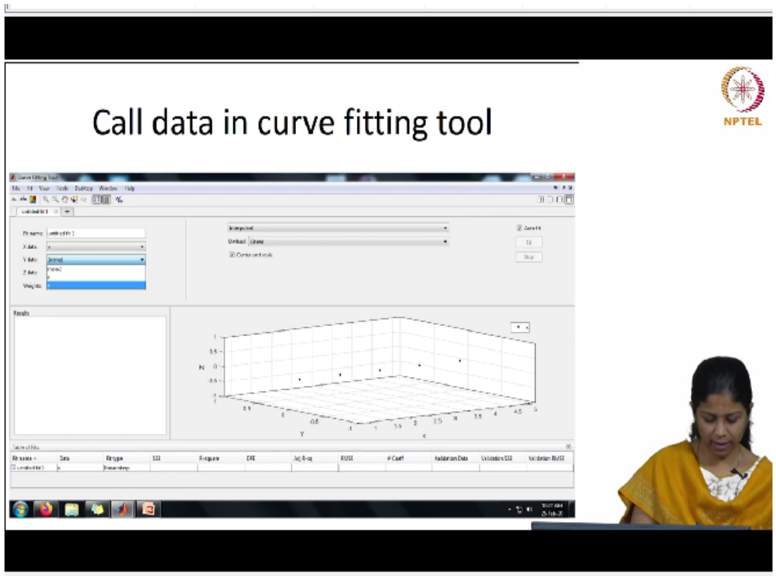


Next, we will input some data in Matlab because to start your curve fitting, we have to choose x and y. We cannot input x and y in your curve fitting app. So, we have to give x and y in your Matlab environment, and you can call them in your curve fitting tool. So, what we do here is I am just giving some very simple inputs for x and y. So, let me give 1, 2, 3, 4 and 5 for x, you can see you have to define a variable like this.

Give the variable name equal to open square brackets, write, separate each of the variable by a comma. So 5, I finished this, if I give this semicolon here, then it means that it will be listed. So, if I do not give this what will happen is, it will say x is equal to 1, 2, 3, 4 and 5, the command window will list this. So, I have to give this here and similarly for y, I input y as 6, 7, 8, 9 and 10. So I have input my variables x and y.

And once I click enter, you can see that in the workspace, you can see both x and y, right. So you can see the value for x and the value for y. So it is 1 by 5. It has 1 row and 5 columns. So it has a default way of taking your numbers. So once you enter, it will always take it as 1 row and how many ever number of columns you want. If you want it the other way, if you want it to be in a single column and multiple rows, you do a transpose, right. You input your data here. So now your data is available.

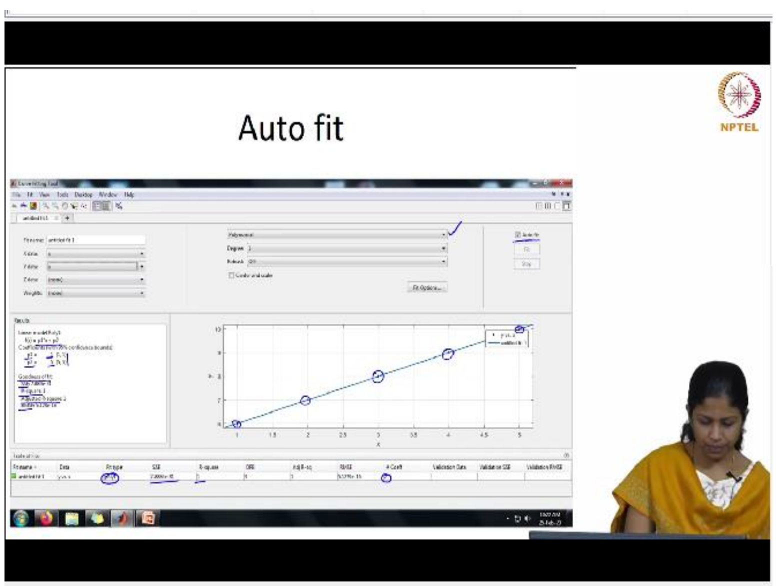
(Refer Slide Time: 12:20)



Now let us go to the Matlab environment, the curve fitting toolbox in the Matlab environment and then select your data. See you have given 2 variables x and y. So when you use this pull-down menu to select your variable, it automatically lists what are the variables which are available, right. So since I have given only x and y, it is listing x and y. If I give 4 variables, all the 4 variables will be listed here.

So, what I do is, I select x for my x data. And then so once I select my x data itself, it has plotted this in a 3 dimensional plot. Next, I choose y for my y data. So, I just choose y here.

(Refer Slide Time: 12:59)



See it has generated a fit. So, once I click y automatically it does all these things, because we have auto fit enabled here. So, what all it has done: it has chosen the type of fit here. So, it has chosen a polynomial fit, one degree polynomial fit, so, we know what a one degree polynomial fit is. So, you can see it is $P_1x+ P_2$, this is a one degree polynomial fit, so, it has also generated a plot, you can see x here, y here, you can see our 5 data points.

So, these are the 5 data points we have given right. So, it has generated a fit and it has also given you what are the coefficients. So, it has said there are 2 number of coefficients, the coefficients are P_1 and P_2, the value of P_1 is 1 and the value of P_2 is 5. So, there are also bounds the lower and upper bounds, in this case the value is same for both the cases 1 for P_1 and 5 for P_2 right. So, it has generated a fit, it has given you what are the coefficients for the model which is used here and it has also evaluated the goodness of fit.

So, this goodness of fit says how good the model can predict your data, we will come to this goodness of fit later. So, when I explain you about the error measures, we will see what each of them mean. So, as of now, you can see it lists 4 kind of error measures your SSE, R square, adjusted R square and RMSE. So, we will see what they are, just wait for a while right. So, it has given you in a table it has specified what are these values.

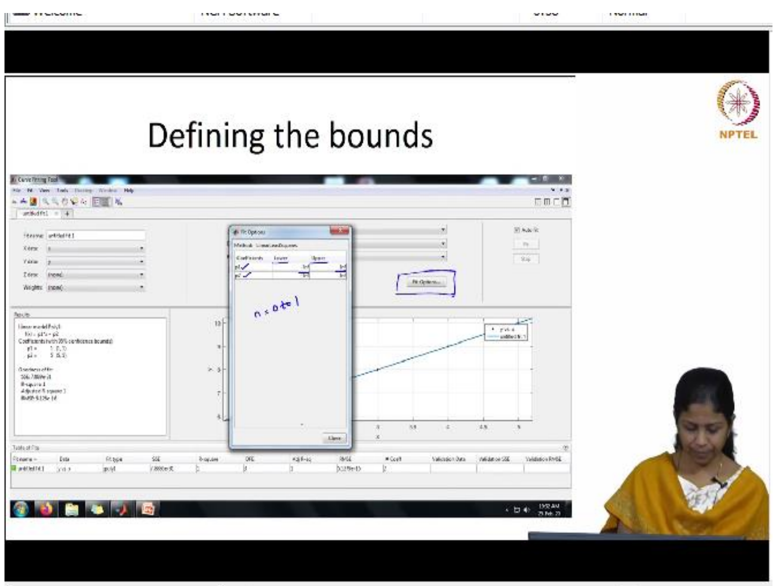
So, your data is y versus x, your fit type is one degree polynomial, so, it has some acronym for that and it has mentioned as poly1, then you can see what is SSE here, R square here. So, you should also know whether SSE has to be minimum or maximum. Similarly R square should be lower or higher for a good fit. So, and again your RMSE. So, all these things also we will see subsequently. So, this is your basic data you get once you enter your x and y data without just doing anything at all.

(Refer Slide Time: 15:24)

Now we will see, if you want to choose some other form, say I do not want Matlab to choose some particular type of the fit for this data. So I want to give me a custom equation right, so then I can choose from the available list of models which is given in Matlab, I can need, so the first one is I can give my custom equation or I can choose an exponential fit Fourier, Gaussian, interpolant, a lot of options which are available here.

You can also give your custom equation and ask Matlab to fit it for that particular model, so this option is available here.

(Refer Slide Time: 16:02)



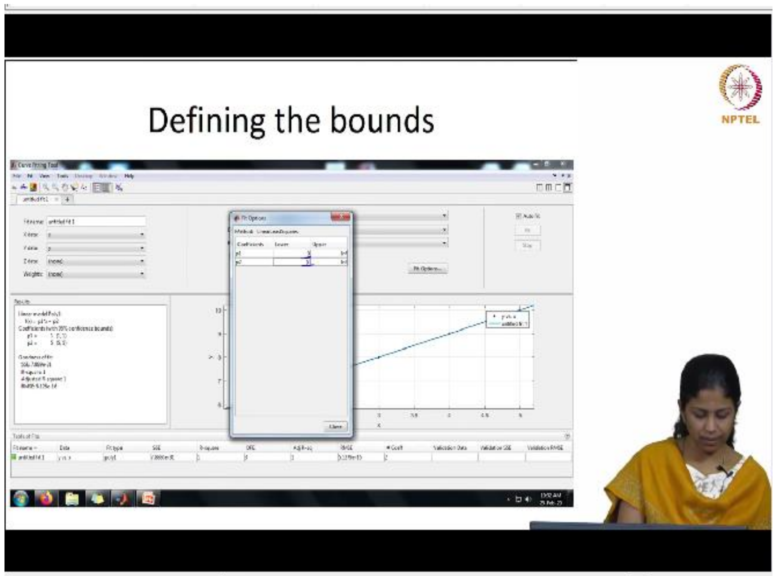
Now what we will see is the fit options which are given here. So, you have this fit options tab here right. So, once you click on this fit options, it lists the coefficients. So, we said in our model for a one degree polynomial, we have 2 coefficients your P_1 and P_2. So, for these 2 coefficients P_1 and P_2, the lower bound and the upper bound is specified here. So, this is very important when you fit some models which have lots of solutions, not a single solution but lots of solutions.

So, in that case, you know, in what range a variable can vary. So if we have a say if I take air temperature, I know that it can vary from say -40 to a maximum of you know, 50 or 60 right. So I know the range, so we get our data from some sources in field. So we know in what range this particular data can vary. So, accordingly you can give you a bound so that Matlab will not consider the other solutions for this particular parameter right.

So, and you can also say that if I measure some modulus or some stiffness parameter it cannot be negative in most cases. So, we know in reality the range in which the variable can vary. So, accordingly, we can choose our coefficients here. By default, it is specified as minus infinity to plus infinity, the lower is minus infinity, the upper is plus infinity. So, if I say in a power law model we say that the n will vary only from 0 to 1.

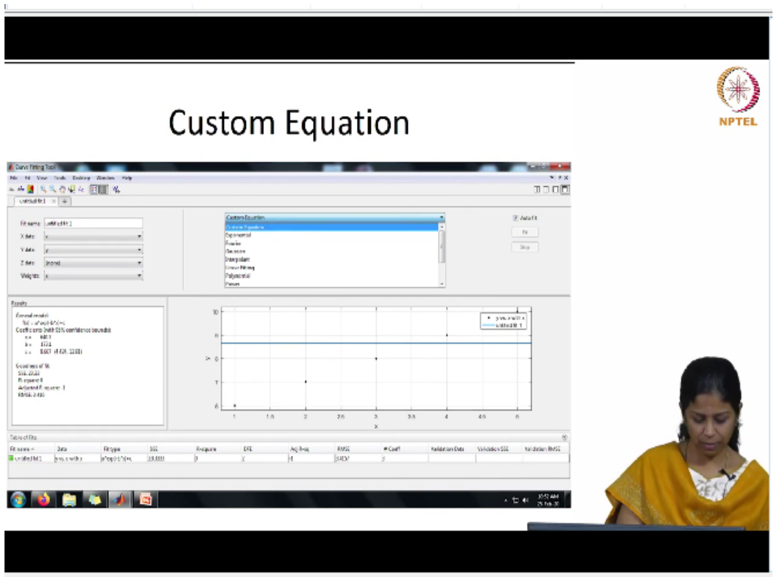
So, in that case I can fix this as 0 and 1, I can change it so that it will only look for solutions where n will vary in that range alone. So, this is one possibility. And in many cases, we have to specify this for Matlab to give us a solution with the coefficients which are realistic to what we are expecting.

(Refer Slide Time: 18:02)



So, here I have changed it to 0 to 1 right just to illustrate, I have changed the lower limit from 0 to 1. So, you can give any range here.

(Refer Slide Time: 18:14)



So, here I am trying to choose a custom equation here. So let us say that I want to type in my equation, so I am going to use a custom equation here.

(Refer Slide Time: 18:23)

So, I give, so it says y right, which is your dependent variable is a function of all your independent variables. So, here x is only the independent variable. You can also give a number of independent variables. So, that is where you have to define first y as a function of its independent variables. So, here we write f of x alone, right. So, then here, I write instead of fitting $y = mx + C$ or which is given here right, which is $P_1 x + P_2$ right.

I just want to do $y=mx$ alone. So, what I do here is I give $y =ax$, I type in my customized equation right. So, then I try to evaluate the fit for this equation. So, once I type this you can see the new fit is generated because we have an auto fit option. So, once I just type these ax, it automatically generates a fit for this particular equation. So, we see here x and y are given here, these are the data points and we can see that the fit is not correct because the constant term is missing here.

You can see it has specified the coefficient of a, its lower and upper bound, so it is again for 95% confidence. I will not get into all these statistics. So, you can see what a 95% confidence is and you can see how the lower and upper bounds are calculated and how much it will vary depending upon your confidence interval, right. It is also evaluated here, SSE, R square, adjusted R square and RMSE.

And it has also given you a warning. So it says a negative R square is possible if the model does not account for the constant term, right. So, it also gives you some suggestions to improvise your model. So, this is your customizing equation. So, I have just used a very simple equation, you can use any of these equations for which or the models for which you want to fit your data.

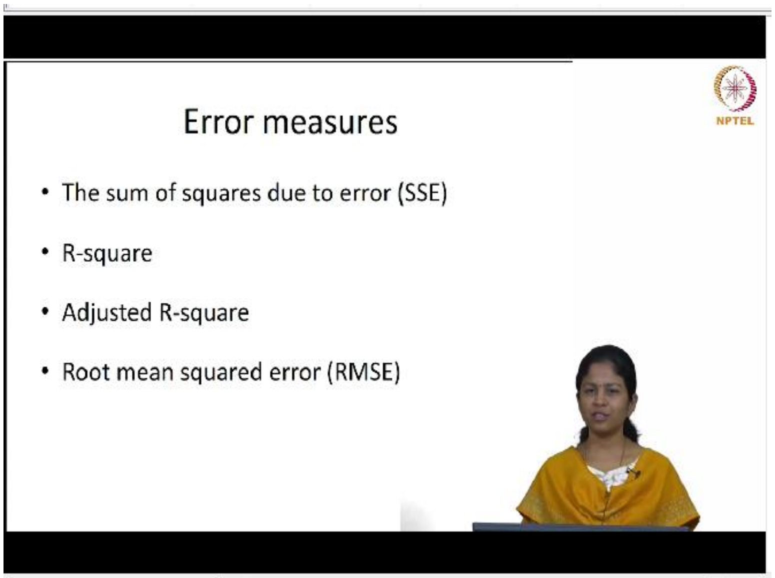
(Refer Slide Time: 20:36)

Next, I will show you how to compare 2 fits. So, here we can see, initially we have kept it as untitled fit 1, right now, I want to generate a new fit. And then I want to compare the coefficients or compare the error measures for these 2 fits. So whatever I want to do, I want to keep all these things in a single space and then view them. So, for that you go to the fit option which is given here and then you choose new fit, right.

So, once you chose this new fit, you can see an untitled fit 3 is generated here. So, again by default it takes the most appropriate model, it does all the fitting here and it shows you the results for untitled fit 3. So, again you can compare what are the error measures for these 2 cases right. So, this also has 2 coefficients. So, like this, you can generate multiple number of fits, save them in the same window and then compare them.

So, it need not be a polynomial alone. One can be a customized equation vis-a-vis the second can be some standard form of model, which can fit your data. So, you can make those comparisons here. So, next we will move on to what are the error measures that are available in Matlab.

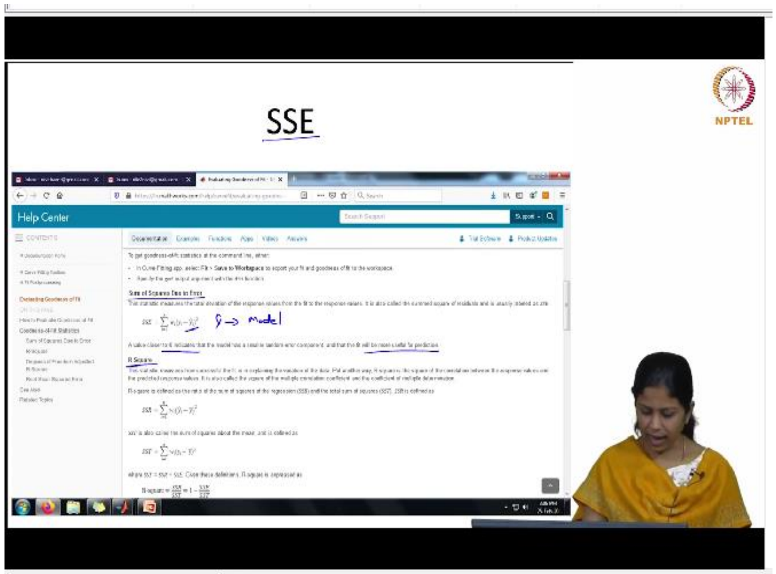
(Refer Slide Time: 21:57)



So, Matlab lists 4 types of errors measures by default to define the goodness of fit for any type of fit. This is not only for Matlab, whenever we use a model to predict some data, we evaluate how well the model can predict your data, or how closely the model can predict your data. So, for evaluating that, we need to associate some numerical value and that we do by using any of these error measures. The first one is sum of squares due to error, which is called SSE.

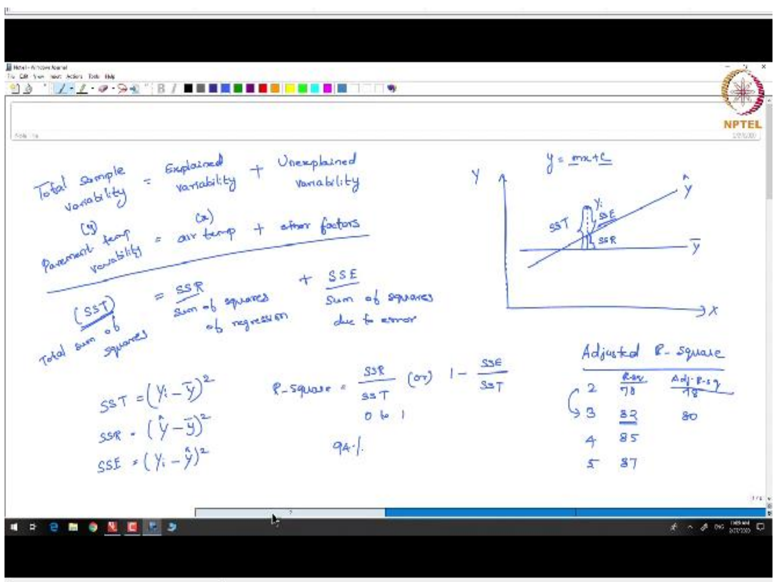
So, I was telling you earlier right, so, we saw in Matlab that when you evaluated the goodness of fit, so we had SSE, we had R square, we had adjusted R square and we also had root mean squared error. So it had listed these variables. So there are a lot of other error measures also. But for this lecture, we will contain it to these 4 error measures.

(Refer Slide Time: 22:55)



And as I have mentioned earlier, SSE, which is sum of squares due to error is one of the error measure. But before seeing about this error measure in detail, I will give you some background about the fitting which is carried out.

(Refer Slide Time: 23:12)



So, the total sample variability can be split into 2 terms, one is the explained variability and the other is the unexplained variability. Let me give you an example. We all know that the pavement temperature depends mostly on air temperature. So, if you know how the air temperature varies, you can also predict how the pavement temperature will vary, but not all subtle variations in your pavement temperature is due to air temperature.

Some amount or some percentage of the variability in your pavement temperature is due to air temperature and the remaining things are due to other factors. So, in the total variability of my pavement temperature, how much of the variability can be attributed to air temperature and how much can be attributed to other factors. So, we can explain this here. So, if I say the pavement temperature variability, how much can be explained my by air temperature and how much is attributed to the other factors.

So, this total sample variability is called SST which is nothing but total sum of squares. And this explained variability is nothing but SSR which is sum of squares of regression and this unexplained variability we call it as SSE, which is nothing but sum of squares due to error. So, my total variability I split it into the variability which can be attributed to the model and the variability which cannot be attributed to this which occurs in a random.

So, this is how I evaluate this. Let me now use an example to explain this, let us say we have x and y here, let us say that we have mean y bar here, we have a model, I will denote the model by $\hat{y}$. And I will say that my data point y_i is here, let us just take it as an example, just for the sake of illustration, right. Now, with this I will define what is SST, SSR and SSE right.

So, from my mean right, whatever is the variability, this I call it as SST, that is the total variability from my mean. So, I will write SST is nothing but y_i - $\bar{y}$, the whole square because we said it is sum of squares. So, SST is nothing but the variability of my data point y_i from the mean right, that I defined by SST and we said SSR is sum of squares due to regression. So, how much of this variability is attributed to the model right.

So, from my mean my value is varying so much. So, how much of this variation is explained by the model is due to the variability in my independent variable x. So, that is this which I call it as SSR. So, this SSR is nothing but the variability that is explained due to my model. So, this is nothing but $\hat{y}$ which is the model minus the mean value. So, in my total variability, here, this much of it is explained by the model.

So, now, whatever is the unexplained variability, we call it as SSE, which is sum of squares due to error. So, it is due to some random error and not attributed to the model and this SSE is nothing but now you will be able to say how we define SSE, it is the variability of y i minus the model. So, that we can write it as y_i - $\hat{y}$ the whole square right. So, this is how we define these 3 terms SSE, SST and SSR.

And we have said that the first parameter is sum of squares due to error. So, it directly takes this value, sums it up for all your y values and then it calculates SSE. Now, how the next one is calculated which is nothing but your R square value, R square value says how much of the total variability right, so, I said the total variability is SST. So, in this total variability, how much is explained by the model.

So, which is nothing but your R square is $\frac{SSR}{SST}$ or it is 1- $\frac{SSE}{SST}$. So, I have a total variability. If my model is able to predict most of its variability, then my R square value is higher. Let us come back to the same example pavement temperature, air temperature variability. Let us say I am trying to fit the pavement temperature variation using air temperature. This is my only y and this is my x.

I am trying to explain the variability in pavement temperature associating it to air temperature. So, if I get an R square value of 94% for this case, then which means that 94% of the total variability in my pavement temperature can be explained by air temperature. So, that is why when we get a higher R square value, we say it is the model is good or the fit is good, right. So, this is why we say, higher R square value is better, and we can see from this that it is a ratio.

So, it can vary from 0 to 1. In some cases we can also see a negative R square value, but let us just not worry about that now. So, here we can see the R square varies from 0 to 1 right. Now, there is something called as adjusted R square right. So, we will see what this adjusted R square is, let us say I have a model, right in which there are only 2 terms, right. So, let us say right let us just take this simple form of model, there are 2 terms m and c right, there are 2 terms.

Now, I want to add one more term to this model. So, now, if I add one more term, because of the addition of the other term, the model will be, the fit will be the goodness of fit will improve right. So, the error measures will be reduced because I add one more term to this model. Now, we want to see the addition of the other term is because it is explaining the variability in data or it is because of some random phenomena, right.

So, I want to see the addition of an extra variable is explaining the model in a better way, or it is just happening because of some randomness which is occurring because I add another parameter. So, then this adjusted R square compares the goodness of fit and adjusts it according to the number of parameters in the model. Let us say I have a 2 parameter model, the goodness of fit is 78.

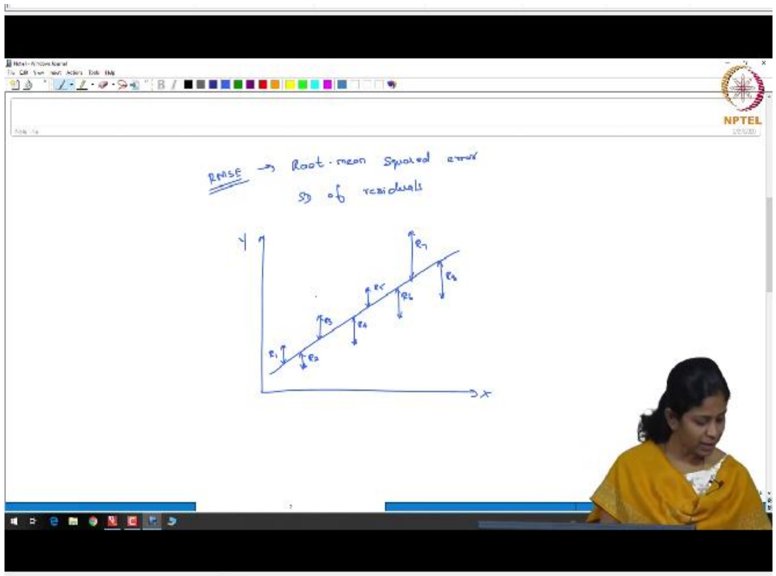
I have a 3 parameter model, the goodness of fit is improved to 82, I have a 4 parameter model, this became 85, I have a 5 parameter model, this became 87. Let us just assume a case like this. Now, I have added 1 parameter to this 2 parameter model. So, this increase in your R. This is just R square not adjusted R square, it is simply R square. So, this increment right from 78 to 82. whether this is because this parameter is trying to explain the variability or it is happening due to some random phenomena.

So it will evaluate that and it will calculate something called as adjusted R square. So, if I add one more parameter by probability, you can say that maybe additional 5% of the variability can be explained by addition of one more parameter. So, now, it will evaluate whether this variability is that probability which it has predicted or it is due to some other phenomena. So, then accordingly it will adjust your R square and it will say, this adjusted R square. So, here it will be like this.

So, here it means that if you get an adjusted R square of 80, then it means that only 2% among this 4% the additional parameter is explaining the variability and the remaining 2% are actually a random phenomena. So, that information you will be able to get by using an adjusted R square. So, it is not that we will be using R square alone and you can use any number of parameters.

How many parameters if you use you will get a better fit, that you will be able to evaluate when you calculate adjusted R square and the last parameter is root mean square error RMSE. So, this RMSE is a standard deviation of let me write them in a separate sheet.

(Refer Slide Time: 34:20)

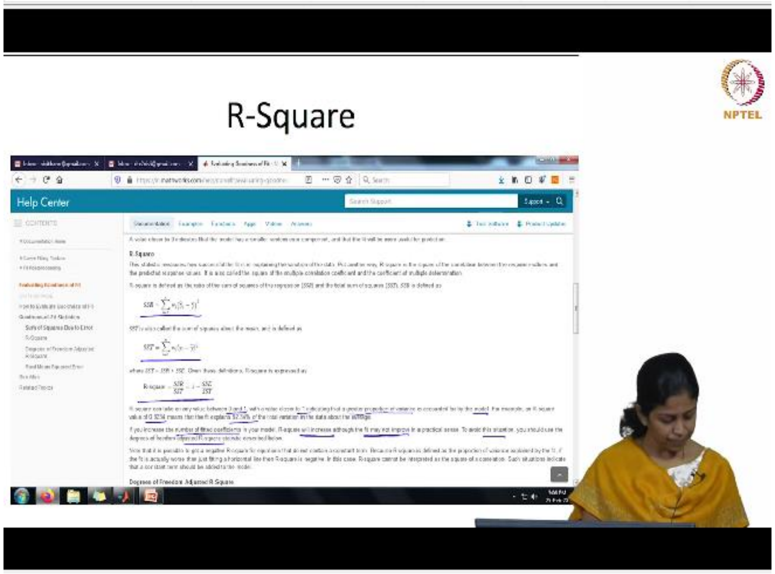


So, we have this RMSE which is nothing but right, it is standard deviation of the residuals. So, what do we call by residuals, so, I have ax and y here, this is my model. So, these are my data points right. So, what is the variability between my model and my data point right. So, this we call it as residual. So, this is R_1, let us say R_2, R_3, R_4, R_5. So, it has calculated the residuals of all these data points right.

So, this residual, the standard deviation of this residual is called as your root mean square error. So, by looking at this you will be able to say whether your RMSE has to be minimum or maximum for a good fit, right. So, this says how close are your data points around your fit, is the variability large. So, that is why it uses a standard deviation measure, right. So, these are the error measures in brief. Now, let us go back and look at what is defined in Matlab.

So, you can see the first one is sum of squares due to error. So, that we said it is the difference between your model and your data point. So, that you can see the model is given as this y cap is your model and this y_i is your data point right. So, this value closer to 0 indicates that the model has smaller random error component and the fit will be more useful for prediction. So, that is a conclusion you get depending upon the value of your SSE, the next one is R square. So we have seen what R square is right.

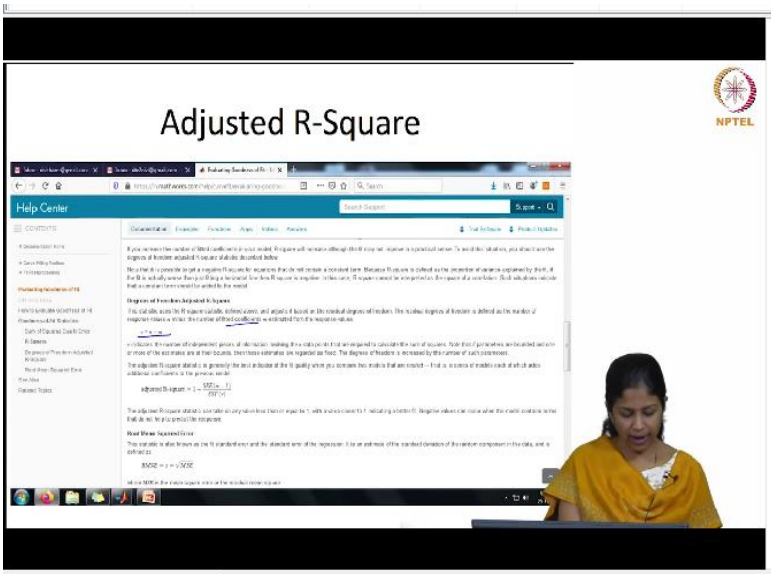
(Refer Slide Time: 36:51)



So we have calculated SSR. We know what it is, it is the sum of squares due to regression. So, it is the variability between your model and the variability from the mean to the model and this is SST the total and we can see how R square is defined. So, R square can take any value between 0 to 1 and a value closer to 1 indicates that greater proportion of variance is accounted by the model. So, we like we discussed earlier if you have an R square of 0.8234 which means that the fit explains 82.34% of the total variation.

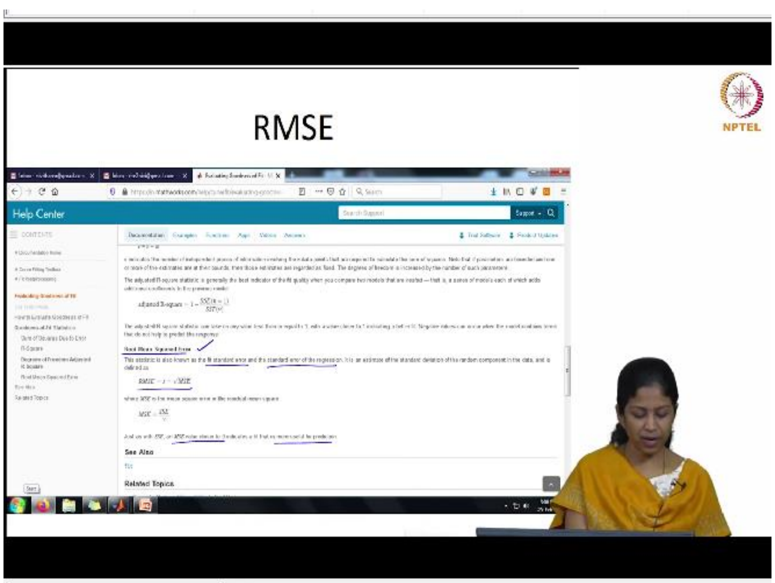
And here you can see the discussion related to adjusted R square. If you increase the number of fitted coefficients, R square will increase, but the fit may not improve. So, to avoid this, we should be calculating adjusted R square.

(Refer Slide Time: 37:50)



And this is adjusted R square. It uses something called as degrees of freedom. You can read about this, I will not get into the details about calculation of degrees of freedom, response values, fitted coefficients and all those things, but this is what adjusted R square means and the last one is root mean square error.

(Refer Slide Time: 38:12)



So, this root mean square error is known as the fit standard error and standard error of the regression right. So, this is how they have defined root mean square error, so, which is nothing but your SSE by degrees of freedom and as we have defined earlier for SSE, an MSE value closer to 0 indicates that the fit is more useful. So, we said that it is the standard deviation of the residuals.

So, we want to minimize the residuals as much as possible or the variability in the residuals as much as possible and that is calculated by root mean square error. So, these are the different error measures that are given in Matlab.