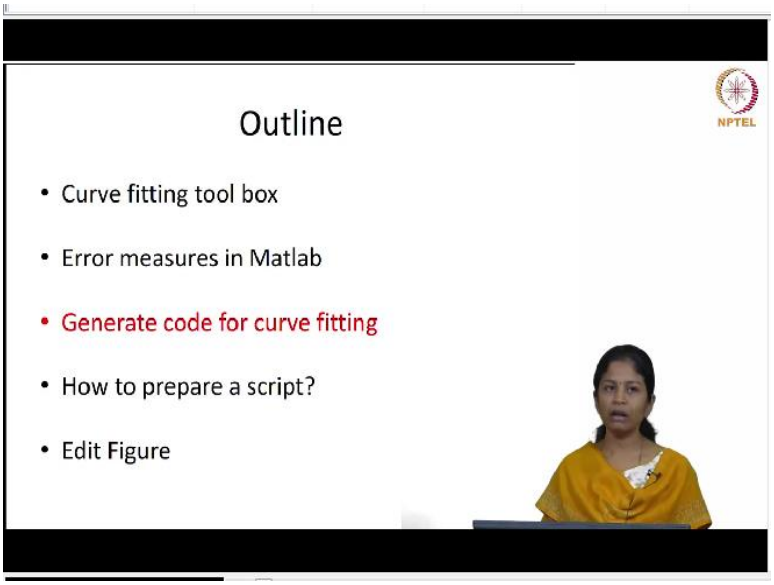
Mechanical Characterization of Bituminous Materials
Prof. Dr. M.R Nivitha
Department of Civil Engineering
PSG College of Technology-Coimbatore

Lecture-54
Introduction To Curve Fitting Using Matlab

Part 02

(Refer Slide Time: 00:14)



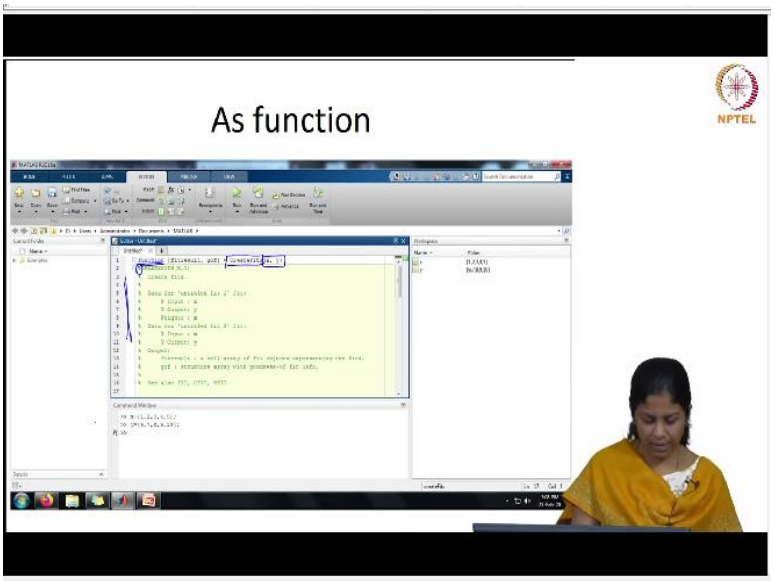
Now we will move on to the next session, which talks about generating code for curve fitting. So previously, I explained you how to fit a curve using the curve fitting toolbox. But what happens here is, whenever I have a data, I have to go to the curve fitting toolbox, select my x, select my y, and then I have to do all these things, whatever I do each and every time. So instead if we have a code and if we want to do some set of, some follow some steps in fitting, we can save all of them.

And then you just use the code to give if you input your x and y values and if you use the code, it will automatically generate the fit. So, that is very convenient when you have to fit models for multiple sets of data. Maybe identical fits will be very easy. So, we will see how to generate code for curve fitting.

(Refer Slide Time: 01:07)

So, once you do this right, so far what we have done is we have generated x, y we have called x and y and we have created 2 fits, untitled fit 1 and fit 3 and we have used a 1 degree polynomial. So, we have done all these things right. So, whatever you have done so far, it will be saved in the memory and when you say file generate code. So, this option is available here, when you say file, generate code, the code for whatever you have done so, far will be generated.

(Refer Slide Time: 01:41)



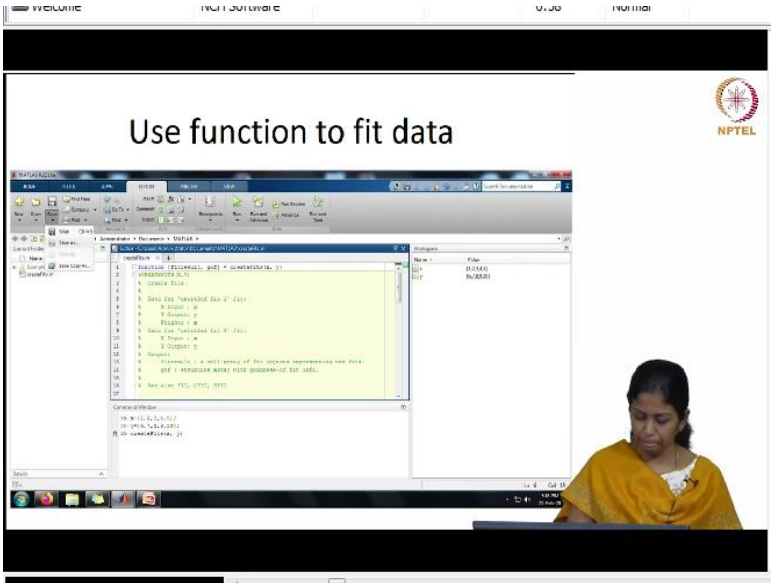
So, you can see it is a function; it is generated as a function which means that you can call this function whenever you want to do something. So, if I have to fit, you know like this curve fitting is part of some big operation, I save the codes related to curve fitting separately. So, that is what

we call it as a function, I save the codes related to curve fitting separately and in that big part, if I want to call curve fitting, I just say create fit and then I give the input values.

It will do whatever is defined in this code right, we will see about it. So it is generated as a function. So, this name here is the name of the function and it has 2 variables, x and y right. So, this is what is required to curve fit for any other set of data and we will see what is listed here. So, these see whatever is commented here something like this percentage; it is just for you to understand and Matlab will not consider that for execution.

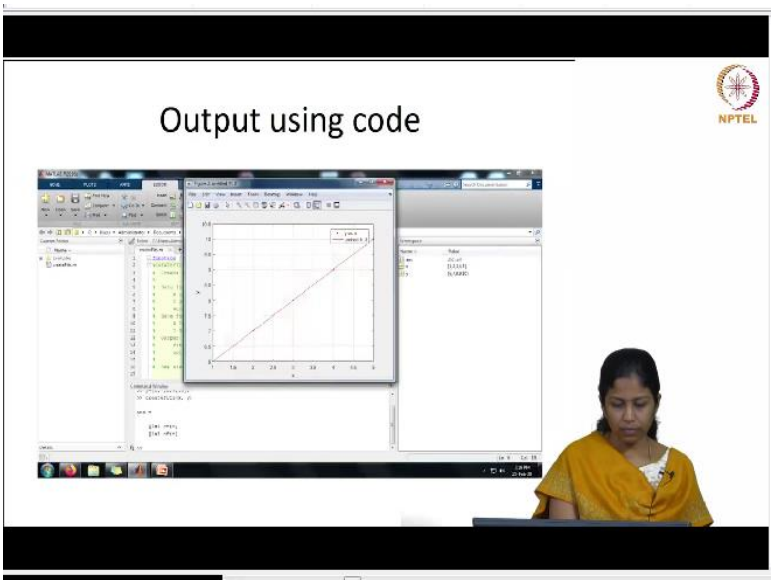
So, that is what is given as a percentage symbol here. So, if you want to you know understand, you know remember something, you can just use this percentage symbol and that will just remain there; it is for your reference. So, if you have done some 10 steps, you want to say in this step I am doing this, in this step I am doing this, you give percentage and you just write some comments. It is just a commented line, it will not be considered for execution. So, we will see.

(Refer Slide Time: 03:18)



So, first we have to save this fit, because it is a function file. So, to call that function you have to save it first. So, without saving this if you just try to create with x and y, it will actually show you an error, right. So, you have to save that first, so you go to save, click save as and then you just save this right. So once you save this, it is saved in this folder, create fits is saved here.

(Refer Slide Time: 03:47)

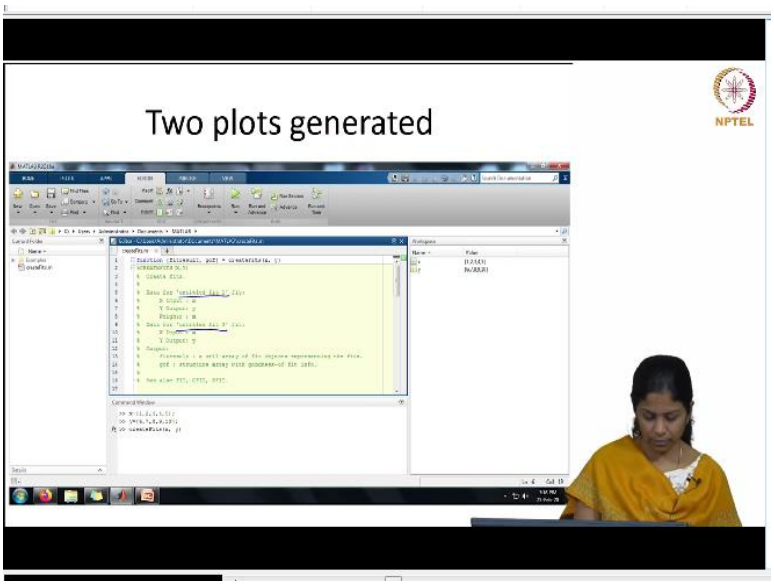


So now I give create fits of x and y right, so once in the command window I type create fits x and y, my x and y is already available here. So once I give create fits of x and y. So, you have to give the function name and the 2 inputs, so I said x and y are required here right, the 2 variables. So, you have to give values for these 2 variables. So, once you give x and y, the fit is generated here.

So, whatever we have done previously, you can see, the same thing is available here. But you have to note something, we are seeing 2 figures here, one for untitled fit 1, and another one with the name untitled fit 3. This is because previously when we saved the session, we saved it for 2 different fits, because in the lower tab, you would have seen that there were 2 fits which we have done and it has actually saved whatever was there in its memory.

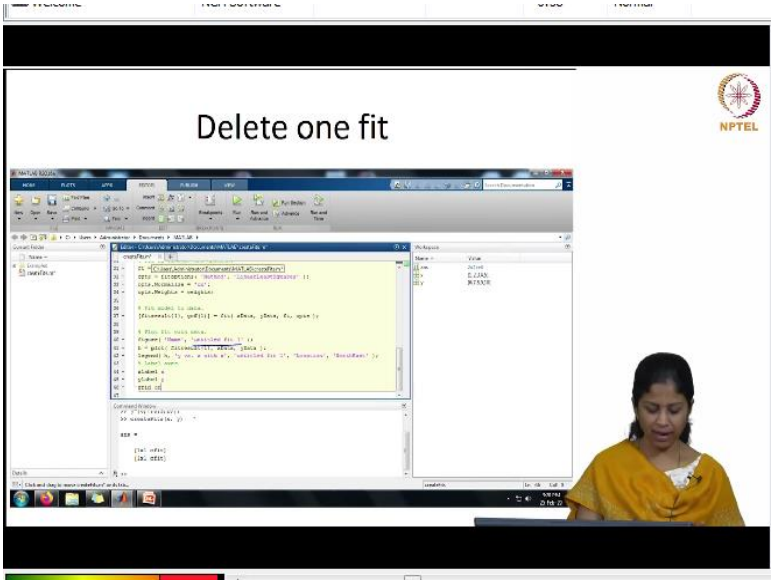
So, when you generate the code, you should remember that whatever you have done in that session, right from opening that curve fitting app, whatever you have done, everything will be in the memory. So once you finalize your procedure, you just go do the exact procedure, then if you generate code, it will be convenient. So, I have just shown this here to illustrate that particular aspect. So, now we have 2 different fits here. So, I do not want 2 fits. So, I will go to the code right and then I will delete 1 fit from this right.

(Refer Slide Time: 05:12)



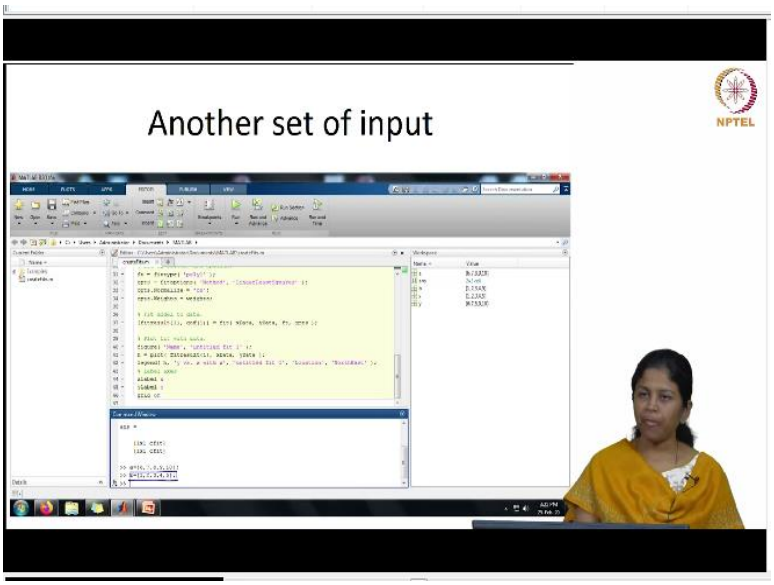
So, you can see, it is for untitled fit 1 and fit 3, both are considered here. So, I will go delete the fitting options which are available for one of these fits right.

(Refer Slide Time: 05:26)



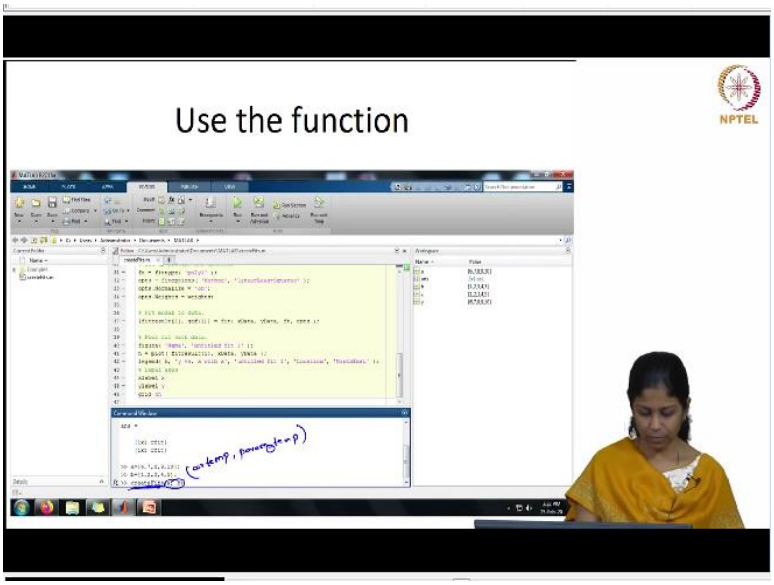
So that is deleted here and you can see what are the other aspects of the code, we will come to that, I will show you what are the other things. So, here I have just kept untitled fit 1 and we have deleted untitled fit 3. So we have only 1 fit now and now if you use this to fit it, we will be getting only one window here, right. So next what we are going to do here is, we are going to see how to use this to fit for some other data. So we already have 2 variables x and y, now, I am going to give some other variable.

(Refer Slide Time: 06:03)



And this variable I have defined it as a and b. So, you can see I have given a as 6, 7, 8, 9 and 10 and b as 1, 2, 3, 4 and 5 some other data right. So, you can have any other data for this or for this some other data, I want to use this function, I want to do the same fit for this particular data. So, what I do here is I do not have to go to the curve fitting toolbox, click all those things and do it.

(Refer Slide Time: 06:34)

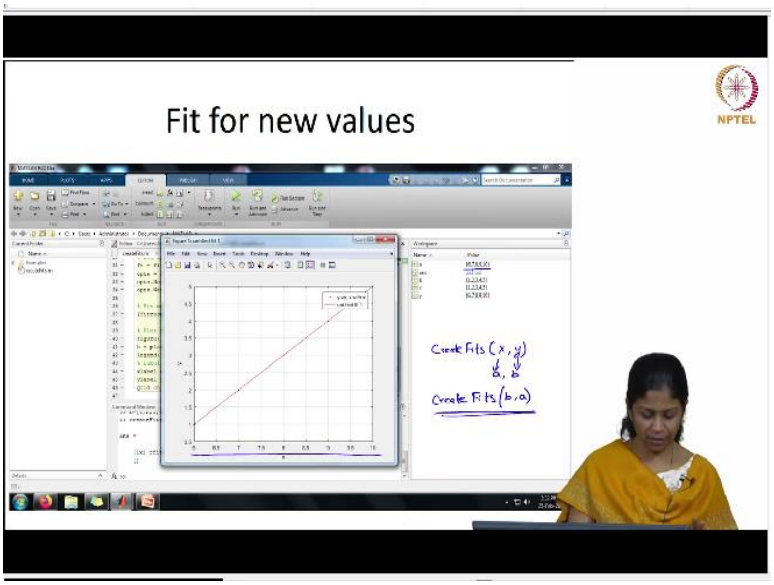


I just do create fits a and b, because it is not x and y, x and y is not fixed, whatever is the name of your variable you can change it here, the function name will remain the same. So, when I give create fits, it will remain the same, but I can change my variable. So, I can have a b, any name which is given here. I can also have it as, say for example, air temp comma, something like this

also I can define 2 variables. Any name I can give as long as these variables are previously defined.

So, you should have defined what this a and b are and they should be available in your workspace, only then you will be able to use in your create fits function, if they are not previously defined, then you will not be able to use them here and again you should see why I have given an underscore here. Matlab will not consider spaces between words, right. So, for that whenever you define a variable, If you want to give multiple names you can give, but then use an underscore and this entire thing will be considered as 1 variable by Matlab. So, this is how you use the function to fit any other data that you have.

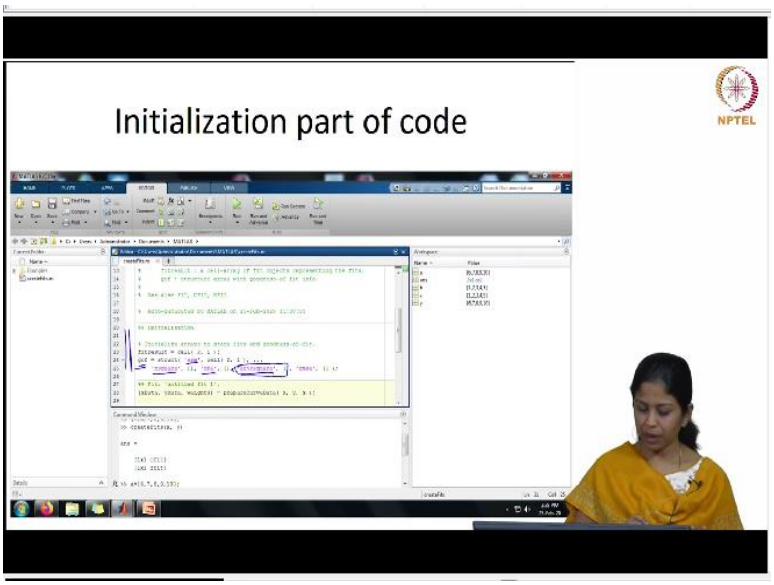
(Refer Slide Time: 07:58)



And you can see that we have now got a fit using for a and b, you can see that x is now 6, 7, 8, 9 10 because I have defined a as 6, 7, 8, 9 and 10 and I have specified a in terms of the x. So, I had create fits x, y is how the function is defined, I have used a, b. So, x will be a and y will be b for the fit, if I want it the other way, then I should be doing it as create fits b, a, then I will get the b values on the x axis and a values on the y axis.

So, Matlab does not consider anything automatically; it depends upon how we consider it right. So, this is how we use the function to generate some other fit.

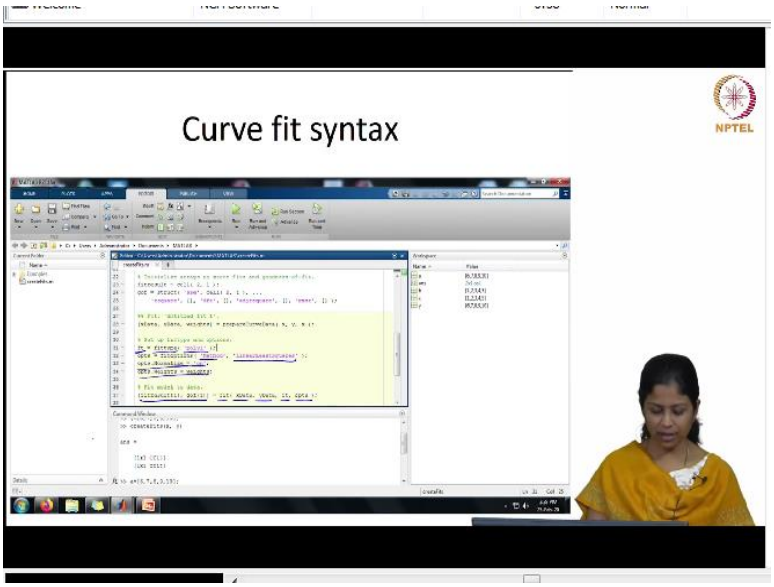
(Refer Slide Time: 08:55)



Now, let us look into the code as I mentioned earlier, I am not going to teach you how to write a code. So, this is just understanding the structure of this code and using it to modify some subtle aspects. So, the first one is the initialization of the code. So, it says it is fit result and then this is how it is defined, because we saw a lot of parameters, which were listed as tables here. So, it has to generate all these things.

And so, it is specifying all these parameters. So, which means that in your fit result it has to have all these parameters. So if you do not want adjusted R square, you just remove this part okay, like that you can do. So, this much is enough in the initialization part of the code.

(Refer Slide Time: 09:43)

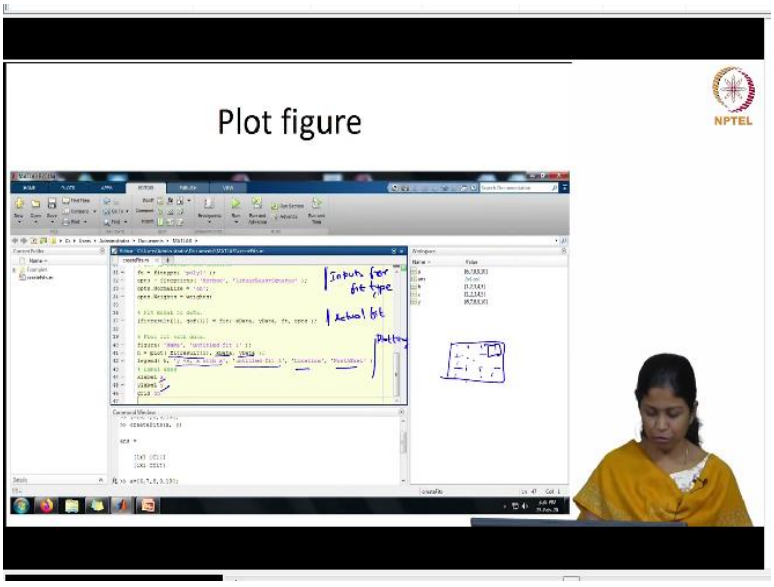


Next, we will move on to the actual fitting part. You can see this ft is defined as the fit type poly 1, poly 1 is the acronym which Matlab uses to fit a 1 degree polynomial. So if you go to the help which is available in Matlab, you can see that it is very user friendly. In fact, you can, you know, search for the options which are available in Matlab, and then you can try to create lots of fits, modify a lot of options which is used here.

So, this poly1 is a 1 degree polynomial, if you want some other standard form of it, then you can go and refer its acronym. And the options are, the method is linear least squares regression, okay, use this method to estimate the best fit parameters. And the options are normalize and you can give weights. So we will not worry about that now; since it is an auto generated code you have all these things.

You can also write your own code, wherein you can just finish it in 2 lines, right. And it has fit, the fit result is defined like this, you have the fit for which x data, y data, the fit and the options will be generated.

(Refer Slide Time: 10:57)

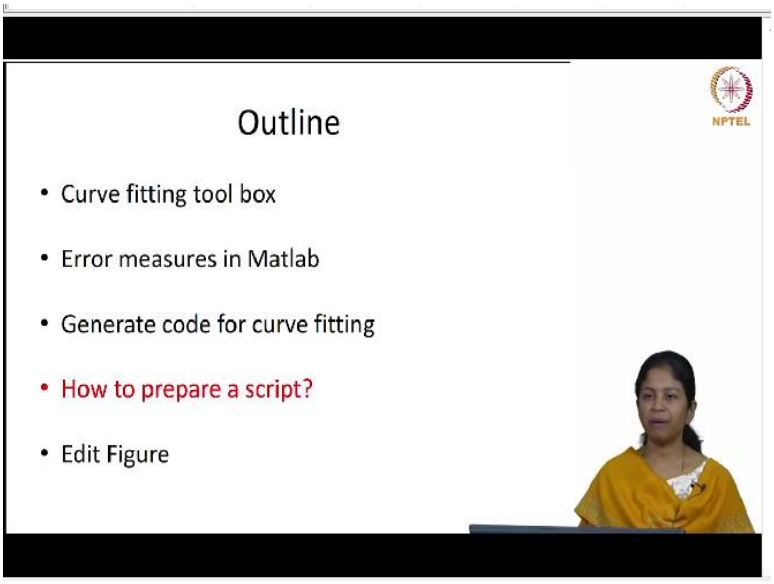


So this is the structure of the code and then you can see the next one is, so, here we have given the data, defined the type of fit, then we have actually done the fitting part, this is input for fit type, this is the actual fitting part and this is plotting, okay. So, in this case you can see you create a figure name and the name is given as untitled fit 1. Because that is how we have defined

it earlier. In the previous case when we use the curve fitting toolbox, if we give some other name, that name will be reflected here. Then you use h, plot it; so, we have fit results for x data and y data and then these are all the things which is given in the legend. You say that legend has y versus x with x and then the name untitled fit, the location if you see the figure, you will see that the figure is here the legend was here right. So, the location is north east. So that is also specified.

And label axis, by default, it is given as x and y only and we have this grid right, these boxes which are available in the background. So, that is called as grids and the grid is on here. So, we will be able to see the grid. So, these are the some basic terminologies which are used in this code.

(Refer Slide Time: 12:28)

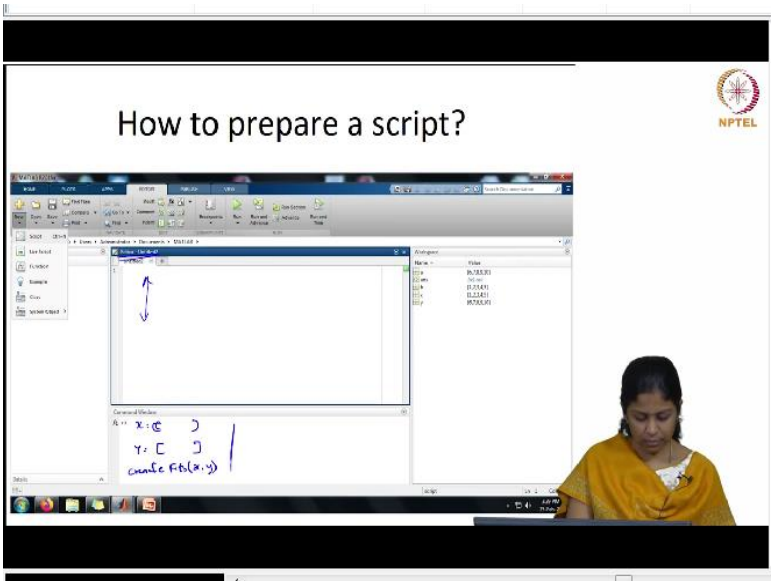


Now, how do we prepare a script? So, you know that there is a function which is available; you can do some specific step or methodology in your curve fitting toolbox. You can generate code for that, use it as a function, use it for fitting new data. Now, I do not want to do this because my curve fitting will not be one step, I will have data from somewhere else, I will do a lot of processing on that data.

And at one point I will fit the model to the data, so my curve fitting is only a part of it, right. So in that case, I do not want to write this every time or I do not want to you know, write all the

steps involved here every time. So, what I do here is I prepare a script and then I just run it, it will do all the steps which are listed there. So, how to prepare a script?

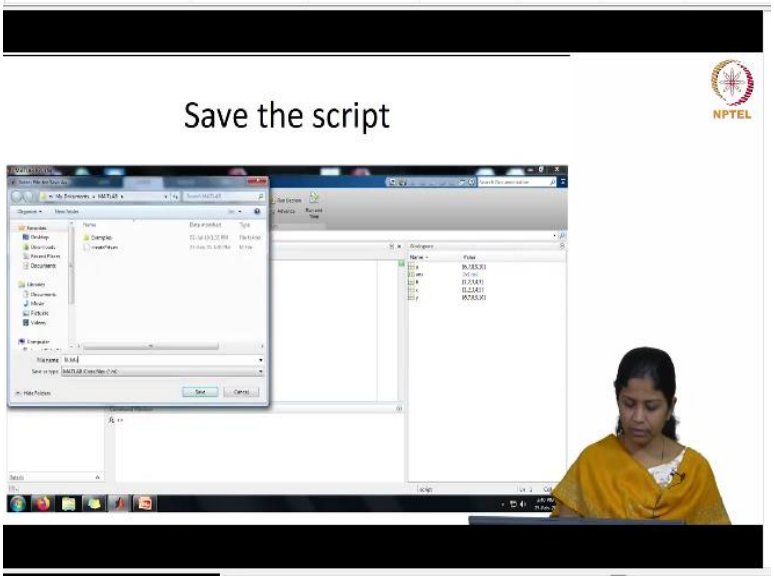
(Refer Slide Time: 13:22)



So, previously, what we are done if you have to create fits, you define x is all these things, right, and then y as all these things in the command window, and then you have to write create fits x, y, then it will do it. But this command window, it will not be saved, right. It is just running in time. So what I do here is I create an editor, I give all these steps here, and then I will be able to save it. So I do not have to define my variables every time.

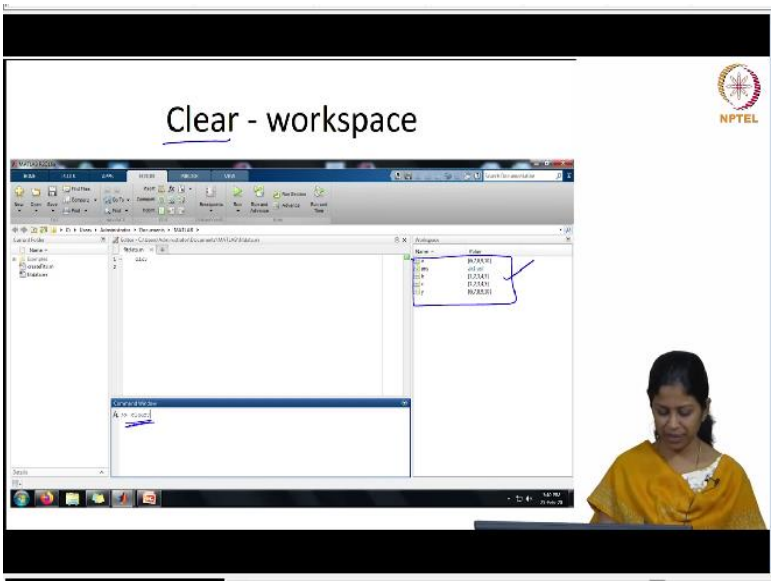
And I do not have to write all these things every time I want to fit for a data. So, how to create a script. So, you go to a new, you give as a new script, and then you will get your editor here. So, then you have your editor, you can write all your commands here.

(Refer Slide Time: 14:15)



So, this editor I am saving it as fit data. So, this particular file I am saving it in the name of fit data, already you can see the create fits function is saved as create fits dot m as a Matlab file right. You will be able to open this only using a Matlab tool. So, I just click save and then this is now saved.

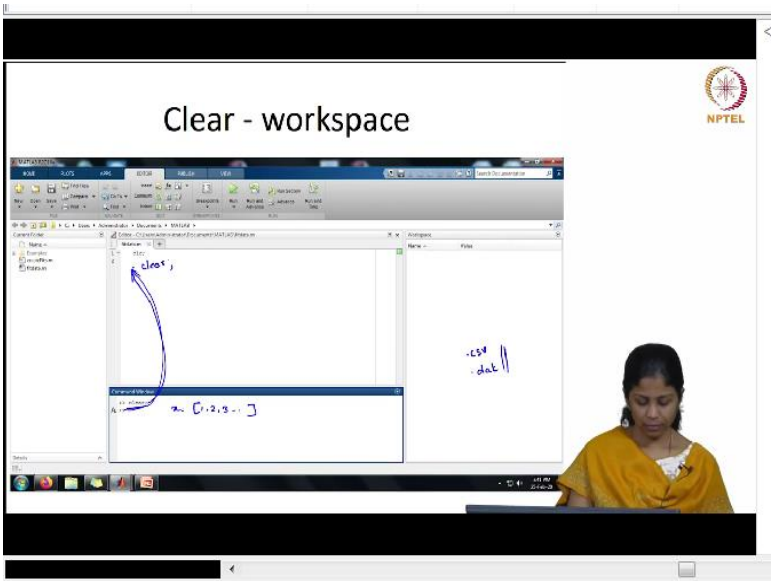
(Refer Slide Time: 14:43)



Now, I move on to creating my script right. The first one I write is clc. So this clc will clear whatever is there in your command window, right. So this is only optional, you can give or you can ignore this also, but generally it is a good practice to clear your workspace and command window before you start your particular code, right. So you can always give clc as the first step. Next one is the clear.

So when you give this clear option, it clears all the variables in your workspace. Maybe previously, I was trying to do something; I did a lot of random things. So I had a lot of variables generated in my workspace. Now I want to start something new, I want to clear off all these variables, then you type clear here. So you can see I have typed clear in my command window. So I have previously have all these variables here. Once I type clear in my command window, you can see that all these variables are gone.

(Refer Slide Time: 15:44)

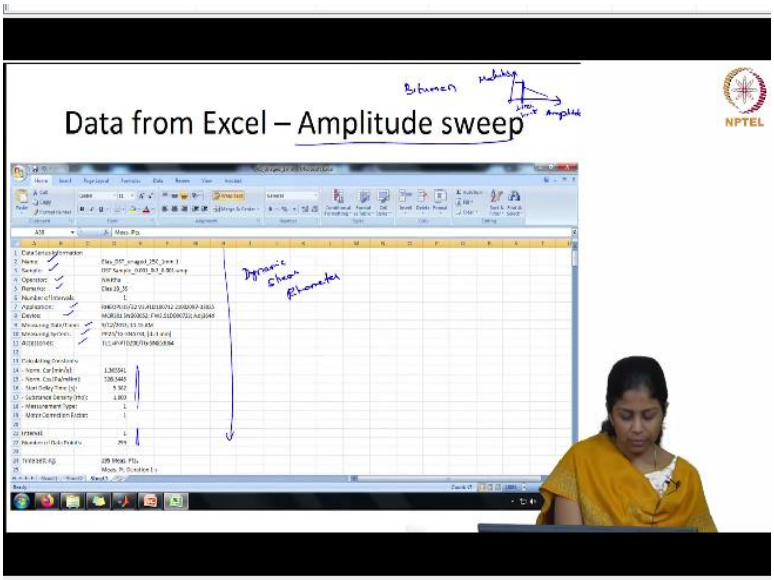


So we write this clear here in our script, so that it will clear the workspace every time you start doing something, right. So we have this clc and clear here right. Now we have to first import our data, only after we import the data and we will tell Matlab what to do with the data right. So, mostly we will not be having data which you can write something like this right. So, it is possible for an illustration.

But when you have experimental data which is voluminous, then it will not be possible to write it manually. So, you need Matlab to read the data and then extract whatever you want from the data so that you can use it for subsequent analysis. So, I will show you an illustration how to import data from excel, because excel is, these days, most of the equipment generate data in the form of excel because using Matlab or otherwise, it is easy to process.

But sometimes we get it as csv files, something like dot csv or dot dat. All these extensions are also possible and Matlab can read all these data types right. So for this course we will just contain to importing data from excel.

(Refer Slide Time: 17:05)



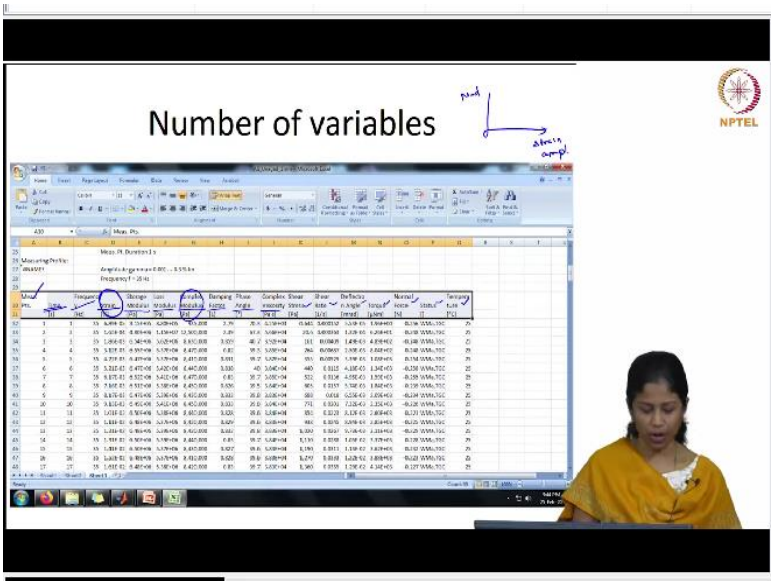
So, I have shown you a data for bitumen wherein an amplitude sweep is performed. So, what is this amplitude sweep? we have this strain amplitude, measure the modulus, we see how the modulus varies with strain amplitude. So, what is expected is maybe it will remain something like this in the linear regime and then it will start dropping off. This is what you had seen earlier also when you were discussing some basics related to binder rheology.

So, this portion where the modulus begins to show a dependency with the strain amplitude, this is called as the linear limit right. So now we know that the variation of modulus with amplitude is almost linear for other regions. So I am taking this particular data, I am trying to use the same fit, what we have defined earlier using the function create fits. So, this is all information regarding your data.

Again, when we conduct an experiment, the data, when it is generated will contain all necessary basic information regarding the equipment and the test what we are conducting here. So this is a result from a dynamic shear rheometer. You can see the name of the sample, right and what is the sample, operator, remarks, the number of intervals, the application, the device for which you

are using, measuring time, measuring system, accessories and your constants which are used here, the data collection interval, so everything is listed previously, right. We do not want all this data when we are doing the analysis, we just want the numbers right.

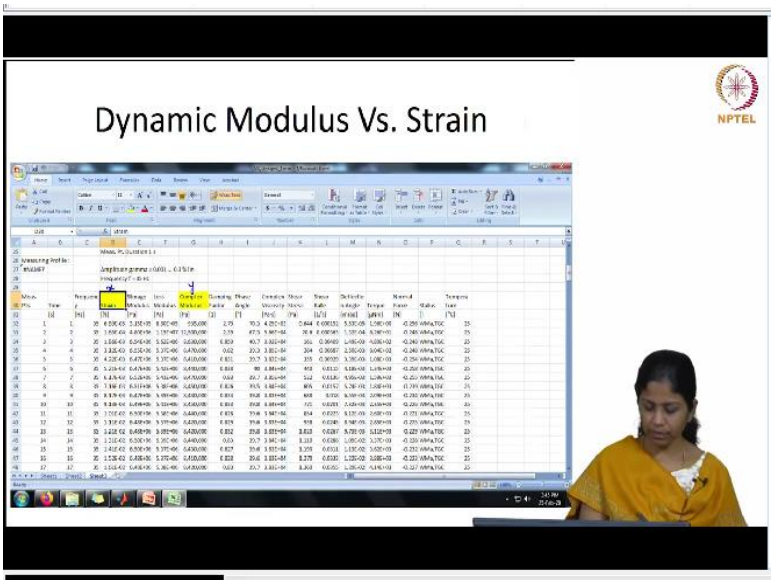
(Refer Slide Time: 19:06)



So, now we see that a lot of variables are here, we have measuring points, again you can specify what are all the variables you want from your dynamic shear rheometer. It will collect a number of variables, but you can choose what you want to tabulate and then you know export after you have finished your experiments. So, the first one is measuring points, the time, frequency, strain amplitude, your storage modulus, loss modulus, complex modulus, damping factor, phase angle, complex viscosity, shear stress, shear rate, deflection angle, torque, normal force, status and temperature.

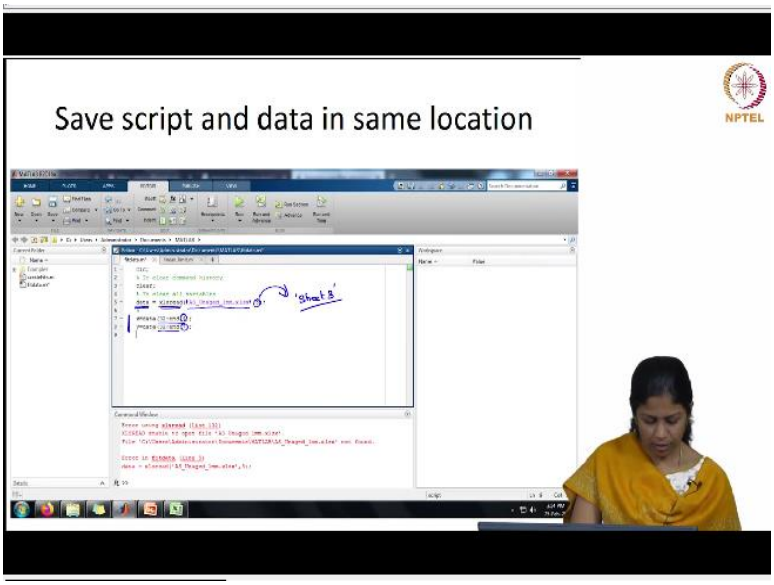
So, these are some of the variables that you will get as output from your DSR. And what we are interested here is just to generate a plot between your modulus right and your strain amplitude right. So, I do not want any other variable from this. I am interested only in the strain amplitude and the modulus which is shown here.

(Refer Slide Time: 20:17)



So, if you see, these are my interest, I do not want any other data. Now, I want to extract only this data, my strain variable is strain amplitude is x and my modulus is y. So, I want only this x and y and I want to fit it using the function.

(Refer Slide Time: 20:35)

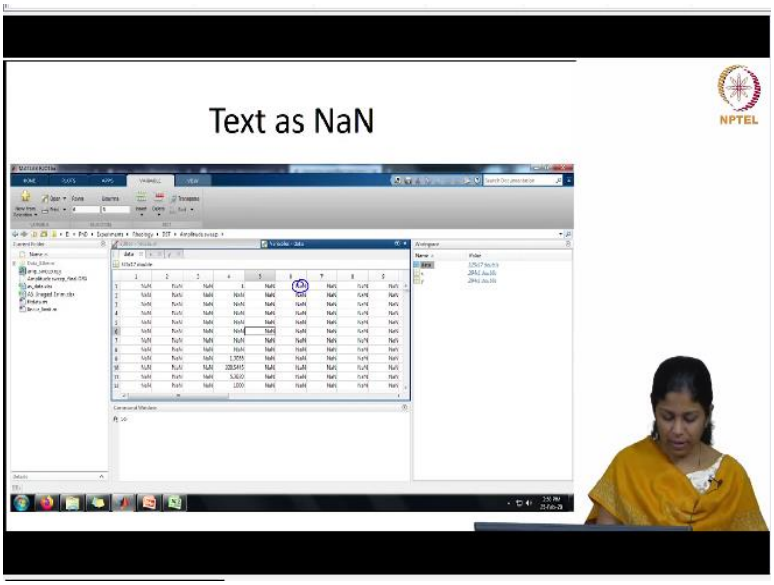


So, what you do here is, we write a code right, this data is a variable. This data can be anything x y, p, q whatever it is. So this is a variable right. So, this data is equal to xls read, to read an excel file, we have to specify it as xls read, you provide the file name exactly within a single quote. This is very, very important, you have to specify the name of the file with extension. So this is AS Unaged 1mm.

You can see the previous slide. See it is AS underscore Unaged underscore 1mm right. So, the same name I have to give with an extension xlsx and the sheet number. Previously, when you see, we were analyzing the sheet in sheet 3, right. So, you can either give sheet number, so that is 3. So the first sheet is 1, the second is 2, third is 3. That is how Matlab takes, but if you want to specify the name, you can also give it as sheet 3 within quotes like this after, in place of this 3; you can also write it this way alright.

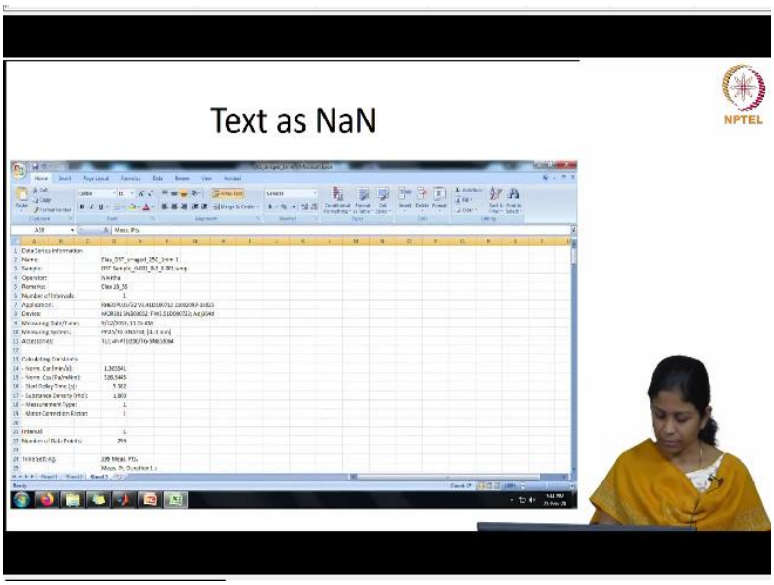
So that Matlab will open a sheet with the name sheet 3, or else it will automatically open the third sheet, whatever be it right it will automatically open that. So, that you have to specify correctly. So, then we specify what are the x and y values. So, the x is data from 32nd point till the end and y is data 32nd point till the end and this is fourth column and this is seventh column.

(Refer Slide Time: 22:19)



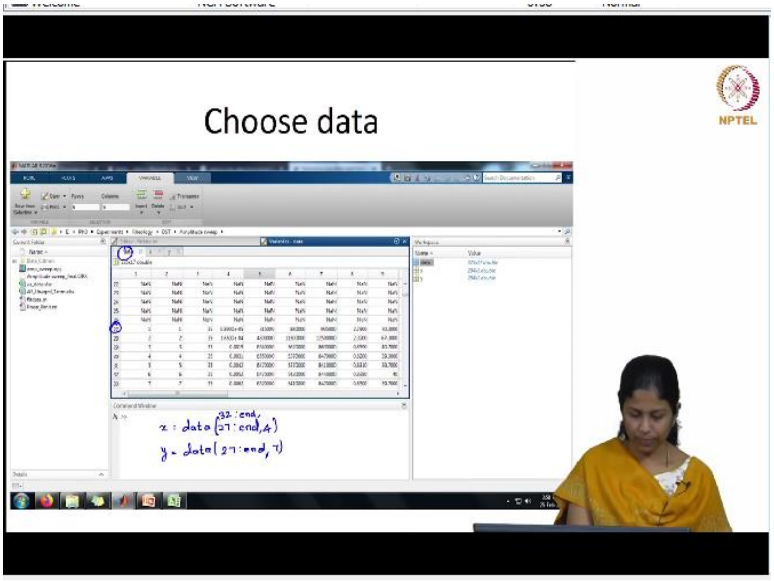
You can see, this is how your data is defined here, right, all the information right previously I was telling you it had the name, operator and other information regarding the equipment and the tests; all that is not a number. So, when we define an array, it will only take the numerical value. So, whatever is not a number it says as NAN, right, we do not want all these things, we only want whatever is the experimental data with numbers.

(Refer Slide Time: 22:53)



So, you can see whatever was given here right this information is only shown as NAN here. So, we want to ignore all these things.

(Refer Slide Time: 23:04)



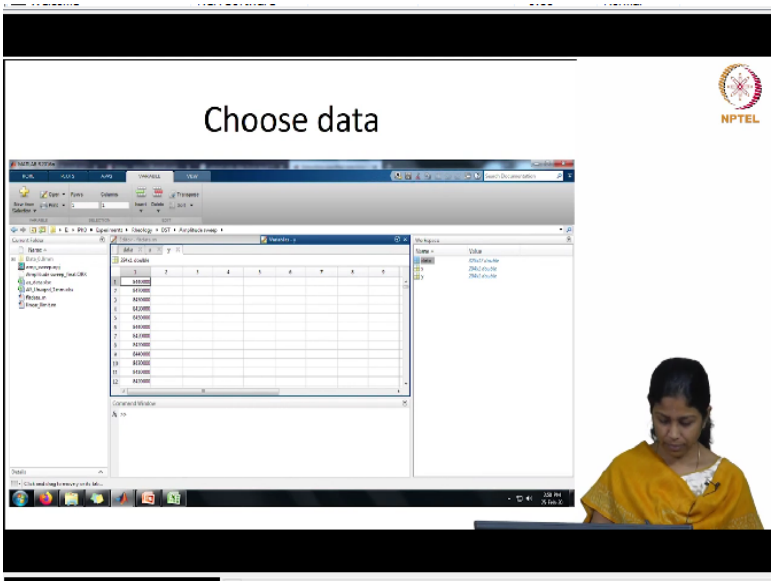
And you can see the data starts from 27th data point right. So, this is where your experimental data starts. So, I want whatever is the data that is present here from my 27th row. So, then what I will do here is, I will write, so this name of the variable this data right x is data of 27 till end. So, it starts this 2 is your row number. So, I can give from 1 to 5 or whatever is it. So, all these are NAN. So, we are going to ignore it.

We are going to start only from 27 till the end, how many ever data points are there because you might have some 1000 data points or 1000 230 data points. You do not have to go count every time how many data points are available. You can just give end, it will take till the end right, if I want only from a 32nd data point for whatever reason, maybe my initial 5 points are some trial points, I do not want, I want to ignore them, I will give it as from 32 to end right.

So, you can choose comma, this is the row right. So, we have defined the row and the second one is column. So, when you see the previous case, you saw that strain is defined in the fourth column. Yeah, see, first, second, third and fourth. So strain is in the fourth column, complex modulus is 7, right. So, x is in the fourth column, y is in the seventh column, so that I need to specify. So here, x is in the fourth column, right.

So similarly, I give y is equal to data of 27 till end comma 7 right. So, once I give this, it will automatically extract your x and y alone.

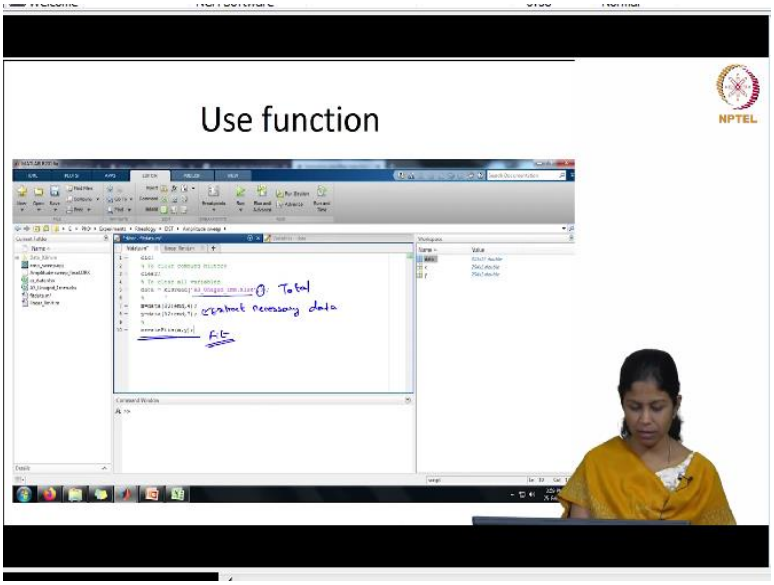
(Refer Slide Time: 25:07)



So, this is like I have already given y is 32 to end and this is x is fourth one and y is the seventh column. So, that is already defined here and you can see it is extracted; here also x and y are generated here. So, you can see x the values are given here and for y you will see the values are given here, right. So, we have now from the total data extracted our x and y, then we can proceed with fitting.

So, this extraction right, to select x and y, if you have some ten experimental data, then you have to do it individually for every set of data. So, when you prepare this script, you just give all these things, you just keep changing the sheet numbers or your file name, it will automatically do all the subsequent actions right. So, that is advantage of using a script.

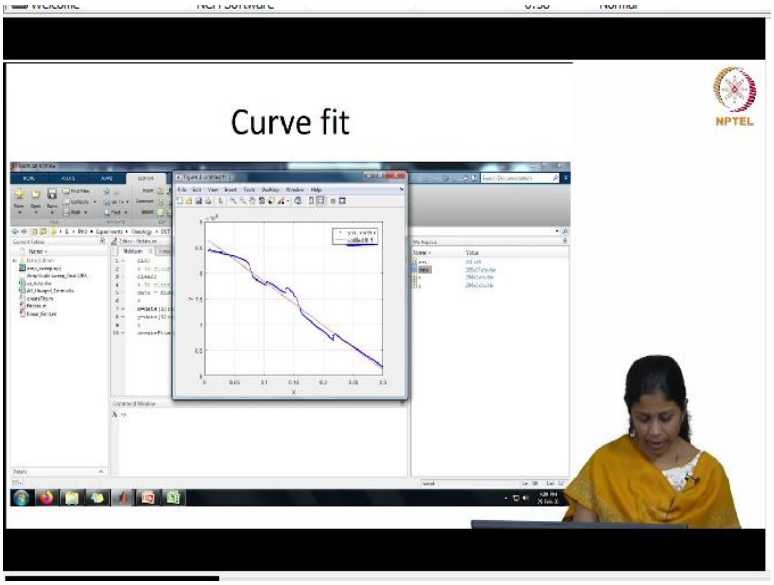
(Refer Slide Time: 26:03)



So then we just give create fits x and y that is all because there are only 4 steps, this to clear the command history, this is to clear all the variables, we are now reading the data. So, calling the xls file and then reading the file named so and so, and taking the data from sheet 3. So, this is calling the data. Then from the total data so, your total data will be displayed here, from the total data you have to extract the necessary data.

That is what we have done here and then we are trying to fit right. So, in this create fits function whatever was there everything will be carried out when you call this function create fits right. So, this is the actual fitting part.

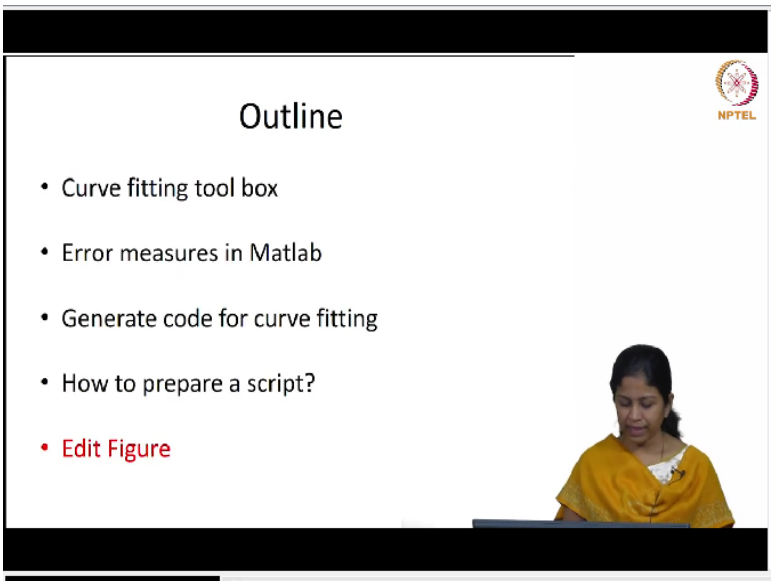
(Refer Slide Time: 26:58)



So once you give this you will see that it automatically fits a 1 degree polynomial for your x and y data right. So, this is my the blue one is my actual data and the red one is the fit and it has named the fit, like how we have defined it in the function file. So, if you want to change the name you can go change it here or every time when you fit also you will be able to change the name. I am not getting into all those details.

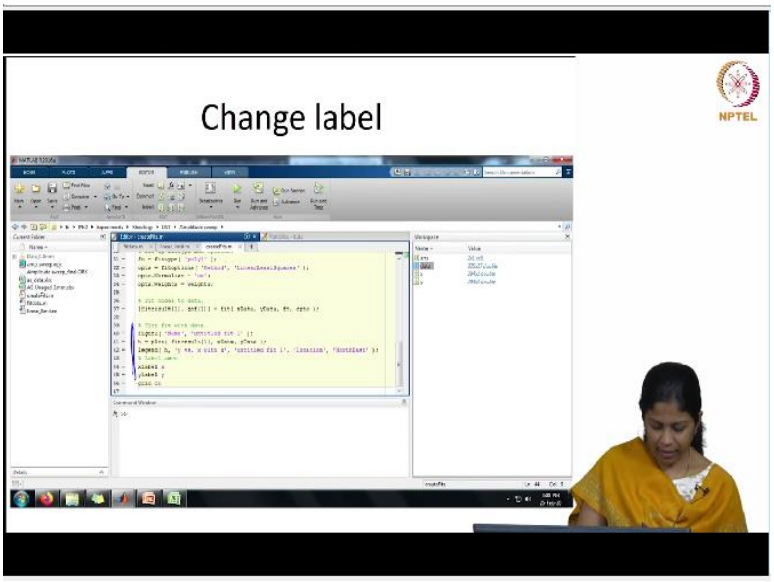
Similarly, when you read data from excel, there are also a lot of other options wherein you can just read ignore all these NANs and only read what is the numeric data. So again for that you need to write some 3 or 4 lines and to you know explain that I need you to I need to explain lots of other parameters also. So that is why I have not explained that also. Whatever I have shown here is may not be the optimistic way, but it is a simplistic way of doing this.

(Refer Slide Time: 27:54)



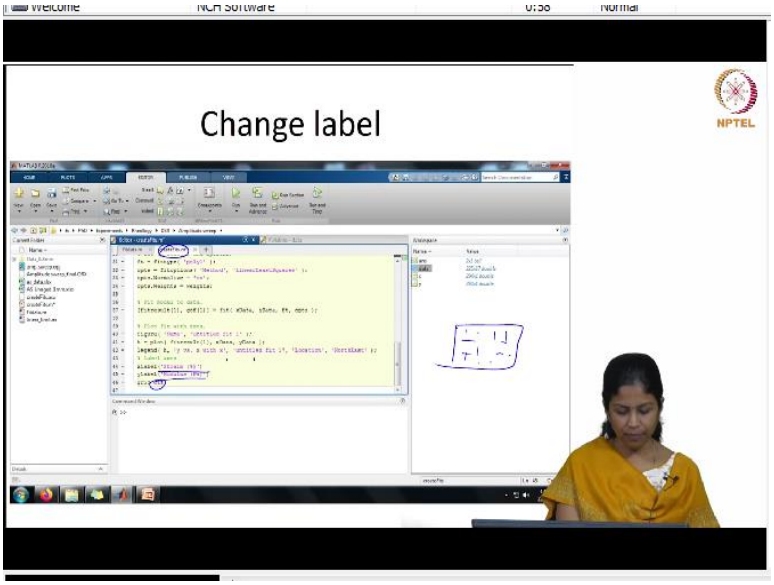
Then the final step is how to edit your figures.

(Refer Slide Time: 28:00)



So, we have defined the plot in the function file right. So you can go to the function file and change them some things so that every time when the plot is generated all these changes will be incorporated.

(Refer Slide Time: 28:18)

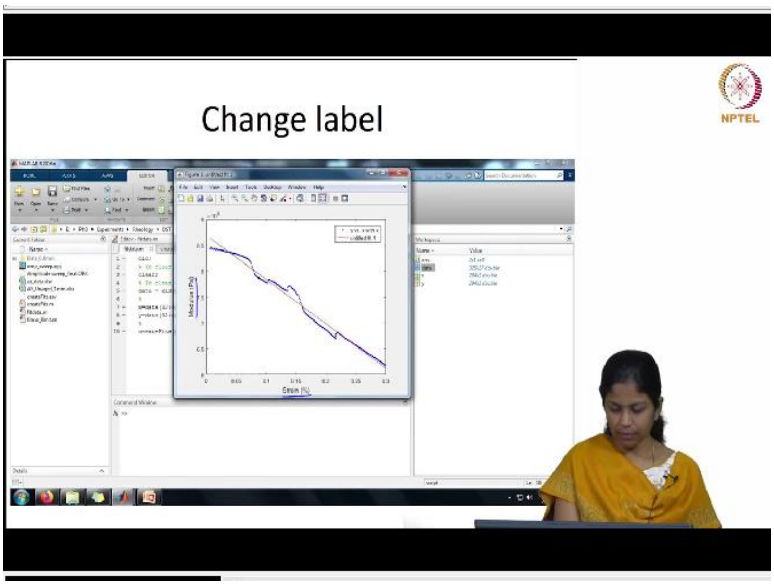


So we will see here the first one is how to change your x label because previously every fit which was generated here right. So had only x and y, so we do not know what this x is, what this y is. So when the fit is generated, you want Matlab to say what x is and what y is. So, for that we have this option here x label. So, you can see previously it was x and it was y. So Matlab printed it as x and y.

Now we will go to this and change the x as strain in percentage and y as modulus in Pascal. So whenever you write a text or a string, you have to put it within single quotes. So in this function, so I am doing this in the create fits function file. So you go to this create fits function file and then you change x label and y label to whatever name we want. So here we are giving it as strain in percentage for x label and modulus in Pascal for your y label.

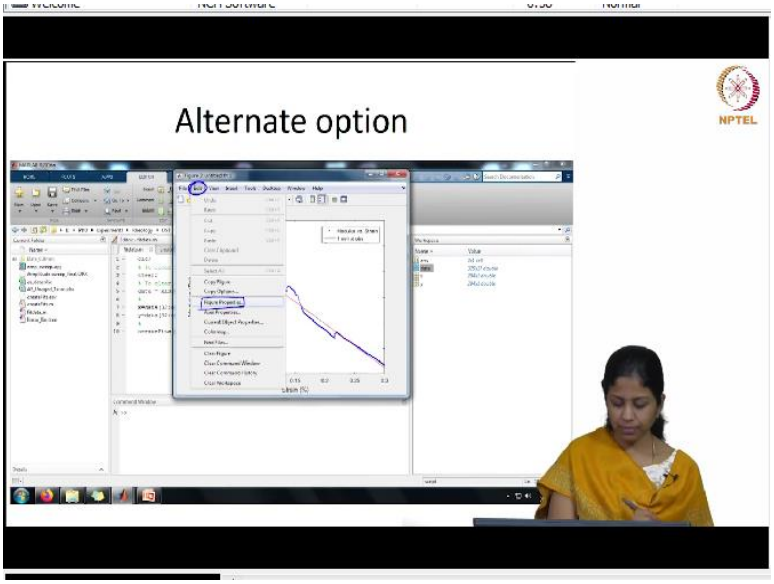
And I do not want those you know gridlines, which are there in the background, right. I do not want them. So I want to switch off those grid lines. So here grid, I give it as off right.

(Refer Slide Time: 29:27)



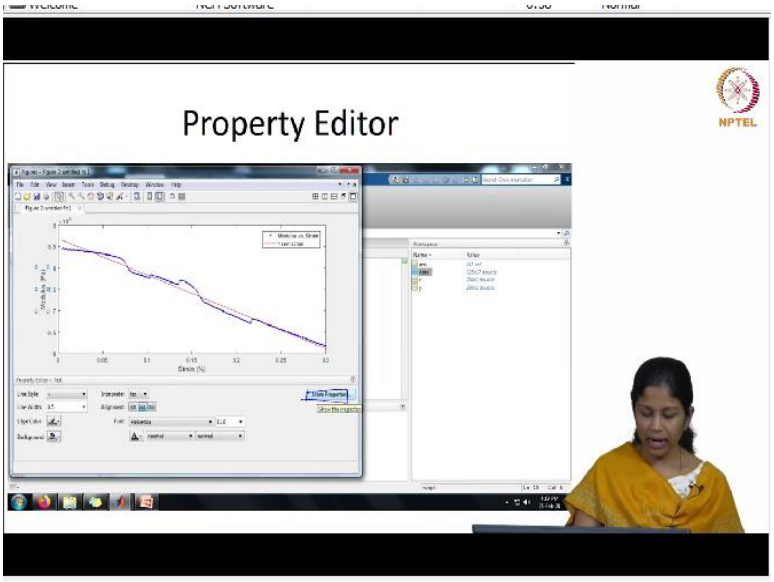
So once I do all these things, my plot is generated, see, the gridlines are gone. The x label and the y label are also changed now.

(Refer Slide Time: 29:36)



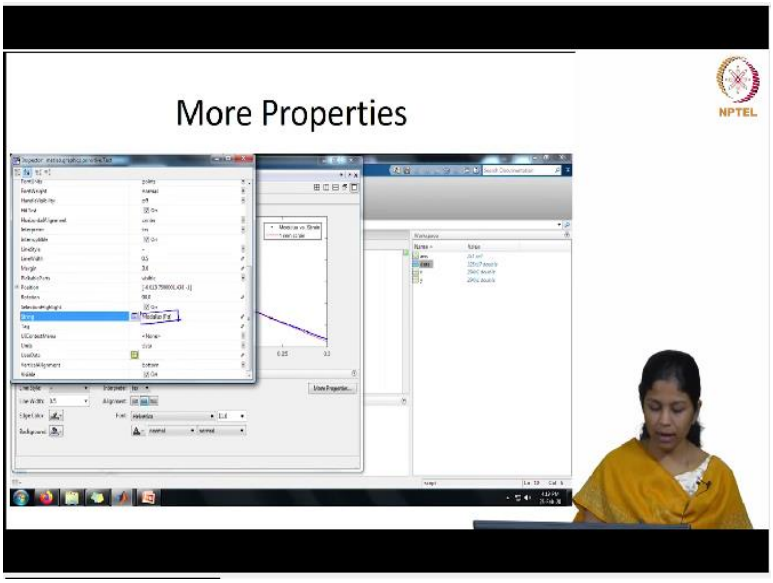
Alternatively, instead of going to code and then changing it, we can also do it in the toolbar, edit properties in your figure toolbar. So when you click this edit in your figure toolbar, you have lot of options. You just click figure properties, right.

(Refer Slide Time: 29:58)



When you click this, you will get into the property editor. So, you can see that below that the property editor is displayed, you can change line style, you can change line width, the color, alignment, lot of things are here.

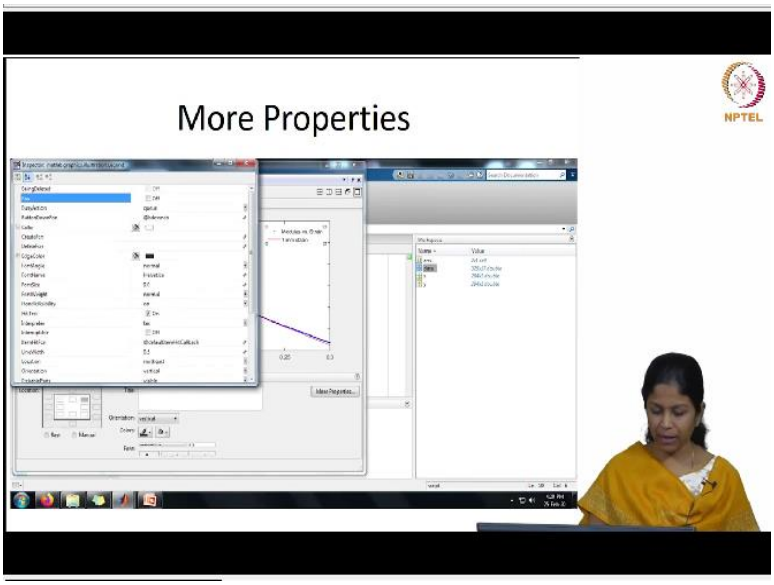
(Refer Slide Time: 30:13)



And when you click more properties, you will see a number of properties are now displayed right, you can see you can there are lot of parameters which you can change and you can go through it one by one. So, you can modify this plot however you want. And all of it can be done using codes also. But for beginners, you can use these options which are here, maybe generate a code and then use that in your function code so that you will get figures customized in the manner in which you want.

So here also see instead of changing it in the code, I can also go here when I have this string right, modulus in Pascal, I can also change the name of my variable. So, all these things are possible. So, it has actually a lot of parameters and you can customize it in any manner you want.

(Refer Slide Time: 31:00)



Similarly, I can switch off, this box here, right. Previously, we had a box here around the legend, I want to switch off the box. So you go to box, click this, uncheck this off, and then this box will be gone. There are a lot of options. You can go through it if you are interested. But basically we have generated a plot with x and y and we have labeled them. So I will stop my lecture on this introduction to Matlab here.

Dr. Padmarekha will continue few lectures down, and she will explain you how to use data collected specifically for bitumen and fit some models something like a Burgers model for a creep and recovery data. So she will show you a lot of things how to do curve fitting using Matlab. So, in that case, all these functions and writing a script whatever we have discussed so far will come in handy.